Non-visual Web Browsing: Beyond Web Accessibility

I.V. Ramakrishnan, Vikas Ashok, and Syed Masum Billah(✉)

Department of Computer Science, Stony Brook University, Stony Brook, NY, USA
{ram,vganjiguntea,sbillah}@cs.stonybrook.edu

Abstract. People with vision impairments typically use screen readers to browse the Web. To facilitate non-visual browsing, web sites must be made accessible to screen readers, i.e., all the visible elements in the web site must be readable by the screen reader. But even if web sites are accessible, screen-reader users may not find them easy to use and/or easy to navigate. For example, they may not be able to locate the desired information without having to listen to a lot of irrelevant contents. These issues go beyond web accessibility and directly impact web usability. Several techniques have been reported in the accessibility literature for making the Web usable for screen reading. This paper is a review of these techniques. Interestingly, the review reveals that understanding the semantics of the web content is the overarching theme that drives these techniques for improving web usability.

Keywords: Web accessibility · Web usability · Screen reader · Web content semantics

1 Introduction

The Web has permeated all aspects of our daily lives. We use the Web to obtain and exchange information, shop, pay bills, make travel arrangements, apply for college or employment, connect with others, participate in civic activities, etc. It has in effect become the indispensable ubiquitous "go-to utility" for participating in society. A 2016 report by Internet World Stats shows that Internet usage has skyrocketed by more than 1000% since 2000, to include almost half of the global population in 2016 (over 3.6 billion people) [38], making it one of the most widely used technologies.

About 15% of the world's population are living with some form of physical/sensory/cognitive disability [57]. The Web has the potential to provide an even greater benefit to such individuals who once required human assistance with many of the activities mentioned earlier. The Web opens up opportunities to do them without assistance and thereby foster independent living.

People with disabilities rely on special purpose assistive software applications for interacting with the Web. It is left to web developers to ensure that their web sites are accessible, i.e., the web sites work with such assistive software. To aid web developers in this process, the W3C Web Accessibility initiative [55] has formulated the Web Content Accessibility Guidelines [56] on how to make web pages accessible. These guidelines are essentially recommendations to web developers. An example

M. Antona and C. Stephanidis (Eds.): UAHCI 2017, Part II, LNCS 10278, pp. 322–334, 2017.
DOI: 10.1007/978-3-319-58703-5_24

recommendation states that web developers should provide text equivalents for images and semantically meaningful labels to links in web pages.

People with vision impairments browse the Web non-visually. They form a sizeable fraction of people with disabilities. Specifically, there are nearly 285 million people with vision impairments worldwide – 39 million blind and 246 million with low vision [58]. In the U.S. alone there are over 23 million Americans suffering from vision loss and over 1.5 million of them use the Internet [3]. This paper reviews the state of the art in non-visual browsing. Ever since the advent of the PC, visually impaired people have used Screen Readers (SRs), a special-purpose software application, to interact with digital content. SRs serially narrate the content of the screen using text-to-speech engines and let users navigate the content using touch or keyboard shortcuts.

Over the years there has been much progress on screen reading and more broadly on assistive technologies for a broad range of disabilities. It has been driven by several factors: (1) federal mandates such as the ADA [2] and the 21st Century Communications and Video Accessibility Act [1]; (2) companies specializing in the development of assistive technologies [18, 25, 33, 41]; large IT companies like Google, Apple and Microsoft incorporating support for accessibility in their products and services (e.g. Microsoft's MSAA and UI Automation [34, 35], Apple's NSAccessibility [9], and GNOME's ATK and AT-SPI [16]); (3) business and educational institutions adopting assistive technologies to enhance employment and educational opportunities for people with disabilities. Because of all this progress, these days visually impaired people have several high quality SRs to choose from, e.g., JAWS [25], Window-Eyes [33], Super-Nova [18], NVDA [41] and VoiceOver [53].

For visually impaired people, SRs remain the dominant technology for non-visual web browsing. Web sites that are designed based on WCAG guidelines are accessible to SRs. But making web pages accessible in and of itself does not make them usable – a problem that is primarily concerned with the "how to's" of providing a rich user experience in terms of ease of use, effectiveness in getting tasks done, etc. In this regard, SRs are not very usable or efficient for web browsing [14] and have several notable drawbacks [27]. Firstly, to be efficient, SR users have to remember an assortment of shortcuts and learn a number of browsing strategies; however, most users rely only on a small basic set of sequential navigation shortcuts, which leads to excessive interaction with computers even while performing simple browsing tasks [14]. Secondly, because one cannot judge the importance of content before listening to it; blind users typically go through reams of irrelevant content before they find what they need, thereby suffering from information overload. Thirdly, SRs are typically oblivious of the fact that web content is organized into semantic entities (e.g., menus, date pickers, search results, etc.), where each entity is composed of many basic HTML elements; the user may not know what entities are present on the page, whether s/he is navigating inside or outside an entity, where the entity's boundaries are, etc. These problems become particularly acute when performing tasks in content-rich web sites; for example, while sighted users can purchase something online or make a reservation in just a few minutes, screen-reader users often take 10 min or more [14, 45]. Yet another serious problem of not knowing the entity boundaries is that the SR's sequential readout intersperses content from different semantic entities, which can confuse and disorient the user. Lastly, in addition

to not being able to get a quick overview of the entire web page and having to read through content one element at a time, blind users also have to endure the fact that SRs navigate web pages at the syntactic level instead of the semantic one. Consequently, while sighted people see the semantic structure of the web page, blind people have access only to its syntactic structure, and most often have to navigate and listen to individual HTML elements.

The root cause of the usability problems stems from the SR's limited knowledge of the semantics of web content. Research efforts in accessibility have sought to rectify this situation by incorporating *semantic awareness* in non-visual browsing. At their core, the techniques for semantic awareness infer the semantics by analysis of the content using syntactic and structural cues in web pages, optionally supplemented by explicit knowledgebase encoding the semantics of domain-specific web sites such as travel web sites, shopping web sites, etc. Semantic awareness goes beyond web accessibility. It embodies the state of the art in making web browsing usable for SR users. In the following sections this paper reviews how semantic awareness is incorporated in non-visual browsing with SRs.

2 Semantic Awareness in Non-visual Web Browsing

We begin with some terminology: A web page can be viewed as a collection of semantic entities. Informally, we define a semantic entity to be a meaningful grouping of related HTML elements. As an illustration, Fig. 1 is a web page fragment with six semantic entities, numbered 1 to 6. The number associated with an entity is shown in red at the corner of that entity. For example, entity numbered 4 corresponds to the search-result entity showing the results for an available flight. Notice that it is a grouping of related links, button, images and text. Similarly, an article entity in a news web page is a collection of paragraphs and possibly links; a list of items entity can be a simple HTML list or a tabulated list of products with their prices and short descriptions. Observe how the semantic entities in Fig. 1 have clear visual boundaries. Sighted people can easily identify and interact with these entities because of these boundaries and moreover are easy to distinguish from each other. In contrast, blind people have to use the screen reader to figure out and guess where the entity starts and ends and how it is organized. Early on there has been a lot of research effort on identifying the boundaries of semantic entities. The basis of these efforts is segmentation, described next.

2.1 Segmentation

A segment of a web page corresponds to a contiguous fragment of web elements in the page that are "semantically" related (e.g., the news headline and article summary, menu of categories, search results, etc.). As illustration: the fragments enclosed within the rectangles in Fig. 1 are examples of segments.

Organizing a web page into segments lets users navigate between "meaningful" pieces of information and results in much better comprehension of the content. This is

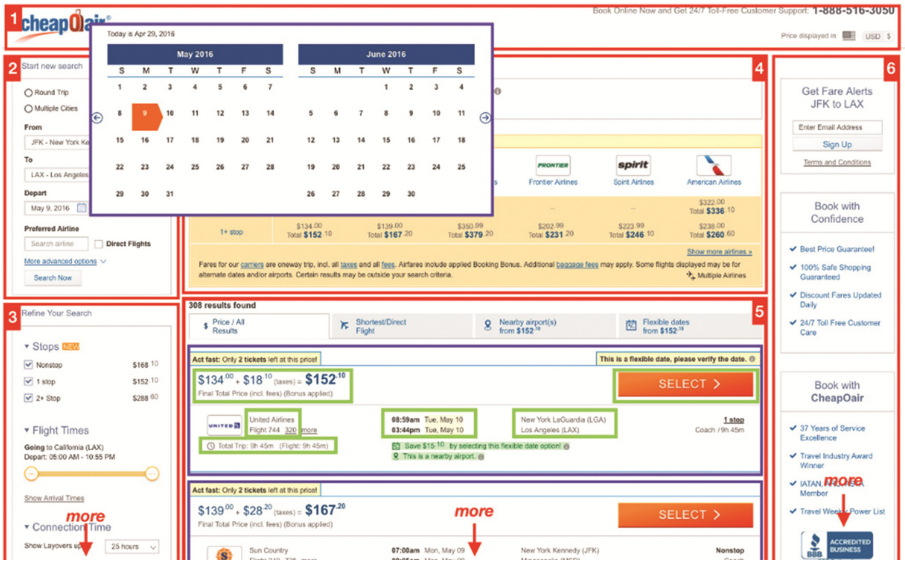

Fig. 1. Web page segmented into semantic entities (Color figure online)

especially useful for small screen devices where display area is at a premium, making it all the more important to focus on coherent and relevant chunks of content.

Several techniques for segmenting web pages have appeared in the research literature (e.g. see [5, 15, 20, 42, 49, 60–62]). They utilize a range of features in the pages from visual cues, to spatial locality information, to presentational similarity, to patterns in the content, etc.

Segmentation has been used in a variety of applications such as adapting content on small screen devices (e.g. [61]), data cleaning and search (e.g. [60, 62]) and web data extraction (e.g. [5, 49]). Recognizing the importance of segmentation, Apple's VoiceOver also segments web pages with its "auto web spot" feature. More importantly segmentation is an important component in many techniques that have been developed to enhance web usability for people with visual impairments. We review these techniques now.

2.2 Segmentation-Based Techniques for Enhancing Web Usability

Clutter-Free Browsing

As SR users browse the Web, they have to filter through a lot of irrelevant data, i.e., clutter. For example, most web pages contain banners, ads, navigation bars, and other kinds of distracting data irrelevant to the actual information desired by the users. Navigating to the relevant information quickly is critical for making non-visual web browsing usable. For finding information quickly, SRs allow keyword searching. This assumes that users already know what they are looking for, which is not necessarily true in all cases, especially in ad hoc browsing.

The relevance of different entities on any page is subjective. However, as soon as the user follows a link it is often possible to use the context of the link to determine the relevant information on the next page and present it to the user first. A technique described in [22] uses the context of a link, defined as the text surrounding it, to get a preview of the next web page so that users could choose whether or not they should follow the link. The idea of using the words on the link as well as those surrounding it is used in [31] to more accurately identify the beginning of main content relevant to the link, on the following page. For example, clicking on a news link, it will directly place the reading position to the beginning of the news article on the next page. The user can now listen to the article clutter-free.

This focus on removing "clutter" in a web page for readability purposes motivated the Readability [47] tool and the "Reader" button in the Safari browser. Both employ heuristics driven by visual and structural cues (such as link density in a node, text length, node position in the tree, representative font size, tags like headers and div) for extracting the main content in a web page. More precise clutter-removal is done in [24] by tightly coupling visual, structural and linguistic features.

Online Transactions

Web transactions broadly refer to activities such as shopping, registrations, banking and bill-payments online. Such transactions involve several steps that typically span several web pages. This can significantly exacerbate information overload on SR users and affect their productivity. In this regard, as was mentioned earlier, while sighted users can purchase something online or make a reservation in just a few minutes, SR users often take 10 min or more [14, 45].

Usually one needs to browse only a small fragment of a web page to perform a transaction. This observation is the basis of the method in [51] for doing web transactions

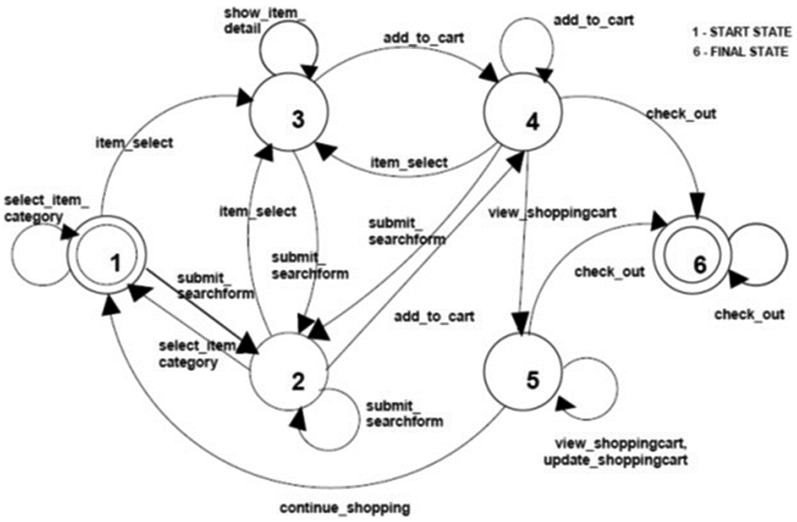

Fig. 2. Automaton fragment for online shopping

more efficiently and with less information overload. Specifically, a web transaction is modeled as a process automation (see Fig. 2). In that automaton, each node/state represents transaction-specific semantic entities (e.g., search results or product description) it expects to find in the web page that is given as the state's input and the edges/arrows are possible user actions corresponding to the entity represented by the state (e.g., clicking a button). Segmentation is used to identify transaction-specific semantic entities in that page.

Stepping through a transaction corresponds to making transitions in the automaton; at each step, only the entity relevant to that step in the transaction is presented thereby skipping all the other content in the page. The process automaton is learned from labeled training sequences gathered from click streams of user actions. In [30], the construction process of the automata was completely automated.

3 Skimming and Summarization

Skimming and summarization are complementary techniques that help the users obtain the gist of a text document. Summarization is a snippet of text, explicitly constructed from the document's text, which conveys the essence of the information contained in the document. Skimming, on the other hand, conveys it by identifying a few informative words in the document. These two topics have attracted the attention of the Information Retrieval and Natural Language Processing research community [32, 46]. Summarization and skimming are naturally applicable to web pages, especially for non-visual browsing as they give the users a "peek" into the content without having to make them listen to it in its entirety.

Using the notion of the context of a web page as a collection of words gathered around the links on other pages pointing to that web page, [17] uses this context to obtain a summary of the page. Summarization using context is also explored by the InCommonSense system [7], where search engine results are summarized to generate text snippets. In BrookesTalk [52] the web page is summarized as a collection of "significant" sentences drawn from the page. These are sentences containing key trigrams (phrases containing three words) that are identified using simple statistics. The AcceSS system [43] makes web pages accessible through a transcoding process comprised of summarization and simplification steps. The former uses the idea of context of a link from [22] to get a preview of the page and the latter identifies "important" sections to be retained. The Summate system picks a maximum of four sentences from a web page as its gist summary [23].

A method for non-visual skimming is described in [4]. It works as follows: Firstly, every sentence is parsed to extract grammatical relations amongst its words. Secondly, a lexical tree based on these relations is constructed, where each node of the tree represents a word in the sentence. Thirdly, for every word in this tree, it's grammatical (i.e., POS tags) as well as structural features (related to in-degree/out-degree, etc.) are extracted. These features are given to a trained classifier to determine whether or not to include the word in the skimming summary. Finally, a subtree consisting of the selected

words is constructed. This subtree represents the skimming summary that users interact with via an interface.

4 Speech-Based Browsing

Speech input has long been recognized as a powerful interaction modality for non-visual browsing because of its potential to alleviate the shortcomings of a SR's keyboard-based press-and-listen mode of interaction. An exposition of these shortcomings and how speech modality can address them appears in [10]. Several systems support browsing with spoken commands.

Voice browsers like PublicVoiceXML [44] and JVoiceXML [48] have an interactive voice interface for browsing web content. Browsing the Web with these voice browsers requires the conversion of web pages to JVoiceXML [54], a document format that operates within a controlled domain. In some cases, voice navigation is used for improving one particular aspect of browsing, e.g., [21] focuses on making the menus and submenus appearing on a webpage voice-accessible; Windows Speech Recognition (WSR) [36] makes it possible to follow a link by speaking its ordinal number and enables a few other basic commands. Alas, neither is accessible to blind users. An Android accessibility service, JustSpeak [59], can be used with any Android app or accessibility service, and is able to process chains of commands in a single utterance. It is limited to a few basic browsing-related commands, specifically: activate, scroll, toggle switch, long press, toggle checkbox. Dragon NaturallySpeaking Rich Internet Application feature [40] enables the user to control certain websites by voice. It provides limited support to select parts of only four (4) websites in specific browsers, and lists many additional caveats and limitations, both general and browser specific. But usage of visual cues and a graphical user interface for listing possible utterances/commands significantly reduces Dragon's accessibility for blind people. Capti-Speak is speech-augmented Screen Reader developed recently [10]. Capti-Speak translates spoken utterances into browsing actions and generates appropriate TTS responses to these utterances. Each spoken utterance is part of an ongoing dialog. It employs a custom dialog-act model [11] that was developed exclusively for "speech-enabled non-visual web access" to interpret every spoken utterance in the context of the most recent state of the dialog, where the state, in some sense, encodes the history of previous user utterances and system responses.

5 Web Automation

Web automation broadly refers to methods that automate typical web browsing steps such as form filling clicking links and more generally any kind of repetitive steps, on behalf of the user. They therefore play an important role in making non-visual web browsing more usable.

There are several research prototypes that automate web browsing. The traditional approach to Web automation is via macros, which are pre-recorded sequences of instructions that automate browsing steps. The recorded macros are later replayed to automate the same sequence of recorded steps. Macros are usually created by the well-known

process of programming by demonstration, where the developer demonstrates to the macro recorder what steps need to be done and in what sequence. With the exception of Trailblazer [13], which is built on top of the CoScripter system [28], most of the macro-based web automation systems are meant for sighted users.

The CoScripter system is a tool for recording macros to automate repetitive web tasks and replay them. CoCo [26] takes user commands in the form of (restricted) natural language strings in order to perform various tasks on the Web and maps these natural language commands to macros stored in the CoScripter Wiki [28] and CoScripter Reusable History (ActionShot [29]). While both CoScripter and CoCo are meant for sighted users, TrailBlazer [13] allows SR users to provide a brief description of their tasks for which it dynamically cerates new macros by stitching together existing macros in the CoScripter Wiki. It also attempts to adapt macros explicitly recorded for one website to similar tasks on other sites.

The main drawback with macros is that they lack the flexibility necessary to allow the user to deviate from the prerecorded sequence of steps, or to choose between several options in each step of the macro. Those difficulties make macro-based approaches too limiting to be useful for people with vision impairments. A flexible, macro-less model-based approach to web automation is described in [45]. The model is constructed based on the past browsing history of the user. Using this history and the current web page as the browsing context, the model can predict the most probable browsing actions that can be performed by the user. The model construction is fully automated. Additionally, the model is continuously and incrementally updated as history evolves, thereby, ensuring the predictions are not "outdated".

6 Web Screen Reading Assistants

Voice-activated Assistants are in vogue these days. The recent wave of such assistants include Apple's Siri [50], Samsung's S Voice [8], Google Now [19], Nuance's Nina [39], Microsoft's Cortana [37] and Amazon's Echo [6]. These assistants are typically used for factual question answering and doing common tasks such as finding restaurants and managing calendars. Users are finding them to be invaluable, so much so that it is fast becoming an integral part of their digital world. But Assistants fall short when it comes to general-purpose web browsing. Some, e.g., Siri, fall back to simple web search when they are unable to answer user's requests. Regardless, Assistants have the potential to become a transformative assistive technology and remains mostly unexplored in accessibility research. But a recent work that explores the applicability of Assistants in web screen reading suggests that it has the potential to significantly enhance web usability for SR users [12]. In this work, the web screen reading assistant, SRAA, is rooted in two complimentary ideas: First, it elevates the interaction to a higher level of abstraction - from operating over (syntactic) HTML elements to operating over semantic web entities. Doing so brings blind users closer to how sighted people perceive and operate over web entities. Second, the SRAA provides a dialog interface using which users can interact with the semantic entities with spoken commands. The SRAA interprets and executes these commands.

SRAA is driven by a Web Entity Model (WEM), which is a collection of the semantic entities in the underlying webpage. The WEM is dynamically constructed for any page using an extensive generic library of custom-designed descriptions of commonly occurring semantic entities across websites. The WEM imposes an abstract semantic layer over the web page. Users interact with the WEM via natural-language spoken commands (They can also use keyboard shortcuts). By elevating interaction with the web page to the more natural and intuitive level of web entities, SRAA relieves users from having to press numerous shortcuts to operate on low-level HTML elements - the principal source of tedium and frustration. Figure 3 below depicts a scenario snippet of how a user interacts with SRAA to review the search results for making a flight reservation in Expedia, depicted in Fig. 1.

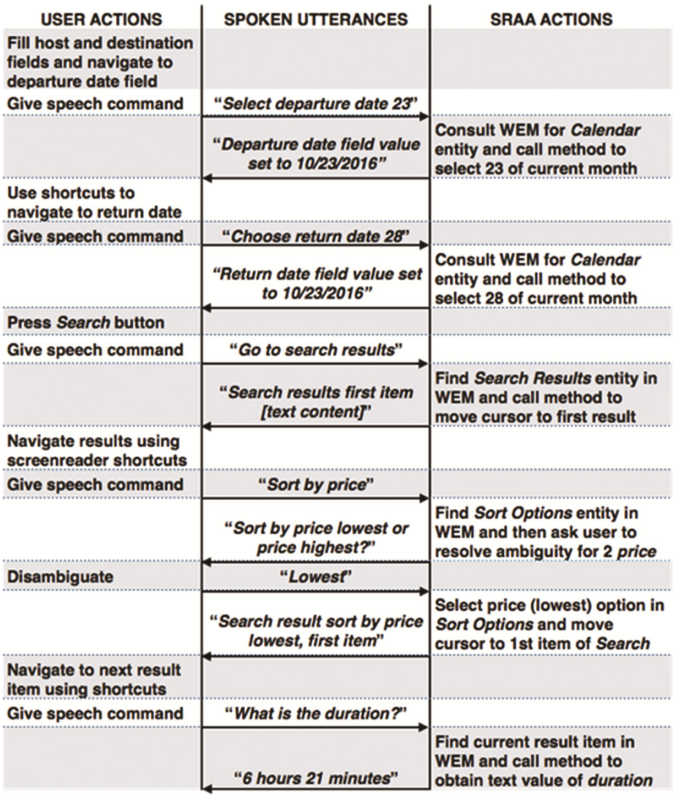

Fig. 3. Example user interaction scenario with SRAA

User actions (with keystrokes) and SRAA's internal operations corresponding to user commands appear in the left and right column respectively. Arrows pointing right and left in the middle column correspond to user's spoken commands and SRAA's synthesized-speech responses. The scenario sequence flows from left-to-right and top-to-bottom. As seen in Fig. 3, users no longer need to spend time and effort locating and

getting the information they need; instead, they simply use speech commands to delegate this task to the SRAA, which also resolves any ambiguity in the process (e.g., "sort by price"). Observe the simplicity and ease of interaction with SRAA compared to using only a vanilla screen reader. While sighted users' interaction with the Web is implicitly driven by the semantics of web entities, SRAA makes it explicit to the blind users. It brings blind users closer to how sighted people perceive and interact with the Web – which is the highest degree of web usability any technology can expect to achieve.

7 Conclusion

This paper reviewed some of the important techniques reported in the accessibility research literature, for making web sites usable for screen reading. The review included clutter-removal techniques, support for online transactions, skimming and summarization, interacting using speech, web automations and Assistants. The overall aim of these techniques is to make the Web easy to use and navigate, and reduce information overload. The common thread underlying these techniques was their use of the semantic knowledge of the web content to improve the usability of the Web.

The reviewed techniques mostly focused on desktop computing as this is still the primary way visually-impaired people use computers at home, at school, and at work. Nowadays, smart phone devices are becoming an indispensable device in people's lives, including people with disabilities. These devices have numerous apps that assist users in performing various day-to-day activities. This raises several interesting research questions regarding the usability of these apps for people with vision impairments and how it can be further improved.

Acknowledgement. This research was supported in part by NSF awards: IIS-1447549, CNS-1405641; National Eye Institute of NIH award: R01EY026621; NIDILRR: 90IF0117-01-00. NIDILRR is a Center within the Administration for Community Living (ACL), Department of Health and Human Services (HHS). The content is solely the responsibility of the authors and does not necessarily represent the official views of NIH nor represent the policy of NIDILRR, ACL, HHS.

References

1. 21st Century Communications and Video Accessibility Act (CVAA). https://www.fcc.gov/consumers/guides/21st-century-communications-and-video-accessibility-act-cvaa
2. Introduction to the ADA. https://www.ada.gov/ada_intro.htm
3. AFB. Facts and Figures on American Adults with Vision Loss (2017). http://www.afb.org/Section.asp?SectionID=15&TopicID=413&DocumentID=4900
4. Ahmed, F., Borodin, Y., Soviak, A., Islam, M., Ramakrishnan, I.-V., Hedgpeth, T.: Accessible skimming: faster screen reading of web pages. In: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, pp. 367–378. ACM: Cambridge, Massachusetts, USA (2012)
5. Álvarez, M., Pan, A., Raposo, J., Bellas, F., Cacheda, F.: Finding and extracting data records from web pages. J. Sig. Process. Syst. **59**(1), 123–137 (2010)

6. Amazon. Echo (2015). www.amazon.com/echo
7. Amitay, E., Paris., C.: Automatically summarizing web sites: is there a way around it? In: CIKM 2000: Proceedings of the Ninth International Conference on Information and Knowledge Management, pp. 173–179. ACM Press (2000)
8. Android, S Voice (2012). http://www.androidcentral.com/tag/s-voice
9. Apple, NSAccessibility. https://developer.apple.com/reference/appkit/nsaccessibility
10. Ashok, V., Borodin, Y., Puzis, Y., Ramakrishnan, I.-V.: Capti-Speak: a speech-enabled web screen reader. In: Proceedings of the 12th Web for All Conference. ACM, Florence, Italy (2015)
11. Ashok, V., Borodin, Y., Stoyanchev, S., Ramakrishnan, I.-V.: Dialogue act modeling for non-visual web access. In: The 15th Annual SIGdial Meeting on Discourse and Dialogue, SIGDIAL, Philadelphia, PA, USA (2014). http://www.aclweb.org/anthology/W/W14/W14-43.pdf#page=143
12. Ashok, V., Puzis, Y., Yevgen, B., Ramakrishnan, I.-V.: Web screen reading automation assistance using semantic abstraction. In: 22nd ACM International Conference on Intelligent User Interfaces (2017)
13. Bigham, J.-P., Lau, T., Nichols, J.: Trailblazer: enabling blind users to blaze trails through the web. In: Proceedings of the 13th International Conference on Intelligent User Interfaces. ACM, Sanibel Island, Florida, USA (2009)
14. Borodin, Y., Bigham, J., Dausch, G., Ramakrishnan, I.-V.: More than meets the eye: a survey of screen-reader browsing strategies. In: Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A), pp. 1–10. ACM, Raleigh, North Carolina (2010)
15. Cai, D., Yu, S., Wen, J.-R., Ma, W.-Y.: VIPS: a vision based page segmentation algorithm. Microsoft Technical report (2004)
16. GNOME Accessibility Architecture (ATK and AT-SPI). https://accessibility.kde.org/developer/atk.php
17. Delort, J.Y., Bouchon-Meunier, B., Rifqi, M.: Enhanced web document summarization using hyperlinks. In: Proceedings of the 14th ACM Conference on Hypertext and Hypermedia, pp. 208–215. ACM, Nottingham, UK (2003)
18. Dolphin, SuperNova Screen Reader. http://www.yourdolphin.com/productdetail.asp?id=1
19. Google. Google Now. http://www.google.com/landing/now/#utm_source=google&utm_medium=sem&utm_campaign=GoogleNow
20. Guo, H., Mahmud, J., Borodin, Y., Stent, A., Ramakrishnan, I.-V..: A general approach for partitioning web page content based on geometric and style information. In: Proceedings of the International Conference on Document Analysis and Recognition (2007)
21. Han, S., Jung, G., Ryu, M., Choi, B.-U., Cha. J.: A voice-controlled web browser to navigate hierarchical hidden menus of web pages in a smart-tv environment. In: Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion. International World Wide Web Conferences Steering Committee (2014)
22. Harper, S., Goble, C., Stevens, R., Yesilada, Y.: Middleware to expand context and preview in hypertext. In: Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility (2004)
23. Harper, S., Patel, N.: Gist summaries for visually impaired surfers. In: Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility (2005)
24. Islam, M.-A., Ahmed, F., Borodin, Y., Ramakrishnan, I.-V.: Tightly coupling visual and linguistic features for enriching audio-based web browsing experience. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 2085–2088. ACM, Glasgow, Scotland, UK (2011)

25. JAWS. Screen reader from Freedom Scientific (2013). http://www.freedomscientific.com/products/fs/jaws-product-page.asp
26. Lau, T., Cerruti, J., Manzato, G., Bengualid, M., Bigham, J., Nichols, J.: A conversational interface to web automation. In: Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology. ACM, New York, USA
27. Lazar, J., Allen, A., Kleinman, J., Malarkey, C.: What frustrates screen reader users on the web: a study of 100 blind users. Int. J. Hum.-Comput. Interact. **22**(3), 247–269 (2007)
28. Leshed, G., Haber, E.-M., Matthews, T., Lau, T.: CoScripter: automating & sharing how-to knowledge in the enterprise. In: Proceeding of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems. ACM, Florence, Italy (2008)
29. Li, I., Nichols, J., Lau, T., Drews, C., Cypher, A.: Here's what I did: sharing and reusing web activity with ActionShot. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM (2010)
30. Mahmud, J., Borodin, Y., Ramakrishnan, I.-V., Ramakrishnan, C.-R.: Automated construction of web accessibility models from transaction click-streams. In: Proceedings of the 18th International Conference on World Wide Web. ACM, Madrid, Spain (2009)
31. Mahmud, J.-U., Borodin, Y., Ramakrishnan, I.-V.: CSurf: a context-driven non-visual web-browser, In: Proceedings of the 16th International Conference on World Wide Web. ACM, Banff, Alberta, Canada (2007)
32. Manning, C.-D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge University Press (2008)
33. GW Micro - Window-Eyes. http://www.gwmicro.com/Window-Eyes/
34. Microsoft, Microsoft Active Accessibility: Architecture. https://msdn.microsoft.com/en-us/library/windows/desktop/dd373592(v=vs.85).aspx
35. Microsoft, UI Automation Overview. http://msdn.microsoft.com/en-us/library/ms747327.aspx
36. Microsoft. Common commands in Speech Recognition (2014). http://windows.microsoft.com/en-us/windows/common-speech-recognition-commands#1TC=windows-7
37. Microsoft. Cortana contextual awareness (2014). http://www.bing.com/dev/en-us/contextual-awareness
38. Miniwatts Marketing Group. Internet Usage Statistics: The Internet Big Picture World Internet Users and Population Stats (2016). http://www.internetworldstats.com/stats.htm
39. Nuance. Nina (2012). http://www.nuance.com/landing-pages/products/nina/default.asp
40. Nuance. Dragon Naturally Speaking Rich Internet Application (2014). http://nuance.custhelp.com/app/answers/detail/a_id/6940/~/information-on-rich-internet-application-support
41. NVAccess, NV Access: Home of the free NVDA Screen Reader. http://www.nvaccess.org/
42. Zhai, Y., Liu., B.: Web data extraction based on partial tree alignment. In: Proceedings of the 14th international conference on World Wide Web. ACM (2005)
43. Parmanto, B., Ferrydiansyah, R., Saptono, A., Song, L., Sugiantara, I.-W., Hackett, S.: AcceSS: accessibility through simplification & summarization. In: Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A). ACM, Chiba, Japan (2005)
44. Public Voice Lab, S., PublicVoiceXML. 2002
45. Puzis, Y., Borodin, Y., Puzis, R., Ramakrishnan, I.-V.: Predictive Web Automation Assistant for People with Vision Impairments. In: Proceedings of the 22th International Conference on World Wide Web. ACM, Rio de Janeiro, Brazil (2013)
46. Radev, D.-R., Hovy, E., McKeown, K.: Introduction to the special issue on summarization. Comput. Linguist. **28**(4), 399–408 (2002)
47. Readability. https://www.readability.com/

48. Schnelle, JVoiceXML (2013). http://webdesign.about.com/gi/o.htm?zi=1/XJ&zTi=1&sdn=webdesign&cdn=compute&tm=171&f=00&tt=14&bt=3&bts=31&zu=http%3A//jvoicexml.sourceforge.net/
49. Zhu, J., Nie, Z., Wen, J.-R., Zhang, B., Ma., W.-Y.: Simultaneous record detection and attribute labeling in web data extraction. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2006)
50. Siri: The Personal Assistant on Your Phone (2013). http://siri.com/
51. Sun, Z., Mahmud, J., Ramakrishnan, I.-V., Mukherjee, S.: Model-directed Web transactions under constrained modalities. In: ACM Transactions on the Web (2007)
52. Zajicek, M., Powell, C., Reeves, C.: Web search and orientation with BrookesTalk. In: Technology and Persons with Disabilities Conference (CSUN) (1999)
53. VoiceOver, Screen reader from Apple (2015)
54. VoiceXML. W3C - Voice Extensible Markup Language (2009). http://www.w3.org/TR/voicexml20
55. WAI. W3C Web Accessibility Initiative (1997). http://www.w3.org/WAI/
56. WCAG. W3C Web Content Accessibility Guidelines (2009). http://www.w3.org/TR/WCAG10/
57. WHO-disability-data (2011). http://www.who.int/disabilities/world_report/2011/report/en/
58. WHO. Visual impairment and blindness (2014). http://www.who.int/mediacentre/factsheets/fs282/en/
59. Zhong, Y., Raman, T.-V., Burkhardt, C., Biadsy, F., Bigham, J.-P.: JustSpeak: enabling universal voice control on Android. In: Proceedings of the 11th Web for All Conference. ACM, Seoul, Korea (2014)
60. Yi, L., Liu, B.: Eliminating noisy information in web pages for data mining. In: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (2003)
61. Yin, X., Lee, W.-S.: Using link analysis to improve layout on mobile devices. In: Proceedings of the International World Wide Web Conference (WWW). ACM (2004)
62. Yu, S., Cai, D., Wen, J.-R., Ma, W.-Y.: Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In: Proceedings of the International World Wide Web Conference (WWW) (2003)