ENABLING ACCESSIBLE CHARTS THROUGH INTERACTIVE NATURAL LANGUAGE INTERFACE FOR PEOPLE WITH VISUAL IMPAIRMENTS

MD ZUBAIR IBNE ALAM

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO
OCTOBER 2022
© MD ZUBAIR IBNE ALAM, 2022

Abstract

Web-based data visualizations have become very popular for exploring data and communicating insights. Newspapers, journals, and reports regularly publish visualizations to tell compelling stories with data. Unfortunately, most visualizations are inaccessible to readers with visual impairments. For many charts on the web, there are no accompanying alternative (alt) texts, and even if such texts exist they do not adequately describe important insights from charts. To address the problem, we first interviewed 15 blind users to understand their challenges and requirements for reading data visualizations. Based on the insights from these interviews, we developed *SeeChart*, an interactive tool that automatically deconstructs charts from web pages and then converts them to accessible visualizations for blind people by enabling them to hear the chart summary as well as to interact through data points using the keyboard. Our evaluation with 14 blind participants suggests the efficacy of *SeeChart* in understanding key insights from charts and fulfilling their information needs while reducing their required time and cognitive burden.

Dedication

I would like to dedicate this thesis to my parents, **Md Shamsul Alam** and **Shameema Alam**, who inculcated a love of science, learning, and knowledge in me since my childhood. Also, a big thank you goes to my wife, **Farah Tabassum** and little son, **Muhammad Bin Zubair**. I am very grateful to them for backing me up with all possible ways to excel in my journey of higher education and reach my goals. Their envisioned dreams for me will be achieved by this accomplishment.

May Almighty bless them endless happiness and healthiness.

Acknowledgements

First of all, I would like to express my sincerest gratitude to my thesis supervisor, **Prof. Enamul Hoque Prince**, for his endless support from the very first day and throughout my masters degree. His valuable feedback and suggestions helped me much to improve the quality of this thesis work. I am very thankful to him for teaching and guiding me to conduct high-quality research in the area of accessibility. It was a great honor for me to work with him for two years.

I would like to thank **Prof. Scott MacKenzie** for being a committee member of my thesis and spending his valuable time on it.

I am thankful to **Jason Obeid**, who started creating the interface that we extended further. Special thanks to **Shehnaz Islam** and **Ahmed Masry** for their support.

Finally, I would like to thank **Prof. Hasan Mahmud** who is an assistant professor at Islamic University of Technology (IUT) in Bangladesh where I completed my BSc in Computer Engineering degree. I started my very first research experience with him during my time at IUT, and learned much from him about HCI.

Table of Contents

Abstract ii

Dedication iii

Acknowledgements iv

Table of Contents v

List of Tables ix

List of Figures x

1 Introduction 1

1.1 Motivation 1

1.2 Our Approach 4

1.3 Contributions 7

1.4 Outline 8

2 Literature Review **10**

2.1 Accessible Visualization on Web 10

2.2 Deconstructing Visualizations 15

2.3 Chart Summarization 16

3 Needfinding Interviews **18**

3.1 Participants 18

3.2 Procedure 19

3.3 Findings 20

3.3.1 Inadequate information with the alt texts 21

3.3.2 Chart data remains out of reach 22

3.3.3 Needs for chart data filtering 23

3.4 Design Implications 25

3.5 Summary 27

4 The SeeChart System **28**

4.1 Overview of *SeeChart* System 28

4.2 Visualization Deconstructor 29

4.3 Natural Language Description Generator 32

4.3.1 Chart Summary Corpus Study 32

4.3.2 Chart Summarization Method 34

4.4 The *SeeChart* Interface 43

4.4.1 Get an Audio Overview of the Chart 44

4.4.2 Access to Chart Data 45

4.4.3 Interactive Summarization 47

4.4.4 Search by Voice Command 49

4.4.5 Control Audio Speed 51

4.4.6 Keyboard Shortcuts 51

5 User Study **53**

5.1 Goal of the Study 53

5.2 Methodology 54

5.2.1 Participants 54

5.2.2 Apparatus 55

5.2.3 Study Design 55

5.2.4 Tasks 56

5.2.5 Procedure 58

5.3 Results 60

5.3.1 Task Performance 60

5.3.2 Post-study Feedback 63

5.3.3 Analysis of user interactions 66

5.4 Summary 67

6 Discussion and Future Work **69**

6.1 Summary of Contributions 69

6.2 Limitations and Future Work 70

6.3 Final Remarks 73

Bibliography **74**

A Appendix **83**

A.1 Script for Needfinding Study 84

A.2 Questionnaires 87

B Appendix **88**

B.1 Script for User Study 89

B.2 Questionnaires 92

B.2.1 Pre-Study Questionnaires 92

B.2.2 Post-Study Questionnaires 93

B.2.3 Charts and Tasks 94

List of Tables

4.1 Extracted information from charts. 31

4.2 Ranked list of identified statistics and data trends according to their occurrence ratio. 35

4.3 List of available keyboard shortcuts in *SeeChart*. 52

List of Figures

1.1 *SeeChart* automatically deconstructs a D3 chart from (A) web page and converts it into an accessible visualization for blind people. The tool generates and (B) presents a textual summary from the extracted data along with other metadata such as (C) the chart title. Users can adjust (D) the length of the summary. The interface allows users to (E) navigate through the data points via keyboard shortcuts and get (F) audio feedback. (G) The voice search feature allows users to search various statistics (e.g., Extrema, Average, etc.) from chart data and the corresponding y-axis value of an x-axis value. 6

3.1 Initial mock-up prototype of *SeeChart* interface. 20

4.1 An overview of the system. 29

4.2 A D3 bar chart (a). D3 deconstructor extracts data fields “Country” and “Number of Fighter Jet” of data types “nominal” and “quantitative” respectively. Also, the axes, including the data field they represent are extracted. The deconstructed representation for the bar chart is a Vega-Lite specification (b) 30
4.3 Architecture of template-based description generator. 36
4.4 Sample bar chart from Statista and its human-written summary as well as summaries generated by our template-based algorithm (both mid and max lengths). 38
4.5 Sample line chart from Statista and its human-written summary as well as summaries generated by our template-based algorithm (both mid and max lengths). 39
4.6 Sample grouped bar chart from Statista and its human-written summary as well as summaries generated by our template-based algorithm (both mid and max lengths). 40
4.7 Sample multi line chart from Statista and its human-written summary as well as summaries generated by our template-based algorithm (both mid and max lengths). 41
4.8 Accessing chart data with keyboard navigation. 46

4.9 Initial summary of a line chart (A), and partial summary based on selected data points (B). 47

4.10 Summary generated from discontinuous sets of data points 49

4.11 Search bar of *SeeChart* tool. 50

5.1 Example of multi line chart and single bar chart for Task B. 57

5.2 Example of single bar chart and multi line chart for Task B. 58

5.3 Mean task completion time for summary and task types (A), chart types (B), and application of interactive summarization. 62

5.4 Percentage of correct answers based on summary type (A), task type (B), and chart type (C). 63

5.5 User study responses to post-study interview. 64

5.6 Sequence of interactions when a chart is presented to the participants. . 66

5.7 Usage of various features and components of *SeeChart* tool. 67

B.1 Charts used in the user study along with corresponding task and reference URL. 95

B.2 Charts used in the user study along with corresponding task and reference URL. 96

B.3 Charts used in the user study along with corresponding task and reference URL. 97

B.4 Charts used in the user study along with corresponding task and reference URL. 98
B.5 Charts used in the user study along with corresponding task and reference URL. 99
B.6 Charts used in the user study along with corresponding task and reference URL. 100
B.7 Charts used in the user study along with corresponding task and reference URL. 101
B.8 Charts used in the user study along with corresponding task and reference URL. 102

1 Introduction

1.1 Motivation

Web-based visualizations have gained a lot of traction because people frequently use them to examine datasets and share crucial discoveries. To assist readers understand data, visualizations are commonly used in news articles, financial reports, scientific papers, and blogs. Particularly during the COVID-19 pandemic, we have seen the need for visualization to predict incoming waves of new infections and make appropriate preparations. It also aided in our understanding of how vaccinations are used all across the world. Numerous websites (such as Worldometer[1] and John Hopkins[2]) release real-time statistics on reported cases of illness, coronavirus-related deaths, and other important information.

Globally, there are at least 253 million people with visual impairments including 36 million people who are blind[14]. Access to information visualizations is often not

[1] https://www.worldometers.info/

[2] https://coronavirus.jhu.edu/map.html

available to people who are blind or visually impaired. Visually impaired users rely on screen-reader tools to use computers and browse the internet. When a screen-reader tool detects HTML tags (such as `<p>`, `<h>`, `<ol>`, `<table>` , and etc.) in a web page, it allows users to move between tags of the same type. Screen-readers that recognize the use of header `<h>` tag allow users to hop between the headers similar to how sighted individuals might scan a page by reading the headings. And when the screen-reader comes across an image, video, or other media, it reads the alternative (alt) text provided in the `alt` attribute of `<img>` tag. If an alternative text is not provided, the screen-reader tool ignores the image completely.

Unfortunately, most of the websites do not provide useful alt texts with images and other media content and blind users do not even get to know about the existence of such content on those websites. In the case of web-based visualizations like charts and graphs, the need for having an alt text is even more crucial as these visualizations depict data through lines or bars which is not always explained in the original article considering that people would take a look at the charts or graphs and analyze the data if necessary. People who have visual impairments often cannot interpret and understand web-based visualizations, as there is little or no accessibility support for them [48, 54]. Providing a useful alt text with charts is different than an image as it engages with numerical data. As a result, even if alt texts exist for charts they do not adequately describe important insights like chart type, axes, or data trends[48] which are critical to

people with blindness [56]. Thus, without alt text or alt text with insufficient information, people with visual impairments are unable to fulfill their information needs from charts.

Even with providing an effective alt text, blind people do not get access to the data presented in a chart unless the data is provided in a table using `<table>` tag that turns on particular screen-reader commands. The user can use their keyboard to move horizontally and vertically through the table when a screen-reader detects the `<table>` tag. Charts can be complex (e.g., multi-series line charts) or may have a large number of data values, and it becomes challenging to translate them into tables. Moreover, navigating through the table cells and comprehending data simultaneously often cause too much information to process and can lead to a cognitive overload [3] for people with blindness [70]. As a result, unlike sighted people who can explore and comprehend web-based visualizations, people who are blind are often given no choice to understand charts, aggravating inequalities in access to information among individuals with disabilities in comparison to those who are not living with a disability [48].

In 1999, a list of standard guidelines known as Web Content Accessibility Guidelines [87]), was published by the Web Accessibility Initiative of the World Wide Web Consortium [84] that defines how to make Web content more accessible to people with disabilities. However, a survey [32] found that more than 80% of websites are not developed following

[3] A cognitive overload is, by definition, when a person is given too much information at once or too many tasks to complete simultaneously, which prevents them from performing or processing the information as they would if the amount were sustainable.

those standards and do not always provide alt texts. Even when there are alt texts embedded with visualizations, they do not follow the prescribed guidelines, namely, the WCAG guidelines [87], the Diagram Center's guidelines [25], the Penn State's accessibility guidelines [64], and CFPB's guidelines [26]. Prior study[76] showed that screen-reader users can extract 61.48% less accurate information from web-based visualization after spending 210.96% more time than sighted people. Beyond alt text, software tools like SAS [2] and research prototypes (e.g., [28, 90]) used an approach called data sonification that utilizes non-speech audio (e.g., piano tones) to convey information where the values of each bar are represented by the pitch of the corresponding piano note. However, studies showed that participants prefer speech over sonification because of cognitive overload [70] and integrating such techniques in a web environment can be challenging [56].

1.2 Our Approach

In this thesis, we aim to address the aforementioned challenges by applying user-centered design approaches by designing accessible visualizations on Web for readers who are blind. We first attempted to understand how people with blindness browse the internet, explore commonly used charts on the Web, and what challenges they face during such exploration. For this purpose, we conducted a needfinding study with 15 participants who are blind since birth or visually marginalized. In this thesis, we use 'visually marginalized' to imply 'blind since birth'.

Our participants of the needfinding study were quite familiar with screen-reader tools (e.g., JAWS [73], NVDA [1]) as they use them to get audio versions of alt texts. The interviewees reported that chart descriptions given as alt texts provide very few insights about the chart. They also confirmed from their past experience that in most cases alt texts for charts do not provide any chart-specific information or insights about the data. Similarly, blind users can rarely access the data points in a chart but even if they get access then it is difficult to get interesting patterns, trends, or outliers by navigating through data points. From the analysis of the interview transcripts, we characterize critical user requirements including 1) the need for providing a textual summary of the chart including important patterns, trends, and outliers to provide the users an overview of what the chart tried to present, 2) facilitating navigation through the chart data points in order to let users examine the data values; for example, the value of each bar of a bar chart, and 3) enabling interactive selection to summarize a subset of the dataset for the purposes of getting insights from a specific region of the chart that the users might find interesting.

Based on these user requirements, we distilled several design implications which guided the design of *SeeChart*, an interactive web-based system that facilitates users with blindness in understanding charts through automatically generated natural language descriptions and explore the chart through the keyboard.

SeeChart works as a Chrome browser extension that scans a web page to identify the presence of charts within SVG (Scalable Vector Graphics) elements. As soon as it detects

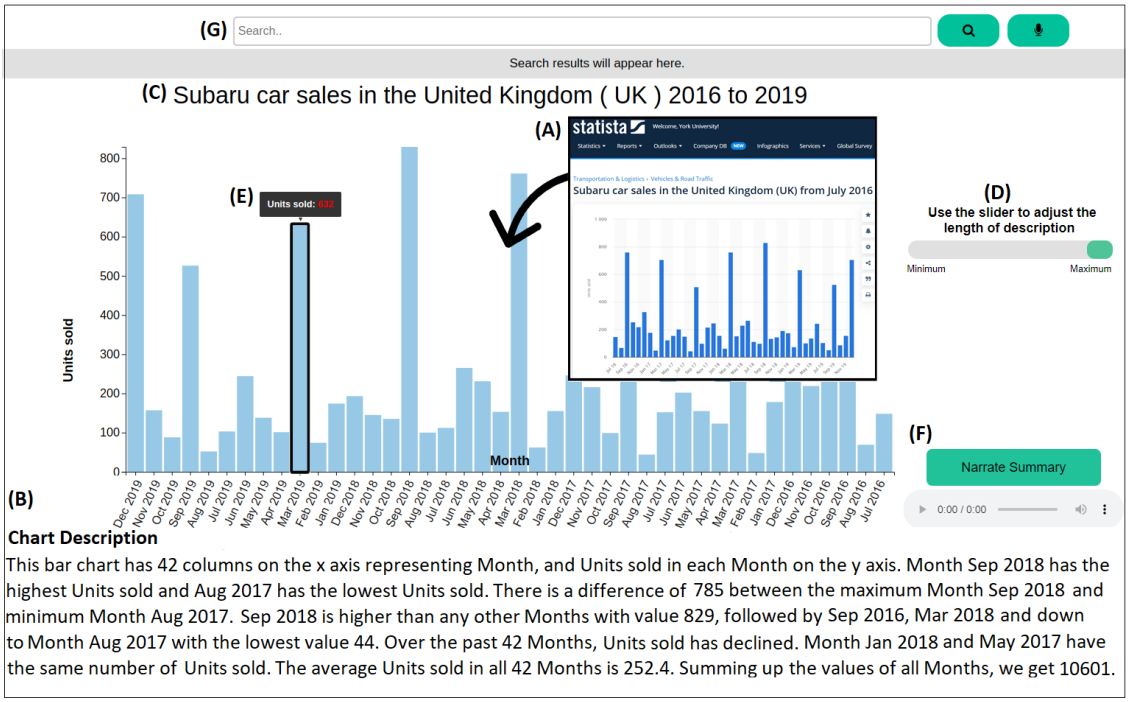


Figure 1.1: *SeeChart* automatically deconstructs a D3 chart from (A) web page and converts it into an accessible visualization for blind people. The tool generates and (B) presents a textual summary from the extracted data along with other metadata such as (C) the chart title. Users can adjust (D) the length of the summary. The interface allows users to (E) navigate through the data points via keyboard shortcuts and get (F) audio feedback. (G) The voice search feature allows users to search various statistics (e.g., Extrema, Average, etc.) from chart data and the corresponding y-axis value of an x-axis value.

a chart on a web page it extracts the underlying data and visual encodings (i.e., how data is mapped to graphical marks such as bars and circles) from the chart that we refer to as *Deconstructing Visualizations*. With the extracted data, *SeeChart* generates a description of the chart using our unique natural language generation (NLG) method that is known as *Chart Summarization* [74, 59, 67, 89]. We have adopted a template-based approach where the tool first calculates significant statistical information (such as maximum, minimum,

trend, etc.) from the chart data and then generates natural language sentences following predefined templates. The system then regenerates an interactive version of the chart that allows users to move through the data points (e.g., bars, dots) with keyboard shortcuts and read their values. It also presents a set of features to let blind users access the data and descriptions easily and efficiently. Figure 1.1 shows an overview of the *SeeChart* interface. We describe more about the system architecture and available features of *SeeChart* in chapter 4.

We evaluated our system through a user study with 14 visually marginalized participants with who used our system to read charts and provided us feedback through semi-structured interviews. The quantitative analysis and qualitative feedback revealed that readers found the *SeeChart* system to improve their ability to understand charts which would be very difficult otherwise.

1.3 Contributions

In this thesis work, our contributions can be summarized as follows:

- To enable effective access to online data visualizations for blind users without requiring them to take on too many additional user tasks, we compiled a set of design criteria that an assistive tool should have by conducting a needfinding study with participants with blindness.

- In order to aid users with blindness in exploration and analysis, we designed and developed the *SeeChart* tool, which combines a natural language generation algorithm to generate a summary of a chart with interactive access to chart data.

- The results of our evaluation study and participant feedback revealed that blind users were able to use the *SeeChart* tool with minimal training and they were 6.25% more accurate when locating significant statistical data (such as the maximum or trend) when using the summary produced by *SeeChart* than when using a human-authored summary.

- Finally, we discuss the key takeaways from the study by explaining the issues users confronted and potential future approaches for further creations of interactive natural language interfaces to serve readers with visual impairments.

1.4 Outline

The remaining sections of this thesis are organized as follows. Chapter 2 provides a survey of the literature on accessible visualization, visualization deconstruction, and chart summarization. We go over the chart deconstruction technique we employed in the *SeeChart* tool in that chapter. The needfinding research we carried out to establish the design principles of the *SeeChart* tool and better understand user requirements is covered in Chapter 3. We go into great detail about the *SeeChart* tool's architecture in chapter

4. This also applies to the method we used to find and extract chart data, as well as our special template-based summary generation. Chapter 6 concludes with a discussion of the tool's shortcomings, the proposed course of action, and a concluding observation.

2 Literature Review

Our work is based on following main areas of prior work: (1) accessible visualization, (2) deconstructing visualizations, and (3) chart summarization. In this chapter we discuss about the existing research work, developed tools that addressed the similar areas.

2.1 Accessible Visualization on Web

The ability of people with disabilities to use a piece of equipment, service, or vehicle is known as accessibility [4]. Regardless of technology, language, or location, users with a wide variety of sight, hearing, and cognitive abilities should be able to use the Web because it is primarily built to operate for everyone. In order to ensure that no one is deprived from using these services, it is crucial for developers to create websites of the understanding the requirements of the people with disabilities.

While there have been significant research and development on improving accessibility on Web, very few works have focused specifically on chart accessibility. Recently, Kim

[4] https://en.wikipedia.org/wiki/Accessibility

et al. [54] defined the design space for chart accessibility and identified the current knowledge gap in this research space. Elavsky et al. [36] enlisted a set of heuristics (e.g., Small text, No interaction cues, No title or summary, etc.) that facilitate designers to evaluate the accessibility of interfaces developed to present visualizations. Screen-reader tools such as JAWS [73], NVDA [1] support people with visual impairments to navigate through web pages and read texts through keyboard shortcuts. To further improve web accessibility various guidelines have been developed. For example, W3C standards [84] established the Web Content Accessibility Guidelines [87] so that people with visual impairments can navigate through websites. The guidelines also provide examples of alternative texts which are invisible text block that can be embedded with image elements in HTML with the `alt` attribute. There are also specific guidelines on formulating alt text for various graphics including charts and graphs, such as the guidelines from WCAG [87], Diagram Center's guidelines [25], the Penn State's accessibility guidelines [64], and CFPB's guidelines [26]. Yet, many charts on the web do not accompany alt texts at all. As a result, most of the charts are entirely invisible to the screen-reader tools [91] and even if some charts come with alt texts, such texts often do not follow the above guidelines and fail to provide important information like chart type, axes, or data trends [48]. Chundury et al. [29] attempted to understand how individuals with blindness perceive surroundings using non-visual senses and based on their findings, they outlined several design considerations for accessible visualizations. Murphy et al. [61] ran an

empirical study involving 30 blind and partially sighted computer users to find out the day-to-day challenges they face while using the Internet.

A key question here is what information should be presented in a textual summary of a chart for effective comprehension by users with blindness. A recent interview study involving people with visual impairments revealed how they want to utilize visualizations and how they draw the image in their head by listening to alt texts [48]. From the analysis of interview data, the authors recommended that the alt text should include chart type, axis labels, data trends, and users should have access to data points. Another study that collected natural language summaries and analyzed them with both sighted and blind users concluded that the summaries should focus more on chart's overall trends and key statistics rather than explaining its visual constructions only [56]. In this work, we have considered these recommendations along with findings from our needfinding studies to inform our chart summarization approach.

There have also been some automatic techniques for navigating charts and tables through textual summaries. The iGRAPH-Lite system [39] provides chart accessibility for blind users via generating captions and supporting navigation through the chart using the keyboard. It uses templates to provide a short description of what the chart looks like. They found that the metadata such as chart title, axis labels, and chart types are important information to blind readers. EvoGraph [75] is a jQuery plugin that aims to create screen-reader friendly charts with information like chart title, data values, and

statistics such as mean and standard deviation. The SIGHT system[34, 23] is an interactive accessibility system that presents brief summaries of charts with their salient features and allows users to access statistical information (e.g., extrema, change in trends, etc.) of the charts through a menu interface. Others have utilized the automatically extracted data from chart images to navigate through the recovered data table [27]. Overall, this work focused on helping people navigate through data points through audio descriptions without providing high-level insights conveyed by the chart. Sharif et al. developed VoxLens [77], an interactive JavaScript plug-in that uses multi-modal approach to provide a comprehensive summary of a visualization. VoxLens allows users to interact with the visualization verbally by using pre-defined commands. Tools made for developers are often not welcomed by them due to compatibility issues and the end users are deprived of the intended benefits. We aimed to make sure that the developers do not need to take any action to make graphics accessible by developing a browser plug-in that will make millions of existing visualizations accessible.

Beyond natural language-based audio feedback, others have considered sonification (the mapping of data to non-speech sounds) [77, 28, 90, 2] and tactile systems like braille displays [57]. For example, SAS [2] uses sonification with keyboard navigation where values of each bar is represented by the pitch of the corresponding piano note. Sio et al. [78] proposed a heuristics-based approach combining sonification with textual description to generate audio data narratives from a time-series dataset and Infosonics [44] presented

an unconventional method to present the audio analogue of an infographics combining interactive sonifications, spoken introduction, and harmonious notes. Wang et al. [85] conducted experiments to find how different auditory channels affect the perception of visualization among people with visual impairments and they reported that pitch was the most instinctive channel to understand all data types (e.g., quantitative, ordinal, and nominal). However, data sonification can lead to cognitive overload [70], and integrating such techniques in an online environment is challenging [48]. Therefore, in this work, we are focusing on natural language summaries of charts instead of sonification.

Apart from alt text-based or audio descriptions, providing source data in tabular format is another recommended approach to make data accessible. Brown et al. [22] outlined a set of design guidelines to present tables using non-speech audio. While the tabular representation of a chart provides direct access to data but fails to communicate the general overview of the chart [68], [76]. Cognitive overload commonly occurs while navigating a data table by audio feedback [52], [83] as the audio feedback only provides the numerical value without associating column and row headers. Therefore, users have to map the value to the header names while traversing through the table. In *SeeChart*
, we aid users by providing them with more information when user checks a data point. For example, our tool will say *"In year 1988, the Inflation rate compared to previous year was, 6.94"* rather than only mentioning the data value thus helping the user to process the information faster.

2.2 Deconstructing Visualizations

The goal of deconstructing a chart is to extract the underlying data, the marks (e.g., points, bars) along with their visual attributes (e.g., color, position), and visual encodings that describe the mappings between them. Early work introduced semi-automatic systems to extract the data from the bitmap chart images [72, 49]. Subsequently, others proposed a fully automatic chart data extraction pipeline, however, their methods rely on various heuristics which may not work for many real-world charts and the performance was limited [27, 55]. ChartOCR [58] is another automated system which extracts data from real-world charts with better accuracy but it only predicts the raw data values of marks without extracting visual encodings. Other works focused on using combinations of image processing techniques [59], extracting visual encodings [65] from map charts [66] and scatter plots [30].

Harper et al. [41, 42] developed a technique to extract underlying data from D3 visualizations. Extracting data and visual encodings are relatively easier for D3 charts because the developer can bind the data within the Document Object Model (DOM). Therefore, we applied the D3 deconstructor in our initial work. As deconstructing charts in bitmap image format becomes more accurate they can be used in our *SeeChart* tool to create accessible visualization.

2.3 Chart Summarization

Outlining an effective summary from chart data is important in order to establish the picture a chart aims to draw on someone's mind. In order to find out which information should be prioritized to present and how much information should be laid out in the summary we carefully examine the existing research work and describe them here at length.

Researchers propose various natural language generation (NLG) methods to describe how the chart encodes data and explain "insights" in textual form for understanding charts. Early works [60, 39] followed a planning-based architecture [69] and used templates to generate texts. These systems only explain how to read the chart and do not convey key insights from the chart. Commercial systems like Quill[67] and Wordsmith[89] and research prototypes Voder[81] and Datasite[33] computed key statistics (e.g., extrema, outliers) to present facts from charts. Demir et al. [35] used Bayesian Inference System [37] to generate summaries from XML(Extensible Markup Language) representation of a bar chart images. ChartVi [59] generates summaries by computing statistics using image processing-based extraction of textual information from chart images and the accuracy of such an approach lies exclusively on the textual information provided in the input image. The SIGHT system [23] also used bayesian network to identify the intended message of a graphic. Datashot computes a variety of data facts covering statistics and data trends from

data tables [86].

One notable limitation of the abovementioned approaches is that the sentences generated using pre-defined templates may lack variations in terms of syntactic structure and lexical choices compared to data-driven models. Therefore some recent works applied data driven deep learning models based on transformer [62, 74] or LSTM based encoder-decoder architecture [79]. While some of these works managed to generate very fluent summaries, they also suffered from hallucinations and factual errors as well as difficulties in correctly explaining complex patterns and trends in charts. In our needfinding studies with users, we learned that people prefer factually accurate statements even if they may lack variations over incorrect facts with fluent texts. Therefore, we developed a template-based approach for chart summarization.

3 Needfinding Interviews

We carried out a needfinding study to learn how blind people try to read visuals and how future aids could successfully support their jobs. We paid great attention to how users with blindness typically interpret visualizations on websites, the difficulties they encounter, and the features they would like to see in an accessible visualization tool. We sought to investigate how prior research tools and methodologies had influenced how blind people understood visualizations. During the need-finding interviews, we also hoped to receive early comments and suggestions on the *SeeChart* prototype.

3.1 Participants

For the needfinding study, we recruited 15 participants (9 female, 6 male, age range 45-55 years) who identified themselves as visually marginalized. In order to find a diverse range of population, we shared our study-advertisement on social media platforms (Twitter, Facebook, and LinkedIn) and also engaged with several local and national non-profit organizations for blind populations including Canadian Council of the Blind

[63], Canadian National Institute for the Blind [31], and Fighting Blindness Canada [18]. The participants were proficient with screen-reader tools such as JAWS and NVDA for browsing Web. Seven of them occasionally encounter charts and graphs (several times a month) on websites and articles while others encounter them less frequently. Each participant was assigned an ID (from P01 to P15) randomly.

3.2 Procedure

During the needfinding study, we asked the participants open-ended questions regarding their experiences, challenges, needs, and requirements for reading visualizations. We started by asking "*How do you try to understand a chart that appears in a document or a web page?* and "*How important do you think it is for you to understand data visualizations?*". We also wanted to know if they find the generally provided alt texts helpful to understand visualizations. Finally, we asked them if they use any specific computer-based tool to explore data visualizations and what features they would like to have in an accessible tool for visualizations. On the basis of the interviews, we developed a mock-up prototype of *SeeChart* tool (shown in figure 3.1) to further comprehend the technical requirements in another course of follow-up interviews with each participant and received their feedback. The prototype was equipped with several sample charts with simple user interactions (e.g., reading data point value of a chart by pressing keyboard arrow keys, and hearing description of the chart). We prepared a demo website to host the

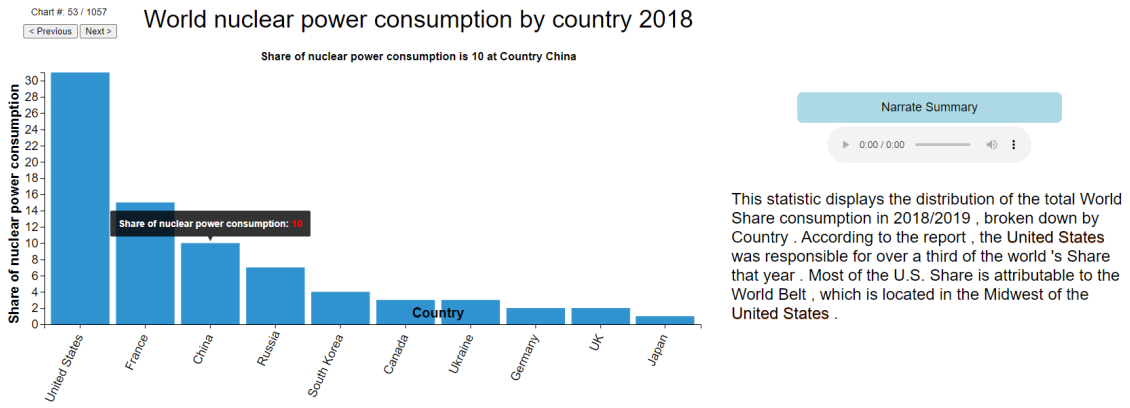


Figure 3.1: Initial mock-up prototype of *SeeChart* interface.

tool and shared the link with the participants. They used the tool by navigating to the link on their own computers.

The semi-structured interviews were conducted over video conferencing and each session lasted 40-45 minutes. As compensation, each participant was gifted 20$. We audio-recorded interviews and transcribed them for analysis. In particular, we used a spreadsheet application to thematically analyze the interview transcripts and notes taken during the interviews using the open coding method [51]. The key themes from our analysis are laid out in the following section.

We present the need-finding questionnaires in Appendix A.

3.3 Findings

We now present the primary themes found from our analysis along with the user requirements (UR) we identified from the analysis to guide the design of our *SeeChart*

system.

3.3.1 Inadequate information with the alt texts

Users with blindness generally rely on screen-reader tools to read a web page by going through tags (e.g., `<h1>`, `<p>` etc.) and they control the navigation and pace according to their needs. Whenever the website provides alternative texts for visualizations using `alt` tags, the users listen to those texts to make sense of the visualizations. Most participants reported that usually alt texts for charts are very generic (summarizing the topic) and do not provide important information like chart type, visual encoding, or any insights from the chart. This finding is consistent with the study from Jung et al. [48] who reported that the majority of alt texts in their collected corpus do not report important information such as chart type and data trends. Participants suggested that alt texts should provide important information such as chart types, the visual encoding structure (e.g. what each axis represents), and summarize key insights covering patterns, trends, and outliers from the visualization (*UR-1*).

Eight participants mentioned that there are many occasions where the web page presents the visualization in bitmap image format without providing any alt text or other textual summaries. In such cases, they may not be even aware of the existence of visualization and skip it inadvertently. They suggested that an assistive tool should notify users of the presence of a chart so that they can delve into the chart (*UR-2*). P03 said,

> "*Most of the time if it's a chart, there's no way to interpret it so I just have to ignore it even if it sounds interesting to me. Overall, the situation is so upsetting that I have stopped expecting to learn useful information that is laid out in a chart. It really makes me feel like I am being deprived of publicly available information because I cannot see.*"

3.3.2 Chart data remains out of reach

Screen-reader users attempt to conceptualize a chart using table navigation technique where columns represent the data attributes encoded in the chart. They use keyboard arrow keys to navigate through the table by first reading through the column headers and then reading through individual data values in each row. According to P05,

> "*If the data is given on a table underneath the chart, I would use JAWS key commands to navigate the table. Then I would read the heading first and use the arrow keys to navigate the columns. Otherwise, I get nothing.*"

Since the users have formed their mental model in accessing raw data using table navigation techniques, it is important to provide a positive transfer effect while designing the navigation for chart data (*UR-3*). Feedback from our participants supports the findings from previous research [76], [68], [83] as tables do not provide an overview of data and screen-reader users find it difficult to keep track of cells and columns. P05 also mentioned,

> "*For large tables, I find it extremely hard to navigate through the data and remember them. I need to associate the data value with the column header and that makes it very difficult when there are many columns of data.*

Eleven of the participants indicated that they frequently visit websites containing charts. However, underlying data tables are rarely provided with online charts. While

sometimes the alt text may mention extremum data values, they may not mention other data points. As a result, a large subset of the data points from the chart are left in darkness and users with blindness do not have a way to drill down to these data points. Similarly, other accessibility techniques like data sonification are also limited in revealing the precise raw data values [48] and prone to cause cognitive overload [70]. P11 who used the data sonification features with SAS mentioned the challenges in discriminating between two tones that indicate two different data values of a pie chart. She said,

> "*SAS makes different tones for different points as it goes up or lower but it does not give me specifics. I don't know what 25% sounds like. Also when there is a group of similar data, it becomes almost impossible to differentiate between them by hearing a tune.*"

3.3.3 Needs for chart data filtering

Three participants who encounter charts occasionally, mentioned that they often need to filter data and to extract partial information from a big chart. P07 said,

> "*A sighted person would be able to visually perceive the whole chart to get a quick overview. But I do not get that privilege. For complex charts, I ask my colleagues to assist me with getting interesting information as there is no way of data filtering.*"

In this context, an assistive tool should encourage this aspiration for exploring chart data by allowing users to select a portion of the chart that they are interested in and summarize the selected portion of dataset and drill down to detailed values on demand (*UR-4*).

3.3.3.1 Needs for Interactive features

When we asked questions specifically about what participants would expect from a future tool for accessible visualization, they suggested introducing interactive techniques that are consistent with their currently used tools. Users with blindness frequently use keyboard shortcuts for website navigation using screen-reader tools and a new tool should introduce interactive features that are learnable and consistent with their current forms of interactions with Web (*UR-5*). Participant P07 said,

> "*I always find it really challenging to get familiar with new technology or tool. I love tools that do not require a lot of training and can be used with a little bit of practice. You can follow the features that we use with JAWS and use similar ones in the assistive tool.*"

We received some useful suggestions by allowing the participants to use the *SeeChart* prototype. Several participants also mentioned that controlling the information granularity in the chart summary is important to them. Often the description of the visualizations is too long for a user to process as continuously listening to a long summary involving numbers can lead to cognitive overload. Therefore, the user should be able to control the length of the chart summary, hear each sentence individually, and replay a sentence (*UR-6*). P03 said

> "*The tool plays the chart summary without any break and it is very difficult to concentrate on the numbers and labels. If there was a way to hear the description line-by-line or word-by-word, I am sure it would be very much advantageous.*"

The prototype was not providing any navigational cue while participants were reading

chart data values using arrow keys. It was puzzling the participants as seven of them found it very difficult to grasp where in the chart they were at and they suggested providing them with some cues so that they get an idea of their current position within the chart (*UR-7*) (e.g., whether they are at the beginning or end of the chart).

3.4 Design Implications

Based on our needfinding study we have identified the following design implications (DI) which formed the basis of the *SeeChart* system. Each of them was derived from one or more user requirements, as follows:

1. **Give users access to data with intuitive navigational cues:** The tool should allow users to navigate through the data points in the chart via arrow keys and play each data value along with the related data attribute. The tool should also provide a positional awareness (e.g., beginning, middle, or end of the chart) as they traverse through the chart (to address *UR-3, UR-7*). The tool should also let users jump to the beginning or end of the chart at any time.

2. **Provide chart details on demand:** When a chart is detected on a web page, the tool should notify the user. If the user is interested in further exploring the chart, more detailed information should be accessible through keyboard shortcuts (to address *UR-1, UR-2*). Also, users should be able to read the summary of the chart first so

that they can decide if they are interested to go through the data points to examine the chart for better analysis.

3. **Provide interactive selection of interesting area of chart:** The tool should allow users to select a range of data points following the same way they select multiple cells of a table through keyboard and generate a summary of data points based on the selected data points (to address *UR-4*). Users should get feedback as they select the data points and the description should be generated in real time. To confirm that the information displayed in the summary is accurate and that they did not choose any items they did not want, users should be able to check the data points they chose from the sources from which the summary was generated.

4. **Provide more control for adjusting the information:** The tool should provide a concise textual summary of the chart with what the chart represents and what are some key insights in the chart with appropriate punctuations. The user should be able to listen to the summary sequentially and pause/skip sentences whenever needed to comprehend the information easily without getting cognitive overload. Finally, the user should be able to choose the desired length of description (addresses *UR-6*).

5. **Keep interactions learnable and consistent:** To enhance learnability, the interface features of the tool should be intuitive and follow familiar shortcuts. Blind users

mostly rely on keyboard shortcuts while browsing Web and they use a number of shortcuts that are assigned to screen-reader tools. Introducing many additional shortcuts would be difficult to remember. Therefore, it is important to carefully choose a small number of shortcuts to interact with visualizations (to address *UR-5*).

3.5 Summary

The needfinding study guided us to understand how people with blindness use screen-reader tools to access the internet. It also helped us to outline the existing challenges and analyze the requirements of a tool to make online visualizations accessible. That ultimately directed us to design and implement the *SeeChart* tool. In the next chapter, we discuss the implementation details of the system.

4 The SeeChart System

We revised the system to reflect the design implications (DI) that we outlined based on the needfinding study after receiving feedback on our initial mock-up prototype. Each feature was created with the understanding that it would not require users to become highly adept at learning new user actions to access it, and that the keyboard shortcuts needed to utilize the *SeeChart* tool would not require more than two keys. In the sections that follow, we go into further detail about each system module.

4.1 Overview of *SeeChart* System

Our system has three main modules. Given a web page with an SVG visualization loaded in the Google Chrome browser, the *Visualization deconstructor* analyzes the SVG representation of the chart to recover the data, visual attributes of graphical marks, and visual encodings that describe the relationship between them. The *chart description generator* then takes the data, marks, and visual encodings as input and automatically generates the chart summary as output. Finally, the *user interface* reconstructs the

visualization with summary descriptions as well as provides several interactive features for selection, navigation and browsing through the chart.

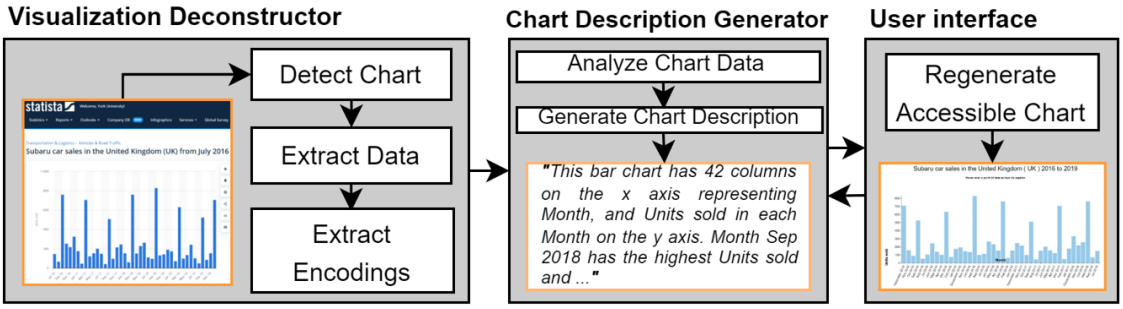


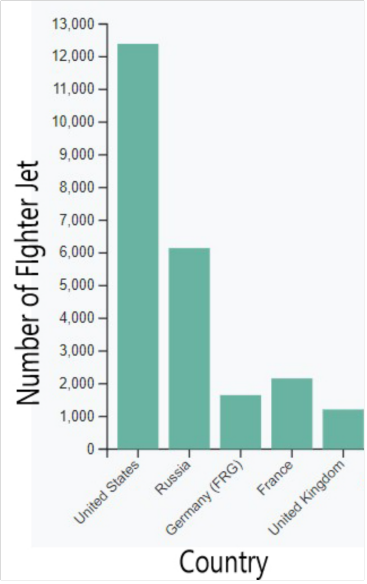
Figure 4.1: An overview of the system.

4.2 Visualization Deconstructor

The first phase of the system is to detect the presence of a D3 chart on the web page and then extract the underlying data from it. Here we describe the method we followed to accomplish that.

We developed the visualization deconstructor as an extension for Google Chrome browser which injects code into the JavaScript environment for the given web page. It first detects a visualization in SVG format and then extracts the underlying data and visual encodings. We have primarily focused on deconstructing visualizations created using the D3.js, a JavaScript library [21] which has become very popular in constructing web-based visualizations.

We utilized the D3 deconstructor tool [41, 45] which analyzes the relationships be-



a) Sample Bar Chart

```
"mark" : "bar",
"encoding": {
"x": { "field": "Country",
       "type": "nominal"}
"y": { "field": "Number of Fighter jet",
       "type": "quantitative"}
"color": { "field": "Country",
           "type": "nominal"}}
"data": {
"values": [
{
    "Country": "United States",
    "Number of Fighter jet": 12252
},
{
    "Country": "Russia",
    "Number of Fighter jet": 6148
},
{
    "Country": "Germany (FRG)",
    "Number of Fighter jet": 1653
}, ...]
```

b) Vega-Lite Specification

Figure 4.2: A D3 bar chart (a). D3 deconstructor extracts data fields "Country" and "Number of Fighter Jet" of data types "nominal" and "quantitative" respectively. Also, the axes, including the data field they represent are extracted. The deconstructed representation for the bar chart is a Vega-Lite specification (b)

tween data values and the values of each mark attribute to recover the encodings that map between them. Figure 4.2 shows an input bar chart and the resulting deconstruction output in Vega-Lite format. In addition to deconstructing D3 visualizations, we also deconstruct visualizations created with the highcharts library[43] where we automatically analyze the content of the SVG to extract the data and visual encodings of charts. Deconstructing visualizations from highcharts[43] was relatively straightforward because charts are generated with declarative specifications and the library internally assigns specific class names for chart elements in the generated SVG chart. For example, when data

values of attributes appear as text labels in the chart they are grouped together under <g> with `class = highcharts-data-labels`. Similarly, we retrieve other chart information such as axes labels using specific class names (e.g., `highcharts-yaxis`). In order to construct the underlying data table of the chart, we simply iterate through each of this <g> elements with `class = "highcharts-data-labels"` to find data values of each data attribute encoded in the given chart. When data-labels are absent we use the bounding box information of data encoding marks which are children of `<g class="highcharts-series-group"`. We then obtain the scale information based on the axis labels and tick marks to recover the data values for these data encoding marks. We have deconstructed a variety of bar charts, line charts and pie charts created by the highcharts library. Table 4.1 shows the information we extract from the chart.

Table 4.1: Extracted information from charts.

Extracted Information
Title
Graph Type
Column Type
Labels
Graph Type
Column Type
Data

4.3 Natural Language Description Generator

In the next phase, the natural language description of the chart data is generated based on the chart type. In this section, we describe how the description generator was populated along with the detailed algorithm. Finally we present some sample charts with corresponding descriptions generated by the *SeeChart* tool.

This module automatically summarizes the given chart explaining how the chart encodes data and what are some key takeaways from the chart. In order to learn what should be the content (*what to say*) and structure (*how to say it*), we analyzed an existing chart description corpus for screen-reader users[56]. Informed by the corpus study, we generate chart summaries followed the natural language generation (NLG) architecture which has been widely adopted for generating text summaries from numeric input data [69]. We discuss the corpus study followed by our chart summarization method below.

4.3.1 Chart Summary Corpus Study

In order to understand what information is important and should be included in the chart summary, we rely on the recent study with blind people by Laundgard et al. [56]. They curated 50 visualizations comprised of 18 bar charts, 21 line charts, and 11 scatter plots from MassVis dataset [19, 20] and were able to gather a corpus of 582 summaries consisted of 2,016 sentences. Based on the semantic content, the sentences were categorized into

four levels: (1) elemental and encoded properties such as chart type, title, axis labels; (2) statistical concepts and relations such as extremas, comparisons, and correlation; (3) perceptual and cognitive Phenomena such as trends and patterns; and (4) contextual and domain-specific insights. In our analysis, we mainly focused on understanding what information is frequently reported in Level 2 and 3 type of contents which describe statistics as well as patterns, trends and outliers in the given chart. We did not focus on Level (4) because generating contextual or domain-specific description that is not given in the chart is extremely challenging and are not of primary interest to blind people[56]. In contrast, the Level 1 content type, which describes the chart type, title, axes labels, range, and encoding channels, is relatively straightforward to generate, therefore we manually build templates for this level.

For our analysis, two authors from our research team independently went through all level 2 and 3 types of summary content and came up with categorization of information presented in these sentences (e.g., *extrema*, *comparisons[relative]*, *trend [global]]*, where each category represents a function along with possible parameters). The two annotators resolved each inconsistency by explaining their categories and then worked together to reach a final consensus on what should be the category for each sentence. Based on this analysis of 464 sentences about bar charts and 818 about line charts For each chart type, we identified the frequently reported statistics and categorized each sentence into one of those categories. The 'others' represent the category that does not fall into any of the

identified major categories (e.g., reporting a random data point in the chart).

We ranked the statistics and trend categories based on their frequency of occurrence as shown in table 4.2. Among the 272 distinct summaries of 11 single bar charts, majority of them highlighted on *extremum* (min/max) values, relative or absolute *comparison* between data points, *order* of data values, *derived* values (e.g., average, total), and other statistical information. Almost similar kind of categories were found in 192 summaries of 6 grouped bar charts. Line charts generally depict changes of trends over a period of time and this was no different as about 60% of 818 summaries of 10 single and 11 multi-series line charts describe positive or negative *trends*. Other notable statistics were about *extrema*, *comparison*, *order*, etc.

4.3.2 Chart Summarization Method

To generate summaries from chart data, we followed the natural language generation (NLG) architecture proposed by Reiter which has been widely adopted for generating text summaries from numeric input data [69]. Following this data-to-text generation architecture, our NLG method works according to the following stages as shown in Figure 4.3.

Stage 1: Analyze data and visual encodings. The goal at this stage is to detect basic statistics and patterns in the numeric input data such as maxima, minima, overall trend in the chart, or a short-term trend within a range of data items (e.g., spikes). Some

Table 4.2: Ranked list of identified statistics and data trends according to their occurrence ratio.

Single Bar		**Grouped Bar**	
Stat Category	*Ratio*	*Stat Category*	*Ratio*
Extrema [min/max]	57 %	Global Extrema [min/max]	36 %
Comparison [relative]	12 %	Trend [global]	18 %
Order [rank]	8 %	Comparison [relative]	13 %
Trend	7 %	Extrema [max difference]	13 %
Computed derived value [avg/sum]	6 %	Trend [local]	7 %
Order [rank/position]	4 %	Order [rank/position]	4 %
Comparison [absolute]	2 %	Local Extrema [min/max]	2 %
Others	4 %	Others	7 %
Single Line		**Multi Line**	
Stat Category	*Ratio*	*Stat Category*	*Ratio*
Trend	62 %	Global Trend	48 %
Extrema [min/max]	22 %	Order [position/rank]	24 %
Comparison [absolute/relative]	10 %	Extrema [min/max]	13 %
Extrema [difference]	4 %	Extrema [local/global]	5 %
Others	2 %	Comparison [absolute/relative]	2 %
		Others	8 %

examples of the resulting statistics and patterns are shown in Figure 4.3 (top block). For characterizing trends, we follow Hullman et al. for identifying visually salient points in line chart [47]. However, their method included some domain-specific features (e.g., stock price and volume traded) to calculate visual saliency of each data point. We simplify the approach by considering the variable that captures the change (slope) over data point in the previous time point by computing the absolute distance between the value for the current date point and the value for the previous time point. We then normalize the value of change by dividing by the maximum change value across the series. Based on the calculated slopes, we compute the global trend over the whole chart as well as local trends

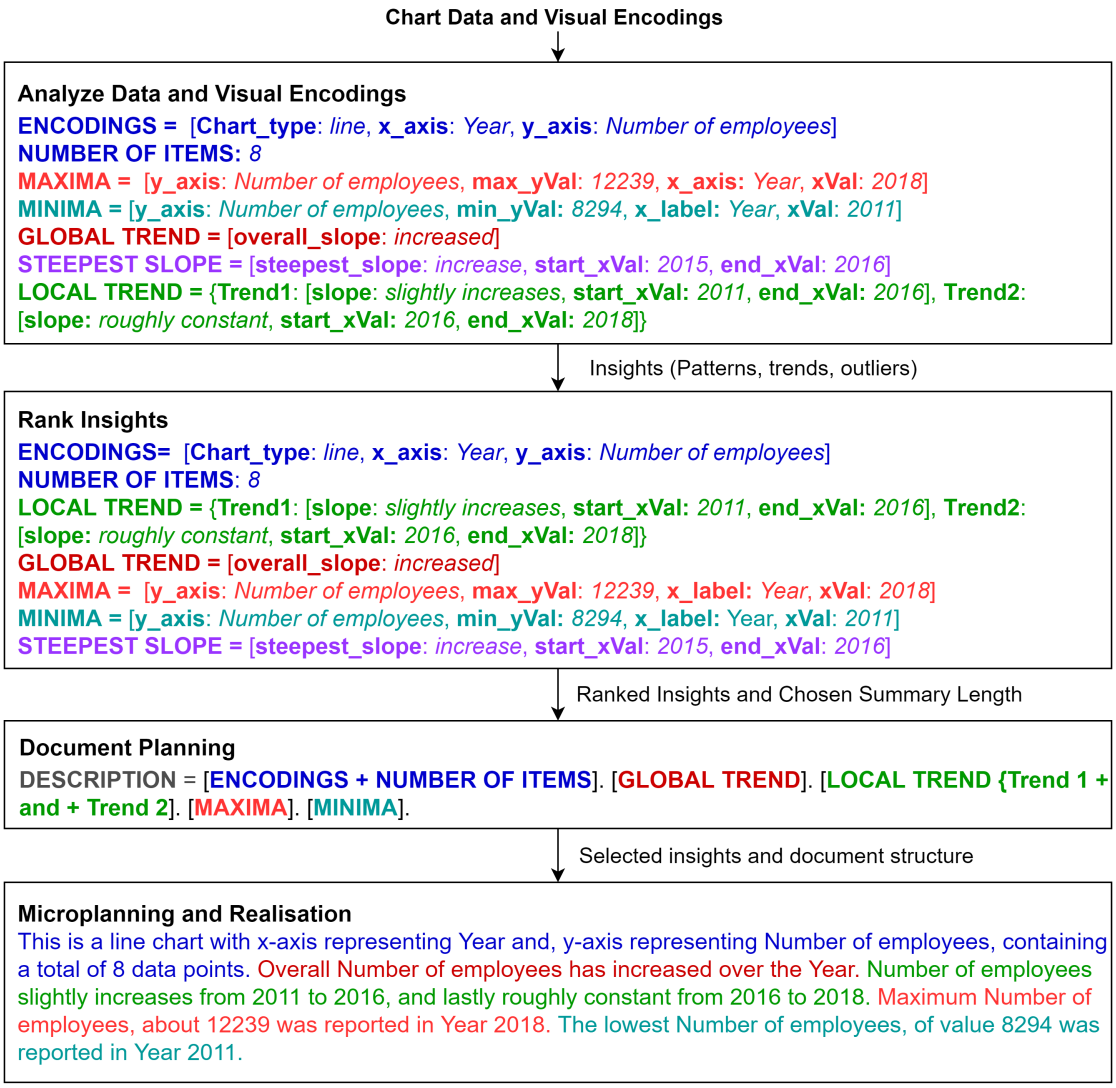


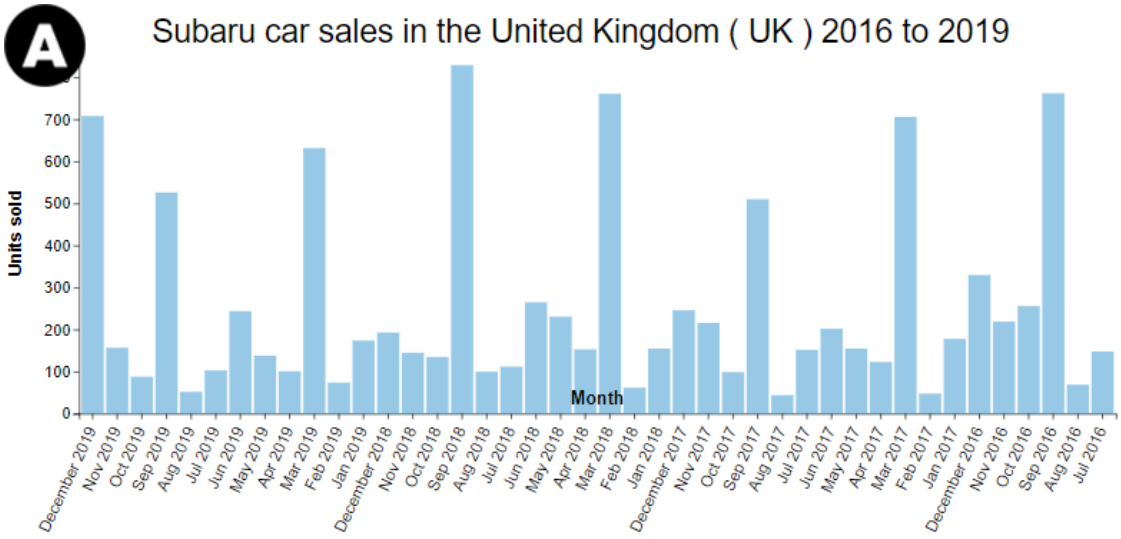
Figure 4.3: Architecture of template-based description generator.

over different portions of the chart. We also used the extreme values in each series of the chart and the change of magnitude and the direction of a slope (e.g., from increase to decrease) at different points to determine the range of a local trend.

Stage 2: Rank insights. At this stage, the goal is to assign an importance for each of

the insights and then rank them according to the importance measure. For ranking each message, we use a simple measure of how frequently the corresponding insight appears in chart summaries for that chart type. According to Table 4.2, people tend to report extrema and comparisons between values for bar chart, whereas for line charts people focus more often on trends. Nevertheless, in future other measures of importance may be combined with this frequency information for ranking statistics.

Stage 3: Document planning. In this stage, the method determines the content (which messages should be included in the final summary) and the structure of the summary (what would be the discourse structure of the text, e.g., how to combine multiple messages into a sentence and how to determine sentence boundary). We first use a set of heuristics to combine related statistics, for example, maxima and minima values into a single sentence. Similarly, multiple short term trends are joined together in a sentence using discourse connectives such as 'and', 'however', 'after that' etc. We then first present the introductory message containing the basic chart information and visual encoding channels such as graph title, type of graph, label of X and Y axes which are identified as important for blind people by the study from Ferres et al. [38]. The remaining messages are presented according to their ranking. For example, when the maximum summary length is 3 sentences, we present the intro sentence (describing chart type and visual encoding channels) followed by the top two messages identified after ranking statistics, which are *extrema (min/max)* and then *comparison (absolute/relative)*.

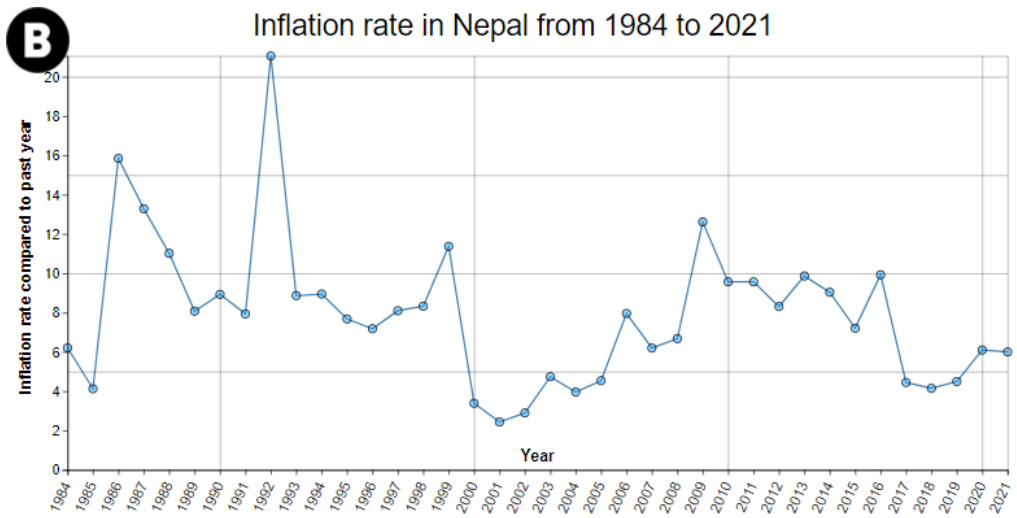


Human-Written: This statistic shows the monthly amount of cars sold by Subaru in the United Kingdom (UK) between July 2016 and December 2019. Peaks in registration numbers were recorded in March and September of each year, which was due to the issuing of license plates by the Driver & Vehicle Licensing Agency in those months. In December 2019, 708 new Subaru cars had been sold.

Template-based (Moderate length): This is a bar chart. It shows Units sold for 42 number of Months. The maximum Units sold 829 is found at Month Sep 2018. The minimum is found at Aug 2017 where Units sold is 44. The highest Month Sep 2018 has 785 more Units sold than the lowest Month Aug 2017.

Template-based (Long length): This bar chart has 42 columns on the x axis representing Month, and Units sold in each Month on the y axis. Month Sep 2018 has the highest Units sold and Aug 2017 has the lowest Units sold. The second highest Units sold is in the Month. There is a difference of 785 between the maximum Month Sep 2018 and minimum Month Aug 2017. Month Sep 2018 is higher than any other Months with value 829, followed by , and Mar 2018. Down to Month Aug 2017 with the lowest value 44. Over the past 42 Months, the Units sold has declined. Month Jan 2018, and May 2017 have the same Units sold. The average Units sold in all 42 Months is 252.4. Summing up the values of all Months, we get total 10601.

Figure 4.4: Sample bar chart from Statista and its human-written summary as well as summaries generated by our template-based algorithm (both mid and max lengths).

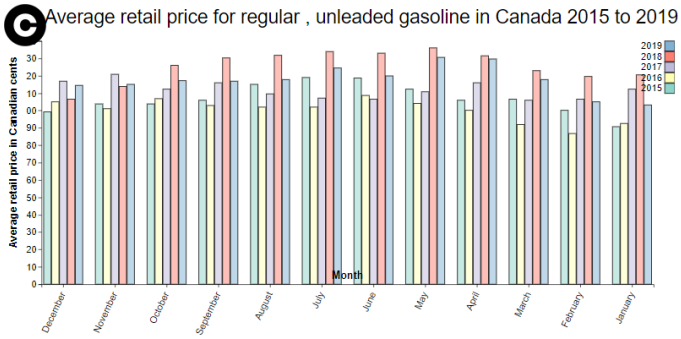


Human-Written: In 2018, the average inflation rate in Nepal was at 4.15 percent, a slight drop compared to the previous year. The inflation rate is calculated using the price increase of a defined product basket. This product basket contains products and services on which the average consumer spends money throughout the year.

Template-based (Moderate length): This is a line chart with an x axis representing Year and a y axis representing Inflation rate compared to previous year, containing a total of 41 data points. In general Inflation rate compared to previous year has fallen over the Year. The most drastic increase took place within the Year 1991 and 1992. Maximum Inflation rate compared to previous year, about 21.06 was reported at Year 1992 The lowest Inflation rate compared to previous year, of value 2.44 was reported at Year 2001.

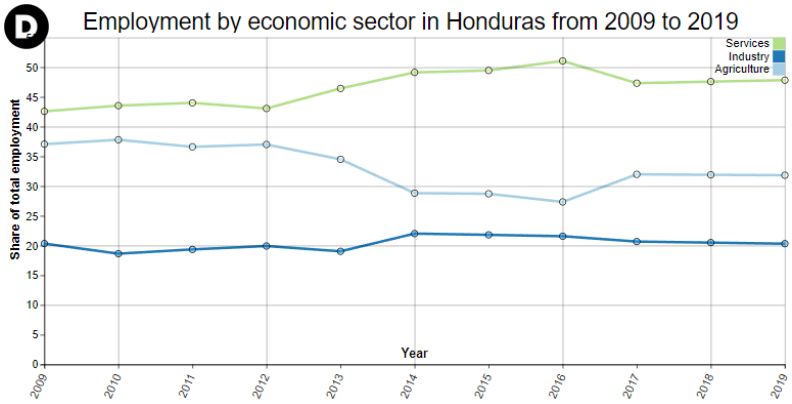
Template-based (Long length): This is a line chart where the x axis denotes Year and a y axis denotes Inflation rate compared to previous year. In total the number of data points it has is 41. Overall Inflation rate compared to previous year has decreased over the Year. The chart in general has a zigzag shape. The most drastic increase took place within the Year 1991 and 1992. Maximum Inflation rate compared to previous year, about 21.06 was reported at Year 1992 The lowest Inflation rate compared to previous year, of value 2.44 was reported at Year 2001 The line rapidly increased by 283.78% from Year 1985 to 1986,increased by 165.24% from Year 1991 to 1992,decreased by 70.21% from Year 1999 to 2000,increased by 75.33% from Year 2005 to 2006, and lastly, increased by 88.92% from Year 2008 to 2009.

Figure 4.5: Sample line chart from Statista and its human-written summary as well as summaries generated by our template-based algorithm (both mid and max lengths).



Human-Written: The average retail price for regular, unleaded gasoline at self-service stations in Canada was 114.5 Canadian cents per liter in December 2019. Prices hit a high of 1.3 dollars per liter in May 2019. The average price in 2018 was around 1.26 dollars per liter, a substantial increase from an average of 1.07 dollars in 2015.	**Template-based (Moderate length):** This is a grouped bar chart showing Average retail price in Canadian cents per liter on the Y-axis throughout 12 Months for 2015, 2016, 2017, 2018, and 2019 on the X-axis. Over the 12 Months, the Average retail price in Canadian cents per liter for both 2016 and 2015 have reduced. For Month May, 2018 had the highest Average retail price in Canadian cents per liter 1359 among the 5 groups and in Feb, 2016 had the lowest Average retail price in Canadian cents per liter 867. Month Jun and Feb had the biggest gap of 219 between the highest and lowest Average retail price in Canadian cents per liter found for 2016.	**Template-based (Long length):** This is a grouped bar chart showing Average retail price in Canadian cents per liter on the Y-axis throughout 12 Months for 2015, 2016, 2017, 2018, and 2019 on the X-axis. All through the Months, similar trend was observed for 2016 and 2015. In both cases, the Average retail price in Canadian cents per liter have reduced. Out of all 5 groups, 2018 had the highest Average retail price in Canadian cents per liter for Month May and 2016 had the lowest Average retail price in Canadian cents per liter for Month Feb. The maximum Average retail price in Canadian cents per liter for 2016 that was found in Month Jun was 1.25 times larger than the minimum Average retail price in Canadian cents per liter observed in Month Feb. Among all 5 groups, 2018 had the gap of 295 between the maximum and minimum Average retail price in Canadian cents per liter observed in Month May and Dec respectively. The second highest Average retail price in Canadian cents per liter 1191 was observed for 2018 in Month May. On average, the Month May had the highest Average retail price in Canadian cents per liter for all 5 groups 2015, 2016, 2017, 2018, and 2019. Whereas Feb had the lowest average Average retail price in Canadian cents per liter.

Figure 4.6: Sample grouped bar chart from Statista and its human-written summary as well as summaries generated by our template-based algorithm (both mid and max lengths).



Human-Written: The statistic shows the distribution of employment in Honduras by economic sector from 2009 to 2019. In 2019, 31.84 percent of the employees in Honduras were active in the agricultural sector, 20.32 percent in industry and 47.84 percent in the service sector.

Template-based (Moderate length): This is a multi line chart with 3 lines representing Agriculture, Industry, and Services. The y axis denotes Share of total employment and the x axis denotes Year. Overall Services is gradually increasing, Agriculture is gradually decreasing, Industry is roughly constant, throughout the Year. Throughout the period, Services generally had the highest Share of total employment relative to others with an average of 46.56, and it reached its maximum at 2016 with a value of 51.07. After Services, Agriculture overall has the second highest values , with a mean value of 33.0, peaking at 2010. Industry mostly had the least Share of total employment with a mean value of 20.38, which peaked at 2014 with a value of 22.02.

Template-based (Long length): This is a multi line chart with 3 lines representing Agriculture, Industry, and Services. The y axis denotes Share of total employment and the x axis denotes Year. Overall Services is gradually increasing, Agriculture is gradually decreasing, Industry is roughly constant, throughout the Year. The lines ordered according to average values of Share of total employment in descending order is: 1. Services, 2. Agriculture, and lastly, 3. Industry. Overall across the Year, Services mostly maintained the highest Share of total employment when compared to others with a mean value of 46.56, and it peaked at 2016. Followed by Agriculture which ranks as the second topmost line , with an average of 33.0 Share of total employment,reaching its highest point at 2010 with a value of 37.8. The bottommost line, Industry, has a mean of 20.38, and peaked at 2014. The minimum Share of total employment about 18.64, occurred at 2010 by Industry.

Figure 4.7: Sample multi line chart from Statista and its human-written summary as well as summaries generated by our template-based algorithm (both mid and max lengths).

Stage 4: Microplanning and realization. The last stage of the architecture generates actual text based on the content and structure determined in the previous stage. To avoid too simple and repetitive structure, we analyzed three different corpora of summary sentences for charts and curated several templates for each message type with varying syntactic structures. To find various variations of templates, we analyzed the dataset shared by Kanthara et al. in their extended work on Chart-to-Text[50]. We randomly pick a different template every time a particular category of message appears to enhance naturalness and diversity in lexicons. For example, the intro message for the single bar chart shown in figure 4.4 (A), our tool will generate following alternative intro messages and pick one of them randomly:

1. "*This is a bar chart representing 42 Months in the x axis and Units sold in the y axis.*"

2. "*This bar chart has 42 columns on the x axis representing Month, and Units sold in each Month on the y axis.*"

3. "*This is a bar chart. It shows Units sold for 42 number of Months.*"

Similarly, for the *extrema(min/max)* category, one of the following sentences would be picked:

1. "*The maximum car sold 829 is found in Month September 2018 and the minimum is found in August 2017 where Units sold is 44.*"

2. "*The Units sold was highest in Month September 2018 and lowest in Month August*

2017."

3. "*The Units sold was appeared to be the highest in Month September 2018 and lowest in Month August 2017.*"

Sample Summary Outputs: Figure 4.4, 4.5, 4.6, and 4.7 show example charts and corresponding summaries: single bar (A), single line (B), grouped bar (C), and multi line (D) taken from Statista. The first column contains the human-written summaries of the charts that we found in Statista. Descriptions generated by our template-based algorithm (with moderate and long length) for corresponding charts are presented in last two columns. Note that *SeeChart* allows user to change the description length of a chart to 3 levels: *short*, *moderate*, and *long* length. *Short* summaries are generally 3 to 4 sentences long depending on whether it is a bar chart or a line chart.

4.4 The *SeeChart* Interface

Informed by the insights gained from needfinding interviews with our participants, we designed and developed the *SeeChart* interface which takes a chart appeared on a webpage and regenerates an interactive version of it with audio description of the chart as well as features for selecting and navigating through charts. During our design process, we involve three participants from the needfinding interviews who interacted with our initial prototype and provided feedback. We have considered the feedback to improve the accessibility features. All these interactive features are accessible through keyboard

shortcuts. An overview of the interface is shown in figure 1.1.

4.4.1 Get an Audio Overview of the Chart

When *SeeChart* detects a chart in a webpage, it reconstructs the chart and generates the chart description in real time. Upon successful reconstruction of the chart summary, the tool plays a *chime* sound to notify the user about the presence of the chart. The sound does not intervene in any ongoing narration being played by the screen-reader tools. At this point, if the user wants to explore that chart, pressing the `SPACE` key opens the reconstructed chart in a pop-up window to let user access the chart through a set of interactive features.

Users can get a quick overview of what the chart is about by pressing `Enter` which provides the chart title along with chart type, e.g., *"This is a Bar chart. It shows Subaru car sales in the United Kingdom (UK) 2016 to 2019"*. The user can access the chart summary by pressing the `Space` key. However, during exploring the *SeeChart* prototype, participants mentioned that going through the long summary at once results in cognitive burden as they need to keep track of the information and overload their working memory. To address their concern, we enable them to hear the summary sentence-by-sentence by pressing `J`, `K`, and `L` keys, where `J` and `L` play the next or previous sentence, and `K` repeats the sentence. Since the right hand of a user typically rests on the `J`, `K`, and `L` keys, we used these keys to make sure users find it easy to learn and remember (*DI-5*). By default,

the description is provided at a moderate length, however, users can adjust the summary length by pressing `1` (short), `2` (moderate), or `3` (long) (*DI-4*).

If there are multiple charts on a web page, the *SeeChart* system reconstructs all of them and user can navigate between them by pressing `N` (next) and `P` (previous) keys. All audio descriptions are generated with proper punctuation so that users can comprehend them effectively (*DI-5*).

4.4.2 Access to Chart Data

Users with blindness generally do not get access to chart data as most of the websites do not provide chart data in a separate table. Therefore, there is a pressing need for browsing chart data in a similar way they browse table data on Web. Participants suggested that when they encounter data tables on a webpage, they read the data of each cell by using the keyboard `arrow` keys. Therefore, *SeeChart* uses such shortcuts so that users explore a chart by pressing *arrow keys* to navigate through the data points from beginning to end or vice versa (*DI-1*). For each data point, the interface plays the information such as the data value and the corresponding data attribute(s). For example, for the bar chart shown in figure 4.8 (A), if the user selects the first data point by pressing the `right arrow` key, the tool will say "*You're at the beginning of the graph. In airlines American, the number of passengers in millions was, 203.*". For bar and single line charts, user can only navigate to left or right by pressing the corresponding `arrow` keys. Tool will say "*This is*

the last point of the graph. In airlines American, the number of passengers in millions was, 110.", when user reaches the last bar of the chart as shown in 4.8 (B). Thus, the interface provides navigational cues by informing where the user is navigating right now. When the user reaches the end of the chart and presses the right arrow again, she would be redirected to the first point and again informed of the current position.

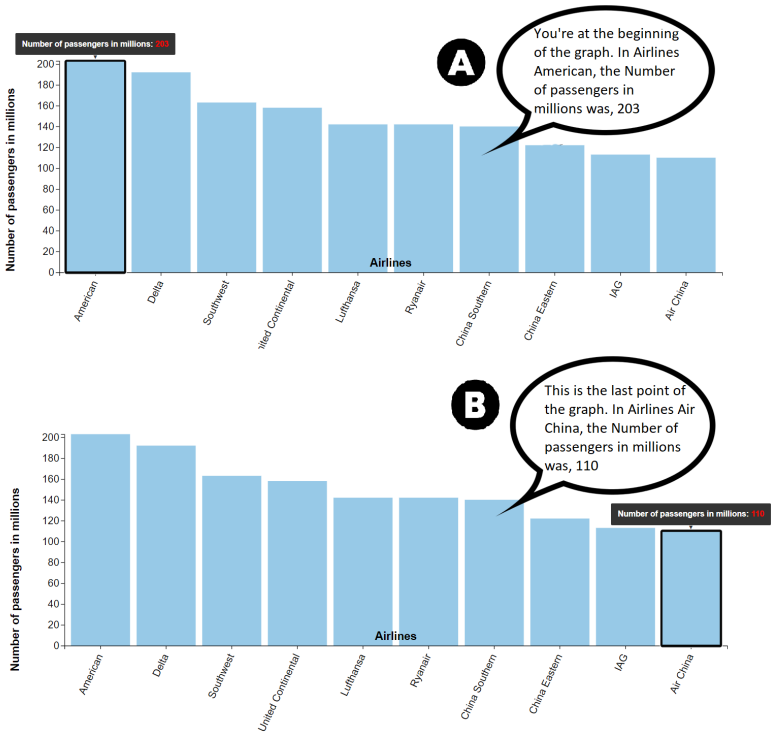


Figure 4.8: Accessing chart data with keyboard navigation.

When line charts have multiple series the user can switch among different line series by pressing `up` or `down arrow` keys. In this case, when users reach the topmost or bottommost line, *SeeChart*informs the user by saying "*This is the topmost line"* or "*This is the bottommost line*". This allows users to explore a chart without getting disoriented.

Similarly for grouped bar charts, user can move between different groups by pressing `up` or `down arrow` keys.

4.4.3 Interactive Summarization

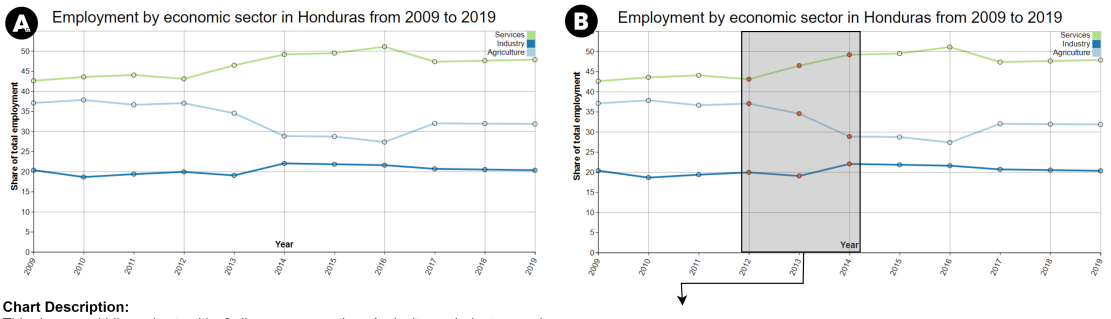


Chart Description:
This is a multi-line chart with 3 lines representing Agriculture, Industry, and Services. The y axis denotes Share of total employment and the x-axis denotes Year. Overall Services is gradually increasing, Agriculture is gradually decreasing, Industry is roughly constant, throughout the Year Over the period, Services generally had the highest Share of total employment relative to others with an average of 46.56, and it reached its maximum at 2016 with a value of 51.07. After Services, Agriculture overall has the second highest values, with a mean value of 33.0, peaking at 2010. Industry mostly had the least Share of total employment with a mean value of 20.38, which peaked in 2014 with a value of 22.02.

Description based on selected data:
Overall Services is gradually increasing, Agriculture is gradually decreasing, Industry is roughly constant, throughout the Year. The ranking of the lines from topmost to bottommost is as follows: 1, Services, 2, Agriculture, and lastly, 3, Industry. During this period, Services generally had the highest Share of total employment relative to others with an average of 46.22, and it reached its maximum in 2014 with a value of 49.14. After Services, Agriculture overall has the second highest values, with a mean value of 33.4, peaking at 2012. The bottommost line, Industry, has a mean of 20.33 and peaked in 2014.

Figure 4.9: Initial summary of a line chart (A), and partial summary based on selected data points (B).

While accessing individual points is helpful to users, they may have difficulty comprehending a large number of data points together. Sometimes they desire to select a region of interest in the chart and get the corresponding summary for that selection. *SeeChart* allows users to select the data points they are interested in and provides a comprehensive description based on the selected points (*DI-3*).

The user can press the `Shift` key to activate the multi-point selection mode, then continue to select using the `Right-arrow`. We adopt this interaction because this is

how we commonly perform selection in various tools, for example, selecting text using a word processing tool (*DI-5*). The interface provides feedback as the user continues to select (e.g., "*Year 2012 is selected*"). Once user selects desired data points and releases the `Shift` key, *SeeChart* generates the summary of the selected range of data immediately using the template-based algorithm as described in Section 4.1. The user can reset the selected points by pressing the "`Escape`" key. The user can also listen which points have been selected by pressing the "`S`" key. Figure 4.9 shows an example of partial summary generated based on selected data points: `year 2012`, `2013`, and `2014`. Finally, users can also get a summary from discontinuous sets of data points from the chart as shown in figure 4.10 where the generated summary is based on the data points of year 2005 to 2007 and 2009 to 2011. Following is the summary for the selection points:

> *Overall Number of units sold has increased over the Year. Number of units sold is roughly constant from 2005 to 2007, and lastly slightly increasing from 2007 to 2011. The steepest increase occurs in between the Year 2010 and 2011. The highest Number of units sold, of value 470004 was reached at Year 2011. The lowest Number of units sold, of value 313437 was reported at Year 2007.*

And here is the overall description of the chart generated by *SeeChart*

> *The chart is a line chart where the x-axis denotes Year and a y-axis denotes Number of units sold. In total the number of data points it has is 16. In general Number of units sold has rose over the Year. The most drastic increase took place within the Year 2014 and 2015. Maximum Number of units sold, about 560415 was reported at Year 2017 Minimum Number of units sold about 313437 was reached at Year 2007.*

SeeChart also supports direct manipulations to choose multiple items through lasso

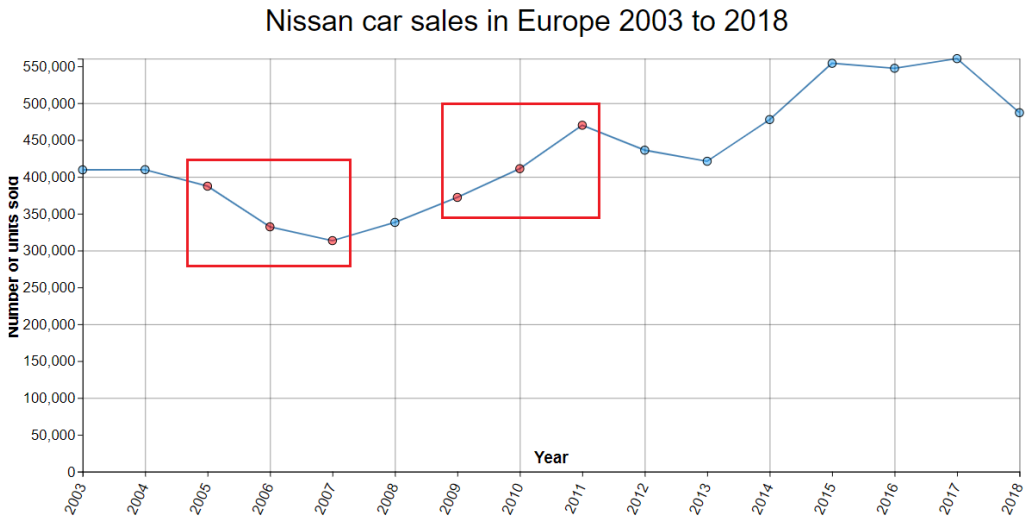


Figure 4.10: Summary generated from discontinuous sets of data points

selection and brushing. However, according to our needfinding study, our participants preferred to use keyboard shortcuts rather than using mouse. Future work may explore how they may perform these interactions with touch and tactile-based interfaces.

4.4.4 Search by Voice Command

Based on the feedback of a some participants during our main user study, we have integrated a voice search feature in the *SeeChart* interface shown in figure 4.11.

We have used Web Speech API[12] to construct the speech recognition system and user can activate the search mode by pressing the "`F`" key and then asking the question. Upon successful recognition of a word or phrase the tool initiates the search operation and looks for predefined keywords (e.g., x-axis data label, maximum, minimum, trend,

etc.) in the search string, *SeeChart* presents the result if found or asks user to ask the question again with some hints of what they can ask. User can also type the question after activating the search mode and press "`Enter`" to get the answer. When the search mode is active, no other assigned shortcut keys will work unless user presses "`ESC`" key or complete the operation by pressing "`Enter`".

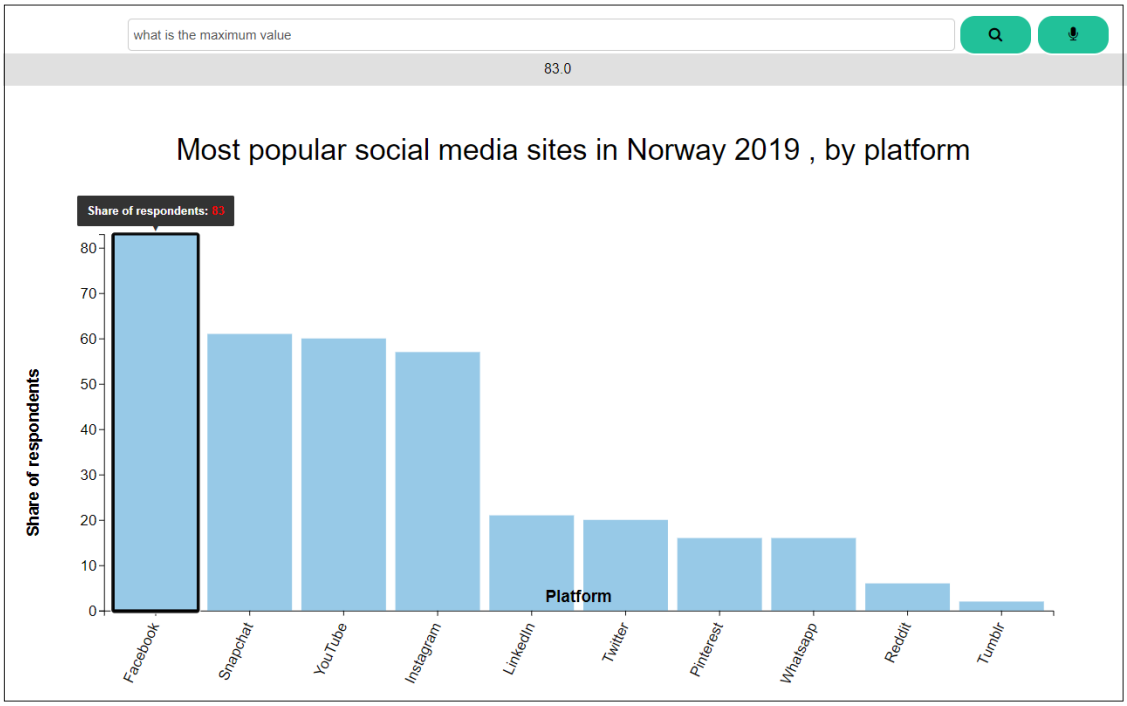


Figure 4.11: Search bar of *SeeChart* tool.

Using this feature, users can find the corresponding y-axis value of an x-axis data label. For the chart shown in figure 4.9, user can either ask "*What is the value of 2011?*" or simple say "*2011*" and if the provided data is accurate, the tool will reply "*We have found multiple values for Year 2011. These are, Agriculture is 36.62, Industry is 19.36,*

Services is 44.02.". Users can also ask different stats given in table 4.2. For the example chart shown in figure 1.1, user can ask "*What is the difference between maximum and minimum values?*" and the reply would be "*785*". Users can learn what type of questions can be asked by checking the help menu.

4.4.5 Control Audio Speed

Screen-reader users rely on auditory feedback to communicate with computer devices. Screen-reader tools let users select faster speech rates to speed-up consuming the digital information. Previous study[40] showed that, at around 278 words per minute (WPM), visually impaired users found the sweet spot between scanning information efficiently and understanding each sentence while average person speaks at 100-130 WPM[88]. In *SeeChart*, we used the *Text-to-Speech* (TTS) service provided by Amazon Polly[16] to convey information from charts (e.g., the value of a data point, the description of a chart, and other commands to guide users). Since the tool primarily describes quantitative information, it allows the user to toggle between different voice rates by pressing the "`T`" button to understand the information at their own pace.

4.4.6 Keyboard Shortcuts

SeeChart comes with a list of shortcut keys for different functionalities as shown in table 4.3.

Table 4.3: List of available keyboard shortcuts in *SeeChart*.

Key/s	Functions
Enter	Plays the chart title along with chart type.
X, Y	Plays the X and Y axis labels.
Space	Plays the chart description.
J, L	Plays the next or previous sentence.
K	Plays the currently selected sentence again.
F	Activate search mode by typing.
Q	Activate voice search mode.
Right or Left Arrow	To navigate through the data points.
Up or Down Arrow	To switch between lines in multi line chart.
Shift + Right Arrow	To select multiple points for partial description.
S	To hear the list of selected points.
Escape	To reset the selected points.
1, 2, 3	To change the length of the description.
T	Toggle audio speed

We tried to keep the number of keys and their combinations minimum. In our needfinding interviews, one participant (P05) suggested: "*If you are planning to assign keyboard shortcuts, try to avoid too many key combinations as I already need to remember many shortcuts for JAWS*". *SeeChart* has only one shortcut that involves multi-keys press: `Shift + Right Arrow` since this is the standard way of selecting texts (Microsoft Word) or cells (Microsoft Excel). Also, we tried to keep finger-travel distance minimum by utilizing the keys where human hands rest for frequently used functions (e.g., `J`, `K`, and `L`).

5 User Study

We designed a user study to see whether blind individuals could effectively operate the *SeeChart* tool with minimal assistance and whether they could successfully complete general tasks such as determining the maximum, minimum, or trend of a line from the representation. In the parts that follow, we discuss the research questions we attempted to answer and an analysis of the study's findings.

5.1 Goal of the Study

In this user study we tried to answer the following questions. First, how the *template-based* natural language summaries generated by our system fare in terms of user's performance compared to the *human-written* summary condition (in which a human-written caption is provided)? Second, how the *interactive summarization* feature affects user's performance to accomplish data exploration compared to when user manually examines each data point individually. Third, how do the users consider *SeeChart* tool in terms of subjective measures (e.g., usefulness and ease of use)? Finally, what can we learn from participants'

feedback to improve the future development of accessible visualization tools?

In addition, we logged participants' keystrokes and the time they spent interacting with different features of the tool. We analyzed the log data to find the interaction patterns and most used features.

5.2 Methodology

In an effort to identify the answers to the research questions, we created an experimental website. In the parts that follow, we go into further detail on our participants, equipment, study design, tasks, and methods.

5.2.1 Participants

We conducted the study with 14 visually marginalized participants (7 female, 7 male, age range 35-65, North-American) who had experience in using screen-reader tools (e.g., JAWS and NVDA). Six of them reported that they encounter charts occasionally (several times a month) and 8 of them said they encounter charts frequently (several times a week). Same to the needfinding study, we assigned a random ID (from P01 to P14) to each of the participants. Six of the participants also participated in the needfinding interviews.

5.2.2 Apparatus

Using HTML-CSS, JavaScript, and Python-Flask[4], we created an experimental website. The Heroku[6] platform served as the website's host. We conducted the study in separate ZOOM meetings[13] because of current COVID-19-related constraints. On their personal PC or laptop, participants ran the *SeeChart* system on a chrome browser. There were no specifications for operating system or screen resolution. In the experiment, there were a total of 12 charts presented to the participants, with 8 different charts for each type of chart (e.g., single bar, multi line). Each chart was taken from Statista[82].

5.2.3 Study Design

We designed the user study with a $2 \times 2 \times 2$ within-subject design with following independent variables respectively: **Chart Type:** with 2 conditions (*single bar*, *multi line*), **Summary Type:** with 2 conditions (*template-based*, *human-written*) and **Task type:** with 2 conditions (*Task A*, *Task B*). Here, *Single Bar* refers to a bar chart with one categorical attribute and *Multi Line* refers to line charts with multiple series. The dependent variables for the experiment were task completion time (measured in seconds) and percent of correct answers (%). Each participant repeated each condition 2 times, thus, there were a total of 16 unique trials per participant. To minimize the ordering effects, we counterbalanced the ordering of the conditions by dividing the participants

into two groups. Participants from different groups started the study with different chart types. Also, we changed the order of *summary type* and *task type* after each participant's session. For example, if a participant was presented with *template-based* summary and *task A*, the next participant was presented with *human-written* summary and *task B*.

In total, in the study we collected a total of 224 trials (14 participants × 2 chart types × 2 summary types × 2 task types × 2 trials).

5.2.4 Tasks

We chose following two types of tasks based on the primitive analytic activity by Amar et al. [15]. The first type of task involved finding extrema (*Task A*). Figure 5.1 shows the corresponding charts[8],[10] for the following questions:

(A) "*Which sector maintained the highest share in GDP over the years?*"

(B) "*In which country, the cost of an angioplasty is the highest?*"

The second type of task is relatively complex as it combines two primitive tasks i.e., filtering followed by finding extrema (*Task B*). Figure 5.2 shows the corresponding chart[9],[11] for the following questions:

(A) "*Among the first five provinces shown in this chart, which province had the least number of snowmobile registrations?*"

(B) " *What was the overall trend of share prices from year 2015 to 2017?*"

8 out of the 16 trials presented *Task A* while the other 8 trials presented *Task B*. For

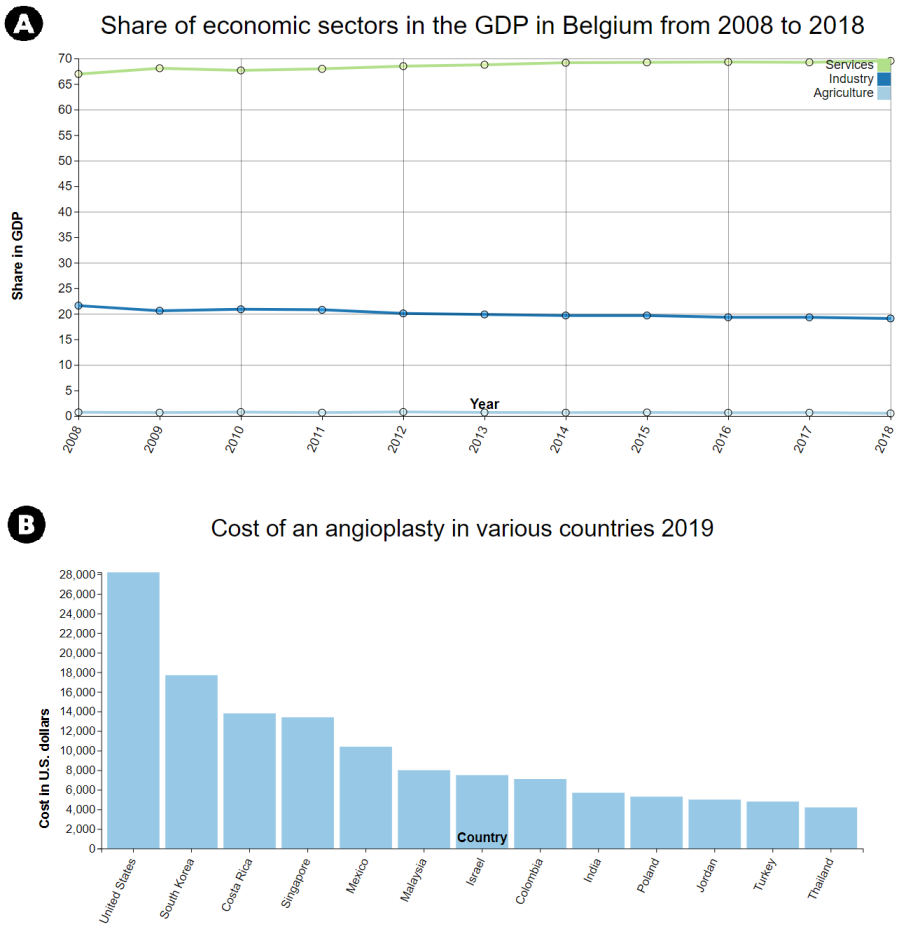


Figure 5.1: Example of multi line chart and single bar chart for Task B.

designing the tasks, we chose 16 different charts from Statista[82] in SVG format along with corresponding human-written summaries that are provided with the charts in Statista. Statista is an online platform that regularly publishes charts on a wide range of topics (e.g. transportation, economy, sports, opinion research, etc.).

We chose charts with similar number of data items for each chart type to reduce the possible effects of confounding factors (data items for bar charts: M = 12.3, SD = 3.4,

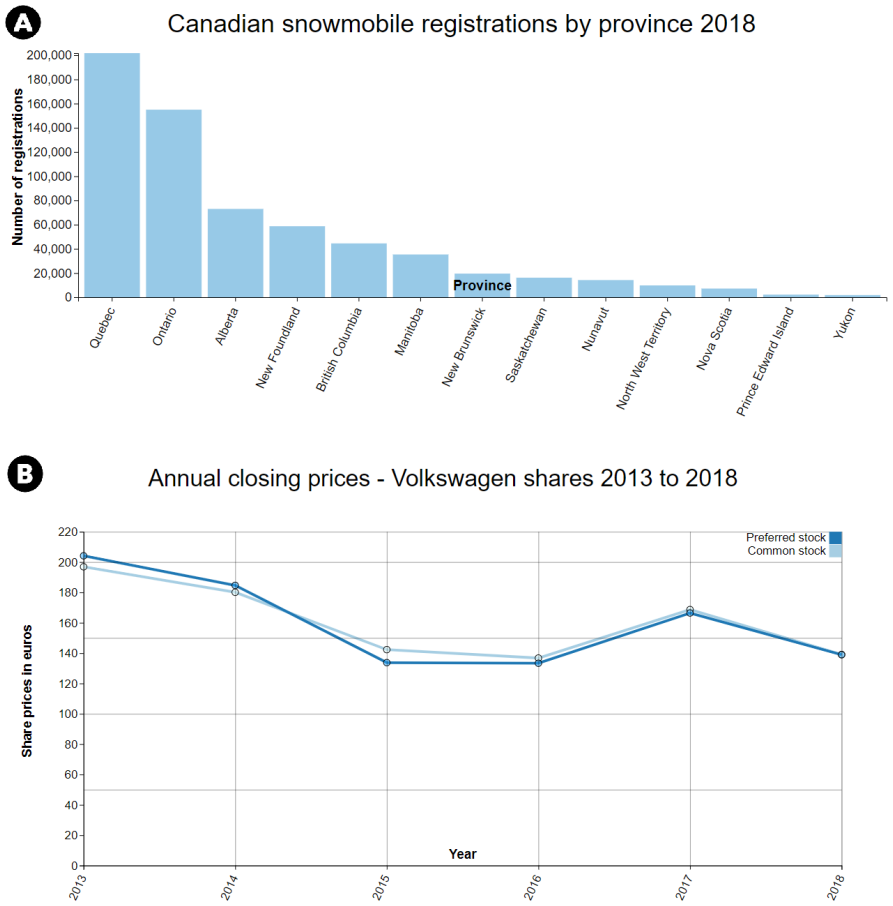


Figure 5.2: Example of single bar chart and multi line chart for Task B.

and for multi line charts: $M = 10.5$, $SD = 1.11$). All of the tasks along with corresponding bar charts (table **??**) and multi line charts (table **??**) are presented in Appendix B.

5.2.5 Procedure

After filling up a pre-study questionnaire, participants went through a tutorial where they learned how to use the tool under different conditions and to complete the task. During the

tutorial, we demonstrated the *SeeChart* tool with sample charts that were not used during the main study. In order to get comfortable with the tool, we allowed the participant to interact with it for 15-20 minutes.

During the main study, each of the 16 trials presented a different chart along with a question. In the *template-based* condition, the interface first presented the automatically generated moderate-length summary from our NLG method and then re-generated the summary whenever the user selected a range of data items. In the *human-written* condition, participants had access to the human-written summary of the chart but interactive summarization of selected data points was not available. In both conditions, participants could navigate through individual data points using `arrow` keys. Participants explored the chart and then mentioned the answer out loud.

After completing all 16 trials, participants rated the *SeeChart* tool on a 5 point Likert scale and expressed their opinions about the tool in a short interview. The experiment took about 90 minutes to complete and each participant received $40.00 CAD for participating in the study. We developed an online experimental system to run the study which was accessed through the chrome browser. All study sessions were conducted remotely through video conference to conform to social distancing protocol due to COVID-19.

In Appendix B, we present the questionnaires used in the user study.

5.3 Results

Here we present the result we found from the analysis of user study data.

5.3.1 Task Performance

Task Completion Time: When we compare among summary types, we observe that participants were faster with *template-based* summaries (M = 67.06, SD = 10.93) than *human-written* (M = 94.39, SD = 13.83). The results are shown in figure 5.3 (A). The analysis of variance (ANOVA) test revealed that the effect of *summary type* on task completion time was statistically significant ($F_{1,13}$ = 25.735, p < .0005) while the effect of *task type* on task completion time was not statistically significant ($F_{1,13}$ = 2.291, p > .05). One possible reason why participants were more efficient with template-based summaries is that they are more effective in answering questions by providing more consistent discourse structure and contents and focusing on statistical information rather than contextual and background information compared to human-written summaries. Furthermore, participants completed the tasks faster (18.34%) with single bar charts compared with multi line charts (Figure 5.3 (B)). This suggests that users with blindness take more time with multi line charts possibly because they are usually more complex than single bar charts.

While accomplishing *Task B*, users could apply the interactive summarization to get a partial summary of the chart to answer the questions in the *template-based* condition which was not available in the *human-written* condition and participants had to examine individual values of filtered data points to find the answers. Figure 5.3(C) shows the difference as participants were 37.98% faster to complete *Task B* when they could utilize the interactive summarization feature and the effect was statistically significant ($F_{1,13}$ = 15.320, p < .005).

Accuracy: Participants were also more accurate with *template-based* summaries (94.79%) than *human-written* summaries (88.54%) as shown in figure 5.4 (A). This indicates that human-written summaries are often generic and do not help to perform the common analytic tasks identified by Amar et al. [15]. As a result, participants had to check the data points individually to find answers. Interestingly, participants were less accurate with *Task A* compared to *Task B* (Figure 5.4 (B)) despite the fact that *Task B* involves multiple operations (filtering followed by finding extrema) whereas *Task A* involves finding extrema only. One possible explanation is that with *Task A*, participants had to go through all the data points and keep track of their values to find extrema which can lead to much more cognitive burden, than finding the extrema from a much smaller number of data points. Finally, more errors occurred with multi line charts than with single-series bar charts (Figure 5.4 (C)), possibly because of the increasing complexity with more data series than single-series bar charts. Eleven participants made at least one

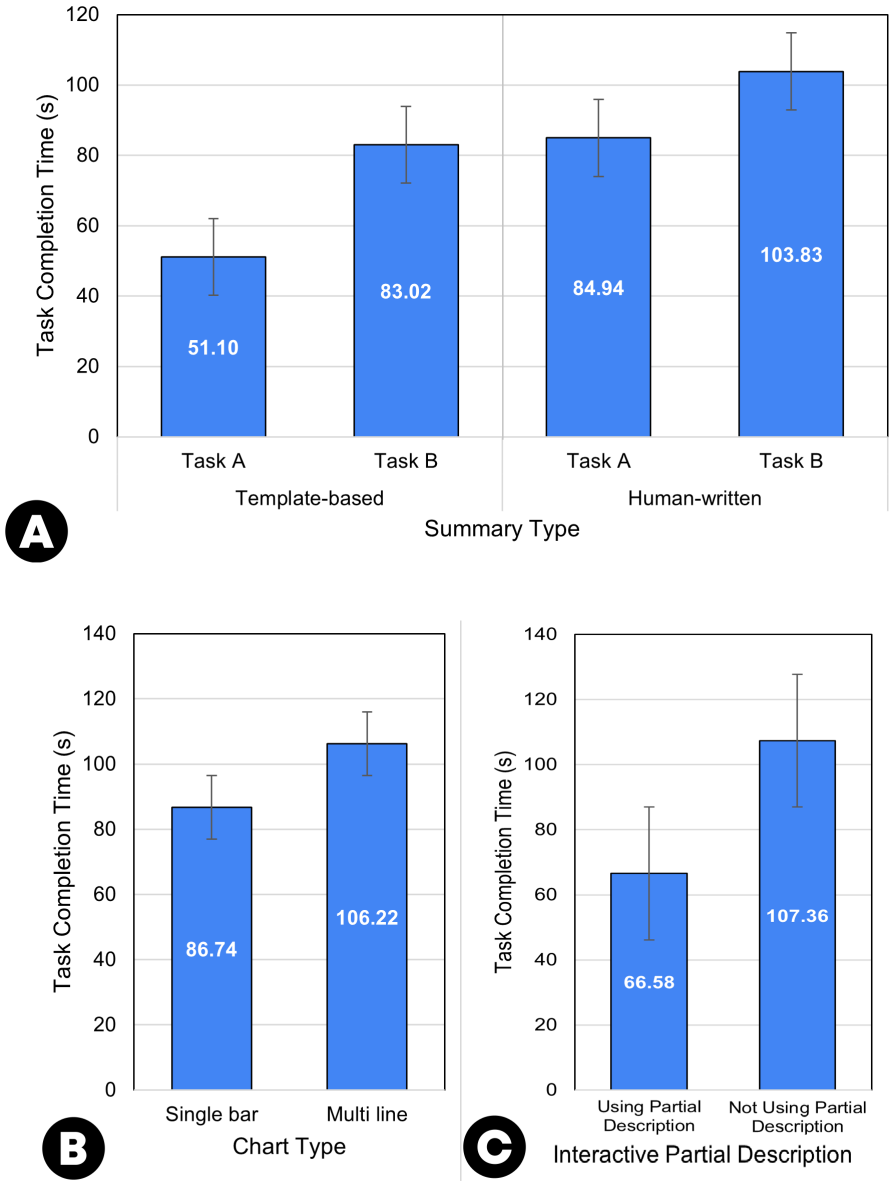


Figure 5.3: Mean task completion time for summary and task types (A), chart types (B), and application of interactive summarization.

error when trials involved the *human-written* condition, compared with 5 participants who made errors with the *template-based* condition.

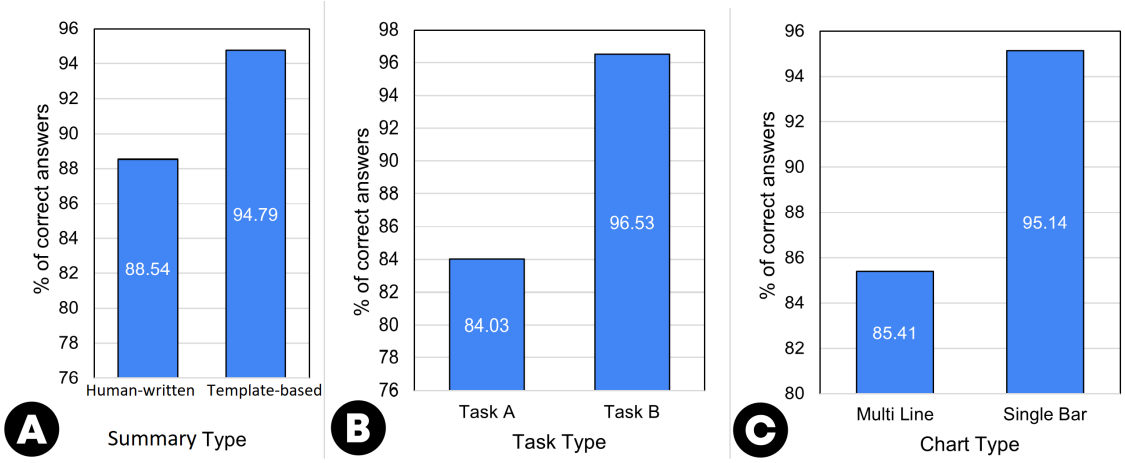


Figure 5.4: Percentage of correct answers based on summary type (A), task type (B), and chart type (C).

5.3.2 Post-study Feedback

At the end of the study, we asked participants for their feedback. Figure 5.5 shows the overall result of the likert-scale questionnaire responses. As we can see all participants agreed or strongly agreed that the *SeeChart* tool is useful for exploring visualizations and it can help them to understand charts. Also, they agreed or strongly agreed that they would recommend this tool to other people with visual impairments.

During interviews, all participants confirmed that previously they did not have any tool to help them interactively explore online charts through audio descriptions. Furthermore, there was no good way to get quick insights from charts based on selected points. P09 said:

> "*This tool is so exciting. I never thought this would be possible to move across a chart and select data points to get such a detailed summary! Integration of*

the voice search just made life easier. Just like the other features, it was very responsive.".

Most participants stated that they liked the interactive feature that generates summaries from the selected range of data points, as P03 said:

"*This feature not only helps blind people to get insights from a selected region of a chart, it will be helpful to anyone interested in data analytics and visualization.* "

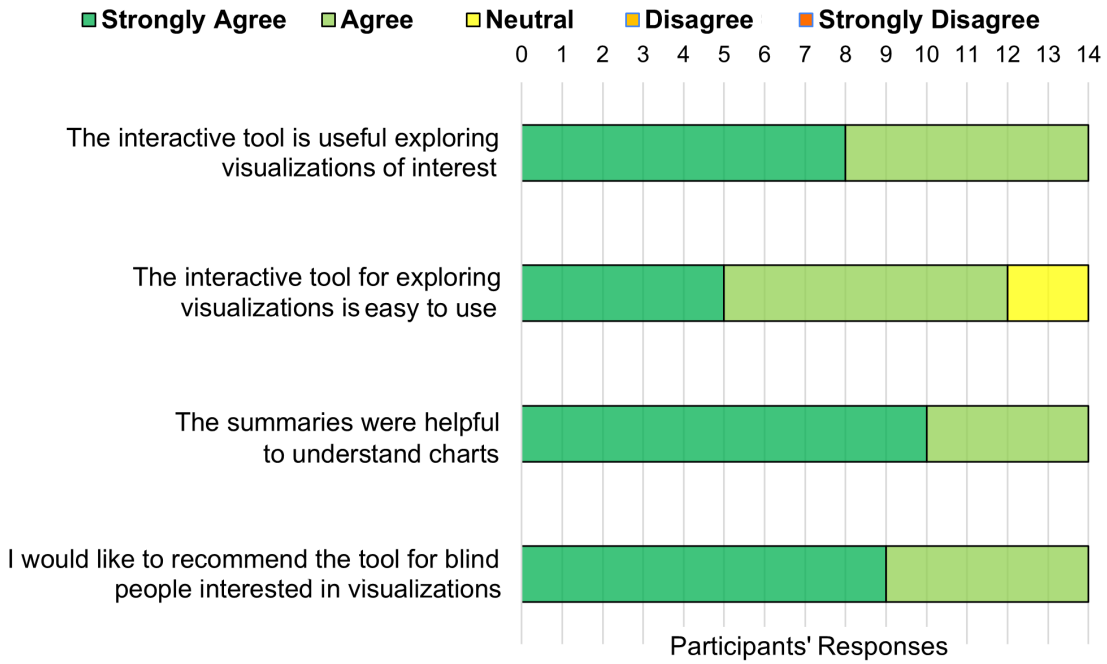


Figure 5.5: User study responses to post-study interview.

Participants also liked the template-based summaries as they could get rich insights from charts that were not possible before. Eight participants liked the feature of accessing chart data with keyboard shortcuts. In general, they liked the audio feedback while

navigating through the chart. Three participants mentioned that they found the shortcut keys to be very conveniently designed and easy to remember.

Five of the participants mentioned that it would be extremely useful to have a search feature for getting statistically significant information (e.g., extrema, average, etc.) from the chart data. P06 said:

> "*I wish there was a quick search feature where I could ask the tool to find answers from the chart data rather than looking for it by myself*".

The voice search feature (described in 4.4.4) was later implemented and integrated it with the *SeeChart* tool.

The interviews also revealed that while *SeeChart* mostly met their needs there were some concerns from participants. In particular, navigating through relatively complex charts such as the line chart with multiple series seemed difficult to at least four participants who reported that they were losing track of where they are as they were switching between different lines. Two of the participants suggested to introduce a search feature so that they can quickly select specific data attribute or values without needing to navigate through all attributes and values. Further, one participant suggested to support natural language question answering so that they can get their question answered through speech to perform simple analytical tasks such as sorting, and filtering data.

5.3.3 Analysis of user interactions

Besides questionnaires, we analyzed the log data to get insights into the interaction patterns of participants. As shown in figure 5.6, when a chart is presented to the participants, majority (67%) of them first listened to the chart description before going through the chart data and 33% of the participants chose to hear the axis labels and chart data first before listening to the chart description.

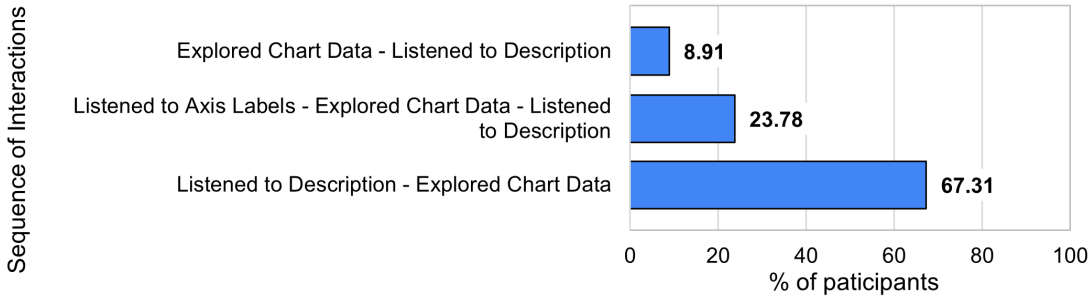


Figure 5.6: Sequence of interactions when a chart is presented to the participants.

Figure 5.7 shows the usage of various features and components of *SeeChart* tool. Anticipatedly, all of the participants listened to the moderate length chart description and navigated through the chart data. 76% participants utilized the interactive summarization feature to perform tasks. We found that very few participants (13%) were interested to hear the long length descriptions and while 34% of them checked short length descriptions. A possible explanation is that the participants found the answer to the questions either hearing the shorter length descriptions or checking the data values. Therefore, they did not require to listen to the long length descriptions. Axis labels were checked by 23.78%

participants who did not chose to listen the descriptions first when a chart was presented. Participants who listened to the description first, were already knew what were the axis labels and they did not require to check the axis labels again. We also noticed that the default audio speed was too slow for 83.74% of participants as they had to increase the tempo.

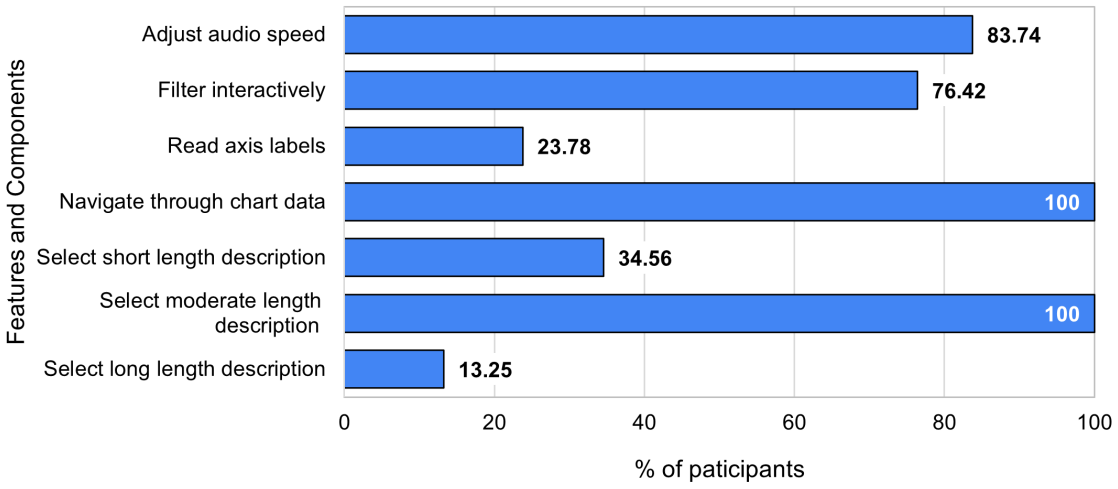


Figure 5.7: Usage of various features and components of *SeeChart* tool.

From the keystroke log data, we found that less than 5% key-presses were wrong. This suggests that the keyboard-shortcut selection was easy for the participants to perceive.

5.4 Summary

The user study reveals that participants were indeed able to use the tool with very little training. Moreover, the availability of chart data navigation features helped them to get

the answers faster. The descriptions generated by the *SeeChart* tool were particularly beneficial to get an overview of the charts. Participants were able to utilize the interactive partial description feature to find out answers faster from filtered data. The keyboard shortcuts were easy to grasp and log data showed that participants did not make too many errors using the shortcuts. The post-study feedback suggests that participants really enjoyed using the tool and they would like to recommend others to try it.

6 Discussion and Future Work

In this chapter, we discuss the limitations that *SeeChart* system currently has and our future plan to address those. We summarized the contributions of the thesis as well as put our final remarks at the end of the chapter.

6.1 Summary of Contributions

The primary objective of this thesis is to assist blind or visually impaired individuals in exploring web visualizations with the aid of a browser extension tool. Currently, charts are frequently inaccessible to those with visual problems, aggravating information disparities [48]. We want to address the issue of inequality by taking a first step in making data visualization more accessible to blind people through our study. Additionally, even though the *SeeChart* method is intended for blind people, by presenting the most important findings from charts in everyday language, it has the potential to make data visualization more inclusive and accessible across a variety of user communities in the future.

6.2 Limitations and Future Work

The evaluation of *SeeChart* suggests that providing natural language summaries using template-based NLG methods are helpful in understanding charts on Web more efficiently and effectively. The study also revealed important lessons and helped us identify the limitations that we would like to lift in future work.

Support complex charts through multiple modalities: Currently, *SeeChart* supports a variety of bar charts (e.g., single-series bar charts, stacked and grouped bar charts), line charts (with single or multiple series), and pie/donut charts. We chose these chart types in our initial prototype because they are frequently found on Web [45] and can be deconstructed relatively easily. Moreover, existing chart summary corpora mainly cover these chart types [56, 74]. In future, we aim to cover more complex charts such as mapchart, scatter-plot matrix and multiple coordinated views. However, with the increasing complexity of charts with larger number of data attributes and items, designing interactions can be more challenging as the participants struggle to keep track of multiple lines solely based on keyboard shortcuts. Future research may introduce additional input modalities such as touch and speech to perform brushing, filtering, and zoom in/out to support complex visualizations such as map charts, multiple coordinated views and improve scalability. Adopting and extending recent works on natural language interfaces for exploring and question-answering with data through speech can be starting

point [46, 80, 53]. In terms of output modalities, an interesting direction could be to combine data sonification along with audio summary based on interactions.

Improve and extend chart deconstruction: *SeeChart* currently supports SVG charts created using D3[21] and highcharts[43] and can deconstruct basic charts like bar, line, pie charts, and scatter plots. In immediate future, we aim to support the extractions of SVG charts generated by other libraries such as Plotly[7], Chartblocks[3], and Graphiq [5]. Unlike chart images for which deconstructing data involves computer visualization challenges, recovering data and visual encodings can be relatively easier from SVG-based charts by extracting features such as position, sizes, and colors of SVG elements [17]. Similarly, Vega-Lite[71] charts provide declarative specifications which can be extracted [53] for re-constructing the accessible version of a chart. While deconstructing image-based charts is still a challenging problem, more advanced deep learning methods have shown promise in extracting data and generating chart summaries from them [74, 62]. In future, such improved methods can be integrated with our *SeeChart* tool to make these image-based charts more accessible.

Explore advanced chart summarization methods: During our design, in addition to the template-based approach, we considered recently proposed deep learning methods based on the transformer architecture [74, 62]. while such advanced data-to-text NLG models usually generate fluent summaries and offer more variations in terms of reported insights, grammatical styles, and lexical choices than template-based summaries, they also

suffer from hallucinations (i.e., outputting irrelevant tokens) and factual errors. However, during our needfinding study participants preferred factually correct insights over rich variations in language, therefore we opted for utilizing template-based summaries. Our summarization method can also be improved in terms of improving the insight generation and more effective ways to rank the insights [86]. However, generating more complex insights covering perceptual and cognitive phenomena is still an open problem [56]. Finally, our summaries do not cover contextual or domain-specific information that is not present within the given chart. However, this type of information has been found to be less important to the users than the insights covering statistics and data trends [56]. Nevertheless, future work may exploit large-scale knowledge sources to inject domain knowledge into the summaries.

Study *SeeChart* in the wild: We ran our initial evaluation through user studies in a controlled setting with specific tasks. While such evaluation allowed us to learn about the efficacy and limitations of our tool, they also lack realism [24]. In the future, we will release the tool as a downloadable Chrome extension, so that a large pool of people with blindness can use it over a long period in their natural settings. This will help us to measure the adoption rate of the tool and get more insights into the interaction behavior of target users in the wild.

6.3 Final Remarks

With the help of keyboard-based interactions and natural language description, the interactive web application *SeeChart* , which we have just introduced, provides accessible visualization. The application automatically disassembles web charts, then creates an accessible version of the relevant chart that includes interactive capabilities for studying the charts and summaries that are also generated automatically. A user study with blind users found that interactive features for chart exploration and such automatic summaries improved the efficiency and effectiveness of their chart reading. We anticipate that *SeeChart* will help make information visualization and analytics more approachable using natural language as a modality and encourage additional study in this largely untapped field. The source code of *SeeChart* is available at `https://github.com/ZubairShovan/SeeChart`.

Bibliography

[1] NV access, 2022. https://www.nvaccess.org/.

[2] SAS, 2022. https://support.sas.com/accessibility/.

[3] Chartblocks, 2022. https://www.chartblocks.com/.

[4] Python-Flask, 2022. https://flask.palletsprojects.com/en/2.2.x/.

[5] Graphiq, 2022. https://www.graphiq.com/.

[6] Heroku, 2022. https://www.heroku.com/.

[7] Plotly, 2022. https://plot.ly/.

[8] Sample Chart from Statista, 2022. Retrieved March 20, 2022 from https://www.statista.com/statistics/328691/share-of-economic-sectors-in-the-gdp-in-belgium/.

[9] Sample Chart from Statista, 2022. Retrieved March 20, 2022 from https://www.statista.com/statistics/420703/canadian-snowmobile-registrations-by-province/.

[10] Sample Chart from Statista, 2022. Retrieved March 20, 2022 from https://www.statista.com/statistics/189973/cost-of-an-angioplasty-in-various-countries/.

[11] Sample Chart from Statista, 2022. Retrieved March 20, 2022 from https://www.statista.com/statistics/466109/annual-closing-share-prices-of-volkswagen/.

[12] Web Speech API, 2022. https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API.

[13] Zoom, 2022. https://zoom.us/.

[14] Peter Ackland, Serge Resnikoff, and Rupert Bourne. World blindness and visual impairment: despite many successes, the problem is growing. *Community eye health*, 30(100):71, 2017.

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

[16] Amazon. Turn text into lifelike speech using deep learning, 2022. `https://aws.amazon.com/polly/`.

[17] Leilani Battle, Peitong Duan, Zachery Miranda, Dana Mukusheva, Remco Chang, and Michael Stonebraker. Beagle: Automated extraction and interpretation of visualizations from the web. In *Proc. of the 2018 CHI*, pages 1–8, 2018.

[18] Fight Blindness. Together let's move research forward, 2022. `https://www.fightingblindness.ca/`.

[19] Michelle A Borkin, Azalea A Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. What makes a visualization memorable? *IEEE TVCG*, 19(12):2306–2315, 2013.

[20] Michelle A Borkin, Zoya Bylinskii, Nam Wook Kim, Constance May Bainbridge, Chelsea S Yeh, Daniel Borkin, Hanspeter Pfister, and Aude Oliva. Beyond memorability: Visualization recognition and recall. *IEEE TVCG*, 22(1):519–528, 2015.

[21] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D-3: Data-driven documents. *IEEE TVCG*, 17:2301–9, 12 2011. doi: 10.1109/TVCG.2011.185.

[22] Lorna M Brown, Stephen A Brewster, SA Ramloll, R Burton, and Beate Riedel. Design guidelines for audio presentation of graphs and tables. International Conference on Auditory Display, 2003.

[23] Sandra Carberry, Stephanie Elzer Schwartz, Kathleen Mccoy, Seniz Demir, Peng Wu, Charles Greenbacker, Daniel Chester, Edward Schwartz, David Oliver, and Priscilla Moraes. Access to multimodal articles for individuals with sight impairments. *ACM Trans. Interact. Intell. Syst.*, 2(4), jan 2013. ISSN 2160-6455. doi: 10.1145/2395123.2395126. URL `https://doi.org/10.1145/2395123.2395126`.

[24] Sheelagh Carpendale. Evaluating information visualizations. In *Information visualization*, pages 19–45. Springer, 2008.

[25] Diagram Center. Specific guidelines – graphs, 2022. `http://diagramcenter.org/specific-guidelines-e.html`.

[26] CFPB. Data visualization guidelines, 2022. https://cfpb.github.io/design-system/guidelines/data-visualization-guidelines.

[27] J. Choi, Sanghun Jung, Deok Gun Park, J. Choo, and N. Elmqvist. Visualizing for the non-visual: Enabling the visually impaired to use visualization. *Computer Graphics Forum*, 38, 2019.

[28] Stephen H. Choi and Bruce N. Walker. Digitizer auditory graph: Making graphs accessible to the visually impaired. In *CHI '10 Extended Abstracts*, CHI EA '10, page 3445–3450, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589305. doi: 10.1145/1753846.1753999. URL https://doi.org/10.1145/1753846.1753999.

[29] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan Lazar, and Niklas Elmqvist. Towards understanding sensory substitution for accessible visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1084–1094, 2022. doi: 10.1109/TVCG.2021.3114829.

[30] Mathieu Cliche, David Rosenberg, Dhruv Madeka, and Connie Yee. Scatteract: Automated extraction of data from scatter plots. In *Conference on Machine Learning and Knowledge Discovery in Databases*, pages 135–150, 2017.

[31] CNIB. Canadian national institute for the blind, 2022. https://www.cnib.ca/en?region=on.

[32] Disability Rights Commission. *The web: Access and inclusion for disabled people; a formal investigation*. The Stationery Office, 2004.

[33] Zhe Cui, Sriram Karthik Badam, M Adil Yalçin, and Niklas Elmqvist. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *Information Visualization*, 18(2):251–267, 2019.

[34] Seniz Demir, David Oliver, Edward Schwartz, Stephanie Elzer, Sandra Carberry, and Kathleen F. McCoy. Interactive sight into information graphics. 2010. doi: 10.1145/1805986.1806009.

[35] Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. Summarizing information graphics textually. *Computational Linguistics*, 38(3):527–574, 2012. doi: 10.1162/COLI_a_00091. URL https://www.aclweb.org/anthology/J12-3004.

[36] Frank Elavsky, Cynthia Bennett, and Dominik Moritz. How accessible is my visualization? evaluating visualization accessibility with chartability. *Computer Graphics Forum*, 41(3):57–70, 2022. doi: https://doi.org/10.1111/cgf.14522. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14522.

[37] Stephanie Elzer, Sandra Carberry, and Ingrid Zukerman. The automated understanding of simple bar charts. *Artificial Intelligence*, 175(2):526–555, 2011. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2010.10.003. URL https://www.sciencedirect.com/science/article/pii/S0004370210001670.

[38] Leo Ferres, Avi Parush, Zhihong Li, Yandu Oppacher, and Gitte Lindgaard. Representing and querying line graphs in natural language: The igraph system. volume 4073 LNCS, 2006. doi: 10.1007/11795018_25.

[39] Leo Ferres, Gitte Lindgaard, Livia Sumegi, and Bruce Tsuji. Evaluating a tool for improving accessibility to charts and graphs. *ACM Trans. Comput.-Hum. Interact.*, 20(5), November 2013. ISSN 1073-0516. doi: 10.1145/2533682.2533683. URL https://doi.org/10.1145/2533682.2533683.

[40] João Guerreiro and Daniel Gonçalves. Faster text-to-speeches: Enhancing blind people's information scanning with faster concurrent speech. In *Proc. ACM SIGACCESS Conference on Computers Accessibility*, ASSETS '15, page 3–11, 2015. ISBN 9781450334006. doi: 10.1145/2700648.2809840. URL https://doi.org/10.1145/2700648.2809840.

[41] Jonathan Harper and Maneesh Agrawala. Deconstructing and restyling d3 visualizations. 2014. doi: 10.1145/2642918.2647411.

[42] Jonathan Harper and Maneesh Agrawala. Converting basic d3 charts into reusable style templates. *IEEE TVCG*, 24, 2018. ISSN 10772626. doi: 10.1109/TVCG.2017.2659744.

[43] Highcharts. Interactive JavaScript charts library, 2022. https://www.highcharts.com/.

[44] Leona M Holloway, Cagatay Goncu, Alon Ilsar, Matthew Butler, and Kim Marriott. Infosonics: Accessible infographics for people who are blind using sonification and voice. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3517465. URL https://doi.org/10.1145/3491102.3517465.

[45] Enamul Hoque and Maneesh Agrawala. Searching the visual style and structure of d3 visualizations. In *IEEE TVCG (Proc IEEE InfoVis 2019)*, volume 26, pages 1236–1245. IEEE, 2019.

[46] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. Applying pragmatics principles for interaction with visual analytics. *IEEE TVCG*, 24(1):309–318, 2017.

[47] Jessica Hullman, Nicholas Diakopoulos, and Eytan Adar. Contextifier: automatic generation of annotated stock visualizations. In *Proc. of the SIGCHI*, pages 2707–2716, 2013.

[48] Crescentia Jung, Shubham Mehta, Atharva Kulkarni, Yuhang Zhao, and Yea-Seul Kim. Communicating visualizations without visuals: Investigation of visualization alternative text for people with visual impairments. *IEEE TVCG*, 2021.

[49] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeong-in Hwang, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. ChartSense: Interactive data extraction from chart images. In *Proc. of SIGCHI*, pages 6706–6717, 2017.

[50] Shankar Kanthara, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. Chart-to-text: A large-scale benchmark for chart summarization, 2022. URL `https://arxiv.org/abs/2203.06486`.

[51] Shahedul Huq Khandkar. Open coding. *University of Calgary*, 23:2009, 2009.

[52] Johan Kildal and Stephen A Brewster. Providing a size-independent overview of non-visual tables. In *12th International Conference on Auditory Display (ICAD2006)*, pages 8–15, 2006.

[53] Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. Answering questions about charts and generating visual explanations. In *Proc. CHI*, pages 1–13, 2020.

[54] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim. Accessible visualization: Design space, opportunities, and challenges. *Computer Graphics Forum*, 40, 2021. ISSN 14678659. doi: 10.1111/cgf.14298.

[55] Xiaoyi Liu, Diego Klabjan, and Patrick N. Bless. Data extraction from charts via single deep neural network. *ArXiv*, abs/1906.11906, 2019.

[56] Alan Lundgard and Arvind Satyanarayan. Accessible Visualization via Natural Language Descriptions: A Four-Level Model of Semantic Content. *IEEE Trans. Visualization & Comp. Graphics (Proc. IEEE VIS)*, 2022. URL `http://vis.csail.mit.edu/pubs/vis-text-model`.

[57] Alan Lundgard, Crystal Lee, and Arvind Satyanarayan. Sociotechnical considerations for accessible visualization design. In *In Proc. IEEE Visualization*, pages 16–20, 2019.

[58] Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. Chartocr: Data extraction from charts images via a deep hybrid framework. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1916–1924, 2021.

[59] Prerna Mishra, Santosh Kumar, M. Chaube, and Urmila Shrawankar. Chartvi: Charts summarizer for visually impaired. *Journal of Computer Languages*, 69:101107, 03 2022. doi: 10.1016/j.cola.2022.101107.

[60] Vibhu O. Mittal, Johanna D. Moore, Giuseppe Carenini, and Steven Roth. Describing complex charts in natural language: A caption generation system. *Computational Linguistics*, 24(3):431–467, 1998. URL https://www.aclweb.org/anthology/J98-3004.

[61] Emma Murphy, Ravi Kuber, Graham McAllister, Philip Strain, and Wai Yu. An empirical investigation into the difficulties experienced by visually impaired internet users. *Universal Access in the Information Society*, 7:79–91, 04 2008. doi: 10.1007/s10209-007-0098-4.

[62] Jason Obeid and Enamul Hoque. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proc. of the 13th International Conference on Natural Language Generation*, pages 138–147. Association for Computational Linguistics, 2020. URL https://www.aclweb.org/anthology/2020.inlg-1.20.

[63] Canadian Council of the Blind. Canadian council of the blind is the voice of the blind in canada, 2022. https://ccbnational.net/.

[64] PennState. Charts accessibility, 2022. https://accessibility.psu.edu/images/charts/.

[65] Jorge Poco and Jeffrey Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. In *Computer Graphics Forum (Proc. EuroVis)*, volume 36, pages 353–363, 2017.

[66] Jorge Poco, Angela Mayhua, and Jeffrey Heer. Extracting and retargeting color mappings from bitmap images of visualizations. *IEEE TVCG (Proc. InfoVis)*, 24(1):637–646, 2018.

[67] Quill. Narrative science, 2020. https://narrativescience.com/quill/.

[68] Rameshsharma Ramloll, Stephen Brewster, Wai Yu, and Beate Riedel. Using non-speech sounds to improve access to 2d tabular numerical information for visually impaired users. In *People and Computers XV—Interaction without Frontiers*, pages 515–529. Springer, 2001.

[69] Ehud Reiter. An architecture for data-to-text systems. In *proceedings of the eleventh European workshop on natural language generation (ENLG 07)*, pages 97–104, 2007.

[70] Prabodh Sakhardande, Anirudha Joshi, Charudatta Jadhav, and Manjiri Joshi. Comparing user performance on parallel-tone, parallel-speech, serial-tone and serial-speech auditory graphs. In *IFIP Conference on Human-Computer Interaction*, pages 247–266. Springer, 2019.

[71] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE TVCG*, 23(1):341–350, 2017.

[72] M. Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and J. Heer. Revision: automated classification, analysis and redesign of chart images. *Proc. of the 24th annual ACM symposium on User interface software and technology*, 2011.

[73] Freedom Scientific. Jaws-job access with speech, 2022. `https://www.freedomscientific.com/products/software/jaws/`.

[74] Xiang Lin Ahmed Masry Megh Thakkar Enamul Hoque Shafiq Joty Shankar Kanthara, Rixie Tiffany Ko Leong. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv preprint arXiv:2203.06486*, 2020.

[75] Ather Sharif and Babak Forouraghi. Evographs - a jquery plugin to create web accessible graphs. volume 2018-January, 2018. doi: 10.1109/CCNC.2018.8319239.

[76] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. Understanding screen-reader users' experiences with online data visualizations. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383066. doi: 10.1145/3441852.3471202. URL `https://doi.org/10.1145/3441852.3471202`.

[77] Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reinecke, and Jacob O. Wobbrock. Voxlens: Making online data visualizations accessible with an interactive javascript plug-in. In *CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3517431. URL `https://doi.org/10.1145/3491102.3517431`.

[78] Alexa Siu, Gene S-H Kim, Sile O'Modhrain, and Sean Follmer. Supporting accessible data visualization through audio data narratives. New York, NY, USA,

2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3517678. URL https://doi.org/10.1145/3491102.3517678.

[79] Andrea Spreafico and Giuseppe Carenini. Neural data-driven captioning of time-series line charts. In *Proc. of the International Conference on Advanced Visual Interfaces*, AVI '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375351. doi: 10.1145/3399715.3399829. URL https://doi.org/10.1145/3399715.3399829.

[80] Arjun Srinivasan and John Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE TVCG*, 24(1):511–521, 2018.

[81] Arjun Srinivasan, Steven M Drucker, Alex Endert, and John Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE TVCG*, 25(1):672–681, 2018.

[82] Statista. Statista, 2022. https://statista.com/.

[83] Tony Stockman, Christopher Frauenberger, and Greg Hind. Interactive sonification of spreadsheets. Georgia Institute of Technology, 2005.

[84] W3C. Standards, 2022. https://www.w3.org/standards/.

[85] R. Wang, C. Jung, and Y. Kim. Seeing through sounds: Mapping auditory dimensions to data and charts for people with visual impairments. *Computer Graphics Forum*, 41(3):71–83, 2022. doi: https://doi.org/10.1111/cgf.14523. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14523.

[86] Yun Wang, Zhida Sun, Haidong Zhang, Weiwei Cui, Ke Xu, Xiaojuan Ma, and Dongmei Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE TVCG*, 26(1):895–905, 2019.

[87] WCAG. Complex images, 2022. https://www.w3.org/WAI/tutorials/images/complex/.

[88] wordcounter. How fast does the average person speak?, 2022. https://wordcounter.net/blog/2016/06/02/101702_how-fast-average-person-speaks.html.

[89] Wordsmith. Wordsmith by automated insights, inc., 2020. https://automatedinsights.com/wordsmith.

[90] Haixia Zhao, Catherine Plaisant, Ben Shneiderman, and Jonathan Lazar. Data sonification for users with visual impairment: a case study with georeferenced data. *In Proc. TOCHI*, 15(1):1–28, 2008.

[91] Jonathan Zong, Crystal Lee, Alan Lundgard, JiWoong Jang, Daniel Hajas, and Arvind Satyanarayan. Rich screen reader experiences for accessible data visualization. 05 2022.

A Appendix

This appendix contains the supplementary materials for the Needfinding Study that we describe in Chapter 3. The script of the study prepared by the experimenter along with the study questionnaires are presented here.

A.1 Script for Needfinding Study

Script for Needfinding Study

STEP 1: PARTICIPANT GREETING

Tell Participant: "Good morning/afternoon/evening. Thank you for participating in our study. In this study, our goal is to understand how to make data **visualizations like bar charts and line charts more accessible to people who are blind or are visually impaired.** The entire process will take approximately 90 minutes. First, you will answer a short pre-study questionnaire. After that, you will participate in an open discussion session where you will be answering a few basic questions about your experience in reading data visualizations like bar charts and line charts. Then, we will demonstrate the prototype of the tool and you will explore its different features. As you use the tool, you will let us know how you feel about it. This session will be recorded for further analysis."

Action: Participants will navigate to https://infovis-userstudy.herokuapp.com/pid[*participantID*] (e.g., https://infovis-userstudy.herokuapp.com/pid1002), read and accept the **consent form**. Participants will be given an ID (e.g., 1001).

STEP 2: PRE-STUDY QUESTIONNAIRES

Tell Participant: "Now we will ask you some questions to understand how you explore different types of charts that appear on websites:

PreQ1: Please indicate your age range

PreQ2: Do you use any assistive tool (like JAWS, Narrator) for using your computer? What is it?

PreQ3: How often do you encounter charts (e.g. Bar charts, line charts, pie charts)

PreQ4: Where do you encounter the charts, in website or paper articles?

Action: Participants will answer the questions.

STEP 3: NEED-FINDING INTERVIEW

Tell Participant: "In this section, we will ask you some questions to understand how you explore different types of charts that appear on websites:

Q1: How important is it for you to understand data visualizations like bar charts, line charts, pie charts, etc?
Q2: How do you try to understand a chart that appears in a document or webpage?

Q3. Sometimes a chart on a webpage may come with alternative text or caption to describe it. Do you find such captions helpful for understanding charts? Are there any limitations of such captions/alt texts?
Q4. Do you use any computer-based tool to understand and explore data visualization or graph charts? If so, what are some useful features they have? Are there any limitations of existing tools??
Q5. How do you access data items such as the value of a bar from a chart?

Action: Participants will answer the questions.

STEP 4: DEMONSTRATING PROTOTYPE

Tell Participant: "In this segment of the study, we will first demonstrate to you a tool that aims to enhance the accessibility of data visualizations. Also, we will train you to be able to use various features of this tool. Please navigate to the link present at the bottom of the PRE-STUDY QUESTIONNAIRE page. Press F11 to set to Full-Screen mode. "
Action: Participants will navigate to the sample page (https://infovis-userstudy.herokuapp.com/original) and switch to full-screen mode. The sample charts will be used to demonstrate the features.

Tell Participant: "Tool Overview: This is a tool that aims to let visually impaired users explore various graph charts (e.g., line chart, bar chart, etc.). It converts static charts presented on websites to interactive ones so that you can access the data points individually and hear their values. You can navigate to different data points of the chart by keyboard shortcuts, and the tool will play audio to describe that point's value. The tool also provides a description of the chart, and you can adjust the length of the description. You can hear the description any time by pressing the SPACE button on the keyboard. Finally, you can select multiple data points in the chart by pressing arrow keys on the keyboard and the tool will give you a summary of the selected points.
Chart Description: The tool now has three approaches to generate chart descriptions. They are Template-based Summary, Chart-to-Text Summary. You can set the length of the Template-based Summary to Minimum, Intermediate, and Maximum length. You can press the SPACE button to hear the description.
Chart Datapoint Navigation: You can navigate through the data points of the interactive chart (e.g., bars of bar charts, dots of line charts) via the ARROW keys and the tool will tell you the value of the selected point.
Partial Summary Generation: Press CONTROL + RIGHT ARROW keys to select multiple data points (like navigating through the data points) and release the buttons to hear the summary generated based on the selected data points. Press the SPACE button to re-hear the summary.

Keyboard shortcuts:
Press D: To hear the general description of the chart.
Press SPACE: To hear the chart summary.
Press Left or Right arrow: To navigate through the bars or points of line charts.

Press Control + Right arrow: To select multiple data points to hear the partial summary
Press ESCAPE: To skip ongoing audio playback
Press H: To hear the list of keyboard shortcuts.

Do you have any questions?"
Action: Participants will go through various features and their feedback will be noted.

STEP 5: DEBRIEFING

Tell Participant: "Thank you very much again for your participation. Would you have any other comments or questions?"

Action: Get the Payment form (https://forms.gle/a6kQkbDdhvCKsPZ86) filled out.

A.2 Questionnaires

Pre-Study Questionnaires

PID: _____

1. **Please indicate your age range:**
    a. 18-24
    b. 25-34
    c. 35-44
    d. 45-54
    e. Over 55

2. **Please indicate your gender:**
    a. Female
    b. Male
    c. Other
    d. Prefer not to answer

3. **Please indicate your occupation**: _____
4. **Field of Study (if student): _____**

5. **On average, how often do you encounter charts (e.g. Bar charts, line charts, pie charts):**
    a. Never
    b. Rarely (several times a year)
    c. Occasionally (several times a month)
    d. Frequently (several times a week)
    e. Very Frequently (Several times a day)

6. **Among the following screen reader tools, which one do you usually prefer?:**
    a. JAWS
    b. NVDA
    c. Apple VoiceOver
    d. Narrator
    e. Other
    f. None

B Appendix

In this appendix, we present the supplementary materials for the user study we conducted to evaluate the efficacy of *SeeChart* tool. This includes the study script, pre-study questionnaires, post-study questionnaires, and task samples.

Note: The charts in this appendix are the charts that we used during the user study. These charts DO NOT reflect the results of the research.

B.1 Script for User Study

Script for User Study

STEP 1: PARTICIPANT GREETING

Tell Participant: "Good morning/afternoon/evening. Thank you for participating in our study. In this study, our goal is to understand how to make data **visualizations like bar charts and line charts more accessible to people who are blind or are visually impaired.** The entire process will take approximately 90 minutes. First, you will answer a short pre-study questionnaire. After that, you will participate in an open discussion session where you will be answering a few basic questions about your experience in reading data visualizations like bar charts and line charts. Then, we will move to the main phase of the study, which will involve you exploring some charts, reading the data points, hearing chart summaries, and finally, answering a few questions with the help of those.

At the end of the study, you will go through a brief interview. This session will be recorded for further analysis."

Action: Participants will navigate to https://infovis-userstudy.herokuapp.com/pid[*participantID*] (e.g., https://infovis-userstudy.herokuapp.com/pid1002), read and accept the **consent form**. Participants will be given an ID (e.g., 1001).

STEP 2: PRE-STUDY QUESTIONNAIRES

Tell Participant: "Now we will have you answer some questions. Please click the NEXT button to see the questions."
Action: Participants will click NEXT and answer the pre-study questions. If they are unable to click the NEXT button, the pre-study link (https://infovis-userstudy.herokuapp.com/new_pre) will be provided.

Action: Participants will answer the questions.

STEP 3: DEMO AND TRAINING

Tell Participant: "In this segment of the study, we will first demonstrate to you a tool that aims to enhance the accessibility of data visualizations. Also, we will train you to be able to use various features of this tool. Please navigate to the link present at the bottom of the PRE-STUDY QUESTIONNAIRE page. Press F11 to set to Full-Screen mode."
Action: Participants will navigate to the sample page (https://infovis-userstudy.herokuapp.com/original) and switch to full-screen mode. The sample charts will be used to demonstrate the features.
Tell Participant: "Tool Overview: This is a tool that aims to let visually impaired users explore various graph charts (e.g., line chart, bar chart, etc.). It converts static charts presented on websites to interactive ones so that you can access the data points individually and hear their values. You can

navigate to different data points of the chart by keyboard shortcuts, and the tool will play audio to describe that point's value. The tool also provides a description of the chart, and you can adjust the length of the description. You can hear the description any time by pressing the SPACE button on the keyboard. Finally, you can select multiple data points in the chart by pressing arrow keys on the keyboard and the tool will give you a summary of the selected points.
Chart Description: The tool now has three approaches to generate chart descriptions. They are Template-based Summary, Chart-to-Text Summary. You can set the length of the Template-based Summary to Minimum, Intermediate, and Maximum length. You can press the SPACE button to hear the description.
Chart Datapoint Navigation: You can navigate through the data points of the interactive chart (e.g., bars of bar charts, dots of line charts) via the ARROW keys and the tool will tell you the value of the selected point.
Partial Summary Generation: Press CONTROL + RIGHT ARROW keys to select multiple data points (like navigating through the data points) and release the buttons to hear the summary generated based on the selected data points. Press the SPACE button to re-hear the summary.

Keyboard shortcuts:
Press D: To hear the general description of the chart.
Press SPACE: To hear the chart summary.
Press Left or Right arrow: To navigate through the bars or points of line charts.
Press Control + Right arrow: To select multiple data points to hear the partial summary
Press ESCAPE: To skip ongoing audio playback
Press H: To hear the list of keyboard shortcuts.

Do you have any questions?"
Action: Participants will close the tab and move back to the study website. The Full-Screen mode should be ON.

STEP 4: TASKs

Tell Participant: "Click the NEXT button to navigate to the task. Press S to hear the question. Answer it by using the interactive tool. Type your answer and click the Submit and Proceed to Next Task button when you are done. You will have 50 minutes to complete all the tasks."
Action: Participants will use the interface and answer the questions.

STEP 5: POST-STUDY QUESTIONNAIRES

Tell Participant: "Congratulations. You have completed all the tasks and reached the last step of today's study. Please fill out the form."
Action: Participants will complete the post-study questionnaires.

STEP 6: DEBRIEFING

Tell Participant: “Thank you very much again for your participation. Would you have any other comments or questions?”

Action: Get the Payment form (https://forms.gle/a6kQkbDdhvCKsPZ86) filled out.

B.2 Questionnaires

B.2.1 Pre-Study Questionnaires

Pre-Study Questionnaires

PID: _____

1. **Please indicate your age range:**
    a. 18-24
    b. 25-34
    c. 35-44
    d. 45-54
    e. Over 55

2. **Please indicate your gender:**
    a. Female
    b. Male
    c. Other
    d. Prefer not to answer

3. **Please indicate your occupation**: _____
4. **Field of Study (if student): _____**

5. **On average, how often do you encounter charts (e.g. Bar charts, line charts, pie charts):**
    a. Never
    b. Rarely (several times a year)
    c. Occasionally (several times a month)
    d. Frequently (several times a week)
    e. Very Frequently (Several times a day)

6. **Among the following screen reader tools, which one do you usually prefer?:**
    a. JAWS
    b. NVDA
    c. Apple VoiceOver
    d. Narrator
    e. Other
    f. None

B.2.2 Post-Study Questionnaires

Post-Study Questionnaires

Participant ID: _____

Q1. The interactive tool is useful exploring visualizations of interest:

Answer:

1. **Strongly Disagree**
2. **Disagree**
3. **Neutral**
4. **Agree**
5. **Strongly Agree**

Q2. The interactive tool for exploring visualizations is easy to use:

Answer:

1. **Strongly Disagree**
2. **Disagree**
3. **Neutral**
4. **Agree**
5. **Strongly Agree**

Q3. How likely would you like to recommend the tool for blind people interested in visualizations:

Answer:

1. **Very Unlikely**
2. **Unlikely**
3. **Neutral**
4. **Likely**
5. **Very Likely**

Q4. Are there any features you find useful? If so, what are they?

Answer:

_____
_____
_____

Q5. Do you have any additional comment or suggestion to improve the tool?

Answer:

_____
_____
_____

B.2.3 Charts and Tasks

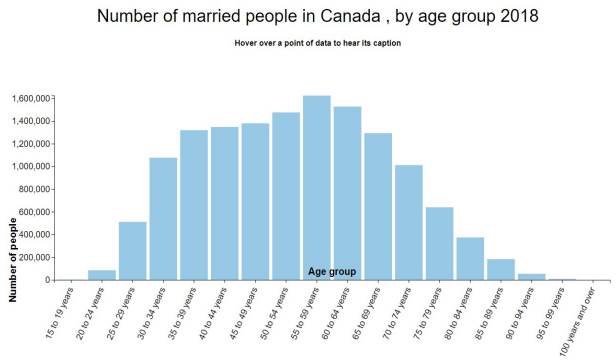
In this section, we present the charts and corresponding task from each chart that we used during the user study described in chapter 5.

B.2.3.1 Single Bar Charts

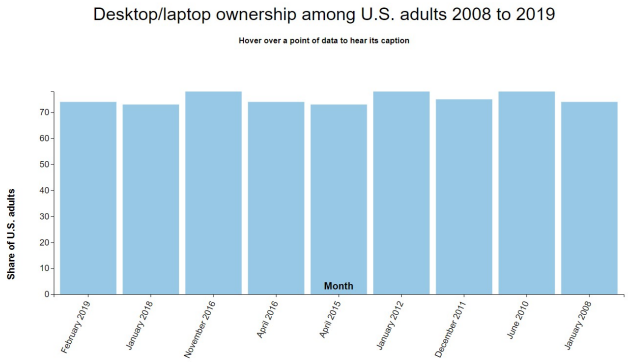
Table B.1, B.2, B.3, B.4 present the reference bar charts and the tasks users performed using the corresponding charts.

B.2.3.2 Multi Line Charts

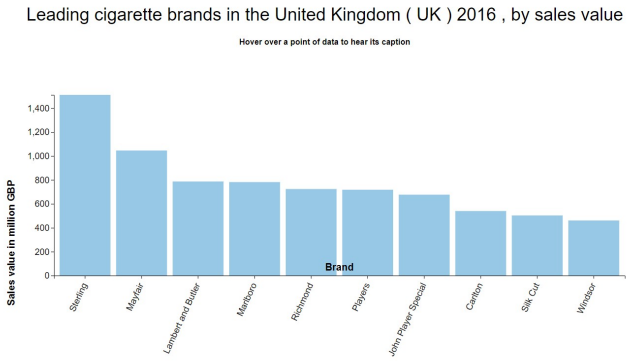
Table B.5, B.6, B.7, B.8 present the reference multi-line charts and the tasks users performed using the corresponding charts.



Task: What Age Group had the maximum number of people? [Extrema].
Reference:
https://www.statista.com/statistics/446181/married-couples-in-canada-by-age-group/

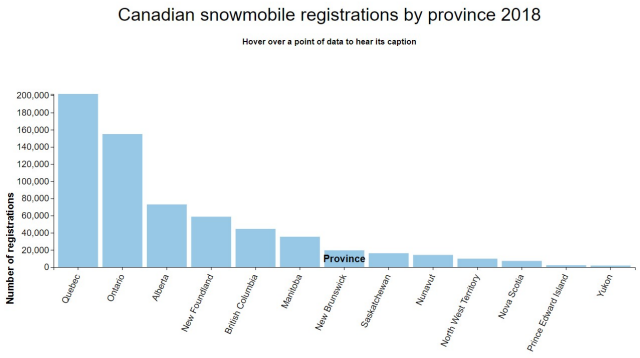


Task: What is the minimum Share of US Adults you found in this chart? [Extrema].
Reference:
https://www.statista.com/statistics/756054/united-states-adults-desktop-laptop-ownership/

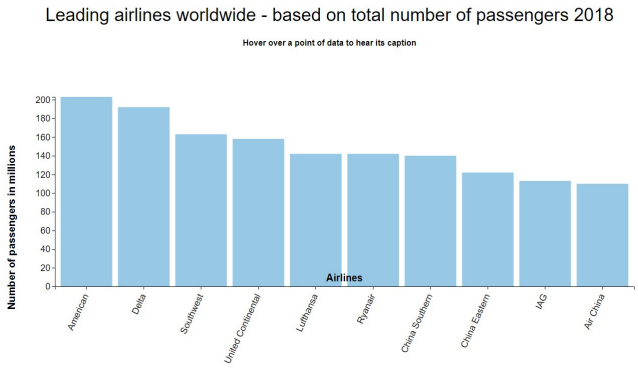


Task: Among Marlboro, Richmond, PLayers, which brand had the lowest Sales Value? [Filter, Extrema].
Reference:
https://www.statista.com/statistics/292123/top-leading-cigarette-brands-in-the-united-kingdom-uk/

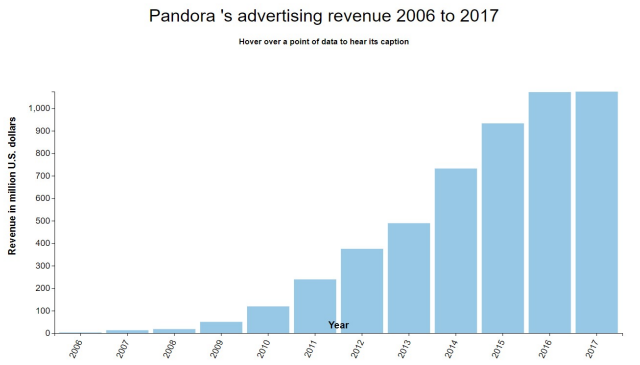
Figure B.1: Charts used in the user study along with corresponding task and reference URL.



Task: Among the first FIVE provinces shown in this chart, which one had the LEAST number of snowmobile registered? [Filter, Extrema].
Reference:
https://www.statista.com/statistics/420703/canadian-snowmobile-registrations-by-province/

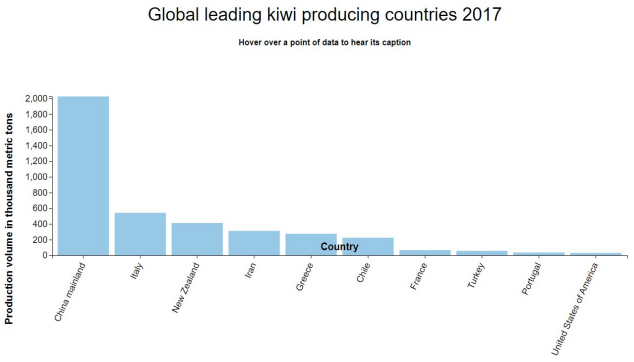


Task: Which airlines had the max and min number of passengers in 2018? [Extrema]
Reference:
https://www.statista.com/statistics/269617/top-10-airlines-worldwide-by-number-of-passengers/

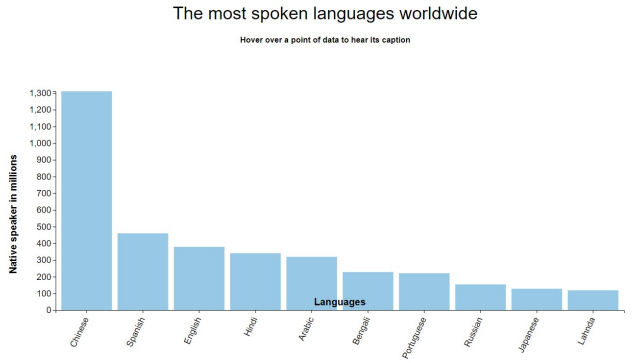


Task: From 2010 to 2016, which year had the minimum Revenue for Pandora? [Filter, Extrema].
Reference:
https://www.statista.com/statistics/190926/advertising-revenue-of-pandora-since-2007/

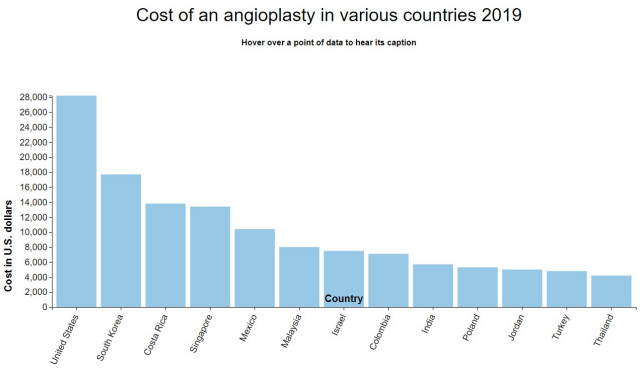
Figure B.2: Charts used in the user study along with corresponding task and reference URL.



Task: Which country had the minimum production of Kiwi in 2017? [Extrema].
Reference:
https://www.statista.com/statistics/812434/production-volume-of-leading-kiwi-producing-countries/

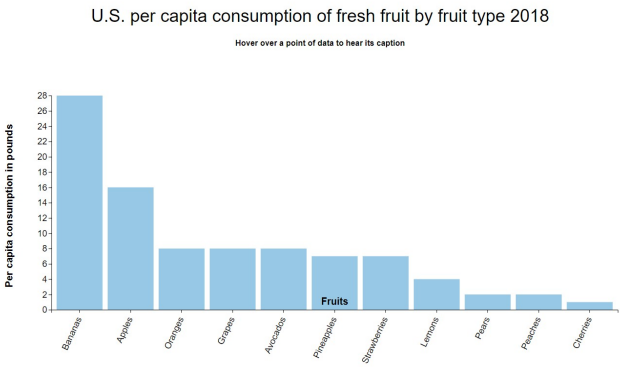


Task: Among Spanish, English, Hindi, which language is spoken the least? [Filter, Extrema]
Reference:
https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/

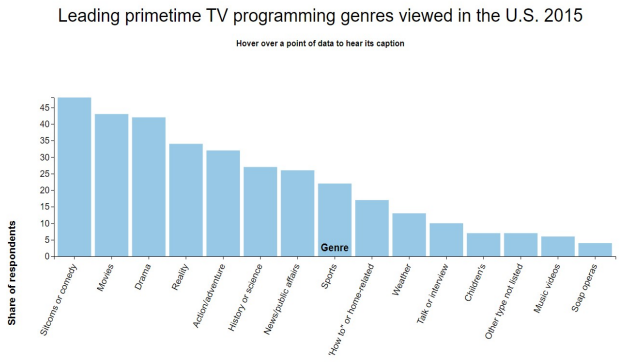


Task: In which country, the cost of angioplasty is the highest? [Extrema].
Reference:
https://www.statista.com/statistics/189973/cost-of-an-angioplasty-in-various-countries/

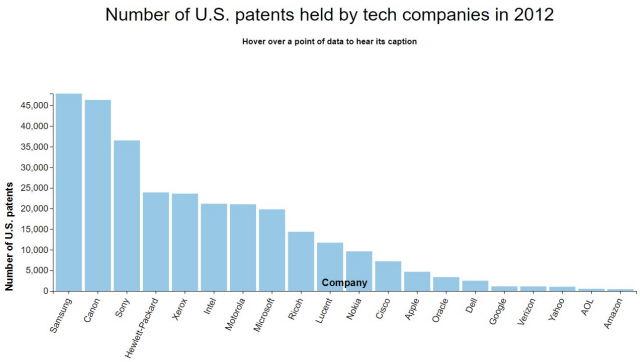
Figure B.3: Charts used in the user study along with corresponding task and reference URL.



Task: In 2018, which fruit was consumed the most in US? [Extrema].
Reference:
https://www.statista.com/statistics/257123/per-capita-consumption-of-fruit-in-the-us-by-subcategory/

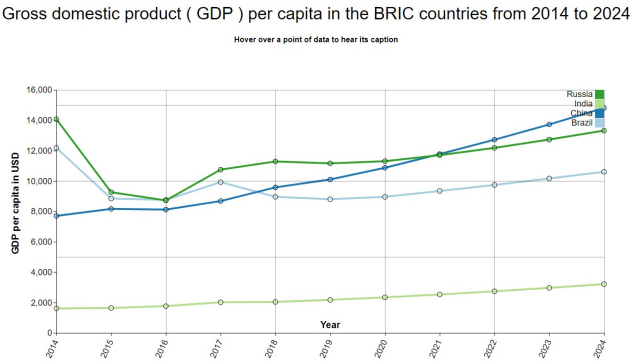


Task: Among Movies, Drama, Reality shows, which one is least favorite? [Filter, Extrema]
Reference:
https://www.statista.com/statistics/495963/primetime-tv-programming-genres-viewing-usa/

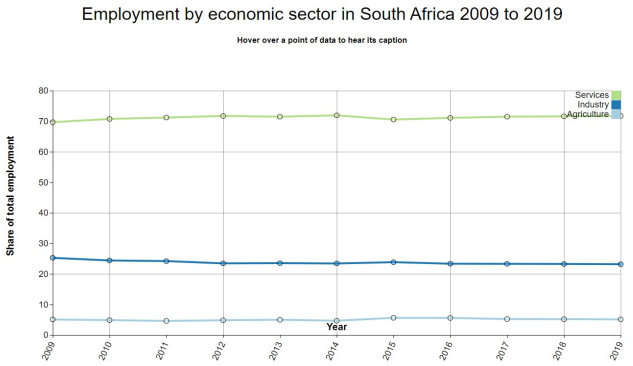


Task: Among Xerox, Intel, Motorola, Ricoh, and Lucent, which Company had the minimum number of patents? [Filter, Extrema].
Reference:
https://www.statista.com/statistics/278790/number-of-us-patents-held-by-tech-companies/

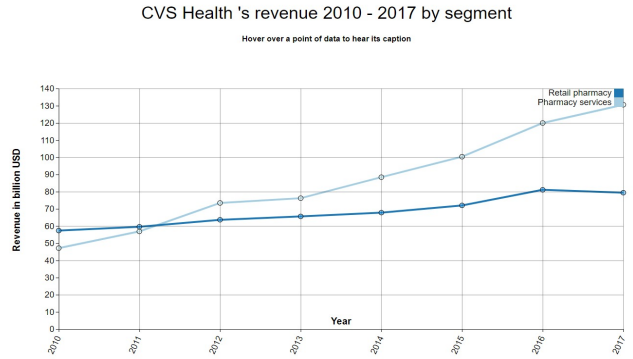
Figure B.4: Charts used in the user study along with corresponding task and reference URL.



Task: Overall, which country had the minimum GDP? [Extrema].
Reference:
https://www.statista.com/statistics/741745/gross-domestic-product-gdp-per-capita-in-the-bric-countries/

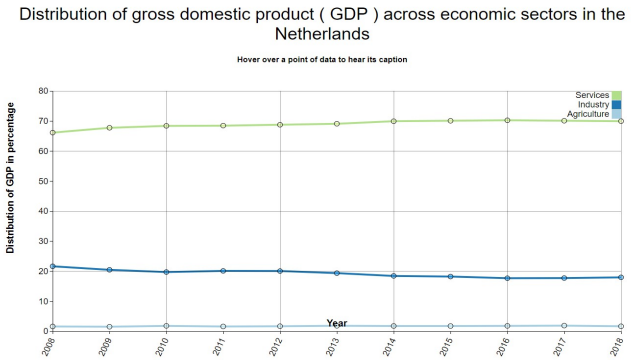


Task: Which sector is providing highest rate of employment? [Extrema].
Reference:
https://www.statista.com/statistics/578944/employment-by-economic-sector-in-south-africa/

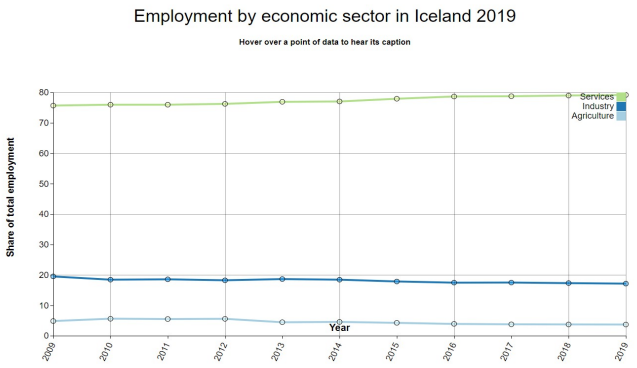


Task: In 2010 and 2011, which segment saw more revenue? [Filter, Extrema].
Reference:
https://www.statista.com/statistics/261282/cvs-caremarks-revenue-by-segment/

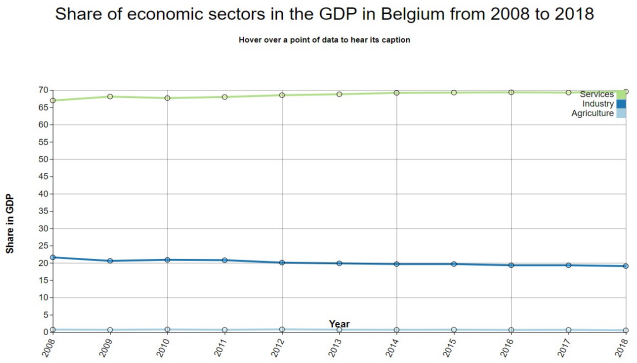
Figure B.5: Charts used in the user study along with corresponding task and reference URL.



Task: What are the trends of Services and Industry from the year 2016 to 2018? [Filter, Trend].
Reference:
https://www.statista.com/statistics/276713/distribution-of-gross-domestic-product-gdp-across-economic-sectors-in-the-netherlands/

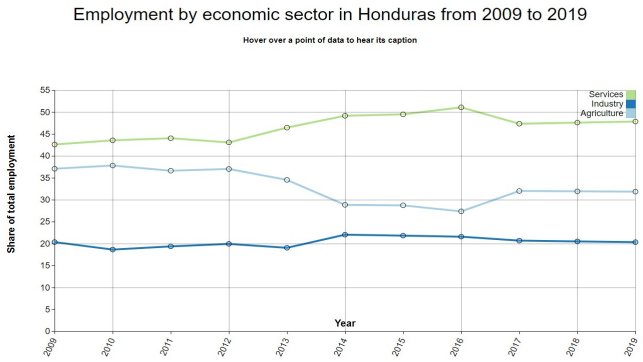


Task: Which sector had the least share of employment? [Extrema].
Reference:
https://www.statista.com/statistics/328691/share-of-economic-sectors-in-the-gdp-in-belgium/

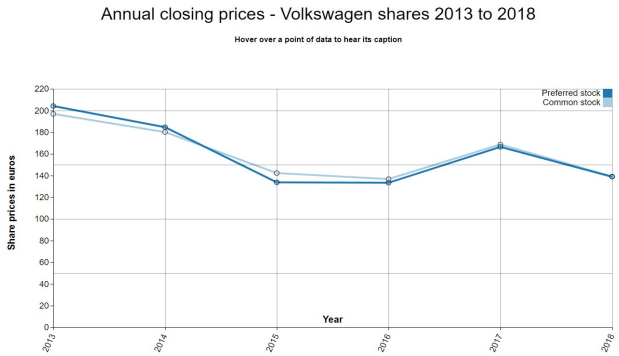


Task: Which sector maintained the highest share in GDP over the years? [Extrema].
Reference:
https://www.statista.com/statistics/328691/share-of-economic-sectors-in-the-gdp-in-belgium/

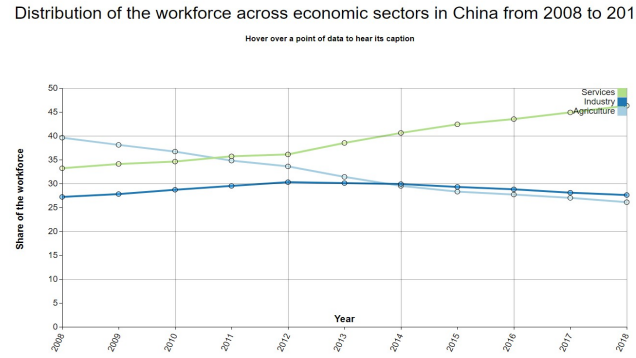
Figure B.6: Charts used in the user study along with corresponding task and reference URL.



Task: What are the trends of Services and Agriculture from the year 2012 to 2014? [Filter, Trend].
Reference:
https://www.statista.com/statistics/510041/employment-by-economic-sector-in-honduras/

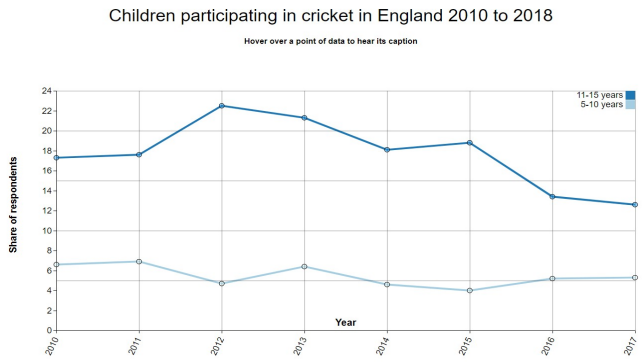


Task: What was the overall trend of stocks from year 2015 to 2017? [Filter, Trend].
Reference:
https://www.statista.com/statistics/466109/annual-closing-share-prices-of-volkswagen/

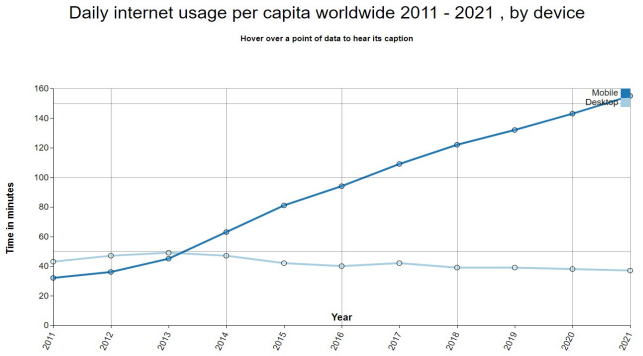


Task: What was the minimum share of workforce in the Agriculture sector? [Extrema].
Reference:
https://www.statista.com/statistics/270327/distribution-of-the-workforce-across-economic-sectors-in-china/

Figure B.7: Charts used in the user study along with corresponding task and reference URL.



Task: When did the highest percentage of children of 11-15 years old play cricket? [Extrema].
Reference:
https://www.statista.com/statistics/421078/cricket-sport-involvment-children-england-uk/



Task: What was the trend of Desktop internet usage from year 2011 to 2013? [Filter + Trend].
Reference:
https://www.statista.com/statistics/319732/daily-time-spent-online-device/

Task: From year 2016 to 2018, which source brought the least revenue? [Filter + Extrema].
Reference:
https://www.statista.com/statistics/251154/revenue-of-manchester-city-by-stream/

Figure B.8: Charts used in the user study along with corresponding task and reference URL.