

# What makes web data tables accessible? Insights and a tool for rendering accessible tables for people with visual impairments

Yanan Wang

yanan@cs.wisc.edu

Department of Computer Sciences, University of Wisconsin-Madison  
Madison, Wisconsin, USA

Crescentia Jung

csjung@wisc.edu

Department of Computer Sciences, University of Wisconsin-Madison  
Madison, Wisconsin, USA

Ruobing Wang

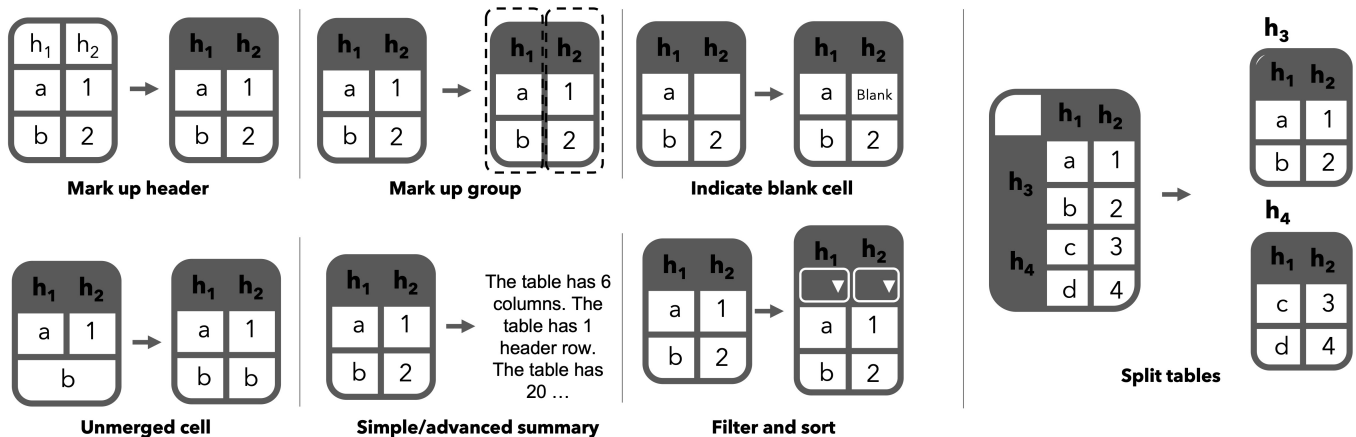
rwang582@wisc.edu

Department of Computer Sciences, University of Wisconsin-Madison  
Madison, Wisconsin, USA

Yea-Seul Kim

yeaseul.kim@cs.wisc.edu

Department of Computer Sciences, University of Wisconsin-Madison  
Madison, Wisconsin, USA



**Figure 1:** Based on the guidelines and our study, we built a browser plug-in to detect a table online and modify it to make it more accessible. The tool has multiple features that promote a better understanding of the table's structure and contents.

## ABSTRACT

The data table is a basic but versatile representation to communicate data. From government reports to bank statements, tables effectively carry essential data-driven information by visually organizing data using rows, columns, and other arrangements (e.g., merged cells). However, many tables online neglect the accessibility requirements for people who rely on screen readers, such as people who are blind or have low vision (BLV). First, we consolidated guidelines to understand what makes a table inaccessible for BLV people. We conducted an interview study to understand the importance of tables and identify further design requirements for an accessible table. We built a tool that automatically detects

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-9157-3/22/04...\$15.00  
<https://doi.org/10.1145/3491102.3517469>

HTML formatted tables online and transforms them into accessible tables. Our evaluative study demonstrates how our tool can help participants understand the table's structure and layout and support smooth navigation when the table is large and complex.

## CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Haptic devices*; *User studies*.

## KEYWORDS

Accessibility, Data Tables, Individuals with Disabilities & Assistive Technologies, Artifact or System, Empirical study that tells us about how people use a system

## ACM Reference Format:

Yanan Wang, Ruobing Wang, Crescentia Jung, and Yea-Seul Kim. 2022. What makes web data tables accessible? Insights and a tool for rendering accessible tables for people with visual impairments. In *CHI Conference on Human Factors in Computing Systems (CHI '22)*, April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 20 pages. <https://doi.org/10.1145/3491102.3517469>

## 1 INTRODUCTION

Tables are a common data representation in the wild. According to the collection assembled by Borkein et al. [34], 17.4% of data visualizations were tables (c.f., 25.4% were bar charts, and 13.4% were line charts). Tables are a great medium to represent data through rows and columns, helping readers read and locate values. As tables carry essential information for data-driven information, WCAG and other organizations offer various accessibility guidelines to aid people who are blind and have low vision (BLV) by suggesting more accessible HTML formats for designing tables online.

Designing an accessible table using HTML can be easy when the table is relatively simple (e.g., one heading without any nesting or merging cells). As long as authors use proper table tags (e.g., `<table>` to wrap the entire table, `th` for heading), the majority of screen readers can parse the structure and access the values. However, many tables in the wild use more advanced visual structures, effectively conveying the data to sighted people. For example, merged cells can immediately give the impression that multiple cells have the same value without an explicit remark. A nested table can indicate a hierarchical relationship without requiring explaining the structure. These are effective visual cues that help sighted people better extract the semantics of the data and its structure. However, while a few guidelines prescribe how authors can mark these structures with proper HTML tags and attributes, it remains unclear how they impact BLV people.

In this work, we set out to understand the experience of BLV people with data tables and improve their experience by developing a tool. First, we consolidate existing guidelines to understand how organizations devoted to accessibility advise authors to create accessible tables online. Then, we conduct an interview study with 13 BLV people to further understand the experience with well-marked complex tables and ill-designed tables. Based on the guidelines and the study, we create a browser plug-in that detects HTML formatted tables and transforms them into more accessible tables. We evaluate the usability and usefulness of our tool and report the findings.

We found that many guidelines emphasize the importance of using proper HTML markups and avoiding complex structures. Some guidelines also provide advice on how to structure complex tables. Our study found that participants can struggle to parse data from tables with multi-headings or merged cells, even when properly marked. We also identified simple features, such as sorting and filtering, that help to locate values and support inference while interacting with tables. We derived design requirements for our tool by synthesizing guidelines and the study results. Participants shared that our tool can help them understand the structure and layout better and it supports them to perform various tasks (e.g., quick value retrievals).

Our contribution is four-fold:

- (1) We survey and consolidate existing guidelines on how to design accessible HTML tables.
- (2) We report findings from a study illustrating what prior experience participants had with data tables, what tasks they carry out with tables, what makes tables inaccessible for them, and how we should improve the design of tables.
- (3) Based on the accessibility issues identified by the synthesized guidelines and the formative study, we develop a Chrome

browser plug-in that reforms the table in an accessible table while BLV people read online documents.

- (4) We report findings that demonstrate how our tool makes table interactions more accessible online.

## 2 BACKGROUND & RELATED WORK

This section presents how an HTML table is designed and how screen readers will read the table. Then, we discuss prior work on resolving accessibility issues around data tables from various research communities.

### 2.1 HTML Formatted Tables

When an author wants to create a table on a web page, the author can use the `<table>` tag and its attributes to structure the table. To be read properly via screen readers, a few tags and attributes should be set. The basic table structuring elements are the following. `<th>` indicates header cells and `<td>` indicates data cells. `<thead>` and `<tbody>` can be used to indicate the headers and actual data values, respectively, to group them semantically. `<tr>` indicates which elements belong to the same row. `<caption>` is used to describe and provide extra information about the table. In the header tag, `scope='colgroup'` or/and `scope='rowgroup'` can be added to provide information of orientations and how the header and the data cells are associated. `rowspan` and `colspan` are used to merge cells.

To demonstrate how screen readers parse simple tables, consider VoiceOver (built-in screen reader on Mac) reading the example shown in Fig 2. Once the user navigates to the table element, VoiceOver would read the caption first and explain the detected structure of the table: "In web content, New reported doses administered by day, table, three columns, five rows." Once the user navigates to the first header (Fig 2 (1) Date), it would read "Date, column 1 of 3. You are currently on a text element, inside of a cell." When the user goes to the bottom cell (Fig 2 (2) Jan 1), it would read "row 2 of 5, January first." In this case, the screen reader unpacks the abbreviation automatically. However, authors could provide their own version by using the `abbr` tag. Once the user navigates to the second column (from Jan 1 to 0 - Fig 2 (3)), VoiceOver would read the column name to indicate the associated heading of the cell and also the location in terms of columns: "Doses administered by day, 0, column 2 of 3." If the author did not mark the header using `<th>`, screen readers would not be able to execute this function. When the user moves to a different row (from 0 to 1,431,168 - Fig 2 (4)), it mentions the current row index and its value without repeating the column name "row 3 out of 5, one million four hundred and thirty-one thousand one hundred and sixty-eight." If the table is vertically laid out, as indicated by `scope='rowgroup'`, this practice will be reversed (e.g., when users move across rows, it repeats the row name).

### 2.2 Data Table Accessibility

In this section, we introduce research efforts put in to enhance table accessibility. Algorithms to detect table layout, representations, interaction techniques, and evaluative metrics and studies have been investigated.

**(a) HTML code**

```

<table>
  <caption>New reported doses administered by day</caption>
  <thead>
    <tr>
      <th>Date</th>
      <th>Doses administered by day</th>
      <th>7-day average</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Jan 1</td>
      <td>0</td>
      <td>255,223</td>
    </tr>
    <tr>
      <td>Jan 2</td>
      <td>1,431,168</td>
      <td>325,882</td>
    </tr>
    <tr>
      <td>Jan 3</td>
      <td>0</td>
      <td>325,882</td>
    </tr>
    <tr>
      <td>Jan 4</td>
      <td>337,504</td>
      <td>348,017</td>
    </tr>
  </tbody>
</table>

```

**(b) Rendered table**

(1) Date	Doses administered by day	7-day average
(2) Jan 1	0 (3)	255,223
Jan 2	1,431,168 (4)	325,882
Jan 3	0	325,882
Jan 4	337,504	348,017

**Figure 2: An example of the HTML code and the rendered table.**

**2.2.1 Detecting Table Structures.** Detecting tables and the associated elements and structure are the first steps for improving accessibility using an automated method. Several methods have been proposed to detect the structure of HTML tables [41, 57, 59, 76, 78]. Often these methods combine modeling approach, HTML tags detection, and heuristics to detect tables and their structure. For example, recent work proposes an automated pipeline to predict the structure and the orientation of web tables based on the similarity of data across rows and columns [57]. Detecting tables and their structures have been investigated in other media formats as well, including Excel (e.g., [42, 47]) and PDF (e.g., [28, 51, 69]).

**2.2.2 Developing Techniques and Representations to Improve Table Accessibility.** Several works introduce interaction techniques to improve accessibility on navigating data tables [33, 53, 61, 79]. For

example, Asakawa and Itoh developed table navigation methods using three features, namely a table cursor, a table pointer, and a cell-jumping key [33]. Using the numeric keypad on the keyboard, a user can move the table cursor horizontally (key 4 or 6) or vertically (key 2 or 8). The table pointer reads out the cells around the current cell without changing the position. The cell-jumping key allows the user to jump around the merged cells. This work considers splitting the merged cells (i.e., “ungridded” table) to support a better navigation experience. Recent work also uses keyboard interaction to support users’ spatial perception while interacting with web interfaces, including data tables [53]. The proposed system, SPRITEs (Spatially Region Interaction Techniques), leverages the 2D nature of keyboards that helps users orient themselves in the space. For example, SPRITEs allows users to navigate data tables using keys around the left edge (e.g., tab, caps lock), which map to each row, and the number keys on the top, which map to each column. While one of the table related tasks took longer time with SPRITEs compared to the baseline condition (i.e., screen readers), the evaluation demonstrated that more participants were able to complete the tasks, including finding specific values or finding all the instances of a value in a given column or row.

To further improve the accessibility of data tables, other modalities of interactions have also been incorporated to communicate data values in tables in addition to speech modality. For example, prior work has explored the used of haptic or tactile devices [35, 43], sonification [67] or both approaches combined [27, 46, 48, 54, 55]. For example, Chiousemoglou and Jurgensen develop a system that can convert tables to a tactile version using Braille display [43]. The main challenge of rendering tables with Braille dots is the size of the haptic screen, often too small to accommodate a table. They overcome the issue by leveraging zooming and scrolling features. Beyond the well-explored alternative modalities, such as tactile and auditory, the prosody of speech also has been investigated to convey the structure of tables by varying tones and accent [70–73].

To facilitate smooth navigation, several works design new representations and the associated system to generate those representations [62–65]. For example, Pontelli et al. first demonstrated the importance of spatial cues by conducting a user study and then proposed a system based on the study that generates a semantic representation of the table by incorporating the table’s structure and the generated description by human [62].

As opposed to improving interactions and representations directly, labels are a more indirect but generalizable method to improve accessibility. For example, Accessible Rich Internet Applications (ARIA) labels enhance the accessibility of every web component, including data tables. ARIA labels are read by screen readers despite not being visible on the browser, describing non-text elements. ARIA labels offer a series of principles to improve the accessibility of a website, and additional HTML attributes to provide extra information to support people with disabilities [45]. ARIA labels are used extensively online, including Wikipedia [39].

**2.2.3 Evaluating Table Accessibility and Accessibility Diagnosis Tools.** While data tables are recognized as a helpful representation to communicate data, some accessibility issues remain, hindering people with visual impairments from accessing and understanding the contents [32, 44, 50]. For example, in 2020, Haider and Yesilada

collected around 16,000 table elements from the Web (i.e., HTML elements used the table tag) and studied the prevalence of specific HTML table tags that influence accessibility [50]. They found that less than 5% of them have a caption element or a summary element, which is supposed to provide an overview of the dataset. Beyond checking HTML elements to ensure accessibility, Amtmann et al. evaluated the impact of “well-designed” (e.g., no blank cell) vs. ill-designed tables by conducting a user study. They demonstrated that proper HTML tags are helpful when tables have a complex structure and help decrease the number of steps taken by users to retrieve necessary information from the table [32]. While it is not specific to data tables, Chartability contains seven principles to ensure the accessibility of data visualizations and information around the underlying data of the visualizations [22]. Chartability provides a concrete way of testing failures, helping visualization authors to make their visualization more accessible. One of the principles (assistive failure) emphasizes the importance of table formats to be readable via screen readers: “All textual information displayed (in data labels, annotations, axes, tables, legends, etc.) must be formatted to an understandable level of content (i.e., “human-readable”). These formats must also be made into versions that can be read and parsed comfortably by screen readers.”

Beyond data tables, several recent works focus on diagnosing general Web accessibility by examining HTML elements. Diagnostic tools often take the form of browser extensions, helping designers check the accessibility of their web pages. Prominent examples include axe DevTools [23], Accessibility Insights for Web [21] and Siteimprove Accessibility Checker [25]. The Siteimprove Accessibility Checker [25] identifies accessibility issues based on the latest ACT (Accessibility Conformance Testing) rules and checks whether a website follows the WCAG 2.0 and 2.1 guidelines across all conformance levels. For example, if a webpage doesn't satisfy conformance level A guideline of Non-text Content, which asked for text alternative for non-text content, the tool may provide feedback like “Image without a text alternative.” In terms of conformance level AA guideline Resize Text, which asked text should be able to be scaled up to 200% without losing any information, the tool may provide feedback like “Text is clipped when resized.” Similarly, the Accessibility Insights for Web [21] is checking possible accessibility issues based on WCAG 2.1 guideline, and the Axe DevTools [23] supports testing accessibility issues with PDF in addition to HTML elements. With the release of the newest WCAG 2.2 guidelines, Boyalakuntla and Venigalla et al. [36] identify human-independent parts of inaccessible components based on the guideline (e.g., detecting how many authentication mechanism exists, whether visible indications are there or not) and implement a browser extension to automate evaluation. Since there are sheer numbers of evaluating tools exists, several works propose frameworks to evaluate these tools (e.g., [30, 37, 75]). For example, Abdullah [30] devises a metric to quantify the performance of an evaluative tool by comparing the number of the detected errors from a specific tool and the total number of errors detected by all the tools in aggregate. In addition, some studies use diagnostic tools to evaluate websites. For example, Mánuez-Carvajal et al. [60] conducted an analysis of the websites of the top 15 higher education centers from Spanish, Chilean and

Mexican using the tools. The results demonstrated that these institutions still have difficulties complying with the WCAG 2.0, even regarding the most basic conformance level A.

Some works evaluate the web pages based on ARIA labels. Carvalho et al. [40] evaluate the effectiveness of rich internet applications interface components (especially those developed with Bootstrap) for mobile screen readers and how they conform with ARIA 1.0, which includes some components that may be used in web data tables such as dropdowns, button groups. Romeo et al. [68] evaluate ARIA's design principles for constructing accessible tables by analyzing the accessibility guidelines, for example, using attribute “colgroup”, “aria-describeby”, etc.

*2.2.4 Using Table Structures as Webpage Layout.* Even though HTML tables still have some accessibility issues, the structured nature of tables offers many benefits when people with visual impairments parse documents and datasets online. By leveraging table structures, several works propose ways of re-designing web documents and web datasets [49, 77]. For example, EnTable is a pipeline that converts a web document containing data-driven information (e.g., an e-commerce website listing all the products with their specifications) to a data table using HTML <table> tags. When users encounter an unstructured web document, they can request EnTable to re-structure the document. Then the system asks sighted crowd workers to re-design the web document as a data table. EnTable offers an interface where sighted workers can easily structure the document by dragging and dropping HTML elements. Then, the system extrapolates the demonstrated patterns to complete the rest of the entities. William et al. evaluated the effectiveness of EnTable on accomplishing daily tasks, such as finding an item on the web page [77]. Through a controlled study and interviews, this work demonstrated that rewriting documents using <table> tags improves task performance (i.e., speed, accuracy) as well as the overall experience. Beyond the table structure, Potluri et al. also investigated the benefit of communicating visual semantics of UI elements for blind and low-vision people [66], highlighting the importance of visual semantics.

Our work is built upon a prior method on detecting table structure [57] and the understanding of the importance of spatial cues in navigating tables [62]. However, our work differs from prior work in several aspects. First, we identified what makes online data tables inaccessible via a user study. While some prior work conjecture some table designs would impact accessibility (e.g., blank cell [32] and merged cells [33]), we expand the list of inaccessible design factors and validate them empirically via a user study. Second, we contribute to the understanding of the role of tables in daily online interactions and what tasks users would perform with tables. Lastly, our approach differs from prior work in how we address accessibility issues. Many tools diagnose accessibility and give feedback to authors but do not fix the issues (e.g., [21, 23]). Our work directly intervenes in the user experience on the fly when accessibility issues happen to improve accessibility without human intervention. Also our approach improve accessibility without requiring users to learn new interactions (e.g., [33, 53]), new modalities (e.g., [67]), or additional devices (e.g., [43]).

### 3 PHASE 1: CONSOLIDATING GUIDELINES

We consolidate existing guidelines to understand how organizations devoted to accessibility prescribe HTML table accessibility. We focus on guidelines that aim at serving BLV people who use screen readers to interact with tables.

#### 3.1 Method

We first conducted Google searches with relevant keywords, such as “data table accessibility” or “HTML table guideline”. We stopped the search when we started seeing the contents deviated from HTML-formatted tables (e.g., describing CSS trick), and the contents did not contain any specific prescription on how to design HTML tables (e.g., describing the concept of `<th>`). We searched through at least 3 pages per keyword to exhaust the set. We merged the search results and removed the duplicates, resulting in 20 web pages.

The first result of all the searches was a tutorial from Web Content Accessibility Guidelines (WCAG) published by the Web Accessibility Initiative of the World Wide Web Consortium, one of the most credible organizations that offer accessibility guidelines. This page contains actionable instructions that meet the criteria of specific WCAG guidelines. When discussing the WCAG tutorial below, we included the citation for the originated WCAG guidelines, the individual techniques, and the Failure of Success Criterion published under WCAG when they are relevant.

All the 20 web pages contain mentions of how to properly use HTML tags and attributes so that the screen readers can correctly parse and communicate tables. Many of the results have examples to demonstrate the accessible data tables. While not many websites cite the WCAG, their contents directly came from the guidelines. We filtered out the web pages that only contained a subset of WCAG tutorials but kept those with any new content, even if the majority overlapped with WCAG tutorials, resulting in 7 sources. For example, we filtered out a web page published by Princeton University [26], which listed as 4th results when we googled with the keyword “data table accessibility”, as they present the dos and don’ts in designing tables by directly bringing the examples from the WCAG tutorial, and citing the WCAG success criteria.

#### 3.2 Findings

Table 1 summarizes each guideline.

**Indicating structure using HTML mark up.** Tables should mark header cells with `<th>` and data cells with `<td>` [2, 15–20]. The purpose of marking up header and data cells is to allow screen readers to identify the cell type. Moreover, table headers should include the scope attribute to associate either column or row to each header [19]. For example, WebAIM, Washington University, Yale University, PSU, and Mozilla specifies that the “scope” attribute can be used as `scope=‘col’` to associate everything within a column to a header, and `scope=‘row’` to associate everything within a row to a header [2, 15–17, 20]. Ultimately, the scope attribute allows associations between data cells and header cells that are read out by the screen reader software.

**Providing contexts by adding summary and caption.** Although the “summary” attribute has been deprecated in HTML5, include a summary to provide orientation information in tables [2, 19]. However, the summary should not repeat the content of the

caption attribute [2]. The summary attribute can be used to provide a summary of the data table structure, not the content [20]. However, Mozilla recommends the `caption` element to be used instead, as the summary attribute cannot be seen by sighted users [16]. Include a caption, which defines the overall topic or the name of a table [2, 15–20].

**Avoiding complex structure.** According to Washington University, a complex data table consists of nested columns or rows [15]. Complex tables are advised to be avoided [17]. Instead, PSU recommends replacing complex tables with a series of linked simple tables [2]. However, if complex tables are used, complex data tables should use tags such as `scope`, `<caption>`, `<summary>`, `abbr`, `acronym`, `<tfoot>`, and `<thead>` [2]. Complex tables should also mark up headers with a unique “id” attribute, for example, `<th id=‘column1a’>`. According to Deque University, headers and data associations must not be used with nested, merged, or separate tables [18]. Furthermore, PSU also recommends to avoid merged cells since screen readers can not distinguish merged cells easily [2].

WCAG provides guidelines regarding tables with irregular and multi-level headers: tables with multi-level headers should use a unique id for each table header, which will be referenced in the headers attribute in data cells. Tables with irregular headers can use various attributes and elements, such as the `<colgroup>` element to define a column group and define a row group with the `<thead>`, `<tfoot>`, and `<tbody>` elements, to define the relationships between header and data cells [19].

### 4 PHASE 2: SEMI-STRUCTURE INTERVIEW STUDY

Guidelines prescribe how to design web data tables. However, many ill-formed and complex tables in the wild do not follow the guidelines, preventing people who rely on screen readers from understanding the tables. We conduct an interview study with three aims: 1) further understanding the importance of data tables to BLV people, 2) characterizing what factors make a table inaccessible to further inform tool designs, 3) deriving insights from BLV people regarding what type of support they need to better understand table structure and its contents.

#### 4.1 Method

**4.1.1 Participants.** We recruited our study participants by circulating an IRB-approved solicitation material through mailing lists of organizations serving BLV people. We recruited participants with criteria of 1) at least 18-year-old, 2) legally blind, and 3) frequent users of screen readers. We recruited participants until the findings were saturated [29], resulting in interviewing 13 participants (Table 2). The age of the participants ranges from 20 to 44 ( $M=35$ ,  $SD=7.0$ ). Of all participants, 8 were blind, and 5 participants had functional vision. Interviews were carried out via Zoom, lasting approximately one hour each ( $M=62$  min,  $SD=11.1$  min). Participants were compensated for their time with a \$20 Visa gift card.

**4.1.2 Study Stimuli.** To situate participants with realistic scenarios, we prepared four HTML formatted data tables that participants can interact with using their screen readers. We collected around 40 HTML tables from various online web pages, ranging from online news outlets to government websites. To observe what factors

Source	Summary
WCAG [19]	This tutorial is an operationalization of WCAG criteria 1.3.1 Info and Relationships [14]. This success criterion prescribes that “Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.” This criterion emphasizes ensuring the same amount of information presented visually and auditory should be perceivable to all users. The tutorial specifically applies this criterion using HTML tags and attributes to represent web tables. For example, tables should use HTML markup to indicate header (<th>) and data cells <td>, as well as define their relationship [5, 8, 11]. Complex tables should be defined with explicit associations using scope, id, and headers attributes [7, 8, 10, 12]. Include a caption, which defines the overall topic of a table [6, 9]. Also, include a summary to provide orientation information in tables [13]. For tables with one header, mark header cells with <th> and data cells with <td> elements. For tables with two headers, set the scope attribute to highlight row to define the direction of each header. Tables with multi-level headers should use a unique id for each table header, which will be referenced in the headers attribute in data cells. Furthermore, captions can also be used to identify the table and include a summary. Tables with irregular headers can use various attributes and elements to define the relationships between header and data cells. In the table markup, use the <colgroup> element to define a column group and define a row group with the <thead>, <tfoot>, and <tbody> elements.
WebAIM [20]	A <caption> element should be used to include brief descriptive text about the table. Row and column headers should be identified with the <td> element for data cells and the <th> element for header cells. Data cells should be associated with appropriate headers using the scope attribute, which identifies whether a table header is a column or a row header. All <th> elements should have a scope attribute. Avoid spanned cells and multiple levels of header cells. The summary attribute can be used to provide a summary of the data table structure, not the content.
Deque University [18]	Table headers must be defined with <th> and a data table header must be meaningful. Data cells are recommended to be defined with the scope attribute and data group headers must be associated via scope=‘‘rowgroup’’ or scope=‘‘colgroup’’. If <th> and scope cannot be used for header/data associations, headers and id attributes must be used, and headers and associations must not be used with nested, merged, or separate tables. Tables should be represented in a <table>. <caption> or a name via aria-label or aria-labelledby should be used; <caption> is recommended to describe the purpose of the table. The name or <caption> should be unique from other tables on the same page. Avoid layout tables and headers in layout tables.
DO-IT, University of Washington [15]	All data tables should have headers identified using the <th> element. A <caption> element is also recommended. The scope attribute is recommended to be used with simple tables; for example, scope=‘‘row’’ or scope=‘‘col.’’ Complex tables should mark up headers with a unique ‘‘id’’ attribute, for example, <th id=‘‘column1a’’>. ‘‘Headers’’ attributes can include multiple headers, for example, <td headers=‘‘column1a column1b’’>.
Yale University [17]	Data tables should include the <th> element, which should have a scope attribute for associations; for example, scope=‘‘col’’ and scope=‘‘row.’’ For short tables and tables with self explanatory data, column headers are sufficient. For longer tables, include row and column headers. Include the <caption> element to include an accessible name. Avoid complex tables with multiple levels of row or column headers. Avoid empty cells to visually format the table; instead, use multiple tables.
Penn State Accessibility [2]	Use the <th> and scope tags to identify row and column headers, for example, <th scope=‘‘col’’>; reference WCAG 2.0 Guidelines 2.4.6 and 1.3.1. The <caption> tag displays a title for a table; the summary tag is placed within the table tag, which can be used to define the organization of a table or a summary of the content. Avoid spanned rows and columns in tables. Any abbreviations and acronyms used in the tables are accessible. A simple table consists of one row and header columns with no merged cells. Complex data tables should use tags such as scope, <caption>, <summary>, abbr, acronym, tfoot, and thead. Consider replacing complex tables with a series of linked simple tables. Avoid merged cells as screen readers can not distinguish merged cells easily.
MDN Web Docs [16]	Include a <caption> element inside the <table element>. The caption should contain a description of the table contents. A <caption> element is recommended in place of a summary attribute. Use <thead>, <tfoot>, and <tbody> to mark up a header, footer, and body section of a table. Add a scope attribute to the <th> element to make associations between data cells and rows/columns. Colgroup and rowgroup values of scope can be used for multiple columns or rows. An alternative for the scope attribute are the id and headers attributes. Add a unique id to each <th> element and add a headers attribute to each <td> element; each headers attribute should contain the ids of the <th> elements. The scope approach is recommended.

**Table 1: Summary of each guideline source.**

hinder participants from understanding table structure and its contents, we chose four stimuli that contain inaccessible components based on the guidelines. Parts of the tables were modified to remove personal information or to shorten its content to conduct the study in a reasonable time frame (Table 3). All participants saw all four stimuli in a randomized order.

**4.1.3 Procedure.** The interview started by asking participants demographic information, including their vision condition, age, occupation, and education level. We also asked about their prior experience with assistive technologies and their experience with interacting with data tables online.

Then we asked participants to open the four study stimuli that we previously sent through their email one at a time. Participants were asked to think aloud while interacting with the stimuli. For each table, we asked them to describe the table’s structure (e.g., how

Pid	Age	Gender	Edu.	Occupation	Light Perception	Onset Age (year)	Screen reader	Years Used
1	30	Female	B.A.	Unemployed	Y	0	JAWS	10
2	30	Male	B.A.	Teacher	Y	0	JAWS, NVDA	15
3	20	Male	B.S.	Student	Y	0	JAWS, VoiceOver	30
4	30	Female	B.S.	Unemployed	N	0	JAWS, NVDA	23
5	35	Female	B.S.	Customer service	N	0	JAWS	20
6	44	Female	B.S.	Self-employed	N	1	JAWS, NVDA	20
7	38	Female	H.S.	Unemployed	Y	0	Chrome box	3
8	36	Female	M.S.	blind rehab specialist	N	0	JAWS, NVDA, VoiceOver	20
9	39	Male	H.S.	Software tester	N	0	JAWS, NVDA	10
10	38	Female	PhD	Assistive technology manager	Y	18	JAWS	15
11	44	Female	M.S.	Assistive technology manager	N	0	JAWS, NVDA	20
12	29	Female	H.S.	Student	N	0	NVDA	10
13	42	Male	M.S.	Rehabilitation counselor	N	0	JAWS	20

**Table 2: Demographics of participants.** *Pid=Participant ID.Edu=Education (H.S.=High School, B.S.=Bachelors of Science, B.A.=Bachelors of Arts, M.S.=Masters of Sciences.*

the rows and columns were distributed). We also asked participants how they would portray what they've learned from the table to a friend in terms of data and insights. Finally, participants were asked to describe their frustrations while interacting with the table, and possible remedies, if they had any.

After examining the four stimuli, participants answered post-task questions, including how the size and the complexity of the tables impact their experience, what additional information they needed to understand table contents better, and what would be a perfect interaction system to read data tables.

**4.1.4 Analysis.** We first transcribed all interviews. With the transcriptions, we conducted a thematic analysis, a method to analyze qualitative data to extract common themes to find patterns around the manuscript [38]. To create the initial codebook, three researchers independently coded three participants' transcripts. Next, the researchers discussed and agreed on consolidating the codes, resulting in 16 high-level themes and 127 codes. This step is not prescribed by Braun and Clarke's method, but we added it in the procedure to get a broader set of codes to cover various aspects appearing in the transcripts. Then, one researcher coded all the transcripts (re-coded for the initial three sets) and revised the codes and the themes in the initial codebook, when the modifications help more accurate characterizations of a quote [38]. Then the researcher reviewed the final codebook and the corresponding quotes to adjust the codebook if necessary. The final codebook results in 14 high-level themes and 107 codes. The codebook is available in Supplemental Material.

## 4.2 Results

**4.2.1 Previous Experience with Data Table. Media and Context.** Participants encounter data tables in various places, including survey forms, websites, web documents, financial statements, etc. Specifically, participants encountered data tables online in the context of technology reports/logs (P9), government data (P11), email (P9, P12), calendar (P3, P8), online banking (P6, P10, P12, P13), online news (P2), survey (P5, P13), menus (P6, P13), online shopping (P7). P9 said they encountered data tables in "the update history for Microsoft." P11 mentioned learning "new covid data" from data tables. P12 stated "I am browsing email several times a day. I mean,

that is tables right there." P8 stated, "it might be dealing with dates like if you're on a calendar area." P2 stated encountering data tables "when I read the newspaper online sometimes." P6 stated encountering data tables on "a menu or things like that" and in "banking stuff." P13 stated, "I encounter tables when I was in college when I am doing mathematics process", "sometimes restaurant menus", and also when "completing my yearly evaluation." P10 stated encountering tables in "college and university websites." The goals of understanding tables identified by participants were to entertain, to learn, and to carry out job-related tasks. As P4 stated, "the different purposes could be anything from educational to just entertainment." P13 stated, "sometimes, maybe some work-related web pages, and I may go to some that have tables in it."

**Frequency** Of all the participants, four participants interact with web data tables on a daily basis (P1, P5, P9, P10), and seven participants interact a few times a week (P2, P3, P4, P6, P8, P11, P13). Two participants indicated they interact with tables not too often (P7, P12). As P5 stated "I probably see a few tables every day" and P10 described the frequency with "daily." P4 stated "I usually encounter them a few times a week" while P12 stated "I don't use too many tables" and P7 also encountering "not a whole lot."

**Reading data tables through screen readers** During the session, we observed that most participants navigated tables with screen readers very fluently. When describing their prior experience with data tables using screen readers, many participants describe it as positive. As P9 stated "I found what I needed." P12 stated "It was very good, it was very easy to navigate."

However, some participants expressed frustration on the number and complexity of commands required to navigate tables. As P11 stated, visually impaired people could be "not very familiar with different commands for reading tables." Two participants (P2, P13) stated that it could be hard to remember all the commands they need, and P13 suggested, "I would have to have a cheat sheet." Two participants (P4, P13) even formed an erroneous understanding of the column and row names that the screen reader provided them, misconstruing the table's orientation and causing confusion while navigating it. Even when people know the commands, the complexity still influences their experience as P3 stated, "it's a little bit different to actually use their navigation keys, plus the arrow

Table ID	Table (First few rows)					Topic	# of Columns / Rows	Inaccessible Components
T1	Date	Doses administered by day	7-day average			Vaccine administration numbers	3/61	Empty cells Long table
	Dec 20	556,208						
	Dec 21	57,909						
	Dec 22	0						
	Dec 23	393,908						
	Dec 24	0						
	Dec 25	0						
	Dec 26	936,560	277,798					
T2	REGION	YEAR	PROSECUTIONS	CONVICTIONS.	VICTIMS IDENTIFIED	NEW OR AMENDED LEGISLATION	Human trafficking law enforcement data	6/30 Merged cells Multi-headers No caption
	Africa	2013	572 (245)	341 (192)	10,096 (2,250)	7		
		2014	811 (49)	317 (33)	9,523 (1,308)	4		
		2015	1,517 (53)	719 (8)	12,125 (3,531)	6		
		2016	1,293 (54)	1,120 (21)	18,296 (13,205)	4		
		2017	1,325 (98)	515 (34)	26,517 (5,902)	2		
		2018	1,253 (37)	1,190 (29)	24,407(3,749)	2		
	East Asia & Pacific	2019	955 (71)	2,112 (32)	42,517 (1,284)	2		
2013		2,460 (188)	1,271 (39)	7,886 (1,077)	3			
T3	Expected benefit	Response	U.S.	U.K.	Australia	Total	Expected benefits of self-driving vehicles	6/42 Multi-headers No mark up for column header Merged cells
	Fewer crashes	Very likely	26.1	23.5	24.2	24.6		
		Somewhat likely	41.7	47.6	48.1	45.8		
		Somewhat unlikely	22.2	21.6	21.4	21.7		
		Very unlikely	10	7.2	6.3	7.8		
	Less traffic congestion	Very likely	19.2	15.2	15.2	16.5		
		Somewhat likely	30.5	32.1	32.3	31.6		
		Somewhat unlikely	32.9	37.4	36.2	35.5		
Very unlikely		17.4	15.4	16.2	16.3			
T4	Account Summary						Bank statement	3/36 Nested tables Empty cells Empty rows No summary No caption
	Beginning Balance			\$1,896.23				
	Deposits and Additions			2,760.85				
	Checks Paid			-1,210.00				
	Electronic & Debit Card Withdrawals			-1217.9				
	Ending Balance			\$2,229.18				
	Transaction Detail							
	Deposits and Additions							
	Posting Date	Description	Amount					
	20-Apr	Zelle Payment From Jane Doe	525					
	26-Apr	Zelle Payment From Jane Doe	5.5					
	30-Apr	University Direct Deposit	2,230.35					
		subtotal	2,760.85					

**Table 3: Study stimuli. We varied the inaccessible elements in designing the stimuli to derive design considerations for each element.**

keys to navigate them, so I wish for a better implementation from the distributor side,” and P13 also echoed, “it was pretty challenging, and I have to use different JAWS keys to get there.” To improve their experience with screen readers, two participants (P3, P11) specifically talked about the generality of the commands. P3 stated that interaction command should be “the same, regardless of what people are using” and “expected to be universal.” P11 provided a similar suggestion that “because most screen readers utilize the same commands or very similar commands”, providing “shortcut tips” can be helpful.

An ill-designed table is another reason why participants had negative experiences. P11 mentioned that “Authors don’t make

their tables based on accessible guidelines” and P13 elaborated that “some bad design in there, which puts all the columns together in one area.” Indicating heading in relation to row and column is essential for navigating tables without problems. P8 also encountered tables that “weren’t properly laid out in terms of rows and column headers, so I had to kind of figure it out myself.” P6 also mentioned not being able to “figure out which columns go with which heading, when they are not formatted correctly.”

**4.2.2 Importance & Roles of Data Table.** Participants described that accessing tables is critical in understanding the surrounding contents. P10 stated that “If a table is a design that designers are going to choose to organize the information, the table is essential to



the understanding of that content.” As a result, P10 says, “I always go through the entire table with the screen reader and the keyboard” because “want to get a good understanding.” Several participants hold a similar opinion. For example, P6 stated that “it is necessary if I need to find something specific [related to the presented article].” P8 noted the importance of equal accessibility: “I think it’s very important because I believe that more and more things now are being put into that format, especially a lot of visual data. And so, it’s imperative that we, who use screen readers, have the same access to that.” P12 “wanted to get a broad sense of the table” and P13 will be interested to learn the table when “working in some type of research.” P5 would “look through and scan through the table in order to understand what’s going on.”

**4.2.3 Data Table Tasks.** We observed several tasks that participants carried out (or wished to carry out during the study).

**Comparing values.** We observed that nine participants carried out comparison tasks while reading tables. When reading the trafficking table T2, P10 was interested in finding “if there’s been any changes between regions or make a comparison.” P8 was also interested in knowing “if the violations were more or less as the years went up.” While reading T2, P4 and P9 both noticed “UK and Australia had the same value.”

**Deriving summary statistics.** During the task, eleven participants generated quantitative insights, such as deriving the data range, frequency, min, max, mode, and mean values. For example, P4 stated “somewhat likely was the most popular category.” Several participants reasoned about the range of the response. For example, P11, stated while interacting with the self-driving car table T3, “goes from very likely to very unlikely.” P9 also verbalized the data range in the vaccine table is “mostly between 300 to 700,000.” P9 wished to know how frequently a certain value is contained in the table: “I would like to know how many times zero appeared” in the vaccine table T1.

**Extracting trend.** The majority of participants talked about trends in data. Many focus on the change in values. With the vaccine table T1, P7 found that “there were some increases, and then it goes down, and then it goes back up,” and P5 stated “The vaccine started fairly slow in the hundreds of thousand and then got going into the millions per day.” Some participants showed interest in understanding the trends within subcategories. In the trafficking table T2, P3 noticed that “one of the data points like the prosecutions, for example, has increased.” Also, P10 was interested to learn “if things have decreased or increased in a specific year.”

**4.2.4 Factors that Make Data Table Inaccessible.** We capture factors that hinder participants from understanding the structure and the contents of tables. Participants shared their thoughts and ideas based on their prior experience as well as the study.

**Missing & wrong mark up** As P11 stated, the table should be in <Table> tag: “sometimes they’re not tagged, and it can be a real pain.” Many participants echoed the importance of header markup (i.e., <th>). P11 mentioned that “sometimes you kind of have to memorize the categories [if there is no header mark up].” P4 and P5 raised a similar statement that “well marked up tables are really important so that it can actually read those headers” and “needs to be labeled correctly.” Otherwise, as P3 stated, “it can be really hard to navigate.” P8 shared a specific problem she encountered because

“the table isn’t laid out correctly”, she “couldn’t do the table layer command.” P10 also mentioned, “There was no automatic table caption announced, so you’d have to actually navigate with the keyboard to the headings to try and figure out what the table is.”

**Merged cell** P11 stated that the problem of merged cells is that “it didn’t have any indication of that.” In addition, the way that the screen reader function made it harder for BLV people to learn about the table, as hinted by P12: “[the screen reader] would jump you back to the top of that section” for merged cell. Because the screen reader is not able to parse the merged cells, P4 and P10 mistakenly thought the value of the last column as the second to last column. Participants perceived the merged cell as a non-uniform structure, having trouble understanding the contexts. P11 stated, “some columns had more rows than others, it was not a uniform table, sometimes they will be a little tricky.” P6 also mentioned that it’s “very time consuming because of the way it’s formatted.” P13 also emphasizes that “stable layouts” are easy to read. P7 suggested that “unmerged” cells would be helpful: “not merged together as much ... may help somebody that’s using a screen reader.” P11 also talked about adding some information in the summary of the table to indicate the use of merged cells. As P11 stated, it would be “great to have a better description of that table. For example, it has some cells that are merged or that they’re organized ... in different sections. ... a little bit more information to just help navigate it.”

**Multi-level headings** When tables contained multi-level headings, participants had difficulty tracking them because screen readers do not provide any indication about the multiple headings, as P9 stated: “there’s nothing telling me what it’s for.” While navigating the table, the lack of header information can be error-prone, as exemplified by P4 and P6, which thought the first level heading was a separate row. To mitigate this concern, P11 suggested, “if it’s a multi-level, I need to know when I’m jumping.” P8 had a similar experience: “I’d have to know what the numbers referred to.” Also, P5 stated, “screen reader doesn’t tell you when you’ve switched questions”, but “when one of those headings over on the very left changed, we need to know what the new heading is, with the screen reader, otherwise you’d have to go up and navigate and figure out what this is.” P3 suggested making “smaller tables” so that “users can digest the information.” P2 further suggested splitting “different parts [of the data] in different tables”. Similarly, P6 suggested “maybe different category can be its own table and make them into a heading”, “we can use the screen navigation to jump to the specific tables.” P10 also echoed the same feeling: “make them separate tables so that the information can be consumed in an easier way sure.”

Sometimes subtitle marked as heading causes confusion (e.g., Account Summary in T4). In this situation, screen readers repeat all the subtitles instead of only reading the heading of the active column under most participants’ settings. As P11 stated, “it reads the whole thing,” “I have to wait for it to read to me”. P5 and P12 echoed similar frustrations.

**Blank Cells and Rows** Blank cells without proper indication often left participants confused. P8 stated, “I would have to scroll a few times after, and it was mainly a blank cell.” In P6’s interaction with table T1, P6 missed the blank cell and lost track of the cursor’s position. P6 stated, “sometimes [the screen reader] says blank, but sometimes it actually has a name. So I think that’s the confusing

part.” P10 noted a similar situation, “it’s kind of difficult to know where you are if the cells are blank.” P8 also reflected on their confusion caused by a blank row: “there were some parts that I was like, it would say the posting date, but it wasn’t really giving information, I think that was maybe because there was a blank row in the table.” To address this issue, P10 suggested signaling blank cells “typically for an empty cell you’d put like a dash.”

**Oversized table** Oversized tables make it hard for participants to find the information they want. This feeling was shared among several participants while interacting with a long table. As stated by P9 “it’s massive, I mean, I’m overloaded.” P3 stated, “to really understand the table, you have to read again” and “what is going on with these numbers is going a bit over my head.” P10 stated that “the longer the table, usually the more issues can arise.” P6 also expected “I wouldn’t need to move through all these years” while reading T2. Oversized tables may impact recall as well. P12 noted “If there’s too much clutter, or things are being repeated, it can make it harder to remember the content.” P4 further mentioned “especially with more rows and columns but especially columns, I think, the harder it is to remember things.” P5, P6, P8, P9, P12 believed that remembering the content of the table is important for them to derive insights. However, as noted by P9, “unless I went back up to the top of the table to see what was for what, I would probably forget very quickly what it was.” To assist in parsing oversized tables, P3 suggested that functionality to filter values could help table navigation: “You know another thing that useful is, for every table on the web, to be able to be filtered.” P9 suggested a similar idea: “You can choose from a list box, what information you’re displaying.” P11 also expected “to select certain parts of a table.” In addition to table filters, better summaries were also identified as useful. P8 noted “Maybe having like a narrative summary instead of having just the raw data.” A summary should include “how many cells are on the table”, “the name and some of the data that’s on the table” as P7 stated, and “what each row is, and . . . what are each value” as P13 said. P12 also asked to include in the summary “information about what is in each column, what each column is covering.” One of the reasons identified for using good summaries is to help readers form a mental picture of the table before navigating it. As expressed by P12, with informative summaries that offer a good overview of the table, “it will be easier for people to pick out what they wanted to find.”

## 5 BROWSER EXTENSION: MAKING TABLE MORE ACCESSIBLE

We set out to develop a tool that helps the structural and informational accessibility problems identified by the guidelines and the interview study. Browser extensions are a natural choice for enhancing the accessibility of published tables online as it allows us to detect table elements and modify tables directly.

### 5.1 Design Goals (G) & Features (F)

Based on the guidelines and the interview study, we derive three design goals and corresponding features that make HTML tables more accessible. Specifically, 1) we automate detectable features that guidelines prescribe, 2) support important tasks identified during the interview (e.g., comparing values, deriving summary

statistics), and 3) address the pain points that we observed during the interview (e.g., wrong markups, merged cells, and others).

- **G1: Supporting constructing a mental model of structure**

- F1: Add mark up for header when it is missing. All surveyed guidelines [2, 15–20] and our participants emphasize the importance of the header. Screen readers will read the marked header whenever users navigate data cells across different cells belonging to the different header.
- F2: Add mark up for Rowgroup or Colgroup, when it is missing. Many guidelines [16, 18, 19] highlight the necessity of indicating grouping information of the table to convey users the orientation information.
- F3: Indicate blank cells. Several participants (P6, P8, P10) unintentionally skipped the blanks, which could lead to misunderstanding the contents of the table.
- F4: Unmerge merged cells. Some guidelines [2] advise people not to use merged cells, and our participants (P4, P6, P10, P11) echoed that the merged cell disrupts them to understand the table contents.

- **G2: Providing additional contexts**

- F5: Provide a simple summary when it is missing. A few participants (P7) pointed out that some versions of screen readers do not provide a simple summary of the table, such as the number of rows and columns. A few guidelines [2, 19, 20] are advised to provide a summary.
- F6: Provide advanced summary. Eleven participants conducted or wished to derive quantitative insights. The advanced summary will support the task better. This feature will facilitate the identified task in our study “Deriving summary statistics”.

- **G3: Facilitating easy navigation**

- F7: Split tables when multi-heading exists. While a guideline [19] offers how to structure the multi-heading tables, 7 participants were confused about the structure, and 2 of them failed to understand the structure of the headings. Splitting the table into multiple simple tables will make users easy to parse the structure.
- F8: Provide filtering and sorting capability. Four participants mentioned that the capability to extract specific information would aid the understanding of the table contents. This feature will facilitate the identified task in our study “comparing value” and “Extracting trend”.

### 5.2 System Preliminary

We developed a Chrome extension that detects HTML tables and modifies inaccessible aspects of the tables. In developing the extension, we followed the accessibility guidelines published by Chrome Developers [1]. The overview of the workflow is shown in Figure 3. The workflow starts by detecting table elements when a user loads a web page. Then, the tool detects the orientation as suggested by prior work [57]. The detected orientation can then be used to infer where the heading should be located and whether cells should be grouped with rows or columns. The prior work infers the table orientation based on its contents instead of solely relying on markup. For example, it leverages an assumption that the type of data (e.g.,

text, numerical value) should be identical along rows (at least more than 50%) in horizontal tables or along with columns in vertical tables. This work [57] also provides ways to detect a “decorative table”, a table whose contents and its understanding do not depend on the table structure (Example of a decorative table is in Fig 4). There are mainly two types of decorative tables. The first type is a table that often consists of full text and contents that are not required to be in a table. Another type is a “layout table,” which uses `<table>` tag to format the layout of the entire web page. We used the same algorithm to detect and filter out the decorative table to apply our tool’s features.

### 5.3 G1: Supporting constructing a mental model of structure

**5.3.1 F1: Marking Up Headers.** Based on the detected orientation, the tool adds `<thead>` and `<th>` tags to the header so that screen readers will repeat the header when users move across the headers. To detect the multi-level headers, the tool calculates the similarity between  $N$ th cell and  $N+1$ th cell for the entire column for each row (if it is a vertical table) and finds  $N$  where has the lowest similarity to determine where the header ends. Then the tool will wrap the header of the table elements with `<thead>`. If the table has a horizontal layout, we will modify the first cell in each row by replacing the first cell to `<th>`, if originally it is `<td>`. The coding details of detecting orientation and marking up the headers are described in Algorithm 1.

The row that has the most number of cells vertically similar to the cells below is regarded as the start of the table body part, and rows before that are regarded as headings, and the tool will wrap those heading rows of the table elements with `<thead>`. If the table has a horizontal layout, we will modify the first cell in each row by replacing the first cell to `<th>`, if originally it is `<td>`.

**5.3.2 F2: Mark up Rowgroup or Colgroup.** Explicitly marking up Rowgroup or Colgroup helps convey the semantics of grouping. Based on the detected orientation, the tool adds the `scale` attribute to the cells. If the table has `<thead>`, for cells that have the `colspan` attribute and the value is bigger than 1, we add “colgroup” as its `scale` attribute. Otherwise, we add “col” as its `scale` attribute. Similarly, if the table has an horizontal layout and it has a `rowspan` attribute with value larger than 1 for `<th>` in the first cell of each row, then we add “rowgroup” as its `scale` attribute. Otherwise, we add “row” as its `scale` attribute.(Algorithm 1)

**5.3.3 F3: Indicating Blank Cell.** We first detected `<td>` elements that do not contain any content and inject the word ‘blank cell’ to provide an indication (Algorithm 2).

**5.3.4 F4: Unmerged Merged Cells.** The tool fills up the merged cells by repeating the values to all cells within the merged span, unless the cell is a header (Table 4). Specifically, if a cell is not inside of `<thead>` or the corresponding row is marked as a subtitle, the tool parses the value of the `colspan` or `rowspan` attribute (i.e., how many cells were merged). Then, it creates the appropriate number of cells and duplicates the same value to each one of them (Algorithm 2).

## 5.4 G2: Providing additional contexts

**5.4.1 F5: Simple Summary.** Most screen readers automatically detect the structure of the table and provide the information to users. Still, some do not (e.g., two versions of JAWS that two participants were using in our study). Our tool generates a summary and adds it to the `<summary>` element. The first part includes a description of the table’s caption (if it exists), type (decorative or data table), and the number of columns and rows.

**5.4.2 F6: Advanced Summary of the Column/Rows.** The advanced summary includes the information about the orientation since knowing orientation guides users on how to navigate the table. The tool further detects the data type of the columns or rows (either text or numerical value) to generate the proper summary. If the column or row contains numerical values, then the tool computes the range of the values. If the column or row contains text, the unique value sets are generated. Then, the tool injects the information inside of the `<caption>` tag to allow screen readers to read them out.

For example, the summary generated for the table in Table 5 was the following.

This is a data table. The table has 6 columns. The table has 1 header row. The table has 12 rows in the main body. The table is vertically laid out. It has 1 level column headers. Level 1 includes Expected benefit, Response, U.S., U.K., Australia, Total. Column 1, Expected benefit, is a text column with 2 unique values, including Fewer crashes and Less traffic congestion. Column 2, Response, is a text column with 4 unique values, including very likely, Somewhat likely, Somewhat unlikely, and Very unlikely. Column 3, U.S., has values ranged from 10 to 41.7. Column 4, U.K., has values ranging from 7.2 to 47.6. Column 5, Australia, has values ranging from 6.3 to 48.1. Column 6, Total, has values ranging from 7.8 to 45.8.

## 5.5 G3: Facilitating easy navigation

**5.5.1 F7: Splitting Tables.** Since there is a simple shortcut (e.g., Control+Enter for JAWS) that allows users to move to the next table, dividing up tables will lower the burden of understanding complex tables without adding too many interactions.

The tool addresses this source of confusion by splitting a large table into multiple tables in two situations. The first case is when multiple tables are joined by an empty row. In this case, the tool splits the sub-table into a standalone table, uses the detected subtitle as its caption, and reuses the headers of the original table.

The second case is when the table has multiple headers. In this case, the tool splits the table by parsing the cells related to each header cell and taking the value in the header cells as the caption. An example is shown in Figure 5.

**5.5.2 F8: Filtering & Sorting.** The tool supports a filter function, allowing users to select target categories or values from the list. To achieve this, the tool collects all the unique values in each column of the table. The tool strips all the tags (e.g., `<a>`, `</a>` and leaves only text. Then the tool creates a `<select>` element with an `<option>` for each of the unique values. This allows users to select any of the

**Algorithm 1:** Detect orientation and Mark up headers

---

```

input : An HTML table
output: An HTML table with headers marked up and information of the orientation and subtitles of the table

1 maxColLen ← 0; fullSizeRowIndexes ← [];
2 for i ← 1 to rowSize(table) do
3   colLen ← 0;
4   for j ← 1 to columnSize(table[i]) do
5     colLen ← colLen + table[i][j].colSpan;
6   if colLen > maxColLen then maxColLen ← colLen;
7 for i ← 1 to rowSize(table) do
8   if colLen(i)==maxColLen and columnSize(table[i])==1 then fullSizeRowIndexes.append(i);
9 if fullSizeRowIndexes[0]==1 then
10  use first row as table title and remove it from table body and modify the indexes in fullSizeRowIndexes;
11 Splitting cells here, calculating similarity matrix and detecting orientation manipulate on a temporary table;
12 for i ← 1 to rowSize(table) do
13   for j ← 1 to columnSize(table[i]) do
14     if table[i][j].colSpan > 1 or table[i][j].rowSpan > 1 then
15       split cell[i][j] based on colSpan and rowSpan (if applicable);
16       copy all attribute except colSpan and rowSpan to added cells;
17       set content as blank for added cells;
18 similarityMatrix ← []; rowOrientation ← []; colOrientation ← [];
19 for cell[i,j] in table do
20   verticalSimilar ← 0;
21   horizontalSimilar ← 0;
22   if similar(cell[i,j], cell[i+1,j]) then verticalSimilar ← 1;
23   if similar(cell[i,j], cell[i,j+1]) then horizontalSimilar ← 1;
24   if verticalSimilar==1 and horizontalSimilar==1 then similarityMatrix[i,j]← ‘b’;
25   else if verticalSimilar==1 and horizontalSimilar==0 then similarityMatrix[i,j]← ‘v’;
26   else if verticalSimilar==0 and horizontalSimilar==1 then similarityMatrix[i,j]← ‘h’;
27   else similarityMatrix[i,j]← ‘n’;
28 for row[i] in table do
29   if count(‘h’ or ‘b’ in similarityMatrix[i,]) / width of table >= 0.5 then rowOrientation[i]← ‘h’;
30   else rowOrientation[i]← ‘n’;
31 for column[j] in table do
32   if count(‘v’ or ‘b’ in similarityMatrix[,j]) / length of table >= 0.5 then colOrientation[j]← ‘v’;
33   else colOrientation[j]← ‘n’;
34 Marking up cells manipulates on the original table;
35 if count(‘h’ in rowOrientation) / length of table >= 0.5 then
36   tableOrientation ← ‘horizontal’;
37   mark first cell in each row as heading (change <td> to <th>) with ‘scale’ set based on ‘rowspan’;
38 if count(‘v’ in colOrientation) / width of table >= 0.5 then
39   tableOrientation ← ‘vertical’;
40   mark cells in rows before the first row that has the most ‘v’ or ‘b’ cells as heading (change <td> to <th>) with ‘scale’ set based
   on ‘colspan’
return : fullSizeRowIndexes, tableOrientation

```

---

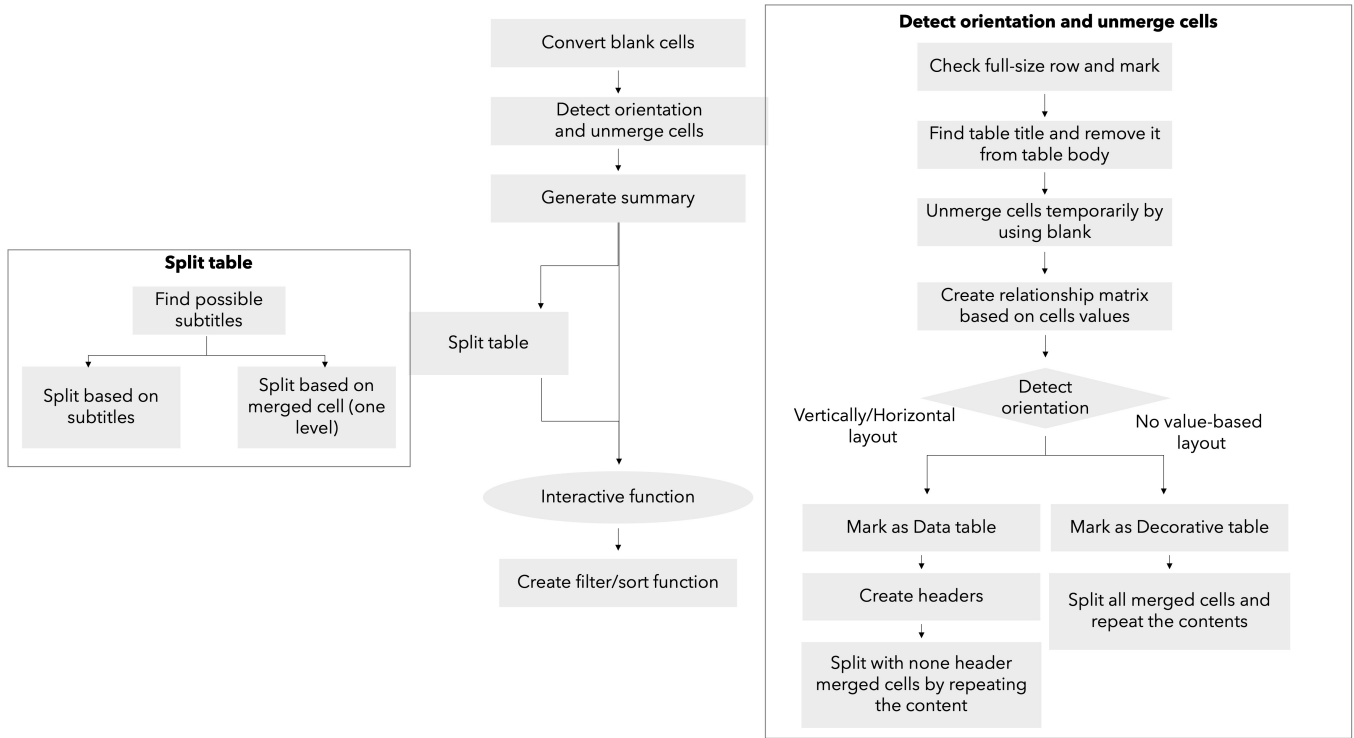


Figure 3: The workflow of the system

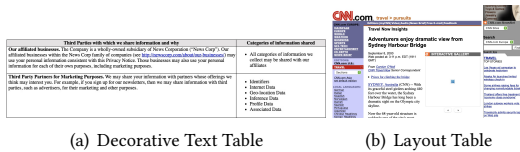


Figure 4: Screenshots of two typical types of a decorative table (Subfigure 1 displays a series of paragraphs that are formatted as a table. Subfigure 2 displays a webpage whose structure is designed as a table.)

values they want to filter the table with. The `<select>` elements are displayed in a new row within the `<thead>`. The tool is triggered by any changes of the options selected and modifies the `display` attribute of each cell in order to display only the rows that match the selected values.

The tool also supports the sorting function. By clicking the column’s header, it sorts the column values. The tool will create a `<button>` after each column name and name it as “sort” so that users know that it will sort the corresponding column. The button will have an `aria-description` attribute to indicate that the button will sort the values in ascending and descending order, in turn. If the first row contains the highest value, clicking the sorting button will sort the table in ascending order. Otherwise, the rows are sorted in descending order.

**Algorithm 2:** Indicate blank cell and Unmerge cells

**input :** An HTML table, *fullSizeRowIndex*  
**output :** A modified HTML table

```

1 for cell[i,j] in table do
2   if content in cell[i,j] is Null then
3     cell[i,j] ← ‘blank cell’
4 for cell[i,j] in table do
5   if i not in fullSizeRowIndex then
6     if cell[i,j].tag==<td> and (cell[i,j].colSpan>1 or
7       cell[i,j].rowSpan>1) then
8       unmerge the merged cells based on colspan or
9       rowspan (if applicable);
       copy all attribute except colspan and rowspan to
       added cells;
       duplicate values to fill in the added cells;

```

**6 EVALUATION**

**6.1 Performance Evaluation**

To evaluate the performance and the generalizability of our tool, we collected 20 data tables online that consist of 10 simple tables and 10 complex tables based on the definition of the guideline from [2]. Since it is challenging to collect web tables systematically, we took a manual approach by querying “table below” with specifying URLs

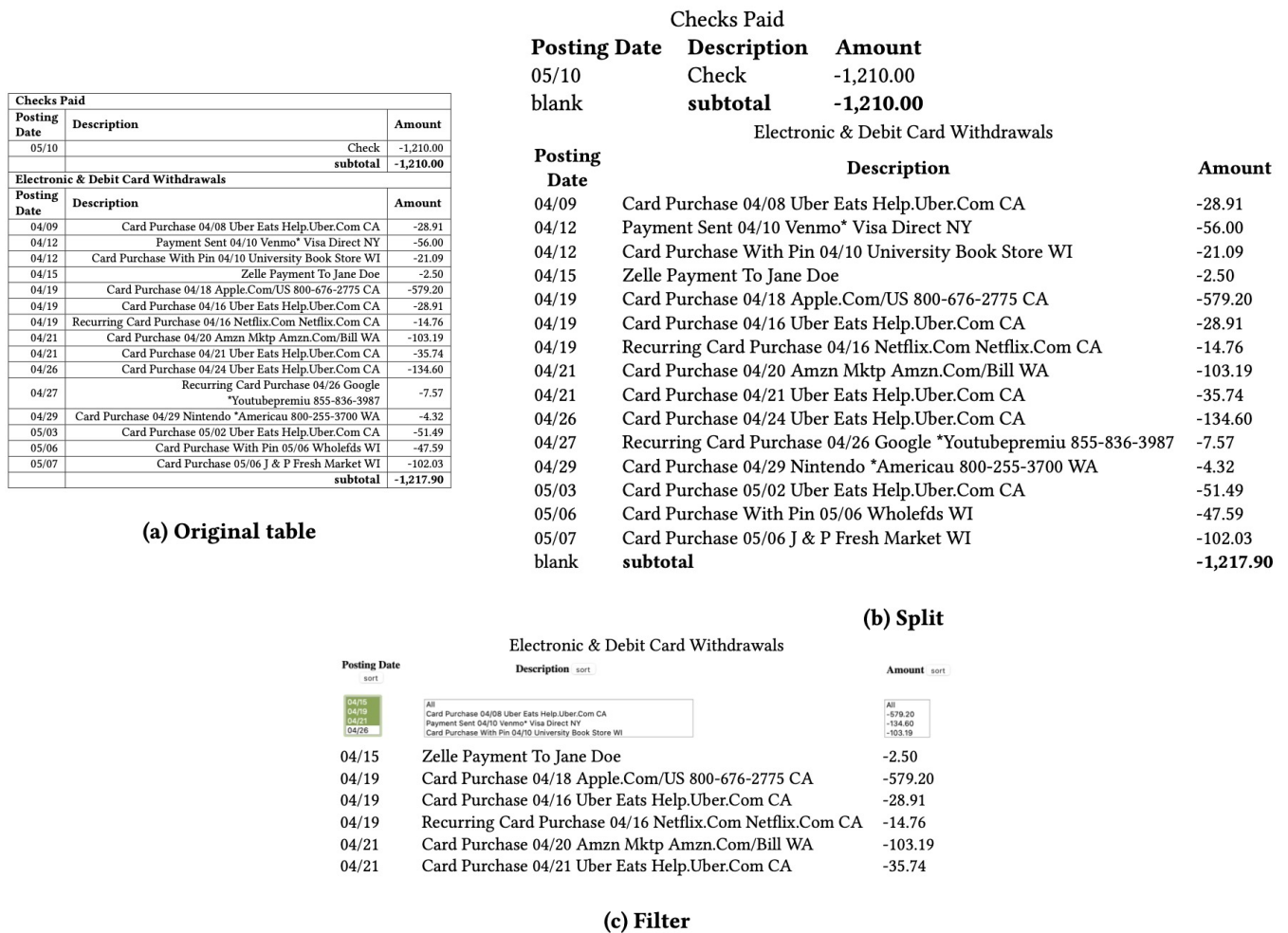


Figure 5: (a) The original table. (b) An illustration of how the tool splits the table into multiple tables with multiple headings. (c) An illustration of the filter features. It offers multiple selections (Green boxes).

of various news outlets government organizations (e.g., querying Google allintext: ["table below"] site: washingtonpost.com).

6.1.1 *Data Preliminary & Analysis Method.* Within 20 collected, three of them were decorative tables [57]. Except for those three, all the other seventeen tables have some types of accessibility issues. Ten do not have a header marked up (i.e., missing <th> or <thead>), all of them do not specify their orientation using colgroup or rowgroup, nine have at least one merged cell, fourteen of them do not have summary tags, two have multiple headers, two have blank cells with no indication, and eight of them are considered oversized, resulting in 62 accessibility issues to solve. We also evaluated whether each table can generate the summary and filtering and sorting features, resulting in 96 items to check. To assess the performance, we ran our tool on each web page and observed the rendered result.

6.1.2 *Results.* The results demonstrated that the tool is able to solve most of the accessibility problems of the tables as we expected

(94/96 cases). Fig. 7, Fig. 8 and Fig. 9 are some distinguished examples. Fig. 7 demonstrated that the tool is able to correctly detect the (sub)titles and the headings and mark them correctly by converting the (sub)titles in the first row to <caption> and marking headings with <th>. Fig. 8 demonstrated the tool is able to split correctly based on the (sub)titles detected, generate meaningful captions and attach headings for each subtable. Fig. 9a and Fig. 9b demonstrated the tool is able to correctly add sort and filter function to a table even when it has multi-level headings.

6.1.3 *Problems.* We analyzed the two cases to inform our next iteration. One was derived from unformatted content. An example is shown in Fig. 9c. Because the original table did not use a uniform format for the data (one of the data values accidentally contains %), that specific value is not able to be correctly sorted.

Another problem came from detecting the decorative table, developed by [57]. We validated via testing that this algorithm can detect most of the decorative tables. However, when a data table is



REGION	YEAR	PROSECUTIONS	CONVICTIONS.	VICTIMS IDENTIFIED	NEW OR AMENDED LEGISLATION
Africa	2013	572 (245)	341 (192)	10,096 (2,250)	7
	2014	811 (49)	317 (33)	9,523 (1,308)	4
	2015	1,517 (53)	719 (8)	12,125 (3,531)	6
	2016	1,293 (54)	1,120 (21)	18,296 (13,205)	4
	2017	1,325 (98)	515 (34)	26,517 (5,902)	2
	2018	1,253 (37)	1,190 (29)	24,407 (3,749)	2
	2019	955 (71)	2,112 (32)	42,517 (1,284)	2
East Asia & Pacific	2013	2,460 (188)	1,271 (39)	7,886 (1,077)	3
	2014	1,938 (88)	969 (16)	6,349 (1,084)	3
	2015	3,414 (193)	1,730 (130)	13,990 (3,533)	10
	2016	2,137 (51)	1,953 (31)	9,989 (310)	7
	2017	2,949 (77)	3,227 (72)	4,915 (669)	0
	2018	2,351 (63)	1,275 (16)	5,466 (291)	1
	2019	3,276 (86)	3,662 (20)	14,132 (7,687)	2

(a) Original table

REGION sort	YEAR sort	PROSECUTIONS sort	CONVICTIONS. sort	VICTIMS IDENTIFIED sort	NEW OR AMENDED LEGISLATION sort
All Africa East Asia & Pacific Europe	2014 2015 2016 2017	All 572 (245) 811 (49) 1,517 (53)	All 341 (192) 317 (33) 719 (8)	All 10,096 (2,250) 9,523 (1,308) 12,125 (3,531)	All 0 1 2
Near East	2015	480 (31)	343 (31)	6,068 (156)	0
South & Central Asia	2015	6,930 (225)	1,468 (16)	24,867 (1,191)	0
Western Hemisphere	2016	1,513 (69)	946 (24)	8,821 (109)	2
Europe	2016	2,703 (201)	1,673 (40)	13,349 (3,192)	3
Africa	2016	1,293 (54)	1,120 (21)	18,296 (13,205)	4
Near East	2016	996 (591)	1,187 (582)	3,292 (185)	4
South & Central Asia	2016	6,297 (72)	2,193 (19)	14,706 (464)	5
Africa	2015	1,517 (53)	719 (8)	12,125 (3,531)	6
Western Hemisphere	2015	1,796 (83)	663 (26)	9,661 (2,118)	6
East Asia & Pacific	2016	2,137 (51)	1,953 (31)	9,989 (310)	7
Europe	2015	4,990 (272)	1,692 (245)	11,112 (3,733)	8
East Asia & Pacific	2015	3,414 (193)	1,730 (130)	13,990 (3,533)	10

(b) Filter after sort

Figure 6: Combination of sort and filter

Prices for climbing the bridge			
	Weekday	Weekend	During Olympics
Day	AU\$117 (US\$66)	AU\$142 (US\$80)	AU\$150 (US\$85)
Night	AU\$142 (US\$80)	AU\$164 (US\$93)	AU\$170 (US\$96)

(a) Original table

Prices for climbing the bridge			
	Weekday	Weekend	During Olympics
Day	AU\$117 (US\$66)	AU\$142 (US\$80)	AU\$150 (US\$85)
Night	AU\$142 (US\$80)	AU\$164 (US\$93)	AU\$170 (US\$96)

(b) Mark up

Figure 7: (a) The original table. (b) An illustration of how the tool detects the (sub)titles and headings and marks them up with <caption> and <th>.

not uniformly distributed, it may wrongly detect it as a decorative table.

While these two problems are beyond the scope of our project, the analysis implies that better detection of human mistakes would improve the user experience. Also, detecting decorative tables would be another module that we should improve upon as the failure cases will confuse users.

## 6.2 User Evaluation

In addition to the performance evaluation, we conducted a user evaluation with 5 BLV people. We aimed at evaluating the usability of our tool and the usefulness of our tool, including to what extent our tool can promote a better understanding of data tables by making the tables more accessible and how users find each feature useful.

**6.2.1 Participants.** We recruited 5 participants from the same email lists we used in our interview study (Table 6). There were no overlapped participants between the interview study and the evaluative study. The age of the participants ranged from 20 to 35 ( $M=30$ ,  $SD=4.9$ ). While all participants were blind, their light perceptions varied. All interviews were conducted via Zoom ( $M=71$ ,  $SD=11.9$ ). Participants were compensated with a \$20 Amazon gift card.

**6.2.2 Procedure.** We used the same four stimuli used in our previous study. The interview started by asking participants demographic information, including their vision condition, age, occupation, education level. We also asked about their prior usage of assistive technologies and their prior experience with using chrome extensions.

We then asked the participants to open four URLs that we sent via email one at a time. Each page contains the original table and the tables reformatted by the tool. We put the two tables on a page to facilitate participants to interact with them and compare them. We explained the difference between the two tables and provided information on the features we applied to the table. Participants were asked to think aloud while interacting with the stimuli. We also asked them to describe the structure of the table and to portray what they've learned from the table to a friend in terms of data and insights.

After examining the four stimuli, all participants were asked post-task questions, including the accessibility and functionality of each feature and how they helped them learn more about the data. In the end, we asked participants to download our tool from the Chrome Web Store to evaluate how accessible it is to download and install a Chrome extension. We did not ask to download at the

Summary Table of Key Financial Measures (dollars in billions)

Summary Consolidated Balance Sheet Data	2020	2019	Change	% Change
Fund Balance with Treasury	\$59.6	\$61.2	\$(1.6)	(3)%
Investments, Net	20.1	19.4	0.7	4%
Property and Equipment, Net	26.3	25.6	0.7	3%
Cash, Receivables, and Other Assets	3.7	2.8	0.9	32%
<b>Total Assets</b>	<b>\$109.7</b>	<b>\$109</b>	<b>\$0.7</b>	<b>1%</b>
Accounts Payable	\$2.6	\$2.3	\$0.3	13%
After-Employment Benefit Liability	25.9	24.2	1.7	7%
International Organizations Liability	2.5	3.8	(1.3)	(34)%
Other Liabilities	2	2.1	(0.1)	(5)%
<b>Total Liabilities</b>	<b>\$33</b>	<b>\$32.4</b>	<b>\$0.6</b>	<b>2%</b>
Unexpended Appropriations	46.9	46.6	0.3	1%
Cumulative Results of Operations	29.8	30	(0.2)	(1)%
<b>Total Net Position</b>	<b>\$76.7</b>	<b>\$76.6</b>	<b>\$0.1</b>	<b>0%</b>
<b>Total Liabilities and Net Position</b>	<b>\$109.7</b>	<b>\$109.0</b>	<b>\$0.7</b>	<b>1%</b>

Summary Consolidated Statement of Net Cost Data				
	2020	2019	Change	% Change
Total Cost and Loss on Assumption Changes	\$39.4	\$38.5	\$0.9	2%
Less Total Revenue	6.8	8.8	(2.0)	(23)%
<b>Total Net Cost</b>	<b>\$32.6</b>	<b>\$29.7</b>	<b>\$2.9</b>	<b>10%</b>

Summary Combined Statement of Budgetary Resources Data				
	2020	2019	Change	% Change
Unobligated Balance from Prior Year Budget Authority, Net	\$32.7	\$31.1	\$1.6	5%
Appropriations	36.4	35.8	0.6	2%
Spending Authority from Offsetting Collections	8	8	0	0%
<b>Total Budgetary Resources</b>	<b>\$77.1</b>	<b>\$74.9</b>	<b>\$2.2</b>	<b>3%</b>

(a) Original table

Summary Table of Key Financial Measures (dollars in billions)

Summary Consolidated Balance Sheet Data	2020	2019	Change	% Change
Fund Balance with Treasury	\$59.6	\$61.2	\$(1.6)	(3)%
Investments, Net	20.1	19.4	0.7	4%
Property and Equipment, Net	26.3	25.6	0.7	3%
Cash, Receivables, and Other Assets	3.7	2.8	0.9	32%
<b>Total Assets</b>	<b>\$109.7</b>	<b>\$109.0</b>	<b>\$0.7</b>	<b>1%</b>
Accounts Payable	\$2.6	\$2.3	\$0.3	13%
After-Employment Benefit Liability	25.9	24.2	1.7	7%
International Organizations Liability	2.5	3.8	(1.3)	(34)%
Other Liabilities	2.0	2.1	(0.1)	(5)%
<b>Total Liabilities</b>	<b>\$33.0</b>	<b>\$32.4</b>	<b>\$0.6</b>	<b>2%</b>
Unexpended Appropriations	46.9	46.6	0.3	1%
Cumulative Results of Operations	29.8	30.0	(0.2)	(1)%
<b>Total Net Position</b>	<b>\$76.7</b>	<b>\$76.6</b>	<b>\$0.1</b>	<b>0%</b>
<b>Total Liabilities and Net Position</b>	<b>\$109.7</b>	<b>\$109.0</b>	<b>\$0.7</b>	<b>1%</b>
blank cell	blank cell	blank cell	blank cell	blank cell

Summary Table of Key Financial Measures (dollars in billions)-  
Summary Consolidated Statement of Net Cost Data

Summary Consolidated Balance Sheet Data	2020	2019	Change	% Change
Total Cost and Loss on Assumption Changes	\$39.4	\$38.5	\$0.9	2%
Less Total Revenue	6.8	8.8	(2.0)	(23)%
<b>Total Net Cost</b>	<b>\$32.6</b>	<b>\$29.7</b>	<b>\$2.9</b>	<b>10%</b>
blank cell	blank cell	blank cell	blank cell	blank cell

Summary Table of Key Financial Measures (dollars in billions)-  
Summary Combined Statement of Budgetary Resources Data

Summary Consolidated Balance Sheet Data	2020	2019	Change	% Change
Unobligated Balance from Prior Year Budget Authority, Net	\$32.7	\$31.1	\$1.6	5%
Appropriations	36.4	35.8	0.6	2%
Spending Authority from Offsetting Collections	8.0	8.0	0.0	0%
<b>Total Budgetary Resources</b>	<b>\$77.1</b>	<b>\$74.9</b>	<b>\$2.2</b>	<b>3%</b>

(b) Split

Figure 8: (a) The original table. (b) An illustration of how the tool detects subtitles splits the table, generates titles, and attaches headings for each subtable.

Cycle	Presidential		State Level		U.S. House	Combined
	Primary	General	Governor	U.S. Senate		
1998	—	—	85%	87%	49%	75%
1999-2000	95%	67%	82	83	53	76
2001-02	—	—	90	81	73	82
2003-04	94	78	70	82	70	78
2005-06	—	—	90	92	72	84
2007-08	79	94	95	95	82	88
2009-10	—	—	85	86	74	81
2011-12	62	82	91	87	71	77
2013-14	—	—	80	77	74	77
2015-16	85	71	68	78	78	78
2017-18	—	—	74	73	80	75
2019-20	79	80	92	72	81	79
All years	82	79	83	81	73	79

(a) Original table

Cycle	Presidential		State Level		U.S. House	Combined
	Primary	General	Governor	U.S. Senate		
1998	—	—	85%	87%	49%	75%
1999-2000	95%	67%	82	83	53	76
2001-02	—	—	90	81	73	82
2003-04	94	78	70	82	70	78
2005-06	—	—	90	92	72	84
2007-08	79	94	95	95	82	88
2009-10	—	—	85	86	74	81
2011-12	62	82	91	87	71	77
2013-14	—	—	80	77	74	77
2015-16	85	71	68	78	78	78
2017-18	—	—	74	73	80	75
All years	82	79	83	81	73	79

(b) Sort & Filter

Cycle	Presidential		State Level		U.S. House	Combined
	Primary	General	Governor	U.S. Senate		
1998	—	—	85%	87%	49%	75%
1999-2000	95%	67%	82	83	53	76
2001-02	—	—	90	81	73	82
2003-04	94	78	70	82	70	78
2005-06	—	—	90	92	72	84
2007-08	79	94	95	95	82	88
2009-10	—	—	85	86	74	81
2011-12	62	82	91	87	71	77
2013-14	—	—	80	77	74	77
2015-16	85	71	68	78	78	78
2017-18	—	—	74	73	80	75
All years	82	79	83	81	73	79

(c) Problem

Figure 9: (a) The original table. (b) An illustration of how the tool adds filter and sort function based on the lowest-level headings. (c) An illustration of how the unformatted content in the table influences the sort function.

beginning of the session because the tool will automatically alter the original tables that we wished to have in the study stimuli.

### 6.2.3 Results.

#### Usefulness of Features

**Summary & Blank Cells Indication** All participants found the summary to be practically useful. As P3 stated, “I really like the summary features a lot, because it tells me what’s coming.” P4 stated, “summary was great, I wish more tables and more articles would put up summaries like that for blind people. That’s really helpful.” Regarding questions about the usefulness of the advanced summary, two participants (P1, P3) mentioned that they would like to learn about the overview of each column before they started to navigate the table. As P3 pointed, “which columns and rows contain what information” is useful. P4 stated, “I don’t know that there’s anything redundant about the descriptions.” Three participants (P1, P2, P5) expressed that understanding the layout prior to diving into

the actual tables can help them be ready for further navigation. As P1 said, “a summary of how the table is going to be laid out is helpful”. P2 noted that it “describes how the table looks like and what it contains”. As a result, P2 stated that “by having that description before I navigate, gives me a better sense of what to do”. P5 also stated that the summary “has more descriptions” about what “I can expect into the table.” The blank cells indication helped P5 recognize the blank cell they missed before while interacting the original tables. As P5 stated, “I wouldn’t be able to understand whether it [the blank cell] even existed or not in the table.”

**Unmerged Cells & Splitting Tables** Three participants (P1, P3, P5) shared the usefulness of unmerging cells as it helps recognize the connections between columns. P1 mentioned, “I’m actually able to see all of the years and connect the different prosecutions to what year was in, and the other information.” This was an improvement over the original trafficking table T2, as the merged cells made P1



Expected benefit	Response	U.S.	U.K.	Australia	Total
Fewer crashes	Very likely	26.1	23.5	24.2	24.6
	Somewhat likely	41.7	47.6	48.1	45.8
	Somewhat unlikely	22.2		21.4	21.7
	Very unlikely	10	7.2	6.3	7.8
Less traffic congestion	Very likely	19.2	15.2		16.5
	Somewhat likely	30.5			31.6
	Somewhat unlikely	32.9	37.4	36.2	35.5
	Very unlikely	17.4			16.3

(a) Before

Expected benefit	Response	U.S.	U.K.	Australia	Total
Fewer crashes	Very likely	26.1	23.5	24.2	24.6
	Somewhat likely	41.7	47.6	48.1	45.8
	Somewhat unlikely	22.2	blank cell	21.4	21.7
	Very unlikely	10	7.2	6.3	7.8
Less traffic congestion	Very likely	19.2	15.2	15.2	16.5
	Somewhat likely	30.5	blank cell	blank cell	31.6
	Somewhat unlikely	32.9	37.4	36.2	35.5
	Very unlikely	17.4	37.4	blank cell	16.3

(b) After

**Table 4: An illustration of how the tool adds th mark-up in the header, fills out blank cells and unmerges the merged cell.**

unable to read the value for all the years. The split function also improves the efficiency of navigating tables. Two participants (P1, P3) explicitly emphasized how splitting the table could help them navigate around tables using a simple keyboard shortcut. As stated by P3 “if you wanted to jump right to a certain spot, it would be easier.” Similarly, P1 stated, “I like that each one is in a separate table, because that makes it much easier like I can just navigate to them just pressing the T button and I don’t have to worry about navigating this giant table.” The split function also improves the experience of fetching information from the screen reader. For example, P1 encountered the problem of repetition of heading, and the split function helped because “it doesn’t say the long heading over and over again, which is really annoying.”

**Filter & sort** The filter function made it easier for the participants to find information. P4 stated the filter function “would make sense if I were trying to search specific information”. P1 also noted, “you can use this list box to find just a very specific date” when reading the vaccine T1 table. Regarding the usefulness, when encountering bank statements, P1 also mentioned that “I really just want to see what happened on this day. I could use the little box and sort by just set day.” The sort function was also deemed useful. P4 said it “ would be useful in a work setting,” and P1 stated that by applying it, “you can see kind of a trend.” P5 stated the filter “allow me to only see the data that I want to see. It’s not like the massive chunk of it, which is very nice.”

#### Accessibility of Chrome Extension

Even though only three participants had experienced Chrome extension before, all participants successfully installed our tool and found the installation process accessible, as P5 stated, “highly accessible.” P3 stated “I really like chrome extensions because you just click add to chrome or remove”, and P2 said “downloading is not an issue.” Regarding the usage, P1 asserted that it should be accessible “once I learned how to do it, like what the steps are.” P2 noted, “it was very smooth and very descriptive.”

## 7 DISCUSSION

Effective communication often entails providing an overview of the contents in advance so that audiences can be prepared to receive detailed information. Especially without access to visual information, a high-level description of the table will impact listeners’ capability to grasp further information. To this end, our tool helps form a reliable representation of tables by providing an extensive (but skippable) summary so that users can be ready to navigate its details. Our tool also unpacks complex table structures into more digestible units (e.g., by splitting tables and unmerging cells) and fills out missing details (e.g., the indication of a blank cell). Our tool successfully supports various table interaction tasks (e.g., locating values of interest) by providing simple but powerful features, such as filtering and sorting. We hope our tool helps more people who rely on screen readers to read tables and carry out data-driven tasks at hand efficiently.

### 7.1 Contextualizing Study Findings with Prior Work

Prior work on improving table accessibility makes conjectures about which table elements might hinder users’ accessibility. For example, Amtmann et al. argue that blank cells would confuse blind individuals [32]. We provided empirical evidence to this argument and demonstrated a simple mechanism to indicate the blank cell can help users understand the tables.

In our formative study, we found that participants performed (or wished to perform) analytical tasks (comparing values, deriving summary statistics, and extracting trends) similar to what prior work identifies in different data reading contexts. For example, a low-level task taxonomy in information visualization by Amar et al. [31] as well as visualization tasks by Brehmer et al. include similar sets of tasks.

### 7.2 Improving Data Visualization Accessibility by Improving Table Accessibility

Our study identified various inaccessible design factors in tables online, which should inform the design of accessible visualizations as well. The data visualization research community has been vocal regarding the necessity of making visualization more accessible by providing alternative data modalities (e.g., [56, 58]). For example, High Chart [24] offers an accessibility layer on top of a visualization enabling screen readers to read out the data when users navigate to a data point. From a user’s point of view, these interactions are similar to table interactions as they hear the data points linearly. Hence, the findings of this work, such as the inadequate perception of blank cells (equivalent to missing values) or the confusion around multi-level headings (equivalent to multi-variate cases in visualizations), should inform the design of accessible visualization interaction.

In addition, tables are used to *supplement* graphs and charts, especially for BLV people. Many accessibility guidelines (e.g., [3, 4]) recommend visualization authors provide a table alongside the visualization to ensure BLV people can access data through screen readers. Recent work by Jung et al. [52] demonstrates the importance of tables in communicating visualizations by interviewing people with visual impairments. Our tool will also enhance the

Expected benefit	Response	U.S.	U.K.	Australia	Total
Fewer crashes	Very likely	26.1	23.5	24.2	24.6
	Somewhat likely	41.7	47.6	48.1	45.8
	Somewhat unlikely	22.2	21.6	21.4	21.7
	Very unlikely	10	7.2	6.3	7.8
Less traffic congestion	Very likely	19.2	15.2	15.2	16.5
	Somewhat likely	30.5	32.1	32.3	31.6
	Somewhat unlikely	32.9	37.4	36.2	35.5
	Very unlikely	17.4	15.4	16.2	16.3

Table 5: Part of an example table that we generate the summary.

Pid	Age	Gender	Edu.	Occupation	Light Perception	Onset Age (year)	Screen reader	Years Used	Has previous experience of Chrome extensions?
1	25	Female	B.S.	Unemployed	Y	15	JAWS	10	N
2	28	Male	B.S.	Solutions consultant	N	9	VoiceOver	6	Y
3	35	Female	B.S.	Unemployed	N	0	JAWS	10	Y
4	35	Male	M.S.	Unemployed	Y	0	JAWS, NVDA	10	N
5	26	Male	H.S.	Language interpreter	N	8	JAWS	10	Y

Table 6: Demographics of participants for evaluation. *Pid*=Participant ID. *Edu*=Education (H.S.=High School, B.S.=Bachelors of Science, M.S.=Masters of Science).

comprehension of visualizations by improving the readability of supplemental tables.

### 7.3 Bridging the Benefit Gap between Sighted and Visually Impaired People

Human vision can process complex information efficiently by leveraging heuristics. For example, Gestalt principles address how human vision can infer the relationship between objects by seeing their organizations [74]. Tables leverage many visual metaphors in their designs. For example, if a cell is *empty*, it is straightforward for sighted people to think that the cell does not have a value. If a cell spans multiple columns, it tells sighted people that the value applies to multiple columns. If a row is empty, sighted people see that as a separation between multiple joint tables. If the table contains multiple headings signaled with bold fonts, then sighted people can immediately associate them as a corresponding row and column. These efficient operations are the reason why tables are popular in communicating data. When these visual compositions need to be serialized using speech, however, it can cause misunderstandings. In this work, we attempt to bridge the gap in benefits that sighted people would gain from table representations and that BLV people would gain.

### 7.4 Limitations & Future Work

To collect the guidelines from online resources, we searched the web pages via Google with a limited set of keywords. Therefore, the collection of guidelines may not be a comprehensive set. We also acknowledge that the process to create the collection may not be fully reproducible since the Google search results are personalized. However, based on our observations, the vast majority of them are based on the WCAG guidelines and their tutorial examples, and most of them illustrate the uses of HTML language where there are only limited sets of tags and attributes. We believe that while the collection procedure may not be perfectly reproducible, the results

and the findings will be very close to what we present in this paper if other researchers attempt to replicate it.

The goal of this research is to enhance the accessibility of authored tables, as opposed to focusing on promoting authors to create an accessible table. While this approach has a benefit where we can make legacy tables more accessible, we would like to note that we do not advocate that accessibility should be implemented in a secondary fashion. Based on our findings of what makes data tables inaccessible, future work will explore a simple and unobtrusive authoring interface that aids authors in creating accessible tables. For example, we can envision a text editor plug-in where it dynamically detects components that make tables not accessible while authors are editing and provide warnings and auto-corrects to enhance the accessibility literacy of authors as well as the accessibility of tables.

While our performance evaluation demonstrated our tool could detect tables most of the time with the sampled collection, we may encounter ill-formatted tables that our tool can not detect when we launch the tool in the wild. As future work, we will add a logging feature that a user can report the problem. When the user encounters the tables that are not readable, they can simply click the problem report button, and then the tool will save the URL. We envision this collection of failure examples will provide insights for further enhancing the robustness and generalizability of our tool.

We performed a qualitative evaluation of our tool to demonstrate how the tool can be useful in interacting with ill-designed tables. Since we only had five participants in our study, the results may have limited generalizability. While we did not measure more objective measures such as accuracy and time, our qualitative results can inform in designing a controlled study to evaluate table accessibility in general. For example, since a few participants shared that jumping around individual tables is better than a gigantic table, we envision designing a controlled study to find an optimal threshold to split the table by measuring their time and performance. In the study, researchers can vary the number of rows and columns, as well as tasks, and the result will provide a quantified benefit and the circumstance where the benefit is maximized. Also, since a few

participants illustrated that knowing what columns to expect before navigating the tables is helpful, a controlled study can further quantify the effectiveness of this feature.

We observed that participants wished to carry out different tasks based on the granularity of the data presented in the table. Participants were more interested in statistics like min, max, and mode when interacting with summary tables (e.g., presenting aggregated numbers by category), such as the trafficking table T2 or the self-driving table T3. While interacting with tables with raw values, such as the vaccine table T1, they were more interested in trends. This observation provided us with future research questions: do people want to conduct different tasks based on the level of aggregation of data? If so, how can we detect and support those cases differently?

This work focuses on the front-end part of the pipeline by modifying the HTML tables to make them more accessible. Future work should explore redesigning screen readers based on the problems identified from the past experiences with screen readers and their testimony shared during the formative study (e.g., complex commands, repetitions of information, and unexpected pointer movement with no indication). It is worth noting that many participants were not aware that settings could be changed to avoid some of the problems. We assume that even if they know, it could be hard always to make changes to the screen readers' settings to satisfy different situations. Therefore, future work can explore how a screen reader can be more adaptable to the table reading scenario, offering the best experience when the screen reader detects a data table online.

## 8 CONCLUSION

We analyzed and consolidated existing accessibility guidelines on how to design accessible web data tables and conducted an interview study with BLV people to investigate how these guidelines align with their actual needs and how the web data tables should be improved. Based on the guidelines and the findings from our study, we designed and implemented a browser plug-in to help BLV people reformat the web data table they may encounter during their daily life. Lastly, we demonstrated the usefulness and the accessibility of the tool with an evaluation study.

## REFERENCES

- [1] 2012. Accessibility (a11y). <https://developer.chrome.com/docs/extensions/mv3/a11y/>.
- [2] 2014. Tables for Data in HTML. <https://accessibility.psu.edu/tableshhtml/>
- [3] 2018. Charts & Accessibility. <https://accessibility.psu.edu/images/charts/>.
- [4] 2020. Specific Guidelines - Graphs. <http://diagramcenter.org/specific-guidelines-e.html>.
- [5] 2021. WCAG2.1 HTML Techniques, W3C World Wide Web Consortium, Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to format tables in plain text content. <https://www.w3.org/WAI/WCAG21/Techniques/failures/F34>
- [6] 2021. WCAG2.1 HTML Techniques, W3C World Wide Web Consortium, Failure of Success Criterion 1.3.1 due to using th elements, caption elements, or non-empty summary attributes in layout tables. <https://www.w3.org/WAI/WCAG21/Techniques/failures/F46>
- [7] 2021. WCAG2.1 HTML Techniques, W3C World Wide Web Consortium, Failure of Success Criterion 1.3.1 for incorrectly associating table headers and content via the headers and id attributes. <https://www.w3.org/WAI/WCAG21/Techniques/failures/F90>
- [8] 2021. WCAG2.1 HTML Techniques, W3C World Wide Web Consortium, Failure of Success Criterion 1.3.1 for not correctly marking up table headers. <https://www.w3.org/WAI/WCAG21/Techniques/failures/F91>
- [9] 2021. WCAG2.1 HTML Techniques, W3C World Wide Web Consortium, Using caption elements to associate data table captions with data tables. <https://www.w3.org/WAI/WCAG21/Techniques/html/H39>
- [10] 2021. WCAG2.1 HTML Techniques, W3C World Wide Web Consortium, Using id and headers attributes to associate data cells with header cells in data tables. <https://www.w3.org/WAI/WCAG21/Techniques/html/H43>
- [11] 2021. WCAG2.1 HTML Techniques, W3C World Wide Web Consortium, Using table markup to present tabular information. <https://www.w3.org/WAI/WCAG21/Techniques/html/H51>
- [12] 2021. WCAG2.1 HTML Techniques, W3C World Wide Web Consortium, Using the scope attribute to associate header cells and data cells in data tables. <https://www.w3.org/WAI/WCAG21/Techniques/html/H63>
- [13] 2021. WCAG2.1 HTML Techniques, W3C World Wide Web Consortium, Using the summary attribute of the table element to give an overview of data tables. <https://www.w3.org/WAI/WCAG21/Techniques/html/H73>
- [14] 2021. Web Content Accessibility Guidelines 2.1, W3C World Wide Web Consortium Recommendation Updated 27 July 2021, 1.3.1: Info and Relationships. <https://www.w3.org/WAI/WCAG21/Understanding/info-and-relationships.html>
- [15] 2021-08-09. How can data tables be made accessible? | DO-IT. <https://www.washington.edu/doi/how-can-data-tables-be-made-accessible>
- [16] 2021-08-09. HTML table advanced features and accessibility - Learn web development | MDN. <https://developer.mozilla.org/en-US/docs/Learn/HTML/Tables/Advanced>
- [17] 2021-08-09. Tables | Usability & Web Accessibility. <https://usability.yale.edu/web-accessibility/articles/tables>
- [18] 2021-08-09. Tables | Web Accessibility Checklist. <https://dequeuniversity.com/checklists/web/tables>
- [19] 2021-08-09. Tables Concepts • Tables • WAI Web Accessibility Tutorials. <https://www.w3.org/WAI/tutorials/tables/>
- [20] 2021-08-09. WebAIM: Creating Accessible Tables - Data Tables. <https://webaim.org/techniques/tables/data>
- [21] 2021-09-01. Accessibility Insights. <https://accessibilityinsights.io/>.
- [22] 2021-09-01. Chartability. <https://chartability.fizz.studio/>.
- [23] 2021-09-01. Deque Systems: Web Accessibility Software, Services & Training. <https://www.deque.com/>.
- [24] 2021-09-01. Highcharts - Interactive javascripts charts library. <https://www.highcharts.com/>.
- [25] 2021-09-01. Siteimprove Digital Marketing Cloud. <https://siteimprove.com/en-us/>.
- [26] 2021-12-22. Digital accessibility Tables. <https://accessibility.princeton.edu/how/content/tables>
- [27] Iyad Abu Doush, Enrico Pontelli, Dominic Simon, Tran Cao Son, and Ou Ma. 2009. Making Microsoft Excel™ accessible: Multimodal presentation of charts. *ASSETS'09 - Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility* (2009), 147–154. <https://doi.org/10.1145/1639642.1639669>
- [28] Byron Acuña, Luiz César Martini, Lucas L. Motta, Julio Larco, and Felipe Grijalva. 2019. Table Detection for Improving Accessibility of Digital Documents Using a Deep Learning Approach. In *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*. 1–6. <https://doi.org/10.1109/LA-CCI47412.2019.9036767>
- [29] Roobaea Alroobaea and Pam J Mayhew. 2014. How many participants are really enough for usability studies?. In *2014 Science and Information Conference*. IEEE, 48–56.
- [30] Abdullah Alsaedi. 2020. Comparing web accessibility evaluation tools and evaluating the accessibility of webpages: proposed frameworks. *Information* 11, 1 (2020), 40.
- [31] Robert Amar, James Eagan, and John Stasko. 2005. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. IEEE, 111–117.
- [32] Dagmar Amtmann, Kurt Johnson, and Debbie Cook. 2002. Making Web-based Tables Accessible for Users of Screen Readers. *Library Hi Tech* 20, 2 (Jan. 2002), 221–231. <https://doi.org/10.1108/07378830210432589>
- [33] Chieko Asakawa and Takashi Itoh. [n.d.]. User Interface of a Nonvisual Table Navigation Method. ([n. d.]), 2.
- [34] Michelle A Borkin, Azalea A Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What makes a visualization memorable? *IEEE transactions on visualization and computer graphics* 19, 12 (2013), 2306–2315.
- [35] Roopa Bose and Helmut Jürgensen. 2014. Accessibility of e-commerce websites for vision-impaired persons. In *International Conference on Computers for Handicapped Persons*. Springer, 121–128.
- [36] Kowndinya Boyalakuntla, Akhila Sri Manasa Venigalla, and Sridhar Chimalakonda. 2021. WAccess-A Web Accessibility Tool based on the latest WCAG 2.2 guidelines. *arXiv preprint arXiv:2107.06799* (2021).
- [37] Giorgio Brajnik. 2004. Comparing accessibility evaluation tools: a method for tool effectiveness. *Universal access in the information society* 3, 3 (2004), 252–263.
- [38] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

- [39] Marina Buzzi and Barbara Leporini. 2009. Editing Wikipedia content by screen reader: easier interaction with the Accessible Rich Internet Applications suite. *Disability and Rehabilitation: Assistive Technology* 4, 4 (2009), 264–275.
- [40] Lucas Pedrosa Carvalho, Lucas Pereira Ferreira, and André Pimenta Freire. 2016. Accessibility evaluation of rich internet applications interface components for mobile screen readers. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. 181–186.
- [41] Hsin-Hsi Chen, Shih-Chung Tsai, and Jin-He Tsai. 2000. Mining tables from large scale html texts. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*.
- [42] Zhe Chen and Michael Cafarella. 2013. Automatic web spreadsheet data extraction. In *Proceedings of the 3rd International Workshop on Semantic Search over the Web*. 1–8.
- [43] Mehmet Chiosemoglou and Helmut Jürgensen. [n.d.]. Setting the Table for the Blind. ([n. d.]), 8.
- [44] Chetz Colwell and Helen Petrie. [n.d.]. Evaluation of Guidelines for Designing Accessible Web Content. ([n. d.]), 3.
- [45] James Craig, Michael Cooper, L Pappas, R Schwerdtfeger, and I Seeman. 2009. Accessible rich internet applications (WAI-ARIA) 1.0. *W3C Working Draft* (2009).
- [46] Seniz Demir, David Oliver, Edward Schwartz, Stephanie Elzer, Sandra Carberry, Kathleen F. McCoy, and Daniel Chester. 2010. Interactive SIGHT: Textual access to simple bar charts. *New Review of Hypermedia and Multimedia* 16, 3 (2010), 245–279. <https://doi.org/10.1080/13614568.2010.534186>
- [47] Iyad Doush and Enrico Pontelli. 2010. Non-Visual Navigation of Tables in Spreadsheets. (Jan. 2010).
- [48] Iyad Abu Doush, Enrico Pontelli, Tran Cao Son, and Dominic A Simon. 2010. Multimodal presentation of two-dimensional charts: An investigation using open office XML and microsoft excel. November (2010). <https://doi.org/10.1145/1857920.1857925>
- [49] Steven Gardiner, Anthony Tomasic, and John Zimmerman. [n.d.]. EnTable: Rewriting Web Data Sets as Accessible Tables. ([n. d.]), 2.
- [50] Waqar Haider and Yeliz Yesilada. 2020. Tables on the web accessible? unfortunately not!. In *Proceedings of the 17th International Web for All Conference*. 1–5.
- [51] Jane Hoffswell and Zhicheng Liu. 2019. Interactive repair of tables extracted from pdf documents on mobile devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [52] Crescentia Jung, Shubham Mehta, Atharva Kulkarni, Yuhang Zhao, and Ye-Seul Kim. 2021. Communicating Visualizations without Visuals: Investigation of Visualization Alternative Text for People with Visual Impairments. *IEEE transactions on visualization and computer graphics* (2021).
- [53] Rushil Khurana, Duncan McIsaac, Elliot Lockerman, and Jennifer Mankoff. 2018. Nonvisual interaction techniques at the keyboard surface. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [54] Johan Kildal and Stephen A Brewster. 2006. Non-Visual Overviews of Complex Data Sets. (2006), 6.
- [55] Johan Kildal and Stephen A Brewster. 2006. Providing a size-independent overview of non-visual tables. In *12th International Conference on Auditory Display (ICAD2006)*. 8–15.
- [56] NW Kim, SC Joyner, A Riegelhuth, and Y Kim. 2021. Accessible Visualization: Design Space, Opportunities, and Challenges. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 173–188.
- [57] Pauli PY Lai. 2013. Adapting data table to improve web accessibility. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. 1–4.
- [58] Bongshin Lee, Eun Kyoung Choe, Petra Isenberg, Kim Marriott, and John Stasko. 2020. Reaching Broader Audiences With Data Visualization. *IEEE Computer Graphics and Applications* 40, 2 (2020), 82–90.
- [59] Seung-Jin Lim and Yiu-Kai Ng. 1999. An automated approach for retrieving hierarchical data from HTML tables. In *Proceedings of the eighth international conference on Information and knowledge management*. 466–474.
- [60] Carlos Máñez-Carvajal, Jose Francisco Cervera-Mérida, and Rocio Fernández-Piqueras. 2021. Web accessibility evaluation of top-ranking university Web sites in Spain, Chile and Mexico. *Universal Access in the Information Society* 20, 1 (2021), 179–184.
- [61] Toshiya Oogane and Chieko Asakawa. 1998. An interactive method for accessing tables in HTML. In *Proceedings of the third international ACM conference on Assistive technologies*. 126–128.
- [62] E Pontelli, D Gillan, G Gupta, A Karshmer, E Saad, and W Xiong. [n.d.]. Intelligent Non-Visual Navigation of Complex HTML Structures. ([n. d.]), 14.
- [63] Enrico Pontelli, D Gillan, W Xiong, Emad Saad, Gopal Gupta, and Arthur I Karshmer. 2002. Navigation of html tables, frames, and xml fragments. In *Proceedings of the fifth international ACM conference on Assistive technologies*. 25–32.
- [64] Enrico Pontelli and Tran Cao Son. 2002. Planning, reasoning, and agents for non-visual navigation of tables and frames. In *Proceedings of the fifth international ACM conference on Assistive technologies*. 73–80.
- [65] Enrico Pontelli, W Xiong, Gopal Gupta, and Arthur I Karshmer. 2000. A domain specific language framework for non-visual browsing of complex html structures. In *Proceedings of the fourth international ACM conference on Assistive Technologies*. 180–187.
- [66] Venkatesh Potluri, Tadashi E Grindeland, Jon E. Froehlich, and Jennifer Mankoff. 2021. Examining Visual Semantic Understanding in Blind and Low-Vision Technology Users. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Number 35. Association for Computing Machinery, New York, NY, USA, 1–14.
- [67] Rameshsharma Ramlooll, Stephen Brewster, Wai Yu, and Beate Riedel. 2001. Using Non-speech Sounds to Improve Access to 2D Tabular Numerical Information for Visually Impaired Users. In *People and Computers XV—Interaction without Frontiers*, Ann Blandford, Jean Vanderdonck, and Phil Gray (Eds.). Springer London, London, 515–529. [https://doi.org/10.1007/978-1-4471-0353-0\\_32](https://doi.org/10.1007/978-1-4471-0353-0_32)
- [68] Katerine Romeo, E Pissaloux, and Frédéric Serin. 2019. Accessible tables in digital documents. *arXiv preprint arXiv:1911.03287* (2019).
- [69] Volker Sorge, Akashdeep Bansal, Neha M Jadhav, Himanshu Garg, Ayushi Verma, and M Balakrishnan. 2020. Towards Generating Web-Accessible STEM Documents from PDF. In *Proceedings of the 17th International Web for All Conference (W4A '20)*. Association for Computing Machinery, New York, NY, USA, 1–5. <https://doi.org/10.1145/3371300.3383351>
- [70] Dimitris Spiliotopoulos, Panagiota Stavropoulou, and Georgios Kouroupetroglou. 2009. Acoustic rendering of data tables using earcons and prosody for document accessibility. In *International Conference on Universal Access in Human-Computer Interaction*. Springer, 587–596.
- [71] Dimitris Spiliotopoulos, Gerasimos Xydias, and Georgios Kouroupetroglou. 2005. Diction based prosody modeling in table-to-speech synthesis. In *International Conference on Text, Speech and Dialogue*. Springer, 294–301.
- [72] Dimitris Spiliotopoulos, Gerasimos Xydias, Georgios Kouroupetroglou, and Vasilios Argyropoulos. 2005. Experimentation on spoken format of tables in auditory user interfaces. In *Proceedings of the 11th International Conference on Human-Computer Interaction (HCI'05)*. 361–370.
- [73] Dimitris Spiliotopoulos, Gerasimos Xydias, Georgios Kouroupetroglou, Vasilios Argyropoulos, and Kalliopi Ikospentaki. 2010. Auditory universal accessibility of data tables using naturally derived prosody specification. *Universal Access in the Information Society* 9, 2 (2010), 169–183.
- [74] Dejan Todorovic. 2008. Gestalt principles. *Scholarpedia* 3, 12 (2008), 5345.
- [75] Markel Vigo, Justin Brown, and Vivienne Conway. 2013. Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated tests. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. 1–10.
- [76] Yalin Wang and Jianying Hu. 2002. A machine learning based approach for table detection on the web. In *Proceedings of the 11th international conference on World Wide Web*. 242–250.
- [77] Kristin Williams, Taylor Clarke, Steve Gardiner, John Zimmerman, and Anthony Tomasic. [n.d.]. Find and Seek: Assessing the Impact of Table Navigation on Information Look-up with a Screen Reader. *ACM Transactions on Accessible Computing* 12, 3 ([n. d.]), 23.
- [78] Yudong Yang, Yu Chen, and HongJiang Zhang. 2003. HTML page analysis based on visual cues. In *Web Document Analysis: Challenges and Opportunities*. World Scientific, 113–131.
- [79] Yeliz Yesilada, Robert Stevens, Carole Goble, and Shazad Hussein. 2003. Rendering tables in audio: the interaction of structure and reading styles. *ACM SIGACCESS Accessibility and Computing* 77-78 (2003), 16–23.