UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

TRANSLATING VISUALIZATION INTERACTION INTO NATURAL LANGUAGE

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

MARYAM NAFARI
Norman, Oklahoma
2014

TRANSLATING VISUALIZATION INTERACTION INTO NATURAL LANGUAGE

A DISSERTATION APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY

Dr. Chris Weaver, Chair

Dr. Matthew Jensen

Dr. Qi Cheng

Dr. Chen Ling

Dr. K.Thulasiraman

For my dad.
Kheily jat khaliye

Acknowledgements

I would like to thank my faculty adviser Dr. Chris Weaver for his valuable advice, patience, and continuous supervision throughout my PhD studies. I truly appreciate the challenges you put me through to make this work happen.

Thanks to the members of my dissertation committee—Matthew Jensen, Qi Cheng, Chen Ling, K. Thulasiraman, and May Yuan—for their valuable suggestions and generous time.

Thanks to Dr. Sridhar Radhakrishnan for encouraging and supporting me to pursue my goals academically and professionally. Your words mean a lot to me.

I am also thankful to the Computer Science Staff, particularly Virginie Perez Woods and Emily Pierce, for making graduate life much easier for me and all other students.

Thanks to the Computer Science department for their generous scholarships over the years which made me focus more on the research.

Cheers to all my friends in Oklahoma—Brenna, Joohee, Estapraq, Nathalie, Hoda, and Sayantani—who made the PhD one of the best experiences of my life.

To my lovely family—my mom, Kave, Ghazaleh, and Saman—for always being there for me and encouraging me to follow my dreams.

Of course I am thankful of my dad, who is the reason that I have a chance to write this acknowledgement. I know from the bottom of my heart that you were always with me in spirit and thank you for all the dreams you had for me and everything you have done for me.

Most of all, I would like to thank my husband, Tristin Forbus, who is my friend, officemate, classmate, labmate, roommate, dancemate, etc. Thank you for your infinite support and love, and I could not do this without you.

Table of Contents

List of Tables

List of Figures

Abstract

Richly interactive visualization tools are increasingly popular for data exploration and analysis in a wide variety of domains. Recent advancements in data collection and storage call for more complex analytical tasks to make sense of readily available datasets. More complicated and sophisticated tools are needed to complete those tasks. However, as these visualization tools get more complicated, it becomes increasingly difficult to learn interaction sequences, recall past queries asked from a visualization, and correctly interpret visual states to forage the data. Moreover, the high interactivity of such tools increases the challenge of connecting low-level acquired information to higher-level analytical questions and hypotheses to support, reason, and eventually present insights. This makes studying the usability of complex interactive visualizations, both in the process of foraging and making sense of data, an essential part of visual analytic research. This research can be approached in at least two major ways. One can focus on studying new techniques and guidelines for designing interactive complex visualizations that are easy to use and understand. One can also focus on keeping the capabilities of existing complex visualizations, yet provide supporting capabilities that increases their usability. The latter is an emerging area of research in visual analytics, and is the focus of this dissertation.

This dissertation describes six contributions to the field of visual analytics. The first contribution is an architecture of a query-to-question supporting system that automatically records user interactions and presents them contextually using natural written language. The architecture takes into account the domain knowledge of experts/designers and uses natural language generation (NLG) techniques to translate and transcribe a progression of interactive visualization states into a log of text that can be visualized.

The second contribution is query-to-question (Q2Q), an implemented system that translates low-level user interactions into high-level analytical questions and presents them as a log of styled text that complements and effectively extends the functionality of visualization tools.

The third contribution is a demonstration of the beneficial effects of accompanying a visualization with a textual translation of user interaction on the usability of visualizations. The presence of the translation interface produces considerable improvements in learnability, efficiency, and memorability of visualization in terms of speed and the length of interaction sequences that users perform, along with a modest decrease in error ratio.

The fourth contribution is a set of design guidelines for translating user interactions into natural language, taking into account variation in user knowledge and roles, the types of data being visualized, and the types of interaction supported.

The fifth contribution is a history organizer interface that enables users to organize their analytical process. The structured textual translations output from Q2Q are input into a history organizer tool (HOT) that imposes reordering, sequencing, and grouping of the translated interactions. HOT provides a reasoning framework for users to organize and present hypotheses and insight acquired from a visualization.

The sixth contribution is a demonstration of the efficiency of a suite of arrangement options for organizing questions asked in a visualization. Integration of query translation and history organization improves users' speed, error ratio, and number of reordering actions performed during organization of translated interactions. Overall, this dissertation contributes to the analysis and discovery of user storytelling patterns and behaviours, thereby paving the way to the creation of more intelligent, effective, and user-oriented visual analysis presentation tools.

Chapter 1

Introduction

1.1 Overview

Visual analysis tools are increasingly used to make sense of large amounts of multidimensional information in a wide variety of knowledge domains. Users of these tools analyze and progressively make sense of a dataset by traversing a set of cognitive stages that comprise the sensemaking loop [1]. In analytic research, Pirolli and Card [1] define the main two cognitive stages in the information seeking loop to be *foraging* and *sensemaking*. This categorization can effectively be extended to data analysis using visualizations as well. In a visualization context, foraging refers to the process of searching, filtering, and extracting information from a visualization. Sensemaking refers to the process of transforming the foraged data into knowledge for constructing and testing hypotheses and eventually presenting gained insights.

The high dimensionality of many data sets calls for complex visualization designs composed of multiple views. In such visualizations, views are often coordinated to determine how the appearance and behavior of data dimensions in each view depend on navigation and selection in other views. Coordinated views are often equipped with interaction techniques to enable users to forage data sets by brushing, drilling down, using overview+detail, semantic zooming, and synchronized scrolling [**?**, 2].

Coordination supports making sense of the data by revealing patterns and complex relationships across and within data dimensions.

Despite the powerful capabilities that coordinated multiple view visualization techniques provide for analyzing data, they introduce usability issues. As visualization tools become more structurally sophisticated, it becomes increasingly difficult for users to interpret visualization states, remember past queries, and grasp the full analysis potential of composed interactions to forage the data. Moreover, the high interactivity of such tools increases the challenge of connecting low-level acquired information to higher-level analytical questions and hypotheses in order to support, reason, and present acquired insight. These usability issues suggest the need for supplementary visual tools to communicate information about current and past visualization states and the interactions that bridge them. Such tools could help make rich visual query interfaces accessible to a much wider and more diverse community of users.

Capturing the history of user interactions is one step toward expanding accessibility. The history of user interactions, which is constructed through analytical steps users take sequentially by interacting with a visualization, can also be referred to as *provenance*. In studying provenance, the highly interactive nature of complex visualization tools warrants attention to how users utilize interactions to accomplish exploration and analysis tasks. Research has focused primarily on recording a history of interactions for *later* use, such as to provide a means to recreate past visualization states (e.g., [3]), share the analytical process (e.g., [4]), or analyze users' reasoning and rationale (e.g., [5]). Little attention has been paid to taking advantage of ongoing interaction *during* an analysis session for learning a new visualization, recalling past states, or improving the reliability of insight acquired.

Approaches that focus on recording a user's visual activities are generally classified into two categories: manual or automatic [6]. Manual recording of provenance is done by users themselves and is effective for capturing their high-level insight (e.g., [7–10]). However, it is distracting, time-consuming, and might result in inconsistent and sometimes inaccurate records of provenance. Conversely, automatic approaches capture a comprehensive and consistent record of user activities (e.g., [4, 5, 11–13]). Such systems generally capture only low-level user interactions such as mouse movements or clicks. Exhaustive recording of such events often results in enormous datasets that must be understood and managed to be most useful. A more effective approach needs to be developed to automatically record user interactions, yet capture the corresponding semantics of those interactions and present them in a way that is understandable and reusable. This would broaden the usage of provenance data from reconstructing visualization states to augmenting the foraging and sensemaking processes of data analysis.

1.2 Motivation

Consider a visualization of migrant boat interdictions (shown in Figure 1.1) which was presented in VAST Challenge 2008 [14] and created using Improvise [15]. The visualization shows data related to illegal migration to the United States from an imaginary island off the coast of Florida.

This visualization is a good example of complicated coordinated multiple view design used to evaluate geographical and temporal migration patterns. It has several types of views: table view, map, calendar view, scatter plot, and time series. It incorporates many interaction techniques, such as cross-filtering, selection, dynamic sliding, and panning and zooming. It visualizes a variety of data types including nominal, categorical, temporal, geospatial, and numerical. The comprehensiveness of the visualization, in displayed data dimensions and supported interactions, gives users the ability to effectively drill down into the data and analyze complex relationships between the dimensions.

The views in Figure 1.1 are labeled to further illustrate some of the capabilities of the visualization. As shown in Figure 1.1, users can:

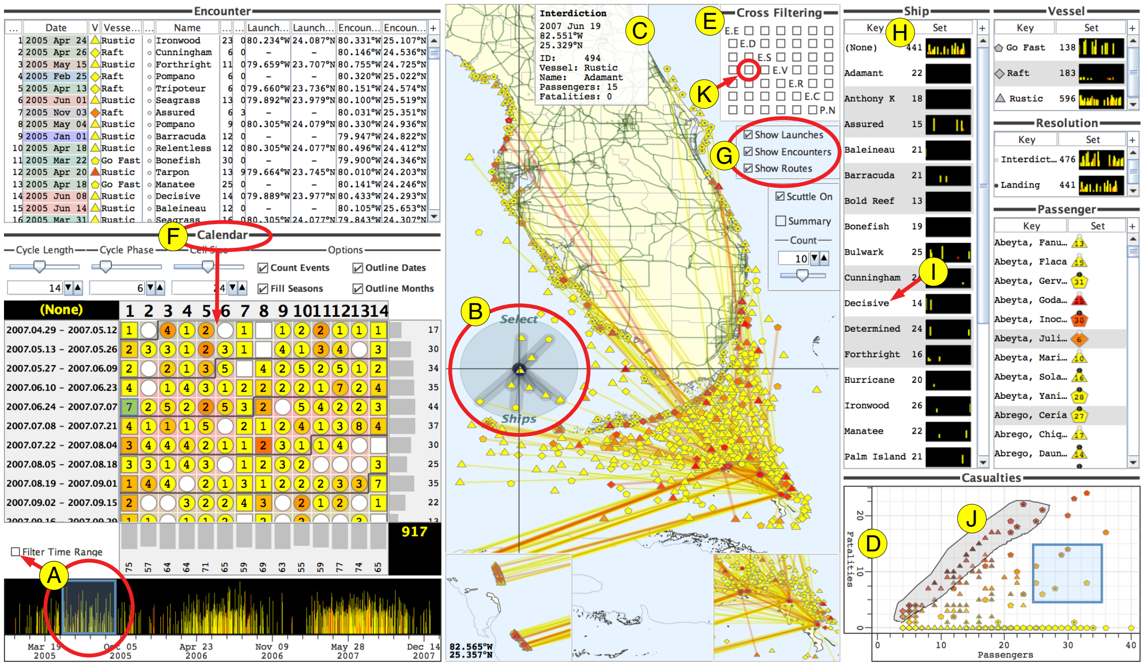


Figure 1.1: Migrant boat interdictions visualization

(A) dynamically filter Vessels, coast guard Ships, Resolutions, and Passenger views on Time;

(B) interact with the map to get the spatial characteristics of the target vessel and nearby vessels within a given radius;

(C) extract detailed information about individual vessels such as when, where, and what type of vessel was involved with an event, and how many passengers the vessels were carrying;

(D) filter range of fatalities on different dimensions such as Time, coast guard Ship, type of Vessels, type of Resolutions, and Passengers;

(E) select and filter multiple dimensions on one anther and extract the complex relationships among them;

(F) navigate the calendar view to get detailed information about events in certain dates; and

(G) express their preferences for data representation.

Despite the richness and high utility of visualizations like the one in Figure 1.1, it is easy to imagine users having trouble utilizing them due to the visualizations' complexity; it might take several interactions until they realize how the visualization can be helpful in their analysis process and how it can reveal the relationships between data attributes. People also tend to forget their sequence of interactions after few visualization states [16]. If users are not able to remember what they are looking for or how they have achieved certain results, they might have difficulty continuing their analytical process.

Consider an example of a sequence of queries performed on the visualization in Figure 1.1, shown in Figure 1.2 and 1.3. This example illustrates the challenge of keeping track and understanding the meaning of each visualization state after a series of interactions:

(A) Selecting *Anthony K* from the Ship table (Figure 1.2, label A)

(B) Filtering the Calendar view on the Ship table (Figure 1.2, label B)

(C) Selecting *April 14th, 2005* from the Calendar view (Figure 1.3, label C)

(D) Filtering Vessel table on the Calendar view (Figure 1.3, label D)

(E) Filtering the Vessel table on the Ship table (Figure 1.3, label E)

Looking at the steps, it might not be clear what information is requested from the visualization. This sequence of interactions results in a state showing the vessels caught by *Anthony K* on *April 14th, 2005*. Even though selection of data items and toggling of checkboxes to apply filtering are the only interactions required to compose the queries presented in the example, it can be difficult to comprehend the meaning of the interactions in the context of the domain being visualized. Moreover, users might have trouble remembering primary interactions and visualization states after performing several interactions (e.g., by the fifth interaction (labeled as E), the first interaction (labeled as A) might be forgotten). The memorability and interpretability of visualization states can be even more challenging in visual analysis tools that are structurally more complicated than the boat visualization.

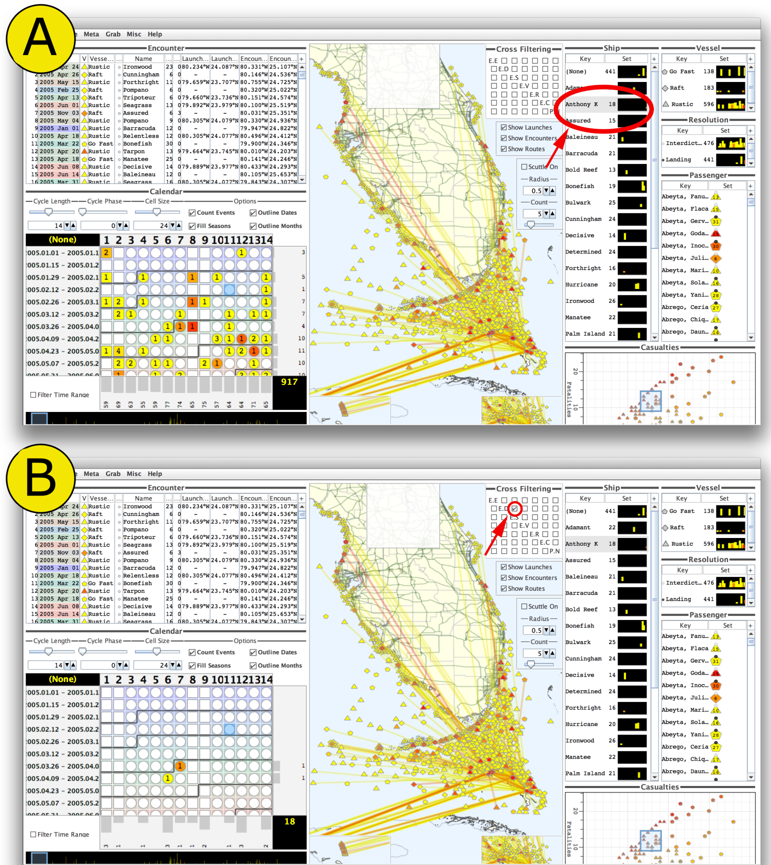


Figure 1.2: Sequence of interactions performed on the migrant boat visualization: (A) Selecting *Anthony K* from the Ship table, (B) Filtering the Calendar view on the Ship table.

The visualization presented in the earlier example can be accompanied by a visual log of text, as illustrated in Figure 1.4. The log dynamically grows as the user performs interactions. Explicit textual representation of user interactions can assist users in making sense of how the interactions they perform correspond to changes in visualization state. These individual inquiries can be combined to form a bigger analytical picture, which promotes pattern discovery and reasoning.

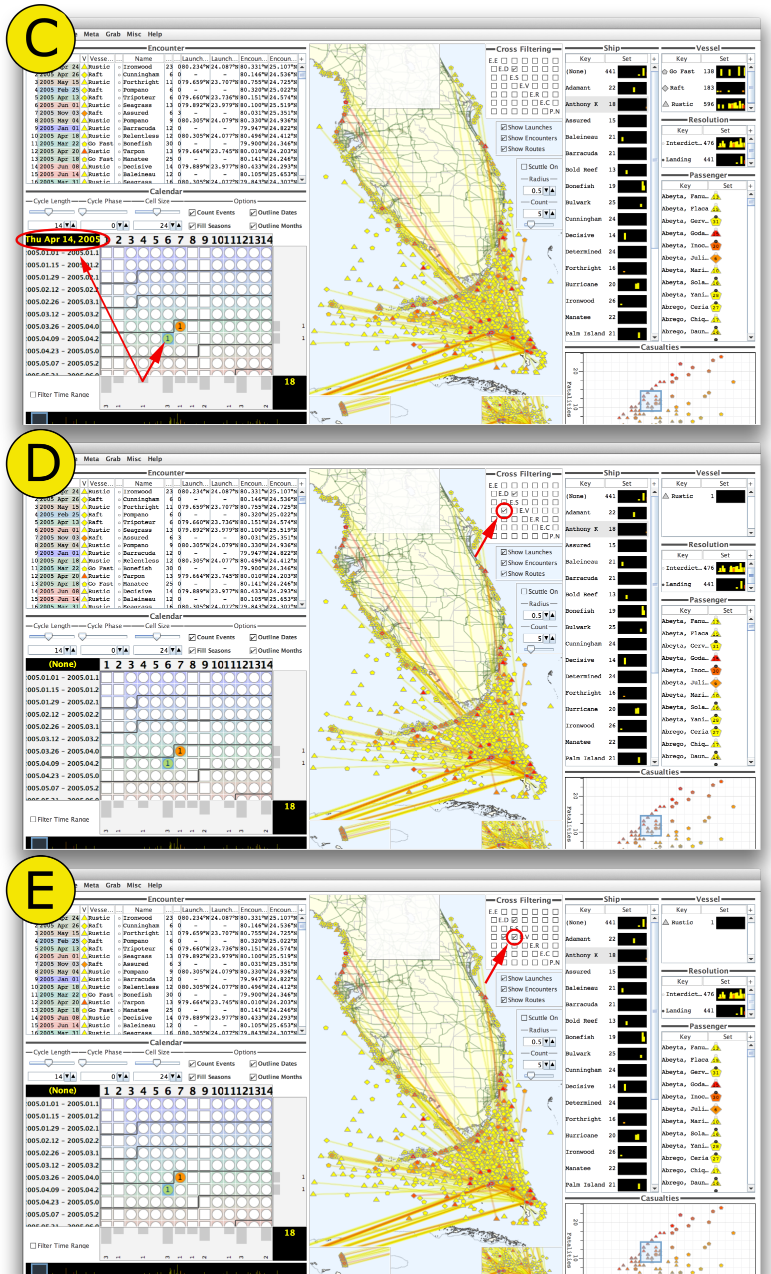


Figure 1.3: Sequence of interactions performed on the migrant boat visualization: (C) Selecting *April 14th 2005* from the Calendar view, (D) Filtering Vessel table on the Calendar view, (E) Filtering the Vessel table on the Ship table.

```
Select Anthony K              -> For Anthony K
Filter Calendar on Ship       -> On what date the ship, Anthony K, interdicted a vessel?
Select April 14th 2005        -> On April 14th 2005
Filter Vessel on Calendar     -> What type of vessels were involved with events on April 14th 2005?
Filter Vessel on ship         -> What type of vessels were interdicted by ship Anthony K on April 14th 2005?
```

Figure 1.4: Logged interactions

1.3 Translation of User Interaction

This dissertation addresses the problem of enhancing usability and utility of complex coordinated visualizations throughout the sensemaking process. The general approach is to utilize users' visual interactions and transform them in a way so as to (1) express the meaning of individual visualization states and their relationships in a bigger analytical context, and (2) construct semantically rich and atomic pieces of information from the interactions that can be used in presentation and storytelling. A textual history of translated interactions can be an easy-to-read reflection of salient observations and low and high-level decisions made throughout the analytical process. It can also expand the usage of visual analysis tools beyond just acquiring certain information, by providing a facility for remembering, reasoning, recreating, sharing, analyzing, and presenting the process of developing hypotheses and reaching conclusions.

In contrast with existing approaches—manual and automatic— used to capture history of user interactions, this dissertation introduces a hybrid approach that automatically record lower-level interactions, then translate the captured intentions into a written language such as English [17]. This approach is able to comprehensively record user interactions as in existing automatic approaches, and capture users' high-level query intentions as in manual approaches, but with little or no end-user involvement.

Following this approach, a user interaction translation architecture is designed and a *Query-to-Question* (Q2Q) system is implemented to capture and translate user interactions into natural language questions. Q2Q uses natural language generation (NLG) techniques to record and display the provenance of interactions and intentions as a visual log of formatted text. The choice of translating user interactions into questions rather than sentences is a means to engage inquisitiveness. The questions work as a way to motivate users to learn more about provided visual representations and interactions and to utilize them for analysis in a more cognizant and effective manner.

Developing a general system for visualization interaction translation is extraordinarily challenging. The design space of visual representations and interaction techniques is large and growing. One direction of generalization is support across knowledge domains. For the system to be widely applicable, one must carefully choose how to interpret interactions and construct language to produce appropriate text that works well both within and across domains. Subtle differences in translation of domain-specific data relationships to an understandable log, even for common visualization techniques, make this path of generalizability both complicated and deeply interesting. Another direction is support for more of the widely varying types of visualization interactions and queries. New language constructions are needed to express the structure of these interactions. There is an enormous space of translation design to explore.

The design of an architecture for Q2Q focuses on supporting *generalizability to many knowledge domains for a set of selected interaction techniques*. In focusing on generalizability across applications, the family of visualization techniques that deals with nominal/categorical data types are studied and Q2Q is developed in the domain-independent cross-filtering views technique [18]. In cross-filtering, users can brush items within data dimensions and filter between data dimensions to dissect multidimensional relationships. However, its usability suffers from a *selection occlusion* effect—in which selected items both cross-filter other views and are themselves cross-filtered out—as well as an *out of sight, out of mind* effect—in which the meaning of visualization states are forgotten after a few subsequent interactions. This mix of benefits and drawbacks make cross-filtering well-suited to examine how Q2Q supports question-centric reasoning, action, verification, and presentation of intention.

The translations that this architecture provides are independent of, yet customizable to, various domains. Though full generalizability of the system to support the large space of all common interaction techniques is beyond the scope of this dissertation, the design of the system is done in a way that is expandable to support new interaction techniques. This approach allows formation of a structured set of text fragments that can be handed to a storytelling tool for rearranging and presenting a history of user activities. This dissertation also describes the design and implementation of an initial storytelling application called the History Organizer Tool (HOT). HOT provides an environment for users to manipulate the questions they ask in visualizations and organize those questions to present and share their analytical process. The question-centric approach to user interaction translation in Q2Q, and to rearrangement and presentation in HOT, aims to increase the usability of visualization by having questions rather than interaction serve as the focus of analytical reasoning and action throughout the sensemaking process.

1.4 Thesis Statement and Research Contribution

My thesis statement is as follows: A domain-independent natural language generation system can be built to translate user interactions in visualizations into structured, contextual questions. Augmenting coordinated multiple view visualizations with a user interaction translation interface improves learnability, efficiency, memorability, and user satisfaction of visualizations. Structured user interaction translations can be used to construct a history organizer tool, which imposes ordering, sequencing, and grouping of the translations. Integration of translation and the history organization improves users' speed, error ratio, and number of reordering actions during rearrangement of their visual activities. Overall, *accompanying multiple coordinated view visualizations with user interaction translation and organization systems enhances data foraging and sensemaking during and after visual analysis.*

This dissertation describes six contributions to the field of visual analytics. The first contribution is an architecture of a query-to-question supporting system that automatically records user interactions and presents them contextually using natural written language. The architecture takes into account the domain knowledge of experts/designers and uses natural language generation (NLG) techniques to translate and transcribe a progression of interactive visualization states into a log of text that can be visualized.

The second contribution is query-to-question (Q2Q), an implemented system that translates low-level user interactions into high-level analytical questions and presents them as a log of styled text that complements and effectively extends the functionality of visualization tools.

The third contribution is a demonstration of the beneficial effects of accompanying a visualization with a textual translation of user interaction on the usability of visualizations. The presence of the translation interface produces considerable improvements in learnability, efficiency, and memorability of visualization in terms of speed and the length of interaction sequences that users perform, along with a modest decrease in error ratio.

The fourth contribution is a set of design guidelines for translating user interactions into natural language, taking into account variation in user knowledge and roles, the types of data being visualized, and the types of interaction supported.

The fifth contribution is a history organizer interface that enables users to organize their analytical process. The structured textual translations output from Q2Q are input into a history organizer tool (HOT) that imposes reordering, sequencing, and grouping of the translated interactions. HOT provides a reasoning framework for users to organize and present hypotheses and insight acquired from a visualization.

The sixth contribution is a demonstration of the efficiency of the suite of arrangement options for organizing questions asked in a visualization. Integration of query translation and history organization improves users' speed, error ratio, and number of reordering actions performed during organization of translated interactions. Overall, this dissertation contributes to the analysis and discovery of user storytelling patterns and behaviours, thereby paving the way to the creation of more intelligent, effective, and user-oriented visual analysis presentation tools.

1.5 Research Questions and Scope

The research presented in this dissertation must answer several questions raised by the thesis.

First, the issue of usability of complex coordinated visualizations has been overlooked in past studies of visual analytics tools. To increase the utility of visualizations, user interaction provenance has been used to recreate visualization states and analyze users' rationale. Can user interaction provenance be used to increase the usability of visualizations during analytical sessions? How can user interactions be presented to be easy to follow and comprehend? Can natural language generation be used to translate user interactions? Can a natural language generation architecture be designed that is domain-independent, yet customizable to particular data domains?

Second, how can a supporting translation tool be built for user visual interactions? What are the requirements for the translation system? What is a useful design of its user interface? What are the expected capabilities of the interface?

Third, how does the textual translation of user interaction affect usability of visualizations? What aspects of usability can be affected and to what extent? Does visualization or familiarity with a domain affect the usefulness of textual translation?

Fourth, what are the considerations in designing a user interaction translation system? In what ways do types of data, types of interactions, and individual differences between end-users affect the translation process? What are the challenges and limitations in designing a user interaction translation system?

Fifth, how can users organize their thoughts and analytical process after interacting with a visualization? Can they present the steps they took to gain certain insight using textual translations of their interactions? How can automatic reordering and grouping be applied to rearrange the textual translations? What are the possible reordering options?

Sixth, which types of reordering options, such as temporal, causal, or free, are more useful? Can a relationship graph representing interaction contextual associations be useful in ordering and finding relevant information? What are the users' preferences while arranging the questions for presentation? What are the other rearrangement behaviors to be considered?

1.6 Organization of the Dissertation

The remaining chapters in the dissertation are organized as follows.

Chapter 2 provides an overview of previous studies in user interaction provenance, natural language techniques, and applications of NLG in visual analytic research, and states the contributions of this dissertation within the existing literature.

Chapter 3 describes the architecture of a user interaction translation system, its modules, and the implemented translation system, Q2Q.

Chapter 4 describes a study of the effects of pairing a visualization with a Q2Q interface on several aspects of usability.

Chapter 5 discusses opportunities and challenges in automatically translating a progression of interactive visualization states into a "visual logbook" of generated text that complements and extends the functionality of visualization tools; focusing on its support for cross-examination and query validation. This chapter also elucidates the challenges by identifying several key factors that strongly influence the generation procedure and the final text, including differences in user intention and usage of text, user level of knowledge, forms of interaction, and data types.

Chapter 6 describes a history organizer tool that enables users to organize their thoughts and steps taken during their analysis using a visualization. It presents several graph models of the relationships between analytical questions asked from a visualization and, describes how the graph models are implemented for automatic reordering and grouping.

Chapter 7 describes a user study of different ordering and rearranging options for organizing users' questions and demonstrates their usefulness in telling a short story about data dimensions in a visualization. Further, it describes the discovery of other rearrangement patterns that users follow during storytelling tasks.

Chapter 8 concludes with a discussion of the benefits and limitations of user interaction translation for increasing the usability of visualizations, and provides an outline of possible future work and extensions.

Chapter 2

Background and Related Work

2.1 Visual Analytics

As Thomas and Cook stated in [19], visual analytics is the science of utilizing interactive visual interfaces for analytical reasoning. In recent years, the data available is rapidly growing due to advancement in data collection and storage technologies. However, the extensive amount of data challenges the ability of effective analysis of these valuable source of information. In response to increasing analytical requirements, interactive visualization tools are becoming indispensable for making sense of complex data. The sophisticated and intuitive visual interfaces accompanied with various interaction techniques enable analysts to directly manipulate and interact with data to make well-informed decisions. In the next two subsections, background and knowledge about visualizations and commonly used interactions are provided. Then, related work in capturing and presenting user interactions are discussed. Finally, existing applications of natural language generation (NLG) in visualization research are presented and contributions of this dissertation are stated.

2.2 Visualization

Visualization research is about designing effective ways to graphically display data. The main goal of visual representation of data is to meaningfully communicate the information carried by large datasets—often containing multiple and diverse data attributes—to users.

Visualization techniques can be classified as static or interactive. Static visualizations are an informative image of data. A well-known example of an effective static visualization showing multiple dimensions is Minard's map of Napolean's March on Moscow [20]. Even though static visualizations are often useful in representing relationships between a few data attributes, their effectiveness dramatically decreases as the number of attributes and amount of data grows. Interactive visualization can overcome limitations in representation to some extent by giving control to the user over what information is shown at any given time.

Visualizations can also be categorized as scientific visualizations or information visualizations based on the kind of data they display. Scientific visualizations mostly focus on quantitative data in fields such as medicine [21], meteorology [22], geography [23], and biomedical engineering [24]. These data generally have spatial characteristics and consist of a large number of records. The immenseness of the scientific data records limits the interactivity of these types of visualizations.

Information visualization deals with representing abstract data using graphical encoding. One difference between scientific visualization and information visualization is that scientific visualization are spatially oriented whereas information visualization may incorporate spatial representation if it suits the data. Information visualizations typically display heterogeneous multidimensional data using various views and graphical representations. High interactivity of some of the visualizations allows natural and effective data exploration and view manipulation to study the data from various perspectives. A few examples of interactive multiple view visualizations are VisTrails [4], Improvise [15], and WireVis [5], and ComVis [25]. The views in these tools are often coordinated for ready exploration of multiple data attributes and their interdependencies.

Coordinated multiple views often involve several displays such as tables, histograms, timeline, or maps that are linked by brushing, selection, and filtering. Users can explore the data by viewing it through different representations. However, the learning time required to use a new visualization, the load on users memory, the context switching time between the representations, and ongoing changes on the linked views due to interaction with another views, introduce major usability issues [16, 26, 27].

This dissertation introduces an approach to improve the usability of coordinated multiple view visualizations while keeping their sophisticated features intact. Improvise [15] is the chosen platform for conducting the research and implementing the supporting system. Improvise users are able to interactively build and browse multiple view visualizations. The rich coordination provided in Improvise and fine grained control given to users in interacting and modifying visualizations make the tool suitable for studying how to increase usability of highly interactive multiple coordinated views in general.

2.3 Interaction

Interactive visualization is an effective way to engage users in exploring and analyzing data. Interaction enables users to focus on a point of interest, manipulate views to show data differently, change content, and navigate to explore.

There are a variety of well-known taxonomies of visualization interactions (e.g., [28–32]). Some of the taxonomies that present interactions at different levels of granularity are summarized next. These taxonomies provide a better understanding of various perspectives towards visualization interactions: from focusing on system-centric to user goals. A taxonomy which is more aligned with the level of interactions considered in this dissertation is described later in detail in this chapter.

Several taxonomies focus on low-level characteristics of interactions. Shneiderman [28] categorizes interactions as seven tasks for single and multi-dimensional data visualization: overview, zoom, filter, details-on-demand, relate, history and extract. Dix and Ellis [29] take a more system-centric approach, classifying interactions as highlighting and focus, accessing extra information, and temporal fusion. Wilkinson [30] categorizes interactions into filtering, navigation, manipulating, brushing, animating, rotating, and transforming. Spence [31] looks at interactions in terms of interaction modes—continuous, stepped, passive, and composite—and data dimensionality. Amar, et al. [32] considers user goals, categorizing interactions as retrieve value, filter, compute derived value, find extremum, sort, determine range, characterize distribution, find anomalies, cluster, and correlate.

Unlike the above taxonomies, Yi, et al. [33] presents a taxonomy that takes into account both user goals and interaction techniques. They categorize visualization interactions into seven general techniques: selection, exploration, reconfiguration, encoding, abstraction/elaboration, filtering, connection, and other.

- *Selection* is marking an item of interest by ways such as highlighting [14], labeling [34], and placemarking for instance pins placed in Google Earth. Selection can sometimes be seen as a preceding action followed by an operation. For instance, filtering operation generally is proceeded after a selection action.

- *Exploration* is examining different subsets of the data over time. Panning [34], searching [35], and hyperlinking [36] are examples of exploration interactions.

- *Reconfiguration* is rearrangement to show different representations of the data. Sorting [37], clustering [38], re-plotting [39], and rotating [40] are examples of reconfiguration to look at data from different perspectives.

- *Encoding* is changing the visual representation such as its color, size, shape, or format, to reveal patterns in data that may not be easy to see otherwise.

- *Abstraction/elaboration* is changing the visual representation to provide an overview or, conversely, show more details about the data, using techniques such as zooming [14] or detail-on-demands [37], respectively.

- *Filtering* is requesting a subset of data based on a condition; for example, using sliders [14], checkboxes [14], or keyword search [35] to filter the data.

- *Connection* is showing related items. Brushing techniques in multiple coordinated view visualizations are examples of connection interaction [39]. Use of connection in visual representations is also common, for instance in network graphs.

- The *Other* category of interaction refers to all other interactions that do not clearly fit into any of the mentioned categories, such as undo/redo.

Note that the above list of interactions identified by Yi, et al. can overlap. For instance, filtering interactions can be viewed as a subcategory of exploration actions.

Considering the interactions listed above, this dissertation focuses on an important subset of interactions: selection, filtering, and connection. This combination of interactions is commonly used in research and commercial visualization tools, making them interesting and challenging to consider with an eye toward increasing the usability of visualizations that employ these techniques.

2.4 User Interaction Provenance

Visual interaction can be seen as a dialog between the user and a visual representation while the user explores data to gain insight. The rich variety of types of interactions offers an opportunity to study the exploratory behaviours and visual activities of users during their interactions with visualizations. In this dissertation, the term *user interaction provenance* is used to refer to a historical record of process, rationale, and visual activities by which an insight is acquired during analytical sessions. Also, the term *visualization state* is used to refer to the current set of parameter settings (determined by interactions) as reflected in the views of a visualization.

There is a lot of interest in studying provenance of users' visual activities, whether to share and recreate visualization states and pipelines (e.g., [11, 41]), revisit past states by providing history (e.g., [3, 42, 43]), analyze user reasoning processes and visual analysis behavior (e.g., [5,44,45]), or support analytical reasoning (e.g., [7,46]). Although these studies follow different procedures, they capture histories of visual activities as sequences of visualization interface states, user interactions, or the state transitions triggered by user interactions.

Many tools provide a history of recorded visualization states in a linear or branching fashion. For instance, Adobe Photoshop and Illustrator record document states as a linear stream that can be revisited using undo/redo operations or random access in a history panel. Another example of linear representation of interaction history is a continuous timeline that shows the duration between actions with sliders for navigation [47]. Branching models, on the other hand, use a tree structure to store history states and enable users to navigate the current branch of actions [3] or across branches of actions [12, 48, 49].

Rather than simply presenting history states for user revisitation, the focus of this dissertation is on capturing and representing the analytic connotation of interactions and state transitions, to help users comprehend the sequence of decisions made during the analysis process. In keeping with like Jankun-Kelly's conceptual framework for recording visualization states and the transition functions applied to them [11], Q2Q automatically records both, at the level of visualization parameter changes. Q2Q then reflects those intentions through designer-guided translation of changes into a written language such as English [17]. This approach promises to facilitate the visual sensemaking activities of a wider group of users having different levels of visualization expertise [50].

Approaches to capture provenance of user interactions can further be categorized loosely as either *manual* or *automatic* [6]. The manual capturing of user's visual activities requires users to manually record and annotate their rationale and acquired insight (e.g., [7–10, 46]) as they are performing their analytical tasks. Manually capturing users' high-level goals is effective in analyzing their reasoning and rationale. However, it is time-consuming and not practical considering the rapid-fire nature of interactivity in many visual analysis applications [51]. Moreover, as Gotz and Zhou [6] point out, users generally record their high-level logic and approach and not the detailed information about the steps that lead to their conclusions. As a result, making sense of their manually annotated approach can be difficult, and consequently retrieving and sharing from manual provenance is limited.

Conversely, automatic capture of interactions can produce comprehensive records of provenance (e.g., [11,12,41]). Most automatic capture systems are event-based and record low-level activities such as mouse movements and clicks. Extensive recording of low-level events, however, often produces data sets much larger than the original data itself. Moreover, due to the limited amount of data analysis context that low-level event data conveys, making sense of these interactions and extracting intention from them can be extraordinarily difficult.

Most similar to Q2Q approach is the HARVEST system [6]. HARVEST automatically captures a set of defined actions and presents them as a trail of icons. Although the system is able to abstract a set of meaningful user actions common across various visualization tools, the meanings conveyed by the icons and their action labels are limited. In contrast, in this dissertation, the meanings of interactions relative to the domain application are expressed to better capture the intended connotation of individual recorded interactions, and also sequences of them, in the context of the actually visualized data values and dimensions of the domain. High-level abstract representation of interaction provenance allows recorded interaction to be used to recreate and share visualization states, which can be useful for tutoring novice users and facilitating use of visual data analysis by experts [50].

2.5 Natural Language Generation

Natural language generation (NLG) is a well-established sub-field of computational linguistics. It has broad applications in information processing, such as description of data structures and relationships [52], narration for medical records [53], summarization of weather forecasts [54], and letter generation [55, 56]. Generally, NLG starts from some non-linguistic representation of information as input, and using knowledge about language and the application domain, automatically produces documents, reports, explanations, and other kinds of texts [57].

The approach presented in this dissertation centers on using NLG to translate the meaning of visualization interactions in a generalizable way. This NLG application is unique and challenging for two reasons. First, generation must work for an open-ended set of diverse analysis domains. It is a practical impossibility to realize both sufficient comprehensiveness and specificity in a pre-built corpus or example repository. Second, generation should produce consistent results, readily associable with visualization states. In particular, generation should respond to ongoing visualization state changes at interactive rates.

NLG techniques are maintainable, scalable, and can generate high quality text. NLG applications typically follow a *rule-based*, *statistics-based*, or *instance-based* approach. Many practical applications use rule-based approaches, e.g., [54], [53]. They require linguistic knowledge, manual development, and are tied to the domain of application. For example, a commercially used NLG system (SumTime) generates marine weather forecasts for offshore oil rigs [54]. In SumTime, the authors analyzed the existing forecast corpus to map forecasters' choices of words to the numerical forecast data, and designed a set of domain specific rules to cover existing scenarios. Though this approach produces high quality texts, its strong dependency on a particular domain makes it inapplicable to user interaction translation of visualizations of data from an open-ended set of possible domains. In [53], which documents a NLG system for electronic medical records, Harris further describes the challenges of commercially used NLG systems that necessitate a rule-based approach. Besides the general challenges such as manual construction of exhaustive domain dependent templates, required linguistic knowledge, and modularization of the system for further modifications, Harris indicates that building a system that is expected to generate accurate and reliable output cannot completely be automated. Q2Q takes this into consideration by using two stage generation that monitors possible outputs and semi-automatically generates meaningful text that is accurate and reliable.

Conversely, statistics-based approaches need less linguistic knowledge and manual development, but require large training corpora to produce high quality text. They also require long computation to perform *over-generation* and *ranking*. These approaches are largely based on Langkilde and Knight's well-known work [58] in applying statistics to NLG in the design of general-purpose machine translators. Even though various systems implemented the idea, it lacks practicality for real world application due to the low quality of output, as well as the cost of full generation and ranking output based on appropriateness.

Approaches that combine statistical and rule-based methods have produced more promising results. Belz [59] focuses on design of a reusable grammar that is applied based on probability. This approach has the advantages of variety in the generated text and better computation time compared to purely statistical methods, such as [58]. Yet, the computation time is not on the order of 100 milliseconds expected for responsive interaction with visualization user interfaces. Like other statistical approaches, combined approaches require a comprehensive corpus to base grammar rules upon and thereby allow probability approaches work. Such corpus repositories are not practical to populate when dealing with applications spanning an open-ended set of domains. Also, the accuracy and quality of outputs expected from a user interaction translation system are not guaranteed using the approach in [59], due to its probabilistic nature.

Instance-based approaches manually construct a repository (smaller than those in statistical-based approaches) of annotated examples that are often particular to the application domain, and use machine learning techniques to generate new text. Since a smaller sample size would result in lower quality output in machine learning methods, some approaches combine rule-based and instance-based methods to increase the accuracy and grammatical correctness of the text [60, 61]. Also, some work has been done to design more time-efficient generation engines compared to statistical-based methods, by incorporating search optimization techniques to find the best output suggestions [61, 62]. However, as in rule-based approaches, instance repositories are generally designed to cover a particular domain, which limits their applicability to user interaction translation.

Work on NLG for visualization use looks at summarizing non-interactive visualizations into accompanying captions [63], describing special patterns in time series data [64, 65], and capturing the decision-making process by summarizing interactive parameter changes [66]. These applications generally follow a rule-based approach. In the approach presented in this research, syntactical grammar rules are combined with a very few domain-specific sentence fragments as examples. This hybrid approach provides the meaning of new data dimensions and relationships without an existing domain specific corpus or example repository, confirms the consistency and quality of generation outputs, and translates domain-specific interactions into high quality text without a perceptible delay. Q2Q implements the first semi-automatic NLG technique that builds a structured ontology on the fly and uses it to generate text across a variety of domains without relying on an existing corpus.

Chapter 3

Query to Question Architecture and Implementation

This chapter provides a detailed description of the architecture of the user interaction translation system. It then presents the considerations that go into the application of the architecture of Q2Q to cross-filtered views. To provide context, the chapter starts with an example of how Q2Q works in conjunction with a cross-filtering visualization and what kind of questions it generates in response to user interactions.

3.1 Q2Q Example

Figure 3.1 shows an intermediate state of a cross-filtered visualization, alongside the Q2Q interface. The visualization displays data from the Retrosheet database[1] of schedules and play-by-play information for both historical and modern major league baseball games. A larger image of the visualization without the Q2Q interface is provided in Figure 3.2 for better readability. This visualization shows games, stadiums, home teams, and away teams in table views; game schedules in a calendar view; stadium locations in a geographic map; and a scatterplot matrix of game dates, game time, and two quantitative dimensions selected from dropdown lists.

[1] http://www.retrosheet.org

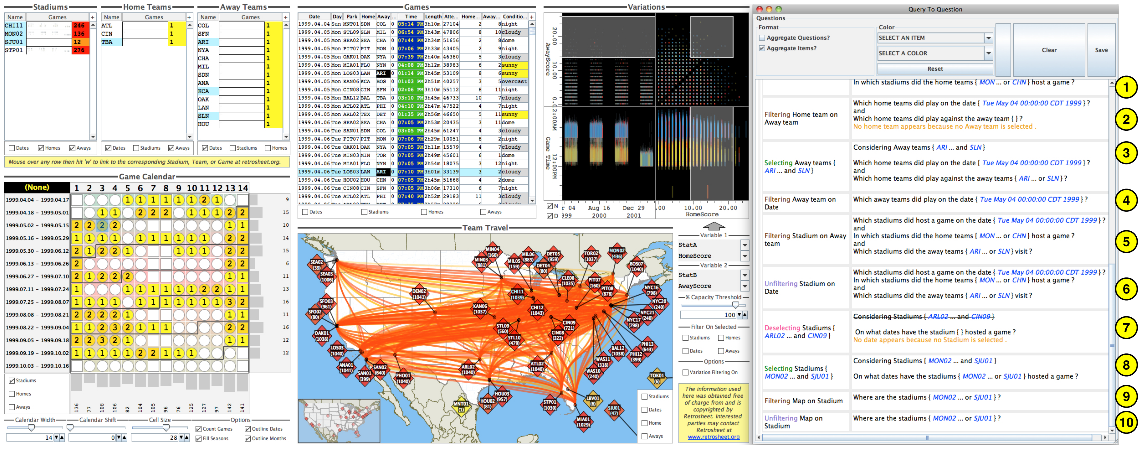


Figure 3.1: Visualization of baseball games in the Retrosheet database (left), accompanied by a Q2Q interface (right).

The Retrosheet visualization is built around cross-filtered views technique. The cross-filtering views support high-dimensional brushing between the views and allow users to filter any of the displayed views (tables, calendar, map, and scatterplot) on one or more other views. For instance, the Home Teams table in Figure 3.2 is filtered by the data values selected in the Dates and Away Teams tables. It shows only the home teams that played on May 4th, 1999 and against away teams ARI (Arizona Diamondbacks), KCA (Kansas City Athletics), and SLN (St. Louis Cardinals).

The Q2Q interface shown on the righthand side of Figure 3.1, and enlarged in Figure 3.3, appends rows of questions and automatically scrolls as the user interacts with the visualization. The left column lists interactions. For instance, toggling the Away checkbox in the Home Teams table in Figure 3.1 results in the text *Filtering Home team on Away team*, which is shown as the second row in Figure 3.3. The right column displays the corresponding translations of interactions into questions. Users can scroll to revisit interactions performed and review queries as questions. The interaction types and data attributes involved in the interactions are highlighted to increase the readability of the translations. For instance, the data value *Tue May 04* appears in blue in the second row of Figure 3.3.

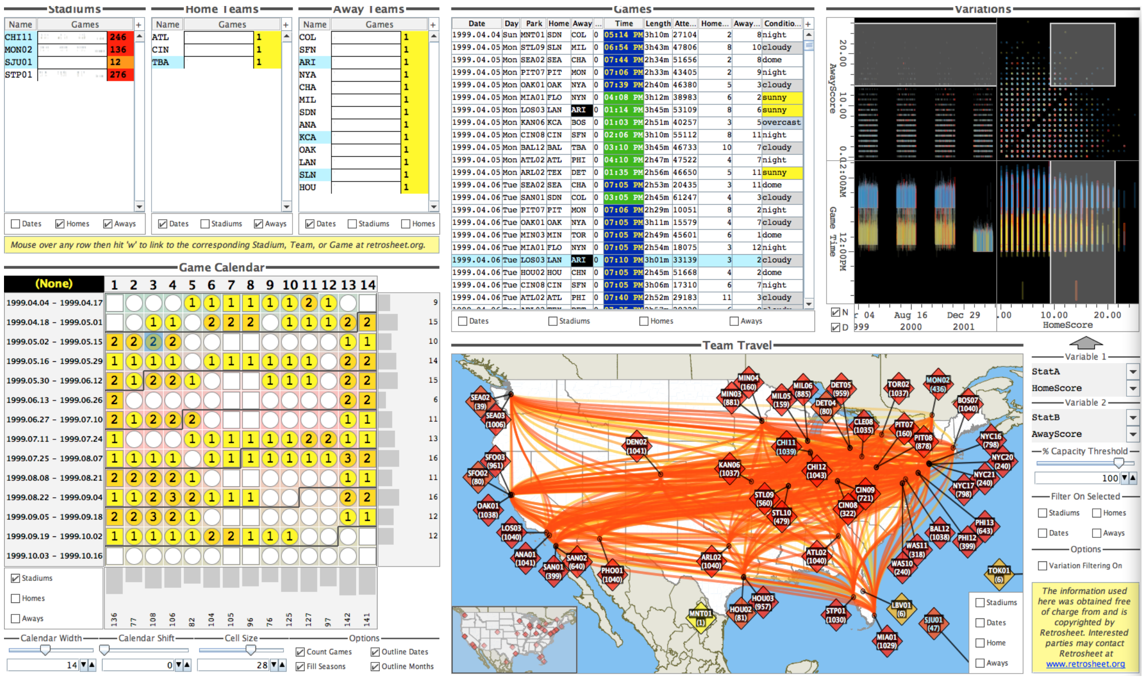


Figure 3.2: Visualization of baseball games in the Retrosheet database

Possibly confusing interactive conditions are color-coded in orange. Consider the filtering interaction shown in the second row of the Q2Q interface in Figure 3.3. The interaction filters Home Team on Away Team. However, no data value from the Away Team table is selected at this point in time, Home Team is filtered on an empty set. This results in the question *Which home teams did play against the away team* {}*?* with no item listed as the Away Team data value. To avoid confusion, Q2Q generates a warning message—*No home team appears because no away team is selected*—to inform users about what might be the issue and why the visualization responded in a certain way, in this case that this interaction results in disappearance of all the items in Home Team table. Chapter 5 provides more examples of this type of interaction.

Q2Q uses strikeout to copyedit rather than replace defunct questions. For instance, unfiltering Stadium on Date in Figure 3.3, sixth row, results in striking out the previously generated question *Which Stadiums did host a game on the date* {*Tue May 04 00:00:00 CDT 1999*}*?* and generating the questions that are still valid

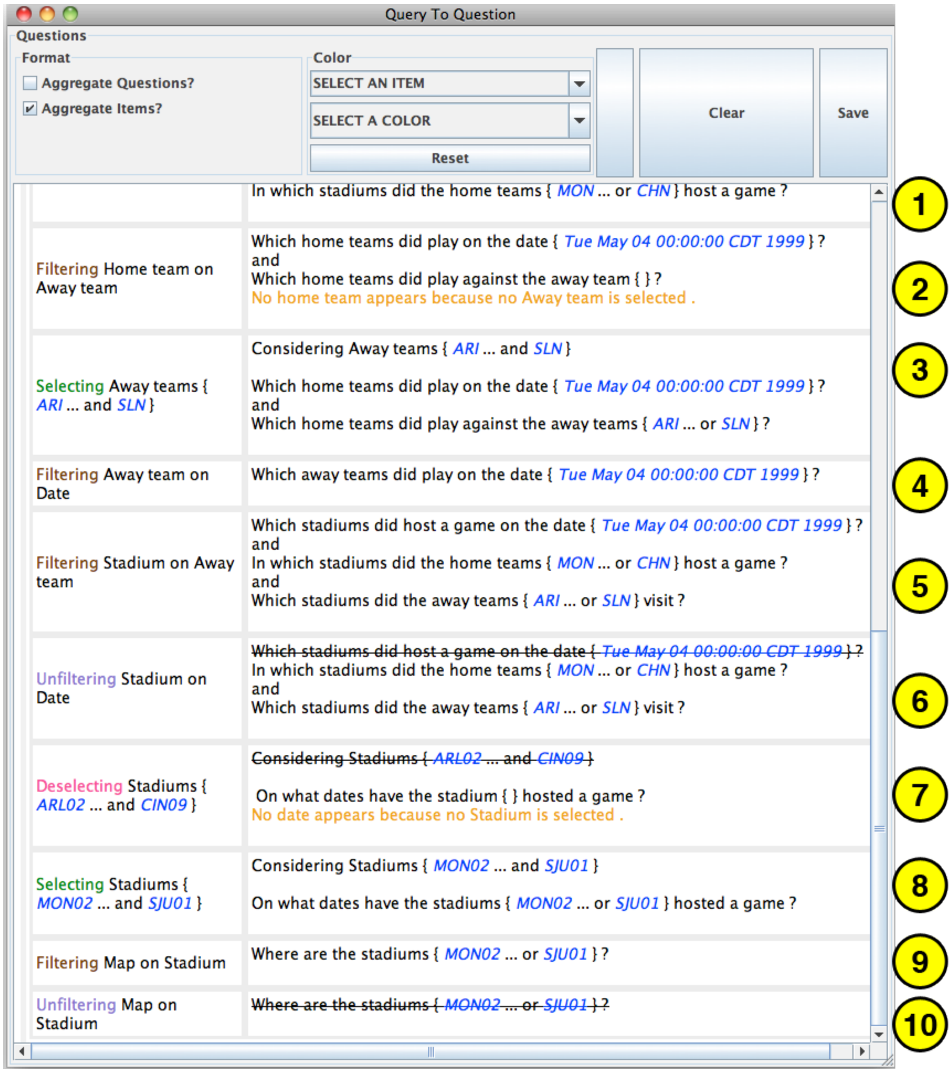


Figure 3.3: Q2Q interface that accompanies the Retrosheet visualization.

after the unfiltering interaction. Similarly, deselecting a set of stadiums from the Stadium table in Figure 3.1, seventh row, is reflected by striking out the deselected items and updating the previously generated questions (Figure 3.3, row seven, shows the questions in a larger figure).

Arbitrarily long lists of attribute values are displayed using an ellipsis as a graphical placeholder. Users can also customize the interface by changing the color of the interaction names and the data values of each type.

This example illustrates several types of question—*where, when, what/which*—that can be generated as users interact with a visualization. For example, the interaction to filter the geographical map is translated (ninth row) into *Where are the stadiums MON02, . . . , or SJU01?* A deselection interaction affects the Calendar view and results in a query translated (eighth row) as *On what dates have the stadium MON02, . . . , or SJU01 hosted a game?* An example of a what/which question (second row) is *Which home teams did play on the date Tue May 04 1999?* Translation of a selection interaction generates a phrase using the keyword *Considering* to reflect brushing of attribute values within a dimension (third row). If the selection occurs with filters on, the phrase acts as a qualifier in questions. If not, it serves as a sentence fragment to indicate activity in progress.

To accommodate variation in user preferences, needs, and analytical roles, Q2Q provides options to aggregate items within questions, and separately to aggregate the questions themselves, resulting in four different question formats. The Aggregate Items checkbox toggles item aggregation. If an interaction involves three or more data values, Q2Q aggregates them and presents them using an ellipsis: *Which stadiums did the away teams ARI, . . . , or SLN visit?* Item aggregation is on by default. The full list of attribute values can be presented on demand by unchecking the checkbox: *On what dates have the stadium MON02, PHI13, SAN02, or SJU01 hosted a game?*

The upper left corner of the interface provides a check-box to Aggregate Questions. Non-aggregated questions are concatenated with *and*, which is easy to read but somewhat clumsy: *Which stadiums did host a game on the date Tue May 04 1999? and In which stadiums did the home team MON, . . . , or CHN host a game? and Which stadiums did the away team ARI, . . . , or SLN visit?* Conversely, aggregated questions present meanings of interactions more colloquially: *Which stadiums did host a game on the date Tue May 04 1999, the home team MON, . . . , or CHN host a game in, and the away team ARI, . . . , or SLN visit?*

The formatted text can also be saved—currently in HTML format—for later use, such as to recreate past sessions and share sessions with others. Saved translations are also provided as an input to the History Organizer Tool developed in this dissertation and described in Chapter 6. The tool takes advantage of the structured format of the questions to suggest a set of reordering and grouping options and let users build a story of their own.

Users can also clear the log; this option is provided as an initial crude way to edit a record of interactions, such as to forget an unfruitful exploration path. More flexible editing and filtering of generated questions can be done after analytical sessions for the purpose of sensemaking and presentation, using the History Organizer Tool.

Throughout this dissertation, various examples show Q2Q applied to different data domains and dimensionalities to suggest its generalizability. The Retrosheet example which appears in Figure 3.1, is exemplary of cross-filtering visualizations, and is thus used to describe the architecture of the translation system in detail throughout this chapter.

3.2 Overall Architecture

The high-level architecture of the Q2Q translation system is shown in Figure 3.4. The translation process consists of two main components: *offline generation* and *online generation*. Offline generation is the process of linguistic specification for a new data domain and data set. Online generation is the dynamic generation of questions as a user interacts with a visualization. A *relationship table* populated by offline generation is given as input to online generation.

To set up Q2Q for a new visualization, a visualization designer or domain expert inputs a set of data relationship sentences into the interface of the offline generation engine. The sentences exemplify relationships between pairs of data dimensions in the visualization. The offline generation engine processes the sentences and outputs a set of possible queries in the form of questions. Once the designer confirms the

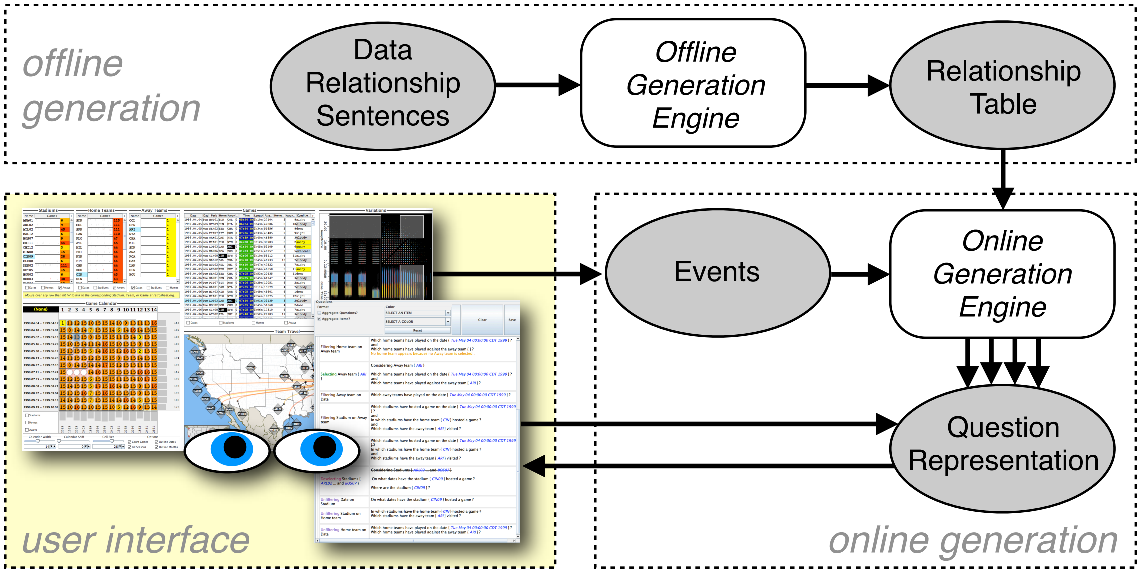


Figure 3.4: The Q2Q interaction translation architecture

correctness of the queries, the linguistic specification of the questions is stored in the relationship table. The finalized relationship table is used to generate questions as the user interacts with the visualization.

Interaction in the visualization sends events to the online generation engine, which uses the relationship table to generate question(s) that represent the user's supposed query corresponding to that event. The queries are expressed as what, who, when, and where questions. The online engine generates multiple formats of questions, at different levels of aggregation. Based on the current format setting, the *question representation* module fetches and visually encodes the corresponding question in the Q2Q interface. The user can rapidly switch between formats.

The design of the overall translation architecture is based on the well-known NLG Pipeline Method. In the Pipeline Method, text generation follows a sequential procedure that starts with *knowledge acquisition*, proceeds with *document planning* and *microplanning*, and ends with *realization*. Each stage of the generation introduces different levels of abstraction. Certain decisions need to be made in each level, which makes the generation process highly application dependent. Thus, common implementations of the Pipeline Method are constructed around a set of domain specific rules [53, 54].

To diminish the domain dependency of the generation process, a syntactical context-free grammar (CFG) is used to design domain-independent grammar rules for the generation engines. This way, the designed translation system is applicable to various visualization domains, as well as expandable to different visual interactions. This approach suggest a two stage generation process to acquire domain specific information about the data being displayed in offline generation, and use the domain-independent syntactical grammar to generate the questions in online generation. This approach substantially reduces the considerable amount of time and linguistic knowledge required to design a domain-specific translation engine for any given visualization, yet produces translations of user interaction approximate to the given data domain. The generation process and its components are described in detail in the following subsections.

3.2.1 Offline Generation

The heart of the system is the offline generation engine, which accepts data relation sentences as input and outputs possible questions for each target dimension in every directed pair. Once the designer/expert confirms the questions, the specifications of the relations are stored in a relationship table to be retrieved during online generation. Figure 3.5 shows the architecture of the offline generation system.

3.2.1.1 Input

In Figure 3.5, the box labeled as the user interface gets information about the data dimensions displayed in the visualization and outputs sets of example questions for the user (a visualization designer or domain expert) to confirm. The input to the offline generation engine is a set of descriptions of relationships between data dimensions. It is supplied by the visualization designer or domain expert as a set

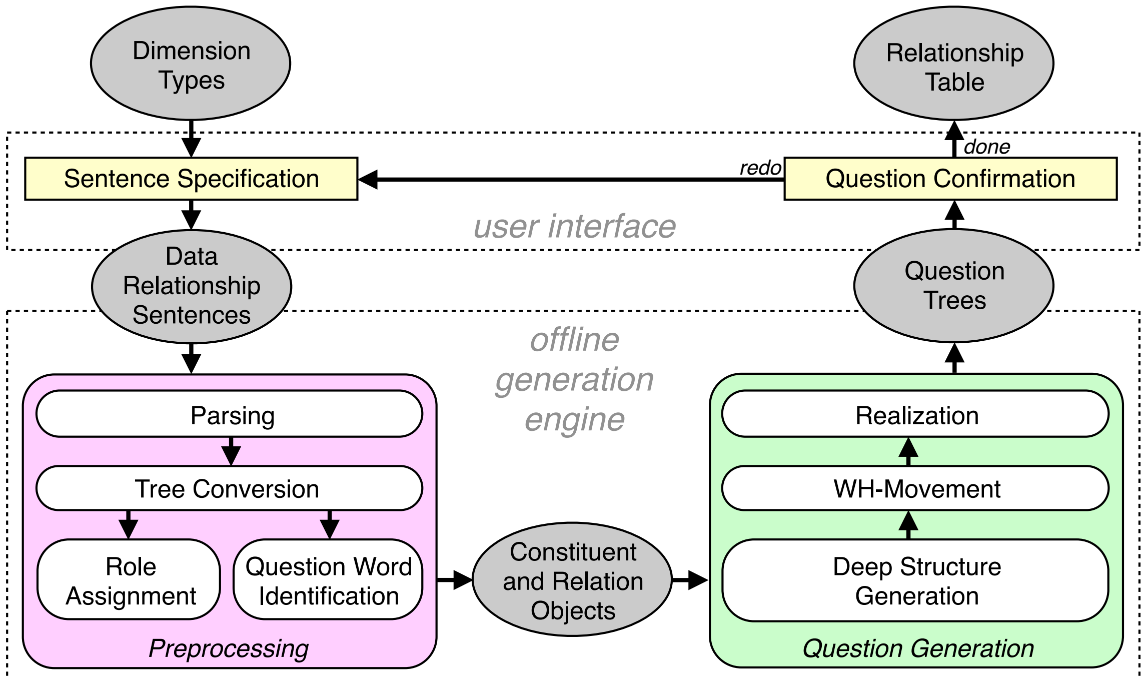


Figure 3.5: The Q2Q offline generation architecture

of simple sentences. Prior to acquiring relation sentences, the designer or domain expert has the option to characterize data dimensions. For example, they can specify the type of data as number, person, place, time, concept, or object (see Figure 3.6). If the user skips this step, the specifications are either set to default values (e.g., Number is set to Singular) or is automatically identified by the system (e.g., Type is set using the Stanford Named Entity Recognizer [67]). After this step, an input interface (shown in Figure 3.7) is presented to the user to enter relation sentences. For instance, the relationship between dimensions *Stadiums* and *Home Teams* in Figure 3.1 can be described as "home team hosts a game in stadium" and given to the system as textual input. Rather than allowing free text input, a pair of combo boxes is used to guide and constrain the choice of pairs of dimensions. This semi-structured input mechanism is much like when school kids are asked to use words in a sentence, and is designed to encourage users to compose simple sentences with phrasing based on binary relationships.

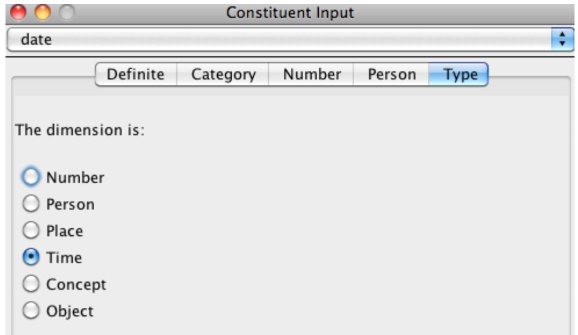


Figure 3.6: User interface to enter data dimension specifications.

3.2.1.2 Preprocessing

Preprocessing consists of four steps to transform the input data into an structured format that can be used in Question Generation. The steps are parsing the relationships, converting the parse tree to a generation tree, assigning role tags, and identifying the question words.

Parsing Given the data relationship sentences as input, the process starts by parsing the input sentences. This sub-module outputs a parse tree of the grammatical structure of each sentence. The tree has detailed information about the sentence constituents (the definition of tags in parse trees are provided in Appendix B). For instance, "stadium" and "home team" are tagged as a singular noun (NN) and "hosts" is tagged as a third-person singular present verb (VBZ), in parsing the input sentence "home team hosts a game in stadium" into the following parse tree:

```
(ROOT
  (S
    (NP (NN home) (NN team))
    (VP (VBZ hosts)
      (NP (DT a) (NN game))
      (PP (IN in)
        (NP (NN stadium))))))
```

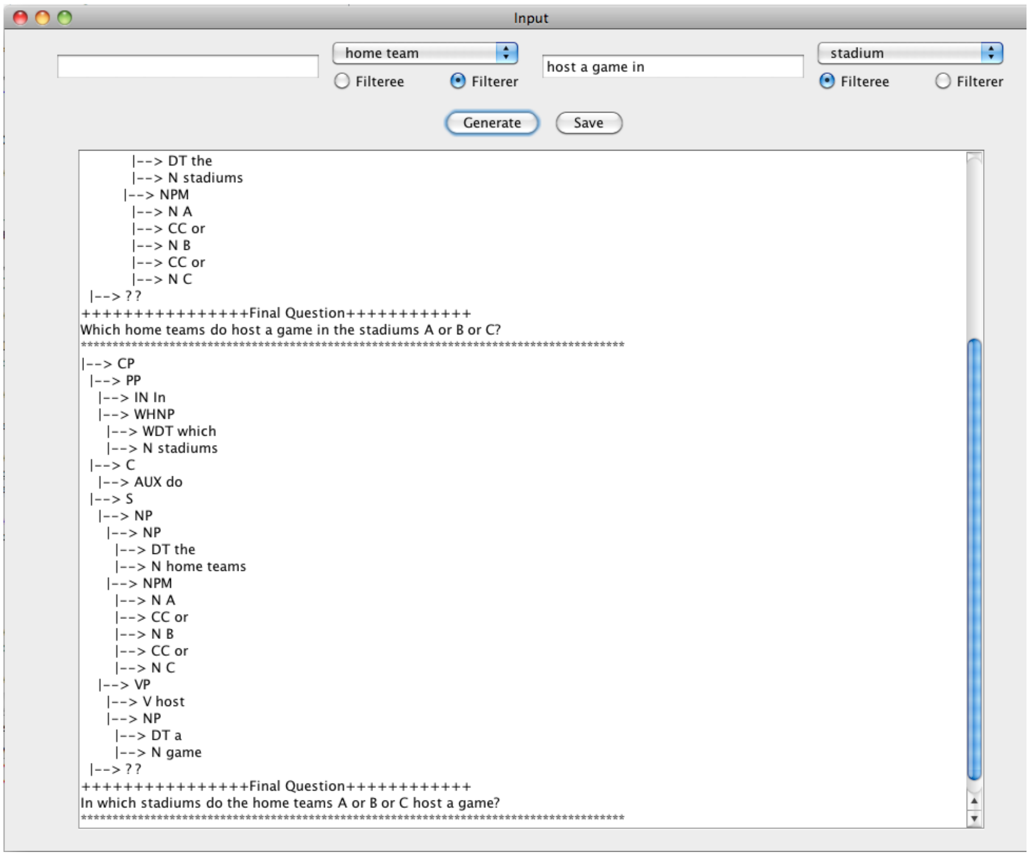


Figure 3.7: User interface to enter descriptions of relationships between dimensions.

Converting to Generation Tree The parse tree is then converted into an abstract tree called a *generation tree*. The generation tree is more general than the parse tree, and constituents are not tagged with detailed labels. For instance, “stadium”, “Home” and “team” are tagged as noun(N), and “hosts” is tagged as verb (V):

(ROOT
  (S
    (NP (N home) (N team))
    (VP (V hosts)
      (NP (DT a) (N game))
      (PP (IN in)
        (NP (N stadium))))))

This conversion is performed because the detailed information captured during parsing is subject to change during the generation, while the general grammatical structure stays the same. For instance, a singular noun “stadium” can turn into a plural noun “stadiums” during online generation if the user selects more than one stadium from the Stadiums table view in Figure 3.1. For this reason, the grammar used to generate the questions is designed to be more abstract (details about the grammar are discussed in Section 3.2.1.3.) In converting a parse tree to a generation tree, the detailed information provided by the parsing step is carried over. This information is considered a default specification of the constituent if it is not overwritten during generation.

Assigning Role Tags Given the generalized constituency structure and part of speech (POS) tags from the previous step, Role tags such as Subject, Object, and Verb are assigned. These tags determine which grammar rule should be used to generate questions. For instance, the role tags shown in parenthesis for “home team hosts a game in stadium” are: *(pre-modifier)home (subject)team (verb)hosts (Determiner)a (object)game (preposition)in (indirect object)stadium.*

Identifying Question Words The appropriate question words are identified based on the semantic types of the data dimensions involved. The semantic types considered are PERSON, TIME, LOCATION, ORGANIZATION, NUMBER, and NO ENTITY. These semantic types correspond to the question words *Who, When, Where, How many/How much,* and *What/Which*, respectively. The information about the semantic type of a data dimension is either given as an input or recognized using an entity recognizer tool [67]. In some cases, alternate question words can be linguistically appropriate; for instance, a data dimension Date can have *When* or *What* (e.g., “On what date ...”) as the question word. The system lets the designer specify a preference for each data dimension, falling back on a default question word for its semantic type when none is given.

3.2.1.3 Question Generation

The question generation stage in offline generation is the same as in online generation. Offline generation is done separately to ensure that the questions presented to users are confirmed by designers or domain experts. To generate example questions, after each sentence is preprocessed, question generation is performed on each relation sentence. Since each of the sentences involves two visualized data dimensions, at least two example questions can be produced, each with one dimension as the target (for example, the output questions in Figure 3.7). If the generated questions are not satisfactory to the designer or domain expert, they can edit the sentence relation so the system generates another set of questions. As the designer or domain expert confirms questions, the specifications of relationships between data dimensions are stored for later use as input to online question generation.

Q2Q applies a set of syntactical context-free grammar rules to realize sentences that reflect query interactions. To develop these rules, nearly one hundred question structures were studied from four application domains, and classified by hand based on the part of speech tags of the targets of the questions. The grammar is represented by $G = (W, N, S, R)$, in which W is a set of terminals, N is a set of nonterminals, $S \in N$ is the start symbol, and R is a set of production rules. Each rule has the form $n \rightarrow \alpha$, $n \in N$, $\alpha \in (W \cup N)^*$ in which W and N are disjoint. To capture the context sensitivity of English grammar, the nonterminals are n–ary relations $f(o_1, ..., o_n)$ in which $n \geq 0$ and $o_1, ..., o_n$ are either constituent objects or relation objects that define relationships between constituents. For example, *stadium* is considered as the head of a constituent with all its specifications such as singular, definite, and so forth constructing the constituent. Accordingly, a relation object for *stadium* and *home team* would have information about the verb *host* that connects them. Constituent and relation objects are both outputs of the preprocessing step.

This design of text generation has the advantages of reusability and expansion. The previously added syntactic rules can be reused and new rules can be added to generate new types of questions [59]. This will be particularly important for future extension of Q2Q to include types of interactions other than filtering and selection. Higher-level syntactical rules can be added based on the expected structure of the translation, and lower-level rules can be reused in the definition of higher-level rules.

Grammar rules in Q2Q are designed at a general syntactic level, rather than at a fully specified level. A fully specified level would construct generation rules based on lower level syntactical specifications (several of this are discussed in examples below). Two reasons for this choice are grammar size and missing information. Another reason, mentioned in Section 3.2.1.1, is that specifications in the parsing stage are subject to change during online generation.

Full specification of grammar rules would increase the size of the grammar drastically. A general rule like S $\rightarrow$ NP VP can expand to numerous specific grammar rules, such as S $\rightarrow$ NN VBD, S $\rightarrow$ NNS VBG, S $\rightarrow$ NNP VBP, or S $\rightarrow$ NNPS VB[2]. To keep the grammar small, general grammar rules are combined with a set of n-ary production rules, such as $\mathrm{S}(x, y, z, r) \rightarrow \mathrm{NP}(x, y, r)\mathrm{VP}(z, r)$, in which x, y, z are constituent objects and r is a relation object that specifies relations among the constituents. Full specifications involving NP and VP are expanded using the constituent and relation object variables and applied in the later stages when non-terminals are substituted by terminals (lexicalization).

Occasionally, detailed information about constituents and relations can only be retrieved at later stages of generation; for instance, to determine whether "Stadium" needs to appear as plural or singular. Moreover, sometimes certain specifications cannot be acquired at all due to missing information, yet questions can be constructed without concern for all details. For example, if a given input sentence is not complete and does not have a verb, an abstract grammar based on a generation

[2]S: sentence, NP: noun phrase, VP: verb phrase, NN: singular noun, NNS: plural noun, NNP: singular proper noun, NNPS: plural proper noun, VBD: past tense verb, VBG: present participle verb, VBP: singular present non-3rd verb, VB: base form verb. More tags definitions are provided in Appendix B.

tree can still be used to generate questions. A partial question is generated in which the nonterminal VERB is not substituted by a terminal and is left to be specified by the designer. By using a placeholder for missing information, designers or domain experts can better spot the issue and possibly edit the input descriptive sentences accordingly.

Below is an example of a set of grammar rules applied to construct a question for the input "home team hosts games in stadium." The NP $\rightarrow$ DET N rule is an example of how similar rules can be reused with different variables. H refers to *Home Team*, S refers to *Stadium*, ARI refers to *Arizona Diamondbacks* stadium. C and R are constituents and relation objects, respectively.

$$\mathrm{S}(c_H, c_S, c_{ARI}, r_{H,S}) \rightarrow \mathrm{NP}(c_H)\mathrm{VP}(c_S, c_{ARI}, r_{H,S})$$
$$\mathrm{VP}(c_S, c_{ARI}, r_{H,S}) \rightarrow \mathrm{AUXV}(r_{H,S})\mathrm{NP}(r_{H,S}.Object)\mathrm{PP}(c_S, c_{ARI}, r_{H,S})$$
$$\mathrm{AUXV}(r_{H,S}) \rightarrow \mathrm{AUX}(r_{H,S})\mathrm{V}(r_{H,S})$$
$$\mathrm{PP}(c_S, c_{ARI}, r_{H,S}) \rightarrow \mathrm{P}(r_{H,S})\mathrm{NP}(c_S, c_{ARI})$$
$$\mathrm{NP}(r_{H,S}.Object) \rightarrow \mathrm{DET}(r_{H,S}.Object)\mathrm{N}(r_{H,S}.Object)$$
$$\mathrm{NP}(c_{ARI}, c_S) \rightarrow \mathrm{DET}(c_S)\mathrm{N}(c_S)\mathrm{NPM}(c_{ARI})$$
$$\mathrm{NP}(c_H) \rightarrow \mathrm{DET}(c_H)\mathrm{N}(c_H)$$
$$\mathrm{NP}(c_S) \rightarrow \mathrm{DET}(c_S)\mathrm{N}(c_S)$$
$$\mathrm{NPM}(c_{ARI}) \rightarrow \mathrm{N}(c_{ARI})$$
$$\mathrm{DET}(x) \rightarrow a$$
$$\mathrm{DET}(x) \rightarrow the$$
$$\mathrm{DET}(x) \rightarrow \mathrm{WDT}(x)$$

Using the described grammar, an abstract representation of the text is constructed in the next step.

Deep structure generation builds an abstract representation of a sentence that contains its core semantic relations. This representation will be mapped on to the surface structure (outward form) of a sentence via transformations. The preprocessing stage outputs constituency structures with the POS and role tags in the form of constituent and relation objects. Based on the role tags of the data dimensions, appropriate grammar rules are applied to generate questions. The questions correspond to the given role tags, such as subject questions and object questions. The target of the question is tagged as the WHPhrase. One of the data dimensions will be the target of the question. The grammar above outputs the following deep structure:

$\mathrm{WDT}(c_{HomeTeam})\mathrm{N}(c_{HomeTeam})\mathrm{AUX}(r_{HomeTeam,Stadium})\mathrm{V}(r_{HomeTeam,Stadium})$
$\mathrm{DET}(r_{HomeTeam,Stadium}.Object)\mathrm{N}(r_{HomeTeam,Stadium}.Object)$
$\mathrm{P}(r_{HomeTeam,Stadium})\mathrm{DET}(c_{Stadium})\mathrm{N}(c_{Stadium})\mathrm{NPM}(c_{ARI})$

This structure will be later replaced by specific words to construct the first question shown in Figure 3.7.

Next a transformation is needed to map the *deep structure* into a *surface structure*. This transformation is called **WH-Movement** and consists of three steps:

1. Auxiliary verb insertion. If the sentence has no auxiliary verb, insert an appropriate form of ‘to do’ (‘do’, ‘does’, ‘did’).

2. Verb form adaptation. If the main verb of the sentence is not the auxiliary verb, reduce it to its base form.

3. WHPhrase movement. Move the auxiliary verb to the start of the sentence (*Inversion*), then place the WHPhrase in front of it. Sometimes, more than one WH-question can be formed from the same deep structure. For instance, when a prepositional phrase is the target of the question, WH-movement can be performed either on the prepositional phrase that contains the WHPhrase or on the WHPhrase itself.

Finally, **realization** converts the output of WH-Movement to actual text. In this stage, the system takes care of grammar, constituent numbers, verb tense, and punctuation. It takes into account the context of the interaction and the data dimensions involved. For instance, it makes sure that cardinalities (plural or singular) of constituents are aligned with the visual interaction, that the verb tense conveys the correct contextual relationship (e.g., using past tense in the home team and stadium question), that the first word in the question is capitalized, and that appropriate punctuation is used (such as the question mark at the end of question sentences).

3.2.1.4 Populating the Relationship Table

Once a question is output from the realization and confirmed by the designer, the lexical and structural specification of the relationships between data dimensions leading to the question is stored in a database for retrieval during online generation. Relationship specifications are often applicable across domains. The relation objects stored for pairs of dimensions can incrementally populate a database for use with future visualizations with involving known data dimensions and similar contextual relationships. Over time, the database can grow into a repository for rapid reuse in offline design or direct use in online generation [68].

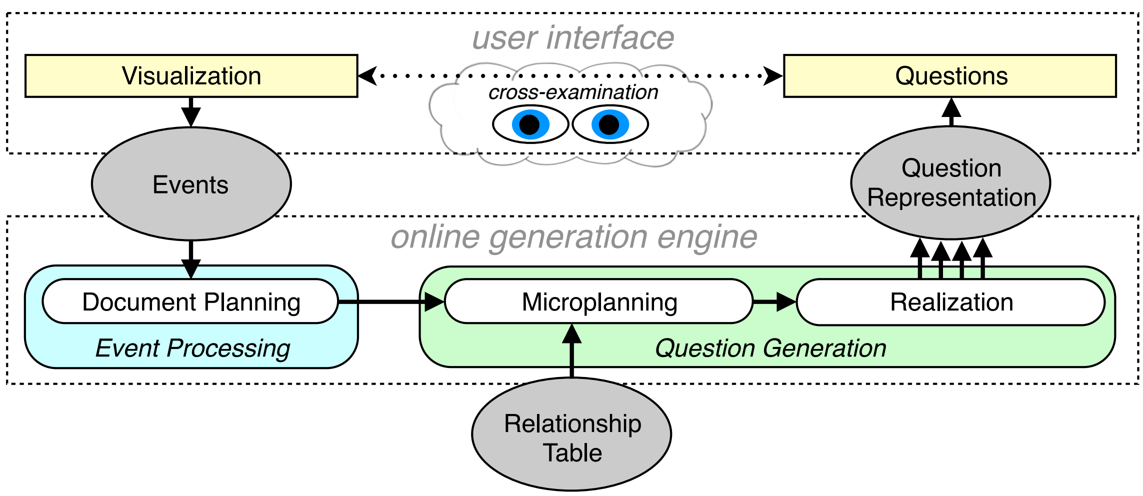


Figure 3.8: The Q2Q online generation architecture.

3.2.2 Online Generation

As users interact with a visualization, their interactions are translated into questions expressing their queries. The interaction information and questions are visually encoded and presented in the Q2Q online interface. The online generation architecture is illustrated in Figure 3.8.

The online generation engine takes input in the form of interaction events captured at the level of visualization parameter changes. This is similar to the level of *action exploration* [6], in which the user accesses and explores data to search for insights. Supplied with an interaction event and the relationship table populated in offline generation, the online generation engine looks up the linguistic specification of the relationship between data dimensions involved in the interaction.

Online question generation follows the three stages of the Pipeline Method [57]: document planning, microplanning, and realization. Document planning determines the content of the text and the information that needs to be communicated. Microplanning determines how the selected information can be communicated linguistically. Realization is responsible for generating grammatically correct text.

In document planning, information is determined about an interaction event to be included in the text. For instance, if all but one item are selected in a table view, the text can describe the selected items, only the non-selected item, or do both. The ordering of items in text is also determined. Given the interaction event, the document planner detects all affected dimensions and tells the microplanner to generate questions for all queries triggered by the recent interaction. For example, in the Q2Q interface in Figure 3.1, the fourth full row reflects an interaction to filter *Stadium* on *Away Team*. The corresponding changes in visualization state depend on three dimensions: *Date*, *Home Team*, and now *Away Team*. Therefore, three questions are presented for this interaction.

Microplanning uses the same question generation module as offline generation (the green areas labeled as "Question Generation" at bottom right in Figures 3.5

and 3.8), except that additional input data is provided by document planning. Supplied with details of the interaction to be communicated in the text, the generation system looks up information about constituent and relation objects from the relationship table, chooses appropriate grammar rules, and generates the abstract form (deep structure) of questions. Finally, lexicalization replaces non-terminals with terminals.

Realization accepts an abstract representation of the question and turns it into grammatically correct text. Offline and online generation use the same realization module.

The grammar rules might output multiple deep structures of questions. Based on the current settings in the Q2Q interface, *Question Representation* chooses from the list of questions output by the online generation engine, applies visual encoding to the text, and displays it in the Q2Q interface. Users can request a different choice and format of questions at any time during their interaction with the visualization.

3.3 Implementation

Q2Q is implemented, including offline and online generation, in Java. The Improvise visualization builder and browser environment [15] was extended to dynamically output visualization parameter changes to the Q2Q online generation engine and to input result sentences for visualization into a window alongside the primary visualization.

The implementation of Q2Q focuses on translation of user interactions in cross-filtered views [18]. By covering an important subset of information visualization interaction types—brushing and filtering—Q2Q builds a foundation and works as a stepping stone for applying natural language generation to other types of interactions in the future.

The key parameters of selection and filtering interactions are the filterer attributes, the filteree attributes, and the selected set of data values of the filterer attributes. Interaction translation in the Q2Q implementation focuses on these parameters.

3.3.1 Input

Cross-filtering interactions involve at least two data attributes: filterer and filteree. Filterer is the dimension that filters other dimensions based on its one or more selected data values. Filteree is the dimension being filtered by the filterer.

The input interface for offline generation is designed in a way to effectively get information about the pairwise relationships between the dimensions. These dimensions can play either or both roles—*filterees* and *filterers*—during live translation in online generation. The drop-down comboboxes shown in Figure 3.9 contain all the dimensions that are part of the cross-filtering matrix in the corresponding visualization. The radio buttons under the comboboxes are designed to further guide users to enter descriptive text fragments that capture the meaning of the data attributes, taking into account each dimension as the source of action or as a filterer. Even though a change in roles often does not require different input descriptive sentences, as is the case in the example in Figure 3.9, it occasionally might suggest alternative input sentences when the relationship is expressed differently based on the subject of the sentence. For instance, in a visualization of email metadata, with senders and receivers as two data dimensions, these two dimensions can be related using the verbs "send" and "receive": senders *send* emails to receivers and receivers *receive* emails from senders. Two input descriptive sentences cover both directions of this asymmetric pairwise relationship.

The Generate button in Figure 3.9 is particularly helpful to allow users experiment with the descriptive text fragments and see the questions that result. Users can then decide if they want to provide different input before they save the resulting specification.

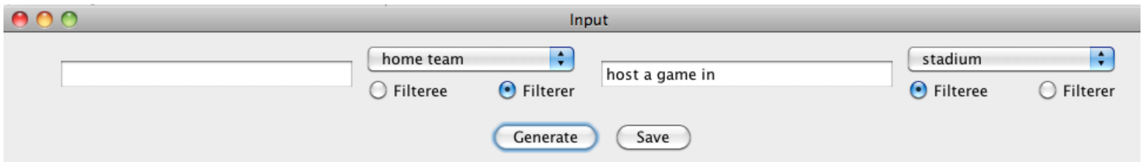


Figure 3.9: The Q2Q offline input interface for cross-filtering interactions.

3.3.2 Preprocessing

As described in Section 3.2.1.2, preprocessing consists of parsing input, converting a parse tree to a generation tree, assigning role tags, and identifying questions words.

To build an ontology based on the given input, descriptive sentences are parsed to get their part of speech tags. These tags are used for assigning role tags. The role tags are used to identify the grammar rules for generating the questions. The parsing sub-module of the user interaction translation architecture is implemented in Q2Q using the Stanford Parser [69]. The Stanford Parser is a probabilistic natural language parser which uses knowledge of language based on a large set of hand-parsed sentences to produce the most likely structure of new sentences. Q2Q uses a Java implementation of Stanford Parser to extract the part of speech tags of the input sentences. Other natural language processing tools include the Berkeley Parser [70] and the Charniak Parser [71]. The compatibility, precision, and adaptability of Stanford Parser make it particularly suitable for the purpose of Q2Q.

Given the part of speech tags, a tree abstraction module converts the parse tree, which has detailed information about sentence components and structure, to a more general generation tree (see Section 3.2.1.2).

A role tag assignment module is implemented based on the expected structure of basic active and passive sentences, e.g., "subject verb object". The current Q2Q implementation also covers role tags for questions that involve indirect objects or noun phrase modifier. For example, in the text fragments "Co-PIs receive award from program managers" and "vessels carrying passengers are interdicted by ships", "program managers" and "carrying passengers" are the indirect object and noun phrase modifier, respectively.

Question word identification is done once role tags are determined and the subject of the potential question is set. Given the relationship between filterer and filteree in cross-filtering interactions, the subject of generated questions is the filteree dimension. For instance, consider a visualization of movies with two data dimensions, *genre* and *movie*. A possible translation resulting from filtering *genre* on *Movie* with selected value *Avatar* is, "What is the genre of the movie Avatar?" In this question, *genre* is both the filteree dimension and the subject of the question. Given a pairwise relationship sentence, Q2Q generates at least two questions, each taking one of the specified dimensions as the filteree and to be the subject of the question.

The type of the question-word corresponds to the entity type of the subject of the question (filteree). If the entity type of the filteree is not specified by the visualization designer or domain expert as a part of offline input, the Named Entity Recognizer [67] is used to determine if the data dimension is PERSON, TIME, LOCATION, ORGANIZATION, NUMBER, or NO ENTITY. Q2Q generates *Who, When, Where, What,* and *Which* questions. It does not currently generate *How* or *Why* questions. *How* and *Why* questions are mainly related to higher-level analysis and pattern discovery, which may be impossible to capture through action level interactions alone; e.g., "How does the date affect the score of a baseball game?"

3.3.3 Question Generation

The implementation of Q2Q defines and applies a set of syntactical context free grammar (CFG) rules. Figure 3.10 shows a subset of rules defined to generate questions for cross-filtering interactions. In the n-arry relations of the (non)terminals, the constituents object given to the (non)terminals are generally the filteree data dimension, filterer data dimension, and the selected items within the filterer data dimension (e.g, x, y, z variables in Figure 3.10). Based on the role tags determined in preprocessing stage, one of the main top level sentence rules (e.g. $S(x, y, z, r) \rightarrow NP(x, y)VP(z, r)$ in Figure 3.10) is selected. Then, following the grammar rules, a set of abstract questions are output. WH-movement described in Section 3.2.1.3 turns the abstract representation of the questions (deep structure) into a surface representation. Q2Q uses WordNet [72] to find base forms of the verbs and SimpleNLG [73] to later adjust numbers for noun phrases and verbs, verb tenses, and punctuation, to realize the questions.

Implementing the question generation engine of Q2Q based on syntactical context free grammar makes it practical to extend the current grammar to cover other interaction techniques using additional sentence structures. This can be done by adding sentence-level and phrase-level rules to the grammar and reusing existing lower-level rules.

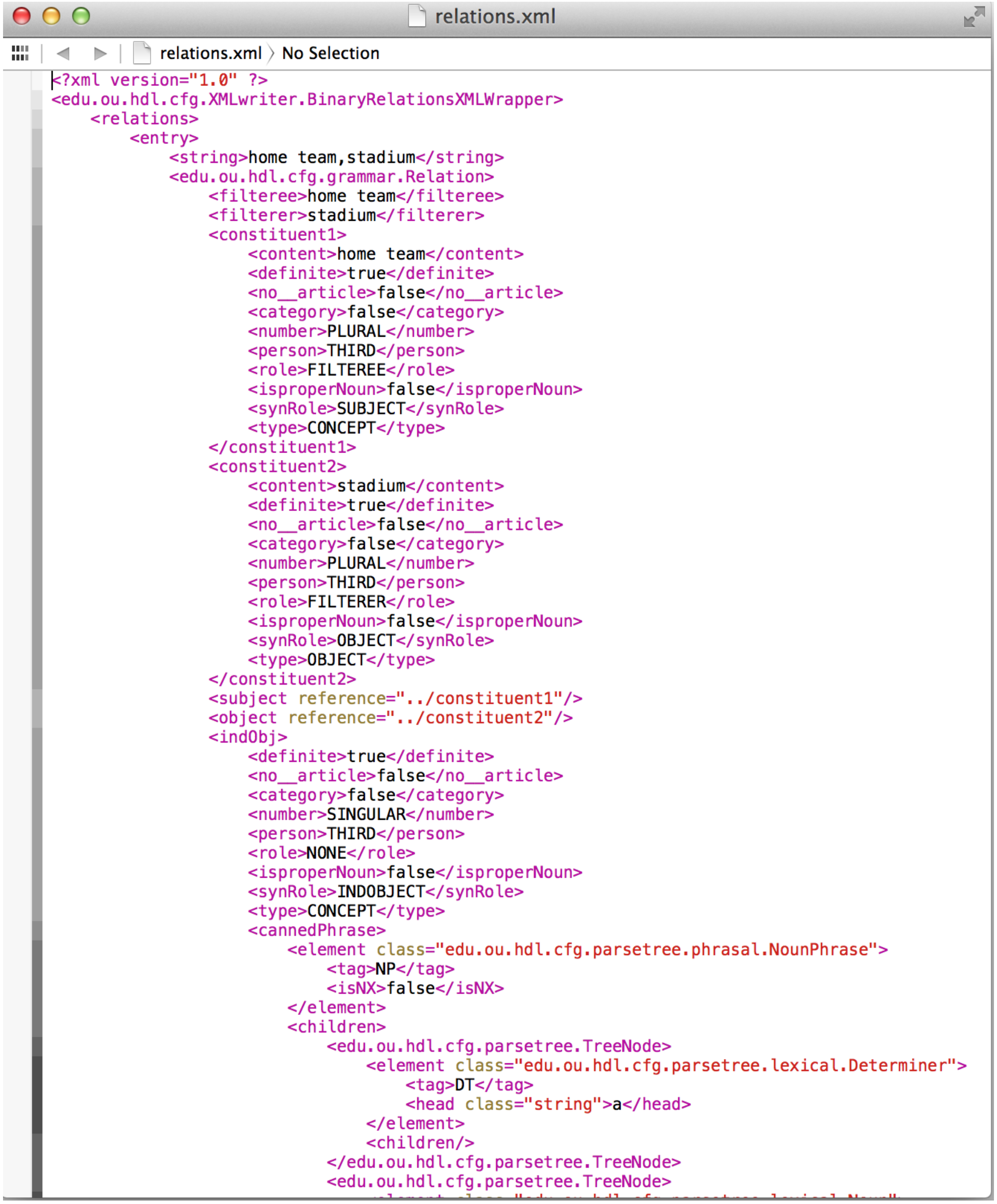
3.3.4 Populating the Relation Database

Once the questions are realized and confirmed by the users, the specification of the questions are stored in a XML file to be used in online generation. The relationships specified between pairs of dimensions are parsed, tagged, characterized, and stored in a relation object. The relation object contains information about the constituents of the relation and syntactical role tags such as verb, indirect object, or prepositions. The relation object also includes information about the contextual role (filteree or filterer) of the input relationship. This is particularly important when constituents with different role tags require different relation objects (i.e., different descriptive sentences as input). Once the example questions are confirmed, the constructed relation object—which is also used to form the example questions—is converted to a XML format to be readily used by online generation. This conversion is done for all pairwise relationships given as input and stored in an aggregated XML file.

```
*
* x, y, z are Constituent, r is a relation, s is string, w is conjunction and xs is an array of Constituents
* CP(x,y,z,r)  -> S(x,y,z,r)
*
* S(x,y,z,r)   -> NP(x,y) VP(z,r) //rule 1
* S(x,y,z,r)   -> NP(x) VP(y, z, r) //rule 2
* S(x,y,z,r)   -> NP(r) VP(x,y,z,r) // rule 3
* S(x,y,z,r)   -> NP(x,y,z,r) VP(r) //rule 4
* S(y[], r)    -> VP(y[], r)//selection, e.g. "considering selected_item_names", or "Selecting a, b, c"
* S(x, r)      -> VP(x, r) //selection all, e.g. "considering any of the table_name" , n can be any, all, ...
*
* NP(x,y)      -> Det(x) N(x) NPM(y) // NPM stands for noun phrase modifier
* NP(x,y)      -> Det(y) N(y) NPM(x) // this rule is optional
* NP(x,y)      -> Det(x) N(x) p("of") N("type") NPM(y)
*
* NP(x,y,r)    -> NP(r.modifier) NP(x,y)
*
* NP(x,y,z,r)  -> det(x) N(x) PP(y,z, r.modPP, r)
*
* NP(x)        -> Det(x) N(x)
* NP(x)        -> Det(x) N(x) PP(x) // if x.category is true
* NP(x)        -> N(x) // in the especial case of "who" which means "which people"
* NP(xs,w)     -> NP(xs1) Conj(w) NP(xs(excluding xs1), w) // xs is an arraylist of constituents
* Np(xs,w)     -> NP(xs1) Conj("...") NP(xsn)
* NP(r)        -> Det(r.indObj) N(r.indObj) //rule 3-8
*
* NPM(x)       -> N(x) // NM refers to an NP without a determiner since it is a modifier
* NPM(xs,w)    -> NPM(x1) Conj(w) NP(xs, excluding x1) // xs is an array of constituents
*
* N(s)         -> s // s is a string. e.g. if x.category is true s can be "types"
* N(x)         -> x.content
*
* P(s)         -> s    // s is a string. e.g. if x.category is true s can be "of"
* p(r)         -> r.preposition
*
*
* VP(r)        -> AUXV(r) N(r.obj)
*
** VP(z,r)     -> AUXV(r) NP(z,r) PP(z,r)
* VP(z,r)      -> AUXV(r) PP(z,r) //rule 1
* VP(z,r)      -> AUXV(r) NP(z,r) //rule 1
* VP(z,r)      -> AUXV(r) NP(r.indObj) PP(z,r) // rule 1-7
* VP(x,r)      -> AUXV(r) N(r.subject) PP(x, r.p) // considering any of "table_name"
** VP(z,r)     -> AUXV(r)
*
* VP(y,z,r)    -> AUXV(r) NP(y,z) //rule 2
* VP(y,z,r)    -> AUXV(r) PP(y,z,r) //rule2
* VP(y,z,r)    -> AUXV(r) NP(r.indObj) PP(y,z,r) //rule 2-3
* VP(y,z,r)    -> AUXV(r) PP(r) PP(y,z,r) //rule 2-4
* VP(y,z,r)    -> AUXV(r) NP(r.indObj) NP(y,z) //rule 2-10
*
* VP(x,y,z,r)  -> AUXV(r) NP(y,z) NP(x) //rule 3-8
* VP(x,y,z,r)  -> AUXV(r) NP(y,z) PP(x,r) //rule 3-8
* VP(x,y,z,r)  -> AUXV(r) NP(x) PP(y,z,r.indobj) //3-9
*
* AUXV         -> AUX(r) V(r)
* AUXV         -> V(r) // in the case that the auxiliary verb itself is the main verb. e.g. this is a dog.
*
* PP(a,b)      -> PP(a) PP(b)
*
* PP(a, s)     -> P(s) NP(a) // s is a string
*
* PP(z,r)      -> P(r) NP(z) //rule 1
*
```

Figure 3.10: A subset of syntactical context free grammar rules defined to translate user interactions into questions in Q2Q.

Figure 3.11 shows a portion of the XML format of the relation between Home Team and Stadium displayed in the visualization in Figure 3.1.



relations.xml

relations.xml 〉 No Selection

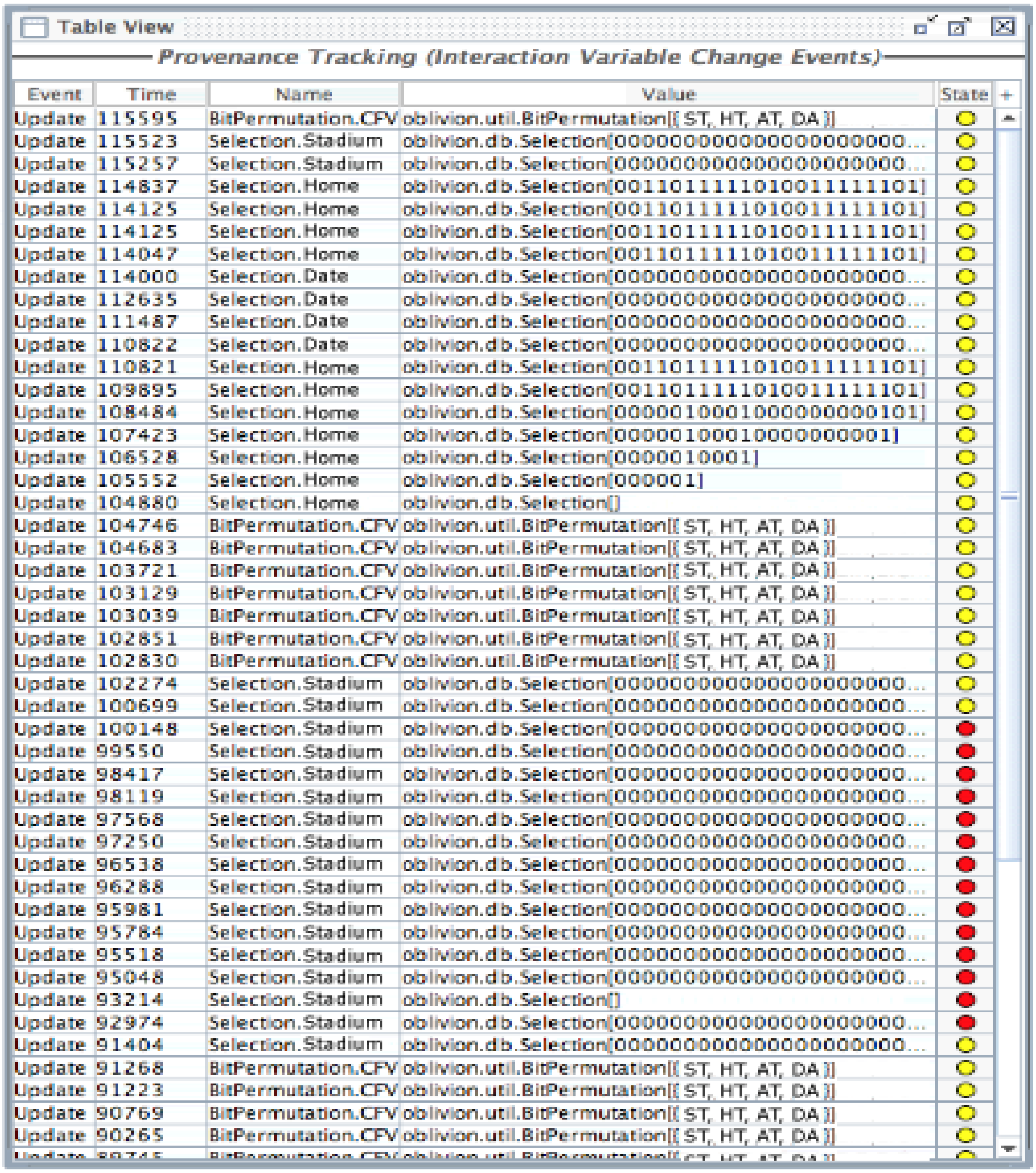

```xml
<?xml version="1.0" ?>
<edu.ou.hdl.cfg.XMLwriter.BinaryRelationsXMLWrapper>
    <relations>
        <entry>
            <string>home team,stadium</string>
            <edu.ou.hdl.cfg.grammar.Relation>
                <filteree>home team</filteree>
                <filterer>stadium</filterer>
                <constituent1>
                    <content>home team</content>
                    <definite>true</definite>
                    <no__article>false</no__article>
                    <category>false</category>
                    <number>PLURAL</number>
                    <person>THIRD</person>
                    <role>FILTEREE</role>
                    <isproperNoun>false</isproperNoun>
                    <synRole>SUBJECT</synRole>
                    <type>CONCEPT</type>
                </constituent1>
                <constituent2>
                    <content>stadium</content>
                    <definite>true</definite>
                    <no__article>false</no__article>
                    <category>false</category>
                    <number>PLURAL</number>
                    <person>THIRD</person>
                    <role>FILTERER</role>
                    <isproperNoun>false</isproperNoun>
                    <synRole>OBJECT</synRole>
                    <type>OBJECT</type>
                </constituent2>
                <subject reference="../constituent1"/>
                <object reference="../constituent2"/>
                <indObj>
                    <definite>true</definite>
                    <no__article>false</no__article>
                    <category>false</category>
                    <number>SINGULAR</number>
                    <person>THIRD</person>
                    <role>NONE</role>
                    <isproperNoun>false</isproperNoun>
                    <synRole>INDOBJECT</synRole>
                    <type>CONCEPT</type>
                    <cannedPhrase>
                        <element class="edu.ou.hdl.cfg.parsetree.phrasal.NounPhrase">
                            <tag>NP</tag>
                            <isNX>false</isNX>
                        </element>
                        <children>
                            <edu.ou.hdl.cfg.parsetree.TreeNode>
                                <element class="edu.ou.hdl.cfg.parsetree.lexical.Determiner">
                                    <tag>DT</tag>
                                    <head class="string">a</head>
                                </element>
                                <children/>
                            </edu.ou.hdl.cfg.parsetree.TreeNode>
                            <edu.ou.hdl.cfg.parsetree.TreeNode>
```


Figure 3.11: Part of the XML file showing the relation between Home Team and Stadium in the Retrosheet Visualization.

3.3.5 Online generation

The online generation engine takes interaction events as inputs. The interactions that are considered in the current implementation of Q2Q are brushing and filtering. Brushing can be done through mouse clicks, lassoing, and keyboard shortcuts. Filtering is done through toggling checkboxes. An example of a set of interaction events, captured in Improvise is shown in Figure 3.12. The table lists selection (e.g. "Selection.Stadium") and filtering settings (represented collectively as "BitPermutation" objects) user interactions in the Retrosheet visualization. Each row shows the event type, the relative time it occurred, and the affected parameter value. Given this record of low-level interactions, online generation uses the Pipeline Method to transform them into a sequence of questions. The implementation of online generation consists of three modular pipeline stages: document planning, microplanning, and realization. Changes to one stage have minimal effect on other stages, making it possible to extend the system to cover other interaction techniques.

In the application of Q2Q to cross-filtered views, the document planning stage primarily identifies the filterees, filterers, and selected items involved in a given interaction. Later, it decides how much information about filteree, filterer, and selected items needs to be communicated. For instance, it determines if a generated question should explicitly refer to a filterer data dimension, or, on the other hand, it should implicitly include the filterer by only listing the data values selected within the dimension, for example,"In which movies the *actor* Naomi Watts played a role?" versus "In which movie Naomi Watts played a role?" The level of abstraction of selected items is also specified in the document planning stage. For instance, the full names rather than the abbreviated names of stadiums can be used in the questions in the Retrosheet visualization, e.g. Saint John's University rather than SJU. Other decisions, such as inclusion of certain selected items in questions, inclusion of particular sets of questions in the translation, and the structure and order of those questions, are made in the document planning stage.



Event	Time	Name	Value	State
Update	115595	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	115523	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	115257	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	114837	Selection.Home	oblivion.db.Selection[0011011111010011111101]	
Update	114125	Selection.Home	oblivion.db.Selection[0011011111010011111101]	
Update	114125	Selection.Home	oblivion.db.Selection[0011011111010011111101]	
Update	114047	Selection.Home	oblivion.db.Selection[0011011111010011111101]	
Update	114000	Selection.Date	oblivion.db.Selection[000000000000000000000...	
Update	112635	Selection.Date	oblivion.db.Selection[000000000000000000000...	
Update	111487	Selection.Date	oblivion.db.Selection[000000000000000000000...	
Update	110822	Selection.Date	oblivion.db.Selection[000000000000000000000...	
Update	110821	Selection.Home	oblivion.db.Selection[0011011111010011111101]	
Update	109895	Selection.Home	oblivion.db.Selection[0011011111010011111101]	
Update	108484	Selection.Home	oblivion.db.Selection[0000010001000000000101]	
Update	107423	Selection.Home	oblivion.db.Selection[00000100010000000001]	
Update	106528	Selection.Home	oblivion.db.Selection[0000010001]	
Update	105552	Selection.Home	oblivion.db.Selection[000001]	
Update	104880	Selection.Home	oblivion.db.Selection[]	
Update	104746	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	104683	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	103721	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	103129	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	103039	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	102851	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	102830	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	102274	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	100699	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	100148	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	99550	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	98417	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	98119	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	97568	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	97250	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	96538	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	96288	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	95981	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	95784	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	95518	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	95048	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	93214	Selection.Stadium	oblivion.db.Selection[]	
Update	92974	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	91404	Selection.Stadium	oblivion.db.Selection[000000000000000000000...	
Update	91268	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	91223	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	90769	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	
Update	90265	BitPermutation.CFV	oblivion.util.BitPermutation[[ST, HT, AT, DA]]	

Figure 3.12: An example of low-level user interactions in the Retrosheet visualization in Figure 3.1 captured and displayed in Improvise

Microplanning mainly focuses on generating the abstract form of questions and lexicalizes them using the relationship table populated in offline generation. Given the structure of the questions and what needs to be included, the microplanning module looks up the required information in the relations XML file, forms the questions, and assign words to the abstract representation.

Microplanning also performs aggregation on the questions. Aggregation is the process of removing redundant information in text. Aggregated questions are more natural and resemble human authored text. Dalianis and Hovy described five types of aggregation: Syntactic, Elision, Lexical, Unbound Lexical, and Referential [74]. *Syntactic* aggregation keeps at least one of the repeated items in the text to explicitly carry the meaning, and removes all other redundant information. Any part of speech, such as subjects, objects, and predicates, can be aggregated. For example, "Sara is sick. John is sick." can be syntactically aggregated to "Sara and John are sick."

Elision aggregation removes information that can be implicitly inferred and keeps no items in the text that carries that information. For instance, the sentence "My daughter visit me before school." is an aggregated version of "My daughter visit me before she goes to school."

Lexical aggregation transforms information into an aggregated but recoverable form. For instance, "Sara goes to work on Wednesdays, Saturdays, and Sundays" can be transformed into "Sara goes to work on Wednesdays and weekends." In this type of aggregation there is no information loss, unlike Unbound Lexical Aggregation.

Unbound Lexical aggregation is similar to Lexical aggregation, except that the aggregated information is not precise and hence not recoverable. For instance, "Sara has a part time job in Walmart and Target department stores." can be aggregated into "Sara has a part time job in department stores."

Finally, *referential* aggregation removes redundant information, but refers to them using references, such as pronouns. For example, "John and Sara are in graduate school. John and Sara study Computer Science." can be aggregated to "John and Sara are in graduate school. They study Computer Science."

In implementing aggregation for Q2Q, mainly syntactical aggregation is used. For example, "Who are involved with Avatar movie? and Who are involved with the action genre?" is aggregated to "Who are involved with the Avatar movie and the Action genre?". In a few situations, lexical aggregation is also applied. For example, if all the items are selected in the genre table and movies are filtered on genre, rather than listing them all in the question, the qualifier "all" is used in the translation: "What movies are in all genres?".

Translations in Q2Q are generated to satisfy reliability, precision, and straightforward recovery of information. Thus, Elision and Unbound aggregations are generally unsuitable for meeting Q2Q translation requirements. The current implementation of Q2Q does not use even unambiguous Referential aggregation due to its complexity. However, it may be considered in future work.

The grammar rules in the current implementation of Q2Q support four types of aggregation within and across questions, which results in four different deep structures of questions. Based on the current setting in the Q2Q user interface (the checkboxes labeled *Aggregate Questions?* and *Aggregate Items?* in Figure 3.3), *Question Representation* chooses one of the four questions output from the online generation engine, applies visual encoding, and displays it in the Q2Q interface. Users can request a different format and type for the questions at any time during their interaction with the visualization.

3.3.6 Time Complexity

In the implementation of Q2Q for cross-filtered views, the average time for offline generation to output a question is less than a second. For n data dimensions, the designer must specify $\binom{n}{2}$ data relationships. Given that most visualizations display fewer than 10 dimensions, the average time to set up Q2Q for a new visualization is reasonable: approximately two minutes plus the time needed for the designer to decide on the descriptions and input them to the system. Offline generation proper is spent primarily in parsing. Online generation is effectively interactively immediate, taking on the order of 50 milliseconds, which is often much faster than the queries themselves.

3.4 Summary

This chapter describes the architecture of a translation system, a hybrid automatic-manual system to capture and represent the meaning of visual query interactions as natural language. It also describes the implementation of Q2Q with considerations that are taken into account for the cross-filtered views case. Q2Q provides comprehensive recording of user activities at a level of abstraction suitable for visual analysis of multidimensional data. It is shown in this chapter that building off of existing NLG techniques makes it practical to design a system that is generalizable to various data domains and can be expanded to translation of different interaction techniques.

The content of this chapter is accepted and will be published in *IEEE Transaction on Visualization and Computer Graphics Journal* [87].

Chapter 4

Evaluation of Q2Q

4.1 Evaluation

The primary goal of the design of Q2Q is to increase the usability of multiple coordinated view visualizations created in systems like Improvise [15], and thereby to make them more accessible to a wider community of visualization stakeholders. To gain understanding of how successfully Q2Q enhances usability, we study and analyze differences in user performance when a visualization includes and does not include Q2Q. (For the remainder of this chapter, a visualization that includes Q2Q is referred to as *vis+Q2Q*. One that does not is refer to as *vis-Q2Q*.) We have designed a task-based experiment to assess user performance on three metrics: *speed*, *error ratio*, and *number of interactions*. We calculate these metrics in four different use scenarios. The scenarios are designed to expose how differences in visualization complexity and user familiarity affect the usefulness of Q2Q. An analysis of the data collected shows that Q2Q becomes more beneficial as complexity and visualization experience increase.

We also analyze the effects of translation on learnability, efficiency, and memorability. These aspects of usability are considered from the perspective of visualization interaction as a visual language for expressing data queries. The evaluation results suggest that Q2Q as a visual language translator plays an effective role in increasing the learnability and efficiency of visualization tools. Even though Q2Q also has a positive effect on memorability, user errors and feedback suggest that additional training on Q2Q would increase its effectiveness.

This chapter starts by describing the usability aspects considered. It then presents the metrics used to measure the usability attributes mentioned above, following a detailed description of the experiment design and observations. Finally, based on the observed quantitative and qualitative results, it presents an analysis of the translation system from a visual language perspective.

4.1.1 Usability Perspectives

The evaluation of the effectiveness of Q2Q on usability of visualization is based on a set of well-known attributes described independently, yet similarly, by Nielsen [75] and by Shneiderman [76]. The attributes, adapted to visualization user interfaces, are:

- *Learnability* is the ease of learning when a user encounters a visualization for the first time.
- *Efficiency* is the speed of performance after a user has learned the visualization.
- *Memorability* is the ease of remembering past queries when a user wants to interpret the current state of the visualization.
- *Satisfaction* encompasses the opinions of the user about the visualization.

In the original definitions of attributes given by Nielsen [75] and by Shneiderman [76], memorability refers to the amount of effort needed to retain knowledge about operation between sessions when users return to the interface after a time. This perspective on memorability focuses on long-term recall of past interactions across analysis sessions. This evaluation, however, consider the short-term memorability of a visualization within a session and the study of long-term memorability left as future work.

Both Nielsen and Shneiderman identified user errors as a characteristic of the individual. In this study, it is instead considered as a metric to assess the four usability attributes. The evaluation studies the attributes in an integrated way with the goal of achieving deeper insight about when and how Q2Q is beneficial and how its effectiveness can be improved.

4.1.2 Metrics

This study uses three metrics—*speed*, *error ratio*, and *number of interactions*—to assess the chosen usability attributes.

Speed is the time it takes a user to complete a visualization query task. To perform a task, users need to know which interactions must be performed and what the consequences of their interactions will be. If a user has insufficient understanding of a tool's functionality, they spend more time to comprehend the tool, particularly to make sense of their actions and the corresponding visualization reactions.

Error ratio is the total number of unsuccessful task performances divided by the total number of tasks. A successful task is one performed correctly and completely. It assumed that a user's ultimate goal is to gain information from the visualization. If due to errors in task performance the displayed results are not what the user seeks, it implies the visualization does not effectively serve their needs. A tool with high usability can avoid potential errors by providing easy to use and straightforward data representations and interactions.

Number of interactions is the number of discrete interaction steps users take to complete a task. The interactions—selecting items in a table and toggling checkboxes to apply filtering—are recorded at the level of data exploration actions, as defined in [6], rather than at a lower level such as mouse events. While it is hard to fully categorize the time spent by a user during task performance, (particularly if they do not talk aloud), the number of interactions allows calculation of how much time is spent on each interaction, and the total time spent on interaction sequences to accomplish a task. Since different tasks require different minimum numbers of steps, tasks are normalized by subtracting the known minimum number of steps from the actual number of steps performed. This calculation produces the extra, presumably unnecessary steps performed to complete a task.

Both *speed* and *error ratio* relate to the user's ability to extract information from a visual display. The study seeks to determine whether Q2Q can help overcome issues that either cause the user to slow down or that lead to errors. Even though *number of interactions* has a strong correlation with *speed*, *number of interactions* provides greater insight into users' decision making processes, such as whether their interaction decisions are based on trial and error, or are planned. Using this metric, the aim is to find out if a visualization with Q2Q helps users avoid unnecessary steps to accomplish their tasks, which if so suggests that Q2Q helps them make more sense of their interactions.

All three metrics are used to assess a visualization's learnability and efficiency (with and without Q2Q). Memorability is examined using only *speed* and *error ratio*, since the tasks that were defined for this aspect do not involve direct interaction with the visualization. The qualitative data from a questionnaire is used to assess user satisfaction with the visualization's usability (with and without Q2Q).

4.1.3 Design

Twelve participants, nine males and three females, were recruited, including undergraduate and graduate students from five different majors: seven from Computer Science, two from Industrial Engineering, one from Civil Engineering, one from Mathematics, and one from Electrical Engineering. They all had basic experience interacting with user interfaces, but not with Improvise visualizations.

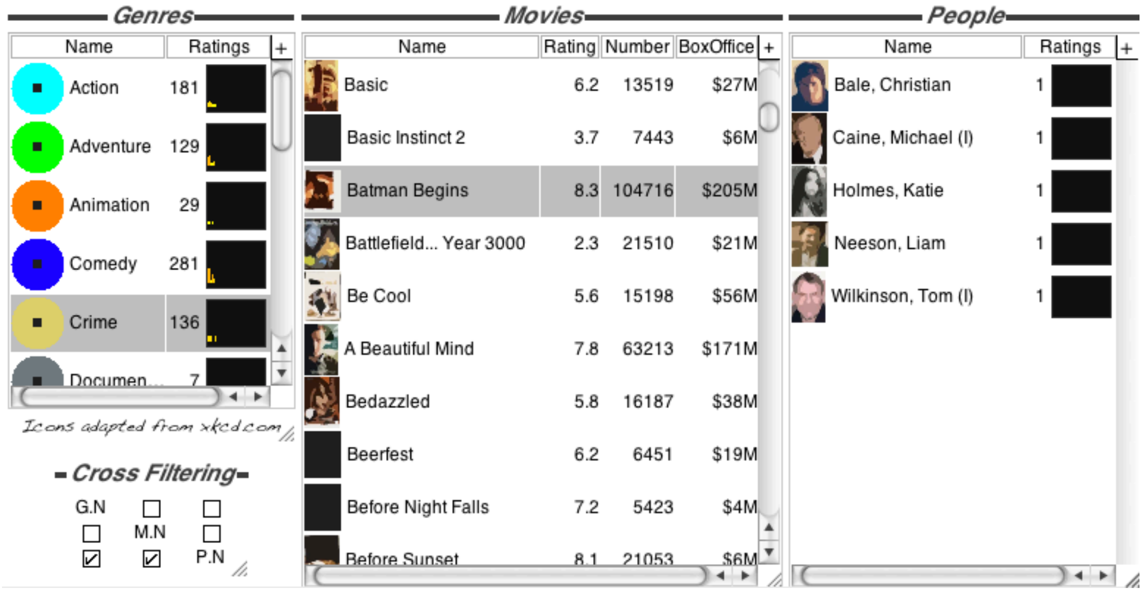


Figure 4.1: Visualization of movies in the Internet Movie Database. Three tables list genres, movies, and people.

Prior to the study, each participant watched a 10 minutes training video on cross-filtering and a 5 minutes video on Q2Q. Each experiment began immediately after training and lasted from 25 to 60 minutes. Two factors were considered: (1) participants' previous experience with the visualization and (2) their familiarity with the data attributes displayed. We used two Improvise visualizations from different domains to measure the effectiveness of Q2Q in terms of these factors.

The first visualization (*vis1*, see Figure 4.1) displays a data set about movies, a domain familiar to most participants. This visualization is a simplified version of the Cinegraph visualization presented in the InfoVis 2007 Contest [77]. The visualization in includes three table views—of Genres, Movies, and People— and a cross-filtering matrix (checkboxes to toggle filtering between each pair of dimensions).

The second visualization (*vis2*, see Figure 4.2) visualizes a data set about NSF awards, a domain unknown to most participants. The NSF visualization includes nine table views: of Directorates, Organizations, States, Program Managers, Programs, Application Fields, Institutions, Principle Investigators, and Co-PIs. It also

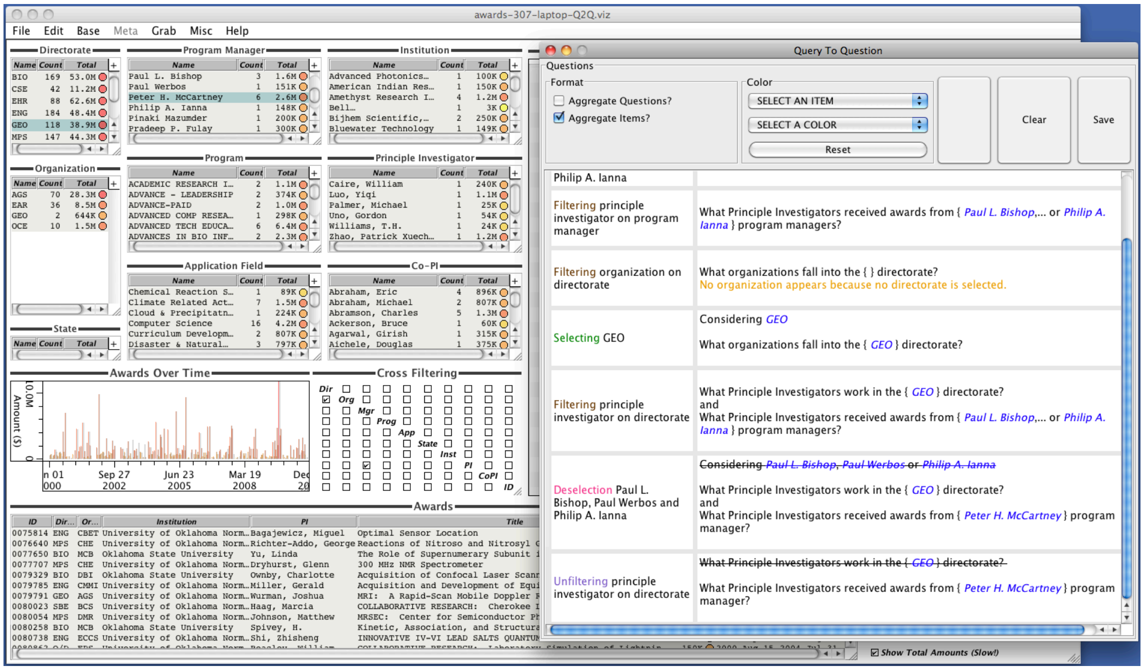


Figure 4.2: Visualization of NSF awards, accompanied by a Q2Q interface (the frontmost window on the right).

has a drill-down table of detailed awards information, a time-line showing variation in award amounts over time, and a cross-filtering matrix.

Vis2 is structurally more complicated in that it displays more data dimensions (nine) than *vis1* (three). We asked participants to perform tasks on table views only; we instructed them to ignore other views. We added Q2Q to each visualization, for a total of four user interfaces: *vis1+Q2Q*, *vis1-Q2Q*, *vis2+Q2Q*, and *vis2-Q2Q*. We divided the participants into two groups of six, each group presented with one of two sequences of interfaces, A or B:

	S1	S2	S3	S4
A:	*vis1+Q2Q*	*vis1-Q2Q*	*vis2-Q2Q*	*vis2+Q2Q*
B:	*vis1-Q2Q*	*vis1+Q2Q*	*vis2+Q2Q*	*vis2-Q2Q*

In both sequences *vis1* is shown before *vis2*. The difference between the sequences is the ordering in which Q2Q is included with the visualization or not. This variation reduces learning effects and allows comparison of the usability of *vis+Q2Q* and *vis-Q2Q* across the two sequences. These orderings enables the comparison of user performance in four scenarios:

***S1*:** A participant interacts with *vis1* for the first time.

***S2*:** A participant interacts with *vis1* for the second time.

***S3*:** A participant interacts with *vis2* for the first time.

***S4*:** A participant interacts with *vis2* for the second time.

These scenarios are studied for each participant in the order (*S1*–*S4*) presented above. As the experiment proceeded, each participant's experience and knowledge of the cross-filtering technique used in the visualizations increased. The purpose of *S1* is to observe the effects of Q2Q on novice users' ability to learn a new visualization. *S2* is designed to examine the efficiency of user performance in the presence of Q2Q when they have prior experience using the visualization. Once participants reach the more complex visualization in *S3*, they are familiar with how to work with visualization, but have no background knowledge about the domain displayed by *vis2*. The *S3* scenario is designed to examine the effects of Q2Q on user ability to learn and understand a new visualized domain. *S3* allows us to examine the effects of the complexity of domain, independent of familiarity with allowed interactions. Finally, *S4* is designed to analyze the case in which participants know how to use the supported interactions and have at least basic understanding of the domain. The *S4* also provides insight into whether participants use Q2Q in this scenario, determine how helpful it is, and observe whether they use it for other purposes, such as to review past interactions and corresponding visualization states.

In each scenario, participants performed several tasks on a combination of *vis* and *Q2Q*. Each task is to answer an *interactive* or *descriptive* question about the visualized data. *Interactive* questions ask participants to identify a set of data values using selection and filtering interactions. The questions progressed from simple to complex. Simple questions required only one selection and filtering step, e.g., questions 1 and 2 in Table 4.1. Complex questions required multiple steps, e.g., questions 3 and 5 in Table 4.1.

Descriptive questions ask participants to describe a portion of query results in terms of other data dimensions. For such questions, some data values that affect visual state are hidden after selection and filtering actions, e.g., question 7 in Table 4.1. This task requires remembering past states if Q2Q is not present. For both *vis+Q2Q* and *vis-Q2Q*, participants answered six questions for *vis1* and seven for *vis2*, for a total of twenty six questions. Also, each participant filled out a questionnaire to express their opinions of Q2Q, whether they prefer to interact with the visualization with or without Q2Q and how helpful they found Q2Q to be.

The complete sets of tasks performed in this study are listed in Appendix C.

1. Identify the organization(s) that is in the GEO directorate.
2. Identify the program manager(s) working in GEO directorate.
3. Identify the organization(s) that the programs Aeronomy or Climate & Large-scale Dynamic fall into, and program manager Jay S. Fein works under, and it is in GEO directorate.
4. Identify the organization(s) that the Application Fields Chemical Reaction Systems or System Theory are in.
5. Identify the Application Field(s) that the program managers from Muriel G. Poston to Wendell Talbot Hill work under.
6. Identify the Program manager(s) who work under the selected application field(s).
7. Characterize the resulting program manager(s) in terms of other data dimensions.

Table 4.1: Interactive (1–6) and descriptive (7) questions asked of participants about the interface in Figure 4.2.

4.1.4 Quantitative Results

For each scenario, the usability measurements are averaged over all performed tasks and over all participants involved in the scenario. Considering the extracted differences between averaged results of the group of participants interacting with *vis-Q2Q* and the other group interacting with *vis+Q2Q*, a Student's unpaired t-test is performed to assess their significance. Table 4.2 shows the scenario-based results. Three quantitative hypotheses are tested:

- *Users are faster on average with Q2Q than without.*
- *Users perform fewer errors on average with Q2Q than without.*
- *Users perform fewer interactions on average with Q2Q than without.*

Table 4.2 shows user performance on the usability measurements in each of the four scenarios. The results show improvements (shown in **bold** in Table 4.2) in both speed and number of interactions in all scenarios for *vis+Q2Q*. Even though the error ratios for *vis+Q2Q* do not improve over *vis-Q2Q* in *S1* and *S3*, they do improve for *S2* and *S4*.

	S1					S2				
	vis1+Q2Q		*vis1*-Q2Q		p	*vis1*+Q2Q		*vis1*-Q2Q		p
	AVG.	SD.	AVG.	SD.	value	AVG.	SD.	AVG.	SD.	value
Speed (sec)	**47.67**	23.75	67.39	45.42	0.19	**44.67**	21.86	56.75	26.63	0.21
Error ratio	0.16	0.11	0.14	0.19	0.39	**0.14**	0.09	0.33	0.21	0.04
#Interactions	**2.44**	2.69	5.78	6.38	0.14	**2.53**	1.3	4.94	3.75	0.09
	S3					S4				
	vis2+Q2Q		*vis2*-Q2Q		p	*vis2*+Q2Q		*vis2*-Q2Q		p
	AVG.	SD.	AVG.	SD.	value	AVG.	SD.	AVG.	SD.	value
Speed (sec)	**52.98**	25.53	67.35	45.09	0.26	**44.69**	16.98	58.02	19.48	0.12
Error ratio	0.21	0.22	0.12	0.14	0.2	**0.14**	0.18	0.28	0.24	0.14
#Interactions	**1.22**	1.25	2.0	1.85	0.21	**0.78**	0.41	1.44	0.62	0.03

Table 4.2: Averages and standard deviations of usability metrics. Improvements for visualizations with Q2Q are shown in **bold**.

To test the significance of the improvements, a relatively large threshold, $p \leq 0.15$, is used due to the small sample size and large standard deviations. The threshold allows us to analyze the results of differences in usability metrics for *vis+Q2Q* and *vis-Q2Q*. The results can still be meaningful even if they are not significant based on smaller thresholds. In *S1*, in which participants interact for the first time with a visualization of a familiar dataset, they performed statistically significantly fewer interactions when they had access to the Q2Q interface. There was also considerable improvement in the amount of time spent on average per task (30% improvement in *vis+Q2Q*; see Table 4.2). The time difference was not statistically significant, however. In *S2*, in which participants had gained some experience with the interaction techniques from *S1*, a statistically significant fewer number of interactions and a lower error ratio occurred. As in *S1*, even though participants perform a task approximately 20% on average faster when the visualization includes Q2Q, the difference was not statistically significant.

S3 had no statistically significant differences between *vis2+Q2Q* and *vis2-Q2Q*. However, the actual difference in speed was considerable. Not being familiar with the domain, and interacting with larger number of data dimensions compared to *S1* and *S2*, might be two reasons why participants had difficulty performing the tasks even in the presence of Q2Q. Performing tasks in *S3* helped participants achieve a basic understanding of the new domain, leading to significant performance improvements in speed, error ratio, and number of interactions in *S4* when they had access to Q2Q. The differences between *vis2-Q2Q* and *vis2+Q2Q* for all metrics were statistically significant. *S4* suggests that Q2Q is particularly useful for cases in which the domain is complex—in terms of relationships among data attributes—and is not familiar, but that participants have at least some experience with the visualized data. Note that by the time they encounter *S4*, participants had more experience using Q2Q during their analytical tasks.

When participants encounter a new visualization, in *S1* and *S3*, it takes them time to make sense of the relationship between the visualization and Q2Q. As long as the relationship is not clear to them, they cannot fully utilize Q2Q to perform their tasks. However, once they understand this relationship, in *S2* and *S4*, they are able to take advantage of the capabilities of Q2Q to perform tasks more efficiently. This observation suggests a future follow-up experiment to evaluate the effects of Q2Q when users are fully trained on the functionality of the translation system, specifically to see if results for scenarios *S1* and *S3* improve.

Further data analysis studies the overall performance of participants over entire analytical sessions. For each participant, the study measures the number of interactions, time to perform the tasks, and error rate in each session, both with (*vis+Q2Q*) and without (*vis-Q2Q*) the Q2Q display. It calculates the differences between vis-Q2Q and vis+Q2Q for each metric, then averages the differences in measurements over all participants to calculate an average difference for each metric.

A Student's paired t-test is performed to assess the three hypotheses (users are faster, perform fewer errors, and perform fewer extra interactions on average when the visualization includes Q2Q). The results for these hypotheses are considered as statistically significant if $p \leq 0.05$. The increase in sample size due to studying the performance over all the scenarios, allows for using standard 0.05 p-value over 0.15. Table 4.3 shows the average and standard deviation of each defined metric (including both visualizations) for both *vis+Q2Q* and *vis-Q2Q*. The results show that participants' performances in terms of time to perform tasks, number of errors, and number of interactions improve when they interact with vis+Q2Q. The improvement in the time it takes them and the number of extra interactions they perform to complete the experiment are statistically significant.

	vis+Q2Q		*vis*-Q2Q		p
	AVG.	SD.	AVG.	SD.	
Time (min)	95	39.14	124.76	61.40	$0.002 < (0.05)$
Error ratio	0.33	0.25	0.44	0.3	$0.08 > (0.05)$
#Interactions	3.48	2.41	7.08	5.69	$0.006 < (0.05)$

Table 4.3: Averages and standard deviations of usability metrics.

4.1.5 Qualitative Results

At the end of the experiment, a follow-up survey asked participants to express their opinions about Q2Q and whether they preferred to use the visualization with Q2Q or without. If they preferred to use Q2Q, the survey provides a set of reasons Q2Q could be preferred and asked participants to select reasons that applied to them. If they did not preferred to use Q2Q, the survey asked participants to provide their reasons. Also it asked participants whether they found different types of generated languages (aggregated/not aggregated) to be beneficial during task performance. The survey also encouraged Free form commentary.

Of the twelve participants, eleven (91%) preferred to use Q2Q. The reasons they chose are listed in Table 4.4. Some participants provided more than one reason. The only participant (9%) who did not prefer to use Q2Q declared that *"...the cross-filtering technique is very complicated and I do not prefer to use this visualization with or without Q2Q."* It is worth mentioning that this participant's results with Q2Q showed better performances than without Q2Q.

For language configuration, five participants (41%) used the *Aggregate Items?* option provided by Q2Q to toggle aggregation/non-aggregation of selected data items in the text. The main reason they mentioned for using it was *"It is easier to see the results."* Four (57%) participants did not manipulate the type of text using aggregation options. All four admitted that they forgot this option existed. Two others (29%) stated *"...different types of text did not seem useful in their case."*

In the comments received, three participants (27%) mentioned that Q2Q helps them to verify which queries they asked. A few comments were received about

Reasons	Rates
I can see the result of my interaction easier.	73%
Q2Q makes the visualization more understandable.	54%
Q2Q makes the visualization easier to use.	45%
Q2Q helps me remember how to perform a task if I have to perform a task more than once.	36%
I can find what I am looking for faster.	36%

Table 4.4: Satisfaction rates for the eleven participants.

the need for more training at the outset. One participant expressed a preference about the order of the text in the translation log. In the current system, beside the outer temporal ordering of the interactions, there is an inner ordering “within” each interaction to present a set of dependent queries in a consistent manner. The participant expressed a preference for the existing choice of inner ordering. She suggested allowing the user to adjust the interaction order for the inner ordering as well, rather than keeping the inner ordering constant.

4.2 Discussion

The main role of Q2Q in the communication cycle between a user and a visualization is to transform the visual representation language into a semi-textual language that is close to human written language. Rather than users relying solely on perception and comprehension to translate the visual language directly into their mental language, they can also access an accurate and consistent written language for easier, indirect translation. The quantitative and qualitative experimental results verify the effectiveness of such indirect translation through text. This section analyzes the reasons behind how Q2Q improves learnability, efficiency, and memorability, from the perspective of a visual language model.

From a computational perspective, a visual language is a system of communication used to facilitate interactions between computer and human. This study assumes that the system consists of three components: a computational system, a cognitive system, and a visual language [78]. Interaction can be viewed as a cycle of transformations between languages used by the cognitive and computational systems, as described by Abowd and Dix [79]. To perform an interaction, users translate their cognitive steps into a mechanical language that is expressed through interaction techniques. The computer system parses and interprets the language and responds to the user query in the same form of language. Users perceive, comprehend, and reason about the computed response by translating the visual language into the mental language in which they form their mental models. Successful interaction depends on the ability to formulate, understand, and quickly translate between these different languages. The analysis in this section is based on this visual language model as it applies to visual querying in particular (see Figure 4.3).

Learnability

The learnability of a visualization depends on how easily users can express their queries in the interaction language and comprehend the corresponding result when they see the visualization for the first time. Scenario *S1*, described in Section 4.1.3 was designed to study the usability effects of Q2Q, particularly on learnability. The results in Table 4.2 show promising improvements in performance in terms of speed and number of interactions when a visualization is accompanied by Q2Q.

While performing tasks, users spent a significant amount of time translating the language of coordinated visualization and the corresponding interaction sequences into their cognitive language. Users not only need to translate the visual language into their mental language, but also make sense of the textual translation provided by Q2Q. It may seem having an external translation system such as Q2Q would lengthen performance time. However, the quantitative observations and statistical

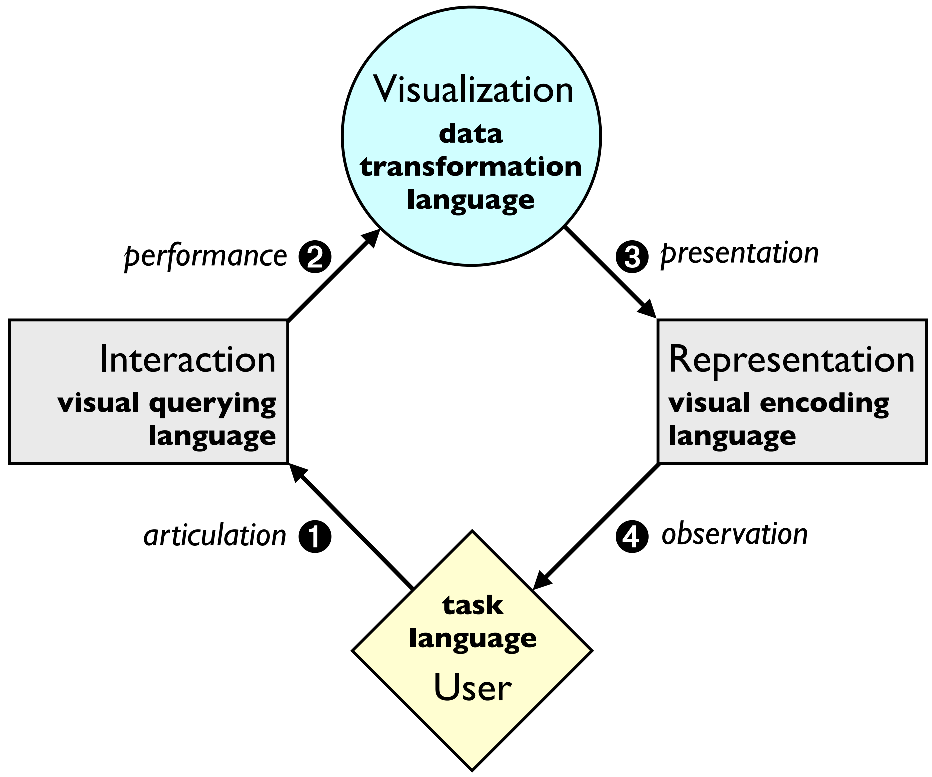


Figure 4.3: Visual language model, specialized for visual querying.

results suggest that users spend less time relying on their cognitive interpretation; instead, they accept the textual translation as a means to understand the language of visualization interactions. As a result, a user's time spent on translation between languages is considerably reduced.

In terms of error ratio, Q2Q does not produce any improvement when learning interactions (see *S1* in Table 4.2). One major reason can be the learnability of the Q2Q interface. The observations revealed that most errors in *vis1+Q2Q* under *S1* were due to user misinterpretation of Q2Q translations. These misunderstandings resolve as users progress to later scenarios and become familiar with Q2Q.

There was a considerable reduction in the number of superfluous interactions when participants had access to Q2Q in *S1* (see Table 4.2). Participants tried to learn and understand the interaction language by performing many trial and error interactions. However, they had difficulty comprehending the computed responses, and thus the consequence of their interactions (in the visual language) when the visualization did not include Q2Q. They also repeated a large portion of their interactions. When Q2Q translations are available, they did not perform as many repeated interactions. These results (Table 4.2) show that the translation system facilitates understanding of the visual language while users perform analysis tasks.

Efficiency

In this study, efficiency is defined in two ways: (1) the speed of performing individual interaction tasks, and (2) the overall speed of performing a sequence of tasks as a function of the percentage correct. The second definition is highly correlated with effectiveness of performing individual tasks. For any given large number of tasks, if users make fewer errors, they are accomplishing more in a given amount of time.

Complex visualizations with coordinated multiple views suffer from low efficiency since it is more difficult for users to maintain a mental model of interactive dependencies between views [16,26,27]. In particular, exploration or analysis often requires switching between tasks that involve different views or different kinds of interactions. Switching from a set of views to another set imposes a heavy cognitive overhead. Errors can be made because users are managing evolving mental models of overlapping subsets of visualization components. Consequently, there is potential for a large number of errors to be made over a long period of time, affecting the efficiency of successful operation overall. Q2Q is able to speed up this process by simultaneously reflecting the set of all queries that led to the current state of the visualization. Users have correspondingly less difficulty constructing their mental model since they have access to interaction events of current and recent states in semi-textual form.

Scenarios *S2* and (especially) *S4* are designed specifically to support analysis of the efficiency of visualization using Q2Q. The results show considerable improvement in usability metrics in both scenarios in which Q2Q is available. Without Q2Q, the visual language translation in the cognitive system would take time due to the complexity of the visualizations. With Q2Q, the response of the computer system to the interaction is expressed in user language and describes all of the cross-filtering relationships that exist between views. This not only accelerates the language transformation process and therefore increases the efficiency of the users in interacting with the visualization, but also mitigates inaccurate and misleading analysis by bringing all affected dimensions into visual display—in the limit of available screen space—and thus brings them to the user's attention.

In this experiment, we observed a few outstanding situations in which the presence of Q2Q helped users have less trouble understanding the visual language. We categorized these situations into three cases based on the character of the *responses* of the computer system after a single interaction or sequence of them.

- *No response.* Users generally expect each action to have a response in visual form. When there is no visual feedback, users interpret this as if they did not express their query correctly in the interaction language. Consequently, they try further interactions, especially if they are unfamiliar with the domain and they do not know what to expect as the result of their interactions. However, it is observed that with Q2Q, users have better understanding of the visual responses—in this case, no response—despite the fact that the feedback is in textual rather than visual form. They can make better sense of the current visual state and the reasons why the most recent interaction did not change the visual display.

- *Misleading response.* In the case of *selection occlusion* in cross-filtering (described in Chapter 1.3), the visualization obscures the relationships between views. Generally, if the views in a visualization show incongruous results in response to an interaction, the user's understanding of the visual language expressed by these views would be incorrect and would result in misunderstanding of the computer responses. In such cases, the Q2Q text does not match what the visualization shows. Users need to learn to trust the textual response over the visual one.

- *Abnormal response.* Comprehension of the visual language is difficult for users when they perform an interaction that is an "error". For instance, filtering a table on nothing results in the disappearance of all the items in the filtered table, which often confuses users. With the warning messages provided by Q2Q such as the warning in orange in the second row of the Q2Q interface in Figure 4.2, users can make sense of the visual responses and modify their interaction to interpret this situation.

In the *abnormal* and *no response* cases, the presence of Q2Q results in substantial improvements in the usability metrics (i.e., better performance on interactive tasks). In the case of *misleading response*, however, Q2Q provides less benefit (i.e., limited improvements on descriptive tasks). There is a need for more training on Q2Q to make it more effective in *misleading response* cases. In Chapter 5, these situations are discussed further using examples in Immigrant Boats Visualization.

Memorability

In this study, a visualization is considered memorable if users are able to recall past states as a part of making sense of the current visualization state. To see the effect of Q2Q on memorability, we measured how quickly and correctly participants can answer descriptive questions.

Table 4.5 shows the results of measurements of memorability tasks in the four scenarios. The task performances with Q2Q in scenarios *S2*, *S3*, and *S4* (shown in **bold** in Table 4.5) are promising. The outcome for *S1* is poor. The improvement in speed is statistically significant in all scenarios except *S1*. The error ratio

	S1					S2				
	vis1+Q2Q		*vis1*-Q2Q		p	*vis1*+Q2Q		*vis1*-Q2Q		p
	AVG.	SD.	AVG.	SD.	value	AVG.	SD.	AVG.	SD.	value
Speed (sec)	36.08	39.31	26.67	14.61	0.3	**16.42**	9.61	43.75	18.19	0.008
Error ratio	0.42	0.37	0.25	0.41	0.24	**0.42**	0.37	0.75	0.41	0.09
	S3					S4				
	vis2+Q2Q		*vis2*-Q2Q		p	*vis2*+Q2Q		*vis2*-Q2Q		p
	AVG.	SD.	AVG.	SD.	value	AVG.	SD.	AVG.	SD.	value
Speed (sec)	**16.83**	16.20	33.33	17.78	0.07	**24.33**	10.63	37.5	22.68	0.12
Error ratio	**0.33**	0.51	0.5	0.54	0.3	**0.5**	0.54	0.67	0.51	0.3

Table 4.5: Averages and standard deviations of speed and error ratio for memorability tasks. Improvements are shown in **bold**.

significantly improved only in *S2*. These observations suggest that the main factor of poor performance in *S1* in *vis1+Q2Q* is the lack of sufficient training on Q2Q. The considerable performance improvements in the later scenarios, in which users have more experience with Q2Q, affirm that conclusion. A significant drop in error ratio is anticipated if users are trained to use Q2Q properly (e.g., more training or one-on-one training) to perform memorability tasks, in particular by trusting that translated questions correctly reveal the data values hidden in the visualization.

Based on our observations, the difficulty of performing memorability tasks when Q2Q is not present mainly occurs: (1) when the sequence of previous interactions involves several cross-filterings (the *out of sight, out of mind* problem described in Chapter 1.3); and (2) when, as a result of several cross-filterings, some items of interest are no longer visible, i.e., there is a misleading response. These effects decrease memorability and degrade efficiency. In these cases, the memorability tasks require users not only to comprehend the recent visual responses of the visualization, but also to recall previous "conversations" between the cognitive and computational system that led to the current state. Q2Q displays the history of interactions in more human-familiar textual language. Users can recall their conversations with the visualization and review their prior queries to make sense of the most recent visualization state.

4.3 Summary

This chapter describes an evaluation of the effects of a translation system, Q2Q, on certain aspects of usability—learnability, efficiency, memorability, and satisfaction—of multiple coordinated view visualizations. The evaluation revealed considerable improvements in learnability, efficiency, and memorability of the speed and length of interaction paths followed. It also showed modest improvements in error ratio. Using scenario-based analysis, the study revealed that as users' experience with a visualization increases, the more that Q2Q helps to improve their performance. Questionnaires further revealed improvement in user satisfaction. By analyzing these observations from a visual language perspective, we conclude that Q2Q, as an interface component that translates visual language to a semi-textual human language, helps users to better comprehend and operate visualization interfaces, particularly when visual representations of data hide items or relationships due to filtering.

The content of this chapter was published in different form in *Computer Graphics Forum, the International Journal of Eurographics Association* [50].

Chapter 5

Opportunities and Limitations

Translation of user interactions provides substantial improvements in usability of coordinated multiple view visualizations, as described in Chapter 4. This chapter provides more examples and presents Q2Q as a tool extension to support *cross-examination* of visualization interaction. The chapter then describes key visualization design factors, corresponding guidelines, and challenges for effective translation of interactions.

5.1 Cross-Examination Using Q2Q

Throughout an analytical session, users express questions by performing interactions that trigger queries. They interpret visual responses to their actions based on the questions they intended to express and the queries they think they have performed. A potentially large gulf of evaluation [80] can arise between what is requested from the visualization and what is discerned in changes to visualization state. Users of sophisticated tools with multiple visual representations and many interactive interdependencies have difficulty making sense of the interaction sequences and consequent visualization responses. They might also have difficulty remembering their past queries in order to correctly interpret the current state of the visualization [17].

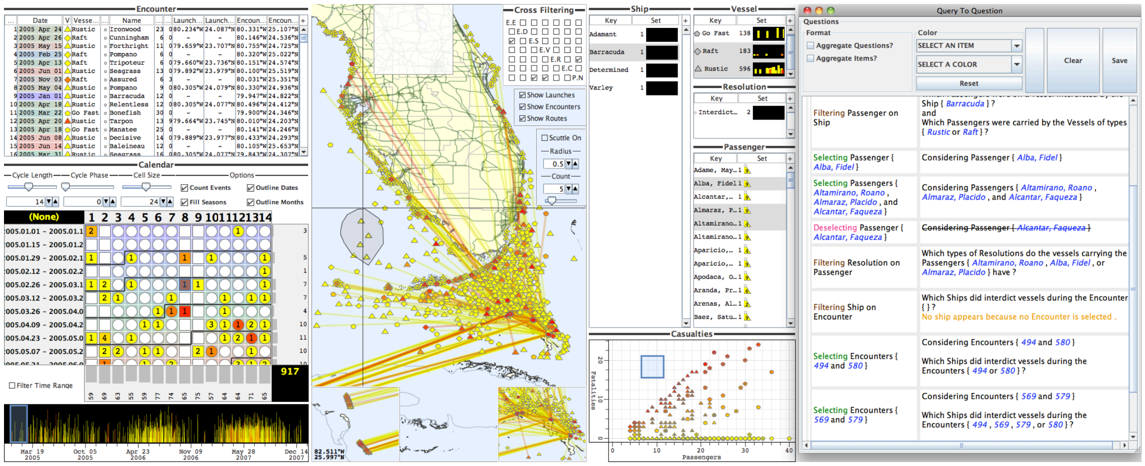


Figure 5.1: Migrant interdiction visualization in an intermediate state (left), alongside the corresponding Q2Q interface (right).

This results in misinterpretation of visual responses relative to the intention of interaction and consequently unreliable analysis and insight. Capturing and representing user interactions explicitly can assist users in making sense of how the interactions they perform correspond to changes in visualization state. Thus, translation can support a visual *cross-examination* process in which users can cross-check the validity of their questions as an integral part of reasoning.

Providing an accurate record of the questions users have asked helps them validate their queries and continue their analyses. The term “cross-examination” is used by way of loose analogy to legal questioning of witnesses on the stand, and similar forms of interrogation. Much like an attorney asks questions to elicit accurate testimony from a witness, a visualization user interactively constructs queries to elicit accurate depictions of relationships from the data. However, visualization is much like a (potentially) unreliable witness, in that visual states (the witness’ answers) do not necessarily convey the queries (questions) that the user intended to ask. Careful attention to queries/questions as they seem to be interpreted by the visualization/witness can help to address this gulf of evaluation [80].

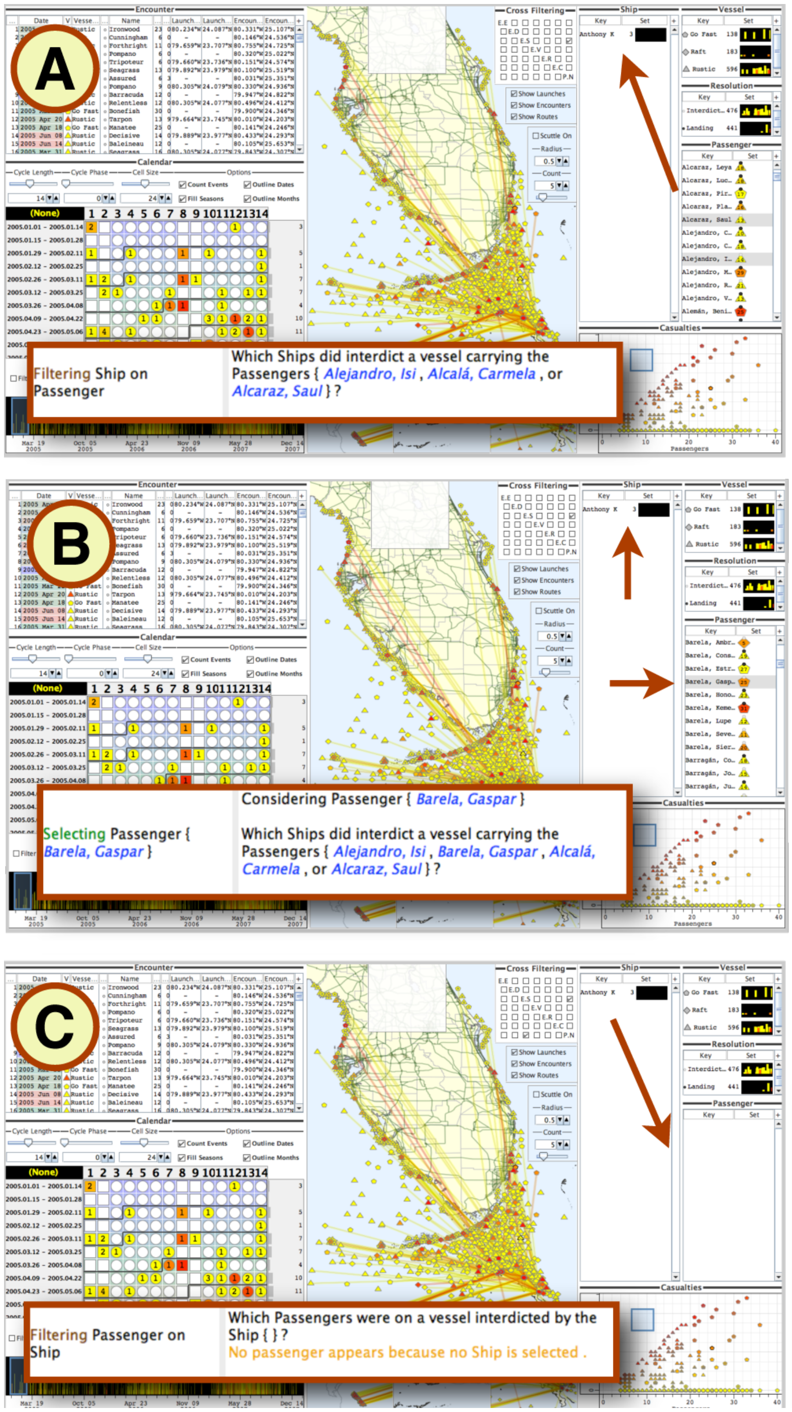


Figure 5.2: Sequence of visualization states resulting from interactions with the visualization in Figure 5.1: (A) initial state; (B) a state showing a *no response* situation; (C) a state showing an *anomalous response* situation.

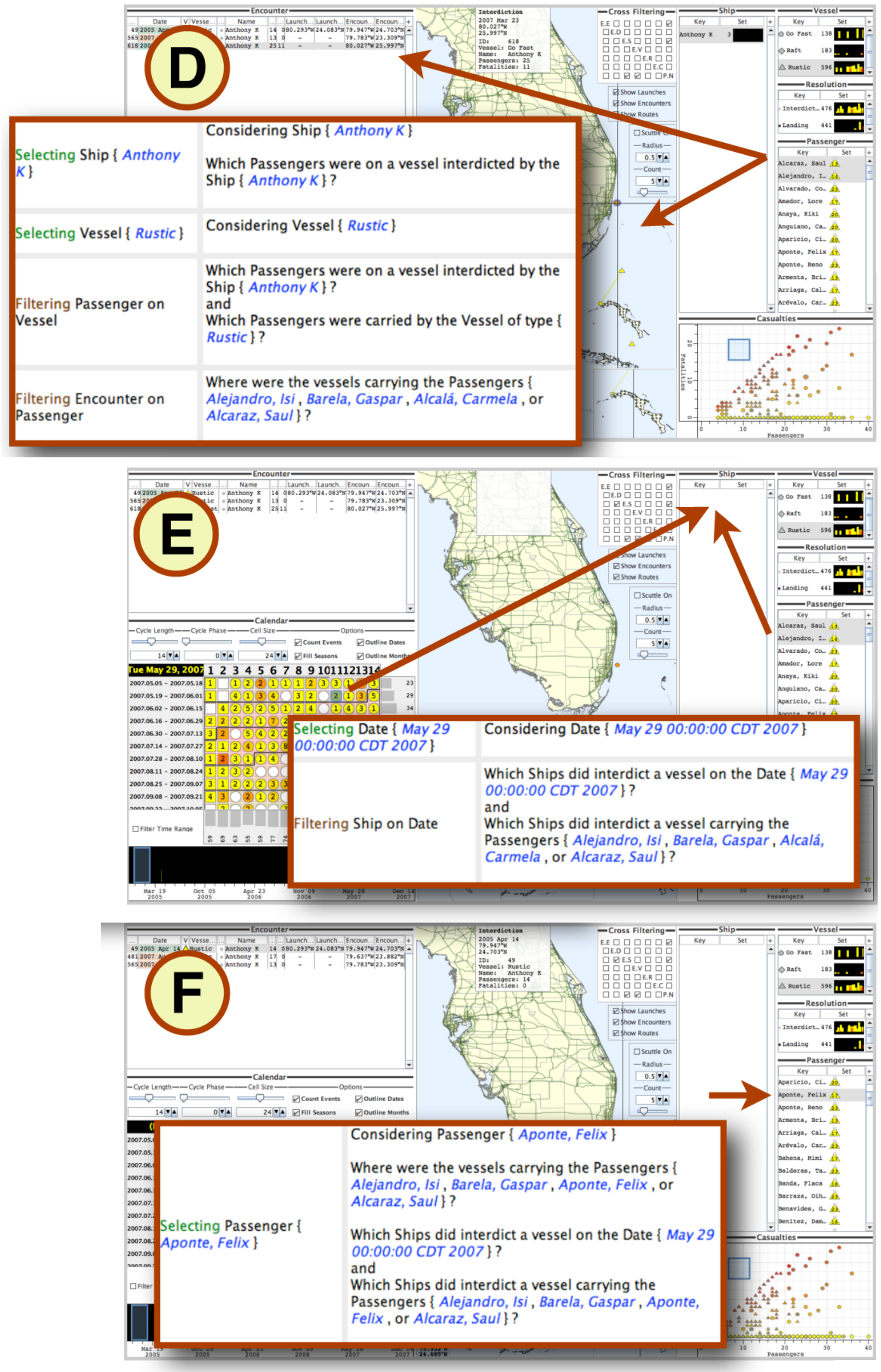


Figure 5.3: Sequence of visualization states resulting from interactions with the visualization in Figure 5.1: (D) a state showing a *misleading response* situation; (E-F) a pair of consecutive states showing the *out of sight, out of mind* effect.

This chapter describes four situations to show how translation of user interactions can help users better interpret visualization states.

Three of the situations in which users have difficulty interpreting visualization states were discovered in relation to the evaluation of Q2Q described in Chapter 4: when the visualization gives *no response*, a *misleading response*, or an *anomalous response* to an interaction [50]. The forth situation is when the *out of sight, out of mind* problem affect recall [18]. These situations are described further below.

Consider a visualization of a migrant boat interdiction and landing scenario [14] (Figure 5.1) and a sequence of visualization states in it (Figures 5.2 and 5.3). The sequence starts from an intermediate state in which three passengers (*Alejandro Isi*, *Alcala Carmela*, and *Alcaraz Saul*) are selected in the passengers table view and dimension *Ship* is filtered on those passengers (Figure 5.2A). (In each subfigure in Figure 5.2 and 5.3, the question(s) generated by Q2Q in response to the most recent interaction is enlarged and shown in an inset box for greater readability.)

No response refers to a situation in which an interaction has no apparent effect on query outputs. Figure 5.2B shows a state in which the user selects another passenger, *Barela Gaspar*. The *Ship* table does not show any visual change as a result of the selection interaction, meaning the ship that interdicted the recently selected passenger is the same as the one that interdicted the previously selected passengers. As was observed in the study in Chapter 4, in this case, if users are not familiar with the domain and do not know what to expect from their interactions, they often think they have performed the query incorrectly and try to formulate it in a different way. However, the last question generated by Q2Q (the inset in Figure 5.2B) reflects the query asked. Users are able to confirm their queries and alleviate their potential confusion about whether a query happened.

Misleading response refers to a situation in which an interaction obscures query inputs, e.g., *selection occlusion* in cross-filtering, which can lead to incorrect insights. In the third row of the Q2Q view in Figure 5.3D, the *Passenger* table is filtered on

Rustic vessel. This results in a state in which only two of four selected passengers—*Alejandro Isi* and *Alcaraz Saul*—are visible in the table, even if one scrolls it. Later, the user filters *Encounter* on the previously selected passengers; encounters are shown both in a table and on the map. The two passengers hidden in the *Passenger* table might imply that the resulting encounters on the map are the ones that just the two visible passengers were involved in. However, textual representation of queries reveals that the query that is taking place also involves two other passengers, *Alcala Carmela* and *Barela Gaspar*. Q2Q explicitly presents query inputs that cannot be or are not made apparent in the visualization itself due to filtering.

Anomalous response refers to a situation in which an interaction has an unexpected effect on query outputs. For instance, filtering a table on nothing (an empty set) causes all of its items to disappear, which often confuses users. In Figure 5.2C, the user filters *Passenger* on *Ship* to see which of the passengers were interdicted by the ship shown in the *Ship* table. Since no ship is selected in the *Ship* table, no passenger is shown in the *Passenger* table. To relieve confusion, Q2Q shows a warning message with the query that corresponds to the interaction. Later, the user selects *Anthony K* from the *Ship* table, resolving the anomaly (Figure 5.3D). This interaction is revealed in the questions in Figure 5.3D.

Out of sight, out of mind occurs when a user has difficulty remembering visualization states after only a few subsequent interactions. Figure 5.3E shows a state in which a user filters *Ship* on *Date*. Q2Q not only reflects recent interactions, but also reminds the user of prior filters that affect the current query (e.g., that a filter on *Ship* occurred in Figure 5.2A). Q2Q does that by presenting all the questions about previous filters that affect the current query. In Figure 5.3F, the user selects another passenger for exploration. Q2Q translates all queries that are triggered by this interaction. Such queries may be forgotten after subsequent interactions, but still affect the visualization's current state. Based on the observations in evaluation study described in Chapter 4, by providing a history as a means to quickly see and recall interactions beyond the horizon of memory, the memorability and reliability of actions and of the analysis process overall are increased.

These situations call attention to the importance of support for query verification, particularly in visual analysis tools that support sophisticated interaction techniques to compose complex visual queries. Q2Q can serve as a general means to support a more reliable analysis process.

In addition to the benefits provided by Q2Q, the user experiment in Chapter 4 as well as the process of designing and implementing a translation system in Chapter 3 led us to identify a set of key factors that influence the design of the translation. The next section identifies and discusses these key factors.

5.2 Design Factors

Successful design of NLG techniques depends on having examples of human-written text. There exist neither corpora of translations of user interactions in information visualizations nor guidelines for how translated text should be designed and structured. Developing an architecture for natural language translation of visualization interaction involves a variety of design factors. Three design factors that we explicitly take into account in the design of Q2Q are: *users of the text*, *data involved in interaction*, and *types of interaction*.

5.2.1 User Knowledge and Roles

The design of Q2Q—as it is expected of any interaction translation system—depends on the knowledge and roles of users. We further classify user knowledge as: *domain knowledge*, *technical knowledge*, and *linguistic knowledge*.

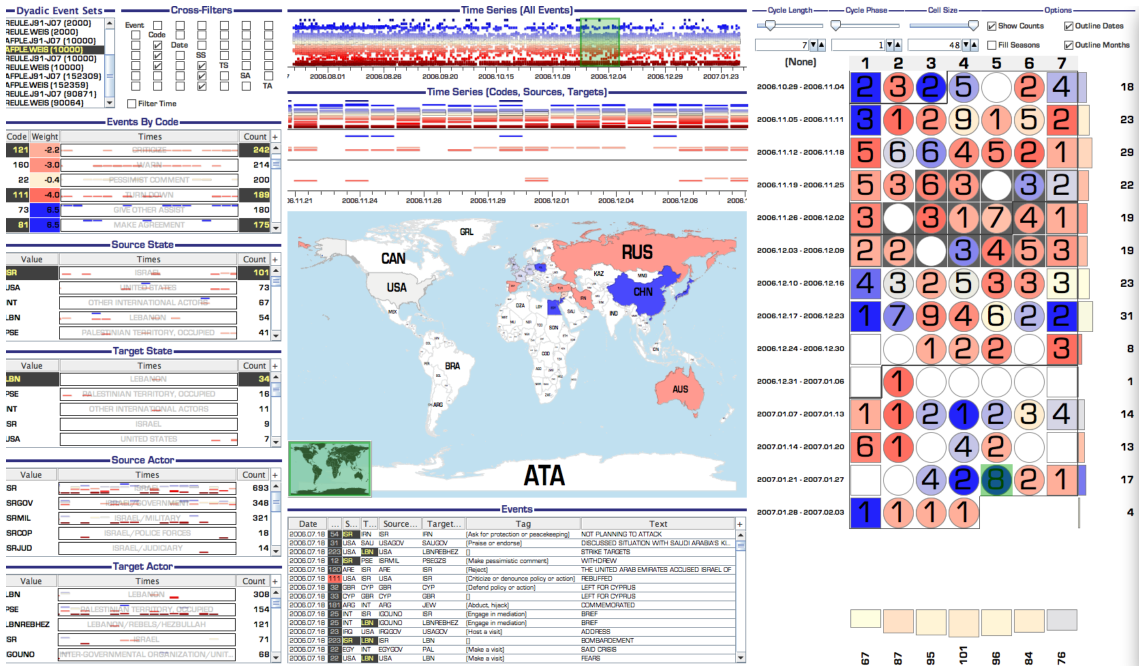


Figure 5.4: Visualization of events in newswire stories.

5.2.1.1 User Knowledge

Domain knowledge refers to the knowledge specific to a domain of application, in particular the domain of application that provides the data being visualized. Domain knowledge directly affects linguistic decisions such as lexicalization and aggregation. If users have limited knowledge of an application, translations can help to learn relationships between data attributes and acquire basic knowledge of the domain. In that case, a generator should prefer not to describe relationships in abstract language or using domain-specific vocabularies. Consider political events in newswire stories recorded in the Kansas Event Data System (KEDS) [81]. The visualization in Figure 5.4, shown with its Q2Q interface in Figure 5.5, can answer questions related to geographical and temporal patterns of international political activities reported by major news services. The visualization supports cross-filtering on event codes and abbreviated source and target states and actors. Although trained political scientists may prefer to interact directly with codes and abbreviations, new users are likely to struggle to map codes to defined meanings. For example, to translate filtering

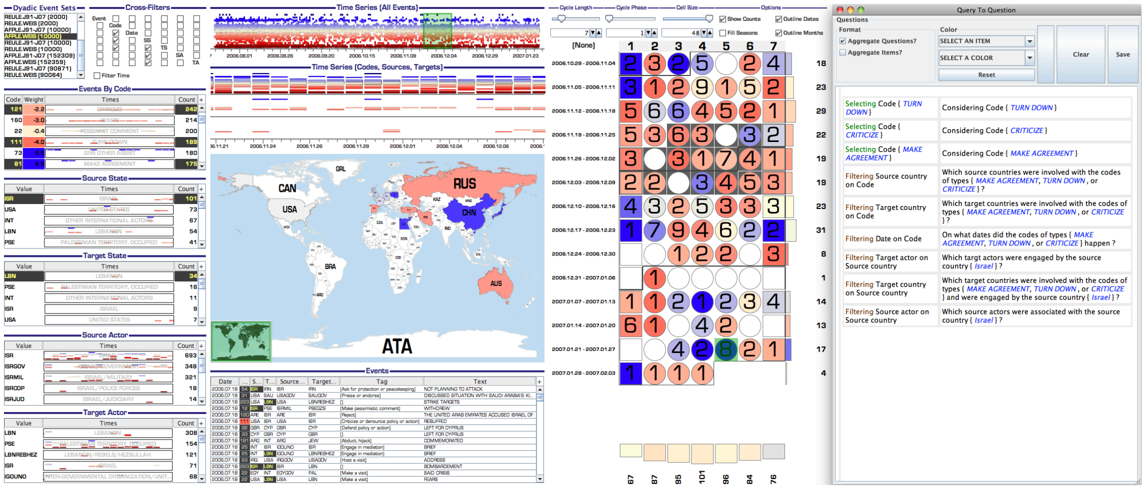


Figure 5.5: Visualization of events in newswire stories (left), accompanied by the corresponding Q2Q interface (right).

of *Source Country* on *Events By Code*, two possible questions are "Which source countries were involved with codes of types MAKE AGREEMENT, TURN DOWN, or CRITICIZE?" and "Which source states are involved with codes of types 81, 111, or 121?" The former question expands event types, shown as numerical codes in the latter, to descriptive phrases. The choice of domain information representation is adjusted in the content planning stage of online generation and reflected in the output questions, as shown in the example above (see Figure 5.6, row four).

Similarly, relationships between data dimensions often can be expressed more abstractly by using general-purpose, abstract phrasing such as "involved with". Wording that explicitly defines relationships between dimensions, such as the translation of filtering of *Resolution* on *Passenger* in Figure 5.1, can be customized (in the offline stage) to cover users of all knowledge levels.

Technical knowledge refers to theoretical and computational skills in the field of visualization. Technical knowledge can be exploited using more formal and/or precise words and phrasings to describe the structure and behavior of interactions. However, doing so can decrease the naturalness of text and its accessibility to users unfamiliar with visualization concepts. Some interactions may make more sense if

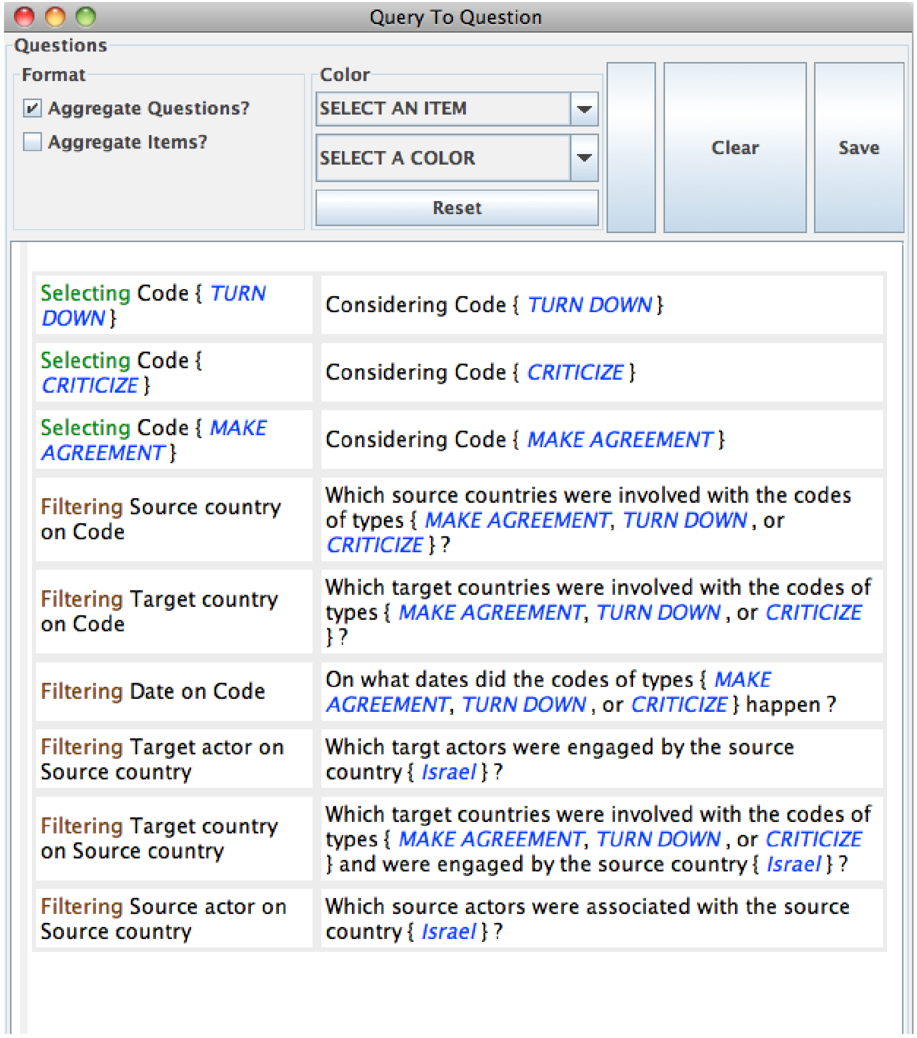


Figure 5.6: The Q2Q interface that accompanies the visualization of events in newswire stories shown in Figure 5.5.

described in technical words. For example, when a user applies a jitter interaction in Spotfire [82], two (of many) possible translations are "apply jitter" and "shift each displayed item randomly by a small spatial increment." The former is more expressive and straightforward for visualization designers, whereas the latter is likely to be more understandable to non-visualization experts. Q2Q provides both technical words for interactions (Figure 5.5, left column) and also presents meanings of their queries as natural language questions (right column).

Linguistic knowledge refers to the abilities enabling speakers or writers of a language to communicate with each other. Linguistic knowledge varies from user to user and suggests how much sophistication they will accept. A visualization tool for educational purposes might have users of many ages. The text generator should be able to provide simple text with short sentences and fewer referring expressions for younger students, and coherent, more natural text for older students. The same condition applies to non-native speakers and people at low reading level. Long, aggregated sentences with rich vocabulary and compound phrasing are less likely to be readable by these groups. Q2Q offers two levels of abstraction, in the form of two options for aggregating within and across generated questions, to support different linguistic preferences. For example, Figure 5.5, eighth row, the queries containing more than one question are aggregated and in Figure 5.7, sets of more than three items are aggregated within questions.

The customized translations in Q2Q can also serve the needs of users in their various role(s). User roles are described in the next section.

5.2.1.2 User Roles

In the context of information visualization, users have three main roles: *developer*, *operator*, and/or *consumer*.

Developers are visualization system developers and tool designers. System developers implement algorithms, languages, and architectures, and write system libraries. Interaction translation provides a way to analyze the query semantics and expressiveness of new interaction approaches. Designers build tools using existing interaction techniques and libraries. They can use translation to choose interaction approaches.

Both system developers and tool designers can use translations to evaluate applications of interaction, such as to determine which visualization parameters users manipulate frequently or rarely. Harder to understand interactions are also likely to be more difficult to translate. Translations generated for the Migrant Boats visualization hint that the cross-filtering technique might be less usable when not all the data dimensions involve in the filtering have total participation in the relationship. For instance, *Ship* and *Resolution* in Figure 5.1 are partially associated with one another; if a vessel landed, it means that it did not get interdicted. Thus the question "Which ship interdicted a vessel with resolution of type *Landing*?" does not make sense. This suggests a modification to the design of the cross-filtering technique in the visualization, such as to deactivate the checkbox leading to this question.

Operators are people who browse, interact with, and otherwise run the visualization. They perform interactions and are the ones who most directly consume the results of the translation, interleaved with interaction itself. They also organize translations for dissemination to other operators and consumers.

The benefits of translation differ for experienced users and trainees. Experienced users perform tasks such as overview, zoom, filter, and extract [28] to gather information and acquire insight. Text can help them remember their past actions, plan future actions, understand details of interactions, and collaborate with other analysts in different locations. Trainees can use text to help learn the capabilities of a visualization and its interaction techniques, so that they can perform the same tasks as analysts in the future. For instance, Figure 5.2A-C and 5.3D-F show how Q2Q helps trainees perform tasks when they are not acquainted with the domain or the visualization. Designers can specify evocative details, tailored to particular operators, in the offline stage. In Figure 5.2A, the phrase "a vessel carrying" clarifies what "interdiction" means in translations of queries about *Passengers* and *Ships*. Corresponding phrasing in Figures 5.2B-C and 5.3D-F serves the same purpose for other combinations of dimensions. Additional text generated to highlight potentially misleading interaction responses (e.g., the orange text in Figure 5.2C) and overt punctuation around enumerated data values in question translations (e.g., the curly brackets around blue text in Figure 5.3D-F) help operators verify that their filtering and selection actions are triggering intended, correct queries.

Consumers are the people who use the results of visualization without necessarily being involved directly in live interaction. For instance, stakeholders who perform sensemaking on the results of analysts' foraging are consumers. They may peruse the question log from alternate perspectives, assess precise questions that have been asked, and interpret visualization snapshots in terms of questions (and vice versa). To them, textual output is a representation of fine-grained analysis activities. They use the text to reason about foraging and understand what happened, and how, in the broader analysis process, and integrate translations for use in results dissemination and reporting. Offline generation lets designers capture nuanced yet consistent domain relationships for easier reading and comparison, such as time in the past tense and spatial context for *Ships-Passengers* (Figure 5.2B, "did interdict a vessel carrying"), *Passengers-Ships* (Figure 5.3D, "were on a vessel interdicted by"), and *Encounter-Passengers* (Figure 5.3F, "Where were the vessels carrying...").

NLG can employ user modeling to improve the understandability and effectiveness of generated text [83]. User models can be based on either customized rule sets [84] or training corpora of examples for particular groups of users [85,86]. Given the goal of broad application of NLG for translation of interactions in a variety of visualization tools, translation designers must characterize the interaction activities of a broad spectrum of users, yet customize the text generation process based on specific analysis needs. They can delegate customization to visualization designers, who are likely to be more familiar with user needs. The system may also provide users with a variety of text representations to choose ones most fitting their needs.

Q2Q incorporates user modeling by following a two stage offline-online approach. This maintains the generalizability of the translation system while supporting the flexible text representation needs of different users. Offline generation lets the visualization designer (or domain expert) not only express the relationships between data dimensions, but also decide on the representation of relationships as text. It also lets them adjust relation sentences to output desired questions for targeted users. This diminishes the need for constructing a comprehensive user model prior to generation.

Currently, the output from Q2Q is static HTML, using fixed CSS-like markup to embed data dimension and value references and format them for display in generated text. Chapter 6 describes an integration of an interactive web interface into the translation system for presentation, storytelling, and report generation, with search on embedded references, general text matching, and reordering and grouping options.

5.2.2 Data

Translation of data into written natural language depends on the characteristics of information conveyed. Translation in Q2Q takes into account the *dimensional type*, *informational complexity*, *dimensional cardinality*, and *linguistic ordinality* of the visualized data values and parameters of interaction. In this dissertation, the focus is on how specific aspects of data influenced the design of the current Q2Q system for initial application to cross-filtering visualizations. A comprehensive treatment of these concepts to information visualizations in general is left as a large body of future research.

Dimensional type is the classification of data values into well-defined categories, whether syntactically (integers, strings), statistically (nominal/categorical, ordinal, ratio, interval), semantically (name, date, watershed region), or structurally (tree, graph). In text, information about dimensional type can be conveyed lexically (as a word, token, or self-contained phrase) or grammatically (spread out across one or more sentences). For instance, length, area, or volume can be presented lexically using various forms of numbers, or a residential location can be described grammatically using its address representation. Translation should convey syntactic and semantic information about data types as well as the data values themselves. Data types may be implicit in translated data values, or can be translated explicitly. Both data values and data types can also be conveyed via formatting, markup, or embedded graphics. For instance, including an image of a geometrical shape in the text provides information about the type of the data and possibly its value (as approximate area). The challenge here is to reduce the structure of data values into the primarily linear structure of text. Exploiting common conventions of text structure, such as parentheticals, may allow more data structure to be conveyed. More exotic ways of structuring text, such as indentation used in poetry, may provide additional meaning to convey information about data type. To support the various user roles, however, the goal is to generate recognizably natural language, rather than descriptive code.

Informational complexity is the amount of information carried by data. It is generally straightforward to translate "atomic" data values, including spatial, temporal, and N-dimensional data points. General translation of structure involving composition, association, topology, and geometry is a deep and potentially open-ended research challenge.

Translation of visualization interactions focuses on the information complexity of data structures that are actually represented and manipulated in a visualization, for instance the polygon shape of a lasso or the correspondences between data points duplicated in brushed views. Generated translations can enumerate data structure details exhaustively, potentially allowing users to scan for patterns in sentences directly (see the last row of the Q2Q interface in Figure 5.1). However, it is a challenge to stay within reasonable limits of sentence length for sake of understandability and readability. For instance, if the number of *Encounters* listed in the question "Which Ships did interdict vessels during the Encounter *494, 569, 579, or, 580*?" increases, not only would the question not sound natural, but also making sense of the list and discovering patterns in it could become more challenging.

Generated sentences may alternatively summarize data structures, reducing the length and complexity of text while preserving the abstract character needed for user scanning. For example, unbound lexical aggregation (described in Chapter 3.3.5) can be used to aggregate selected *Encounters* (Figure 5.1, last row of the Q2Q interface) to describe them as "boats in the highlighted circle off the west coast".

Dimensional cardinality is the number of data dimensions involved in visual queries. Queries in coordinated multiple views can arbitrarily combine locally encoded and remotely filtered dimensions. Flexible query dimensionality is a hallmark of recent tools [4,15]. Readability and comprehension of compound phrasing in sentences generally limits the number of dimensions that can be effectively translated. Moreover, the semi-linear nature of text means that one must chain small subsets of dimensions in the clause and phrase structure of sentences, or in the progression of a set of sentences. Q2Q generates alternative sequences of question sentences with a variety of aggregated clauses, and lets the end-user choose from them.

Linguistic ordinality is the expected ordering of both dimensions and data values in text. In English, time phrases typically follow space phrases. Names typically appear alphabetically and dates from past to future. Such "natural" ordering of data values can inform effective ordering in text. However, many data types, such as geographical coordinates, have no single natural ordering. Translation can rely on designer specification of contextual ordering conventions—such as on the order of item selection, interaction, or domain specific criteria—to avoid arbitrary textual ordering of data values.

Q2Q focuses on data values that are—or like calendar dates and geographic locations can be treated as—nominal or categorical, since the cross-filtering technique targets these types. Nominal or categorical data values are low complexity, one-dimensional, and are often inherently lexical, making them straightforward to translate. Dates are converted into a standard text format and geographic locations can be "nominalized" as *(latitude, longitude)* or as a place identifier. For example,

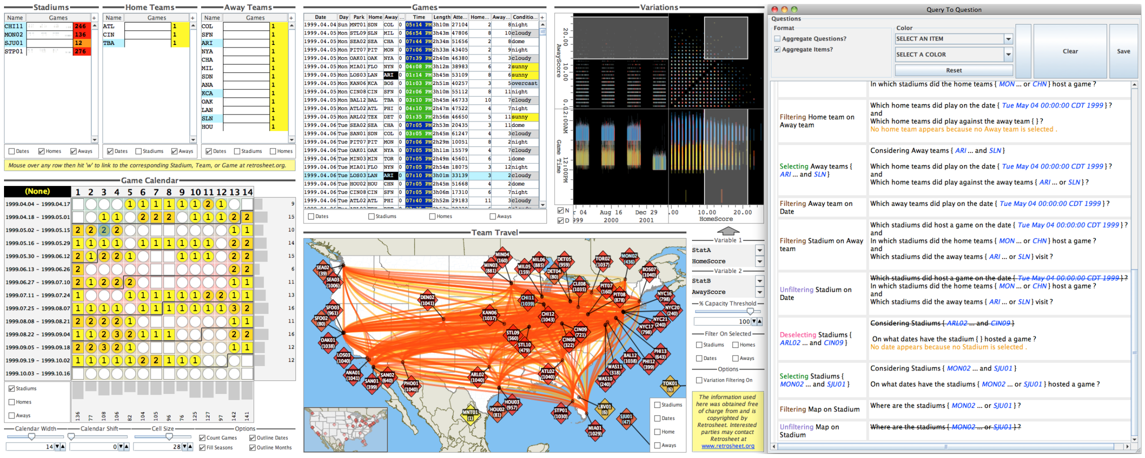


Figure 5.7: Visualization of baseball games in the Retrosheet database (left), accompanied by a Q2Q interface (right).

in the last two rows in the Q2Q interface in Figure 5.1, encounters are named by numeric identifier rather than coordinate. Identifiers are easy to aggregate in sentences (e.g., "494, . . . , or 580") but are hard to spot in most views.

The dimensional cardinality of cross-filtering queries can vary. To assist comprehension of how complex questions correspond to multidimensional queries, a non-aggregation option translates a query with $n + 1$ dimensions into n independent questions. The fourth full row in the Q2Q interface in Figure 5.7 shows an example of a four-dimensional query translated into a set of three non-aggregated questions.

Q2Q follows convention for linguistic ordinality: names of people are alphabetical; dates are past-to-future; one-dimensional quantities are in numerically sorted order. Conventional orderings work in favor of aggregation by allowing understandable presentation of long lists of data values through elision. For instance, several of the rows in the Q2Q interface in Figure 5.7 use an ellipsis to elide long lists of teams and stadiums.

5.2.3 Interaction

This section looks at translation of interaction in terms of two interrelated dichotomies: *selection versus navigation* and *discrete versus continuous*.

Selection lets users keep track of data items of interest by marking them, generally as a preceding action to subsequent operations to query on the selected set [33]. *Navigation* is a process of exploring different areas of data [28], such as zooming a map. In general, selection manipulates data objects and navigation manipulates visualized space. Queries involving selection and navigation can be described in terms of data values and visualization parameters, respectively. Selection entails one bit of information per data item, with each item associated with a label value. (In cross-filtering, the label is the group-by key value of each item in the corresponding data dimension.) Navigation typically sets a 1-D or 2-D point (by mouseover or click), a rectangular region (by rubberband), or a polygonal region (by lasso). Linguistically, navigation is also associated with meaningful spatial references, typically a location or a region in a domain-specific coordinate system.

For cross-filtering, Q2Q reflects selection by enumerating and highlighting the types of interaction (green text for Selection and brown for Filtering in Figure 5.5) and the data values involved (blue text within the ellipsis). It also linguistically identifies the selected dimensions and connects them to selected items. For instance, "Filtering source country on code" identifies two selected dimensions "country" and "code" and "...source country {Israel}..." identifies Israel as one of the selected countries.

Navigation interactions are implied by the progression of discrete questions rather than directly reflected in individual questions. In general, navigation is hard to describe within questions directly; it often involves parameters with higher informational complexity and dimensional cardinality, and lower linguistic ordinality, than the data values involved in selection. For instance, describing the details of a lassoed region as text would be verbose. Although precise, a list of polygon coordinates is likely to be hard for a reader to interpret as a meaningful nugget of information, either in relation to visualized spaces or to steps in analytical reasoning.

Selection and navigation can be either *discrete* or *continuous*. We define *Continuous* as interactions involving a sequence of intermediate visualization states between an initial state and a goal state. Selection by lassoing and navigation by panning (both through mouse drags) are examples. Examples of *discrete* interactions include brushing a single item, clicking a checkbox to toggle filtering, and pressing the right arrow key to advance a time series.

Intermediate states make effective translation of continuous interactions much more difficult than for discrete interactions. Rapid accumulation and modification of text for every change would be hard to read. Sampling or "punctuating" the translation can preserve sufficient differences between key interactions while reducing the total amount of generated output. Punctuated representation of translations of key changes as text would require knowledge (or specification) of the analytic relevance of intermediate states, which is highly unpredictable in general. Consider a continuous selection by lasso of encounters in the map in Figure 5.1, ending when the user releases the mouse. Q2Q generates two questions, one to reflect the selection state when lassoing starts and one to reflect when it ends. Both questions list selected encounters—optionally aggregated with an ellipsis when lassoing in a dense region makes the list long—rather than, for instance, describe the geometry of the lasso itself. Reflecting the selection state at intermediate points during lassoing can also be a valuable indication of analysis steps. However, translation might require knowledge about the end point, which would not be yet known.

For Q2Q of cross-filtering, the natural granularity of the analysis actions supported by a visualization's interactions is reflected largely one-for-one in the progression of interactions and questions recorded and displayed. Ways to generate readable summaries of navigation and continuous selection interactions, including aggregation across and abstraction within questions, are left to future exploration.

5.3 Summary

This chapter presents Q2Q as a means to support a cross-examination process in which questions rather than interactions are the focus of analytical reasoning and action. The chapter describes four particular situations in which users have difficulty interpreting visualization states. Then, through a set of examples, it shows how Q2Q can benefit users by helping them to validate their queries and correctly interpret visualization states. In addition, the chapter presents a set of design factors identified throughout the design, implementation, and evaluation of the Q2Q translation system. These design factors—user knowledge and roles, user interaction, and types of data—introduce a set of challenges and open ended issues which were discussed in this chapter.

The content of this chapter is accepted and will be published in *IEEE Transaction on Visualization and Computer Graphics Journal* [87].

Chapter 6

History Organizer Tool

6.1 Storytelling

6.1.1 Introduction

The primary focus of most visual analytic research has been on providing a platform for users to perform data analysis and exploration. Less attention has been paid to presentation and communication of the steps users take during analytical sessions. The analytical process is particularly important to communicate since people who analyze the data are often not the people who make decisions. The insight gained from visualizations and the reasoning leading to those insights need to be communicated and presented to decision makers to support well-thought-out decisions.

In the iterative sensemaking process of intelligence analysis [1], visualization users start with selection and navigation to filter and search data sets, and bring subsets of information to their attention. They use visualizations to view their data, explore it, and form hypotheses. They then present a hypothesis, reasoning, and possible conclusions to an audience. The feedback from the audience might suggest more analysis of the data and iteration of the process.

Presentation of a process in the form of a story is a popular way to engage an audience and communicate insights [88,89]. There is a need for presentation tools for construction of stories that communicate information achieved during data analysis, and the steps taken to get to that information.

In this dissertation, we developed a storytelling tool that enables users to take the questions they have asked by interacting with a visualization, and interactively organize them into a coherent and meaningful representation. The questions generated by Q2Q suggest certain orderings and groupings. The tool uses these dependencies between questions to assist users in making understandable and reasonable organization choices. The tool also provides an environment to conduct studies of user storytelling behavior, which are presented in Chapter 7. The rest of this chapter provides background and related work about storytelling. It then describes a motivation scenario to further define the problem space. Finally, it presents the history organizer tool (HOT) and the algorithms used by the tool.

6.1.2 Background

This document defines storytelling as the presentation of a series of events in a certain order with clear relationships between them [88, 89]. Stories generally start with an introduction, followed by main content, and ending with a conclusion. Storytelling presentation can use various visual elements such as text, graphics image, and videos. Regardless of their content, stories can be told in wide variety of forms. Segel and Heer [88] define seven styles of storytelling: *magazine style*, *annotated chart*, *partitioned poster*, *flow chart*, *comic strip*, *slide show*, and *animation*. These styles are different in terms of the visual elements they use, the number of frames they contain (for instance, multiple visual elements showing different periods of time), and the order of the visual elements. The top level structure and order of the visual elements forms the core of the story. Elements can appear in chronological order or in a non-linear order, such as to include flashbacks in time. Events in a story might not even be tied to particular time, and instead appear as a function of their importance or interestingness.

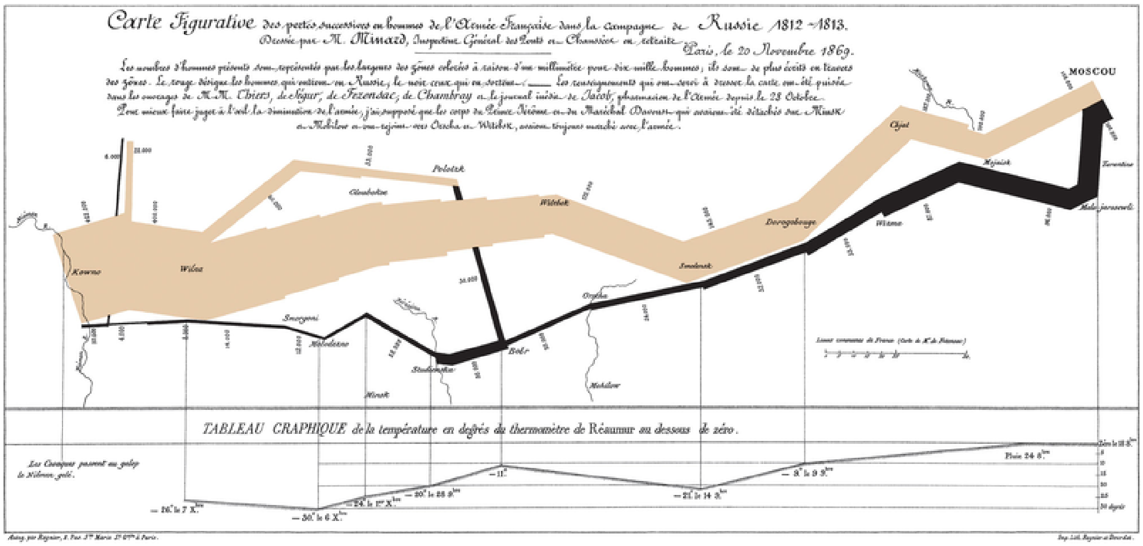


Figure 6.1: Minard's visualization of Napolean's march on Moscow, 1812–1813

Static data visualizations are often used to provide supporting evidence or additional details to a story that is conveyed primarily through text, particularly in the magazine style [88]. However, static visualizations have also been used as a standalone storytellers. For instance, Charles Minard's visualization of Napolean's 1812–1813 Russian campaign is an example of a single, coherent graphic that describes an event over time. This visualization, shown in Figure 6.1, illustrates the tremendous losses of Napolean's army, along with other factors such as geographical locations, temperatures, and travel directions over time [20].

Increasingly, online newspapers such as the New York Times, the Washington Post, the Economist, and Guardian accompany the textual description of events with interactive visualizations that provide an engaging experience for the audience. Interaction with a visualization also invites the audience to be part of the storytelling process and construct a story of their own. In visual analytics, storytelling components are starting to be incorporated into visualization systems. For instance, GeoTime [7] has adapted elements of storytelling by allowing users to manually annotate and bookmark key states of a visualization. In Tableau [3], users can review a history of visualization states and organize selected ones for later use. Online visualization applications, such as Many Eyes, have also been used to tell stories with visualizations in collaborative settings [90]. Heer, Viegas, and Wattenberg designed a web-based system called *sense.us* for asynchronous collaboration of information visualization [8]. Sense.us allows users to bookmark, annotate, and share views. These elements let the user construct trails of visualization states for the purpose of storytelling. DecisionSite Posters[1] allows distribution of interactive visualizations and viewing of a selected set of interesting states for further analysis. Collaborative discussion of analysis is also possible using CommentSpace [91]. In CommentSpace, the visualizations are overlaid with tagging and commenting features that can be linked, filtered, and searched to construct a view of the analytical process and facilitate collaboration.

Segel and Heer [88] further studied various visualizations and their capabilities for storytelling. They identified three common approaches in telling stories in the visual analytic sense: *Martini Glass*, in which the story starts with a broad introduction, then focuses on key points, and finishes with the big picture; *Interactive Slideshow*, in which users explore the domain in sequential form, focusing on a few aspects one at the time; and *Drill-Down Story*, in which users are presented with general information and can drill down into aspects that are interesting to them. These three approaches are categorized based on the balance between author-driven narrative and reader-driven narrative. Hullman and Diakopoulos [92] analyzed different rhetorical strategies used in narrative visualizations, including selective information representation, explicit notation of data source and uncertainty, usage of visual metaphor and intended obscuration of data values, intentional presentation of contradicting of redundant information, usage of visual encoding for emphasis, and enforced particular ordering.

[1] http://spotfire.tibco.com

Hullman, et al. [93] surveyed sequences of visualization states as they appeared in linear presentations. They considered various types of transition between visualization in sequence: *Dialogue* transitions, in which the state answering an analytical question appears after the state reflecting the question; *Temporal* transitions, in which visualizations are arranged based on a time data attribute of the underlying data set; *Causal* transitions, in which a sequence of visualizations reflects causal relationships, with each the consequence of previous; *Granularity* transitions, in which visualization states appear in order of general to specific or specific to general, in terms of the level of information they reveal; and *Comparison* transitions, in which visualization states are placed to compare values of a data attribute with respect to changes in other attributes. The study showed that users' preferences of visualization order are: temporal, comparative, and finally on granularity. They did not include causal and dialog transitions in their study. They also found that users prefer a high level of consistency between consecutive visualizations over low consistency. They defined consistency as the number of changes in data dimensions between two visualization states.

Even though studies in visualization storytelling are recently emerging, there is a large space of research to be explored on various approaches toward creating, organizing, reordering, and sequencing snippets and pieces of stories as a part of visual analysis. In this dissertation, the focus is on providing a platform for users to organize and present their general data exploration and analysis process, specifically for the questions they have asked by interacting with data visualization tools. The study conducted by Hullman, et al. [93] focuses on sequencing of visualization states. The research in this dissertation takes advantage of their approach to study the sequencing of questions asked in visualizations through interaction sequences. The goal is not only to provide an effective presentation tool, but also to gain a better understanding of the strategies that people use to organize the questions asked in a visualization throughout an analysis session.

6.2 Motivation Scenario

Users perform analytical tasks to form hypotheses and find solutions to analytical problems. To do this, they can ask various questions of data using visualizations. The questions they can ask are limited by the interaction and representation techniques provided by the visualization tool. However, these questions are not always asked in a deliberate fashion. Users often branch off to explore various dimensions and attributes. They also often come back to previously visited points and regions in the visualization space. Thus, simply providing questions in the order that they have been asked might not form a comprehensible story of their exploration process. Such a list would most probably include a set of states that are important and interesting, as well as a set of states that are less useful and or even distracting in a story of analysis. Even if all states are worth including in a story, the order of exploration is not necessarily the desired presentation order. For instance, users might prefer to present their conclusion first, and then go into detail of how they reached them. A tool that provides questions to users, and incorporates capabilities to search, filter, group, and reorder those questions, can facilitate the process of designing an easy-to-follow story of the analysis process. Such a tool can also provide a platform to identify user strategies for organizing, reordering, and grouping questions for later reference or presentation. Insights about users' storytelling habits can be reflected in the design of presentation tools to improve their storytelling capabilities.

6.3 History Organizer Tool (HOT)

In this dissertation, we present the design and implementation of a history organizer tool (HOT). HOT inputs automatically generated questions from Q2Q and provides features to search them on keywords, reorder them based on various ordering options, filter them based on their internal relations, and group them based on relevancy. The HOT web application is designed in a way that can be used after each analysis session (in a visualization + Q2Q setting). Thus, it enables users to import questions generated by Q2Q and edit, share, and present them.

HOT provides two ordering options, *temporal* and *causal*, which impose a set of ordering constraints when users are reorganizing the questions. A *relationship suggestion* option is also provided by HOT for effective search of related questions to a given question. HOT can also be set to Group mode to apply grouping constraints to the questions. These features are further explained in this chapter.

6.3.1 Architecture

The overall architecture of HOT and its relation with Q2Q is shown in Figure 6.2. During an analytical session, user interacts with a visualization. The interactions are translated into questions and shown in Q2Q interface. The structured questions are also passed into a graph formation module for constructing a graph of questions and their relationships for usage in HOT. When the user exists the visualization, the relationships between the questions calculated based on the graph representation are passed to the HOT web application which follows the general model view controller architecture.

The raw, internal representation of HOT's data (the model) is separated from the user through a presentation layer (the view). The user interacts with the view to organize questions asked in the visualization. This sends events to the controller for manipulating or getting data from the model. The controller looks up the ordering, grouping, and relationship constraints and imposers, which are pre-calculated from the graph and are stored in the model, to update the view. This architecture is common among web applications. The primary benefit of this architecture is that the model, view, and controller are modular and reusable. The web-based design of HOT has the advantage of being platform-independent. It can be run on any machine with a modern browser, meaning its content can easily be shared for presentation purposes.

The model is constructed after a preprocessing stage which transforms the questions generated by Q2Q into a graph representation. The graph representation of the questions informs the ordering and grouping procedures that can be applied during user question reorganization. The preprocessing performed to calculate orderings, relationship connections, and groupings are described in Sections 6.3.3.1, 6.3.3.2, and 6.3.4.

The view presents information from the model, and gives users a platform for interacting with, rearranging, organizing, and preparing their previously asked queries for presentation. The interface of HOT is described in detail in Section 6.3.2.

The controller is responsible for handling the user's interactions in the view, and responding appropriately, either enforcing ordering on the model, filtering, or highlighting questions, based on the features chosen by the user. Section 6.3.3 describes how the controller responds when a violation of ordering occurs.

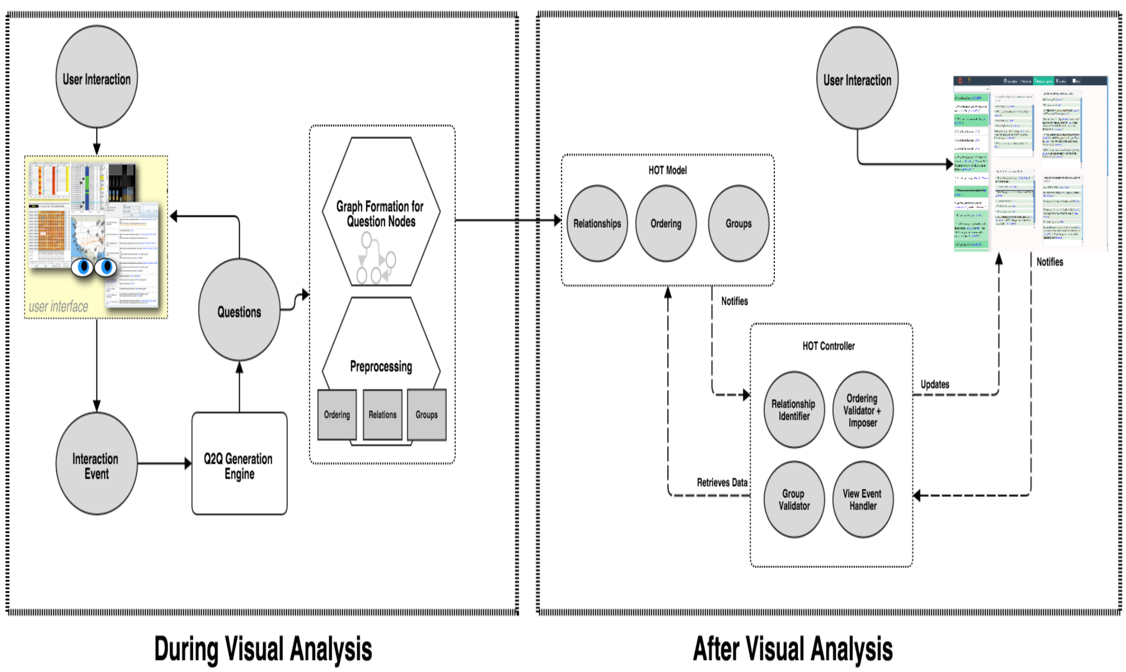


Figure 6.2: The overall architecture of HOT.

6.3.2 Interface

The user interface for HOT is implemented in HTML and JavaScript. Figure 6.3 shows the HOT interface being used to organize the questions asked in the Migrant Boats Visualization (Figure 1.1).

HOT presents the questions generated by Q2Q on the righthand side of the interface, in the same order as they were asked in the visualization. Users scroll a list to find questions of interest. They can also search questions on keywords using a search bar located at the top of the question view, thereby filtering the questions based on what is interesting to them. HOT uses filtering the questions identified by keyword search—over other means of showing related items, such as highlighting. The list of questions can be arbitrarily long. Filtering results by showing a subset of the questions reduces the amount of displayed data and is thus more effective for search and scanning tasks.

The questions are presented with a numeric identifier (ID), which is associated with the order in which the questions were generated by Q2Q, and thus also when in the analysis process they arose. IDs are given a fractional-part greater than zero. The fractional part means that the corresponding row (question/sentence) belongs to a larger set of questions/sentences generated as a result of a single interaction. (For instance, question 10.1 on the righthand side of the interface—shown in Figure 6.3. Identifiers 10.0 and 10.1 are generated to mark a short sequence of related selection interactions.)

Several key features are provided by HOT interface to support storytelling:

- Freely movable panels,
- Reordering options,
- A Relationship Suggestion option,
- Group mode, and

- Saving and loading question organizations.

To better organize the questions, users can add questions to *panels*. An add panel option is provided (see the plus sign on the top left corner of the interface). This feature allows users to build a hierarchy of panels and add/remove questions to/from those panels. One panel is active at a time. Users add questions to the active panel. Users can also title the panels to summarize their contents. Panels can move freely in a history organizer board to construct an infographic of the questions. Panels can also be deleted at anytime, which will result in removal of the questions within the panel but not deletion from the main list.

Questions in a panel can be reordered. HOT provides three ordering options:

- *Temporal Order* forces questions to follow the order that they were asked during visualization interaction.

- *Causal Order* makes sure that text generated in response to selection interactions comes before the corresponding questions that result from filtering interactions.

- *Free Order* allows users to reorder the questions without any constraints.

To facilitate the storytelling process, HOT provides a *relationship suggestion* option, which helps users to find all questions related to a given question, and add them to a panel. Users turn on the relationship suggestion option and click on a question in a panel to find related questions, which appear highlighted. Darker highlighting color represents stronger relationships. The strength of the relationships is based on the similarity of the dimensions involved in the questions (described in Section 6.3.4).

Users can enable a *Group* mode while they are adding questions. This mode restricts questions that can be added to a panel. If a question is unrelated to all the other questions in a panel, the application does not allow the question to be added to that panel.

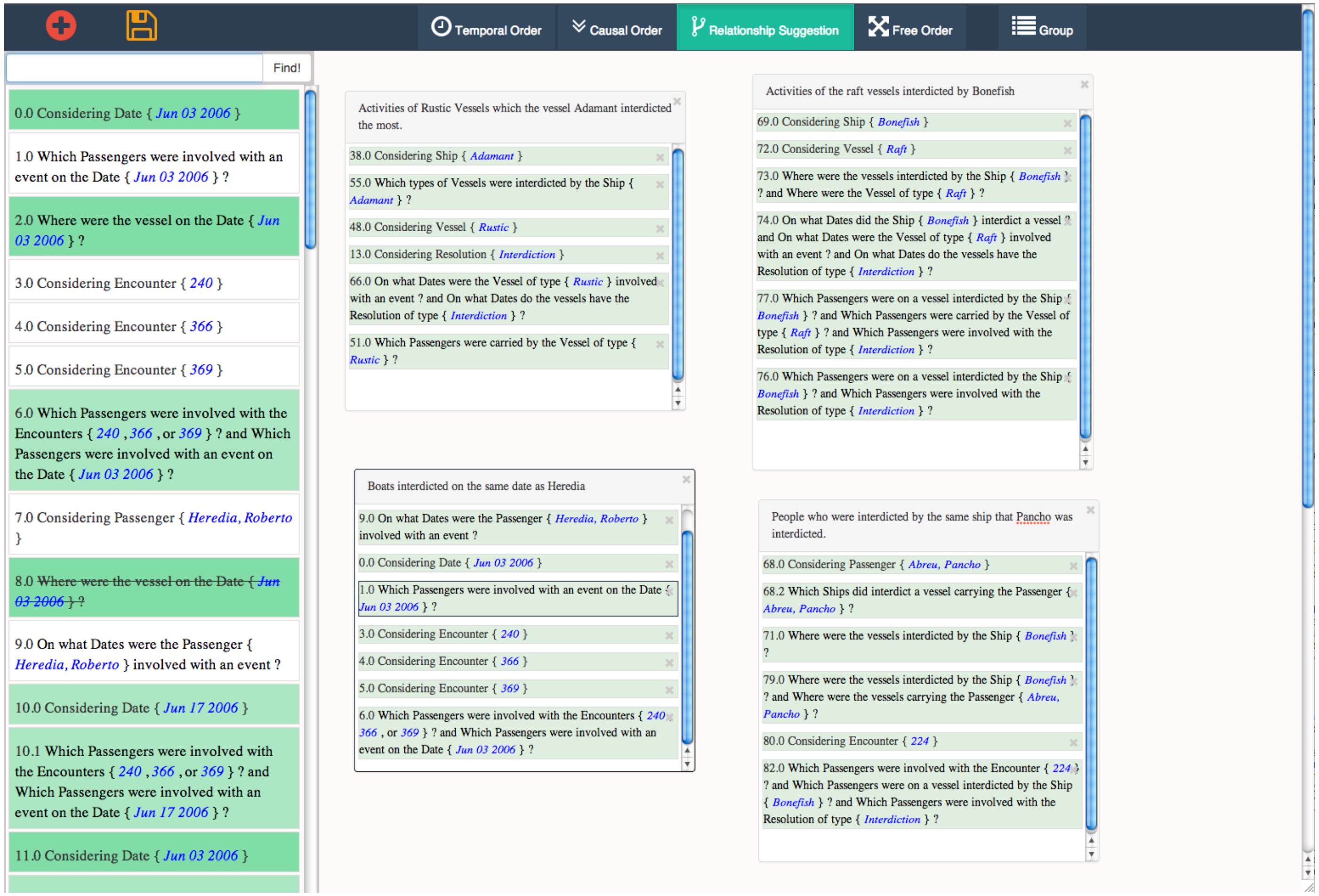


Figure 6.3: Intermediate state of the History Organizer Tool, showing a set of panels populated by questions. Relationship suggestion option is turned on, highlighting all questions related to the question that is selected in the bottom left panel (1.0).

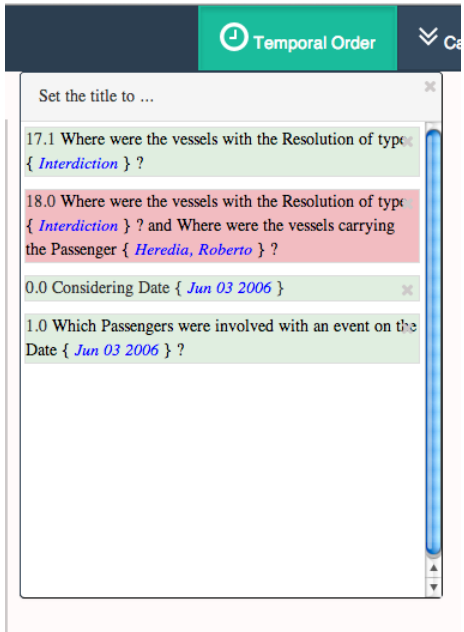


Figure 6.4: The question with identifier 18.0 is highlighted to indicate a violation of the imposed temporal ordering.

Users can also *save* and load the state of the HTML at any time for ongoing use. In the following sections, the algorithms behind the ordering and relationship suggestion options are described.

6.3.3 Reordering

HOT provides two imposed ordering options: *temporal* and *causal*. If either of these options is turned on, a set of restrictions is applied while the user reorders the questions. If users try to reorder against an imposed ordering, the HOT UI highlights the questions that are in violation. For example, Figure 6.4 shows a panel from the HOT interface. This panel contains questions about the Immigrant Boats Visualization shown in Chapter 5.1. The user has tried to move question 17.1 after question 18.0. Based on the temporal ordering, question 17.1 should always appear before 18.0. The system snaps 17.1 back to its original place and highlights question 18.1—the source of the "error"—in red.

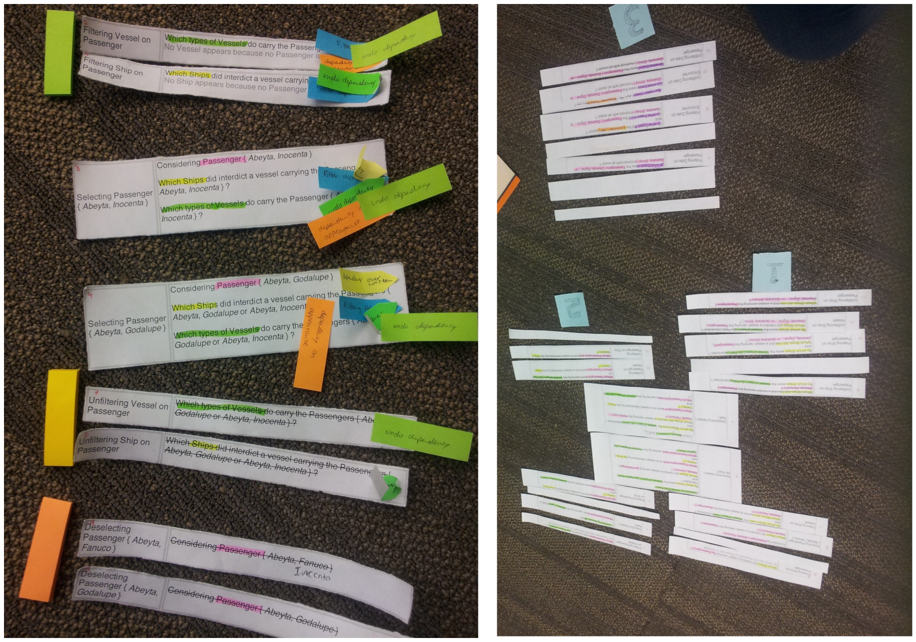


Figure 6.5: Ordering experiment, with printed interaction translations.

To acquire a better understanding of the process of constructing a coherent story of the data analysis sessions and particularly ordering options, we performed an experiment. In the experiment, we placed a set of printed interaction translations on a board. The interactions were generated using Improvise and translated into questions using Q2Q (see Figure 6.5). The questions were generated over several simulated analysis sessions, including exploration of various dimensions and data attributes. We reorganized those questions by moving them around, grouping them, ordering them, highlighting the data dimensions within each group, and labeling the groups on a free board. This required omitting unimportant questions and organizing the important ones into orders that were easy to follow.

The experiment suggested a graph-driven structure that identifies possible transitions in a question set, represented as nodes in a graph. A set of ordering options, which are connections between graph nodes, can be automatically applied while users are organizing the questions in the HOT interface. This results in a directed graph that specifies allowed presentation orderings of questions based on possible paths and directions from one node to another. The next few sections provide detail about the ordering algorithms.

6.3.3.1 Temporal Ordering

User interactions with visualizations are often exploratory. They might include various analysis branches and intermediate steps. A sequence of interactions reflects a set of steps taken, consciously or unconsciously, in an order that is desirable to preserve for presentation. Temporal ordering preserves the overall order of interactions as they occurred, while giving the users an ability to skip and not present particular interaction steps.

In temporal ordering, the focus is on *preserving the order followed by the primary analysts who used the visualization, but only for the questions asked about shared dependent variables*. To better understand the concept of temporal ordering, consider the following example. In Figure 6.4, both 17.1 and 18.0 are questions about the location of vessels, but with different levels of constraints. That is, 17.1 is looking for the vessels that were only interdicted, whereas 18.0 is looking for a vessel that both interdicted and was also carrying passenger Heredia. The IDs indicate that 17.1 was asked before 18.0 during data analysis. Temporal ordering preserves this order since both questions are about the location of vessels. Accordingly, the graph representing the temporal order (Figure 6.6) has a directed edge from node 17.1 to 18.0, reflecting the order they were originally asked in the visualization. On the other hand, in Figure 6.4, even though 1.0 occurred before both 17.1 and 18.0, it can be located after those questions since it examines a disjoint set of dependent variables (here is the variable, Passengers) and is conceptually independent from the other questions in the panel. Thus, there is no edge connecting 1.0 to 17.1 or 18.0 in the graph.

Algorithm The algorithm to apply temporal ordering to a set of interaction questions generated by Q2Q follows a graph-driven approach. As users interact with a

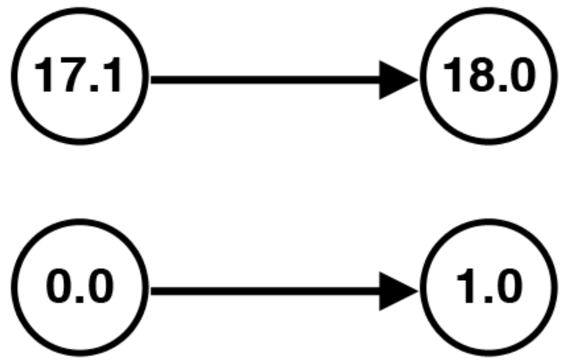


Figure 6.6: A graph of the temporal ordering dependencies of questions asked in the Migrant Boats visualization, as shown in the HOT panel in Figure 6.4.

visualization, the system constructs a dependency graph of the questions generated from queries.

The algorithm accepts translated queries as input and constructs a graph. The individual questions form the nodes of the graph. The dependency relationship between the nodes form the edges of the graph.

Figure 6.7 shows the flow of information through the algorithm and its components. Q2Q receives each interaction event as an input from the underlying visualization system. Q2Q outputs questions corresponding to the interaction. A single interaction might result in generation of more than one question/sentence. For instance, if a filtering or selection interaction results in changes in more than one view, a single question—which can be a combination of several questions concatenated with "and", or aggregated into a single question/sentence—is generated for each affected visualization view. Given a group of questions/sentences, the algorithm ignores any warning sentences generated during user visual interaction and constructs nodes for each remaining question/sentence. This approach allows users to freely organize the questions that result from individual interactions. The experiment performed using printed interactions on a board suggest associating nodes with individual questions rather than interactions. During the experiment, organizing questions as a group (associated with a single interaction) provided less flexibility than rearranging individual questions within a group.

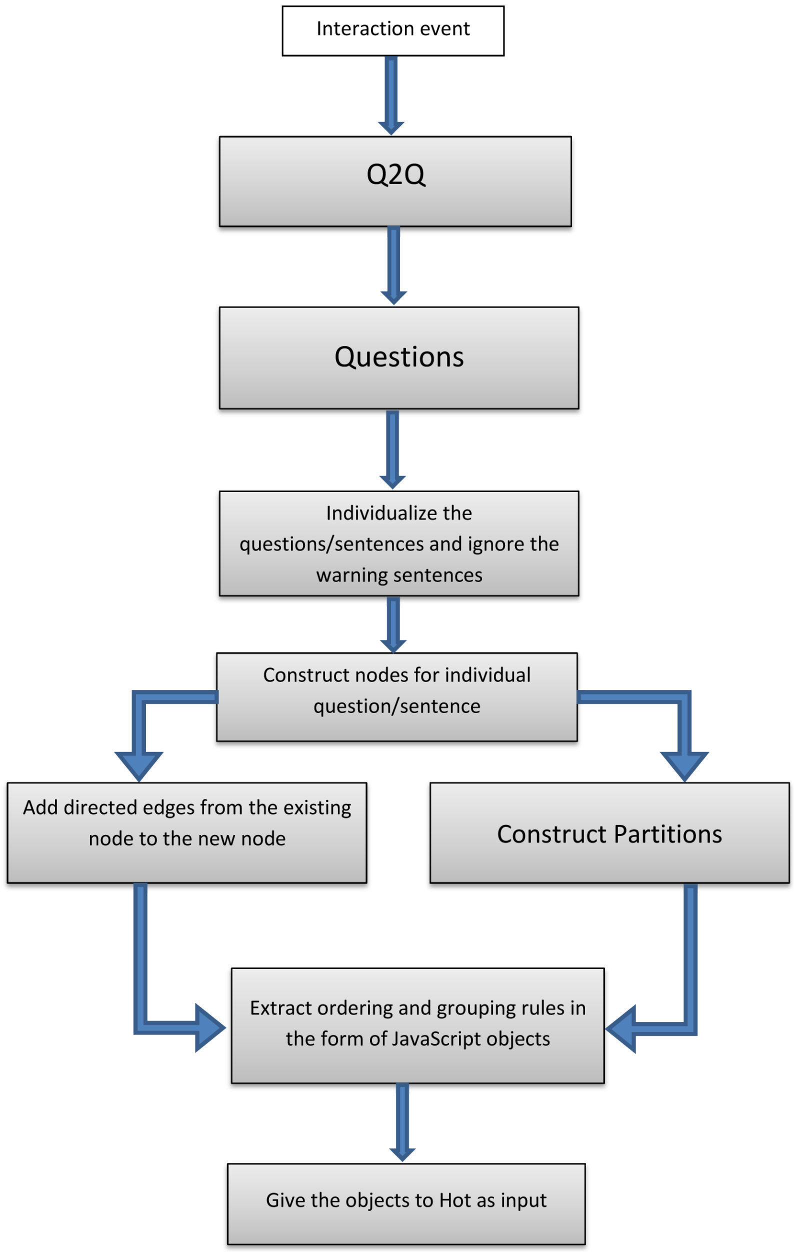


Figure 6.7: The flow of information through the components of the *temporal* ordering algorithm.

New nodes form as interactions occur. Each node in the dependency graph contains information about the ID of the interaction, interaction type, and the parse tree constructing the question/sentence. It also has detailed information about the interaction including the filteree dimension, the filterer dimensions, and the selected data values in each dimension. This information is extracted from the question/sentence parse tree and is used later to determine the edges of the graph.

An edge exists between two nodes if there is a dependency between the question/sentence attributes of the nodes. The direction of the edge defines the order they need to appear in a storytelling UI. If a new node (constructed by the most recent interaction) is determined to be connected to an existing node, an edge from the existing node to the new node is defined, preserving the interaction order. In temporal ordering, the questions are considered related if the subject of the question is the same, meaning they ask a question about the same data dimension.

To determine if two nodes are related, each node listens for certain interaction events related to the question the node is representing. Once a related event to the node occurs, the system creates an edge from the existing node to the new node. This way, the direction of the edge represents the order of a set of related questions/sentences as they actually occur in an analysis session.

In a question generated by a filtering or a selection interaction, the dependent variable is called the *filteree*, and the independent variable is called *filterer*. For instance, in the question "Which passengers were involved with an event on the date June 03, 2006?", the dependent variable is "passengers", the independent variable is "date", and the value of the independent variable is "June 03, 2006". Similarly, the data dimension "date" in a qualifier sentence "Considering Date June 03 2006" is the independent variable and "June 03 2006" is its value. Determining whether a new node is related to an existing node is based on the value of the filteree (dependent variable) and filterers (independent variable) in the questions/sentences being represented.

A node can contain a question, qualifier sentence, invalid question, or invalid qualifier sentence. For these four types of nodes, the following list describes which types of node relate to one another and which variables (dependent and independent) in any given pair of nodes should match to form a relation:

Question and Invalid Question nodes connect to

- a question node with the same dependent variable, and
- a qualifier sentence node that has the same independent variable.

Qualifier Sentence nodes connect to

- a question node with the same independent variable, and
- a qualifier sentence node that is invalid and has the same independent variable.

Invalid Qualifier Sentence nodes connect to

- a question node with the same independent variable, and
- a qualifier sentence node that is valid and has the same independent variable.

For grouping purposes, the algorithm calculates partitions as it adds new nodes to the graph. Three cases can occur while adding a new node to the graph:

(1) If there is no edge between the new node and the existing nodes, the new node constructs its own partition.

(2) If there is at least one edge between the new node and the existing nodes, and all the related existing nodes belong to the same partition, the new node joins that partition.

(3) If the related nodes belong to different partitions, adding the new node results in merging the existing partitions into one.

Once the graph is constructed, the dependency specification of each node is extracted by looking at incoming and outgoing edges from the node. The dependency specification is stored in the form of lists of nodes that comes before and after each node and the partition the node belongs to. This information is converted into a JavaScript object for applying ordering and grouping constraints in the HOT interface.

6.3.3.2 Causal Ordering

Causal ordering is motivated by the logic behind performing interactions. Some forms of interaction are naturally performed in certain orders. For instance, in selection and filtering interactions, the interactions are designed in such a way that it makes more sense to perform selection first, and then filtering. Thus, causal ordering causes selection interactions to appear before the corresponding filter interactions. More generally, causal ordering puts qualifier sentences before the questions affected by those qualifiers. This type of ordering gives users freedom to organize questions while adhering to constraints that facilitate construction of easy-to-understand stories of the analytical process.

Algorithm Similar to temporal ordering, causal ordering follows a graph-driven approach. The graph contains the same four types of nodes: question, invalid question, qualifier sentence, and invalid qualifier sentence. The question/sentence nodes are added to the graph as users interact with the visualization. However, the rules for connecting two nodes with an edge are different from those used for temporal ordering. Three main rules determine the existence of edges and their directions:

(1) There is an edge from a qualifier sentence to a question if both have the same independent variable and the values of the independent variable overlap (i.e., they share at least one value).

(2) There is an edge from a qualifier sentence to an invalid qualifier sentence if they both have the same independent variable and the values of the independent variable overlap.

(3) There is an edge from a question to an invalid question if they both have the same dependent and independent variables.

The evaluation of HOT (Chapter 7) revealed that presenting invalid qualifier sentences and invalid questions result in increased user confusion during organization. An updated version of the HOT interface does not display both invalid qualifier sentences and invalid questions in the list of interactions. This makes rules (2) and (3) above inapplicable in causal ordering.

6.3.4 Sequencing

Imposed orderings help user arrange questions to convey sequences of analysis steps. They also suggest related questions/sentences to ease the search and extraction of questions from a long list of possibilities.

HOT provides a relationship suggestion that allows users to see all the related questions/sentences to a selected question/sentence with the degree of their relatedness (described below). That is, HOT displays not only which questions/sentences are related, but also the strength of the relationship. For instance, in Figure 6.3, a user can select question 1.0 ("Which passengers were involved with an event on the Date June 03 2006?") in the panel in the bottom left corner of the board, causing all the related questions/sentences to be highlighted. HOT uses different shades of green to represent the strength of relationships. Darker colors represent stronger relationships. In this example, question 2.0 ("Where were the vessel on the Date June 03 2006?") is more related to the original question, 1.0, than to question 6.0 ("Which passenger were involved with the Encounters 240, 366, or 369? and which passengers were involved with an event on the Date June 03 2006?").

The identification of relationships between questions/sentences and calculation of their strength is inspired by the research of Hullman, et al. [93]. They performed a survey to discover individual preferences in sequencing visualization states for different state transitions and transition costs. They defined the cost of a transition as the number of mismatching attribute values of successive visualization states. They concluded that users prefer low cost transitions to high cost transitions when switching between visualization states. Even though their focus is on visualization states and not the textual translation of queries triggered in each state, a similar idea can be used to calculate the transition cost between two query translations. The algorithm for relationship suggestion in HOT calculates the cost transition between each node of the graph based on the number of mismatches in data attribute values that appear in questions/sentences. Then, it assigns edges if the cost of the transition is smaller than a specific threshold defined by the system. The constructed graph is an undirected weighted graph.

The transition cost between question/sentence nodes in filtering and selection interactions is calculated by looking at differences in dependent variables, independent variables, and the value of independent variables. After counting the non-matching attributes, the cost is normalized over the maximum total number of attributes that appear in the new node and existing node. The value of the cost transition is between 0.0 and 1.0, inclusive. A value 0.0 indicates that the two questions/sentences are exactly the same, while 1.0 indicates that they have no attributes in common. For example, in Figure 6.8, the questions related to question 78.1 ("Where were the vessels interdicted by the ship Bonefish?") are highlighted on the left hand side of the interface. Since question 71.0 is exactly the same question as 78.1, but occurred at a different time in the analytical process, the cost is 0.0 and consequently shown in dark green. Question 69.1 ("Which types of vessels were interdicted by the ship Bonefish?") has a strong relationship with 78.1, since it shares two attributes out of three—independent variable Ship and the value Bonefish—resulting in a cost of

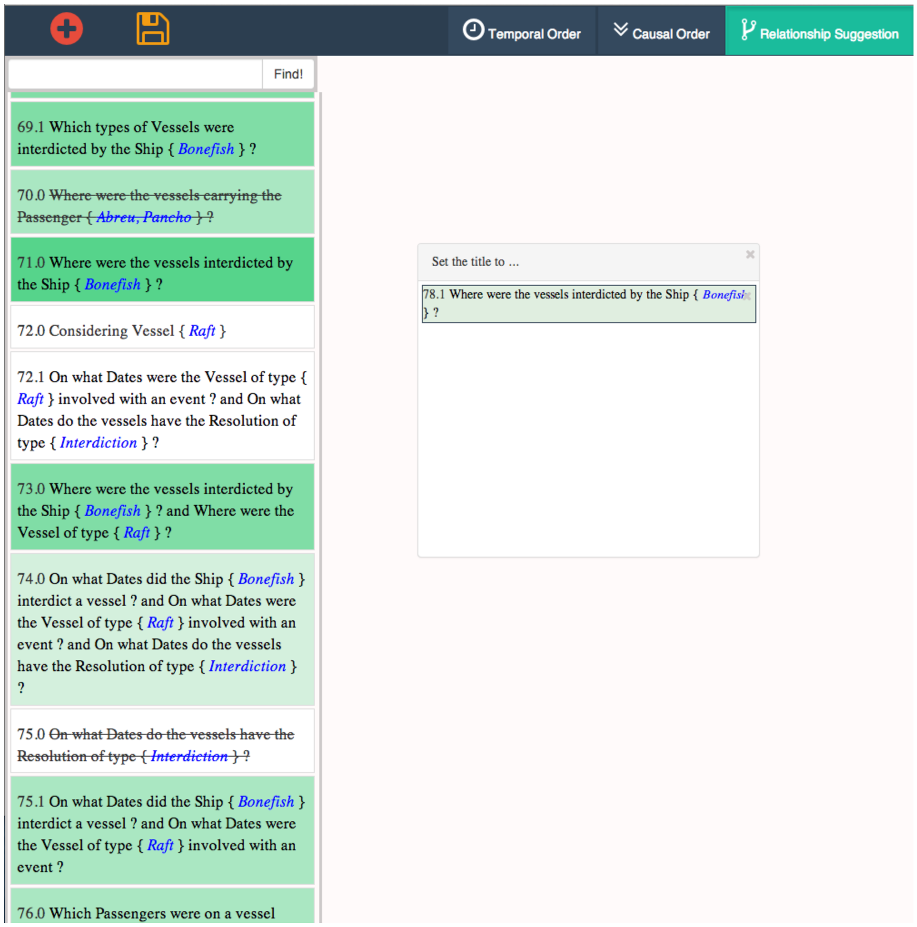


Figure 6.8: The relationship suggestion feature highlights related questions/sentences on how strongly they are realted using different shades of green. Here, the relationship suggestion option is activated. Selecting question 78.1 highlights all the other related questions in the list as shown.

0.33. On the other hand, question 74.0 ("On what dates did the ship Bonefish interdict a vessel? and on what dates were the vessel of type Raft involved with an event? and on what dates do the vessels have the resolution of type Interdiction?") is not highly related to question 78.1, since only two attributes out of nine attributes match, resulting in a cost of 0.78.

Questions/sentences that are not highlighted have a cost of 1.0, which means no matching attributes. The effectiveness of this approach decreases as the number of involved attributes increases. A larger number of attributes results in lower cost values during normalization, since the number of mismatches is divided by the total number of attributes. This problem can be disregarded in practice since the number of the data attributes in most visualizations is small (generally fewer than ten).

6.4 Summary

This chapter describes a history organizer tool (HOT) that is integrated with Q2Q user interaction translation system. It accepts questions generated by Q2Q and provides a user interface to organize them. Users are able to search, filter, use automatically ordering and grouping to organize the questions into panels. Rearrangement options such as temporal and causal reordering, relationship suggestion, and grouping, are provided to facilitate the reorganization of questions into a coherent presentation of the steps taken during analysis sessions. Such facilities for grouping and sequencing the questions asked during visualization interaction has not been studied in visual analytic research. Deeper study is needed to understand the possible benefits that automatic reordering and relationship configuration can provide to support storytelling. The next chapter describes a user study of facilities for organizing questions translated from visual queries. Focusing on the benefits and drawbacks of the features in the HOT user interface, it further analyzes observed behaviors of users while organizing translations of their queries to form stories.

Chapter 7

Evaluation of User History Organization

7.1 Introduction

During data analysis and sensemaking using highly interactive visualizations, the organization of analytical steps and presentation of the process is nearly as important as the data foraging and sensemaking. Arranging the steps taken to acquire knowledge is essential to convey reasoning and conclusions, to make sense of the process, to better present and receive feedback about the analysis strategies, and eventually accept or reject hypotheses. With tools like Q2Q that capture and translate user interactions in visualizations into natural language, analysts have access to the questions they have asked and can arrange them in a presentation format to describe the whole analytical process. The HOT interface is an initial design of a tool that enables users to flexibly rearrange, reorder, and group their query intentions to construct a story about the analysis process. HOT is also a suitable platform for studying users' rearranging and grouping behaviors since various users have different preferences in organizing the analytical process.

This chapter describes an evaluation of user behavior in selection and ordering of their translated queries from visualizations in general, and the effectiveness of the automatic ordering and selection features provided by HOT in particular.

7.2 User Experiment

The goal of the history organizer tool (HOT) is to ease the process of rearranging users' analytical thoughts captured in the form of their translated queries, so that users are able to present the questions asked from the visualization to others, or record them for themselves. This process can be done by removing redundant, irrelevant, and distractive questions, and by grouping and ordering the relevant questions together. In order to gain understanding of how successful HOT is at facilitating the organization of translated queries, we conducted an experiment to study and analyze differences in user performances when users use different features of HOT. Beyond the evaluation of the features provided by HOT, this study can inform the design of other ordering and rearranging options that can be incorporated into HOT in the future.

7.2.1 Design

Twenty undergraduate and graduate students (12 females and 8 males) were recruited from various different majors including Computer Science, Electrical Engineering, Civil Engineering, Human Resources, Petroleum Engineering, Mechanical Engineering, Physics, Mathematics, and Industrial Engineering. They all reported basic experience interacting with user interfaces and visualization of some sort.

Prior to the study, we showed an Improvise visualization to the participants to give them a sense of the types of interactions that can be performed, focusing on selection and filtering only, and how those interactions are translated into questions using Q2Q. Participants worked through an example task to learn the software's mechanics and terminology. This practice also familiarized the participants with the data domain encountered in the experiment. Then, we explained the history organizer tool and its capabilities. In particular, we explained how the tool accepts

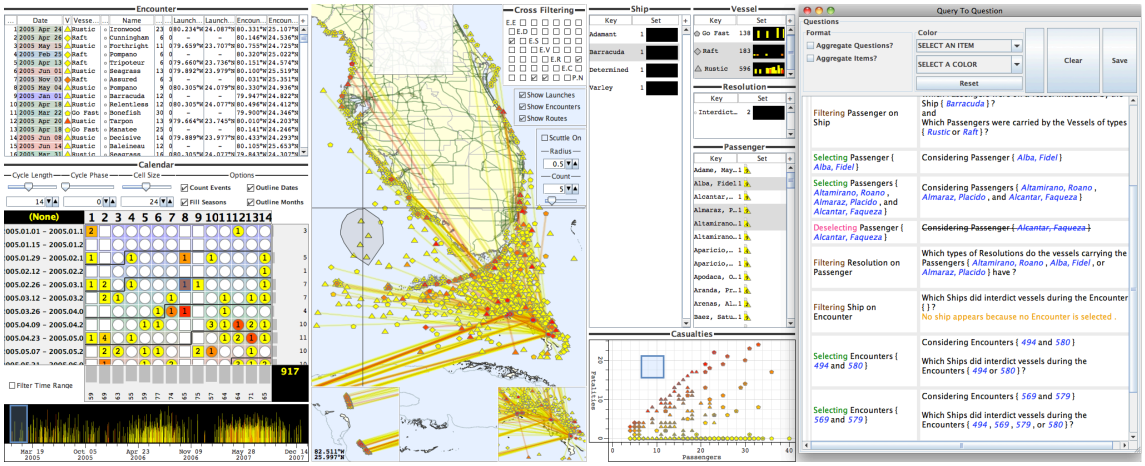


Figure 7.1: Migrant boats visualization accompanied with Q2Q and used for conducting the user study on HOT interface

translation of interactions as an input and how participants can use the provided features for organizing those interaction translations.

One Improvise visualization—Migrant Boats Visualization shown in Figure 7.1—was used to simulate an analytical session involving various queries on several data dimensions. The questions generated from simulated interactions with the visualization were displayed in HOT to conduct the experiment. We gave participants a set of questions. We asked them to use the HOT interface to organize the interaction translations to answer those questions. In each organization task, we asked them to use a specific feature provided by the application, resulting in four scenarios to study:

- a participant only uses the *temporal* ordering option to organize the questions
- a participant only uses the *causal* ordering option to organize the questions
- a participant only uses the *relationship suggestion* option to organize the questions
- a participant only uses the *free* ordering option to organize the questions

We asked each participant to perform one task in each of the four scenarios. The order of the scenarios in the experiment was assigned randomly to reduce learning effects.

In each scenario, we asked participants to organize queries that might be of interest in intelligence analysis of coast guards interdictions, as supported by the Migrant Boat visualization. The organization tasks involve two main sub-tasks: grouping related questions and reordering questions in groups. We instructed the participants to use HOT to first group and then reorder (if necessary) the translated interactions to provide answers to the evaluation questions given to them. Example of the questions asked in each scenario are shown in Table 7.1. After each task, we asked participants to express how difficult the task was and how confident they were about their answers.

We also asked participants to express their opinion about the quality of questions generated by Q2Q.

7.2.2 Data Collection

In this experiment, two types of data were collected: performance data and behavioral data. Performance data was gathered to compare the effectiveness of the various ordering and selection features of HOT. Behavioral data was gathered to acquire a better understanding of how people organize their questions to form a story.

7.2.2.1 Performance Data

The measurements collected in this user study are: *number of errors*, *time taken to group*, *time taken to reorder*, and *number of reordering actions*.

- *Error* indicates whether the participants correctly and successfully completed a task. A task is completed correctly if the questions that are grouped by

Temporal Ordering	Where were vessels of type Raft interdicted by Bonefish, and during what time they were interdicted, who was traveling on those boats? Who were the passengers that Bonefish interdicted? (Any vessel type)
Causal Ordering	Ship "A" interdicted Passenger Abreu Pancho (in the event "B"): What other boats did "A" interdict? Who else was on the boat that Abreu Pancho was on? Hint: First ask the questions to get the answer Bonefish for "A". Then get the answer 224 for event "B".
Relationship Suggestion	Adamant interdicted different types of vessels. Most commonly, Adamant interdicted vessels of the type "A". Discover the dates vessels of this type were interdicted. Discover the passengers who traveled with this type of vessel. Discover the passengers who were interdicted by Adamant. Hint: First ask questions to get the answer Rustic for "A".
Free Ordering	Passenger Heredia Roberto was interdicted on the date "A", June 3, 2006. Which passengers were interdicted on the same date as Heredia, Roberto? More specifically, who were on the boats (240, 366, or 369) on that date? Hint: First ask the questions to get the date (June 3, 2006) for "A", and then answer the questions.

Table 7.1: Questions posed to participants in each of the four scenarios. To complete each scenario, we asked participants to group and reorder translated interactions to answer each of the questions.

the participants answer the questions that are handed to them. The level of abstraction and the order they appear are not the concern in this measurement. Error has a binary value, meaning the participants can either perform a task correctly or incorrectly. This measurement is particularly useful in studying the effects of having the *relationship suggestion* option for selecting relevant questions. Ordering options can have indirect effects on choosing the right set of questions as well. This requires a more fine-grained study and is an avenue for future research.

- *Time Taken to Group* indicates the time spent to select the relevant questions and add them to a panel in the HOT interface. This is collected from the time participants start to look for the first question to the time they add the last question. The amount of time spent to reorder the selected questions is not part of this measurement. This measurement is collected to see the effects of *relationship suggestion* directly, and ordering options indirectly, on the user's speed in finding relevant pieces of information.

- *Time Taken to Reorder* indicates the amount of time it takes to reorder the questions after they are grouped in a panel. The time is recorded from when the participants are done grouping the questions and starting the reordering action to the time they finalize the order. The time is recorded even if the participant does not perform any ordering actions. This measurement is beneficial in studying the effects of automatic ordering options—*temporal* and *causal*—on constructing a coherent story from the translation of user queries.

- *Number of Ordering Actions* indicates the number of ordering actions performed by the participants during their organization of the interaction translations. This measurement is collected to study the effects of *temporal* and *causal* ordering on the performance of storytelling tasks. Recording only the time taken for reordering might not be sufficient to determine whether participants are satisfied with the automatic ordering options. Even if they spend considerable amount of time during the reordering tasks, they might spend the time reading and examining the suggested orders, but eventually agree with the suggestion with a few or no changes in the order of the textual translations.

7.2.2.2 Behavioral Data

The behavioral data that is collected in this user study are presented below.

Number of generalized questions versus number of specific questions. During selection and grouping of translated queries, participants can pursue different strategies in forming their story and completing their task. From the list of possible questions to choose from, they have the option to choose a general question or a specific question, both of which would answer the question that is presented to them. This behavioral data is valuable input for creating design guidelines of a history organizer board like HOT, in that it gives a better understanding of how users express their thoughts; either they present an overview, a detailed view, or a mixture of both. Thus, it can be helpful to determine whether a storytelling tool should provide/suggest more general questions, more specific questions, or both.

Order from general to specific questions versus order from specific to general questions. Once participants have selected their questions, if they have chosen both specific and general questions, how do they order them? This information is particularly valuable for designing an automatic ordering tool to assist the organization of user stories.

Questions and Variables. Another set of behavioral data collected focuses on how participants order the translations corresponding to the underlying interactions. In this dissertation, the focused interactions are filtering and selection. Filtering interactions are associated with questions, and selection interactions are associated with qualifier sentences. An example of a question and a qualifier sentence are

(a) Presenting all the variables first (b) Presenting all the variables last

Figure 7.2: Two panels showing all the variables appear (a) first and (b) last.

"Which types of vessels were interdicted by the ship Bonefish?" and *"Considering ship Bonefish"*, respectively. In this study, we categorized the participants' arrangements of questions and the variables into four ways listed as below (also see Figures 7.2a, 7.2b, 7.3a, and 7.3b):

- presenting all the variables first,
- presenting all the variables last,
- presenting variables before the associated questions, and
- Presenting variables after the associated questions.

Order of WH questions. We collected behavioral data about how participants order questions as a function of WH-question words. In this study, we examined four types of WH questions: *What*, *When*, *Where*, and *Who*. This covers questions about categorical, temporal, spatial, and nominal types of data. It is interesting

(a) Presenting variables before the associated questions

(b) Presenting variables after the associated questions

Figure 7.3: Two panels showing variables (a) before and (b) after questions.

to see how participants order their questions based on the types of data that are queried. This information can be useful in designing the automatic ordering features in the history organizer tool. In this experiment, since the given tasks might suggest particular orderings for the WH-questions, only the orderings that do not follow the task ordering are considered.

Order of independent variables. In addition to studying the order of interaction translations based on the WH-questions, we also examined the order based on the independent variables. We gathered this information to see whether participants organize the translated queries with similar values of independent variables close to each other in a group. Similarly, this information would guide an automatic reordering tool.

Keyword searched. We studied the keyword that participants used to find the relevant questions/text. This information is useful in providing a more advanced search feature for an organizer tool. Keywords fall into seven categories: *action*, *dependent variable*, *independent variable*, *value of independent variable*, *WH-question word*, *other data dimension*, and *other*. To have a better understanding of what each of these categories covers, consider the example question: "Which ship did interdict a vessel carrying a passenger John Smith?". The keywords for searching this question can be in one or more of the seven categories:

The *action* category refers to what describes the action, state, or occurrence. It is generally some form of the verb that is used in the question, here "interdict" and "carry".

The *dependent variable* category refers to the subject of the question. In the example above, "Ship" is a dependent variable.

The *independent variable* category refers to a data dimension that a dependent variable is questioned about. "Passenger" is an independent variable in the example above.

The *value of independent variable* category refers to the data values of an independent variable. "John Smith" is the value for the independent variable "Passenger" in the example question.

The *WH-question word* category refers to question words such as what, which, when, where, and who that can be used as a keyword to search a question. In the example, "which" is the WH-question word.

The *other data dimension* category refers to other data dimensions that are displayed in the visualization, but are not directly part of the query. For instance, "vessel" in the question above is a dimension that is not directly part of this query, but is semantically related to the query.

The *other* category refers to any other words that can be searched and do not fall into the other six categories. For instance, "did" belongs to the *other* category in example question.

7.2.3 Performance Results

For each performance aspect measured, we perform a F-test to assess the differences in performance between four features: *temporal ordering*, *causal ordering*, *relationship suggestion*, and *free ordering* in the HOT interface. If a significant difference is observed, we perform a Student's paired t-test between pairs of features to identify the feature that reveals significant difference in performance measurements. We test four hypotheses:

1 *Participants perform tasks with fewer errors using relationship suggestions.*

2 *Participants perform grouping tasks faster using relationship suggestions.*

3 *Participants perform reordering tasks faster using temporal or causal Ordering.*

4 *Participants perform fewer reordering actions using temporal or causal Ordering.*

Table 7.2 shows participants' performance data for the measurements described in Section 7.2.2.1. Differences in data performance for the four features are considered significant with $p \leq 0.05$. Even if the differences are not statistically significant, the considerable differences are shown in **bold** and reviewed for analysis.

The first hypothesis tests the effect of relationship suggestion on the number of errors users make while organizing a question/sentence. When participants use the relationship suggestion option to find related questions or sentences, they perform considerably fewer errors compared to other features (see Table 7.2, first row). However, this difference is not considered statistically significant. Based on the observations, participants are less distracted by other, low relevance questions when they use the relationship suggestion option. Color coding of related questions, not only accentuates relevant questions based on shared keywords, but also points out textual translations related to corresponding visualization states.

	Temporal Ordering		Causal Ordering		Relationship Suggestion		Free Ordering		
	AVG.	STD.	AVG.	STD.	AVG.	STD.	AVG.	STD.	P-value
Error	0.19	0.33	0.1	0.31	**0.03**	0.17	0.2	0.41	$0.33 > 0.05$
Time to Group	226.125	89.35	267.9	111.7	**206.4**	64.97	236.05	98.34	$0.21 > 0.05$
Time to Reorder	58.91	47.25	58.95	61.20	59.35	30.59	62.9	64.01	$0.99 > 0.05$
Number of Reorderings	**1.57**	1.39	**1.5**	1.67	2.36	1.86	5.95	4.35	$5.593E - 07 < 0.05$

Table 7.2: Averages and standard deviations of performance measurements for each of the four features, along side the p-value for all features over all participants.

The second hypothesis tests the effects of the relationship suggestion option on the speed of users while performing a grouping task. Similar to the number of errors, participants' performances improve, 37 seconds on average faster compared to other features, when they use relationships suggestion (see Table 7.2, second row). Even though the difference in time is considerable, it is not statistically significant. The observations indicate that participants are able to find related questions that require several searches (using the keyword search box) in fewer steps. (For instance, one can click on one of the questions that is already in the group to find the rest of the related questions.)

The third hypothesis tests the effect of automatic ordering on the time it takes users to reorder a group of questions or sentences. As shown in Table 7.2, third row, there is no considerable difference between the ordering features and other features in terms of participants' speed in reordering. The differences in standard deviations for the four features result in inconsiderable difference in speed. (Note that the time is recorded even if the participants do not perform any reordering actions and only spend their time examining the suggested or existing orders.)

The forth hypothesis tests the effects of automatic ordering on the number of reordering actions users perform while organizing the textual translations. As shown in Table 7.2, fourth row, there is a considerable and statistically significant difference between the number of reordering actions performed. A Student's paired t-test is performed to identify the features that are significantly different. The t-test reveals that temporal and causal ordering are significantly different from free ordering, with p-values of 0.0002 and 0.00004, respectively. Despite the considerable difference between temporal/causal ordering and relationship suggestions, the difference is not statistically significant. The difference between temporal and causal ordering themselves are also not considerable. The observations support the fact that even though users might still spend time on examining the suggested orders (third hypothesis), they end up agreeing with the current order and do not perform a considerable amount of reordering actions when they use temporal and causal ordering. This suggests that the effect of automatic ordering might become significant with the continuous use of the tool.

7.2.4 Behavioral Results

We collected behavioral data to study the grouping and ordering strategies of users while organizing their question/sentence. Thus, we collected this data across all four main tasks (performing grouping and reordering using the four available features in HOT).

Generalized Questions versus Specific Questions

The first set of behavioral data focuses on users' rearrangement activities based on the level of details that questions/sentences convey. The two corresponding hypotheses that have been tested are as follows:

1 *Participants use both general questions and specific questions with no significant difference.*

2 *Participants order questions from general to specific, or from specific to general, without significant difference.*

Table 7.3 shows the results for the number of general and specific questions used in the experiment. The collected data have been normalized, thus the average values presented in Table 7.3 indicate the percentage of time that questions having two levels of details were considered by the participants. A Student's paired t-test

General Questions		Specific Questions		
AVG.	STD.	AVG.	STD.	P-value
0.52	0.445	0.48	0.445	$0.38 > 0.05$

Table 7.3: Averages and standard deviations of the **number of general questions and specific questions** that users used in the experiment, with p-values

reveals that there is no significant difference in usage of general and specific questions in this experiment, with $p \leq 0.05$. These observations support the fact that some participants use both general and specific questions to cover a more comprehensive story of the steps needed to answer analysis questions. On the other hand, some participants preferred only general questions while others preferred only specific questions.

Table 7.4 shows the preferences in ordering the questions for the participants who included both types of questions in their stories. A Student's paired t-test shows no significant difference in users' preferences for these two types of ordering. However, there is a statistically significant difference in participants' preference in using one of the above mentioned ordering over no particular order (see Table 7.5). Participants generally order their pieces of information based on the levels of abstractions presented by that information.

General to Specific Order		Specific to General Order		
AVG.	STD.	AVG.	STD.	P-value
0.38	0.494	0.31	0.471	$0.32 > 0.05$

Table 7.4: Averages and standard deviations of occurrence of **general to specific and specific to general questions ordering** in the experiment, with p-values.

Order based on level of details		No particular Order based on levels of details		
AVG.	STD.	AVG.	STD.	P-value
0.66	0.494	0.484	0.484	$0.04 < 0.05$

Table 7.5: Averages and standard deviations of occurrence of **ordering based on the level of details in the text, versus no particular ordering based on the level of details**, with p-values.

Variables First		Variables Last		Variables Before Questions		Variables After Questions		
AVG.	STD.	AVG.	STD.	AVG.	STD.	AVG.	STD.	P-value
0.343	0.464	0.018	0.101	0.572	0.478	0.067	0.206	$5.77E-23 < 0.05$

Table 7.6: Averages and standard deviations of occurrence of **arrangement based on variables and questions**, with p-values.

Questions and Variables

The order that the textual translations are organized in a group can be also viewed in terms of variables and questions. We tested a hypothesis:

Participants order variables before the associated questions.

We collected four different types of ordering data with respect to the arrangement of variables and questions: *all variables presented first*, *all variables presented last*, *variables presented before associated questions*, and *variables presented after associated questions*. Table 7.6 shows the average number of times that each of the four above arrangements occurred in the experiment. An F-test reveals that there is a significant difference between occurrences of the four arrangements. A Students' paired t-test further narrows down the differences and reveals that the pattern of *variables before associated questions* occurred much more significantly (statistically, with p-value $0.01 < 0.05$) than the other three patterns. Occurrence of *all variables presented first* is also statistically significant compared to the remaining two patterns. This shows that participants generally tend to arrange variables before questions and, if not, mostly arrange them at the beginning.

Order of the WH-questions

Another ordering aspect explored is how participants order the questions with respect to WH-question words. Since the given tasks might suggest particular orderings of WH-words, we only study the orderings that do not match the task order. This results in 40 sub-tasks, with 38 *Who* questions, 29 *When* questions, 25 *What*

	Probability of occurrence in the text	% of time occurrence
what before **when**	**0.325**	**72%**
when before what	0.125	28%
what before where	0.150	55%
where before what	0.125	45%
who before what	0.20	27%
what before **who**	**0.55**	**73%**
where before **who**	**0.35**	**64%**
who before where	0.20	36%
where before **when**	**0.225**	**64%**
when before where	0.125	36%
when before **who**	**0.55**	**63%**
who before when	0.325	37%

Table 7.7: The probability of **occurrence of pairs of WH-questions in certain order**, along with the percentage of time that the order occurred compared to the reverse order.

questions, and 17 *Where* questions. We perform two types of analysis regarding the occurrence of WH patterns. The first analysis looks at the order of occurrence for pairs of WH-questions regardless of their exact position in the story. This analysis focuses on discovering the probability of occurrence of one WH-question type before or after another, not necessarily directly before or after. The second analysis is done by looking at the occurrence of the *exact* pattern of different subsets of WH-questions.

Table 7.7 shows the results of occurrence of pairs of WH-questions. The probability values in Table 7.7 indicate the probability of occurrence of two WH-questions in a certain order. The percentage indicates the percentage of the time that the WH-questions occurred in order, if ever. The ordered pairs shown in **bold** occurred more than 60% of the time compared to the reverse order of the pair. For instance, *What* questions occurred before *When* questions 72% of the time, compared to 28% for occurrence of *When* questions before *What* questions. Based on the results shown in Table 7.7, the most frequent pattern is:

What/Where When Who

	Average number of times occurred
when who	0.2033
who when	0.1685
who what	0.1225
when who what	0.1220
what who when	0.1008
what who	0.1008
when who where	0.0813
who when where	0.0674
where who when	0.0632
where who	0.0632

Table 7.8: The top highest patterns of WH-question words **across all tasks**, listed in order of highest probability of occurrence.

We perform further analysis to discover the probabilities of certain WH-patterns. The analysis studies the probability of occurrence of all permutations of four types of WH-questions across the entire experiment. We construct a joined probability model to calculate the joined probability distribution over the values of the four variables (when, who, where, and when). The chain rule of probabilities is used to decompose the joined probability expression:

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \\ P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i_1})$$

The joined probabilities are calculated for all combinations of subsets of size two, three, and four for the set {When, Where, Who, What}. Table 7.8 shows top ten highest probability of occurrence of WH patterns across all tasks. The patterns are

	Probability of occurrence
when who	0.2146
who when	0.1626
when who where	0.1430
where when	0.1393
where when who	0.1045
who where	0.0813
when who what	0.0715
where who when	0.0696
where what	0.0696
where who	0.0696

Table 7.9: The top ten patterns of WH-questions words for **temporal task**, listed in order of highest probability of occurrence.

sorted based on the probability values. A comprehensive table, including all the observed patterns with their probabilities, is provided in Appendix D, Table D.3.

The results show that participants generally use *Who* questions in between two other types of questions. Direct observations suggest that users tend to filter down into specific data dimensions, ask about people involved in events, then possibly focus on those people to ask more questions about other data dimensions. The conflict of these results with those of the first analysis is discussed in Section 7.3.2.

The occurrences of the patterns are also broken down by task. Table 7.9 shows the probability of occurrence of the patterns when users perform temporal ordering tasks (Table 7.1). The tasks consist of *Where*, *When*, and *Who* questions. However, some participants included other types of questions as well (e.g., What/Which). Eleven out of twenty participants ordered their questions differently from the order that the tasks are given. We considered these eleven orders for the analysis. Similar to the overall results, *Who* questions are asked generally in between two other types of questions. Also, the probability of a *Where* question occurring before a *When* question (either directly before or having some other type of question in between) is higher than the probability of a *When* question appearing before a *Where*.

	Probability of occurrence
who what where	0.2578
who what	0.2578
what who	0.1562
what where	0.125
where who	0.1124
who when	0.0859
who when where	0.0859
where what who	0.0750
where what	0.0750
what where who	0.0416

Table 7.10: The top ten patterns of WH-questions words for **causal tasks**, listed in order of highest probability of occurrence.

Table 7.10 shows the probability of occurrence of WH-question patterns for the casual ordering tasks listed in Table 7.1. The tasks consist of *What*, *Who*, and *Where* questions. However, *When* questions are also used when participants performed this task. Similar to the temporal ordering task, we consider eleven out of twenty participants' orders that do not match the task order. In this sub-task, unlike the overall occurrence of patterns in Table 7.8, *Who* questions appeared before other types of questions. Also, *What* questions appeared mostly before *Where* questions. The complete set of observed patterns with their probabilities are provided in Appendix D, Table D.2.

Table 7.11 shows the probabilities of occurrence of the tasks listed as *relationship suggestion* tasks in Table 7.1. These tasks involve *What*, *When* and *Who* questions. Twelve orders out of twenty are considered for this analysis. Based on the results shown in Table 7.11, *Who* questions mostly appear before *When* questions. The occurrence of *Who* questions in between two other types of questions is frequent.

Table 7.12 lists the probability of occurrence of WH patterns in the free ordering tasks in Table 7.1. The tasks involve *When*, *Who*, and *Where* questions. Participants also used *Where* questions to perform these subtasks. Only five orders are considered in this subtask, since the rest matched the task order. As shown in the Table 7.12, the occurrence of *Who* after *When* is the most probable. *Who* questions appeared also often in between two other types of questions.

	Probability of occurrence
when who	0.3
when who what	0.3
who when	0.1875
what who when	0.15
what who	0.15
what when	0.125
what when who	0.125
who what	0.1124
who what when	0.0749
when what	0.0375

Table 7.11: The top ten patterns of WH-questions words for **relationship suggestion tasks**, listed in order of highest probability of occurrence.

	Probability of occurrence
when who	0.3636
when who what	0.3636
who when	0.2272
who what	0.2272
where who when	0.0909
where who	0.0909

Table 7.12: The top ten patterns of WH-questions words for **free tasks**, listed in order of highest probability of occurrence.

Order of Independent Variables

In this experiment, we have also analyzed the orders that text translations appeared in the stories based on the position of independent variables in the questions. Similar to the study of WH-question orders, in this sub-study we only consider the orders that do not match the task order, which consists of 40 tasks involving 109 questions. In each task, we examine the order of the questions to see if the questions are arranged in a way that similar independent variables are set close to each other. In 21 out of 40 tasks, questions were reordered on independent variables (52%). Moreover, during the experiment, 10 participants mentioned that they prefer to order questions based on the independent variables.

	Avg.	STD.
Action	0.06	0.13
Dependent Variable	0.10	0.16
Independent Variable	0.004	0.037
Value of Independent Variable	**0.80**	**0.27**
WH-question Word	0.01	0.042
Other Data Dimension	0.005	0.039
Other	0.006	0.067

Table 7.13: Average and standard deviations for the number of times a **keyword category** is used to search.

Keyword Search

We have also studied the keywords that participants used to search for relevant questions. The keywords are categorized into seven categories listed in Section 7.2.2.2.

Table 7.13 shows the results for the average number of times a keyword category is used to search during the experiment. Performing an F-test on data from all the keywords categories reveals that there is statistically significant difference (p-value: $1.4536E - 194 < 0.05$) between the average number of keywords searched in each category. Based on the results from Students' paired t-test, *value of independent variables* was predominantly searched to find related questions (80%).

7.2.4.1 Quality of Q2Q Generated Questions

In addition to the data collected to evaluate features in HOT and study users' rearrangement behaviors, we also asked participants to express their opinion about the quality of the questions generated by Q2Q on a Likert-type scale. Participants declared how difficult it is to understand the meaning of questions on the scale in which 1 is very difficult, 2 is difficult, 3 is neutral, 4 is easy, and 5 is very easy. The results show that participants considered the automatic generated questions almost *easy to understand* (average 3.71 with 0.93 standard deviation), which is highly promising for a semi-automatic natural language generation system.

7.3 Discussion

7.3.1 Evaluation Preparation

We performed this user study to both evaluate features of the HOT interface and also to gain understanding of how users perform grouping and ordering tasks while arranging their translated interactions. This information is essential for designing an effective history organizer tool to facilitate the process of story telling and presentation.

We prepared four main tasks to study three features of the HOT interface. These tasks are designed carefully to have the same level of complexity. At the end of each task, we collected data about how difficult the tasks are and how confident participants are about their answers. To answer the questions, participants chose from a five-point Likert-type scale in which 1 means "very difficult or not at all confident" and 5 means "very easy or extremely confident". The results show no significant difference between each task in terms of difficulty or confidence (see Table 7.14). This provides assurance that the tasks are at the same level of difficulty, which in turn simplifies analysis and comparison of various features.

	Temporal Ordering		Causal Ordering		Relationship Suggestion		Free Ordering		
	AVG.	STD.	AVG.	STD.	AVG.	STD.	AVG.	STD.	P-value
Difficulty	3.3	0.75	3.6	1.18	3.6	1.2	3.5	0.77	$0.69 > 0.05$
Confidence	3.4	0.88	3.6	1.51	3.8	0.90	3.65	1.08	$0.68 > 0.05$

Table 7.14: Averages and standard deviations of task difficulty and confidence in performance, with corresponding p-value over all features and participants

7.3.2 Observations

In addition to collecting the performance data, we recorded general comments from participants.

Relationship Suggestion

The tasks are given to participants in a random order to decrease the learning effect. Excluding the data from participant who performed relationship suggestion task at the end, thus considering only 16 participants, we observed that 65% of participants either explicitly (by verbally asking) or implicitly (by clicking on relationship suggestion option) requested this feature to aid in their task. This indicates that users found the relationship suggestion feature effective. The other 35% who did not request relationship suggestion were either strictly following the experiment protocol or did not see the need.

Automatic Ordering

During the experiment, on a few occasions, we observed that participants do not agree with the order that is automatically applied to the textual translations. To have a better understanding, we further examined temporal ordering and causal ordering by recording the times that participants preferred an order that is not logically consistent with the chosen kind of ordering. We averaged this data over all participants. The results in Figure 7.15 show that participants prefer an order that is inconsistent with temporal ordering on average 0.525 times per task and with causal ordering on average 0.2 times per task. Performing a Student's paired t-test, the difference between temporal and causal ordering is not statistically significant, but is considerable. This reveals that participants tend to agree more with the order of arranging questions and sentences based on the concept of variables and questions about those variables than with the order that questions were originally asked in the visualization. These observations also suggest that automatic ordering should be

	Temporal Ordering		Causal Ordering		
	AVG.	STD.	AVG.	STD.	P-value
Num. of orderings preferred but not allowed	0.525	0.73	0.2	0.61	0.07 > 0.05

Table 7.15: Averages and standard deviations of the number of orderings preferred by participants but not allowed under temporal and causal ordering logic, with p-values

provided to users as suggestions, and that the user should have the ability to apply their own ordering even if it does not match the logic of automatic ordering.

WH-Questions Ordering Results

We have performed two types of analysis to study the user preference in ordering the questions based on WH-question words. The first analysis looks at the order of pairs of WH-words regardless of their position, while the second analysis looks at the exact occurrence of patterns of WH-words. The results from the first analysis suggest that the predominant order is *What/Where*, *When*, then *Who*. In contrast, the second analysis suggest that *Who* questions generally appear between two other types of WH-questions. This might be due to the differences in viewpoints in the analyses. The first analysis only looked at the occurrence of the order of pairs of WH-questions regardless of their position. For instance, to study the number of times a *When* question word occurs before a *Who* question word, the *When, What, Where, Who* pattern and *When, Who, Where* patterns are both considered. The second analysis looked for occurrence of fixed series of WH-questions. In addition to the fundamental differences in the analysis approaches, the second analysis would require a larger sample size to construct a more reliable probability model for predicting exact patterns. This also explains why the patterns that involve *Who* questions are in the top ten over all tasks. This is due to the larger number of *Who* questions that are involved in the tasks compared to the other types of WH-questions. It also allows more analysis on *Who* questions than on other WH-questions.

Input to History Organizer Tool

The output from Q2Q contains questions and qualifier sentences. It also contains questions and qualifier sentences that are crossed out. These types of text translations are generated to resemble undo-like actions, e.g., unfiltering to turn off an earlier filtering. As part of our observations, we noted that participants did not use the crossed out text in their stories. Not only is this type of text not used, it also resulted in confusion for the participants. To avoid confusion and have a shorter list of questions/sentences, input to a history organizer tool might include only regular questions/sentences and not the crossed out ones.

Other Ordering Options

In this experiment, we not only explored and evaluated the current ordering features (temporal and causal ordering), we also observed and discovered other ordering possibilities. These include ordering based on WH-questions, on independent variables, and on the level of detail that questions/sentences convey.

In the experiment presented in the previous section, ordering based on WH-questions and independent variables are two of the interesting ordering behaviors that we observed. Similar to temporal and causal ordering, these ordering patterns could be incorporated into the history organizer tool. For WH-question word patterns, a more comprehensive corpus of questions is required for constructing a more reliable model and discovering a general pattern for incorporating WH-question ordering suggestion.

In terms of ordering based on the level of details questions/sentences convey, we observed both general to specific and specific to general question ordering with no considerable difference. This suggest that both options could be provided to users to satisfy different user preferences.

The ordering patterns are not limited to the ones mentioned above. More complex interaction techniques can result in more compound and descriptive translations that accordingly affect the pieces of information used in the storytelling. This suggests that there are many types of ordering that can be studied in the future.

7.4 Summary

This chapter describes a user study for evaluating the history organizer tool described in Chapter 6. We studied three different ordering and grouping options provided by HOT. The evaluation results indicate that the relationship suggestion option in HOT provide considerable improvements in user speed and number of errors during question rearrangement. Temporal and causal ordering also considerably reduce the number of reordering actions that users need to perform. However, the automatically reordering options does not show considerable effects on their reordering speed.

In addition to the evaluation study, we also performed a behavioral analysis to study the grouping and ordering strategies of users while organizing their question/sentence fragments. The analysis suggests the viability of various other orderings and groupings, such as ordering based on levels of abstraction, the data values that appear in questions, and on WH-question words themselves. Specific insight was also acquired about users' searching behaviors. The user study contributes insights into the design of more effective and flexible information rearrangement capabilities for use in visual analytics research or presentation and storytelling tools.

Chapter 8

Conclusion

Query-to-Question (Q2Q) is a designer-guided translation system that captures user interactions with visualization, and transforms the interactions into natural language questions. Q2Q enhances foraging of data by increasing learnability, efficiency, and memorability in cross-filtering visualizations. Combined with the History Organizer Tool (HOT), Q2Q facilitates the sensemaking process with more reliable and presentable analytical steps. Integrating visualizations with Q2Q+HOT, users can understand the meaning of user interactions, recall their past queries, validate the questions they ask from the visualizations, reorganize the analysis steps taken, and present and share them with others in collaborative settings. Q2Q+HOT is a supporting system for bridging the analysis, foraging and sensemaking loops by merging data exploration, analytical reasoning and presentation into one coherent platform.

8.1 Contribution

The thesis statement of this dissertation is: *accompanying multiple coordinated view visualizations with user interaction translation and organization systems enhances data foraging and sensemaking during and after visual analysis.* This dissertation supports and validates six contributions:

- Query-to-Question Architecture. An architecture for automatically translating user interactions into natural language questions.
- Q2Q Implementation. An implemented system that realized the architecture to transform low-level user interactions into higher level English descriptions.
- Evaluation. An evaluation and analysis of the effects of accompanying visualizations with textual translation of user interactions.
- Design Guidelines. A set of guidelines for applying the design space of user interaction translation to visualizations.
- History Organizer Tool (HOT) Implementation. An implementation of a history organizer board for sequencing, rearranging, and grouping the steps in an analysis process as represented by questions asked.
- Rearrangement Evaluation and Analysis. Increased understanding of users' strategies for rearranging analytical questions, grounded in user experiment.

These contributions in visual analytics illustrate the effectiveness of the supporting tools (Q2Q + HOT) that are implemented and incorporated to improve the usability of existing visualizations without sacrificing the benefits of sophistication and richness offered by complex and highly interactive visual representations. The Query-to-Question system is generalizable yet customizable to various data domains. It has been integrated into visualizations of data from diverse sets of knowledge domains, spanning popular entertainment, sports, journalism, politics, and intelligence. The architecture of Q2Q is designed so that the current system can readily be expanded to cover new interaction techniques. Combined with the History Organizer Tool (HOT), the output of Q2Q can be utilized in reasoning, supporting, and presentation of the analytical process, hypotheses, and conclusions. This provides a platform for studying user strategies in reformatting the questions they pose to data through visual queries, and how different automatic sequencing can assist them in that process.

8.2 Benefits

Accompanying visualizations with Q2Q + HOT provides many benefits to Visual analytic researchers, visualization designers and developers, and visualization users.

The benefits provided to *Visual Analytic Researchers* are:

- introducing a new perspective toward capturing and meaningfully transforming user interaction provenance;
- providing a working translation system for studying combinations of textual descriptions and visualizations;
- providing capabilities to integrate and extend the current translation system to a new set of visualization and interaction techniques, for studying the nuances they introduce; and
- providing an initial platform for studying and implementing storytelling tools.

The benefits provided to *Visualization Designers and Developers* are:

- increasing the accessibility of the visualizations they design and the techniques they develop;
- providing guidelines for appropriateness and understandability of the techniques they develop and combine into tools (assuming that understandable translations correlate with the user comprehension of the techniques); and
- limiting the need to provide descriptions of unfamiliar visualization domains or interaction techniques by using the translation as a proxy explanation tool.

The benefits provided to *Visualization Users* are:

- learning a new visualization by having access to an understandable summary of the meaning of its interactions, visual representations, data domain, and interdependencies of data attributes;
- augmenting the usability of interactions for experts by validating the questions they ask from a visualization and providing a history of interactions that can be reviewed and shared;
- facilitating reasoning through support for hypothesis formation, analysis, and validation, by providing an interactive board to reorganize and present analytical steps; and
- enabling effective accessibility of sophisticated visualizations to end-users.

8.3 Future Work

8.3.1 Language Quality

Q2Q aims to be a general-purpose interaction translation system that accurately reproduces the detailed domain semantics of queries performed in a visualization. Toward this goal, visualization designers/domain experts guide the process of generating reasonable translations in the offline generation stage by specifying relationships between data dimensions in the form of sentences. Generation of high-quality text depends heavily on the quality of this input. Translations can better convey the meaning of interactions when the description of involved data values and relationships accurately and precisely reflects type semantics. From a grammatical perspective, given a descriptive sentence as an input, the system currently gives back an error in the case of grammatical incorrectness. The system does this naively by checking parts of speech and their arrangement in a sentence. This can be improved by adapting more advanced rules to check grammar.

8.3.2 Language Complexity

Q2Q currently accepts embedded reduced relative clauses to encourage designers to express relationships between data dimensions effectively. For instance, the relationship between *Passenger* and *Resolution* in Figure 5.1 is expressed as "vessels carrying passengers have resolution" rather than "vessels *that were* carrying passengers have resolution". Consequently, Q2Q is able to output complex relationships without dealing with overly complex forms of sentence structure. This, however, limits the formats of the descriptive sentences that Q2Q can process. In the future, the system can be modified to also accept subordinate clauses, allowing for more detailed description of inter- and intra-dimensional relationships.

Moreover, pairs of dimensions are sometimes unclearly or ambiguously related, making it hard for the designer to express dimension relationships in a clear language. Consider the two dimensions *Source Actors* and *Source States* in Figure 5.5. *Source States* are the "home" countries, if any, of *Source Actors* that include national and international organizations, groups, and ethnicities, as well as the *Source States* as acting entities themselves. The multiple subtle relationships between different data values of the two dimensions are hard to unify into language to describe a single, more abstract, common relationship between all potential pairwise data values. Specialization of question generation to capture such nuances is another avenue of future exploration.

8.3.3 Relationships Database

Including an offline generation stage has several benefits. It allows application of Q2Q to new domains and data sets without the substantial cost of constructing either a centralized corpus or individual example repositories. It allows designers/experts to make sure that generated text conveys the necessary, sufficient, and precise meaning desired. It caches partially evaluated linguistic specifications of dimensions and relationships, with rapid completion through simple substitution during later interaction. Offline generation also consumes considerably less design time compared to corpus construction in example-based and statistics-based methods. It introduces less variation in generated text compared to these methods, which may decrease perceived quality and naturalness, but may increase efficiency and effectiveness of scanning and reading many interactions that have been translated along complex paths of inquiry over time.

Relation specifications given as input in offline generation are often applicable across domains. Over time, the database populated by these specifications can grow into a repository for rapid reuse in offline design. In the future, with enough domain-spanning examples, example-based generation [68] could be integrated into the current system to augment (or perhaps even substitute for) Sentence Specification in the offline architecture. This future addition would take advantage of the dimensional scalability of example-based approaches, yet through Question Confirmation continue to meet the special requirements of interaction translation.

8.3.4 Interaction Techniques

Gotz and Zhou [6] characterize interaction activities during visual analysis, including *actions*: intermediate sets of steps that connect low level interaction events (e.g., mouse clicks) to high-level analytical processes (e.g., goals and hypotheses). By recording the visualization parameter changes that trigger dynamic queries, we "capture" analytic intention at the action level of abstraction and in a way that generalizes across visualization applications. Gotz and Zhou further categorize actions as *data exploration actions*, *visual exploration actions*, *insight actions*, and *meta actions*. The Q2Q implementation currently focuses on selection and filtering interactions as used in cross-filtering. Although capturing and presenting these data exploration actions is the current focus of Q2Q, ongoing generalization of translation system capabilities can consider other types of actions. The current syntactic grammar can be extended to translate queries triggered by other types of interactions, such as the visual exploration action of sorting or the insight actions of bookmarking and annotating. New grammar rules can be added for types of relationships and new interaction techniques not covered by existing grammar rules.

8.3.5 Story Telling

The questions generated by the Q2Q architecture are coherent, modular units of provenance. They are represented internally as first class data for visual analysis of individual and collaborative analytical process, storytelling, and collection into reports. HOT takes this data as input and enables users to reorganize and regroup the questions either freely or using the algorithmic suggestions from the system. This initial form of telling the story with questions paves the way for a fully interactive environment in which live streams and recorded logs of questions can be formatted, ordered, edited, and annotated like any other data. This makes it possible to visualize and interact with analytic provenance at a more natural level of cognitive abstraction in the future.

Moreover, the user study presented in Chapter 7 informs a set of user preferences and behaviors in arranging the analytical process. These insights can be incorporated and applied in the history organizer tool to enhance the current ordering and grouping features. An effective set of suggestions combined with user preferences can lay out a basic narrative of the analytical process, and then allow the users to build the story of their own upon that foundation. Studying the practicality of an intelligent storytelling tool is an interesting and relatively new area of research in visual analytics to be explored moving forward.

8.3.6 Evaluation

In this dissertation, we conducted an evaluation to see the effects of a translation system on certain aspects of usability: learnability, efficiency, memorability, and satisfaction in individual use. As part of a longitudinal study in the future, we could analyze the effects of Q2Q's presence on the usability of visualizations in collaborative applications, focusing on different-time-different-space cases in which the time difference is substantial.

Similarly, in the evaluation performed to study user strategies for reorganizing the questions asked from visualizations, the focus was on individual usage. In a more comprehensive study, the effectiveness of reorganization could be studied when it is shown to a third person. This could inform the algorithmic rearrangement system how to sequence and group questions in a way more understandable to collaborators.

In designing the architecture of Q2Q, we identified a set of design factors that affect translation, including user knowledge, user roles, visualized data characteristics, and types of interaction involved. These factors serve to define a set of guidelines for other researchers to explore and expand the design space of user interaction for translation. Generated language has many design aspects, such as the overall quality, formality, and formatting (e.g., HTML-like markup). How these and other design aspects affect the utility and usability of Q2Q is a promising avenue for future exploration.

8.4 Conclusion

My main goal in designing the translation and reorganization systems is to make sophisticated, interactive visualization tools more accessible to wider groups of users with diverse knowledge levels and backgrounds. I believe that Q2Q + HOT achieves this goal for an important subset of coordinated multiple view visualization techniques. This opens a new path for visual analytics researchers to pursue a new area of research, expand it to a wide variety of visualization techniques, and thereby explore user interaction provenance from a new perspective to reach greater understanding of how it can be utilized throughout the analysis process.

Bibliography

[1] Peter Pirolli and Stuart Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," *Proceedings of International Conference on Intelligence Analysis*, pp. 2–4, 2005.

[2] Jonathan C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," *Coordinated and Multiple Views in Exploratory Visualization, International Conference on, Coordinated and Multiple Views in Exploratory Visualization, International Conference on 2007*, pp. 61–71, 2007.

[3] Jeffrey Heer, Jock D. Mackinlay, Chris Stolte, and Maneesh Agrawala, "Graphical histories for visualization: Supporting analysis, communication, and evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1189–1196, November/December 2008.

[4] Louis Bavoil, Steven P. Callahan, Patricia J. Crossno, Juliana Freire, Carlos E. Scheidegger, Cláudio T. Silva, and Huy T. Vo, "VisTrails: Enabling interactive multiple-view visualization," in *Proceedings of the IEEE Conference on Visualization (VIS)*, 2005, pp. 135–142.

[5] Dong Hyun Jeong, Wenwen Dou, Felesia Stukes, William Ribarsky, Heather Richter Lipford, and Remco Chang, "Evaluating the relationship between user interaction and financial visual analysis," in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, Columbus, OH, October 2008, pp. 83–90, IEEE.

[6] David Gotz and Michelle X. Zhou, "Characterizing users' visual analytic activity for insight provenance," in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, Columbus, OH, October 2008, pp. 123–130, IEEE.

[7] Ryan Eccles, Thomas Kapler, Robert Harper, and William Wright, "Stories in GeoTime," *Information Visualization*, vol. 7, no. 1, pp. 3–17, February 2008.

[8] Jeffrey Heer, Fernanda B. Viégas, and Martin Wattenberg, "Voyagers and voyeurs: Supporting asynchronous collaborative information visualization," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, San Jose, California, 2007, pp. 1029–1038, ACM.

[9] Russell A. Lankenau, M. Andrew Eick, Alexander Decherd, Maxim Khailo, Phil Paris, and Jesse Fugitt, "Second place, corporate category: DECIDE™," in *Conference Compendium of the IEEE Symposium on Visual Analytics Science & Technology (VAST)*, Baltimore, MD, October 31–November 2 2006.

[10] Fernanda B. Viégas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon, "Many Eyes: A site for visualization at internet scale," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1121–1128, November/December 2007.

[11] T. J. Jankun-Kelly, Ma Kwan-Liu, and M. Gertz, "A model and framework for visualization exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 357–369, March-April 2007.

[12] M. Kreuseler, T. Nocke, and H. Schumann, "A history mechanism for visual data mining," in *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*. October 2004, pp. 49–56, IEEE.

[13] Anthony C. Robinson and Chris Weaver, "Re-visualization: Interactive visualization of the process of visual analysis," in *Proceedings of GIScience Workshop on Visual Analytics & Spatial Decision Support*, Münster, DE, September 2006.

[14] Donald Pellegrino, Chi-Chun Pan, Anthony Robinson, Michael Stryker, Junyan Luo, Chris Weaver, Prasenjit Mitra, Chaomei Chen, Ian Turton, and Alan MacEachren, "Grand challenge award: Data integration visualization and collaboration in the VAST 2008 challenge," in *Conference Compendium of the IEEE Symposium on Visual Analytics Science & Technology (VAST)*, Columbus, OH, October 2008.

[15] Christopher E. Weaver, *Improvise: A User Interface for Interactive Construction of Highly-Coordinated Visualizations*, Ph.D. thesis, University of Wisconsin–Madison, Madison, WI, June 2006.

[16] Maximilian Scherr, "Multiple and coordinated views in information visualization," in *Trends in Information Visualization*, 2008.

[17] Maryam Nafari and Chris Weaver, "Poster: Translating cross-filtered queries into questions," in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST) (Compendium)*, Salt Lake City, UT, October 2010, IEEE.

[18] Chris Weaver, "Cross-filtered views for multidimensional visual analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 2, pp. 192–204, March-April 2010.

[19] James J. Thomas and Kristin A. Cook, "Illuminating the path: Research and development agenda for visual analytics," 2005.

[20] Edward R. Tufte, *The Visual Display of Quantitative Information*, 2001.

[21] Alexander Bock, Erik Sundn, Bingchen Liu, and Burkhard Wuensche, "Coherency-based curve compression for high-order finite element model visualization," in *IEEE Transactions on Visualization and Computer Graphics*, 2012, pp. 2315–2324.

[22] PhilipC. Chen, "Applications of scientific visualization to meteorological data analysis and animation," in *Computer Animation 90*, Nadia Magnenat-Thalmann and Daniel Thalmann, Eds., pp. 31–38. Springer Japan, 1990.

[23] Jason Dykes and Susanne Bleisch, "Quantitative data graphics in 3d desktop-based virtual environments an evaluation," in *International Journal of Digital Earth*, 2014.

[24] Paul Craig, Alan Cannon, Robert Kukla, and Jessie Kennedy, "Matse: The microarray time-series explorer," in *2012 IEEE Symposium on Biological Data Visualization (BioVis 2012)*, 2012, pp. 41–48.

[25] Kresimir Matkovic, Wolfgang Freiler, Denis Gracanin, and Helwig Hauser, "Comvis: a coordinated multiple views system for prototyping new visualization technology," in *Proceedings of the 12th International Conference Information Visualisation*, 7 2008.

[26] C. M. Burns, "Putting it all together: Improvising display integration in ecological displays," *Human Factors*, pp. 226–241, 2000.

[27] J. G. Trafton, S. S. Kirschenbaum, T. L. Tsu, R. T. Miyamoto, J. A. Ballas, and P. D. Raymond, "Turning pictures into numbers: Extracting and generating information form complex visualizations," in *Journal of Human-Computer Studies*. 2000, vol. 5, pp. 827–850, Association for Computational Linguistics.

[28] Ben Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings of the IEEE Symposium on Visual Languages*, Boulder, CO, September 1996, pp. 336–343.

[29] Alen Dix and Geoffrey Ellis, "Starting simple: adding value to static visualisation through simple interaction," *Working Conference on Advanced Visual Interfaces*, 1998.

[30] Leland Wilkinson, "The grammar of graphics," New York, NY, USA, 2005, Springer.

[31] Robert Spence, "Information visualization: Design for interaction," 2007, Prentice Hall.

[32] Robert Amar, James Eagan, and John Stasko, "Low-level components of analytic activity in information visualization," in *IEEE Symposium on Information Visualization*, 2005, pp. 111–117.

[33] Ji Soo Yi, Youn ah Kang, John T. Stasko, and Julie A. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1224–1231, November/December 2007.

[34] Ji Soo Yi, Rachel Melton, John Stasko, and Julie A. Jacko, "Dust and magnet: multivariate information visualization using a magnet metaphor," *Information Visualization*, vol. 4, pp. 239–256, 2005.

[35] M. Wattenberg and J. Kriss, "Designing for social data analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 549–557, 2006.

[36] Benjamin B. Bederson, Jesse Grosjean, and Jon Meyer, "Toolkit design for interactive structured graphics," *IEEE Transaction on Software Engineering*, vol. 30, pp. 535–546, 2004.

[37] Ramana Rao and Stuart K. Card, "The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information," in *Conference on Human Factors in Computing Systems (CHI '94)*, Boston, MA, USA, 1994, pp. 318–322.

[38] Jeffrey Heer and Dana Boyd, "Vizster: Visualizing online social networks," in *Presented at IEEE Symposium on Information Visualization*, Minneapolis, MN, USA, 2005, pp. 33–40.

[39] Christopher Ahlberg, "Spotfire: an information exploration environment," in *SIGMOD Record*, 1996, pp. 25–29.

[40] Mei C. Chuah, Steven F. Roth, Joe Mattis, and John Kolojejchick, "SDM: selective dynamic manipulation of visualizations," in *presented at ACM Symposium on User Interface and Software Technology*, 1995, pp. 61–70.

[41] Claudio T. Silva, Juliana Freire, and Steven P. Callahan, "Provenance for visualizations: Reproducibility and beyond," *Computing in Science and Engineering*, vol. 9, no. 5, pp. 82–89, September/October 2007.

[42] Scott R. Klemmer, Michael Thomsen, Ethan Phelps-Goodman, Robert Lee, and James A. Landay, "Where do web sites come from?: Capturing and interacting with design history," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 2002, pp. 1–8.

[43] Steven P. Callahan, Juliana Freire, Emanuele Santos, Carlos E. Scheidegger, Cláudio T. Silva, and Huy T. Vo, "Managing the evolution of dataflows with vistrails," in *Proceedings of the International Conference on Data Engineering Workshops (ICDEW)*, 2006, pp. 71–75.

[44] Frank L. Greitzer, "Methodology, metrics and measures for testing and evaluation of intelligence analysis tools," Tech. Rep. PNWD-3550, Pacific Northwest Division Battelle Memorial Institute, Richland, WA, 2005.

[45] Ernest Hampson and Paula Cowley, "Instrumenting the intelligence analysis process," in *Proceedings of the First International Conference on Intelligence Analysis Methods and Tools*, McLean, VA, May 2005, Battelle Memorial Institute, number PNWD-SA-6893.

[46] Yedendra Babu Shrinivasan and Jarke J. van Wijk, "Supporting the analytical reasoning process in information visualization," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, Florence, Italy, 2008, pp. 1237–1246, ACM.

[47] Catherine Plaisant, Anne Rose, Gary Rubloff, Richard Salter, and Ben Shneiderman, "The design of history mechanisms and their use in collaborative educational simulations," *Proc. CSCL*, pp. 348–359, 1999.

[48] M. Derthick and S. F. Roth, "Enhancing data exploration with a branching history of user operations," *Konwldge Based Systems*, vol. 14, no. 1-2, pp. 65–74, March 2001.

[49] Nazanin Kadivar, Victor Chen, Dustin Dunsmuir, Eric Lee, Cheryl Qian, John Dill, Christopher Shaw, and Robert Woodbury, "Capturing and supporting the analysis process," *IEEE Symposium on Visual Analytics Science and Technology*, pp. 131–138, 2009.

[50] Maryam Nafari and Chris Weaver, "Augmenting visualization with natural language translation of interaction: A usability study," in *Proceedings of the Eurographics Conference on Visualization (EuroVis)*, Leipzig, Germany, June 2013, vol. 32.

[51] Yang Chen, Jing Yang, and William Ribarsky, "Toward effective insight management in visual analytics systems," in *Proceedings of the Pacific Visualization Symposium*, April 2009, pp. 49–56.

[52] Paul S. Jacobs, "Knowledge-intensive natural language generation," *Artificial Intelligence*, vol. 33, no. 3, pp. 325–378, 1987.

[53] Mary Dee Harris, "Building a large-scale commercial NLG system for an EMR," in *Proceedings of the Fifth International Natural Language Generation Conference (INLG)*, Salt Fork, Ohio, 2008, pp. 157–160, Association for Computational Linguistics.

[54] Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy, "Choosing words in computer-generated weather forecasts," *Artificial Intelligence*, vol. 167, no. 1-2, pp. 137–169, 2005.

[55] Stephen Springer, Paul Buta, and Thomas C. Wolf, "Automatic letter composition for customer service," in *Proceedings of the Third Conference on Innovative Applications of Artificial Intelligence*, 1991, pp. 67–83.

[56] Ehud Reiter and Roma Robertson, "The architecture of the STOP system," in *Proceedings of the Workshop on Reference Architecture for Natural Language Generation*, 1999, pp. 1–9.

[57] Ehud Reiter and Robert Dale, *Building Natural Language Generation Systems*, Cambridge University Press, 2000.

[58] Irene Langkilde and Kevin Knight, "Generation that exploits corpus-based statistical knowledge," in *Proceedings of the International Conference on Computational Linguistics*, 1998, pp. 704–710.

[59] Anja Belz, "Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models," *Natural Language Engineering*, vol. 14, pp. 431–455, 2008.

[60] Shimei Pan and James Shaw, "SEGUE: A hybrid case-based surface natural language generator," in *Natural Language Generation*, Anja Belz, Roger Evans, and Paul Piwek, Eds., vol. 3123 of *Lecture Notes in Computer Science*, chapter SEGUE: A Hybrid Case-Based Surface Natural Language Generator, pp. 130–140. Springer, 2004.

[61] Sebastian Varges and Chris Mellish, "Instance-based natural language generation," *Natural Language Engineering*, vol. 16, pp. 309–346, 2010.

[62] David DeVault, David Traum, and Ron Artstein, "Practical grammar-based NLG from examples," in *Proceedings of the International Natural Language Generation Conference (INLG)*, June 2008, pp. 77–85.

[63] Vibhu O. Mittal, Giuseppe Carenini, Johanna D. Moore, and Steven Roth, "Describing complex charts in natural language: A caption generation system," *Computational Linguistics*, vol. 24, no. 3, pp. 431–467, Sept. 1998.

[64] Somayajulu G. Sripada, Ehud Reiter, Jim Hunter, and Jin Yu, "Summarizing neonatal time series data," in *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics (EACL)*, Budapest, Hungary, 2003, vol. 2, pp. 167–170, Association for Computational Linguistics.

[65] Jin Yu, *SumTime-Turbine: A Knowledge-Based System to Generate English Textual Summaries of Gas Turbine Time*, Ph.D. thesis, University of Aberdeen, 2004.

[66] Kathleen McKeown, Karen Kukich, and James Shaw, "Practical issues in automatic documentation generation," in *Proceedings of the Fourth Conference on Applied Natural Language Processing (ANLC)*, Stuttgart, Germany, 1994, pp. 7–14, Association for Computational Linguistics.

[67] Jenny Rose Finkel, Trond Grenager, and Christopher Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, June 2005, pp. 363–370.

[68] Shimei Pan and Wubin Weng, "Designing a speech corpus for instance-based spoken langauge generation," *Int. Conf. on Natural Language Generation*, , no. 49-56, 2002.

[69] Dan Klein and Christopher D. Manning, "Fast exact inference with a factored model for natural language parsing," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. December 2003, vol. 15, pp. 3–10, MIT Press.

[70] David Hall John Canny and Dan Klein, "A multi-teraflop constituency parser using GPUs," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, Washington, USA, October 2013, pp. 1898–1907.

[71] Eugene Charniak, "A maximum-entropy-inspired parser," in *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, Stroudsburg, PA, USA, 2000, NAACL 2000, pp. 132–139, Association for Computational Linguistics.

[72] Christiane Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998.

[73] Albert Gatt and Ehud Reiter, "SimpleNLG: A realisation engine for practical applications," in *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG)*, Athens, Greece, March 2009, number 4, pp. 90–93, Association for Computational Linguistics.

[74] Hercules Dalianis and Eduard H. Hovy, "Aggregation in natural language generation," in *Selected Papers from the Fourth European Workshop on Trends in Natural Language Generation, An Artificial Intelligence Perspective*, London, UK, UK, 1996, EWNLG '93, pp. 88–105, Springer-Verlag.

[75] Jakob Nielsen, *Usability Engineering*, Boston, Academic Press, 1993.

[76] Ben Shneiderman, *Designing the user interface: Strategies for effective human-computer-interaction*, Reading, Mass, Addison Wesley Longman, 1998.

[77] Chris Weaver, "Infovis 2007 contest entry: Cinegraph," *IEEE Visualization 2007 Conference Compendium*, October 2007.

[78] N. Hari Narayanan and Roland Hubscher, *Visual language theory*, chapter Visual language theory: towards a human computer interaction perspective, pp. 87–128, Springer-Verlag New York, Inc., New York, NY, USA, 1998.

[79] Gregory D. Abowd and Alan J. Dix, "Integrating status and event phenomena in formal specifications of interactive systems," in *Proceedings of the Symposium on Foundations of Software Engineering (SIGSOFT)*, New York, NY, USA, 1994, pp. 44–52, ACM.

[80] Donald A. Norman, "Cognitive engineering," in *User Centered System Design: New Perspectives on Human-Computer Interaction*, Donald A. Norman and Stephen W. Draper, Eds., pp. 31–61. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[81] Philip A. Schrodt, *Event Data in Foreign Policy Analysis*, pp. 145–166, Prentice-Hall, New York, 1994.

[82] Christopher Ahlberg, "Spotfire: An information exploration environment," *SIGMOD Record*, vol. 25, no. 4, pp. 25–29, December 1996.

[83] Ingrid Zukerman and Diane Litman, "Natural language processing and user modeling: Synergies and limitations," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1-2, pp. 129–158, Mar. 2001.

[84] Ehud Reiter, S. Sripada, and S. Williams, "Acquiring and using limited user models in NLG," in *Proceedings of the European NLG workshop (ENLGW)*, Budapest, Hungary, 2003, pp. 87–94.

[85] Nancy Green, "Generation of biomedical arguments for lay readers," in *Proceedings of the International Natural Language Generation Conference (INLG)*, Sydney, Australia, June 2006, pp. 114–121.

[86] Chrysanne DiMarco, H. Dominic Covvey, P. Bray, D. Cowan, V. DiCiccio, E. Hovy, J. Lipa, and D. Mulholland, "The development of a natural language generation system for personalized e-health information," in *Proceedings of the World Congress on Health Informatics*, 2007.

[87] Maryam Nafari and Chris Weaver, "Query2question: Translating visualization interaction into natural language," in *IEEE Transactions on Visualization and Computer Graphics*, 2015.

[88] Edward Segel and Jeffrey Heer, "Narrative visualization: Telling stories with data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1139–1148, November 2010.

[89] Robert Kosara and Jock Mackinlay, "Storytelling: The next step for visualization," *IEEE Computer*, vol. 46, no. 5, pp. 44–50, 2013.

[90] Catalina M. Danis, Fernanda B. Viégas, Martin Wattenberg, , and Jesse Kriss, "Your place or mine?: Visualization as a community component," *CHI*, 2008.

[91] Wesley Willett, Jeffrey Heer, Joseph M. Hellerstein, and Maneesh Agrawala, "Commentspace: Structured support for collaborative visual analysis," *CHI*, 2011.

[92] Jessica Hullman and Nicholas Diakopoulos, "Visualization rhetoric: Framing effects in narrative visualization," *Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2231–2240, December 2011.

[93] Jessica Hullman, Steven Drucker, Nathalie Henry Riche, Bongshin Lee, Danyel Fisher, and Eytan Adar, "A deeper understanding of sequence in narrative visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, 2013.

Appendix A

Author's List of Publications

A.1 Publications

- Maryam Nafari and Chris Weaver. "Query2Question: Translating Visualization Interaction into Natural Language". IEEE Transaction on Visualization and Computer Graphics 2015.
- Maryam Nafari and Chris Weaver. "Augmenting Visualization with Natural Language Translation of Interaction: A Usability Study". The Eurographics Conference on Visualization (EurVis 2013).
- Chris Weaver and Maryam Nafari. "Capturing Connotation in Interactive Visualization". Workshop on Analytic Provenance: Process + Interaction + Insight, Vancouver, BC, May 2011.
- Maryam Nafari and Chris Weaver. "Poster: Translating Cross-Filtered Queries into Questions". IEEE Conference on Visual Analytics Science and Technology 2010, Salt Lake City, UT, October 2010.

Appendix B

Tags

B.1 Clause level

CP complement

ROOT root

S sentence

SBAR clause introduced by a subordinating conjunction

SBARQ Direct questions by wh-word or wh-phrase

SINV Inverted declarative sentence

SQ Inverted yes/no question, or main clause of a wh-question

B.2 Phrase Level

ADJP Adjective phrase

ADVP Adverb phrase

CONJP Conjunction phrase

C Auxiliary for questions

FRAG Fragment

INTJ Interjection

LST List marker. Includes surrounding punctuation

NAC Show scope of certain prenominal modifiers within a NP

NP Noun phrase

NPM Noun phrase modifier

NX Used within certain complex NPs to mark the head of the NP

PP Prepositional phrase

PRN Parenthetical

PRT Particle

QP Quantifier phrase

RRC Reduced relative clause

UCP Unlike coordinated phrase

VP Verb phrase

WHADJP Wh-adjective phrase. e.g. “how hot”

WHAVP Wh-adverb phrase

WHNP Wh-noun phrase, e.g. “which people”

WHPP Wh-prepositional phrase. e.g. “of which”

X Unknown tag

B.3 Word level

AUX Auxilliary verb

CC Coordinating conjunction

CD Cardinal number

DT Determiner

EX Existential there

FW Foreign word

IN Preposition or subordinating conjunction

JJ Adjective

JJR Adjective, comparative

JJS Adjective, superlative

LS List item maker

MD Modal

N Noun

NN Noun, singular or mass

NNS Noun, plural

NNP Proper noun, singular

NNPS Proper noun, plural

PDT Predeterminer

POS Possessive ending

PRP Personal pronoun

PRP$ Possessive pronoun

DQ Quantifiers

RB Adverb

RBT Adverb, comparative

RBS Adverb, superlative

RP Particle

SYM Symbol

TO To

UH Interjection

V Verb

VB Verb, base form

VBD Verb, past tense

VBG Verb, gerund or present participle

VBN Verb, past participle

VBP Verb, non-3rd person singular present

VBZ Verb, 3rd person singular present

WDT Wh-determiner

WP Wh-pronoun

WP$ Possessive wh-pronoun

WRB Wh-adverb

PCC Punctuation comma

PCE Punctuation exclamation point

PCS Punctuation fullstops

PCQ Punctuation question mark

Appendix C

Tasks

1. Identify the people who either played or directed the movie *21 Grams*.
2. Identify the people who played or directed the movie *21 Grams* and also they play or direct in the movies with *Crime* genre.
3. Identify the movies that Naomi Watts has played in besides *21 grams*.
4. Identify the Genres of the movie *The Ring*.
5. Describe what the movie table means.
6. Describe what the people table means.

Table C.1: Interactive (1–4) and descriptive (5–6) questions asked of participants about the interface in Figure 4.1.

1. Identify the genres of the movies *The Prestige*.

2. Identify the people who played or somehow associated with *The Prestige*

3. Identify the genres that *Christian Bale* have a movie in and also the genres that the movie *The Prestige* is.

4. Identify the movies that are in *Drama* genres.

5. Describe what the genre table means.

6. Identify the people who are in the movies *25 hour* to *50 First Date*.

7. Identify the movies which are in *Crime* genre.

8. Identify the genres which the selected movies are in.

9. Identify the movies that genres got filtered on.

Table C.2: Interactive (1–4, 6, and 7) and descriptive (5, 8, and 9) questions asked of participants about the interface in Figure 4.1 accompanied with Q2Q interface.

1. Identify the programs which are managed by *Anne Maglia or Brian M. Patten*.

2. Identify the Application fields that *Anne Maglia or Brian M. Patten* work under.

3. Identify the programs that are in the organization *AST* and are managed by *Anne Maglia or Brian M. Patten* and also the Application field *other applications* is in it.

4. Identify the program that are managed by *Aixa Alfanso...,Alexander Warzkopf*.

5. Identify the program managers that work under the organization *CBET*.

6. Identify the directorates under which selected program managers work.

7. Explain the resulted directorate.

Table C.3: Interactive (1–5) and descriptive (6 and 7) questions asked of participants about the interface in Figure 4.2.

Appendix D

Comprehensive Tables

	Probability of occurrence
when who	0.2146
who when	0.1626
when who where	0.1430
where when	0.1393
where when who	0.1045
who where	0.0813
when who what	0.0715
where who when	0.0696
where what	0.0696
where who	0.0696
who when where	0.0542
when where what	0.0536
when where	0.05365
what where when	0.0487
what where when who	0.0487
what where	0.0487
who what	0.0406
where what when	0.0348
where what when who	0.0348
what when	0.0243
what who	0.0243
what when who	0.0243

Table D.1: Probability of occurrence of patterns of WH-question words for temporal tasks, listed in order.

	Probability of occurrence
who what where	0.2578
who what	0.2578
what who	0.1562
what where	0.125
where who	0.1124
who when	0.0859
who when where	0.0859
where what who	0.0750
where what	0.0750
what where who	0.0416
when where who	0.0312
what when	0.0312
what when who	0.0312
when where	0.0312
when who	0.0312

Table D.2: Probability of occurrence of patterns of WH-question-words for causal tasks, listed in order.

	Average number of times occurred
when who	0.2033
who when	0.1685
who what	0.1225
when who what	0.1220
what who when	0.1008
what who	0.1008
when who where	0.0813
who when where	0.0674
where who when	0.0632
where who	0.0632
what when who	0.0588
what when	0.0588
who what where	0.0525
what where	0.0504
where when	0.0421
where what	0.0421
who what when	0.0350
where when who	0.0316
who where	0.0306
when where	0.0305
when where who	0.0152
when where what	0.0152
where what when	0.0140
where what who	0.0140
where what when who	0.0140
what where when	0.0126
what where who	0.0126
what where when who	0.0126
when what	0.0101

Table D.3: The probability of occurrence for WH patterns across all tasks, sorted based on probability values.