Interaction with Multiple Data Visualizations Through Natural Language Commands

by

Jillian Aurisano
M.S., University of Illinois at Chicago, 2014; A.B., University of Chicago, 2006

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2021

Chicago, Illinois

Defense Committee:
Andrew Johnson, Chair and Advisor
Debaleena Chattopadhyay, University of Illinois at Chicago
Barbara Di Eugenio, University of Illinois at Chicago
G. Elisabeta Marai, University of Illinois at Chicago
Moira Zellner, Northeastern University

To my family.

ACKNOWLEDGMENT

This dissertation exists in large part thanks to the support and help of many people in my life.

I would like to first acknowledge my advisor, Andrew Johnson, for mentoring and supporting me throughout my graduate career. I am so thankful to have been able to work with an advisor who allowed me to pursue my own ideas and to take risks on projects that I believed in. Thanks to my committee for their feedback on my research : Barbara Di Eugenio, Liz Marai, Debaleena Chattopadhyay, and Moira Zellner. I owe special thanks to Barbara for her years of support working with me on Articulate. I also want to thank Jason Leigh, for advising me during my MS degree and for giving me my start in the EVL. Thanks to Maxine Brown and Dana Plepys for their support and words of wisdom. Thanks to Tanya Berger-Wolf her supporting me, and sending interesting projects my way, and for generally supporting my career. Thanks to Daria Tsoupikova for being a sounding board in the next step in my career. Thanks to Isabel Cruz for the support graduate school and beyond. Thanks to Liz Marai for supporting my work within the Vis and BioVis community. Thanks to Bob Sloan and Dale Reed for their support of my work with the Women in Computer Science organization. Thanks also to John Lillis for teaching my first intro to CS course, which convinced me to give computer science a try, and for steering me toward the EVL. Thanks also to Rick Stevens, Fangfang Xia, and Mike Papka at Argonne National Laboratory, for supporting my research during my PhD.

I am thankful for the all the people I worked with in the Electronic Visualization Laboratory, (EVL) and the culture in the EVL. Before I came to the EVL, I had given up on the idea that I could have a career that I would be passionate about. Finding my way into the EVL, was one of the greatest strokes of luck in my life. Working here showed me that academic research could be fun, exciting and inspiring, and I am so thankful. I was lucky to be able to work with so many inspiring students, who also helped me along the way. In particular, thanks to Abeer Alsaiari, for being an amazing and supportive collaborator, who was willing to be a sounding board on many of my ideas, and who helped me finish my evaluation when I could not come to campus due to the pandemic. Thanks to Abhinav Kumar, for being an amazing collaborator throughout our work on Articulate. Thanks to Arthur Nishimoto, for all his work setting up interaction servers in the EVL, and debugging issues with the Kinect and touch on the wall in Continuum, and just generally being an inspiring creator in the EVL. Thanks to Lance, for helping me with all the technical challenges along the way, as well as for friendship, advice, support and encouragement. I also want to thank Krishna Bharadwaj, Andrew Burks, Ashwini Naik, John Novak, Dennis Chau, Alessandro Febretti, Abhinav Kumar, Timothy Luciani, Chihua Ma, Tommy Marrinan, Paul Murray, Victor Mateevitsi, Kyle Almryde, Alex Simes, Steve Conner, Paul Grenning, Kristine Lee, Khairi Reda, and JD Pirtle for friendship, support and encouragement during my time in the EVL. Thanks for the Munchkins games, the all-you-can-eat sushi outings, the rock-wall climbing and general antics that made my time in graduate school an absolute pleasure. And thank you to all the other EVL students during my time here: Julian Chen, Hyejung Hur, Ratko Jagodic, Ed Kahler, Sangyoon Lee, Michael Lewis, Qin Li, Sungwon Nam, Shiwangi Singh, Yiwen Sun, Muxuan Wang, and Shi Yin.

Also, thanks to all the undergrads I worked with during my time at the EVL: Ishta Bhagat, Oliver San Juan, Tyrone Harris, James Hwang, Vasanna Nguyen, Krupa Patel, Ryan Fogarty, Joe Borowicz, and Vijay Mahida. Thanks to Vasanna, Krupa, Ryan, Joe and Vijay for working with me on systems that ultimately became the core of 'Ditto'.

Thanks to all my friends- particularly Brian Gomoll and Tristan Clark- for being there for me through it all. Thanks to Kisa Marx for showing me that childcare can be beautiful, which allowed me to work and go after my dreams knowing that my kids were in exactly the place they needed to be. Thanks to all the teachers and caregivers for my kids, who have allowed me to work knowing they were happy and doing interesting things.

I cannot thank my parents enough. Thanks to my Dad, for persisting in suggesting that I take a computer science course, because he really thought it would be a good a fit for me. It took around a decade for this advice to be followed, but he was right. This was a good fit for me. And thank you for carrying all of my babies around, and taking them on walks, so that I could do my work. Especially during the pandemic- I could not have written this thesis if it you had not come and helped with all the kids when we lost our childcare due to the pandemic. Thanks to my Mom for encouraging me to leave my wet-lab biology research job because it as not a good fit for me. You were right. It was a terrible fit for me. Fortunately, it did not take me a decade to realize this, and I got out in time to discover something new and better. And thank you for taking care of all the kids during the pandemic, so I could work and finish this thesis. And thank you for the constant encouragement, through this whole journey.

Thanks to Rick and Maria Aurisano for being amazing cheerleaders and wonderful in-laws and grandparents. Thanks to my grandparents for supporting me: Grandma Jane for her scientific, feminist wisdom; Grandpa Don for always believing in me; Grandma Margaret and Grandpa Ron for their constant support. Thanks to my sister, and all of my extended family, for being a constant source of encouragement and support.

Thanks to my kids for your curiosity, your energy, and for keeping me grounded. Kids: Thanks for being patient with me- I know I sometimes had to write papers on my phone while nursing you, and I may have dropped the phone on your heads by accident a few times while trying to get out a particularly complicated thought, but you seem to be doing okay in spite of this. Thanks for coming to conferences with me- I am not sure how one gives a poster without holding a baby, is it even possible? Thanks for adapting when I had to work, and you wanted to play, or when I had to work during bedtime. Normally, I worked when you were in school, but with the pandemic, and homeschooling, we really had to figure out a way for me to spend time you and still get my thesis done. I know that I am a happier Mom for being able to pursue my dreams while having 1,2,3,...4! babies and watching you grow into amazing people, and I hope I can show you that you can do amazing, hard things in your lives too!

And thanks to Adam. None of this would have happened without you. Your support is everything.

JMRA

CONTRIBUTIONS OF AUTHORS

Chapter 1 provides an introduction to the topic of this dissertation. **Chapter 2** provides an overview of related work, touching on visualization construction, natural language interaction with visualization, large displays for visualization, and visualization recommendations. **Chapter 3** includes portions of a published paper, (Aurisano et al., Many At Once: Capturing Intentions to Create And Use Many Views At Once In Large Display Environments, 2019), for which I am the first author. This work was completed in collaboration with Abhinav Kumar, Abeer Alsaiari, Andrew Johnson and Barbara Di Eugenio, with vital contributions from Khairi Reda, Alberto Gonzales and Jason Leigh. **Chapter 4** presents an interaction technique to support breadth in data exploration, through generating sets of visualizations using natural language commands. This work is primarily my own, but was conducted with feedback from Abeer Alsaiari, Abhinav Kumar, Andrew Johnson, Barbara Di Eugenio, Moira Zellner, and Anuj Tiwari. **Chapter 5** extends the approach developed in Chapter 4 to large display environments, and multi-modal speech and mid-air gesture interactions. This work is primarily my own, but I collaborated with Abeer Alsaiari and Abhinav Kumar in developing and testing the interface, and mentored a number of incredible undergraduate students, who assisted in developing the speech and mid-air gesture interaction system (Vasanna Nguyen, Krupa Patel, Ryan Fogarty, Joseph Borowicz, and Vijay Mahida). Andrew Johnson, Barbara Di Eugenio, Jason Leigh and Moira Zellner provided valuable feedback on this Chapter. The dissertation was guided by Andrew Johnson and Barbara Di Eugenio, with valuable feedback from my thesis committee.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

FIGURE **PAGE**

EXECUTIVE SUMMARY

Data exploration stands to benefit from environments that permit users to examine and juxtapose many views of data, particularly views that present diverse selections of data values and attributes. Large, high-resolution environments are capable of showing many related views of data, but efficiently creating and displaying visualizations in these environments presents significant challenges. In this dissertation, I will present my research on “multi-view data exploration interactions” that enable users to create sets of views with coherent data value and attribute variations, through multi-modal speech and mid-air pointing gestures in large display environments. This work enables users to rapidly and efficiently generate sets of views in support of multi-view data exploration tasks, organize these views in coherent collections, and operate on sets of views collectively, rather than individually, to efficiently reach large portions of the 'data and attribute space'. I will present three contributions: 1) an observational study of data exploration in a large display environment with speech and mid-air gestures, 2) 'Traverse', an interaction technique for data exploration, based on this study, which uses natural language to create and pivot sets of views, and 3) 'Ditto', a multi-modal speech and mid-air pointing gesture interactive environment, which utilizes the multi-view data exploration technique, in large display environments.

CHAPTER 1

INTRODUCTION

We live in a data-driven world, where our lives increasingly intersect with large and opaque data, so we need technical and computational approaches that help us grapple with this complexity in ways that keep humans 'in the loop' with their data (2). Visualization is a key tool in empowering people to manage and command data-driven problems (3). However, as data grows in scale and complexity, new visualization and interaction approaches may be needed.

J. Tukey describes exploratory data analysis (EDA) as a process of "looking at data to see what it seems to say" (4). EDA is important in helping us grapple with data, because it aims to allow analysts to approach data prior to statistical analysis, and prior to having precisely formed analysis goals. Through EDA, analysts first develop an understanding of data, and then use this understanding to develop computational or statistical approaches (5). Through EDA, analysts often approach a large dataset incrementally, by segmenting data into meaningful partitions and representing these parts in multiple, related views (6). EDA proceeds from both targeted questions, open-ended inquiry and observation driven steps through a dataset, and the analyst gradually shifts between points of interest based on prior knowledge, evolving goals and observations (7). In exploring data, users develop an understanding of a dataset incrementally. From a visual design standpoint, this means that the goal is not to present data in a single, comprehensive overview, instead the goal is to enable users to create many views of their data, that show varied selections of data values and attributes, and make it easy for users to transition from one set of interests to another (8). Through this process, users can build a mental model of the data as they expand the focus of their exploration to new portions of the 'data and attribute space' (5; 9; 10; 11).

However, a major obstacle for exploratory data analysis is visualization construction, which has been found to be complex and error-prone with users facing obstacles to selecting appropriate templates and mapping data to visual elements (12). EDA entails multiple rounds of visualization construction, which requires users to repeatedly translate their questions about the data into concrete representations, on top of already complex sensemaking tasks such as insight and goal formation (5; 8; 11).

One approach to this challenge is to enable users to directly specify intended visualizations, using interaction modalities beyond mouse and keyboard (13). The idea driving this line of research is that other interaction modalities might be more expressive, and may enable users to translate their intentions more easily into visualizations, allowing them to focus on their data, their questions and their analysis tasks. Natural language has been of particular interest, because it allows users to pose high-level questions without manually specifying data encodings and visual templates, or learning a complex graphical interface. There has been tremendous growth in this research area, with promising results (14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29).

In parallel, researchers have considered how to provide an environment for creating and displaying multiple views of data to support exploratory data analysis. One approach is to provide an environment for EDA that externalizes the incremental and evolving exploratory process. The idea here is to capture transitions between points of interest, so that users can review and revisit their process (11). Often this takes the form of a flexible space, such as a large virtual canvas, where visualizations can be created, freely positioned and juxtaposed to enable sensemaking over many views. Techniques for creating visualizations in these contexts often employ methods to act directly on prior views in order to create new ones, such as duplicating and pivoting a visualization from one set of interests in the data to another, creating a trail of exploratory steps (30; 31; 32; 33; 34; 28) .

Research on large displays have proceeded in a related direction. Interest in large displays grew out of findings that analysts use abundant display spaces to organize analysis artifacts into conceptually meaningful configurations in support of sensemaking tasks(35). Large displays have also been found to provide cognitive benefits, such as using spatial memory and embodied cognition to navigate large data spaces (35; 36; 37). In addition, large displays allow users to organize visualizations into meaningful configurations (38; 39), and develop complex and integrative insights across multiple views of data (40).

However, interaction in large display environments can pose challenges, and more research is needed to develop interaction techniques that take advantage of the unique properties of large displays for visual data exploration. Input modalities beyond mouse and keyboard, such as speech, mid-air gestures, touch, proxemics and mobile devices, are promising methods for interaction with large displays, but there are things we do not fully understand about how to use these modalities and how to accommodate the potential synergy between multiple interaction modalities in real data analysis scenarios (41).

1.1 Overview of Contributions

One vision for addressing interaction challenges for exploratory data analysis in large display environments is an 'Attentive Space'. This environment would responds to a range of inputs from users, particularly inputs such as speech and mid-air gestures that allow users to move around and engage in collaborative discussion with others, and create visualizations of their data on-the-fly without breaking 'the flow' of data exploration. An attentive space would leverage the cognitive and perceptual benefits of large displays for sensemaking in data exploration (35; 42; 40; 43), while also addressing visualization creation challenges in this environment (41), by offloading visualization construction and refinement onto a system that responds more directly to their exploratory interests. While these ambitious goals are not fully realized in this dissertation, it contributes to this long-term effort.

In this dissertation, I first set out to understand how people use large display environments to explore data, in particular how they might use conversational speech and gestures to convey their intentions to a 'smart assistant', who could act on their behalf and respond to their requests. With my collaborators, I conducted an observational, exploratory study to examine this more closely, and we found that participants in our study adopted an interaction style that was effective in requesting many views of data, that could be arranged in coherent configurations, and that supported complex tasks that spanned more than one view of the data, using a combination of 'targeted' requests, 'cast-a-net' requests- through which participants expressed multiple, complex points of interest in one command, and 'referential requests', which utilized prior views as templates for requesting new visualizations. We describe these findings in more detail in Chapter 3, and in our publication (44), as well as in our publications within the natural language processing research community (20; 21; 22; 23).

This initial study raised a set of questions, because the interaction approach expressed by participants was not reflected in current data exploration techniques. First, we needed to figure out an approach for using natural language to create not just one view, in targeted requests, but potentially many views at once, based on multiple expressed points of interest (cast-a-net requests). This presented challenges because prior work in this area did not offer formalisms for translating multiple data value and attribute interests into coherent sets of views, and this had not been explored for natural language inputs. Second, we needed to develop approaches for characterizing the 'referential actions' we observed in our initial study. Addressing these questions involved designing novel interactions in support of multi-view exploratory tasks. We present our designs and an implemented system 'Traverse', which takes natural language commands about interests in the data, and creates coherent sets of views, that we term 'view collection.' These view collections can then be referenced and acted upon, either expanded or copied and pivoted, to help users 'traverse' a dataset, through multiple, evolving points of interest. We found that participants were able to efficiently use both direct and referential actions, to create multiple views, and views with diverse compositions of data values and attributes, and that many found data exploration in this context to be enjoyable and easy to learn. This design process is described in Chapter 4.

Finally, we returned to the large display and multi-modal speech and mid-air gesture context, and considered how to realize the interaction approach we observed in the first study. This entailed both building an input system to accommodate multi-modal speech and mid air gesture interactions, and considering how to combine these inputs together for data exploration tasks. We also needed to consider the role of spatial positioning and custom configurations of views, that are useful in a large display environment for complex sensemaking tasks. We evaluated these design choices in a system we call 'Ditto', which was a name chosen to highlight the core 'duplicate and pivot' mechanism of our technique for multi-modal speech and mid-air gestures. We observed that participants used both direct and referential actions, and used our multi-modal speech and gesture input system, to create and then copy and pivot sets of views to explore data. Participants organized these views into meaningful groups on the large display, using touch, and then indicated one or many of the views, to duplicate and pivot in new exploratory directions. In addition, participants dynamically referenced views from prior points in their exploration, which suggests that the large display, and the ability to reference prior visualizations, enabled exploration of complex and evolving points of interest. These contributions are described in Chapter 5.

To my knowledge, **this dissertation, and closely coupled prior work completed with my collaborator Abhinav Kumar (20; 21; 22; 45; 23; 44), contributes the first multi-modal speech and mid-air gesture system for large displays and for data exploration.** This realizes a longstanding ambition in human computer interaction, dating back to work from MIT media lab in the late 1970s and 1980s in 'Put that there' (46). Given the trajectory of large displays, which are becoming easier to build and drive with a single machine (47; 42), as well as the trend toward capturing and responding to natural language spoken inputs (48), as well as in capturing detailed physical movements of people in environments through depth sensors (49), I anticipate that interest in multi-modal speech and mid-air gestures for visualization in large display environments will increase in the next few years. The contributions of this dissertation will further efforts to leverage the combined properties of abundant display space, and natural language spoken inputs with mid-air gestures, for new approaches to longstanding challenges in data visualization.

CHAPTER 2

RELATED WORK

Data exploration is an iterative process of generating insights about a dataset, prior to in-depth analysis involving statistical models or hypothesis testing (4). Exploring a dataset involves a combination of open-ended inquiry, where there may not yet be a precisely formulated goal, as well as focused inquiry, based on the knowledge or interests of the user (7). Visualization is an essential part of data exploration, because it can present data in an accessible format that leverages the powerful human visual system for identifying interesting patterns or features within the data - essentially using vision to think (3). When visualization is used to support data exploration, the exploratory process can be 'bottom-up', grounded in observed features in the data. The end point of data exploration can be a precisely formed analysis goal, that can be addressed through computational or statistical approaches (5).

With large and complex data, it is difficult to construct perceptually meaningful and accessible visual overviews, which presents problems for the canonical information visualization paradigm 'overview first, zoom and filter, details on demand' (50; 51; 52). One approach to visual data exploration is to instead segment data into small and meaningful selections, based on user interests and data driven observations, allowing a user to gain familiarity with the data incrementally, and identify potential analysis methods and priorities (30; 39).

There are several major approaches in visualization research that aim enable visual data exploration.

The first considers reducing challenges in constructing visualizations (12). This includes an overview of view construction models and theory, as well as tool sets and graphical interfaces that make it easy to construct a view of data (53; 54; 55; 56; 57; 58; 59). If it is easy to visualize data, then it will be possible for people to construct views in support of their exploratory tasks. I will refer to this as Area 1, and will provide an overview of this work.

The second approach considers ways to use interactions 'beyond mouse and keyboard', or 'post-WIMP'- post windows, icons, menus and points- to enable users to construct visualizations more naturally, and without learning how to navigate a complex graphical interface (13). While this includes interaction modalities such as sketching (60; 61), tangibles (62), direct manipulation (27) and touch (63; 64), among others, we will focus on natural language interaction to generate views of data (14; 15; 16; 26; 25; 20; 21; 23; 22; 45). I will refer to this as Area 2.

The third considers ways to externalize the visual data exploration process, and directly support exploratory tasks. Many of these adopt a 'flexible canvas' environments for displaying visualization, which also retains a history of incremental exploratory actions- 'mutating' steps through different selections of data values and attributes (30). These approaches also provide an environment that accommodates more than one view of the data- to show exploration history across a series of views (30; 31; 32). The benefit of this approach is that analysts can offload memory onto perception (65). I will refer to this as Area 3.

The fourth approach considers the benefit of large display environments for visual data exploration and sensemaking (35). In particular, how abundant display space allows users to externalize many views of data produced during the exploratory process and perform tasks that span more than one view, and arrange these views into conceptually meaningful groups (38; 39), that can be viewed both up-close and at a distance (66). There is also work in this area that relates to area 2- post-WIMP interactions- where new interaction modalities are considered in response to specific interaction challenges pertaining to large display environments (41; 49). I will refer to this as Area 4.

The fifth approach focuses on guidance and recommendation for data visualizations (6). This includes helping users consider next steps in their exploration of data (1), and includes avoiding fixation and rabbit holes, that may lead to erroneous conclusions (67). While our work does not directly contribute to research in visualization recommendation, it is informed by theory about the recommendations for users based on likely next steps in typical exploratory processes (1). I will refer to this as Area 5.

Area 1	Visualization Construction	2.1
Area 2	Natural Language Interaction	2.2
Area 3	Data Exploration	2.3
Area 4	Large Displays	2.4
Area 5	Visualization Recommendations	2.5

This research began with a novel observational study which combined aspects of several of these areas. The observation study was conducted in a large display environment (Area 4), where users requested views of data in support of data exploration using combined speech and mid-air gestures (Area 1 and Area 2), and responses were visualizations that externalized their exploration process (Area 3). These combined factors led to observations of interaction approaches that were not well characterized or reflected in current visual data exploration tools.

In this section I will discuss major research contributions in each of these areas and how they relate to the contributions of this dissertation.

2.1 Visualization Construction Challenges and Approaches

Visualization construction, broadly, may target a variety of goals (68), including tasks that contribute to developing an understanding of a data set in a larger exploratory process. In this section, I will discuss the view construction process, view construction roadblocks and challenges, and briefly mention tools for manual view specification, that are designed to make visualization construction easier.

2.1.1 View construction and roadblocks

When creating a visualization, raw data needs to be transformed into a visual encoding and presented to the user. There are several models for this process, but the model from Card et al. is prominent. It presents a step-wise model for visualization construction. First, data must be transformed from its raw format, filtered and aggregated, into data tables. These data tables are then mapped to visual templates, such a bar charts or line charts, so that the data can be visually represented to the user. After rendering the view, users might interact to transform the view by zooming or changing the level of abstraction (3). Other models include Chi's data state model (69), and Ware's model incorporating the human perceptual system (70). But, these models share an emphasis on step-wise transitions from data, to abstractions, to visual encodings that can be represented to the user.

In practice, view construction can be difficult and error prone. Grammel et al. performed an empirical study of visualization construction. In this work, 'infovis novices', those not trained in visualization construction, were recruited to participate in a laboratory study, where they verbally specified intended views to a remote mediator, who translated their requests into realized views using Tableau.

The first outcome from this study was a model of the view construction process. They found an iterative process in designing views: data attribute selection, visual template selection, visual mapping specification. This process may proceed in different orders, as illustrated in the state diagram depicting these activities.

The second outcome from this study were descriptions of view construction roadblocks, or places where participants struggled to describe intended views. Infovis novices struggled to translate questions into appropriate visualizations, including template selection, choosing mappings, layouts and encodings. In addition, they found that partial specification was common, and the authors list different partial specification types (such as failing to specify visual mappings for selected data attributes or failing to specifiy abstractions and groupings, such as levels of abstraction for temporal variables). Grammel et al. argue that partial specification arises because the users' mental models are simplistic and they omit elements that are assumed to be inferred from the present context in which their query is posed (eg. the current view or prior views provided).

The take-away from this study is that even when a user interface is intuitively designed, and when there is no need for the user to learn coding or scripting for visualization creation, users may still make errors, and that tools should make some of these selections for users by supplying defaults or providing view suggestions.

This dissertation focuses on enabling users to generate views of their data. However, I focus less on overcoming visualization construction roadblocks, and more on enabling breadth in data exploration- a diversity of data value and attribute selections- and repeated cycles of visualization construction. However, I build on Grammel et al.'s finding that participants often utilized visual templates or prior views in posing requests for specific visualization. This finding provides additional empirical support for the 'referential operations' described in this dissertation.

2.1.2 Tools to facilitate view creation

There are a variety of tools that assist with visualization construction. In many contexts users specify views manually, through direct interaction with a graphical interface or through specifying view components directly. With these tools and languages, users select data variables (eg. columns from a tabular dataset) as well as data transformations, such as aggregation or binning, and apply filters to select subsets of the data of interest. Users then select a template (such as a bar or line graph) through which to visually encode the data, and then may apply colors and scales and for the selected data and view, producing the final result.

This can be done through high-level languages- such as protovis (53), d3 (54), ggplot2 (55)- high-level coding grammars, such as Vega (57), vegalite (56) or through interaction with a visualization generation interface such as through Polaris/Tableau (59), ivisdesigner (58), and lyra (71). Graphical-interface based view generation allow users to select data variables through selections within a menu or through drag-and-drop operations. These actions translate into queries on databases, which in the case of Tableau produce views that can be seen one-at-a-time or in multi-view dashboards.

Polaris, which is a foundation of Tableau, a widely adopted software for view generation, uses a drag-and-drop interaction style to facilitate view creation without coding. The user selects view templates through a menu and maps data variables to axes, shapes or colors. Tableau has added additional features, such as natural language view creation, to lower the barrier to view construction. (59).

2.2 Natural Language Interaction With Visualizations

In the previous section, I discussed research suggesting that visualization construction can be difficult and error prone. One approach to address this challenge is to consider visualization generation through alternate interaction modalities that are more expressive. These include sketching interfaces that combine pen and touch interactions (60; 61), interactions with tangibles (62). and direct interaction (27).

Natural language queries also present a promising alternative to manual specification, by allowing users to directly pose questions without translating high-level queries into low-complexity interaction primitives with a potentially complex visual interface. There has been growing interest in natural language interfaces (NLI's) for visualization systems and applications, both focusing on database queries (72; 73; 74; 75; 76; 77; 78) and for visualization applications (14; 15; 16; 17; 18; 19; 20; 21; 24; 25; 26; 27; 28; 29).

In this section, I will give a brief overview of recent work toward NLIs for visualization. This dissertation primarily contributes toward the design of novel interactions for data exploration, and it utilizes an NL approach. However, the primary contribution is not toward robust NL interpretation, or a contribution to natural language research areas. Our contribution is towards how NL might enable breadth in data exploration, and coupling NL and mid-air pointing gestures for large display interactions. Using the terminology presented in Srinivasan et al., this dissertation presents a **restricted NLI**, which responds to a small range of NL commands (25). In this section, I will present other recent work in NL techniques for data visualization, with a focus on the target application or problem area, to put the work of this dissertation into context.

Srinivasan et al. reviews NLIs for data visualization, and characterizes them by three dimensions: visualization capabilities, which allow users to specify visualizations, data focused capabilities, which allow users to pose questions about a dataset, and system control capabilities, which allow users to move windows among other features (48).

Articulate was an early tool that provided visualization capabilities, and it provided several alternate charts in response to user queries (14). Eviza and Evizeon focused on conversational requests, and on iterative refinement of a visualization's design (16; 79). DataTone presented a technique for resolving ambiguity in visualization queries (15). A number of other approaches combine database queries with NL and provide visualizations (80), or use recurrent neural networks to generate visualizations (81). NL4DV is a toolkit for generating visualization specifications based on user queries, and is designed to be adaptable for diverse application contexts.

However, it only accommodates requests for new visualizations, not requests to modify or reference a prior visualization (29).

There are a number of commercial tools that integrate NLIs, such as Power BI, IBM Watson and Ask Data feature in Tableau (82). Arklang is used by Ask Data to describe NL queries in a structured format, which can be understood by VizQL for Tableau (24; 83).

InChorus focuses on multi-modal pen and touch and speech interactions. InChorus also presents an argument in favor of creating restricted NLIs, in which the focus of the research is on interface or interaction design, rather than complex parsing of NL queries. They found that users were able to use a restricted NLI, which responded to a set of specific NL query formats, leaving as future work the possibility of building more complete NLIs for their novel multi-modal interaction technique (25).

This dissertation adopts the approach of InChorus, to focus on novel challenges posed by NLIs for visualization problems by using a restricted NLI command interface. This approach allows us to explore a specific interesting issue, with the development of a more robust NL interpretation technique as future work.

I have collaborated to work in the area of NL for data visualization, in collaboration with Abhinav Kumar and Abeer Alsairi, with Andrew Johnson, Jason Leigh, and Barbara DiEugenio, and our has been presented in visualization and natural language processing venues (45; 84; 20; 21; 23; 22).

2.3 Externalizing Data Exploration

One approach to enabling data exploration addresses cognitive challenges in data exploration, by focusing on externalizing the exploration process.

Visual exploration can be cognitively intensive. It is an iterative process, involving multiple rounds of actions, toggling between view construction activities and observation and insight generation activities (9; 10; 11). One particular challenge is memory. An analyst needs to be able to recall previous findings, to make decisions on what to explore next. They need to be able to return to past visualization states, and iterate on them (85; 30). They need to be able perform tasks that span more than a single view- such as comparing or correlating results (86; 39).

To address this challenge, a variety of interaction techniques have been developed in recent years that focus on a particular environment for displaying visualizations histories- a flexible canvas environment. This environment often consists of a virtual canvases with pan and zoom interaction. In other cases, it consists in a large display with abundant space for showing many views of data at once (38). We will discuss the particular features of large display cases in the section on large displays and data exploration, and focus here on flexible canvases more generally.

Broadly, virtual canvas tools aim to enable users to generate multiple views of their data, and position these views freely. The goal is to allow users to view their data from different perspectives and arrange views to reflect their sensemaking process. These environments are motivated by the finding in Andrews et al., where analysts performing sensemaking tasks over large volumes of text documents within a large display environment used 'space to think', by offloading conceptual relationships between analysis artifacts onto the display through spatial positioning (35). The large, flexible display area allowed the analysts to flexibly express conceptual schema. For instance, analysts might cluster groups of related documents, or construct timelines.

In this section, I will describe several prominent flexible canvas environments, both ones designed for virtual canvases and ones designed for large display environments. I will describe how views are created and utilized to create more views, as well as design decisions about view composition, management and layout.

2.3.1 Systems to support data exploration

First, I will describe the contributions of Javed et al, in Explates (30). ExPlates targets exploratory data analysis. The authors note that exploratory data analysis can involve high cognitive load activities, including use of memory and reasoning. The tool aims to externalize and spatialize the EDA process, and help analysts with recall and reflection.

Referring to the Card et al. pipeline (3), they define views as a tuple of D,M,V, where D represents data transformation, M represents visual mappings, and V represents view transformations (eg. Navigation). Each data transformation, which they term a mutating interaction, produces a new view of the data, because this stores a history of exploratory steps where the representation or the underlying data changes.

Components of their technique also reference a 'data flow' style interface. They show two kinds of plates- data plates, which depict data transformations, and visualization plates, which show the views. They use the technique of showing data wires and data anchors to visually depict the flow from one view to another. Data plates include database actions, as well as operations like filter, join, intersect, sort. Visualization plates include major view types (scatterplot, line graph, pie chart, bar chart, tag cloud, world map, US map). New plates are created either by selecting data or visualization plates from a menu, or by mutating operations within an existing view.

They provide automatic layout, using a grid-based layout algorithm. The algorithm focuses on showing the history as a branching tree, with space for the wires and anchors. In the case of mutating operations, views are positioned next to the parent view. In the case of new views, the user can position them or the algorithm finds a free space. Users can re-position views, through dragging interactions, and views the intersect are shifted aside and the algorithm works to keep the layout intact.

We utilize the idea of presenting visualizations where the data state has changed as separate views. We differ in that we focus on actions to modify the data state in multiple views. We also focus on an NL approach to this task, and a large display environment, which produces a large volume of views than is seen in ExPlates.

There are several other prominent data exploration systems that utilize the flexible canvas metaphor in support of data exploration.

In Gratzl et al, they present Domino (31). Domino consists in a visualization technique for exploring related subsets in a multivariate dataset. The focus is on enabling users to examine and manipulate subsets of interest across multiple views, with metavisualization techniques to connect subsets across multiple, related views. Specifically, the application shows subsets of the data, data associated with those subsets, and relationships between subsets. Domino uses blocks, or rectangular regions, to represent subsets of the data. These blocks can be related using different techniques- including parallel coordinates or sets. Users can interact with items or interact with blocks. This thesis does not target between-view relations at the level of granularity shown in Domino. In addition, while Domino is in a flexible canvas environment, the authors do not focus on flexible canvas interaction, view organization, or view scalability.

Zgraggen et al. present PanoramicData (32), which uses a boundless canvas, basic views and connections that are boolean expressions. PanoramicData presents to the user a scheme viewer, which contains the attributes for the user to select from. They drag and drop attributes onto the display, and each are depicted in a separate view. For instance, a user may select marital status and salary less than 50k, and each will appear in a view. These two attributes can then be connected, and one chart can be filtered on the other when the user makes a selection. Users can also drop selections onto existing views, and there are two drop targets for attributes: the group target, which allows users to specify what is summed over, and the color target which specifies how entities ought to be colored. Handwritten tags can be added, with notes to be returned to later.

The PanoramicData UI uses pen and touch gestures for interactions. The authors argue that this gives the user expressive power and reduces the cognitive load in interacting. PanoramicData has a 2d canvas with pan and zoom. The canvas is unbounded and elements can be freely, and manually, positioned.

Authors do not explicitly mention scalability, but as with other tools that feature an unbounded canvas with manual positioning, as the number of views grows the effort to find specific views and manage new content will also likely grow.

Bavoli et al, present VisTrails (87). VisTrails focuses on presenting a data flow approach, where multiple stages of data transformation are shown as a trail. VisTrails allows users to specify multiple view endpoints for a single trail.

Van den Elzen and Van Wijk focus on exploratory transitions, but in a specific context-transitioning from large overviews, to small multiples. They pursue this work in a 'trail-like' environment, which presents the exploration history. They provide interactions that make it easy to go from visual overviews to small sets,, in a trail that preserves the exploration context (34). Our work has similar methods to expand out a visual overview, through a specific category of referential action, however, we focus on a large, flexible canvas environment, where views can be freely positioned, and not in a layout that emphasizes visualization provenance. We also pursue other interaction types, that copy and pivot a target visualization.

2.3.2 Externalizing exploration and this dissertation

This dissertation shares a focus on presenting multiple views, generated over the course of data exploration, as an approach to externalize the exploratory process, and enable tasks that span more than one view of data. I pull from this body of work the notion of 'copying and pivoting', as a form of incremental exploration, that is presented visually to the user (30; 31; 32; 33). I also pull from this body of work an understanding of the distinction between mutating actions- that alter the underlying data values and attributes presented in a view- from other kinds of interactions, that alter the encodings and layouts within a view (30).

However, **unlike the techniques described above, this dissertation targets interactions that create and spatially organize many views at once, in support of multi-view analysis tasks. This means that we adapt copying and pivoting approaches to a multi-view context.** In addition, this dissertation focuses both on enabling incremental steps, which can be understood through a sequence of views, but stresses the need for conceptual groupings of visualizations, based on common features.

2.4 Large Displays and Visualizations

Large display environments have been found to be beneficial in a variety of contexts relevant to visual data exploration (42). They enable users to leverage movement and embodied cognition (88; 89; 37), for improved memory in data intensive tasks. With encodings that are perceptually scalable, research suggests that users can perform visual queries over both large datasets and many related views(43; 90; 91). In response to these findings, applications have been designed for large visualization environments viewing at different distances from the display (92), collaboration (93; 94; 95; 96), presentation of large volumes of data (97; 98), performing complex tasks (99; 98; 40; 96) and integration of 2d and 3d views (47; 95; 100). This dissertation focuses on benefits for sensemaking and integrative insights across many views of data, and considers interactions that will support visual data exploration on the display.

There are two issues that I focus on in this dissertation with respect to large displays and visualization. The first concerns using abundant display space to support sensemaking (35) and data exploration that spans more than one view of data, which includes what we know about how people organize content on a large display and how to support these actions through metavisualization (38; 39). The second concerns overcoming interaction challenges with large displays, with a particular focus on post-WIMP interactions.

2.5 Large Displays, Many Views of Data

In 'Space to Think' Andrews et al. describe how analysts use abundant display space to offload their sensemaking process onto the display, by physically grouping conceptually meaningful analysis artifacts together on the display. However, this insight- that spatial organization on large displays can support sensemaking in the analysis of complex data- has been under-investigated with respect to information visualizations, rather than text documents (35).

Knudsen et al. extended this idea to look at visualizations and data analysis, focusing on how analysts used abundant display space to support their tasks. The authors of this work used whiteboards to elicit interactions, particularly mid-air gestures, and noted that participants manipulated views to facilitate multi-view comparisons. They note that users wanted to spread out a set of views, to compare them, often ordering these views or composing these views into coherent sets, so that they could efficiently reason over them (38). Our findings in our pre-design evaluation, and in our evaluate of Ditto, echo these observations. In addition, our approach to providing coherent sets of views to users is consistent with the observed action the Knudsen et al. study- that abundant display space can be used to segment, and then visually juxtapose many related views of data.

Chung et al. discuss considerations in large display environments and describe the value of juxtaposing many heterogeneous views of data. Their scheme of describing the kinds of coordinated data views mirrors the scheme to be developed in this thesis. They discuss juxtaposing views to **compare**, **complement** to see different aspect of a dataset (similar to the browse category described in this proposal), **split** which decomposes visualizations into multiple views, and two categories of hierarchical views: focus+context and overview+detail. They also discuss the need for use of space to convey relationships. However, these design considerations are applied to display ecology scenarios, and not in service of a specific final application (101).

Knudsen and Carpendale explore the topic of abundant display spaces and visualization, noting that multiple views help us grapple with increasingly large dataset, because they allow analysts to segment and group data into accessible pieces, breaking down information into meaningful chunks. Then, people can compare and reason about their data, by considering different sets of data variations in multiple views. They note that large displays have the potential to enable these kinds of tasks, with views showing diverse data value and attribute selections. In particular, large displays have sufficient space and resolution to allow users to spatially organize information, forming implicit **metavisualizations**- or visualizations of visualizations. One of the challenges indicated by this paper is how to use both abundant display space and human spatial organization capabilities to enable interaction in visual analytics. They describe the need for *'formalisms' to develop a better conceptual understanding of the new design space of large displays* (39).

Metavisualization is explored further by Knudsen et al. in an exploratory study, to capture design ideas for visually representing between view relations. This work focuses on pair-wise between view relations (102).

This dissertation aims to contribute toward the effort to develop a better conceptual understanding of the new design space for large displays, both in proposing a technique for providing coherent multi-view responses to data-centered queries, and by using spatial positioning to enable efficient view creation actions in data exploration. We also contribute formalisms for multiple views, through our description of view collections, which are sets of views with coherent between-view relationships that are generated by our technique, both for natural language and for speech with mid-air gestures.

2.6 Large Displays and Interaction

Interaction with large displays presents challenges, and standard approaches, such as interaction through mouse and keyboard inputs, prevent users from fully leveraging the cognitive and perceptual benefits of large displays, described above (13; 41; 49).

David Norman described one of the primary challenges in designing computer systems as reducing the Gulf of Execution, or the barriers that prevent users from executing actions through a computer interface (103). Reducing this Gulf of Execution in the context of visual data exploration for large displays entails interaction design that brings users to their goals more efficiently and directly, by making it easier to create views of data on the display.

There has been considerable interest in 'Natural User interfaces' (NUI), that capture and respond to so called natural user behaviors, such as physical movements or gestures on multitouch displays (49). Realizing these interfaces has been particularly enticing for large and immersive contexts, where users are physically present within a digital space such that they are interacting with objects on the display at human scale (42; 41).

Knudsen et al. describe capturing some of these desired interactions, that are embodied and realized through gestures, in their whiteboard study, noting that participants would gesture to indicate suggested layouts or ways to manipulate visualizations on the display (38).

There has been research examining interaction in large display environments for visualization through multiple devices (104; 105; 95; 100; 94; 106), through touch and direct manipulation (107; 33; 108) , and through proxemic interactions (92; 66; 36). Many of these approaches for large displays focus on interaction that modifies the presentation of information within a view, such as zooming or selecting new encodings or layouts.

Several papers consider touch based approaches to generate visualizations for a large display, such as VisWall and PADE, which use a drag-and-drop style interface to create visualizations, and direct interaction with visualizations to create new ones (33; 107).

We depart from this line of work in several respects. First, to my knowledge, this is the first contribution of speech and mid-air gestures for creating visualizations on large display environments. In the HCI community, there is early work in the 1980s toward multi-modal speech and mid-air gesture interactions with a large display in Bolt's work 'Put that There' and in Hauptmann's "Speech and gestures for graphic image manipulation" (46; 109) In recent years, researchers have explored this further (110), including for interaction with a smart TV (111). However, the unique affordances of combined speech and mid-air gestures have not previously been considered for visual data exploration on a large display. Given the interest in Natural User Interfaces, and growing interest in large displays for data exploration, our research stands to contribute to several important research directions in visualization and human computer interaction.

2.7 Visualization Recommendations

Given challenges in creating views, there are systems that aim to provide visualization suggestions using mixed-initiative approaches. Mixed Initiative systems aim to reduce work for users by taking initiative and automatically performing tasks that are difficult for humans to complete (112). Although this dissertation does not directly contribute to visualization recommendation research, it does have similar motivations- addressing challenges in data exploration- and use similar models, which seek to understanding data exploration transitions.

2.7.1 Breadth oriented data exploration

Wongsuphasawat et al. present Voyager (6), which targets exploratory visualization and used mixed-initiative approaches which allow users to explore data without manually specifying views. Rather than transform raw data into tables, and map the table to visual elements or templates in specifying views, users select from an interface which displays a schema of the data, as well as transformations of the data, and then this is used by a visualization recommendation engine to automatically generate views. Voyager focuses on breadth-oriented exploration and prioritizes data variation, by supplying views with diverse data selections, over design variation, which would prioritize alternate representations of a fixed data and attribute selection. The system uses Vega-Lite (56) which specifies views using a grammar based in

Grammar of Graphics (113) and VizQL (83). Users can bookmark views that they found useful, and these are stored in a separate area for later access. Voyager2 builds on this work, by offering options for partial specification, where gaps are filled in with wildcards.

This dissertation shares a focus on breadth oriented exploration, by supplying many views of data that feature data and attribute variations. We utilize the terminology developed in this paper- breadth-oriented exploration and multiple views with data variations- in developing our design goals.

2.7.2 Other recommendation approaches

There are a number of visualization recommendation systems, some of which focus on providing users with a broad range of visualizations suggestions, that present diverse selections of data values and attributes, and also incorporate effectiveness criteria, to constrain the space of possible visualizations.

Mackinlay et al. designed APT (114), which presents a large space of potential views, by first enumerating many visual encoding options and then winnowing down the suggestions using a set of effectiveness criteria from Bertiin et al (115). In Tableau, "Show me", adopted similar features, by providing design suggestions (116). In a similar vein, Sage is a visualization recommendation system that uses a taxonomy of data properties to recommend visualizations (117). Other work in automatic design of visualizations includes (118; 119; 120; 121; 122; 123; 122).

Approaches that automatically create sets of views on behalf of the user include Design Galleries (124), which give design variations based on a starting point supplied by the user, essentially providing alternate views. VizDeck, provides a gallery of recommended charts, but focuses on statistical properties, rather than design variations (125).

These approaches share a focus on providing multiple views to users, in support of data exploration. We add a focus on natural language interaction, translating user interests into sets of views, and follow-up referential interactions, to extend the exploration focus.

CHAPTER 3

MANY AT ONCE: CAPTURING INTENTIONS TO CREATE AND USE MANY VIEWS AT ONCE IN LARGE DISPLAY ENVIRONMENTS

This chapter presents a study of visual data exploration in a large display environment, where participants expressed their exploratory intentions through speech and mid-air gestures. The analysis of this study informs the subsequent development of the multi-view data exploration technique discussed in Chapters 4 and 5 .

This chapter is published in Computer Graphics Forum 2020 (44), and was presented at EuroVis 2020. Coauthors of the work include Abeer Alsaiari (AA), Abhinav Kumar (AK), Barbara Di Eugenio (BDE) and Andrew Johnson (AJ). The contributions from each author are: AK and I designed and conducted the study, with extensive input from BDE and AJ. AK and I reviewed the videos and developed an initial coding scheme. I refined and expanded this coding scheme, in consultation with AA and AK. I wrote the text, and created the images, with input from AA and AJ. I am the first author of this publication.

3.1 Introduction

Large, multi-view environments present a variety of benefits in visual data exploration (42), particularly in contexts where users of the environment wish to juxtapose and arrange many views of data (35; 126), and generate integrative insights across these views (47). However, interaction in these environments remains an area of active research (127).

In this Chapter, I will describe results from an observational, exploratory study of visual data exploration in large, multi-view environments, using an approach similar to Grammel et al. (12), where participants express their intentions to a remote mediator who responds on their behalf. This approach allowed us to study the intentions of the participants independent of any particular view generation paradigm or graphical interface, giving us access to what users of a large, multi-view environment would like to do when unconstrained by the design choices of existing tools.

Existing large display or virtual canvas environments for visualization (eg. (128; 31; 30; 87; 32; 28; 107; 33)) typically allow users to explore their data by producing one view at a time, either through drag-and-drop operations through a menu, through actions on elements within a single view, through trails of copied and pivoted single views, or through data-flow diagrams.

In contrast, we observed that when participants expressed their intentions without constraint, they frequently posed requests for many views of data, by asking for many subsets of the data and many data attributes at once. We term this **'casting a net'**. These requests were accomplished both through direct queries and by utilizing prior views that were persistently displayed on the canvas. When using existing views, participants frequently posed requests to **copy and pivot these view**s, but they often did so in ways that **'scaled-up' their intentions, expressing multiple, parallel copy+pivot actions to perform on a single view target, or by collectively copying and pivoting whole sets of views in one command**.

These 'cast-a-net' requests enabled participants to **efficiently produce sets of views with conserved features- or features in common across the visualizations- which**

utilized the display space and allowed them to perform tasks that spanned many views.

In this Chapter, I will present a detailed description of how participants efficiently expressed intentions to 'cast a net' to target many subsets of the data and data attributes. This includes use collective and parallel actions on prior views on the display. We contribute a description of how these actions facilitated data exploration and discuss the design implications for multi-view environments.

3.2 Background

3.2.1 Large Displays

Recent research suggests a variety of potential benefits for information visualization in large display environments. One of the early findings about large display surfaces is that, when provided with "space to think", analysts use large displays to organize analysis artifacts, encoding conceptual relationships by positioning related text documents together in space (35). Large displays also enable users to leverage movement and embodied cognition (88; 89) for improved memory in data intensive tasks. When perceptually scalable encodings are applied to data attributes, there is evidence to suggest that users can perform visual queries over large volumes of data, and over many related views of data (43; 90; 91). Finally, given the ability to display more related views of data (129), users appear to formulate integrative hypothesis that make use of these views(40). In response to these findings, applications have been designed for large visualization environments targeting hybrid display of information (92), collaboration (93), presentation of large volumes of data (97), and integration of 2d and 3d views (47; 95; 100).

Interaction with visualizations on large displays is understood to present challenges and opportunities (42; 127). Recent work has examined movement or proxemics as an input to visualization environments (66; 130), as well as multi-touch (108) and an ecology of devices through which users interact with the large, shared display (105; 93; 106; 94).

Our work contributes to this body of work by examining use of a large display for a real visual data exploration scenario, but we capture intentions for views independent of a realized interface. Some of our findings echo Knudsen et al. (38), where a whiteboard workshop captured interactions over many visual artifacts on large display surfaces. Our work complements this analysis by observing similar tasks that spanned many views and utilized large display areas.

3.2.2 View Construction and Multi-View Environments

A variety of flexible canvas environments have been created for information visualization, including virtual canvases with pan and zoom interaction. Broadly, these tools aim to enable users to freely generate views of their data and position these views freely, and aim to allow users to view their data from different perspectives, (eg. (128; 31; 30; 87; 32; 28)). View creation in these environments have been explored using a variety of interaction techniques. Initial views are often added to the flexible canvas through interaction with a menu, such as through drag and drop operations onto the canvas. Alternatively natural language queries can create views, in systems such as FlowSense, which feature a NLP interface to the data flow model, where data is selected and transformed through a set of views (28).

Many of these systems also present ways to create new views through actions on existing views. One approach is to allow participants to copy and pivot a view target, or to create new views from selections within an existing view, to drill down into more focused portions of a dataset. These actions can facilitate the creation of visualization provenance trails, and aim to enable backtracking and revisions along the trail. (128; 87)). The complexity of multi-view environments has prompted exploration of multiple coordinate view techniques, which are summarized in Tobias et al. (64). Given many views, there is also interest in capturing and visually representing the relationships between views through metavisualization (39; 102).

Our work contributes to this line of research by observing how participants create views and use existing views as tools or reference points for further view creation.

Grammel et al. explored how “InfoVis Novices” construct visualizations, by asking them to create views in the absence of a graphical interface, using a remote moderator. View construction was found to pose challenges and be error prone. Vis novices struggled to translate questions into appropriate visualizations, including template selection, choosing mappings, layouts and encodings. The take-away from this study is that even with robust graphical interfaces, which remove the need to learn coding or scripting for visualization creation, users may still make visualization construction errors. (12).

3.3 Methodology

In this section we present the design decisions in our evaluation and how these designs allowed us to address our research goals. We also discuss limitations and how we address these limitations in our analysis.

Our research goal was to observe visual data exploration in a large, wall-sized display environment to derive design goals for future systems that are grounded in how participants request new views, utilize and reference existing views on the display and utilize the display space in support of data exploration tasks.

Our research questions are :

1. In a multi-view environment how did participants request views?
2. How did participants use existing views to pose subsequent requests for new views?
3. How did the display space support analysis tasks that involved more than one view?

To address these questions, we had three broad goals in our study design: 1) **realism**: capture interactive intentions expressed in response to real visualizations of data within a realistic data exploration scenario in a large display environment; 2) **unrestricted expression of intentions**: capture interactive intentions independent of existing interfaces or interaction modalities, in effect to capture what participants wanted to do when reasonably unconstrained; 3) **multiple rounds of view generation**: examine these intentions over a complete analysis session, with many rounds of visualization generation, in support of completing a realistic data exploration task.

To meet these design goals we conducted an observational exploratory study in a laboratory setting, using a protocol that mirrored Grammel et al. (12). Recruited participants were given a data exploration task, and told to verbally express their intentions to a remote mediator, (a PhD student in data visualization), who was located in an adjacent room monitoring spoken and gestural communication from the participant over video and audio feeds. By locating the mediator in a different room, we distanced the participant from the interface used to generate new views. This allowed us to examine their behavior in an interface-agnostic setting. Like Grammel et al. participants were informed that the remote mediator was a person, and we do not simulate a system as in a Wizard-of-oz study, as is used in other studies of interaction modalities in InfoVis, [eg. (131)].

Unlike Grammel et al., we use a large, multi-view flexible canvas environment to persistently display prior responses, allowing us to look at how participants used these past views and the large display. In addition, we did not ask participants to specify an intended single view, but rather to ask anything that might aid in the exploration and analysis of the data. Study of abundant display space mirrors Knudsen et al, but we focus on data exploration tasks and use a digital environment with real views of data and many cycles of view construction. (38).

3.3.1 Piloting

To arrive at our final study design, we conducted pilots in two phases. In the first piloting phase, we performed an offline pilot with four remote subjects, who were presented with a document summarizing the data variables and a data analysis task, and had the opportunity to pose analysis or clarification questions over a two week period via email. This enabled us to refine the materials and add focused data exploration sub-tasks. Then, we conducted a pilot study with five participants in a laboratory environment. We refined our approach to responding to participant queries, particularly our approach to managing new visualizations as they were added to the display. We refined the experimental setup by shifting the cameras to ensure a better view of gestures.

3.3.2 Participants

14 participants (7 male and 7 female, ages 18 to 34), were recruited for the study, with an additional 4 participants in the first stage pilot and 5 in the second stage pilot. The participants were drawn from diverse departments and fields including computer science, communications, business, speech-pathology education, biology and medicine. Participants had varied experience with visualization and data analysis, ranging from daily data analysis tasks (close to 50 percent of participants), to almost never conducting data analysis (20 percent of participants). All participants were familiar with common data visualization types and used computers daily. A few participants had used the large display environment for class or meetings, but they had not used it for data exploration.

Given this diversity, we do not draw conclusions that are specific to any particular background or level of expertise. Domain experts or novice analysis could be an area of focus in future work. However, since all participants were either students (10 participants) or professionals in data driven fields (4 participants), this group is appropriate to target for future realized systems.

3.3.3 Apparatus, Environment and Materials

We performed our study in a laboratory setting, allowing us to control the interface and environment, as well as manage the communication channels between participant and remote mediator.

We opted to perform this study in a digital context, as opposed to a whiteboard to capture multiple, continuous rounds of interaction, with visualization responses that were generated

Participant Data			
Participant	*Requests*	*Visualizations*	*Ratio*
#1	11	36	3.27
#2	11	18	1.63
#3	13	23	1.77
#4	13	28	2.15
#5	21	34	1.62
#6	11	25	2.27
#7	23	30	1.30
#8	16	32	2.00
#9	9	35	3.89
#10	32	34	1.06
#11	13	31	2.38
#12	23	35	1.52
#13	24	33	1.375
#14	22	32	1.45

TABLE I: Participant data: number, request count, number of views produced, view:request ratio.

from real data. The mediator created visualization responses to participant requests using Tableau and presented these visualizations to the users through SAGE2, a flexible canvas tiled-display wall collaborative system (93).

The environment for the participants consisted in a large display wall (6.675 by 2.01 meters and 6144 by 2304 pixels) shown in Figure 1. Participants could refer to onscreen textual descriptions of 1) the data, including attributes and their values, and 2) the overall goal and sub-tasks. Paper copies of the task description and data description were removed, so the participant directed their attention to the display and gestured freely. On the top of the display we created a status bar, with an animation indicating when the remote mediator was working on responding to the request, and a chat box, in which the remote mediator could enter messages. This is depicted in Figure 1.

The remote mediator was isolated from the participant in a nearby room. As in Grammel et al. (12), this allowed us to shield the participant from the interface used to generate new views, avoiding biasing effects and removing the influence of verbal or non-verbal feedback from the remote mediator.

An in-room aide, a graduate student in computer science, was present in the room with the participant to explain the study protocol, address technical questions during the study and conduct the final interview. The remote mediator was not introduced to the participant until the study was complete. We chose to shield the participant from the mediator to encourage direct and honest feedback during the final interview.

The remote mediator was provided with two video streams, showing the participant from the front and from behind, to capture pointing gestures and gaze, as well as facial expressions. The remote mediator viewed two 4k displays that mirrored the participants large display, enabling the remote mediator to ensure optimal placement and sizing of the provided views.

The remote mediator generated visualizations using Tableau on a laptop, and dropped exported static images of these views onto the large display using a collaborative tiled-display wall middleware, SAGE2 (93), which also supplied a laptop interface for re-sizing and positioning these views freely. The remote mediator used a chat box, through a live-streamed text editor, and a status bar, to show when the remote mediator was producing new views in response to participant queries. This is depicted in Figure 1

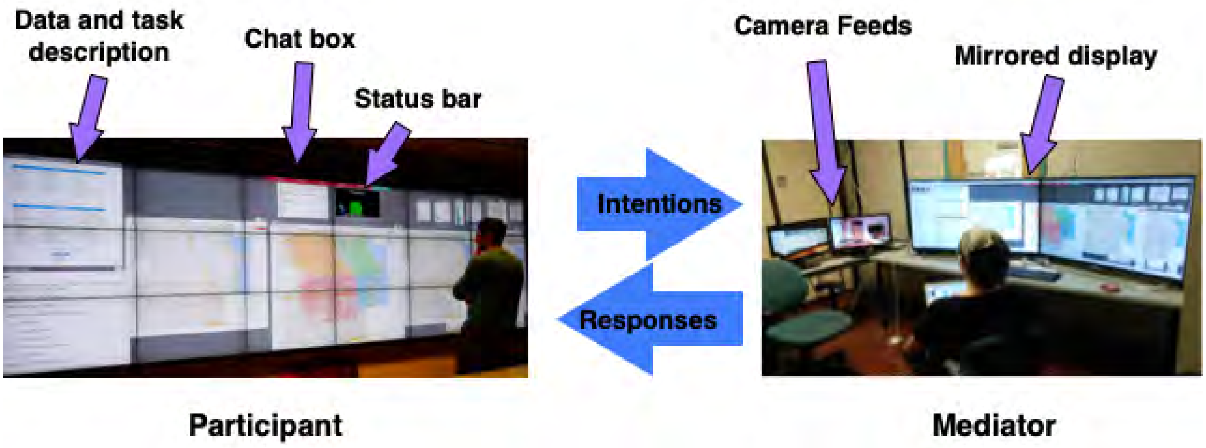


Figure 1: We conducted our study in a laboratory setting with the remote mediator in an adjacent room. The participant's environment consisted in the large display, with data and task description on the left side of the display, and a status bar and chat box at the top. The remote mediator had access to two video feeds, and two 4k displays mirroring the large display content. The remote mediator used a laptop to generate views using Tableau, and dropped them to the user through Sage2.

3.3.4 Task

Their task was to explore crime data from a local, public data repository in order to decide how to deploy additional policing resources. We chose this task because we believed the data would be familiar to participants and that they would be motivated to explore this data out of personal interest. Each crime incident in the dataset included a **GPS coordinate**; a **neighborhood identifier** (from one of four local neighborhoods); a date and time, which were used to infer the **time of day**, **day of the week**, **month of the year**, and **year** when the crime took place; a classification of the primary **crime type** (eg. theft, burglary, assault...); as well the primary **location type** where the incident took place (eg. street, residence, business). This data is described in detail in Appendix A.

We supplied a list of general sub-tasks (eg. examine changes over time, look for hotspots), to provide direction and a starting point to the participants. The addition of subtasks was based on feedback from participants in the first phase pilot.

3.3.5 Procedure

During the instruction phase, the in-room aide provided the data description and analysis tasks to the participants and instructed them to 'ask anything that would aid in your analysis', and were given no restrictions in the kinds of queries they could pose. The aide explained the interface, pointed out the location of cameras, and demonstrated through a short social exchange that the remote mediator could respond to spoken and gestural requests. Participants were encouraged to think aloud and describe their findings as they inspected provided views.

We opted to not include a learning phase in our study. We opted to allow analysts to discover system capabilities during the session, rather than through a learning phase to give as much time as possible to a single analysis. We opted to not provide views at the start because we did not wish to direct the analysis in a particular direction and we wished to capture initial queries as well as follow-up queries. We did not provide visualization templates, because we wanted participants to pose questions freely rather than specify views directly, which has already been investigated by Grammel et al. (12). Participants began with a blank canvas and a data and task description.

The remote mediator responded with visualizations where possible, including situations that could be answered with textual responses (eg. *"which crime occurred the most often?"*). When participants posed a request, the mediator generated one or several views in response to their request using Tableau. If a request produced a multi-view response, all views were presented at once. Provided visualizations were numbered by request and given a title that communicated the subset of the data contained in the view (eg. *Theft and Battery*), and the visualized data attributes (eg. *Time of the Day*). All visualizations were saved and used in our analysis.

In situations where the expected outcome to a request was unclear, the remote mediator made an appropriate guess rather than asking extensive follow-up questions. We opted for this response style because we did not want participants to feel that they needed to precisely specify views. We wanted them to pose requests freely. Participants were instructed to correct the mediator if the responses were not what they wanted. The mediator would select appropriate templates for any spoken data attributes, would filter based on selected subsets of the data. Colors and scales were generally the defaults supplied by Tableau.

The visualizations presented to the participant were static images exported from Tableau, not interactive views. The benefit of this approach was that it allowed us to focus on view generation actions in a large display context, and bracket the challenge of view modification and multiple coordinated views, which would have introduced a large range of design choices to our study (132). The view coordination problem could be addressed in future work. We made the decision to respond to all requests for new subsets of the data, new data attributes or new visual templates with new views. In contrast, when participants wanted to modify the encodings, scales or layouts within a view (eg. adding labels, changing color schemes), we treated these as view modification requests and the old static image with a new static image reflecting the requested change, and we distinguished between these request types in our analysis.

The remote mediator had control over view positioning. We opted to position the views automatically for the user for several reasons. First, we learned during the second phase of piloting that participants struggled to interpret a set of views and make decisions about where to position them, and that verbal positioning instructions were time consuming. Second, views generated during the pilot study very quickly filled the display which made it challenging for participants to pose new requests. We could have provided a secondary device or interaction modality for view positioning, but this would have tethered the participants to a device and we wished to encourage interaction through the mediator. A limitation of this choice is that we captured fewer view layout requests, and this could be an interesting direction for future work. The layout protocol involved deciding whether to move aside prior views and arranging the new views in the central region of the display, which we call the 'active' region.

Participants decided when to end the session, based on when they felt they addressed the initial data exploration task. As seen in table Table I, the mediator provided an average of 30 views to an average of 17 requests. Following the session, the participant took a computerized survey and completed an interview with the aide. The remote mediator would visibly exit the session before the participant began the survey by deleting the chat box and status bar, in order to encourage candid responses during the interview.

3.3.6 Analysis

We opted for qualitative analysis methods in order to capture rich behavior within a realistic scenario, and we used a grounded approach (133).

The recorded video was transcribed in full. We also used the stored and numbered static visualizations and chat transcripts from the sessions. A team of three researchers reviewed a subset of the participants transcript and video. This team met several times to discuss high-level themes. We note that 1) participants expressed their intention to generate visualizations either directly or by utilizing existing views on the display and 2) participants frequently generated many views that could be arranged into coherent group with relatively few interactions through the mediator. These themes informed the adopted coding approach.

A primary coder created a visual record of each participant's sessions. For each request, a visual 'scene' was created that depicted 1) snapshots from the video showing the state of the display before the request, 2) the transcript of the request, 3) snapshots from the video showing the participant's gestures to onscreen views, 4) the images of the views provided to the participant and 5) snapshots from the video showing the state of the display following the request. We also created scenes containing changes to the view layouts and participant think aloud. We adopted this approach because we needed to rapidly review of the transcript alongside the display state, the provided views and the participant's movement and actions. Over 550 scenes were compiled in total, with 23-64 per participant. We isolated from these scenes 215 scenes which contained requests for new visualizations. These visualization scenes were the primary unit of analysis.

The primary coder used an open coding approach to refine a set of codes to apply to the visualization scenes. Codes were developed through an iterative, multi-pass process. These codes were discussed with two coding reviewers. The coding reviewers posed questions and flagged ambiguous cases. After discussion, the codes were modified through several passes. This review and discussion process was repeated several times, until the codes were relatively stable and addressed the themes.

The final codes capture both how views on the display were related to participant requests, and how many data attributes and subsets of the data were requested, an approximation of how many tasks participants performed that spanned more than one view. We noted unusual features within what we term referential requests, and developed a set of codes specific to this request type.

3.4 Findings

In this section we describe our coded observations from participant visual data exploration sessions.

Our coding scheme is divided into three parts. The first part identifies the ways in which participants utilized or did not utilize existing views to express their intentions. Essentially, these codes identify *how* participants expressed their intentions. We identified three strategies - direct (41 percent), referential (42 percent), and selection (17 percent). This primary division helped us to isolate different strategies participants used to express complex intentions efficiently.

The second part of our coding scheme looked at the whether a participant's request targeted a single data attribute and a single subset of the data, which we term a **targeted** request that could be presented in one view, or whether the request **cast a net** around several subsets of the data and/or several data attributes within the information space. We divide cast-a-net requests into browse, compare and complex multifaceted. This coding gives us access to *what* multi-view, multi-subset, multi-data-attribute intentions participants requested through the mediator.

Note: For the purposes of this discussion, we define a subset of the data as a selected set of rows from a tabular data set, where the rows are selected based on a set of one or several filters. By data attribute we mean the columns of a tabular data set, such as the day of the week the crime occurred on, the crime type, the neighborhood of the crime.

These two code parts and their frequencies are summarized in Figure 2.

The third part is applied specifically to referential requests. In this part, we examined the number of views are that are targeted in a referential request and the number of actions that are specified on the target(s). This allowed us to concretize our observation that, when unconstrained, participants 'scaled-up' their intentions to create or operate on many views at once, to extend their exploration to data subsets and data attributes, and to perform tasks that spanned more than one view.

3.4.1 Direct, Referential and Selection Requests

We observed that participants were able to efficiently pose complex requests through the mediator. A significant way that they accomplished this was by using existing views, either as templates or for selection and drill-down. To isolate these requests, we looked at how

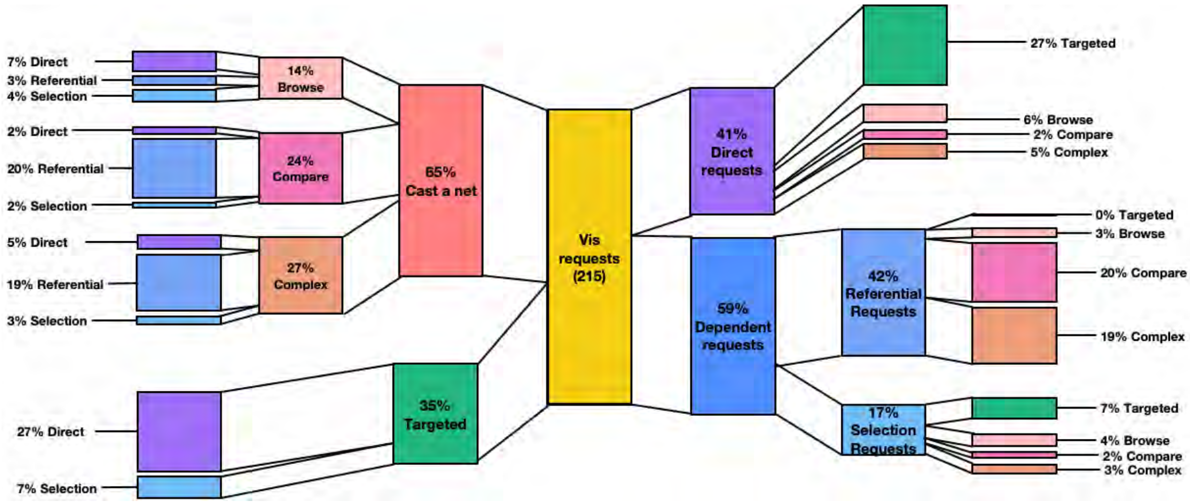


Figure 2: This figure presents an overview of the coded observations of how participants utilized existing views to pose their requests, and how many data attributes and subsets of the data they requested in one command.

participants referenced or utilized ‘active views’, or views in the center of the display, in formulating their request, and any verbal or gestural indication toward these views. In cases where participants referenced view targets in posing their requests, we labeled these as **dependent requests**, because they relied in some way on existing views. These constituted 59 percent of visualization requests. The remainder we termed **direct requests**. In these requests the participant specified the intended view with no reference to existing views and they represented 41 percent of view requests.

Of dependent view requests, most were labeled as referential. Referential view requests came in the form of ”Can I see this, but...”, where participants indicated a view target or targets, using speech and/or gestures, expressed an action or actions, to perform on the target(s) which resulted in the outcome, a new view or views on the display. In effect, participants specified a desire to copy and pivot the target or targets to a new portion of the "information space". Of the 215 visualization requests, 92 were referential requests, around 42 percent.

In the third major category, selection, participant requested a new view that focuses on a selected region from a target view. For instance, a participant might be looking at a breakdown of crimes by day of the week, and may then ask to see Friday crimes, selecting the 'Friday' subset of the data, by the neighborhood data attribute. These represented 17 percent of all visualization requests. The resulting trail of views would have a hierarchical relationship which includes a parent view that is selected from, and a child view which displays the selection.

Given the flexibility in participant requests, we noted 5 cases where participants both used selection and referential approaches in one command- such copying and pivoting a view to a filter selected from another view.

3.4.2 Target vs Cast a Net

In the second part of our coded observations, we distinguish requests that were **targeted** to a single data attribute (eg. day of the week) and a single subset of the data (eg. Thefts in 2014), from requests that **cast a net** over a number of data attributes and subsets of the data. We observed three major categories of these requests- browse, compare and complex multifaceted- based on the particular data subsets and attributes enumerated in the request. These codes captures whether participant explored by expressing a focused question to which a singular response could be provided, or whether they wished to look across many portions of the dataset at once. Most requests were cast-a-net requests, and we explore the implications of this in the discussion.

3.4.2.1 Targeted

A targeted request is one where the participant specified a single subset of the data and a single attribute of interest, which could be responded to within a single view. A targeted request might include asking for a map of thefts or frequencies of thefts by day of the week. Alternate views could be provided, such as using alternate visual templates, colors, scalings and other encodings. However, the participant expresses an intention to view one portion of the dataset with respect to one data attribute. Overall, 35 percent of requests were labeled as targeted.

3.4.2.2 Cast a net: Overview

The remaining 65 percent of requests overall were requests that spanned more than one subset of the data and/or more than one data attribute. These requests typically elicited either a set of views or one or several large multi-faceted views, where several aggregations of data are presented within one window, such as a divided bar chart or a multi-line chart. We classified 'cast a net' requests into 3 categories: compare, browse and complex multifaceted requests, depicted in Figure 3, Figure 4 and Figure 5.

All referential requests were coded as cast-a-net, even if one new view is produced from the request. We did this because the new view possessed a relationship to the target and the participant generally used the resulting pair or set of views to perform tasks that spanned more than one view.

Figure 3: In browse cast-a-net requests, a participant would indicate one subset of the data, in this graphic represented by the long blue bar, with respect to more than one data attribute, in the graphic represented by the gray square and diamond. Participants used these requests to focus on a subset of interest, and explore several data attributes.

3.4.2.3 Cast a net: Browse

In browse cast-a-net requests, participants expressed a single subset of the data that they wished to focus on (eg. Thefts on Saturday), but requested several data attributes within that subset. In effect, the participant expressed the intention to browse several attributes and views within a focused area. The resulting views would allow the participant to browse for trends, features and patterns within the subset of the data of interest. Cast-a-net browse requests constituted 14 percent of all visualization requests.

3.4.2.4 Cast a net: Compare

In compare cast-a-net requests, participants requested different subsets of the data with respect to a common data attribute. For example, a request to examine two neighborhoods by distributions of the types of crime, would be classified as a comparison request, because the

Figure 4: In compare cast-a-net requests, a participant would indicate one data attribute of interest, in this graphic represented by the gray square, with respect to more than one subset of the data, in the graphic represented by the long blue and purple bars. Participants used these requests to compare subsets with respect to a single data attribute.

participant specified one data attribute and several subsets of the data. Sensible responses to these requests include multiple views in separate windows or in multifaceted views within the same window, such as multi-line charts or grouped bar charts. These views allowed participants to compare distributions, trends, or spatial hotspots across multiple subsets of the data. Cast-a-net compare requests constituted 24 percent of all visualization requests.

3.4.2.5 Cast a net: Complex Multifaceted

In complex multifaceted requests, participants would express interest in several subsets of the data and several data variables. Responses to these requests would include multiple views with permutations of the subsets and variables of interest. At times, participants might request views that allowed them to simultaneously browse within several subsets of the data, and compare these subsets against a set of common data attributes, with each dimension presented

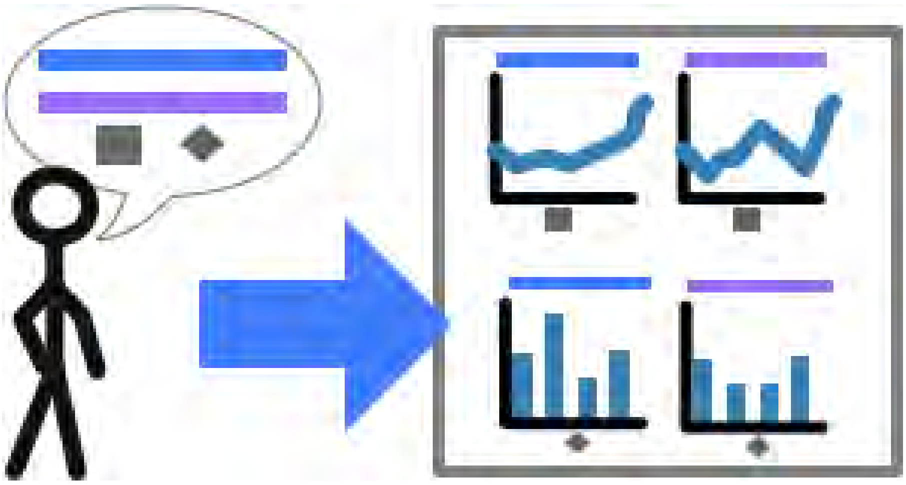


Figure 5: In complex cast-a-net requests, a participant would indicate more than one data attribute and/or subset of the data. In this graphic they requested two subsets of the data with respect to two attributes, and the responses were depicted in a grid. Participants used these requests to explore complex combinations of attributes and subsets.

in a grid. At other times, complex requests might warrant combinations of multifaceted views, to enable participants to facet the data in different ways. Cast-a-net complex multi-faceted requests constituted 27 percent of all visualization requests.

3.4.3 Creating Many Views with Referential Requests

Within referential requests, we captured the number of targets, actions and outcomes of the referential request. Of the 92 referential requests, the majority targeted a single view, expressed a single operation to perform on that view, producing a single outcome (37 requests). The remaining referential requests were coded as one-to-many (25 requests), many-to-one (6 requests), and many-to-many (21 requests). These requests enabled participants to efficiently express desires for complex sets of views.

3.4.3.1 One-to-One

In one-to-one referential requests, the participant indicated a single view and specified a single operation to perform on this view, which would produce a single outcome, in our case a new view on the display. For example, one participant pointed to a multi-line chart, displaying data by day of the week, and asked *"Can I have a look at this (pointing to the target) by month of the year?"*. By referencing this complex template, the participant was able to shift from one data attribute (day of the week) to another (month of the year), while still retaining the other components of the view. The conserved features serve to link these views together.

3.4.3.2 One-to-Many: Parallelized Copy+Pivot

One-to-many referential requests occurred where participants referenced a single view target, but expressed an intention to perform **multiple operations in parallel on this view**, producing a set of views unified by preserved features from the original template. For instance, in Figure 6, the participant pointed to a view showing thefts by time of the day, asked *"Give me the same of this (pointing) with battery, deceptive practices, criminal damage and assault, please"*. The mediator took the specified view, preserved the template and x-axis, and repeatedly changed the filter from theft to the enumerated crime types, producing a new view for each of the specified types. The participant then scanned the set of views and identified differences in the hourly distribute of battery crimes, when compared to the other crime types. Parallel actions of this kind are highly efficient. Rather than request each new view with a new filter, when unconstrained participants opted to bundle the actions together within a single request.

3.4.3.3 Many-to-Many: Collective Copy+Pivot

In 21 cases, we observed participants making referential visualization requests by indicating many view targets, through pointing or speech, and then expressing one or several operations to perform **collectively** on the indicated targets. We term these actions 'many-to-many', because many views were targeted by the participant, with the intention to produce many views and extend the reach of their exploration.

In one case, pictured in Figure 7, the participant points to two views, one of which shows theft by month and the other theft by day of the week. The participant asks *"Can I get these same charts but just for battery."* To respond to this request, the moderator pivoted the two views, producing two new views. When all the views were juxtaposed on the display, the participant then compared the number of battery and theft crimes by day of the week and month, and identified several differences.

In many of these cases, participants collectively operated on sets of views with conserved features. For instance, a set of views with a common set of filters could be pivoted to a new set of filters. A set of views with a common visual template could be pivoted to a new set of data attributes. We speculate that these commonalities served to signal to the participants that sets of views could be referred to collectively and acted on as unit.

3.4.4 How, What and How Many

Examining coded observations in combination, several interesting features emerge. While direct requests most frequently targeted single views, direct cast-a-net requests were around a third of all direct requests. Participants would pose direct browse requests either by asking for

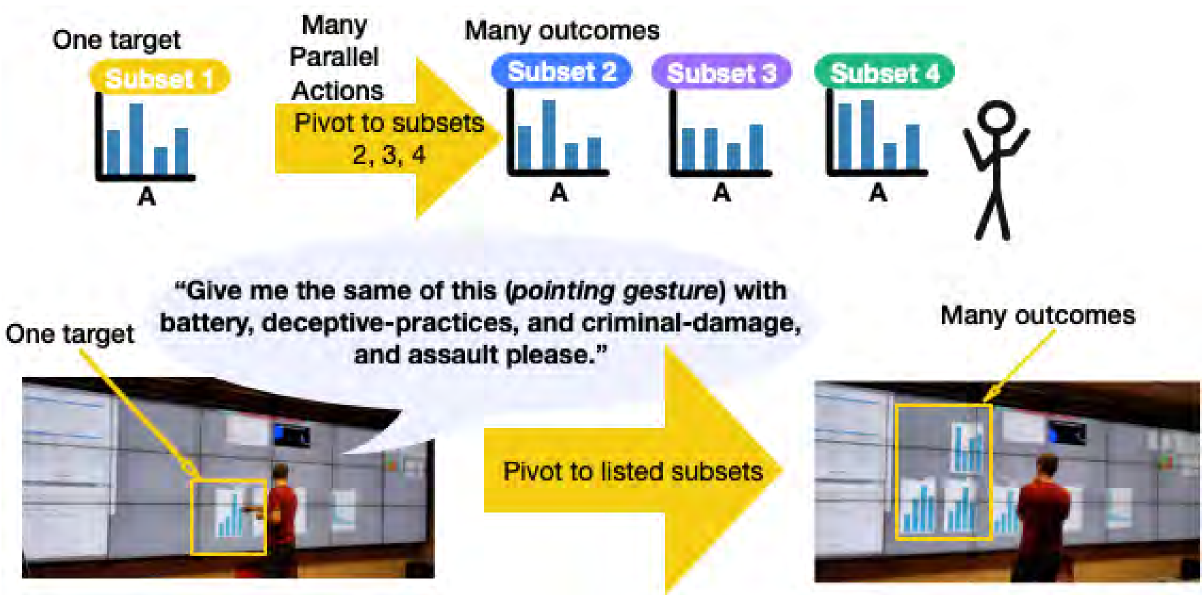


Figure 6: In this one to many referential request, a participant indicates a view target and expresses multiple operations to perform in parallel on the target, enabling a comparison task across the resulting views.

general information about their area of interest or by bundling several data attributes together, often using language that applied to several data attributes (eg. 'where crimes occur' included several data attributes) or by wanting to know when crimes occurred, and failing to specify temporal aggregation. Direct browse requests could be seen as related to underspecification of intentions or the high cognitive load of interaction in the absence of a visual interface, which came up during interviews with participants.

Comparison cast-a-net requests were most frequently accomplished via-reference to existing views. In these cases, participants might be examining a view and then would ask to pivot to a new filter. Essentially participants wanted to know if their observations extended to other

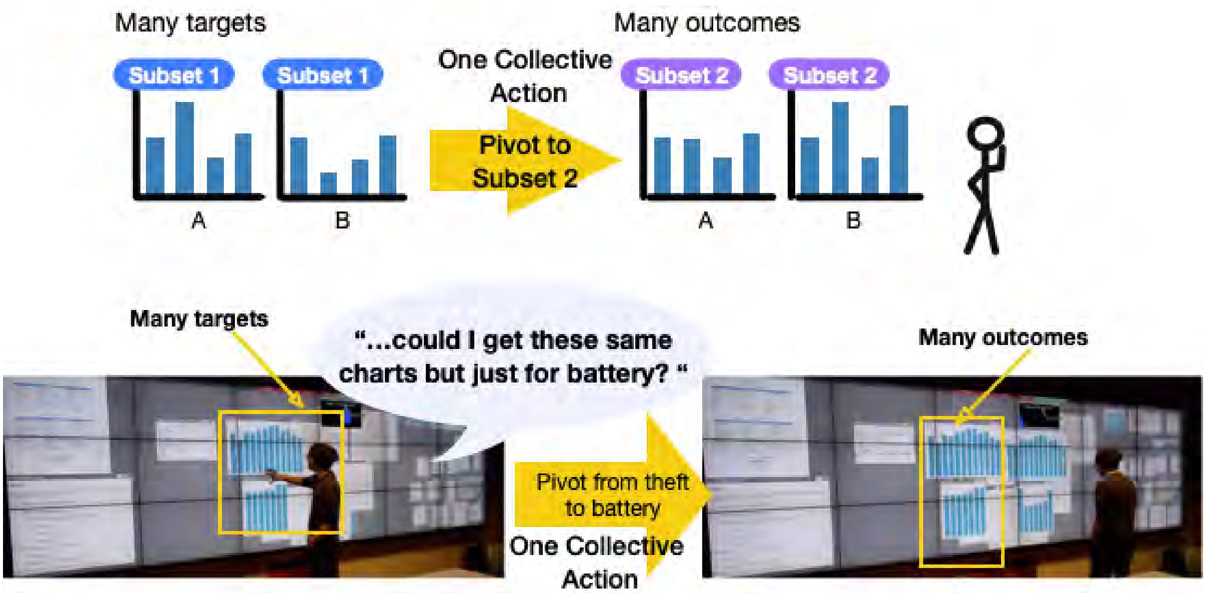


Figure 7: In this many to many referential request, a participants collectively duplicates a pair of views, and performs tasks on the resulting view grid.

subsets of the data. Complex referential requests frequently involved either faceting the target view, such as by subdividing the view by an additional data attribute, or by a collective many-to-many copy and pivot operation, which would produce a grid of views with conserved features in each dimension, allowing for smooth movement between two different multi-view tasks, one accomplished by scanning horizontally across views within the grid and the other by scanning vertically.

While many selection requests were targeted, requesting the selected subset of the data with respect to a single data attribute, other cases were more complex. In half of the selection requests, participants created a net around the selection, either requesting several data attributes for that selection (browse, 25 percent of selection requests), or faceting this selection with respect to several data attributes (complex, 17 percent of selection requests). In other cases, participants made several selections from the target view and compared these selections across a conserved attribute (12 percent of selection requests).

3.4.5 Cases

We observed that the cast-a-net requests and referential requests produced coherent sets of views and enabled a variety of analysis tasks that integrated information across many views. Several participants used repeated requests of these types to efficiently create many views in a few interactions with the mediator. The views could then be positioning in grids and clusters, to perform simultaneous multi-view analysis tasks, browsing, comparing, trend identification, and faceted exploration.

The first case we wished to highlight, involved a participant who used four queries to produce 29 visualizations. She began with a direct browsing request focusing on the neighborhood around the university, which resulted in 7 views focused on the university. This many-to-many referential operation was repeated, for two more neighborhoods resulting in a screen state with 28 views, seven for each neighborhood and four for each attribute, in just 3 requests. These views were presented in a grid that permitted her to perform a between-neighborhood comparison task across pairs of views and a within-neighborhood browsing task within several views showing different data attributes.

In the second case study that we wished to highlight, the participant made a series of referential requests, resulting in three multi-faceted views that showed a common subset of the data (crimes in 2014) and a common aggregation in a multifaceted bar chart by the four neighborhoods. Then in her final request, she targeted 3 views for a collective and parallel copy+pivot operation to produced 15 views, which collectively covered 4 data attributes within 5 years. Walking from left to right, and scanning vertically, she could smoothly move between trend analysis within a neighborhood, comparing trends across neighborhoods, as well as browsing for interesting patterns within a neighborhood and year. This case is pictured in Figure 8.

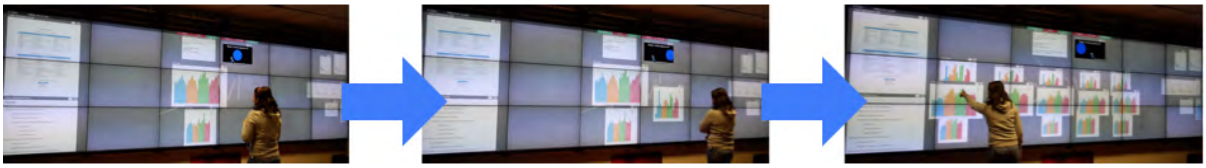


Figure 8: In three requests, a participant generates 15 views of their data. In the final request, the participants references a set of views and poses a complex request to collectively and repeatedly copy and pivot these views. The final result is a grid of views, which the participant used for an integrative data analysis task that spanned the views as a collection.

3.4.6 Interviews and Surveys

Our interview and survey with participants enabled us to examine how participants experienced the data exploration sessions, their subjective impressions of the quality of the visualizations and their response to exploring data in a large, multi-view, flexible canvas environment.

3.4.6.1 Overall impressions

Many participants directly commented that they they liked the experience. One participant stated *"The experience was amazing. Most of my queries were satisfied through the visual* *presentations. The data provided enough data for my understanding of the visualizations. The data analysis expert understood all my questions and I got a prompt visual response".*

In the survey the participants responded to a set of questions on a five-point Likert scale. All of the participants felt that the mediator always or usually understood their requests (50 percent for 'always' and 'usually'). 66 percent felt that the responses always helped them analyze the data and the remaining 33 percent felt that the responses usually helped them analyze the data. 93 percent of participants felt the responses met their expectations all (53 percent) or most (40 percent) of the time.

Participants noted that responses with multiple views were valuable. For example, one participant stated *"It was impressive to see the data and be able to compare & contrast it in many different ways. Each visual makes you consider a new aspect and/or want to inquire about new data to find new patterns."* . Another stated, *"The multiple responses were very helpful. Sometimes the additional responses helped answer a complex question, and could be used to compare more detailed responses to more general ones."* . In survey responses, 60 percent of participants preferred getting multiple responses, with the remaining either holding no opinion (33 percent) or preferring one response (7 percent).

3.4.6.2 Blank canvas challenges

Several participants described challenges related to knowing where to start, formulating requests verbally, and facing a blank canvas. One participant noted *"The cognitive load of like thinking about what I want to visualize and translating that is just more steps vs like, I want to look at that, click click click, doing it myself."*

In contrast, some participants appreciated the ability to offload tasks onto the mediator. One participant stated "*It is much more convenient to just say and get things done, rather than implementing your own. It lets you, at least in my case, I could completely focus on what I wanted to do, instead of 'do I click here, should I draft that?'. What am I trying to solve, that is all I focused on. I loved that part."*

Another noted that the process of verbalizing their intentions may have helped them with planning and decision making stating *"...sometimes the act of describing a chart helps you figure out exactly what you want. Or, in some cases, you realize that what you're asking for doesn't make sense and you change your mind.'*

3.4.6.3 View organization challenges

Participants who commented on the window positioning approach adopted in the study, where the mediator automatically positioned views for the participant, tended to have more negative impressions. These challenges are noteworthy, because even though the mediator had extensive experience positioning and displaying views for the participants, and managing large numbers of views, doing so manually posed challenges and was imperfectly executed at times.

A few participants wanted control over view positioning and suggested a touch screen to enable this. While this was not possible in our study, it would be sensible in a realized system to provide some direct control the user in managing the views on screen. However, from the study pilots, we knew that some automatic decisions in managing the views was needed from the mediator, otherwise visual clutter was a significant barrier.

3.5 Discussion

In this section we integrate our coded observations from the data exploration sessions with participant comments from the interview and survey in order to consider the design implications of our findings.

3.5.1 Arriving at Many, Not Just One

We found that cast-a-net view generation was a common request style. Multiple view responses were appreciated by our participants, and we describe many cases where the groups of views produced from cast-a-net requests enabled tasks that spanned more than one view. Essentially, this request style allowed participants to create groups of views that were useful together as a collection.

In contrast, many visualization environments aim to help participants arrive at a single view, or a series of single views, that address their questions. Some systems do this through intuitive interface design, such as Tableau and its precursor Polaris (59). In systems where alternate views are presented (15), often these are framed as alternate options to help users find useful single views, or as ways to accommodate ambiguity in the expression of intentions. In other cases, many views are presented to users in order to help guide a faceted exploration of visualization recommendations, as in Voyager (6). But, since Voyager uses a bookmarking mechanism, allowing users to mark useful single views, the end goal is still framed as helping users create a set of useful single views of their data.

Flexible canvas environments often adopt the single view framing, by focusing on how the environment accommodates the display of a trail of single views (39). But, if we return to the original insight from 'space to think' (35), the value of a flexible canvas for sensemaking was in arranging and grouping analysis artifacts around conceptual relationships and these groups of artifacts were useful when considered together.

We suggest that flexible canvas environments should target more than displaying a chain of useful views, but instead displaying sets and collections of views that are valuable together. Developing interactive interfaces that enable cast-a-net view generation would help users arrive at these sets and collections of views efficiently and leverage the large, flexible display area.

3.5.2 Collective and Parallel Actions

We observed that participants used collective many-to-many copy+pivot referential requests to create many views of data at once. It is possible that when groups of views have common features, users might be inclined to act on this set collectively, rather than one at a time. For instance, when a set of views has a common filter, it may seem intuitive to pivot this set to a new filter in one command.

Between-view relationships are often framed as an aid to visually presenting views in flexible canvas environments (102; 39), but these relationships may also be an aid to facilitating efficient interactions in multi-view environments. Expressing many parallel actions on a single view target, appeared to be an efficient way to create sets of views for multi-view tasks, such as comparison. Further exploration of how to enable these efficient interactions, and the contexts in which users would like to act on sets of views collectively is an interesting area for future research.

3.5.3 View organization: Not yet realized

The mediator used flexible positioning of views to create custom groupings and arrangements that reflected the content of the views. Based on piloting this study, positioning views for the participants served to help with visual clutter and to communicate complex multi-view responses using spatial positioning.

However, freely positioning and re-positioning the many views of data generated during each study posed significant challenges and was imperfectly realized. While it is clear that users of flexible canvas environments want to have control over view positioning, it is also clear that free positioning is time consuming and potentially difficult. Algorithmic positioning approaches, such as tiling, are fast but do not take into account the content of the windows being arranged, which limits their utility.

It is generally assumed that the benefits of flexible, manual positioning in large, multi-view environments outweigh the costs, in time and visual clutter. Although this study does not directly challenge this assumption, we suggest that additional algorithmic view positioning tools that take into account between-view relations might make it is easier to manage many views on data.

3.6 Conclusions

We contribute observations of intentions to create many views at once to accommodate tasks that span more than one view. Using a methodology in which participants explore data on a large display by directly expressing their data exploration intentions to a remote mediator, we were able to examine how participants want to explore data independent of realized visual interfaces, tools or interaction approaches. We noted that participants posed requests in ways that cast a net around sets of data attributes and subsets of the data. They accomplished this both using direct requests, and by referencing and selecting from existing views. We describe how participants used these actions to create sets of views which accommodated tasks that spanned more than a single view. The take-aways from this study are that flexible canvas systems should consider techniques to facilitate creation of many views at once for multi-view analysis tasks.

CHAPTER 4

TRAVERSE: NATURAL LANGUAGE DATA EXPLORATION TECHNIQUE FOR CREATING, AND PIVOTING, COHERENT SETS OF VIEWS

This chapter describes a novel interaction technique, motivated by the findings presented in the previous chapter. This is primarily my own work. Abeer Alsaiari (AA) helped to refine the query and response types, and provided input on the design of the interface and the user study problem description. Moira Zellner (MZ) and Anuj Tiwari (AT) provided a dataset and decision problem, which enabled me to expand and test the design of the technique, and also develop a compelling use case for evaluation. Andrew Johnson (AJ) helped me evaluate early ideas, and refine the overall scope and direction of the project. Discussions with Barbara Di Eugenio (BDE) and Abhinav Kumar (AK) helped me to define more clearly the distinction between direct questions, and referential questions, and highlighted the approach of extracting data values and attributes for NL interpretation.

4.1 Introduction

In data exploration, analysts toggle between open-ended inquiry, targeted questions around data values and attributes of interest, and incremental steps to new portions of the data and attribute space, driven by both the interests of the analysts and their observations from generated views of data (9; 10; 11; 8). In the process, analysts may generate many views of their data, which show different combinations of filters, data attributes and aggregations (6). By examining and juxtaposing multiple views that feature **data variations**- a diversity of data and attribute selections- analysts can perform exploratory tasks that span more than one view, such as comparing subsets of the data against each other or putting a focused observation into a general context (85; 30; 86; 39; 101; 38).

These multi-view exploratory tasks benefit from rapid, on-the-fly view generation and juxtaposition (30; 31; 32; 33). However, view construction itself can be time consuming and error prone, with road blocks in selecting appropriate templates and mapping data to visual elements (12). There has been interest in creating systems that allow users to directly specify desired visualizations using natural language (NL) interactions. By allowing users to express their desired visualization directly, using NL inputs, users can focus on their data and tasks, rather than on learning how to navigate a graphical user interface within a visualization tool (48). Recent research has explored NL interactions for view creation and refinement (14; 16; 134; 28; 29), resolution of ambiguity (28), conversational interaction (135; 20; 21; 22; 23), and multi-modal interaction combining NL with other modalities, such as touch and pen on tablets (25).

In this Chapter, we present **Traverse, an interaction technique to efficiently create sets of views with coherent data variations for visual data exploration using natural language commands**. The focus of this technique is on rapid, on-the-fly view generation that supports *breadth in data exploration* (6)- helping users *traverse*, or travel through, the data. Our technique uses a small set of actions, with a corresponding natural language command grammar to invoke them. This technique responds to requests based on data values and attributes of interest- both **'targeted' requests** that can be responded to in a single view, and **multi-view 'cast-a-net' requests**. In addition, we create a novel set of **'referential actions'**, which allow users to refer to a prior view or set of views and ask to pivot the targeted visualizations to new selections of data values and attributes. Our technique also externalizes the exploration process (30), and preserves a history of the questions posed by the user and the responses provided by Traverse. These actions together allow users to explore the data based on their interests, and iteratively expand the focus of their exploration. Our technique is designed to produce sets of views with coherent between-view relationships (39), that can be arranged in coherent configurations in support multi-view exploratory tasks.

The Chapter contributes:

- Design decisions in support of a novel interaction technique for data exploration, which uses natural language commands to create collections of views with coherent data value and attribute variations.
  - A description of targeted and 'cast-a-net' requests, connected to a set of responses that we term 'view collections', which contain coherent sets of views that present both conserved and varied data values and data attributes.
  - A description of referential actions, that target these view collections, and then expand or pivot them to new points of interest- new selections of data values and attributes - for iterative exploratory interactions.

- An evaluation of Traverse, a prototype system that implements this technique, examining how these actions are used together to enable breadth in exploration.

4.2 Background

4.2.1 Data Exploration

First, this work contributes to research in supporting data exploration (5; 9; 10; 11), in particular externalizing the exploratory process (85), juxtaposing multiple views of data (35; 38), and aiming to support breadth in data exploration (6).

Within this body of research, there have been several techniques to support externalizing and juxtaposing multiple views of data. These tools (30; 31; 32; 33; 34; 28) enable users to create views of their data in data exploration by interacting with a graphical interface or through direct actions with elements in the visualization itself, to create new views of data. Many also support incremental steps, where a view serves as a starting point and is copied and pivoted to a new view, which depicts new portions of the data and attribute space.

This work contributes to this body of research in several ways. First, this technique adopts the approach of copying and pivoting targeted visualizations, for exploring new parts of a large data set, but applies it this technique to multiple view targets, not just one. The primary way that it accomplishes this is by representing views in coherent collections, which is a novel direction for this line of research. Second, it presents a natural language grammar for accomplishing these copy and pivot actions, along with actions to expand the reach of a view collection, which reflects exploratory transitions documented in the literature (1). There are benefits to considering an NL approach to data exploration, and mapping NL commands to exploratory transitions, which we accomplish in this work.

4.2.2 Natural Language Interaction

Recognizing that view creation is challenging and error prone (12), there has been interest in recent years to develop natural language interfaces for data visualization, that allow users to directly ask for the visualizations that they need, rather than learn a complex interface. Much of this work is summarized in Srinivasan and Stasko in 2017. (48). This work includes a number of approaches that address requesting and refining visualizations (14; 16; 80; 29), responding to ambiguity in visualizations (15), using conversational inputs (20; 21; 23; 22; 135), and enabling multi-modal pen and touch and speech interactions (25), among others (26). However, natural language interaction has been under-examined with respect to data exploration, particularly contexts where users externalize their exploration process, and juxtapose multiple views of their data.

In the previous chapter, I presented our work published in Aurisano et al. (44), which examined data exploration in a qualitative, observational study, where participants expressed data exploration intentions using speech and mid-air gestures, and responses were provided by a mediator who acted on their behalf. We found that participants used several techniques to efficiently arrive at sets of views with data variations. First, participants posed direct requests for new views of the data in ways that 'cast-a-net' around a set of data values and attributes, in effect requesting many views of data at once. Because these views were requested in a single query, they had shared features that allowed them to be presented to the user in coherent arrangements, such as in a grid of views. Then, when given these sets of views, participants in the study would refer to them collectively and ask to see the same thing but with new data values or attributes, in effect pivoting many views at once to new portions of the data value and attribute space. This allowed the participants to efficiently express complex intentions, and perform exploratory tasks that spanned more than one view of the data. This research suggested an approach for NL for data exploration, which consists in a combination of direct requests for data values and attributes of interest, and then actions that reference and pivot existing sets of views. This work suggests that NL interactions with these characteristics can aid in breadth of exploration arriving at many views of data, in coherent arrangements, with varied data values and attributes.

In this chapter, we build on the work the findings in the previous chapter, coupling targeted, cast-a-net and referential operations- to enable efficient expression of complex interests within the data.

4.3 Usage Scenario

In this section, I will motivate the design of the data exploration technique, and the design of Traverse- the implemented prototype system that utilizes our technique- through a data exploration scenario. This scenario is drawn from the work in the previous Chapter, which is published in (44). This scenario will refer to a dataset of consisting in a table of crime incidents classified by 1) crime type (eg. theft, assault...), 2) location type (eg. street, residence...), 3) neighborhood (eg. Downtown, Near East Side...), 4) Time of the day, 5) Day of the week, 6) Month of the year, and 7) Year. The data consists in thousands of rows (crimes) and 7 data-

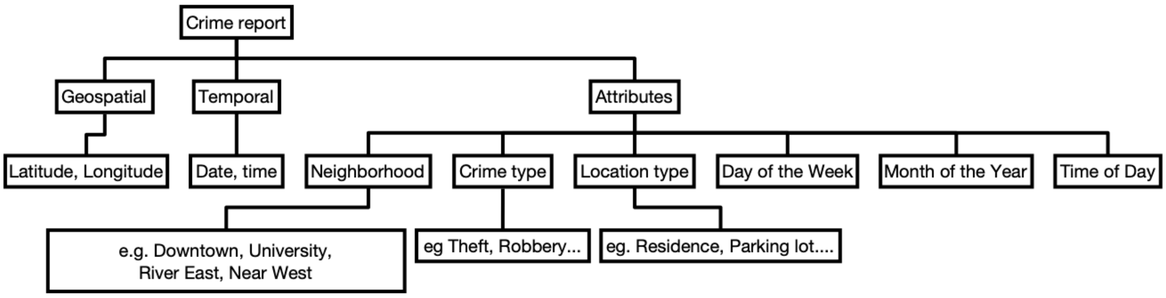


Figure 9: Data used to design the usage scenario, and data used for training in the evaluation of the technique. A crime dataset with data attributes and values related to crime incidents in four Chicago neighborhoods.

attribute columns. This 'crimes' dataset will be referenced throughout the rest of this section. This data is shown in Figure 9, and described in Appendix A.

After loading the data, the analyst wants to gain an understanding of the data. They would want to ask 'Show me an overview of the data', and browse a set of representations that show each of the data attributes in the dataset, in an appropriate visual representation. We term this a **'browsing request'**- browsing within a set of data attributes across a conserved filter. So far, she has posed one question, and surveyed 8 views of the data.

Two of the charts in this browsing collection capture her attention- the frequencies of all crimes broken down by neighborhood, and the frequencies of all crimes broken down by crime type. Examining the chart showing frequencies of all crimes broken down by crime type, she may notice that theft is the most common crime type, followed by burglary. She wants to know if this is true in her neighborhood. She might ask 'Can I see crimes by crime type in the University neighborhood?'. This **targeted request** has a specific data value and attribute focus (data value: neighborhood = University; data attribute: month of the year), and could be responded to in a single view. Now, she has asked two questions and examined 9 views of her data.

Looking at this single view, she notices that theft and burglaries are most common in the University neighborhood. She wants to know if this is also true in the downtown neighborhood, where she often goes on the weekends. She might ask, 'Can I see this but for the Downtown neighborhood'. She wants to keep everything about the initial view- the data attribute, crime types- but now she wants to *swap* data values (1) through an action that we call a subset-pivot, with new filter criteria, 'neighborhood = Downtown'. The resulting new view is similar to the original, in that both have the same data attribute in the same visual template, but now they show different selections of data values. We term the result **a comparison collection**, because it would enable the analyst to compare two selections of data values, which serve as filter criteria for each chart, (University neighborhood vs Downtown neighborhood) with respect to a common data attribute. At this stage, she has posed 3 questions, and explored 9 views of the data.

The analyst might shift their focus to thefts in the University neighborhood, requesting one attribute ("Can I see thefts near the University by time of the day?"- **targeted request**), then another ("Can I see this but by day of the week?"- **referential data attribute pivot**), then another ("Show me this but by month of the year."). These incremental operations would retain the filter criteria- thefts in the University neighborhood- but *swap data attributes* (1).

The growing set of views would be considered a browsing collection, like the first she generated, but unlike that first set, she produced this one incrementally. At this stage, she may have posed 6 questions, and will be examining 12 views of her data.

After surveying a set of views, all focusing on the University neighborhood and theft, she may want to say "Can I see all these views, but for burglaries". She would want to *swap data value* (1) selections- a component of the filter criteria, the crime type- for a different selection, a different area of interest. This **collective referential action** would copy each view the focuses on thefts, and produce a new view, now that focuses on burglaries. The resulting grid would line up data attributes and filter criteria, in a **'browse and compare grid'**. At this stage she would have posed 7 questions and generated 15 views of her data.

The analyst may continue her exploration - posing both targeted and 'cast-a-net' queries, followed by referential copy and pivot operations- to shift or extend her focus to cover more data value and attribute selections. The many views she produced have coherent variations in data value and attribute selections, and they can be presented in coherent juxtapositions- lines and grid. She also did not need to specify intended visual encodings or layouts, which has been shown to be difficult for many analysts (12), but she has posed all of her questions around her *interests in the data.* By making it possible to create, pivot, expand and group many views together based on these coherent patterns, the analyst can perform exploratory tasks that touch on her evolving points of interest, allowing her to incrementally develop an understanding of interesting features within the data.

In this short usage scenario, the analyst posed 9 questions, to generate 34 views, each of which have a unique combination of data value selections, which are used as filter criteria, and data attributes. The many views she produced have coherent variations in data value and attribute selections, and they can be presented in coherent juxtapositions- lines and grid. She also did not need to specify intended visual encodings or layouts, which has been shown to be difficult for many analysts (12), but she has posed all of her questions around her *interests in the data.*

The results visualizations from each question, can be considered collections of views with different properties- different types of similarities and differences. By making it possible to create, pivot, expand and group many views together based on these coherent patterns, the analyst can perform exploratory tasks that touch on her evolving points of interest, allowing her to incrementally develop an understanding of interesting features within the data.

4.4 Design

In this section, I will describe the design of our natural language (NL) interaction technique for data exploration, including our design goals, design process our ultimate design decisions. This design is motivated by the scenario presented in the previous section, and it informs the implementation of Traverse, which I will describe in the next section.

4.4.1 Design goals

In this section, I describe the design goals that inform the ultimate design of our NL data exploration technique. I will indicate in the text prior work that we referenced in developing these goals.

4.4.1.1 Design Goal 1: Enable exploration by capturing data interests, rather than design interests

Our first design goal, which guided the development of our technique and implemented system, Traverse, was to **capture and respond to a user's interests in the data- the selections of data values and attributes that they wished to examine- rather than capturing and responding to requests for specific views- interests in visualization design** (1). We prioritize expressions of data interests rather than design interests (6). We will refer to this as Design Goal 1.

In practice, this meant that we focused on responding to NL queries such as 'Can I see theft by day of the week", or "Can I compare thefts on Tuesday to burglaries, by neighborhood?" or "Show me some information about crimes that occur on streets in the River East neighborhood." We did not focus on enabling queries like 'Show me a heat map colored by number of crimes from light red to dark red", or "Can I have a bar chart showing the frequency of crimes by crime type, sorted by frequency ." We also did not respond to refinement operations that changed the template, encodings or layouts within a view, such as 'now change the axis to a logarithmic scale', or 'highlight the top 10 crime types, in yellow'.

We chose this goal because, while there is extensive prior work on NL interfaces for specifying and refining visualizations (14; 15; 16; 24; 28; 25), there is less work that focuses explicitly on enabling data exploration through NL interaction, specifically enabling rapid creation of sets of views featuring diverse selections of data values and data attributes. NL presents benefits for visual data exploration, as it does for view creation. In particular, users can focus on the data and their interests, rather than on navigating an unfamiliar user interface or considering how to map their interests to visualization templates, encodings and mappings, which has been shown to be challenging and error prone (12).

Our focus on capturing data interests, rather than design interests, allowed us to prioritize enabling breadth in data exploration (6), in particular enabling users to create many views reflecting different combinations of filters and attributes. The challenge in our system was to determine how to automatically and coherently map what users expressed, into views that were useful for data exploration.

4.4.1.2 Design Goal 2: collections of views

Our second design principle, which guided the developed of our visual data exploration technique, was that our technique would **allow users to express multiple points of interest- multiple data attributes and data values- in a single command, and the response will consist in a set of views that cover the range of expressed interests in a coherent structure.** The goal is to allow users to efficiently express a *breadth of exploratory interests*- such as interests in several data attributes and/or several data values. By doing this repeatedly, users can rapidly cover many combinations of parameters, to identify meaningful patterns or features in the data. We will refer to this as Design Goal 2.

Some prior systems for view creation in data exploration in multi-view, flexible canvas environments allow users to create views one at a time, through a graphical interface, selecting a few data attributes and filters to apply to create a new visualization (33; 30). Other data exploration environments begin with a single point of interest and provide a set of recommendations that guide users in their exploratory process (6). Our approach differs from these in part due to the flexibility of NL expression. In the previous Chapter, and in our publication (44), we noted that participants expressed intentions for many data attributes and values, when unconstrained by the design choices of existing interfaces. This mode of 'casting a net' around a set of interests within a dataset, allowed participants to communicate efficiently their exploratory intentions.

We term the set of views provided to users in our technique **'view collections'**, because they exhibit coherent variations in data values and attributes. By coherent we mean that there are conserved data values or attributes, and features that vary according to a comprehensible pattern or rule. For example, a coherent collection might contain three views, all of which show crime frequencies by day of the week, but the first shows only crimes classified as thefts, the second view shows only burglaries and the third shows only trespassing. This example view collection we call a *comparison collection*, because it juxtaposes views that vary according to non-overlapping subsets of the data, to identify similarities and differences between those subsets, with respect to a common data attribute. We describe the range of view collections supported in our technique in the next section.

4.4.1.3 Design Goal 3: iterative exploratory steps

Data exploration is a process of iteratively stepping through the data and attribute space. Users might begin with one focused point of interest, and then wish to step to a new point of interest. These iterative steps are systematically described in Lee et al. (1). Examples include adding new filter criteria, or swapping data attributes. These exploratory steps are beneficial, because they allow users build their understanding of the data incrementally, or to spontaneously direct their path through the data based on unexpected observations or insights that arise from visualizations they create during the exploratory process (5). We will refer to this as Design Goal 3.

These incremental steps are echoed in the previous chapter, where participants would refer to prior views, and ask 'Can I see this but...', with a change to the selection of data values and attributes. We termed those actions **'referential operations'**, a label we will adopt here in our exploratory technique.

Our technique enables **iterative data exploration using 'referential operations', that allow users to take an existing view or view collection and extend it to cover new data values or attributes, or to collectively copy and then pivot the collection to new selections of data values and attributes.** Both operations take advantage of the coherent structure within the collection- data interests that are conserved, as well as data interests that vary according to a pattern or a rule.

The benefit of this design approach is that users can extend or shift the focus on their exploration from one region in the data and attribute space, to another, through a referential operation. A second benefit is that referential operations allow users to avoid re-articulating a complex request. Instead, they can say "Can I see this but..." and express the change they wish to see enacted on the selected values or attributes within the view. This action furthers design goal 1 and 2, in that users continue to express multiple points of interest, and the system continues to respond with collections of views that have coherent structure. It adds the feature of iterative steps, because users can chain together multiple referential operations to build large collections, or iteratively pivot views to new points of interest.

4.4.1.4 Design Goal 4: Use a simple, NL command grammar

Where possible, we opted for a simple NL command structure, to focus on how participants used different kinds of actions, rather than to enable complex specifications. This is similar to the approach in InChorus, by Srinivasan et al. (25), which put forward a defense of using a restricted NLI in order to resolve issues with a new interaction approach.

4.4.2 Design process

Referring to the usage scenario and our design goals, we 1) systematically defined a set of direct request types, with different numbers of data values and attributes, 2) created a proposed mapping between these combinations of parameters to view collection responses, and 3) exhaustively considered all actions that would modify the data values or attributes within each collection.

We focused our work on visualizations with a primary data attribute (such as a bar chart, or a single line chart), or a primary and secondary data attribute (such as a multi-line chart or a heat table). These are common and accessible chart types, that apply to many kinds of data. For the purposes of this work we did not consider view types with a three or more data attributes- such as a colored scatterplot- but we discuss the extension of this approach to visualizations like this in our discussion section.

These actions and responses were discussed by a team of two PhD students, and refined through a formative evaluation with 5 participants, all of whom are PhD researchers in human computer interaction. These 5 participants used a prototype version of our system, Traverse, and offered feedback on the requests and responses. In response to this feedback, we refined our approach to responding to data attributes and pivoting a secondary data attribute, and refined the visual templates to make the distinctions between views with different data attributes more clear.

For our work, we focused on two datasets- a city crime dataset and a Covid19 dataset (136). We focused on these data because they had varied data attribute and values and mapped to varied visualization templates, and both were associated with data exploration scenarios that were accessible and relevant to users in our formal evaluation.

4.4.3 Translating data interests to view collections

Here we describe how we extract data value and attribute interests from the user's NL query, considering both 'targeted' queries with focused data value and data attribute interests, and 'cast-a-net' queries, with a set of data value and attribute interests.

4.4.4 Filters

We focus on enabling users to express data value interests, such as an interest in *thefts and burglaries in the University neighborhood on Saturdays*, and our technique automatically translates these into one view or a set of views in a view collection (Design goal 1). We allow users to express many data value interests, with the expectation that the system will automatically translate these into visualizations, to enable breadth of exploration (Design Goal 2). We needed approaches that allowed users to express these interests through simple NL commands (Design Goal 4).

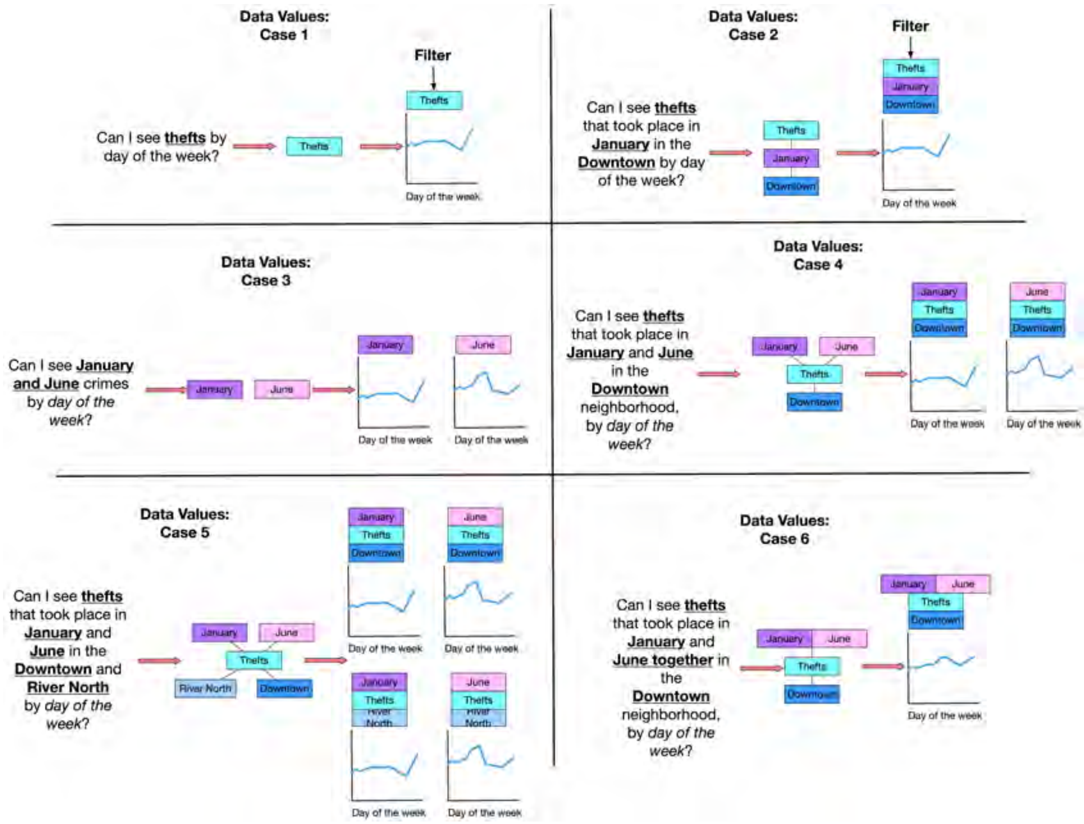


Figure 10: When many data value interests expressed by the user, these are translated into filter criteria. The approach is designed to give multiple views, derived from combinations of the expressed data interests.

There are several possible options, in our technique, for interpreting data value interests. First, when a user expresses an interest in one data value, interpretation is straightforward. This data value servers as a single filter criteria. However, if they express multiple data values, but all are from different data attributes (eg. *"Can I see theft in the University neighborhood in 2012?"*), this also produces a single filter, but with multiple filter criteria. These are Cases 1 and 2, in Figure 10

The second option, arises in instances where users mention several data values from the same data attribute, such as asking to see 3 different crime types, such as *"Can I see theft and burglary and trespassing..."*. We interpret this as a comparison request, and place these three filter criteria in separate filters. This is Case 3, in Figure 10. We defaulted to this approach, dividing into separate filters, because it helped enable breadth of exploration (Design Goal 2) through simple commands around data interests (Design Goal 4).

The third option, applies to cases where users enumerate a complex list of data values, some of which are from the same data attribute, and some of which are not. An example of this type is "Can I see theft and burglary in River East and in 2012?". We show two examples of this, Cases 4 and 5, in Figure 10. This approach allows us to give complex combinations of visualization (Design Goal 2) reflecting diverse points of interest (Design Goal 1), without expecting users to verbalize complex filter criteria (Design Goal 4).

Finally, we added one option for users to combine filter criteria together, by saying "Can I see theft and burglary together in one view", and our interpreter would recognize that they wished to combine these criteria in a single filter. This is Case 6 in Figure 10

Future work may consider how to enable efficient expression of more complex filter criteria, building on prior work from database retrieval through NL (72; 73; 74; 75; 76; 77; 78), though it is possible that this would be achieved through additional input modalities (41; 25).

After interpreting the enumerated data values, the result is either one list of filter criteria, which may be applied to one or several views, depending on the data attributes of interest expressed by the user. Or, a list of separate filter criteria, with potentially overlapping features, but that would be used as filters in different views.

4.4.5 Data attributes

As with data values, users can express one or many data attribute interests, such as an interest in crime types and neighborhoods (Design goal 1). We focus on translating these points of interest into one or many visualizations, to enable breadth of exploration (Design goal 2).

Requests for a single attribute, are the most straightforward case. Single data attributes are mapped to a single view type, based on the attribute characteristics, such as whether the data is categorical, temporal, spatial, etc. In our data, this included bar charts, line charts and maps.

Our responses to multiple data attributes evolved based on user feedback in formative evaluations. The main challenge with multiple enumerated data attributes is whether to allow any of the attributes to be treated as 'secondary' data attributes, such as asking for a line chart by year colored by neighborhood, where the primary attribute is 'year', shown on the x-axis, and a secondary attribute is 'neighborhood', reflected in the color scheme. We wanted to avoid an overly complicated approach, so we opted to capture the enumerated attributes as a list, by default, which elicited a specific interpretation as a view collection (a complex collection, which we describe in the next section), or to look for specific keywords that indicated that users wanted to treat some of the enumerated attributes as secondary attributes. Specifically, participants in the formative evaluation wanted to express a desire to use language such as 'versus', 'colored by' or 'split' (as in split this single line into multiply lines), to specifically treat one of their expressed attribute interests as a secondary attribute. An example of this kind of query is, "Can I see crimes by year versus neighborhood", which we interpreted as a multifaceted request, where 'year' is the primary attribute and 'neighborhood' is the secondary. Variants of this style of interaction include 'Can I see crimes by year and month colored by neighborhood", which would produce a list of two primary attributes (year, month) and one secondary attribute (neighborhood). These differences are depicted in Figure 11.

4.4.6 View collection creation

After extracting the data value and attribute interests from the user, and considering particular keywords or regular expressions, our technique would have a list of filters, consisting in one or several filters with one or several filter criteria, and a list of primary and/or secondary data attributes. We then mapped the set of interests to a view collection response type. The specific number of filters, primary data attributes and secondary data attributes, determined which view collection type to give to the user. The complete description of this approach, and the types of view collections we provided to users, is depicted in Figure 12.

For each view collection, there was a protocol for translating these lists of filters and attributes into a set of views, often accomplished by creating combinations of filters, primary and secondary attributes. For each data attribute, or pair of primary and secondary data attributes, the view creator would assign to a visual template, based on the attribute type (eg. temporal data to line charts, ordinal categorical data to sorted bar charts, geospatial data to a map).

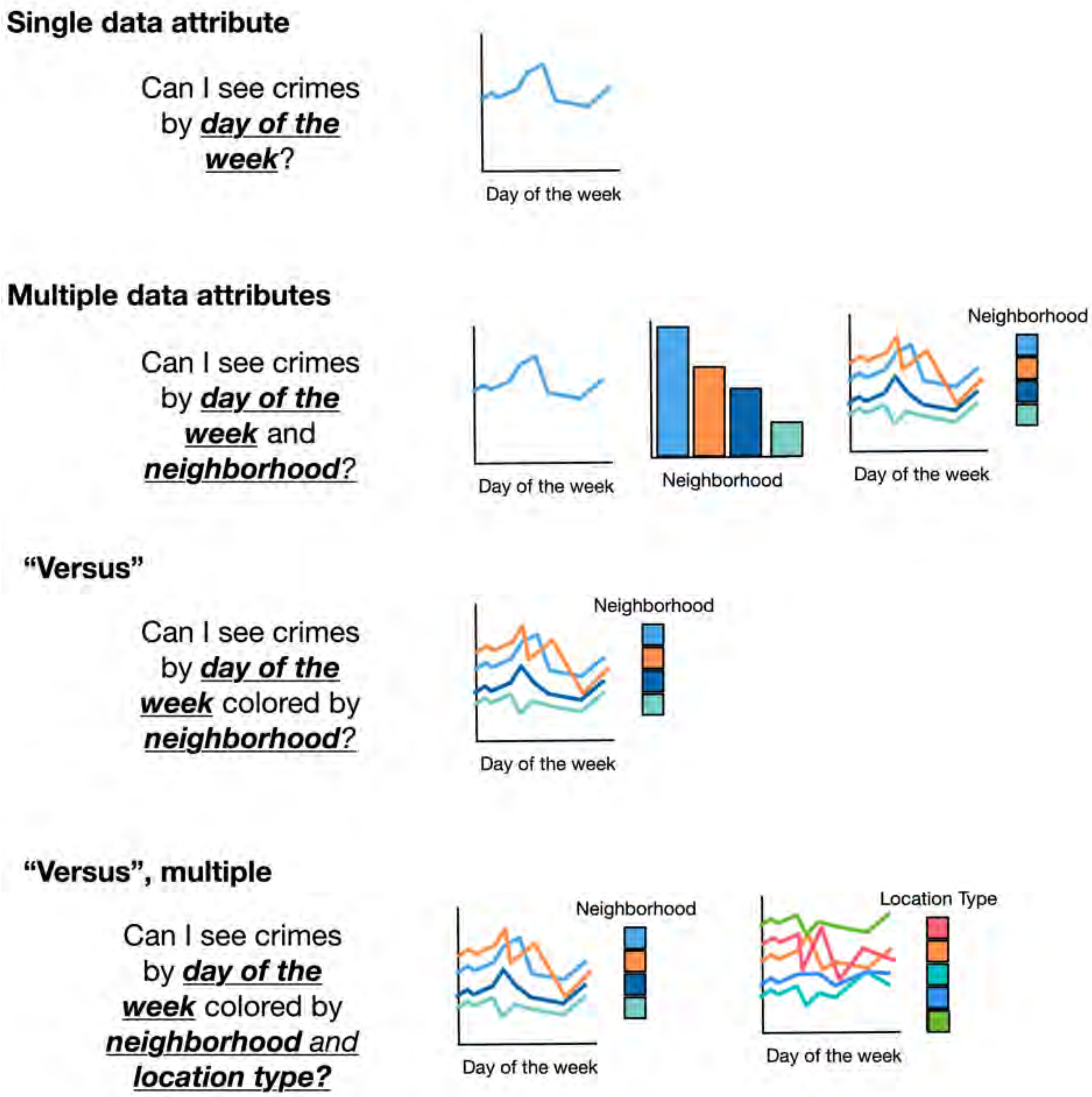


Figure 11: When many data attribute interests expressed by the user, these are translated into different view collections, depending on the number of attributes and other keywords.

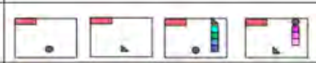
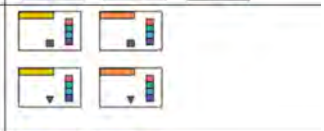
Type	Example questions	# Subsets	# Primary Data Attributes	# Secondary Data Attributes	Form	Figure
Target	"Can I see theft by day of the week?"	One	One	None	Single	
Multifaceted	"Can I see theft by day of the week *versus* neighborhood?"	One	One	One	Single	
Browse	"Can I see some information about theft?"	One	None-> Wildcard-> Multiple	None	Line	
Compare	"Can I see theft, burglary, trespassing and criminal damage by day of the week?"	Multiple	One	None	Line	
Complex	"Can I see theft by neighborhood and location type?"	One	Multiple	None	Line	
Subset+attr	"Can I see theft by day of the week, versus neighborhood and year?"	One	One	Multiple	Line	
Multifaceted Compare	"Can I see theft, burglary, trespassing and criminal damage by day of the week, versus neighborhood?"	Multiple	One	One	Line	
Multifaceted Browse	"Can I see some information about theft, versus neighborhood?"	One	None-> Wildcard-> Multiple	One	Line	
Browse and Compare	"Can I see some information about theft and burglary?"	Multiple	Multiple	None	Grid	
Multifaceted Browse and Compare	"Can I see some information about theft and burglary, colored by neighborhood?"	Multiple	Multiple	One	Grid	
Comparison combinations	"Can I see theft and burglary in June and July by day of the week?"	Multiple	One	None	Grid	
Comparison combination, multifaceted	"Can I see theft and burglary in June and July by day of the week versus neighborhood?"	Multiple	One	One	Grid	
Multifaceted Compare x Sub+Attr	"Can I see theft, burglary, trespassing and criminal damage by day of the week, versus neighborhood and year?"	Multiple	One	Multiple	Grid	
Multifaceted Browse x Sub+Attr	"Can I see some information about theft, versus neighborhood and year?"	One	Multiple	Multiple	Grid	
Complex by Compare Grid	"Can I see theft and burglary by neighborhood and location type?"	Multiple	Multiple	None	Grid	

Figure 12: View collection types, depending on the number and composition of data values and data attributes.

Figure 13: Figure showing the decision we needed to make on color choices- a neutral, de-saturated color, for all basic charts- or color schemes for each attribute.

4.4.7 Encoding decisions

A significant design decision involved selecting visual encodings in a way that allowed users to visually distinguish the visualizations on the display. During our formative evaluation, participants noted that it was difficult to distinguish charts, particularly as they generated more views during data exploration. We initially opted to show all single data attribute bar charts and line charts with a neutral de-saturated blue color. But this resulted in mostly blue colored charts, which was difficult for users to follow. In response, we applied a unique color scheme, selected from a color brewer API (137), that would visually distinguish charts with different data attributes.

We also made some encoding decisions to avoid visual clutter and to make it easier to examine many views at once. We opted to automatically filter views to show the top 10 of each category to avoid visual clutter. This approach is depicted in Figure 13.

4.4.8 Referential actions

Our technique aims to enable data exploration both through direct requests for views of data, as described in the previous section, and through iterative exploratory actions that extend and shift the focus of exploration (design goal 3). We accomplished this through what we termed a 'referential operation'. Referential operations echo the finding in the previous Chapter and in our publication (44), where participants in an observational study posed requests to expand or pivot an existing view or set of views, to reach new portions of the data and attribute space.

In our technique, a referential operation is one where a user asks "Can I see this but....", referring to a prior view or set of views, and then they specify a change- a new set of data value or data attribute interests- that when applied to the referenced views, produce a new area of focus. The response to this request depends on both interests expressed by the user and the view or view collection that they referenced. We depict example pivot actions in Figure 14.

To resolve referential operations, we use an approach that involves 2 steps. First, the detection of data values and attributes of interest. This follows the same approach described in the previous section, and the result is a set of lists of filter criteria and data attributes. Second, the challenge is to decide how to use the context, the targeted view collection, to produce an appropriate response that extends the exploratory focus for the user. The response depends on the conserved and varied features of the targeted collection.

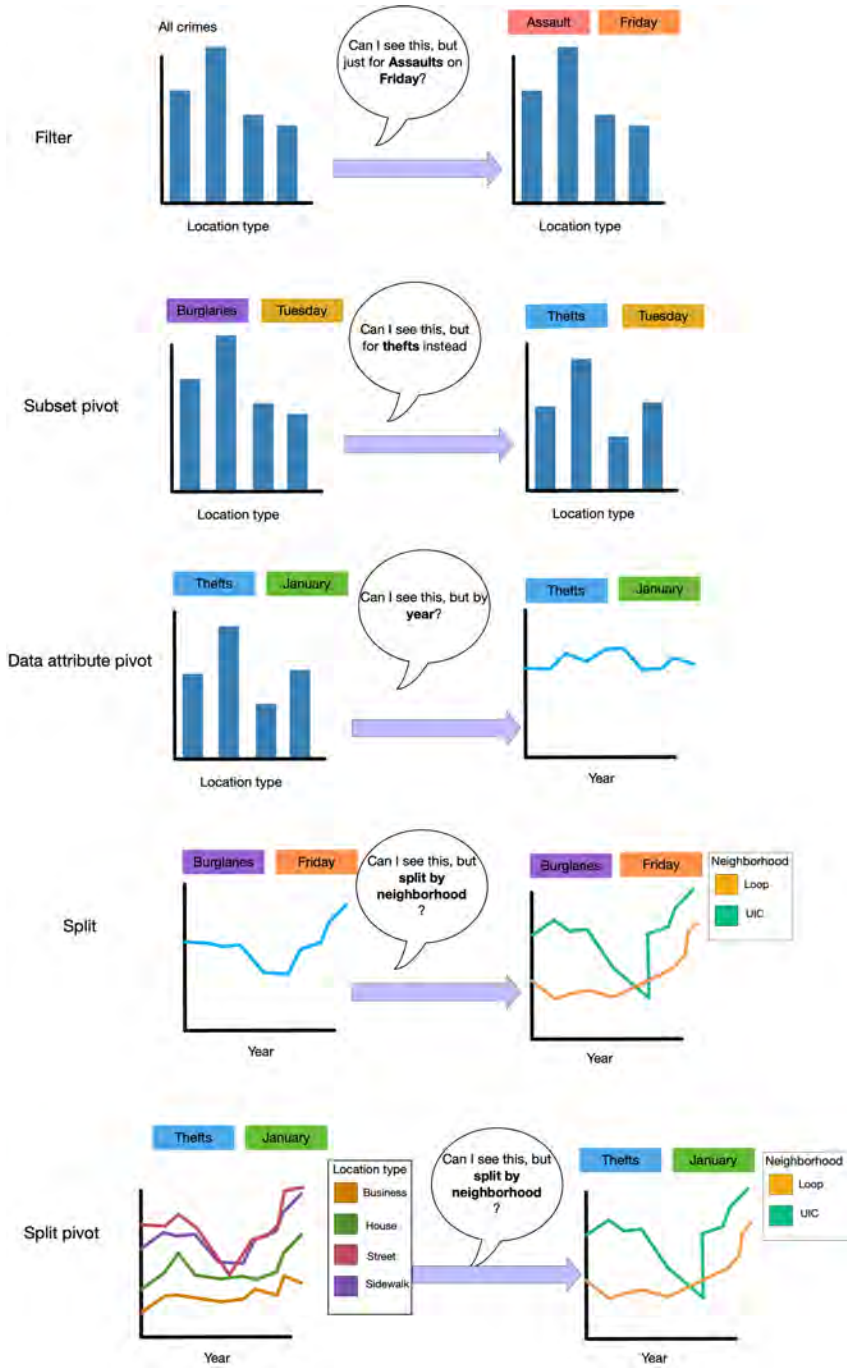


Figure 14: This diagram depicts basic pivot examples. Filtering actions, add in filter criteria. Subset pivots, swap filter criteria, for example from burglaries to thefts. A data attribute pivot switches the primary data attributes, for example from location type to year. Split referential actions, introduce a data attribute. Split pivots swap one secondary data attribute (such as location type) for another (such as neighborhood).

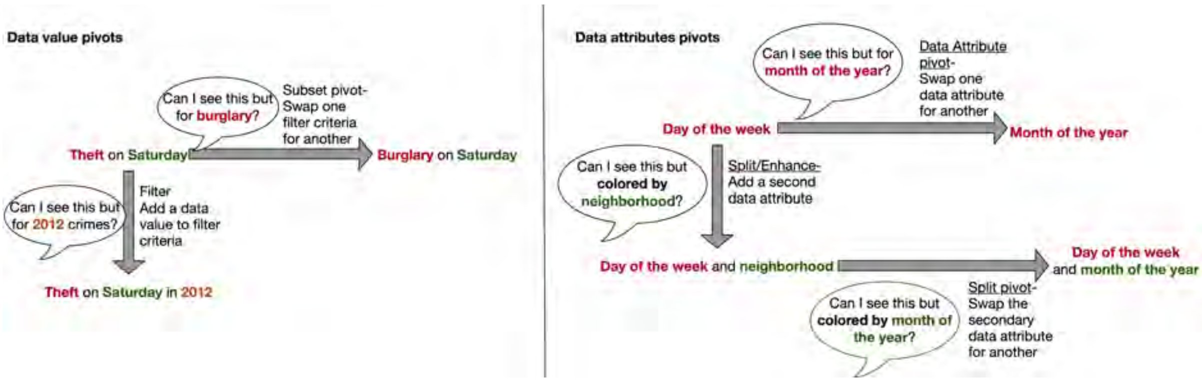


Figure 15: Data value changes expressed by the user. This figure depicts two types of data value pivots. In the first transformation (horizontal), there is a data value swap requested, or a **subset-pivot**, which exchanges theft for burglary, but retains the rest of the filter criteria. In the second (vertical), a new filter criteria is added, a **filter pivot**, with the rest of the filter criteria retained. This figure also depicts three types of data attribute pivots. First (top horizontal) a data attribute pivot, from 'day of the week' to 'month of the year'. Second, (vertical) a **split operation**, or enhance operation (1), to add in a new attribute criteria. Third (bottom horizontal), to exchange secondary attributes, but retain the primary attribute, which is a **split pivot**.

4.4.8.1 New data value and attribute interests

One component of referential requests was detecting what users wanted to see altered. The major types for referential requests are depicted in Figure 15. These operations are related to the analytical operations in Lee et al. (1), and other work on exploratory analysis.

When users list a new data value interest, that is from the same data attribute as an existing filter criteria (eg. "Can I see this but for burglary", on a set of views that show thefts), we interpreted these as requests to *swap filter criteria*. We term this action a **'subset pivot'**, and it is equivalent to the 'swap data value' operation in Lee et al. (1)

Alternatively, users might ask to *add a new filter criteria*- from thefts on Saturday, to thefts on Saturday in 2012. This would apply when users specified a data value from a data attribute not included as filter criteria in views in the current set. This is a filter operation, similar to the 'add data value' operation in Lee et al. (1)

Referential actions involving data attributes involved several possible response types, as depicted in Figure 15. Users may ask to swap a data attribute for another, such as from a plot showing crimes by day of the week, to a plot showing crimes by neighborhood. We termed this an attribute pivot, and Lee et al. termed this an 'attribute swap' (1)

Users may ask to add a new data attribute, which we call a 'split pivot'. We capture these actions through a request such as *"Can I see this but versus (new attribute)"* or *"Can I see this but split by (new attribute)"* or *"Can I see this but colored by (new attribute)"*. We called this a split operation. In cases where the collection already had a secondary data attribute, a request to *"see this but split by (new attribute)"*, would be responded to with a swapped secondary data attribute.

We found in our formative evaluation that users struggled to distinguish between attribute pivots and secondary attribute pivots, particularly when the secondary attribute was the most visually salient, as in colored multi-line charts or colored maps. In this case, users expected a request for a new data attribute to produce a new secondary attribute, without saying 'versus' or 'colored by' or 'split by'. We accommodated this in our technique, defaulting to pivoting the secondary data attribute, for certain chart types.

4.4.8.2 Copy and pivot

When the user expresses a change to a conserved feature across the collection, we call this a **'copy and pivot'** action. An example of this kind of action is shown in Figure 16. Copy and pivot actions include the previously described major types - subset pivot, filter, attribute pivot, split and split pivot.

View collections are characterized as possessing at least one shared data value or data attribute that unifies the views in the collection. For instance, a browsing collection shares a common focus- a common subset of the data. A comparison collection shares a common primary data attribute. A 'subset+attribute collection' shares a common set of filter criteria, and a common data attribute. These similarities can be found in Figure 12. **When referential operations specify a change in a shared feature, our technique translates this into a collective copy+pivot action, where all views in the collection are duplicated, and changed to reflect the new interests expressed by the user.**

In some instances, this results in an enlarged and re-classified initial collection, that now has more dimensions of shared features than before the referential action. This could mean a copy and pivot operation that transforms a single view, into a line of views. Or a copy and pivot operation that transforms a line of views into a grid of views.

For example, suppose a user referenced a browsing collection that showed a set of views with a common filter criteria "theft on Saturday'. Suppose they asked *"Can I see this view, but for burglaries?"*. The response would be to copy each view in the initial browsing collection and pivot it with swapped filter criteria, from theft to burglary, but retaining the rest of the view parameters- the data attribute and the filter criteria 'day of the week=Saturday'. We term this operation a 'subset pivot', since it pivots a set of views with a conserved list of filter criteria- showing a conserved subset of the data- to another list of filter criteria.

In another example, suppose a user referenced a comparison collection showing theft and burglary by neighborhood- a shared data attribute with respect to varied filter criteria. A user may ask 'Can I see this but for day of the week'. This operation will copy both views in the comparison collection, and pivot them, keeping the filter criteria in each but swapping data attributes, from neighborhood to day of the week.

In both of these cases, the new views can be added to the views in the original collection, but the original collection becomes a new type. In both cases, these collections become 'browse and compare' collections, showing a grid of views, with shared filter criteria and shared data attributes in each dimension.

4.4.8.3 Extend a collection

In our technique, some referential operations act to extend the reach of an existing collection, without modifying its underlying type. This occurs when a target collection has varied data values or attributes, and then the user poses a request for new data attributes or values in a way that follows the pattern of the target collection.

For example, suppose the user referenced a browsing collection in their request- specifically a collection of 2 views showing 1) thefts on Saturday by month of the year and 2) thefts on Saturday by day of the week. They might ask "Can I see this but for year?". In our technique, we are retaining the filter criteria in the set, but adding new data attributes. Since the original

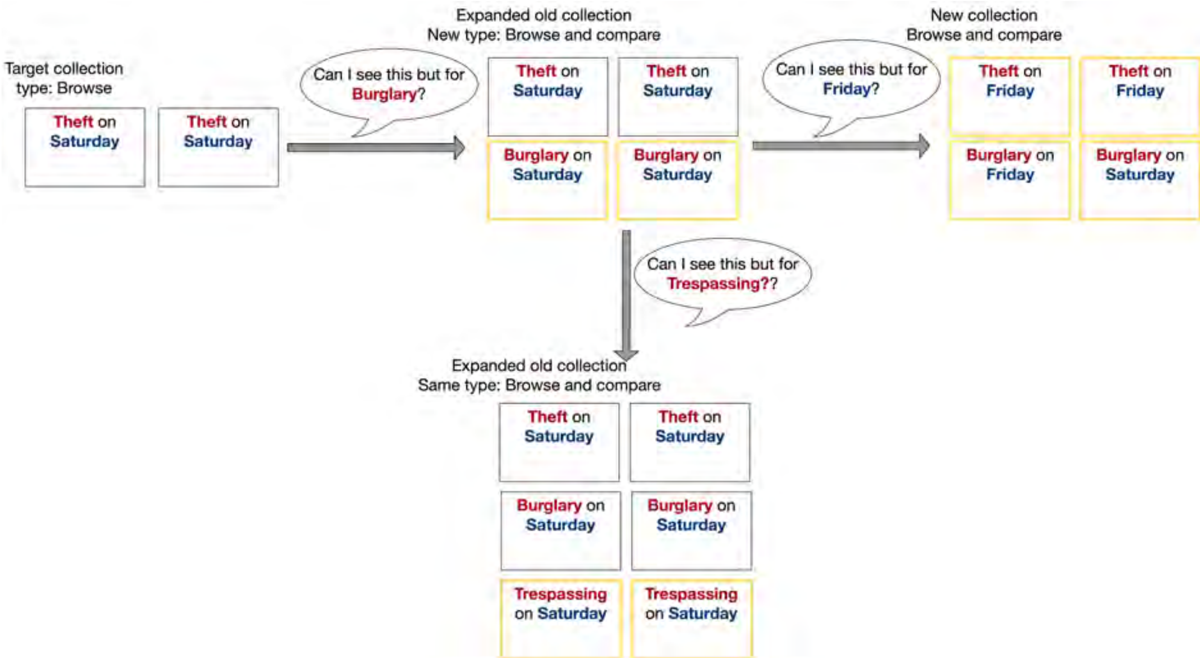


Figure 16: This example highlights two response approaches to referential actions on a target collection. One which extends an existing collection, another produces a new collection.

collection is extended to enable tasks across a wider set of data value and attribute selections, we term this an 'extension' operation. Unlike the copy and pivot operations described above, this action does not copy each view in the collection, but extends the set. An example extension operation is shown in Figure 16.

4.4.9 Exploratory paths

When users combine direct and referential operations, both copy and pivot and extension operations, they can build collections of views with interesting features. There are also multiple paths to arrive at similar view collection types, depending on the user's evolving points of interest. Two example paths through the data are shown in Figure 17 and Figure 18

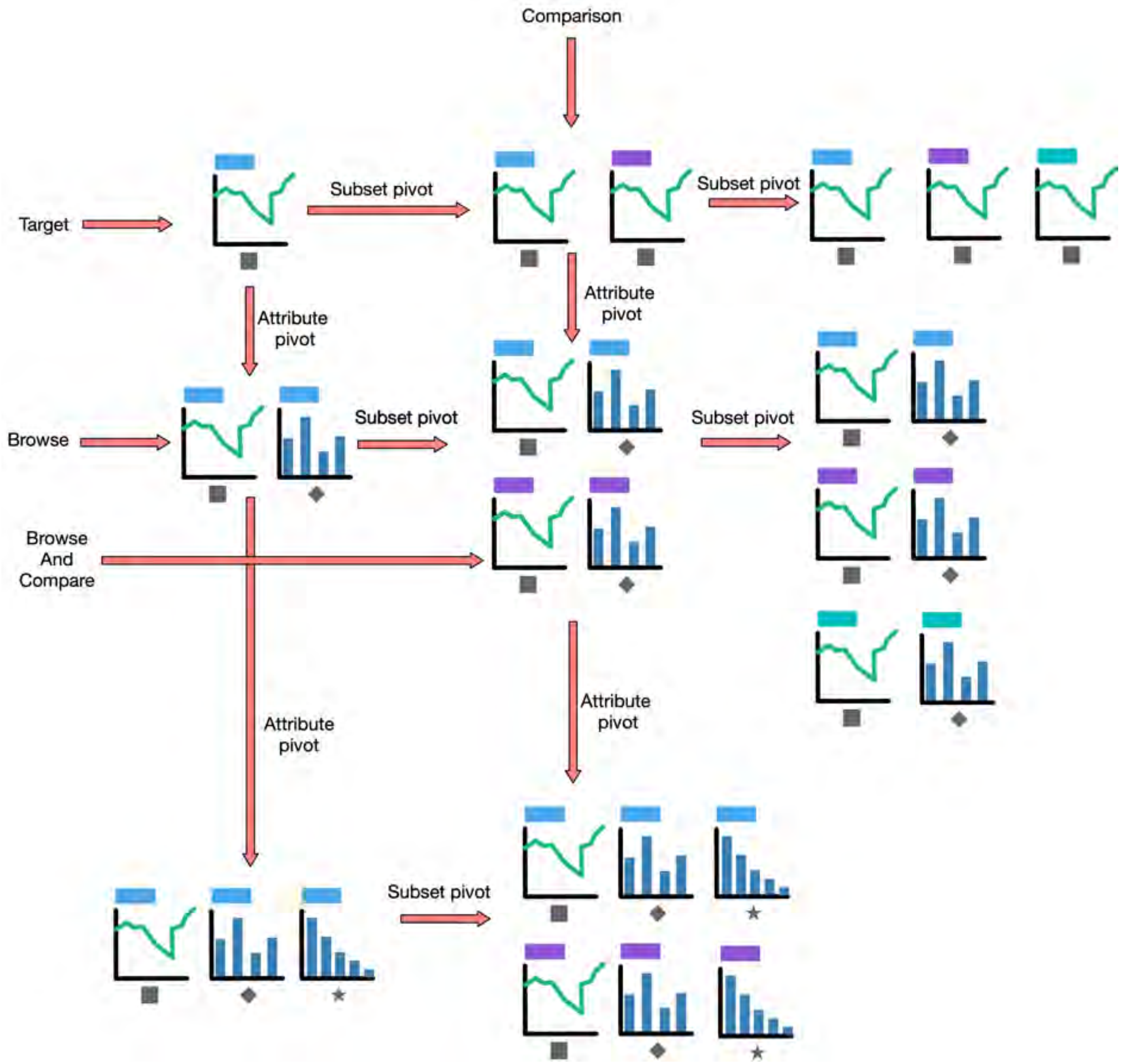


Figure 17: In this referential pivot paths, we show alternate routes to different collections of views. The types on the side and top, are direct question types, and the transitions are different types of referential actions.

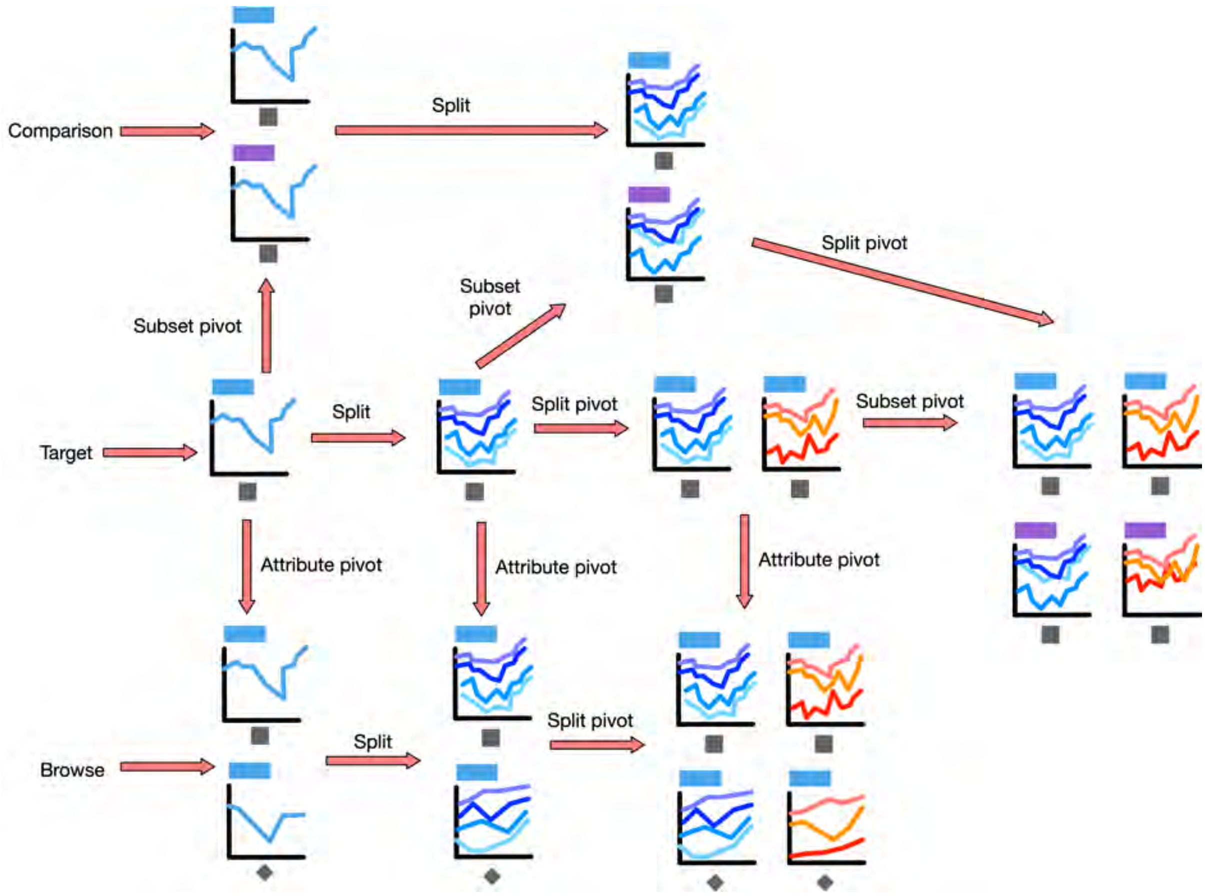


Figure 18: This is a second example of a set of referential pivot paths. We show alternate routes to different collections of views. The types on the side and top, are direct question types, and the transitions are different types of referential actions.

4.4.10 View Collection Layouts

Visualizations without a view collection are presented in either a line or a grid, with corresponding visualizations juxtaposed together in each dimension. As view collections are updated, such as expanding to add new visualizations, or transforming from one collection type to another, we ensured that effective layouts were used, for visual comparisons and browsing within each dimension of a grid.

4.5 Implementation

In this section we describe Traverse, a prototype system which implements our interactive data exploration technique. We named the system 'Traverse', because it enables users to efficiently traverse through many combinations of data values and attributes in exploring the data. The development of Traverse allowed us to refine and evaluate our approach, and capture how users explored data using the technique. Our implementation focuses on interactive data exploration actions, specifically target, cast-a-net and referential pivot and expand actions- to produce collections of views with coherent data variations. We constructed an interface which used this technique for view creation around the data value and attribute interests, allowing us to examine how participants in our evaluation used these actions for data exploration.

Traverse was implemented in Javascript, with a node.js server at the backend that handled data, nlp processing, and generating visualization specifications, in this case VegaLite specification objects. We used a simple flat table, or set of tables, for our data, and used a small set simple custom scripts to filter, aggregate and retrieve data values. We focused on this lightweight solution, for speed and simplicity, to allow us to develop the mechanics of the technique. The frontend was implemented in javascript, with visualizations rendered using VegaLite.

For natural language command interpretation, we used node-nlp to extract attributes and values from queries, along with a set of manually defined keywords, and regular expressions, which mapped to specific operations. Traverse is a form of 'restricted' natural language interface, as is described in Srinivasan et al. (25), which uses simple command grammar. but explores a novel aspect of NL interaction. There has been significant progress in the visualization research around the interpretation of complex, ambiguous or underspecified queries, but we opted to focus on a restricted NLI for several reasons. First, we wished to focus on our novel interaction technique, rather than NL interpretation challenges, and felt that introducing a more complex NL system would distract from this core focus. Second, we wanted our users to learn a specific interaction approach and felt that a simple grammar with a set of clear interaction rules would be easier to learn in our evaluation. Future work could extend our restricted NLI to incorporate more complex NL interpretation techniques, such as the techniques developed in our related work (20; 21; 23; 22).

The Traverse architecture is depicted in Figure 19.

4.5.1 The Traverse User Interface

The traverse interface, depicted in Figure 20, consists in a 1) query entry area, where users can enter their question, 2) a data value and attribute description panel, where users can explore and interactively select data attributes and values of interest, and 3) a query and visualization

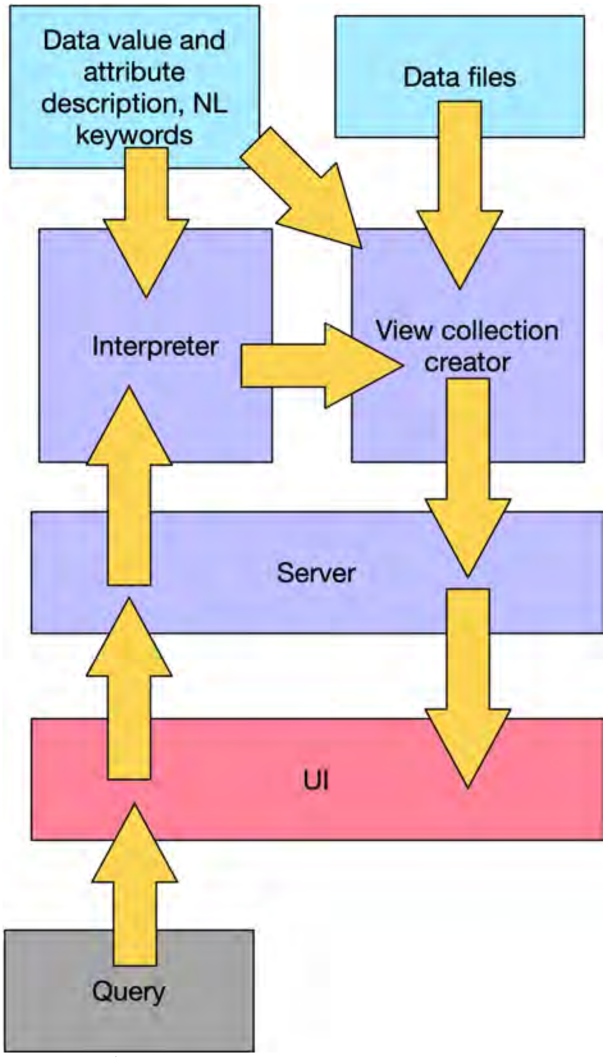


Figure 19: The system design of Traverse, which includes a node.js server, an nlp interpreter, and a view collection creator.

Figure 20: The traverse user interface consists in a query entry area, an interactive data description area, a query and view collection response area, with previous collections shown in a scrolling history. The examples shown above are from the COVID19 data and problem description which is described in detail in Appendix A.

collection scrolling history area, where responses to queries are displayed, in coherent collections, with a scrolling chat-style history that displays prior queries and collections.

4.5.1.1 Query entry and data value and attribute exploration area

Users can enter their query through typing into a form at the top of the interface. We opted to use typing rather than speech, because we wanted to focus on user queries and responses without the complications of speech-to-text translations, particularly since our NLI used a simple NL grammar rather than conversational inputs. Future research could extend our approach to spoken interactions, which have different affordances.

We also supplied a data and attribute exploration panel, which presented an interactive tree representation of the data. Data attributes could be clicked to reveal assigned data values (eg. the attribute 'day of the week' could be clicked to reveal a set of child data values - Monday, Tuesday....etc). Based on feedback during formative evaluations of the interface, we added a feature where users could double click to select a data attribute or value, and add it to the growing query, in the event that users did not want to type a long name.

4.5.1.2 View collection display and history area

After entering their query, the responses are presented in the view collection display and history area. For queries that produce a new view or a new collection of views, either through new queries or pivoting queries, these views are presented in a line or a grid, which is delineated from previous collections by background color. Collections are assigned a background color in alternating neural shades of gray and white, in order to visually separate the view collections. The query associated with the new view collection is displayed above the views, on the left, top side in the view collection area.

For queries that expand this collection, such as adding new views to the line or grid, or transitioning a linear set to a grid, the collection on the top is updated. For instance, suppose a user is viewing a collection with one visualization- a display of crimes by crime types in January. If the participants asks 'Can I see this but for June', this set will update with a new view. The view collection is transformed from a targeted collection to a comparison collection, displaying January and June crimes by crime type. As this collection is updated, the background is highlighted with a bright gold color, that gradually fades to the previous neutral gray or white color, in order to ensure that participants understand that this top collection has been updated with new views. The views are shown in a new layout- such as in a longer line or a grid- corresponding to the view collection type. All queries associated with this collection are displayed in order at the top of the collection, allowing the user to see the iterative process they used to build the view collection.

If participants express a pivoting query, it is shown in a new collection. Only the query associated with the new collection directly is shown. Users can scroll down to see the history of queries that produced the 'parent' collection from which the new collection was created. We opted for this approach to displaying the history because we wanted to avoid visual clutter and prevent any confusion in surveying in the query and collection history.

All prior collections are retained, along with the history of queries, in chronological linear order. We opted to keep this order fixed, rather than allow users to rearrange the collections. During formative evaluation of the interface, we observed that participants would copy and paste a prior query or set of queries they wanted to repeat in order to build a new collection, and that this action did not interrupt their flow or the ease of use of the interface. Future work on the design of the interface could explore techniques to go back in the history and create new branches of exploration, that start from past queries and view collections.

In addition to highlighting updated collections as new views come in, we also implemented an 'active collection' highlight on the left side of the collection, in a golden yellow color. This helped to indicate which collection would be expanded or pivoted for referential actions. As the user typed their question into the query bar, the page used a simple parsing scheming to detect the keywords for a referential operation- for instance detecting 'Can I see this but...". Upon detecting these keywords, the active collection indicator would expand, visually indicating that the system understood that the users was referencing the highlighted question in their query.

4.6 Evaluation

To evaluate our data exploration technique, which we implemented in our prototype system 'Traverse', we conducted a formal user study. Through our user study, we wished to examine how participants used both targeted, cast-a-net and referential actions to express their interests and how they 'traversed' the data and attribute space.

4.6.1 Participants and Setup

We recruited 10 participants, 5 male and 5 female, ages 19 to 37. Participants were largely students in computer science and engineering, pursuing bachelor's (5), masters (1) and PhDs (4). Participants received a 20 dollar gift card as compensation for their time. This number of participants allowed us to observe a range of behavior, and gave us enough data to understand how our technique and system was used for different users, with different exploratory strategies.

Participants reported varied experience with data visualization and data analysis. They self reported frequency of using or creating visualizations (Daily-1, Weekly-2, Monthly-3, Yearly-3, Almost never-1), and their frequency in engaging in data analysis or exploration tasks (Daily-0, Weekly- 4, Monthly- 4, Yearly-1, Almost Never-1). Participants reported having experience with a range of tools for data visualization and data analysis, such as excel, Tableau, Power BI, google sheets, d3, and Observable notebooks.

Our evaluations were conducted through a video conferencing system (Zoom), due to Covid19 precautions. Participants were required to have a desktop or laptop computer, with the Chrome browser, and they reported their operating system (Windows-7, Mac- 2, Chromebook-1), and screen resolution (from 1366 x 768 to 3840 x 2160).

4.6.2 Procedure and Tasks

Our study had three phases: a training phase, a data exploration phase, and a survey/debriefing phase. The protocol was refined through 5 piloting sessions.

4.6.2.1 Training

Participants first filled out a consent form. Then they walked through a description of the interaction technique with the researcher. The technique was presented using simple graphics that explained the principles of expressing their interests and allowing the system to translate those interests into sets of views. They were told that the goal of the system was not to help them create a specific desired visualization, but rather to pose questions about parts of the data that they were interested in learning more about.

Participants were shown the interface and the researcher explained how the different parts of the interface worked. They were then able to test the interface and system, by entering a set of scripted examples pertaining to city of Chicago data. By using a script, we were able to ensure that participants had an opportunity to see the major direct and referential request types during the training phase. As responses to queries came in, the researcher would explain the responses, highlighting how the request was translated into a set of views. Participants were encouraged to ask questions during the training.

4.6.2.2 Task and Data

After the training, participants were given a new dataset and a prompt for exploring the data. The dataset pertained to US counties and health and demographic data related to COVID19. We chose this dataset because we felt it would be of interest to our participants, it reflected real datasets that could be targeted by our approach, and it had interesting features that our participants could uncover through exploration.

The data included a county-level measure called a 'COVID19 vulnerability index', which was a machine learning prediction of the risk of a county to COVID19 outbreaks, and to having insufficient resources to address this outbreak (136). Along with this risk prediction, we also included measures pertaining to health care (doctors per capita, uninsured rate), diseases associated with COVID19 (cardiovascular disease rates and diabetes rates), poverty rate and percent of the population over 65. Each county was also classified into a region (Midwest, Southwest, Southeast...etc), and a county type (rural, urban, suburban and small city). We also included a time series of COVID19 cases per month, for each county in the US. This data is described further in Appendix A.

The participants were given an exploratory task to examine differences between the regions in the dataset, and the county types, with respect to COVID19 vulnerability scores, and health and demographic data. They had a list of suggested sub-tasks to address, but they were encouraged to ask questions about their interests and observations, and to explore freely. Participants were allowed to explore until they had found interesting features within the data, and had suggestions for where to target resources to address COVID19 outbreaks, or to be prepared for potential future pandemics.

Following the study, participants were given a brief survey, and had the opportunity to discuss their experience with the researcher.

4.6.2.3 Captured Data

We logged participant queries, as well as the views and view collections Traverse produced in response to these actions We captured screen shots of their complete data exploration session, which shows both queries and responses. All sessions were also video recorded, through the video chat interface, which included the shared screen which depicted the participant's interface and webcam recordings of the participant and the experimenter. We conducted a survey at the end of the session. The researcher also took notes during the study.

4.7 Findings

An overview of the questions, visualization response types and length of the session is in Figure 21. On average, participants explored the data for 30 minutes (min: 17, max 49), and posed 35 queries (min 21, max 71). These queries generated an average of 56 unique combinations of data values and attributes (min- 31, max 137), and an average of 80 visualizations that were preserved by the user in their exploration history (min- 36, max- 162).

Participants utilized a mix of direct requests (63 percent of visualization requests) and referential requests (37 percent of visualization requests). Direct requests included targeted requests (27 percent of direct requests)–one set of filter criteria and one data attribute- or multifaceted requests (27 percent of direct requests)- combining 2 data attributes together, in

Category / Participant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total	Average
Minutes	34	17	49	43	29	38	35	21	23	47	336	**33.6**
Queries	25	21	49	32	20	26	68	22	23	71	357	**35.7**
Visualizations created and preserved	38	36	126	59	51	162	118	49	39	120	798	**79.8**
Unique data value, attr.combinations	31	32	79	26	31	137	73	28	35	93	565	**56.5**
Direct	13	10	29	17	16	23	30	17	18	36	209	20.9
Target	5	4	6	6	2	5	6	2	5	15	56	5.6
Multifaceted	3	2	7	4	4	4	5	9	7	11	56	5.6
Cast a net	5	4	16	7	10	14	19	6	6	10	97	9.7
Referential	11	11	18	14	3	0	29	4	5	29	124	12.4
Subset pivot	1	0	4	0	0	0	0	0	0	8	13	1.3
Attribute pivot	1	5	1	2	0	0	4	3	0	4	20	2
Filter	2	0	0	5	0	0	2	0	0	1	10	1
Split	0	0	0	1	0	0	2	0	0	4	7	0.7
Split pivot	2	1	3	1	0	0	2	0	2	7	18	1.8
Extend	5	4	10	5	3	0	17	1	3	5	53	5.3

Figure 21: Breakdown of participant sessions.

one view- as well as 'cast-a-net' requests, in which participants asked for many data values and attributes (46 percent). The proportions of actions are depicted in Figure 22. One question might be to consider how these breakdowns compare to those in the previous chapter. It is difficult to direct compare, because we did not classify subcategories of referential actions in Chapter 3, and we used a slightly different classification scheme to capture 'cast-a-net' requests. However, there were more referential requests than direct requests in our pre-design study, and the opposite is true here. This may be due to differences in the environment or due to how participants expressed themselves through speech versus a typed interface.

Using logged data, and the captured screenshots of the visualization history, we examined more closely how participants used these actions together, and how they traversed the data and attribute space.

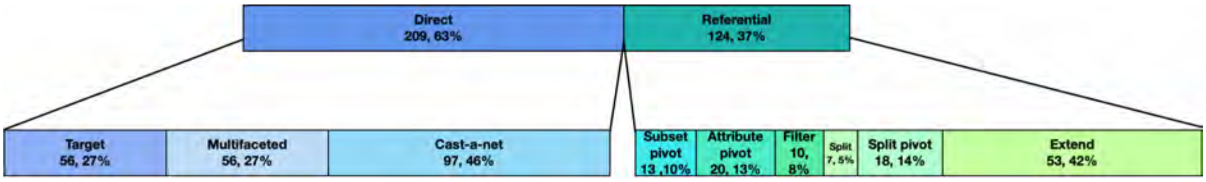


Figure 22: Breakdown of participant actions.

4.7.1 Analysis of exploratory actions

We analyzed the interplay between targeted, cast-a-net and referential requests, in Figure 23. This figure shows the sequence of actions taken by each participant, colored by whether the action was a direct or referential request, and with the number of views created visually indicated, which highlights focused queries versus ones that covered multiple data values and/or attributes at once.

From this visualization, we can see varying exploratory strategies. At one extreme, participant 6 did not use referential operations, and acted almost exclusively through cast-a-net requests. Looking at this participant's exploration history we could see that they posed repeated browsing cast-a-net requests, in which they selected regions of interest, and then viewed these regions with respect to multiple data attributes. Other participants, such as participant 3 and 10, used repeated referential actions to extend the focus of their investigation, and build large view collections, or sets of related view collections.

4.7.2 Analysis of breadth of exploration

We evaluated the breadth of exploration, through a representation that we call an 'exploratory grid' Figure 24. For each request posed by the user, we took the visualizations

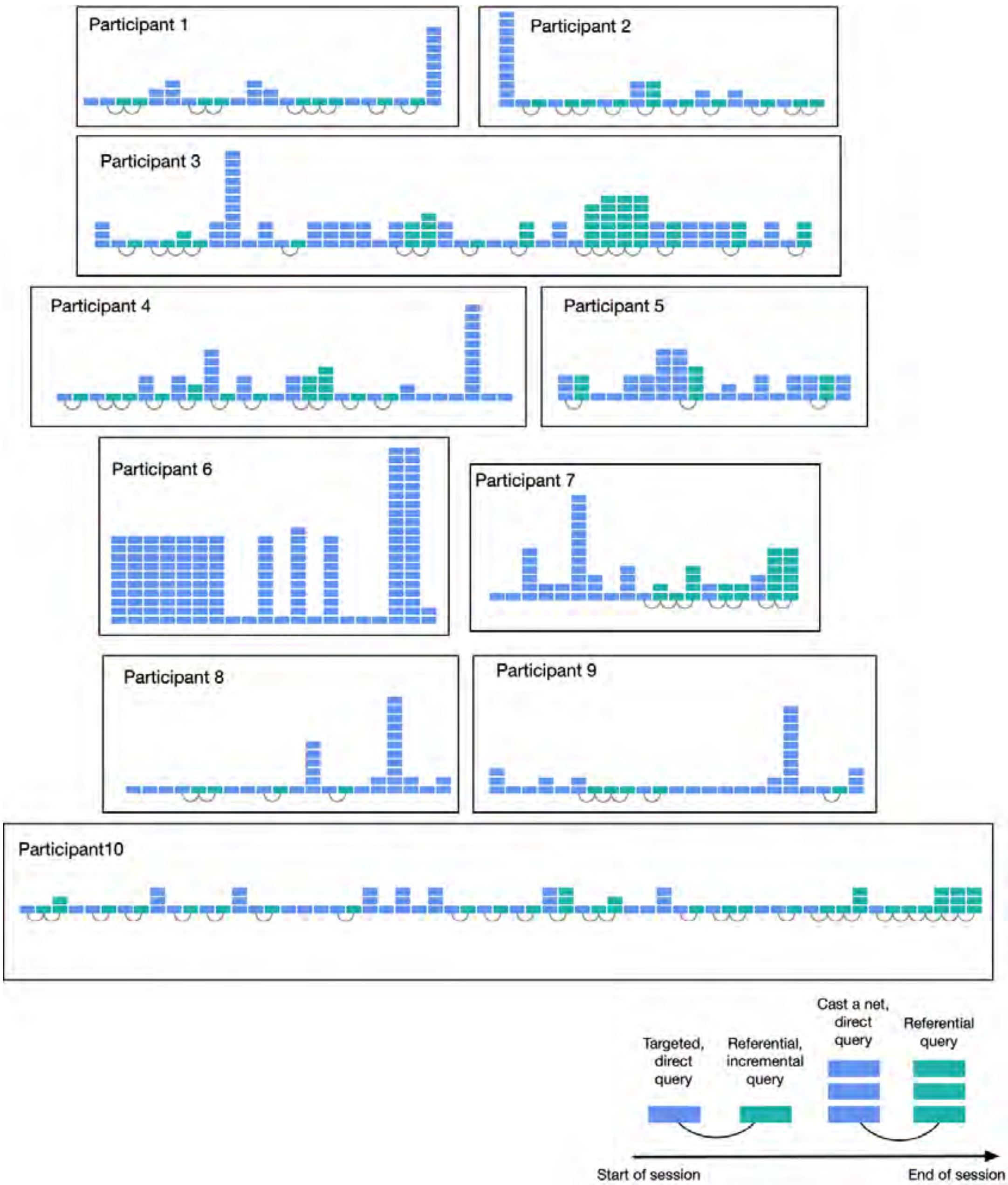


Figure 23: This figure shows how each participant used a combination of direct and referential actions to explore the data. Each participant's session begins on the left. Each visualization produced is shown as a rectangle. Blue rectangles are produced from a direct request. Green rectangles are visualizations produced from a referential request. If multiple rectangles are in a stack, they were produced together, from one query. The arcs connect the referenced view or view collection, to the new views that are created. A sequence of arcs, and a sequence of green rectangles, indicates that the participant repeatedly used a referential operation to expand or copy and pivot visualizations, expanding the focus of their exploration.

created by the system and added a ‘box’ to the grid, and assigned the rows in the grid to each unique filter criteria created during the session, and each unique single or pair of data attributes. As the participants retained a focus on a particular filter criteria, but added new data attributes, this would add new entries to the grid horizontally, filling in spaces from left to right over time. Alternatively, retaining a focus on a specific data attribute or pair of data attributes, but examining different filter criteria, would add entries vertically to the grid, filling in spaces from top to bottom over time. Repeated requests for a particular set of filter criteria, will add more squares to the assigned row, and the same for each data attribute’s column. Repeated requests for the same filter criteria and data attributes, would darken the color in each grid position. This occurs when participants recreate previous views of the data, in order to juxtapose this view in a new view collection, or in order to start a collection that they intended to build in a new exploratory direction.

Some participants explored mostly data attribute variations, focusing on overviews, and they did not drill down into different subsets of the data. This can be seen in participants with few columns but many rows in the exploratory grid. Other participants pursued a combination of data attribute and data value diversity, resulting in patches of filled in data values in the visualizations. In particular, participant 6, pursued a strategy which ’covered’ selected data values and attributes evenly. What we mean by this is that participant 6 produced a more evenly filled in grid, because they posed repeated browsing cast-a-net requests, which covered a set of selected filter criteria with respect to a consistent set of data attributes. In contrast, Participant 10 covered a wide range of data attributes, suggested by the length in the vertical

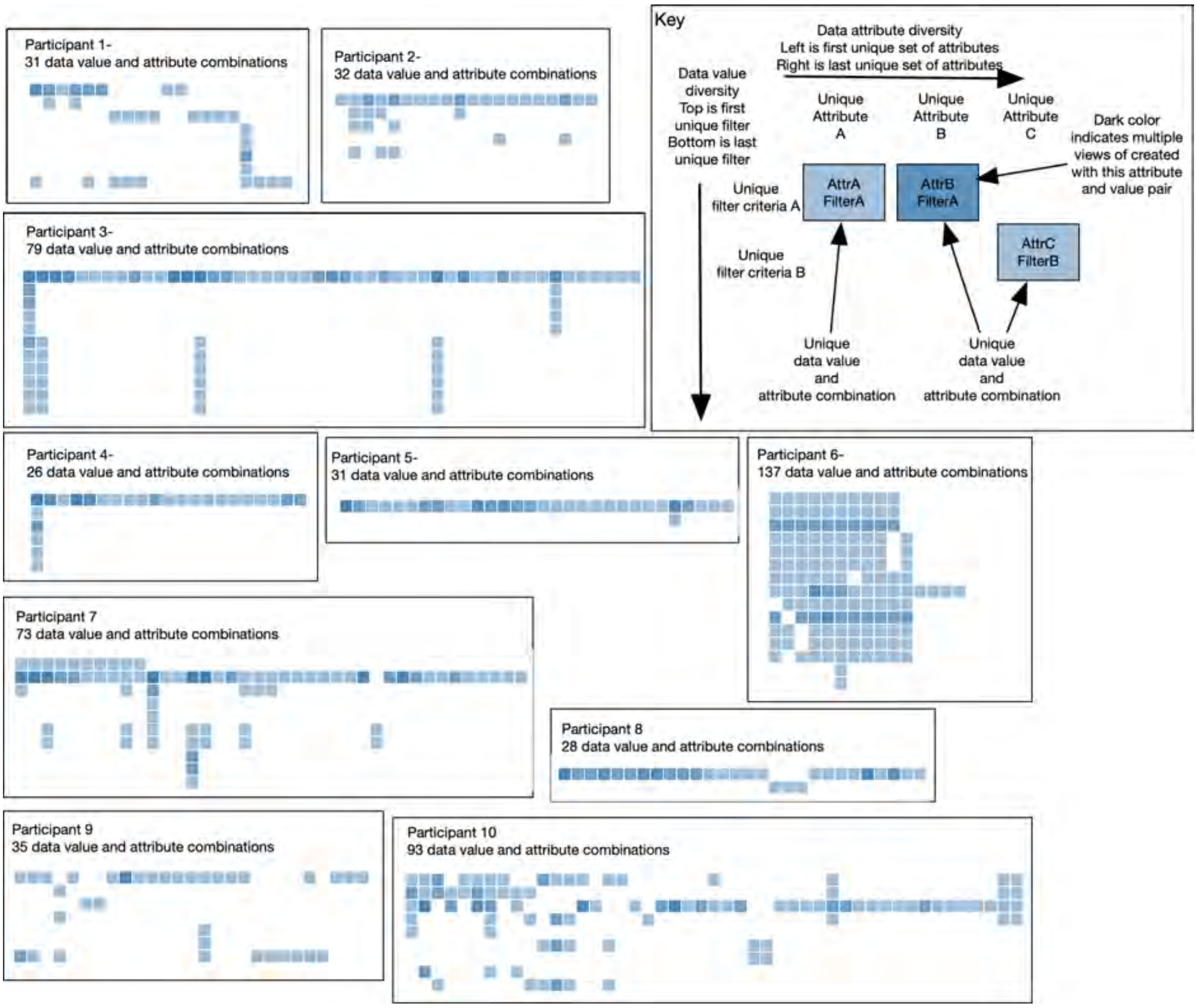


Figure 24: This figure shows how each participant explored unique sets of data values and attributes, over the course of their session. Along the x axis, for each participant, are unique data attributes (either one, or a pair) that traverse produced from the start (left) to the end (right) of the session. Along the y-axis, are unique filters, from the start of the session(top), to the end (bottom). Darker colored squares occur when a participant creates that data attribute and filter more than once.

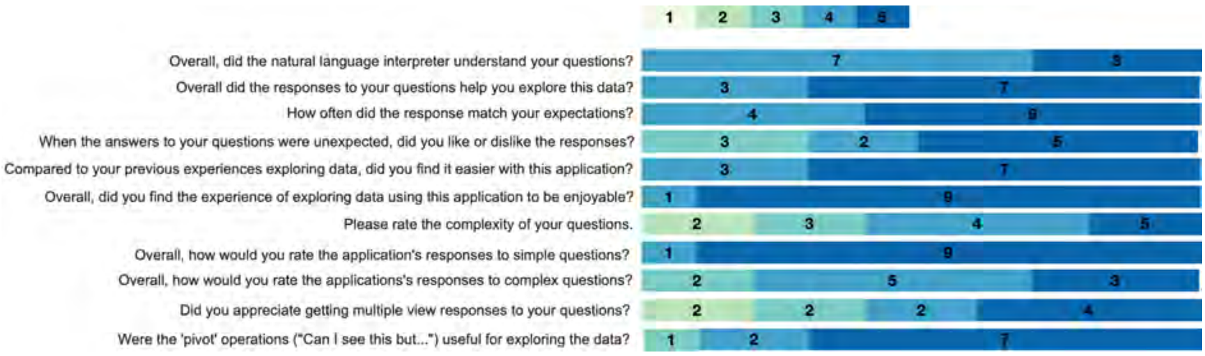


Figure 25: Participant Likert score responses to survey questions. 1 is strongly disagree, 5 is strongly agree. Each question is shown, next to a stacked bar. The color of each bar is based on the score. Width of the bars is based on the number of participants who responded with that score.

direction of their exploratory grid, and a variety of different filter criteria. Participant 3 similarly explored a wide range of selected data attributes, and a smaller set of varied filter criteria.

4.7.3 Qualitative and Quantitative Feedback

Participants responded to survey questions on their experience, and provided 1-5 Likert score response to express agreement (5) or disagreement(1). They also were given an area to type comments about their experience, and describe places where we could improve. Figure 25 displays the results of this survey, with participant scores.

4.7.3.1 Feedback on Interaction Using Natural Language

Participants ranked the natural language interpreter highly, indicating that it always (3/10) or almost always (7/10) understood their questions. They also ranked the experience as 'very enjoyable' (9/10) or 'somewhat enjoyable' (1/10). When asked how this compared to previous experiences they had exploring data, participants ranked it as 'significantly easier' (7/10) or 'somewhat easier' (3/10). These responses are echoed in participant comments. One participant stated *"I found it very convenient to type questions and get results. Writing queries in simple English was very helpful."*

Another participant described a benefit of NL for data exploration, in allowing them to remain focused on the data and their tasks, rather than on the mechanics of creating visualizations. This participant stated *"It was interesting to use and was fairly ergonomic to input commands while maintaining my train of thought."* One participant noted the benefits of NL for quickly generating visualizations, stating *'It was moderately expressive and very fast to generate a lot of charts.'*

4.7.3.2 Feedback on View Collection Responses

Participants also scored the responses to their questions, and felt that the responses were very helpful in exploring the data (7/10) or somewhat helpful (3/10). One participant noted that the ability to state their general interests and get visualizations in response was helpful when they were not sure what they were looking for- *"I liked that I could just type what I was thinking without having a real sense of what I was looking for, and the plots produced generally answered my vague question."* We asked participants about whether they felt the responses matched their expectations, and the answered that they did 'most of the time' (6/10) or 'some of the time' (4/10). Participants were mixed on whether they liked unexpected responses, with 5/10 'strongly liking' and 2/10 'somewhat liking' unexpected responses, and 3/10 feeling neutral about unexpected responses. One participant noted that they appreciated that the application generally attempted to give a response, saying *"I also liked how the application always generated a response instead of a simple "try again" error."*

4.7.3.3 Feedback on Referential Actions

Participants generally appreciated the referential operations, with 7/10 describing these as 'very useful', 2/10 as 'somewhat useful', and 1/10 as neutral. One participant stated *"I liked that you could easily switch the filter for a particular set of data (I want to see this but for...), that it was easy to compare plots/the data filtered by different attributes ..."* . Another commented *"I thought the features of the application were pretty amazing (mainly, being able to pivot)."*. A third commented *"It felt natural using this to explore data because of the way you can change the parameters of what data you are seeing"*.

4.7.3.4 Reflections on Exploratory Process

We asked participants to discuss issues around the complexity of the questions they could ask, and how they felt the application responded to simple and complex queries. They ranked the complexity of their questions as 'I asked very complex questions' (2/10), 'I asked somewhat complex questions' (4/10), 'neutral' (3/10) and I asked somewhat simple questions' (2/10). Participants strongly liked answers to simple questions, with 9/10 saying it answered simple questions 'very well', and 1/10 saying it answered these questions 'somewhat well'. Participants were mixed on how well the application responded to complex questions, saying it responded to complex questions very well (3/10), somewhat well (5/10) and neutral (2/10).

4.7.3.5 Feedback on Multi-View Responses

Participants had mixed impressions of situations where the application provided multiple view responses, ranking that they preferred multi-view responses 'strongly' (4/10), 'somewhat' (2/10), neutral (2/10) or preferred single view responses 'somewhat' (2/10). In cases where the response involved combinations of data attributes, specifically the complex response type, some participants during the study commented to the experimenter that they wished it only responded with the multifaceted, two-attribute views, rather than the set of monovariate and multifaceted views. This may be due to difficulties using the 'color by', 'versus' or 'split' language, or expressing this intention but in a format the interpreter did not understand. One participant said ' *I feel like when I wanted to compare two attributes, it would have been more helpful for the heat map to come up before the bar charts of the individual attributes."* Another commented *"I don't feel like I used .... "split by" very often, so I'm not sure how useful they are."* which suggests that this approach to adding multiple data attributes to a single chart may be difficult to understand for some users, but that the approach to give both monovariate and multifaceted responses may be unwelcome, for other users.

4.7.3.6 Feedback on Restricted NLI

Participants also expressed some thoughts about phrasing their questions using our restricted NLI. One participant commented *"I feel as if the application was a little constrained in terms of what you could ask and how you worded it."* Another commented *"Sometimes the verbiage and figuring out what you want to explore takes a bit to get used to."* Other participants echoed this difficulty in knowing what to explore, by asking for guidance from the interface.

One wanted a feature that *'suggests graphs or visuals other users have look at in case someone doesn't know where to start.'* This interest suggests a potential future area of research, which is seen in other work on visualization guidance and recommendations (6; 1)

4.7.3.7 Feedback on Uses for Collaboration

Finally, although this study targeted single user exploration, one participant suggests the value of this approach for collaborative, synchronous data exploration, by noting "*This experience is far nicer and would be much more convenient during a presentation or a meeting when there are questions and I want to be able to answer them in a visualization quickly.*"

4.8 Discussion

Participants largely were able to use our novel NL data exploration technique to explore the data. They used multiple request types, toggling between direct and referential actions, as well as targeted and cast a net approaches. Participants gave largely positive feedback with respect to the interaction approach, and found referential copy and pivot actions to be easy to use in exploring the data.

4.8.1 Limitations

There were several limitations in our approach to responding to articulated data value and attribute interests, that would benefit from future work.

One particular difficulty involved resolving multiple data attribute requests. Our response to this request type contained combinations of parameters, which means that it often also contained views they did not need. For instance, some participants did not want to include visualizations that depicted more than one attribute, and others expressed that they only wanted to see views with the pairs of data attributes. When these participants in our evaluation wanted to pivot this collection, they often expressed that they did not want to see the undesired views. One option would be to respond to some categories of requests, such as multiple data attributes of interest within multiple response types, and let users select one to keep in the history. This approach might be worth considering for other request types as well, to generally allow for more selection in the options provided to users, while still allowing users to express data interests rather than design interests, using simple commands.

A second limitation, concerns design variations. Although we did not aim to provide design variations, there are contexts where alternate representations would be valuable. But, view collections would become crowded, and the coherent structure obscured, if we responded with alternate representations to user interests. However, perhaps a secondary display area could include alternative representations, that could be selected by the user.

Finally, some participants expressed difficulty in situations where they wanted to achieve a specific intended visualization, which was not a design goal for this work. However, it is possible that this approach could complimented by other techniques to create visualizations.

4.8.2 Future opportunities

Some participants expressed a desire to interact within a window, rather than just produce new views of the data. For future work, this could be an interesting area to explore, particularly looking at how the view collection formalism impacts multiple coordinated view approaches (64; 39; 138; 102).

Participants also suggested that this approach could be valuable in a collaborative context, since they could explore data without breaking away from the conversation. In a collaborative context, there are additional questions around co-developing these groups. While this interface provided a linear history, in a collaborative context it might be valuable to consider a branching history. Alternatively, this approach could adapted for flexible canvas environments which give more freedom in terms of the exploratory process, which might be more compatible with collaborative scenarios.

4.8.3 Accommodating new data

As we noted in our design, we focused on visualization types that presented one or several views of the data, and did not consider view types with three of more attribute interests. Since participants had difficult deciding whether to pivot a primary or secondary attribute using our restricted NLI, future work is needed to determine a cleared approach to communicate data attribute pivots using NL. One approach could involve iteratively adding data attributes, rather than directly specifying a primary and secondary attribute. This would mean that some view collection types would need to be created in multiple NL commands, the first to create views based on a primary data attribute, and the second to add in a secondary data attribute, and then a third step to add in a third data attribute. This is an area worth pursuing through future research.

4.9 Conclusion

Since the motivation for this technique grew from observations in the previous chapter, where participants posed requests in a large display environment, with a flexible canvas to juxtapose many views of data, in the next Chapter I will describe how we adapted this technique to a large display environment, with a point-and-speak interaction approach.

CHAPTER 5

DITTO: A MULTI-MODAL SPEECH AND MID-AIR POINTING TECHNIQUE FOR VISUAL DATA EXPLORATION IN LARGE DISPLAY ENVIRONMENTS

This chapter describes a novel interaction technique with multi-modal speech and mid-air gesture interactions for large displays, motivated by the findings presented previously, and using the Traverse view collection generation approach described in the previous chapter.

This research has been collaborative, but I am the project lead for this work. Abeer Alsaiari (AA) helped to develop the multi-modal interaction system, in particular resolving pointing gestures. She also assisted with testing of the system and running the evaluation. Arthur Nishimoto helped with integrating inputs from the Omicron system, and developed the touch interface. Abhinav Kumar assisted in developing the speech input system, and helped to debug and test different speech input approaches. Andrew Johnson (AJ) and Barbara Di Eugenio (BDE) has assisted in the scope and direction of the project. As in the previous chapter, Moira Zellner (MZ) and Anuj Tiwari (AT) provided a dataset and decision problem, which enabled me to expand and test the design of the technique, and also develop a compelling use case for evaluation. A number of undergraduate researchers also contributed to developing the input system: Vasanna Nguyen, Krupa Patel, Ryan Fogarty, Joseph Borowicz, and Vijay Mahida.

5.1 Introduction

Visual data exploration is an iterative process, where an analyst toggles between targeted queries, open-ended exploration, and iterative steps from current points of interest to new ones, based on observations or evolving exploratory goals (4; 85; 5). During this process, an analyst may produce many views of their data, that show diverse selections of data values and data attributes, and that partition a dataset into meaningful pieces (39; 30). After observing one set of interests, an analyst may wish to pivot to a new set of interests, and juxtapose the resulting visualizations with the previous ones, so they can perform tasks that span more than one view, such as comparing or putting a focused observation into context. Through multiple rounds of view creation, based on evolving points of interest, an analyst gradually expands the focus of their exploration and incrementally moves through the 'data and attribute space' of a large dataset (6; 1). To support this process, analysts need techniques for rapid, on-the-fly view creation, based on data values and attributes of interest, and an environment for displaying and juxtaposing multiple views in configurations that support evolving exploratory tasks (38; 35; 39; 44).

Large display environments present many potential benefits for visual data exploration (44), because they provide abundant space for displaying and juxtaposing the multiple views produced during the exploratory process (38; 35; 42). However, interaction in a large display environments presents significant challenges, and recent research has considered ways to enable interaction using modalities beyond mouse and keyboard inputs, such as touch (107), proxemic interaction (66), and external devices (104; 105; 106; 94; 100; 95).

In this chapter, **we present Ditto, an interaction technique for generating many views of data in visual data exploration on large displays using synchronous speech and mid-air pointing gestures**. Our technique aims to enable analysts to efficiently create views of data around multiple, evolving points of interest in data exploration. We couple these multi-modal view creation actions with touch for view organization on the display, which allows analysts to juxtapose related views, in support of tasks that span more than one view of the data. We use a set of 'referential view creation actions', based on the work in the prior chapter, which are initiated by combined speech and mid-air pointing interactions. These referential actions take advantage of the potential of users to organize views into conceptually meaningful groups on the display. We named this system 'Ditto' because a core feature of our multi-modal speech and mid-air gesture technique is an efficient set of **copy-and-pivot actions, that enable efficient, iterative steps from old points of interest to new ones, in a way that leverages the spatial positioning decisions of users**. The result is a technique for **rapid, on-the-fly creation of views around diverse combinations of data values and attributes, to support evolving data exploration goals.**

Our primary contributions are:

- Design: We discuss design considerations for multi-modal speech and mid-air pointing interactions for large display environments, to target data exploration
  - We describe how our technique considers the interplay between synchronous speaking and pointing in communicating visual data exploration intentions, and leverages abundant display space and human spatial organizational capabilities.

  - We propose a technique which combines: 1) direct requests around data values and data attributes of interests, 2) view positioning, into meaningful configurations, and 3) referential 'copy-and-pivot' requests, to iteratively extend reach of the user's exploration into new portions of the 'data and attribute space'.

- Application: We present an implemented environment and application, called Ditto, which utilizes our design, and visualizes two data sets, drawn from real world use cases. The prototype implementation allows us to study exploratory actions of users, both their requests, their view positioning decisions, and how they used speech and mid-air pointing gestures together to express their intentions in data exploration.

- User study: we present observations and findings from an empirical study, with recruited participants, who explore a dataset in the large display environment with multi-modal speech and mid-air gesture inputs. We discuss how users group and then reference prior views to create new ones, and how their strategies relate to evolving data exploration interests.

In this chapter, I will described our work towards designing and implementing this technique, informed by the findings in our observational study in Chapter 3, and using the natural language data exploration technique developed in Chapter 4.

5.2 Background

5.2.1 Large Displays, Visualization and Interaction

Large display environments present many potential benefits for visual data exploration (44). First, large displays have abundant space that is capable of displaying many views of data at once, at sufficient resolution (42). Second, large displays can serve as a large, flexible canvas for creating custom arrangements of visualizations, in configurations that support reasoning over the varied selections and combinations of data values and attributes produced during the exploratory process (38; 35; 42). These features of large displays- abundant display space and human spatial organizational capabilities- (39) have been examined with respect to supporting sensemaking over large volumes of analysis artifacts (35), for visualizing between-view relations through metavisualization (102), and for research in coordinated multiple views (39).

However, interaction with large display environments presents challenges. Traditional modalities for interaction in visualization, such as mouse and keyboard, are ill-suited to large displays, because they require a user to be tethered at one location (36; 37; 88; 89). This limits mobility and reduces the opportunity to fully utilize the display space. As a result, there is growing interest in input modalities for large displays beyond mouse and keyboard. In the data visualization research community, researchers have examined touch (107; 108; 33), proxemic interaction (66), and separate devices (104; 105; 106; 94; 100; 95). Since these modalities have varied affordances, there are many open questions around how to select and combine multiple input modalities together in ways that leverage the benefits of large displays, for diverse visualization tasks (41).

5.2.2 Gestures, proxemics and multi-modal interaction

(130) suggested an implicit interaction with visualizations on large displays using body position, orientation, and movement. These proxemic attributes can be used to implicitly trigger specific tasks like zooming, selection and navigation. (127) designed a hybrid approach that utilizes both proxemics and gestures and found that gestural interaction is suitable for direct actions while proxemics interaction is beneficial for tasks such as navigation and collaboration. In both studies, proxemic inputs and mid-air gestures are employed for a limited set of visualization tasks, and are not envisioned as a larger part of the visualization process, which is the focus of our study. Our work does not directly examine proxemics, but mid-air gestures as explicit interactions with the system, in concert with speech.

Multi-modal natural language and gesture has been examined in 'Put that there' (46) in 1980, describing a system which allowed users to place objects on a large display using a mid-air pointer, controlled through gesture, and spoken inputs, which was echoed in Hauptmann et al's work in 1989 (109).

5.3 Motivating Scenario

To motivate our design, we consider an exploratory data analysis scenario in a large display space where users could make use of multi-modal speech and mid-air pointing gestures for exploratory interactions. This usage scenario will help define the requirements for our interaction technique. We will describe this scenario using the dataset about crimes in a major US city, as in prior Chapters, where an analyst has been tasked with exploring the data and identifying differences in crime patterns between neighborhoods. This data consists in a list of crime incidents, with fields that include- the neighborhood name, the crime type (theft, burglary...), the location type (street, sidewalk...), the day of the week, the time of the day, the month of the year, and the year. This dataset is depicted in Figure 9.

Suppose the analyst wishes to explore the data in the crimes dataset, beginning by looking at their neighborhood- the university neighborhood. They may want to first understand which types of crime are most common in the university neighborhood. Then, they may want to look at what time of the day crimes typically occur in that neighborhood, followed by what day of the week. These three questions focus on a common exploratory thread- interest in the University neighborhood. In effect, they want to retain the focus on the university neighborhood, but pivot their focus to new data attributes.

After creating these three views of the data- one for each data attribute, and all filtered to just show crimes near the university- they might position the views in the center of the display, surveying to see if any significant features stand out. They make three observations: 1) crime is low late at night and in the early morning, and then rises over the day, peaking at noon, 2) crime peaks on Friday near the university, but is otherwise relatively stable, and 3) thefts are most common in the university neighborhood.

The analyst may then wonder if these observed features in the data are shared by the downtown neighborhood, which is adjacent to the university neighborhood. They would want to take the same plots but see *all of them* for the new filter (neighborhood=downtown). They would want to *duplicate each view, and swap one filter criteria for another* (1). These three new plots might then be situated by the user next to the prior ones, in a grid of views, with rows defined by the different neighborhoods and columns defined by the different data attributes (139; 38).

Suppose the analyst wanted to continue comparing neighborhoods, but with respect to two new data attributes, such as the month of the year that crimes occur, and the location type (street, sidewalk, etc). In effect, they want to retain two points of focus (the two neighborhoods) and pivot to look at two new data attributes. They would want to select two of the views, and *duplicate these but swap data attributes* (1), creating 4 new views of the data, that can be added to the existing grid view at the center of the display.

Their exploratory process may then continues in a new direction, looking at particular crime types or location types, and they create new sets of views in support of these questions. Then, after a time, they may wish to return to their initial thread- comparing neighborhoods- and take this inquiry in a new direction. They might, for example, return back and ask to see the grid of views and want to retain the current data values and attributes represented in this set, but *add new filter criteria* (1)- thefts on Friday- based on their new findings. They want to retain the focus on the two neighborhoods, and the data attributes, but they want to pivot to add new points of interests. The resulting views could be placed in a separate grid, or could be juxtaposed with respect to either collection of views on the display.

This process might repeat itself, where analysts wish to explore the data with respect to different exploratory threads. They want to retain aspects of prior views, but copy and pivot these views to new points of interest. By preserving prior views of the data, arranged in coherent groupings on the display, they can follow multiple threads, and keep their exploration history and findings visible.

There are aspects of this process that would be difficult to accomplish on a small display, where it is difficult to display more than a few visualizations at once, at sufficient resolution. However, on the large display, **an interaction approach is needed to support view creation, in particular the action of duplicating and pivoting views to new points of interest.** Otherwise, the analyst would need to engage in lots of view construction actions, which can be time consuming and error prone (12).

5.4 Design Considerations

In this section, we use the motivating scenario described previously, to generate a set of design considerations, focusing on the combined affordances of speaking, mid-air pointing gestures, and organizing views on a large display. Several of these design goals are shared by our work in the previous chapter, but I will describe them below, in the context of the large display and multi-modal interaction environment.

5.4.1 Design Goal 1: Focus on enabling breadth in data exploration, and evolving points of interest

In our multi-modal speech and mid-air gesture interaction technique, we focus on data exploration. In particular, we focus on initial exploration of a dataset, where a user wants to gain basic familiarity with the features of the data, and select potential targets for a subsequent focused analysis. We therefore focus on allowing users to express their interests in particular data values and attributes that they wish to explore, and we designed our system to translate these interests automatically into appropriate views of the data.

We specifically focus on enabling breadth in data exploration through multi-modal speech and mid-air gestures on the large display. By exploration breadth, we mean enabling users to examine many views with diverse data values and attributes, rather than design breadth, where many varieties of visual encodings are provided to the user. This is our focus for several reasons. First, it is a good target for large displays. Large displays are capable of showing many views of data, at high resolution, and they present benefits in contexts where users can spatially organize views into conceptually related groups, for sensemaking tasks (35; 38; 44). This fits with the demands of data exploration, where users consider many views with 'data variations'- or varied combinations of filter criteria and data attributes. Second, data exploration is an excellent target for natural language interactions, since it allows users to directly express their interests, without learning a complex graphical interface (12; 41)

We also focus on evolving points of interest in data exploration. Enabling users to generate diverse views in support of following evolving threads of interest, is a good target for large displays, where visualizations from prior threads can be persistently accessed and displayed (30; 85).

We do not focus on interactions in support of view coordination in this work. The reason for this is that it is difficult to precisely target entities within a visualization- such as visual entities like bars or lines- using mid-air pointing gestures at a distance (49; 41). This is better suited to other interaction modalities, such as touch (107; 33), where users can interact up close to the display with precision, or through a secondary device (94). We also do not focus explicitly on guidance or recommendation in support of data exploration (6; 1), though there are avenues for this contribution in future work, which we touch on in the discussion section.

5.4.2 Design Goal 2: Leverage combined features of mid-air pointing gestures, NL commands, and large display areas

We focus in this Chapter on enabling users to explore data by communicating their intentions through speaking, and through synchronous speaking and pointing toward visualizations on a large display. There is prior work examining NL interaction for visualization (14; 15; 16; 17; 18; 19; 20; 21; 24; 25; 26; 27; 28; 29), as well as prior work examining physical movement for large display interaction (66; 130), and prior work on using large displays in information visualization (129; 97; 42; 92; 43; 90; 96; 95). Considering multiple factors together presents an opportunity to explore novel interactions.

Items on a large display are effectively in our world at human scale (42), and we routinely point to items in real physical spaces when communicating our intentions to others. So, it is reasonable to consider how this real-world interaction style can be ported to visual data exploration for a large display. In addition, this style of interaction may be conducive to collaborative contexts, where users may want to create visualizations of their data without disrupting the flow of the collaborative discussion. Although we do not explicitly target collaboration in this work, it is reasonable to explore interaction approaches that are compatible with potential future use cases.

5.4.3 Design Goal 3: Design to enable multi-view exploratory tasks

Our design focuses on providing multiple view responses to situations where users express multiple data value and attribute interests. We make this a design goal because users may express multiple points of interests in data exploration (38). In our previous work, we found that users in a large display context frequently expressed a desire for multiple views of data, through expressing many data values and attributes interests in a single request. Responding to these kinds of requests with multiple views, with diverse but coherent combinations of data values and attributes, is an excellent target for large display environments. The display can accommodate many views, and if these views have coherent similarities and differences they can be arranged by the user in configurations that support tasks across the set of views (44).

This design goal is appropriate because it targets the large display and multi-modal speech+mid-air gesture interactions, that are the focus of this work. Systems that aim to help users arrive at an optimal single view response are better suited to small displays, rather than large displays. Single view responses are also better suited to contexts where users have a specific desired visualization in mind, rather than for an exploratory data analysis scenario. In addition, this approach leverages the affordances of spoken interactions, where users frequently express vague or general questions that are best responded to with multiple views. Speech also may not be the best approach for contexts where users have a highly specific intended visualization in mind, and a graphical interface or sketching interface might be more appropriate.

5.4.4 Design Goal 4: Leverage human spatial organization for collective copy-and-pivot referential actions

Large displays provide users with a large flexible canvas for positioning many views of data, and there are suggestions that users organize views on the display to reflect meaningful relationships or groupings that aid in sensemaking (35; 38). One of our design goals is to take this feature of large displays into account, but use it to enable multi-modal speech and mid-air pointing interactions.

Our technique allows users to indicate groups of views on the display through pointing, and pose spoken requests for new views of data based on the indicated views. This 'copy-and-pivot' approach has been explored for "flexible canvas" style systems (33; 30; 32; 107), but because these systems target mouse and keyboard, touch, or pen-and-touch interactions, they typically allow users to target a single view at a time. However, through speech and mid-air pointing gestures users are capable of indicating several views at once, particularly if they are located near each other on the large display. **Our interaction technique allows users to gesture over a set of views, and pose requests that copy-and-pivot the indicated set of views collectively, not just individually.** We will describe the mechanics for enabling users to perform these referential requests, to target, and then ask to copy and pivot sets of views with similar features that have been positioned together.

Spatial organization has typically been considered as beneficial for reasoning and sensemaking on large displays (35; 38). In our technique we focus on how positioning decisions by users can also aid in view creation actions for data exploration.

One of the consequences of this design goal, is that we do not provide algorithmic positioning options, or provide fixed view locations. Supplying these options would potentially constrain or overly guide user choices, and we wanted to capture their view positioning decisions in relation to our point-and-speak interaction technique.

5.5 Design

After presenting our design goals for point-and-speak data exploration interactions in the previous section, we will now present our interaction technique, which is used in our implemented system 'Ditto'.

We envision this technique working for large display walls, that can accommodate 50+ visualizations, with an aspect ratio that accommodates multiple groupings of visualizations, based on user interests. We also assume the room allows for movement in front of the display, and is capable of tracking pointing gestures, from a distance of 1-4 meters from the display, and capabilities to respond to touch interactions for view positioning. Details of our implementation and physical environment are in the next section, but these are the assumed environment characteristics for our technique.

There are three main components of our technique:

- Direct requests to create visualizations, based on user expressed points of interest in the data. We focus on responding to both focused requests, which we call 'targeted requests', as well as 'cast a net' requests, which include open-ended queries and queries that express multiple simultaneous points of interest. This component is accomplished through speech only. These queries are comparable to the direct requests described in the previous chapter, for the Traverse system.

- User spatial organization of views into conceptually meaningful arrangements. These actions are accomplished through touch gestures on the large display.
- Referential 'copy and pivot' actions, to use existing views as templates for new views, allowing users to efficiently specify their intentions, and incrementally extend the focus of their exploration. These actions are performed through synchronous spoken queries and mid-air pointing gestures. We accommodate both referential actions that target a single view, and referential actions that target many views at once. If the request is for several views at once, we determine how these views are similar in order to properly respond to the participant's request.

These approaches are drawn from our prior observational study (44), the previous chapter developing Traverse, and the exploration actions from Lee et al. (1). We also consider the multi-modal affordances for immersive analytics, described in Badam et al (41).

Although I have described some of these actions previously, In the next section, I will describe each of these action types in more detail, relate these to our design considerations, particularly how they utilize the large display and the multi-modal speech and mid-air gesture inputs, and I will explain how they address varied data exploration tasks.

5.5.1 Direct Requests

Visualization construction can be challenging and error prone (12), and there is extensive work on creating individual views through graphical interfaces (53; 71; 58; 59), sketch (131), or through NL inputs (14; 15; 16; 17; 18; 19; 20; 21; 23; 22; 24; 25; 26; 27; 28).

In a large display context, view construction is also challenging. First, the user needs to be able to get visualizations onto the display, ideally without being tethered to a fixed device. In many cases in prior work, researchers focus on interactions when visualizations are already present on the display, and they do not consider how the views are created and organized (106; 94). This suggests that visualization creation for large display environments is a potentially under-explored topic. In our design, we aim to reduce the barrier to creating visualizations on the display, by allowing users to directly express their interests in the data through speech, by listing data values and attributes that a user wants to explore, and we then use Traverse to respond, and populate the display with one or several visualizations.

Spoken interactions allow users to enumerate multiple points of interest- both data values and attributes. We use the approach described in Traverse, to respond to these multiple points of interest with sets of views that contain coherent data value and attribute variations. As in the previous chapter, we term these sets 'view collections', to highlight that they contain multiple views and at times have coherent between-view relationships.

We made a few changes to this technique for Ditto. First, we added some additional options for expressing multiple data attributes, based on user feedback on Traverse. Users can ask questions such as "Can I see (data attribute) for each of the (data attribute)", which we found was a desired interaction approach in our evaluation in the previous chapter. Second, we adapted our response to queries with multiple data attributes. If a participant asked, "Can I see (data attribute) and (data attribute)", we opted to default to a list of views for each data attribute, and would only give the multifaceted responses with multiple data attributes when participants asked to see views 'colored by' or 'split by' or 'versus' a secondary parameter. These changes were made in response to user feedback, and to accommodate the new environment.

5.5.2 Referential "Copy and Pivot" Commands

Ditto enables copy-pivot actions that combine speaking and pointing interactions. This work builds on our pre-design study in Chapter 3, where participants would indicate a prior view and say 'Can I see this but..." for a new data value and/or data attribute. The user's intention in this request was to make a new visualization that retained features from the view they targeted, but with a change in filter or data attributes. These actions to make small changes to the data values and attributes in a target visualization or set of visualizations, are part of incremental exploratory tasks, and are described the typology of Lee et al. (1).

There has been interest in previous work in externalizing the exploration process, through large, flexible canvas systems where a 'trail' of visualizations are displayed, and each view reflects these incremental steps. In these systems, users can take an existing visualization as a starting point, and then duplicate it using a graphical interface, and make modifications to the copy. This is a helpful interaction technique for data exploration, because it allows the user to pivot from one set of interests to another. The new visualization can be juxtaposed with the prior visualization, and used for tasks that span more than one view (30; 107). We apply this approach in our technique, with some adaptations for the environment and input modality.

5.5.3 Referential actions: multiple targets, multiple pivots

In our design, we extend this copy and pivot idea to multi-modal speech and mid-air gesture inputs. Through referential actions, we enable a user to indicate a view of interest through mid-air pointing, and then pose their request to see this view but with a change in the data represented in the view. In addition, we enable referential actions that target more than one visualization. We allow users to specify multiple view targets and/or to specify multiple parallel actions (pivots) to perform on the targets, to copy and pivot many views at once, or to copy and pivot a single view repeatedly in one command. To target multiple views, a user has to point to more than one visualization. To express multiple actions, a user needs to enumerate multiple data value and/or data attribute interests.

We can classify referential actions as:

- **'One-to-One',** for a single view target and a single pivot actions
- **'One-to-Many'**, for a single view target, and multiple pivots, producing multiple views
- **'Many-to-One'**, for multiple view targets and one new view, which extends the collection by one view
- **'Many-to-Many'**, for multiple view targets, with one or multiple collective pivots

Some examples of these referential action types are shown in Figure 26.

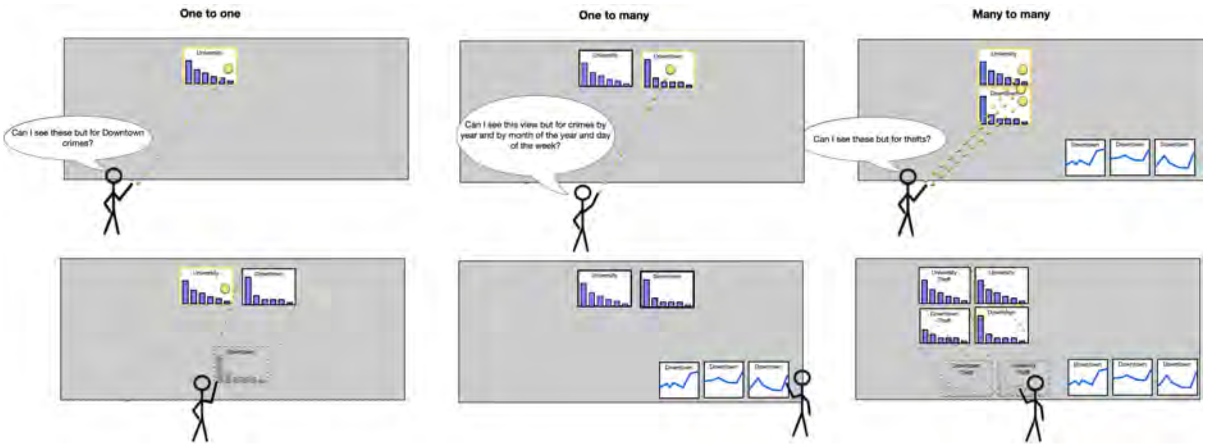


Figure 26: Figure depicting referential interactions of different types. Some target a single view, to produce a single new view (One to One). Others target one view, but express multiple pivots producing multiple views (One to Many) Finally, some referential interactions target several views and express a collective copy and pivot action on the targeted set (Many to many).

In the previous chapter, we described the mechanics for executing these actions. Here we add in the ability for users to create their own sets of views, that can copied and pivoted. This is enabled by spatial positioning decisions, described below.

5.5.4 Role of spatial positioning

Andrews et al. established that analysts examining large numbers of text documents tended to use spatial positioning to group related items together, in ways that reflected conceptual relationships (35) Applying this principle to information visualization, has been of recent interest (38; 102; 39). Typically, these groupings are seen as a way to help build an understanding of the data, because users can freely juxtapose and arrange views, in configurations that allow for tasks that span more than one view, such as comparisons.

In this dissertation, we use spatial positioning as a feature to enable view creation for data exploration, an example of which is shown in Figure 27. Items near each other are easier target with a mid-air pointing gesture. These items may be produced together, for instance in a cast-a-net direct request. They may be produced completely independent of each other, through separate direct requests. Or they may be from a referential request, to copy and pivot a prior view (which we will describe in the next section). In all of these cases, we needed to design actions to take custom sets of views, that a user may target, and provide an appropriate response.

Because users can freely position visualizations on the display, they can create custom view collections and request collective pivots on these self-defined view collections. This differentiates our interaction technique from Traverse, where views were created in collections in temporal order, and it was not possible to freely return to prior collections or individual views.

In Ditto, users can position views that were created at different stages of the analysis together. They can then refer to these views together, to copy and pivot them at the same time. Because users can cluster views together based on conceptual relationships, not their order of creation, we aim to allow them to move between threads of their analysis more easily. In effect they can explore one set of interests, and then another, and return to the first thread later. Or weave together multiple analysis threads.

5.5.5 How to copy and pivot self-defined collections

After users have requested an action on a self-defined set of views, our system needs to determine what features these views have in common. These shared features provide informa-

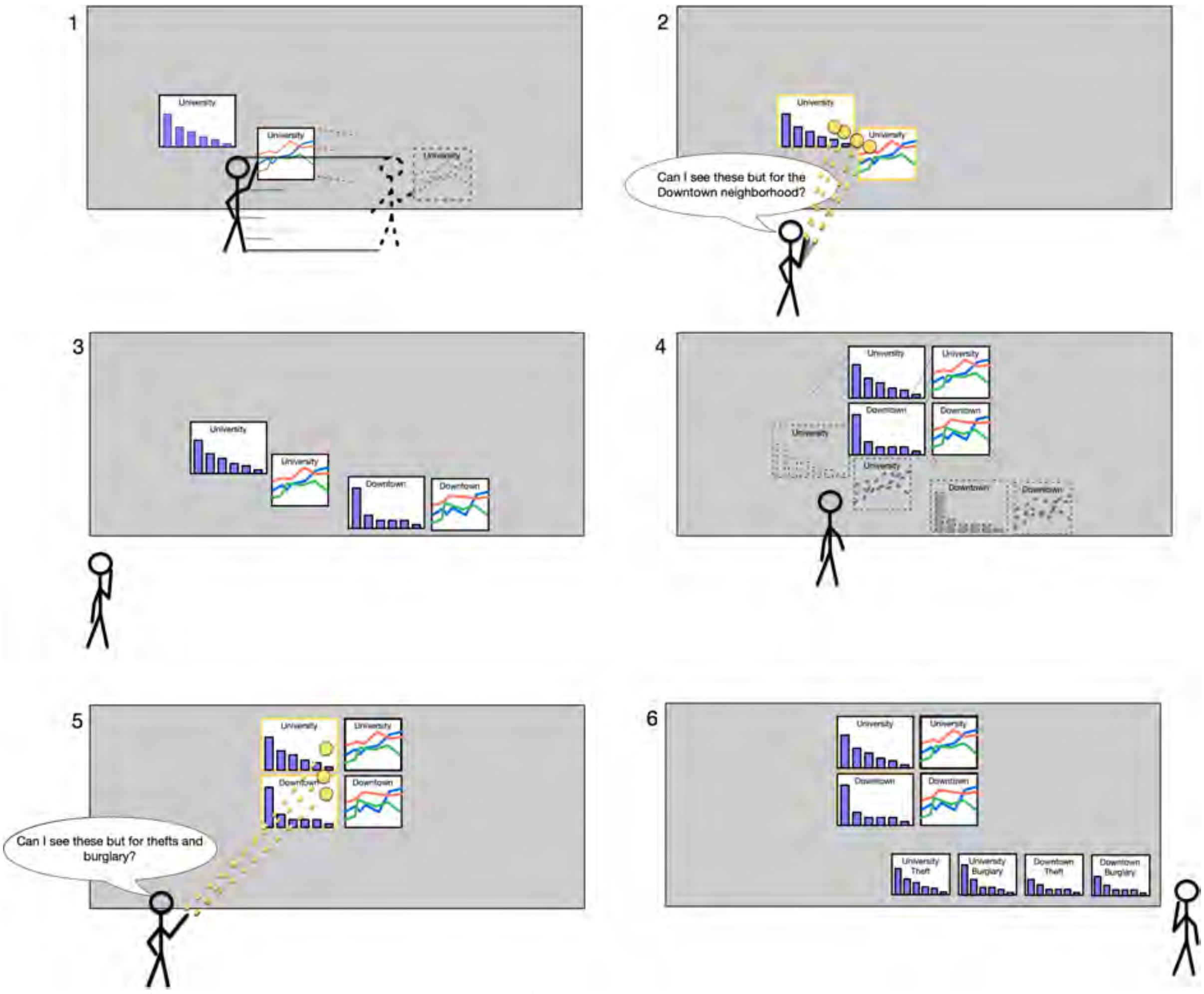


Figure 27: Spatial positioning decisions of the user can enable our technique, because adjacent visualizations are easy to point to. The figures are ordered 1 to 6. In the first scene, the user moves one the visualization closer to another, both figures are about the university (1). Then, they can reference this pair of views to copy and pivot them to a new subset of the data, the Downtown neighborhood (2 and 3). Then, these views can be grouped together in a grid (4). Then, in this grouping, the user can point to views with a common data attribute (the bar charts), and pivot them to a new set of data values (5 and 6).

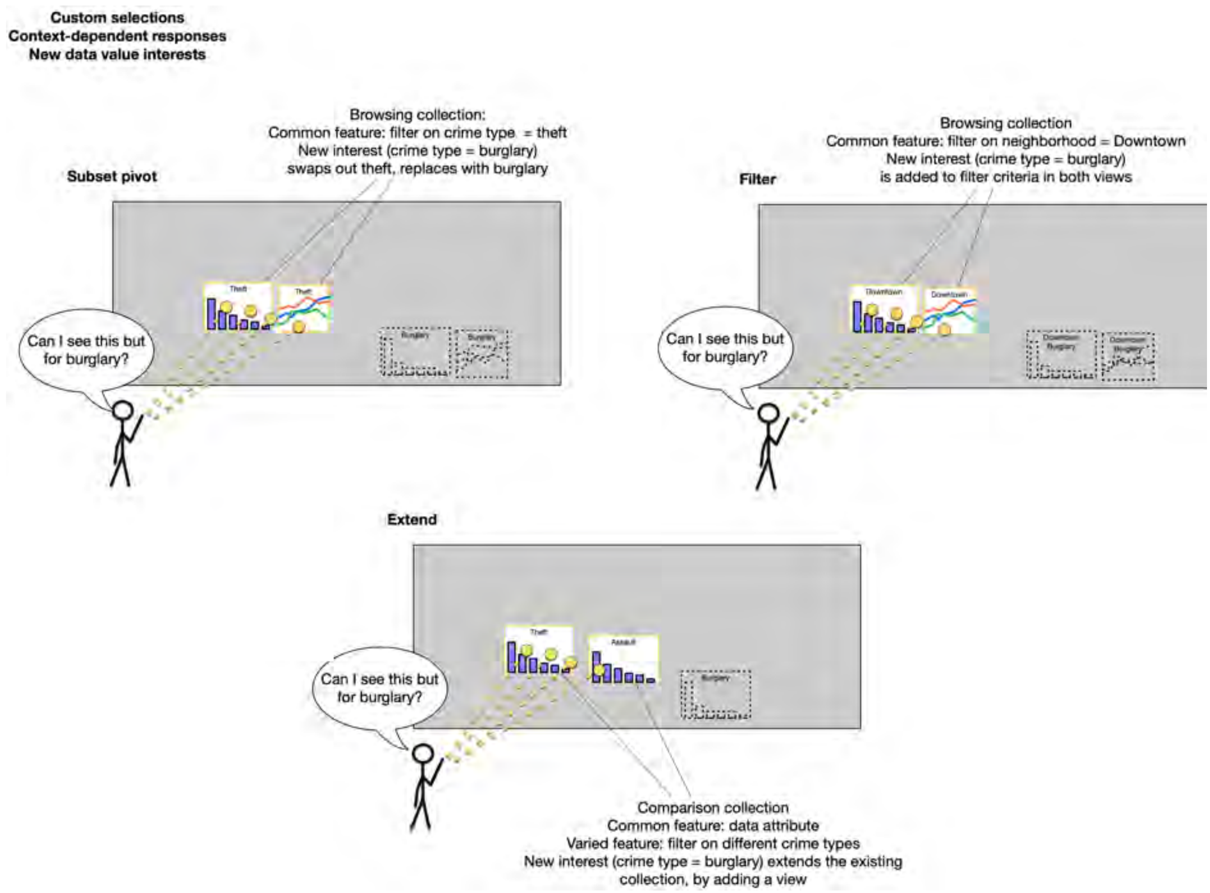


Figure 28: Context dependent responses to the same referential request, involving new data values of interest. Three different cases, with three different responses, that depend on the targeted views, and an assessment of their common and varied features

tion about how the user expects the system to respond to their referential action. To respond to self-created collections, we first determine how the referenced views are similar, and classify them into a collection type. Then, responses proceed as in Traverse, with a few examples shown in Figure 28, Figure 29 and Figure 30.

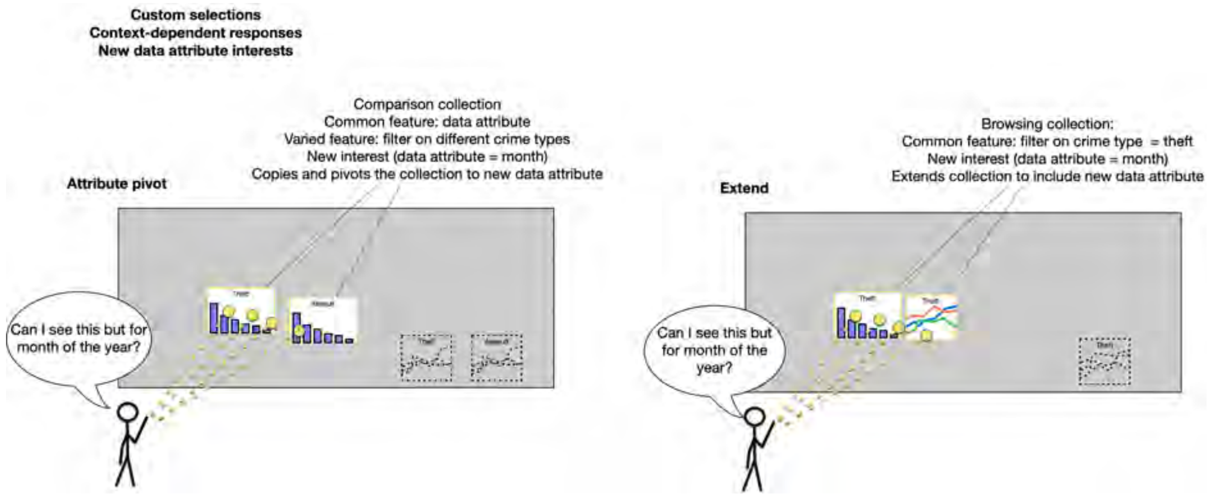


Figure 29: Context dependent responses to the same referential request, involving new data attributes of interest. Two different cases, with two different responses, that depend on the targeted views, and an assessment of their common and varied features

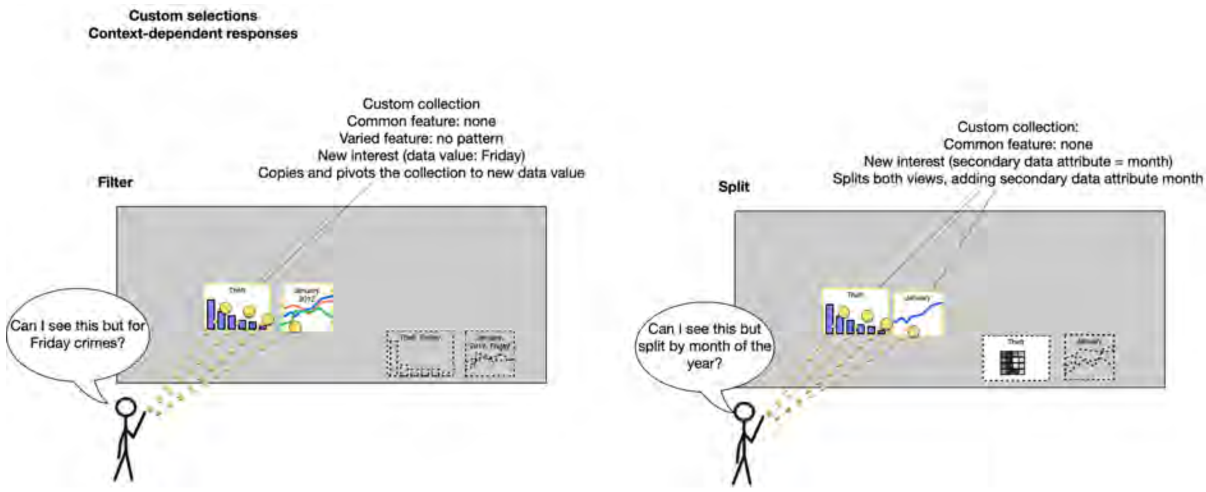


Figure 30: Context dependent responses to the same referential request. Two different cases, with two different responses, that depend on the targeted views, and an assessment of their common and varied features. In these cases, the user indicated views that did not have clear conserved features, but Ditto can respond with filter and split actions in some cases.

If this set of views cannot be classified in a collection, we classify it as other, and attempt to respond to the request. Filter requests can typically be performed on any set of views Figure 30. Adding a data attribute (a split operation) can be performed on views with single data attributes Figure 30.

5.6 Implementation

In this section, I will describe our implementation of our multi-modal speech and mid-air gesture data exploration technique, which we call Ditto. Ditto is a colloquial term, originally from Italian 'detto' meaning 'said', and is used to convey "the same thing again", or to indicate that something already said applies again. This name reflects the core mechanics of our technique- to refer to onscreen views, which are themselves the product of something previously said, and which show one particular selection of data values and data attributes, and to repeat this but with a change, with new selections of data values or attributes. I will describe how we translated our design goals into implementation requirements, and the decisions we made to realize these requirements.

5.6.1 Implementation requirements

Based on the design goals, we developed a list of implementation requirements.

First, we needed a speech input system, capable of capturing spoken 'actionable' requests from the user, and isolating these requests from other speech that might occur in the room. Since the user would be using this system repeatedly, it needed to be easy to active, and activation should have minimal errors. It needed to capture short, actionable segments of speech with reasonably high fidelity.

Second, we needed a pointing detection system, capable of capturing pointing gestures, and the onscreen targets of pointing gestures. These pointing gestures need to be precise at the level of whole visualizations, not entities within visualizations. We do not need to record all pointing gestures, just pointing gestures that co-occurred with spoken, actionable requests.

Third, we needed a touch system, that allowed users to position views on the display. We did not focus on other uses for touch gestures- such as interaction within a visualization, to change encodings or layouts, or to select elements within a visualization. This work would fall under the umbrella of multiple coordinate view research, and is out scope for this research contribution.

Fourth, we needed a display platform, capable of showing visualizations, in user-defined layouts. This platform needed to also allow us to display information about the state of the system, and the queries and responses, in a chat-box history.

Fifth, we need a view generator system, that would interface with these components, and supply visualizations in response to user queries and the targets of their pointing gestures. This generator needed to supply one or several views, and respond to referential operations as well, and referential operations on user-defined collections of views.

5.6.2 Implementation of the input system

Based on our design goals, we isolated a set of implementation decisions that pertained to our input system, which were resolved through formative evaluations with users. Here we will describe these decisions and how we resolved them. An overview of this system design is presented in Figure 31.

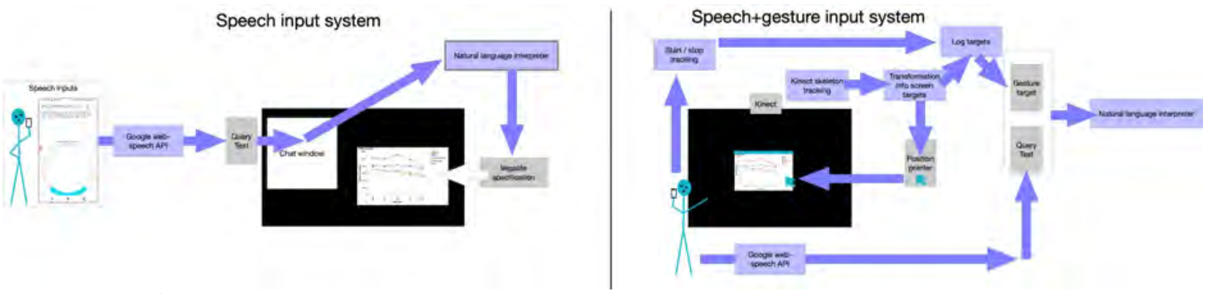


Figure 31: A diagram of the speech input system, and the speech and mid-air gesture input system.

5.6.2.1 Capturing spoken requests

We needed an input system which captured spoken 'actionable' requests (20; 44; 45), and distinguished these requests from other spoken utterances that might occur during data exploration. There are two basic approaches to capturing spoken requests from the user.

The first approach that we tested used continuous listening, and would use some approach to decide when the user wanted a response from the system. At the moment, separating 'actionable' speech from non-actionable think aloud, is an ongoing research question (23). So, to accomplish this for our system, we tested continuous listening systems with keyword-based activation, such as Alexa or Google Home. In these tests, the user would say 'Computer' in order to signal that they were going to pose a request, and then would use language to direct the request to our visualization generator server, by saying 'Ask Ditto to....". However, we found in testing that there were frequent missed requests, and misunderstood requests. Sometimes the Alexa would not detect the first query (Ok computer) and then other times it would miss the request to direct the query to the our visualization generation system. In addition, it was very wordy to express requests in this, for example “Ok Computer, ask (name of system) Can I see a chart for thefts in the Loop?”. Due to these combined difficulties, we did not pursue further development with continuous listening, keyword activated platforms (22).

Next, we tested interfaces that gave users direct control over the activation of speech-to-text transcription. We tested two approaches, both of which used an interface on an Android phone. The first required users to control both the start of recording and the end of recording. The second required users to activate recording, and then recording would stop when users stopped speaking. During pilot studies of the interface (with 4 participants) we found that participants strongly preferred the second approach- direct activation and with recording halted by pausing speech. One of the limitations of this decision, is that the recording may deactivate before the user has completed their request, such as if they pause in the middle of deciding what to ask. This was a source of some errors for users, which we discuss in our results section.

The visual interface for spoken interactions is depicted in Figure 32. It consists in the activation button, which changes color when listening and transcribing spoken requests, and an area to display the transcribed text. Once activated, it will send any captured spoken utterances to the system. Our interface also supplies a sound when activated and deactivated, for an additional source of feedback.

5.6.2.2 Capturing pointing gestures

We needed an input system to capture references to onscreen targets while participants spoke. We used a Kinect mounted to the center of the display, which captures a user’s skeleton data. Kinect data was captured with the Omicron multiple input and devices framework (140).

Figure 32: Three different states of the input interface, that runs on a mobile device.

Skeleton data consists in a set of joint positions in 3d space, with axes expressed relative to the Kinect device.

The Kinect on the display is directed at a 45 degree downward angle, from vertical, capturing the user from above. We perform a transformation on skeleton joint coordinates, to put the skeleton in a coordinate system that is relative to the display, and is easier to work with. This means that we first rotate the skeleton, so skeleton positions are parallel to the display, and lateral movements of the user to either side of the display are recorded as changes in x position, vertical changes in the position of a joint, such as raising an arm, are recorded as as changes in y position, and moving towards or away from the display are recorded as changes in z position. This is accomplished by considering the floor plane, and applying a rotational matrix, based on the angle of the Kinect relative to the floor plane. We also take into account the display size, and position, and the height and position of the Kinect.

The skeleton is continuously tracked, but we begin to detect pointing gestures when the user of the system raises their right arm away from their sides, above a threshold angle of 90 degrees. When we believe that the user is pointing, we activate an onscreen visual pointer, to provide visual feedback. This visual pointer can be seen in Figure 33 and the approach we adopt is shown in Figure 34.

As the user moves their raised arm, we move the on-screen pointer. Movements of the arm are related to movements on the screen, by creating a virtual display around the user. This virtual display is in proportion to their head, the length of their arms and the distance of their arm toward the display, relative to their torso joints. As users reach the edge of this virtual screen, they also reach the edge of the large display, and as their movements reach the center of their virtual screen, it also reaches the center of the large display (141). We also used a 1Euro filter, to smooth the skeleton data, and smooth the pointer movement on the display (142).

Logging the window targets of the participant's pointing gestures is based on whether the onscreen pointer has brushed over the window during the user's spoken inputs. Logging these targets begins when the user activates the speech input system, to meet our goals of capturing synchronous point and speak interactions.

We wanted to capture these gestures without requiring users to wear a motion tracking device. Based on initial tests, the Kinect could track movements of users skeletons with sufficient precision to allow for clear detection of point targets.

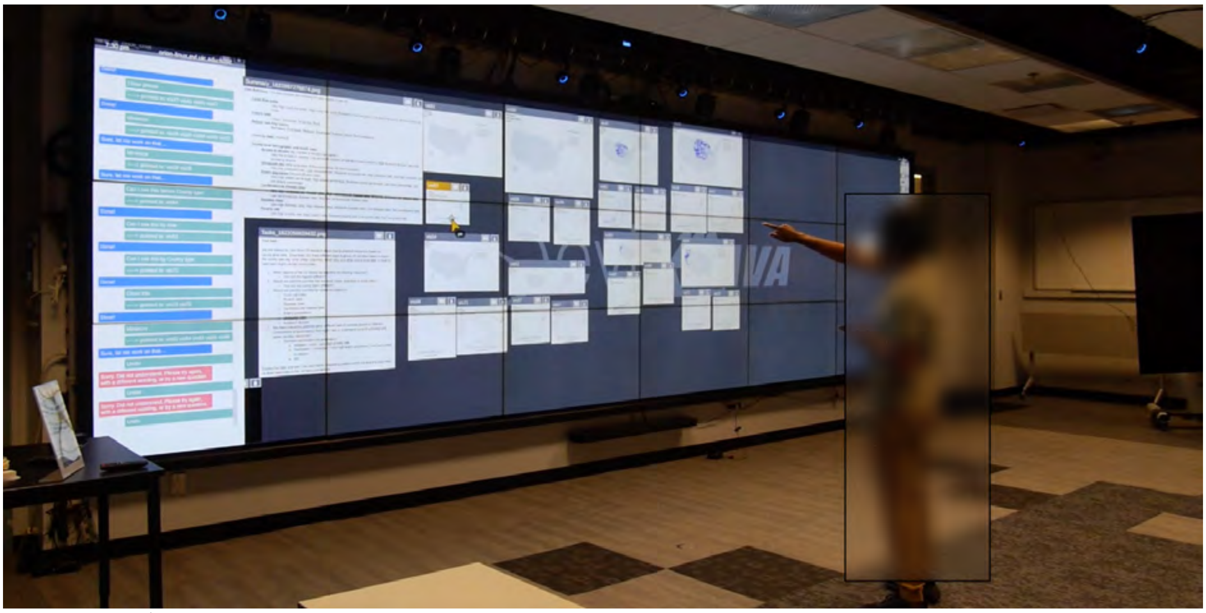


Figure 33: A participant in our study points to an onscreen visualization, to perform a referential operation using the targeted view.

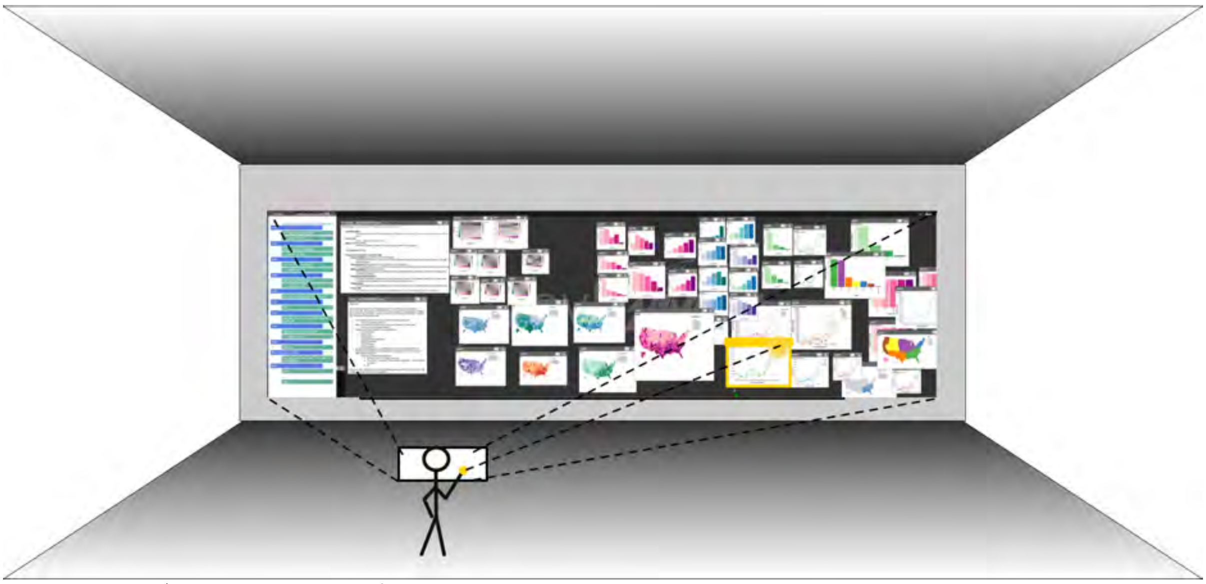


Figure 34: An illustration of how we mapped mid-air pointing movements to an on-screen pointer.

5.6.2.3 Display environment and Touch inputs

Ditto runs within the SAGE2. SAGE2 is an application for visualization and multi-user interaction with visualizations on large, collaborative displays. SAGE2 is a web-based framework, consisting in a node.js server and and html/javascript front end (93). Ditto uses a custom branch of Sage2, to integrate our system for synchronous pointing and speaking inputs, and to accommodate visualization creation through an external view creation server, Traverse, described in the previous chapter. Touch events are detected within Omicron (140), and we use the SAGE2 touch manager, to enable users to position views on the display.

5.6.3 Display interface

The display interface for Ditto was in SAGE2, and is shown in Figure 35. The main application, displayed within SAGE2, consisted in a scrolling chatbox, that displays query history, and some responses from the Traverse view collection generator. It shows user queries in green, and system responses in either blue, for a normal system state, or red to report an error. The chat interface presented both the user's queries and the visualization identifiers of any views they referenced in their questions, and responses from the system, such as 'Processing', 'Done' and "Error'.

The visual interface for SAGE2, consists in the display area, and windows that can be freely positioned by the user. A colored 'pointer' object was shown when participants would raise their right arm, and this pointer would visually highlight windows as it passed over them, changing the color of the title bar.

Figure 35: A screenshot from a study session, showing the different portions of the display, and the visual interface.

5.6.4 Traverse+ Natural Language data exploration system

Spoken inputs were sent to Traverse+, an augmented version of our NL for data exploration system that we described in the previous chapter. Traverse+ responds to direct requests, both targeted and 'cast-a-net' requests, and responds to referential pivots.

Traverse+ differs from Traverse, in that it allows users to send custom sets of views, for pivoting actions, and the system attempts to assign this collection of views to one of the existing view collection types, to generate an appropriate response. In the event that views cannot be assigned to a view collection type, Traverse+ has an additional approach to decide whether to respond. This implementation executes the design principles presented in the previous section.

We also augmented Traverse to include window positioning requests to 'move views to the side', or requests to 'undo' an action or 'delete' a visualization.

5.6.5 Formative Evaluation

We conducted an early evaluation of our interaction technique, where we refined the input speech system, and established that participants could speak and point comfortably. The results of this evaluation are presented in Kumar et al (21; 22).

5.7 Evaluation

We conducted a user study of our implemented multi-modal speech and mid-air gesture data exploration technique for large displays. We were interested in understanding how participants created views directly and how they used speech and gestures together to reference and then copy and pivot the referenced views. We wanted to see whether participants used the display space to enable referential interactions, and used both referential actions with a single target and many targets, as well as referential commands with one expressed pivot operation, as well as many.

We designed an evaluation around a data exploration scenario, to explore how recruited participants utilized our interaction technique. Participants engaged in a brief training, to learn how to use our interaction technique, and then they completed an open exploration of a data set. Our experimental setup and study protocol was refined through a pilot evaluation with 4 participants. In the pilot study, we clarified our user instructions and task description, and tested different arrangements of study materials on the display wall. We also refined thresholds for marking targets of pointing gestures- to limit unintentional selections of views- and improved options for closing and moving visualizations to the side.

5.7.1 Participants

We recruited 8 participants (4 M and 4 F), ages 24 to 33, pursuing PhDs (7) or MS degrees (1), from largely computer science or engineering fields. They had reported using data visualizations daily (3), weekly (2), monthly (2) or yearly (1). Participants all were fluent in English, though 6 of the participants were from a non-English speaking backgrounds.

5.7.2 Apparatus

Our study was conducted in a laboratory setting, on a 24 x 6.75 feet (7.3 x 2 meters) , 37MPixel display wall. A Kinect, centered and angled downward, tracked participant movements, and detected mid-air pointing gestures with the right hand. Participants were given an Android Pixel 2 phone (5-inches (130 mm) with a resolution of 1920 $\times$ 1080), which connected with the sage2 node.js server, through wireless internet.

Two experimenters observed and guided the participants in the study. The first was seated behind the user. The second observed the sessions via a laptop Zoom interface. Both experimenters assisted with training and documenting user behavior.

On the display was the chat interface, which presented both the user's queries, the visualization identifiers of any views they referenced in their questions, and responses from the system, such as 'Processing', 'Done' and "Error'. This interface was on the left side of the display. Next to the chat interface, were two documents describing the data and the tasks. During the training phase, participants were also shown a list of scripted examples to follow, on the wall near the center of the display, slightly to the right and near the top.

The center of the display was reserved for new visualizations. Participants could ask to move views to the side, through the pointing and speaking interface, and the minimized views were moved to the right hand side, in a descending stack of visualizations. New views were positioned in a horizontal stack at the bottom, center of the display.

5.7.3 Procedure

The study was conducted in 4 phases: 1) explanation of the data exploration technique and input system, 2) training, 3) open exploration and 4) survey and debrief.

5.7.3.1 Training

During the training phase, participants were given an explanation of the interaction technique, by looking at a document that presented a set of example queries and simple graphics of example responses. Participants were told that the environment was designed to provide simple visualizations in response to their interests in the data. They were given an explanation about asking focused questions vs 'cast a net' questions vs referential questions, with a set of examples.

Then participants were shown the large display and the phone interface, and they were given a set of instructions for using the environment. The experimenter would perform a couple of example queries, showing the participant how to activate the speech to text listening system through the phone interface and showing the participant how to indicate on-screen visualizations, with examples of point-and-speak referential interactions. They demonstrated use of the touch screen, to position views on the display.

Participants were then given a list of queries and referential interactions to perform, in order to become familiar with the interface and with the kinds of data exploration NL commands Ditto could accommodate. This portion used the city crime dataset, described in our usage scenario. This portion of the study lasted an average of 30 minutes.

5.7.3.2 Exploration

After training, participants were given a description of an exploratory problem and a dataset, described below. They were instructed to explore the data freely. They were given a set of sub-tasks, which were persistently visible on the display, on the left hand side. They were told that the sub-tasks were suggestions, and that they could also explore based on their interests and observations.

Participants were told to position visualizations as they came in using touch, and to choose whatever configuration supported their understanding of the data. We asked participants to think-aloud, and relay their observations and thought process. We told participants that at the end we wanted them to tell us a bit about what they found, through the visualizations they created.

We provided the participants with some visualization arrangement aids- in particular the ability to point at a visualization or a set of visualizations to close them, and the ability to move a visualization or a set of visualizations 'to the side'.

5.7.4 Task and Data

Participants explored a COVID19 related dataset about US counties, which was used in the evaluation of Traverse in the previous chapter. It consisted in a county classification, as rural, urban, suburban or small city, a regional classification (Midwest, Southeast...), and county-level Centers for Disease Control health indices (poverty rate, diabetes rate, cardiovascular disease rate, percent over 65, uninsured rate, primary care doctors per capita), and county-level COVID19 case numbers for each month from April 2020 until January 2021. In addition, for each county we used a machine learning prediction for 'COVID19 vulnerability' (136), and instructed participants to examine this machine learning based prediction with respect to the provided county-level health, demographic and COVID19 case data, and to look for interesting features by region or county type (rural, urban...). This data is described in greater detail in Appendix A.

5.7.5 Captured Data

We collected video and audio from the sessions. We captured a complete screenshot of the display after the participant organized new visualizations, and if they made significant changes to visualization positioning between queries. We also captured a complete log of participant queries, pointing gesture targets, the responses of Traverse+ to these multi-modal inputs. This included visualization responses, requests to position visualizations or delete them, undo actions, and errors.

5.8 Results

5.8.1 Analysis methods

We analyzed survey responses and logs of participant queries, referenced views and system actions and related these logs to the captured screenshots and videos, along with notes captured during the study. We found that participants used direct targeted and 'cast-a-net' queries in concert with referential actions, to explore the data. In the next section we will describe how participants used the technique, how referential actions appear to have enabled iterative exploration, that picked up on current and prior points of interest. We also describe how participants were able to use referential actions that targeted both single views and sets of views, and expressed one or several simultaneous pivot actions, to rapidly create views in support of exploring multiple points of interest.

5.8.2 Overview of participant exploration actions

Using logged participant actions, captured screenshots of the display after every query+position cycle, and open coding, we captured an overview of how participants used the available NL interaction command types.

In Figure 36 we present the frequencies of different action types, broken down by major category and participant.

We captured a total of 307 visualization queries with responses from Ditto. Of these, around responded to 55 percent (168) were 'direct' queries, where participants only used speech to express their intentions, and 45 percent (139) were 'referential', involving simultaneous speech and mid-air pointing gestures. Of the direct requests, 21 percent (36) were 'cast-a-net' queries, where participants expressed several points of interest, and Ditto provided several views.

Within referential requests, participants expanded their exploration to new data value interests, through 'subset pivots' and 'filter' operations (29 percent of referential operations), and through data attribute pivots or additions through the 'split' operation (69 percent of referential operations).

Category / Participant	P1	P2	P3	P4	P5	P6	P7	P8	Total
Minutes	69	55	28	26	42	23	53	58	354
Visualization queries	51	50	17	23	38	21	47	60	307
Window manipulation	20	29	18	6	26	13	19	21	152
Errors	15	18	10	12	12	3	3	35	108
Direct	**20**	**23**	**4**	**18**	**35**	**13**	**30**	**25**	**168**
targeted	8	8	4	4	12	3	6	10	55
multifaceted	9	8	0	6	15	9	19	10	76
cast a net	3	7	0	8	7	1	5	5	36
Referential	**31**	**27**	**13**	**5**	**3**	**8**	**17**	**35**	**139**
Subset pivot	4	0	0	2	0	0	8	7	21
Attribute pivot	10	5	8	2	1	0	5	14	45
Split	0	1	5	0	0	0	0	0	6
Split pivot	11	13	0	1	1	8	3	9	46
Filter	5	8	0	0	1	0	1	4	19
Extend	3	0	0	0	0	0	0	1	4
Split pivot extend	1	0	0	0	0	0	0	0	1
Referential #inputs/#outputs	**31**	**27**	**13**	**5**	**3**	**8**	**17**	**35**	**139**
One to one	25	18	12	2	3	8	11	24	103
One to Many	3	1	1	1	0	0	2	1	9
Many to One	1	1	0	0	0	0	0	0	2
Many to Many	2	7	0	2	0	0	4	10	25
Errors	**15**	**18**	**10**	**12**	**12**	**3**	**3**	**35**	**108**
speech to text	6	8	1	3	8	1	0	16	43
speech input system error	6	2	1	5	4	2	1	9	30
pointing detection error	0	4	0	0	0	0	1	2	7
query is out of scope	0	3	8	3	0	0	0	0	14
non-response / unclear query	3	3	0	1	0	0	1	8	16
Window operations	**20**	**29**	**18**	**6**	**26**	**13**	**19**	**21**	**152**
Close	5	15	15	1	9	9	7	13	74
Move to the Side	2	2	0	0	2	1	0	5	12
Undo	13	12	3	5	15	3	12	3	66

Figure 36: An overview of participant actions and system responses.

Participants on average used both direct requests, that are posed through speech, and referential point-and-speak requests. Some participants strongly favored direct requests (participants 4 and 5), others used both equally, and a few used referential requests more often (participants 1, 2 and 8). In the next section, we will focus on how participants wove between direct and referential questions during data exploration.

5.8.3 Referential operations, and evolving points of focus

In order to understand how participants used referential operations to explore the data, we created the visualization in Figure 37. To generate this view, we took logged interaction data and visually represented the session as a sequence of views provided in response to user queries, and arcs to show when a view is referenced, and copied to produce a new view. Arcs show how a referential request takes an indicated view or views, and produces a new visualization. The gray boxes show when a group of views is produced in response to one query, such as in a cast-a-net request, or a copy-and-pivot operation with a single view target and multiple pivots, or a copy-and-pivot operation with multiple view targets that are copied and pivoted collectively.

One observation that we can make from these figures is that our participants often referenced views from early in their data explorations session, to copy and pivot in new directions later in the session. For example, we can see from this visualization that participant 7 referenced the very first view they produced, to copy-and-pivot it in their very last query of the session. Participant 2, referenced their first two visualizations repeatedly, which means they copied and pivoted these two views repeatedly, to extend this initial focus in a new direction. This data

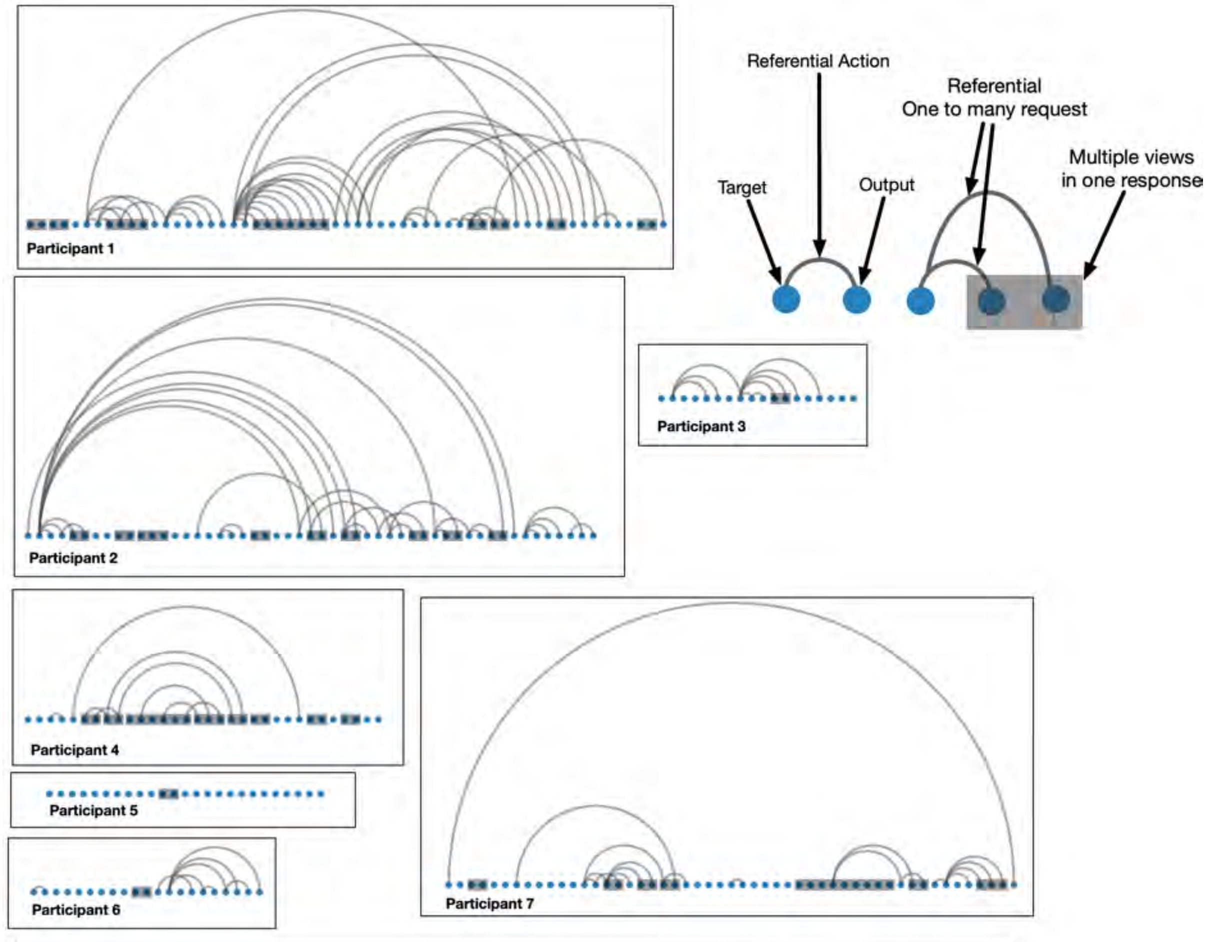


Figure 37: A visualization of participant actions. For each participant, we show a list of visualizations, blue circles, in the order they were created. Arcs connect visualizations that were referenced, copied and pivoted. Views that were created together in one action, such as a cast-a-net action or a many-to-many or one-to-many referential action, are shown with a gray box.

exploration style suggests that participants may have started to explore a particular set of data and attribute interests, creating a set of views at one point in their session, and then moved to a new thread of their analysis, creating a different set of visualizations, but participants returned later to their initial exploration thread. The large display accommodates this data exploration pattern, because views can remain persistently present on the display wall, and can be returned to later.

We can also see instances where participants repeatedly pivoted a single view target, to produce many new visualizations. In effect, after finding a visualization that suited their interests, they incrementally extended the focus of their exploration, either through repeated single pivots (one to one) or multiple parallel pivots (one to many).

5.8.4 Referential operations, one to one, one to many, many to many

Using our logged data, we found that participants most frequently used one-to-one referential operations (75 percent of referential operations) specifying a single view target, through pointing, and a new data attribute and/or value interest, which Ditto responds to with a single view. Participants would often use repeated referential one-to-one operations to incrementally build up a set of views, which contained both conserved and varied features. We can see instances of this in Figure 37. For example, Participant 3, used this one-to-one referential interaction style repeatedly, referencing their second visualization, to create 3 new visualizations, and referencing their eighth visualizations to create 7 more, only one of which was a 'one-to-many' operation (which is evident from the pair of views in the gray box).

Participants occasionally performed ‘one-to-many’ pivots, (6 percent of referential interactions) where they indicated a single view target and asked for multiple new data values and/or attributes, which Ditto would use as a basis for providing a set of views.

Participants performed many-to-many pivots for around eighteen percent of referential interactions, pointing to several views during their spoken command, and indicating a new data value and/or attribute interest. Because they indicated several view targets, Ditto would copy and pivot each indicated view to the new points of interest, producing a set of duplicated and pivoted views. In effect, users could express a ‘batch action’, that operated on several views at once.

5.8.5 Case Study: Targeted and Referential, For Evolving Exploration Goals

We present a case study, showing one participant’s use of space and a combination of direct and referential actions to explore the data. We used captured screenshots from the session, along with captured screenshots from the video recording, to construct this description of her exploratory actions. A photo of the final state of her session is displayed in Figure 38.

This participant created 66 views of the data, in 69 minutes, using 20 direct (39 percent of visualization requests) and 31 referential (61 percent of visualization requests) interactions. Overall, she mostly employed targeted or multifaceted (eg. two data attributes) (85 percent of direct interactions) and one-to-one referential requests (80 percent of referential requests), employing an iterative exploration approach. She created a total of 6 partially overlapping exploratory clusters, depicted in Figure 39. The evolution of her analysis and her use of space can be seen in Figure 40, which contains a sampling of snapshots from her session.

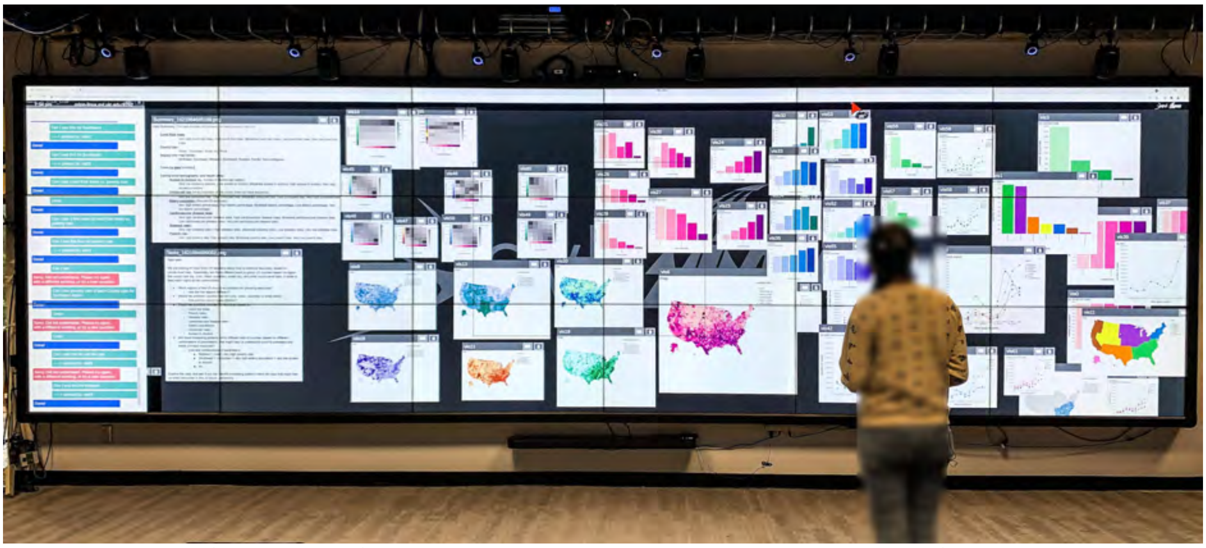


Figure 38: The final state of one participant's analysis session, which we focus as a case study to illustrate how targeted and referential actions enabled her to build coherent sets of views, which she positioned in coherent groups on the large display.

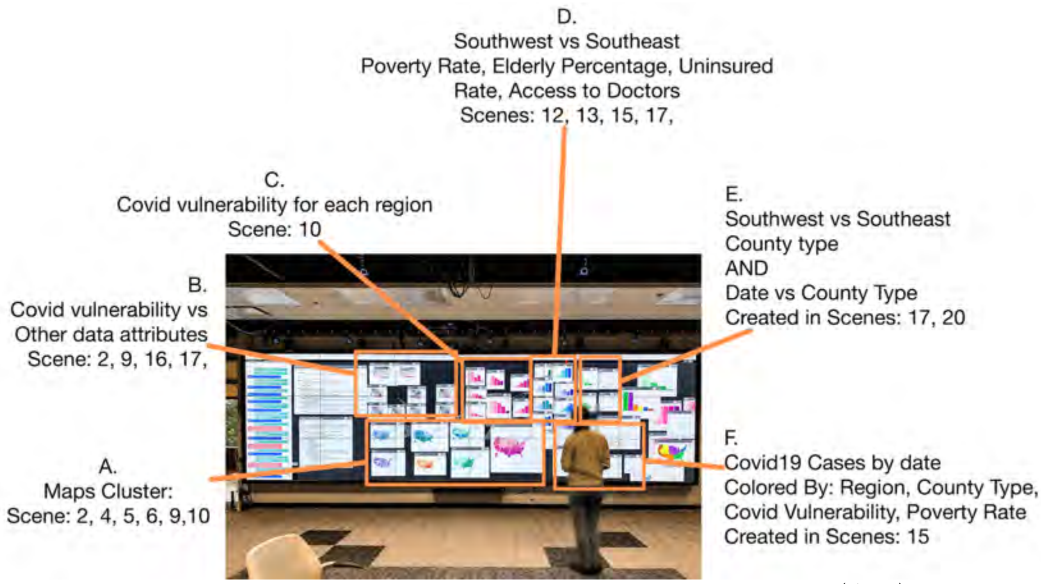


Figure 39: This participant generated several clusters of visualizations (A-F), which are coherent view collections, using our data exploration technique. She created these views largely through targeted and one-to-one referential actions.

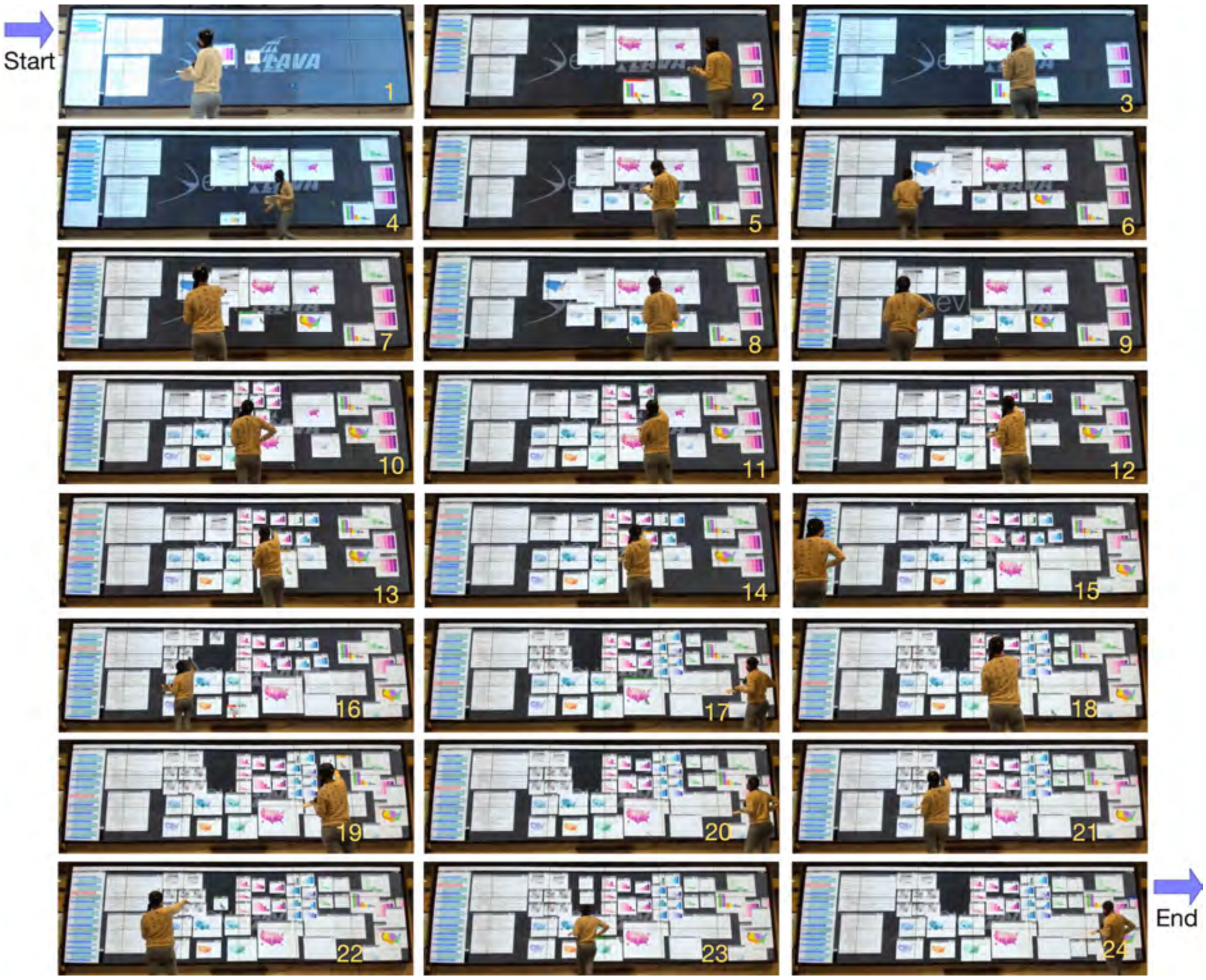


Figure 40: This figure shows selected snapshots from the video recording, ordered in time from the start in the upper left to the end in the lower right. The evolution of the participant's exploratory session can be seen. She dynamically followed evolving threads, based on her interests, frequently returning to earlier threads.

At the start (scenes 1 through 4), she created a set of views starting point visualizations, getting familiar with the data. Then she began to focus on exploring maps, colored by data attributes of interest. She expressed this intention first with a direct , targeted request ("Can I see a map of Covid19 risk?"). Then she followed up on this map, through a set of referential one-to-one operations, performed in between other requests, generating maps of access to doctors, uninsured rate, elderly population, cardiovascular disease rate, poverty rate and diabetes rate. These were all one-to-one referential operations, and she gradually build her view collection of maps, colored by varied secondary attributes. She positioned these in a loose cluster.

In between creating maps, she also asked questions about the correlation between Covid19 risk scores versus regions ("Can I see covid risk index by ridge(sic) versus region"), in a targeted direct request. She returned to this line of inquiry, after following another analysis thread, to ask a second targeted direct request about correlation. Then she returned to this analysis thread in the second half of her session, to create several more of these plots, first through another targeted, direct request, and then through a series of referential one-to-one and one-to-many pivots, targeting the existing views in this cluster to create more. She placed these carefully in a cluster, after each was created.

Early in the session, she created a set of visualizations showing Covid19 risk index for each of the regions (Cluster C) (one to many, referential subset pivot). She selected plots associated with the southeast and southwest from Cluster C, and used these as a starting point for the plots in Cluster D, followed by Cluster E. In between, she explored Covid19 cases, by date versus other secondary attributes.

This case illustrates use of targeted and one-to-one referential actions to build sets of views. This participant also illustrates use of the display to externalize their interests, and then because they are visible throughout the session, she returns to them later, to extend them. This participant generally positioned views that she referenced, near other views, because she was using these one-to-one referential actions to expand a prior collection.

5.8.6 Case Study: Many to Many referential operations

Several participants used many-to-many referential operations to duplicate a set of views. One example, involves a participant who had just created 4 visualizations showing maps colored by elderly population, in the Midwest, one showing all Midwest counties, and the others filtered by the three of the four county types in the dataset (rural, urban, suburban, small city). These four maps had been generated by a targeted direct request (”Can I see a map for elderly percentage in the Midwest), followed by three one-to-one referential pivots, which filtered the map by three of the county types (”Can I see this but for rural counties?”, ”Can I see this but for Suburban?”, ”Can I see this word(sic) for Urban?”). Then, the participant pointed to all four of the maps, and asked ”Can I see these but for the Pacific?”. The resulting 4 plots were placed in a grid, so the participant could compare the geographic distribution of high and low elderly populations, in different regions and in counties of different densities.

5.8.7 Errors: Observed points of difficulty

In addition to logging user requests and Ditto's responses, we also automatically logged errors. We focus on situations where the Ditto could not offer a response. We reviewed the logged data, and coded the non-response cases, based on the primary issue that led to the

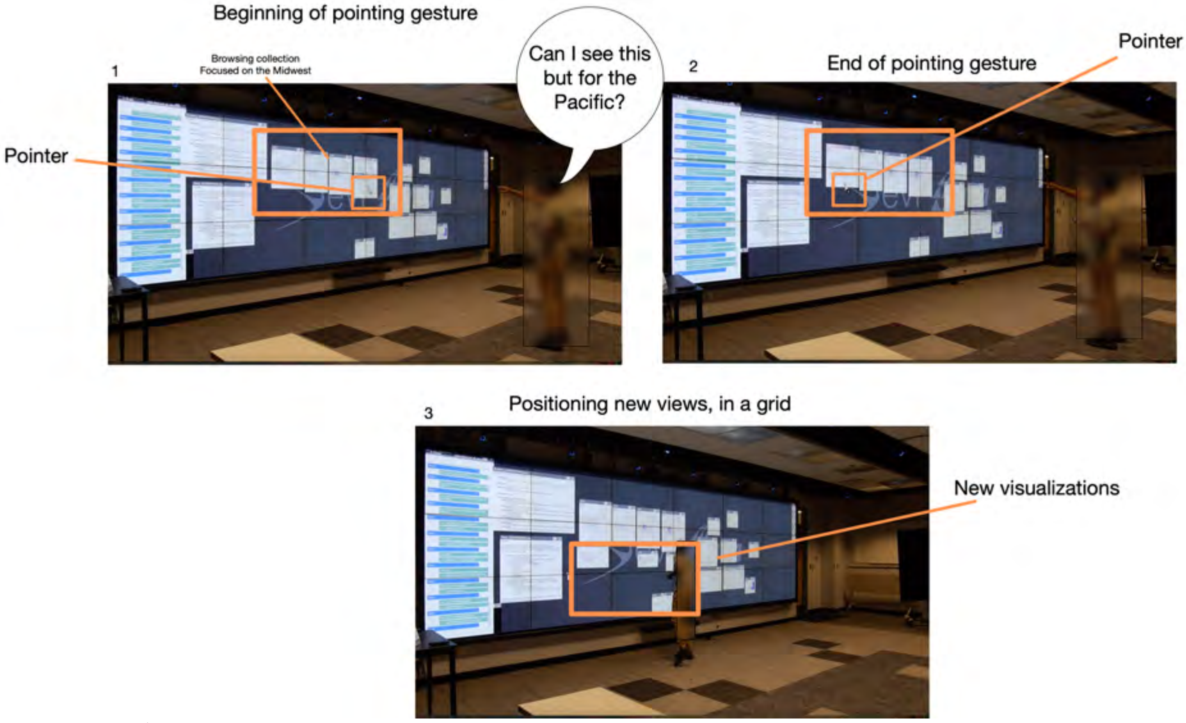


Figure 41: A participant in the study first points to a set of visualizations, containing a custom set of views, all showing Midwest counties colored by elderly population, but varying in filter criteria around county types. The participant asks to see these views but for a different region "Can I see this but for Pacific?", and Ditto provided four visualizations that were duplicates of the original, but displaying the Pacific region, rather than the Midwest region.

system error. We used an open coding approach, and applied these codes through multiple passes of the logged data.

We found that the most frequent sources of error involved the speech input system. Close to 40 percent of the errors, were due to speech to text interpretation issues. This is where the spoken request of the user is mis-transcribed by the speech to text interpreter.

Speech to text issues will likely resolve as speech to text translation systems improve. Given the interest in natural language interfaces, there are likely to be improvements in these systems in the next few years. In addition, in future work in this area, we could explore training the speech to text interpreter on each participant's voice and accent, and augmenting interpretations with a corpus related to the data, so keywords that are most critical to interpretation of a user's intent are easier to detect.

System input errors typically involved the time-out feature on the listening system. This is where speech to text transcription ends when the user pauses for a few seconds. The represented 28 percent of the non-response cases. We believe that this is an interesting challenge for spoken NL systems, which is addressed in my colleague Abhinav Kumar's forthcoming thesis, and in our prior work (23; 22). As described in the implementation section, we tested several approaches for capturing spoken inputs, and we found that participants in early pilots of our interface did not find other approaches to be easy to use- including alternate input devices (Alexa, Google home..) or input capturing methods - involving both direct activation or listening and direct activation of terminating listening. It is possible that for data exploration scenarios, users need more time to formulate their query, and the standard APIs for capturing spoken inputs may time out too soon.

In an additional 27 percent of the cases, the system was not designed to respond to the question users posed (out of scope), such as asking about population data or for specific visualization types not currently available in our implementation, or the question posed by users was unclear to the NL interpreter, and a response could not be provided.

Pointing detection errors were less frequently a source of non-responses. This includes situations where users intended to point to a view, but it was not detected, and situations where the system detected a pointing action to a visualization that the user did not intend, such as accidentally brushing over a view during the pointing gesture.

5.8.8 Feedback

After the study ended, participants were given a set of survey questions allowing them to comment on aspects of the data exploration experience with Ditto. They were provided a set of questions, which they scored on a 1-5 Likert score, to express agreement (5) or disagreement(1). They also were given a set of questions with an area to type more extensive comments. In this portion, we asked them to comment on their experience, and discuss places where we could improve. Figure 42 displays the results of this survey, with participant scores.

5.8.8.1 Feedback on exploration and responses

Participants largely felt that Ditto helped them explore the data, with 2/8 participants ranking it was 'very helpful' and 6/8 ranking it as 'somewhat helpful'. Participants also ranked the experience as 'enjoyable', with 5 out of 8 participants ranking it as 'very enjoyable', 2 in

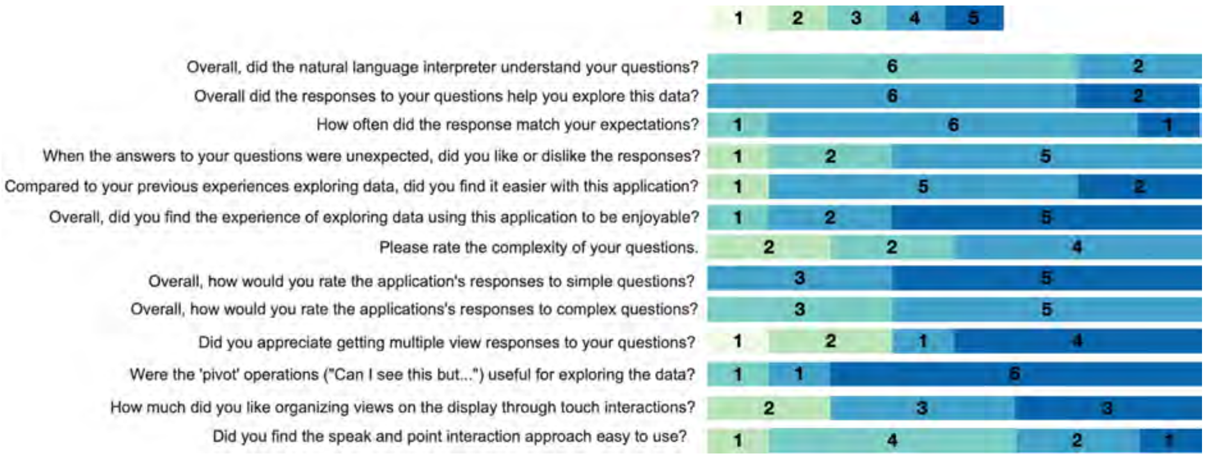


Figure 42: Participant Likert score responses to survey questions. 1 is strongly disagree, 5 is strongly agree. Each question is shown, next to a stacked bar. The color of each bar is based on the score. Width of the bars is based on the number of participants who responded with that score.

8 ranking it as 'somewhat enjoyable', and 1 in 8 neutral. Many of the participants found date exploration easier with Ditto, when compared against their prior experiences, with 2/8 saying it was much easier with Ditto, 5/8 saying it was somewhat easier.

Participants also ranked the copy and pivot, referential operations highly, with 6/10 saying the pivot operations were 'very helpful', 1/8 'somewhat helpful' and 1/8 'neutral'. One participant commented "I liked that I can create a lot of visualizations quickly and also can create new charts from existing ones. This seems to help me in understanding which data attributes I would like to have . . . ".

Participants generally felt that the responses to their questions matched their expectations, with 1/8 saying this was true 'all the time', 6/8 saying this was true some of the time, and 1 participant neutral. Participants generally were not unhappy with unexpected results, with 4 out of 8 somewhat liking unexpected results, 2/8 neutral, and 1 out of 8 disliking unexpected results ‘somewhat’.

5.8.8.2 Feedback on input system

Some participants felt that the combinations of inputs addressed their needs. One participant commented “The combination of gestures (touch and mid-air) and voice worked smoothly. It was easy to learn and use.” In the rankings, participants had mixed reviews of how well the pointing and speaking interactions worked for them. 1/8 felt these interaction were very easy to learn, 2/8 felt they were somewhat easy to learn, 4/8 were neutral and 1/8 ranked these interactions as somewhat difficult to learn.

Some participants were unsure about how long you needed to point to a visualization in order for their pointing gesture to be registered. One participant commented “ For gestures, you needed to point to a chart for during the whole command to avoid giving the command to other charts by accident. This could be a little tiring.” Another commented “I also think that maybe the pointing gesture could be shorter delay so there is less fatigue.” The duration of time that a view needed to be indicated is short in practice, but from these comments it appears that participants needed better visual feedback on this action, to know that they have successfully indicated a particular view. Although the visualization title bar would change color, when the participant’s pointing gesture brushed over the view, this signal may not have been strong enough, to register with the participants. Future work on visually indicating and controlling pointing actions could help address this challenge.

5.8.8.3 Feedback describing challenges

Participants expressed mixed responses to aspects of data exploration using Ditto. Participants ranked the NL experience as understanding their questions 'some of the time' (2/8) or neutral (6/8). This issue could be related to the speech to text interface, as in did the speech to text capture their intention correction, or it could relate to the interpretation of their request, as in deciding what response to provide.

With respect to the speech to text input system, participants commented on challenges with this system. One wrote "I found it interesting but also frustrating. Speaking-wise, the system had a bit of trouble with my accent and the time-frame given to record the command seemed short; hence, I found myself having to close many charts and re-doing the process." Another participant commented "The interval for voice input can longer, there were couple of times that it stopped listening while I was just remembering a particular property. ." As we discussed in the previous section on system errors, this was a frequent source of error, and addressing this should be addressed in future work.

Participants also commented on challenges with formulating questions using the NL command structure that Ditto supported. One participant commented "Deciding how to word commands was the more difficult part. Especially if the system failed to produce the expected result. " One participant noted that they had particular desired visualizations they wanted to create, something that Ditto is less well suited to, and they commented "Finally, i'd think of a way to remember the commands or the keywords to use to trigger the chart that I want. As it was open-ended, I found myself forgetting how to trigger the chart that I wanted or how to match a variable to the desired color or axis." Some participants suggested better visual interface to help users remember the kinds of queries Ditto responds to. One participant commented "I think a wider wording flexibility will be helpful especially if the users are non-native speaker like myself. A work around that I am thinking of is having hints open on the large display. I think the speech to text interpreter worked fine, a little bit less fine than my Google assistant, so it could use a nudge."

5.8.8.4 Feedback on Large Display

Many of the participants commented that they appreciated the large display and the ability to position multiple views in different arrangements on the display. One participant commented "The system was easy to bring up visualizations and spread them across the wall.". Another noted the ability to return back to prior views, writing "Having the big display helps in having a lot of views at the same time so it's easy to go back to refer to others and also compare.".

But participants expressed mixed reactions to managing views on the large display, and the multi-view responses to queries from Ditto. One participant commented "I found that I spent a lot of time dragging and resizing the graph windows." It is possible that this is one of the reasons that participants had mixed rankings for multi-view responses, with 4/10 strongly preferring multi-view responses, 2/10 somewhat preferring multi-view responses, 2/10 somewhat preferring single view responses and 1/10 strongly preferring single view responses.

Participants discussed view positioning challenges, and offered ideas for how to address these challenges in future. One participant commented "Touch worked well to reorganize, but some more initial grouping might be helpful. I.e. generated next to related graph that was referenced.” Another offered the suggestion to provide options to re-organize views from a distance, saying “I would focus on gesturing actions that can accelerate how to place elements on the screen as dragging the charts one-by-one seem slow. For instance, I’d like to use a lasso tool to then show the items in a grid or select all filtered by the same attributes and apply a change of location to all of them. “ Another suggested, “ I would have preferred if charts popped up into already empty space by default and were bigger by default. Having preset sizing options (say big, medium, small) could also help in addition to having manual resizing “.

5.9 Discussion

Based on the feedback and the usage patterns, we can see that aspects of our design for multi-modal speech and mid-air gestures for data exploration on large displays was successful. Participants in our study were able to learn the technique, after a short training, and use both direct requests and referential requests to generate sets of views. Participants used both one-to-one referential actions, targeting a single view to create a new view of the data, showing a new selection of interests, as well as many-to-many referential actions, which acted to copy and pivot sets of views collectively. We also noted that participants returned to prior views, later in their session, suggesting that externalizing their exploratory process on the display allowed them to pursue evolving points of interest and follow different exploratory threads.

5.9.1 Addressing speech and mid-air gesture input challenges

There were some limitations with the technique, that involved the input technology and points of error, such as mis-translating a request or not capturing an intended target of a mid-air gesture. There are several ways to address these limitations. First, the trajectory for speech to text translation is promising, and likely there will be improvements in this area in the next few years. Second, future work with this approach could use training or augment the speech to text with a corpus of keywords, to improve accuracy (25). Finally, it may be possible to explore using the speech and pointing together in ways that account for potential errors, such as inferring the a missed pointing gesture from context, or inferring a mis-transcribed spoken request using the pointing gesture targets as context. This would be valuable to consider as future work.

In addition, there may be design decisions that could improve aspects of the user experience. Some users suggested circling sets of visualizations, in a lasso gesture, rather brushing over them, which would be a great approach. Second, there may be ways to highlight groups of views visually, and then users could point to a smaller target area when referring to a group of views. Third, sophisticated natural language processing techniques could resolve references and detect referential requests better, than with our reduced NLI system (20; 21; 22; 23). In addition, we could explore ways to visually highlight the targets of pointing gestures more clearly.

Finally, we could explore using other devices or other approaches to capture spoken inputs, such as a watch or a microphone, or in-room microphones that persistently listen and detect requests. This would allow for device-less speech inputs, which might reduce issues with activation and de-activation of the speech-to-text input system.

5.9.2 Visualization organization and metavisualization

In addition, participants suggested that view positioning presented challenges, and they could see the need or value for future work that assisted with this time-consuming task. This is an interesting finding, because there is a assumption in our community that free positioning of visualizations is optimal, because users can then express their idiosyncratic sensemaking process through positioning decision. However, as the number of visualizations on the display grows, it becomes more difficult to manually position content. Several of the suggestions, such as positioning views next to the referenced target, or using an algorithm to find an available space for new visualizations, are a good direction for future research in this area. This is an also a design challenge where our view creation technique, Traverse, might be able to help. Traverse generates line and grid layouts for visualizations, that were used in our previous chapter. Future work could examine how to utilize these layouts in a flexible canvas environment, in conjunction with user decisions about visualization positioning.

In addition, there has been recent interest in metavisualization, which considers how to effectively show between view relations in large, multi-view environments. Our work considers a group-wise approach to between-view relations, as opposed to a pair-wise approach, which may have benefits for coherently displaying visualizations. Future work is needed to explore how to use this group membership formalism for metavisualization to assist with tasks that span more than one view of data.

5.9.3 Future work

There are several important directions for future work. First, considering a collaborative context, and whether the ability to use natural language and mid-air gestures assists in collaborative data exploration. In a collaborative context, the ability to pose questions about visualizations on the fly, and generate views of data without stepping away from a collaborative discussion, could be quite powerful. However, the challenge in this context is in disambiguating multiple input signals, particularly if collaborators toggle between independent and collaborative work using the speech and mid-air gesture system.

There could be value integrating other input modalities, such as pen and touch, with speech and mid-air gestures, as well as portable devices. These input modalities offer varied affordances, and would likely compliment each other (41; 95; 100).

Finally, future work could consider how to use spoken inputs along with large displays to capture analysis provenance. In non-natural language interaction contexts, the challenge for analysis provenance involves inferring high-level sensemaking tasks from low-level actions with a graphical interface. In a context where most actions occur through speech, it is possible to capture and visually represent a data exploration session, without making inferences. This would be a valuable direction for future research.

5.10 Conclusion

In conclusion, in this chapter we present a multi-modal interaction technique that combines speech and mid-air gestures for data exploration in large display environments. This work is motivated by the findings in 3, and utilizes Traverse, the technique for data exploration in support of breadth in exploration. Through our design, implementation and evaluation, we found that participants used speech, mid-air pointing gestures and touch interaction modalities to generate multiple views of data, position these views on the display, and the point and speak to reference and then copy and pivot views, based on evolving data exploration interests.

CHAPTER 6

DISCUSSION

6.1 Multiple Views, Large Displays, and Cognitive Load

Environments that accommodate the display and juxtaposition of multiple views of data have generally presented advantages for complex data exploration scenarios. For example, the ability to select and segment data, and represent in multiple views, allows users to explore based on their interests and view data from different vantage points. However, recent work by Chen at al., on multi-view layouts in visualization literature, has identified potential limits on the number of views that users prefer and limits on how many views are typically utilized in published visualization applications. This work considers the possibility that more visualizations might be overwhelming for a user, and they may prefer small sets of 5-6 views (143).

This presents challenges for large display environments, which often consider scenarios involving dozens of views of data (94; 129; 144; 38; 40). The technique presented in this dissertation accounts for this potential preference for small sets of visualizations in several ways. First, the selection and arrangement of visualizations with our technique in Ditto is under user control. Although the Traverse view generator provides visualizations automatically in response to user interests, views can be filtered and arranged by the user, which means they can exert their preferences in the number of views they retain and consider, reducing the risk of cognitive overload.

Second, we noted that participants clustered visualizations into meaningful groups during their exploratory sessions. This was typically accomplished through either cast-a-net requests, or referential requests, to create small sets of views. Our technique is therefore less an approach for filling the display with many visualizations automatically, such as generating a grid with dozens of small multiple views, for example, and instead a technique that allows users to create custom groupings. Our work is not inconsistent with the finding that users prefer small sets of related views, and we could frame this contribution as enabling users to arrive at several meaningful sets of views on the large display.

Finally, to address the potential of cognitive overload on large displays it is essential to explore techniques for view generation in this environment, not just interaction or view coordination. Systems that begin with a pre-populated set of visualizations, where the user can't control the number and arrangement of views, may be overwhelming to the user. In addition, the ability to select and then group visualizations into related clusters may have implications for other interaction problems, such as multiple coordinated views. Potentially, users may consider visualizations in small groups, and only want to see coordinated actions, such as coordination in brushing and filtering actions, within the group. The work of this dissertation may contribute to large questions around interaction with multiple visualizations on large displays (39).

6.2 Exploration Risks

This technique focuses on data exploration where vews are provided based on user interests that are expressed on-the-fly. By using a restricted natural language interface, this work aims to overcome some roadblocks in visualization construction that have been documented in the literature, such as difficulties selecting a visual template or navigating an unfamiliar graphical interface (12). However, there are still potential avenues for error.

First, users may misinterpret the provided visualizations. This may arise from unfamiliarity with specific view types or lack of experience with data visualizations. However, this may also arise in our technique from the fact that users do not manually specify visualizations. In other systems, a user may select filters, visual templates, data attributes and aggregation methods themselves (59). For Traverse and Ditto, participants need to read titles and axes labels to fully understand what the visualization is presenting. There were instances in both the pre-design study, and in evaluation of Traverse ad Ditto, where participants did not fully read the information on a provided visualization and then they misunderstood what the chart depicted. Accounting for interpretation errors may be a valuable direction for future work, either by listening for these errors in spoken think-aloud, or by providing help and guidance after presenting the user with visualizations.

Another potential source of error is in following exploratory paths based on biases or preconceived ideas, or potentially posing questions based on tunnel vision rather than based on observations from the data itself. These issues have been explored previously (67; 145; 146; 147; 148; 149), and they present significant challenges to the data visualization community as a whole. These issues may arise with the data exploration technique presented in this dissertation. For instance, users may view focused selections of the data, without first viewing the context for these selections. leading to drill-down errors, which arise when users fail to put an observation into context. Second, users might follow a path in which they fail to consider alternative ideas. For instance, perhaps a user explores to support a particular hypothesis, but does not explore in a way that would consider alternate explanations, and they may arrive at erroneous conclusions.

This challenge might be interesting to consider in future work. The large display environment presents space to present alternative suggestions, or comments from the system to help steer the user away from these errors. Second, there may be potential in the future to detect these issues through overhearing systems, that listen continuously to user think-aloud or collaborative dialogue.

6.3 Integration with other interaction modalities

This work focuses on data exploration through rapid view generation, in response to on-the-fly queries. The reason for this focus is that we want to be responsive to high-level questions, which a user may form quickly. In addition, although a collaborative context is not explicitly evaluated, the ability to generate views rapidly during discussions around data is a potential use case for this technique in future work. Another motivation for providing rapid responses to questions is the goal of helping users avoid visualization construction roadblocks and errors that they may encounter in systems that require manual specification (12), or to avoid difficulties learning to navigate a complex graphical interfaces.

However, not all interaction modalities aim at rapid visualization generation, or visualization generation around points of interest in the data. A number of techniques focus on manual creation activities, such as sketching (60; 131) or direct manipulation (27) or bottom-up view creation (150) through manipulation of tangibles (151). These techniques can be slower, are require more deliberate and sustained effort, but also potentially provide an opportunity for the user to realize visualizations that may not be produced by more rapid techniques. It would be interesting, as future research, to explore the interplay between fast and slow, and the interplay between a modality like speech and a modality like sketching, for data exploration on a large display Sketching interactions require users to express a visualization that they can envision, whereas speech requires users to express questions and points of interest. They address different needs, and might complement each other in interesting ways. One potential result for this interplay could be precise development of a specific view, through sketch, followed by rapid iteration through referential interactions expressed using speech. Or, rapid generation of initial visualizations through speech, followed by slower refinement through sketching. This interplay has been explored on tablet devices (25), but it would be interesting to consider this interplay on a large display.

6.4 Future Work

There are several next steps for this work. The first is to extend this technique to new datasets, and to new user communities, which will enhance our understanding of data exploration using natural language and multi-view interactions. The second is to expand upon the multi-modal interaction environment, and consider how to use our view collections formalism to address challenges in multiple coordinate views on large displays and in metavisualizaiton, to more effectively organize visualizations and highlight between view relations. The third, is to consider how this technique adapts to a collaborative context, to see how multiple users coordinate data exploration activities when they are able to offload visualization construction tasks onto a virtual collaborator, who provides visualizations on their behalf. Finally, a next step may be to consider how additional interaction modalities, such as pen and touch, as well as incorporating new systems for natural language interpretation for visualization, such as work by my collaborator Abhinav Kumar, to resolve references and disambiguate actionable from non-actionable utterances (20; 21; 22; 23) or work such as NL4DV (29), can allow for more expressive visualization construction and refinement tasks.

CHAPTER 7

CONCLUSION

In this dissertation I set out to contribute to our understanding of data exploration in large display environments and using multi-modal speech and mid-air gesture inputs.

The contributions of this dissertation are:

- An observational exploratory study characterizing how participants expressed their data exploration intentions through speech and mid-air gestures
- Development of a technique for data exploration using natural language commands, targeted breadth of exploration and multi-view responses
- Development of a multi-modal speech and mid-air gesture interaction technique for data exploration in large display environments

This dissertation, along with work conducted with my collaborator Abhinav Kumar, presents the first multi-modal speech and mid-air gesture technique for data exploration for large display environments. This work contributes toward the vision of realizing an attentive environment for data visualization, which couples abundant display spaces, with multi-modal interactions that allow for movement and collaboration and are conducive to interactions with multiple views, and that leverage technical trends towards larger, higher resolution display environments, systems that respond to natural language inputs and sensors that track user movement and behavior.

APPENDICES

Appendix A

DATA AND PROBLEM DESCRIPTIONS

There are two datasets and problems that were used in the development of this dissertation. The description of these datasets is provided below.

A.1 Chicago crime data

In the pre-design study, described in Chapter 3, participants explored a city of Chicago crime dataset. Their task was to explore this data, and to identify spatial or temporal hot spots, along with interesting patterns of features in the data. This data consisted in a table of crime incidents, each of which contained the following fields:

- An identifier for each crime incident
- A neighborhood: either UIC, the Loop, the Near West side or River North
- A crime type- such as theft, assault, battery, burglary, deceptive practice.
- A location type- such as street, sidewalk, residence, department store, CTA bus stop
- Year, from 2010-2015
- Month (eg. Jan-Dec)
- Day of the week (Mon-Sun)
- Time of the day (either hourly, or morning, afternoon, evening and night).
- GPS coordinate of the crime incident

Appendix A (Continued)

This dataset was also used in the development of Traverse, described in Chapter 4 and of Ditto, described in Chapter 5, and was used in the training. For these applications, single data attributes and pairs of data attributes were associated with a visual template and data retrieval and aggregation method. Categorical attributes (Neighborhood, Crime type, location type) were assigned to frequency bar charts, temporal attributes (year, month of the year, day of the week, time of the day) are assigned to a frequency line chart. Pairs of attributes are assigned to a visual template as well- two categorical attributes are assigned to a frequency heat table, pairs of temporal attributes are assigned to a multi-line chart, and a categorical and temporal pair are assigned to a multi-line chart.

A.2 COVID19 Risk Data

This dataset and problem was drawn from collaborative work with Moira Zellner (136), using a novel COVID19 Risk Index which used machine learning and a number of data points to predict the vulnerability of a community to COVID19. This vulnerability included the risk of a severe outbreak as well as whether the community lacked resources to respond to this outbreak. In the development and evaluation of Traverse and Ditto, we developed a data exploration problem around the task of understanding this risk, in relation to regional differences (such as comparing the Southeast to the Midwest) and in relation to differences between counties of different densities and population levels (using the CDC's classification of counties into categories: urban, suburban, small city or rural). We also incorporated a selection of health and demographic data from the Center for Disease Control (152), and the John's Hopkins COVID19 cases tracker (153).

Appendix A (Continued)

Specifically, this dataset consisted in two tables. The first table, recorded county-level health data and the COVID19 risk prediction (136), with a unique identifier for each county (FIPS). This data was drawn from the Center for Disease Control, binned into equally sized groupings and labeled from 'very high' to 'very low' (see below). The second table, consisted in monthly cases for each county, with the same FIPS unique identifier. This data is as follows:

- County code- a unique identifier for each county (FIPS code)
- COVID19 Risk Index : a ranking of how vulnerable a county is to a severe COVID19 outbreak (136).
  - Very high COVID19 risk index, High COVID19 risk index, Moderate COVID19 risk index, Low COVID19 risk index, Very low COVID19 risk index
- County type
  - Urban, Rural, Suburban, Small City
- Region
  - Northeast, Southeast, Midwest, Southwest, Rockies, Pacific, Non-contiguous
- Access to doctors(eg. number of doctors per capita)
  - Very low access to doctors, Low access to doctors, Moderate access to doctors, High access to doctors, Very high access to doctors
- Uninsured rate(what proportion of the county does not have insurance)

Appendix A (Continued)

  - Very low uninsured rate, Low uninsured rate, Moderate uninsured rate, High uninsured rate, Very high uninsured rate

- Elderly population (Percent 65 and older)
  - Very high elderly percentage, High elderly percentage, Moderate elderly percentage, Low elderly percentage, Very low elderly percentage

- Cardiovascular disease rates
  - Very high cardiovascular disease rates, High cardiovascular disease rates, Moderate cardiovascular disease rates, Low cardiovascular disease rates, Very low cardiovascular disease rates

- Diabetes rates
  - Very high diabetes rates, High diabetes rates, Moderate diabetes rates, Low diabetes rates, Very low diabetes rates

- Poverty rate
  - Very high poverty rate, High poverty rate, Moderate poverty rate, Low poverty rate, Very low poverty rate

The COVID19 case data, consisted in monthly cases from April 2020 through January 2021, for each county. Counties were identified with the same county identifier (FIPS code).

As with the Chicago crime data, single data attributes and pairs of data attributes were associated with a visual template and data retrieval and aggregation method. Ordinal attributes

Appendix A (Continued)

(COVID19 Risk Index, Access to Doctors, Uninsured Rate, Elderly Population, Cardiovascular Disease Rate, Diabetes Rate, Poverty Rate), were assigned to frequency bar charts, temporal attributes (Cases by Date) were assigned to a frequency line chart. Pairs of attributes were assigned to a visual template as well- two ordinal attributes were assigned to a frequency heat table, and an ordinal and temporal pair are assigned to a multi-line chart. In addition, we could display each county in a map, and filter or color this map by the other data attributes in the table.

This dataset was used in the development of Traverse and of Ditto, and was used in the exploratory phase of evaluation. Participants in the study were instructed to explore the data, looking for regional differences or differences between county types, in terms of COVID19 risk and other health and demographic data.

Appendix B

PERMISSION FOR REUSE

Chapter 3 presented previously published work (44). The following presents written permission from the journal's/publisher's website outlining their copyright policies Figure 43.

Appendix B (Continued)

JOHN WILEY AND SONS LICENSE
TERMS AND CONDITIONS

Jul 22, 2021

This Agreement between Jillian Aurisano ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.

License Number	5114030420509
License date	Jul 22, 2021
Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	Computer Graphics Forum
Licensed Content Title	Many At Once: Capturing Intentions to Create And Use Many Views At Once In Large Display Environments
Licensed Content Author	Jillian Aurisano, Abhinav Kumar, Abeer Alsaiari, et al
Licensed Content Date	Jul 18, 2020
Licensed Content Volume	39
Licensed Content Issue	3
Licensed Content Pages	12
Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Title	Interaction with Multiple Data Visualizations Through Natural Language Commands
Institution name	University of Illinois at Chicago
Expected presentation date	Aug 2021

Figure 43: Publishing rights

Appendix C

INSTITUTIONAL REVIEW BOARD

Below is the Institutional Review Board approval letter for the pre-design study, in Chapter 3, Figure 44 and Figure 45.

Below is the Institutional Review Board approval letter for the evaluation of Traverse and Ditto, in Chapter 4 and 5, Figure 46 and Figure 47.

Exemption Granted

July 6, 2021

Barbara Di Eugenio, PhD
Computer Science
Phone: (312) 996-7566 / Fax: (312) 413-0024

RE: **Protocol # 2021-0728**
"Articulate+: Retrospective Analysis of the Articulate data (previously UIC Exempt Research Protocol #2014-0554)"

All data was initially collected under UIC Research Protocol #2014-0554 (UIC Exemption Period July 10, 2014 – June 18, 2020). No additional data will be collected. The remaining research-related activities are limited to the analysis of existing data only.

PIs must complete a COVID-19 Human Subjects Research Review Worksheet for a protocol COVID safety assessment prior to initiating or re-starting any research activities that require in-person contact between research subjects and staff during the COVID-19 pandemic.

For additional information about this process, please refer to the Human Subjects Research Review page on the OVCR website. If you need assistance, questions may be directed to research@uic.edu.

Dear Dr. Di Eugenio:

Your Claim of Exemption was reviewed on **July 6, 2021** and it was determined that your research meets the criteria for exemption as defined in the U.S. Department of Health and Human Services Regulations for the Protection of Human Subjects [45 CFR 46.104(d)].

Exemption Granted Date:	July 6, 2021
Sponsor:	NSF
Institutional Proposal (IP) #:	00542255
Grant/Contract No:	NSF 2007257
Grant/Contract Title:	HS: Small: Collaborative Research: Articulate+ - A Conversational Interface for Democratizing Visual Analysis

The specific exemption category under 45 CFR 46.104(d) is: 2

You are reminded that investigators whose research involving human subjects is determined to be exempt from the federal regulations for the protection of human subjects still have responsibilities for the ethical conduct of the research under state law and UIC policy.

Page 1 of 2

UNIVERSITY OF ILLINOIS AT CHICAGO

Figure 44: This is the first page of the IRB approval letter for the pre-design study, described in Chapter 3.

Please remember to:

→ Use your research protocol number (2021-0728) on any documents or correspondence with the IRB concerning your research protocol.

→ Review and comply with the policies of the UIC Human Subjects Protection Program (HSPP) and the guidance ***Investigator Responsibilities***.

We wish you the best as you conduct your research. If you have any questions or need further help, please contact me at choehne@uic.edu or (312) 355-2908, or the OPRS office at (312) 996-1711. Please send any correspondence about this protocol to OPRS via OPRS Live.

Sincerely,
Charles W. Hoehne
Assistant Director, IRB #7
Office for the Protection of Research Subjects

cc: Robert Sloan, Computer Science

Page 2 of 2

UNIVERSITY OF ILLINOIS AT CHICAGO
Office for the Protection of Research Subjects
201 AOB (MC 672)
1737 West Polk Street
Chicago, Illinois 60612

Figure 45: This is the second page of the IRB approval letter for the pre-design study, described in Chapter 3.

Exempt Research
UIC Amendment #1

April 23, 2021

Jillian Aurisano
Computer Science
Phone: (312) 996-3002 / Fax: (312) 413-7585

RE: **Protocol # 2021-0200**
"Interaction with Multiple Data Visualizations Through Natural Language Commands"

Sponsor:	NSF
Institutional Proposal (IP) #:	2007257
Grant/Contract No:	Not available
Grant/Contract Title:	HS: Small: Collaborative Research: Articulate+ - A Conversational Interface for Democratizing Visual Analysis

Please be reminded of the need to also submit amendments to the non-UIC sites (University of Hawaii at Manoa and Northeastern University).

PIs must complete a COVID-19 Human Subjects Research Review Worksheet for a protocol COVID safety assessment prior to initiating or re-starting any research activities that require in-person contact between research subjects and staff during the COVID-19 pandemic.

For additional information about this process, please refer to the Human Subjects Research Review page on the OVCR website. If you need assistance, questions may be directed to research@uic.edu.

Dear Jillian Aurisano:

The amendment to your exempt research was reviewed on **April 23, 2021**. It was determined that your amended research continues to meet the criteria for exemption as defined in the U.S. Department of Health and Human Services Regulations for the Protection of Human Subjects [45 CFR 46.104(d)].

UIC Amendment Approval Date: April 23, 2021

Summary: UIC Amendment #1: The only change to the proposal is in the role for the personnel at the non-UIC sites. Now they will be included on publications of this work. They will not, however, interact or intervene with participants, nor will they access identifiable data. Instead, they will analyze only de-identified data. They will not be involved in data collection,

Page 1 of 2

UNIVERSITY OF ILLINOIS AT CHICAGO

Figure 46: This is the first page of the IRB approval letter for the evaluation of Traverse and Ditto, described in Chapters 4 and 5.

processing or analysis, but will consult on this study as part of the larger NSF grant funding this study.

The specific exemption categories under 45 CFR 46.104(d) are: 2, 3

You are reminded that investigators whose research involving human subjects is determined to be exempt from the federal regulations for the protection of human subjects still have responsibilities for the ethical conduct of the research under state law and UIC policy.

Please remember to:

- → Use your research protocol number (2021-0200) on any documents or correspondence with the IRB concerning your research protocol.
- → Review and comply with the policies of the UIC Human Subjects Protection Program (HSPP) and the guidance ***Investigator Responsibilities***.

We wish you the best as you conduct your research. If you have any questions or need further help, please contact me at (312) 355-2908 or choehne@uic.edu, or the OPRS office at (312) 996-1711. Please send any correspondence about this protocol to OPRS via OPRS Live.

Sincerely,
Charles W. Hoehne
Assistant Director, IRB #7
Office for the Protection of Research Subjects

cc: Robert Sloan
Andrew E. Johnson

Page 2 of 2

UNIVERSITY OF ILLINOIS AT CHICAGO

Figure 47: This is the second page of the IRB approval letter for the evaluation of Traverse and Ditto, described in Chapters 4 and 5.

CITED LITERATURE

1. Jung-Lin Lee, D., Setlur, V., Tory, M., Karahalios, K., and Parameswaran, A.: Deconstructing categorization in visualization recommendation: A taxonomy and comparative study. arXiv e-prints, pages arXiv–2102, 2021.

2. Heer, J., Van Ham, F., Carpendale, S., Weaver, C., and Isenberg, P.: Creation and collaboration: Engaging new audiences for information visualization. In Information visualization, pages 92–133. Springer, 2008.

3. Card, M.: Readings in information visualization: using vision to think. Morgan Kaufmann, 1999.

4. Tukey, J. W. et al.: Exploratory data analysis, volume 2. Reading, Mass., 1977.

5. Alspaugh, S., Zokaei, N., Liu, A., Jin, C., and Hearst, M. A.: Futzing and moseying: Interviews with professional data analysts on exploration practices. IEEE transactions on visualization and computer graphics, 25(1):22–31, 2018.

6. Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., and Heer, J.: Voyager: Exploratory analysis via faceted browsing of visualization recommendations. IEEE transactions on visualization and computer graphics, 22(1):649–658, 2015.

7. Keim, D. A.: Information visualization and visual data mining. IEEE transactions on Visualization and Computer Graphics, 8(1):1–8, 2002.

8. Russell, D. M., Stefik, M. J., Pirolli, P., and Card, S. K.: The cost structure of sensemaking. In Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems, pages 269–276, 1993.

9. Jankun-Kelly, T., Ma, K.-L., and Gertz, M.: A model for the visualization exploration process. IEEE, 2002.

10. Groth, D. P. and Streefkerk, K.: Provenance and annotation for visual exploration systems. IEEE transactions on visualization and computer graphics, 12(6):1500–1510, 2006.

11. Ragan, E. D., Endert, A., Sanyal, J., and Chen, J.: Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. IEEE transactions on visualization and computer graphics, 22(1):31–40, 2015.

12. Grammel, L., Tory, M., and Storey, M. A.: How information visualization novices construct visualizations. Visualization and Computer Graphics, IEEE Transactions on, 16(6):943–952, 2010.

13. Lee, B., Isenberg, P., Riche, N. H., and Carpendale, S.: Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. IEEE Transactions on Visualization and Computer Graphics, 18(12):2689–2698, 2012.

14. Sun, Y., Leigh, J., Johnson, A., and Lee, S.: Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In Smart Graphics, pages 184–195. Springer, 2010.

15. Gao, T., Dontcheva, M., Adar, E., Liu, Z., and Karahalios, K. G.: Datatone: Managing ambiguity in natural language interfaces for data visualization. In Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, pages 489–500. ACM, 2015.

16. Setlur, V., Battersby, S. E., Tory, M., Gossweiler, R., and Chang, A. X.: Eviza: A natural language interface for visual analysis. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology, pages 365–377. ACM, 2016.

17. Hoque, E., Setlur, V., Tory, M., and Dykeman, I.: Applying pragmatics principles for interaction with visual analytics. IEEE transactions on visualization and computer graphics, 24(1):309–318, 2018.

18. Kassel, J.-F. and Rohs, M.: Valletto: A multimodal interface for ubiquitous visual analytics. In Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, pages 1–6, 2018.

19. Kim, D. H., Hoque, E., and Agrawala, M.: Answering questions about charts and generating visual explanations. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pages 1–13, 2020.

20. Kumar, A., Aurisano, J., Di Eugenio, B., Johnson, A., Gonzalez, A., and Leigh, J.: Towards a dialogue system that supports rich visualizations of data. In Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 304–309, 2016.

21. Kumar, A., Di Eugenio, B., Aurisano, J., Johnson, A., Alsaiari, A., Flowers, N., Gonzalez, A., and Leigh, J.: Towards multimodal coreference resolution for exploratory data visualization dialogue: Context-based annotation and gesture identification. In The 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2017–SaarDial)(August 2017), volume 48, 2017.

22. Kumar, A., Aurisano, J., Di Eugenio, B., and Johnson, A.: Intelligent assistant for exploring data visualizations. In The Thirty-Third International Flairs Conference, 2020.

23. Kumar, A., Di Eugenio, B., Aurisano, J., and Johnson, A.: Augmenting small data to classify contextualized dialogue acts for exploratory visualization. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 590–599, 2020.

24. Setlur, V., Tory, M., and Djalali, A.: Inferencing underspecified natural language utterances in visual analysis. In Proceedings of the 24th International Conference on Intelligent User Interfaces, pages 40–51, 2019.

25. Srinivasan, A., Lee, B., Henry Riche, N., Drucker, S. M., and Hinckley, K.: Inchorus: Designing consistent multimodal interactions for data visualization on tablet devices. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pages 1–13, 2020.

26. Srinivasan, A. and Stasko, J.: Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. IEEE transactions on visualization and computer graphics, 24(1):511–521, 2018.

27. Srinivasan, A., Lee, B., and Stasko, J. T.: Interweaving multimodal interaction with flexible unit visualizations for data exploration. IEEE Transactions on Visualization and Computer Graphics, 2020.

28. Yu, B. and Silva, C. T.: Flowsense: A natural language interface for visual data exploration within a dataflow system. IEEE transactions on visualization and computer graphics, 2019.

29. Narechania, A., Srinivasan, A., and Stasko, J.: Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries. IEEE Transactions on Visualization and Computer Graphics, 27(2):369–379, 2020.

30. Javed, W. and Elmqvist, N.: Explates: spatializing interactive analysis to scaffold visual exploration. In Computer Graphics Forum, volume 32, pages 441–450. Wiley Online Library, 2013.

31. Gratzl, S., Gehlenborg, N., Lex, A., Pfister, H., and Streit, M.: Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. IEEE transactions on visualization and computer graphics, 20(12):2023–2032, 2014.

32. Zgraggen, E., Zeleznik, R., and Drucker, S. M.: Panoramicdata: Data analysis through pen & touch. IEEE transactions on visualization and computer graphics, 20(12):2112–2121, 2014.

33. Knudsen, S. and Hornbæk, K.: Pade: Supporting collaborative visual analysis of patient administrative systems data with a large touch display system. In 2019 IEEE Workshop on Visual Analytics in Healthcare (VAHC), pages 9–16. IEEE, 2019.

34. van den Elzen, S. and van Wijk, J. J.: Small multiples, large singles: A new approach for visual data exploration. In Computer Graphics Forum, volume 32, pages 191–200. Wiley Online Library, 2013.

35. Andrews, C., Endert, A., and North, C.: Space to think: large high-resolution displays for sensemaking. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 55–64. ACM, 2010.

36. Ball, R., North, C., and Bowman, D. A.: Move to improve: promoting physical navigation to increase user performance with large displays. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 191–200, 2007.

37. Ball, R. and North, C.: Realizing embodied interaction for visual analytics through large displays. Computers & Graphics, 31(3):380–400, 2007.

38. Knudsen, S., Jakobsen, M. R., and Hornbæk, K.: An exploratory study of how abundant display space may support data analysis. In Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, pages 558–567. ACM, 2012.

39. Knudsen, S. and Carpendale, S.: Multiple views in immersive analytics. 2017.

40. Reda, K., Johnson, A. E., Papka, M. E., and Leigh, J.: Effects of display size and resolution on user behavior and insight acquisition in visual exploration. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pages 2759–2768. ACM, 2015.

41. Badam, S. K., Srinivasan, A., Elmqvist, N., and Stasko, J.: Affordances of input modalities for visual data exploration in immersive environments. In Workshop on Immersive Analytics, IEEE Vis, 2017.

42. Andrews, C., Endert, A., Yost, B., and North, C.: Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. Information Visualization, 10(4):341–355, 2011.

43. Yost, B. and North, C.: The perceptual scalability of visualization. IEEE Transactions on Visualization and Computer Graphics, 12(5):837–844, 2006.

44. Aurisano, J., Kumar, A., Alsaiari, A., Eugenio, B. D., and Johnson, A.: Many at once: Capturing intentions to create and use many views at once in large display environments. In Computer Graphics Forum, volume 39, pages 229–240. Wiley Online Library, 2020.

45. Aurisano, J., Kumar, A., Gonzales, A., Reda, K., Leigh, J., Di Eugenio, B., and Johnson, A.: Show me data": Observational study of a conversational interface in visual data exploration. In IEEE VIS, volume 15, page 1, 2015.

46. Bolt, R. A.: "put-that-there" voice and gesture at the graphics interface. In Proceedings of the 7th annual conference on Computer graphics and interactive techniques, pages 262–270, 1980.

47. Reda, K., Febretti, A., Knoll, A., Aurisano, J., Leigh, J., Johnson, A., Papka, M. E., and Hereld, M.: Visualizing large, heterogeneous data in hybrid-reality environments. IEEE Computer Graphics and Applications, 33(4):38–48, 2013.

48. Srinivasan, A. and Stasko, J.: Natural language interfaces for data analysis with visualization: Considering what has and could be asked. In Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers, pages 55–59. Eurographics Association, 2017.

49. Büschel, W., Chen, J., Dachselt, R., Drucker, S., Dwyer, T., Görg, C., Isenberg, T., Kerren, A., North, C., and Stuerzlinger, W.: Interaction for immersive analytics. In Immersive Analytics, pages 95–138. Springer, 2018.

50. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In The craft of information visualization, pages 364–371. Elsevier, 2003.

51. Van Ham, F. and Perer, A.: "search, show context, expand on demand": supporting large graph exploration with degree-of-interest. IEEE Transactions on Visualization and Computer Graphics, 15(6):953–960, 2009.

52. Luciani, T., Burks, A., Sugiyama, C., Komperda, J., and Marai, G. E.: Details-first, show context, overview last: supporting exploration of viscous fingers in large-scale ensemble simulations. IEEE transactions on visualization and computer graphics, 25(1):1225–1235, 2018.

53. Bostock, M. and Heer, J.: Protovis: A graphical toolkit for visualization. IEEE transactions on visualization and computer graphics, 15(6):1121–1128, 2009.

54. Bostock, M., Ogievetsky, V., and Heer, J.: D^3 data-driven documents. IEEE transactions on visualization and computer graphics, 17(12):2301–2309, 2011.

55. Wickham, H.: ggplot2. Wiley Interdisciplinary Reviews: Computational Statistics, 3(2):180–185, 2011.

56. Satyanarayan, A., Moritz, D., Wongsuphasawat, K., and Heer, J.: Vega-lite: A grammar of interactive graphics. IEEE transactions on visualization and computer graphics, 23(1):341–350, 2016.

57. Satyanarayan, A., Wongsuphasawat, K., and Heer, J.: Declarative interaction design for data visualization. In Proceedings of the 27th annual ACM symposium on User interface software and technology, pages 669–678. ACM, 2014.

58. Ren, D., Höllerer, T., and Yuan, X.: ivisdesigner: Expressive interactive design of information visualizations. IEEE transactions on visualization and computer graphics, 20(12):2092–2101, 2014.

59. Stolte, C., Tang, D., and Hanrahan, P.: Polaris: A system for query, analysis, and visualization of multidimensional relational databases. IEEE Transactions on Visualization and Computer Graphics, 8(1):52–65, 2002.

60. Lee, B., Smith, G., Riche, N. H., Karlson, A., and Carpendale, S.: Sketchinsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction. In 2015 IEEE Pacific Visualization Symposium (PacificVis), pages 199–206. IEEE, 2015.

61. Xia, H., Henry Riche, N., Chevalier, F., De Araujo, B., and Wigdor, D.: Dataink: Direct and creative data-oriented drawing. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, page 223. ACM, 2018.

62. Huron, S., Jansen, Y., and Carpendale, S.: Constructing visual representations: Investigating the use of tangible tokens. IEEE transactions on visualization and computer graphics, 20(12):2102–2111, 2014.

63. Drucker, S. M., Fisher, D., Sadana, R., Herron, J., et al.: Touchviz: a case study comparing two interfaces for data analytics on tablets. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 2301–2310. ACM, 2013.

64. Tobiasz, M., Isenberg, P., and Carpendale, S.: Lark: Coordinating co-located collaboration with information visualization. IEEE transactions on visualization and computer graphics, 15(6):1065–1072, 2009.

65. Scaife, M. and Rogers, Y.: External cognition: how do graphical representations work? International journal of human-computer studies, 45(2):185–213, 1996.

66. Ballendat, T., Marquardt, N., and Greenberg, S.: Proxemic interaction: designing for a proximity and orientation-aware environment. In ACM International Conference on Interactive Tabletops and Surfaces, pages 121–130. ACM, 2010.

67. Lee, D. J.-L., Dev, H., Hu, H., Elmeleegy, H., and Parameswaran, A.: Avoiding drill-down fallacies with vispilot: Assisted exploration of data subsets. In Proceedings of the 24th International Conference on Intelligent User Interfaces, pages 186–196, 2019.

68. Brehmer, M. and Munzner, T.: A multi-level typology of abstract visualization tasks. IEEE transactions on visualization and computer graphics, 19(12):2376–2385, 2013.

69. Chi, E. H.-h. and Riedl, J. T.: An operator interaction framework for visualization systems. In Proceedings IEEE Symposium on Information Visualization (Cat. No. 98TB100258), pages 63–70. IEEE, 1998.

70. Ware, C.: Information visualization: perception for design. Elsevier, 2012.

71. Satyanarayan, A. and Heer, J.: Lyra: An interactive visualization design environment. In Computer Graphics Forum, volume 33, pages 351–360. Wiley Online Library, 2014.

72. Blunschi, L., Jossen, C., Kossman, D., Mori, M., and Stockinger, K.: Soda: Generating sql for business users. arXiv preprint arXiv:1207.0134, 2012.

73. He, P., Mao, Y., Chakrabarti, K., and Chen, W.: X-sql: reinforce schema representation with context. arXiv preprint arXiv:1908.08113, 2019.

74. Herzig, J., Nowak, P. K., Müller, T., Piccinno, F., and Eisenschlos, J. M.: Tapas: Weakly supervised table parsing via pre-training. arXiv preprint arXiv:2004.02349, 2020.

75. Li, F. and Jagadish, H. V.: Nalir: an interactive natural language interface for querying relational databases. In Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pages 709–712, 2014.

76. Pasupat, P. and Liang, P.: Compositional semantic parsing on semi-structured tables. arXiv preprint arXiv:1508.00305, 2015.

77. Popescu, A.-M., Etzioni, O., and Kautz, H.: Towards a theory of natural language interfaces to databases. In Proceedings of the 8th international conference on Intelligent user interfaces, pages 149–157, 2003.

78. Wang, C., Tatwawadi, K., Brockschmidt, M., Huang, P.-S., Mao, Y., Polozov, O., and Singh, R.: Robust text-to-sql generation with execution-guided decoding. arXiv preprint arXiv:1807.03100, 2018.

79. Hoque, E., Setlur, V., Tory, M., and Dykeman, I.: Applying pragmatics principles for interaction with visual analytics. IEEE transactions on visualization and computer graphics, 24(1):309–318, 2017.

80. Dhamdhere, K., McCurley, K. S., Nahmias, R., Sundararajan, M., and Yan, Q.: Analyza: Exploring data with conversation. In Proceedings of the 22nd International Conference on Intelligent User Interfaces, pages 493–504. ACM, 2017.

81. Dibia, V. and Demiralp, Ç.: Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. IEEE computer graphics and applications, 39(5):33–46, 2019.

82. Tableau: Tableau ask data, 2018.

83. Hanrahan, P.: Vizql: a language for query, analysis and visualization. In Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pages 721–721. ACM, 2006.

84. Aurisano, J., Kumar, A., Gonzalez, A., Leigh, J., DiEugenio, B., and Johnson, A.: Articulate2: Toward a conversational interface for visual data exploration. In IEEE Visualization, 2016.

85. Heer, J., Mackinlay, J., Stolte, C., and Agrawala, M.: Graphical histories for visualization: Supporting analysis, communication, and evaluation. IEEE transactions on visualization and computer graphics, 14(6):1189–1196, 2008.

86. Wright, W., Schroh, D., Proulx, P., Skaburskis, A., and Cort, B.: The sandbox for analysis: concepts and methods. In Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 801–810, 2006.

87. Bavoil, L., Callahan, S. P., Crossno, P. J., Freire, J., Scheidegger, C. E., Silva, C. T., and Vo, H. T.: Vistrails: Enabling interactive multiple-view visualizations. In VIS 05. IEEE Visualization, 2005., pages 135–142. IEEE, 2005.

88. Jakobsen, M. R. and Hornbæk, K.: Is moving improving?: Some effects of locomotion in wall-display interaction. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pages 4169–4178. ACM, 2015.

89. Endert, A., Andrews, C., Lee, Y. H., and North, C.: Visual encodings that support physical navigation on large displays. In Proceedings of Graphics Interface 2011, pages 103–110. Canadian Human-Computer Communications Society, 2011.

90. Bezerianos, A. and Isenberg, P.: Perception of visual variables on tiled wall-sized displays for information visualization applications. IEEE Transactions on Visualization and Computer Graphics, 18(12):2516–2525, 2012.

91. Reda, K., Offord, C., Johnson, A. E., and Leigh, J.: Expanding the porthole: leveraging large, high-resolution displays in exploratory visual analysis. In CHI'14 Extended Abstracts on Human Factors in Computing Systems, pages 2047–2052. ACM, 2014.

92. Isenberg, P., Dragicevic, P., Willett, W., Bezerianos, A., and Fekete, J.-D.: Hybrid-image visualization for large viewing environments. IEEE transactions on visualization and computer graphics, 19(12):2346–2355, 2013.

93. Marrinan, T., Aurisano, J., Nishimoto, A., Bharadwaj, K., Mateevitsi, V., Renambot, L., Long, L., Johnson, A., and Leigh, J.: Sage2: A new approach for data intensive collaboration using scalable resolution shared displays. In Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on, pages 177–186. IEEE, 2014.

94. Langner, R., Kister, U., and Dachselt, R.: Multiple coordinated views at large displays for multiple users: Empirical findings on user behavior, movements, and distances. IEEE transactions on visualization and computer graphics, 25(1):608–618, 2018.

95. Alsaiari, A., Johnson, A., and Nishimoto, A.: Polyvis: Cross-device framework for collaborative visual data analysis. In 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pages 2870–2876. IEEE, 2019.

96. Burks, A., Renambot, L., and Johnson, A.: Vissnippets: A web-based system for impromptu collaborative data exploration on large displays. In Practice and Experience in Advanced Research Computing, pages 144–151. 2020.

97. Aurisano, J., Reda, K., Johnson, A., Marai, E. G., and Leigh, J.: Bactogenie: a large-scale comparative genome visualization for big displays. BMC bioinformatics, 16(11):S6, 2015.

98. Rajabiyazdi, F., Walny, J., Mah, C., Brosz, J., and Carpendale, S.: Understanding researchers' use of a large, high-resolution display across disciplines. In Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces, pages 107–116. ACM, 2015.

99. Liu, C., Chapuis, O., Beaudouin-Lafon, M., Lecolinet, E., and Mackay, W. E.: Effects of display size and navigation type on a classification task. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 4147–4156, 2014.

100. Alsaiari, A., Aurisano, J., and Johnson, A. E.: Evaluating strategies of exploratory visual data analysis in multi device environments. In EuroVis (Short Papers), pages 91–95, 2020.

101. Chung, H., North, C., Joshi, S., and Chen, J.: Four considerations for supporting visual analysis in display ecologies. In 2015 IEEE conference on visual analytics science and technology (VAST), pages 33–40. IEEE, 2015.

102. Knudsen, S. and Carpendale, S.: View relations: An exploratory study on between-view meta-visualizations. In Proceedings of the 9th Nordic Conference on Human-Computer Interaction, page 15. ACM, 2016.

103. Norman, D.: The design of everyday things: Revised and expanded edition. Basic books, 2013.

104. Badam, S. K., Fisher, E., and Elmqvist, N.: Munin: A peer-to-peer middleware for ubiquitous analytics and visualization spaces. IEEE Transactions on Visualization and Computer Graphics, 21(2):215–228, 2014.

105. Badam, S. K., Fisher, E., and Elmqvist, N.: Munin: A peer-to-peer middleware for ubiquitous analytics and visualization spaces. IEEE Transactions on Visualization and Computer Graphics, 21(2):215–228, 2015.

106. Horak, T., Badam, S. K., Elmqvist, N., and Dachselt, R.: When david meets goliath: Combining smartwatches with a large vertical display for visual data exploration. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, page 19. ACM, 2018.

107. Agarwal, M., Srinivasan, A., and Stasko, J.: Viswall: Visual data exploration using direct combination on large touch displays. In 2019 IEEE Visualization Conference (VIS), pages 26–30. IEEE, 2019.

108. Jakobsen, M. R. and Hornbæk, K.: Up close and personal: Collaborative work on a high-resolution multitouch wall display. ACM Transactions on Computer-Human Interaction (TOCHI), 21(2):11, 2014.

109. Hauptmann, A. G.: Speech and gestures for graphic image manipulation. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 241–245, 1989.

110. Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X.-F., Kirbas, C., McCullough, K. E., and Ansari, R.: Multimodal human discourse: gesture and speech. ACM Transactions on Computer-Human Interaction (TOCHI), 9(3):171–193, 2002.

111. Li, X., Guan, D., Zhang, J., Liu, X., Li, S., and Tong, H.: Exploration of ideal interaction scheme on smart tv: Based on user experience research of far-field speech and mid-air gesture interaction. In International Conference on Human-Computer Interaction, pages 144–162. Springer, 2019.

112. Horvitz, E.: Principles of mixed-initiative user interfaces. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pages 159–166. ACM, 1999.

113. Wilkinson, L.: The grammar of graphics. In Handbook of Computational Statistics, pages 375–414. Springer, 2012.

114. Mackinlay, J.: Automating the design of graphical presentations of relational information. Acm Transactions On Graphics (Tog), 5(2):110–141, 1986.

115. Bertin, J.: Semiology of graphics; diagrams networks maps. Technical report, 1983.

116. Mackinlay, J., Hanrahan, P., and Stolte, C.: Show me: Automatic presentation for visual analysis. IEEE transactions on visualization and computer graphics, 13(6):1137–1144, 2007.

117. Roth, S. F., Kolojejchick, J., Mattis, J., and Goldstein, J.: Interactive graphic design using automatic presentation knowledge. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 112–117, 1994.

118. Seo, J. and Shneiderman, B.: A rank-by-feature framework for interactive exploration of multidimensional data. Information visualization, 4(2):96–113, 2005.

119. Wills, G. and Wilkinson, L.: Autovis: automatic visualization. Information Visualization, 9(1):47–69, 2010.

120. Gilson, O., Silva, N., Grant, P. W., and Chen, M.: From web data to visualization via ontology mapping. In Computer Graphics Forum, volume 27, pages 959–966. Wiley Online Library, 2008.

121. Gotz, D. and Wen, Z.: Behavior-driven visualization recommendation. In Proceedings of the 14th international conference on Intelligent user interfaces, pages 315–324. ACM, 2009.

122. Zhou, M. X. and Chen, M.: Automated generation of graphic sketches by example. In IJCAI, volume 3, pages 65–71, 2003.

123. Casner, S. M.: Task-analytic approach to the automated design of graphic presentations. ACM Transactions on Graphics (ToG), 10(2):111–151, 1991.

124. Marks, J., Beardsley, P., Andalman, B., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., et al.: Design galleries: A general approach to setting parameters for computer graphics and animation. In Proceedings of SIGGRAPH. Association for Computing Machinery, 1997.

125. Key, A., Howe, B., Perry, D., and Aragon, C.: Vizdeck: self-organizing dashboards for visual analytics. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pages 681–684, 2012.

126. Chandler, T., Cordeil, M., Czauderna, T., Dwyer, T., Glowacki, J., Goncu, C., Klapperstueck, M., Klein, K., Marriott, K., Schreiber, F., et al.: Immersive analytics. In 2015 Big Data Visual Analytics (BDVA), pages 1–8. IEEE, 2015.

127. Badam, S. K., Srinivasan, A., Elmqvist, N., and Stasko, J.: Affordances of input modalities for visual data exploration in immersive environments. October 2017.

128. Dunne, C., Henry Riche, N., Lee, B., Metoyer, R., and Robertson, G.: Graphtrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history. In Proceedings of the SIGCHI conference on human factors in computing systems, pages 1663–1672. ACM, 2012.

129. Reda, K., Aurisano, J., Febretti, A., Leigh, J., and Johnson, A.: Visualization design patterns for ultra-resolution display environments.

130. Jakobsen, M. R., Haile, Y. S., Knudsen, S., and Hornbæk, K.: Information visualization and proxemics: Design opportunities and empirical findings. IEEE transactions on visualization and computer graphics, 19(12):2386–2395, 2013.

131. Walny, J., Lee, B., Johns, P., Riche, N. H., and Carpendale, S.: Understanding pen and touch interaction for data exploration on interactive whiteboards. IEEE Transactions on Visualization and Computer Graphics, 18(12):2779–2788, 2012.

132. Koytek, P., Perin, C., Vermeulen, J., André, E., and Carpendale, S.: Mybrush: Brushing and linking with personal agency. IEEE transactions on visualization and computer graphics, 24(1):605–615, 2017.

133. Creswell, J. W., Hanson, W. E., Clark Plano, V. L., and Morales, A.: Qualitative research designs: Selection and implementation. The counseling psychologist, 35(2):236–264, 2007.

134. Sedlmair, M., Meyer, M., and Munzner, T.: Design study methodology: Reflections from the trenches and the stacks. IEEE transactions on visualization and computer graphics, 18(12):2431–2440, 2012.

135. Hearst, M. and Tory, M.: Would you like a chart with that? incorporating visualizations into conversational interfaces. In 2019 IEEE Visualization Conference (VIS), pages 1–5. IEEE, 2019.

136. Tiwari, A., Dadhania, A. V., Ragunathrao, V. A. B., and Oliveira, E. R.: Using machine learning to develop a novel covid-19 vulnerability index (c19vi). Science of The Total Environment, 773:145650, 2021.

137. Brewer, C. A.: Color brewer. accessed July 2021, 2021.

138. Knudsen, S. and Carpendale, S.: The visualization view as a toolbox. In ACM CHI 2018 Workshop on Rethinking Interaction, 2018.

139. Aurisano, J., Cha, M., and Snyder, M.: User-driven predictive visual analytics on multivariate, spatio-temporal incident reports. In Proc. IEEE VIS Workshop Vis. Predictive Anal, 2014.

140. Marrinan, T., Nishimoto, A., Insley, J. A., Rizzi, S., Johnson, A., and Papka, M. E.: Interactive multi-modal display spaces for visual analysis. In Proceedings of the 2016

ACM International Conference on Interactive Surfaces and Spaces, pages 421–426, 2016.

141. Jing, P. and Ye-Peng, G.: Human-computer interaction using pointing gesture based on an adaptive virtual touch screen. International Journal of Signal Processing, Image Processing and Pattern Recognition, 6(4):81–91, 2013.

142. Casiez, G., Roussel, N., and Vogel, D.: 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 2527–2530, 2012.

143. Chen, X., Zeng, W., Lin, Y., Ai-Maneea, H. M., Roberts, J., and Chang, R.: Composition and configuration patterns in multiple-view visualizations. IEEE Transactions on Visualization and Computer Graphics, 27(2):1514–1524, 2020.

144. Cetin, G., Stuerzlinger, W., and Dill, J.: Visual analytics on large displays: exploring user spatialization and how size and resolution affect task performance. In 2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA), pages 1–10. IEEE, 2018.

145. Armstrong, Z. and Wattenberg, M.: Visualizing statistical mix effects and simpson's paradox. IEEE transactions on visualization and computer graphics, 20(12):2132–2141, 2014.

146. Gotz, D., Sun, S., and Cao, N.: Adaptive contextualization: Combating bias during high-dimensional visualization and data selection. In Proceedings of the 21st International Conference on Intelligent User Interfaces, pages 85–95, 2016.

147. Guo, Y., Binnig, C., and Kraska, T.: What you see is not what you get! detecting simpson's paradoxes during data exploration. In Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics, pages 1–5, 2017.

148. Zgraggen, E., Zhao, Z., Zeleznik, R., and Kraska, T.: Investigating the effect of the multiple comparisons problem in visual analysis. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pages 1–12, 2018.

149. Wall, E., Blaha, L. M., Franklin, L., and Endert, A.: Warning, bias may occur: A proposed approach to detecting cognitive bias in interactive visual analytics. In 2017 IEEE Conference on Visual Analytics Science and Technology (VAST), pages 104–115. IEEE, 2017.

150. Méndez, G. G., Hinrichs, U., and Nacenta, M. A.: Bottom-up vs. top-down: trade-offs in efficiency, understanding, freedom and creativity with infovis tools. In Proceedings of the 2017 CHI conference on human factors in computing systems, pages 841–852. ACM, 2017.

151. Bishop, F., Zagermann, J., Pfeil, U., Sanderson, G., Reiterer, H., and Hinrichs, U.: Construct-a-vis: Exploring the free-form visualization processes of children. IEEE transactions on visualization and computer graphics, 26(1):451–460, 2019.

152. for Disease Control, C.: Interactive atlas of heart disease and stroke. accessed July 2021, 2021.

153. Dong, E., Du, H., and Gardner, L.: An interactive web-based dashboard to track covid-19 in real time. The Lancet infectious diseases, 20(5):533–534, 2020.

VITA

NAME	Jillian Aurisano
EDUCATION	A.B., Biological Sciences and Religious Studies, University of Chicago, Chicago, IL, 2006
EDUCATION	M.S., Computer Science, University of Illinois at Chicago, IL, 2014
PUBLICATIONS	Jillian Aurisano, Abhinav Kumar, Abeer Alsaiari, Barbara Di Eugenio, Andrew Johnson. "Many at Once: Capturing Intentions to Create And Use Many Views At Once In Large Display Environments" Computer Graphics Forum, EuroGraphics Association (EuroVis 2020).
	Abeer Alsaiari, Jillian Aurisano, Andrew Johnson. "Evaluating Strategies of Exploratory Visual Data Analysis in Multi Device Environments". EuroGraphics Association (EuroVis 2020).
	Abhinav Kumar, Barbara Di Eugenio, Jillian Aurisano, Andrew Johnson. "Augmenting Small Data to Classify Contextualized Dialogue Acts for Exploratory Visualization" In Proceedings of the the Twelfth International Conference on Language Resources and Evaluation (2020).
	Abhinav Kumar, Jillian Aurisano, Barbara Di Eugenio, Andrew Johnson. "Intelligent Assistant for Exploring Data Visualizations" In The Thirty-Third International Flairs Conference (2020).
	Abhinav Kumar, Barbara Di Eugenio, Jillian Aurisano, Andrew Johnson, Alberto Gonzalez and Jason Leigh. "Multimodal Coreference Resolution for Exploratory Data Visualization Dialogue: Context-Based Annotation and Gesture Identification" The 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDIAL). Saarburken, Germany August 2017.
	Luc Renambot, Thomas Marrinan, Jillian Aurisano, Arthur Nishimoto, Victor Mateevitsi, Krishna Bharadwaj, Lance Long, Andrew Johnson, Maxine Brown, Jason Leigh. "SAGE2: a Collaboration Portal for Scalable Resolution Displays" Future Generation Computer Systems (2016)

Abhinav Kumar, Jillian Aurisano, Barbara Di Eugenio, Andrew Johnson."Towards a dialogue system that supports rich visualizations of data" The 17th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL 2016).

Andrew Johnson, Arthur Nishimoto, Alessandro Febretti, Jillian Aurisano."Immersive Geospatial Analytics" NSF Workshop on Geospatial Data Science (2016).

Jillian Aurisano, Khairi Reda, Andrew Johnson, G.Elisabeta Marai, Jason Leigh. "BactoGeNIE: A La¬¬rge-Scale Comparative Genome Visualization for Big Displays" BMC Bioinformatics Journal (BioVis2015).

Jillian Aurisano, Amruta Nanavaty , Isabel Cruz. "Visual Analytics for Ontology Matching Using Multi-Linked Views" Visualizations and User Interfaces for Ontologies and Linked Data (VOILA 2015)

Tuan Dang, Paul Murray, Jillian Aurisano, Angus Forbes "ReactionFlow: An Interactive Visualization Tool for Causality Analysis in Biological Pathways" Proceedings of the 5th Symposium on Biological Data Visualization (BioVis 2015)

Thomas Marrinan, Jillian Aurisano, Arthur Nishimoto, Krishna Bharadwaj, Victor Mateevitsi, Luc Renambot, Lance Long, Andrew Johnson, and Jason Leigh "SAGE2: A New Approach for Data Intensive Collaboration Using Scalable Resolution Shared Displays." IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2014) [**BEST PAPER AWARD**]

Khairi Reda, Alessandro Febretti, Aaron Knoll, Jillian Aurisano, Jason Leigh, Andrew Johnson, Mike E. Papka and Mark Hereld. 'Visualizing Large, Heterogenous Data in Hybrid Reality Display Environments.' ." Computer Graphics and Applications, IEEE Computer Society, (2013)

Molly Schumer, Ruthie Birger, Chayant Tantipathananandh, Jillian Aurisano, Marco Maggioni, Peter Mwangi. "Infestation by a Common Parasite is correlated with Ant Symbiont Identity in Plant-Ant Mutualism." Biotropica (2013)

Jillian Aurisano, Miriam Cha, Michael Snyder. "User-Driven Predictive Visual Analytics on Multivariate Spatio-temporal Incident Reports". Visualization for Predictive Analytics workshop at IEEE VisWeek (2014)

Khairi Reda, Jillian Aurisano, Alessandro Febretti, Andrew Johnson, Jason Leigh. "Visualization Design Patterns for Ultra-Resolution Display Environments." Workshop on Visualization Infrastructure and Systems Technology at SuperComputing13 in September 2013.

POSTERS

Jillian Aurisano, James Hwang, Andrew Johnson, Lance Long, Margaret Crofoot, Tanya Berger-Wolf. "Bringing the Field into the Lab: Large-Scale Visualization of Animal Movement Trajectories Within a Virtual Island."Presented at the Large Data Analysis and Visualization (LDAV) symposium at IEEE VisWeek in Vancouver, CA in October 2019. [**BEST POSTER AWARD**]

Jillian Aurisano, Abhinav Kumar, Abeer Alsaiari, Barbara Di Eugenio, Andrew Johnson. "Evaluation of scalable interactions over multiple views in large display environments." Presented at the Information Visualization conference at IEEE VisWeek in Vancouver, CA in October 2019.

Jillian Aurisano, Abhinav Kumar, Alberto Gonzales, Jason Leigh, Barbara Di Eugenio, Andrew Johnson. "Articulate2: Toward a Conversational Interface for Visual Data Exploration." Presented at the Information Visualization conference at IEEE VisWeek in Baltimore, MD in October 2016.

Jillian Aurisano, Abhinav Kumar, Alberato Gonzales, Khari Reda, Jason Leigh, Barbara Di Eugenio, Andrew Johnson. "Show Me Data." Observational Study of a Conversational Interface in Visual Data Exploration". Presented at the Information Visualization conference at IEEE VisWeek in Chicago, IL in October 2015. [**HONORABLE MENTION**]

Jillian Aurisano, Amruta Nanavaty , Isabel Cruz. "AlignmentVis: Visual Analytics for Ontology Matching." Presented at the Information Visualization conference at IEEE VisWeek in Chicago, IL in October 2015.

Jillian Aurisano, Khairi Reda, Jason Leigh, Andrew Johnson. "Bacterial Gene Neighborhood Investigation Environment: A Large-Scale

Genome Visualization for Big Displays". Presented at the Large Data Analysis and Visualization conference at IEEE VisWeek in Paris, France on November 9th. [**BEST POSTER AWARD**]

Jillian Aurisano, Khairi Reda, Jason Leigh, Andrew Johnson. "Bacterial Gene Neighborhood Investigation Environment: A Large-Scale Genome Visualization for Big Displays". Presented at the Biological Data Visualization conference in Boston, MA on July 11th.

Jillian Aurisano, James Radosevich, Jason Leigh. "Toward Systems-Level Visualizations of Molecular Networks on Large-Scale, High-Resolution Displays". Presented at the IEEE Symposium on Biological Data Visualization in Providence, RI on Oct. 23rd during IEEE VisWeek 2011.

WORKSHOPS

Jillian Aurisano, Lance Long, Andrew Johnson. "An Immersive Dialogue with Data". To be presented at the IEEE Workshop on VisFutures: Designing Fiction Methods for Envisioning Tomorrow's Visualizations in a virtual conference on Oct. 26th during IEEE VisWeek 2020.

TALKS

Jillian Aurisano. "Designing Visual Interfaces for Data-Driven Science", presented at University of Cincinnati on Mar 6, 2020.

Jillian Aurisano. "Scalable, multi-modal speech+gesture interaction through metavisualization in immersive analytics", presented at IEEE Vis Week doctoral colloquium on Oct 19 2019.

Jillian Aurisano. "Comparative genomics, visualization and big displays." Presented Art of Science Chicago on June 9, 2016.

Jillian Aurisano "Visual Analytics, Visualization Technology and the Research Pipeline", talk presented at Monsanto lunchtime symposium on July 14th, 2011 at Monsanto-Chesterfield, Chesterfield, MO.

POSTERS

Jason Leigh, Maxine Brown, Andrew Johnson, Luc Renambot, Thomas Marrinan, Jillian Aurisano, Arthur Nishimoto, Victor Mateevisti, Krishna Bharadwaj. "System and methods for facilitating use of various forms of displayed content." U.S. Patent Application 62/011,855, filed June 2014.

AWARDS

Best Poster, Symposium on Large Data Analysis and Visualization (LDAV) 2019

UIC Image of Research Honorable Mention, 2016

UIC Chancellor's Graduate Research Award, 2016-2017

Poster Honorable Mention, IEEE InfoVis 2016

UIC Chancellor Student Service and Leadership Award, 2015

Best Poster, Symposium on Large Data Analysis and Visualization (LDAV) 2014

Best Paper, International Conference on Collaborative Computing, Networking and Applications 2014

UIC Computer Science travel award to Grace Hopper Celebration of Women in Computing, 2014

UIC Computer Science Graduate Student Service Award, 2014

Grace Hopper Celebration of Women in Computing travel scholarship, 2013

Google Anita Borg Memorial Award recipient, 2012

Bachelor's degree conferred with honors, June 2006

Dean's List 2002-2006

Howard Hughes Medical Institute Summer Research Fellowship, Summer 2004

First-Year Undergraduate Summer Research Fellowship, Summer 2003