Articulate: Creating Meaningful Visualizations from Natural Language

Yiwen Sun[1], Jason Leigh[1], Andrew Johnson[1], Barbara Di Eugenio[2]
[1]*Electronic Visualization Laboratory*
[2]*Natural Language Processing Laboratory*
University of Illinois at Chicago, USA

ABSTRACT

This chapter presents an approach to enable non-visualization experts to craft advanced visualizations through the use of natural language as the primary interface. The main challenge in this research is in determining how to translate imprecise verbal queries into effective visualizations. To demonstrate the viability of the concept, we developed and evaluated a prototype, *Articulate*, which allows users to simply ask the computer for questions about their data, and have it automatically generate visualizations that answer these questions. We discovered that by relieving the user of the burden of learning how to use a complex interface, we enable them to focus on articulating better scientific questions and wasting less time in producing unintended visualizations.

INTRODUCTION

Nearly one third of the human brain is devoted to processing visual information. Vision is the dominant sense for the acquisition of information from our everyday world. It is therefore no surprise that visualization, even in its simplest forms, remains the most effective means for converting large volumes of raw data into insight. Over the past three decades, much has been investigated in the design of sophisticated visualization tools in a variety of disciplines. However, the effort end-users have to make to craft a meaningful visualization using these tools has been mostly overlooked. The users of such tools are usually lay people or domain experts with marginal knowledge of visualization techniques. When exploring data, they typically know what questions they want to ask, but often do not know, or do not have the time to learn, how to express these questions into a series of interactions that produce an effective visualization.

A 2008 National Science Foundation report "Enabling Science Discoveries through Visual Exploration" [Ebert08] also noted that one of the main barriers hindering the adoption of advanced visualization tools is the steep learning curve associated with them. 2010 findings by Grammel [Grammel10] showed that novices to Information Visualization still tended to use traditional bar, line and pie charts over other chart types by more than 70% because of their familiarity with them. Modern visualization tools offer such an expansive array of capabilities that they can only be wielded by an expert trained in visualization. In some ways it's like expecting someone to know how to build a house by simply sending them to Home Depot[i].

Meanwhile, the 2008 NSF report noted "there is a strong desire for conversational interfaces that facilitate a more natural means of interacting with science." In other words, scientists "simply" want to tell the computer what they want to see and have it just create it. They do not want to have to become visualization experts. Even a decade ago this would have seemed far-fetched, but today we are seeing renewed interest in the use of natural language as an interface to computing. For example, survey results according to search engines like Ask.com show that a third of search queries are entered as natural language questions rather than keywords. Siri, the intelligent personal assistant on iPhone 4S, allows users to send messages, schedule meetings, and place phone calls by directly speaking into their smartphones. The field of natural language processing has made great strides in the last decades, with a variety of models that are able to understand the meaning of sentences in recommender systems, educational technology and health applications.

This inspired us to consider the use of a conversational interface for the automatic generation of visualizations. A system such as this would allow an end-user to pose natural language inquiries, and then let the system assume the burden of creating the most appropriate visual representation of the inquiry. It is hoped that such a capability can potentially reduce the learning curve necessary for effective use of visualization tools, and thereby expand the population of users who can successfully conduct visual analysis. Note however that in this work we are not simply translating explicit visualization commands such as "*make a plot of temperature vs pressure*"- though it is certainly possible within the context of this research. Instead the expectation is that our approach will enable a user to ask deeper questions about data, such as "*what is the correlation between temperature and depth with temperature below zero*", without having to follow or memorize a strict grammar or command structure, as has been in the past. Furthermore users will be able to ask follow-up questions where the computer has some knowledge of what has already been asked and visualized. Therefore the fundamental value of this approach is that it enables the end-users to focus on the scientific question they are trying to ask rather than the specific visualization task they are trying to perform.

The remainder of this chapter is organized as follows. We first describe prior work in related fields. Then we explain in detail our methodology for translating conversation into a precise visualization. Next, we present details of our user studies. While the initial prototype produces information visualizations, we will also explain and show through a case study how the approach is conceptually extensible to scientific visualizations as well. Lastly we outline directions for future work.

BACKGROUND

The problem of deriving a meaningful visualization from natural language highlights many interesting areas for computer science research. It involves researching the steps needed to translate and classify spoken queries that can then be meaningfully visualized. It also involves discovering how to create, modify and explain visualizations automatically, and understanding the benefits and issues related to such an approach.

As creating visualizations remains a skilled and time-consuming task, researchers began to investigate computational approaches to simplify the crafting process. One of the early pieces of work was Mackinlay's APT system [Mackinlay86]. It introduced a composition algebra to describe various graphical encoding and developed expressiveness and effectiveness criteria to ensure meaningful design. The SAGE system [Roth94] extended the concepts of APT, providing a richer set of data characterizations and generating a wider range of composite views through interaction. The previous work on automatic presentation focused primarily on single views of data; however, Show Me [Mackinlay07] provided support for small multiple views. It included an integrated set of user interface commands and default settings that automatically generate small multiple views based on VizQL – an algebraic specification language. Users place data fields into columns and rows in the interface panel to specify VizQL commands. In order to generate insightful visualizations, an understanding of the relationships between columns and rows is needed. While the above work focused on identifying and encoding data in discrete graphics, there is another trend in addressing the issues of communicating with users in visual discourse. Feiner's Apex system [Feiner85] set the foundational work in this area. It attempted to automatically create visual discourses - a series of animated visual illustrations for explaining complex information to users. His work was extended in IMPROVISE [Zhou98], which used an AI planning-based approach to automatically design and create such discourses.

In recent years, related approaches have targeted specific application domains, rather than proposed a more general methodology. Kerpedjiev and Roth introduced AutoBrief [Kerpedjiev01], a system that automatically summarizes transportation schedules as text and graphs. In this system, they proposed a set of rules to map communicative goals to low-level operational tasks, such as lookup or compare. However, the system only focused on generating explanations of problems and relations existing in the data but not in response to user's ad-hoc requests. Gilson et al. [Gilson08] proposed an approach for automatic generation of visualizations via ontology mapping and applied the approach to web data for music. The web data was first mapped to domain ontology, and then projected to visual representation ontology, which was finally depicted as a specific visualization using external visualization toolkits. The mapping between domain and visual representation ontologies was represented by semantic bridging ontologies, which were defined from expert knowledge. By comparison, our approach uses a more flexible meta-learning algorithm to automatically translate language into visualization intentions. Another interesting approach is VisMashup [Santos09], which simplified the creation of customized visualization applications with a pipeline. Once the pipeline is constructed, the application can be generated automatically. While this infrastructure enables an application designer to assemble custom applications quickly, the designer still needs some visualization background to build up the pipeline from components or templates. Although *Articulate* shares some of these same goals, it goes a step further by allowing the users to verbally articulate what they want to see with minimal a priori knowledge of how to use the user interface.

A decade ago the possibility of widespread use of speech interaction seemed far-fetched, for both cognitive and technical reasons. Shneiderman [Shneiderman00] argued that speech input has limited value in human-computer interaction except in niche applications - such as for the disabled or answering service systems. The key criticism cited was that problem-solving and recall competed with speech articulation and interpretation in their use of working memory. However, Dennett [Dennett92] argues that problem solving is in fact enhanced when more areas of the brain are engaged such as when you are speaking and hearing your own words (i.e. thinking a problem out loud). Hanks [Hanks10] argued that "Humans are social animals, and language is the instrument of their sociability, as well as the vehicle of their thought processes." Language interfaces in a variety of settings, from educational technology to health care, have been shown to improve the user's experience and performance [Grasso98, Hallet08, Kersey09, Schulman09].

Technically, the considerable renewed interest in the use of speech and natural language as an interface to computing is due in large part to significant computing power and new powerful statistical models that are brought to improve speech recognition and natural language interpretation. Google's director of research Peter Norvig, believes that being able to converse with computers is "the future". NLP, the processing of language beyond the recognition of words in speech, also made great strides in the last decade, with a variety of models that are used to understand the meaning of sentences, as shown by successes such as that of IBM Watson which defeated the two best human champions in Jeopardy! and Wolfram Alpha - a knowledge engine developed by Wolfram Research that is capable of responding to natural language based questions with computed answers and relevant visualizations instead of a list of web pages as a traditional search engine provides. Additionally, a number of speech-based computational models have been developed that help users to access information using a conversational paradigm. JUPITER [Zue00b] for example allows users to obtain worldwide weather forecast information over the phone using spoken dialogue. It is a mixed initiative system [Zue00a] that requires zero user training, and accepts a large range of user inputs. This approach has been extended to similar domains where the vocabulary is sufficiently limited to support practical conversational paradigm, such as travel planning [Seneff00], health information access [Sherwani07].

Recent work in the information visualization community has applied various design principles to the automatic generation of visualizations though none had used natural language nor approached scientific visualizations. *Articulate* attempts to combine these advanced techniques together in exploring how to automatically translate natural, and potentially ill-defined, conversational language into meaningful visualizations of data in a generalizable way that enables users who are not visualization experts to make use of modern advances in visualization.

THE APPROACH

In brief, the approach involves: extracting syntactic and semantic information from a verbal query; applying a supervised learning algorithm to automatically translate the user's intention into explicit commands; and finally, determining an appropriate type of visualization based on the translated commands and properties of the meta-data (see Figure 1.)

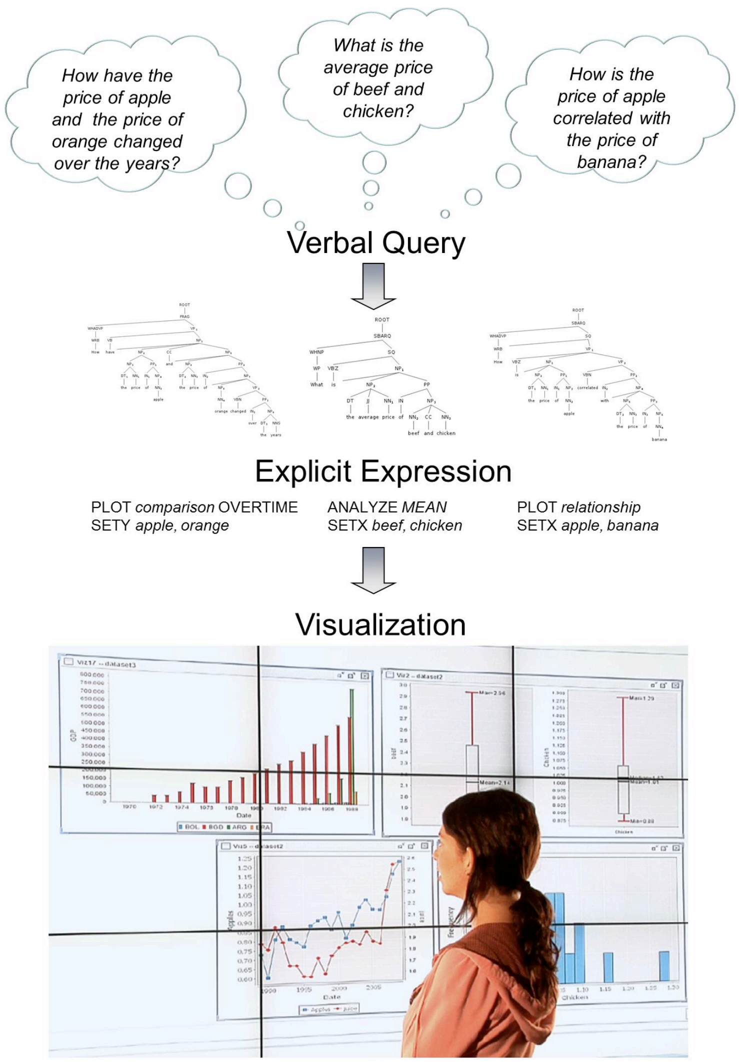

Figure 1. A conceptual pipeline for translating imprecise verbal queries into visualizations.

To demonstrate the concept is viable, we developed a small prototype - *Articulate*. There are three essential parts to its framework: the Data Parser, the Input Translator, and the Visualization Executer. The Data Parser collects information on the attribute names of the data and their data types. The Input Translator takes natural language queries spoken by the user and translates them into a set of commands that follows a precise grammar, which we call SimVL (Simplified Visualization Language) – analogous to small subset of Wilkinson's Grammar of Graphics [Wilkinson00]. The precise SimVL commands and information on the properties of various types of visualization, are given to a Visualization Executer which determines the most appropriate visualization to produce. Figure 2 outlines the major components of the Input Translator. In what follows, we present the ideas underlying these major components.

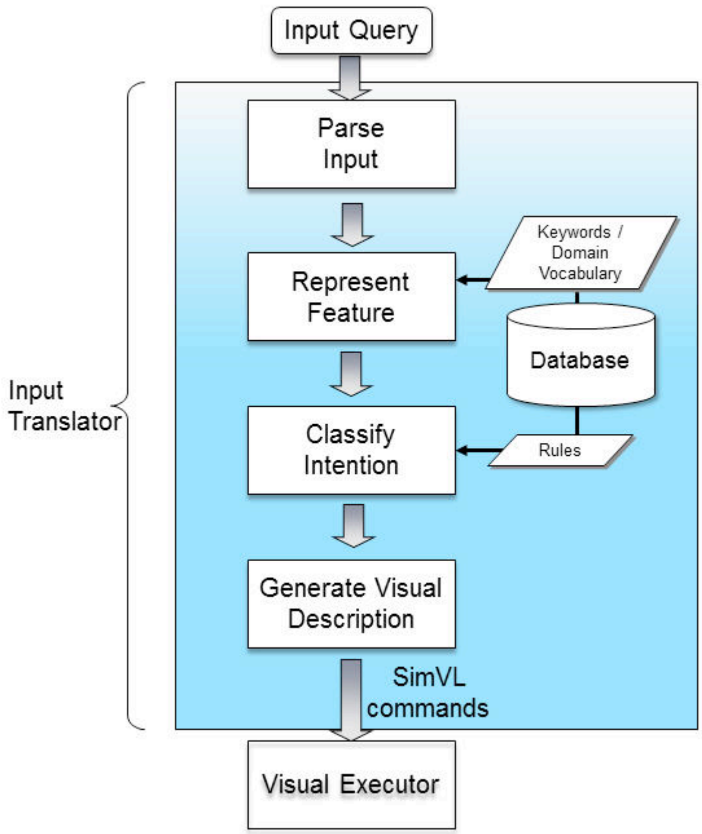

Figure 2. Basic workflow in the Input Translator.

Parsing the Input Stream

The user's initial input to the system is a natural language query. The query sentence is parsed into a syntax tree where leaf nodes store the words and internal nodes show the part-of-speech or phrasal labels. This tree provides an intuitive perception of the structure of the sentence (Figure 3). Additionally, a dependency diagram is obtained via the Stanford Parser [Klein03] to describe the grammatical relationships between pairs of words. Using these structures, the function of each word can be identified which helps to recognize the feature of the query.

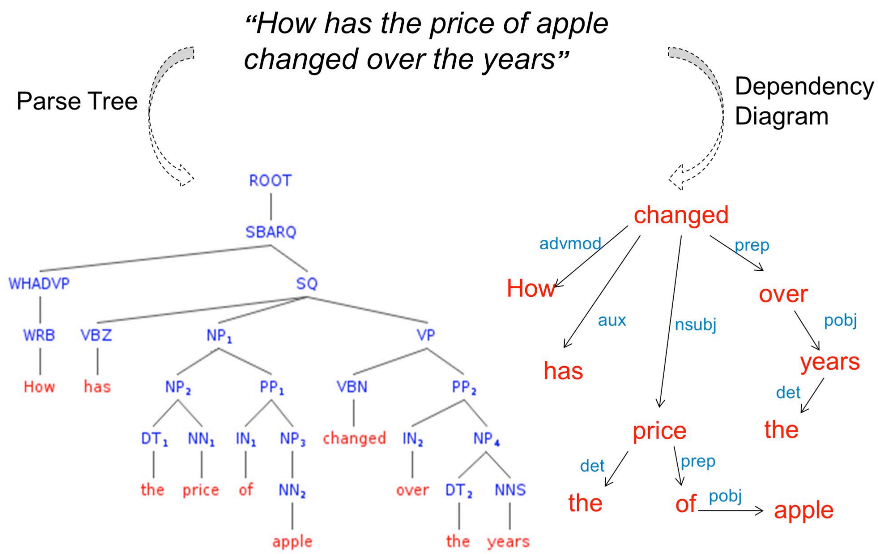

Figure 3. An example of syntax tree and dependency diagram parsed from a sample sentence.

Ideally when articulating a query, the user will mention an attribute name as it appears in the raw data. But in reality this is not always the case, as discovered in our earlier study [Sun10]. For example, in a dataset regarding average food prices, data values may include apple, orange and banana. However users may pose queries such as: "*compare the price of fruit*". Clearly, searching for the exact data value in the query will not lead to a desired result. Hence, meta-data properties are gathered such as data units and semantic relations, which provide a brief context for interpreting the data. To obtain the semantic relations, we use WordNet [Miller95], a lexical database that groups English words into sets of cognitive synonyms (synsets) and expresses various semantic relations between these synsets. Using this, we can expand each attribute into an insightful meta-word and send that to the Visualization Executer to help determine the most appropriate graph to produce.

Representing Syntactic and Semantic Features

The results from the language parsing step provide complex information about the features of the query. Some of the information is not essential in the procedure of identifying the user's general intention. Hence, we defined six categories of keywords: *comparison, relationship, composition, distribution, statistics* and *manipulation*, to represent the semantics of user's general intentions. The keywords in each dictionary are selected according to empirical knowledge and domain vocabulary. For example, *associate, correlate, link, relate, relevant* are often used in the queries intended for tasks regarding relationship or connection between two or more variables, so they are entered into the relationship dictionary.

Besides that, the findings from the preliminary user study show that some queries were not correctly answered due to the ambiguity of query's feature. It is possible to improve feature identification by capturing the syntactic characteristics of those queries. Through close examination, several shallow linguistic features were found that might help the classification of the query, for example clause type, query contains a comparative or superlative adjective or adverb, query contains a cardinal number, and query contains a quantifier.

Based on the above syntactic and semantic feature analysis, a smaller feature space is derived to represent the most important aspects of the query. This feature space is defined as a fourteen-dimensional space.

Each dimension describes one feature found in the query. Specifically, the features are: comparison, relationship, composition, distribution, statistics, manipulation, time-series, visual_primitive, superlative, cardinal, quantifier, filter, clause_type, and number_of_attributes. The first twelve are Boolean values. For instance, if a query involves the comparison between two data attributes, the comparison feature is tagged as true, and the number_of_attributes is declared as 2. In this way, a query sentence can be simply represented as a feature vector.

Classifying the Intended Tasks

The feature vector essentially identifies the words that describe the intended visualization. It does not however guarantee that the user is sufficiently precise in their use of their wording. Therefore the feature vector is given to a task classifier that attempts to derive the user's true intent. Three widely used supervised learning algorithms were considered for this classification job: Decision Tree, Bayesian Network and Support Vector Machine. Each model generated by one machine-learning algorithm can be regarded as an expert. It is more reliable to take into account the opinions of several experts rather than relying on only one judgment. Therefore, we combined a decision tree inducer, a Bayesian network learner, and a support vector machine to form a meta-learner, which takes the majority votes from the three basic classifiers. This meta-learner was formed by applying the meta-learning method over a corpus of sample queries tagged into seven classes of visualizations, based on Shneiderman's task taxonomy [Shneiderman96] and Abela's chart classification model [Abela81].

The visualization task recognized using the meta-learner gives the user a solution based on the limited training corpus. However it may not always be the "best" choice. To help users find the truly intended visualization, a means to allow *Articulate* to suggest alternative visualizations was explored. The algorithm employed to select candidates for suggestion is based on a context-aware meta-classifier. This classifier is similar to the idea used in task classification process discussed above, in which a decision tree inducer, a Bayesian network learner, and a support vector machine are combined together. But different from the task classifier, the immediately preceding tasks were taken into account as context for suggestion, which maintains the coherence of attention in successive tasks.

Given a successful classification, and having identified the attributes to be visualized, a set of precise commands in the form of SimVL is generated. Recall that SimVL is our formal grammar for describing visualization tasks in a precise manner. SimVL's commands comprise four categories: sketch commands, analysis commands, manipulation commands and filter commands. Sketch commands consist of commands that describe the semantics of general visualization tasks, which often perform the actual task of drawing a graph. Analysis commands consist of tasks that involve the use of statistical methods such as taking the average or standard deviation. Manipulation commands describe ways to alter an existing visualization- such as remapping colors to data attributes. Lastly, filter commands are mainly used to select the desired pieces of data.

Obtaining Meaningful Visual Results

The Visualization Executer reads the SimVL command, as well as the properties of various types of visualizations, and uses a heuristic algorithm to choose the most appropriate visualization to execute [Sun12]. Just as a visualization expert might weigh the pros and cons of different graphs in the determination of a desired plot type, the Executer works as an agent carrying out a similar reasoning process. For sketch commands, the choice of a specific visualization is contingent upon factors including the property of the data (such as the number of variables, data types, and whether the variables change over time), and the effectiveness of different visual primitives (such as position, length, direction, area, volume, shading and color saturation). Figure 4 gives an example. For analysis commands, which usually focus on the statistical features of data (such as minimum, maximum, average, percentile), a box-and-whisker chart is a convenient way of graphically depicting these features, as illustrated in Figure 5. Manipulation commands are typically adjustments made to an existing visualization, for example choosing to color data points based on the values of another attribute, as illustrated in Figure 6. Filter commands are often attached to sketch commands or analysis commands with constraints on data values, so the Executer's job is to understand the constraints and filter out unintended data before plotting it.

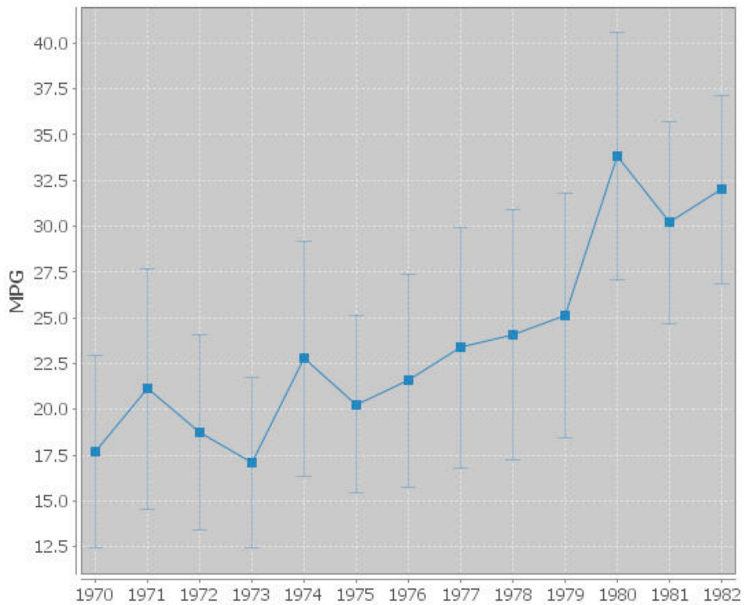

Figure 4. Result for sketch and filter commands translated from "how has MPG changed since 1970" with regard to a 1983 ASA Data Expo dataset on automobiles.

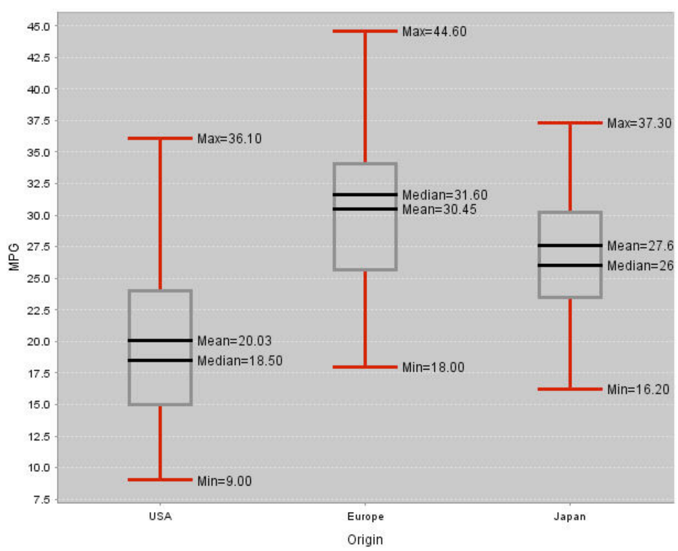

Figure 5. Result for analysis commands translated from "what is the average MPG by country of origin" with regard to the same dataset as Figure 4.

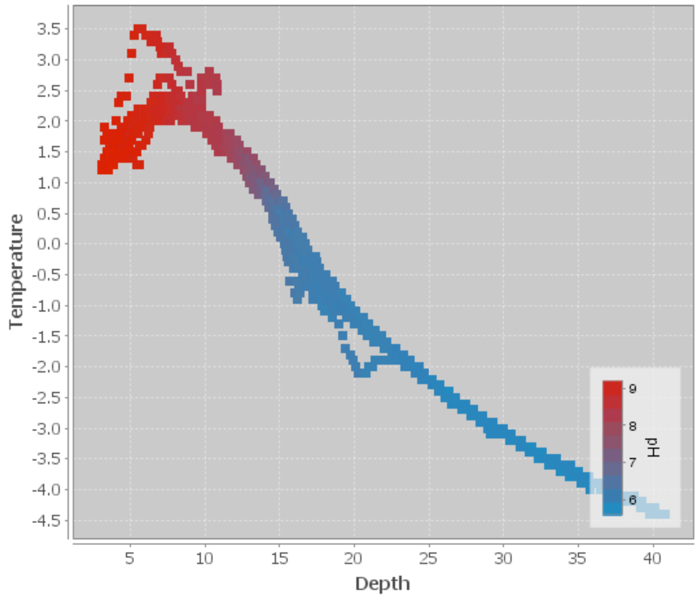

Figure 6. Result for manipulation commands translated from "can you color the points by pH" following a query "what is the correlation between depth and temperature" with regard to a 10-attribute hydrological dataset.

EVALUATION

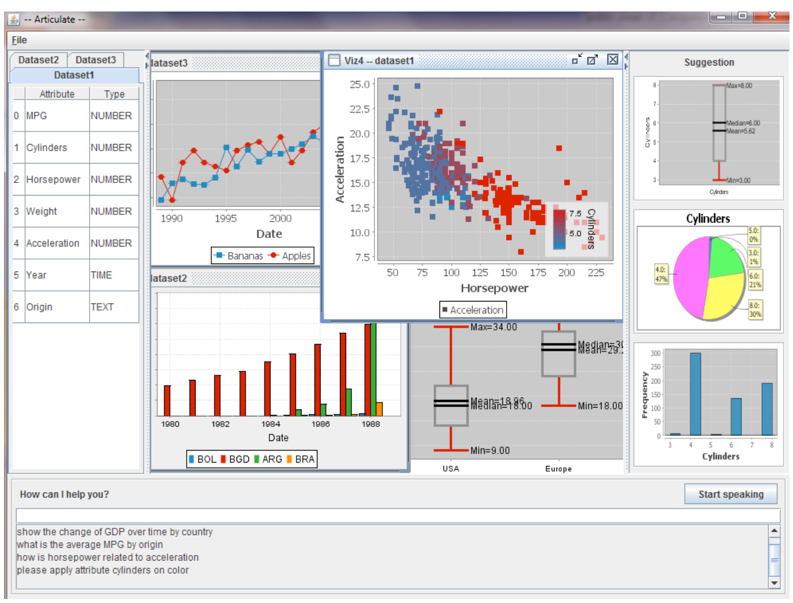

Figure 7. The user interface for the Articulate system.

A prototype of *Articulate* has been implemented in Java. Figure 7 shows a screenshot of the user interface for the prototype. A user speaks into the system without any a priori knowledge of a grammar and the translated text is displayed to the user as well as the resulting visualization. Both the suggestion panel and data window are collapsible to allow users to focus on the intended visual result. Speech recognition is achieved using the Sphinx toolkit, a leading speech recognition toolkit developed at Carnegie Mellon University with various packages used to build speech applications in Java. The Stanford Parser is leveraged for natural language parsing. The task classification prototype employs a meta-learner combining a decision tree inducer, a Bayesian network learner, and a support vector machine implemented using WEKA API. As described earlier, since this is only a proof-of-concept to validate the approach, we chose to initially target simple visualizations such as bar charts, scatter plots, etc. Therefore we employed the JFreeChart and Prefuse graph engines to perform the graph generation.

We conducted a preliminary study to evaluate the viability of our approach. The study consisted of a comparison between users of *Articulate* versus a popular graphing tool such as Microsoft Excel. Subjects in the experiment were provided with a number of datasets ranging from hydrologic data to census data and were given 20 minutes to perform 3 tasks: find meaningful correlations among the data attributes; find as many trends as possible about the data; find other interesting features about the data. We tracked the number of graphs produced, their types, and the duration needed to create a graph that resulted in a discovery. The findings were highly encouraging. We found that at least half of the Excel users had to create more than one chart and call additional Excel functions (such as sort, min, max) to describe a query that could have been expressed with a single sentence in *Articulate*. Furthermore, on average users took twelve times longer to realize a graph for a query in Excel than in Articulate.

Case Study

While the initial prototype focuses on producing information visualizations, this methodology can be applied to scientific visualizations by adjusting certain steps in the framework. The experience of a case study for the Endurance application illustrates this idea.

ENDURANCE (Environmentally Non-Disturbing Under-ice Robotic ANtarctic Explorer) is a NASA funded project involving the development of an autonomous underwater vehicle (AUV) capable of generating 3-D bio-geochemical datasets in the extreme environment of a perennially ice-covered Antarctic dry valley lake, which offers a blend of statistical as well as scientific visualization problems. In addition, they have been using the visualization tools developed in our lab to support the analysis of the data coming from the ENDURANCE mission.

The first step in the expansion of *Articulate* towards the ENDURANCE application is to understand their usage scenario. Preliminary observations of interactions between a visualization expert and several of the participating domain scientists using the existing visualization tool were conducted. Through the observation, a couple of distinctive behavioral features were found: data is preferred to be presented in location-based color-coded 3D representation; multiple parameters are plotted and compared side by side frequently; data are often selected or filtered by time and location. Based on these observations, we adjusted the *Articulate* system in certain aspects. The first adjustment occurs in the meta-data deriving step: all the biological and geo-chemical measurement parameters are extracted from the original data files, as well as some domain specific terms used by the scientists, such as bathymetry, slope, and scale. Secondly, feature representations are tailored to reflect the domain scenario. For example, a couple of shallow linguistic cues are identified to distinguish 3D visualization requests from 2D counterparts, such as the appearance of keywords like 3D, volumetric, bathymetry in a query. Finally, in the Visualization Executer, adaptions are made in the graph reasoning algorithm to accommodate 3D views. In the current prototype, VTK (the Visualization Toolkit) [Schroeder06] was employed to visualize scalar data. Figure 8 gives an example, where a 2D lake map with grids is overlaid on top of the 3D glyphs to highlight the location information.

Before given access to *Articulate*, the domain scientists had to interact with a visualization expert to create various visualizations needed in their data analysis. With the help of *Articulate*, scientists were able to use language to create visualizations and modify those that have been created previously. Since the framework of *Articulate* is composed of distinct modules, each of them handles a single process such as data processing, language parsing, or graph generation; it is very flexible to be applied to different domain science by adjusting certain modules in the framework.

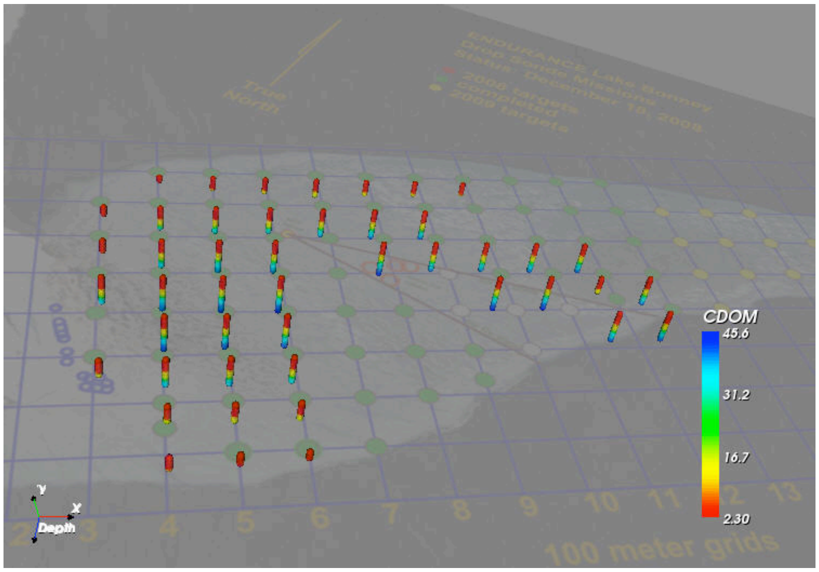

Figure 8. 3D visualization for the query "Show me the relationship between depth and CDOM in a 3D view".

User Study

The framework of *Articulate* has a number of theoretical advantages. However, the success of an approach will ultimately depend on whether the users perceive improvements in their experience. For that reason we designed a between-subject experiment to discover whether the users using *Articulate* can: produce more appropriate visualizations and fewer irrelevant visualizations, and make discoveries that would not have occurred otherwise. Seventeen post-graduate subjects participated in the study. Each of them was presented with multi-country wealth and health development data extracted from Gapminder World's data repository. The participants were expected to perform three tasks at two different levels of complexity (two simple direct tasks and one complex open-ended task):

- Task 1: What are the top 2 countries that emit most CO2 recently?
- Task 2: Is there any big difference between the trends of Japan and Philippines?
- Task 3: Based on this data set which attribute(s) do you think are important factors to life expectancy?

Subjects were asked to make verbal queries to *Articulate* to explore the data and summarize their findings based on the visual results showing on the display.

All participants completed the tasks, but their total completion time varied (mean = 26.8 minutes, SD = 9.3) and the number of queries initiated were different (mean = 8, SD = 3.4). To investigate the different behavior and the effectiveness of the system, we performed statistical analysis on the factor of query per task, which measured the number of queries user initiated to complete each task. We wanted to find out if the number of queries was related with the type of task. A one-way analysis of variance (ANOVA) with $\alpha = 0.05$ was used for this analysis. For task 1 and 2, on average about 1.5 queries were needed to solve the problem, which clearly showed the efficiency of the system. Task 3 required more queries (mean=4.8) to solve due to its complexity. The ANOVA result also indicated that the level of complexity of the task had a significant impact on the number of queries user initiated ($F_{2,48} = 12.45$, $p < 0.0001$).

Furthermore, we analyzed the subjective ratings in the post-study survey pertaining to the effectiveness of the suggestion function. The rating is based on a 5 point Likert scale (1 = Never helpful; 5 = Always helpful). The result was positive (mean=3.7, SD=0.8). The main reason subjects liked the suggestion function was that it provided alternative visualizations that gave insights on different perspectives of the data, which could potentially help users find their solution quickly.

Learning any new system can increase cognitive burden, and in this case, that means the transition from mind-hand coordination to mind-mouth coordination. In this study, we were encouraged to find that all of the subjects were able to alter their working styles to adapt to the new environment. Their feedback on using *Articulate* compared with other traditional visual analytic tools showed that the subjects were more favorable towards the natural language guided interface:

"I think being able to just speak my question instead of having to type it is very helpful."

"In other tools like Excel I have to put in the data, find the right chart, make everything organized. But for this, I asked a question, it pulls the data for me. It gives me easier access to different types of graphs without me having to go back and fill in the place again."

"I like the suggestions because they were not necessarily things I was looking for myself. … Most of time it understood what I was looking for, but it also brought in things that I wasn't looking for that ended up being helpful. It did a good job translating my requests but also giving me alternatives."

FUTURE RESEARCH DIRECTIONS

The work presented in this chapter offers a new approach for automatically creating visualizations from verbal descriptions; however, there are still limitations that need to be addressed. For example, in the current graph engine, only standard 2D graphs and plots, and basic 3D scalar visualizations are supported, which we fully intend to extend in the future, to accommodate more advanced visualization techniques such as those commonly used in scientific visualization (for instance volumetric or streamline visualizations).

Another direction in this research is to provide a textual or verbal explanation to the user on how and why the resulting visualization was derived by the system. When answering the user's questions, which may include requests for information about how to visualize the data and requests for clarification on what the visualization represents, *Articulate* needs access to its own knowledge about the implementation of the current visualization, such as the mapping from data attribute to visual primitives. Providing explanations of the visualization can help users learn visualization techniques they did not know before, and make better understanding about the data.

In terms of interaction modality, enabling gesture input as a complementary interface to voice input is a promising area to investigate. During a conversational interaction, verbal communication is often enhanced or further clarified through a variety of different gestures. As observed in the user study, subjects tend to point and gesture to the screen when saying "this", "these", "here" in deictic expressions. Recognizing pointing gestures can help the interpretation of references used in the verbal description. With devices such as a multi-touch screen and optical trackers (such as the Kinect or Leap), it will be relatively easy to extend *Articulate's* framework with the gesture interaction.

Lastly, one of the emerging trends in large scale data visualization is the use of display-rich environments (Figure 9) to facilitate the integration and interpretation of multiple visualizations simultaneously [Leigh12]. A natural language and gesture based interface is ideally suited to such an environment where the traditional mouse and keyboard are typically cumbersome.

Figure 9. A scalable display wall environment (called an OptIPortal).

CONCLUSION

This chapter has demonstrated *Articulate* - a generalized approach for the creation and modification of meaningful visualizations from user's imprecise verbal descriptions. Unlike traditional visualization tool, this approach brings together natural language processing, machine learning and visualization techniques, to enable the interaction with speech, hence relieve the users of the burden of having to learn how to use a complex interface for their data exploration.

The major components of *Articulate* include: Data Parser, Input Translator, and Visualization Executer. A Data Parser collects information from the data file and prepares them in metadata formats to provide a brief context for interpreting user's interests. The Input Translator recognizes the intended data and translates user's natural language descriptions into specific visualization tasks expressed in a formal grammar. These explicit expressions together with the impact of various graphical primitives guide the Visualization Executer in determining the most appropriate graph. A key benefit of this approach is that it enables the end-users to focus on the scientific question they are trying to ask rather than the specific operations that must be performed to produce a representative visualization.

The contributions of this research are three-fold: First, the incorporation of a speech interface and natural language parser enables the user to "tell" the computer what they want to see, and have the system intelligently create the graph rather than having to struggle with yet another esoteric user-interaction device.

Second, the introduction of a meta-learning algorithm to automatically interpret a user's intent based on linguistic features. We devised a multi-dimensional feature space to represent both syntactic and semantic characteristics of a verbal query. The query is then converted to a feature vector, which is mapped to a visualization task by applying a supervised learning algorithm, and finally translated into explicit graphical commands.

Third, the capability of suggesting related queries and visualizations. Besides the primary recommended visualization, a list of "next-best" candidates are provided to the users, which take into account their previous preferences. Such context-aware suggestions can potentially help them consider alternative perspectives on the data that they may not have originally envisioned.

Formal studies via between-subject experiments were conducted to evaluate the approach. *Articulate* users took less time and created fewer charts to produce a desired result as compared to using a popular graphing program. The studies also showed this approach was able to provide meaningful visualizations to the user, help them make discoveries that would not have occurred to them otherwise, and eventually speed up their data exploration process.

The presented research results, although far from constituting a complete set, offer direction for the future investigation of the automatic generation of data visualization in a natural interaction environment. As the technology needed for building such environments becomes more affordable, we will likely see natural language interfaces permeate a broad range of everyday use cases. Since the Internet has made it possible for everyday citizens to access vast amounts of data instantaneously and at very low cost. There is tremendous interest amongst researchers in the visualization community finding better ways to harness data to make it useful to individuals, businesses, and companies. The work described here will contribute greatly to that effort by making it possible for non-visualization experts to be able to leverage modern advances in visualization and help them interpret data that is growing exponentially in scale and complexity.

REFERENCES

[Abela81] Abela, A. (1981). *Advanced Presentations by Design: Creating Communication that Drives Action*. San Francisco, CA: Pfeiffer.

[Dennett92] Dennett, D. (1991). *Consciousness Explained*. New York: Little, Brown and Company.

[Ebert08] Ebert, D., Gaither, K., Gilpin, C. (2008). Enabling science discoveries through visual exploration. *NSF Workshop report*. Washington, D.C.

[Feiner85] Feiner, S. (1985). APEX: An Experiment in the Automated Creation of Pictorial Explanations. *IEEE Computer Graphics and Applications,* 5(11), 29-37.

[Gilson08] Gilson, O., Silva, N., Grant, P.W., & Chen, M. (2008). From Web Data to Visualization via Ontology Mapping. *Computer Graphics Forum*, 27(3), 959-966.

[Grasso98] Grasso, M. A., Ebert, D. S., & Finin, T. W. (1998). The Integrality of Speech in Multimodal Interfaces. *ACM Transactions on Computer Human Interaction*, 5(4), 303–325.

[Grammel10] Grammel, L., Tory, M., Storey, M-A. (2010). How Information Visualization Novices Construct Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 943-952.

[Hallet08] Hallett, C. (2008). Multi-modal presentation of medical histories. *Proceedings of the 13th International Conference on Intelligent User Interfaces*, pp. 80–89. ACM.

[Hanks10] Hanks, P. (2010). How People Use Words to Make Meanings. *Proceedings of the 7th International Workshop on Natural Language Processing and Cognitive Science*, 3(13).

[Kaufmann07] Kaufmann, E., Bernstein, A. (2007). How useful are natural language interface to the semantic web for casual end-users? *Proceedings of the 6th International Semantic Web Conference*, pp. 281–294.

[Kerpedjiev01] Kerpedjiev, S., Roth, S.F. (2001). Mapping Communicative Goals into Conceptual Tasks to Generate Graphics in Discourse. *Knowledge-Based Systems*, 14(1), 93-102.

[Kersey09] Kersey, C., Di Eugenio, B., Jordan, P. W., Katz, S. (2009). Knowledge co-construction and initiative in peer learning interactions. *In the 14th International Conference on Artificial Intelligence in Education*.

[Klein03] Klein, D., & Manning, C.D. (2003). Accurate Unlexicalized Parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423-430.

[Koop08] D. Koop, C. Scheidegger, S. Callahan, J. Freire, and C. Silva. (2008). Viscomplete: Data-driven suggestions for visualization systems. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 1691–1698.

[Leigh12] Leigh, J., Johnson, A., Renambot, L., Peterka, T., Jeong, B., Sandin, D., Talandis, J., Jagodic, R., Nam, S., Hur, H., Sun, Y. (2012). Scalable Resolution Display Walls. *Proceedings of the IEEE*, Issue 99.

[Mackinlay86] Mackinlay, J.D. (1986). Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*, 5(2).

[Mackinlay07] Mackinlay, J.D., Hanrahan, P., Stolte, C. (2007). Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6).

[Miller95] Miller, G.A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), 39-41.

[Roth94] Roth, S.F., Kolojejchick, J., Mattis, J., Goldstein, J. (1994). Interactive graphic design using automatic presentation knowledge. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 112-117.

[Santos09] Santos, E., Lins, L., Ahrens, J., Freire, J., Silva, C. (2009). VisMashup: Streamlining the Creation of Custom Visualization Applications. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 1539-1546.

[Schulman09] Schulman, D., & Bickmore, T. (2009). Persuading users through counseling dialogue with a conversational agent. *Proceedings of the 4th International Conference on Persuasive Technology*, pp. 1-8.

[Schroeder06]Schroeder, W., Martin, K., Lorensen, B. (2006) *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics*, 4th Edition, Kitware, Inc. publishers.

[Seneff00] Seneff, S., Polifroni, J. (2000). Dialogue Management in the Mercury Flight Reservation System. *In ANLP/NAACL 2000 Workshop on Conversational Systems*, pp. 11-16.

[Shneiderman96] Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Proceedings of the IEEE Symposium on Visual Languages*, pp. 336-343.

[Shneiderman00] Shneiderman, B. (2000). The Limits of Speech Recognition. *Communications of the ACM*, 43(9), 63-65.

[Sherwani07] Sherwani, J., Ali, N., Tongia, R., Rosenfeld, R., Memon, Y., Karim, M., Pappas, G. (2007). HealthLine: Towards Speech-based Access to Health Information by Semi-literate Users. *Proceedings of Speech in Mobile and Pervasive Environments*.

[Sun10] Sun, Y., Leigh, J., Johnson, A., and Lee, S. (2010). Articulate: A Semi-automated Model for Translating Natural Language Queries into Meaningful Visualizations. *Proceedings of 10th International Symposium on Smart Graphic*, pp. 184-195.

[Sun12] Sun, Y. (2012). *Articulate: Creating Meaningful Visualizations from Natural Language*. Doctoral dissertation, University of Illinois, Chicago.

[Wilkinson05] Wilkinson, L. (2005). *The Grammar of Graphics*, 2nd Edition, Springer.

[Zhou98] Zhou, M., & Feiner, S. (1998). IMPROVISE: Automated Generation of Animated Graphics for Coordinated Multimedia Presentations. *Second International Conference on Cooperative Multimodal Communication*, pp. 43-63.

[Zue00a] Zue, V., Glass, J.R. (2000). Conversational Interfaces: Advances and Challenges. *Proceedings of the IEEE*, pp. 1166-1180.

[Zue00b] Zue, V., Seneff, S., Glass, J.R., Polifroni, J., Pao, C., Hazen, T.J., Hetherington, L. (2000). JUPITER: A Telephone-Based Conversational Interface for Weather Information. *IEEE Transactions on Speech and Audio Processing*, 8(1), 85-96.

ADDITIONAL READING SECTION

[Amar05] Amar, R., Eagan, J., & Stasko, J. (2005). Low-level components of analytic activity in information visualization. *Proceedings of the IEEE Symposium on Information Visualization*, pp. 111-117.

[Bavoil05] Bavoil, L., Callahan, S., Crossno, P., Freire, J., Scheidegger, C., Silva, C., & Vo, H. (2005). Vistrails: Enabling interactive multiple-view visualizations. *IEEE Visualization*, pp.135–142.

[Bostock09] Bostock, M. & Heer, J. Protovis: A Graphical Toolkit for Visualization. (2009) *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 1121-1128.

[Chih08] Chih, C., Parker, D. (2008). The persuasive phase of visualization. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, NY, USA, pp.884-892.

[Cleveland84] Cleveland, W. S., & McGill, R. (1984). Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association*. 79(387), 531-554.

[DiEugenio10] Di Eugenio, B., Zefran, M., Ben-Arie, J., Foreman, M., Chen, L., Franzini, S., Jagadeesan, S., Javaid, M. and Ma, K. (2010). Towards Effective Communication with Robotic Assistants for the Elderly: Integrating Speech, Vision and Haptics. *Dialog with Robots, AAAI 2010 Fall Symposium*, Arlington, VA, USA.

[Duke05] Duke, D.J., Brodlie, K.W., Duce, D.A., Herman, I. (2005). Do You See What I Mean?. *IEEE Computer Graphics and Applications*, pp. 6-9.

[Duke09] Duke, D. J., Borgo, R., Wallace, M., and Runciman, C.. Huge Data But Small Programs: Visualization Design via Multiple Embedded DSLs". *Proceedings of the 11th International Symposium on Practical Aspects of Declarative Languages*, pp.31-45.

[Gotz08] Gotz, D., & Zhou, M. (2008). Characterizing Users' Visual Analytic Activity for Insight Provenance. *IEEE Visual Analytics Science and Technology*, Columbus, Ohio.

[Gotz09] Gotz, D., & Wen, Z. (2009). Behavior-driven visualization recommendation. *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, New York, NY, USA, pp. 315-324.

[Hall09] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.,(2009). *The WEKA Data Mining Software: An Update, SIGKDD Explorations*, 11(1).

[Hansen05] Hansen, C.D., Johnson, C.D. (2005). *The Visualization Handbook*. Elsevier.

[Harris00] Harris, R.L., *Information Graphics. A Comprehensive Illustrated Reference*. Oxford University Press, 2000.

[Heer05] Heer, J., Card, S.K., & Landay, J.A., (2005). Prefuse: A Toolkit for Interactive Information Visualization. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 421-430, Portland, Oregon.

[Heer10] Heer, J., Bostock, M., (2010). Declarative Language Design for Interactive Visualization. *IEEE Transactions on Visualization and Computer Graphics,* 16(6): 1149-1156, 2010.

[Keim02] Keim, D. Information Visualization And Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics* 8(1), 1-8.

[Munzner06] Munzner, T., Johnson, C., Moorhead, R. H., Rheingans, P. & Yoo, T. S.. NIH-NSF visualization research challenges report summary. *IEEE Computer Graphics and Applications*, 26(2):20–24, 2006.

[Stolte02] Stolte, C., Tang, D., Hanrahan, P.. (2002). Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. IEEE *Transactions*. on Visualization and Computer Graphics, 8(1):52-65.

[Tufte83] Tufte, E.R. (1983). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, USA.

[Tufte06] Tufte, E.R. (2006). *Beautiful Evidence*. Graphics Press, Cheshire, Connecticut, USA.

[Viegas07] Viegas, F. B., Wattenberg M., van Ham, F., Kriss, J. & McKeon, M. ManyEyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.

[Ward10] Ward, M., Grinstein, G., Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. A K Peters.

[Ware04] Ware, C. (2004). *Information Visualization: Perception for Design*. Morgan Kaufmann.

[Wehrend90] Wehrend, S., and Lewis, C., (1990). A Problem-Oriented Classification of Visualization Techniques, *Proceedings of the 1st conference on Visualization*, *IEEE*, pp. 139-143.

[Witten11] Witten, I. H., Frank, E., Hall, M. A.. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed., Morgan Kaufmann.

---

[i] Home Depot is a large store in the US that sells all items related to home construction and maintenance.