Augmenting Visualizations with Interactive Data Facts to Facilitate Interpretation and Communication

Arjun Srinivasan, Steven M. Drucker, Alex Endert, and John Stasko

Abstract—Recently, an increasing number of visualization systems have begun to incorporate natural language generation (NLG) capabilities into their interfaces. NLG-based visualization systems typically leverage a suite of statistical functions to automatically extract key facts about the underlying data and surface them as natural language sentences alongside visualizations. With current systems, users are typically required to read the system-generated sentences and mentally map them back to the accompanying visualization. However, depending on the features of the visualization (e.g., visualization type, data density) and the complexity of the data fact, mentally mapping facts to visualizations can be a challenging task. Furthermore, more than one visualization could be used to illustrate a single data fact. Unfortunately, current tools provide little or no support for users to explore such alternatives. In this paper, we explore how system-generated data facts can be treated as interactive widgets to help users interpret visualizations and communicate their findings. We present *Voder*, a system that lets users interact with automatically-generated data facts to explore both alternative visualizations to convey a data fact as well as a set of embellishments to highlight a fact within a visualization. Leveraging data facts as interactive widgets, Voder also facilitates data fact-based visualization search. To assess Voder's design and features, we conducted a preliminary user study with 12 participants having varying levels of experience with visualization tools. Participant feedback suggested that interactive data facts aided them in interpreting visualizations. Participants also stated that the suggestions surfaced through the facts helped them explore alternative visualizations and embellishments to communicate individual data facts.

Index Terms—Natural Language Generation; Mixed-initiative Interaction; Visualization Recommendation; Data-driven Communication;

1 Introduction

Recently, an increasing number of visualization tools have begun to incorporate natural language generation (NLG) capabilities into their interfaces. NLG-based visualization systems typically leverage a suite of statistical functions to automatically extract key facts about a visualization's underlying data and surface them as natural language sentences alongside the visualization. In a sense, these sentences complement the visualization by allowing people to verify inferences they derived from the chart or identify data facts they might have missed. Furthermore, the natural language sentences may assist with difficulties in interpreting a visualization resulting from high visual complexity, unfamiliar visualization types, or a person's low visual literacy [15]. Additionally, people may use these system-generated sentences as a way to get a sense of potentially important data facts that can be shown using the visualization.

Since they are expressed as natural utterances, these sentences also enable sharing and communication of data facts using visualizations. For example, consider a visualization like the bar chart of sales by district and a set of associated data facts in Figure 1. A person can simply copy the system-generated data facts s/he finds interesting and send them along with a snapshot of the visualization when communicating with colleagues via email. Alternatively, the visualization and associated data facts can be added to a slide deck and discussed further during a presentation. For the remainder of this paper, we use the term **data fact** to refer to a textual description of the result of one or more statistical functions applied to the data used to create a visualization.

NLG-based visualization systems are still in their infancy, however, and have their own limitations. First, most systems today typically require users to read the data facts and generally lack appropriate means to highlight facts in the visualization. In other words, there is no 'visual' linking between a data fact and the visualization. Without appropriate visual cues to help users 'observe' facts in a visualization, these systems impose upon the user the added responsibility of mentally mapping system-generated data facts back to the visualization. Second, given a data fact, there may be more than one visualization that could be used to illustrate the fact (e.g., a pie chart can also be used to illustrate the last two data facts in Figure 1). Similarly, given a visualization and a data fact, the visualization can be embellished in multiple ways to effectively highlight the fact (e.g., changing opacity, adding stroke, enlarging labels). Choosing the right combination of visualization and embellishments is particularly important when sharing or communicating data facts since this may have a strong effect on the audiences' understanding. Unfortunately, current systems lack support for helping users explore alternative visualizations and embellishments based on data facts they are interested in.

To overcome some of these limitations, we explore how system-generated data facts can be treated as interactive widgets (as opposed to plain text). We hypothesize that such *interactive data facts* may have applications during data exploration as well as to aid people communicate data facts with visualizations. For example, could systems present facts that are related to but not directly observable in a visualization as a way

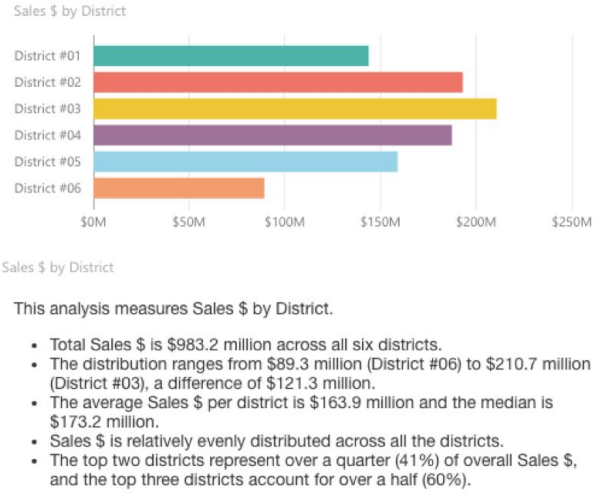

Fig. 1. Result of the Quill NLG plug-in applied to a visualization in Microsoft Power BI. (image source: [2])

- *Arjun Srinivasan, Alex Endert, and John Stasko are with Georgia Institute of Technology. E-mail: arjun010@gatech.edu, endert@gatech.edu, stasko@cc.gatech.edu*
- *Steven M. Drucker is with Microsoft Research. E-mail: sdrucker@microsoft.com*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

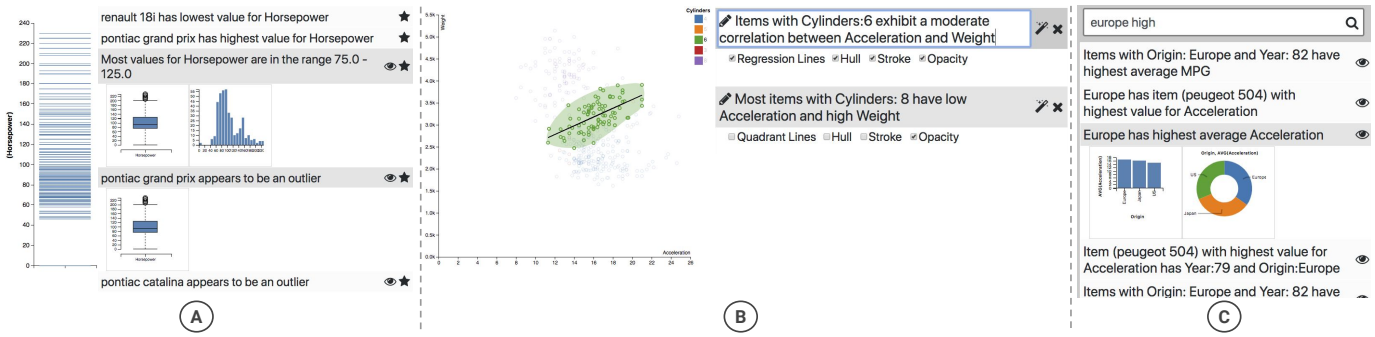

Fig. 2. Examples of interactive data facts being used for: (A) Suggesting alternative visualizations. In this case, a box plot and a histogram are suggested as alternative visualizations for illustrating a distribution-related data fact while only a box plot is suggested to show outliers. (B) Suggesting alternative embellishments. In this case, multiple embellishment options are suggested to highlight a group correlation-based and quadrant distribution-based facts. Interacting with the group correlation fact applies the selected embellishments. (C) Enabling fact-based visualization search. In this case, the search query "*europe high*" results in data facts referring to high values for Europe. Interacting with a data fact shows visualizations that can be used to communicate the fact.

to help users pivot between visualizations showing different aspects of the data (e.g., aggregated vs. data case-level values)? Alternatively, can a data fact be thought of as a point one wishes to communicate with a visualization and correspondingly, be used as a way to suggest possible embellishments that can make it easier to observe the fact using the visualization? To explore the potential of interactive data facts in such cases, we employ them in a prototype of a visualization tool, *Voder*.

Voder helps people explore data through manual view specification, a technique used in well-known visualization systems like Tableau [41]. Once a visualization is created, Voder uses a set of predefined heuristics to generate a list of data facts associated with the specified visualization. As users hover on data facts, parts of the visualization corresponding to the fact are dynamically highlighted. Furthermore, through direct interaction with data facts, Voder allows users to: (1) explore alternative visualizations to illustrate a data fact and (2) explore alternative embellishments to highlight a data fact within a visualization. Voder also lets users issue keyword-based queries to search for data facts. This helps users rapidly search for visualizations to illustrate their intended data facts or identify data facts pertaining to data cases and attributes they are interested in. The primary contributions of our work are twofold:

- We illustrate potential applications of interactive data facts in the context of a manual view specification-based visualization tool to facilitate data exploration. We also show how interactive data facts can be used to suggest alternative visualizations and embellishments to communicate a fact.
- We report our observations from a qualitative user study exploring how participants with varying levels of experience with visualization tools used Voder. Participant feedback indicated that interactive data facts aided them in interpreting visualizations, and that the suggestions surfaced through the facts helped them explore alternative ways to communicate their findings. Furthermore, non-expert users, in particular, stated that they found Voder intuitive to use and said that interactive data facts helped them better understand visualizations.

2 Related Work

2.1 Automated-Insights and NLG-based Visualization Systems

An increased level of interest has recently emerged for exploring how data analysis systems can automatically extract "insights" from a dataset. For example, Tang et al. [42] presented a computation framework to automatically extract *top-k insights* from a dataset. They focused on two types of insights– namely, point insights and shape insights corresponding to outliers and trends, respectively. Foresight by Demiralp et al. [10, 11] is a visual data exploration tool that lets users rapidly explore large high-dimensional datasets by automatically generating different classes of insights including distribution, correlation, and outliers, among others. Visualizations corresponding to various

Fig. 3. Examples of automatically generated insights in Microsoft Power BI (image source: [43]) and Google Sheets

insights are presented as "guideposts" that can be bookmarked and used to explore other similar guideposts. More recently, Cui et al. presented DataSite [9], a visual data exploration tool that proactively generates insights using a library of automatic algorithms. DataSite presents the system-generated insights along with an accompanying visualization to show individual insights as part of a dynamic feed similar to posts in a social media feed. In addition to these systems developed within the database and visualization research community, industry systems such as Microsoft Power BI [28] and Google Sheets [13] also provide similar automatic insight generation features (Figure 3). In the aforementioned systems, a suite of statistical algorithms are executed over a dataset and statistically important results ("insights") are expressed as natural language sentences using basic NLG [46].

A related line of work includes NLG plug-ins like Quill [29] and Wordsmith [49]. These are similar to auto-insight systems in that they also leverage statistical techniques to infer potentially interesting or relevant facts about the data. When coupled with visualization systems, these plug-ins typically use template-based NLG [33, 46] to surface system-generated data facts as natural language sentences alongside user-created visualizations. As opposed to auto-insight systems that largely focus on enabling rapid exploration of insights from the data, such template-based NLG systems are often intended to aid users in interpreting visualizations and using them for communication purposes. In our work, we focus on this latter class of systems and explore how data facts in NLG-based visualization systems can be treated as interactive widgets (referring to them as interactive data facts). We explore how such interactive data facts can aid visual data exploration and help users explore alternative visualizations and embellishments to communicate a data fact.

From a terminology standpoint, since they both stem from a range of statistical functions applied to data, some may refer to sentences in both Figure 1 and Figure 3 as "insights". However, given the subjective nature of the term and the varying definitions of "insight" within the

visualization and visual analytics community [7, 30], as stated earlier, in this paper, we refer to any textual descriptions of data accompanying visualizations as data facts.

2.2 Visualization Recommendation

Several tools have been developed both commercially and within the visualization research community that facilitate visual data exploration by recommending visualizations (e.g., [26, 35, 44, 47, 48]). Such tools typically recommend visualizations based on selected data attributes (e.g., [44, 47]) or a specified visualization (e.g., [26, 48]). While such recommendations help users surmount their "*visual mapping barrier*" and facilitate comparisons between alternative visualizations [15], in current systems, it is left up to the user to examine these alternatives and make inferences from individual visualizations. As stated earlier, making inferences from visualizations may be challenging and is dependent on both the properties of visualization itself and the user's background. Furthermore, depending on the data facts a user is interested in, some visualizations may no longer be appropriate. For instance, a strip plot and a histogram are both fairly standard visualizations recommended for a quantitative attribute [26, 48]. However, if the user is interested in identifying data facts pertaining to individual data cases, the histogram is no longer applicable since it aggregates data cases into bins. Unfortunately, existing systems provide no means for users to express their interest in specific data facts.

As part of our work, we investigate how the system-generated data facts in NLG-based visualization tools can be leveraged to allow users to express their interest via data facts. Furthermore, building upon prior work on task-based visualization recommendation [4, 6, 14], we also explore how visualization systems can leverage the context of data facts to suggest targeted alternative visualizations to illustrate a data fact.

2.3 Embellishing Visualizations For Communication

As stated earlier, most current NLG-based visualization tools require users to read the system-generated text and mentally map it back to the visualization. To help users map system-generated facts to visualizations, some auto-insight systems [9, 10, 43] pre-apply embellishments such as opacity and trend lines to the corresponding visualization. For example, in Figure 3 (bottom-left), the category referred to in the data fact is highlighted in the donut chart to emphasize that the data fact refers to '*Packaging*'. However, in cases where multiple data facts are associated with a single visualization (Figure 1), this approach of pre-applying embellishments would no longer be feasible as different facts may require highlighting different parts of a visualization. A potential strategy to overcome this issue is to enable a dynamic visual mapping between individual data facts and visualizations. For example, Kong et al. [21, 22] have shown how allowing users to interact with textual sentences and highlighting components in visualizations corresponding to a sentence can aid in reading documents containing data-driven graphics (e.g., news articles, reports). Similarly, to facilitate better mapping between a visualization and system-generated data facts, we also enable a brushing-and-linking [3] style interaction between individual data facts and the corresponding visualization.

Visual embellishments (e.g., changing opacity levels, adding stroke, adding labels) help show or highlight specific aspects of a visualization and are particularly helpful when one is using visualizations to communicate data findings [20, 23, 34]. Minimal embellishments such as changing opacity levels might be sufficient to help users interpret data facts associated with a visualization when performing exploratory data analysis. However, when communicating data facts using visualizations, one may want to modify these default embellishments so that it is easier for the audience to understand a data fact in the context of the visualization. Tools like ChartAccent [34] by Ren et al. make it easier for users to add data-driven embellishments (or annotations) to a visualization by providing a suite of embellishment options through a GUI. Unfortunately, these tools do not incorporate the context of data facts a user wishes to communicate with a visualization. Hence, users have to manually select components in the visualization (e.g., data points, axis labels) and explore alternative embellishment options corresponding to their selections. We investigate how visualization systems could leverage the context of data facts to prune the space of possible embellishment options. Given a visualization and a data fact, we explore how systems can automatically present users with a focused set of embellishment options to choose from in order to communicate the data fact with the specified visualization.

3 VODER

In this section, we first discuss the design considerations we had in mind while developing Voder. We then describe Voder's interface and provide the accompanying implementation details.

3.1 Design Considerations

The primary goal of this research was to explore how system-generated data facts in NLG-based visualization tools can be extended to be more than just descriptions of the underlying data. Specifically, we wanted to explore the possibilities that emerge when data facts are treated as interactive widgets (as opposed to plain text). Accordingly, we decided to investigate potential ways in which interactive data facts could be incorporated into a manual view specification-based visualization system. We faced many design decisions while developing Voder. For instance, we wanted to provide comparable capabilities to existing systems like Tableau [41] or PoleStar [47]. However, we also wanted to incorporate system-generated facts without making the interface too overwhelming or requiring users to rely entirely on the presented data facts. To guide the design, we developed a list of considerations on how and why interactive data facts should be incorporated into Voder's interface. These considerations were informed by prior work on exploratory data analysis systems [9, 48], mixed-initiative systems [16], and visualization embellishment [22, 23, 34], then refined through our experiences across multiple design iterations.

DC1. Visually link data facts to visualizations to aid interpretation. As stated earlier, a potential benefit of data facts is that they may help people interpret visualizations or confirm their own inferences. Accordingly, once a visualization is specified, the system should explicitly show data facts that correspond to the specified visualization. However, reading these facts and mentally mapping them back to the visualization may be non-trivial or even challenging. To aid in "observing" data facts in a visualization, similar to prior work by Kong et al. [22], the system should employ interactive visual linking between individual facts and the corresponding elements of the visualization.

DC2. Suggest more detailed or aggregated data facts to aid exploration. While scanning data facts based on a visualization, it may help users to also have access to other related facts. These facts may refer to more aggregated or more detailed aspects of the data being visualized, thus allowing users to explore the visualized data at multiple abstraction levels. To enable this, in addition to presenting facts directly mapping to a given visualization, the system should also present data facts pertaining to transformed (e.g., sum, average) variations of the active visualization.

DC3. Facilitate exploration of alternative ways to communicate data facts:
(a) Show alternative visualizations for a data fact. More than one visualization could be used to illustrate a data fact. The system should enable exploration of possible visualizations that could be used to show a fact by allowing users to directly interact with the data fact itself.
(b) Show alternative embellishments for a data fact. Given a visualization and a data fact, the visualization could be embellished in multiple ways to highlight the fact. In addition to applying default embellishments such as a change in opacity, given a visualization and a data fact, the system should allow users to explore other embellishment options by interacting directly with the data fact.

DC4. Facilitate data fact-based visualization search. Another potential application of system-generated data facts is to facilitate visualization search. Rather than visually exploring data to surface data facts, users may want to directly find visualizations corresponding to specific aspects of the data they are interested in. To enable this, the system should allow users to search for data facts based on data cases,

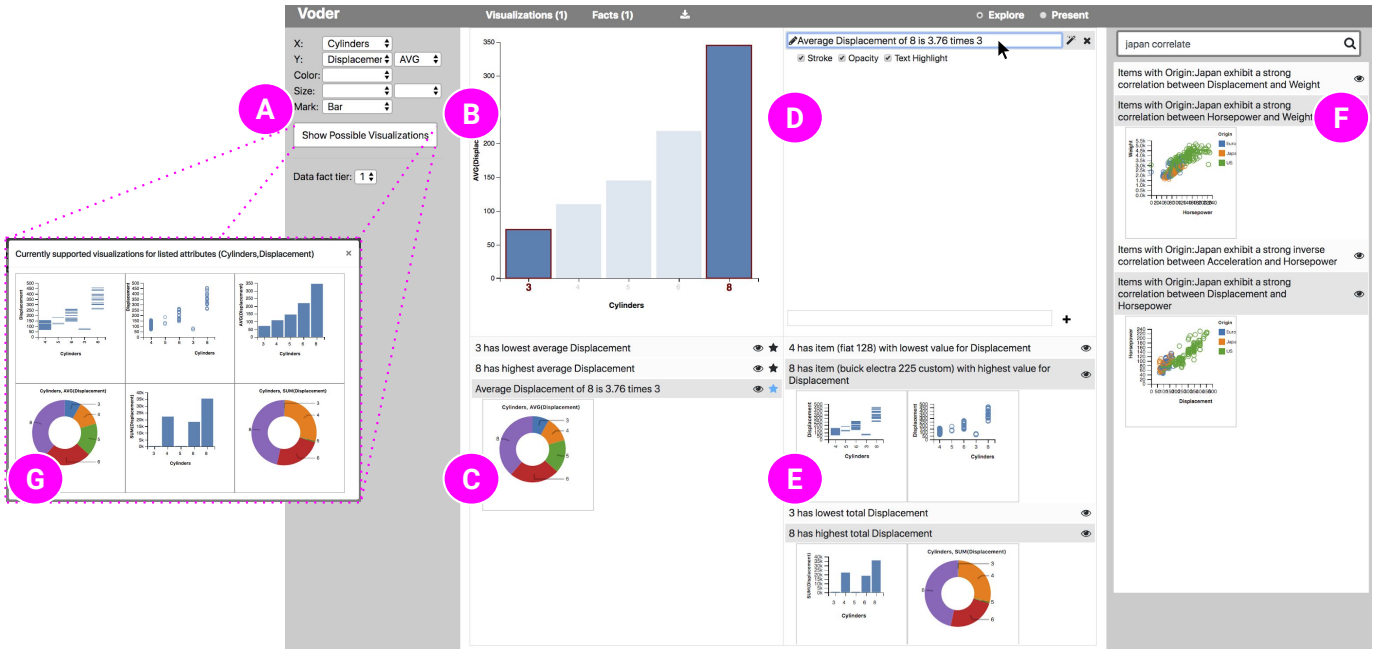

Fig. 4. Voder's user interface in the 'explore' view. (A) Visualization specification and data fact tier selection, (B) Active visualization, (C) System-generated data facts observable in the active visualization, (D) Starred data facts for the active visualization, (E) System-generated data facts related to a different configuration of the active visualization's attributes, (F) Data fact query panel, and (G) Supported visualizations that would be shown for a partial specification consisting of *Cylinders* and *Displacement* attributes. In this case, the user is hovering the cursor over a starred data fact in (D), highlighting it in the active visualization (B).

data attributes, or even specific types of data facts (e.g., correlation between attributes). The data facts can then, in turn, be used to identify potentially relevant visualizations.

3.2 User Interface

We now provide an overview of Voder's user interface (Figure 4). To give a sense of how one might use the system, we contextualize the interface features under three high-level activities— namely, exploring data facts, exploring alternative ways to present a fact, and collating a bookmarked set of facts into a slideshow or dashboard. More details regarding how data facts are generated and ordered, how alternative visualizations and embellishment options are suggested, and how search queries are processed are described in later sections.

3.2.1 Exploring Data Facts

As stated earlier, we developed Voder to explore how interactive data facts can be incorporated into popular visualization tools like Tableau [41]. Accordingly, Voder lets users create visualizations using manual view specification, a technique commonly used in many visualization tools today (e.g., [9, 41, 48]). Users construct visualizations using the visualization specification panel (Figure 4A). The panel provides dropdown menus to map attributes to different encoding channels (*X*, *Y*, *Color*, *Size*), select transformation functions (*count*, *average*, *sum*, *bin*), and select a mark type (e.g., *bar*, *point*). While the manual view specification approach provides flexibility and lets users tweak data transformations and visualizations, it can be challenging for less experienced users to fully specify visualizations [15]. As we expected to evaluate the tool with users having varying levels of experience with visualizations, similar to existing tools [41, 48], we also let users provide partial specifications. Users can populate one or more dropdowns with attributes they are interested in and click the "*Show Possible Visualizations*" button. In response, a modal window appears showing all supported visualizations based on data transformations and mark type variations for the specified attributes (Figure 4G). Users can click a visualization to automatically specify it.

Each time a visualization is specified, Voder automatically generates data facts based on the active visualization (Figure 4B). Data facts appear below the visualization in the active view data facts panel (Figure 4C) (**DC1**). Voder also generates data facts that are related to the specified attributes but are not directly observable in the active visualization. For example, in Figure 4, the active visualization is a bar chart showing the average engine *Displacement* of cars by the number of *Cylinders*. However, if users wanted to observe a data fact pertaining to a specific car as opposed to an average of all cars with a certain number of cylinders, they would need to change the active aggregated visualization to an unaggregated one. Such related data facts are shown separately in the related data facts panel (Figure 4E) (**DC2**).

While reading facts in the active view data facts panel, users can hover the cursor over individual facts to highlight them in the active visualization (**DC1**). If they find a fact interesting, users can save it against a visualization by clicking the ★ icon next to the fact. Saved facts for a visualization are shown to the right of the active visualization (Figure 4D). There may be cases where users identify something about a visualization that they find interesting but their observation is not captured by the system-generated data facts. For such cases, or for simple note taking purposes, Voder also lets users manually add textual statements associated with an active visualization (Figure 4D, bottom). A ■ icon is added to manually entered statements to indicate that they are user-specified sentences and not system-generated data facts.

Instead of specifying visualizations and exploring data facts associated with them, Voder also lets users directly search for data facts and use them as a starting point to specify visualizations (**DC4**). With the data fact query panel (Figure 4F), users can search for relevant data facts using keyword-based queries (e.g., "*bmw*", "*weight correlation*").

3.2.2 Exploring Alternatives to Present Data Facts

As users explore system-generated data facts, an 👁 icon attached to a data fact in Figures 4C,E,F indicates that a data fact can be clicked on to see other visualizations that can be used to observe the fact (**DC3a**). The absence of an 👁 icon indicates the active visualization is the only available visualization that can be used to convey a data fact (e.g., first two facts in Figure 2A).

Saved facts associated with an active visualization can be further edited for presentation purposes. Users can directly edit the displayed text (✎) in Figure 4D by clicking on the data fact. A 🪄 icon next to a data fact indicates that the fact can be clicked on in order to see embellishment options that can be configured to highlight the fact in the active visualization (**DC3b**). Expanding a fact in Figure 4D shows

the available embellishment options as check-boxes. Users can apply or remove embellishments by toggling the check-boxes. Changes to embellishment options are applied immediately and are reflected in the corresponding visualization whenever the user hovers the cursor on the data fact in Figure 4C or 4D.

3.2.3 Organizing Saved Visualizations and Data Facts

Once they have saved one or more visualizations and corresponding facts, users can switch to Voder's 'Present' view. In this view, saved facts and visualizations can be explored in a slide-like layout showing individual visualizations accompanied by their corresponding data facts and user notes (Figure 5, top). Users can continue to add/remove embellishments and edit the textual descriptions of data facts. To view data facts as bullet points (as opposed to the editable widgets), users can turn off the edit option. Turning off the edit option displays the data facts as plain text. Users can also choose between showing facts as bullet points or combine them into a paragraph format. Even with the edit mode turned off, hovering the cursor on data facts still highlights them in the corresponding visualization. Instead of the slide layout, users can also choose a dashboard layout that groups all visualizations and data facts separately (Figure 5, bottom). Once they are done editing data facts and embellishments, users can export the contents of the slide or the dashboard layout as an interactive HTML page (⬇).

3.3 Generating and Ordering Data Facts

Voder currently uses a set of predefined heuristics to generate potentially useful data facts associated with a specified visualization. The data facts in Figure 1 and Figure 3 can be mapped to one or more low-level analytic tasks such as those from Amar et al.'s taxonomy [1]. For example, consider two of the data facts in Figure 1 "*Total Sales % is $983.2 million across all six districts*" and "*Sales $ is relatively distributed across all districts*". Mapping these to Amar et al's [1] taxonomy, the first data fact maps to the *derived value* task whereas the second maps to the *characterize distribution* task. Based on this similarity between analytic tasks and data facts, we defined a basic set of heuristics to generate different types of data facts. Table 1B summarizes the task categories we currently support (based on [1]) along with the criteria specified to generate the corresponding data fact.

In addition to the four unique task categories (*Find Anomalies*, *Correlation*, *Characterize Distribution*, *Find Extremum*) in Table 1B, we also cover *Derived Value* and *Filter* tasks. However, we do not list these exclusively in Table 1B since they are not used individually to generate facts but are used in conjunction with the other listed tasks. For example, the fact "*Average Retail Price of SUV is 1.76 times Sedan*" inherently includes a *Derived Value* task (average) in addition to *Characterize Distribution*. Similarly, a fact like "*Items with Origin: Japan exhibit a strong correlation between Horsepower and Weight*" inherently includes a *Filter* task (Origin: Japan) in addition to *Correlation*.

Our choice of which data fact types to support was primarily based on observing the types of facts generated by existing NLG plug-ins like Quill [29] and Microsoft Power BI's insights feature [43]. As listed in [43], some examples of these categories of data facts include *Category outliers*, *Correlation*, *Major factors*, and *Steady share*, among others. With respect to analytic tasks the facts could be mapped to, we considered multiple visualization and analytic task taxonomies (e.g., [1, 5, 37, 45]) but finally decided to use to Amar et al.'s [1] taxonomy. We chose this task taxonomy because the tasks listed were most similar to data fact categories in existing systems (e.g., *Correlation* in both Power BI and Amar et al's taxonomy, *Steady share* in Power BI [43] ∼ *Characterize Distribution* in Amar et al's taxonomy).

Depending on the number of data cases and the variation in values, there may be scenarios where these heuristics result in multiple data facts being generated for the same task category. For instance, many outliers may exist in a visualization of a quantitative variable. Or in the case of a bar or donut chart with several categories, multiple facts pertaining to relative distributions between category values may be generated. Displaying all generated data facts simultaneously could make it difficult for users to scan through facts and interpret the visualization. Accordingly, we adopt a strategy similar to Quill's [29]

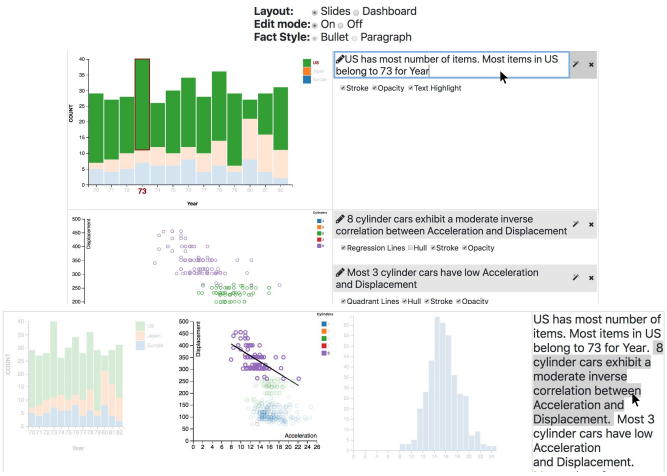

Fig. 5. Saved visualizations and data facts shown in the Present View using the slide layout (top) and the dashboard layout (bottom).

'verbosity' management feature and group data facts into three tiers (*tier 1*, *tier 2*, *tier 3*). Tier 1 consists of the most prominent data facts (e.g., highest/lowest values, top outliers), tier 2 consists of facts that may be important but are not the most extreme cases (e.g., second highest/lowest values), and tier 3 consists of all the remaining data facts generated based on the criteria in Table 1B. In cases where only one data fact can be generated (e.g., range of values), the data facts are added to tier 1. By default, Voder only shows tier 1 data facts. Users can choose the data fact tiers via the dropdown menu below the manual view specification panel (Figure 4A). Note that the data fact tiers are additive (i.e., selecting tier 2 also shows tier 1 facts). This approach ensures that the most prominent facts (tier 1) are always shown and users can request for additional facts on-demand.

By default, data facts in Voder's interface are grouped by task categories. For instance, in the example shown in Figure 2A, extreme value related data facts are followed by distribution based data facts which are then followed by data facts about outliers. However, once one or more data facts are starred, Voder keeps track of the task categories the starred facts belong to. Thereafter, Voder orders the data facts in both Figures 4C and 4E based on the task categories of starred data facts.

3.4 Mappings Between Data Facts, Visualizations, and Embellishments

In our current prototype, we implement a basic set of charts to visualize numerical and categorical attributes. Table 1A summarizes the attribute combinations and corresponding visualizations Voder currently supports for each combination. Table 1A also shows the tasks and corresponding data facts that map to each visualization, as well as the embellishment options provided to highlight a fact in a visualization. We developed the mappings and heuristics in Table 1 through iterative informal feedback from fellow researchers and students with varying levels of experience with visualizations. This feedback was gathered using an initial version of Voder's interface that only consisted of a specification panel and a set of system-generated data facts corresponding to the specified visualization (Figure 4A,B,C). The facts were generated based on a set of heuristics we defined to cover the different categories of facts shown in existing systems [9, 10, 43]. Individuals were asked to state which facts they thought were useful as system suggestions. We also encouraged them to mention any additional types of facts that they would have liked the system to show. This initial review and feedback led us to discard some initial types of data facts that people felt were not very useful to have as system-generated statements. For example, we initially had data facts corresponding to the *Determine Range* task [1] listing the min/max values for numeric attributes (e.g., "*Acceleration has values in the range 8-24.8*"). Similarly, we also excluded plain *Derived Value* facts like "*Average MPG across Origins is 23.47*". The feedback also helped us add certain types of data facts that we initially did not consider. For instance, data facts pertaining to quadrant-based

Attribute Combination	Example Data Fact	Task(s)	Visualization	Embellishments						
				O	S	IL	TH	RL	CH	QL
N	Pontiac Grand Prix has highest value for Horsepower	Find Extremum	Strip plot	■	■	■				
	Most values for Horsepower are in the range 75 - 125	Characterize Distribution	Strip plot	■	■					
			Box plot	■	■					
			Histogram	■	■					
	Pontiac Catalina appears to be an outlier	Find Anomalies	Strip plot	■	■	■				
			Box plot	■	■	■				
C	Europe has the least number of items	Find Extremum	Bar chart	■	■		■			
	Number of items in US is 2.57 times the number of items in Europe	Characterize Distribution	Donut chart	■	■		■			
N x N	Acceleration and Displacement have a strong inverse correlation	Correlation	Scatterplot					■		
	Most items in the dataset have high Horsepower and low MPG	Characterize Distribution		■	■	■				■
C x N	Average Retail Price of SUV is 1.76 times Sedan	Characterize Distribution (+Derived Value)	Bar chart Donut chart	■	■	■				
	Japan has highest average MPG	Find Extremum (+Derived Value)		■	■	■				
	Europe has item (Fiat 128) with lowest value for Displacement	Find Extremum	Strip plot Scatterplot	■	■	■				
C x C	The largest group of items in the dataset have Origin:Europe and Cylinders:5	Find Extremum	Stacked bar chart Scatterplot + Size	■	■	■				
	US has most number of items. Most items in US belong to 8 for Cylinders	Characterize Distribution (+Find Extremum)		■	■	■				
N x N x N	Most items with low MPG and low Weight also have low Horsepower	Characterize Distribution	Scatterplot + Size	■	■	■				■
C x N x N	Overall, Displacement and Weight have a strong correlation	Correlation	Scatterplot					■		
	Items with Origin:Japan exhibit a strong correlation between Displacement and Weight	Correlation (+Filter)	Scatterplot + Color	■	■			■	■	
	Most items with Origin: Europe have low Displacement and low Weight	Characterize Distribution (+Filter)	Scatterplot + Color	■	■				■	■
			Scatterplot + Size	■	■	■				
C x C x N	datsun 1200 with lowest value for Weight has Cylinders:4 and Origin:Japan	Find Extremum	Strip plot + Color Scatterplot + Color	■	■	■				
	Items with Origin: Japan and Cylinders: 4 have lowest AVG(Weight)	Find Extremum (+Derived Value)	Strip plot + Color Scatterplot + Color	■	■	■				
			Scatterplot + Size	■	■	■				

(A)

Task: Find Anomalies
Example: pontiac catalina appears to be an outlier
Generation criteria: value<*Q1*-1.5**IQR* or value>*Q3*+1.5**IQR*
(where *Q1*, *Q3* are first and third quartiles, and *IQR* is the interquartile range)

Tier 1: Top 2 outliers, **Tier 2:** 3-5 of top 5 outliers

Task: Correlation
Example: Acceleration and Displacement have a strong inverse correlation
Generation criteria: $r < -0.5$ or $r > 0.5$
(where r is Pearson's correlation coefficient)

Tier 1: Strong correlations ($r < -0.7$ or $r > 0.7$),
Tier 2: Moderate correlations ($r < -0.5$ or $r > 0.5$)

Task: Characterize Distribution (relative values)
Example: Average Displacement of 8 is 3.76 times 3
Generation criteria: value1 >= 1.5*value2
(where values correspond to two categories)

Tier 1: Pair with maximum difference,
Tier 2: 2-3 of top 3 difference pairs

Task: Find Extremum
Example: Europe has the least number of items
Generation criteria: MIN, MAX

Tier 1: Data cases with max/min values,
Tier 2: 2-3 of top and bottom 3 values

Task: Characterize Distribution (common range of values)
Example: Most values for Horsepower are in the range 75 - 125
Generation criteria: Q1-Q3
(where Q1, Q3 are first and third quartiles)

Task: Characterize Distribution (quadrant-based)
Example: Most items with Origin: Europe have low Displacement and low Weight
Generation criteria: >75% of items in one quadrant
(where quadrants are based on mid-points of range of values for an attribute)

(B)

Table 1. (A) Currently supported attribute combinations (**N**: Numeric, **C**: Categorical) along with corresponding data facts, analytic tasks from [1], visualizations, and embellishment options. **O**: Opacity (highlight/fade marks), **S**: Stroke (add a boundary around marks), **IL**: Item Label (add an item label), **TH**: Text Highlight (highlight a label in axis or a color legend), **RL**: Regression Line (add regression line in a scatterplot), **CH**: Convex Hull (draw a hull around points in a scatterplot), **QL**: Quadrant Lines (show lines to divide a scatterplot into four regions). A blue cell indicates an embellishment is suggested for the combination of a data fact and visualization corresponding to that row. A black stroke around a cell indicates embellishments that were applied by default. (B) Heuristics applied to generate data facts.

distribution in scatterplots emerged as a result of the feedback sessions. To create the final list of data facts and mappings in Table 1, we used an affinity diagramming approach to identify the most common data facts for individual visualization types across users with varying levels of experience with visualizations.

In terms of the embellishments, by default, Voder applies opacity (for all facts) and regression lines (for correlation facts) to highlight facts in a visualization. In other words, unless toggled on by the user, embellishments such as the stroke and category label highlight in Figure 4 B or the hull and stroke in Figure 2B would not appear when hovering on the data fact. The reason for this choice of default embellishments was twofold. First, existing auto-insight tools [9,43] primarily only use embellishments like opacity and trend lines as predefined ways to highlight facts in visualizations (e.g., Figure 3, bottom-left). Accordingly, we wanted to ensure that we provide comparable capabilities. Second, we initially implemented Voder to show all embellishments by default. However, feedback during pilot studies indicated that users preferred adding embellishments on-demand as opposed to removing them in cases where they felt the multiple embellishments were overwhelming.

Note that Table 1 is by no means intended to be exhaustive, nor is our goal to provide a definitive set of mappings and heuristics to generate data facts. These are merely a preliminary set of mappings and heuristics we defined in order to develop a prototype and test if the idea of treating data facts as interactive widgets has merit.

3.5 Keyword-based Search Queries

As stated earlier, Voder lets users flexibly search for data facts by issuing keyword-based queries in the data fact query panel (Figure 4F). Queries can not only include data cases, attributes, and values, but also general words like 'low', 'outlier', 'correlate', 'range', 'compare' etc., that may map to analytic tasks listed in Table 1. For example, as shown in Figure 2C, a query like "*europe high*" results in all data facts pertaining to high values for *Europe* (a data case for the attribute *Origin*). Similarly, a query like "*MPG correlation*" would result in all data facts highlighting strong and moderate (depending on the active data fact tiers) correlation-related facts corresponding to the attribute *MPG*. Furthermore, Voder also has built-in mechanisms to check for both syntactic (e.g., misspelled words) and semantic (e.g., synonyms) word matches. Similar to recent natural language interfaces for visualization [12,36,40], we use the cosine similarity [24] between a keyword and the target string to check for syntactic matches and the Wu-Palmer similarity score [50] when checking for semantic matches.

The ability to issue flexible keyword-based queries enables at least three use cases during visual data exploration. First, in cases where users have a sense of the data cases, attributes, or types of facts they are looking for, they can seed their visualization search with this a priori knowledge. Second, the query feature can be used for rapid hypothesis testing. For example, if users wanted to see if *MPG* and *Horsepower* were correlated, they could simply type in a query like "*MPG horsepower correlate*" and check if the system returns any results. Third, if a user identifies a data case of interest while exploring the data, the query feature makes it easy to rapidly find other facts and visualizations pertaining to the data case. For example, in the case of the cars dataset, looking at a box plot for *Acceleration*, a user may identify that the car *peugeot 504* appears to be an outlier. To see other facts related to the car, the user can simply issue a query with the car name to get other relevant data facts for the *peugeot 504* and visualizations corresponding to those facts.

4 Preliminary User Study

We conducted a preliminary user study to understand if Voder helps people explore their data and communicate their findings with visualizations. More specifically, we had two goals in mind while conducting the study: (1) assess whether the generated data facts aid users in inter-

preting visualizations during exploration, and (2) assess whether the visualization and embellishment suggestions surfaced via data facts aid users in exploring alternative ways to communicate their findings. Given these goals, we decided that an open-ended task that required participants to explore their data to discover facts and then communicate their findings would be most suited for the study. We felt an open-ended task like this would allow us to observe user interactions with and reactions to the various system features, and also gather useful subjective feedback. Accordingly, we asked participants to use Voder to explore a dataset with the intention of creating a slideshow or dashboard to communicate their findings. We explicitly asked for the latter in order to accomplish our second goal of assessing the utility of suggesting visualization and embellishment alternatives via data facts.

4.1 Participants and Experimental Setup

We recruited 12 participants (three females) between 23-40 years of age. To see if there were any differences in usage patterns based on prior experience levels with visualization tools, we recruited three groups of participants (four in each group): experts (P1-P4), intermediate-level users (P5-P8), and novices (P9-P12). Expert users were analysts (P1, P3) or product managers (P2, P4) at Microsoft. They were well acquainted with visualizations and frequently used visualization tools as a part of their work to explore data and create reports or dashboards to share their findings. Three experts (P1, P3, P4) were even aware of NLG-based visualization plug-ins like Quill [29], with P1 and P3 having prior experience of using the plug-in. Intermediate-level users were students who were currently enrolled in or had taken a graduate-level data visualization course. The course required them to use Tableau as the primary visualization software. Consequently, they were comfortable with creating basic visualizations with Tableau. However, none of participants considered themselves experts at visualizations or frequently used visualizations outside the course. Lastly, the novice user group composed of three graduate students and a software developer (P9), none of them having any prior experience with visualization tools. Two novices (P11, P12) did have some prior experience of performing data analysis with Python, however.

All 12 participants interacted with Voder running on Google Chrome on a 15-inch laptop screen set to a resolution of 1920 x 1080. An external mouse was used for all sessions. Sessions were conducted in-person at the participants' offices or universities in quiet meeting rooms. Participation in the study was voluntary and participants were not financially compensated for their time. Recruitment emails were sent out to mailing lists and interested participants were recruited on a "first come first serve" basis.

4.2 Procedure

Sessions lasted between 50-70 minutes. Participants were first given an introduction to Voder's interface and features (∼10 min). Participants were then allowed to try out the system and were encouraged to ask any questions they had regarding the tool and its usage (∼10 min). Introduction and training was conducted using a dataset about cars[1]. Next, participants were given a dataset of 1300 US colleges with 18 attributes per college[2]. These included categorical and numerical attributes such as *Region*, *Control*, *Median Debt*, and *Average Faculty Salary*, among others. A dataset summary consisting of attribute names, corresponding data types, and a sample range of values were provided separately as a printed table. None of the participants had encountered the dataset before. The task was fairly open-ended: participants were asked to explore the dataset using Voder with the goal of creating a presentation (slide deck) or dashboard to communicate their findings (∼30 min). For cases where participants did not find the system-generated data facts useful but still wanted to save a visualization, we encouraged participants to manually add notes corresponding to what they were trying to communicate with the visualization. Participants were also asked to verbally "present" their findings towards the end of the session using their saved visualizations, data facts, and notes.

[1]Cars dataset available at: https://goo.gl/9G1egz

[2]Colleges dataset available at: https://goo.gl/hqp3HW

An experimenter (either the first or second author) observed each session and took notes. Participants were encouraged to think aloud and interact with the experimenter throughout the session. All sessions were screen-captured and audio-recorded for later review. At the end of each session, participants also completed an exit questionnaire and short interview in which we asked about their experiences with the tool and feedback on specific features of the system (∼10 min).

4.3 Participant Feedback

4.3.1 Usage Overview

A total of 86 visualizations and 119 data facts corresponding to those visualizations were saved as part of the final slide decks or dashboards. Individual sessions resulted in between 4-12 visualizations and 4-17 data facts. Of the 119 total facts, 17 (14%) were user-entered facts (as opposed to system-generated). Among the 17 manually entered facts, six were facts comparing ranges of individual categories in a strip or scatterplot visualizing a [Categorical x Numerical] attribute pair, seven were facts about observed outliers in a scatterplot, and four were cases were participants felt the system should have generated a fact regarding a correlation but did not do so. Of the 102 system-generated data facts that were saved, participants further modified the default embellishments for 70 facts (69%). However, participants explored possible embellishments (i.e. clicked to see available embellishments) for 85 of the 102 starred facts (83%). Nine participants (including all four novices) used the search feature at least once. The total number of search queries across sessions was 31 (1-9 per session). Among these, there were 18 instances (58%) where participants expanded a fact shown as part of the search result.

4.3.2 Using Data Facts to Aid Visual Analysis

Overall, participants felt that the system-generated data facts were useful and often helped them identify notable points about a visualization's underlying data. Non-expert users particularly said that the facts were largely similar to ones they would typically look for in a visualization. Six participants (P4, P6, P7, P8, P11, P12) said that the simpler facts such as extremes in a bar chart were particularly useful in saving time. For instance, P8 said "*even if the system just showed these and other basic facts, that's still very useful. Especially when there are many values close to each other, I don't have to spend time inspecting the visualization.*" However, as we expected, there were varied opinions regarding the complexity of data facts among experts. For example, one expert (P2) felt the presented facts were okay as quick highlights about the visualization's data but did not provide deep "insights" resulting from more sophisticated analytic functions. Another expert (P4) said she found the system-generated facts to be useful and sufficient because she did not know anything about the dataset. However, if she was to use the tool to explore a dataset she was more well versed in, she would like the system to highlight facts more unique to the dataset. Four participants (P3, P7, P10, P11) explicitly stated that they liked the related data facts in Figure 4E. Participants perceived the system presenting facts based on aggregated/unaggregated versions of their current visualizations as intelligent behavior as they felt the system was anticipating their questions. For example, when the system suggested the fact "*New England has lowest average Admission Rate*" in the related data facts panel while P11 was looking at a strip plot of admission rates for individual colleges by region, he reacted saying "*wow that's exactly what I was thinking of figuring out next.*"

Novices and intermediate-level users particularly also stated that the interactive data facts in Voder helped them better understand visualizations and the types of inferences they could make using a visualization. For instance, P11 said "*It's almost like this tool is training me by showing facts based on a visualization. Now I can use this the other way around like if I wanted to show a fact, I know which visualization I need to check.*" Talking about the separation of directly observable (Figure 4C) and related (Figure 4E) data facts and visualization suggestions associated with those, P10 said "*the facts definitely helped me understand the key takeaways from the chart. Another thing I liked was that you had primary bookmarks and secondary bookmarks* (data facts) *This helped me understand related things that I couldn't show with one*

visualization but then it let me find other visualizations that could show the secondary bookmarks. After some point, I was able to predict the chart I needed to use even before I saw the suggestions." P7 indicated that data facts helped him understand more about mappings between data and visual channels. He said "*in a manner of speaking, facts made me more aware of the type of data I was dealing with... Towards the end I was, like, okay this attribute has a list of values per college and this attribute has a list of values per college so I can actually compare them using that colored point graph*" (referring to a colored scatterplot visualizing a combination of [Categorical x Numerical x Categorical] attributes showing average values for the numerical attribute.)

Interactive Highlighting. All participants found the brushing-and-linking style interaction between facts and the visualization helpful. For instance, P6 stated "*the dynamic highlighting made it easy to smoothly scan through facts in the visualization*". Referring to charts like strip plots or scatterplots showing individual colleges, six participants (P3, P4, P6, P7, P8, P10) also mentioned that highlighting via facts was particularly useful in these cases since the visualizations had cluttered or overlapping marks making it difficult to examine individual items.

4.3.3 Interactive Data Facts For Communication

Alternative Visualization Suggestions. There were varying opinions regarding data facts presenting multiple possible visualizations to illustrate a data fact. For example, some experts (P2, P3) felt the system should automatically select the best visualization (i.e., only suggest an alternative if it the system thinks it is better than the active visualization). On the other hand, another expert (P1) said "*it's nice to have these options show up even if I don't always pick them. It makes me more aware of the possibilities.*" This line of thought was echoed by the fourth expert (P4) as well as non-expert users. For instance, P4 said "*it's helpful to have these especially when I'm going into a customer presentation. It shows me if there's a simpler chart I can use and that is always good when dealing with customers.*"

Alternative Embellishment Suggestions. Voder's suggestion of alternative embellishment options via data facts received positive feedback from all participants. Participants generally modified embellishments for unaggregated strip plots, scatterplots, stacked bar charts, and donut charts. In most cases, participants found the default embellishments (opacity) to be sufficient for bar charts and aggregated scatterplots. Some novices (P9, P10) felt the system could automatically add more enhanced embellishments for some chart types instead of providing the minimal defaults. However, both P9 and P10 mentioned that they would still like to have control over the embellishment options just in case the system defaults are overwhelming or unclear. In line with our initial feedback sessions and pilot studies, most participants stated they liked Voder's minimal defaults approach towards embellishments. P6, for instance, said "*it was nice to have the system consistently show facts in the visualization by fading things out. Since getting to possible styling options was easy, I could simply go in and format a chart further when I wanted to.*" None of the participants thought it would be better to have all embellishments applied by default. In the final results, there were only 14 facts out of 102 (14%) for which participants applied all supported embellishments.

Interactive Highlighting. Participants were very enthusiastic about the ability to interactively embellish visualizations by hovering on facts while they presented their findings. For example, P2 said this feature was a "*clear winner*" and would be great to have in reports and dashboards generated in Microsoft Power BI. Another participant (P7) stated "*I love this! This is great because now I can show the whole visualization and then focus on parts of the chart I'm talking about.*"

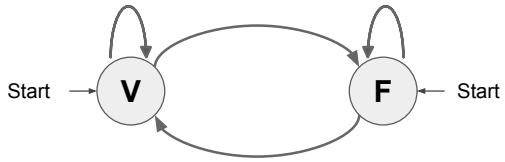

Fig. 6. Transitions between visualizations (V) and data facts (F) during data exploration with Voder.

5 Discussion

5.1 Iterating between Visualizations and Facts during Exploration

During the study, we observed that interactive data facts enabled different strategies for visual data exploration. Observed patterns during the user study are represented as transitions between visualizations (**V**) and data facts (**F**) and are summarized in Figure 6.

With Voder, users can begin exploring data by manually specifying a visualization. Starting with a visualization is the common strategy supported in most existing tools and was adopted by nine participants during the study. This strategy is useful especially when users do not have prior hypotheses or questions, and lets them explore their data using one or more attributes at a time. Once they specify a visualization, users can then leverage Voder's system-generated data facts related to the visualization to make inferences about the underlying data (**V**→**F**). Examples included participants bookmarking facts in Figure 4C or scanning through other related facts in Figure 4E. While scanning through facts, users can click on a fact to see possible visualizations that can be used to illustrate it. By clicking a visualization thumbnail, users can then specify one of these as their active visualization (**F**→**V**). For example, by clicking the second data fact in the related data facts panel in Figure 4E, one could replace the active bar chart with a scatterplot. Alternatively, instead of going through the data facts, users can directly switch between different visualizations for the attributes in the current specification using the "*Show Possible Visualizations*" button (**V**→**V**).

As opposed to beginning by specifying a visualization, users can take a different approach altogether and start by querying for data facts directly. If they find one of the resulting facts interesting, users can expand the fact to see corresponding visualizations and specify them directly (**F**→**V**). During the study, three participants (P3, P5, P9) began exploring data using Voder's query feature. Starting with data facts enables a "start with what you know" or a "find a visualization to see what you want" strategy. For instance, P9 started the session with a query "*Private For-profit Median Debt*" to look for visualizations highlighting *Mean Debt*s for *Private For-profit* (data case for the *Control* attribute) colleges. When asked why he started with the search, he said "*looking at the data summary I felt it's better to directly start with something that I care about as opposed to trying to create different visualizations and find facts.*" The query feature also enables rapid exploration of data facts based on knowledge gained via other system-generated data facts (**F**→**F**). We observed this transition in the case of four participants (P1, P4, P8, P11). For example, P4 identified "*Southern California Institute of Architecture*" as an outlier for *Median Debt* based on a system-generated data fact. To quickly see other facts pertaining to the college, she typed in the college name in the query box and got a list of facts highlighting it as an outlier and extreme value for some other attributes. This made her note the college as an important data case to investigate further. In fact, in the case of P1 and P11, while they both started with a visualization, once they began using the search feature, they changed their exploration strategy mid-session, performing more **F**→**F** and **F**→**V** transitions.

Drawing an analogy to Pirolli and Card's sensemaking loop [32], these transitions illustrate how the combination of visualizations and interactive data facts let users adopt both bottom-up and top-down strategies, even allowing them to switch between the two.

5.2 Potential Risks: Trust and Deception

One of our observations during the study pertained to the level of trust some participants placed on the system-generated data facts. For instance, three participants (P5, P6, P10) frequently skipped visualizations for which the system did not yield any facts (e.g., scatterplots without notable correlations). They said that after creating the first two or three visualizations and looking at the system-generated facts, they felt they could rely on the system to tell them if there was anything useful to note. On the other hand, looking at correlation-based data facts, one expert (P3) said that he would have liked to see the metric

used to determine the correlation. He also said he would possibly even want the ability to "define" what moderate and strong correlations are, indicating that he did not feel comfortable accepting the system defaults. In contrast, five non-expert users (P6, P7, P9, P11, P12) said that they liked that the system did not give them complex statistical values and kept things simple. These mixed reactions suggest exploring design alternatives such as the use of embedded configurable widgets as a part of a data fact's text [18, 38] as an open direction for future work.

Overall, these comments and observations highlight that an important consideration when designing NLG-based systems like Voder is that of trust. While systems need to ensure that they abstract out low-level details and make information easy for users to consume, they also need to provide users effective means to understand the system's reasoning for generating content. Furthermore, data facts do not incorporate domain knowledge and are generated based on heuristically-defined statistical functions. Correspondingly, by relying solely on system-generated facts, users may overlook their own domain knowledge and thus overlook facts they may have found interesting otherwise. Hence, it is also important that systems clearly indicate that the generated content is not exhaustive. Regardless of the suite of algorithms used, systems should not only facilitate but also encourage users to incorporate external information based on any additional inferences they make and not rely entirely on system results for decision making.

With the query feature, Voder lets users directly find facts they want to show about a dataset. Combined with the suggestions of all possible visualizations to illustrate a fact and embellishments to highlight a fact in a visualization, Voder gives users tools to communicate their desired data facts. During the exit interview, P7 stated "*you're allowing for people to see it the way you intend your presentation to be seen and I like that*" indicating that he liked the ability to select a visualization of his choice and add multiple embellishments to highlight data facts. While this was said in a positive sense, the comment also highlights a potential risk with systems like Voder. While our intention of providing system-generated data facts and communication alternatives was to help users make informed choices, users may also unintentionally (or intentionally) select a highly embellished version of a less suited or a potentially "deceptive visualization" [31] to communicate a fact. As recently highlighted by Correll and Heer [8], an important consideration is how to prevent or at least make users (and audiences) aware of potentially deceptive visualizations being used to communicate a fact.

6 Limitations and Future Work

Limitations of the current user study. The qualitative study with the fairly open-ended exploration task helped us collect useful observational data and participant feedback regarding the use of interactive data facts. However, these observations and subjective feedback cannot substitute for a formal evaluation especially to measure the effects of Voder's features on aspects such as visualization interpretation. Consequently, isolating specific features of a system like Voder and running controlled studies to scientifically understand those features and assess their impact is an important next step.

Integration with partial view specification-based tools. Voder currently provides a minimalist manual view specification interface and places more focus on supporting **V**→**F**, **F**→**F**, **F**→**V** transitions as compared to **V**→**V** transitions. However, during the study, it was clear that a better specification interface was required to enable more effective analysis. Going forward, we believe there is potential in incorporating Voder's interactive data facts into a tool like Voyager2 [48] that better enables visual analysis through its partial specification interface and organized visualization recommendations. This would allow users to more easily create visualizations to explore their data and also make it easier for them to interpret and explore communication-oriented alternatives for individual visualizations.

Recommending exploratory facts and visualizations based on user interest. The ability to bookmark data facts in addition to visualizations can allow recommendation-based visual data exploration tools to present more personalized suggestions. For example, a bookmarked data fact like "*Japan has highest average MPG*" gives the system the ability to infer not only the attribute (*MPG*), but also the data case (*Japan*), and tasks (*find extreme*, *derived value*) a user is interested in. With this knowledge, a system can then suggest both additional facts and visualizations that are more tailored to the user's interest captured not only in the form of data attributes, but also specific data cases and analytic tasks. Exploring such recommendation options and the best practices to present them is another open area for future work.

Integrating NLU and NLG. As stated earlier, there may always be cases where automatic techniques do not capture what users find interesting in a visualization. Such cases present the opportunity to combine the ideas presented by natural language interfaces for visualization [39] which typically focus on natural language understanding (NLU) with systems that focus on NLG. For example, in Voder's current version, user-entered data facts are not automatically processed by the system (i.e., the system does not suggest embellishment options to highlight a fact). However, using NLU techniques, systems could provide users with presentation suggestions based on the facts they enter. Alternatively, natural language interfaces that generate a visualization in response to user utterances can leverage NLG techniques to proactively help users ask follow-up questions. For instance, if a user query resulted in a colored scatterplot such as that in Figure 2B, the system could automatically generate follow-up questions regarding correlation between the visualized attributes or specific groups of points (e.g., "*Is there a group of cars exhibiting a correlation between Acceleration and Weight?*"). Exploring such synergies between NLU and NLG techniques is an exciting open area for future work.

Generating narratives and facilitating interactive storytelling. While we have primarily focused on leveraging NLG to interpret and communicate with basic visualizations, an increasingly common application of NLG in visualization is for storytelling. Plug-ins like Wordsmith [49] are automatically generating explanations (as opposed to individual data facts) based on visualizations and dashboards created in Tableau. However, the challenge of a missing 'visual' link between the text and the visualization still persists in these cases. Along the lines of prior work exploring ways to interactively couple text and visualizations [21, 22, 25, 27], expanding the notion of interactive data facts to sentences in such explanations and exploring how they could be used to facilitate interactive storytelling is another open area for future work. Furthermore, incorporating findings from existing work on sequencing visualizations [17, 19], future systems could also investigate how to recommend sequences of facts to convey coherent stories.

7 Conclusion

A growing number of NLG-based solutions are being proposed to help users interpret visualizations and communicate their findings. We explored how data facts generated by these systems can be treated as interactive widgets (as opposed to plain text). Through a prototype of a visualization tool, Voder, we discussed potential applications of interactive data facts in the context of visual data exploration and communication. Specifically, we showed how interactive data facts aid in interpretation by dynamically highlighting parts of a visualization they are referring to. We also demonstrated how systems can present alternatives in the form of visualizations and embellishments to communicate data facts users are interested in. We reported observations from a qualitative user study with 12 participants to discuss how interactive data facts facilitated visualization interpretation and communication. Based on observed behavior and participant feedback, we also discuss how interactive data facts afforded varying data exploration strategies and highlighted potential risks associated with automated data fact generation systems like Voder.

Acknowledgments

We thank the anonymous reviewers for the detailed and helpful feedback on the article. This work was supported in part by the National Science Foundation grant IIS-1717111 and by DARPA FA8750-17-2-0107. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

REFERENCES

[1] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, pages 111–117, Oct. 2005.

[2] Announcing a new collaboration between microsoft and narrative science. https://narrativescience.com/Resources/Resource-Library/Article-Detail-Page/announcing-a-new-collaboration-between-microsoft-and-narrative-science, Apr 2016.

[3] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[4] F. Bouali, A. Guettala, and G. Venturini. Vizassist: an interactive user assistant for visual data mining. *The Visual Computer*, 32(11):1447–1463, 2016.

[5] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.

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

[7] R. Chang, C. Ziemkiewicz, T. M. Green, and W. Ribarsky. Defining insight for visual analytics. *IEEE Computer Graphics and Applications*, 29(2):14–17, 2009.

[8] M. Correll and J. Heer. Black hat visualization. In *Workshop on Dealing with Cognitive Biases in Visualisations (DECISIVe), IEEE VIS*, 2017.

[9] Z. Cui, S. K. Badam, A. Yalin, and N. Elmqvist. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *arXiv preprint arXiv: 1802.08621v1*, Feb 2018.

[10] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Rapid data exploration through guideposts. *arXiv preprint arXiv:1709.10513*, 2017.

[11] C. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending visual insights. *Proc. VLDB Endow.*, 10(12):1937–1940, 2017.

[12] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of ACM UIST*, pages 489–500, 2015.

[13] Google sheets. https://www.google.com/sheets/about/.

[14] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of ACM IUI*, pages 315–324, 2009.

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

[16] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of ACM CHI*, pages 159–166, 1999.

[17] J. Hullman, S. Drucker, N. H. Riche, B. Lee, D. Fisher, and E. Adar. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2406–2415, 2013.

[18] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of ACM CHI*, pages 3363–3372, 2011.

[19] Y. Kim, K. Wongsuphasawat, J. Hullman, and J. Heer. Graphscape: A model for automated reasoning about visualization similarity and sequencing. In *Proceedings of ACM CHI*, pages 2628–2638, 2017.

[20] H.-K. Kong, Z. Liu, and K. Karahalios. Internal and external visual cue preferences for visualizations in presentations. In *Computer Graphics Forum*, volume 36, pages 515–525. Wiley Online Library, 2017.

[21] N. Kong and M. Agrawala. Graphical overlays: Using layered elements to aid chart reading. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2631–2638, 2012.

[22] N. Kong, M. A. Hearst, and M. Agrawala. Extracting references between text and charts via crowdsourcing. In *Proceedings of ACM CHI*, pages 31–40, 2014.

[23] R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. *Computer*, 46(5):44–50, 2013.

[24] N. Koudas, A. Marathe, and D. Srivastava. Flexible string matching against large databases in practice. *Proceedings of the VLDB Endowment*, 30:1078–1086, 2004.

[25] B. C. Kwon, F. Stoffel, D. Jäckle, B. Lee, and D. Keim. Visjockey: Enriching data stories through orchestrated interactive visualization. In *Poster Compendium of the Computation+ Journalism Symposium*, volume 3, 2014.

[26] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007.

[27] R. Metoyer, Q. Zhi, B. Janczuk, and W. Scheirer. Coupling story to visualization: Using textual analysis as a bridge between data and interpretation. In *Proceedings of ACM IUI*, pages 503–507, 2018.

[28] Microsoft power bi. https://powerbi.microsoft.com/en-us/.

[29] Narrative science. https://narrativescience.com/.

[30] C. North. Toward measuring visualization insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006.

[31] A. V. Pandey, K. Rall, M. L. Satterthwaite, O. Nov, and E. Bertini. How deceptive are deceptive visualizations?: An empirical analysis of common distortion techniques. In *Proceedings of ACM CHI*, pages 1469–1478, 2015.

[32] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, volume 5, pages 2–4, 2005.

[33] E. Reiter and R. Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, 1997.

[34] D. Ren, M. Brehmer, B. Lee, T. Höllerer, and E. K. Choe. Chartaccent: Annotation for data-driven storytelling. In *IEEE PacificVis*, pages 230–239, 2017.

[35] S. F. Roth, J. Kolojejchick, J. Mattis, and J. Goldstein. Interactive graphic design using automatic presentation knowledge. In *Proceedings of ACM CHI*, pages 112–117, 1994.

[36] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of ACM UIST*, pages 365–377, 2016.

[37] R. R. Springmeyer, M. M. Blattner, and N. L. Max. A characterization of the scientific data analysis process. In *Proceedings of the 3rd Conference on Visualization '92*, pages 235–242, 1992.

[38] A. Srinivasan, H. Park, A. Endert, and R. C. Basole. Graphiti: Interactive specification of attribute-based edges for network modeling and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):226–235, 2018.

[39] A. Srinivasan and J. Stasko. Natural language interfaces for data analysis with visualization: Considering what has and could be asked. In *Proceedings of EuroVis*, volume 17, pages 55–59, 2017.

[40] A. Srinivasan and J. Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):511–521, 2018.

[41] Tableau software. https://www.tableau.com.

[42] B. Tang, S. Han, M. L. Yiu, R. Ding, and D. Zhang. Extracting top-k insights from multi-dimensional data. In *Proceedings of ACM SIGMOD*, pages 1509–1524, 2017.

[43] Types of insights supported by power bi. https://docs.microsoft.com/en-us/power-bi/service-insight-types.

[44] M. Vartak, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: automatically generating query visualizations. *Proceedings of the VLDB Endowment*, 7(13):1581–1584, 2014.

[45] S. Wehrend and C. Lewis. A problem-oriented classification of visualization techniques. In *Proceedings of the First conference on Visualization'90*, pages 139–143, 1990.

[46] What is natural language generation? https://narrativescience.com/Resources/Resource-Library/Article-Detail-Page/what-is-natural-language-generation.

[47] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016.

[48] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the ACM CHI*, pages 2648–2659, 2017.

[49] Wordsmith by automated insights, inc. https://automatedinsights.com/wordsmith.

[50] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, 1994.