Accessible Text Descriptions for UpSet Plots

Ishrat Jahan Eliza* University of Utah
Jake Wagoner University of Utah
Jack Wilburn University of Utah
Nate Lanza University of Utah
Daniel Hajas University College London
Alexander Lex † University of Utah

2024 1st Workshop on Accessible Data Visualization (AccessViz) | 979-8-3315-1731-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/AccessViz64636.2024.00005

Abstract

Data visualizations are typically not accessible to blind and low-vision users. The most widely used remedy for making data visualizations accessible is text descriptions. Yet, manually creating useful text descriptions is often omitted by visualization authors, either because of a lack of awareness or a perceived burden. Automatically generated text descriptions are a potential partial remedy. However, with current methods it is unfeasible to create text descriptions for complex scientific charts. In this paper, we describe our methods for generating text descriptions for one complex scientific visualization: the UpSet plot. UpSet is a widely used technique for the visualization and analysis of sets and their intersections. At the same time, UpSet is arguably unfamiliar to novices and used mostly in scientific contexts. Generating text descriptions for UpSet plots is challenging because the patterns observed in UpSet plots have not been studied. We first analyze patterns present in dozens of published UpSet plots. We then introduce software that generates text descriptions for UpSet plots based on the patterns present in the chart. Finally, we introduce a web service that generates text descriptions based on a specification of an UpSet plot, and demonstrate its use in both an interactive web-based implementation and a static Python implementation of UpSet.

Index Terms: Accessibility, Data Visualization, Text Descriptions

1 Introduction

Being able to read and understand data visualizations is essential for both personal and professional reasons. Consumers of visualizations may encounter them in newspapers, in government dashboards, in work reports, or in scientific articles. However, most data visualizations we encounter are not accessible to people that have visual impairments, such as users with reduced or residual vision, or even blind users [2, 26].

Accessibility has been an area of increased interest in the visualization research community in recent years. Various assistive tools have been developed to aid visually impaired users [28, 5, 9]. A common remedy to make images of all kinds accessible is to provide text descriptions [13] which are easily readable by most modern screen readers [28]. A study by Lundgard and Satyanarayan has shown that rich text descriptions can significantly enhance the understanding of data to sighted and visually impaired users [17]. At the same time, most charts in scientific journals do not have any text descriptions at all [18].

Clearly, there are organizational and social challenges that cause this lack of text descriptions; yet automatically generating text descriptions has been shown to be a partial—if imperfect—remedy [27]. Our work aims to help address these shortcomings by generating effective text descriptions for a complex visualization, the UpSet Plot [16]. Widely used plots, such as scatter plots, exhibit well-understood patterns, such as clusters or outliers, that algorithms can identify to extract information and use that information to generate meaningful text descriptions. For complex plots, like the UpSet plot, trends are less clear to novice users, and their patterns may not be well understood yet.

Given the wide adoption of UpSet plots in academic publications, especially in the biomedical domain, we have chosen UpSet as a representative example of a complex chart to generate text descriptions.

To do this, we first identify common patterns in UpSet plots, based on a survey of published UpSet figures. We then introduce a grammar that can be used to write specifications that describe UpSet plots, and use this specification as a platform-independent, machine-readable representation of UpSet plots. To generate the text descriptions, we developed a Python web service that ingests the data and the specification of an UpSet plot. Our implementation is successfully deployed with our interactive web-based UpSet implementation and is also compatible with a static Python implementation of UpSet.

2 UpSet Plots

UpSet, shown in Figure 1, is a set visualization technique suited for data that is commonly shown in Venn diagrams. Unlike Venn diagrams, UpSet works with more than three sets by plotting the set intersections as a matrix. Each column corresponds to a set, and bar charts on top show the size of the set. Each row corresponds to an intersection: the filled-in cells show which set is part of an intersection. The size of the intersections are displayed as bar charts to the right of the matrix. The matrix can be sorted in various ways to surface trends; a common way is to sort by size of the intersections. As a more concrete example, the UpSet plot in Figure 1 shows movie data. Movies have genres (sets) like Drama, Comedy, or Action. A movie can have a single genre, or it can have multiple genres. Movies that span multiple genres are counted in the set intersections.

3 Related Work

We review relevant prior work on text descriptions and accessibility in visualization in general. Accessibility has been considered an important element in terms of visualization in recent years [15, 17, 22], and different researchers have created different accessibility tools for visualization for blind or low-vision users [5, 10, 1, 4].

Studies have suggested that text summaries could significantly aid users in understanding complex graphical data, comparing favorably to sighted individuals' comprehension levels [20, 14, 17]. However, the usage of text descriptions (also called alternative text, or alt-text) in scientific publications is extremely limited. A study by L'Yi et al. [18] reveals that people with visual impairments use screen readers as the most common assistive technology, but that data portals and journal websites do not have proper text descriptions associated with visualizations. On the other hand, text descriptions are much more common in generic images on the web [1] and on social media platforms [27, 12], partially aided by the automatic generation of text descriptions on platforms such as Facebook [27].

*e-mail: ishratjahan.eliza@utah.edu
†e-mail: alex@sci.utah.edu

979-8-3315-1731-1/24/$31.00 ©2024 IEEE
DOI 10.1109/AccessViz64636.2024.00005

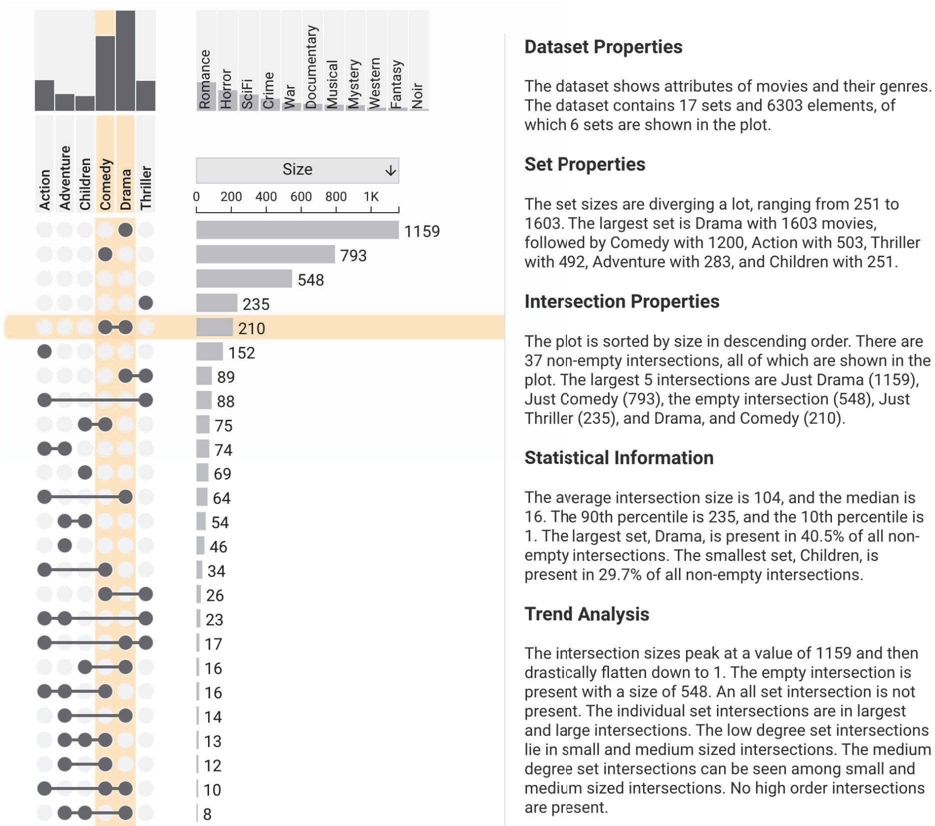

Figure 1: An UpSet plot (left) and the generated text description for the plot (right). The data shown are movies and their genres. The biggest intersection is between Comedy and Drama (highlighted) with 210 movies. The text description shows the long version we generate, with section covering both basic information and higher-level trends.

Studies have shown that figures in scientific publications contain important information and results, and descriptions with proper context are needed for blind and low-vision readers to engage with them [3]. For graphics that contain rich information, such as visualizations, lack of, or poorly written alternative texts can worsen the information access inequality for people with visual impairments [13]. These limitations are major hurdles for people with visual impairments entering science careers [18].

Researchers have been studying the quality of accessibility via text descriptions for years [7]. The text descriptions found in practice often provide very little information [13, 19, 21], and users are unable to dig deeper into the data behind the visualization. Most figures in scientific articles include captions; however, captions are usually not written to summarize the content of a figure, and hence are typically not effective enough to provide insights into the data to visually impaired readers [8].

While attempts have been made to use AI to automatically generate natural language summaries for charts [23] and to create figure captions [24, 25], these summaries are limited to common chart types, a limitation we address in this paper.

4 Patterns in UpSet Plots

To generate text descriptions, we first have to identify general patterns of UpSet plots, or set data in general. To do this, we surveyed published UpSet plots, analyzed the plots by tagging them, and categorizing the patterns we identified.

4.1 Collecting UpSet Plots

There are over 4000 papers citing the two publications that describe UpSet [16, 6]. Most of these papers cite UpSet because they are using an UpSet plot. Since we couldn't reasonably analyze all of these plots, we drew a convenience sample, as described below, and conducted our coding and codebook development as we kept adding plots. We stopped adding papers when we couldn't identify any new patterns (i.e., reached saturation of our codes). In total, we collected 79 UpSet plots from 40 published papers. First, we searched for papers citing the original UpSet paper [16] in Google Scholar, resulting in 30 papers. We then searched Google for the keyword "UpSet Plot" and found another 8 papers that we included in our sample. We further collected plots from two papers cited in our corpus. For coding, we entered all plots into an online tabular tagging framework (Airtable). We also saved the metadata of the plots, such as bibliographic details, the caption of the plots, the paragraph that referred to the UpSet plots in the paper, the scientific fields the paper was published in, tools that were used to generate the plots, the raw data of the plot (if provided), and the presence of alternative text attached to the plots. Figure 2 shows an overview of our collection.

4.2 Coding Patterns in UpSet Plots

To generate a textual description that can provide insights into the data, we first need to understand the types of patterns that are exhibited in Upset plots. To this end, we coded the patterns we observed in our dataset. Two of the authors took part in the coding process. First, one of them coded each plot individually based on the presence of the largest set in the largest intersection, the smallest sets in the smallest intersection (and vice versa), how the set sizes diverge, how the intersection sizes diverge, the presence of all set intersections, etc. Then, the other author revised the tags and gave an opinion on the tags, what we could include, or what we could remove from the tags. After two iterations, the coding team reached a consensus and finalized the tags.

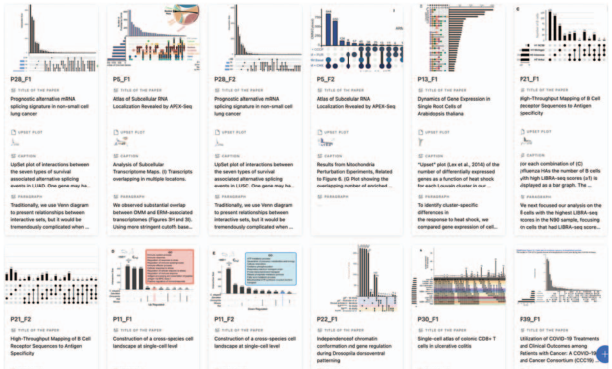

Figure 2: Selected thumbnails of UpSet plots collected for our classification of patterns found in UpSet plots.

4.3 Categorizing the Patterns

To categorize the patterns found in UpSet plots, we use the code defined in the previous step. We categorized the collected plots according to the set number in the intersections, regions based on the intersection size, and general trends falling into those regions, the differences among the set sizes as our representative plots.

UpSet plots are most useful for visualizing and analyzing data with more than three sets and their intersections. The sets can be of potentially different sizes, and the intersections can diverge in different patterns based on the underlying characteristics of a dataset. For example, when analyzing shared genes between species (a common use case of UpSet), we expect a lot of high-order overlaps (many shared genes between many or even all species). In contrast, when analyzing sparse datasets, such as movie genres associated with movies (movies typically have 1-3 genres), we expect little high-order overlap. Another important factor is the relative size of set. For example, there are many more “drama” than “adventure” movies, so an analysis has to take this divergence into account.

For some UpSet plots, the presence of intersections with no set, or with intersections that include all sets, is important information. Thus, we also categorize them as specific attributes. We ended up with the overarching categories General Trends, General Trend Sizes, Set Sizes, and Specific Artifacts/Attributes.

The categories are:

- **Intersection Patterns:** Are there many independent sets (intersections containing only 1 set). Are there many low-set intersections (intersections containing 2-3 sets), medium-degree set intersections (intersections containing 3-n/2 sets), or high-order set intersections (intersections containing n/2 to n sets).
- **Intersection Size Patterns**: what are the intersection sizes associated with the intersection patterns? Are the higher-order intersections (involving most sets) large or small? Are the low-set intersections large or small? We classify these into small, medium, large, and largest.
- **Specific Attributes:** Is the all-set intersection present? Is the empty intersection present? Are the all-set and empty intersections the largest, among the largest, or small?
- **Set Sizes:** Are set sizes roughly equal, diverging a lot, or moderately diverging.

5 Text Description Generation

We generate our text description for UpSet plots using a custom text generation Python package published on PyPI that reads in a JSON grammar describing the UpSet plots. There are dozens of different software libraries implementing UpSet plots, and the grammar allows us to capture the relevant patterns independent of a specific implementation and programming language, thereby allowing us to make our text description generation compatible with various implementations. We implemented the ability to export a specification based on the grammar for two UpSet libraries: our interactive web-based implementation—UpSet 2.0[11], and a popular Python implementation of UpSet, UpSetPlot.

This specification is then passed into the text generation package, where we parse the grammar to extract information about data. We perform a statistical and trend analysis based on our categorization of patterns and the parsed data, resulting in a list of salient trends for a particular upset plot. We used an iterative feedback process with our blind collaborator and co-author to develop text descriptions at two levels of details that are succinct yet fully descriptive of the dataset, so that the text confers a similar amount of information as the equivalent chart.

We generate text based on a template that has programatically replaceable tokens in it. These tokens are linked to functions that use the parsed and analyzed data from our grammar model to compute strings. We then iterate over the grammar model and replace all tokens with their computed string values. These strings are returned as human-readable text descriptions.

We have exposed this text-generating Python package to UpSet 2.0 and UpSetPlot through a Python-backed web API, demonstrating that such a text-generation function could be used for a variety of implementations of the same visualization technique. The API returns text in JSON format that includes a technical description, a 2-3 sentence short description, and a long description. The *Technical Description* is an introductory sentence about the UpSet plot; the *Short Description* is a brief snippet that contains the most salient trend and that can be used consistent with browser’s ‘alt’ properties; the *Long Description* is a comprehensive description designed to communicate similar amounts of information to the chart.

Figure 1 shows the resulting long description next to a web-based UpSet implementation. The long description includes Introduction, Dataset Properties, Set Properties, Intersection Properties, Statistical Information, and Trend Analysis. The first four sections have basic information about the type of the plot, how many sets are involved, how many intersections are seen, the count of visible sets, the title of the sets, sorting order, largest intersection name, largest five to ten intersections, etc. For the statistical information section, we compute metrics such as percentile information, the percentage of the presence of the largest set and smallest set in all the intersections, the average value, and the median divergence of the intersections, etc. Finally, the trend analysis section describes trends that we determine to be present based on patterns and categories discussed in the previous section.

6 Future Work

One immediate avenue for future work is evaluating our approach. We are planning a two-pronged approach: a crowdsourced study with sighted participants and an interview study with low-vision and blind users. In the former, we will evaluate learning and understanding in UpSet plots under three different conditions: visualization only, text only, and combined visualization and text.

Our interview study will focus on the learnability of UpSet plots based on text descriptions alone, and the quality and understandability of our text descriptions.

Beyond this immediate project, another avenue for research is to apply our methods for generating text descriptions for other complex scientific charts. Finally, it will be intriguing to develop a corpus of UpSet (and other) plots, associated grammars, and vetted text descriptions to use as a training dataset for LLMs.

7 Acknowledgements

We gratefully acknowledge Kiran Gadhave, and funding by the Chan Zuckerberg Initiative through the EOSS5 program.

References

[1] J. P. Bigham, R. S. Kaminsky, R. E. Ladner, O. M. Danielsson, and G. L. Hempton. WebInSight: Making web images accessible. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '06, pp. 181–188. Association for Computing Machinery, New York, NY, USA, Oct. 2006. doi: 10.1145/1168987.1169018 1

[2] M. Butler, L. Holloway, K. Marriott, and C. Goncu. Understanding the graphical challenges faced by vision-impaired students in Australian universities. *Higher Education Research & Development*, 36(1):59–72, Jan. 2017. doi: 10.1080/07294360.2016.1177001 1

[3] S. S. Chintalapati, J. Bragg, and L. L. Wang. A Dataset of Alt Texts from HCI Publications: Analyses and Uses Towards Producing More Descriptive Alt Texts of Data Visualizations in Scientific Papers. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '22, pp. 1–12. Association for Computing Machinery, New York, NY, USA, Oct. 2022. doi: 10.1145/3517428.3544796 2

[4] J. Choi, S. Jung, D. G. Park, J. Choo, and N. Elmqvist. Visualizing for the Non-Visual: Enabling the Visually Impaired to Use Visualization. *Computer Graphics Forum*, 38(3):249–260, 2019. doi: 10.1111/cgf.13686 1

[5] P. Chundury, B. Patnaik, Y. Reyazuddin, C. Tang, J. Lazar, and N. Elmqvist. Towards Understanding Sensory Substitution for Accessible Visualization: An Interview Study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1084–1094, Jan. 2022. doi: 10.1109/TVCG.2021.3114829 1

[6] J. R. Conway, A. Lex, and N. Gehlenborg. UpSetR: An R package for the visualization of intersecting sets and their properties. *Bioinformatics*, 33(18):2938–2940, Sept. 2017. doi: 10.1093/bioinformatics/btx364 2

[7] F. Elavsky, C. Bennett, and D. Moritz. How accessible is my visualization? Evaluating visualization accessibility with Chartability. *Computer Graphics Forum*, 41(3):57–70, 2022. doi: 10.1111/cgf.14522 2

[8] S. Elzer, S. Carberry, D. Chester, S. Demir, N. Green, I. Zukerman, and K. Trnka. Exploring and Exploiting the Limited Utility of Captions in Recognizing Intention in Information Graphics. In K. Knight, H. T. Ng, and K. Oflazer, eds., *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 223–230. Association for Computational Linguistics, Ann Arbor, Michigan, June 2005. doi: 10.3115/1219840.1219868 2

[9] C. Engel and G. Weber. A User Study to Evaluate Tactile Charts with Blind and Visually Impaired People. In K. Miesenberger and G. Kouroupetroglou, eds., *Computers Helping People with Special Needs*, Lecture Notes in Computer Science, pp. 177–184. Springer International Publishing, Cham, 2018. doi: 10.1007/978-3-319-94274-2_24 1

[10] H. A. Faucett, K. E. Ringland, A. L. L. Cullen, and G. R. Hayes. (In)Visibility in Disability and Assistive Technology. *ACM Transactions on Accessible Computing*, 10(4):14:1–14:17, Oct. 2017. doi: 10.1145/3132040 1

[11] K. Gadhave, H. Strobelt, N. Gehlenborg, and A. Lex. UpSet 2: From Prototype to Tool. In *Proceedings of the IEEE Information Visualization Conference – Posters (InfoVis '19)*, 2019. 3

[12] C. Gleason, A. Pavel, E. McCamey, C. Low, P. Carrington, K. M. Kitani, and J. P. Bigham. Twitter A11y: A Browser Extension to Make Twitter Images Accessible. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pp. 1–12. Association for Computing Machinery, New York, NY, USA, Apr. 2020. doi: 10.1145/3313831.3376728 1

[13] C. Jung, S. Mehta, A. Kulkarni, Y. Zhao, and Y.-S. Kim. Communicating Visualizations without Visuals: Investigation of Visualization Alternative Text for People with Visual Impairments. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1095–1105, Jan. 2022. doi: 10.1109/TVCG.2021.3114846 1, 2

[14] G. Kim, J. Kim, and Y.-S. Kim. "Explain What a Treemap is": Exploratory Investigation of Strategies for Explaining Unfamiliar Chart to Blind and Low Vision Users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pp. 1–13. Association for Computing Machinery, New York, NY, USA, Apr. 2023. doi: 10.1145/3544548.3581139 1

[15] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim. Accessible visualization: Design space, opportunities, and challenges. *Computer Graphics Forum*, 40(3):173–188, 2021. doi: 10.1111/cgf.14298 1

[16] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. UpSet: Visualization of Intersecting Sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1983–1992, Dec. 2014. doi: 10.1109/TVCG.2014.2346248 1, 2

[17] A. Lundgard and A. Satyanarayan. Accessible Visualization via Natural Language Descriptions: A Four-Level Model of Semantic Content. *IEEE Transactions on Visualization and Computer Graphics*, 28(1), Jan. 2022. doi: 10.1109/TVCG.2021.3114770 1

[18] S. L'Yi, T. C. Smits, A. Lex, and N. Gehlenborg. Digital Accessibility of Life Sciences Data Portals and Journal Websites. *OSF.io*, Feb. 2024. doi: 10.31219/osf.io/5v98j 1, 2

[19] K. Mack, E. Cutrell, B. Lee, and M. R. Morris. Designing Tools for High-Quality Alt Text Authoring. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '21, pp. 1–14. Association for Computing Machinery, New York, NY, USA, Oct. 2021. doi: 10.1145/3441852.3471207 2

[20] P. Moraes, G. Sina, K. McCoy, and S. Carberry. Evaluating the accessibility of line graphs through textual summaries for visually impaired users. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility*, ASSETS '14, pp. 83–90. Association for Computing Machinery, New York, NY, USA, Oct. 2014. doi: 10.1145/2661334.2661368 1

[21] M. R. Morris, J. Johnson, C. L. Bennett, and E. Cutrell. Rich Representations of Visual Content for Screen Reader Users. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 1–11. Association for Computing Machinery, New York, NY, USA, Apr. 2018. doi: 10.1145/3173574.3173633 2

[22] A. Nazemi and I. Murray. A Method to Provide Accessibility for Visual Components to Vision Impaired. *International Journal of Human Computer Interaction*, 4(1):54–69, 2013. 1

[23] J. Obeid and E. Hoque. Chart-to-Text: Generating Natural Language Descriptions for Charts by Adapting the Transformer Model. In B. Davis, Y. Graham, J. Kelleher, and Y. Sripada, eds., *Proceedings of the 13th International Conference on Natural Language Generation*, pp. 138–147. Association for Computational Linguistics, Dublin, Ireland, Dec. 2020. doi: 10.18653/v1/2020.inlg-1.20 2

[24] X. Qian, E. Koh, F. Du, S. Kim, and J. Chan. A Formative Study on Designing Accurate and Natural Figure Captioning Systems. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, pp. 1–8. Association for Computing Machinery, New York, NY, USA, Apr. 2020. doi: 10.1145/3334480.3382946 2

[25] X. Qian, E. Koh, F. Du, S. Kim, J. Chan, R. A. Rossi, S. Malik, and T. Y. Lee. Generating Accurate Caption Units for Figure Captioning. In *Proceedings of the Web Conference 2021*, WWW '21, pp. 2792–2804. Association for Computing Machinery, New York, NY, USA, June 2021. doi: 10.1145/3442381.3449923 2

[26] K. C. Shahira and A. Lijiya. Towards Assisting the Visually Impaired: A Review on Techniques for Decoding the Visual Data From Chart Images. *IEEE Access*, 9:52926–52943, 2021. doi: 10.1109/ACCESS.2021.3069205 1

[27] S. Wu, J. Wieland, O. Farivar, and J. Schiller. Automatic Alt-text: Computer-generated Image Descriptions for Blind Users on a Social Network Service. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pp. 1180–1192. ACM, Portland Oregon USA, Feb. 2017. doi: 10.1145/2998181.2998364 1

[28] J. Zong, C. Lee, A. Lundgard, J. Jang, D. Hajas, and A. Satyanarayan. Rich Screen Reader Experiences for Accessible Data Visualization. *Computer Graphics Forum*, 41(3):15–27, 2022. doi: 10.1111/cgf.14519 1