Semantic Interaction for Symmetrical Analysis and Automated Foraging of Documents and Terms

Michelle V. Dowling

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science & Application

Christopher L. North, Chair
Michael A. Horning, Co-chair
Joseph L. Gabbard
Edward A. Fox
Leanna L. House
Bohdan A. Nebesh

March 18, 2020
Blacksburg, Virginia

Keywords: Semantic interaction, semantic interaction foraging, symmetry, interactive visual analytics, exploratory data analysis

Semantic Interaction for Symmetrical Analysis and Automated Foraging of Documents and Terms

Michelle V. Dowling

(ABSTRACT)

Sensemaking tasks, such as reading many news articles to determine the truthfulness of a given claim, are difficult. These tasks require a series of iterative steps to first forage for relevant information and then synthesize this information into a final hypothesis. To assist with such tasks, visual analytics systems provide interactive visualizations of data to enable faster, more accurate, or more thorough analyses. For example, semantic interaction techniques leverage natural or intuitive interactions, like highlighting text, to automatically update the visualization parameters using machine learning. However, this process of using machine learning based on user interaction is not yet well defined. We began our research efforts by developing a computational pipeline that models and captures how a system processes semantic interactions. We then expanded this model to denote specifically how each component of the pipeline supports steps of the Sensemaking Process. Additionally, we recognized a cognitive symmetry in how analysts consider data items (like news articles) and their attributes (such as terms that appear within the articles). To support this symmetry, we also modeled how to visualize and interact with data items and their attributes simultaneously. We built a testbed system and conducted a user study to determine which analytic tasks are best supported by such symmetry. Then, we augmented the testbed system to scale up to large data using semantic interaction foraging, a method for automated foraging based on user interaction. This experience enabled our development of design challenges and a corresponding future research agenda centered on semantic interaction foraging. We began investigating this research agenda by conducting a second user study on when to apply semantic interaction foraging to better match the analyst's Sensemaking Process.

Semantic Interaction for Symmetrical Analysis and Automated Foraging of Documents and Terms

Michelle V. Dowling

(GENERAL AUDIENCE ABSTRACT)

Sensemaking tasks such as determining the truthfulness of a claim using news articles are complex, requiring a series of steps in which the relevance of each piece of information within the articles is first determined. Relevant pieces of information are then combined together until a conclusion may be reached regarding the truthfulness of the claim. To help with these tasks, interactive visualizations of data can make it easier or faster to find or combine information together. In this research, we focus on leveraging natural or intuitive interactions, such organizing documents in a 2-D space, which the system uses to perform machine learning to automatically adjust the visualization to better support the given task. We first model how systems perform such machine learning based on interaction as well as model how each component of the system supports the user's sensemaking task. Additionally, we developed a model and accompanying testbed system for simultaneously evaluating both data items (like news articles) and their attributes (such as terms within the articles) through symmetrical visualization and interaction methods. With this testbed system, we devised and conducted a user study to determine which types of tasks are supported or hindered by such symmetry. We then combined these models to build an additional testbed system that implemented a searching technique to automatically add previously unseen, relevant pieces of information to the visualization. Using our experience in implementing this automated searching technique, we defined design challenges to guide future implementations, along with a research agenda to refine the technique. We also devised and conducted another user study to determine when such automated searching should be triggered to best support the user's sensemaking task.

Dedication

To my friends and family, who have supported me tirelessly throughout this journey, through listening to me panic about my research, helping ease the stress with food or games, or even just being a friendly face.

And to my committee for helping me grow and learn as a person and as a researcher through their continual guidance and entertaining so many wacky project ideas. I can't imagine having a better group of people helping show me the way.

Acknowledgments

This research was partially supported by NSF grant IIS-1447416, NSF grant DGE-1545362, UrbComp (Urban Computing): Data Science for Modeling, Understanding, and Advancing Urban Populations, as well as by General Dynamics Mission Systems.

Attribution

In this work, we have included several published or presented papers. We describe each such paper here, including the contributions of other researchers. Firstly, Chapter 3 is based on the paper titled, "A Bidirectional Pipeline for Semantic Interaction," which was presented at the IEEE VIS 2018 workshop on Machine Learning from User Interactions for Visualization and Analytics [32]. Adam Binford created an initial version of a computational pipeline for semantic interaction, which was then adapted and refined based on additional visual analytics systems we created from this pipeline. The development of these systems was accomplished alongside the BaVA@VT research group under the guidance of Peter Hauck. John Wenskovitch, Peter Hauck, Nicholas Polys, and Chris North provided additional assistance in writing and editing this paper.

Similarly, Chapter 4 is based on the paper titled, "SIRIUS: Dual, Symmetric, Interactive Dimension Reductions," which was presented at IEEE VIS 2018 (© 2018 IEEE. Reprinted, with permission, from [31]). J.T. Fry, Scotland Leman, and Leanna House assisted with technical descriptions of the SIRIUS-based system described, with additional edits assisted by John Wenskovitch and Chris North. Mai Dahshan, Ian Crandell, and other members of the BaVA@VT research group helped develop this system.

Additionally, "Interactive Visual Analytics for Sensemaking with Big Text," as published through the Big Data Research special issue on "Big Data Exploration, Visualization and Analytics" [33], was used as the foundation for Chapter 6. This article represents a large collaborative effort, including system development and technical writing assistance from Nathan

Wycoff, Brian Mayer, Scotland Leman, and Leanna House. These development efforts were largely led by Peter Hauck, who guided researchers from the BaVA@VT group towards the system represented in the article. John Wenskovitch, Nicholas Polys, and Chris North provided additional writing assistance.

Although papers have not yet been submitted for publication yet, we wish to also recognize the contributions of other researchers in the material presented in Chapter 5 and Chapter 7. Michael Horning and Chris North provided guidance in these research efforts and assisted with editing the material presented in these chapters. Additionally, Nathan Wycoff assisted with the data analyses presented in both chapters, with additional analyses from Chapter 7 performed by Chreston Miller. John Wenskovitch also assisted in writing and editing the design challenges detailed in Chapter 7. The user studies performed in each of these chapters have IRB approval; the associated IRB approval letters accompany this dissertation as separate files.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The **Sensemaking Process** describes how people accomplish tasks by finding different pieces of information and combining them together in a coherent manner. As such, the Sensemaking Process, as defined by Pirolli and Card [90] and depicted in Figure 1.1, encompasses a series of iterative steps that represent the cognitive processes necessary to transform raw data into a formalized hypothesis. The iteration between these steps reflects a notion of incremental formalism [106], through which the analyst iteratively defines and refines their hypothesis. Additionally, these steps can be divided between two main loops: the Foraging Loop and the Synthesis Loop. The Foraging Loop describes how the analyst searches for new information, whereas the Synthesis Loop identifies the steps to combine information into a formalized hypothesis.

The Sensemaking Process can be used to describe a variety of tasks, from uncovering terrorist plots from different pieces of information, to predicting civil unrest events by cross-referencing news articles with social media, to journalism professionals determining the truthfulness of a claim. As such, finding methods to support the Sensemaking Process has far-reaching implications. However, supporting the iterative nature of the Sensemaking Process as well as the requirements of both the Foraging Loop and Synthesis Loop impose unique challenges. How can such iteration and incremental formalism be supported? How can the Foraging Loop and Synthesis Loop be facilitated? Can these loops be facilitated simultaneously? To put these questions into context, we focus on the example of journalism professionals investigat-

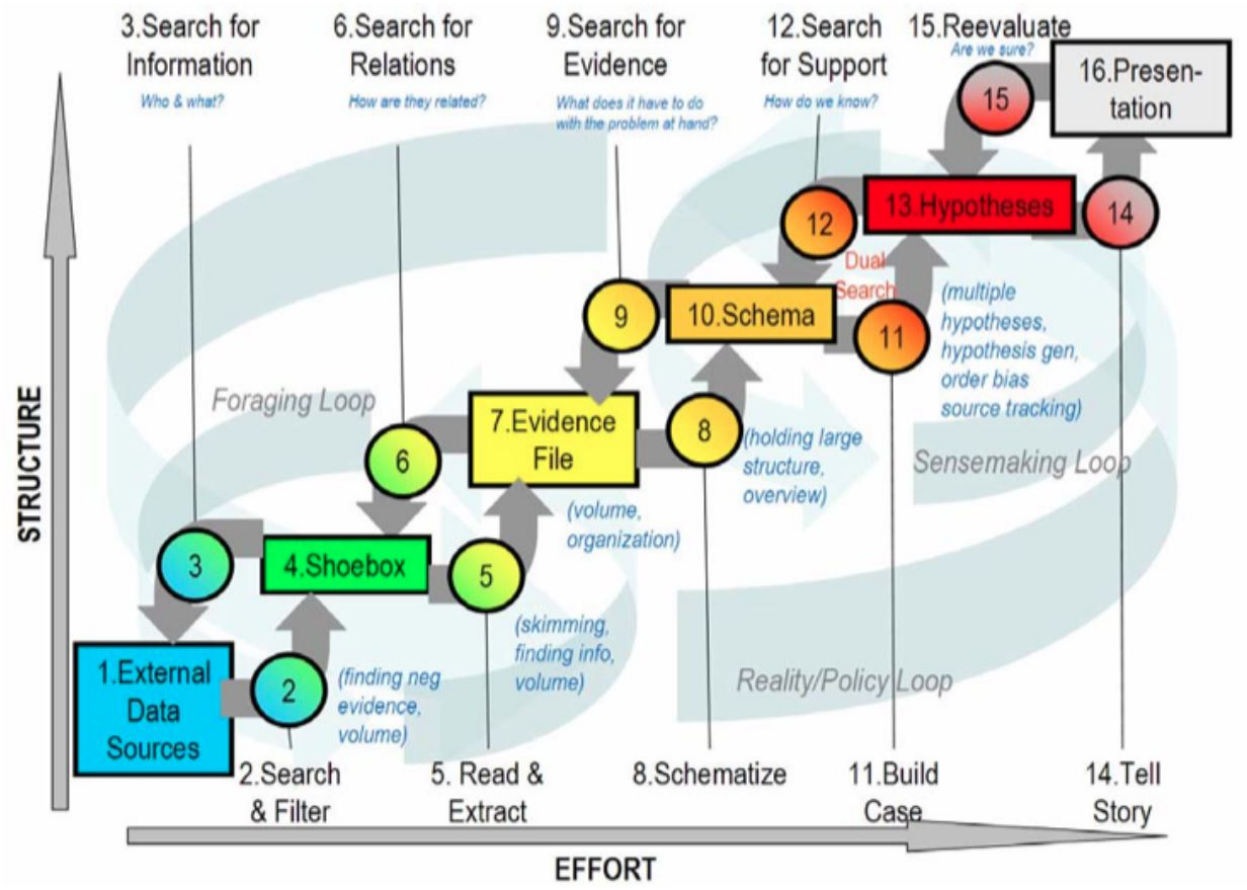

Figure 1.1: The Sensemaking Loop as defined by Pirolli and Card [90] is a series of iterative steps that describe the analyst's cognitive processes in transforming raw data into a formalized hypothesis. This process is divided between two large loops: the Foraging Loop and the Sensemaking Loop. To reduce confusion with the term "Sensemaking Process," we use the term "Synthesis Loop" to refer to the Sensemaking Loop. (Included under Fair Use, 2020.)

ing the truthfulness of a claim. What does the Sensemaking Process by which journalism professionals make such truthfulness decisions look like? How can this process be supported? Since this task involves foraging for information, making truthfulness and relevance determinations about each piece of information, and synthesizing different pieces of information together, how can each of these parts of their Sensemaking Process be facilitated? These questions become even more complicated to answer when considering how every journalism professional will complete their Sensemaking Process differently, mandating any potential

solution to be flexible enough to support many paths to completion.

We continue to use the example of journalism professionals determining the truthfulness of a claim to help motivate our work. Looking at the broader impacts of supporting such tasks, we find that riots, strikes, protests, and other instances of civil unrest in cities are becoming more common in today's political climate. These instances are fueled by the media, which can perpetuate fake or unverified news or present news in an inflammatory manner, thus further escalating such issues. Although efforts like the EMBERS project [91] seek to predict instances of civil unrest, we can also research how best to support journalism professionals in their analyses of claims as well as how they reach their decisions on how truthful or trustworthy such claims are. Such research may reveal new understandings in how these types of decisions are made. This understanding can then be extended to help inform, educate, and cause reflection in non-professionals regarding how they themselves determine the trustworthiness of various news stories. Such empowerment would hopefully lead people to more accurately judge the trustworthiness of news stories, thus mitigating the effects of fake, unverified, and inflammatory news.

To support such Sensemaking Processes, especially with large text datasets, the research area of **visual text analytics** has uncovered effective methods for visualizing and interacting with text datasets to enable thorough explorations and understanding of the data [12, 13, 15, 16, 39, 51, 52, 63, 115, 132]. In particular, **semantic interaction** is a technique that enables analysts to directly interact with visualizations of complex data in a natural manner [40, 41]. To accomplish this, semantic interaction reflects notions of Human-in-the-Loop analytics [42] by combining the computational power of machines with the analytical and reasoning capabilities of the analyst [12, 37, 41, 102]. Towards this goal, semantic interaction first captures interactions within the visualization and then interprets them to determine what is interesting or relevant to the analyst. This interpretation is then

reflected by *learning* model parameters through machine learning algorithms to produce an updated visualization. Such learning is at the heart of semantic interaction; it abstracts the algorithmic details behind the interaction to allow the analyst to interact with the visualization in a natural manner, thereby enabling the analyst to remain within their cognitive zone [106].

The inherent complexity of semantic interaction means that the designers of semantic interaction in visual analytics systems must determine which visualizations and interactions best support the analyst's Sensemaking Process as well as how the system should interpret the interactions themselves. Such complexity hints at the many facets of semantic interaction design to explore. Thorough exploration of semantic interaction design alternatives leads to a natural question of how to model and express the complexity of semantic interaction. Such a model must include the mathematical algorithms involved in learning new model parameters based on user interest as well as when and how these algorithms are used. However, as thoroughly described in Section 2.2, no such model or pipeline previously existed to properly capture this level of detail or complexity.

One example of the many facets of semantic interaction to explore is which portions of the data to involve in the visualization and interaction techniques. As described further in Section 2.3, current visual analytics systems for high-dimensional data are either observation-centric– enabling exploration that focuses on the observations– or attribute-centric– enabling exploration that focuses on the attributes[1]. For text analytics, this means that it is either the documents or their attributes (e.g., terms or topics) that dominate the visualization and are the primary thing that analysts interact with. However, observations (including documents) are understood based on their attributes, and the attributes are understood based on the

[1]In this work, we use the convention of referring to individual data items as "observations" and their dimensions, features, or variables as "attributes." A value that a given observation has for a specific attribute is the observation's "attribute value."

observations. For example, how similar two documents are depends on the terms that appear in them. Likewise, how similar two terms are depends on the documents in which they appear. This concept indicates a **cognitive symmetry** between how observations and attributes are understood. Such cognitive symmetry is also reflected in tasks, including identifying how different attributes affect the perception of the overall trustworthiness of news articles (which is a required task when determining the truthfulness of a given claim). In this example, how similar two documents are in their "trustworthiness" is based on the attributes of the documents, and how similar the attributes are in their "trustworthiness" depends on the documents themselves. Thus, enabling simultaneous exploration of both the observations and the attributes would support this symmetry. However, providing such support implies a number of design considerations, including how to provide an accurate projection of both observations and attributes as well as how to leverage a single interaction to update both projections. Likewise, it is necessary to understand **the impact that such symmetry has on the performance of a variety of analytical tasks**, such as the complexity of insights gained and the speed at which analysts can gain such insights.

Although there are many such facets of semantic interaction to explore, it is equally important to understand **how different components of a given visual analytics system map to the Sensemaking Process**. Such an understanding would highlight how the analyst's Sensemaking Process is being supported and, therefore, how such support could be altered or further improved. For example, Wenskovitch and North define **semantic interaction foraging** for text analytics [125]. This technique is demonstrated in StarSPIRE [12, 37], which leverages semantic interactions that enable analysts to express desired similarities/dissimilarities between documents as well as denote information of interest. Based on these interactions, the system then *learns* which terms the analyst is interested in. This new information is used to reproject all documents in an updated visualization. Additionally,

documents that contain these terms are then automatically foraged and added to the visualization. This semantic interaction foraging proved to be incredibly effective in helping analysts uncover relevant information, even when the model for determining which new documents to recommend was relatively simplistic [125]. However, there were previously no models that map computational components of a visual analytics system, like this semantic interaction foraging, to the Sensemaking Process, making it difficult to determine precisely how systems like StarSPIRE support the Sensemaking Process.

Additionally, **no guidelines previously existed for implementing semantic interaction foraging**. This gap in the existing literature impedes research into how semantic interaction foraging can be incorporated in visual analytics systems or how to further improve implementations of semantic interaction foraging. For example, **no previous research explored *when* automated foraging should occur**. Continuing with the example of StarSPIRE, semantic interaction foraging always occurred after every semantic interaction; the analyst had no control over this feature. However, the analyst may simply want to explore the current space for the moment and forage later. In such scenarios, automated foraging may result in too much information being displayed for the analyst's current task. Therefore, it is important to understand how and when the analyst would choose to use automated foraging techniques in order to have the system better adapt to the analyst's Sensemaking Process.

Given these existing challenges, we define our research questions to be:

1. How can we model semantic interaction to capture the complexity of how algorithms process and learn from the interactions?

2. How can we model symmetry in analytical tasks for both observations and attributes in the context of semantic interaction?

3. How does such symmetry affect analysts' time on task, accuracy, and their cognitive cardinality and dimensionality when performing sensemaking tasks? Are there certain tasks that symmetry best supports or hinders?

4. How can we model sensemaking in the context of semantic interaction for text analytics, including the interactions between foraging and synthesis processes?

5. When integrating semantic interaction foraging with these three models to support text analytics (e.g., journalism professionals determining the truthfulness of a claim), what are the design challenges for implementing semantic interaction foraging?

6. In a symmetrical system that includes semantic interaction foraging, how and when do analysts decide to use such automated foraging techniques?

Answering these research questions provides stepping stones towards the larger goal of understanding how to better support text analytic tasks, such as journalism professionals determining the trustworthiness of a claim.

To explore these research questions, we include here a series of chapters that highlights each research question in turn. First, we define a new, generalized computational pipeline to model and capture the complex steps necessary for visual analytics systems to process semantic interactions (Chapter 3). Then, we develop a mathematical model for symmetric visualization and interaction techniques to support symmetrical analyses of both observations and attributes of high-dimensional data simultaneously (Chapter 4). Using a testbed system developed directly from this model, we study the effect such symmetrical visualizations and interactions have on analysts' ability to perform a various analytic tasks, as measured by time on task, accuracy, and cognitive cardinality and dimensionality[2] (Chapter 5). Additionally,

[2]We use the same definitions of cardinality and dimensionality as in [103], where cardinality refers to the number of observations that an analyst uses in their analytical reasoning and dimensionality is the number of attributes.

we explore how to model semantic interactions in the context of the analyst's Sensemaking Process (Chapter 6). We demonstrate this model through a new visual analytics system for big text data. The model for this system reflects how semantic interaction foraging for documents supports the Foraging Loop so that the analyst can remain focused on their Synthesis Loop. We then combine these different research avenues to develop a visual analytics system that encompasses both semantic interaction foraging as well as symmetric visualization and interaction design to assist the Sensemaking Process when performing text analytics tasks. This system is used to exemplify how design challenges for implementing semantic interaction foraging may be addressed. We conclude by using this new system to evaluate how and when analysts, such as journalism professionals who determine the trustworthiness of claims, choose to use automated foraging in their analysis process (Chapter 7).

Chapter 2

Literature Review

Here, we focus on the underlying components to this research to identify existing literature related to semantic interaction, modeling such interactions using pipeline representations, modeling symmetric interface and interaction design, and modeling how semantic interactions map to the Sensemaking Process. We also discuss notions of automated foraging, which provides a foundation for semantic interaction foraging specifically.

2.1 Semantic Interaction

Endert defines semantic interaction by the following steps [40]:

1. Capture the interaction
2. Interpret the associated analytical reasoning
3. Update the underlying model

To exemplify semantic interaction, ForceSPIRE was introduced as a text analytics tool that leverages natural or intuitive interactions, such as highlighting text, to iteratively build a user interest model centered on terms within the corpus to drive the visualization [41]. Thus, the system *learns* which terms are important to the analyst to then automatically update the visualization. For the example of highlighting text, this interaction would lead to tighter

clustering of documents based on the highlighted terms. As such, semantic interaction abstracts model parameters to enable the analyst to remain in their cognitive zone [106], thereby supporting the analyst's Sensemaking Process [90]. Additionally, this abstraction means that the analyst does not have to have a deep knowledge or understanding of the underlying models in order to interact with the system.

Bradel et al. built upon ForceSPIRE to create StarSPIRE [12], which includes the ability to perform semantic interaction foraging [125]. Using this automated foraging technique, analysts are able to see new, relevant documents appear on the screen based on the user interest model formed. For example, highlighting text in StarSPIRE creates the same update to the visualization as in ForceSPIRE with the addition of new documents being included (i.e., foraged) in the visualization based on the highlighted terms.

This broader concept of semantic interaction – in which the system interprets the user intent based on the interaction to then provide an updated visualization – has been used to develop other interaction techniques applicable outside of text analytics systems. For example, Endert et al. developed Observation-Level Interaction (OLI), in which an analyst can directly manipulate a similarity-based spatialization of high-dimensional data to express desired similarity and dissimilarity relationships between observations [37]. In response, the system *learns* how to update the underlying model parameters to reflect these similarity/dissimilarity relationships and ultimately produces an updated visualization to reflect this learning. OLI was applied in a visual analytics system for quantitative high-dimensional data called Andromeda [102]. Through this system, Self et al. demonstrated the power of such interactions to enable more complex insights [103]. Throughout this work, we focus on these types of examples of the broader concept of semantic interaction to incorporate related work in areas outside of text analytics or aside from work that directly cites semantic interaction techniques [6, 15, 16, 23, 30, 34, 62, 70, 78, 83, 87, 95, 121].

2.2 Semantic Interaction Pipelines

To fully capture the complexity of semantic interactions, our first research question seeks to define a new conceptual pipeline that depicts the structure of the feedback loop between the various data processing components of the semantic interaction-enabled visual analytics pipeline. We justify the need for such a pipeline by surveying the current state of commonly-referenced pipeline models in information visualization and visual analytics as well as exploring the breadth of pipelines used to model existing visual analytics tools.

2.2.1 Existing Pipelines for Visual Analytics Processes

The fields of information visualization and visual analytics rely on computational and visual pipelines to convert data into visual displays. For example, Figure 1.1 shows the Sensemaking Process defined by Pirolli and Card. [90], which identifies the different mental processes involved in transforming raw data into a presentation of a formalized hypothesis. Figure 2.1 depicts both the information visualization pipeline of Card et al. [19] and the visual analytics task process of Keim et al. [61].

These pipelines model a high-level representation of how raw data is transformed to a final visualization or presentation and are quite generalizable, but the resulting trade-off is that these pipelines abstract any details of the mathematical model(s) and visualization(s) into single nodes in the graph. For example, the emphasis on a high-level abstraction on visual analytics task processes means that the pipeline defined by Keim et al. does not explicitly discuss interaction. Similarly, the focus on the mental processes in the Sensemaking Process means that mathematical models are not considered in this pipeline. In contrast, the interactions described in the pipeline from Card et al. represent methods to directly manipulate parameters of mathematical models, such as slider interactions. While this is

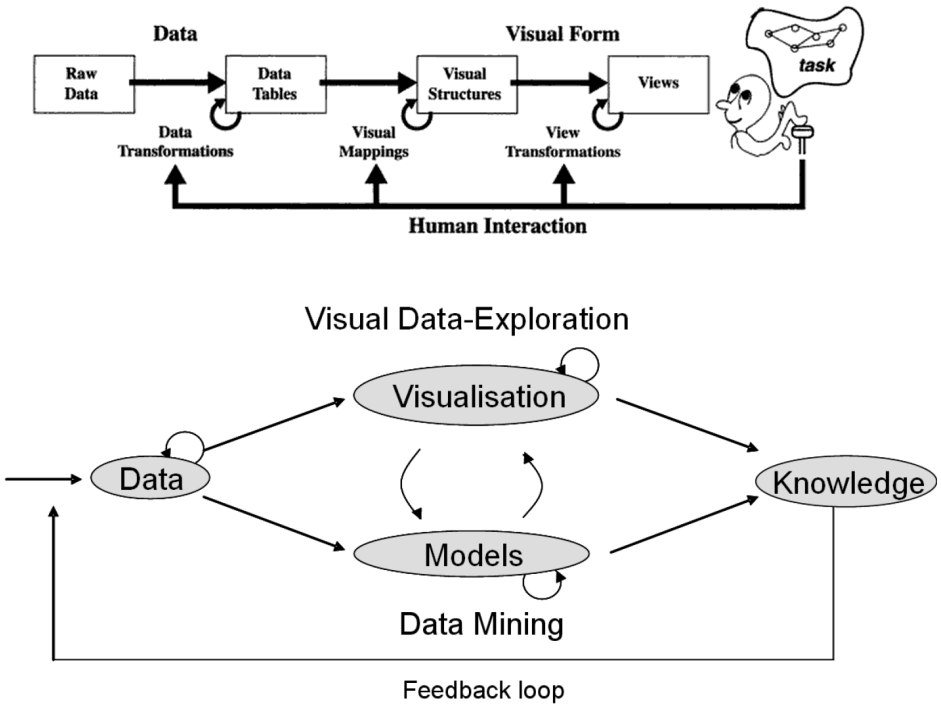

Figure 2.1: **(top)** The information visualization pipeline presented by Card et al. [19] does not specifically model semantic interactions. (Copyright Elsevier 1999.) **(bottom)** The visual analytics model provided by Keim et al. [61] provides a high-level overview of the structure of visual analytics knowledge discovery, but lacks detail in defining how mathematical models are used to interpret semantic interactions. In order to support semantic interaction, a different pipeline structure is necessary. (Copyright © 2008, Springer-Verlag Berlin Heidelberg)

interaction, we do not define this to be *semantic* interaction as no interpretation is necessary by any model for this interaction; the value provided by the analyst is simply stored and used. Thus, the precise mechanisms used to process and visualize the data or to interpret semantic interactions are not adequately captured in either of these pipelines. Looking at other pipeline representations, such as those presented in a survey of analytical pipelines by Wang et al. [122], we find the same limitations for semantic interaction tool design. Thus, these pipelines are insufficient for capturing how to support semantic interaction in visual analytics systems.

In contrast to these pipelines, Virtual to Parametric Interaction (V2PI) [73] provides a

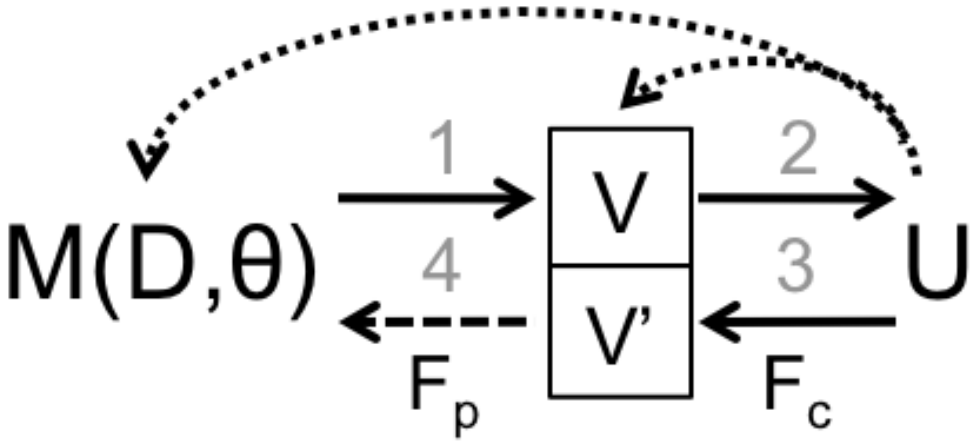

Figure 2.2: V2PI [73] is a mathematical representation of semantic interaction. This framework supports the creation of a visualization V. When the analyst U manipulates V to form V' via a semantic interaction, this triggers a manipulation of the parameters θ that influence model M. The parameterized feedback (F_p) represents an inverse process similar to what is described by the Sensemaking Loop, in which the interaction is interpreted as a set of updates to model parameters. (Included under Fair Use, 2020.)

statistical semi-supervised machine learning methodology for realizing semantic interaction. The V2PI pipeline (Figure 2.2) supports interactivity for visualizations and relies on both proven statistical methods and the analyst's judgment. In this pipeline, a visualization is created by processing data and parameters through a mathematical model. This visualization is presented to the analyst to evaluate. The analyst can directly manipulate the visualization, referred to as cognitive feedback. This cognitive feedback is translated into parameterized feedback, typically via machine learning, which updates the model through newly learned parameter values. As a result, a new visualization is created based on the analyst's interaction. Given this definition of V2PI, we assert that V2PI appropriately captures the basics of semantic interaction. However, V2PI only permits the exploration of a single mathematical model to accomplish such interactions. Thus, while V2PI may be able to capture simple visual analytics tools which contain a single mathematical model, it is not capable of representing tools with multiple models, such as StarSPIRE [12].

2.2.2 System-Specific Pipelines

Semantic interaction tools have become increasingly varied in interaction methods and purpose. To incorporate semantic interaction, some systems leverage the V2PI framework previously discussed, including ForceSPIRE [41], StarSPIRE [12], and Andromeda [101]. In each of these systems, the analyst directly interacts with observations in a dimension-reduced projection of data. These interactions drive a model that learns the relative importance of the attributes in the high-dimensional data space.

Additional systems also support interacting with a projection, but were not explicitly created with the V2PI framework in mind. Examples include the LAMP framework described by Joia et al. [59] and the extension to iLAMP [30], the technique described by Mamani et al. [78], Dis-Function [15], the technique defined by Paulovich et al. [87], and the system developed by Molchanov et al. [83]. However, semantic interaction can extend beyond interactions with projected observations to learn attribute weights. For example, both InterAxis [62] and AxiSketcher [70] use interactions on observations in the projection to update the axes of the projection. Intent Radar [95] introduces interactive intent modeling, allowing an analyst to provide feedback by dragging or clicking keywords, increasing the relevance by moving the keyword closer to the center or decreasing it by moving it outward in the radar interface. Moving away from projection-based tools entirely, Podium [121] is a tabular ranking system in which an analyst reorders rows (i.e., observations) in the table while the tool learns the attributes important to the current ranking scheme. iCluster [34] provides analysts with the ability to interactively move documents into clusters, learning the attributes important to the current clustering scheme. Similarly, ReGroup [6] interactively learns a model of group membership as an analyst adds members to groups.

Although each of these systems employ semantic interaction, few of the papers offer associ-

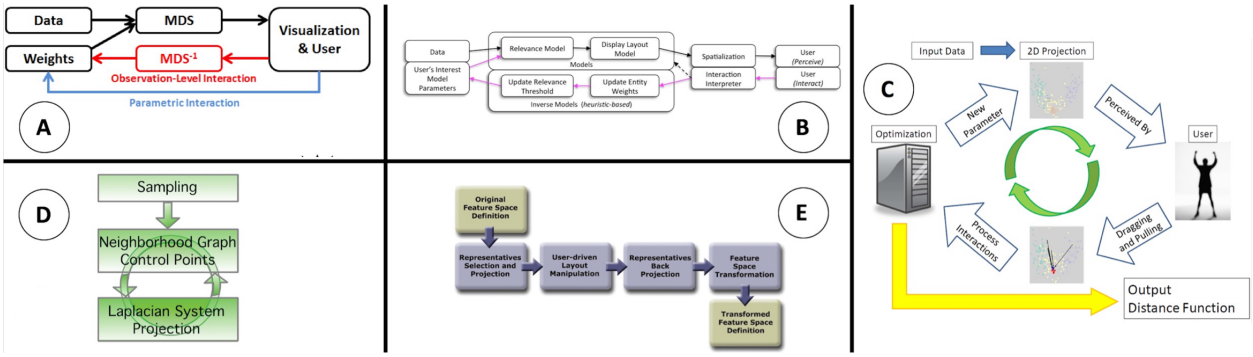

Figure 2.3: A series of pipeline representations of existing systems, including from (A) Andromeda [102] (© 2016 ACM), (B) StarSPIRE [12] (© 2014 IEEE), and (C) Dis-Function [15] (© 2012 IEEE), (D) Piecewise Laplacian Projection [87] (© 2011 The Author(s) Journal compilation © 2011 The Eurographics Association and Blackwell Publishing Ltd.), and (C) Mamani et al. [78] (© 2013 The Author(s) Computer Graphics Forum © 2013 The Eurographics Association and Blackwell Publishing Ltd.). We revisit these pipelines in Chapter 3 (Figure 3.3).

ated pipelines to properly capture the complexity involved with the interaction. Moreover, the pipeline representations are diverse, ranging from high-level abstractions to more detail-oriented representations. In Figure 2.3, we show a subset of these pipeline representations, including Andromeda [102], StarSPIRE [12], Dis-Function [15], Piecewise Laplacian Projection [87], and the pipeline provided by Mamani et al. [78] to describe their technique. Although the pipelines for Andromeda and StarSPIRE arguably achieve the highest level of detail to capture the semantic interaction therein, we feel that these pipelines can be improved to focus even more on the mathematical models used to create the visualization and interpret the semantic interactions therein (which is further explained in Section 3.4). In contrast, the pipelines for Dis-Function, Piecewise Laplacian Projection, and the technique by Mamani et al. are high-level pipelines which focus on the general concepts behind how the associated tools and techniques work. The trade-off in these, just as with the pipelines by Card et al. and Keim et al., is that the mathematical models used are abstracted away, making it difficult to determine how the semantic interactions therein are accomplished. Thus, we see a need for defining a new conceptual pipeline that can capture the complexity

of semantic interaction in visual analytics systems.

2.3 Symmetric Interface and Interaction Design

Here, we provide a brief survey of interactive visual analytics techniques for exploratory data analysis with high-dimensional data to highlight a lack of connection and symmetry between observations and attributes. We directly address this limitation via SIRIUS, as described in Chapter 4. In the following discussion, we refer to visualizations as simplistic if they do not incorporate many dimensions, and interactions as simplistic if they result in a trivial interpretation to change a mathematical model used to process or visualize the data.

2.3.1 Displaying and Interacting with Attributes

The attributes of high-dimensional data are visualized using a variety of techniques, ranging from simplistic (e.g., a raw data table or data matrix [15]) to more complex and informative (e.g., MDS projections [23, 117] or PCA (Principal Component Analysis) projections [16, 58, 76, 117, 138, 139]). In most cases, attribute visualizations implement a more simplistic technique like visual encodings such as color [23, 54, 95, 99, 105, 137, 138], size [4, 23, 95], or labels [4, 16, 63, 95]. Another method is to show the attribute values for observations along a one-dimensional line [54, 102]. Individual axes in parallel coordinates [54, 66, 133] create a similar visualization of attributes. As the complexity of the visualizations grow, we begin to see visualizations capable of conveying more information, such as scatterplots [118], histograms [104], heatmaps [15, 133], and polar coordinates [95, 134, 135]. Some visualizations implement a specific method for conveying information with this level of complexity, such as the arrows used by Brown [16], representing attributes as “magnetic”

nodes that pull on observation nodes [84, 137], and the "Axis Rainbow" in AxiSketcher [70]. The most complex examples of visualizations for attributes include the aforementioned MDS and PCA projections.

Interactions with the attributes tend to also be more simplistic. For techniques that map an individual attribute to an axis, the axis can be enlarged, shrunk, or rotated to alter how the given attribute influences the visualization [54, 60, 102]. Another method for altering the visualization of the attributes is to change the color mapping [4]. To see data associated with a particular attribute, analysts can sometimes hover over or click on nodes [16, 23]. Brushing and linking can also be used to highlight attributes [118, 138]. Searching mechanisms allow new attributes to be added to the visualization [95], while sorting enables analysts to easily find specific attributes [54]. Some techniques also support clustering of attributes [95, 138]. With respect to attributes, there are few examples of more complex interactions, such as dragging the attribute nodes [84, 95, 137], the focus and context interactions described by Turkay et al. [118], the update features described by Brown [16], and altering the attribute values in the aster plot in AxiSketcher [70]. These types of interactions adjust the underlying visualization mechanisms to update the visualization itself based on the analyst's interaction.

2.3.2 Displaying and Interacting with Observations

Visualization techniques used to display the observations of high-dimensional data also range widely in complexity, but they tend to be more complex in comparison to those used to display the attributes. The least complex of these are raw data [15], color [23, 63, 104, 138], size [23], and lists [95]. Increasing in complexity, we again see visualizations like heatmaps [104] as well as frequency plots [66], dendrograms [104], and scatterplots [54, 59, 70, 118]. Many visualizations for observations attempt to incorporate all the attributes explicitly

in visualizations [54, 60, 66, 84, 133, 137]. Examples of implicitly including all attributes can be seen in scatterplot-like projections of the data (e.g., MDS projections [23, 102], PCA projections [15, 66, 118, 138], t-SNE [63], and force-directed layouts [4, 126]).

Similarly, the interactions on the observations tend towards more complex interactions. Simplistic interactions include hovering or clicking on representations of observations to see the associated data [15, 16, 23, 60, 66, 84, 102, 126] and altering how color is mapped in the visualization [23, 99, 104]. Common, but still simple, interactions such as filtering [60, 84, 104, 133, 138], searching [16, 133], and brushing and linking [15, 16, 104, 118, 138] are often included. However, many visualizations for the observations enable direct manipulation of the visualization itself, such as how the attributes are used for the axes [54, 60, 104], manipulating the projection to alter an underlying mathematical model [15, 59, 102], selecting the clustering algorithm used or at what level clustering occurs [104, 138], manipulating how different attributes influence the visualization of the observations [4, 84, 102, 126, 137], drawing lines through the visualization to redefine the axes [70], and using a lens to separate groups of observations [63].

2.3.3 Projecting Attributes and Observations

Although at first glance it may appear that techniques such as the Data Context Map [23], Dimension Projection Matrix/Tree [138], and the visualization defined by Turkay et al. [118] provide symmetry in how observations and attributes are visualized and interacted with, there are important differences in their visualization methods, interaction methods, or both. In the Data Context Map [23], observations and attributes are plotted in the same MDS projection. Such a projection enables insights regarding similarity-based relationships between observations and attributes while contextualizing the projection of the observations.

However, the tradeoff is that this technique necessarily distorts either the projection of the observations, the attributes, or both. This distortion is caused by the fact that each additional piece of data plotted in an MDS projection influences the projection of all other data. Therefore, the observations and/or attributes can appear more or less similar than what they actually are. Furthermore, the interactions for the Data Context Map focus on drawing contours around observations based on ranges of attribute values; there is no interaction to draw contours based on ranges of observations.

As for the Dimension Projection Matrix/Tree [138], both observations and attributes are visualized using PCA projections. However, the observation projection can be split into multiple projections based on specific subsets of attributes, whereas the attribute projection always remains a single projection. Additionally, the projections of the observations are given colored axes based on their corresponding subset of attributes; such information about the observations is not provided in the attribute projection. Thus, while this visualization and the interactions therein enable exploration of how attribute subspaces affect projections of the observations, the obvious tradeoff in this technique is that such exploration of observation subspaces is not supported.

Lastly, the visualization by Turkay et al. [118] provides three scatterplots: one for observations using one attribute for each axis, an observation projection using two principal components, and one visualizing attributes using their mean and standard deviation. Despite observations and attributes being displayed in separate scatterplots, the manner in which each portrays information is inherently different, meaning these projections do not have a strong symmetry. Additionally, interactions include brushing and linking between observations selected in the first scatterplot and associated data in the other two scatterplots, as well as focus and context interactions based on selected attributes in the attribute scatterplot. Thus, the interactions for observations are very different than those for attributes, creating

further disparity between observations and attributes. Therefore, while this visualization technique enables deep exploration of the observations, the tradeoff is that such exploration for attributes is not well-supported.

Given the above discussion, attributes are generally treated very differently than observations, yet many tasks that analysts have regarding attributes are symmetrically similar to those regarding observations. Therefore, there is an opportunity to explore a new part of the design space of semantic interaction in visual analytics in which observations and attributes are displayed and interacted with in a symmetric and interconnected manner. Thus, we propose a new, symmetric exploratory data analysis technique for visualizing and interacting with both observations and attributes of high-dimensional data called SIRIUS, detailed in Chapter 4.

2.4 Modeling the Sensemaking Process with Semantic Interaction

Our approach for modeling the Sensemaking Process [90] with semantic interaction [40] is designed from a synthesis of the following concepts from the literature and advances previous work in this area [12, 41, 101, 126].

2.4.1 Information Synthesis

A variety of visual analytics systems incorporate various synthesis models, including network-based synthesis [79, 115], entity profile synthesis [10], spatial synthesis [38, 62], and interactive clustering [96]. We focus this discussion on spatial synthesis, in which space is used to represent the cognitive model of the analyst. This often takes the form of a "proximity ≈

similarity" visual metaphor, in which similar documents and data points are displayed near each other while dissimilar items are positioned at a distance.

Previous studies have shown that human analysts often make use of physical space to organize and synthesize text data [8, 38, 94]. Such synthesis techniques have been implemented in a variety of systems. For example, Analyst's Workspace [7] supports a manual approach for spatial synthesis of documents, whereas ForceSPIRE [41], StarSPIRE [12], and BigSPIRE [13] add computational support to assist with the spatial organization. However, these methods were based on heuristics. Building on lessons learned from these existing techniques, systems such as Andromeda [102], SIRIUS (from Chapter 4), and InterAxis [62] use a semi-supervised machine learning approach for spatial synthesis. We leverage a similar approach in Cosmos (described in Chapter 6) to enable the interactive positioning of documents within a visual display in a statistically valid and data-supported fashion.

2.4.2 Information Foraging and Retrieval

Many foraging models have also been developed for information retrieval. These include techniques such as simple keyword search foraging, creation of user interest models [12, 95], data-based dynamic query expansion [91], query-by-example systems [9], and recommender systems [92]. The foraging system used by Cosmos falls under the user interest model category. We take a "content-based filtering" approach, in which documents are assigned a score determined by profiles of the item in question and the analyst exploring the collection of all items [17]. Past work has shown that these user models can both broaden queries and help analysts to overcome bias in their foraging process [125].

2.4.3 Learning through Interactive Visual Feedback

Both the synthesis and foraging processes described in the last two subsections can be learned incrementally through iterative user feedback as part of a human-in-the-loop [42] process. To achieve this incremental learning, some visual analytics systems incorporate semantic interactions [39], emphasizing a contextualized feedback loop between the system and the analyst. While methods implemented in past text analytics systems [12, 41] make use of semantic interactions, the learning process implemented in those systems is heuristic. To scale up the benefits of semantic interaction, more rigorous modeling is needed.

Visual analytics frameworks such as V2PI [73] and BaVA [56] offer a potential solution to this modeling challenge. For example, Andromeda [102] provides analysts with the ability to interactively steer weighted multidimensional scaling (WMDS) projections of quantitative data. Analysts position a subset of the observations in the space to communicate desired similarity relationships to the system. Andromeda uses those positions to learn a distance metric that is applied to the full projection. This technique for manipulating projections is referred to as observation-level interaction (OLI) [37, 101] and is characterized by the learning step undertaken to generate the distance metric. In other words, the analyst's intent is inferred from his or her interactions, leading to a learned parameter change in the system. This is in contrast to parametric interaction (PI), in which the analyst directly communicates a desired parameter change to the system [101]. In Cosmos, we adapt these methods to support text data.

2.5 Supporting Automated Foraging

In this subsection, we describe how automated foraging techniques, such as semantic interaction foraging (SIF) [125], can benefit the analyst's Sensemaking Process.

2.5.1 Supporting the Sensemaking Process

The **Sensemaking Process** [90] is a cognitive model that describes the process of incrementally formalizing [20, 106] raw data into a supported hypothesis through a series of iterative steps. This process is divided into two equal subcomponents: the Foraging Loop and the Synthesis Loop. The Foraging Loop captures how the analyst finds relevant information, whereas the Synthesis Loop reflects how the information is combined to develop a final hypothesis.

In visual analytics, the Sensemaking Process is supported by providing an interactive visual representation of the data that enables the analyst to explore the given data more quickly, efficiently, or accurately. In particular, **Semantic Interaction (SI)** has proven useful in achieving this goal [12, 37, 95, 102, 124, 126] by leveraging natural or intuitive interactions to *learn* how to update underlying model parameters and help analysts in their task [40]. Thus, the analyst does not need to be an expert in any of the underlying models to use the given visual analytics system, nor do they need to focus on manipulating individual parameters of the underlying models directly. As an example, Andromeda [102] affords projection interactions (PrI)[1]. This interaction technique enables the analyst to directly manipulate a similarity-based projection of high-dimensional data to denote desired similarity/dissimilarity relationships between specific observations (i.e., individual data items) based on their

[1]The term used in [102] is observation-level interaction (OLI) [37, 102]. However, given our leveraging of symmetrical system design in Chapter 4 and the associated re-terming of OLI to PrI, we follow the convention of calling such interaction PrI instead.

attribute values. In response, the system *learns* weights on the attributes (i.e., dimensions or features of the data items) to apply on the attribute values of each observation to create such similarity/dissimilarity relationships. This learning is crucial to performing **semantic interaction foraging (SIF)**, which acts as a recommender system that leverages the information learned from SI to automatically forage for new, relevant data on behalf of the analyst [125]. To our knowledge, only Centaurus, StarSPIRE [12], and Cosmos (described in Chapter 6) employ SIF.

As shown in Chapter 4, a notion of **symmetrical visualization and interaction design** shows promise in assisting analyses of both the observations and the attributes simultaneously. Although this concept is applicable to any high-dimensional dataset, the clearest example of the usefulness of symmetry can be seen in text analytics. For example, if a analyst searches for the term "cat," then documents that mention "cat" should be displayed to the analyst. Understanding the relationships between these documents relies on their similarity, which is ultimately defined by their common terms. Therefore, it is also useful to understand what these common terms are between these "cat" documents (e.g., "cute" and "furry") and the relationships between these terms. However, relationships between terms are defined by the documents in which they appear, highlighting a notion of cognitive symmetry between observations and attributes. Thus, symmetrical visualization and interactions for both observations and attributes supports this cognitive symmetry and the desire to understand the relationships between both observations and attributes.

This technique of enabling simultaneous and symmetric exploration of both observations and attributes, called SIRIUS (described in Chapter 4), underlies Centaurus (detailed in Chapter 7). However, Centaurus also incorporates SIF to forage for more documents (i.e., observations) and terms (i.e., attributes) based on the analyst's interactions with the system. With the addition of SIF, Centaurus is transformed into a system that performs both

document and query space modifications that is iteratively constructed and manipulated based on the analyst's interactions. We describe how SIF is implemented in Centaurus in Section 7.2, including by an example analysis in Section 7.2.2. We then use Centaurus to exemplify how the design challenges for SIF might be addressed in Section 7.3. Additionally, we begin to explore how to better refine these design challenges in Section 7.4.

2.5.2 Text Analytics Systems

In visual text analytics, a visual representation of the documents, terms, and/or topics is provided to assist analysts in understanding their data and finding relevant information. **Spatial projections** of documents, where the relative proximity of the documents denotes their similarity, has proven to be a natural and intuitive method for understanding relationships between documents [7], as reflected by the prevalence of such visualization techniques for text [12, 25, 26, 47, 51, 52, 63, 93, 95, 109, 130]. However, other methods for visualizing text data have also been leveraged, such as depicting the change in topics over time [67, 123], similarity and relevance relationships between terms [95], or even categorical binning of documents [72, 111].

In order for analysts to understand the relevance of documents to their analysis, recommender systems employ a **ranking algorithm**. While many ranking algorithms exist (e.g., those listed by Yang [136]), a simplistic ranking algorithm first represents each document as a bag-of-words (e.g., term frequency – inverse document frequency (TF-IDF) values) to then apply a weight on each term within the bag-of-words, thereby creating a vector space model with TF-IDF weights. As a result, terms that receive a higher weight will have a greater influence in determining the relevance of a given document [65, 136]. For example, if an analyst performs traditional keyword search foraging (KSF), the term(s) used in the query would

receive a higher weight, directly resulting in documents containing those terms receiving a higher estimated relevance.

In many text analytics systems (e.g., [12, 21, 72, 95, 98, 111, 123]), only a subset of documents or terms are displayed. This is because visualizing all documents or terms (e.g., as an overview of the data [108]) becomes infeasible as the size of the data grows. Not only can attempting to display big text data result in visual clutter, but it also becomes more difficult for the analyst to focus only on data relevant to their investigation. Thus, a relevant subset of such data must be selected to better assist the analyst's task. Additionally, only displaying a subset of data has the benefit of enabling a system to scale to larger datasets.

To determine such a subset of data, the same ranking algorithm may be used in conjunction with thresholds, such as only the top n data may be added to the display or only documents above a specified relevance threshold may be displayed. Using a **top n threshold** [12] ensures that the analyst will not be overwhelmed by too many new data appearing in the display. A **relevance threshold** [12] ensures that the analyst is not shown irrelevant or distracting data. However, a tradeoff in implementing a such thresholds is that the analyst may feel that not enough data was returned by the algorithm. This issue may be mitigated by enabling the analyst to ask for more data if desired (e.g., by repeating the same interaction or clicking a "Forage for More" button). As an alternative, these thresholds may dynamic, adjusting to the analyst's need either by enabling the analyst to directly provide input or by automatically performing such adjustments based on the analyst's interactions.

In Centaurus, we leverage the information learned from SI to apply a simplistic ranking algorithm and employ both a top n threshold and a relevance threshold to use in SIF, as described in Section 7.2. Both SI and SIF are demonstrated in Section 7.2.2. In so doing, we assert the prowess of SIF in a symmetrically-designed system, even when such simplistic methods are used, as described in Section 7.5.

Chapter 3

Modeling Semantic Interaction

3.1 Introduction

Semantic interaction is a powerful interaction methodology, allowing analysts to explore and discover relationships in data [39]. For example, a number of semantic interaction systems and techniques have been developed that make use of OLI-like interactions [12, 37, 41, 56, 73, 101]. These interactions allow an analyst to continue exploring and understanding relationships in the data without pausing to manipulate model parameters manually. This frees the analyst's cognition to focus on high-level analysis concepts rather than low-level parameter details [38]. As the analyst continues to perform such semantic interactions, the system learns more about the analyst's reasoning, and the visualization incrementally adjusts to reflect the current data exploration [41].

Though a number of systems that use semantic interactions have been developed, each is described in a distinct manner to highlight the purpose for which the system was built. Although there are more generalized pipelines to model or describe the concepts behind such visual analytics systems, such as those proposed by Card et al. for information visualization [19] and Keim et al. [61] for visual analytics tasks, they do not incorporate sufficient focus on the interactions to fully capture the power and complexity of semantic interaction. Thus, it can be difficult to understand how semantic interactions affect the underlying mathematical models and how these mathematical models work together in a single system.

To address this need for capturing the complexity involved in semantic interactions for visual analytics systems, we begin by exploring the characteristics of semantic interactions in such systems. With these characteristics to guide us, we define a new pipeline that can properly communicate how the visualization is created and how semantic interactions are interpreted. We then demonstrate this new pipeline's capabilities by discussing the pipeline representations for a set of existing systems as well as a selection of new visual analytics systems. Finally, we discuss other implications of using this new pipeline, such as the ability to more thoroughly explore the design space of semantic interaction or enable rapid prototyping, as well as the limitations.

Specifically, we note the following contributions:

1. A review of the necessary components to accomplish semantic interactions in visual analytics systems (i.e., model composability, inverse computations, pipeline bidirectionality);

2. A new conceptual pipeline that incorporates these necessary components to model semantic interaction in visual analytics systems;

3. A set of examples demonstrating how this pipeline is capable of modeling semantic interaction designs in both existing and new visual analytics systems.

3.2 Characteristics of Semantic Interaction in Visual Analytics Systems

When comparing the characteristics of the visual analytics systems discussed in Section 2.2, we note that there are several commonalities. Combined with the ideas from the Sensemaking

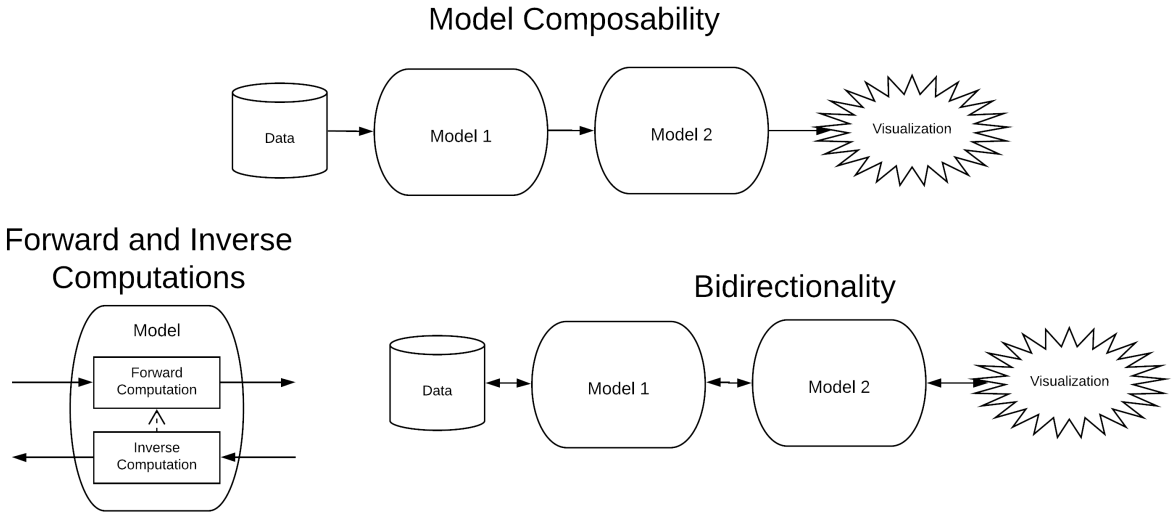

Figure 3.1: A representation of our three characteristics for a new semantic interaction pipeline: Model Composability, Bidirectionality, and Model Inversion. Model Composability refers to how different mathematical models must work together to produce the desired visualization. Bidirectionality allows interactions to drive updates to the underlying models. Model Inversion refers to the pairs of a forward computation with an inverse computation. The inverse computation supports the translation of semantic interactions into manipulations of model parameters.

Process [90], we define three properties as necessary for supporting semantic interaction in visual analytics systems. Each of these properties map directly to structures required to represent the complexity involved in modeling semantic interactions in a generalized pipeline for visual analytics systems.

3.2.1 Model Composability

The first characteristic we identified is that each mathematical model used to process the data as it works its way to the final visualization has specific input and output requirements. This hints at how these mathematical models must be composed to work together within the pipeline in order to produce the desired visualization. For example, Principle Component Analysis (PCA) requires numerical high-dimensional data as input to produce

low-dimensional coordinates as output. Therefore, any models preceding PCA must produce these high-dimensional data, and any models after PCA must be able to work with the low-dimensional coordinates as input. As another example, Weighted WMDS accepts numerical high-dimensional data as well as a set of attribute weights as input to produce low-dimensional coordinates as output. Thus, while the output is the same as with PCA, the input requirements have changed. This change must be accounted for in either data preprocessing steps or in a mathematical model that precedes the WMDS model. Therefore, model composability is a fundamental characteristic of semantic interaction and is represented by the top row of Figure 3.1.

3.2.2 Forward and Inverse Computations

While the model composability characteristic may seem simple or intuitive, it has important implications for the structure of a pipeline that captures semantic interaction. For example, Andromeda [102] uses WMDS to produce low-dimensional coordinates given a set of attribute weights. However, OLI expands the WMDS model by providing new low-dimensional coordinates from which to *learn* a new set of attribute weights. Given that the dataset is treated as a constant, this effectively inverts the WMDS computation.

We find this type of computation inversion common in systems with semantic interaction [12, 15, 78, 87, 102, 126]; it is this inversion which defines the learning or interpretation necessary to realize semantic interaction. Therefore, we propose that computation inversion is a required characteristic for visual analytics systems that support semantic interaction. Thus, our new pipeline must capture both forward and inverse computations for a given model. This concept is represented by the bottom right of Figure 3.1. Combined with the aforementioned model composability, this means that each mathematical model must fulfill

composability requirements for its inverse computation as well as its forward computation.

3.2.3 Looping Sensemaking via Bidirectionality

Taking the model composability and forward and inverse requirements a step further begins to imply a required bidirectionality in how the models are used together. In other words, each model must fulfill composability requirements for both its forward and inverse computations. Combine this with the fact that the forward computations help produce the given visualization and the inverse computations help interpret an interaction, then the pipeline must be bidirectional to support a looping structure. This bidirectional structure can be seen in both StarSPIRE [12] and Andromeda [102], where each use inverse computations of their models to interpret semantic interactions is followed by the forward computations to generate updated visualizations.

Referring back to the Sensemaking Process [90], we see a similar structure between pairs of processes that allow for information to be progressively transformed. These pairs of processes allow the transformation to occur in both forward and inverse directions, implying that there is a concept of looping between these collections of information. Thus, bidirectionality in a pipeline to represent semantic interaction mimics this natural process of incrementally building information to generate an output and then reassessing and refining information to produce a better output. This approach captures the concept of *incremental formalism* [7, 8, 106] in the cognitive sensemaking processes, in which analysts incrementally improve their mental models of the data through interaction, and represents that cognitive process formally as a machine learning process.

However, the Sensemaking Process as well as existing semantic interaction systems [78, 87, 102, 126] also indicate that it is not always necessary to iterate through the entire pipeline

and all models to generate the desired results. As an example, Andromeda [102] uses the aformentioned semantic interaction of OLI. When this occurs, an inverse computation is triggered that determines new attribute weights given a set of low-dimensional coordinates. However, since all the observations are already visualized, there is no need to pull any additional data into the pipeline. Thus, there is no need for any new data processing, meaning processing can skip to immediately recalculating new low-dimensional coordinates for all observations using the learned attribute weights. In the Sensemaking Process, a similar concept is represented by the fact that the analyst does not have to go all the way back to the external data sources every time he/she wishes to refine information. For instance, an analyst refining an evidence file may only need to reread or perhaps read more of a file that has already been accessed rather than foraging for a completely new file.

These examples reveal an important feature with respect to this bidirectionality characteristic: the ability to short circuit the rest of the pipeline when appropriate. This is a key new feature of a multi-model pipeline not found in earlier definitions [12]. Short circuiting happens when the inverse computation of a model does not need to send the interaction any further down the pipeline. Thus, instead of running the entire pipeline, we can short circuit to skip over unnecessary components of the pipeline, executing the forward computations beginning with the last model used to perform an inverse computation. From there, other models that were also updated should also have their forward computations rerun to produce an updated visualization. While the bidirectionality characteristic is represented in the lower-left of Figure 3.1, this short circuiting concept is depicted by the upward arrow between the inverse computation and the forward computation in the lower-right of Figure 3.1.

3.3 Components of a Semantic Interaction Pipeline for Visual Analytics Systems

3.3.1 A New Semantic Interaction Pipeline

When evaluating traditional visual analytics models (e.g., Keim et al. [61]), we note that there is rarely a distinction between different models that may be used in the pipeline. Thus, model composability is not well-represented in these existing models. Furthermore, while bidirectionality may be represented on some level, the manner in which the visual analytics pipeline handles this bidirectionality is not discussed or represented in detail. Additionally, there is no representation of inverse computations within the models. Therefore, there is a need for a new pipeline for visual analytics systems that better captures these characteristics of semantic interaction.

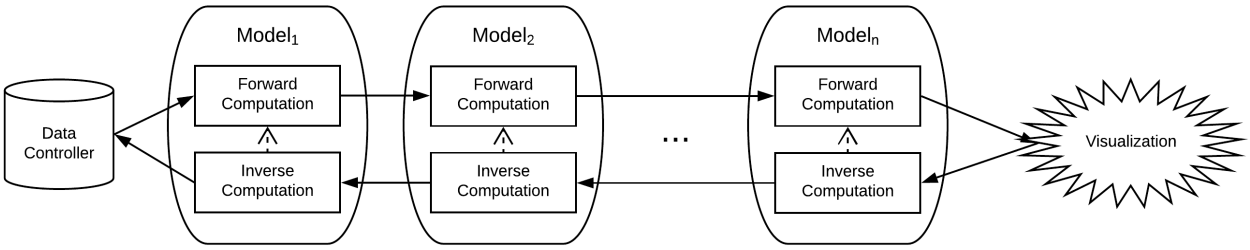

Figure 3.2: Our new pipeline for semantic interaction in visual analytics systems, created from the combination of the three characteristics shown in Figure 3.1. Model composability is shown through the chaining of a series of models horizontally in the pipeline. Bidirectionality results from the separated forward (top) and inverse (bottom) paths through the models. Model inversion is shown through the pairing of a forward computation and an inverse computation in each of the models. This representation also shows short circuiting arrows that connect the inverse and forward computations in the Models. The resulting structure captures how data is transformed into a Visualization and how semantic interactions are interpreted to update the parameters of the forward computations of the different Models.

For our proposed new pipeline, we require properties of the pipeline to map back to the model composability, model inversion, and bidirectionality characteristics discussed previously. To

capture these characteristics, we define this new pipeline to consist of three components, which are further described in the following subsections: a Data Controller, a set of Models, and the Visualization[1]. This new pipeline is shown in Figure 3.2. The forward and inverse computation characteristic is addressed by having each Model represent a set of such computations. Arrows between Models and other pipeline components indicate how the input and outputs requirements for each component line up[2], thereby addressing the model composability characteristic. The bidirectionality of the overall pipeline is handled through transitions between these computations, using the forward computations in the projection direction and the inverse computations in the interaction direction to loop through the pipeline in response to a semantic interaction. Upward arrows between inverse computations and forward computations of a given Model show when the pipeline short-circuits rather than iterating through the entire pipeline to interpret a semantic interaction. Thus, this proposed structure accurately captures the power and complexity of semantic interactions.

3.3.2 Models

As is evident from our discussion thus far, the primary focal point of our proposed pipeline is the Models. This is because the Models alone must encompass two of the three identified characteristics of semantic interaction: model composability and model inversion. To capture the inversion characteristic, each Model consists of a set of computations: a forward computation that is used to help produce the desired Visualization and at least one inverse computation that is used to help interpret an interaction by updating the inputs to the forward computation. These inverse computations can come in many forms, including precise

[1]To differentiate common terms (e.g., a mathematical model) from pipeline components, we capitalize pipeline components (e.g., Model).

[2]As shown in the pipeline in Section 3.5.2, it is certainly possible to create non-linear pipelines that model how subsets of models collaborate to handle different groups of semantic interactions.

mathematical inverses [101], heuristic inverses [126], and probabilistic inverses [56].

The forward and inverse computations of the Model naturally have input and output requirements, hinting at the given Model's composability with other pipeline components, whether they be other Models in the pipeline, the Data Controller, or the Visualization itself. These input and output requirements and how they are addressed is implied by how the Model connects to these other components in the pipeline. In Figure 3.2, this connectivity between a given Model and other pipeline components is represented by the arrows between these pipeline components. These arrows therefore represent the process used to both create the desired Visualization and interpret interactions within the Visualization. However, it is important to note that while these arrows provide an overview of how each Model is composed within the pipeline, the specific details of how composability requirements are met are left to the corresponding text accompanying the pipeline. To help provide more details for these composability requirements through the pipeline itself, the pipeline can be further annotated. For example, the arrows throughout the pipeline can be annotated with mathematical variables used to represent the inputs and outputs of each pipeline component. The trade-off in doing so is that such annotations may lead to visual clutter or initial confusion as to what these annotations mean.

Since these arrows represent the processes of Visualization production and interaction interpretation, we begin to note how the bidirectionality requirement is also addressed through this new pipeline. That is, there are a set of arrows that flow through each Model in the pipeline in a forward direction to produce the given Visualization as well as a set of arrows that flow in a backwards direction to interpret interactions within the Visualization. Thus, the manner in which the Models are represented in the pipeline alongside the other pipeline components denotes the pipeline's bidirectionality, thereby capturing this final characteristic of semantic interaction.

However, there is an additional nuance of bidirectionality that is also captured within each Model: being able to short-circuit the pipeline when no further computation is needed to interpret the given interaction. This is represented by an arrow between the inverse computation of a Model and its forward computation. Referring to our previous example with Andromeda, OLI does not need to communicate with any other pipeline component. Therefore, there is no need to send this interaction further down the pipeline, allowing the Model to short-ciruit. This immediately triggers a recalculation of the low-dimensional coordinates of the data using the newly calculated attribute weights, enabling the Visualization to update as soon as possible.

3.3.3 Data Controller

Which Models are supported in a pipeline is highly dependent on the data being used. Therefore, to better contextualize the Models in the pipeline, our pipeline necessitates a Data Controller to serve as the main access point to the underlying data that is being visualized. Its key purpose is to retrieve the raw data and any possible metadata as well as to transform this data into a form usable to the Models through data preprocessing. Thus, the Data Controller can enable analysts to view the raw data directly or allow the pipeline to pull additional data to process and visualize. This means that a Data Controller is specific to a particular type of data or dataset.

3.3.4 Visualization

Finally, it is difficult to understand or appreciate a Model without understanding the Visualization being used and the interactions enabled therein. Therefore, our new pipeline also requires a Visualization component. Firstly, the Visualization must define how the output

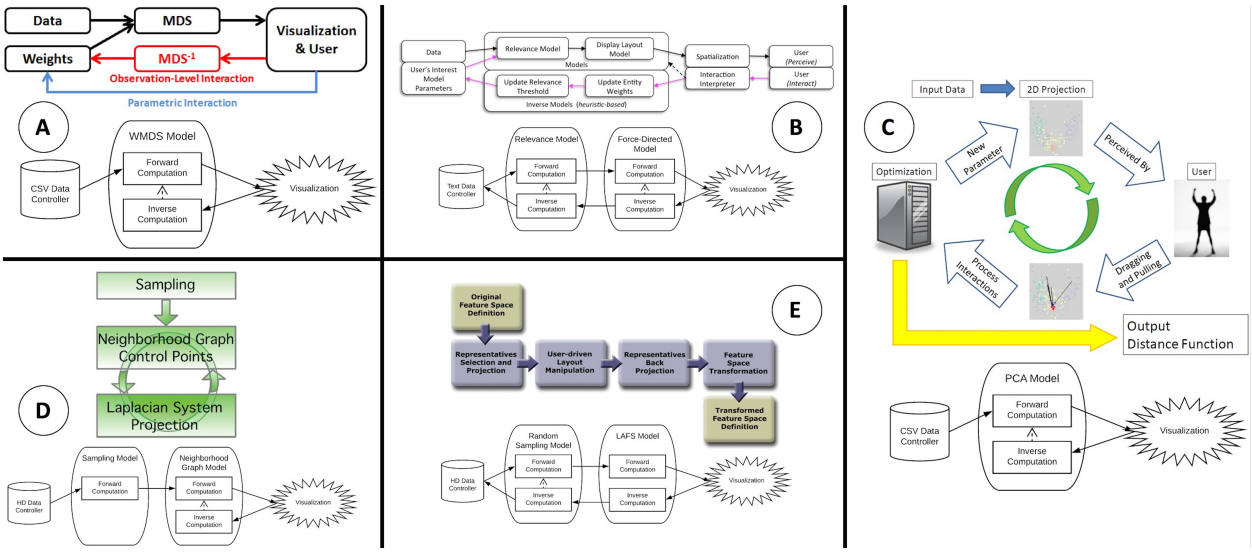

Figure 3.3: Using the proposed semantic interaction pipeline shown in Figure 3.2, we can now model the behavior of existing semantic interaction systems like (A) Andromeda [102] (© 2016 ACM), (B) StarSPIRE [12] (© 2014 IEEE), and (C) Dis-Function [15] (© 2012 IEEE), (D) Piecewise Laplacian Projection [87] (© 2011 The Author(s) Journal compilation © 2011 The Eurographics Association and Blackwell Publishing Ltd.), and (E) Mamani et al. [78] (© 2013 The Author(s) Computer Graphics Forum © 2013 The Eurographics Association and Blackwell Publishing Ltd.).

from the Models is mapped to different visual elements in the Visualization. Additionally, this pipeline component determines how the visual elements are interacted with and which Model(s) should be used to interpret this interaction. Thus, an interaction within the Visualization initiates inverse computations in the Models of the pipeline to interpret the given interaction and produce an updated Visualization.

3.4 Using the Pipeline for Existing Visual Analytics Systems

With this pipeline structure, we have the ability to well describe the complexity of semantic interaction in existing visual analytics systems. To exemplify this, we focus on the same

five visual analytics systems and techniques represented in Figure 2.3. Figure 3.3 shows a side-by-side comparison of the pipelines provided in the perspective papers for each system or technique and how to represent each using our newly proposed pipeline. From A to E in Figure 3.3:

Andromeda [102] provides a scatterplot projection of numerical high-dimensional data using WMDS. In this projection, the analyst can perform OLI to provide new low-dimensional coordinates for a subset of the observations. From these observations, new attribute weights are learned, which are then used to update the low-dimensional coordinates of all the observations. Both these interactions manipulate the parameters for the WMDS mathematical model. Therefore, three pipeline components are needed to represent Andromeda using our new pipeline: a CSV Data Controller, a WMDS Model, and the Visualization. The CSV Data Controller reads in a specified CSV file of numerical high-dimensional data and normalizes each attribute using z-scores. It also initializes each attribute weight to be $1/p$, where p is the number of attributes in the dataset. Using this normalized data and attribute weights, the forward computation of the WMDS Model determines the low-dimensional coordinates for each observation. The Visualization component displays the low-dimensional coordinates and the attribute weight values. The inverse computation then determines new attribute weights based on the analyst-defined low-dimensional coordinates. At this point, the pipeline always short-circuits to run the forward computation and determine (and then display in the Visualization) new low-dimensional coordinates for all observations; the CSV Data Controller is never needed beyond the data preprocessing steps since all observations are always displayed, meaning no further computation from the Data Controller is needed.

StarSPIRE [12] provides a scatterplot-like projection for queried text data. Thus, the analyst must perform a query followed by subsequent queries or interactions in order to pull documents into the visualization. To represent this process in our new pipeline, four pipeline

components are needed: a Text Data Controller, a Relevance Model, a Force-Directed Model, and a Visualization. The Text Data Controller initializes a set of extracted entities from the document set and ensures each document has an associated TF-IDF value for every entity. After providing references to the locations of the documents themselves and a set of entity weights (each initialized to $1/p$) the forward computation of the Relevance Model computes the relevance of all documents according to the current entity weights. Only the top n documents above a given threshold will be added to the visualization. Thus, the Relevance Model acts as a query filter for which documents are passed to the Force-Directed Model. The forward computation of the Force-Directed Model determines the low-dimensional coordinates of each document passed to it, using the same entity weights to place similar documents near each other. The Visualization then uses both the low-dimensional coordinates and the relevance (mapped to node sizes) to display the documents. Semantic interactions in StarSPIRE can cause the Force-Directed Model and the Relevance Model to learn new entity weights in their inverse computations. Thus, when the analyst manipulates the document positions or relevance values, new entity weights representing the analyst's interest are learned and then used in the forward computation to update the visualization accordingly.

Dis-Function [15] displays, among other views, a scatterplot of projected pairwise distances using PCA. An analyst is able to perform semantic interactions by providing new low-dimensional coordinates for observations, causing the system to learn new attribute weights for PCA and reprojecting the observations using these new weights. This behavior is quite similar to Andromeda, thereby using a PCA Model in place of Andromeda's WMDS Model in its pipeline.

Mamani et al. [78] propose a system similar to that of Paulovich et al., though the projection is based on local affine mappings rather than Laplacian. Still, the basic process of

sample first and project second remains the same in the forward computations. This means that the pipeline representation of this system uses a Local Affine Force Scheme Model in place of the Neighborhood Graph Model described above. However, the process of responding to semantic interaction also incorporates the inclusion of a new set of samples, thereby incorporating an inverse computation in both the Local Affine Force Scheme Model and the Random Sampling Model.

Paulovich et al. [87] present a Piecewise Laplacian projection system in which samples are drawn from a full dataset, control points are created for each sample, and a neighborhood graph is constructed for the full dataset. As an analyst manipulates the projection through semantic interactions, these control points and neighborhood graphs dynamically update. Using our new pipeline, we can model this process using a Sampling Model to perform the sampling step and a Neighborhood Graph Model to perform the projection. In the Sampling Model, the forward computation performs the initial sampling step, which then feeds into the forward computation of the Neighborhood Graph Model to specify the control points and project the data. Semantic interactions can be performed by directly manipulating the projection, causing the Neighborhood Graph Model to learn a new projection through its inverse computation and short-circuiting to rerun its forward computation. Since there is no semantic interaction defined that alters the Sampling Model, this model effectively has no inverse computation defined, highlighting the possibility for additional semantic interactions to be included in this type of system.

3.5 Using the Pipeline for New Visual Analytics Systems

In this section, we illustrate three visual analytics prototypes that have been developed using our new visual analytics pipeline to further explore the design space for semantic interaction. These prototypes handle different types of data (numerical and text) and alter similar Models to create distinct Visualizations and semantic interactions therein. Each of the prototypes are discussed in the following format:

- **Motivation:** We begin by motivating the creation of the prototype, describing why such a system is useful and what we could learn from it.
- **Visualization and Semantic Interactions:** We describe the Visualization developed and the semantic interactions enabled therein to provide context for the various pipeline components.
- **Pipeline:** We discuss how the given Visualization and semantic interactions are accomplished mathematically through our new visual analytics pipeline. Since we define the Visualization previously, we effectively separate the discussion of this pipeline component from the others to improve clarity.

3.5.1 Cosmos

Motivation

The Cosmos pipeline was created to explore how to incorporate the Relevance Model from StarSPIRE [12] with a stricter notion of similarity than a force-directed layout, such as is accomplished in Andromeda's WMDS Model [102]. With these two models, analysts can

query for specific terms or documents in the dataset, view the raw text from a document, manipulate a document's relevance, and directly manipulate the projection of the documents. Further details on a version of Cosmos that includes a third Model as well as a different Data Controller are provided in Chapter 6.

Visualization and Semantic Interactions

As shown at the bottom of Figure 3.4, the Cosmos Visualization consists of two panels. While the left panel is an interactive WMDS projection of the documents, the right panel displays the details for a single selected document. Unlike in Andromeda, the WMDS projection is initially empty, requiring the analyst to perform a search to bring documents into the Visualization. After documents are placed on the screen, their relevance calculations are mapped to the sizes of the projected observations. The analyst can then use an array of interactions to manipulate the Visualization. For example, double-clicking an observation populates the panel to the right of this projection with information specific to the corresponding document. This includes an interactive relevance slider, the label of the projected observation, and the raw text of a document and associated notes. The analyst also has the ability to remove a document from the Visualization by clicking a button on this panel.

As in Andromeda, the analyst can perform the semantic interaction of OLI in Cosmos by clicking and dragging documents of interest in specific locations (to denote their desired similarity/dissimilarity) and clicking an "Update Layout" button. This triggers a recalculation of attribute weights using only the low-dimensional observations the analyst interacted with. However, Andromeda stops there and reprojects the entire dataset to create a new Visualization; Cosmos continues its interpretation of this interaction by automatically performing a query for more documents on behalf of the analyst, guided by these new attribute weights. After combining the new documents with the old documents, the relevance of each document

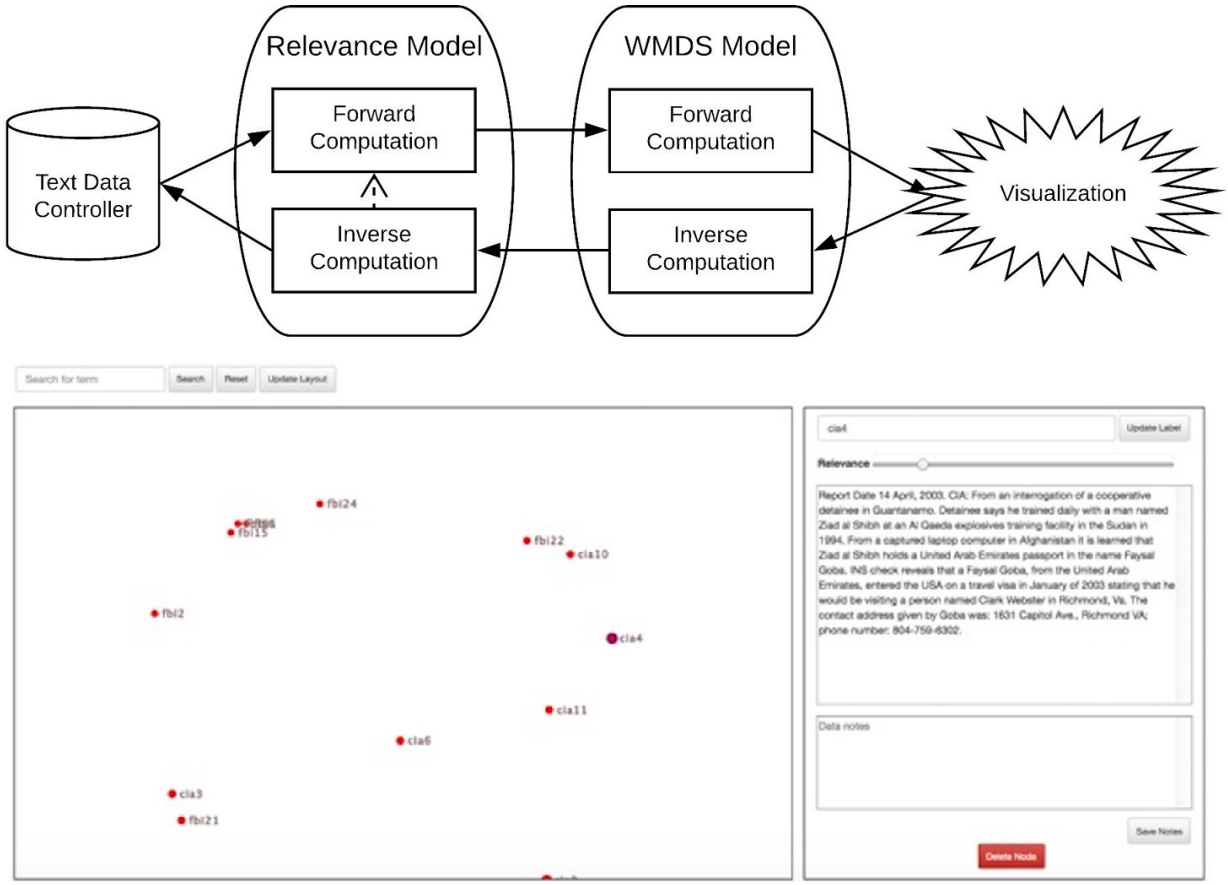

Figure 3.4: **(top)** Our pipeline representation of Cosmos consists of a Text Data Controller, Relevance Model, WMDS Model, and a Visualization. The Relevance and Similarity models each handle a different component of manipulating the data to create the Visualization. **(bottom)** The Cosmos interface allows analysts to interact with documents, manipulating their similarity and relevance throughout the exploration of the dataset.

is recalculated, and the data is reprojected to generate a new Visualization.

In addition to OLI, Cosmos affords an additional semantic interaction through its "Relevance" slider. This slider is available for a selected document, as seen in the details panel at the bottom of Figure 3.4. When this slider is manipulated by the analyst, new attribute weights are calculated which best estimate the analyst-defined relevance for the given document. If the relevance for the document is increased, then this interaction is interpreted as reflecting a document that the analysts likes and would want to see more of. Therefore,

this interaction also triggers an automatic query for more documents, which uses these new attribute weights. Otherwise, the pipeline simply continues its process by recalculating all document relevancies and reprojecting all documents to create a new Visualization.

Pipeline

The Cosmos pipeline is shown at the top of Figure 3.4. Note that this pipeline is similar to the StarSPIRE pipeline. In addition to replacing the Display Similarity Model with the WMDS Model, we have modified the Data Controller and Visualization as well. Each component of this pipeline is described below:

Text Data Controller: For this pipeline, we modified the Data Controller from Andromeda to work with text documents. To do so, we first assume that the uploaded CSV file contains the TF-IDF values for entities extracted from the document set. This assumption allows us to skip additional preprocessing steps to focus instead on the Models themselves and their influence on the Visualization.

Once uploaded, this Text Data Controller reads the data from the CSV file and preprocesses the data in the same manner as Andromeda's Data Controller to ensure each entity is treated equally by the Models. This is accomplished by normalizing each entity's TF-IDF values to be within a standard deviation of 1. Additionally, this Text Data Controller adds references to the flat files for each document, which are assumed to be in a single directory.

The final role of the Text Data Controller is to initialize a set of entity weights for the Relevance Model and Similarity Model to use, thus fulfilling the composability requirements for the forward computations in these Models. We initialize these weights to be $1/p$, where p is the number of extracted entities in the uploaded CSV file. These weights, along with the other document-related data, are sent along the pipeline to the Models.

Relevance Model: We drew inspiration from StarSPIRE [12] to create our Relevance Model. The Relevance Model uses the same set of attribute weights that the WMDS Model does (which is described next), but in a different manner. In the forward computation, this model computes the relevance of a document given a set of attribute weights as a linear combination of those weights and the document's TF-IDF values. This simple relevance calculation combined with a threshold determines which documents are passed on to the WMDS Model. That is, the Relevance Model acts as a filter that determines which documents are visualized. The forward computation has a matching inverse computation to calculate the entity weights that produce a relevance value for a given document.

Additionally, the Relevance Model is responsible for querying for new documents to display, whether the query was initiated by the analyst by searching for a term or automatically by Cosmos itself (e.g., after OLI or increasing a document's relevance). Using the entity weights, the Relevance Model finds the top n most relevant documents that are above the relevance threshold. This ensures that only highly relevant documents are displayed while also guaranteeing that the analyst will not be overwhelmed by too many documents appearing in the Visualization at once. If querying is not necessary to interpret the given interaction (e.g., when the relevance value for a document is decreased), then the Relevance Model simply short-circuits, allowing for immediate recalculation of the relevance values of all documents currently being displayed.

WMDS Model: The role of the WMDS Model is to spatialize documents according to their similarity based on a given set of attribute weights, just as is accomplished in Andromeda [102]. However, Cosmos relies on data passed from the Relevance Model to define which documents should be used as well as the entity weights. The forward computation uses these weights to project the high-dimensional data in the Visualization.

The WMDS Model also uses the same inverse computation defined by Andromeda, enabling

OLI interactions. This calculates the entity weights based on low-dimensional coordinates of documents in the Visualization. However, Cosmos also performs an automatic query based on these new entity weights. Since this automatic query always occurs after OLI and because the Relevance Model is responsible for such querying, the WMDS Model never short-circuits. After the Relevance Model recalculates document relevance values, the WMDS forward computation is run to determine new low-dimensional coordinates for all documents to be displayed in the Visualization.

3.5.2 A SIRIUS-Based System

Motivation

In Chapter 1, we noted that analysts often think about the observations and attributes in similar manners. In other words, there is a symmetry between how analysts analyze attributes and observations of a dataset. Therefore, there is a need to develop visual analytics systems that afford this symmetric thought process, leading us to develop SIRIUS (Symmetric Interactive Representations In a Unified System), a technique for symmetric visualizations and interactions for both observations and attributes. While further details of SIRIUS are described in Chapter 4, we focus on its pipeline representation of a prototype system using SIRIUS here.

Visualization and Semantic Interactions

The SIRIUS-based Visualization in Figure 3.5 consists of two main panels: a left panel for a projection of the observations and a right panel for the projection of the attributes. Both projections are WMDS projections, with node sizes and opacities reflecting the importance of the given observation or attribute. Both of these panels enable the same semantic interaction

of OLI previously described. However, instead of only updating one projection, this semantic interaction updates both projections in the Visualization.

Below these two main panels is a third panel that provides an interactive "Importance" slider that allows the analyst to define the importance of a selected observation or attribute. The associated raw data is also provided in the text field in this panel for the analyst's convenience. Manipulation of this "Importance" slider is a semantic interaction that triggers a recalculation of attribute weights and observation weights, thereby resulting in updates to both projections in the Visualization.

Pipeline

CSV Data Controller: The Data Controller used in this SIRIUS-based prototype is virtually the same as the one used in Andromeda. The main difference is that the Data Controller in this prototype must normalize both the original data as well as its transpose separately. This enables the projections to represent all observations and attributes without an artificial emphasis placed on any one attribute or observation due to naturally higher values (e.g., height vs. weight of a person).

Importance Model: We again drew inspiration from StarSPIRE's Relevance Model as it provides a simple method for the forward computation to calculate the importance (i.e., relevance) for any one observation or attribute using a linear combination of attribute or observation weights and the associated data for the given observation or attribute (respectively). However, these calculations also make it easy to translate the importance of attributes to the importance of observations and vice versa by expanding the importance calculation for a single observation or attribute to calculate the importance of all observations or all attributes at once. Thus, these importance calculations enable a recalculation of observation weights

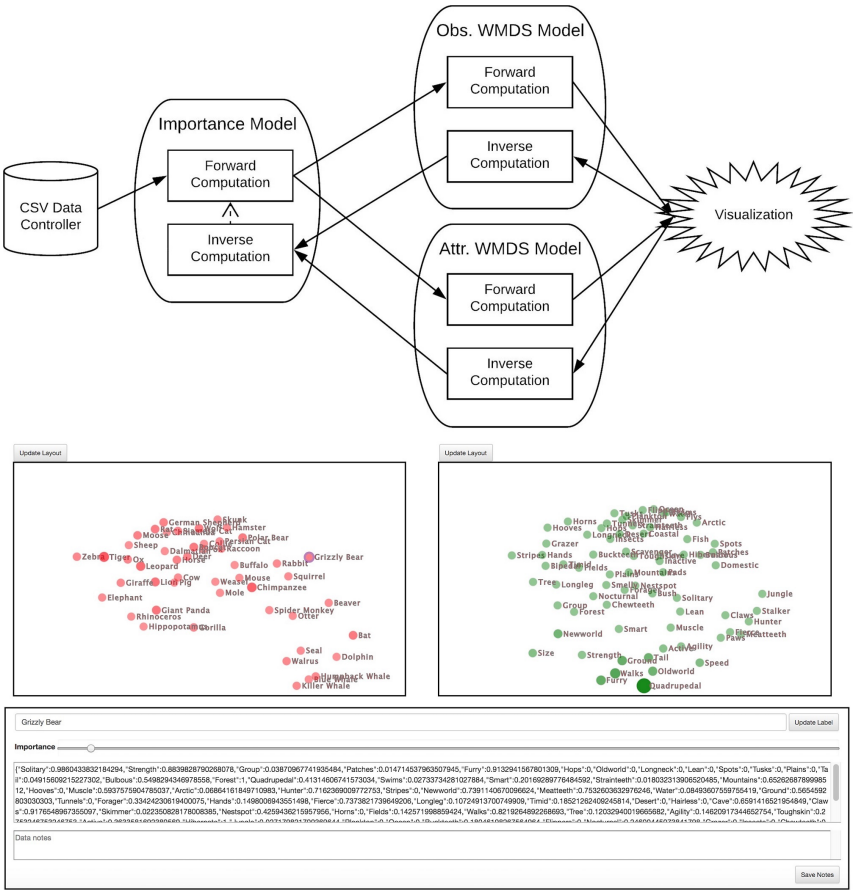

Figure 3.5: **(top)** Our pipeline representation of how our SIRIUS-based prototype produces the observation and attribute WMDS projections and how this system interprets semantic interactions therein using our new proposed pipeline. This is accomplished using a CSV Data Controller, Importance Model, two WMDS Models, and a Visualization. **(bottom)** This Visualization consists of two interconnected, interactive WMDS projections: one for the observations and one for the attributes of a high-dimensional dataset.

based on entity weights and vice versa. Since the WMDS Models (discussed next) use the same set of weights, this means that both projections update based on a single interaction in either projection.

To enable semantic interactions, the Importance Model's inverse computations begin when the analyst manipulates the "Importance" slider. This triggers an inverse calculation of the weights that produce the analyst-defined "importance" value using equations similar to those used in the Relevance Model's inverse computation from Cosmos. For example, if the analyst

manipulates the "importance" value for an attribute, then the observation weights to produce that "importance" value are calculated using one of these inverse computations. However, these new weights are then used to determine new attribute weights. To enable more insights for attribute similarities/correlations, these attribute weights from the inverse computation are then used to recalculate new observation weights in the forward computation. These final sets of weights are then used in the WMDS Models to reproject the data in the Visualization.

The Importance Model performs a similar set of calculations on OLI. For example, if OLI is performed in the observation panel, then a new set of attribute weights are determined in the WMDS Model. However, to translate this change to changes in the attribute projection as well, new observation weights must be determined. As a result, both new sets of weights are passed to the WMDS Models to determine the new positions of the nodes in both panels.

It is important to note that just as with Andromeda, this prototype assumes that all observations and attributes are used from the beginning. Since no querying for new data is performed, this system never needs to communicate with the CSV Data Controller again, causing the Importance Model to always short-circuit.

WMDS Models: As seen in Figure 3.5, this SIRIUS-based prototype consists of two WMDS Models: one for the projection of observations and one for the projection of attributes. The Observation WMDS Model uses the normalized form of the original dataset and the same attribute weights used by the Relevance Model to determine the low-dimensional coordinates for each observation using the same WMDS equation from Andromeda and Cosmos. Similarly, the Attribute WMDS Model uses the normalized form of the transposed dataset and the same observation weights used by the Relevance Model to determine the low-dimensional coordinates for each attribute.

Each of these WMDS Models enable OLI separately. That is, OLI can only be performed

on one panel at a time. Then, an inverse WMDS computation similar to the computation described in [102] is used to calculate a new set of weights. These weights are then passed to the Importance Model to enable updates in both panels.

3.5.3 A Cluster-Based Visualization

Motivation

We have also begun investigating how the introduction of explicit clustering assignments affect the ways in which analysts perceive and interact with projections [126]. The technique itself can make use of a variety of layout and clustering techniques, but the following implementation describes an instantiation using a force-directed layout for similarity projection and k-means clustering on the projection to automatically group similar observations. Analysts directly interact with the projection using semantic interactions to alter clustering assignments of the observations in order to manipulate the underlying mathematical models.

Visualization and Semantic Interactions

The Visualization for this system (bottom of Figure 3.6) simply consists of a large projection space accompanied by a column of attribute weights. Individual observations are still rendered as labeled nodes in the display, but are grouped by saturated convex hulls. The distance between pairs of observations is a weighted dissimilarity computation, in which the resting length of each link corresponds to the difference between the observation endpoints across all attributes.

Once again, analysts can perform OLI interactions on the observations. However, these semantic interactions only affect the mathematical models when an observation has been

reclassified into a new cluster (i.e., manipulating the distance between observations within a cluster has no effect on the learned weights). When an analyst adds an observation to a cluster or removes an observation from a cluster (or both), the attribute weights are recalculated based on a dissimilarity measurement between the relocated observation and the centroid(s) of the involved cluster(s). After this computation, the resting length of each link is recalculated based on the new attribute weights. As a result, additional observations may reclassify themselves as the force-directed layout repositions the observations in the projection.

Pipeline

CSV Data Controller: The Data Controller used in this system is identical to that used in Andromeda: numerical high-dimensional data is simply read in from a CSV file and normalized, and attribute weights are initialized to equal values.

Dissimilarity Model: The forward computation of the Dissimilarity Model computes a distance between each pair of observations, taking into account both the differences between the attributes values and the weights that have been learned for those attributes. This dissimilarity matrix is then passed to the Force-Directed Model to be rendered. The inverse computation aims to understand why the analyst decided that this observation does not belong to its original assigned cluster and/or better belongs to its analyst-assigned cluster. This is accomplished by computing a distance between the observation and the involved cluster centroid(s), ranking the attributes based on dissimilarity, and then updating the attribute weights accordingly.

Force-Directed Model: After a distance has been computed for every observation pair, these distances are loaded into a force-directed node-link visualization. The force-directed

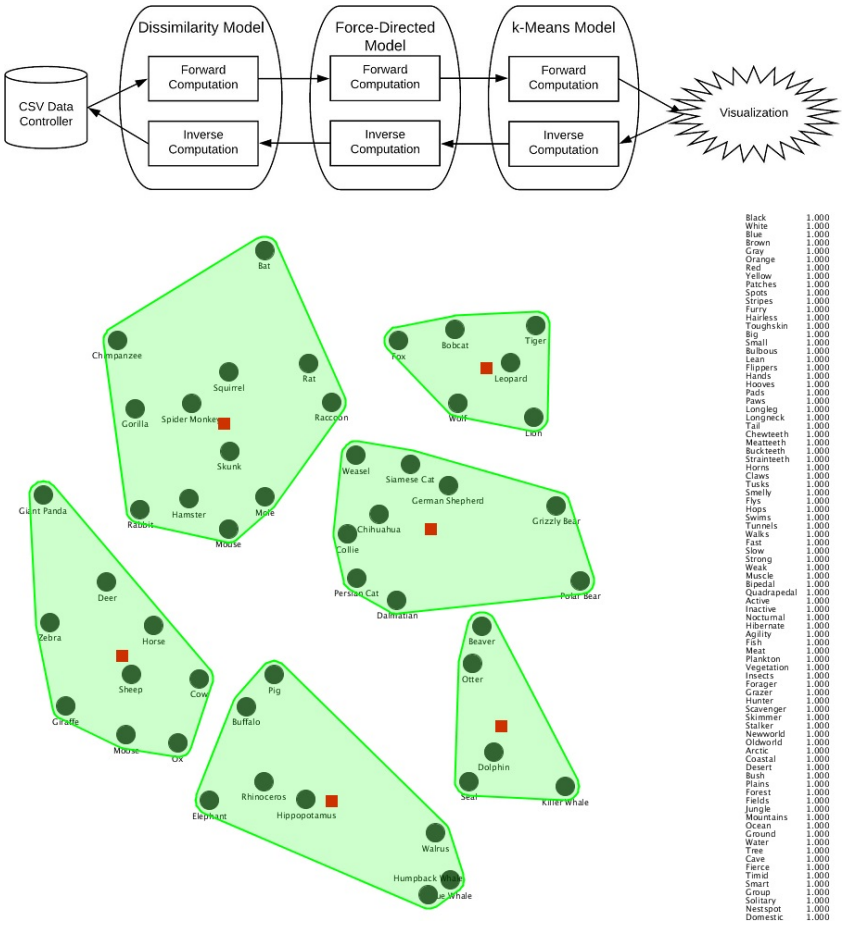

Figure 3.6: **(top)** Our pipeline representation for how the cluster-based visualization by Wenskovitch and North [126] is created and semantic interactions therein are interpreted. This is accomplished using a CSV Data Controller, Dissimilarity Model, Force-Directed Model, k-Means Model, and Visualization. **(bottom)** The clustering interface allows analysts to explore related groups of observations depending on the learned attribute weights.

graph then stabilizes to a low-energy layout. There is no inverse computation for this model.

k-Means Model: Clusters are computed continuously using a modified k-means algorithm in the forward computation of the k-Means Model. This computation has been altered from traditional k-means to include a maximum cluster radius that allows some nodes to exist external to all clusters. As the force-directed graph reaches a stable layout, individual observations may transition into and out of clusters as they move closer to and further from each cluster centroid. The inverse computation of this model detects analyst-initiated changes in

observation clustering assignments, and it passes the old and new cluster information to the next inverse computation.

3.6 Discussion

In this section, we discuss the implications of our new pipeline that is capable of capturing the complexity of semantic interactions in visual analytics systems. This discussion includes how this pipeline highlights the various semantic interaction possibilities in any given visual analytics system, the ability to leverage this pipeline for rapid prototyping, and the limitations of this pipeline.

3.6.1 Exploring the Design Space of Semantic Interaction

With the greater emphasis placed on the mathematical models in our proposed new pipeline for visual analytics systems, opportunities for semantic interaction are highlighted. This is due to the fact that every Model should have both a forward computation and an inverse computation. If a given pipeline does not have an inverse computation for a Model, such as in the Sampling Model in the Piecewise Laplacian projection system [87], then perhaps there is a missed opportunity for implementing a semantic interaction. Even for those that already have inverse computations, there may still be the potential to implement an additional or alternative inverse computation for the same Model.

For example, the forward computation in Andromeda's WMDS Model uses two parameters (the high-dimensional data and a set of attribute weights) to produce a single output (low-dimensional coordinates). However, the "inverse WMDS" computation described only computes new attribute weights given new low-dimensional coordinates, thereby assuming

the high-dimensional data is static. However, what if instead the assumption was that the attribute weights were static and new high-dimensional data for an undefined observation was desired? Such an interaction may be triggered by the analyst clicking in an empty space of the projection not already occupied by an observation, effectively interpolating what attributes a high-dimensional observation would have if it were projected in that location.

Additionally, our previous research has identified a variety of ways to combine a similarity-based Model with a clustering Model to create different types of projections [127]. With this multitude of pipelines possible, there are naturally many methods of enabling semantic interactions with just two Models. Thus, in cases such as these, our newly-proposed pipeline can help further explore the design space of semantic interaction by highlighting the numerous possibilities, even when there are few Models involved.

3.6.2 Rapid Prototyping to Explore Design Trade-Offs

To quickly and efficiently explore the design space of semantic interaction, the ability to rapidly prototype several techniques from the visual analytics literature, and augment them with semantic interaction, would be immensely helpful. Trade-offs in different implementations may imply different Models being used or perhaps the same ones being altered to produce different results. For example, Cosmos is very similar in appearance to Andromeda, yet functions more like StarSPIRE (as evidenced by the similarities in their pipeline representations). The SIRIUS-based prototype and the visualization by Wenskovitch and North are both similar to Cosmos in their own manners as well, yet these systems operate in distinctly different manners by adding additional Models to the pipeline.

However, the current issue in experimenting with these kinds of trade-offs is that many visual analytics systems are created such that changing one Model for another is difficult;

too often, the program structure for the given system is heavily reliant on the specific Models being used. By defining the pipeline components and creating a pipeline such as the ones that we present here, we assert that our new visual analytics pipeline can help promote more modularized code. This is because every pipeline component has composability requirements, which are described by the arrows between the pipeline components. By structuring the program for the system in this manner, the composability requirements help enforce modular code design. This modularity then makes interchanging different Models– and even different Visualizations and Data Controllers– trivial, thereby making exploring different areas of the design space of semantic interaction even easier.

For example, it may be apparent from Figure 3.4 and Figure 3.5 that Cosmos and the SIRIUS-based prototype have very similar-looking Visualizations. This is because the Cosmos pipeline– including its Models and Visualization– were all leveraged and adapted to enable the SIRIUS-based prototype. In fact, we have been able to separate each pipeline component to the point that we are able to interchange Cosmos and the SIRIUS-based prototype at will.

3.6.3 Limitations

Despite the power and flexibility of our proposed new visual analytics pipeline for semantic interaction, it is not without limitations. We briefly address several of these limitations here.

Requirements Limitations

The primary limitation of our semantic interaction pipeline lies in the requirement of providing an inverse computation for each forward computation. We assert this requirement as essential for enabling semantic interaction, yet we provide no guidance for how to determine

what such an inverse computation should be. That is, the inverse computation can be mathematically rigorous, heuristic, or probabilistic, but the creation of the inverse computation lies with the Model creator.

Similarly, the pipeline requires Models to be composable with other pipeline components, but we provide limited instructions for defining this composability. For example, if the Text Data Controller for Cosmos were altered to use dynamic document sets, then at some point before any of the data preprocessing steps could be performed, the TF-IDF values would need to be computed on the fly. This would allow the other pipeline components to remain unchanged. If instead the TF-IDF values were not computed and no data preprocessing steps occurred in the Data Controller, the responsibility for doing so (to maintain data composability with the WMDS Model) would either fall to the Relevance Model or to a new Model that would rest between the Data Controller and Relevance Model. How such an alteration might be accomplished is described in the version of Comsos described in Chapter 6.

Limitations of Pipeline Components

Another potential limitation is that we define a Model to be comprised of any forward computation and at least one accompanying inverse computation. It may very well be that there are only a few different categories or types of such Models (e.g., data manipulation Models that work the raw data into a form usable to the Visualization, projection Models that determine the overall type of projection used in the Visualization, and other Models that seek to augment the Visualization with additional information). Such a categorization may be useful to define the nuanced differences between how Models may be used and which ones definitely should have inverse computations to interpret semantic interactions. However, we do not attempt to make any such categorization; instead, we focus on the overall pipeline structure to generate discussion and critical thinking regarding which Models *should* be

included, *how* the Models fit together to realize an interactive visual analytics system, and the various manners in which semantic interaction *can* be realized.

Limitations of the New Visual Analytics Systems

Rather than creating fully-featured systems, we use this pipeline to quickly and efficiently prototype visual analytics systems to explore the semantic interaction design space. As a result of this design decision, the prototypes that we implemented in Section 3.5 only support a limited number of semantic interactions. However, we argue that each of our prototypes can support additional semantic interactions with the addition of more Models or alterations of existing Models in each pipeline.

3.7 Conclusion

In this work, influenced by the Sensemaking Process described by Pirolli and Card [90] as well as the growing body of visual analytics systems that implement semantic interaction, we proposed three characteristics shared by semantic interaction applications: model composability, model inversion, and pipeline bidirectionality. From these characteristics, we proposed a new visual analytics pipeline that enables proper representation of the complexity involved in semantic interactions. This new pipeline is comprised of three main types of components: Data Controllers, Models (containing forward and inverse computations), and Visualizations.

We demonstrated the ability of our new pipeline to capture semantic interactions by providing pipeline representations of existing visual analytics systems. Then, we discussed pipeline representations for new visual analytics systmes, highlighting the extensibility of this new

pipeline new research in this area.

We also briefly discussed how this new pipeline may help further the exploration of the design space of semantic interaction and enable rapid prototyping of new visual analytics systems with semantic interaction. By rapidly prototyping such systems, researchers will be able to quickly create and study many alternative methods of semantic interaction. We intend to continue expanding on our own prototypes and conduct user studies to study how analysts perceive and use different visualizations and interactions. Such research may uncover which methods best support the analyst's sensemaking process and how to develop better visual analytics systems in the future.

Chapter 4

Modeling Symmetric Visualizations and Interactions

4.1 Introduction

Visualizing and interacting with high-dimensional data for exploratory data analysis is an open research area with many facets to explore. In this chapter, we focus on visual analytics techniques for high-dimensional data exploration that use dimension reduction to project 2D scatterplots of the data. Many existing techniques and interactions therein focus on **observation-centric** tasks that reveal relationships between observations as defined by their attributes, such as clustering tasks [5]. For example, with the animal dataset used throughout this chapter [71], analysts could investigate questions such as "Which attributes separate the *Tiger* and *Wolf* from the *Blue Whale* and *Dolphin*?" or "What other animals are similar to those animal groups?" Projection methods often define weight parameters on the attributes that enable analysts to assign different levels of **importance** to each attribute, thus enabling exploration of alternative observation projections [15, 102].

Likewise, there are complementary **attribute-centric** tasks that reveal relationships between attributes as defined by their observations, such as correlation tasks [5]. For example, a follow-up question might be "What other attributes are correlated with the attributes that separate these two groups?" This question is more difficult to answer with observation-

centric projections. Thus, other kinds of visualizations are often used, such as linked distribution plots and dynamic queries [3, 74, 77, 107, 113] or correlation matrices [104], creating asymmetry in how observation-centric and attribute-centric tasks are supported.

A natural **symmetry** between observation-centric and attribute-centric tasks in high-dimensional data can be defined as equivalent tasks on the data table or data matrix and its transpose (which swaps the observations and attributes). For example, this symmetry often arises in visualizations for text analytics. With a vector space model matrix, documents can be projected in terms of their word usage [4, 24, 41]. Alternatively, with the matrix transpose, words can be projected in terms of their usage in documents [16, 28, 86, 95].

We propose that this task symmetry between observations and attributes reflects a symmetry in the cognition of multidimensional data, and therefore is better supported by a symmetry between how the observations and attributes are visualized and interacted with. Such a symmetry would give analysts equal power to investigate both observations and attributes, using the same visual representations and interactions for both. Additionally, previous work has shown cognitive bias towards symmetric stimuli, as well as an association between asymmetry and disgust [43]. Based on this, we assert that asymmetric visualization and interaction design should increase cognitive load in comparison to symmetric designs as they require analysts to simultaneously interpret different approaches to observations and attributes.

To address this need for symmetry, we define a **symmetric dual projection** technique in which the projection of observations is defined by the attributes, and the projection of attributes is defined by the observations. We further define **interactions that connect** the observations and attributes between the symmetric projections, resulting in a manipulation of the projection of observations influencing the projection of the attributes and vice versa. These interactions reflect a notion of a deep connection between the observations and the attributes that mirrors the analyst's notion of how observations and attributes are

Figure 4.1: The initial, interactive symmetric dual projections of a multidimensional dataset using SIRIUS. Observations (animals) are projected in the left panel, while attributes (animal characteristics) are projected in the right panel. Both panels project similar items closer together based on a weighted high-dimensional distance function in which the weights reflect a conceptual notion of "importance." These weights are reflected by the node sizes and opacities in the opposing panel. For example, *Quadrupedal* has a higher weight in the left projection of animals, and *Tiger* has a slightly higher weight in the right projection of characteristics.

interconnected.

Specifically, our contributions in this work are:

1. Defining a technique called SIRIUS (Symmetric Interactive Representations in a Unified System) that models symmetric, interconnected projections of observations and attributes to directly address the lack of symmetry in current visual analytics techniques (detailed in Section 4.2).

2. Creating an implemented instantiation of SIRIUS using WMDS, as represented in Figure 4.1 (described in Section 4.3).

3. Demonstrating how SIRIUS allows analysts to gain insight on both observation-centric and attribute-centric tasks (shown in Section 4.4).

Table 4.1: A description of the commonly used variables and functions in the equations throughout this paper.

O	A	The original data matrix (observations; O) and its transpose (attributes; $A = O^T$), with the columns for each matrix normalized
n	p	The number of observations n or attributes p, as represented by the number of rows in O or A, respectively
O_i	A_i	The high-dimensional data for the i^{th} observation (O_i) or i^{th} attribute (A_i), as represented by the i^{th} row of O or A, respectively
$\hat{O}$	$\hat{A}$	The dimensionally reduced matrices derived from O and A. In SIRIUS, the dimensionally reduced space is 2D, enabling easy projection onto a computer screen
W_O	W_A	The observation weights (W_O) or attribute weights (W_A), each represented as a single vector
W_{O_i}	W_{A_i}	The i^{th} observation weight (W_{O_i}) or attribute weight (W_{A_i})
$wDist_O$	$wDist_A$	A matrix of high-dimensional pairwise weighted distances between observations ($wDist_O$) or attributes ($wDist_A$)
$hDist_O$	$hDist_A$	The weighted high-dimensional distance function to calculate similarities between observations ($hDist_O$) or attributes ($hDist_A$). In SIRIUS, we use weighted Euclidean distance for each of these distance functions.
$lDist$		The low-dimensional distance function to calculate the pairwise distances between rows of a dimensionally reduced matrix. The same function is used for both observations and attributes, which further promotes symmetry in the presentation, interaction, and interpretation of both projections. In SIRIUS, we use 2D Euclidean distance.

4.2 A Symmetric, Interactive Projection Technique

To enable exploratory data analysis with high-dimensional data using symmetric visualization and interaction techniques between the attributes and observations, we designed a new technique called SIRIUS. We assert that in SIRIUS the analyst must be able to:

Goal 1: View similarity-based relationships between observations and similarity-based relationships between attributes of high-dimensional data.

Goal 2: Explore different projections of the data by altering the importance of specific observations or attributes.

Goal 3: Understand how importances of observations affect the importances of attributes and vice versa.

Each of these goals are further described in the following subsections. To clearly exemplify these concepts, we use a subset of the animal dataset from Lampert et al. [71] containing 13

observations and 13 attributes. These observations and attributes were selected to provide a clear and intuitive example (e.g., by excluding attributes like *Newworld*) while ensuring variance (e.g., by only including *Horse* and not *Zebra*). More complex examples are presented in Section 4.4. Variables and functions are defined in Table 4.1.

4.2.1 Goal 1: Visualize Similarity-Based Relationships

This first goal combines the tasks of seeing similarities between observations and seeing similarities between attributes. For example, in the animal dataset where the observations are animals, the *German Shepherd* is similar to the *Wolf* but not very similar to the *Elephant*. These similarities between the observations can be visually represented via a projection method. There are many different methods of doing so, including but not limited to PCA [58, 89, 131], t-SNE [120], and MDS [68, 69, 116]. We generalize the lower-dimension projection of the observations to be the output of the function $project_O$:

$$\hat{O} = project_O(wDist_O)$$

Similarly, an additional projection method should enable analysts to see similarities between attributes. For example, in the animal dataset, the attribute *Strength* is more similar to the attribute *Size* than to *Grazer*. We generalize the lower-dimension projection of the attributes to be the output of a function, $project_A$, as follows:

$$\hat{A} = project_A(wDist_A)$$

As noted in our evaluation of the Data Context Map [23] in Section 2.3.3, visualizing the observations and the attributes in a single projection necessarily distorts the projection

of the similarities between observations, attributes, or both. Thus, the observations and attributes must be projected into separate spaces to maintain an accurate representation of their similarities. This means we need two similarity-based projections: one for observations ($project_O$) and one for attributes ($project_A$).

Furthermore, to reduce confusion between the two projections, we propose that $project_O$ and $project_A$ should produce projections that are understood in the same manner by the analyst. This is best reflected by a symmetry in the manner in which observations and attributes are visualized (and later interacted with). The easiest way to accomplish this is to use the same projection method for both $project_O$ and $project_A$ (e.g., MDS), but they can differ if the analyst perceives them in the same manner (e.g., MDS and PCA).

4.2.2 Goal 2: Explore Different Projections

Exploring different projections stems from the need to gain new insights based on domain knowledge or a hypothesis, or for general exploratory analysis. These insights can be gained by redefining the similarity between observations or attributes, which implies interaction that alters at least one projection. Such interaction can be accomplished by either altering the parameters that generate the high-dimensional pairwise distances or by directly manipulating the projection.

Explore Projections of Observations

To exemplify what is meant by each of these interaction methods, we first focus on $project_O$. From a projection of the observations, the analyst may want to understand how placing more importance on different attributes affects the similarities between the observations. For example, the analyst may want to investigate how animals differ based on their *Water*

attribute. By placing more importance on this one attribute, animals like *Otter* should be reprojected much closer to the *Dolphin* and *Blue Whale* and farther away from the *Siamese Cat*.

Alternatively, the analyst may want to see how altering similarities between observations influences the level of importance that should be placed on each attribute. For example, if the analyst drags nodes for *Dolphin* and *Blue Whale* in one corner of the animal projection (to denote their desired similarity) and *Elephant* to the opposite corner (to denote its desired difference from the other two animals), the technique should reflect a higher level of importance for attributes that describe the differences between these two groups. In this case, the *Walks* and *Grazer* attributes describe the differences between these two groups, implying they should be given higher levels of importance.

In both of these types of interactions within the observation projection, the importances of attributes are altered. This reflects the fact that observations are understood based on their attributes. Given that the importance of attributes can be represented by weights on the attributes, the similarity between two observations, O_i and O_j, can be generalized with the following weighted high-dimensional distance function:

$$wDist_{O_{i,j}} = hDist_O(W_A, O_i, O_j)$$

There are many weighted distance functions that could be used here, such as weighted variants of Euclidean distance, Manhattan distance, cosine distance, Gower distance [50], Pearson coefficient [88], and Bray-Curtis dissimilarity metric [14, 16]. Which distance function to use is often determined by the tasks supported and data used, however it must be compatible with the chosen projection method and desired outcome of the projection itself. For example, while PCA can be used to accomplish Goal 1, it emphasizes variance rather

than strictly pairwise distances. Thus, using a weighted Euclidean distance with PCA may not produce the desired results.

Explore Projections of Attributes

To maintain the desired symmetry with the observations, the same interactions are enabled in the projection of the attributes. Thus, the analyst can alter the importance of a specific observation to understand how this affects the similarities between attributes. For example, increasing the importance for *Cow* should result in attributes like *Walks*, *Size*, and *Strength* being placed closer together but far away from *Stripes*. Additionally, the analyst can alter the similarities between attributes to understand how the importances of observations are affected. For instance, if the analyst drags nodes for *Grazer* and *Size* to one corner of the projection and *Water* to another, *Horse* describes the differences between these two groups of attributes and should therefore receive a higher weight to denote its increased importance.

In both of these types of interactions within the attribute projection, the importances of observations are altered. This reflects the fact that attributes are understood based on the observations. Since the importance of observations can be represented by weights on the observations, the similarity between two attributes, A_i and A_j, can be generalized with the following weighted high-dimensional distance function:

$$wDist_{A_{i,j}} = hDist_A(W_O, A_i, A_j)$$

Note the symmetry between this equation and the equation for weighted high-dimensional distances between observations.

4.2.3 Goal 3: Relate Importances to Each Other

As stated previously, the analyst understands the observations based on their attributes and vice versa. This hints at a connectedness between the observations and the attributes themselves. In the equations in the previous subsection, this connectedness is initiated by the importance of the attributes affecting the similarity of the observations and the importance of the observations affecting the similarity of the attributes. However, the notion of interconnectedness goes beyond these equations: attributes that are given more importance indicate which observations should be given more importance and vice versa.

As a common example of this, a keyword search for relevant documents results in those keywords (i.e., attributes of documents) being given a high level of importance. This means that documents that are better described by those keywords are, in turn, more important as well, and hence should be returned in the query results. This example implies that observations that have higher values (e.g., keyword frequencies) for an attribute are more important, and vice versa.

Using the animal dataset again to exemplify this, increasing the importance of the *Water* attribute should result in animals like *Dolphin*, *Blue Whale*, and *Otter* also being given a high level of importance in addition to reprojecting their nodes closer together. *Siamese Cat*, on the other hand, should have a low level of importance as it is not well described by the *Water* attribute. However, these updated importances for the different animals also denote which attributes should be considered important, beyond the single *Water* attribute that was interacted with. This means that given important animals like *Dolphin*, *Blue Whale*, and *Otter*, attributes like *Speed*, *Active*, and *Smart* are also more important than other attributes that do not describe these animals as well.

Therefore, the importance of attributes should affect the importance of observations and vice

versa to reflect the analyst's notion of the interconnectedness between the attributes and the observations. This can be accomplished with the following equations, where $Importance_O$ and $Importance_A$ compute the importance for one observation or attribute, respectively:

$$W_{O_i} = Importance_O(O_i, W_A)$$
$$W_{A_i} = Importance_A(A_i, W_O)$$

This interconnectedness between the importances of the observations and the importances of the attributes enables a new visual analytics technique that treats the observations and the attributes in a symmetric manner while enabling rich interaction between both projections.

Given this interconnectedness in SIRIUS, the weights that reflect the importances of observations and attributes are crucial components of the technique. They are used to project the data via weighted high-dimensional distance functions to explore different projections and to relate the importances of observations and attributes to each other. We demonstrate how this can be accomplished in Section 4.3.

4.3 An Implementation of SIRIUS

The generalized SIRIUS technique above supports a design space of possible projection methods, distance functions, and interaction methods that can be inserted. We present a particular implementation of SIRIUS that addresses the goals of Section 4.2 in the following manner:

1. Using WMDS for both $project_O$ and $project_A$, and weighted Euclidean distance for $hDist_O$ and $hDist_A$, to create both a projection of the observations and a projection

of the attributes.

2. Using the notions of parametric interactions (which we call PaI) and OLI as described by Self et al. [102] to manipulate the weights W_O and W_A of the weighted Euclidean distance function. Since we use the concept of OLI in both the observation projection and the attribute projection, we have renamed OLI as projection interactions (PrI) to reduce any potential confusion[1].

3. Relating the importances of observations and attributes to each other by defining $Importance_O$ and $Importance_A$ as a dot product between the original data matrix or its transpose and a set of attribute weights or observations weights (respectively). Ultimately, these equations for importance result in interconnecting both projections by using an interaction in one projection to alter both projections.

We made these particular design choices based on previous research in visualizing and interacting with high-dimensional data [12, 37, 41, 102, 126]. However, these are not strict constraints; any distance function, projection method, or interaction method that properly addresses the goals defined in Section 4.2 may be used in place of our choices here.

The following subsections describe how we accomplished each goal in detail, using the same subset of the animal dataset from the previous section to exemplify each concept.

[1]Briefly, PaI refers to direct alteration of a model parameter via some control mechanism, such as a slider or textbox. An analyst may update that parameter with a precise value. In contrast, PrI *learns* a set of model parameters based on *an interpretation* of analyst alteration of the projection itself.

Figure 4.2: An initial projection of a subset of the animal dataset using SIRIUS, which maps "importance" to node size and opacity to provide a deeper semantic connection between observations and attributes. This allows analysts to determine at a glance which animals best describe the attribute projection (from the observation panel) and which attributes best describe the animal projection (from the attribute panel).

4.3.1 Goal 1: Visualize Similarity-Based Relationships

Observation Projection

Using SIRIUS, we designed the left projection to depict similarities between observations. To visualize these similarities, we use weighted Euclidean distance in WMDS [37], as defined by the following equation:

$$\hat{O} = \underset{\hat{O}_1,\ldots,\hat{O}_n}{\arg\min} \sum_{i=1}^{n-1} \sum_{j>i}^{n} \Big(lDist(\hat{O}_i, \hat{O}_j) - hDist_O(W_A, O_i, O_j) \Big)^2 \tag{4.1}$$

Before the data can be projected, some preprocessing must occur. To overcome any potential distortions in the projection caused by attribute values on different scales, O is z-score normalized prior to visualization.

Additionally, a set of attribute weights, W_A, that reflect the importances of each of the attributes must be defined before the high-dimensional distance matrix can be calculated. W_A has two constraints: $0 \leq W_{A_i} \leq 1$ and $\sum_i W_{A_i} = 1$. Thus, weights are interpreted

as proportions of the analyst's interest in each attribute (i.e., its level of importance). Although this is a minor point in the initialization of our implementation, it greatly affects the interaction methods, as discussed in the following subsections. For now, we will say that W_A is initialized by determining the importances of the attributes, with details discussed in Section 4.3.3. The initial observation projection is shown in the left panel in Figure 4.2, which accurately shows that *German Shepherd* and *Wolf* are more similar to each other than to the *Elephant* as desired.

Attribute Projection

Given the desire for symmetry between the observations and the attributes, a second projection is used to depict the similarities between the attributes. It also uses weighted Euclidean distance in WMDS, as defined in the following equation:

$$\hat{A} = \underset{\hat{A}_1,\ldots,\hat{A}_p}{\arg\min} \sum_{i=1}^{p-1} \sum_{j>i}^{p} \Big(lDist(\hat{A}_i, \hat{A}_j) - hDist_A(W_O, A_i, A_j) \Big)^2 \tag{4.2}$$

Again, the data must be preprocessed before it can be projected. This includes z-score normalizing A, as well as initializing an observation weight vector, W_O. These weights reflect the importances of each observation and must hold to the same two constraints as W_A. We again leave detailed discussion for how we initialize this set of weights for Section 4.3.3. This initial projection is shown in the right panel of Figure 4.2, which demonstrates that *Strength* and *Size* are more similar to each other than to *Grazer*, as desired.

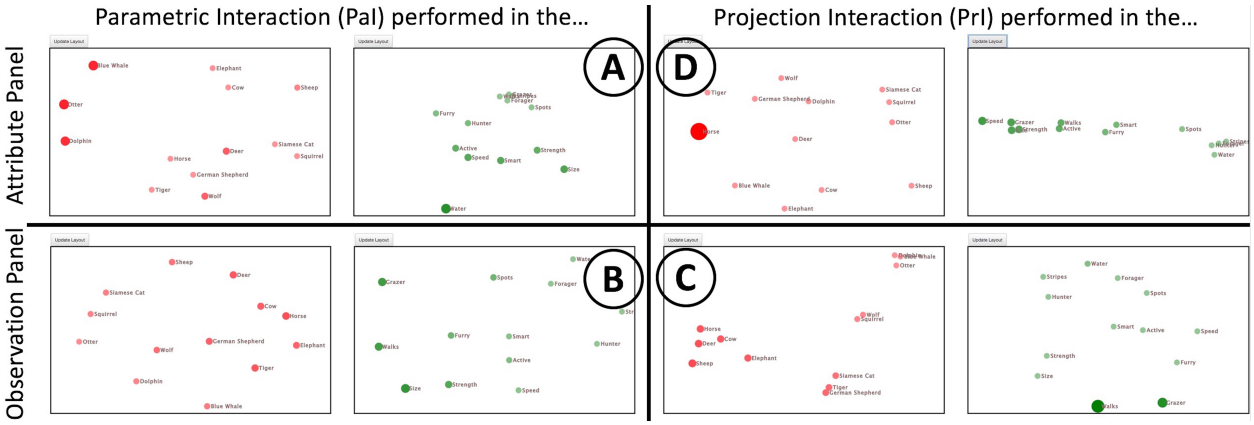

Figure 4.3: The results of two examples of PaI and two examples of PrI described in Section 4.3 with Figure 4.2 as the initial projection of the data and continuing to map "importance" to node size and opacity: **A** PaI performed on the *Water* attribute; **B** PaI performed on the *Cow* observation; **C** PrI performed by dragging the *Dolphin* and *Blue Whale* observations into one corner and the *Elephant* observation into the opposite corner; and **D** PrI performed by dragging the *Grazer* and *Size* attributes into one corner and the *Water* attribute into the opposite corner.

4.3.2 Goal 2: Explore Different Projections

In our implementation of SIRIUS, we use weighted Euclidean distance to define both the similarities between observations and the similarities between attributes. Thus, exploration of different projections is accomplished by manipulating the associated weights, thereby allowing the use of PaI and PrI as described by Self et al. [102] to enable rich interactions.

Parametric Interaction (PaI)

To explore how attribute importances affect similarities between observations, PaI is enabled via an "Importance" slider, which is accessible by clicking a node in the attribute projection. The analyst can alter the attribute weight (i.e., importance) by manipulating the slider. During this interaction, all attribute weights are re-normalized to adhere to the previously described sum-to-1 and 0-to-1 constraints. The change in attribute weights is reflected in up-

dates to the size and opacity of the attribute nodes, which visually reflects their importance. All observations are then reprojected using Equation 4.1 with the updated attribute weights to show the effect of the analyst's interaction. An example of PaI on the *Water* attribute is depicted in Figure 4.3-A, which pulls *Otter* closer to the *Dolphin* and *Blue Whale* than to the *Siamese Cat*.

Symmetrically, PaI is also used to explore how observation importance affects similarities between attributes via the same "Importance" slider. The analyst can click an observation and adjust the slider to alter the weight (i.e., importance) for the given observation, and all observation weights are re-normalized. The size and opacity of the observation nodes are updated to reflect their new weights. All attributes are then reprojected using the updated observation weights in Equation 4.2. An example of PaI on the *Cow* observation is shown in Figure 4.3-B, which correctly results in the *Walks*, *Size*, and *Strength* attributes being placed close together but far apart from the *Stripes* attribute to reflect their similarity in "cow-ness."

Projection Interaction (PrI)

To explore how observation similarities affect attribute importances, analysts can use PrI in the observation projection. This is accomplished by directly manipulating the observation projection via clicking and dragging observation nodes of interest to redefine their relative similarities. Once the analyst is done manipulating the projection, an "Update Layout" button above the observation panel is clicked. This triggers a semi-supervised re-learning of the attribute weights using *only* the observation nodes the analyst interacted with, $\hat{O}^*$, in

the following optimization, essentially inverting the WMDS process in Equation 4.1:

$$W_A = \underset{W_{A_1},\ldots,W_{A_p}}{\arg\min} \sum_{i\in\hat{O}^*} \sum_{j\in\hat{O}^*} \Big(lDist(\hat{O}_i^*, \hat{O}_j^*) - hDist_O(W_A, O_i, O_j) \Big)^2 \tag{4.3}$$

This optimization must also adhere to the sum to 1 and 0 to 1 constraints for W_A. From the new attribute weights, the attribute node sizes and opacities are updated to reflect these new levels of importance, and Equation 4.1 is re-executed to reproject all observations. For example, as depicted in Figure 4.3-C, dragging the nodes for *Dolphin* and *Blue Whale* to one corner of the projection and *Elephant* to the opposite corner, results in an increase in the importances of the *Walks* and *Grazer* attributes, which distinguish these two groups of animals.

Symmetrically, the projection of the attributes permits exploring how attribute similarities affect observation importances. This is accomplished via PrI by dragging attribute nodes and clicking the "Update Layout" button above the attribute panel. This triggers a very similar algorithm that effectively inverts the WMDS process in Equation 4.2 using *only* the attribute nodes the analyst interacted with, $\hat{A}^*$, and following the same 0 to 1 and sum to 1 constraints for W_O:

$$W_O = \underset{W_{O_1},\ldots,W_{O_n}}{\arg\min} \sum_{i\in\hat{A}^*} \sum_{j\in\hat{A}^*} \Big(lDist(\hat{A}_i^*, \hat{A}_j^*) - hDist_A(W_O, A_i, A_j) \Big)^2 \tag{4.4}$$

Using the new observation weights, the observation node sizes and opacities are updated to reflect these new levels of importance, and Equation 4.2 is re-executed to reproject all attributes. To demonstrate, Figure 4.3-D shows that dragging the nodes for *Grazer* and *Size* far away from *Water* results in an increase in the importance for *Horse*.

4.3.3 Goal 3: Relate Importances to Each Other

As our SIRIUS-based implementation has been described thus far, it is somewhat similar to Andromeda [102]. The main difference is that instead of listing the attributes, an attribute projection is provided alongside the observation projection. However, we now introduce two equations to interconnect observation importances and attribute importances: $W_{O_i} = O_i \bullet W_A$ and $W_{A_i} = A_i \bullet W_O$. These equations can be more generally expressed as:

$$W_O = O \bullet W_A \tag{4.5}$$

$$W_A = A \bullet W_O \tag{4.6}$$

Both of these equations are used in initializing the projections as well as both PaI and PrI, as shown in Figure 4.4, thereby interconnecting the projections of the observations and attributes together. This interconnectedness between observation importances and attribute importances has crucial implications in revealing new relationships and affording additional insights by providing methods to alter node size, opacity, and position for both observations and attributes after any interaction. Thus, analysts are afforded insights such as correlations between observations or between attributes at a glance during any point of analysis.

Our use of these equations is loosely based on simple approaches to relevance computations in information retrieval and recommender systems, such as the HITS (Hubs and Authorities on the Internet) Algorithm [64], which underlies Google's PageRank query technique [85]. However, a major difference is that HITS iterates over these two equations until convergence, whereas our implementation of SIRIUS only iterates once to enable explorations of alternative projections via PaI and PrI. However, it might be interesting to iterate until convergence during initialization.

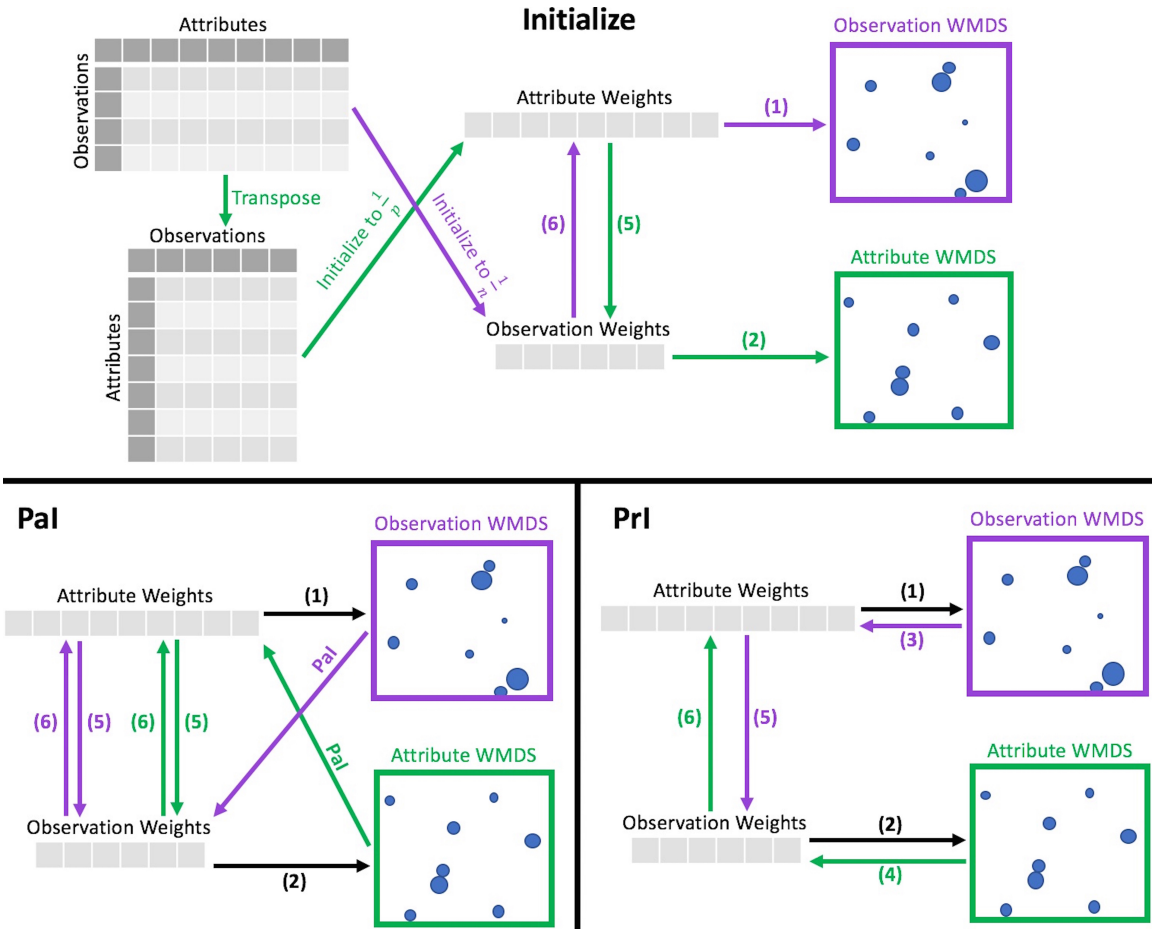

Figure 4.4: A flowchart depicting how Equation 4.5 and Equation 4.6 are used in conjunction with Equations 4.1–4.4 on initialization or when PaI or PrI occur. Arrows and their associated equation numbers are colored based on whether they are used for the observation panel (purple), attribute panel (green), or both (black). Note that Equation 4.5 and Equation 4.6 are both used in PaI, whereas only one of these equations is used in PrI.

Interconnecting the observation importances and the attribute importances is first seen when initializing each of the projections. For the projection of the observations, an observation weight vector is first initialized so that each observation has a weight of $1/n$. This reflects an equal level of importance for each observation while maintaining the sum to 1 and 0 to 1 constraints for the weight vector. Then, Equation 4.6 is used to determine the attribute importances. However, this equation is not constrained as the attribute weights are. Therefore, to use these attribute importance values for the attribute weights in Equation 4.1, they

are normalized to sum to 1. Similarly, the projection of the attributes is initialized by first generating a set of attribute weights in which each weight is $1/p$. Then, Equation 4.5 is used and normalized to sum to 1 to generate the observation weights for Equation 4.2.

After projecting the data using Equation 4.1 and Equation 4.2, as depicted in Figure 4.2, the node sizes and opacities can be interpreted as visualizations of (left) which observations are most important or best describe the differences between the attributes and (right) which attributes are most important or best describe the differences between the observations. For this initial projection, the manner in which the importances of the observations and attributes are determined result in emphasizing items that are most "popular" (i.e., have the highest sum across the dataset, as explained in Section 4.4.1). Additionally, these projections show similarities (i.e., correlations) between attributes or similarities between observations.

These equations for importances are also used during each interaction. For example, when the analyst performs PaI on an attribute, a new set of observation importances is calculated using Equation 4.5. These importance values are used to update observation node size and opacity. Then, the observation importances are used to calculate a new set of attribute importances using Equation 4.6. This results in an update to attribute node size and opacity. Additionally, both observations and attributes are reprojected via Equation 4.1 and Equation 4.2 (respectively) after normalizing both new sets of importances to sum to 1. Thus, PaI results in node size, opacity, and position being updated in both projections. This effect is demonstrated in Figure 4.3-A and Figure 4.3-B.

Similarly, PrI in the observation panel produces a new set of attribute weights. These weights are used in Equation 4.1 to reproject the observations and in Equation 4.5 to determine new sizes and opacities for the observation nodes. Then, the new observation importances are used in Equation 4.6 to update the attribute node size and opacity. To update the attribute node positions, the new observation importances are normalized to sum to 1 and used in

Equation 4.2. Thus, PrI also results in node size, opacity, and position being updated in both projections. This effect is depicted in Figure 4.3-C and Figure 4.3-D.

In the above use of dot products to relate observation importances to attribute importances and vice versa, there is an implied assumption that higher data values represent more importance. For example, we assume that a higher importance for the *Water* attribute indicates that animals that have a high value for the *Water* attribute are more important than animals with a low value. While this assumption is appropriate for some applications, such as in text analytics where values represent word occurrences, it may be less appropriate in other applications, such as wanting to emphasize both extrema. However, the technique for relating observation and attribute importances as described in Section 4.2 is purposefully generic to support a variety of mathematical definitions for relating importances, including one which might generate higher importance values for animals with extreme low *Water* values as well as those with extreme high *Water* values.

When interpreting the dual projections, it is important to understand that, while the node sizes and opacities in one projection describe the spatial layout of the other projection, the projections themselves do not map onto each other. That is, since the projections represent separate high-dimensional spaces, the spatial positions of nodes in one projection do not specifically relate to the node positions in the other. Attempting to align all spatial positions would create a projection similar to the Data Context Map [23], which necessarily distorts the similarities between the observations, attributes, or both, as discussed in Section 2.3.3. Although one of the projections can be rotated and reflected to better match the layout in the other projection (e.g., rotate the attribute panel so that the *Water* and *Grazer* attributes are roughly in the same positions as the animals that are higher in those attributes), the potential tradeoff is that this may lead analysts to conclude that the two projections *can* be mapped onto each other. An inaccurate conclusion such as this can lead to significant

misinterpretations of how the projections relate to each other. For these reasons, we do not attempt any such alignment between the two projections.

4.4 Examples of Data Analysis with SIRIUS

To demonstrate how Equations 4.1–4.6 coalesce to enable exploratory data analysis with diverse high-dimensional datasets, we provide examples of animal data, intelligence analysis data, and breast cancer data in our implementation of SIRIUS. These examples show that SIRIUS can be used to explore quantitative data, textual data, and large datasets, respectively. A demonstration video for each of these examples is available at https://youtu.be/TzBjImkrbDU.

Since Equations 4.1–4.6 rely on strictly numerical data, some of the example datasets had to be altered to change categorical attributes to numerical representations. Additionally, any rows that contained missing data were removed. Such issues could be better addressed through the use of alternative distance functions, such as Gower distance [50].

As we move away from more intuitive datasets like the one by Lampert et al. [71], it is important to note that raw data for a selected observation or attribute is displayed alongside the "Importance" slider. For numerical datasets, this raw data is expressed as a simple list of key-value pairs. For text datasets, selecting a document instead displays the associated raw text. Thus, analysts can readily explore the entire dataset without having to reference spreadsheets or outside tools to interpret the projections.

4.4.1 An Animal Dataset

The Full Dataset

Given the initial projection of the entire animal dataset by Lampert et al. [71] depicted in Figure 4.1, we can already begin gaining insights about the dataset. For example, while there are no strongly distinguished groups or clusters of animals, more water-dwelling animals appear in the upper right whereas more land-dwelling animals appear in the lower left. Despite the hypothesis that the differences between animals are best described by their *Water* attribute, the size and opacity of the attribute nodes indicate that *Quadrupedal* is the correct answer. Since this is the initial projection of the attributes, this also means more animals have a high value for *Quadrupedal* than for any other attribute. Therefore, many animals in the dataset are *Quadrupedal* and that this attribute is the most "popular" attribute in the dataset. In addition to these insights, the projection of the attributes shows that there are strong correlations between certain attributes, such as *Quadrupedal* and *Furry*, since they are projected closely together. Thus, an animal that has a high value for *Quadrupedal* is likely to also have a high value for *Furry*. Similarly, *Grazer* and *Hooves* are somewhat correlated, but since they are projected on the opposite side of the attribute panel, they are not correlated with *Quadrupedal* and *Furry*.

Analysis on a Subset of the Dataset

Using the same subset of the animal dataset as in Figure 4.2 and Figure 4.3 for clarity, consider an analyst who wants to gain insights based on the three related questions mentioned in Section 4.1 using this dataset:

1. What attributes separate the *Tiger* and *Wolf* from the *Blue Whale* and *Dolphin* as

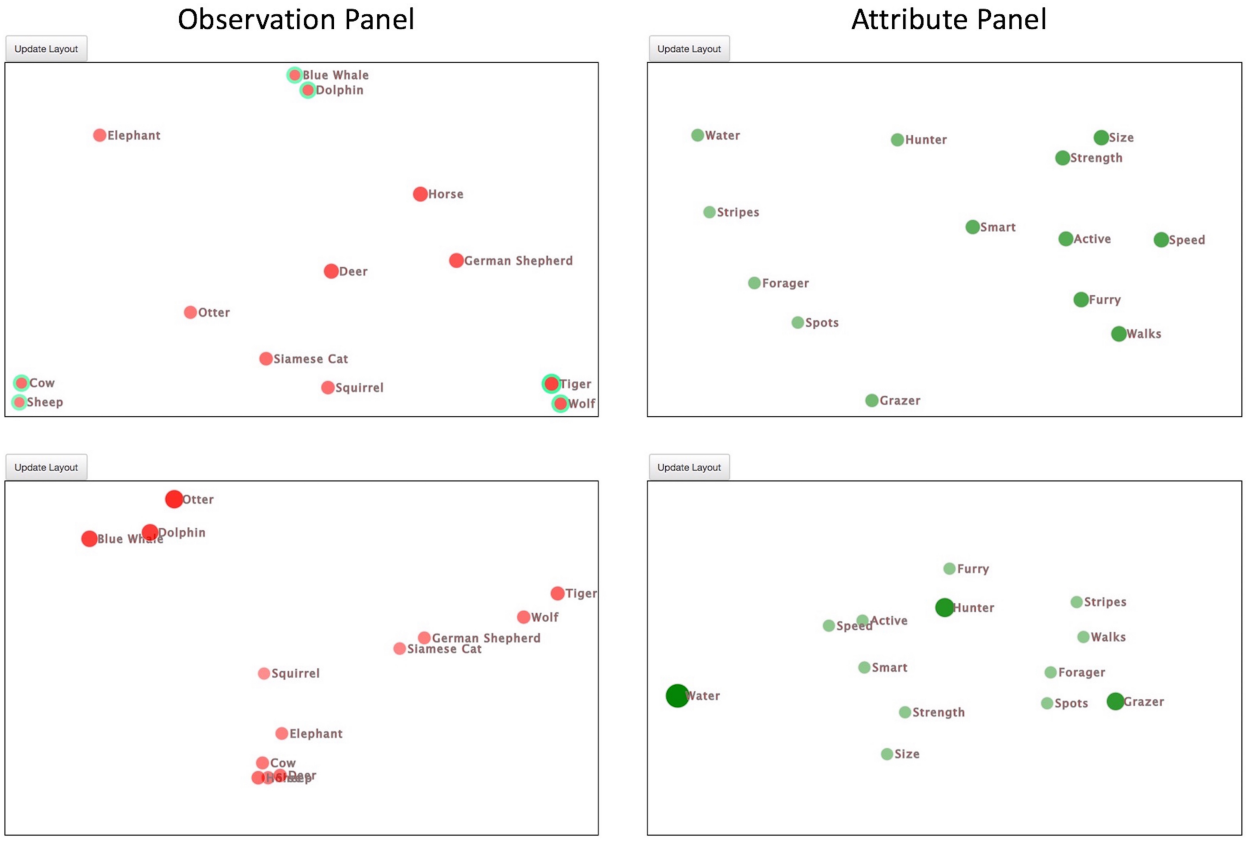

Figure 4.5: Given the initial projection shown in Figure 4.2, **(Top)** the analyst can move animals to express their desired similarities or differences to begin investigating their three questions about this animal dataset. **(Bottom)** After clicking "Update Layout," the data is reprojected with new attribute weights and observation weights. The analyst can now use node position, size, and opacity to determine the answers to all three questions without performing any further interactions.

well as from the *Cow* and *Sheep*?

2. What other animals are similar to those three groups?

3. What other attributes are correlated with the attributes that separate these three groups?

To answer these questions, after the data is initially projected (as shown in Figure 4.2), the analyst would begin by using PrI to move the nodes for the animals of interest into three

groups, as depicted in the top row of Figure 4.5. Clicking "Update Layout" results in the final projection shown in the bottom row of Figure 4.5.

Following this reprojection, the analyst can answer Question 1 by observing that the attributes *Water*, *Hunter*, and *Grazer* are large and opaque, thus leading to the insight that they describe the differences between the three groups of animals. Additionally, this projection gives the analyst the insight that *Horse*, *Deer*, *Elephant*, and *Squirrel* are similar to the *Cow* and *Sheep*; the *Siamese Cat* and *German Shepherd* are most like the *Tiger* and *Wolf*; and the *Otter* is similar to the *Dolphin* and *Blue Whale*. Thus, Question 2 is also answered from this projection.

However, Question 1 and Question 2 can be answered by existing techniques such as Andromeda [102]. What makes SIRIUS unique is that Question 3 can also be determined at a glance via the relative node positions in the attribute panel; more similar (or correlated) items will appear closer together in the projections. Thus, analysts can easily gain the insight that *Furry*, *Active*, *Speed*, *Smart*, *Strength*, and *Size* are all more correlated with *Hunter* than with *Grazer* or *Water*. Similarly, *Stripes*, *Walks*, *Forager*, and *Spots* are most correlated with *Grazer*. The *Water* attribute, in comparison to *Hunter* and *Grazer*, is not correlated with any other attributes.

Connecting these answers for Question 3 back to the animals, this means that the animals that have a high value in *Hunter* are more likely to have higher values for *Furry*, *Active*, *Speed*, *Smart*, *Strength*, and *Size* than animals that have a high value in *Grazer* or *Water*. Inspection of the animals in the observation panel (or domain knowledge) provides the insight that animals that are high in *Hunter* are animals like *Tiger* and *Wolf*. Similarly, animals that have a high value in *Grazer* (like the *Cow* and *Sheep*) are more likely to have higher values for *Stripes*, *Walks*, *Forager*, and *Spots* than animals that have a high value in *Hunter* or *Water*. Animals that have a high value in *Water* (like the *Dolphin* and *Blue Whale*) are

not as likely as animals that have a high value in *Hunter* or *Grazer* to have higher values in any of the other attributes.

4.4.2 A Text-Based Dataset

As mentioned in Section 4.3.3, our notion of importance, defined by Equations 4.5–4.6, results in a high importance for observations that have high values for important attributes and vice versa. This definition of importance is well-suited for text-based datasets in which a higher value for an attribute (e.g., an extracted entity) denotes that the associated extracted entity appears more often in that document (observation).

TF-IDF Data vs. Topic Modeled Data

To demonstrate how SIRIUS can be used to explore textual data, we extracted entities from a synthetic intelligence analysis dataset [57] and created a TF-IDF matrix in data preprocessing steps. Using SIRIUS to visualize this data (depicted in Figure 4.6-A), we can immediately see that *Charlottesville* is greatly emphasized over other attributes. Given this is the initial projection, the emphasis on *Charlottesville* indicates that this entity has the highest sum of TF-IDF values across the entire dataset, hinting that there may be something nefarious occurring there. Since *Charlottesville* is the entity that has the largest influence on the documents in the observation panel, documents that mention *Charlottesville* (the 5 documents towards the top of the observation panel) are separated from the ones that don't (in the middle of the observation panel). Inspecting these *Charlottesville* documents provides insight on a terrorist plot in Charlottesville involving several individuals.

However, Figure 4.6-A has many of the observations and attributes overlapping with each other, making them harder to distinguish from each other. Although the attribute node

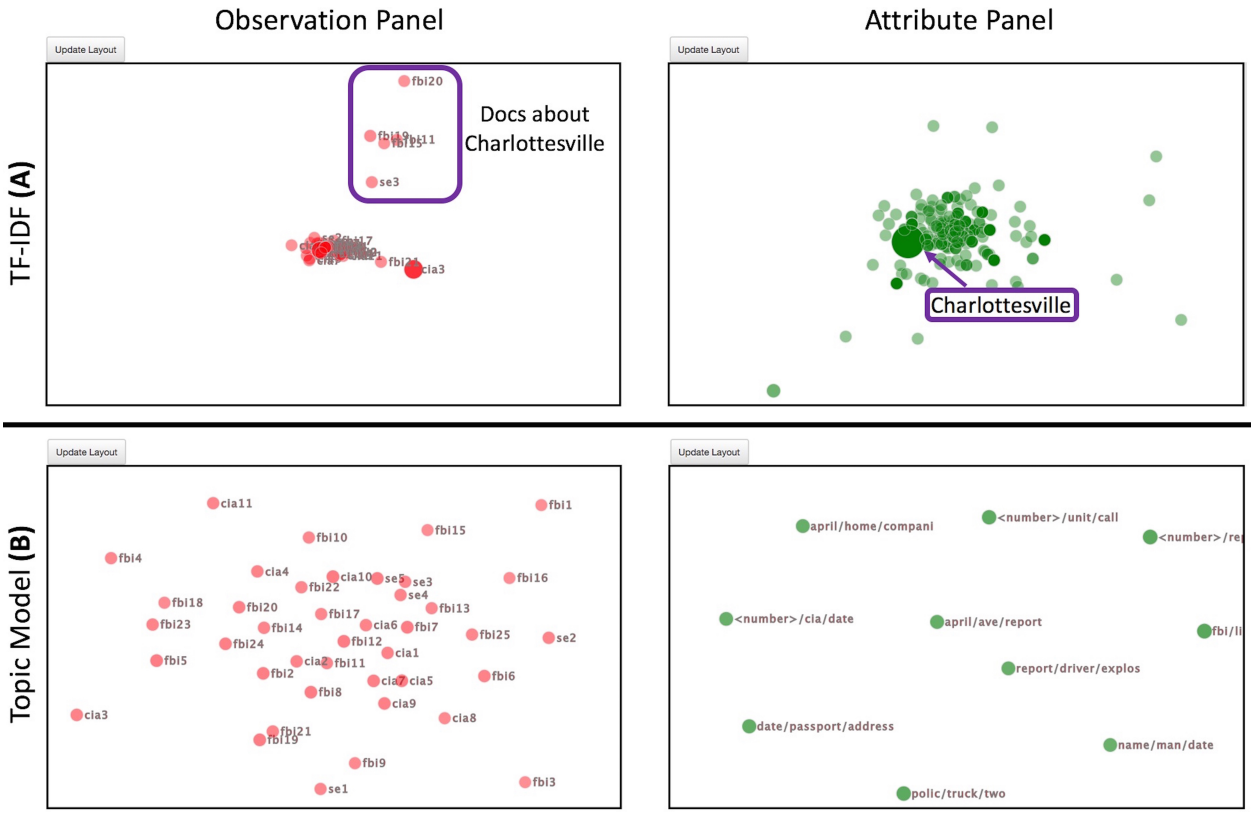

Figure 4.6: The panels labeled **A** show an initial projection using SIRIUS with all extracted entities as attributes of a textual dataset, which immediately emphasizes *Charlottesville* as an important entity. The panels labeled **B** show an initial projection with topics learned through topic modeling as the attributes of the dataset. While this makes both the projection of the observations and the projections of the attributes clearer, the initial insight about *Charlottesville* is lost.

labels have been removed to improve clarity in the projection, these labels are still accessible by hovering over a node.[2] However, this issue can also be alleviated using topic modeling to essentially group attributes together (and thus better separate the documents) during an additional preprocessing step. Visualizing the topics in place of the extracted entities results in the much clearer initial projections shown in Figure 4.6-B. The tradeoff in doing so is that the previous initial insight that Charlottesville may be the center of some nefarious activity

[2]There are many methods for improving the display of labels in scatterplot-like visualizations [27, 44]. However, this is not the main focus of this paper; we instead focus on new interactive projection techniques for displaying both observations and attributes of high-dimensional data and highlight the insights that can be gained.

is lost.

Example Analysis

To show analysis on text data with SIRIUS, we use the topic modeled data to improve clarity. A reasonable starting point with a dataset like this is to pick out attributes (i.e., topics) that seem more indicative of nefarious activity than others. Examining the topics shown in the attribute panel of Figure 4.6-B uncovers a number of such topics, including one that focuses on passport information (*dates/passport/address*), another on police activity (*polic/truck/two*), and a third on explosions (*report/driver/explos*). In contrast, topics like <*number*>*/cia/date* seem perhaps generic or less useful to the initial investigation. Performing PrI by dragging the three attributes of interest into one corner to and <*number*>*/cia/date* into the opposite corner to expresses their desired similarities/dissimilarities (depicted in the top row of Figure 4.7) and clicking "Update Layout" above the attribute panel results in the visualization depicted in the bottom row of Figure 4.7. This visualization gives the insight that the topics <*number*>*/report/phone* and *april/ave/report* are very closely correlated with the three topics of interest that were moved. These new topics, along with *fbi/list/*<*number*> (which is now the most emphasized topic), are all worthy of further investigation.

However, most notably, this interaction resulted in the document *fbi11* being highly emphasized in the observation panel, giving insight on its strong association with the emphasized topics in the attribute panel. Reading the contents of this document reveals information that happens to be central to one of the three main terrorist plots contained within the dataset, as further analysis can confirm.

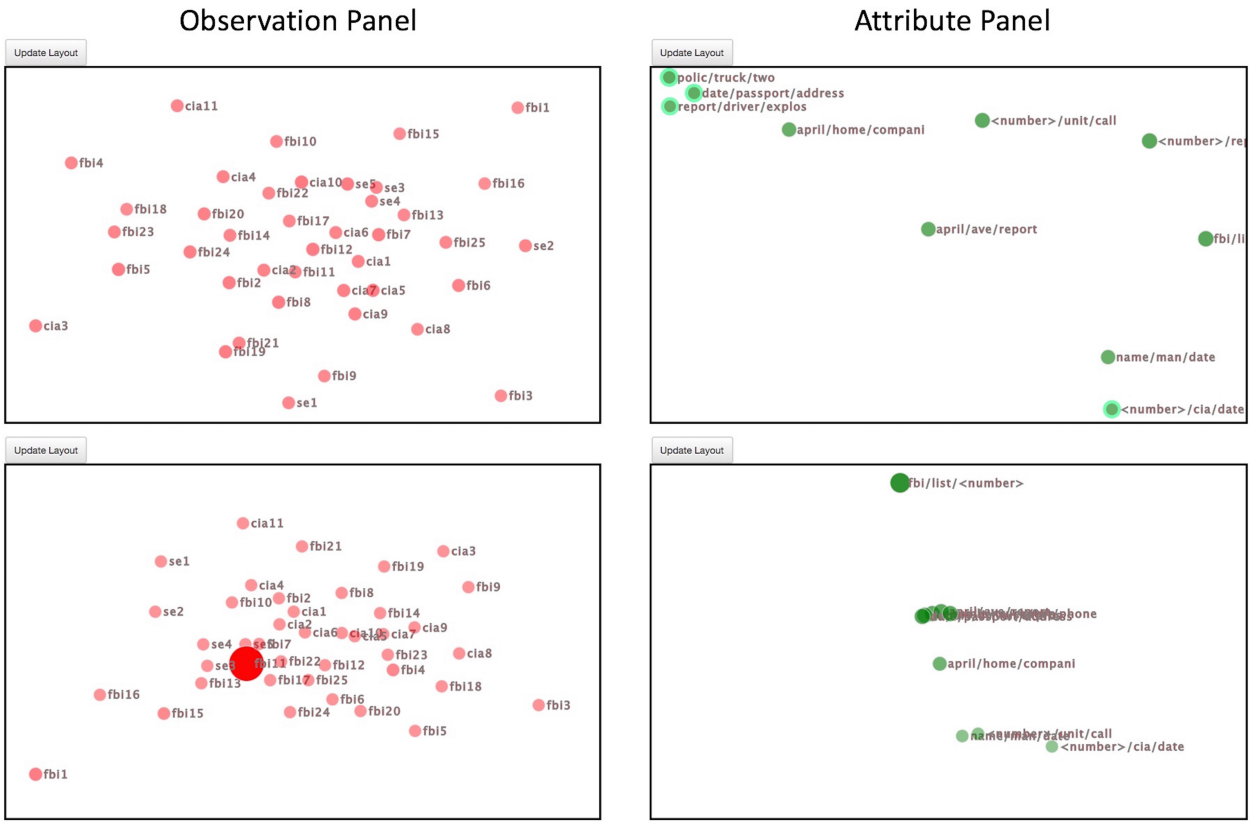

Figure 4.7: From the initial projection of the topic modeled data shown in Figure 4.6-B, nefarious activity can be uncovered by **(top)** using PrI on the attributes to separate topics of interest from generic or uninteresting topics. Clicking "Update Layout" produces **(bottom)** a visualization which reveals other topics that are very closely correlated with topics of interest. Additionally, the combination of emphasized attributes results in *fbi11* in the observation panel being highly emphasized. This document reveals crucial information to one of the three main terrorist plots in this dataset.

4.4.3 A Breast Cancer Dataset

To demonstrate the ability of SIRIUS to enable exploration of larger datasets, Figure 4.8 shows the "Breast Cancer Wisconsin (Original)" dataset from the UCI Machine Learning Repository [35]. Similar to Figure 4.6, the observation labels have been removed to improve clarity. In this visualization, many observations representing benign tumors naturally group together in the lower left of the observation panel, separating themselves from those representing cancerous tumors.

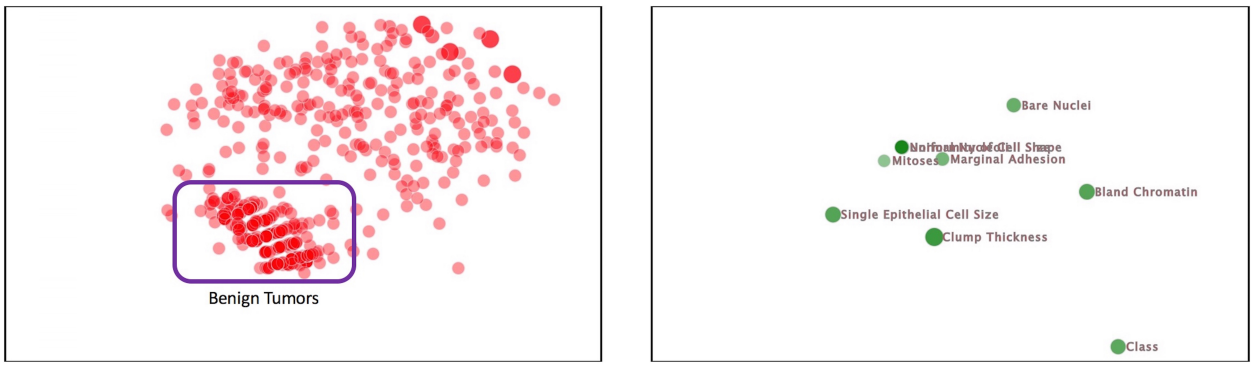

Figure 4.8: An initial projection of the "Breast Cancer Wisconsin (Original)" dataset [35] using SIRIUS. Note that the dense group of nodes in the lower left of the observation panel correspond to benign tumors. The attribute projection reveals that the observation projection is best described by the *Clump Thickness* attribute. However, this attribute, along with *Single Cell Epithelial Size* and *Bland Chromatin* are the attributes that are most closely correlated with the *Class* attribute and therefore may be useful in diagnosing breast cancer in patients.

Looking at node size and opacity in the attribute panel, we can see that the attributes that describe this separation are *Bland Chromatin*, *Single Cell Epithelial Size*, and, most notably, *Clump Thickness*. Thus, an analyst can immediately gain the insights that *Bland Chromatin* and *Single Cell Epithelial Size* do help distinguish between observations, but *Clump Thickness* describes the separation between the different observations better than the others. Additionally, since *Clump Thickness* is the closest attribute node to *Class*, this means that *Clump Thickness* is the attribute that has the strongest correlation with *Class*. This correlation explains why observations seemed to be well-separated by class. Theses insights also mean that doctors may be able to use clump thickness, bland chromatin, and single cell epithelial size to help distinguish between cancerous and non-cancerous tumors. While these insights may be obvious to medical practitioners, the fact that SIRIUS immediately uncovers them demonstrates its ability to easily reveal critical information in high-dimensional datasets.

4.5 Comparing SIRIUS to Existing Techniques

Here, we highlight that SIRIUS enables exploratory data analysis on observations and attributes simultaneously and efficiently, as evidenced in Section 4.4. This includes insights such as attribute correlations while exploring observation similarities, which can be gained through few, simple interactions.

Using Andromeda [102] to represent a contrasting example technique, PaI and PrI are enabled on a single observation projection (accomplishing half of Goals 1 and 2). A separate Andromeda instance would need to be run simultaneously to display the attributes to enable the same interactions on an attribute projection (for the other half of Goal 1 and Goal 2). While this would provide the two projections and the same interactions within each, they would remain disconnected (falling short of Goal 3). Thus, the analyst would be forced to estimate or even guess how to manipulate one projection to reflect changes in the other. This process would be time-consuming and error-prone, easily resulting in incorrect conclusions. Therefore, we assert that SIRIUS provides a more powerful platform than existing techniques for performing data analytics tasks that incorporate both the observations and the attributes of a dataset, such as the example tasks in Section 4.4.

To directly compare SIRIUS with a variety of existing visual analytics techniques for high-dimensional data, we compare the capabilities of SIRIUS and the techniques exemplified by Andromeda [102], Dust & Magnet [137], Star Coordinates [60], Dis-Function [15], LAMP [59], Dimension Projection Matrix/Tree [138] (shortened to "DP Matrix/Tree in Table 4.2), the visualization proposed by Turkay et al. [118], Data Context Map [23], Intent Radar [95], and Doc-Function [16]. These comparisons are summarized in Table 4.2, which emphasize how these other techniques meet the three Goals described in Section 4.2. Note that these comparisons are based on how the visualization is presented in their respective publications

Table 4.2: A summary of the comparisons between SIRIUS and existing visual analytics techniques for high-dimensional data. In some cases, a visual analytics system is used to exemplify a technique. "O" or "A" denotes that the given technique has the specified ability, whereas "o" or "a" denotes that the specified ability is only partially supported or only supported under certain circumstances.

		SIRIUS	**Andromeda** [102]	**Dust & Magnet** [137]	**Star Coordinates** [60]	**Dis-Function** [15]	**LAMP** [59]	**DP Matrix/Tree** [138]	**Turkay et al.** [118]	**Data Context Map** [23]	**Intent Radar** [95]	**Doc-Function** [16]
SIRIUS Goals	Goal 1: Similarity-based projection of observations (O or o) and attributes (A or a) (Section 4.5.1)	OA	O	O	O	O	O	OA	oa	oa	A	A
	Goal 2: Manipulate attribute importance (A or a) or observation importance (O or o) to explore observation similarities or attribute similarities (e.g. PaI on the attributes or observations), respectively; Section 4.5.2)	OA	A	A	A							
	Goal 2: Manipulate observation similarities (O or o) or attribute similarities (A or a) to explore attribute importances or observations importances (e.g. PrI on the observations or attributes), respectively (Section 4.5.2)	OA	O			O	o					a
	Goal 3: Relate attribute importances to observation importances (O or o) or vice versa (A or a) (Section 4.5.3)	OA									O	
Other	Distribution of observations across attributes (O or o) or attributes across observations (A or a)		O	oa	o	Oa		o	Oa	o		OA
	Clustering of observations (O or o) or attributes (A or a)			o			O	OA			A	

and whether the visualization directly enables the given task or directly provides the given information. For example, Andromeda by default provides a similarity-based projection of the observations of high-dimensional data. Although a projection of the attributes could be achieved by using a transpose of the original data matrix as input, this additional projection is not automatically given as part of the visualization. Therefore, we consider Andromeda to *not* provide a similarity-based projection of the attributes. The following subsections provide further details on why we filled each cell of Table 4.2 in the manner presented.

4.5.1 Goal 1: Similarity-Based Projections

As discussed in Section 2.3, most visual analytics techniques, including Andromeda, DisFunction, and LAMP, focus on the observations. Thus, only a similarity-based projection is provided for the observations in many of the techniques in Table 4.2. However, DocFunction provides a similarity-based projection of the attributes, and the Intent Radar maps the similarity of the attributes to the angle around the radar. As discussed in Section 2.3.3, the Data Context Map projects the observations and the attributes into the same space using MDS, which necessarily distorts the projection of the observations, the projection of the attributes, or both. In the visualization from Turkay et al., three scatterplots are provided: one that visualizes the observations using one attribute for each of the axes, one that visualizes the observations using two principle components that eliminate outliers, and one that visualizes the attributes based on their mean and standard deviation. While each of these scatterplots could arguably be a visualization that uses a simplified definition of similarity to produce the scatterplot, we consider these scatterplots to be too simple to be classified as true similarity-based projections as they don't use all the observations or attributes in a similar manner to MDS or PCA. Therefore, we propose the Data Context Map and the visualization described by Turkay et al. only partially support this goal.

4.5.2 Goal 2: Exploring the Projections

Manipulating Importances to Explore Similarities

Given the general focus on projections of observations and interactions therein as opposed to that of attributes, it is perhaps expected that none of our selected comparison techniques enable this manipulation of observation importances to explore different projections. However, Andromeda (via PaI), Dust & Magnet (via increasing the magnitude of a magnet), and

Star Coordinates (via increasing the length of an attribute's axis) enable manipulation of attribute importances to explore observation similarities.

Manipulating Similarities to Explore Importances

Techniques such as Andromeda (via PrI) and Dis-Function (via dragging and dropping nodes) enable direct manipulation of the observation similarities to explore the attribute importances. In Andromeda, the result of this interaction is reflected in the position of the attribute sliders, whereas Dis-Function portrays this information in a bar graph. While LAMP affords a similar interaction, the importance given to each of the attributes is not portrayed to the analyst. Likewise, Doc-Function enables analysts to use a similar interaction technique on a projection of attributes, but the importance given to each observation is not available to the analyst. We therefore argue that LAMP and Doc-Function both only partially support this goal.

4.5.3 Goal 3: Relating Importances to Each other

While no other techniques related observation importances to attribute importances, the Intent Radar is the only other technique in our list that relates attribute importances to observation importances. This is accomplished by visually encoding each attribute's importance as its distance from the center of the radar. This information is then used to determine the importance of each document, with the documents provided to the right of the radar visualization.

4.5.4 Other Mechanisms to Generate Insight

Although our implementation of SIRIUS provides a unique interface that enables powerful interactions and insights, it is not a comprehensive implementation; there are other insights commonly afforded in other exploratory data analysis techniques for high-dimensional data. For example, distributions show how a particular piece of data compares to all others or how common certain values are. This helps analysts understand the given dataset at a high level as analysts generally have low cognitive dimensionality, as described by Self et al. [103]. Similarly, clustering data helps analysts be able to automatically group data together, which also helps give a high level overview of the dataset. While we note that SIRIUS doesn't explicitly prohibit the inclusion of such insights, we discuss each of these types of common insights in detail in the context of our selection of techniques.

The most common insight afforded by our selection of techniques is the distribution of observations across attributes, which can be seen in Andromeda (seeing the raw data values for selected nodes along the attribute sliders), Dis-Function (the parallel bars view), the visualization provided by Turkay et al. (by manipulating the axes of the first scatterplot), and Doc-Function (through searching and the Highlight feature). The Data Context Map only partially supports this insight by allowing analysts to manipulate the ranges for the contour lines, through which analysts can eventually learn the distribution. Star Coordinates also partially supports this insight by allowing analysts to select value ranges of interest for each axis, which can reveal the distribution of observations. Alternatively, analysts can manipulate the size and orientation of the attribute axes to view the distribution of observations across a single attribute. Similarly, the Dust & Magnet visualization partially supports viewing distributions of observations across attributes by having observations move faster towards a moved attribute if it has a higher value for that attribute. However, Dust & Magnet and Dis-Function also partially support insights regarding the distribution of

attributes across observations through visual encodings of node color and size, and the raw data matrix (respectively). Dimension Projection Matrix/Tree partially supports this insight by allowing analysts to refine the attributes used in a single projection of observations in the matrix to one attribute for each axis. Doc-Function directly affords analysts this insight by hovering over or clicking keywords.

Finally, LAMP, and Dimension Projection Matrix/Tree show clustering of observations. LAMP determines clusters both via k-nearest neighbors and via a silhouette coefficient. Dust & Magnet partially supports this insight by coloring the observations by a analyst-selected categorical attribute. Dimension Projection Matrix/Tree directly supports automatic clustering of observations or of attributes by clicking on a corresponding button in a toolbar which performs the clustering using a kNN graph. However, the Intent Radar clusters attributes by mapping the results from agglomerative clustering to both color and position, thereby directly supporting this insight.

4.6 Limitations and Future Work

Using SIRIUS comes with several limitations. Firstly, there is a potential tradeoff in how one projection cannot be transposed on top of another (as exemplified by the bottom row in Figure 4.5), which may be confusing for some analysts. Despite this, we highlight that every projection provided in SIRIUS is a valid projection that can provide rich, meaningful insights, as demonstrated in Section 4.4.

Another potential limitation of our implementation of SIRIUS is in the usability and understandability of PrI caused by the fact that only a subset of nodes are used to calculate a new set of weights, which are then applied to all nodes. While Self et al. explore this limitation in [102], the user study described in [100] highlights the benefits that PrI can bring to the

analysis process.

Additionally, a limitation of our implementation of SIRIUS is a "jumping" effect, which is most evident when performing PaI on each panel in sequence. This effect stems from our use of Equation 4.5 and Equation 4.6. For example, assume that PaI was just performed in the attribute panel. The weights for the changed attributes, W_A, are first fed into Equation 4.5. The observation weights, W_O, resulting from that equation are then fed into Equation 4.6 to determine a final set of attribute weights, W_A. Repetitions of this interaction follow the same flow; for small, subsequent changes to W_A, this series of steps results in similarly small changes to W_O. However, when an observation's importance value is then manipulated, the first step instead becomes that the new W_O is fed into Equation 4.6 to determine a new W_A. Since the W_A here is very different than the previous W_A, the projection of the attributes changes greatly. Therefore, although an analyst may expect both projections to change minimally, a large change is reflected in the attribute projection.

One cause of this "jumping" effect is that the cycle of Equation 4.5 and Equation 4.6 have a single convergence point, as suggested by the HITS algorithm [64]. However, since we want to explore alternative projections, we necessarily consider other pairs of weight vectors for which the cycle is not converged and therefore "jump" when the analyst switches their interaction back and forth between the projections.

A related cause of the "jumping" effect is the memorylessness of the interaction. When a new interaction is performed, our implementation of SIRIUS only uses the most recent interaction to update the visualization. One possible way to address this issue is to alter Equations 4.5–4.6 to incorporate previous interactions or projections. For instance, Leman et al. [73] suggest a weighted average of the weight vectors produced by the most recent interaction with vectors from previous interactions.

Lastly, a limitation to our implementation of SIRIUS is the size of data that is supported due to the n^2 and n^2p^2 computational complexities for projection and interaction optimizations (respectively). However, recent performance improvements in the computation of WMDS projections [29] as well as PrI interactions (which others in the BaVA@VT group are researching) can greatly increase the size of datasets that can be supported with interactive response times. However, this issue may also be alleviated for some datasets by implementing foraging features (e.g., searching and filtering for text data) in which only the most important observations and attributes are projected, such as in StarSPIRE [12]. We describe one example of a SIRIUS-based system that incorporates foraging in Chapter 7.

We intend to address these issues and limitations by continuing to develop our implementation of SIRIUS. This will also enable us to improve other aspects as well, such as the placement and size of the node labels in the projections or adding trails to highlight the impact the interaction had on the nodes' locations. With a more refined implementation in hand, we will be able to provide thorough evaluations on the time complexities of our refined algorithms as well as run user studies to evaluate interpretability and usability.

4.7 Conclusion

In this chapter, we identified an opportunity for dual, symmetric, interactive projections of high-dimensional data to support the interconnectedness between observations and attributes in exploratory data analysis tasks. Given this need, we defined a generalized technique called SIRIUS, which models three principles: (1) dual projections of observations and attributes, (2) symmetric interactions on importances to explore projections, and (3) symetrically relating importances of observations to importances of attributes. To concretize these principles, we described a specific implementation based on WMDS, weighted Euclidean distance, para-

metric and projection interactions, and dot-product importance calculations. A set of examples then demonstrated how SIRIUS provides insights across a range of diverse datasets, and we compared SIRIUS against a suite of existing techniques. SIRIUS offers new insight into both observations and attributes of high-dimensional data and their interrelationships, while maintaining a consistent symmetric mental model of each.

Chapter 5

Effects of Symmetry on High-Dimensional Data Analysis

5.1 Introduction

To support analyses of high-dimensional data, research has shown that dimensionally reduced projections (e.g., [12, 16, 23, 39, 52, 58, 76, 102, 117, 118, 126, 138, 139]) assist the sensemaking process [90] by using an intuitive "proximity ≈ similarity" visual metaphor. However, it has recently been noted that many such projections support analyses that are observation-centric (where the individual data items are the dominant component that is visualized or interacted with) or attribute-centric (where the features or dimensions of those data items are the dominant component). As such, these systems do not support a cognitive symmetry in which analyses concern *both* observations and attributes [43].

SIRIUS (as described in Chapter 4) was developed to address this need. While the analytical prowess of a symmetrical system was shown, it was not clear which specific types of analytic tasks benefited from such symmetry. More specifically, when compared to a system like Andromeda [102], we find that SIRIUS imposes two types of symmetry: visual symmetry (where the system provides a projection of the observations as well as a projection of the attributes) and interaction symmetry (where the system translates a single interaction into updates in both projections). These two types of symmetry lead to a question of which type

yields the most benefit in a given analytic task. Moreover, does a particular type of symmetry help analysts in considering more observations or attributes in their analytical reasoning? Additionally, is both visual and interaction symmetry necessary to yield a benefit, or is just visual symmetry (which is arguably easier to implement) enough?

To explore these questions, we present a between-subjects user study in which participants are assigned a system that incorporates a specific blend of visual and interaction symmetry. Using this system, they must complete a variety of analytic tasks. By comparing participants' performance between conditions, we can compare which type(s) of symmetry best support a given type of analytic task. As such, our main contribution is a user study on the effects of visual and interaction symmetry on analytic tasks in terms of 3 measures of performance:

- **Time on task:** how quickly participants were able to perform each type of analytic task.
- **Accuracy:** how correct participants' responses to the given analytic questions were
- **Cognitive cardinality and dimentionality** [103]: how many observations and attributes (respectively) were used in participants' analytical reasoning

Using the results from this study, future systems can better leverage visual and/or interaction symmetry to support specific analytic tasks. Specifically, our research questions and hypotheses are:

1. **Does visual symmetry, interaction symmetry, or both improve time on task for analytic tasks?** We hypothesize that only certain analytic tasks will benefit from symmetry, such as *Cluster* tasks or *Complex* tasks that involve multiple, interrelated analytic goals. In contrast, we hypothesize that interaction symmetry will have a negative impact on other tasks which require analysts to express a specific importance

for a given observation or attribute, which may be seen *Correlate* tasks (as described in Section 5.3). Additionally, we expect to see the combination of visual symmetry and interaction symmetry have the greatest influence (positive or negative) on time on task.

2. **Does visual symmetry, interaction symmetry, or both improve accuracy in analytic tasks?** We hypothesize that the results for accuracy in analytic tasks will be similar to results for time on task.

3. **Does visual symmetry, interaction symmetry, or both improve cognitive cardinality or dimensionality?** We hypothesize that at least visual symmetry will improve both cognitive cardinality and dimensionality.

5.2 Background

Given the computational and visual similarities between Andromeda [102, 103] and SIRIUS (as described in Chapter 4), we use Andromeda as a baseline in our user study, as described in Section 5.3. Additionally, we drew inspiration from a user study performed with Andromeda [103] to develop our own study. As such, we provide an overview of Andromeda and SIRIUS as well as how they have been altered to better contextualize our different user study conditions. To assist with these explanations, Figure 5.1 depicts how Andromeda and SIRIUS compute the effects of both parametric interactions (PaI) and projection interactions (PrI) [37, 103].

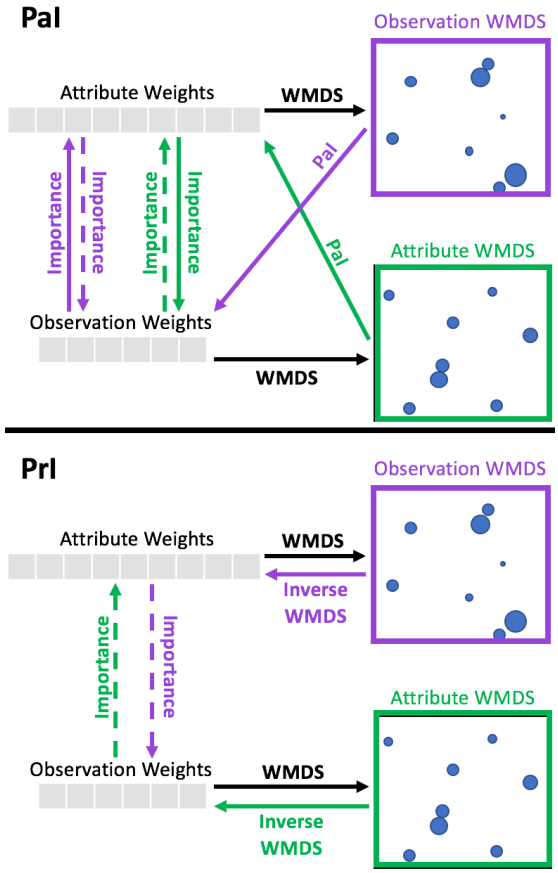

Figure 5.1: A depiction of the computations used to realize parametric interactions (PaI) and projection interaction (PrI) in the NS, VS, and BS conditions. The BS condition uses all the computations represented by the arrows, whereas the VS condition omits the dashed "Importance" arrows. The NS condition only uses the set of arrows related to the observation WMDS projection and the attribute weights, therefore completely omitting the computations represented by the "Importance" arrows.

5.2.1 Andromeda

Andromeda is classified as an interactive, observation-centric visual analytics system for high-dimensional data. As such, the main component in Andromeda is a dimensionally reduced projection of observations that uses weighted multidimensional scaling (WMDS) [68, 69] in which the nearer observations are projected, the more similar they are. The weights used to create the projection are weights on the attributes, which are represented by interactive sliders to the right of the projection. These sliders provide an intuitive interface for analysts

to understand and interact with these weights.

For the purposes of our user study, we use an updated version of Andromeda hosted on a website, as depicted in Figure 5.2, which predominantly changes visual encodings to more closely mirror those of SIRIUS to eliminate these differences as confounding variables. As such, the projected observations are all red in color, and larger node sizes and opacities reflect which observations have high values (on average) for the highly weighted attributes. For instance, Figure 5.2 shows that *Tiger* has relatively high values amongst each of the equally weighted attributes in comparison to other animals such as *Chihuahua*. Hovering over or clicking on a node in this projection represents a **surface-level interaction**. Hovering turns the node yellow until the mouse is moved away from the node, while clicking turns it orange until it is clicked again. Simultaneously, points of the same color appear along the attribute sliders to indicate the relative raw values associated with the given observation. An additional surface-level interaction was also implemented to allow analysts to hover over an attribute slider to see purple borders around the observation nodes appear to reflect the relative values each observation has for the given attribute. This interaction is shown with the *Bulbous* attribute in Figure 5.2.

Another type of interaction is a **parametric interaction (PaI)**. This interaction is performed by directly manipulating an attribute slider to reflect a desired level of importance for a given attribute. Following this interaction, the system directly uses the slider's new value to alter the weight for the given attribute. As a result, the projection updates to reflect this change in the attribute weights. Additionally, the sizes and opacities of the nodes change, where larger and darker nodes depict which observations have relatively high values for the attributes that have higher weights.

Finally, the user can perform a **projection interaction (PrI)** by clicking or dragging nodes in the projection to express desired similarity/dissimilarity relationships between specific

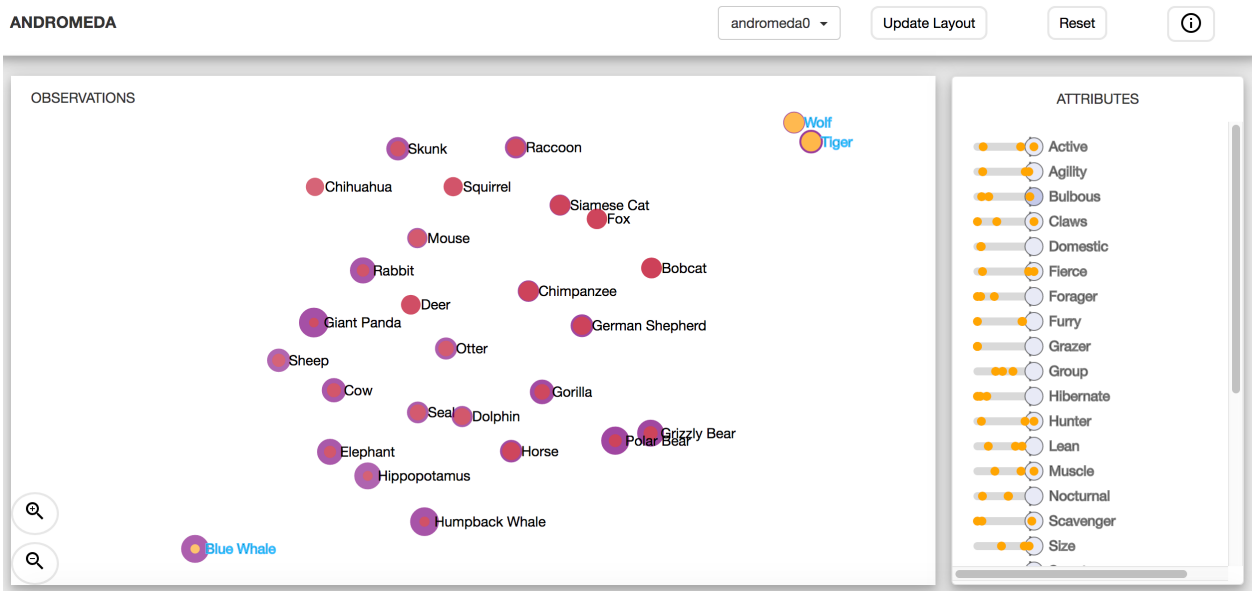

Figure 5.2: The Andromeda interface used for NS condition in the user study described in Section 5.3. Details for this interface are provided in Section 5.2.1.

observations, as shown with *Blue Whale*, *Wolf*, and *Tiger* in Figure 5.2. When the analyst is satisfied with their projection, an "Update Layout" button is clicked. This button triggers an inverse WMDS calculation in which relative positions of the orange nodes are used to automatically update the attribute weights. These new weights are then used to update the projection of all the observations and to move the attribute sliders accordingly. As such, PrI enables analysts a natural and intuitive method for exploring similarity/dissimilarity relationships without necessitating a deep understanding of WMDS or forcing the analyst to manipulate the attribute sliders by hand to create a desired projection.

5.2.2 SIRIUS

To support symmetrical and simultaneous analyses of observations and attributes of high-dimensional data, SIRIUS builds upon Andromeda to provide interactive WMDS projections of both observations and attributes, as shown in Figure 5.3. Therefore, both observations

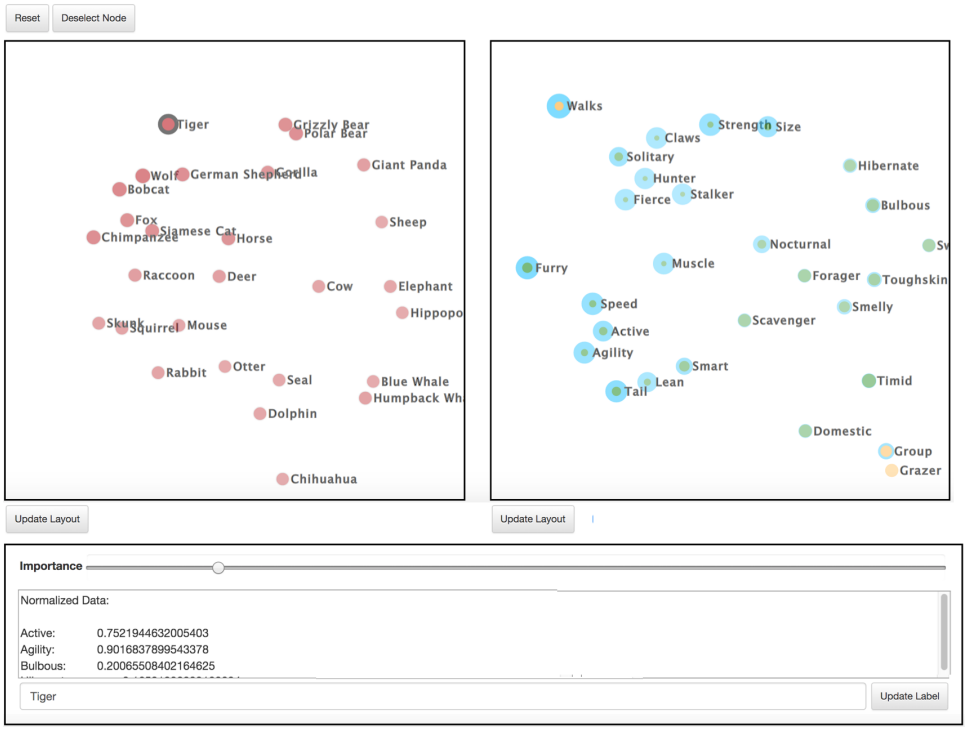

Figure 5.3: An overview of the updated SIRIUS interface, as described in Section 5.2.2.

and attributes have the same precedence in the system and can be interacted with in the same way. Observations are depicted as red nodes, while attributes are green to visually distinguish the two projections. Double clicking a node in either projection is a **surface-level interaction** that provides information about the associated observation or attribute, including borders around nodes in the opposing panel which reflect the relative raw values associated with the given observation or attribute. For example, if an observation is double clicked, such as *Tiger* in Figure 5.3, blue borders around the attribute nodes in the opposing panel will show the relative values the given observation has for each attribute. Purple boarders are used around the observation nodes when an attribute is double clicked. In addition to these visual encodings, the numerical data for these values is provided in a data panel below both of the projections.

This data panel also contains an “Importance” slider, which reflects the weight associated

with the given observation or attribute. As such, SIRIUS uses a set of observation weights as well as a set of attribute weights. Manipulating this slider is considered a **PaI** that initially sets the weight for the given observation/attribute, followed by a calculation of new weights for the opposing set. These new weights are then used to recalulate the original weight vector to highlight correlations. For example, if an observation weight is manipulated, then only that weight is initially altered. The observation weights are then used to calculate a new set of attribute weights to determine which attributes are embodied by the more highly weighted observations. Then, these new attribute weights are used to recalculate the observation weights to determine which other observations might be correlated with the observations that were previously more highly weighted. With these new weight vectors, both WMDS projections are updated, including the node sizes and opacities to visually reflect the weight associated with each observation/attribute.

Similar to Andromeda, clicked or dragged nodes in either projection turn orange to denote that they will be used in a **PrI** triggered by clicking "Update Layout" beneath the associated panel. While PrI only initially updates a single set of weights, these new weights are used to update the other set as well to enable updates in both projections. For instance, PrI can be performed by clicking or dragging attributes such as *Walks*, *Group*, and *Grazer* (as depicted in Figure 5.3) to define relative similarity/dissimilarity relationships between them. These nodes are tuned orange and are subsequently used in the inverse WMDS calculation to determine a new set of observation weights. These new weights are then used to calculate a new set of attribute weights. With each set of weights now updated, the observation and attribute projections are both updated to reflect these changes.

5.3 User Study Design

To determine the effects of visual and interaction symmetry on different analytic tasks, we first defined 3 study conditions:

- **No symmetry (NS)**, where there is neither visual nor interaction symmetry. This condition is represented by Andromeda as described in Section 5.2.1.
- **Visual symmetry (VS)**, where interaction symmetry has not yet been introduced. This condition is represented by a SIRIUS-like interface in which PrI doesn't use any importance calculations and PaI only uses 1 importance calculation instead of 2 (as represented in Figure 5.1).
- **Both visual and interaction symmetry (BS)**, which is represented by the full SIRIUS interface as described in Section 5.2.2.

By analyzing the differences between each pair of user study conditions, we are able to isolate the effects of visual and interaction symmetry on analytic tasks.

Next, we chose our participants to be students from an undergraduate data science course who had previously learned about dimension reduction techniques, including WMDS, and had used a previous version of Andromeda. This prior exposure meant participants needed less training about how to use or interpret the systems they would be using. We randomly divided participants between each of the 3 conditions to create a between-subjects study. Due to student absences, we ultimately had 21 participants for the NS condition, 19 participants for the VS condition, and 24 participants for the BS condition.

To reduce confounding variables such as time of day or location in which participants performed their analytic tasks, we designed an in-class quiz using Qualtrics. However, this

choice imposed a 1 hour time constraint, limiting the number of tasks we could include in this study. To determine precisely which tasks to include, we drew inspiration from a user study performed with Andromeda [103] and evaluated the 10 analytic tasks described by Amar et al. [5] Ultimately, we decided not to include the *Find Anomalies* task since using visualization techniques for identifying outliers is an open research area (e.g., [18, 60, 129]), which is not the focus of this study. Tasks such as *Compute Derived Value*, *Determine Range*, *Filter*, or *Sort* were also not included in this study since each of these tasks are arguably achievable if the *Retrieve Value* and/or *Find Extremum* tasks can be achieved. As such, these tasks were omitted both because neither Andromeda nor SIRIUS are specifically designed to support these tasks as well as to shorten the quiz to ensure students could complete it during the designated class period. Therefore, the quiz focused on the *Retrieve Value*, *Find Extremum*, *Characterize Distribution*, *Cluster*, and *Correlate* tasks.

To contextualize these tasks, we chose to use a subset of the animal dataset by Lampert et al. [71]. This subset omitted obscure attributes such as *New World* as well as reduced the dataset to 28 animals and 28 attributes to better support symmetry between the animals and their attributes. Then, a pair of questions were developed for each of the 5 aforementioned tasks types: one that was observation-centric and one that was attribute-centric. In addition to these 10 questions, 2 questions were developed with 4 subparts each to impose a more complex task in which each subpart was interrelated. A final question asked participants to provide 3 insights about the dataset. Specifically, these 13 questions are:

1. **Retrieve Value (Observation)**: How much does the gorilla like to forage?
2. **Retrieve Value (Attribute)**: How cow-like is the toughskin attribute?
3. **Find Extremum (Observation)**: Which animal is the most timid?
4. **Find Extremum (Attribute)**: Which attribute does the grizzly bear embody the

most?

5. **Characterize Distribution (Observation)**: Describe the distribution of the agility attribute across the different animals.

6. **Characterize Distribution (Attribute)**: Describe the distribution of the hippopotamus across the different attributes.

7. **Cluster (Observation)**: If we consider that the elephant and giant panda are similar to each other, but are dissimilar to the hippopotamus...

   (a) What other animals are like the elephant and giant panda, but not like the hippopotamus?

   (b) Which attributes describes why these other animals are similar to the elephant and giant panda but dissimilar to the hippopotamus?

8. **Cluster (Observation)**: If we consider strength and claws to be similar to each other, but dissimilar to stalker...

   (a) Which other attributes are like strength and claws but not like stalker?

   (b) Which animals describe why these other attributes are similar to strength and claws but dissimilar to stalker?

9. **Correlate (Observation)**: Which animals are solitary and do a lot of grazing?

10. **Correlate (Attribute)**: Which attributes are both gorilla-like and sheep-like?

11. **Complex(Observation)**

    (a) Which attributes separate the tiger and wolf from the blue whale and dolphin as well as from the cow and sheep?

(b) Given the answer to (a), which other animals are similar to the tiger and wolf?

(c) Given the answer to (a), which other animals are similar to the dolphin and blue whale?

(d) Which other attributes are correlated with the attributes that separate these three groups of animals?

12. **Complex(Attribute)**

(a) Which animals separate strength and walks from group and grazer as well as from claws?

(b) Given the answer to (a), which other attributes are similar to strength and walks?

(c) Given the answer to (a), which other attributes are similar to group and grazer?

(d) Which other animals are correlated with the animals that separate these three groups of attributes?

13. **Insights**: Use the tool to learn about the data. Write a list of 3 or more interesting insights you gain from the data and justify each with appropriate rationale and evidence to back up your claims.

As an example for how participants might approach these questions, we focus on the *Correlate (Attribute)* task. The easiest way to determine which attributes were both gorilla-like and sheep-like in the NS condition is to click both of these observations to perform a surface-level interaction, showing the relative values these observations have for each attribute via points along the attribute sliders (similar to how the values for *Wolf*, *Tiger*, and *Blue Whale* are shown along the attribute sliders in Figure 5.2). Participants could then look across these attribute sliders to find which attributes these animals both had higher values for. In contrast, participants in the VS condition could upweight both *Gorilla* and *Sheep* using the

"Importance" slider, which would alter the attribute projection. Participants could then look for attributes that were projected closer together and use a surface-level interaction to check whether an example attribute in a given group had a high value for either *Gorilla* or *Sheep* (similar to how the surface-level interaction in Figure 5.3 shows that *Tiger* has a high value for *Furry*). This information would reveal which attributes best describe both animals. Alternatively, participants in the VS condition could use just surface-level interactions to determine the attribute values for both *Gorilla* and *Sheep* individually and compare them either visually (by watching the blue rings around the attributes change in size) or by reading the raw values in the data panel. In contrast, participants in the BS condition would find using these surface-level interactions to complete this task easier than using the "Importance" slider. If the participant were to use this slider in the BS condition, the interaction symmetry would ultimately change the importance of these observations rather than just using the value initially defined by the "Importance" slider as it does in the VS condition (as represented in Figure 5.1). As such, the attribute projection may not change in the manner in which the participant envisioned or intended, resulting in greater difficulty in achieving this task. An example of this difference in how using the "Importance" slider for just *Gorilla* affects the visualization in the VS and BS conditions is shown in Figure 5.4.

To conduct this quiz, we developed 3 versions of a Qualtrics survey, 1 for each study condition. Each version of the survey was the same except for the URLs linking participants to the proper system as well as the training session at the beginning of the survey. This training session ensured participants were familiar with their assigned system before answering the questions on the animal dataset. As such, training sessions comprised of a demo video for their assigned system as well as an optional practice session using the Automobiles dataset from the UCI Machine Learning Repository [35] (where observations with missing data were removed). If the participant was still practicing with their system after 10 minutes,

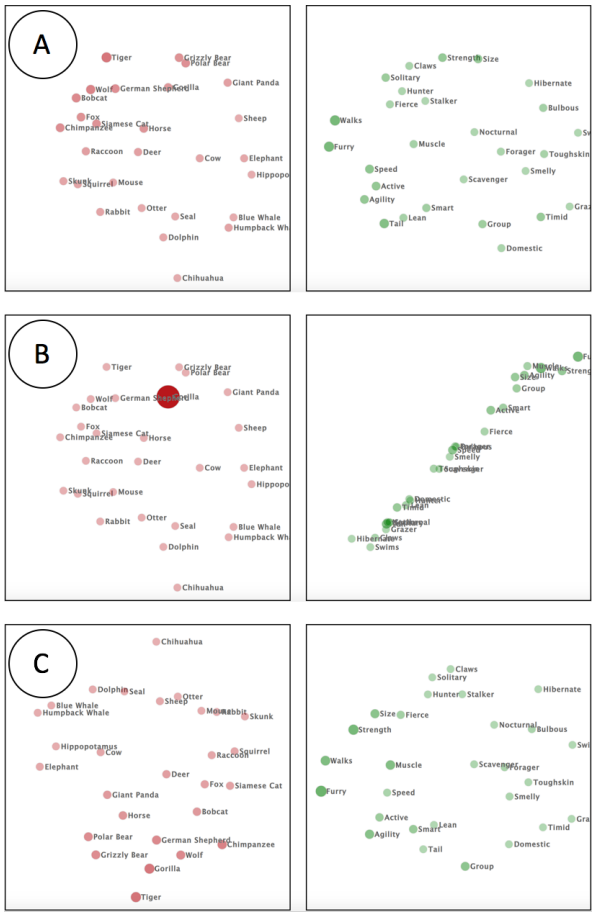

Figure 5.4: An example of how parametric interactions differ between the VS and BS conditions. After the initial projection **(A)**, the "Importance" slider for *Gorilla* is dragged up. In the VS condition, **(B)**, this specified value is directly used as the weight for *Gorilla*, where as the BS condition **(C)** recalculates the observation weights to determine which other animals are correlated with *Gorilla*.

the system automatically showed a message encouraging them to continue with their quiz. Otherwise, participants moved through the quiz at their own pace.

Each of the 13 questions of the quiz was placed on its own page in Qualtrics (including all subparts). The "back" function was disabled to take advantage of Qualtric's timing feature to obtain accurate time-on-task data. As such, the data collected from this study included time-on-task data and participants' responses for each question.

5.4 Data Analysis

With the data gathered from the Qualtrics surveys, our goal was to evaluate the effects of visual and interaction symmetry in participants' time on task and accuracy in each of the various analytic tasks as well as their cognitive cardinality and dimensionality.

5.4.1 Time on Task

To evaluate participants' time on task, we first had to account for the fact that the web server on which all 3 systems were running crashed during the study. Since participants were using Qualtrics to provide answers for each analytic task, the time on task recorded by Qualtrics was longer than how long it actually took participants to perform the required task and provide their answer. Because the crash affected all participants at the same time, we chose to first compare the total duration participants spent on their quiz across each of the conditions to provide an overview for time on task comparisons. To accomplish this, we first normalized each participants' total duration using a logarithmic transformation, which was then input into a t-test assuming unequal variances to compare the differences between each pair of user study conditions. Using a p-value of 0.05, we found no statistically significant differences in how long participants took to complete the entire quiz.

To separate our analysis by analytic task, we first determined precisely when the crash occurred and when the web server was relaunched using the `journalctl` command. With this information, we determined that the server was down for a total of 62 seconds. With this knowledge regarding the server crash, we determined which question each participant was attempting to answer when the server restarted and thus which question needed time-on-task data adjusted. We then determined if the participant submitted an answer for a question while the server was down. If so, we took this to mean that the given participant

was able to utilize part of the server's down time effectively. By comparing the timestamp for when they submitted their answer in Qualtrics against the server's restart time, we were able to determine more precisely how much each participant was truly affected by the server crashing and adjust their time on task accordingly.

Using this adjusted data, we normalized each participants' time on task for each question using a logarithmic transformation. Then, we used a t-test assuming unequal variances to compare the differences of the time on tasks for each of the 13 questions between each pair of user study conditions, resulting in a total of 39 comparisons. Using a p-value of 0.05, we show the statistically significant results in Table 5.1, where a negative number is indicative of participants in the given condition completing the given task type faster than in the NS condition.

5.4.2 Accuracy

To determine participants' accuracy in performing their analytic tasks, we focused on their responses for the 10 questions for the 5 tasks identified from the task taxonomy by Amar et al. [5] as well as the 2 complex questions. Each question (including any subparts) was graded by hand to determine whether participants were able to successfully answer their given question. For the purposes of this data analysis, grades were reduced to 0 (to indicate that the response was incorrect) or 1 (to indicate that the response was correct or mostly correct). For questions that had subparts (i.e., for the *Cluster* and *Complex* tasks), the manner in which the question was asked meant that the first subpart dictated the context for participants' answers to the remaining subparts. As such, if the first subpart was incorrect, we graded the remaining subparts in the context of the answer to the first subpart to determine if these other responses were reasonable given this context. Additionally, we realized while

Table 5.1: The statistically significant results for the time on task and accuracy data analyses. The only significant results are with comparisons to the NS condition. As such, we show the time on task results when comparing the VS and BS conditions to the NS condition as a difference in the average amount of time it took to complete a given task (in seconds). Similarly, we show the accuracy results as a difference in the average number of points participants were awarded for a given question. Instances where the participants in the given condition performed better than in the NS condition are in bold. Dashed lines are used to separate rows which concern the same analytic task.

Task	**Time on Task**		**Accuracy**	
	Visual Symmetry	**Both Symmetry**	**Visual Symmetry**	**Both Symmetry**
Retrieve Value (Observation)	32.05548621	65.34000595		
Retrieve Value (Attribute)		**-34.24095238**		
Find Extremum (Observation)	33.68002508	54.9947619		-0.363095238
Find Extremum (Attribute)			-0.373433584	-0.410714286
Characterize Distribution (Observation)		92.84421429		-0.335119048
Characterize Distribution (Attribute)			-0.36716792	-0.266071429
Cluster (Attribute)	**-244.6564612**	**-277.0289107**		
Cluster (Attribute) – subpart a				**0.214285714**
Correlate (Observation)	87.71490977	138.1306905		-0.485119048
Correlate (Attribute)		70.98960714		
Complex (Attribute)	**-169.0260602**	**-188.6180536**		
Complex (Attribute) – subpart b			**0.22556391**	**0.25297619**
Complex (Attribute) – subpart c			**0.346115288**	**0.316071429**

grading responses that some students may have been confused by the intent behind certain questions. For example, in the question for *Complex (Observation) subpart a*, students could have misunderstood that the *Tiger* and *Wolf*, *Cow* and *Sheep*, and *Dolphin* and *Blue Whale* were supposed to form 3 clusters, not 2. Thus, we considered responses correct if they separated *Tiger* and *Wolf* from the other 4 animals.

Using these grades, we then determined if 80% or more of participants within a given condition got the question right or if 80% or more got the question wrong. If so, we added two additional 0s and two 1s [2] in preparation for a z-test for proportions to compare scores for each question/subpart across each pair of conditions. This analysis resulted in a total of 60 comparisons. The statistically significant results ($p > 0.05$) are shown in Table 5.1, where

a positive number is indicative of participants in the given condition achieving a higher score than in the NS condition.

5.4.3 Cardinality and Dimensionality

In order to determine the effect of visual and interaction symmetry on participants' cognitive cardinality and dimensionality, we evaluated their responses to the final insight question at the end of the quiz, which asked participants to provide 3 insights about the dataset. For each response, we counted how many animals or attributes were explicitly mentioned. Similar to the the user study performed on Andromeda [103], generally referring to "animals" or "attributes" was assigned a cardinality or dimensionality score of 0 (respectively). However, clearly referring to a specific group of animals or attributes was assigned a number depending on the size of the given group. For example, referring to "cats" was assigned a cardinality of 3 since there are 3 cats in the dataset. Invalid responses (e.g., commenting on the quiz itself) were not included in this analysis. As such, most participants produced 3 cardinality and 3 dimensionality scores. These scores were then normalized by taking the square root of each individual cardinality and dimensionality score. The normalized scores were compared between each pair of conditions using a t-test assuming unequal variances. There were no statistically significant results ($p > 0.05$) from this analysis.

5.5 Discussion

Given the data analyses described in the previous section, we describe the implications of this user study here.

5.5.1 Time on Task

The fact that there were no significant results when comparing the total duration participants spent on their quiz meant that between the conditions, time saved on one question was spent answering another question. However, there were statistically significant results for specific tasks. Given the results shown in Table 5.1, time on task performance for participants in either the VS or BS conditions was significantly different compared to the NS condition for the *Retrieve Value (Observation)*, *Find Extremum (Observation)*, *Cluster (Attribute)*, *Correlate (Observation)*, and *Complex (Attribute)* tasks. These comparisons indicate that for these tasks, the fact that the visualization was symmetrical had the greatest impact. Evaluating these results more closely, it would appear that such visual symmetry helped participants answer questions faster for the *Cluster (Attribute)* and *Complex (Attribute)* tasks; symmetry instead slowed participants when performing the other previously mentioned tasks.

Additionally, time on task performance was better for participants in the BS condition than in the NS condition for the *Retrieve Value (Attribute)* task but worse for the *(Characterize Distribution (Observation)* and *Correlate (Attribute)* tasks. This indicates that the combined effects of both visual and interaction symmetry (but neither by themselves) helped improve or worsen time on task accordingly for these tasks.

As such, these results indicate that for *Cluster* or *Complex* tasks visual symmetry is recommended; analysts would be able to complete these types of analyses on attributes faster without slowing analyses on observations. Incorporating interaction symmetry as well would produce further benefits for these tasks. Symmetry should instead be avoided for *Find Extremum*, *Characterize Distribution*, and *Correlate* tasks since symmetry only has the potential to slow analyst down. Connecting back to our hypothesis regarding time on task,

it appears that we were correct in that symmetry improved time on task for *Complex* and *Cluster* tasks and worsened for *Correlate* tasks. However, we were incorrect in that it is just visual symmetry, not visual and interaction symmetry combined, that has the greatest influence on time on task.

5.5.2 Accuracy

In our analysis of participants' accuracy, we find that participants in either the VS or BS condition performed significantly differently when compared with the NS condition for the questions on *Find Extremum (Attribute)*, *Characterize Distribution (Attribute)*, *Complex (Attribute) subpart b*, and *Complex (Attribute) subpart c*. Of these, participants in the VS or BS condition scored higher than participants in the NS condition for both of the aforementioned subparts for the *Complex (Attribute)* question, indicating that the visual symmetry was helpful for these questions. In contrast, the visual symmetry worsened participants' accuracy for the other previously mentioned questions.

However, participants in the BS condition also had statistically significant differences in their scores when compared with in the NS condition for the questions on *Find Extremum (Observation)*, *Characterize Distribution (Observation)*, *Cluster (Attribute) subpart a*, and *Correlate (Observation)*. These results indicate that the combined effect of visual and interaction symmetry that improved accuracy when investigating observation correlations but hindered when performing these other tasks.

Given these results, visual symmetry is recommended for supporting better accuracy in *Complex* tasks as analysts would be able to perform these types of tasks on attributes more accurately without decreasing accuracy for tasks on observations. A combination of visual and interaction symmetry may similarly assist analysts in performing such tasks but is not

required to see such benefits in performance. However, such a combination of symmetry is recommended for *Cluster* tasks since this combined symmetry improves accuracy for such tasks on attributes without decreasing accuracy for observations. Symmetry in any form should instead be avoided for *Find Extremum*, *Characterize Distribution*, and *Correlate* tasks since symmetry only has the potential to decrease accuracy for such tasks.

In relation to our hypothesis regarding similarity between time on task and accuracy results, we see that there such similarities do exist. Namely, it is visual symmetry, as opposed to interaction symmetry or their combination, that best characterizes the effects on participants' performance. However, symmetry more negatively affected accuracy than time on task in *Find Extremum* and *Characterize Distribution* tasks. Additionally, symmetry had less of a negative impact for accuracy than for time on task in *Correlate* tasks. This implies that time on task and accuracy cannot always be considered equally when determining whether to implement symmetry to support certain tasks, as described further in Section 5.5.4.

5.5.3 Cardinality and Dimensionality

Since there were no significant results between either of the user study conditions for cardinality or dimensionality, this means that neither visual nor interaction symmetry had a role in how many observations or attributes participants used in their analytical reasoning. As such, supporting cognitive cardinality or dimensionality should have no influence when deciding whether to incorporate visual symmetry, interaction symmetry, or both in a given system. Additionally, our hypothesis that symmetry would support an increase in cognitive cardinality and dimensionality appears to be incorrect.

5.5.4 Broader Implications

In comparing the results from these data analyses, we see that visual symmetry most positively impacted the *Cluster (Attribute)* and *Complex (Attribute)* tasks, helping improve both time on task and accuracy. Thus, systems that seek to assist analysts with these sorts of tasks should employ visual symmetry, especially since there is no worsening in performance in the observation-centric counterparts for these tasks. In contrast, visual symmetry slowed time on task and lowered accuracy in the *Find Extremum (Observation)* and *Correlate (Observation)* tasks, indicating that visual symmetry should be avoided when attempting to support these tasks.

However, it is interesting to note the relative degree of impact visual symmetry had in participants' performance. Focusing first on time on task, these tasks where visual symmetry improved participants' performance had a relatively large impact compared to tasks where performance worsened. For example, visual symmetry improved time on task by 245 seconds for the *Cluster (Attribute)* task, while this symmetry worsened time on task by 88 seconds for the *Correlate (Observation)* task. This means that for the tasks that visual symmetry assists in, it may save 3 times more time than what is lost in a task that it hinders. In comparison, the relative degree of impact for accuracy shows a different pattern, where the increase in accuracy caused by visual symmetry in the *Complex (Attribute)* task does not always make up for how accuracy is worsened in other tasks, such as *Find Extremum (Attribute)*. As such, a system that supports tasks that are both assisted and hindered by visual symmetry deserve careful consideration to determine the tradeoff in implementing such symmetry.

5.6 Limitations and Future Work

Despite the interesting findings of this user study, it is important to contextualize them in the limitations of the study itself. First and foremost, participants were comprised of students who had prior exposure to Andromeda. This prior exposure may have assisted participants in the NS condition to complete their tasks faster or more accurately than they would have if they had to learn how to use their assigned system as the participants in the other conditions did. For example, participants performed better on the *Retrieve Value (Observation)* task in the NS condition than in either of the other conditions. In light of the improved performance in the VS and BS conditions compared to the NS condition for the *Retrieve Value (Attribute)* task, we suspect this difference in performance may be influenced by the fact that participants in the VS or BS conditions had to learn different surface-level interactions to extract the same observation-related information (but were then better supported in extracting attribute-related information). As such, it is worth repeating this study with participants who are either all familiar or all unfamiliar with both Andromeda and SIRIUS to remove familiarity with these systems as a potentially confounding variable.

If the results of this second user study are similar to our results here, it may be that Andromeda supports certain tasks better than either version of SIRIUS due to other notions of usability beyond symmetry. For example, in Section 5.3, we described how participants might approach performing the *Correlate (Attribute)* task. It may be that participants simply have an easier time using and understanding Andromeda's surface-level interactions to perform this task than SIRIUS's interactions, regardless of any prior exposure to Andromeda. For instance, how well did participants in the BS condition understand that using surface-level interactions would be easier than trying to use the "Importance" slider? This hints at potential usability issues with SIRIUS (even in spite of a demo video that attempted to highlight these types of nuances) which deserves more direct analysis than this user study

on analytic task performance provides.

Additionally, the time constraint impacted both the construction of the analytic tasks in the quiz and potentially influenced participant responses due to time-related stress. As such, performing a user study that directly tests all 10 analytic task types and removes the time constraint may reveal different results. For example, it may be that there were no statistically significant results for participants' cardinality and dimensionality due to time-related stress, which led them to write shorter/simpler responses than they may without a time constraint. Similarly, the fact that there were no statistical differences in the total duration participants spent on the quiz may have meant that participants self-imposed a maximum amount of time they would spend attempting to answer a given question to help ensure they would complete the quiz in the allotted time. As such, this time constraint could have also affected how much time participants actually spent on different questions on the quiz.

While participants were performing the study itself, the server crashed, as mentioned in Section 5.4.1. While the total down time for the server was short, it was long enough for some participants to begin talking with each other, leading to some participants being more heavily impacted than others by the crash. Moreover, we realized after the fact that participants talking with each other like this may have led some participants to share their URLs with each other as they attempted to reconnect to the server. If such sharing were occurred, this would have led to participants using a different system than the one they were assigned to, thereby skewing our results for their assigned condition. Unfortunately, there is no method for definitively determining whether this URL sharing may have occurred with the data we gathered.

Finally, it is worth noting that this study only focused on one type of data. If the study was repeated using a different dataset (e.g., documents and extracted terms), results may be very different. Therefore, the generalizability of the results obtained from this user study

may be determined by performing similar studies using a variety of datasets.

5.7 Conclusion

In this chapter, we presented a user study evaluating the impacts of visual and interaction symmetry on a variety of analytic tasks. A total of 64 students who were randomly split between each of our 3 user study conditions: no symmetry, only visual symmetry, and both visual and interaction symmetry. All participants took the same quiz in the form of a Qualtrics survey, which was comprised of 5 analytic tasks and a complex task for both observations and attributes. With the addition of a question regarding insights about the data, there were a total of 13 questions in the survey.

In evaluating participants' cognitive cardinality and dimensionality in analytical reasoning, we did not find any statistically significant results. Thus, supporting cognitive cardinality and dimensionality should have no influence on the decision for whether to implement symmetry in a given system. However, there were statistically significant results for both time on task and accuracy when comparing participants' performance in either the VS or BS conditions to the NS condition. Generally, these results revealed that visual symmetry has the biggest impact on participants' performance, assisting in tasks such as *Cluster (Attribute)* and *Complex (Attribute)*, and hindering in tasks such as *Find Extremum (Observation)*,and *Correlate (Observation)*. Since the *Cluster (Observation)* and *Complex (Observation)* tasks were not hindered by symmetry, visual symmetry is generally recommended for supporting *Cluster* and *Complex* tasks. As such, interaction symmetry is not necessary to see benefits in performance in these tasks. In contrast, any form of symmetry should be avoided for *Find Extremum* and *Correlate* tasks since symmetry only has the potential for hindering such tasks. These results mean that if a system is designed to support a mix of tasks that are

both assisted and hindered by such symmetry, careful consideration is needed in determining whether to employ visual symmetry in a system.

Chapter 6

Modeling the Sensemaking Process with Semantic Interaction

6.1 Introduction

The overarching goal of this chapter is to computationally augment human sensemaking capabilities in the context of big text analysis problems. For example, intelligence analysts must forage large collections of text for relevant information and synthesize a coherent story from fragments. Such sensemaking activities are modeled by Pirolli and Card's Sensemaking Process [90], which is composed of two primary, interconnected loops: the Foraging Loop and the Synthesis Loop. Traditionally, much of this sensemaking activity, especially synthesis, requires human cognitive intelligence. However, to efficiently scale up sensemaking to big data, more semi-automated augmentation is needed. To support the human cognitive activity, it is important that the automation fits naturally into the human sensemaking workflow.

The Sensemaking Process is a cognitive model. Thus, to support automation, one challenge is to concretize this process into a more computationally-oriented model with formalized subcomponents. In this chapter, we formally model the Sensemaking Process as an interactive data structuring process, and the Foraging Loop as an interactive relevance model driven by the result of the structuring model. A related challenge is the high-dimensional nature

of text data, which makes it difficult to support real-time, interactive structuring methods. Our approach is to exploit topic modeling methods to reduce dimensionality between the foraging and the synthesis models.

Yet a further challenge in enabling this automation lies in the human-centered, interactive, and iterative nature of sensemaking. For example, in the "dual search" process [90] that connects synthesis and foraging, analysts simultaneously identify hypotheses that synthesize the supporting evidence while also foraging for additional evidence for the hypotheses. Through iteration, analysts *incrementally formalize* [106] their hypotheses and arguments. To support this user-driven nature of the models, we exploit the principles of semantic interaction [41] to steer semi-supervised machine learning algorithms, updating the models based on learned user interest. Semantic interaction methods seek to learn users' cognitive sensemaking intents by observing their interactions, such as their interactive structuring activities in the Synthesis Loop. This enables analysts to stay focused on their familiar Sensemaking Process rather than thinking about manipulating underlying statistical models. For our computational sensemaking model, this requires designing machine learning "inverses" (as described in Chapter 3 and in [103]) for the synthesis and foraging models that learn from user's structuring and searching actions. To support high-dimensional text data, the topic modeling approach therefore also needs to interactively update based on semantic interactions.

To address these challenges, we designed a sensemaking computational pipeline (summarized by Figure 6.1), embodied in a novel visual analytics system for big text called Cosmos (Figure 6.2). Specifically, our contributions are:

1. Computational modeling of the Sensemaking Process using a semantic interaction pipeline to connect synthesis models to foraging models. The pipeline makes use of a user interest model based on weights on document terms and topics, which are learned

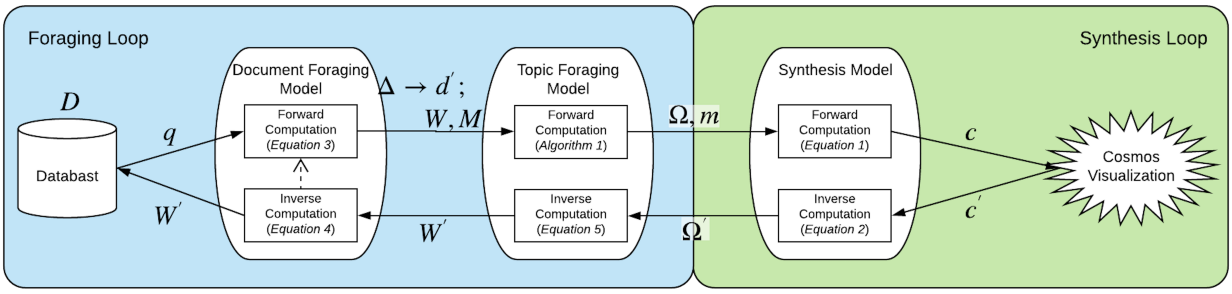

Figure 6.1: A computational representation of how the Sensemaking Process [90] can be supported for big text analytics, following the conventions for depicting semantic interaction from Chapter 3. This pipeline is annotated with the variables from Table 6.1 to show the transformation of data throughout the pipeline, including which equations and algorithms we use in our prototype implementation of Cosmos.

via semantic interaction feedback.

2. Modeling the synthesis process as an interactive dimension-reduction spatialization in which users can express similarity relationships in collaboration with the user interest model.

3. Modeling the foraging process in two parts that collaborate with the user interest model:

   (a) a document foraging process that filters documents (acquired from a search engine) based on relevance to the user interest model,

   (b) and a dynamic topic foraging process that reduces dimensionality and updates in the presence of user interaction.

6.2 Sensemaking Pipeline for Big Text

The Sensemaking Process described by Pirolli and Card is comprised of two main sub-loops: the Foraging Loop and the Synthesis Loop [90]. Thus, we model the Sensemaking Process

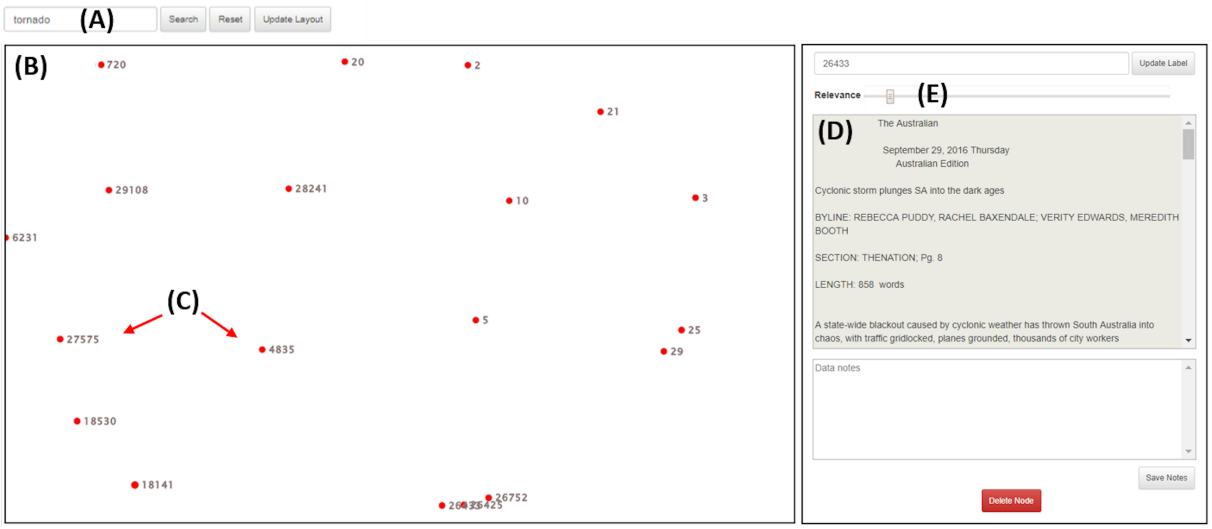

Figure 6.2: An overview of the Cosmos system. **(A)** Analysts use keyword search foraging with a text field to begin populating **(B)** the synthesis visualization of the foraged subset of documents. **(C)** Documents within the visualization are projected according to similarity to each other. To the right of this visualization, **(D)** a selected document's text can be read in a scrolling panel. Just above, **(E)** the document's relevance and label can be updated.

by combining models for foraging and synthesis processes into a computational pipeline represented by Figure 6.1. For reference, Table 6.1 describes most variables used throughout the equations in this chapter.

6.2.1 Synthesis Model

From the analyst's perspective, the ultimate goal of sensemaking with big text is to synthesize information to formulate and support a hypothesis. This places particular emphasis on the Synthesis Loop, in which the analyst must explore relationships between documents and determine the relevance of information gathered. This is often accomplished by iteratively examining information, organizing it, and returning to the Foraging Loop to gather more. This process may also include testing alternative hypotheses, or rejecting or refocusing the hypothesis as additional information is synthesized. Thus, the Synthesis Loop is accom-

Table 6.1: A list of variables used throughout this chapter and their descriptions. Variables that appear with a ' indicate a change or update to that variable.

Variable	Description
D	Full set of documents in the corpus
q	Set of documents returned by a query to D
Δ	Set of relevant documents to add to the visualization; $\Delta \in q$
d	Set of documents to be visualized; $\Delta \cup d = d'$
c	Set of low-dimensional coordinates for each $d_i \in d$
R	A vector representation of relevances for all documents in q
T	All terms in D
t	Topics learned from d
M	A $d \times T$ matrix that describes each $d_i \in d$ in terms of each $T_i \in T$
m	A $d \times t$ matrix that describes each $d_i \in d$ in terms of its probability of belonging to each $t_i \in t$
W	A vector representation of weights on all terms in T
Ω	A vector representation of weights on all topics in t

plished iteratively over time. As such, a model of the synthesis process must support this continuous exploration and organization of information.

Specifically for text datasets, we propose that there are two main methods for synthesis: leveraging document relevance or using document similarity. Representing similarity spatially for sensemaking has proven to be intuitive and powerful in other visual analytics systems [12, 15, 42, 59, 62, 102, 120, 126], including those with big text [13]. These methods for visualizing similarities also enable exploration of different similarity and dissimilarity relationships by manipulating a weight vector. Thus, we propose that the Synthesis Model should be a projection method that takes document-topic matrix m and topic weight vector Ω to produce coordinates c in the visualization. This concept can be represented by the equation $c = Synthesize(m, \Omega)$. Ω thereby forms the first necessary component of a user interest model that represents how interested the analyst is in each topic, with topics defined by another model in the pipeline.

6.2.2 Foraging Models

The Synthesis Loop becomes difficult for analysts to perform when there is too much information for the analyst to organize manually all at once. This leads analysts to direct their attention to the most relevant information to their investigation first and then forage for additional, related information that is also relevant to the investigation to either support or refute their hypothesis.

The use of relevance to forage for additional documents resulted in modeling one foraging component with a Document Foraging Model. This model filters documents to ensure that only highly relevant documents are displayed, focusing synthesis on just the documents relevant to the investigation. Such filtering is represented by the equation $R = DocRel(q)$. By applying thresholds, relevant documents Δ (and therefore visualized documents d') can be determined.

In addition to foraging for specific documents, analysts may wish to forage based on a topic, t_i, that appears in the dataset. Indeed, analysts performing keyword search foraging often relate certain keywords with each other (e.g., synonyms or co-occurrences). Therefore, we argue that keyword search foraging is actually performed based upon topics of interest rather than one specific keyword. Additionally, using term-based representations of documents is problematic due to the sparsity of the data (i.e., most terms do not occur across many documents), leading to documents being represented by a vector consisting of mostly zeros. This sparsity complicates the Synthesis Model, hindering its ability to scale to large datasets. Reducing the data to a "medium"-dimensional space by transforming sets of terms into topics permits the Synthesis Model to run more efficiently, even as the number of documents visualized increases.

To accomplish this translation of terms to topics, a Topic Foraging Model is also necessary.

This model is responsible for dynamically detecting topics in the visualized documents based on the equation $m, \Omega = TopicForage(M, W)$. In other words, term weight vector W forms the second necessary piece of the user interest model by capturing how interested the analyst is in each term.

Performing this topic foraging on only the visualized documents and not the entire dataset keeps the foraged topics closer to the analyst's notion of which topics exist in the dataset. This is because the analyst only knows of the subset that have already been investigated. Another benefit is that which documents to forage can be based on a set of weights on terms rather than just on the single term that is queried. This can produce richer results from the Document Foraging Model for the Topic Foraging Model to utilize, culminating in more relevant topics for the Synthesis Model to visualize. Using topic weights to reflect analyst interest in each topic results in a more accurate reflection of the analyst's notion of document similarity. Thus, the results of these models produce a succinct yet accurate representation of relevant documents for the analyst to synthesize via a similarity-based projection.

6.2.3 User Interest Model

To support the Sensemaking Process, the foraging and synthesis models must *learn* from the analyst's interactions and respond according to their sensemaking activity. This learning means the analyst's interest must be modeled (i.e., a user interest model). Closer inspection of the Foraging and Synthesis Models reveals an interesting feature: all three can be tied together using an interest model that is centered on W and Ω. Alterations to W and Ω can be accomplished by *learning* from user interactions. Following semantic interaction techniques, this learning can be accomplished by pairing each model's computation that helps produce the visualization with an inverse computation. This inverse computation *learns* the analyst's

intent and updates the interest model appropriately. While this concept is more thoroughly discussed in Chapter 3, we note our use of a semantic interaction pipeline to represent our sensemaking pipeline in Figure 6.1.

An example of this inversion is when an analyst drags documents within the projection to redefine their similarities/dissimilarities. This interaction then triggers a learning process to determine a new Ω that describes the analyst-defined layout. Thus, $\Omega' = Synthesize^{-1}(c', m)$. This supports the analysts as their notion of similarity changes between investigations or throughout the course of a single investigation while their hypothesis becomes more refined. Analysts could also perform a semantic interaction to assert a desired relevance for a selected document, $M_{i,*}$, as described by $W' = DocRel^{-1}(R'_i, M_{i,*})$ for a user-specified relevance R'_i. These interactions leads to the Document Foraging Model *learning* a new set of term weights that best mirrors the analyst's interest.

These interactions could also be leveraged to perform semantic interaction foraging, an automated foraging technique defined by Wenskovitch and North [125] that queries for new documents on behalf of the analyst. Which documents are foraged is based on the interest model (specifically the term weights derived from topic weights using $W' = TopicForage^{-1}(m, \Omega'))$.[1]

As a result, these interaction techniques – namely, (1) dragging documents within the projection to express synthesized relationships, or (2) adjusting the relevance rating of documents to express foraging feedback – provided through the synthesis and foraging models provide a natural interface to large-scale text data. By displaying documents most relevant to the analyst's investigation, we avoid overwhelming the analyst. By using semantic interaction foraging, we *learn* which documents may also be relevant to the analyst, and automatically add them into the visualization to further assist in synthesizing information.

[1]Note that M is not part of $TopicForage^{-1}$ since which terms appear in which documents never changes, meaning it is not necessary to include M in this inverse equation.

6.3 Example Prototype

Here, we describe a prototype based on our sensemaking pipeline for big text.

6.3.1 Design Goals

In developing this prototype, we note a number of high-level design goals, with design choices relating to each model described in the following subsections. The primary design goal is to provide a simple prototype interface that naturally reflects the analyst's Sensemaking Process. The emphasis on a simple prototype means that Cosmos is meant to demonstrate how the different models of the pipeline support sensemaking activities as opposed to being a fully-functional system ready for thorough usability evaluation.

The goal of intuitive use implies an interface that avoids needing knowledge of the underlying algorithms to interact with the interface, thereby enabling the analyst to remain focused on their synthesis processes rather than trying to learn the mathematical underpinnings for this specific system [38, 41, 42]. The tradeoff in achieving this goal is that the details of the mathematical algorithms or user interest model will be hidden; there is no method for analysts to directly access this kind of information, including W and Ω. If the analyst feels that the visualization is missing information or has an inaccurate representation of the documents (i.e., an inaccurate W or Ω, or missing documents), then they use one of Cosmos's interactions to rectify the situation.

An additional goal is to help analysts focus on their synthesis tasks. As previously mentioned, this is the part of the Sensemaking Process that analysts particularly excel in.Therefore, the similarity-based projection of the documents (the ultimate product of the Synthesis Model) dominates the visualization to assist with synthesis tasks. While a text field is also provided

to take notes on a single document to further support this goal, we acknowledge that analysts may draw a variety of relationships between documents and want to externalize them in the form of a report. Such relationships may include content similarity, when they occur in relation to each other, coverage of a specific topic, and others [8, 12]. Rather than focusing on supporting all possibilities in this final phase of the Sensemaking Process, we focus on demonstrating how a simple prototype of our proposed pipeline supports sensemaking tasks with big text, allowing analysts to perform report writing activities using other mediums outside of Cosmos (e.g., a word processor, hand-written notes, a flow chart, etc.). In particular, we focus on synthesizing by spatially organizing and grouping document icons in a 2D space. However, this prototype can easily be augmented to better support other forms of synthesis that may be involved report writing activities, as discussed in Section 6.5.

6.3.2 Interface and Interactions

In order to accomplish the aforementioned design goals alongside supporting the necessary components implied in the pipeline itself, we developed the web-based visualization depicted in Figure 6.2. A similarity-based projection of the documents is the most prominent component of the visualization (Figure 6.2-B), which reflects the system's focus in supporting the analyst's synthesis processes. Here, we project each document $d_i \in d$ such that more similar documents are projected closer together and dissimilar documents farther apart (Figure 6.2-C). In recognition of the fact that analysts may also wish to leverage document relevance, we map the relevance, R_i, of each d_i to the radius of the document's corresponding node in the projection. To the right of this projection, detailed information is provided for a single selected document, including the document label, its calculated relevance, and the document's contents (Figure 6.2-D & E). A text field is also provided for the analyst to externalize notes on a selected document. Both the field to view the document's contents and to take notes

Table 6.2: A list of additional variables used to describe the algorithms in our prototype implementation and their time complexities.

Variable	Description
N	The total number of documents in D or $\|D\|$
n	The total number of documents in d or $\|d\|$
P	The total number of terms in T or $\|T\|$
k	The total number of topics in t or $\|t\|$

scroll to fit their contents, allowing these fields to take up a fixed amount of space in the visualization while still scaling to varying amounts of text.

Within Cosmos, a number of interactions are afforded. Following interactions to alter the similarity-based projection, document nodes smoothly transition from one location to another, with new nodes appearing in one corner of the projection and transitioning to their specified location. The simplest interaction is keyword search foraging, which is enabled through a search box above the document projection (Figure 6.2-A) and results in a higher weight for that term in W. Then, new topics, t, and their weights, Ω are learned, causing the projection to update. The corresponding document node can also be clicked, causing the document's label, relevance value, contents, and notes to populate the area to the right of the projection (Figure 6.2-D & E).

Three semantic interactions are also afforded, which are enabled by learning new term weights, W, followed by calculating new topic weights, Ω, to update the projection. The first is to delete a node from the projection by clicking the "Delete Node" button below these fields. This decreases the term weight, W_i, for each term in that document. Two semantic interactions trigger automated foraging: OLI (dragging document icons) and manipulating the relevance slider. OLI always triggers semantic interaction foraging, but foraging after manipulating the relevance slider is conditional. When the relevance slider is increased, this is interpreted as analyst interest in the given document, implying a wish to see more similar documents. In contrast, decreasing the relevance slider, much like deleting a document node,

only informs the system of what the analyst is *not* interested in, and does not provide enough information for what the analyst *is* interested in to perform such foraging. How our prototype enables these interactions is described in the remaining subsections, with additional variables used described in Table 6.2.

6.3.3 The Synthesis Model

To develop our Synthesis Model, we drew inspiration from interactive dimension reduction systems like Andromeda [102], SIRIUS (from Chapter 4), and InterAxis [62] to support synthesis as an interactive process performed within a similarity-based projection of a small set of documents using WMDS [69]. We chose WMDS due to its ability to enable analysts to express their synthesis process through manipulation of document proximities to reflect their perceived similarity. Also, WMDS supports a variety of similarity metrics in both high- and low-dimensional space. Thus, WMDS enables us to fulfill the goals of the Synthesis Model while affording us the flexibility to define the high-dimensional similarity, $dist_H$, as a weighted Euclidean similarity and the low-dimensional-similarity, $dist_L$, as the projected Euclidean similarity. In Equation 6.1, each document is represented as a single row of m, or $m_{i,*}$. Implemented as an iterative algorithm, this equation has a time complexity of $O(n^2k)$ per iteration.

$$c = \underset{c_1,\ldots,c_n}{\arg\min} \sum_{i=1}^{n-1} \sum_{j>i}^{n} \left(dist_L(c_i, c_j) - dist_H(\Omega, m_{i,*}, m_{j,*})\right)^2 \tag{6.1}$$

We enable analysts to change these topic weights through OLI to denote the perceived similarity/dissimilarity between them. The Synthesis Model can then *learn* new topic weights (i.e., amount of interest) using the following equation:

$$\Omega' = \underset{\Omega'_1,\ldots,\Omega'_k}{\arg\min} \sum_{i=1}^{n-1} \sum_{j>i}^{n} \left(dist_L(c'_i, c'_j) - dist_H(\Omega', m_{i,*}, m_{j,*})\right)^2 \tag{6.2}$$

Note that this equation effectively inverts Equation 6.1, now learning topic weights Ω rather than projection coordinates c. This inversion is achieved via gradient-based iterative minimization. As the objective's derivative takes $O(n^2k)$ operations to evaluate and must be evaluated for each of k variables to optimize, the time complexity of this algorithm is $O(n^2k^2)$.

After learning new term weights from these topic weights the term weights can be leveraged to perform semantic interaction foraging [125], thereby automatically revealing additional relevant information after OLI. Further details on how we accomplish semantic interaction foraging in our prototype are provided in Section 6.3.4 and Section 6.3.5, with an example provided in Section 6.4.2.

6.3.4 The Document Foraging Model

In addition to traditional keyword search foraging, our Document Foraging Model enables semantic interaction foraging to automatically bring additional, relevant documents into the visualization after semantic interactions like OLI. To create this model, we drew inspiration from StarSPIRE [12], which uses a simple yet effective calculation for document relevances combined with thresholds to determine which of the foraged documents to display in the projection [125].

In our implementation, all foraging is accomplished through queries to an Elasticsearch database [49]. Each query to the database has a time complexity of $O(1)$ since the documents within the database are indexed and hashed. After receiving the query results, the model then determines the top 10 documents that are above a fixed relevance value[2]. This ensures that only highly-relevant documents are displayed while also guaranteeing that the analyst

[2]These thresholds were set based on our particular dataset and use case, which are described in Section 6.4. For other datasets or use cases, these thresholds may be altered to display greater or fewer new documents at each iteration.

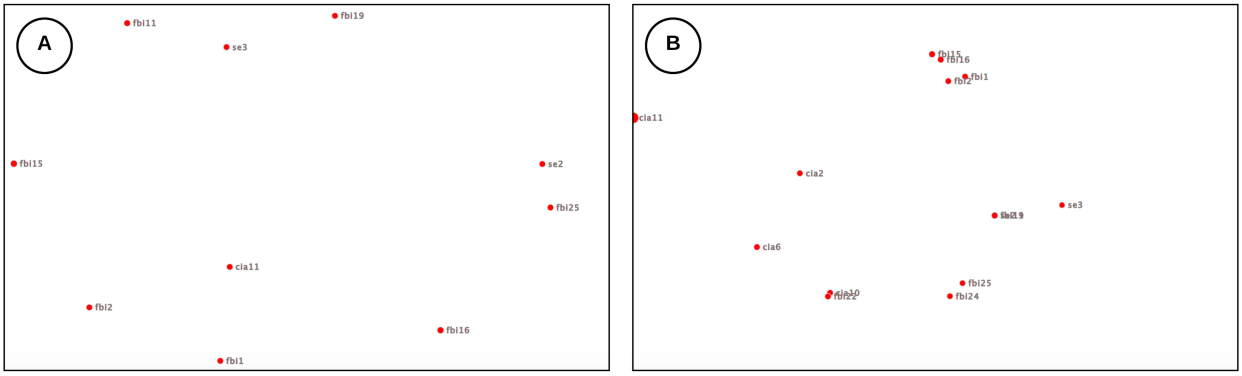

Figure 6.3: A depiction of how the interaction with the relevance slider in Cosmos works. After a query for a person's name, **(A)** 5 documents appear in Cosmos. One of these documents (cia11) seems particularly relevant to the analyst's investigation. After increasing the relevance of this document using the slider, **(B)** new documents related to the document the analyst interacted with appear using semantic interaction foraging. The node sizes and positions in document projection are also updated based on this interaction.

will not be overwhelmed by too many documents appearing at once. Thus, the analyst is provided with a simple yet effective interface to large scales of text data.

Our relevance computation represents each document as a vector of TF-IDF values, which effectively represents the document data as a Bag of Words or Vector Space Model [45]. These TF-IDF values combined with term weights leads to the following equation to compute the relevance of a single document, represented as a single row of document-topic matrix M or $M_{i,*}$:

$$R_i = M_{i,*}^T W, \tag{6.3}$$

Initially, these term weights are set to 1, and they update to reflect their level of importance to the analyst's investigation. Since this process is repeated for each document, the time complexity of this computation is $O(nP)$.

To support the semantic interaction of manipulating the relevance slider, we must determine how to calculate the specified relevance value, R_i' for a given document. This necessitates an inverse computation to determine new term weights to produce R_i'. The following equation

performs this computation, where $M_{i,*}$ represents a single row of document-topic matrix M:

$$W' = W + M_{i,*}\frac{(R_i - R'_i)}{M_{i,*}^T M_{i,*}} \tag{6.4}$$

This equation rescales term weight vector W by another vector proportional to the document's TF-IDF values, $M_{i,*}$, whose relevance is being changed from R_i to R'_i. The time complexity of our implementation of $DocRel^{-1}$ is $O(P)$.

With the new term weights, automated foraging is performed, followed by recalculating the relevance for all displayed documents using Equation 6.3. After learning new topics via Algorithm 1, the document projection is updated. Thus, manipulation of the relevance slider triggers semantic interaction foraging enabled by the Document Foraging Model *learning* which terms are most important to the analyst's investigation. An example of this interaction is depicted in Figure 6.3. Semantic interaction foraging after OLI is similarly handled by the Document Foraging Model once the Topic Foraging Model translates topic weights to term weights, as described in the next subsection.

6.3.5 The Topic Foraging Model

To transform a term-based representation of documents into a topic-based representation usable to the Synthesis Model, our Topic Foraging Model produces a vector of probabilities that mirror the prevalence of each topic in each document. This requires learning which terms belong to which topics, expressed as a probability distribution across terms. Given the terms that appear in each document, inferences can be made on the topic probabilities of that document.

We generate this topic-based representation of documents using Latent Dirichlet Allocation (LDA) [11].We chose LDA because it enables us to represent the prevalence of top-

ics within each document, which provides more fine-grained information to the Synthesis Model. Specifically, we use a weighted modification of the uncollapsed variational algorithm presented by Blei et al. [11] which has a time complexity of $O(nkP)$ per iteration. We fix the number of topics, k, to 5 for the purposes of our prototype[3]. The generative model for the topics and document contents is shown in Algorithm 1. Each topic, represented as a simplex valued variable, is denoted as t_i. We denote the vector containing document lengths as N, and $z_{i,j}$ a latent variable indicating which topic the j'th word of document i references. The proportion of each topic in document d_i, also simplex valued, is denoted by $m_{i,*}$. t_i and $m_{i,*}$ are each endowed with Dirichlet prior distributions with exchangeable concentration using parameters η and α respectively, which represent the number of times each word or topic reference was observed *a priori*. Once estimated, the $m_{i,*}$ for each document is passed to the Synthesis Model to use in projecting the documents in Cosmos.

Algorithm 1 Generative Model for LDA

for $i = 1 : k$ **do**
    $t_i \sim \text{Dirichlet}(\eta)$
for $i = 1 : n$ **do**
    $m_{i,*} \sim \text{Dirichlet}(\alpha)$
    for $j = 1 : N_i$ **do**
        $z_{i,j} \sim \text{Multinomial}(m_{i,*})$
        $M_{i,j} \sim \text{Multinomial}(t_{z_{i,j}})$

These topics are also given weights to use in the Synthesis Model to denote the analyst's levels of interest across the different topics. Initially, these weights are set to 1. However, our foraging technique necessitates weights on terms rather than on topics. In such cases where new term weights, W', must be learned from topic weights, Ω, we use the following

[3]In general, choosing the number of topics is a challenging task and an open research problem (see, e.g., [22]). Given the goal of the Topic Foraging Model is to mirror the analyst's perception on the number of topics along with research suggesting people generally have difficulty thinking in more than 2–3 dimensions simultaneously [103], we chose a smaller k.

formula:

$$W_i' = t_i' \Omega \tag{6.5}$$

Repeating this equation for each W_i' transfers importance to words which are most probable in topics with a time complexity of $O(Pk)$.

6.4 Use Case Scenario

This example demonstrates the use of Cosmos in a realistic scenario to accomplish a sense-making task, showing how the synthesis and foraging models can interact jointly to accomplish an analytical goal. The scenario centers on a dataset comprised of over 30,000 documents, primarily consisting of news articles of varying lengths. These documents were obtained from LexisNexis [1] using a keyword search on "adelaide" to create the foundation of a scenario focused on the worst series of storms that hit South Australia in 50 years [110]. These storms occurred on 28 September 2016. Additionally, 30 hand-picked articles were added to the dataset that relate specifically to the storm scenario.

In response to reports of tornadoes and severe weather, the United States is planning to send humanitarian assistance to Adelaide, Australia [53]. A hypothetical analyst is tasked to assess the impact of these storms from the news dataset and to determine the level of support needed. This example analysis took approximately 75 minutes to complete, including reading approximately 20 documents that each averaged 700 words and interacting with Cosmos. Given that the longest interaction took about 4 seconds for the server to respond (with most interactions taking 1–2 seconds), this means that the vast majority of the analyst's time was spent reading and synthesizing the information within the documents. However, it is important to note that a different analyst may take a different approach in the investigation, leading to a different set of initial insights regarding the impact of the storms.

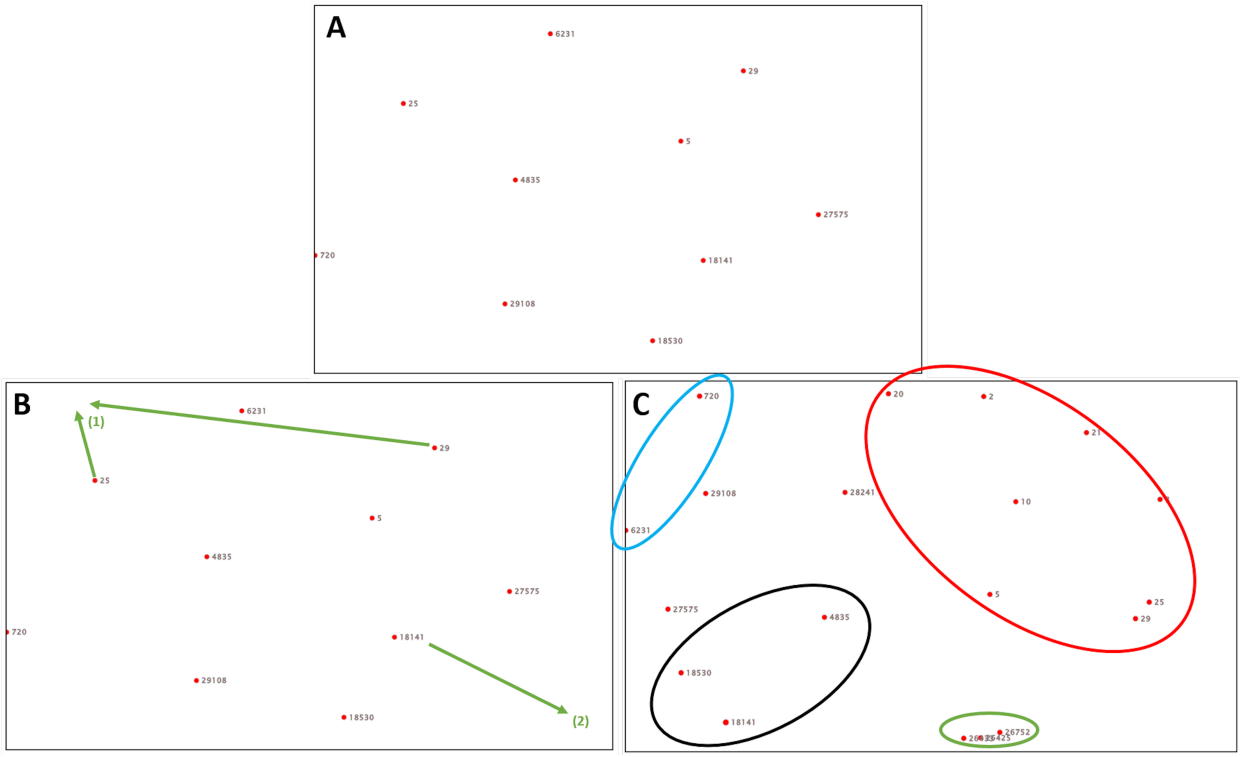

Figure 6.4: After searching for the term "tornado," Cosmos **(A)** visualizes an initial set of documents. The analyst then **(B)** uses OLI to express the perceived similarities/dissimilarities between documents, resulting in **(C)** an updated document projection that includes new documents from semantic interaction foraging.

6.4.1 Initiating the Investigation

To begin the investigation, the analyst queries documents using the search term "tornado." Ten foraged documents then populate the document projection (Figure 6.4-A). Three of the documents (181411, 18530, and 4835) reference previous storms. One document (720) is about a sports team called the "Tornadoes." Another document (6231) references a person named "Adelaide" and is clearly unrelated to the storms. Two other documents (29108 and 27575) mention the storm but do not focus on it. The final three documents (29, 25, and 5) are all planted documents related directly to the storm.[4]

[4]All hand-picked documents have an ID less than 30.

6.4.2 User-Driven Synthesis Modeling Using OLI

The analyst begins to form an initial mental model about the storms, focusing on how some discuss previous storms and others describe the recent storms of interest. To clarify this distinction, the analyst uses OLI to express these perceived similarities/dissimilarities between the documents. This is accomplished by moving the document nodes related to the recent storm (25 and 29) together in one corner (Figure 6.4-B-1) and a document node referencing previous storms (1814) in an opposing corner (Figure 6.4-B-2). The analyst then clicks "Update Layout," which triggers the Synthesis Model to *learn* which topics in the dataset best reflect these similarities/dissimilarities. After the Topic Foraging Model *learns* new terms weights, the Document Foraging Model uses these term weights to automatically forage for new documents to add to the visualization.

6.4.3 Exploring Foraged and Synthesized Documents

The resulting document projection (Figure 6.4-C) includes nine new documents from the semantic interaction foraging triggered by OLI. Documents related to the storm are in the red group. The unrelated documents in the blue group are farthest from the red group. The historical storm documents in the black group are slightly closer but still separate. All newly foraged documents were related to the storm due to the Document Foraging Model's ability to identify their relevance from the OLI made by the analyst. The visual structure created by the Similarity Model reinforces the analyst's mental model about how these documents relate to each other and helps in quickly sorting through the newly added documents (which were placed in the red and green groups).

The analyst now explores the new documents in the important red group and the nearby (and therefore similar) green group. The analyst reads one of the newly foraged document

nodes (26433) from the green group and finds out that the storms have caused statewide power outages (Figure 6.5-Left). Based on this, the analyst determines that disaster relief will need to consider massive power outages in addition to assistance for the physical damage that the storm has caused. Since this document does not directly reference tornadoes, this insight was possible due to semantic interaction foraging triggered by OLI.

Following this insight, the analyst investigates another document in the same group (26425) to find that areas north of Adelaide are also without power and are suffering from extensive flooding (Figure 6.5-Right). Therefore, disaster relief must also consider flood-related issues, such as people trapped in their homes. This discovery is particularly unique since this document is not directly focused on the storm and would likely not be found simply reading through storm reports. Additional insights can be gained by continuing to read and interact with the documents on the screen, which may again trigger semantic interaction foraging and quickly uncover relevant information on the storm's impact.

6.5 Discussion

Cosmos provides a foundation for future research in sensemaking for big text. As shown in Section 6.4, our prototype enables analysts to investigate large sets of documents and begin to quickly draw conclusions from them. Our multi-model approach to forage for documents and allow analysts to synthesize this information is at the core of our approach. We next discuss some limitations of the Cosmos system and describe future work to resolve the issues that we have uncovered.

Firstly, we recognize that the concepts we chose to model (as represented in Figure 6.1) may not be optimal. For example, there may be other concepts that better reflect an analyst's notion of synthesis than similarity-based methods. Alternatively, perhaps more models are

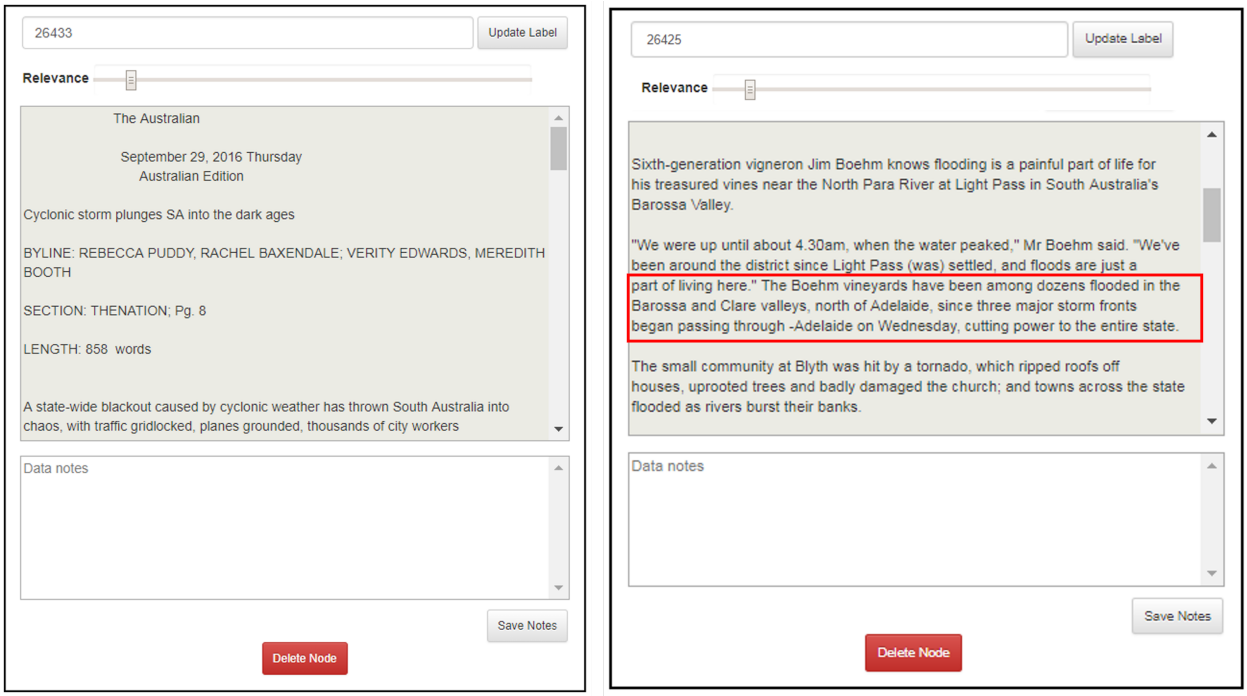

Figure 6.5: The contents of foraged documents in Cosmos that reveal how the storms have impacted Adelaide and surrounding areas.

needed to properly capture the complexity of the analyst's foraging and synthesis processes. These types of alternatives warrant further investigation (perhaps through comparative user studies) regarding the tradeoffs in different computational models and visualizations, and whether they accurately embody components of the Sensemaking Process.

We also note several limitations to our Cosmos prototype. Beginning with the visualizations, the time complexities of WMDS (Equation 6.1) and its inverse (Equation 6.2) limit the number of documents that can be efficiently projected into the display and interacted upon. We are already researching optimizations of both of these equations to permit visualization and interaction on even larger sets of documents. Similarly, we recognize that altering the relevance threshold would impact the documents added to the visualization. We plan to reformulate the Document Foraging Model to enable variable relevance thresholds, allowing the analyst to control how many documents should be foraged and the density

of the projection. This can be accomplished via direct user input (e.g., slider bars) or by developing new semantic interactions that *learn* these parameters.

Additionally, our implementations of each model were based on our previous research in the area [12, 102, 125] as well as algorithm commonality, simplicity, and flexibility. We acknowledge that there are many alternative methods for implementing the same model components (e.g., using t-SNE [120], PCA [131], LAMP [59], or other similarity-based methods in place of WMDS; Rocchio [97] or PageRank [85] can replace our relevance calculations; and LSA [55] or clustering algorithms could be leveraged as alternatives to LDA).

We can address these limitations regarding how we implemented Cosmos by performing user studies using the iteration of Cosmos as described in this chapter against alternative versions. These other versions would implement existing concepts in different manners (e.g., using other similarity-based projections), perhaps drawing inspiration from other systems that support synthesis or foraging models (e.g., TIARA [123]). Such user studies would be immensely informative for future visual analytics systems for big text by helping researchers understand the tradeoffs implied by different modeling and implementation methodologies.

So far, we have focused on supporting spatial structuring forms of synthesis. An interesting future opportunity is to explore other forms of synthesis that are incorporated in the final step of the Sensemaking Process: report writing. In this final step in the Sensemaking Process, the analyst must synthesize all relevant text into a narrative. Such a narrative may involve a number of relationships between documents, as indicated in Section 6.3.1. Thus, to support externalizing such a narrative, Cosmos should be extended to incorporate additional visual components or interactions.

Other useful extensions to Cosmos include additional visualization components to depict information from the Topic Foraging Model to allow analysts to see and interact with the top

terms in the top topics or the top terms among the visualized documents. Such interactions may enable analysts to manipulate the term or topic weights more directly. Alternatively, the document projection can be augmented with additional information, such as uncertainty in the projection resulting from Equation 6.1. This may help the analysts accurately understand the projection itself. Interaction in such a document projection can allow the analyst to express uncertainty, providing additional feedback for the underlying algorithms to learn from. Cosmos could also be augmented with a map visualization that would act as a geographical filter for which documents to forage, thereby also augmenting the user interest model with a geographical component. Finally, Cosmos might also be extended with a collaborative mode, allowing multiple analysts to interact with the same data. Such collaborations may help analysts reach conclusions faster as a team, resulting in higher confidence in the results and assisting with disseminating the information learned.

6.6 Conclusion

This chapter introduced a new computational model of the human Sensemaking Process to enable systems that support interactive big text analytics. This model takes the form of a pipeline, which is comprised of a series of smaller computational models that mirror the Foraging and Synthesis Loops within the Sensemaking Process [90]. Our models ultimately center on a Document Foraging Model, Topic Foraging Model, and Synthesis Model. By leveraging a joint user interest model, these models are connected and interactive, allowing the analyst to iteratively investigate the dataset and dynamically refine their investigation. We demonstrate a prototyped implementation of these models through the creation of Cosmos, a visual analytics system for big text data. We described the mathematics and functionality of each implemented model in detail, and demonstrated how they support the

exploration of a 30,000 document collection with a realistic use case.

Chapter 7

Exploring Design Challenges for Semantic Interaction Foraging

7.1 Introduction

As datasets become increasingly large and complex, it becomes more difficult for analysts to find relevant information to perform a given task. For example, showing an overview of the data first is often recommended [108], but large datasets impose challenges both computationally and visually. A natural solution to this issue is to highlight a subset of the data that is relevant to the analyst's investigation. Indeed, such a solution directly supports the Foraging Loop, a crucial component within the analyst's Sensemaking Process [90]. For instance, performing a keyword search on "cats" for relevant documents should reveal only documents that discuss cats. These documents are often also rank-ordered based on how relevant each document is to the subject of "cats." As such, recommender systems facilitate the Foraging Loop by automatically highlighting potentially relevant data to the analyst based on their preferences or interactions.

Analysts' true prowess is arguably shown through their synthesis abilities rather than foraging. Especially compared to computers, analysts are better at making a variety of complex connections between data, including through real-world knowledge external to the dataset itself. In contrast, if a computer knew how to determine which data were relevant to the

analyst, the computer could automatically show this data without the analyst needing to explicitly ask for it, thereby enabling analysts to focus on their synthesis tasks. Thus, such automated foraging techniques have the potential to greatly facilitate analysts' Sensemaking Process.

One example of an automated foraging technique is **Semantic Interaction Foraging (SIF)** [125], which relies on **Semantic Interaction (SI)**. In SI, the system *learns* which data is important to the analyst based on how they interact with the system [40]. This learning is accomplished by interpreting the analyst's intent behind a given interaction and translating this intent into updates to model parameters (as described in Chapter 3). In so doing, the analyst's understanding of the system is abstracted from the underlying model parameters, enabling the analyst to focus on their Sensemaking Process rather than leaving their cognitive zone [106] to directly alter these parameters. With the knowledge gained from SI, the system can then estimate how relevant additional, unseen data may be to the analyst and highlight the most relevant data. Thus, the goal of performing SIF in this manner is to select a subset of data to highlight to the analyst based on a user-centered interest model, thereby assisting the analyst's Foraging Loop. SIF has already proven to be supportive of the analyst's Sensemaking Process, even when the metric for relevance is simplistic [125].

Despite how useful SIF is, the fact that it's a relatively new automated foraging technique (to our knowledge, only appearing in [125] and in Chapter 6) means that there is little guidance on how to actually implement SIF, even when given a system that already supports SI. What are the design challenges that must be considered in order to implement SIF in such a system? What are the open research questions related to these design challenges? To address these questions, our contributions in this chapter are:

1. A list of design challenges to guide implementations of SIF in a given system, using a system called Centaurus as an example.

2. A series of open research questions related to the design challenges, which form a research agenda for SIF.

3. A user study that showcases an initial investigation into the research question regarding when to use SIF.

7.2 Centaurus

In this section, we describe Centaurus, a prototype, symmetrically-designed system based on SIRIUS (described in Chapter 4) that has been minimally adapted to include SIF, as represented in Figure 7.1. In this figure, the presence arrow going from the Document and Term Foraging Model back to the CSV + Flat File Data Controller represents Centaurus's ability to forage for additional data. Our implementation of SIF relies on both a relevance threshold as well as a top n threshold to determine which subset of unseen documents should be added to the display. Given this approach to the development of Centaurus, we propose this system as a testbed for research into each of the SIF design challenges as opposed to a truly novel system itself. Here, we provide an overview of Centaurus to contextualize the subsequent discussion on each of the design challenges (described in the next section) as well as the user study that was performed on Centaurus, as described in Section 7.4.

7.2.1 Overview

Just as in SIRIUS (as described in Chapter 4), Centaurus (Figure 7.2) displays both observations and attributes by using similarity-based projections. Thus, similarity between two observations or attributes is visually encoded in their proximity to each other in the associated panel. Additionally, node size and opacity operate as dual encodings for relevance to

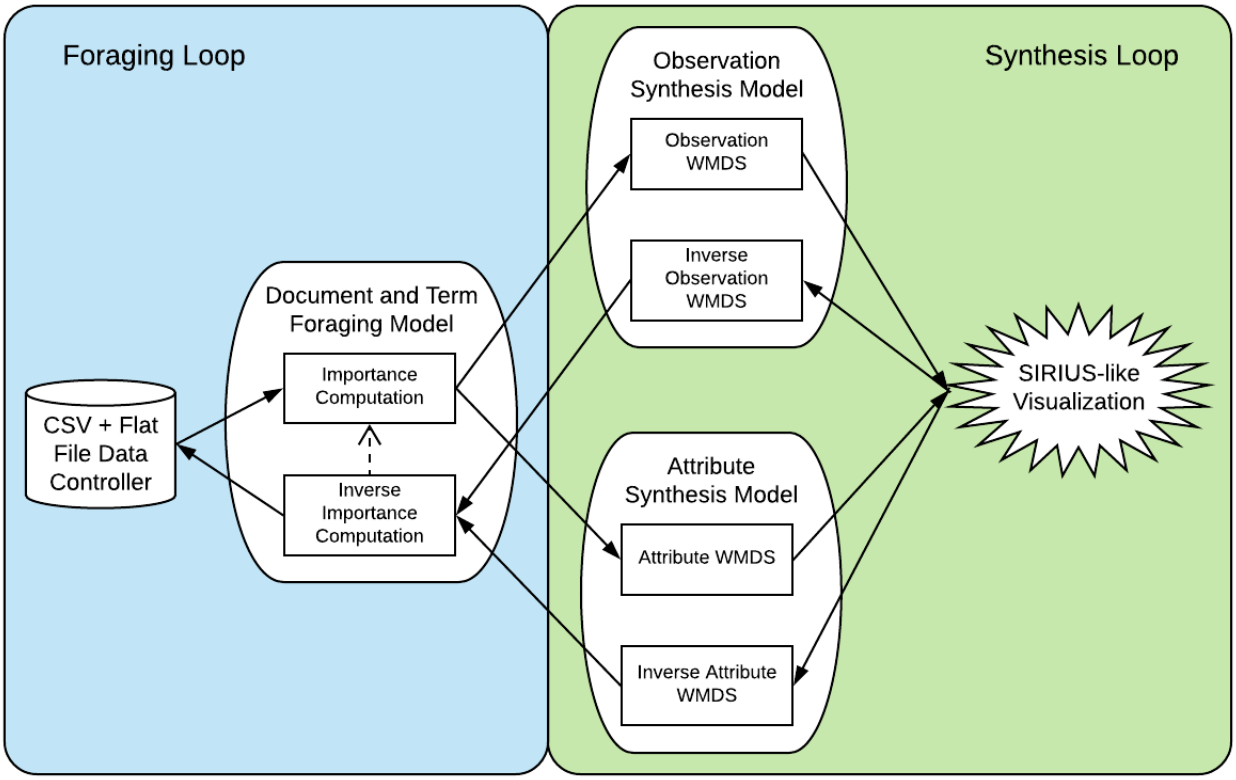

Figure 7.1: A computational pipeline for Centaurus that combines pipeline representations from Chapter 3 and Chapter 6.

allow the analyst to readily determine which observations or attributes are most relevant to their current investigation. Node color is generally used to distinguish between newer nodes (lighter colors, like *fbi7*, *fbi17*, *Sprint*, and *University of Virginia* in Figure 7.2) and older nodes (darker colors). However, an additional orange color, such as for *fbi15* in Figure 7.2, denotes a node that will be used in a PrI (an example of SI in Centaurus) after the analyst clicks the "Update Layout" button.

When an analyst double-clicks a node in either the observation panel or attribute panel, the node gains a darker border to differentiate it from the remaining nodes, as seen with *fbi11* in Figure 7.2. Additionally, the corresponding data fills the panel beneath these projections, including the estimated importance (i.e., relevance), associated raw data (which is also shown

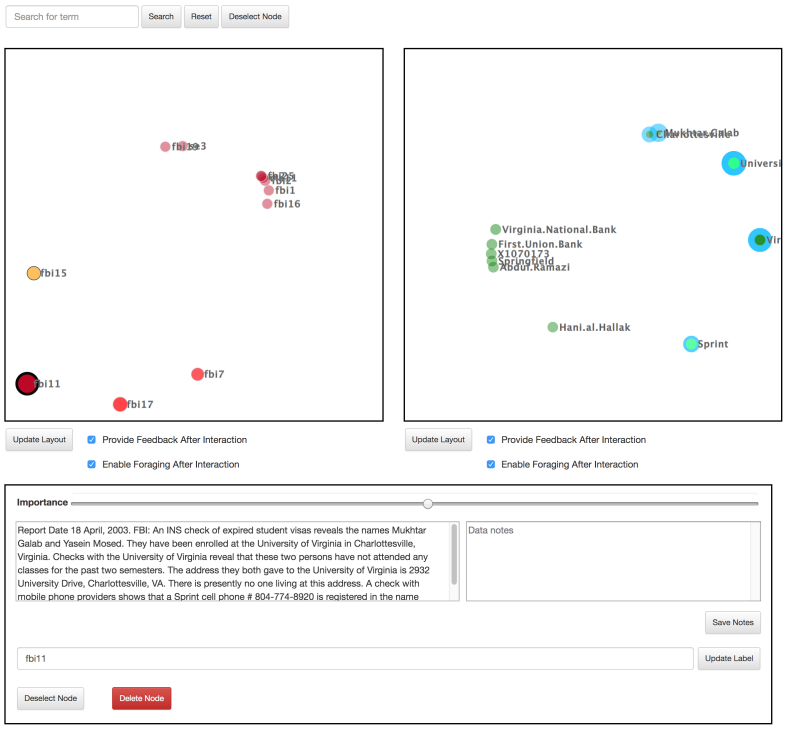

Figure 7.2: An overview of Centaurus, a prototype, symmetrically-designed system that enables semantic interaction foraging (SIF) through projects both observations (i.e., documents in the left panel) and attributes (i.e., terms in the right panel). The visual encodings for Centaurus are described in Section 7.2.1.

visually using colored borders in the opposing panels, as exemplified with the blue borders around the attribute nodes in Figure 7.2), and node label. Manipulating the importance slider is an SI in which the analyst directly communicates to the system a desired relevance for the given observation/attribute. Additional SIs in Centaurus include searching for a specific observation or attribute and deleting an observation or attribute. Each SI and how they influence SIF is described in more detail in the "*Interactions*" subsection.

7.2.2 Example Analysis with Centaurus

In the original description of SIRIUS in Chapter 4, an intelligence dataset called *The Sign of the Crescent* [57] was displayed, as represented in Figure 4.6-A. This dataset is meant to

simulate a scenario in which the analyst must piece together information across multiple documents within a corpus that contains both relevant and irrelevant documents. In total, the dataset is comprised of 41 documents, from which 275 terms have been extracted. However, the projections in SIRIUS were difficult to read due to the visual clutter caused by both the fact that each extracted term typically does not appear across many documents as well as the number of extracted terms in the data. In this section, we provide an example analysis with this same dataset to demonstrate the prowess of SIF in a symmetrically-designed system.

Throughout this dataset, there are 3 main terrorist plots: one in which terrorists are pretending to be students at the University of Virginia to bomb an Amtrak train in Georgia, one where terrorists working in a ship's galley are going to bomb the ship once it arrives in Boston, and a final plot where terrorists will bomb the New York Stock Exchange under the guise of vendors. Across this data set, 15 documents ($\approx 35.7\%$) are completely irrelevant to either of these 3 plots.

An analyst uses Centaurus to evaluate the dataset and uncover terrorist plots, beginning with the knowledge that Abdul Ramazi is someone suspected of being connected to terrorist activity. Therefore, the analyst starts by searching for *Ramazi*. As a result, 5 documents mentioning Ramazi, the *Abdul Ramazi* attribute itself, and 2 additional attributes are foraged (as shown in Figure 7.3-A). The analyst reads the returned documents, beginning with those represented by larger nodes as this indicates the system believes these documents are more relevant to the analyst.

In reading the document *fbi15*, the analyst learns that Mukhtar Galab has been receiving money from Ramazi. By double-clicking on the node, the importance slider becomes available to interact with, which is then dragged up to denote the document's higher level of importance to the analyst. This triggers SIF for 5 attributes and 2 additional documents, *fbi11* and *fbi19*, as reflected in Figure 7.3-B. Reading these documents reveals that Galab is

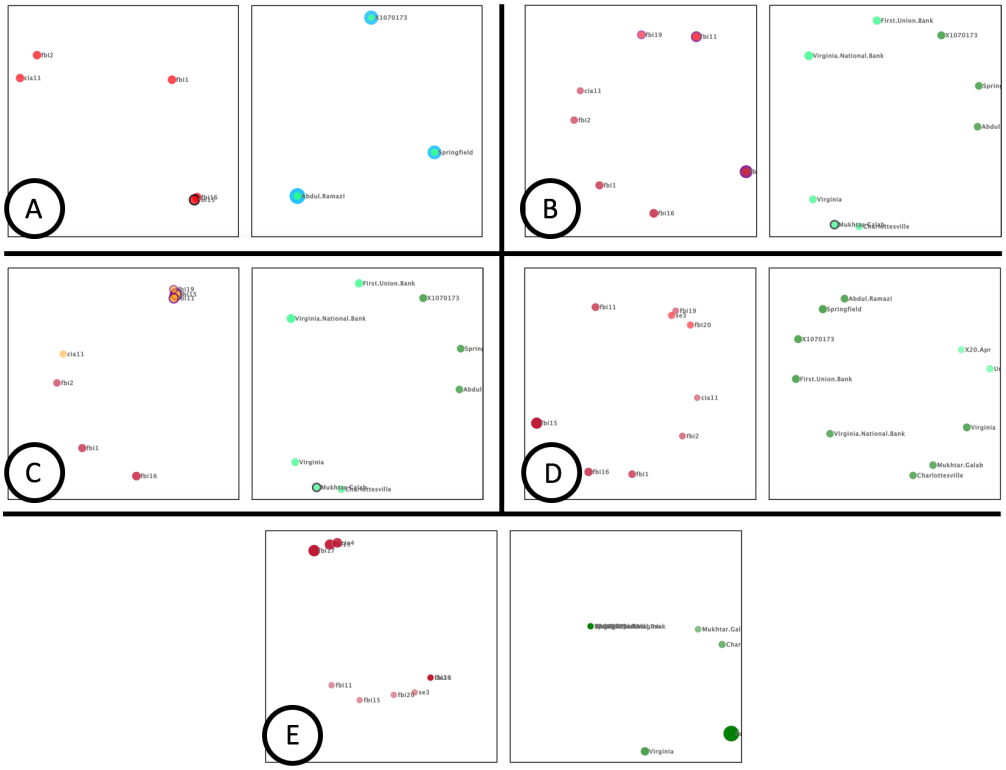

Figure 7.3: An example analysis with Centaurus, beginning with a search for *Ramazi* (*A*). After reading the new documents, the analyst first focuses on information related to *fbi15* and moves the importance slider up for this document (*B*). Upon investigating the newly foraged documents, the analyst chooses to perform PrI (*C*) to attempt to differentiate between relevant and irrelevant documents. However, the resulting SIF (*D*) does not return immediately relevant documents or terms. After deleting an irrelevant term, the analyst searches for *Goba*. The resulting display (*E*) highlights a fourth relevant document, which enables the analyst to complete their identification of one of the three terrorist plots in the dataset.

a registered student at the University of Virginia along with Yasein Mosed, but neither have attended classes for two semesters. Both of their student visas have expired, and neither of them are living at the address on their file. Additionally, Galab and Mosed as well as Faysal Goba have train tickets to travel to Atlanta, GA on April 29.

Suspicious of Galab and his associates, the analyst wishes to differentiate the documents regarding Galab (*fbi15*, *fbi11*, and *fbi19*) from the remaining documents. To accomplish this, the analyst drags the three relevant documents together and highlights *cia11* (a document

represented by a small node and contains no information about Galab, Mosed, or Goba) as a contrasting point, as shown in Figure 7.3-C. By clicking "Update Layout," 2 documents and 2 attributes are added to the display by SIF (as depicted in Figure 7.3-D).

While these new documents and terms do not reveal the type of information the analyst is looking for, the analyst can still perform a keyword search for *Goba* to more directly indicate to the system which data is desired. The system then uses SIF to find the remaining 2 documents on Goba as well as 2 additional attributes, which are then added to the display. However, this interaction results in the attribute *Faysal Goba* ultimately receiving a lower weight than the analysts desires. Thus, the analyst double-clicks the corresponding node and drags up the importance slider, resulting in the display depicted in Figure 7.3-E. Among the large nodes in the upper-left corner of the document panel is *cia4*, which describes how Goba is actually Ziad al Shibh and trained at an Al Qaeda explosives training facility in the Sudan in 1994.

Combining all the information learned from the documents together, the analyst reports a terrorist plot to destroy the train that Galab, Goba, and Mosed have tickets to ride on April 29. As such, the analyst was able to determine one of the 3 terrorist plots using a small series of interactions that incorporated SIF. Although this dataset is small and despite the fact that SIF did not always show documents relevant to the analyst's investigation, 2/4 automatically foraged documents (excluding the initial 5 documents regarding *Ramazi*) were irrelevant to this plot. Thus, we emphasize how SIF helped uncover half (5/10) of the documents relevant to this particular terrorist plot in only 4 interactions. From these documents, the analyst had enough information to determine who the terrorists were, what their plan was, and when they would act out said plan. Thus, we assert that SIF helped the analyst in their task while avoiding clutter, which is consistent with the findings from the original study on SIF [125]. We discuss how Centaurus might be further improved to reduce

these instances where SIF did not show relevant data in Section 7.5.

7.2.3 Interactions for SIF

In Centaurus, there are a total of 4 SIs:

1. Searching for a document or term by name
2. Manipulating the importance slider
3. Projection interactions
4. Deleting a document or term from the display

Here, we highlight each of these interactions and how they influence SIF in turn. To aid in this discussion, we provide Figure 7.4 as a visual representation of the steps taken to perform SI and SIF using a set of observation (i.e., document) weights, W_O, and a set of attribute (i.e., term) weights, W_A. We describe our rationales for choosing our ranking algorithm and relevance thresholds in Section 7.3.

Searching for a Document or Term

At the top of the display is a search bar (see Figure 7.2) in which the analyst may type all or part of the name of an observation or attribute. For text datasets, this results in explicitly foraging for any matching documents or terms and increasing the corresponding weight(s) in either W_O or W_A, respectively.

Using this explicitly foraged data, other data is also foraged, thereby ensuring the additional data is relevant to the initial query. For example, if the search matched the name of a term

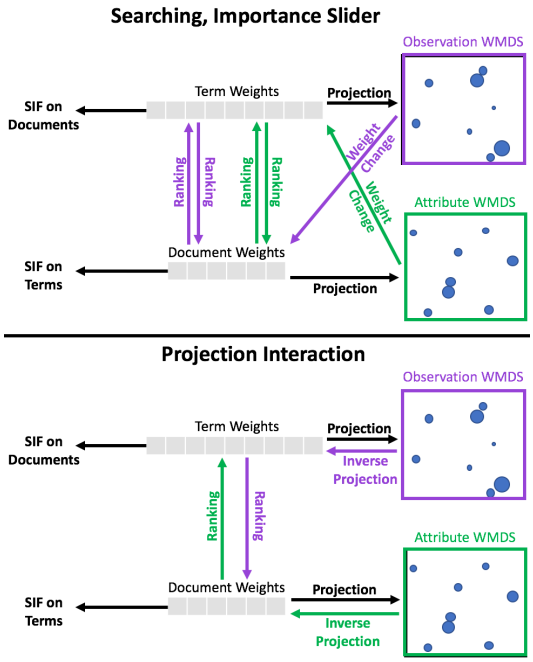

Figure 7.4: A depiction of the mathematical and foraging processes used to perform SI and SIF in Centaurus. When increasing an importance slider or searching for an observation/attribute, the associated weight is directly manipulated. Using this new weight vector, the first instance of foraging occurs. Using this new data, the second weight vector is then redefined using Equation 7.1 or Equation 7.2. This new, second weight vector is then used to perform a second instance of foraging. To be able to compare this set of newly foraged documents/terms with existing ones, Equation 7.2 or Equation 7.1 is used, respectively. In contrast, PrI triggers an "inverse projection" algorithm to determine a new set of weights. These weights are used to perform the first instance of foraging. Then, Equation 7.1 or Equation 7.2 is used to redefine the second weight vector, which is then used to perform the second instance of foraging. Specific examples of these interactions are described in Section 7.2.

(e.g., "cat"), then W_A is altered, and the additional data foraged is comprised of documents (e.g., documents about cats). With these new documents in hand, the new W_A is used to estimate the relevance of each document (via Equation 7.1, as described in Section 7.3). Using these relevance values, the cat documents are then ranked. At this particular point, the system has a high level of confidence that the documents found are relevant to the analyst based on the nature of the interaction itself (i.e., the analyst explicitly asked for data about "cat") and the manner in which these relevant documents were foraged. Therefore, the top

n threshold is set to allow more data to appear on the screen (specifically, up to 5 new documents). Together with the relevance threshold, a subset of documents to add to the display is determined.

However, we also want to find additional terms to add to the display. To accomplish this, we use the relevances of the new documents to estimate the relevances of unseen terms with Equation 7.2 (i.e., the new cat documents are used to automatically forage for additional terms, like "furry" and "cute"). However, the system is less confident in the relevance of this newly learned information since this information is inferred. Therefore, the top n threshold is lowered to return only 2 terms. Thus, a single interaction results in foraging for both documents and terms. As a final step, W_O via Equation 7.1 is updated followed by an update to W_A via Equation 7.2 so that the relevance of old document/terms and new documents/terms incorporate the relevances of the newly foraged data and are properly represented in comparison with each other. This process is similar for when the analyst instead searches for a document name, as depicted in Figure 7.4.

Manipulating the Importance Slider

When an document or term is double-clicked, information related to that document/term populates the data panel, including the estimated relevance (via the importance slider), raw data, and label. When the importance slider is dragged up, this directly indicates that the analyst feels that the given document/term is more relevant to them. Using this analyst-specified value, the corresponding value in W_O or W_A is set. From here, the process of foraging for both documents and terms and updating the weight vectors is similar to when searching for a document or term.

If the importance slider is instead dragged down, then the same manipulation of the two

weight vectors takes place to learn from this SI. However, no SIF takes place since the information learned is that some data is less relevant to the analyst. Thus, the information provided by this interaction is insufficient to determine what *is* relevant to the analyst with a high enough level of confidence to perform SIF. This could also be thought of as the top n threshold being lowered to 0 such that no documents or terms are added to the display.

Projection Interactions

To perform a PrI, the analyst directly manipulates the panel of documents or terms to denote desired similarity/dissimilarity relationships within that projection (e.g., separating documents about dogs from documents about cats). After the analyst clicks an "Update Layout" button, the system learns the weight vector to produce that projection (i.e., W_A). Using either Equation 7.1 or Equation 7.2 the second weight vector is also updated (i.e., W_O). Unlike when searching for a term or manipulating the importance slider, there is no clear indication from the analyst regarding what is important to them; all of the information learned from this interaction is considered an inference. Therefore, the top n threshold is set to return at most 2 documents and 2 terms from this interaction.

Deleting a Document or Term

When deleting a node from the display, the corresponding weight in either W_O or W_A is set to 0. Similar to searching for data or manipulating the importance slider, the second weight vector is then updated, followed by a final update to the original weight vector. For example, if a document is deleted, then its weight in W_O is set to 0. Then, W_A is updated to reflect this change in the document weight vector using Equation 7.2. However, it is likely that documents similar to the deleted one are also less relevant. Therefore, a

final translation of W_A back to W_O via Equation 7.1 leads to a reduction in weight of any such similar documents. SIF is not triggered after this interaction because the information learned is what is *not* relevant to the analyst; there is not enough information regarding what *is* relevant to the analyst to perform SIF with a high level of confidence.

7.3 SIF Design Challenges

In this section, we discuss the design challenges (DCs) for implementing SIF, along with related open research questions that together formulate a research agenda in SIF. These design challenges are also annotated to indicate whether they predominantly concern a computational component of the system (COMP) or a visual component (VIS). Inspiration for these design challenges stems from research on StarSPIRE [12] and Cosmos (as described in Chapter 6) as well as the process of developing Centaurus, as these are the only systems to our knowledge that implement SIF. These systems are used to exemplify each design challenge.

7.3.1 Basics of SIF

To incorporate SIF in any given system, there are several design challenges to guide the implementation. These include:

1. (COMP) How should relevance be measured?
2. (COMP) How should confidence in the information learned from SI influence SIF?
3. (COMP) How can the level of confidence in the information learned from SI be determined?

4. (COMP) How can the system determine when the analyst is ready for SIF?
5. (COMP) What if the system was wrong about the relevance of the information it returned by SIF?
6. (VIS) How should new data affect the display?
7. (VIS) Which data should be initially displayed to the analyst?

We discuss each of these design challenges in turn. Note that the context of SIF highly influences any implementation decision, including the system and data being used as well as the different forms of SI enabled. Therefore, these design challenges are meant to form a guideline for how to implement SIF rather than a firm set of rules that should be followed, especially considering the numerous open research questions related to these challenges.

DC1 (COMP): Relevance Must Connect to SI and Thresholds Must Be Chosen

In Section 2.5, we discussed how SIF seeks to only show a subset of data. Defining a relevant subset of data requires employing a ranking algorithm along with thresholds to ensure the data returned is the most relevant to the analyst. Given how crucial determining such a subset of data is to SIF, a ranking algorithm that reflects the relevance of the data must be determined. However, **it is critical that the ranking algorithm use information learned from SI**, whether directly or through some derivation thereof. In so doing, the analyst's interaction with the system will influence the data returned by SIF, which is one of the main goals of SIF. Thus, choosing an appropriate ranking algorithm is highly dependent on the data that is used, the analytical tasks that should be supported, and the information learned from SI. The fact that choosing a ranking algorithm is highly contextual means that this design challenge can offer little guidance to choose a specific algorithm, but there are

many potential algorithms, such as those described by Yang [136]. However, this choice of algorithm may inform what the relevance threshold or top n threshold should be to determine a reasonable subset of data to display to the analyst.

Because the importance calculations in SIRIUS rely on information learned from SI, we use these calculations for our ranking algorithm in Centaurus. Thus, our relevance metrics for observations (i.e., documents) and attributes (i.e., terms) are as follows, where W_O is the observation weights (or estimated relevances), W_A is the attribute weights, O is the data for visualized observations, and A is the data for visualized attributes:

$$W_O = O \bullet W_A \tag{7.1}$$

$$W_A = A \bullet W_O \tag{7.2}$$

The resulting relevance values are then ranked to form our ranking algorithms, thereby directly connecting the information learned from a given interaction to the subsequently foraged data. Note that these equations are similar to the ranking algorithm used in Cosmos, as described in Chapter 6.

In conjunction with these equations, Centaurus (similar to Cosmos) uses both a top n threshold and a relevance threshold to specify the subset of data to display, as exemplified in the "*Example Analysis with Centaurus*" subsection. Given the nature of these ranking algorithms, the average estimated relevance for documents/terms will decrease as more documents/terms are added to the display. Therefore, our relevance threshold must be dynamic. We chose our threshold to be $1/20N$ for observations and $1/20M$ for attributes, where N is the number of observations currently displayed and M is the number of attributes currently displayed[1]. In the context of the user study described in Section 7.4, we found that

[1]If N or M is 0 (i.e., no observations or attributes are currently being displayed), we set the variable to 5, or the maximum number of observations/attributes returned by SIF (as discussed in DC2), to avoid a

these values seemed to ensure that any documents or terms that might be relevant would be displayed while still omitting those whose estimated relevance is too low. However, more research is needed to determine concretely where these thresholds should be set, as described in DC2 and DC6.

Additionally, we apply a top n threshold to ensure the analyst is not overwhelmed by the number of observations or attributes being added to the display from SIF. This is a particular concern because adding new observations/attributes affects the appearance of existing observations/attributes in the display (as discussed more thoroughly in DC6). Depending on the specific SI and the inherent level of confidence the system has in the information learned, this dynamic threshold changes. We describe the specific influence of confidence in the next two design challenges.

Given the necessary connection between the ranking algorithm and information learned from SI in order to perform SIF, a research question that this design challenge highlights is, are there any ranking algorithms that are incompatible with SI (and therefore cannot be used with SIF)? How many algorithms or which types of algorithms can easily be incorporated with SI/SIF (such as how we used the algorithms from SIRIUS to implement SIF)? For those that necessitate deriving additional information from SI to use in the ranking algorithm, how computationally complex might such derivations be? What are the ideal thresholds for a given ranking algorithm? To best match the analyst's desire for more information, should these thresholds for a particular algorithm be static or dynamic? If the threshold should be dynamic, what should cause a change in these thresholds?

---

division by 0.

DC2 (COMP): Confidence Should Influence SIF

A factor that may influence the relevance threshold, top n threshold, or both is the question of how confident the system is in the information that it has learned from SI and its relevance to the analyst. **If a system is less confident that the information learned is relevant to the analyst, then perhaps it should treat SIF differently**. For example, if a system is less confident, then it may be argued that it should return less data to minimize distractions from data that is more relevant to the current investigation. Conversely, more data can instead be displayed when the level of confidence is low to increase the probability of showing relevant information to the analyst. This tradeoff deserves careful consideration and is dependent on the type of data and tasks that the given system will support. As such, this DC focuses on **how the thresholds should react to different levels of confidence** and mandates that at least a minimum and a maximum threshold be determined.

For example, in Centaurus, we dictate that the level of confidence in the information learned form a particular SI will influence the top n threshold for SIF. We set this threshold to return a maximum 5 observations or attributes when we are confident in the relevance of the information learned by the given SI. When we are less confident, we return a minimum of 2 observations or attributes. No observations or attributes are returned when we have no confidence in the information learned by the given SI. These values were chosen to return at least some data from an interaction that reveals– even through inference– information about the analyst's interest. However, we purposefully set these thresholds low to promote more explicit use of SIF in our user study described in Section 7.4. As mentioned in DC3 and DC6, determining the optimal value of n would require research into areas such as how the displays are specifically used, how clutter is generated, whether such clutter can be avoided, and whether the dataset itself provides context that should further influence the determination of a top n threshold.

This design challenge leads to open research questions such as, how does showing more data vs. less data in less confident scenarios affect analysts' tasks? Is showing more vs. less data a matter of analyst preference, or do certain analytical tasks imply that one choice over another is beneficial? If this choice is a matter of analyst preference, how consistent is this preference across analysts given a particular type of data or analytic task? Could the system detect shifts in analytic tasks or goals to automatically adjust whether SIF should return more or less data in less confident scenarios, or is this preference too ambiguous for the system to detect (thereby necessitating the analyst to explicitly express their preference to the system)? Additional research questions regarding how to determine the limits that the thresholds can have have significant overlaps with DC6 and are left for discussion there.

DC3 (COMP): The Level of Confidence Must Be Determined

Since the level of confidence the system has in the relevance of the information learned from SI to the analyst may vary, **a method for determining how confident the system should be in the information learned from SI must be determined**. These confidence levels will then leverage the decisions made in the previous DC to determine precisely how SIF should react to a given SI. One method is to **use mathematical principles** such as uncertainty, confidence intervals, or other measures of the complexity or shape of the solution space to quantitatively determine the level of confidence of the information learned from a given SI. This enables fine-grained and dynamic notions of confidence, thereby enabling similarly fine-grained and dynamic alterations of either the top n threshold, relevance threshold, or both. However, such mathematical methods may be computationally expensive and therefore prohibitive to maintaining the system's interactivity for the analyst when working with big data.

As a simpler solution, **the level of confidence in the one interaction may naturally be**

higher than in another interaction because of the SI itself. A lower level of confidence of the information learned from SI can be hard coded into the SIF implementation by simply setting the relevance threshold, top n threshold, or both such that different amounts data are returned accordingly. The tradeoff in only relying on the nature of the SI is that the thresholds become fixed based exclusively on the SI performed and is therefore less flexible and dynamic than using mathematical principles to determine the level of confidence. As an example example, Centaurus (like Cosmos as described in Chapter 6) is inherently more confident that the information learned from moving the importance slider up is a better reflection of the analyst's interest than from moving the slider down. This is because moving the slider down only indicates what the analyst is *not* interested in. Thus, the system is programmed to understand that not enough information about what the analyst *is* interested is learned from this interaction to be confident in the relevance of information returned by SIF. As such, moving the relevance slider down alters the top n threshold to a fixed 0, resulting in no new documents added to the display.

Additionally, if a given SI leaves the system to indirectly infer which data the analyst is interested in, then it is less likely that the information learned is directly relevant to the analyst. As such, data returned by SIF has a higher likelihood of being irrelevant or distracting since the information learned may not accurately reflect the analyst's interest, meaning there should be a lower level of confidence in SIs that lead to these scenarios. For example, when a analyst performs PrI on documents in Centaurus, the system attempts to learn the terms that define the similarity/dissimilarity relationships the analyst has expressed. This means that although the analyst has specified similarity/dissimilarity relationships between the documents, SI attempts to learn the relevance of the different terms in the corpus. As such, the information learned by this SI is of a different (and inferred) nature than manipulating the importance slider. In Centaurus, the lower level of confidence reflected by PrI results in

a lower top n threshold (as determined by the previous DC). Other instances of inference in Centaurus include when we learn W_O form W_A or vice versa, which is discussed further in DCs 8 and 9.

This design challenge brings to light research questions such as, what is the overlap (or distinguishing factors) between the level of confidence the system should have and mathematical principles such as uncertainty metrics or confidence intervals? Can metrics for the level of confidence incorporate both the type of SI performed as well as such mathematical principles? Between quantitative metrics, SI classification, and a combined metric, which best matches the analyst's mental model? What are the other tradeoffs between each category of metric (e.g., computational complexity)? How much variation in the optimal metric is there between analysts? Does the optimal metric change depending on data type or analytical task?

DC4 (COMP): Whether The Analyst is Ready for SIF Must Be Determined

Thus far, we have discussed how to measure relevance of information as well as how to leverage the level of confidence to adjust SIF accordingly. However, even if the level of confidence in the information learned by SI is high and we have data above our relevance threshold, this does not necessarily mean that SIF should be performed. For example, in StarSPIRE [12], a analyst can highlight sections of a document, thereby directly denoting which information in the document is important to them. This SI leads the system to determine which other documents with similar information have not yet been displayed, and the system can be reasonably confident that these documents are relevant to the analyst. However, StarSPIRE always displays these new documents, even though the analyst may not have finished reading the given document yet, let alone any others they may have wished to read. Always performing SIF in this manner leads to visual clutter and unnecessary

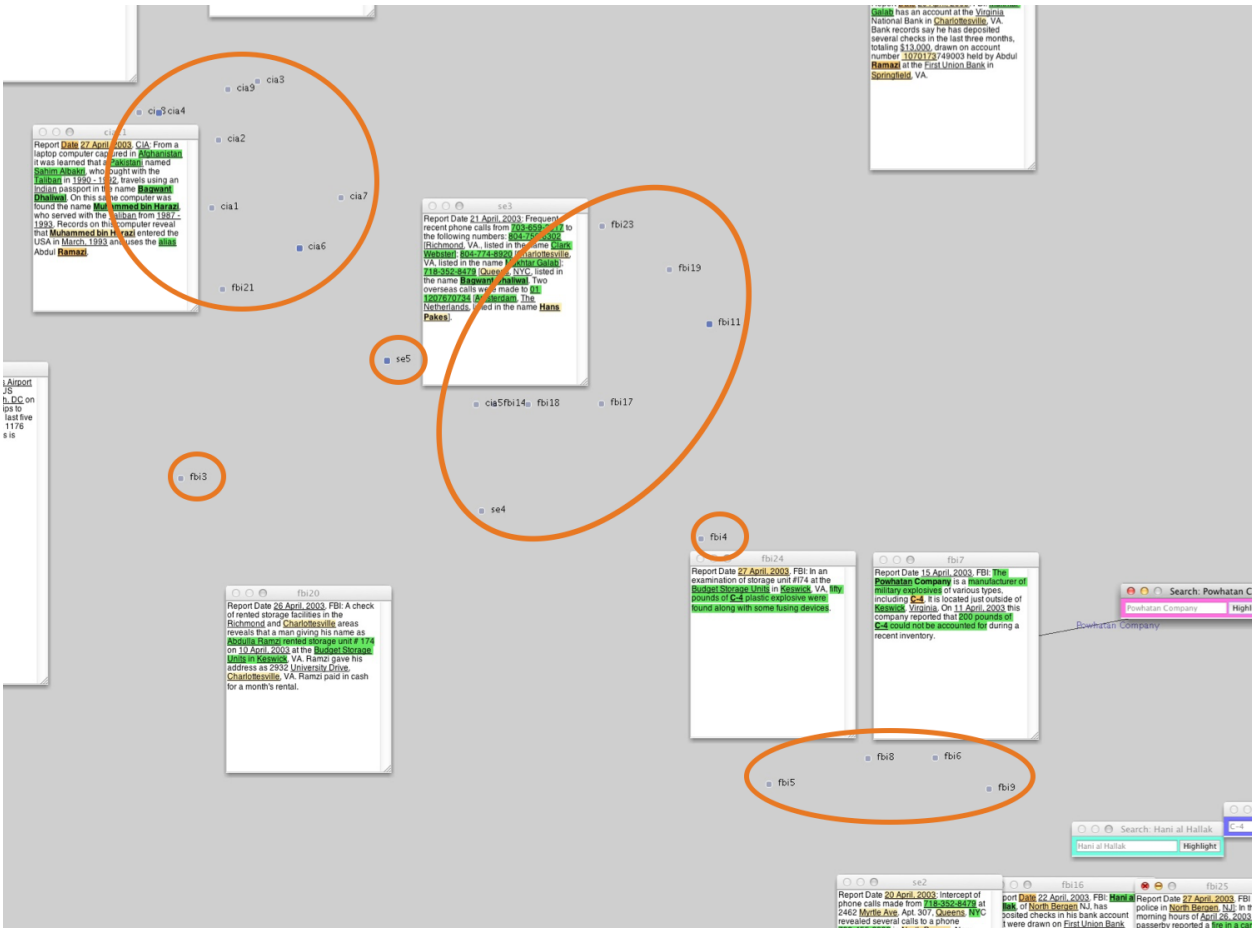

Figure 7.5: A screenshot of StarSPIRE [12] showing documents retrieved by SIF which were never used as part of the analyst's investigation, indicating that SIF was perhaps used too often.

documents being added to the display, as shown in Figure 7.5. Therefore, even though StarSPIRE is confident that the information learned by this SI and has determined a subset of relevant documents that can be added to the display, the analyst is not yet ready for more data. In other words, StarSPIRE assumes that the analyst is always ready for more data, which is not necessarily true.

As such, **the system needs to determine whether SIF should be employed after a given SI**. This can also be thought of as the system trying to determine when the analyst is ready to see more data. However, given how new SIF is as an automated foraging technique,

there is no existing work that seeks to determine a more fine-grained answer than to dictate that a analyst is "never" ready (i.e., SIF is not used) or "always" ready (i.e., SIF is always used, like in StarSPIRE). As such, the open research questions concerning this design challenge include, is there a pattern in analyst rationale or system usage that might inform the system when the analyst is ready to use SIF? If such a pattern does exist, can the system use, detect, or learn this pattern, or must the analyst explicitly dictate when they want to perform SIF? If the system can automatically determine when to use SIF, how detrimental is it to the analyst's Sensemaking Process if it is used to early or too late? Using Centaurus, we begin exploring these questions via a user study in Section 7.4 by allowing the analyst to explicitly choose whether SIF will be performed after an eligible SI. This choice is expressed through checkboxes beneath each projection to denote whether the analyst wants SIF for more observations, more attributes, or both. Thus, Centaurus makes no assumptions about when to perform SIF.

DC5 (COMP): Analysts Should Be Able to Correct SIF

If SIF returned data that was not relevant to the analyst, the analyst should have some method of providing this feedback to the system. A simple example in Centaurus would be for the analyst to express that the unwanted data should be removed from the display (which is also seen in StarSPIRE [12] and Cosmos from Chapter 6). Similar to the moving the importance slider down, **such an interaction would be considered a form of SI that does *not* result in foraging for more data, but rather informs the system of what is *not* relevant to the analyst** and should not be returned by SIF in the near future. However, there is the possibility that the data is simply not relevant to the current investigation, but later becomes relevant as the investigation shifts focus [128]. Therefore, no data should be prohibited from being foraged again unless explicitly indicated

by the analyst.

A more complex approach to this problem may be to employ alternative, competing algorithms and attempt to learn which algorithm best supports a specific analyst's mental model. For example, which of the competing algorithms is ultimately used can be determined by a weighting scheme in which the weight for a given algorithm is increased if the analyst interacts with data that would have been returned from the given algorithm. As a more specific example, say ranking algorithm A would have returned documents W, X, and Y from a given SI whereas algorithm B would have returned documents X, Y, and Z. If both algorithms are equally weighted, then the union of both document sets could be displayed. If algorithm A was more highly weighted, than only documents W, X, and Y would be displayed. If the analyst interacts with document W, then only algorithm A will be upweighted, whereas interactions on document X will result in both algorithms being upweighted to reflect the fact that either algorithm would have returned that specific document. If the analyst does not perform an interaction on a document from a given algorithm, then perhaps that algorithm should be downweighted to reflect that it did not return data that the analyst deemed relevant.

Given the many possibilities implied by this design challenge, research questions are revealed such as, how complex is the design space for feedback specifically for SIF? How much overlap is there in feedback for SIF and feedback for SI? A deep analysis of the possibilities and experimentation with such possibilities will likely reveal a rich research space to explore as well as further develop guidance for how such feedback should be realized.

DC6 (VIS): New Data Should Be Displayed in Context of the Task

Once new data have been returned by SIF, it is important to consider **how the analyst will use or perceive these additional data** in the display. The ultimate approach to this design challenge is highly dependent on the analyst's specific tasks with the data and how best to support those tasks. For example, StarSPIRE [12] assumes that new documents found by SIF should always be added to the display. However, old documents may still be relevant, so all old documents are also preserved; a document is only removed from the display when the analyst explicitly does so. As such, there is no limit to the total amount of documents displayed. Additionally, it is assumed that any opened documents are still relevant. Treating them as pinned nodes, the new documents position themselves around these opened documents, thereby fitting themselves into the analyst's specific spatialization of the data.

In contrast, Centaurus favors new observations and attributes above old observations and attributes under the rationale that these new data have been determined to be immediately relevant to the analyst's most recent interaction. However, we also aim to avoid cluttering either panel with too many observations or attributes. Therefore, we cap the number of nodes projected in either panel to be 30 to support the user study described in Section 7.4. Under this constraint, the least relevant old data are removed as necessary to make room for new data if this threshold is reached.

As a related notion, it is important to consider how new data should be visually encoded in the display, which depends greatly on the visual representations used and the tasks being performed. Some considerations include whether the analyst will want to distinguish newly foraged data from data that previously existed in the display and, if so, for how long. Given the great number of possibilities for visual encodings, we do not enumerate all such

possibilities here. However, Centaurus exemplifies how such visual encodings might be used by having the corresponding nodes for new data appear in the upper left corner of the prospective panel and smoothly transition to their proper location. Old nodes also transition in response to new data being added to the display. Additionally, the new nodes are colored lighter hues to help further distinguish them from the previously existing nodes, as seen in Figure 7.3. After the next interaction, the nodes transition to a standard, darker node color. This allows only the newest data to stand out, thereby helping analysts focus on this data as opposed to data they have already seen. However, given that certain observations or attributes may not have yet been investigated, the visual encodings of bolded node labels and a dark ring around the node help visually distinguish uninvestigated nodes from investigated ones.

From this design challenge, we see research questions such as, what are the possible methods for introducing new data into the display? How does each method affect existing data? How does each method influence the analyst's perception of the space or ability to perform analytic tasks? Is there a limit in the amount of data that can be added to the display at a time before it becomes distracting or overwhelming to the analyst? If so, how much variance is there in this limit between analysts, and what does this limit and variance imply for each threshold used? Is there a limit in the total amount of data that can be displayed before it becomes overwhelming or otherwise too difficult to interact with or understand? If so, how much variance is there in this limit between analysts? How much overlap is there between each of these two limits and the notions of visual clutter and overdrawing (e.g., as described by Splatterplots [80] or the numerous methods in the clutter reduction taxonomy defined by Ellis and Dix [36])?

DC7 (VIS): The Display Must Be Initialized

Once the above design challenges are answered, how to initialize the display may be determined [128]. There are 3 broad categories of initialization decisions: displaying all data, no data, or some data. Displaying all data provides a global overview [108] but can easily lead to visual clutter and/or overwhelm the analyst, especially as the size of the data grows. In contrast, displaying some of the data provides a more scalable alternative but requires a method for determining a relevant subset of the data before the analyst provides any information or feedback. As such, there is a risk of anchoring or biasing the analysis [119]. Finally, displaying none of the data initially is the most scalable solution and avoids anchoring or biasing the analyst. However, the drawback becomes that the analyst must know something about the dataset to perform an initial keyword search for relevant data.

Centaurus, like StarSPIRE [12] and Cosmos (from Chapter 6), initially does not display any documents since the system has not yet learned anything about the analyst's interest and therefore cannot be confident in any inferences made without analyst input. These systems would rather have the analyst specify data of interest than risk biasing the analyst towards irrelevant data. However, the ranking algorithms Centaurus uses can be leveraged to reveal data worthy of being included in an initial projection, similar to how SIRIUS (described in Chapter 4) highlights certain data in an initial projection. We describe how to do this automatically in Centaurus in Section 7.5. Another alternative is to leverage a collaborative model to initialize the display based on previous analyses (for example, Filonik's work on Participatory Data Analytics [46]). In so doing, collaborative systems are capable of displaying initial data to the analyst that the system is confident is relevant to the analyst [75].

7.3.2 SIF in a Symmetrical System

In a symmetrically-designed system, the aforementioned design challenges must be addressed for both observations *and* attributes. However, implementing SIF requires additional design challenges, including:

8. (COMP) How can both observations *and* attributes be foraged from the same instance of SIF?

9. (COMP) How many observations or attributes should be foraged?

DC8 (COMP): Information on Observations Must Be Translated to Attributes and Vice Versa

In order to forage for both observations and attributes, information must be learned about each following a single SI, and this information must connect with the ranking algorithms used (as described in DC1). However, most interactions will only directly influence observations *or* attributes. For example, if a analyst decides that a particular observation is very relevant, then the interaction to denote this will only be on that one observation. Based on this SI, it may be easy to learn information about other observations or about their attributes (but likely not both). Thus, the missing information must be learned as well via a **translation of the newly learned data to the missing data**. For example, if a analyst searches for the term, "cat," then documents about cats should be returned. However, in addition to returning the original term, "cat," which other attributes should be returned? This can be inferred by learning which terms commonly appear in the newly foraged documents about cats (e.g., "furry" and "cute"). In Centaurus, the relevance metrics above (Equation 7.1 and Equation 7.2) can easily perform such translations to turn information

learned about documents into information about terms and vice versa by using the output of one equation as the input to the other.

These examples illuminate research questions such as, how do you translate information on observations to information on attributes and vice versa? Which ranking algorithms for observations and attributes easily or naturally lend themselves to such translation, such as in SIRIUS and Centaurus? For ranking algorithms that do not lend themselves as easily to this translation, how complicated can the translation become? Are there certain ranking algorithms in which such translation is infeasible (and are therefore incompatible with a symmetrically-designed system that incorporates SIF)?

DC9 (COMP): The Number of Observations *and* Attributes to Forage Must Be Determined

The question of precisely **how many observations and attributes should be foraged after a given SI** involves leveraging the thresholds of the ranking algorithm (DC1) based on the confidence of the information learned from SI (DC2, DC3). As noted in DC8, a given SI is likely to be centered on either the observations *or* the attributes, thereby necessitating a translation of information to learn the missing information necessary to perform SIF for both observations and attributes. While the original interaction might be very informative about either the observations or the attributes, the translation may only infer the missing information. Following the example above, when a analyst searches for the term "cat," the system can be fairly confident that the documents about cats are highly relevant to the analyst. However, how confident should the system be that the learned terms "furry" and "cute" are also relevant to the analyst? If the system has a lower level of confidence in the relevance of these terms to the analyst, then the relevance threshold and/or the top n threshold should be adjusted accordingly. For an SI that inherently results in less confident

information (e.g., PrI), how should this lower level of confidence at the beginning of this process affect SIF with the translated information?

As mentioned in DC2, Centaurus returns 5 documents or terms when we are confident about the information learned and 2 otherwise (only returning 0 if we know the information learned by the given SI does not provide enough information about which data is relevant to the analyst). We consider translating information to produce less certain information and therefore only forage 2 documents or terms in this case. For example, searching for "cat" will forage the term "cat" as well as 5 documents about cats (since we are very certain about cats being relevant to the analyst). However, we only forage 2 additional terms since we are less certain about how important other terms are to the analyst. In contrast, a PrI is considered to produce less certain information. Since the information translation also produces uncertain information, we only forage for 2 documents and 2 terms (regardless of which was originally interacted with).

7.4 User Study on When to Use SIF

As mentioned previously, Centaurus was designed to provide a testbed for implementing SIF. The first research questions we focus on concern DC4:

- Is there a pattern in analyst rationality or behavior that predicts when they might be ready to see new data?
- If such a pattern exists, is the pattern something that a system could detect or learn so that it can automatically determine when SIF should be used, or must the analyst explicitly tell the system when they are ready for more data?

In this section, we describe a user study that initiated investigation into these research

questions.

7.4.1 User Study Design

Considering the nature of these research questions, we designed a user study that asked participants to explicitly determine when the system performs SIF using check boxes (as described in DC4). As such, this user study was a think-aloud so that we could learn participants' reasoning behind using SIF for both observations and for attributes as they performed a realistic analysis. Additionally, we logged their interactions so that we could evaluate how they used the system as well. Our hope was to find a pattern in either their rationale or interactions that would reflect when participants wanted to use SIF.

To provide a realistic scenario, we developed a dataset of news articles centered on the claim, "people are marching in the UK because their universal healthcare system is going broke and not working," which is based on a tweet from Donald Trump. Participants were then asked to determine the truthfulness of the claim using the news articles. 29 articles reported information truthfully, as determined by a vetting of this claim by PolitiFact [112]; the remaining 12 articles were fake news related to this claim pulled from known fake news sites [48]. As such, the overlap in content between the fake news and real news articles in addition to the ratio between fake and real news helped ensure participants would encounter both types of articles during their investigation. Regardless of the truthfulness of the article, the information for each article was portrayed in the same manner and included the URL, article title, authors, and main body of the article. In total, there were 41 documents and 874 extracted entities (i.e., terms) that participants could use in their analyses. We asked participants to evaluate the truthfulness of the above claim using this data.

Given the nature of this task, we narrowed our pool of participants to students from the

Communication Department at Virginia Tech, which includes Communication, Public Relations, and Journalism majors who are trained through their coursework to analyze claims in this manner. Since these students regularly perform such a task, our participant pool helps further simulate a more realistic scenario. Due to how late in the semester the study was run, only 9 total participants were able to partake in the study, with 2 of them being used as pilots to help us adjust how to prompt participants for SIF-related information during the study. As such, we propose this user study and its results as an initial exploratory analyses into these research questions on when to use SIF as opposed to a definitive study that concretely concludes when to do so.

Since no participants had prior exposure to Centaurus, we started the study with a hands-on training session. This training session lasted 30 minutes and used *The Sign of the Crescent* [57] dataset. First, participants were guided in how to interact with the system, following an interaction script similar to what is described in the "*Example Analysis with Centaurus*" subsection. This guidance focused on how to properly interpret the display, what each interaction meant, and how to choose which interaction to perform based on analytic intent. Interactions involving decisions on whether to use SIF to automatically find new documents or new entities were particularly highlighted, and differences in performing SIs with and without SIF were shown. In-depth descriptions of the data itself were avoided to encourage participants to interact with the data freely after this initial training. At this time, they were also given the practice task of finding at least 1 of the 3 main terrorist plots within the dataset. As participants practiced with Centaurus, they were asked to think aloud and were permitted to ask any questions about the system. If participants become stuck or otherwise seemed unsure of what to do next, specific interactions were suggested based on their analytic reasoning. Any guidance on which direction to take their analysis, how to interpret the data, or other such components of their analysis was strictly avoided. As participants

performed SIs, they were asked to consider how they wanted the checkboxes set to enable SIF for documents and/or entities as well as their rationality for doing so.

Following this 30 minute practice session, the main task to determine the truthfulness of the aforementioned claim was introduced. The claim was read aloud to the participants, and the fact that the dataset comprised of truthful news articles as well as fake news was explained to them. This information was typed on a piece of paper which was handed to the participant to help remind them of the wording of the claim and the nature of the dataset as they performed their analysis. Participants were given up to 50 minutes to complete this task, although the study ended earlier if participants felt sure about their answer before their allotted time had passed. Similar to the practice session, participants were asked to think aloud, along with providing rationality for setting the SIF checkboxes for documents and entities as they performed SIs.

Both the practice session and the main task were video and screen recorded to capture their rationales in using SIF. Additionally, their open, importance slider, search, and PrI interactions in Centaurus were automatically logged. This data was used to then analyze the patterns in rationales used for determining whether to use SIF for documents and/or entities, as well as interaction patterns between usages of SIF for documents and/or entities. We describe these analyses in the next subsection.

7.4.2 Data Analysis and Results

Given the data collected and analysis goals, we divide our analysis description between specificity of rationale and interaction pattern between SIF.

Specificity of Rationale

For each participant, the recorded audio was first transcribed to capture their rationale in using SIF or not for both documents and entities (separately). These rationales were then briefly summarized and categorized based on how specific the rationale was for using SIF. To enable easier comparison between these rationales, each category was encoded with a numerical value based on the specificity of the rationale. These rationales were then cross-referenced with the participant's interaction logs to determine if any potential rationale for using SIF was missed, which occurred 6 times across all participants. As shown in Table 7.1, an encoding of 0 indicated that the rationale was not captured, whereas an encoding of 8 represented a very complex or specific rationale. Note that the differences in these encodings simply reflect a relative increase or decrease in the level of specificity as opposed to attempting to be a firm measure in the level of specificity.

Given that participants often mixed rationales for choosing whether to use SIF for documents or entities (i.e., providing rationales on both simultaneously), we chose to average their specificity encodings. As such, this data analysis focuses on trends over time in rationales for choosing to use SIF for either documents or entities (simultaneously) across all participants. These averaged encodings were then graphed as a set of timelines for each participant. These graphs were used in combination with a linear regression to determine if any pattern seemed to exist across participants in the specificity of rationales over time. This graph is shown in Figure 7.6. No clear patterns is seen in the timelines, and the linear regression shows only a slight upward trend that is not statistically significant ($R^2 = 0.0083$). As such, this study did not reveal a clear pattern in the specificity of participants' rationales for using SIF. One possible explanation is that these rationales shift too quickly or are often too vague to be reliable predictors of SIF usage.

Code	Rationale Summary
0	No data; Insufficient data
1	Has enough data
2	Wanted more information; Had enough information; Wanted more entities; Has enough entities; Was stuck
3	Wanted to explore existing information; Wanted to focus on existing entities
4	Wanted more similar information; Doesn't want similar documents; Wanted related entities; Wanted opposing information Wanted less information like this document; Doesn't think the document is important
5	Information in document was repetitive; Entities in document were repetitive; Thought nothing new would come up; Did not feel he'd get more useful entities
6	Wanted information related to entity(ies); Wanted entities related to document; Wanted entities related to search
7	Wanted specific entities
8	Did not want to be flooded with too many matching documents

Table 7.1: Each of the rationale specificity encodings that were used along with the types of rationales that were matched to each.

Interaction Pattern Between SIF

To uncover interaction patterns between usages of SIF, we first listed the interaction sequences leading up to (and including) an interaction eligible for SIF. For example, if a participant increased the importance slider, this interaction is eligible for SIF (regardless of whether the participant actually had either box checked to use SIF for this interaction). Leading to this interaction, perhaps the participant read 3 documents. Therefore, the interaction sequence for this set of interactions would be `open;open;open;relevanceEND`, where `END` denotes the SIF-eligible interaction (i.e., the last interaction in a sequence). A second interaction sequence was also generated that further specified whether the each interaction in the sequence was on a document or a entity (i.e., `open-doc;open-doc;open-doc;relevance-docEND`). We referred to these interaction sequences as "generic interaction sequences" and "specific interaction sequences" to differentiate between the two.

These interaction sequences were then cross-referenced with participants' audio recordings to determine if they actually wanted to use SIF for documents or for entities after the given SIF-eligible interaction. Following the previous example, if the participant stated that they

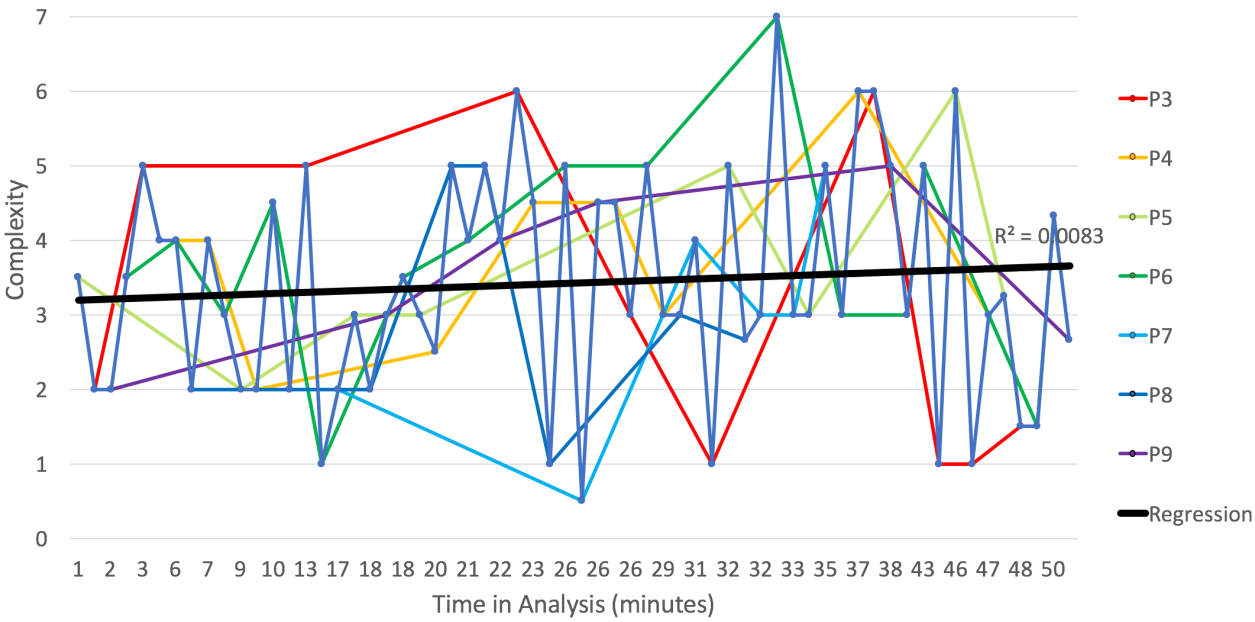

Figure 7.6: Timelines of each participants' average specificity in reasoning for whether to use SIF after an SIF-eligible interaction. A linear regression across all participants shows a slight increase in specificity over time, but this trend is very weak ($R^2 = 0.0083$).

wanted to forage for more documents but not more entities, then the interaction sequences `open;open;open;relevanceEND` and `open-doc;open-doc;open-doc;relevance-docEND` would be matched with a `yes;no` SIF decision. If a rationale was missed (which occurred 6 times), this was treated as a `no;no` SIF decision to focus only on instances in which we were certain that the participant wanted to use SIF. In total, 51 pairs of interaction sequences and their related SIF decisions were captured across all participants. These sequences are visually represented in Figures 7.7 and 7.8 for general and specific interaction sequences, respectively. In these figures, the size of a point on a parallel coordinate reflects how many interaction sequences contained that interaction at that given point in the sequence, and the width of the lines show how many interaction sequences contained that given transition between interactions.

With this data in hand, we determined that participants wanted to use SIF for more documents 69% of the time and SIF for more entities 55% of the time. This would be equivalent

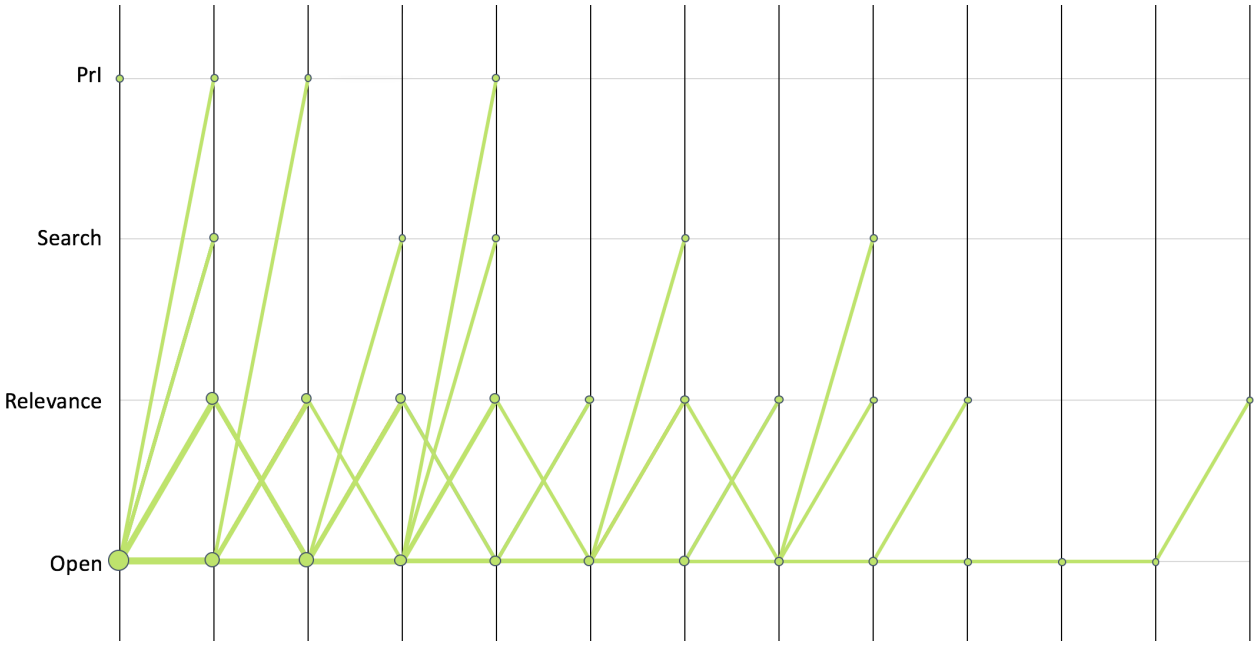

Figure 7.7: A parallel coordinates plot showing the overlap of each of the 51 generic interaction sequences.

to only using the occurrence of an SIF-eligible interaction (i.e., an interaction pattern of *END*) to predict when participants wanted to use SIF, which would be the correct prediction 69% of the time for documents and 55% of the time for entities. We call this metric for interaction patterns its **predictability**. We aimed to improve our predictability by comparing interaction sequences across all participants to extract more complicated interaction patterns that would yield a higher predictability for either documents or entities.

For this pattern extraction, we allowed ambiguous pattern recognition, such as "the participant wanted to use SIF for documents after (at some point since the last SIF-eligible interaction) they opened something followed by at least 1 other interaction." This can also be thought of as using regular expressions, where the pattern matching this description would be `open;.*`. Thus, these patterns dictate a minimum set of interactions that must be contained within a given sequence; other interactions may exist within the sequence as well.

In addition to its predictability, these interaction patterns we evaluated based on how often

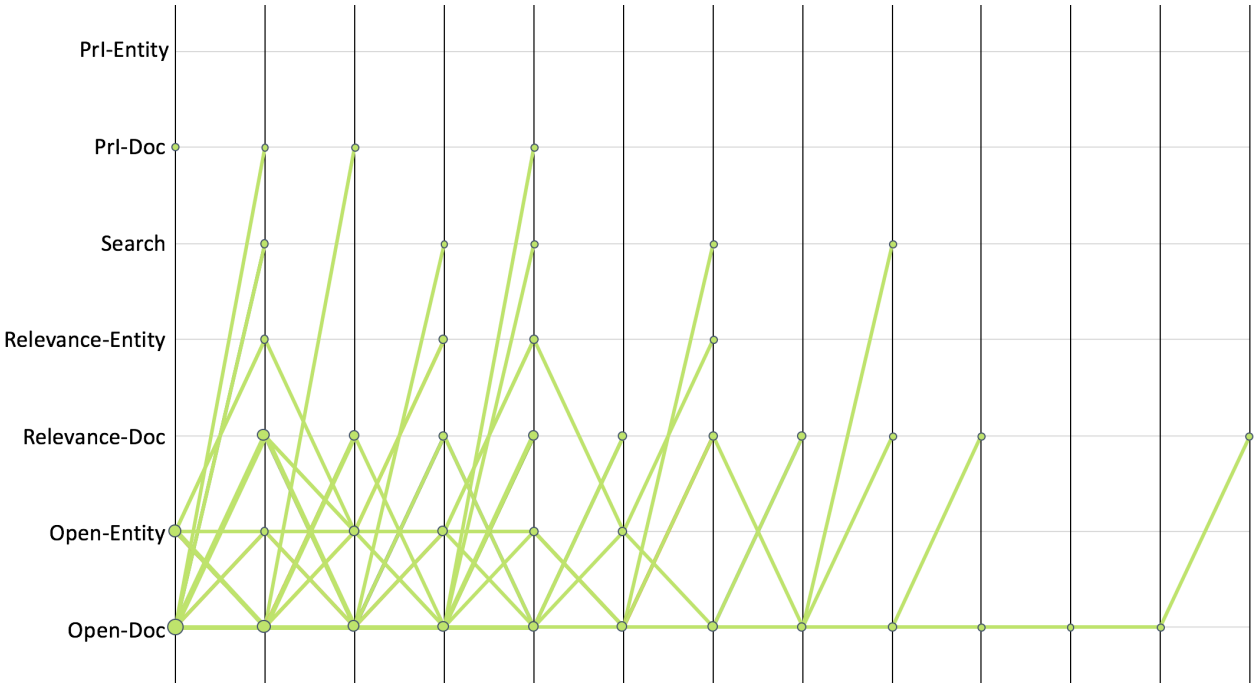

Figure 7.8: A parallel coordinates plot showing the overlap of each of the 51 specific interaction sequences.

the given pattern appeared across all 51 interaction lists, which we define as its **prevalence**. Since we are aiming to improve predictability over 69% for documents and 55% for entities, we used these thresholds to denote indicate potentially interesting patterns. We combined these thresholds with a prevalence threshold of 10 to ensure a better generlizability for each pattern. All such interaction patterns for documents are listed in Tables 7.2 and 7.3, whereas patterns for entities are listed in Tables 7.4 and 7.5. In these patterns, we continue to use `END` to refer to an SIF-eligible interaction (i.e., the last interaction in a sequence). It is important to note that since missed rationales were treated as `no;no` decisions, the predictability of these interaction patterns may actually be higher than reported here. To contextualize where these patterns appear within the interaction sequences and how they overlap, Figures 7.9 and 7.10 represent the where the generic and specific patterns (respectively) occur in participants' interaction sequences. In these figures, the thickness of the lines between interactions represent how many patterns include that particular pair of interactions. Identification of

Predictability	Prevalence	Pattern
0.7	50	.*;.*
0.7	50	.*;END
0.7	50	open;.*
0.7	50	open;END
0.76	41	open;relevanceEND
0.72	32	open;open;END
0.79	28	open;open;relevanceEND
0.83	23	.*;open;open;END
0.81	21	open;open;open;END
0.9	20	.*;open;open;relevanceEND
0.89	19	open;open;open;relevanceEND
0.73	15	open;open;open;open;END
0.85	13	open;open;open;open;relevanceEND

Table 7.2: The extracted generic interaction patterns for wanting to use SIF for more documents. The predictability for participants wanting more documents and prevalence for each pattern is provided.

patterns along these timelines were completed using the techniques described in [81, 82].

These extracted patterns show promise in improving predictions for SIF usage for both documents and entities. For example, it is interesting to see that the interaction pattern `open-doc;.*open-doc; open-doc;END` has a 100% predictability rate for SIF for documents and a prevalence of 10. This means that this one interaction pattern accounts for 29% of the times that participants wanted to use SIF for documents. Further analyses of these patterns would reveal subsets of patterns which combined have a high predictability and prevalence.

However, the manner in which these patterns are chosen and combined will have different implications on their combined generlizability to additional analysts. For example, the patterns `relevance;open` and `open;relevance;open` for predicting SIF for new entities have the same predictability and prevalence. Given how these patterns overlap, it is safe to assume that `relevance;open` is a less specific form of `open;relevance;open` that matches similar interactions within a given interaction sequence. As such, choosing a subset of such patterns

Predictability	Prevalence	Pattern
0.71	35	open-doc;relevance-docEND
0.75	20	open-doc;open-doc;END
0.83	18	open-doc;open-doc;relevance-docEND
0.87	15	.*;open-doc;open-doc
0.85	13	open-doc;open-doc;open-doc;END
1	10	open-doc;.*;open-doc;open-doc;END
0.87	15	.*;.*;open-doc;.*;relevance-docEND
0.92	12	open-doc;open-doc;open-doc;relevance-docEND
0.82	11	relevance-doc;.*;END
0.82	11	open-doc;relevance-doc
0.82	11	open-doc;relevance-doc;.*;END

Table 7.3: The extracted specific interaction patterns for wanting to use SIF for more documents. The predictability for participants wanting more documents and prevalence for each pattern is provided.

Predictability	Prevalence	Pattern
0.59	41	open;relevanceEND
0.68	19	open;open;open;relevanceEND
0.64	28	open;open;relevanceEND
0.77	13	open;open;open;open;relevanceEND
0.75	16	open;.*;open;open;relevanceEND
0.8	10	.*;open;open;open;open;relevance
0.75	12	open;relevance;open
0.75	12	relevance;open
0.75	12	relevance;.*
0.56	50	open;.*
0.56	50	open;END
0.62	21	open;open;open;END
0.67	15	open;open;open;open;END
0.59	32	open;open;END
0.71	21	open;.*;open;.*;END
0.71	14	open;.*;open;.*;open;END
0.73	11	.*;open;open;open;open

Table 7.4: The extracted generic interaction patterns for wanting to use SIF for more entities. The predictability for participants wanting more entities and prevalence for each pattern is provided.

Predictability	Prevalence	Pattern
0.6	35	open-doc;relevance-docEND
0.61	18	open-doc;open-doc;relevance-docEND
0.67	12	open-doc;open-doc;open-doc;relevance-docEND
0.72	18	open-doc;.*;open-doc;relevance-docEND
0.74	19	open-doc;.*;open-doc;relevance-doc
0.75	12	.*;open-entity
0.57	49	open-doc
0.57	49	open-doc;.*
0.57	44	open-doc;END
0.62	13	open-doc;open-doc;open-doc;END
0.71	24	open-doc;.*;.*;END
0.7	23	open-doc;.*;open-doc
0.61	18	open-entity;.*;END

Table 7.5: The extracted specific interaction patterns for wanting to use SIF for more entities. The predictability for participants wanting more entities and prevalence for each pattern is provided.

naturally means choosing between these two patterns. Therefore, a natural question here is, how does the predictability and prevalence of these patterns change if more participants are run through the same experiment? In other words, how well do each of these patterns generalize to more participants, and which pattern produces better results? Questions such as these prompt more research in this direction (as described Section 7.5).

7.5 Discussion

In this section, we describe the broader implications, limitations, and future work of both the user study on when to use SIF as well as the other design challenges more broadly.

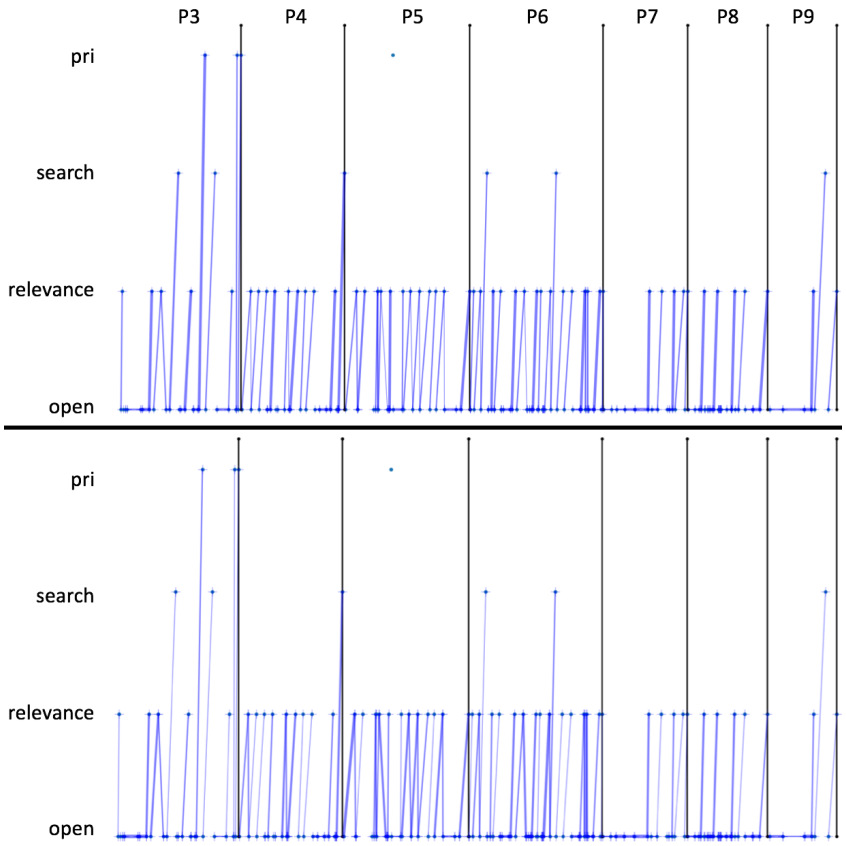

Figure 7.9: Timelines for each participants' interactions in which the lines between interactions represent a match for one or more of the generic interaction patterns. The **top** graph represents patterns listed in Table 7.2 to predict when participants wanted to use SIF for more documents, whereas the **bottom** graph represents patterns listed in Table 7.4 to predict when participants wanted to use SIF for more entities.

7.5.1 User Study on When to Use SIF

The findings of the user study reveal that there are detectable patterns in participants' behavior (but perhaps not their rationality) that predict when they are ready to see new data. The interaction patterns that describe these behaviors are something that the system can use to automatically detect or determine when to use SIF. In particular, combining such patterns together may yield a high prevalence and accuracy, meaning the system will automatically use SIF in most instances in which an analyst would want to use such automated foraging.

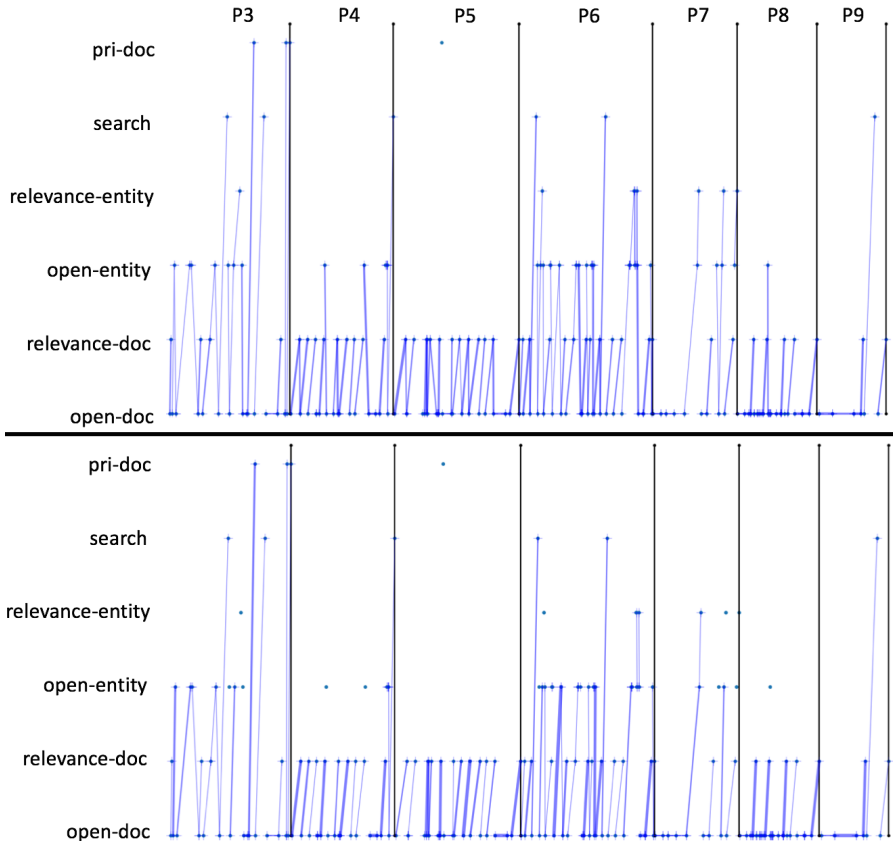

Figure 7.10: Timelines for each participants' interactions in which the lines between interactions represent a match for one or more of the specific interaction patterns. The **top** graph represents patterns listed in Table 7.3 to predict when participants wanted to use SIF for more documents, whereas the **bottom** graph represents patterns listed in Table 7.5 to predict when participants wanted to use SIF for more entities.

However, it is unlikely that these predictions will always be 100% accurate for all analysts. As such, we argue in favor of leaving some method for analysts to explicitly express when they want to see more data.

Despite the promising results for using interaction patters to predict when analysts might wish to use SIF, it is important to consider the limitations of this study. The most obvious limitation is the small number of participants. Thus, it is difficult to say whether the interaction patterns identified would perform similarly well across a broader population, even if same task was performed on the same dataset. Along these lines, it may very well

be that interaction sequences differ significantly depending on the task being performed and the nature of the data. For example, this user study focused on text data, and the main task participants performed was centered on the documents as opposed to the terms within them. As such, participants performed relatively few interactions on the terms. It seems reasonable to assume that their interaction sequences (and thus extracted interaction patterns) would be different when performing tasks that necessitated analyses of both the documents and the terms.

Another analyses that may reveal additional interesting patterns would be to attempt to predict when participants *didn't* want to use SIF. Using these patterns would help the system determine when SIF should be avoided. Combined with patterns for when to use SIF, the predictability and prevalence of the combined patterns may improve significantly. However, such patterns may have similar issues in their generalizability and therefore deserve similar analysis along these lines.

It is also worth noting that our evaluation of participants' specificity of rationales in using SIF we evaluated based on patterns over time, where specificity was treated as a 1-dimensional concept. It is worth further encoding the specificity of rationales to create a high-dimensional dataset to better capture the complexity implied by these rationales. This high-dimensional data can then be used to then perform much more complex analyses into these rationales. As such, we cannot yet rule out the possibility of there being a useful pattern in participants' specificity of rationale that can predict when they wanted to use SIF. If such a pattern exists, they should be evaluated in conjunction with the interaction patterns to determine how generlizable they are as well.

Considering these potential issues in the generalizability of our results, there are 3 main directions to continue research on using interaction patterns to determine/predict when SIF should be used:

- Continue the same experiment with additional participants (whether it is additional undergraduate Communication majors, journalism professionals, or novices to such tasks) to determine if patterns generalize to a wider population.

- Perform a similar user study in which the main task requires participants to analyze both the documents and the terms of the given dataset. Patterns from this additional study can then be compared to those found in our original study here to determine how generalizable patterns are across different tasks.

- Perform a similar study with an entirely different type of dataset (e.g., the animal dataset by Lampert et al. [71]). Patterns from this additional study can then be compared to those found in our original study here to determine how generalizable patterns are across different types of data.

Additionally, potential for different types of interaction patterns may also be worth investigating, such as eye movements and scan patterns [114] or time between interactions. Similarly, incorporating a preview of data that could be added by SIF may help analysts decide which pieces of data they want to see added to the display. Such an interaction can also provide more feedback to the system regarding which data is most relevant to the analyst as well as when to perform SIF to individualize support for their Sensemaking Process. However, even with these types of additions, it is unlikely for a perfect set of interaction patterns to be found that will always accurately predict when every analyst would ideally like to see more data. As such, we can also begin exploring the ramifications for incorrect interaction patterns. More specifically, how incorrect can an interaction pattern be before hindering the analyst's Sensemaking Process?

It is also worth noting that this user study can only be considered guidance for how to continue approaching the research questions regarding when SIF foraging should be used. This

is because there are many other parameters in SIF that also deserve research to determine how best to tune them, such as the relevance and top n thresholds. These open research questions are numerous, as described in Section 7.3. Since tuning these other parameters will alter which data are foraged, such tuning may influence the optimal time to use SIF. We explore the future work regarding these other research directions in the next subsection.

7.5.2 SIF Design Challenges

In the use case presented in the Section 7.2, we highlighted the prowess of SIF in a symmetrically-designed system. Although the analyst did not find all information related to the uncovered terrorist plot, half of the relevant documents specific to this plot were uncovered in only 4 interactions. Additionally, these documents provided enough information for the analyst to confidently report a terrorist plot, including who was involved, what form of terrorism was to occur, where the plot would take place, and when the plot would take place.

However, it is also important to recognize that SIF in Centaurus returned additional data that was not immediately relevant to the analyst's task. For example, PrI did not return any documents or terms relevant to the analyst's investigation on Mukhtar Galab. Thus, the analyst could have reached the same conclusion with only 3 interactions if the analyst did not perform PrI. As such, while SIF is not 100% successful in highlighting relevant data, it often helps guide the analyst in the right direction to continue their analysis, as also showcased in the original research concerning SIF [125].

The fact that Centaurus did not always pull relevant documents alludes to the many improvements that the system could benefit from. For example, the system uses the same thresholds for both documents and terms. However, there are far more terms than documents in a text

dataset. Combined with the fact that terms generally do not appear in many documents, it may be reasonable to return more terms than documents from SIF. To accomplish this, the top n threshold and/or the relevance threshold for terms could be adjusted. Precisely how to adjust these thresholds relies on investigation into the research questions on DCs 1–3.

Along similar lines, an additional possibility for improvement in Centaurus is determining a reasonable initial set of data to display (i.e., researching DC7). As one possibility, collaborative filtering can be used to learn the kind of analyses typically performed with a given dataset to recommend better data from the start. For example, the next analyst to work with *The Sign of the Crescent* may begin their analysis with data used by the previous analyst, thereby enabling a faster evaluation of the dataset. Alternatively, the method in which SIRIUS is initialized (as described in Chapter 4) could be used to determine an initial ranking of observations and attributes. Based on the relevance and top n thresholds, an initial subset of potentially relevant data may be determined.

Such investigations on the numerous research questions surrounding the design challenges involve extensive research. Additionally, the guidance determined by such research may be highly dependent on the order in which the design challenges are investigated. For example, if research into the relevance and top n thresholds is conducted first, then it seems reasonable to make adjustments to these thresholds prior to investigation into DC7. If instead DC7 is investigated before these adjustments are made, then it is possible for the system to determine a different subset of data to initialize the display with. Such a difference would easily lead to a different path in analysts' Sensemaking Process due to anchoring or biasing their investigation [119]. Thus, research into each of these design challenges should also include research into how they interact with or influence each other to help researchers understand how to alter previous findings based on new ones.

The potential interactions between each of the design challenges further highlights an im-

portant limitation: these design challenges can currently only be considered guidance into how to implement and research SIF as opposed to offering any firm resolution. This open-endedness is due in large part to how new SIF and SIF research is; quite simply, not enough has been done yet to offer a more firm resolution. As a result, the design challenges posed in this chapter may not be complete; future research may unveil additional design challenges or refine those we have posed here.

7.6 Conclusion

In this chapter, we presented a series of design challenges necessary for implementing SIF and described a wide set of research questions to help guide future research into SIF. We then showcased how these challenges may be addressed through Centaurus, a prototype system that is symmetrically-designed. Centaurus provides a testbed for SIF research, such as the question of when SIF should be used. This research question was investigated with a user study, which showed promise in using interaction patterns to determine/predict SIF usage. We concluded with discussions on the limitations and future work implied by both the user study specifically as well as the design challenges more broadly. We hope that this work provides a starting point for more thorough research into SIF.

Chapter 8

Conclusions

In this chapter, we summarize the broad strokes of the research described in this dissertation, along with proposed areas for future work.

8.1 Summary

In this research, we explored 6 different research questions to begin an exploration in how to support the analyst's Sensemaking Process for tasks such as determining the truthfulness of a claim. These research questions and how we addressed each are described as follows.

1. How can we model semantic interaction to capture the complexity of how algorithms process and learn from the interactions?

   To address this research question, we described a **model** in Chapter 3 to capture the computational steps necessary for the system to process each semantic interaction. As such, this model enables direct comparison between different systems that incorporate semantic interaction, thereby highlighting future research opportunities to continue exploring how to incorporate semantic interaction in visual analytics systems.

2. How can we model symmetry in analytical tasks for both observations and attributes in the context of semantic interaction?

In investigating this research question we **modeled** a mathematical structure for symmetric visualization and interaction techniques to support symmetrical analyses of both observations and attributes of high-dimensional data simultaneously. This model, called SIRIUS (as described in Chapter 4), directly supports the cognitive symmetry between observations and attributes, thereby enabling richer explorations of high-dimensional data. This model was then used to develop a system to act as a **testbed** for research into how best to employ such symmetry, as described in the next research question.

3. How does such symmetry affect analysts' time on task, accuracy, and their cognitive cardinality and dimensionality when performing sensemaking tasks? Are there certain tasks that symmetry best supports or hinders?

   In Chapter 5, we used our SIRIUS-based testbed system from Chapter 4 to address this research question by **devising and conducting an experiment** that explored which analytic tasks were best supported by symmetry. For this exploration, we divided the symmetry in our system into two components: visual symmetry and interaction symmetry. We then explored the impact of both of these components individually as well as combined on a series of analytic tasks. Ultimately, we found that visual symmetry, compared to more commonly used asymmetrical systems, seemed to support *Cluster* and *Complex* tasks both in terms of time on task and accuracy. However, time on task and accuracy were hindered by symmetry in *Find Extremum* and *Correlate* tasks. As such, our advice is to use visual symmetry when supporting *Cluster* and *Complex* tasks; interaction symmetry is not necessary to see these improvements over an asymmetrical system. In contrast, any symmetry should be avoided when supporting *Find Extremum* and *Correlate* tasks. If some combination of these categories of tasks is to be supported, then careful consideration is needed to determine whether to incorporate

visual symmetry since there will be an obvious tradeoff in analytic performance.

4. How can we model sensemaking in the context of semantic interaction for text analytics, including the interactions between foraging and synthesis processes?

   To explore this research question, we developed a new visual analytics system called Cosmos (as described in Chapter 6) to support big text analytics. With this system, we expanded the model from Chapter 3 to provide a **model** of how each computational component of Cosmos's semantic interaction pipeline explicitly supports the Sensemaking Process. Additionally, this pipeline instance includes semantic interaction foraging for documents to support the Foraging Loop in the Sensemaking Process so that the analyst can remain focused on their Synthesis Loop. In so doing, this enhanced model reveals opportunities for comparing and improving semantic interaction support for specific components of the Sensemaking Process. This model helped form the foundation of our testbed system developed as part of the next research question.

5. When integrating semantic interaction foraging with the three models from our first, second, and fourth research questions to support text analytics (e.g., journalism professionals determining the truthfulness of a claim), what are the design challenges for implementing semantic interaction foraging?

   In Chapter 7, we combined the models from Chapters 3, 4, and 6 to develop a new **testbed** system called Centaurus to incorporate semantic interaction foraging. This testbed provided a means to define and explore the **design challenges** for implementing semantic interaction foraging in a given system. Since semantic interaction foraging is a relatively new technique, the **research agenda** formed around these design challenges is extensive, which we begin exploring in our final research question.

6. In a symmetrical system that includes semantic interaction foraging, how and when do

analysts decide to use such automated foraging techniques?

We used the creation of Centaurus in Chapter 7 to help address our final research question by **devising and conducting a second experiment** regarding how to match the use of semantic interaction foraging with analysts' Sensemaking Process. Our initial results indicate that interaction patterns may be a promising method for a system to automatically determine the ideal time to use semantic interaction foraging, thereby improving on existing semantic interaction foraging techniques which assume such automated foraging should always be used when able. However, additional research is needed to determine the generalizability of the interaction patterns we uncovered. Moreover, research into other aspects of our research agenda on semantic interaction foraging may influence when semantic interaction foraging should ideally be used. As such, our testbed represents a very valuable tool for performing such research to continue improving semantic interaction foraging techniques, especially in regards to how different aspects of this research agenda influence each other.

In examining the intersection between these research questions, we find that our research can largely be divided between developing models of semantic interaction and the Sensemaking Process [90], building testbeds for research in symmetrically-designed systems and semantic interaction foraging, and devising and conducting experiments and research agendas to test hypotheses regarding best practices for when to use symmetry or how to improve semantic interaction foraging techniques.

8.2 Future Work

When looking forward in how to continue along these lines of research, there is the more obvious future work outlined in each chapter that are specific to each research question. For

example, we can next take our testbed for semantic interaction foraging research to explore notions of visual clutter. How much data can be added to the display at one time before analysts begin to feel overwhelmed or their analysis process is otherwise hindered? How much data can be in the display at one time? To explore these questions, we can design an additional experiment, which we anticipate would provide stable limits to how much data can be incorporated into the display, thus providing more specific guidance regarding how to avoid visual clutter in foraging systems. For semantic interaction foraging specifically, this research which would further inform how to set the relevance and/or top n thresholds. Alternatively, we can devise an experiment to uncover understandability and usability issues concerning projection interactions. For example, projection interactions were used infrequently in our experiment on when to use semantic interaction foraging. Was this infrequent use due to the presence of such issues? A more direct exploration into this question may help highlight barriers analysts face in using such interactions, which ultimately lead them to avoid using a potentially useful interaction. These uncovered barriers may reveal additional design considerations for implementing any form of semantic interaction, thereby helping guide further research and implementation of semantic interactions.

However, these future research directions highlight more broadly the need to investigate research questions at the intersection of current research areas such as visual analytics, human-computer interaction, machine learning, explainable artificial intelligence, usability engineering, information retrieval, and cognitive psychology. Indeed, it is the intersection between research areas such as these that produces rich explorations into the complex notions of the Sensemaking Process and how to best support it. This combination of research areas is best reflected by the extensive research agenda for semantic interaction foraging outlined in Chapter 4. For example, the aforementioned experiment on visual clutter would predominantly reflect usability engineering concepts with elements of cognitive psychology.

However, the broader implications for such an experiment would span across visual analytics and information retrieval by helping researchers in these areas understand how much data analysts can be shown throughout their analytic task. As another example, how might information retrieval research refine the semantic interaction foraging design challenges? Encouraging this type of overlap in research areas has already begun with workshops such as "Machine Learning from User Interactions for Visualization and Analytics," a workshop we hosted at IEEE VIS in 2018 and 2019 [128], which will be continued in 2020. This workshop had over 100 attendees in previous years, and we anticipate high attendance again this year. This high level of attendance demonstrates both the interest and the usefulness of interdisciplinary workshops. As such, we would recommend continuing to explore these types of intersections through both individual research efforts, like the experiment on visual clutter, as well as through collaborative efforts reflected by workshops, conferences, and journals that help bring researchers from these different areas together. We hope such collaboration will reveal additional components of the Sensemaking Process that can be automated, or how to incorporate machine learning and automation with the analyst's cognitive abilities.

Bibliography

[1] Adelaide. LexisNexis. http://www.lexisnexis.com/hottopics/lnacademic/, 2017. Accessed: 2017-03-15.

[2] Alan Agresti and Brent A. Coull. Approximate Is Better than "Exact" for Interval Estimation of Binomial Proportions. *The American Statistician*, 52(2):119–126, 1998. ISSN 00031305. URL http://www.jstor.org/stable/2685469.

[3] Z. Ahmed and C. Weaver. An adaptive parameter space-filling algorithm for highly interactive cluster exploration. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 13–22, Oct 2012. doi: 10.1109/VAST.2012.6400493.

[4] Jamal Alsakran, Yang Chen, Ye Zhao, Jing Yang, and Dongning Luo. STREAMIT: Dynamic visualization and interactive exploration of text streams. In *2011 IEEE Pacific Visualization Symposium (PacificVis)*, pages 131–138. IEEE, 2011.

[5] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization*, pages 111–117, Oct 2005. doi: 10.1109/INFVIS.2005.1532136.

[6] Saleema Amershi, James Fogarty, and Daniel Weld. Regroup: Interactive Machine Learning for On-demand Group Creation in Social Networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 21–30, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207680. URL http://doi.acm.org/10.1145/2207676.2207680.

[7] C. Andrews and C. North. Analyst's Workspace: An embodied sensemaking en-

vironment for large, high-resolution displays. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 123–131, Oct 2012. doi: 10.1109/VAST.2012.6400559.

[8] Christopher Andrews, Alex Endert, and Chris North. Space to Think: Large High-resolution Displays for Sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 55–64, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753336. URL http://doi.acm.org/10.1145/1753326.1753336.

[9] Michelle Q. Wang Baldonado and Terry Winograd. SenseMaker: An Information-exploration Interface Supporting the Contextual Evolution of a User's Interests. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 11–18, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258563.

[10] Eric A. Bier, Edward W. Ishak, and Ed Chi. Entity Workspace: An Evidence File That Aids Memory, Inference, and Reading. In Sharad Mehrotra, Daniel D. Zeng, Hsinchun Chen, Bhavani Thuraisingham, and Fei-Yue Wang, editors, *Intelligence and Security Informatics*, pages 466–472, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-34479-7.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003. ISSN 1532-4435.

[12] L. Bradel, C. North, L. House, and S. Leman. Multi-model semantic interaction for text analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 163–172, Oct 2014. doi: 10.1109/VAST.2014.7042492.

[13] L. Bradel, N. Wycoff, L. House, and C. North. Big Text Visual Analytics in Sensemaking. In *2015 IEEE International Symposium on Big Data Visual Analytics (BDVA)*, pages 1–8, Sept 2015. doi: 10.1109/BDVA.2015.7314287.

[14] J. Roger Bray and J. T. Curtis. An Ordination of the Upland Forest Communities of Southern Wisconsin. *Ecological Monographs*, 27(4):325–349, 1957. ISSN 1557-7015. doi: 10.2307/1942268. URL http://dx.doi.org/10.2307/1942268.

[15] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92, Oct 2012. doi: 10.1109/VAST.2012.6400486.

[16] Eli T Brown. *Learning from Users' Interactions with Visual Analytics Systems*. PhD thesis, Tufts University, 2015.

[17] Peter Brusilovski, Alfred Kobsa, and Wolfgang Nejdl. *The adaptive web: methods and strategies of web personalization*. Springer Science & Business Media, 2007.

[18] Nan Cao, Yu-Ru Lin, David Gotz, and Fan Du. Z-Glyph: Visualizing outliers in multivariate data. *Information Visualization*, 17(1):22–40, 2018. doi: 10.1177/1473871616686635. URL https://doi.org/10.1177/1473871616686635.

[19] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[20] Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. Apolo: Making Sense of Large Network Data by Combining Rich User Interaction and Machine Learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 167–176, New York, NY, USA, 2011. ACM. ISBN 978-1-

4503-0228-9. doi: 10.1145/1978942.1978967. URL http://doi.acm.org/10.1145/1978942.1978967.

[21] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. Towards Knowledge-Based Recommender Dialog System, 2019. URL https://arxiv.org/abs/1908.05391.

[22] Zhe Chen and Hani Doss. Inference for the Number of Topics in the Latent Dirichlet Allocation Model via Bayesian Mixture Modelling. *Journal of Computational and Graphical Statistics*, 0(ja):1–44, 2018. doi: 10.1080/10618600.2018.1558063.

[23] S. Cheng and K. Mueller. The Data Context Map: Fusing Data and Attributes into a Unified Display. *IEEE Transactions on Visualization and Computer Graphics*, 22(1): 121–130, Jan 2016. ISSN 1077-2626. doi: 10.1109/TVCG.2015.2467552.

[24] J. Choo, H. Lee, J. Kihm, and H. Park. iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pages 27–34, Oct 2010. doi: 10.1109/VAST.2010.5652443.

[25] J. Choo, C. Lee, C. K. Reddy, and H. Park. UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, Dec 2013. doi: 10.1109/TVCG.2013.212.

[26] Jaegul Choo, Hannah Kim, Edward Clarkson, Zhicheng Liu, Changhyun Lee, Fuxin Li, Hanseung Lee, Ramakrishnan Kannan, Charles D Stolper, John Stasko, et al. VisIRR: A Visual Analytics System for Information Retrieval and Recommendation for Large-Scale Document Data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(1):8, 2018.

[27] Jon Christensen, Joe Marks, and Stuart Shieber. An Empirical Study of Algorithms for Point-feature Label Placement. *ACM Transactions on Graphics*, 14(3):203–232, July 1995. ISSN 0730-0301. doi: 10.1145/212332.212334. URL http://doi.acm.org/10.1145/212332.212334.

[28] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, and H. Qu. Context preserving dynamic word cloud visualization. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 121–128, March 2010. doi: 10.1109/PACIFICVIS.2010.5429600.

[29] S. Dash, A. Verma, C. North, and W. c. Feng. Portable Parallel Design of Weighted Multi-Dimensional Scaling for Real-Time Data Analysis. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 10–17, Dec 2017. doi: 10.1109/HPCC-SmartCity-DSS.2017.2.

[30] E. P. dos Santos Amorim, E. V. Brazil, J. Daniels, P. Joia, L. G. Nonato, and M. C. Sousa. iLAMP: Exploring high-dimensional spacing through backward multidimensional projection. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 53–62, Oct 2012. doi: 10.1109/VAST.2012.6400489.

[31] M. Dowling, J. Wenskovitch, J. T. Fry, S. Leman, L. House, and C. North. SIRIUS: Dual, Symmetric, Interactive Dimension Reductions. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):172–182, Jan 2019. ISSN 2160-9306. doi: 10.1109/TVCG.2018.2865047.

[32] Michelle Dowling, John Wenskovitch, Peter Hauck, Adam Binford, Nicholas Polys, and Chris North. A Bidirectional Pipeline for Semantic Interaction. In *Proceedings of the*

Workshop on Machine Learning from User Interaction for Visualization and Analytics (IEEE VIS 2018), volume 11, 2018.

[33] Michelle Dowling, Nathan Wycoff, Brian Mayer, John Wenskovitch, Scotland Leman, Leanna House, Nicholas Polys, Chris North, and Peter Hauck. Interactive Visual Analytics for Sensemaking with Big Text. *Big Data Research*, 16:49 – 58, 2019. ISSN 2214-5796. doi: https://doi.org/10.1016/j.bdr.2019.04.003. URL http://www.sciencedirect.com/science/article/pii/S2214579618302995.

[34] Steven M. Drucker, Danyel Fisher, and Sumit Basu. Helping Users Sort Faster with Adaptive Machine Learning Recommendations. In Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, pages 187–203, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23765-2.

[35] Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017. URL http://archive.ics.uci.edu/ml.

[36] G. Ellis and A. Dix. A Taxonomy of Clutter Reduction for Information Visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, Nov 2007. ISSN 2160-9306. doi: 10.1109/TVCG.2007.70535.

[37] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 121–130, Oct 2011. doi: 10.1109/VAST.2011.6102449.

[38] A. Endert, P. Fiaux, and C. North. Semantic Interaction for Sensemaking: Inferring Analytical Reasoning for Model Steering. *IEEE Transactions on Visualization and*

Computer Graphics, 18(12):2879–2888, Dec 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.260.

[39] Alex Endert. Semantic Interaction for Visual Analytics: Toward Coupling Cognition and Computation. *IEEE Computer Graphics and Applications*, 34(4):8–15, July 2014. ISSN 0272-1716. doi: 10.1109/MCG.2014.73.

[40] Alex Endert. Semantic Interaction for Visual Analytics: Toward Coupling Cognition and Computation. *IEEE Computer Graphics and Applications*, 34(4):8–15, July 2014. ISSN 0272-1716. doi: 10.1109/MCG.2014.73.

[41] Alex Endert, Patrick Fiaux, and Chris North. Semantic Interaction for Visual Text Analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 473–482, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207741. URL http://doi.acm.org/10.1145/2207676.2207741.

[42] Alex Endert, M. Shahriar Hossain, Naren Ramakrishnan, Chris North, Patrick Fiaux, and Christopher Andrews. The human is the loop: new directions for visual analytics. *Journal of Intelligent Information Systems*, 43(3):411–435, Dec 2014. ISSN 1573-7675.

[43] David W Evans, Patrick T Orr, Steven M Lazar, Daniel Breton, Jennifer Gerard, David H Ledbetter, Kathleen Janosco, Jessica Dotts, and Holly Batchelder. Human preferences for symmetry: subjective experience, cognitive conflict and cortical brain activity. *PloS ONE*, 7(6):e38966, 2012.

[44] Jean-Daniel Fekete and Catherine Plaisant. Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. In Benjamin B. Bederson and Ben Shneiderman, editors, *The Craft of Information Visualization*, Interactive Technologies,

pages 316 – 323. Morgan Kaufmann, San Francisco, 2003. ISBN 978-1-55860-915-0. doi: https://doi.org/10.1016/B978-155860915-0/50040-8. URL https://www.sciencedirect.com/science/article/pii/B9781558609150500408.

[45] Ronen Feldman and James Sanger. *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521836573, 9780521836579.

[46] Daniel Filonik. *Participatory data analytics: Designing visualisation and composition interfaces for collaborative sensemaking on large interactive screens*. PhD thesis, Queensland University of Technology, 2017. URL https://eprints.qut.edu.au/110597/.

[47] D. Fried and S. G. Kobourov. Maps of Computer Science. In *2014 IEEE Pacific Visualization Symposium*, pages 113–120, March 2014. doi: 10.1109/PacificVis.2014.47.

[48] Barrett Golding. UnNews: An index of unreliable news websites, 2019. URL https://www.poynter.org/ifcn/unreliable-news-index/.

[49] Clinton Gormley and Zachary Tong. *Elasticsearch: The Definitive Guide*. O'Reilly Media, Inc., 1st edition, 2015. ISBN 1449358543, 9781449358549.

[50] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27:857–871, 1971.

[51] Michelle L. Gregory, Deborah A. Payne, Dave McColgin, Nick O. Cramer, and Douglas V. Love. Visual Analysis of Weblog Content. 3 2007. URL https://www.osti.gov/servlets/purl/909479.

[52] F. Heimerl, M. John, Qi Han, S. Koch, and T. Ertl. DocuCompass: Effective exploration of document landscapes. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 11–20, Oct 2016. doi: 10.1109/VAST.2016.7883507.

[53] Elizabeth Henson. Seven tornadoes hit SA on day of massive blackout, 2016. URL www.adelaidenow.com.au/news/south-australia/seven-tornadoes-hit-sa-on-day-of-massive-blackout-bureau-of-meteorology-report/news-story/e888d155c01b910778132d68e93c9d6a.

[54] Patrick Hoffman, Georges Grinstein, and David Pinkney. Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations. In *Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation in Conjunction with the Eighth ACM Internation Conference on Information and Knowledge Management*, NPIVM '99, page 9–16, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581132549. doi: 10.1145/331770.331775. URL https://doi.org/10.1145/331770.331775.

[55] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM. ISBN 1-58113-096-1. doi: 10.1145/312624.312649.

[56] Leanna House, Scotland Leman, and Chao Han. Bayesian visual analytics: BaVA. *Statistical Analysis and Data Mining*, 8(1):1–13, 2015. ISSN 1932-1872. doi: 10.1002/sam.11253. URL http://dx.doi.org/10.1002/sam.11253.

[57] F. Hughes and D. Schum. Discovery-proof-choice, the art and science of the process of intelligence analysis-preparing for the future of intelligence analysis. *Washington, DC: Joint Military Intelligence College*, 2003.

[58] Dong Hyun Jeong, Caroline Ziemkiewicz, Brian Fisher, William Ribarsky, and Remco Chang. iPCA: An Interactive System for PCA-based Visual Analytics. *Computer Graphics Forum*, 28(3):767–774, 2009. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2009.01475.x. URL http://dx.doi.org/10.1111/j.1467-8659.2009.01475.x.

[59] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato. Local Affine Multidimensional Projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571, Dec 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2011.220.

[60] Eser Kandogan. Visualizing Multi-dimensional Clusters, Trends, and Outliers Using Star Coordinates. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 107–116, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. doi: 10.1145/502512.502530. URL http://doi.acm.org/10.1145/502512.502530.

[61] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. *Visual Analytics: Definition, Process, and Challenges*, pages 154–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-70956-5. doi: 10.1007/978-3-540-70956-5_7. URL http://dx.doi.org/10.1007/978-3-540-70956-5_7.

[62] H. Kim, J. Choo, H. Park, and A. Endert. InterAxis: Steering Scatterplot Axes via Observation-Level Interaction. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):131–140, Jan 2016. ISSN 1077-2626. doi: 10.1109/TVCG.2015.2467615.

[63] Minjeong Kim, Kyeongpil Kang, Deokgun Park, Jaegul Choo, and Niklas Elmqvist. Topiclens: Efficient multi-level visual topic exploration of large-scale document col-

lections. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):151–160, 2017.

[64] Jon M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, September 1999. ISSN 0004-5411. doi: 10.1145/324133.324140.

[65] Robert R Korfhage. To see, or not to see—is That the query? In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 134–141, 1991.

[66] Josua Krause, Aritra Dasgupta, Jean-Daniel Fekete, and Enrico Bertini. SeekAView: An Intelligent Dimensionality Reduction Strategy for Navigating High-Dimensional Data Spaces. In *IEEE 6th Symposium on Large Data Analysis and Visualization*, 2016.

[67] Miloš Krstajić, Mohammad Najm-Araghi, Florian Mansmann, and Daniel A Keim. Story Tracker: Incremental visual text analytics of news story development. *Information Visualization*, 12(3-4):308–323, 2013. doi: 10.1177/1473871613493996. URL https://doi.org/10.1177/1473871613493996.

[68] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[69] Joseph B Kruskal and Myron Wish. Multidimensional scaling. *Quantitative Applications in the Social Sciences Series, Newbury Park: Sage Publications*, 11, 1978.

[70] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. AxiSketcher: Interactive Nonlinear Axis Mapping of Visualizations through User Drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, Jan 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2598446.

[71] Christoph H Lampert, Hannes Nickisch, Stefan Harmeling, and Jens Weidmann. Animals with Attributes: A Dataset for Attribute Based Classification, 2009. URL https://cvml.ist.ac.at/AwA/.

[72] Bongshin Lee, Greg Smith, George G. Robertson, Mary Czerwinski, and Desney S. Tan. FacetLens: Exposing Trends and Relationships to Support Sensemaking Within Faceted Datasets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1293–1302, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518896. URL http://doi.acm.org/10.1145/1518701.1518896.

[73] Scotland C. Leman, Leanna House, Dipayan Maiti, Alex Endert, and Chris North. Visual to Parametric Interaction (V2PI). *PLoS ONE*, 8(3):1–12, 03 2013. doi: 10.1371/journal.pone.0050474. URL http://dx.doi.org/10.1371%2Fjournal.pone.0050474.

[74] Qing Li and C. North. Empirical comparison of dynamic query sliders and brushing histograms. In *IEEE Symposium on Information Visualization 2003*, pages 147–153, Oct 2003. doi: 10.1109/INFVIS.2003.1249020.

[75] Tianyi Li, Kurt Luther, and Chris North. CrowdIA: Solving Mysteries with Crowdsourced Sensemaking. *Proceedings of the ACM on Human-Computer Interaction*, 2 (CSCW):105:1–105:29, November 2018. ISSN 2573-0142. doi: 10.1145/3274374. URL http://doi.acm.org/10.1145/3274374.

[76] Shusen Liu, Bei Wang, Jayaraman J Thiagarajan, Peer-Timo Bremer, and V Pascucci. Visual exploration of high-dimensional data: Subspace analysis through dynamic projections. *Technical Report UUSCI-2014-003*, 2014.

[77] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. imMens: Real-time Visual Querying of Big Data. *Computer Graphics Forum*, 32(3pt4):421–430, 2013. doi: 10.1111/cgf.12129. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12129.

[78] G. M. H. Mamani, F. M. Fatore, L. G. Nonato, and F. V. Paulovich. User-driven Feature Space Transformation. *Computer Graphics Forum*, 32(3pt3):291–299, 2013. ISSN 1467-8659. doi: 10.1111/cgf.12116. URL http://dx.doi.org/10.1111/cgf.12116.

[79] Robert Harry Mathams. *The intelligence analyst's notebook*. Research School of Pacific Studies, Australian National University, Strategic and Defence Studies Centre, 1988.

[80] Adrian Mayorga and Michael Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE transactions on visualization and computer graphics*, 19(9):1526–1538, 2013.

[81] Chreston Miller and Francis Quek. Interactive data-driven discovery of temporal behavior models from events in media streams. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 459–468, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1089-5. doi: 10.1145/2393347.2393413. URL http://doi.acm.org/10.1145/2393347.2393413.

[82] Chreston Miller, Louis-Philippe Morency, and Francis Quek. Structural and temporal inference search (STIS): pattern identification in multimodal data. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 101–108, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1467-1. doi: 10.1145/2388676.2388702. URL http://doi.acm.org/10.1145/2388676.2388702.

[83] Vladimir Molchanov and Lars Linsen. Interactive Design of Multidimensional Data Projection Layout. In N. Elmqvist, M. Hlawitschka, and J. Kennedy, editors, *EuroVis*

- *Short Papers*. The Eurographics Association, 2014. ISBN 978-3-905674-69-9. doi: 10.2312/eurovisshort.20141152.

[84] Kai A. Olsen, Robert R. Korfhage, Kenneth M. Sochats, Michael B. Spring, and James G. Williams. Visualization of a document collection: The VIBE system. *Information Processing & Management*, 29(1):69 – 81, 1993. ISSN 0306-4573. doi: https://doi.org/10.1016/0306-4573(93)90024-8. URL http://www.sciencedirect.com/science/article/pii/0306457393900248.

[85] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Number 1999-66. Stanford InfoLab, November 1999. URL http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120.

[86] D Paranyushkin. Visualize any text as a network—Textexture. http://textexture.com/, 2012. Accessed: November 18, 2012.

[87] F.V. Paulovich, D.M. Eler, J. Poco, C.P. Botha, R. Minghim, and L.G. Nonato. Piecewise Laplacian-based Projection for Interactive Data Exploration and Organization. *Computer Graphics Forum*, 30(3):1091–1100, 2011. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2011.01958.x. URL http://dx.doi.org/10.1111/j.1467-8659.2011.01958.x.

[88] Karl Pearson. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895. ISSN 03701662. URL http://www.jstor.org/stable/115794.

[89] Karl Pearson. Principal components analysis. *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, 6(2):566, 1901.

[90] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. *Proceedings of International Conference on Intelligence Analysis*, 5, 2005.

[91] Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, Chris Kuhlman, Achla Marathe, Liang Zhao, Ting Hua, Feng Chen, Chang Tien Lu, Bert Huang, Aravind Srinivasan, Khoa Trinh, Lise Getoor, Graham Katz, Andy Doyle, Chris Ackermann, Ilya Zavorin, Jim Ford, Kristen Summers, Youssef Fayed, Jaime Arredondo, Dipak Gupta, and David Mares. 'Beating the News' with EMBERS: Forecasting Civil Unrest Using Open Source Indicators. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1799–1808, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2956-9. doi: 10.1145/2623330.2623373.

[92] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[93] George Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. Data Mountain: Using Spatial Memory for Document Management. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 153–162, New York, NY, USA, 1998. ACM. ISBN 1-58113-034-1. doi: 10.1145/288392.288596. URL http://doi.acm.org/10.1145/288392.288596.

[94] A. C. Robinson. Collaborative synthesis of visual analytic results. In *2008 IEEE Symposium on Visual Analytics Science and Technology*, pages 67–74, Oct 2008. doi: 10.1109/VAST.2008.4677358.

[95] Tuukka Ruotsalo, Jaakko Peltonen, Manuel Eugster, Dorota Glowacka, Ksenia Konyushkova, Kumaripaba Athukorala, Ilkka Kosunen, Aki Reijonen, Petri Myllymäki, Giulio Jacucci, and Samuel Kaski. Directing exploratory search with interactive intent modeling. In *Proceedings of the 22nd ACM international conference on information and knowledge management*, CIKM '13, pages 1759–1764, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2263-8. doi: 10.1145/2505515.2505644. URL http://doi.acm.org/10.1145/2505515.2505644.

[96] D. M. Russell, M. Slaney, Yan Qu, and M. Houston. Being Literate with Large Document Collections: Observational Studies and Cost Structure Tradeoffs. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 3, pages 55–55, Jan 2006. doi: 10.1109/HICSS.2006.73.

[97] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[98] H. Samin and T. Azim. Knowledge Based Recommender System for Academia Using Machine Learning: A Case Study on Higher Education Landscape of Pakistan. *IEEE Access*, 7:67081–67093, 2019. doi: 10.1109/ACCESS.2019.2912012.

[99] Karsten Schatz, Christoph Müller, Michael Krone, Jens Schneider, Guido Reina, and Thomas Ertl. Interactive visual exploration of a trillion particles. In *2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 56–64. IEEE, 2016.

[100] Jessica Zeitz Self, Nathan Self, Leanna House, Jane Robertson Evia, Scotland Leman, and Chris North. Bringing Interactive Visual Analytics to the Classroom for Developing EDA Skills. Technical Report, Virginia Tech, Blacksburg, 2015.

[101] Jessica Zeitz Self, Xinran Hu, Leanna House, Scotland Leman, and Chris North. Designing Usable Interactive Visual Analytics Tools for Dimension Reduction. In *CHI 2016 Workshop on Human-Centered Machine Learning (HCML)*, page 7, 05/2016 2016.

[102] Jessica Zeitz Self, Radha Krishnan Vinayagam, J. T. Fry, and Chris North. Bridging the Gap Between User Intention and Model Parameters for Human-in-the-loop Data Analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, HILDA '16, pages 3:1–3:6, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4207-0. doi: 10.1145/2939502.2939505. URL http://doi.acm.org/10.1145/2939502.2939505.

[103] Jessica Zeitz Self, Michelle Dowling, John Wenskovitch, Ian Crandell, Ming Wang, Leanna House, Scotland Leman, and Chris North. Observation-Level and Parametric Interaction for High-Dimensional Data Analysis. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):15:1–15:36, June 2018. ISSN 2160-6455. doi: 10.1145/3158230.

[104] Jinwook Seo and Ben Shneiderman. Interactively exploring hierarchical clustering results [gene identification]. *Computer*, 35(7):80–86, 2002.

[105] J. Sharko, G. Grinstein, and K. A. Marx. Vectorized Radviz and Its Application to Multiple Cluster Datasets. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1444–1427, Nov 2008. ISSN 1077-2626. doi: 10.1109/TVCG.2008.173.

[106] Frank M. Shipman and Catherine C. Marshall. Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems. *Computer Supported Cooperative Work (CSCW)*, 8(4):333–352, 1999. ISSN 1573-7551. doi: 10.1023/A:1008716330212. URL http://dx.doi.org/10.1023/A:1008716330212.

[107] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11 (6):70–77, Nov 1994. ISSN 0740-7459. doi: 10.1109/52.329404.

[108] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, Sep. 1996. doi: 10.1109/VL.1996.545307.

[109] Sebastian Sippl, Michael Sedlmair, and Manuela Waldner. Collecting and Structuring Information in the Information Collage, 2019.

[110] Michael Slezak. South Australia's blackout explained and no, renewables aren't to blame, 2016. URL https://www.theguardian.com/australia-news/2016/sep/29/south-australia-blackout-explained-renewables-not-to-blame.

[111] G. Smith, M. Czerwinski, B. Meyers, D. Robbins, G. Robertson, and D. S. Tan. FacetMap: A Scalable Search and Browse Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):797–804, Sep. 2006. doi: 10.1109/TVCG.2006.142.

[112] Mica Soellner. Donald Trump wrongly suggests British don't love their health care system, Feb 2018. URL https://www.politifact.com/factchecks/2018/feb/08/donald-trump/donald-trump-wrongly-suggests-british-dont-love-th/.

[113] Square Inc. Crossfilter: Fast multidimensional filtering for coordinated views, 2013. URL http://square.github.io/crossfilter/.

[114] Sandra D. Starke and Chris Baber. The effect of four user interface concepts on visual scan pattern similarity and information foraging in a complex decision making task. *Applied Ergonomics*, 70:6 – 17, 2018. ISSN 0003-6870. doi: https://doi.org/10.1016/j.

apergo.2018.01.010. URL http://www.sciencedirect.com/science/article/pii/S0003687018300188.

[115] John Stasko, Carsten Görg, and Zhicheng Liu. Jigsaw: Supporting Investigative Analysis through Interactive Visualization. *Information Visualization*, 7(2):118–132, 2008. doi: 10.1057/palgrave.ivs.9500180.

[116] Warren S Torgerson. *Theory and methods of scaling*. Wiley, Oxford, England, 1958.

[117] C. Turkay, A. Lundervold, A. J. Lundervold, and H. Hauser. Representative Factor Generation for the Interactive Visual Analysis of High-Dimensional Data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2621–2630, Dec 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.256.

[118] Cagatay Turkay, Peter Filzmoser, and Helwig Hauser. Brushing dimensions-a dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2591–2599, 2011.

[119] A. C. Valdez, M. Ziefle, and M. Sedlmair. Priming and Anchoring Effects in Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):584–594, Jan 2018. doi: 10.1109/TVCG.2017.2744138.

[120] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(11):2579 – 2605, 2008. ISSN 15324435.

[121] E. Wall, S. Das, R. Chawla, B. Kalidindi, E. T. Brown, and A. Endert. Podium: Ranking Data Using Mixed-Initiative Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):288–297, Jan 2018. ISSN 1077-2626. doi: 10.1109/TVCG.2017.2745078.

[122] Xu-Meng Wang, Tian-Ye Zhang, Yu-Xin Ma, Jing Xia, and Wei Chen. A Survey of Visual Analytic Pipelines. *Journal of Computer Science and Technology*, 31(4):787–804, 2016. ISSN 1860-4749. doi: 10.1007/s11390-016-1663-1. URL http://dx.doi.org/10.1007/s11390-016-1663-1.

[123] Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X. Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. TIARA: A Visual Exploratory Text Analytic System. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 153–162, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1. doi: 10.1145/1835804.1835827.

[124] J. Wenskovitch and C. North. Pollux: Interactive cluster-first projections of high-dimensional data. In *2019 IEEE Visualization in Data Science (VDS)*, pages 38–47, Oct 2019. doi: 10.1109/VDS48975.2019.8973381.

[125] J. Wenskovitch, L. Bradel, M. Dowling, L. House, and C. North. The Effect of Semantic Interaction on Foraging in Text Analysis. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 13–24, Oct 2018. doi: 10.1109/VAST.2018.8802424.

[126] John Wenskovitch and Chris North. Observation-Level Interaction with Clustering and Dimension Reduction Algorithms. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, HILDA'17, pages 14:1–14:6, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5029-7. doi: 10.1145/3077257.3077259. URL http://doi.acm.org/10.1145/3077257.3077259.

[127] John Wenskovitch, Ian Crandell, Naren Ramakrishnan, Leanna House, Scotland Leman, and Chris North. Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics. *2017 IEEE Transactions on Visualization*

and Computer Graphics Proceedings of the Visual Analytics Science and Technology (VAST), 24(01), January 2018.

[128] John Wenskovitch, Michelle Dowling, Laura Grose, Chris North, Remco Chang, Alex Endert, and David H. Rogers. Machine Learning from User Interaction for Visualization and Analytics: A Workshop-Generated Research Agenda. In *Proceedings of the IEEE VIS Workshop MLUI 2019: Machine Learning from User Interactions for Visualization and Analytics*, VIS'19, 2019.

[129] L. Wilkinson. Visualizing Big Data Outliers Through Distributed Aggregation. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):256–266, Jan 2018. ISSN 2160-9306. doi: 10.1109/TVCG.2017.2744685.

[130] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proceedings of Visualization 1995 Conference*, pages 51–58, Oct 1995. doi: 10.1109/INFVIS.1995.528686.

[131] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37 – 52, 1987. ISSN 0169-7439. doi: https://doi.org/10.1016/0169-7439(87)80084-9. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[132] Pak Chung Wong, Elizabeth G Hetzler, Christian Posse, Mark A Whiting, Susan Havre, Nick Cramer, Anuj R Shah, Mudita Singhal, Alan Turner, and Jim Thomas. IN-SPIRE InfoVis 2004 Contest Entry. In *INFOVIS*, volume 4, pages 51–52, 2004.

[133] Cong Xie, Wen Zhong, and Klaus Mueller. A Visual Analytics Approach for Categorical Joint Distribution Reconstruction from Marginal Projections. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):51–60, 2017.

[134] Lu Xu, Yang Xu, and Tommy W.S. Chow. PolSOM: A new method for multidimensional data visualization. *Pattern Recognition*, 43(4):1668–1675, 2010. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/j.patcog.2009.09.025. URL http://www.sciencedirect.com/science/article/pii/S0031320309003690.

[135] Yang Xu, Lu Xu, and Tommy W.S. Chow. PPoSOM: A new variant of PolSOM by using probabilistic assignment for multidimensional data visualization. *Neurocomputing*, 74(11):2018–2027, 2011. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2010.06.028. URL http://www.sciencedirect.com/science/article/pii/S0925231211000385.

[136] CS Yang. On Dynamic Document Space Modification Using Term Discrimination Values. *Scientific Report ISR-22, Department of Computer Science, Cornell University*, 1974.

[137] Ji Soo Yi, Rachel Melton, John Stasko, and Julie A. Jacko. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information Visualization*, 4 (4):239–256, 2005.

[138] Xiaoru Yuan, Donghao Ren, Zuchao Wang, and Cong Guo. Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2625–2633, 2013.

[139] Germain Garcia Zanabria, Luis Gustavo Nonato, and Erick Gomez-Nieto. iStar (i*): An interactive star coordinates approach for high-dimensional data exploration. *Computers & Graphics*, 60:107–118, 2016. ISSN 0097-8493. doi: http://dx.doi.org/10.1016/j.cag.2016.08.007. URL http://www.sciencedirect.com/science/article/pii/S0097849316301054.