RESEARCH STATEMENT | ANNE SPENCER ROSS

Mobile applications (apps) provide essential services that need to be accessible to people with disabilities (e.g., banking, transportation). However, many apps fail in this responsibility [2,3,5,6,8] **I use large-scale and multi-factor approaches to characterize and improve app accessibility for people with disabilities**. A selection of my most significant work can be found at the end of this statement.

I created an analytical framework to structure my approach to app accessibility [4]. Using a large-scale, automated analysis of 10,000 apps, I characterized the state of app accessibility (e.g., how often accessibility barriers occur) and identified potential causes of such barriers [5,6,7]. Informed by my data-driven characterization and human-centered research methods, I am creating tools for developers and testers to improve app accessibility. My future work will focus on longitudinal app accessibility analyses, improving tools for app design, development, and testing, and exploring how company ecosystems impact accessible app creation.

Epidemiology-Inspired Analytical Framework

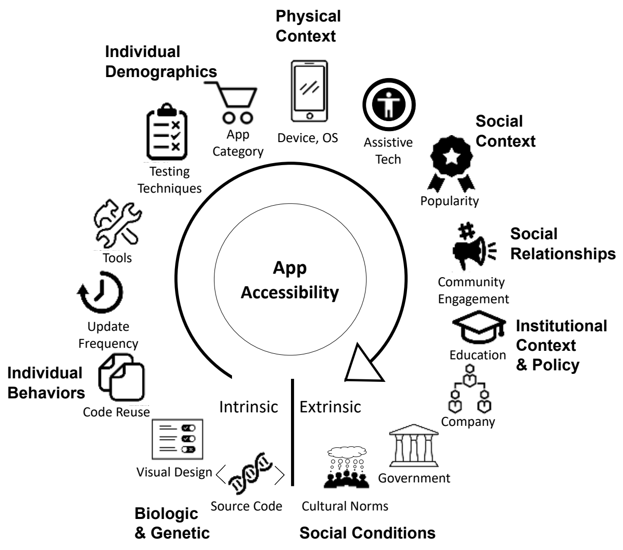


Figure 1: As a systems science, epidemiology can serve as a metaphor that changes the way we think and work with mobile app accessibility. Here, the concept of a multi-factor ecosystem from epidemiology has been applied to mobile app accessibility. An app's accessibility is a product of many factors ranging from individual and intrinsic to population-level and extrinsic.

Existing approaches to characterizing app accessibility generally focus on identifying barriers in individual apps and supporting developers in repairing those barriers. However, app accessibility cannot be fully solved by addressing individual barriers; it must be approached as a systemic and multi-factor problem. Inspired by epidemiology, my framework is the first to conceptualize accessibility barriers as *diseases* within a *population of apps* [4]. The framework emphasizes population-scale and longitudinal analyses as methods for characterizing the state of app accessibility. Further, app accessibility is positioned as a product of *environmental factors,* such as developer awareness and skill, testing tools, company structure and culture, and public opinion. Figure 1 illustrates a range of factors from *intrinsic* factors closely related to an individual app to *extrinsic* factors with more population-level influence. My epidemiology-inspired framework complements individual-focused approaches to app accessibility by providing a more holistic characterization of app accessibility. This multi-factor lens highlights opportunities to address app accessibility by improving interrelated technical, social, and structural influences. For example, rather than only looking at how to improve the accuracy of app accessibility testing tools through technical advancements, I also consider how the company environment in which professional testers use those tools impacts app accessibility.

Understanding the State of App Accessibility

My framework highlights the importance of *determining disease prevalence in a population* (i.e., measuring how often accessibility barriers occur). Advancing this objective, I developed tools to automatically test 10,000 free Android apps for seven accessibility barriers [5,6]. My analysis was orders of magnitude larger than any published analysis to date [3,8]. I identified highly prevalent accessibility barriers and types of screen elements that were likely to have barriers. For example, 23% of apps were missing labels on more than 90% of their image-based screen elements (Figure 2). Image Views and Floating Action Buttons were particularly likely to lack labels, at 92% and 99%, respectively. Another noteworthy trend that emerged was Education apps were disproportionately likely to be completely unusable with assistive technology; apps in the Education category accounted for only 7% of apps tested but made up 19% of apps that were unusable with assistive technology. These results quantify app accessibility as a wide-spread problem and inform population-level remedies to such barriers.

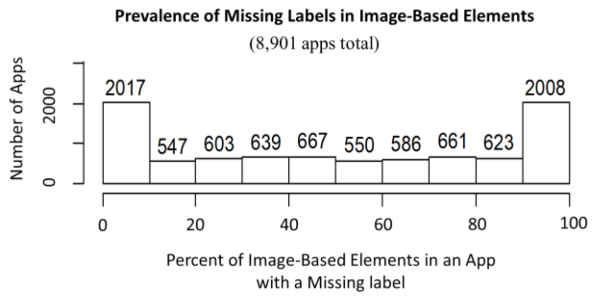
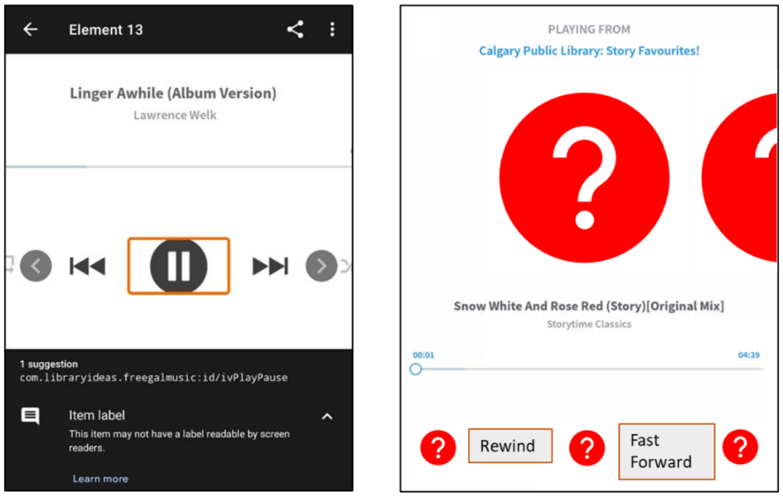


Figure 2: Prevalence of image-based elements missing labels in apps. Out of 8,901 apps with at least one image-based element.

My next epidemiology-inspired objective was *identifying environmental factors* that could affect app accessibility at scale. Guided by my prevalence analysis, I sought factors that could impact image labeling or contribute to apps being completely unusable with assistive technologies. I found Android developer guidelines provided source code examples with missing label and the static analysis Android Lint test suite that ships with the Android Studio IDE identifies missing labels in Image Views but not Floating Action Buttons. Embracing the importance of this research, a team at Google updated the Android developer guidelines to provide labeled examples based on my findings. Certain app creation tools were associated with highly inaccessible apps; for example, 100% of the elements that were identified coming from the Unity Game engine across 63 apps were inaccessible with assistive technology. Identifying accessibility shortcomings in tools is one example of the value of applying my multi-factor epidemiological lens to app accessibility.

Improving App Accessibility

Informed by my large-scale, multi-factor understanding of app accessibility, I am building tools to identify accessibility barriers and guide repairs. Most existing accessibility testing tools either rely on high accessibility expertise (e.g., manual testing, unit testing) or automation (e.g., static analysis, runtime scanners). These tools predominantly support repair through technical documentation. My work focuses on techniques that scaffold a mid-level accessibility expertise and move beyond text-based documentation for guiding repair. For example, in my developer tool prototype, a developer can see a screenshot of their app screen with all images replaced by labels that a person using a screen reader would hear (see Figure 3, right). Compared to creating text-based warnings for each image without a label (see Figure 3, left), this technique is more efficient and understandable.


Element 13
Linger Awhile (Album Version)
Lawrence Welk
PLAYING FROM
Calgary Public Library: Story Favourites!
Snow White And Rose Red (Story)[Original Mix]
Storytime Classics
00:01
04:39
Rewind
Fast Forward

1 suggestion
com.libraryideas.freegalmusic:id/ivPlayPause
Item label
This item may not have a label readable by screen readers.
Learn more

Figure 3:

(left) An existing runtime scanner highlights an image-based pause button, identifies it as unlabeled, and links to technical documentation.

(*right)* My tool replaces all images on an app screen with their screen reader labels. Developers can quickly check the correctness of image labels. Unlabeled images are represented as large red question marks.

Developers are not solely responsible for app accessibility. In some companies, distinct testing and repair teams must collaborate to create accessible apps. Professional testers are not well supported by current tools; automated tools do not leverage their expertise and manual testing is time consuming and difficult to communicate to developers. While working at Google, I added a feature to the Accessibility Test Framework for Android API [1] to allow human annotation of uncertain automated test results. My feature was released in `v3.1` of the API. I additionally prototyped a tool that prompted testers to confirm or correct uncertain automated accessibility test results. Testers valued the annotation feature; it blended the efficiency of automated testing and the correctness of manual testing.

Future Work

I envision a world in which accessibility is integrated into every facet of app development: from the low-level design and implementation details to organization-level structures, incentives, and resources. Toward that goal, my future research agenda has three main focuses: (1) extend my large-scale automated analyses by **measuring the incidence of accessibility issues over time,** (2) improve **design, development, and testing tools** in order to integrate accessibility practices in each stage of app creation, and (3) improve accessibility by **addressing systemic environmental factors** such as communication channels within a company. I briefly describe each of these below.

Longitudinal Large-Scale Analyses

My large-scale analyses were based on a single sample of apps tested for a limited set of barriers. Extending my testing methods and tools to conduct my automated accessibility tests over time gives insights into the evolving state of app accessibility. Longitudinal analyses provide metrics for assessing systemic factors. For example, trends in app accessibility during cycles of major and minor updates can reflect whether an app's development process prioritizes accessibility. As another example, longitudinal analyses can capture whether previously resolved accessibility problems are ever reintroduced. Open research questions for ongoing analyses include how to capture app datasets, what factors should be captured, and how apps should be analyzed.

Achieving this goal necessitates advancing the field through research contributions and applying that research to real-world practices. I will manifest this breadth of impact through interdisciplinary collaborations across campus and with industry connections that I have built, including with teams at Google, Microsoft, and IBM. In addition to collaborations, funding opportunities exist from research-focused organizations like the National Science Foundation (e.g., CRII and CAREER awards) and from industry (e.g., Microsoft Research Faculty Fellowship and Google Faculty Award).

Improving Design, Development, and Testing Tools

Through prototyping and user studies with developers and testers, I have begun building accessibility tools for app developers and testers. My early prototyping has shown promise for some interventions; in future work, I will explore how tools can teach designers, developers, and testers proper accessibility techniques using quantitative (e.g., measures of resultant app accessibility) and qualitative (e.g., interviews, user studies) research methods.

Company Ecosystems

My dissertation research on tools for app developers, professional testers, and third-party repair developers [9] demonstrates promising techniques for improving app accessibility. However, myriad factors prevent real-world adoption of these techniques. My ongoing survey of testing and development teams highlights factors that impact app accessibility such as information discrepancies between teams, power dynamics, and bug reporting infrastructure. My continued research in this area examines tool design in the context of these company structures. In future work, I will investigate how accessibility practices can be effectively integrated into the ecosystem of app development using methods such as in-the-wild tool usage logs, interviews, and focus groups.

References

1. Google. Accessibility Test Framework for Android. Retrieved October 1, 2020 from https://github.com/google/Accessibility-Test-Framework-for-Android.
2. D. Hall. "BECU will restore accessibility for blind customers to its website and mobile banking app". *The Seattle Times,* June 8, 2018.
3. L. Milne, C. Bennett, and R. Ladner. 2014. The Accessibility of Mobile Health Sensors for Blind Users. *Proceedings of Technology and Persons with Disabilities Conference (CSUN '14*).
4. A. Ross, X. Zhang, J. Fogarty, and J. Wobbrock. 2017. Epidemiology as a Framework for Large-Scale Mobile Application Accessibility Assessment. *ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17)*.
5. A. Ross, X. Zhang, J. Wobbrock, and J. Fogarty. 2018. Examining Image-Based Button Labeling for Accessibility in Android Apps Through Large-Scale Analysis. *ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '18)*.
6. A. Ross, X. Zhang, J. Fogarty, and J. Wobbrock. 2020. An Epidemiology-Inspired Large-Scale Analysis of Android App Accessibility. *ACM Transactions on Accessible Computing (TACCESS)*. April 2020, Article No. 4.
7. A. Ross. "An app for everything, but can everyone use it?". *UW CREATE,* May 2020. Retrieved August 26, 2020. https://create.uw.edu/an-app-for-everything-but-can-everyone-use-it/.
8. S. Yan and P. G. Ramachandran. 2019. The Current Status of Accessibility in Mobile Apps. *ACM Transactions on Accessible Computing (TACCESS)*. February 2019, Article No. 5
9. X. Zhang, A. Ross, A. Caspi, J. Fogarty, and J. Wobbrock. 2017. Interaction Proxies for Runtime Repair and Enhancement of Mobile Application Accessibility. *ACM Conference on Human Factors in Computing Systems (CHI '17).*