Multi-Model Semantic Interaction for Text Analytics

Lauren Bradel, Chris North, Leanna House, Scotland Leman

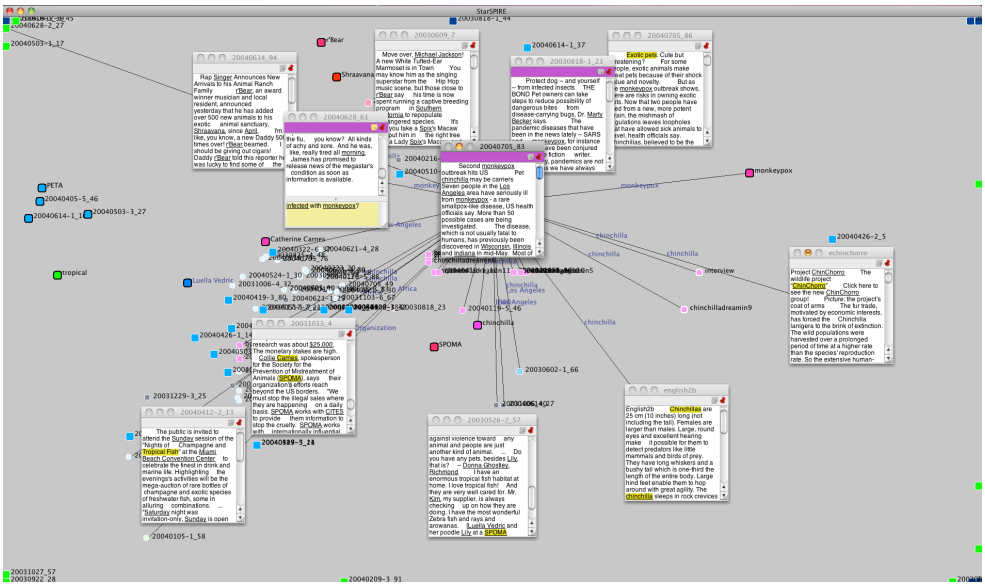

Fig. 1. StarSPIRE spatial workspace showing clusters of open documents and numerous iconified documents selected and arranged through semantic interaction.

Abstract— Semantic interaction offers an intuitive communication mechanism between human users and complex statistical models. By shielding the users from manipulating model parameters, they focus instead on directly manipulating the spatialization, thus remaining in their cognitive zone. However, this technique is not inherently scalable past hundreds of text documents. To remedy this, we present the concept of multi-model semantic interaction, where semantic interactions can be used to steer multiple models at multiple levels of data scale, enabling users to tackle larger data problems. We also present an updated visualization pipeline model for generalized multi-model semantic interaction. To demonstrate multi-model semantic interaction, we introduce StarSPIRE, a visual text analytics prototype that transforms user interactions on documents into both small-scale display layout updates as well as large-scale relevancy-based document selection.

Index Terms— Visual analytics, Semantic Interaction, Sensemaking, Text Analytics.

1 Introduction

The problem of "too much data" has become a significant challenge in unstructured text sensemaking. Analysts are expected to "connect the dots" across many documents [19], requiring analysts to work across multiple models to manage different portions of the sensemaking loop [28].

During foraging, analysts work at the large scale (beyond data displayed on the screen). Because the number of documents available far outweighs the number of relevant documents (e.g. millions of documents with hundreds or fewer relevant documents), the low signal-to-noise ratio makes this a "needle in a haystack" problem. Thus, analysts need methods of honing in on and finding additional relevant documents. Additionally, analysts must find *all* of the relevant documents in order to avoid missing important pieces of information. Relevance models are helpful at this scale.

During synthesis, analysts work at the small scale (e.g. the amount of data that comfortably fits onto a display) with hundreds or fewer documents. A common synthesis strategy is to spatially organize information on the display [1]. Spatialization and dimensionality reduction models are helpful at this scale [10]. The analyst then performs synthesis on these documents to make sense of them, but may have need for additional information.

Thus, the sensemaking process consists of continuous iteration between foraging and synthesis, using multiple models to accomplish different sensemaking-related tasks. However, current tools require the analyst to break from synthesis actions to forage for additional information, which interrupts their cognitive processes.

We propose unifying the sensemaking loop by coupling synthesis with foraging, and therefore coupling the corresponding models and interactions, resulting in a multi-model approach. In other words, synthesis activities can be interpreted to forage for additional relevant information and filter out irrelevant data. Likewise, foraging activities can influence synthesized structure. To accomplish this, a method of usable control over coupled models is needed.

Models which support computing data relevance (foraging) and spatial layout (synthesis) typically require parametric interaction, but most analysts are not experts in these underlying models and are ill-equipped to interact directly with the parameters. Instead, semantic interaction (SI) techniques convert user interactions within a spatialization into parametric feedback, enabling a spatialization that

• *Lauren Bradel, Chris North, Leanna House, and Scotland Leman are with Virginia Tech. E-mail: [lbradel1, north, lhouse, leman]@vt.edu*

Manuscript received 31 March 2014; accepted 1 August 2014; posted online 13 October 2014; mailed on 4 October 2014.

For information on obtaining reprints of this article, please send *e-mail to: tvcg@computer.org*.

is jointly created by user and algorithm [12, 14]. These techniques shield the user from the complexity of underlying spatialization algorithms and allow them to focus on data analysis. However, semantic interaction has been limited to steering a single underlying model with fewer than 1000 data points.

Our goal is to generalize semantic interaction to simultaneously steer multiple models. This involves new challenges in mapping semantic interactions to multiple model parameters in a coordinated way and conveying combined model output via visual feedback. Specifically, we instantiate this for the purpose of leveraging models at different levels of data scale to support larger datasets. In our method, users invoke semantic interaction techniques in order to incrementally adjust a spatial layout model as well as influence what information is presented to them via a relevancy model.

We present three contributions: (1) The concept, named Multi-Model Semantic Interaction (MSI), is an alternative to explicitly controlling parameters in multiple models. (2) We formalize this extension in the form of an updated visualization pipeline that reflects the generalizability of semantic interaction to multiple models. (3) To demonstrate multi-scale semantic interaction, we present StarSPIRE [Figure 1], a visual analytics prototype implementing MSI for unstructured text data, which has been tested on datasets up to 10,000 text documents. We conclude with a discussion of multi-scale semantic interaction and research directions moving forward.

Table 1. Multiple levels of data scale and their associated models, visualizations, and feedback mechanisms.

Scale of Interaction	***Small***	***Large***
Sensemaking Loop	Synthesis	Foraging
Model purpose	Spatially project small scale data points onto the display, e.g. based on similarity	Extract useful data from large scale, e.g. based on relevance or coverage
Usage Description	System lays out displayed data, according to user's spatial organization feedback	System selects data to display based on relevance according to user's interests
Model	Dimensionality reduction	Relevance-based data selection
Model Parameters	Dimension weights	Dimensions weights
Model metrics	Similarity metric	Relevance metric
Visualization	Similarity mapped to visual proximity	Relevance mapped to working set, glyph size, and saturation
Interactive Feedback	Semantic interactions (see Table 2) update the dimension weights	

2 Related Work

Spatializations are frequently employed to aid sensemaking (foraging and synthesis) of unstructured text documents [2, 21, 30, 33, 34]. Large, high-resolution displays in particular have been found beneficial in affording a large, flexible workspace that allows users to externalize knowledge and create semantic schemas [1]. However, this knowledge externalization is typically achieved through parametric interactions (e.g. [22]), many of which require users to go outside the spatial metaphor by manipulating control panels [11]. Furthermore, parametric interaction does not easily scale to big data problems. In unstructured text data, dimensions map to the terms or entities contained in the documents. Thus, the dimensionality of the data grows extremely large as the number of documents increases. Aside from navigating through the flood of dimensions, altering multiple models becomes extremely tedious. If multiple models are used for layout and/or retrieval, the user must update the dimensional weights or parameters for each model. To remove this redundancy, we prefer to contain the interaction within the spatial metaphor and translate interactions into parametric feedback.

For tools that allow users to stay within the spatial metaphor, parametric interaction is still common. For example, Dust & Magnet allows users to manipulate spatial landmarks to adjust the spatialization of multi-variate data [35]. However, these landmarks are attributes of the data, not points themselves. The users only have control over the parameters in the space. Similarly, VIBE allows users to designate keywords as spatial landmarks [27]. In MSI, users can designate specific data points as spatial landmarks. These landmarks attract other data points (e.g. documents) based on the high-dimensional data instead of a single attribute or dimension.

Systems exist which allow users to directly manipulate data points, interpret this feedback via a dimensionality reduction model to generate a new spatialization that better reflect the user's understanding of the high-dimensional data [6, 14, 20]. These methods inherently suffer from scalability issues. Users expect a quick interaction-feedback loop in order to remain in their "cognitive zone" [16], but calculations on thousands, let alone millions, of data points take from minutes to hours to complete. It is more practical to perform dimensionality reduction on a subset of a much larger data set and use information retrieval techniques to retrieve additional information to add to the workspace.

MSI is perhaps most similar to adaptive query-by-example systems. These systems, such as Adaptive Information Retrieval [3], use relevance feedback to augment future retrieval requests to return results that are better tuned to the user(s). Attempts have been made to visualize information retrieval results (e.g. term distribution charts [18], self-organizing semantic maps [25]), but these techniques have not been widely adopted. Information retrieval results are typically visualized as a ranked list of results [26]. Presenting results in this format is suitable for targeted queries where the user may view a handful of results at most (e.g. a web search for a specific culinary recipe). However, when the user is presented with hundreds of viable documents worth reading (e.g. an intelligence analysis task) that relate in complicated, intricate, and fuzzy ways, a linear list becomes less than ideal [5].

Card presents a survey of visualization techniques for huge amounts of unstructured text data [7]. These techniques include, but are not limited to, dimensionality reduction (e.g. [33]), semantic maps (e.g. [25]), hierarchies (e.g. [4]), and link-node diagrams (e.g. [24]). We have chosen to explore dimensionality reduction techniques and link-node diagrams for representing unstructured text data, but we recognize the potential to explore other visual representations in the future.

Choo and Park provide an overview on scaling computational methods to the problem of big data [8]. In our research, we have chosen the data scale confinement solution. By constraining the visualized data to a subset of the actual dataset, dimensionality reduction calculations grow much more efficient than computing across the entire dataset. This motivates our multi-scale approach to sensemaking. After performing information retrieval requests on the entire data set to procure a subset, the subset can be run through a suitable spatial layout model.

We have developed multi-model semantic interaction in order to accommodate the need to work with extremely large amounts of data while staying within the spatial metaphor and interpreting interactions to manipulate multiple data models.

3 Semantic Interaction

Semantic interaction serves as means for analysts to work with data within a spatialization instead of altering algorithms or the raw data [Figure 2]. This is particularly important when the analyst is a non-expert in the layout model(s).

To develop semantic interaction, we first observed analysts, both novice and expert, completing sensemaking tasks and recorded the actions analysts undertook [1, 5, 13]. We then harnessed these

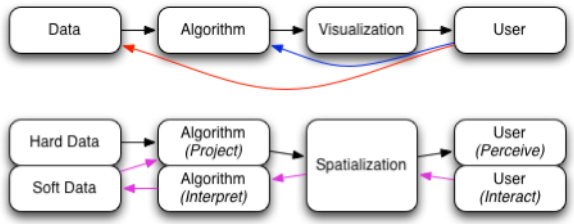

Fig. 2. Top: original visualization pipeline showing user interaction at the algorithmic and data levels Bottom: semantic interaction visualization pipeline showing user interaction within the spatialization, which is then interpreted by the model to extract parameters (stored in the system as "soft data"), which are used to update the spatialization.

actions such that the system could learn from the user which terms were important to them in their analysis, resulting in semantic interaction [12]. Previously, we have applied this technique to unstructured text data in a modified force-directed layout, allowing the semantic interactions to update the spatial layout, which used a "near = similar" metaphor. Alternatively, semantic interaction has been applied to additional dimensionality reduction models, namely Multi-Dimensional Scaling (MDS), Principle Component Analysis (PCA), and Generative Topographic Mapping (GTM) [14]. Semantic interaction has been practically applied to Multi-Dimensional Scaling using multivariate data, although the interactions were limited to moving and highlighting data points [20].

While current forms of semantic interactions have shown to be successful, they are limited in the number of data items they can handle simultaneously (less than 1000) and have been limited to steering a single model (spatial layout). Thus, semantic interaction alone is not adequate for tackling the challenge of big data.

4 Multi-Model Semantic Interaction

We addressed the scalability concern by developing a generalized semantic interaction pipeline where multiple models can be leveraged, providing functionality across multiple levels of data scale. The result of this pipeline is a spatialization with which the user can interact, externalizing their knowledge of the data. These interactions are then converted into parametric feedback in order to update the underlying model(s), and ultimately, update the spatial representation of the data to reflect these changes [**Fig. 3**].

Using [Table 1] as a guide for interaction and visualization at multiple levels of data scale, we see that small amounts of data map to dimensionality reduction models, while large amounts of data map to retrieval models. Using semantic interaction techniques, we seek to communicate with and between these various models in order to update the spatialization, select potentially relevant new information, and filter out irrelevant data.

At the large scale, semantic interactions are mapped to retrieval requests, which serve to constrain the amount of data piped into a display layout model by extracting a working set of relevant documents, which then creates a spatialization with which the user can interact. The interactions done within the spatialization are then interpreted to influence the layout and/or retrieval models. Thus, the user is able to work with multiple models working at multiple levels of scale through interactions done on the data in the spatialization.

For example, if a user executes a search for a term, documents containing this term in the spatial workspace would be drawn closer to the search node and the system would query the larger "behind the scenes" dataset for this term and add the top *n* retrieved documents that surpass a relevance threshold, ranked by the importance the user has given to entities. This is an incremental formalism approach [29] wherein the system considers the history of interactions to gradually construct and refine the user's interest model of the data. In addition to just retrieving documents, multi-scale semantic interaction augments the relevance model to tune the results to the user's interests.

In terms of the sensemaking loop [28], synthesis actions are used to drive foraging activities and many foraging activities are able to be conducted implicitly instead of explicitly. For example, as the user constructs a cluster by dragging documents together, the system can search the entire dataset for documents that are similar to the shared terms in the clustered documents and add them to the workspace. Foraging actions such as these that are conducted through implicit means allow for a richer and more nuanced query than explicit actions. For example, an explicit search for additional documents may take the form of a boolean search. An implicitly constructed query could go beyond boolean values to indicate the relative importance of terms as well as include a far greater number of terms than the user is likely to enter. This method of implicit query formation attempts to return semantically relevant information to the user and seeks to fill in gaps of knowledge that a strict boolean search might miss.

In addition to bringing information into the spatial workspace, multi-model semantic interaction also filters out irrelevant information. If a user indicates that a document or term is uninteresting or not relevant to their current investigation, the system will interpret this interaction to update the user's interest model parameters to reflect this. Accordingly, information related to this document or term would be filtered or removed from the display and would be less likely to be returned from information retrieval requests.

Multi-model semantic interaction conveys the output of the multiple models through visual encodings to convey document relevance and relationships between documents. This serves to give the user immediate visual feedback regarding their interactions.

4.1 Updated Visualization Pipeline

We present an updated visualization pipeline to reflect multi-scale semantic interaction [Figure 3]. The initial spatialization is constructed by taking the data, or a working set of the data as determined by a relevance model, and passing it through a display layout model. The user then perceives the spatialization and has the option of interacting with the data within the spatial metaphor. All interactions are interpreted and directed to the appropriate inverted model(s). The inverted models then are combined, if necessary, and the new parameters are stored in the user's high dimensional model

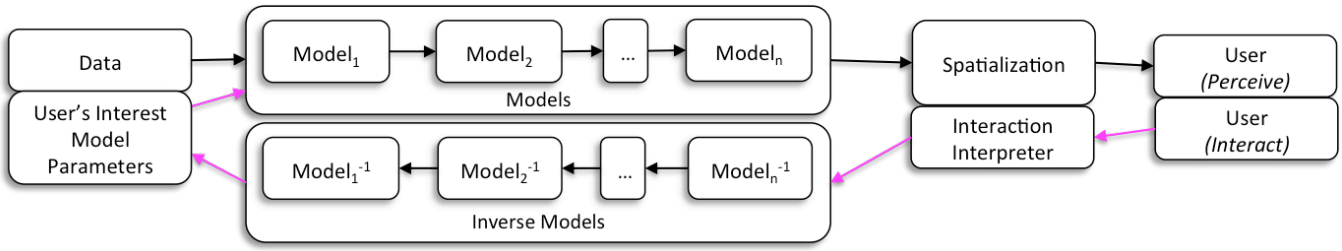

Fig. 3. Generalized multi-scale semantic interaction visualization pipeline. Any number of models can be inserted for use in this pipeline. Once the user perceives the spatialization, they can choose to interact in it. This interaction feedback is interpreted as input to one or many inverted models. The updated model parameters are stored, which are then used, along with the original data, to create an updated spatialization.

of the data. This high dimensional model is then coupled with the dataset to pass through the retrieval and projection portion of the loop, resulting in an updated spatialization. This pipeline currently assumes a single shared set of model parameters. Possible extensions of this pipeline include multiple user models for the data (e.g. the user believes the data should be arranged in a different manner than what the user believes should be displayed).

Not all semantic interactions will necessarily influence every model or have the same impact. We offer a few examples to illustrate this point. Highlighting a phrase in a document typically indicates its importance, while minimizing a document when space is not constricted typically indicates the unimportance of its contents. Moving points around the display would naturally update the display layout, but would not necessarily fetch new data points for the workspace.

Furthermore, updates to the underlying models should be executed wisely. Updating a model that impacts the entirety of the data set will likely be a slow operation, whereas a display layout model operating on a small subset of the data can be executed much quicker. Therefore, it is practical to update the display layout model with each semantic interaction, but it may not be practical to do so for the information retrieval model. Obviously, if a user explicitly queries for information, it should be returned promptly. Otherwise, it may be a better option to check for new potentially relevant information and/or update the underlying model every *n* interactions.

5 StarSPIRE

StarSPIRE (Semantic Translation of Actions for Retrieval – Spatial Paradigm for Information Retrieval and Exploration) is a visual analytics tool prototype that implements multi-model semantic interaction techniques using two models (relevancy and display layout) [Figure 4]. StarSPIRE is built upon the foundation of ForceSPIRE, a semantic interaction visual analytics tool prototype for exploring unstructured text documents [12]. StarSPIRE and ForceSPIRE share a flexible spatial workspace (driven by a modified force-directed layout [12, 15]) and several semantic interactions. This system extends upon previous work to integrate relevance-based retrieval and layout models, provides richer visual encodings, and adds to the semantic interactions leveraged. StarSPIRE dynamically adjusts how many data points are displayed by using heuristic-based relevance metrics. While its predecessor was designed specifically for use on large, high-resolution displays, the push-and-pull nature of displayed data in StarSPIRE has made it usable regardless of display size.

5.1 Visual Encodings

Within the spatial workspace, document nodes are visually encoded to relate their relevance to the user's high dimensional understanding of the data [Figure 5]. Node size and saturation are encoded to reflect how closely a document matches the entities the user has deemed important. Node size and saturation are calculated by summing all of the entity weights in a document, ranking these values, and sorting them into quartiles. Quartiles were chosen instead of absolute ranking to optimize the node drawing process, minimizing the number of calculations and changes required with each user interaction. This was done to promote a quick interaction-feedback loop.

These encodings give the illusion of a third dimension in the workspace where more important documents are in the foreground while less important documents fade into the background. However, unlike a true three-dimensional layout, document nodes cannot overlap each other, preventing occlusion.

Additionally, StarSPIRE provides visual cues for navigating the workspace. Node color is used to indicate search term matches. Instead of showing all links between all documents, StarSPIRE restricts the edges shown to those connected to the selected node. Entities shared between documents are labelled on the edge, but are restricted to the top four entities, determined by their importance weights. All nodes are labelled with their document's titles in order to allow for easier navigation in the space and to allow users to track a specific node's movement throughout the space. Each node's outline color is used to denote its read or unread status in order to allow analysts to see which documents they have read and closed. Within each document, search terms are identified and the text color is changed to allow the terms to stand out for easier identification. These encodings were identified and/or adjusted through an informal usability requirements analysis of StarSPIRE.

5.2 Interactions

StarSPIRE begins with a blank spatial workspace with documents loaded into memory. The user then executes a search to add documents to the workspace. This grants the user flexibility for where to start their analysis and mimics an analyst executing a database search to return a set of documents with which to begin their analysis. Granted, this supported use case assumes that the analyst is conducting a directed sensemaking task. This does not support the use case of being handed a stack of documents and told to "see if there is anything suspicious." In this scenario, other methods, such as topic modelling, would be useful to aid the analyst in finding a starting point for their analysis.

Documents are laid out using a modified force-directed layout where the spring attractive force between two nodes is determined by summing the weights of shared entities. Thus, the layout's input is the displayed data for the current timestep and the weight vector for the previous timestep. The weight vector is determined by interpreting user interactions [Table 2]. The set of displayed documents is determined from a document relevance model.

Users can then interact with the data to incrementally formalize their understanding of the data. These interactions include moving nodes, pinning nodes to create spatial landmarks, resizing nodes, collapsing open nodes, annotating documents, searching for terms, highlighting terms, and linking document nodes. With each interaction, the display layout updates to allow nodes to move about the space to reflect the new entity-weighting scheme. Additionally, the visual encodings are updated to reflect document relevance based on the entity weights.

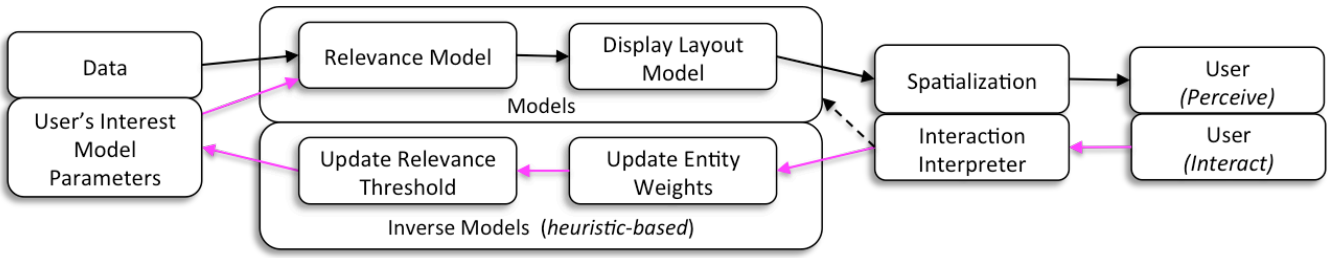

Fig. 4. Implemented version of the multi-scale semantic interaction visualization pipeline. In StarSPIRE, a relevance model and a display layout model are used. With each user interaction, the perceived importance of terms updates, changing the spatial and the working set of data is modified. The dashed black arrow indicates typical force-directed layout interactions that do not influence the user's interest model parameters.

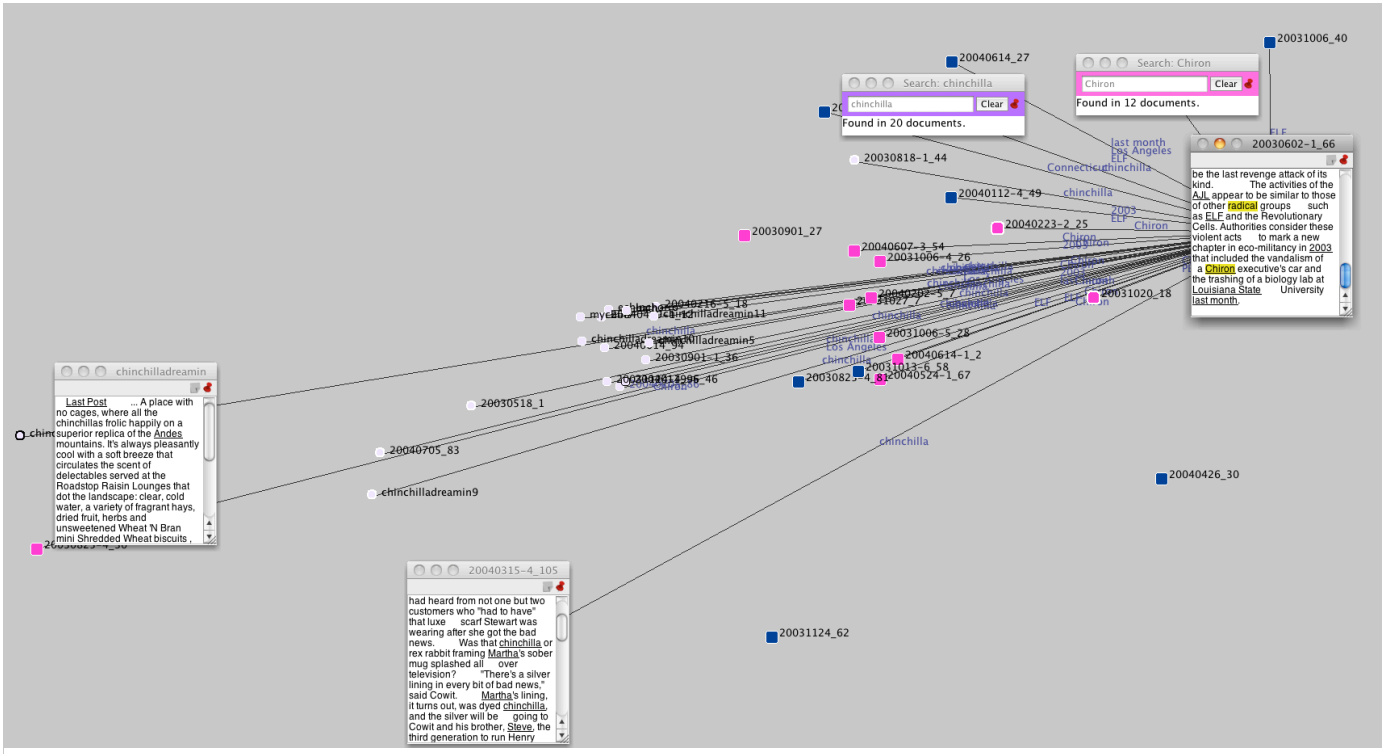

Fig. 5. StarSPIRE workspace, which is a node-link diagram connected by shared entities using a modified force-directed layout. Nodes represent closed documents, which are color-coded based on search terms. Node size and saturation encode document relevance, based on how well the document matches the user-driven entity-weighting scheme. Node outline color denotes read/unread status (white for unread, black for read). All nodes are labelled with their file names for easy tracking of documents as they move in the workspace. Edges radiate from the selected document node, labelled with shared entities.

Moving nodes and ***pinning nodes*** have no impact on the entity weighting scheme, but serve to rearrange the spatial workspace to reflect the user's organizational schema. These are traditional force-directed layout actions.

Resizing a document to make it ***larger*** or ***smaller*** increases or decreases the weight value of each entity contained in the document, respectively. This is interpreted as relevance feedback and the system updates the working set of documents appropriately.

Minimizing a document decreases the weight values of all entities contained in the document. ***Closing a document*** also decreases the weight values of all entities contained in the document, but at a higher magnitude than minimization [Figure 6].

Resizing a node to make it ***larger*** or ***smaller*** increases or decreases the weight values of all entities contained in the document, respectively. Resizing a node is accomplished by selecting a node and using the mouse scroll button to alter the node's size. If a node is made larger, the system queries for additional similar documents. If a node is made smaller, the system tracks this feedback to be less likely to retrieve similar documents in the future.

Annotating a document adds the [new] terms to the typed-in document and increases their weight values. The system retrieves documents matching the entities contained in the annotation.

Searching for a term increases that term's associated weight and retrieves documents matching the search term. This action returns more matching documents than other semantic interactions because it is an explicit request for related information.

Highlighting a term or phrase increases the weight values of all highlighted entities and retrieves documents matching the highlighted entities.

Overlapping documents increases the weight values of all common entities between the two overlapping documents and retrieves documents matching the shared entities between the documents.

With each semantic interaction, the spatial layout updates and, if necessary, the system queries for new relevant documents and adds them, if any, to the workspace. Because StarSPIRE is designed to test the usability of semantic interactions operating across multiple models (and theoretically vastly different levels of data scale), we have thus far only tested the system on smaller datasets (e.g. on the order of 10,000 documents). As a result, StarSPIRE is capable of updating all models (display layout and information retrieval) with each user interaction as well as storing the entire dataset in memory. This will likely not be the case with much larger datasets. Future implementations will likely require database support or leverage cloud-based architectures.

5.3 Relevance-Based Retrieval

We selected a simple modified linear search algorithm to serve as the relevance model. When StarSPIRE increases an entity's importance, it searches the backend database for additional documents to add to the workspace and adds the top *n* search results that exceed a relevance metric [Figure 7]. Currently, a maximum of twenty documents are added if the user executes a search and a maximum of eight documents are added from all other semantic interactions that result in a request for more information. Additional data can be obtained, if available, by repeating the interaction. This allows for progressive disclosure of information to keep too much information being added to the display at one time, which could overwhelm the analyst. The spatial layout then updates to accommodate these new data points.

The current relevancy-based threshold allows for a variable number of documents in the working set of data. By not restricting the number of documents that can be present on the screen, the user is capable of maintaining as much information as inferred to be relevant to their sensemaking task. In the future, this could be updated to allow for additional heuristics, such as the number of opened/closed document nodes, node proximity to the center of the workspace, or how recently a document has been added to the workspace.

Table 2. StarSPIRE's interpretation of semantic interactions in terms of the parametric updates to the model of the user's interests.

Interaction	*Model Parameter Effect*
Resize document	Scale all weights of terms in the document
Minimize document	Down-weight terms by 25%
Close document	Down-weight terms by or 40%, remove from working set
Resize node	Scale all weights of terms in the document
Annotation	Up-weight terms by a constant, add terms to model
Search	Up-weight term by a constant, add terms to model, adjust relevance threshold as needed
Highlight	Up-weight terms by a constant
Overlap documents	Up-weight shared terms by a constant

New information can be added to the display implicitly or explicitly. The user can explicitly query for new documents by executing a search. Implicit queries are constructed using the interpreted semantic interactions [**Table 2**]. These implicit queries are typically more complex than the explicit queries, which include single terms. The implicit queries often include multiple terms and their associated relative importance.

Documents that fall below the current relevance threshold are removed from the display, leaving the user with a working set of documents that match the user's interests in the data.

This retrieval process was chosen in order to support incremental changes to the information on the display as well as real-time interaction. If data were not merely added (or subtracted) from the displayed documents, the user could be presented with an entirely new set of displayed data, which could be disorienting. Thus, we prefer an incremental approach.

The psuedocode for StarSPIRE's retrieval algorithm is as follows:

```
retrieveDocuments(docsDisp, docsHid, W_t, W_t-1, limit):
    //docsDisp = list of documents displayed
    //docsHid = list of documents not displayed
    // W_t, W_t-1 = array of entity weights at timestep t and t-1, respectively
    //limit = maximum documents to add to the display
 1.  docMatches = empty list of documents
 2.  ΔW[] = W_t – W_t-1
 3.  for i = 1 : docsHid.length
 4.      weight = 0
 5.      for j = 1 : ΔW.length
 6.          if(docsHid[i].hasEntity(W_t.entity)
 7.              weight += ΔW[j]
 8.      if(weight > 0)
 9.          docMatches.add(docsHid[i])
10.  for i = 1 : docMatches.length
11.      docMatches[i].relevance =
                  sum(e.weight for each Entity e in docMatches[i]
12.  for i = 1 : docDisp.length
13.      docDisp[i].relevance =
                  sum(e.weight for each Entity e in docDisp[i])
14.  docsRanked[] = Sort(docMatches) based on relevance
15.  for i = 1 : min(limit, docsRanked.length)
16.      docsDisp.add(docsRanked[i])
17.      docsHid.remove(docsRanked[i])
18.  docsDisp[] = Sort(docsDisp) based on relevance
19.  for i = 1 : docsDisp.length
20.      docsDisp[i].rank = i
21.  return docsDisp
```

In the algorithm, the positive changes in entity weights are identified to determine which terms have increased in importance and should be used to identify new documents to add to the workspace. Step two computes the dot product between the entire backend dataset with the change in entity weights vector (ΔW), which results in a single number for each document. To optimize performance, we discard all documents whose value is zero, because they do not contain any entities whose weights were increased within the past timestep. This results in the set of documents docMatches, which are candidates for addition to the workspace. The weights of entities contained in these candidate documents are summed using the current weighting scheme to obtain a score that reflects how well each document matches what the user has deemed important in the dataset thus far. These values are then sorted and the top *n* documents are added to the list of documents included in the spatial workspace at timestep *t* and removed from the set of hidden documents (i.e. documents in the dataset not included in the spatial workspace). This results in the set of documents displayed at the next timestep, *t*+1. The algorithm returns this modified set of documents (which could be the same as the previous timestep if no documents are chosen to be added). The updated rank of each displayed document is stored as an attribute of each document. This rank information allows the system to apply appropriate visual encodings to denote how closely documents match the user-imparted entity importance values.

The documents returned from the information retrieval algorithm are then used as input, along with the current weighting scheme at timestep *t*, to the modified force-directed layout to determine the two-dimensional layout of the data points for timestep *t*+1.

We chose to select candidate documents first instead of applying the weight vector across all documents in the dataset in order to provide an incremental update to the displayed data. If we had applied W(t) to the entire dataset, it is possible that the displayed data would be much different in each iteration.

Similarly, selecting candidate documents and eliminating all documents which do not contain any of the newly increased entities allows us to optimize the retrieval process. This is crucial for maintaining a quick interaction-feedback loop.

The linear nature of this algorithm prevents it from scaling to much larger document collections. More advanced retrieval methods, either running in real time or as a background process, could be substituted in order to handle larger amounts of data.

6 USAGE SCENARIO

To demonstrate StarSPIRE's functionality, we used the VAST 2007 Challenge Dataset ("Blue Iguanodon") [17]. Because StarSPIRE is currently designed to operate on unstructured text documents only, we omitted all images and spreadsheets from the dataset, resulting in approximately 1,500 text files. Blog entries that were included in the data were converted into text files, one for each blog entry. Preliminary entity extraction was done on the dataset.

The challenge task is an open-ended sensemaking task to investigate "unexpected activities concerning wildlife law enforcement, endangered species issues, and ecoterrorism" [17]. We present the following usage scenario to demonstrate how StarSPIRE can leverage the MSI technique.

The user began with a search for "chinchilla." This was unsurprising, because the dataset contained a directory titled "Chinchillas." She read through several documents, arranging them in the display based on document similarity. The user then began highlighting information regarding chinchillas, which branched into additional endangered species. This loosely structured analysis continued until the user read a document concerning a musical artist owning an extremely large number of exotic animals whose actions did not seem to match his words regarding animal conservation. The analyst denoted this as suspicious and began investigating it further. This investigation was driven through highlighting the artist's name and the name of his animal sanctuary, which imported many documents onto the display, some of which had a large node size. The analyst opened the largest new nodes first.

[Figure 8] shows the evolution of the user's spatial organization schemas through the sensemaking task. Clusters of documents were

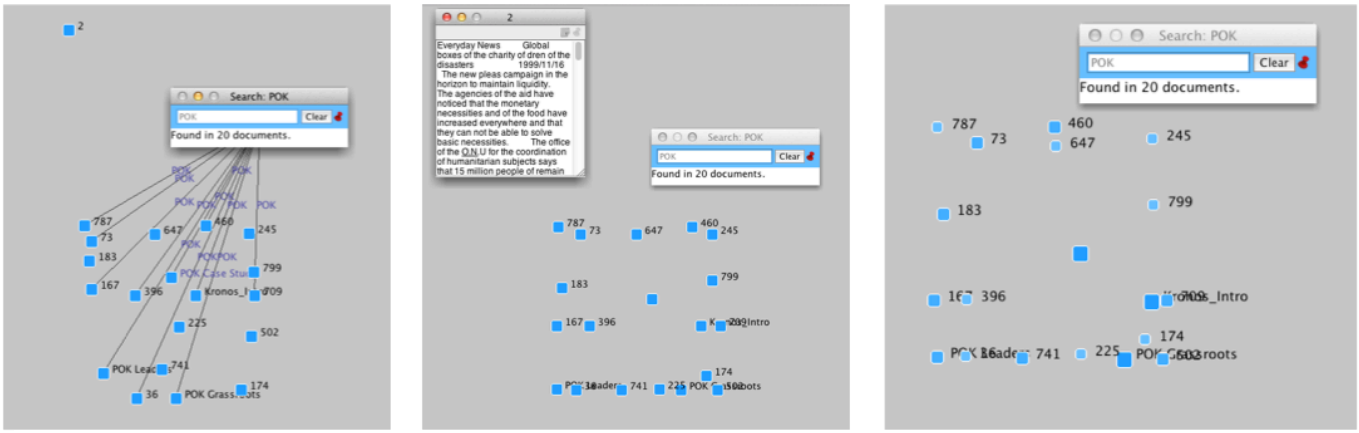

Fig. 6. Multi-model semantic interaction in StarSPIRE: Document relevance feedback. Left: The user explicitly searches for documents containing the term "POK." Documents matching this search term are added to the display and arranged using a "near = similar" metaphor. Middle: The user selects the outlying document, opens it, then closes the document to remove it from the workspace. Right: The system decreases the entity weights of the terms contained in the deleted document. The system updates the visual encodings to reflect this relevancy feedback and updates the display layout.

moved around the screen and a mixture of visual encodings and document proximity motivated the choice of documents to investigate next. Furthermore, it can be seen that the user initially executed two searches to obtain some initial documents, but then opted for other multi-scale semantic interaction techniques to obtain new documents (e.g. highlighting, linking documents – denoted by the purple bars, and annotating documents). Document annotations were used to record hypotheses and insights (e.g. "r'Bert is r'Bear?" and "r'Bear might have monkeypox"). In the later stages of analysis, searches were used primarily to label the space, serving as reminders of which documents concerns which persons or topics. However, they were also used to ensure that important information or documents had not been overlooked.

Once the user identified suspicious activity regarding a large exotic animal reservation, it became apparent that many documents were interconnected via several subplots. As her understanding of the dataset evolved, so did her spatial representation. For example, two documents that were initially considered "not quite relevant, but interesting enough to not minimize" concerning an outbreak of a disease were initially placed in the upper right hand corner of the display. After realizing that the owner of the large exotic animal sanctuary had contracted the same disease, she moved the two documents down next to the exotic animal sanctuary documents.

Highlights, document annotations, and document linking were primarily used to obtain new documents in the workspace. Searches were executed to check for additional information on important persons, but also used to label the spatial workspace. After approximately ninety minutes of analyzing the data, the user concluded that she had a sufficient understanding of the plot and subplots in the data.

The user's results were compared with the known ground truth solution. The user correctly identified four out of five subplots in the data. The use added 145 documents to the workspace, which is 10% of the actual dataset. 47 documents were opened and 33 remained open at the conclusion of the sensemaking session. The user made eight searches, four document annotations, and 21 highlights. 45 documents were added through searches, whereas the remaining 100 documents were added through other multi-scale semantic interactions (e.g. highlight, annotate, document proximity).

Out of 26 documents relevant to the final solution, the user had added 18 of them to the workspace. Six of these 18 documents were added through an explicit search, while twelve were added through implicit multi-scale semantic interactions. 13% (6/45) of documents added through explicit searches were relevant to the solution, and 12% (12/100) of documents added through implicit searches were relevant to the solution. Therefore, the documents that originated from multi-scale semantic interactions were similar in quality to those that originated from explicit searches from the user.

Out of approximately 1,500 documents, 47 were read. Thus, the analyst was able to construct 80% (four out of five subplots) of the solution while only reading 3.13% of the documents in the dataset. While the results of this usage scenario appear promising, further

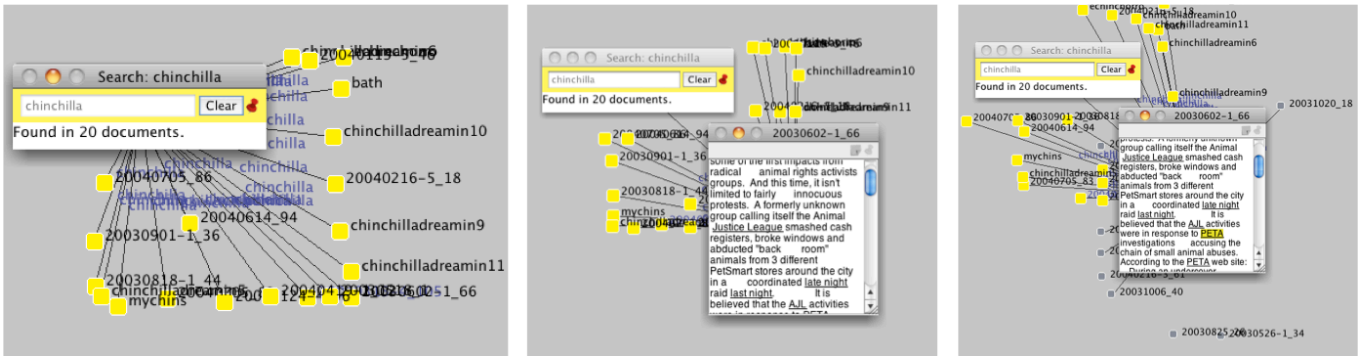

Fig. 7. Multi-model semantic interaction in StarSPIRE. Left: The user explicitly searches for documents containing the word "chinchilla." Documents matching this search term are added to the display and arranged. Middle: The user selects a document to read. To prevent occlusion, nodes are pushed aside but still maintain their relationships to other documents as much as possible. Right: The user highlights the entity "PETA." Eight new documents are retrieved and added to the display. Documents rearrange due to the shift in weighting scheme – documents that contain "chinchilla" and "PETA" (as well as other shared terms) are brought closer together in the middle, documents that contain only "chinchilla" are pushed to the top and left, and documents that only contain "PETA" are pushed to the bottom and right.

work is required to evaluate the performance of MSI techniques as compared to existing SI techniques.

7 Discussion

7.1 Comparison to Existing Techniques

Most similar to our system prototype is ForceSPIRE [10], which implements semantic interaction techniques and allows the exploration of small text datasets. However, ForceSPIRE operates using a single model (display layout), which hinders data analysis compared to StarSPIRE.

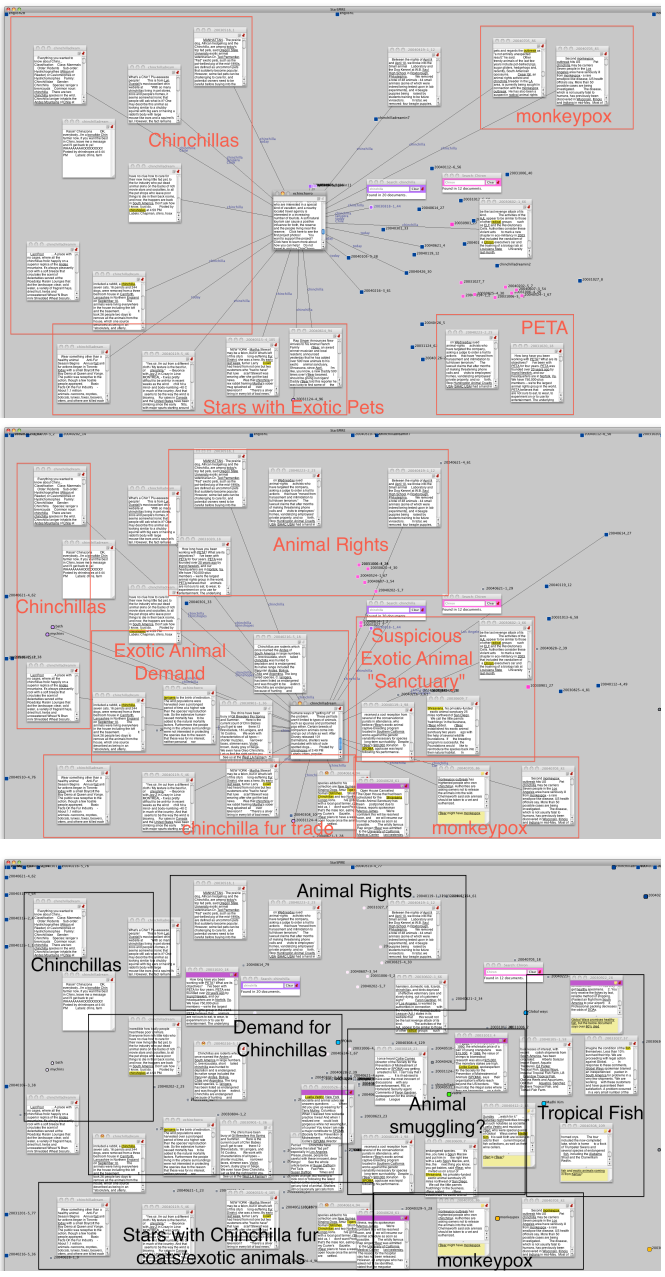

Fig. 8. Organizational schema evolution throughout the use case. Top: Early analysis into chinchillas and endangered species that are growing in popularity with a seemingly unrelated outbreak of monkeypox. Middle: Intermediate analysis that has linked chinchillas, the monkeypox outbreak, and a rapper keeping a suspicious exotic animal sanctuary. Bottom: Final spatial layout showing the relationships between multiple subplots in the dataset, along with searches that have been executed to label the space as well as hypotheses entered as annotations to documents.

Data loading and processing takes much longer in ForceSPIRE (several minutes) than in StarSPIRE (several seconds) for moderately sized datasets on the order of 1,000 to 1,500 documents. Most of this delay is computing the force-directed layout and the relationship between all documents. Because StarSPIRE stores most of the data and only displays a smaller working set of documents, processing is much faster. For these same reasons, the interaction-feedback loop is slower in ForceSPIRE. Thus, the large scale model relieves much of the computational overhead from the small scale model.

Furthermore, we have extended the visual encodings to give a richer overview of the displayed documents and have enabled the users to provide positive and negative relevance feedback, which is reflected in a separate model.

Instead of comparing these tools directly, we will design a comparative user study using StarSPIRE with MSI techniques enabled and with only SI techniques enabled. Users will be presented with a subset of the data on the screen, alleviating ForceSPIRE's inability to display more than a few hundred documents on the screen. With the SI-only condition, they will be required to explicitly request additional information through either through written queries or query-by-example (e.g. "show me more like this document"). This will allow for a comparison between the two techniques in regard to how information is retrieved and interacted with in the workspace.

Many existing systems transform user interactions into model feedback to drive a spatial layout. Dis-function [6] enables users to inject feedback into a spatialization model by repositioning points, allowing the system to incrementally update the distance function driving the low-dimensional projection of high-dimensional data. Similarly, Visual to Parametric Interaction [20] infers analytic reasoning from users moving and/or highlighting data points in a spatial projection of high-dimensional data. These interactions are converted into parametric updates to change the spatial layout. Work in observation-level interaction [14] also allows document repositioning to drive an underlying spatialization model. However, all of these techniques are limited to using a single model, whereas our technique leverages multiple models that are capable of operating at different levels of data scale.

7.2 Document Selection Models

The model used here is only one example of many possible models for document selection. We chose this approach in order to focus on the interactions within StarSPIRE and their mappings to the parameters driving the retrieval results. However, this approach is not practical for extremely large datasets with large numbers of entities. The relevancy-based retrieval algorithm used in StarSPIRE runs in O(nm) time where n is the number of documents and m is the number of entities, due to the initial search process. We have optimized the algorithm to perform the sorting operations on a subset of the possible documents to improve this runtime. However, the worst-case scenario is that all, or nearly all, documents in the dataset match an entity that has been upweighted (e.g. "the"). Even in average and best case scenarios, this algorithm is not an ideal choice for scaling to extremely large datasets. Parallelization is one option for speeding up the algorithm, but we also wish to consider alternative models for retrieval.

Future implementations of multi-model semantic interaction should consider the streaming and ever-growing nature of data. Accordingly, streaming or a mixture of dynamic and static models could be employed. Further methods of handling this type and amount of data could take a multi-threaded or parallel approach.

In addition to optimizations for algorithm performance, different models could be leveraged at the display layout and information retrieval levels. Different models naturally lend themselves to different interactions. For example, moving data points or pinning them as spatial landmarks could be interpreted by algorithms such as Latent Semantic Indexing [9], Principal Component Analysis [23], or Multi-Dimensional Scaling [31], among others, to adjust the lower-

dimensional space of all of the documents to create a representation that better fits the user's high-dimensional understanding of the data, thus producing subjectively better search results.

7.3 Multi-Model Visualization Pipeline

The flexibility of generalized multi-model semantic interaction enables researchers to explore many alternative models, methods of interpreting interactions, and mappings to analytical reasoning.

There are multiple options for routing of interactions to models. For some interactions (e.g. changing data point distances), it may be appropriate to propagate the interaction to each underlying model up the levels of scale. However, for other interactions (e.g. giving relevance feedback on a document), it may be more appropriate to send this feedback directly to a specific model. Further complicating matters, the same interaction may have a different intent based on context. For example, a user may construct a cluster of documents. The clustered documents could be important and relevant to the user, or the user could be grouping them in order to filter out other irrelevant documents from the main display area. In this example, it is possible that this distinction could be captured by the proximity of the cluster to the periphery or center of the display. Investigating alternative approaches to enabling users to naturally express these intents within the visual interactions remains an open research question.

It may be appropriate to maintain several models for each level of data scale and dynamically adapt which is used based on which model is able to best incorporate the user's feedback. For example, having multiple display layout models allows the system to choose the one that converges the best or has the lowest deviation from the user's feedback. We plan on investigating how the notion of competing models changes the performance of the system, both qualitatively and quantitatively. Maintaining multiple models for accomplishing a single task could result in a better approximation of the high-dimensional data, and we will investigate methods for providing visual feedback to inform users of these switches.

Due to the runtime of these algorithms and the time required to invert the models to compute a new representation, it may not be practical to apply interactive feedback to every model at each interaction. Slower models could be told to invert and execute after a certain number of interactions and instructed to run in the background. However, display-level models should be updated with each interaction in order to provide the user with immediate feedback. Therefore, whichever models are chosen to drive the spatial layout should execute quickly.

7.4 Limitations

StarSPIRE is currently designed for text analysis. Multimedia cannot currently be incorporated in the tool. Future implementations could overcome this limitation by using metadata and user-designated tags for multimedia files. Although StarSPIRE is not equipped to handle generic high-dimensional data, multi-model semantic interaction techniques can be applied across data types. As multi-model semantic interaction is an extension of observation-level interaction for high-dimensional data [32], these systems (ex. [6, 20]) are suitable for extension to multi-model semantic interaction.

We have applied multi-model semantic interaction techniques to sensemaking tasks, but have not attempted other analytical tasks, such as social network analysis. Additionally, we have not yet empirically evaluated if users understand and accept the mappings of interactions to model feedback. This will be conducted in future work.

StarSPIRE has currently been tested on over 10,000 text documents that had an entity extractor run on them, resulting in over 20,000 distinct entities. Total loading time was under one minute and interactions could be completed in close to real time (queries are typically executed in under three seconds). Due to the nature of the retrieval model, the execution time depends largely on the size of the set of candidate documents, which need to be sorted and ranked, then compared against a relevance threshold. This problem is exacerbated by document collections with extremely large numbers of entities. Therefore, broad searches tend to have slower response times. This limitation could be overcome by implementing more sophisticated and optimized algorithms. StarSPIRE is not equipped to handle much larger datasets (e.g. on the order of 100,000 documents and higher). Moving to a database or cloud-based architecture and implementing different algorithms could overcome this limitation.

8 Conclusion

In this paper, we have introduced the concept of multi-model semantic interaction, which harnesses user interactions to manipulate underlying models. We have presented an instantiation of this technique that operates across multiple levels of data scale. Along with this technique, we introduced a generalized visualization pipeline for semantic interaction using multiple models. We have shown an example implementation of multi-model semantic interaction techniques through the visual analytics tool prototype, StarSPIRE. Using this prototype, we demonstrated the functionality of multi-model semantic interaction techniques. Finally, we concluded with a discussion of multi-model semantic interaction techniques.

We plan on conducting a comparative user study using StarSPIRE to observe the differences between explicitly constructed queries and the addition of implicitly constructed queries. This will serve to compare multi-model semantic interaction with semantic interaction. The study will use one of the VAST Challenge datasets in order to quantitatively evaluate user performance.

Future work includes investigating additional multi-model semantic interaction techniques, visual encodings, and models. Additionally, we plan on creating a visual representation of the dataset to grant users an overview of the document content. We wish to practically apply multi-model semantic interaction techniques to much larger datasets, including streaming data. This will likely require implementing additional algorithms and cloud-based architectures.

We hope that multi-model semantic interaction will serve as a usable means of interacting with multiple models for data analytics.

Acknowledgments

This work was supported in part by NSF grant IIS-1218346.

References

[1] Andrews, C., Endert, A. and North, C. Space to think: large high-resolution displays for sensemaking *Proceedings of the SIGCHI conference on human factors in computing systems*, ACM, 2010, 55-64.

[2] Andrews, C. and North, C. Analyst's Workspace: An Embodied Sensemaking Environment For Large, High-Resolution Displays *IEEE visual analytics science and technology*, IEEE, 2012, 123-131.

[3] Belew, R.K., Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents. in *ACM SIGIR Forum*, (1989), ACM, 11-20.

[4] Benedikt, M., Cyberspace: some proposals. in *Cyberspace*, (1991), MIT Press, 119-224.

[5] Bradel, L., Self, J.Z., Endert, A., Hossain, M.S., North, C. and Ramakrishnan, N., How analysts cognitively "connect the dots". in *Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on*, (2013), 24-26.

[6] Brown, E.T., Liu, J., Brodley, C.E. and Chang, R., Dis-function: Learning distance functions interactively. in *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, (2012), IEEE, 83-92.

[7] Card, S.K. Visualizing retrieved information: A survey. *Computer Graphics and Applications, IEEE*, 1996, *16* (2). 63-67.

[8] Choo, J. and Park, H. Customizing computational methods for visual analytics with big data. *IEEE Computer Graphics and Applications*, 2013, *33* (4). 22-28.

[9] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R. Indexing by latent semantic analysis. *Journal of the American society for information science*, 1990, *41* (6). 391-407.

[10] Endert, A. Semantic Interaction for Visual Analytics: Inferring Analytical Reasoning for Model Steering, Virginia Polytechnic Institute and State University, 2012.

[11] Endert, A., Bradel, L. and North, C. Beyond control panels: Direct manipulation for visual analytics. *Computer Graphics and Applications, IEEE*, 2013, *33* (4). 6-13.

[12] Endert, A., Fiaux, P. and North, C. Semantic Interaction for Visual Text Anayltics *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 2012, 473-482.

[13] Endert, A., Fox, S., Maiti, D., Leman, S. and North, C. The semantics of clustering: analysis of user-generated spatializations of text documents *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ACM, Capri Island, Italy, 2012, 555-562.

[14] Endert, A., Han, C., Maiti, D., House, L., Leman, S. and North, C., Observation-level interaction with statistical models for visual analytics. in *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, (2011), IEEE, 121-130.

[15] Fruchterman, T.M. and Reingold, E.M. Graph drawing by force-directed placement. *Software: Practice and experience*, 1991, *21* (11). 1129-1164.

[16] Green, T.M., Ribarsky, W. and Fisher, B. Building and applying a human cognition model for visual analytics. *Information Visualization*, 2009, *8* (1). 1-13.

[17] Grinstein, G., Plaisant, C., Laskowski, S., O'Connell, T., Scholtz, J. and Whiting, M., VAST 2007 contest-blue iguanodon. in *IEEE Symposium on Visual Analytics Science and Technology*, (2007), IEEE, 231-232.

[18] Hearst, M.A., TileBars: visualization of term distribution information in full text information access. in *Proceedings of the SIGCHI conference on Human factors in computing systems*, (1995), ACM Press/Addison-Wesley Publishing Co., 59-66.

[19] Hossain, M.S., Andrews, C., Ramakrishnan, N. and North, C., Helping Intelligence Analysts Make Connections. in *Scalable Integration of Analytics and Visualization*, (2011).

[20] Hu, X., Bradel, L., Maiti, D., House, L. and North, C. Semantics of Directly Manipulating Spatializations. *Visualization and Computer Graphics, IEEE Transactions on*, 2013, *19* (12). 2052-2059.

[21] i2. Analyst Notebook, www.i2.co.uk, 2007.

[22] Jeong, D.H., Ziemkiewicz, C., Fisher, B., Ribarsky, W. and Chang, R., iPCA: An Interactive System for PCA,Äêbased Visual Analytics. in *Computer Graphics Forum*, (2009), Wiley Online Library, 767-774.

[23] Jolliffe, I.T. *Principal component analysis*. Springer-Verlag New York, 1986.

[24] Lamping, J., Rao, R. and Pirolli, P., A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. in *Proceedings of the SIGCHI conference on Human factors in computing systems*, (1995), ACM Press/Addison-Wesley Publishing Co., 401-408.

[25] Lin, X., Soergel, D. and Marchionini, G., A self-organizing semantic map for information retrieval. in *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, (1991), ACM, 262-269.

[26] Maron, M.E. and Kuhns, J.L. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 1960, *7* (3). 216-244.

[27] Olsen, K.A., Korfhage, R.R., Sochats, K.M., Spring, M.B. and Williams, J.G. Visualization of a document collection: The VIBE system. *Information Processing & Management*, 1993, *29* (1). 69-81.

[28] Pirolli, P. and Card, S. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis *International conference on intelligence analysis*, 2005.

[29] Shipman III, F.M. and Marshall, C.C. Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. *Computer Supported Cooperative Work (CSCW)*, 1999, *8* (4). 333-352.

[30] Stasko, J., Görg, C. and Liu, Z. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization*, 2008, *7* (2). 118-132.

[31] Torgerson, W. Multidimensional scaling: I. Theory and method. *Psychometrika*, 1952, *17* (4). 401-419.

[32] Vogt, K., Bradel, L., Andrews, C., North, C., Endert, A. and Hutchings, D. Co-located Collaborative Sensemaking on a Large High-Resolution Display with Multiple Input Devices *Conference on Human-Computer Interaction*, Springer, 2011, 589-604.

[33] Wise, J.A., Thomas, J.J., Pennock, K., Lantrip, D., Pottier, M., Schur, A. and Crow, V., Visualizing the non-visual: spatial analysis and interaction with information from text documents. in *Information Visualization, 1995. Proceedings.*, (1995), IEEE, 51-58.

[34] Wright, W., Schroh, D., Proulx, P., Skaburskis, A. and Cort, B., The Sandbox for analysis: concepts and methods. in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, (2006), ACM, 801-810.

[35] Yi, J.S., Melton, R., Stasko, J. and Jacko, J.A. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information visualization*, 2005, *4* (4). 239-256.