

JustSpeak: Enabling Universal Voice Control on Android

Yu Zhong¹, T.V. Raman², Casey Burkhardt², Fadi Biadisy² and Jeffrey P. Bigham^{1,3}

Computer Science, ROHCI¹
University of Rochester
Rochester, NY, 14627
zyu@cs.rochester.edu

Google Research²
Mountain View, CA, 94043
{raman, caseyburkhardt,
biadisy}@google.com

Human-Computer Interaction Institute³
Carnegie Mellon University
Pittsburgh, PA, 15213
jbigham@cmu.edu

ABSTRACT

In this paper we introduce *JustSpeak*, a universal voice control solution for non-visual access to the Android operating system. *JustSpeak* offers two contributions as compared to existing systems. First, it enables system wide voice control on Android that can accommodate any application. *JustSpeak* constructs the set of available voice commands based on application context; these commands are directly synthesized from on-screen labels and accessibility metadata, and require no further intervention from the application developer. Second, it provides more efficient and natural interaction with support of multiple voice commands in the same utterance. We present the system design of *JustSpeak* and describe its utility in various use cases. We then discuss the system level supports required by a service like *JustSpeak* on other platforms. By eliminating the target locating and pointing tasks, *JustSpeak* can significantly improve experience of graphic interface interaction for blind and motion-impaired users.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Voice I/O*; K.4.2 Computers and Society: Social Issues—*Assistive technologies for persons with disabilities*

General Terms

Human Factors, Design

Author Keywords

Universal voice control, accessibility, Android, mobile

INTRODUCTION

Mouse and multi-touch surface have been widely used as the main input methods on computers and mobile devices for their reliable accuracy. But under circumstances when visual access to the display is impossible or hindered, or for users with dexterity issues, it is difficult to point at a target so they are often less effective.

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 ACM 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.

W4A 14, April 07 - 09 2014, Seoul, Republic of Korea.

Copyright 2014 ACM 978-1-4503-2651-3/14/04 ...\$15.00.

<http://dx.doi.org/10.1145/2596695.2596720>

Sighted people also often face this problem, e.g. when driving. As for motion-impaired people, they face challenges when pointing at targets, as most GUI interaction techniques require accurate physical movements of hands or fingers. Voice control is an effective alternative input mode which does not require target locating and pointing, which often costs the majority of time needed to complete interaction according to Fitts's law [12]. Former studies showed that given this advantage, speech input is preferred by blind people when making text entry [3]. Of course there are also challenges with speech interface, such as difficulties in error correction [3], larger learning overhead and lower usability in noisy environments. Existing voice control systems are constrained in many ways. The majority of them are either built on application level (e.g. speech input methods) or limited by pre-defined commands set.

In this paper, we present *JustSpeak*, which is designed and built with a fundamentally different architecture on the system level of Android. *JustSpeak* aims at enabling universal voice control on Android operating system to help users quickly and naturally control the system non-visually and hands-freely. The application runs in the background as a system service which can be activated with simple gestures or a Near Field Communication (NFC) tag no matter which application is running in the foreground. When activated, *JustSpeak* records spoken speech. The audio is then transcribed into plain texts and parsed into computer understandable commands which are automatically executed. Unlike most current systems, *JustSpeak* can accommodate commands inside any installed application. Moreover, *JustSpeak* supports multiple commands in one utterance. For instance, to open the Gmail application then refresh the inbox, two commands can be spoken together "*Open Gmail then refresh*". Those two features enable *JustSpeak* to offer more intuitive and efficient eyes-free and hands-free interaction.

The contributions of this paper include:

- a mobile application, *JustSpeak*, that enables universal voice control on Android devices;
- a description of the system design that empowered accommodation of adaptive commands parsing and chaining; and
- a discussion on leveraging speech to enhance interaction for blind people and people with dexterity issues.

RELATED WORK

Non-visual Interaction

Over the last two decades, many screen readers [1, 4, 9] have been successful in improving GUI accessibility for blind users on multiple platforms. However, blind users still need to adapt to traditional input techniques that require target discovery. On non-touch devices, they often use keyboard shortcuts to iterate through the interface linearly (top to bottom, left to right) to find an object, and use gestures similarly on touch interfaces. Those interaction skills often fail to satisfy blind users' needs, thus it's not surprising to see many work-arounds and strategies were spontaneously developed by them to cope with these kinds of situations[5, 14]. Many research efforts were also invested to address interaction barriers for blind users, especially on multi-touch screens[8, 10]. For sighted people who do not use screen readers, it is almost impossible for them to interact with their devices non-visually. Although this is often merely nuisance for them, sometimes it causes great inconvenience, i.e., when someone wants to check the map while driving.

Voice Control Systems

We define voice control systems as applications or devices controlled by means of human voice. All mainstream smart mobile phones now support different levels of voice control [15]. The most advanced and popular voice control applications are Google Now and Siri.

Google Now is a dialogue system on Android (4.1 and above) that supports execution of a set of actions on the device, including functions of most Google products. [6] Apple firstly introduced simple voice control into iOS as a new feature of iOS 3, they then partnered with Nuance to bring Siri [2] into iPhone 4S and newer models.

Google Now and Siri are both limited by the way they define commands. Essentially each command can only have one key word or phrase which specifies a pre-defined function, for instance, "calendar". If there is no key word or the sentence does not satisfy the grammars, the utterance will be treated as a web search query. Therefore they can not work with third party applications. Also, if a user wants to execute more than one commands, s/he has to repeat the process several times.

Android Accessibility APIs

Android accessibility APIs [13] give developers ability to improve accessibility to users who have special needs. With the APIs, developers can make their own application more accessible, or make their accessibility services which provide enhancements for other applications.

Given permissions from users, developers can access the on screen view hierarchy, accessibility events and enhance user experiences accordingly with the APIs. Those APIs act as a delegate between applications and the system so that an accessibility service can be aware of any interaction and interface changes triggered by the inputs. Developers can also leverage accessibility APIs to fire interaction events.

JUSTSPEAK

JustSpeak is an Android accessibility service designed for use with any existing applications and accessibility services (e.g. TalkBack) on Android 4.2+. The interaction to use *JustSpeak* is intuitive and fully accessible for blind users. It has been released on Google Play Store for free.

In this section, we first discuss how *JustSpeak* handles single command speech, then describe the system designs which support multiple commands in one sentence.

System Description

Once installed, *JustSpeak* can be enabled under *Settings* -> *Accessibility*, this operation only needs to be performed once. *JustSpeak* can also be disabled there.

As shown in Figure 1, *JustSpeak* has three modules. The first is speech recognition, then the utterance parsing processes the recognition results into commands. And finally we search for actionable objects on the screen with command arguments and manipulate the highest matching object or initiate system commands accordingly.

JustSpeak is power efficient as it runs in sleeping mode while not being used. When the user wants to issue voice commands, *JustSpeak* can be easily activated by dragging the home button up or scanning a NFC tag. Then both visual and audio cues are provided to alert the user of activation. In order to let sighted users easily discover valid commands, *JustSpeak* briefly illustrates the labels on top of each actionable object after activation. Blind users learn valid commands intuitively from spoken representation of UI provided by screen readers.

Speech Recognition

JustSpeak uses the Google ASR service, in addition to the reliable performance, Google ASR gives developers flexibility by offering both offline and online recognition [11]. Therefore, *JustSpeak* can be used without internet connection. Of course, there are advantages with online recognition which returns multiple scored results as opposed to single result when using offline recognition. For instance, during a trial when a user said "setting", Google online ASR returned three different results: "setting(0.90), settings(0.08), sitting(0.02)". This feature gives *JustSpeak* ability to increase speech recognition error tolerance and voice commands success rate.

Utterance Parsing

JustSpeak enables voice control of all the commands supported by the Android accessibility APIs as well as some global system commands as listed in the instructions¹.

JustSpeak has a flexible grammar based parsing process. Commands can be expressed in different ways like communication between humans. When *JustSpeak* receives the scored results from ASR, it tries to process each result with the defined grammars. Once a voice recognition result passes grammar check, we wrap it up in the form of command name and arguments, then pass it to the execution module.

¹ *JustSpeak* instructions, <http://goo.gl/dF7YJH>

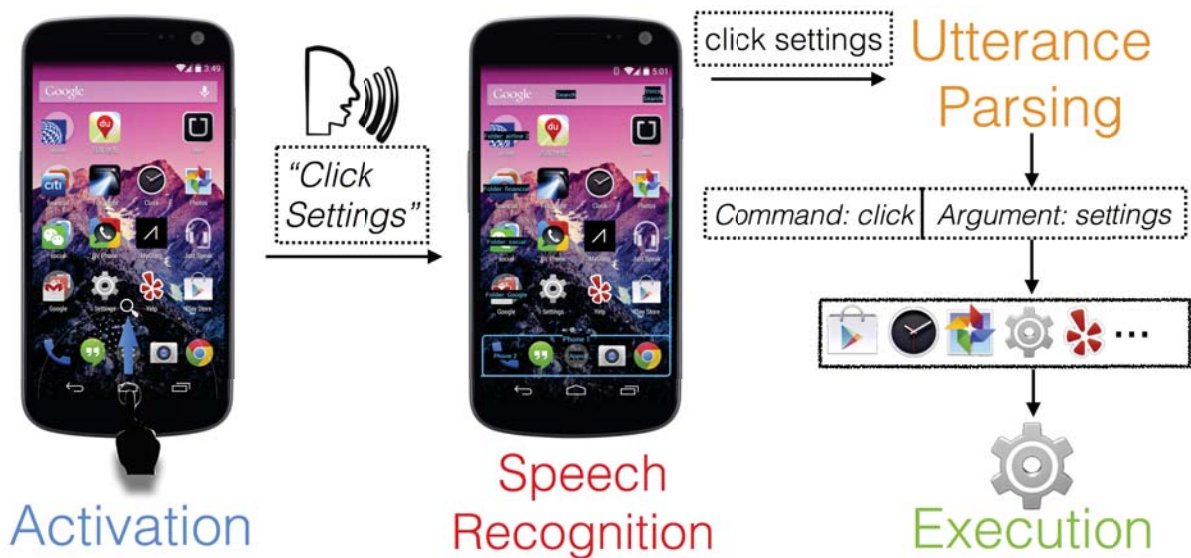


Figure 1. When activated, *JustSpeak* records audio spoken by the user, transcribes it into plain text, then parses the text into formal commands, and then finally finds the correct object on the screen to perform the command.

Commands Execution

Most global commands can be executed straightforwardly. However, local commands have arguments associated with on-screen controls. To locate actionable objects for local commands, we maintain a view indexer in *JustSpeak*, this indexer is a hashmap mapping texts to associated interface elements. Since the screen contents are being constantly updated, *JustSpeak* listens to accessibility events² and dynamically update the indexer.

When a local command arrives, the execution module queries the indexer with its arguments to find matching nodes. Since user inputs don't always match the labels, we used a flexible character overlapping based ranking algorithm. For instance, for command 'compose message', 'compose a new message' yields higher score than 'show message'. Once the indexer returns a node, *JustSpeak* validates whether the command can be performed on it, if not, it continues to a lower ranked node.

As discussed before, online ASR returns multiple results which become several commands, *JustSpeak* tries to process each one by descending scoring order. Speech feedback is provided by default when a command is executed.

Chaining of Commands

Support of commands chaining is an important feature of *JustSpeak* for two reasons. Firstly, it is more efficient to combine multiple commands than repeating the dialog turn; secondly, it is more natural and consistent with the way spontaneous speeches are produced. In *JustSpeak*, all supported functions can be chained into one sentence and then parsed into an array of commands. A challenge here is disambiguation, i.e., sentence 'click reset and home' can be viewed as either one command: clicking button 'reset and home' or two commands: clicking button 'reset' then going to home screen. To overcome

²Android accessibility events, <http://goo.gl/y3N74a>.

this challenge, grammars of each supported action were detailedly defined, *JustSpeak* also assigns higher priority to commands that can be validated on current screen. Therefore, if there is a button 'reset', then the two commands result will be preferred. Since execution of an action usually causes numbers of interface updates, the array of commands can not be executed simultaneously. In fact, the execution module only tries to process the first command, and then waits until all the events fired by the previous command are settled and then proceeds to the next command. Efficiency of chained commands largely depend on processing interface updating speed of the device. ASR only takes a very small amount of time which is almost independent of speech length.

USE CASES AND DISCUSSION

JustSpeak innovatively provides enhancements to all Android applications and the Android system itself by the means of voice control that can be accessed non-visually and hands-freely across the whole platform. Its value is not limited to assisting blind users.

For Blind Users

As the primary user group of non-visual interaction techniques, we believe blind users of Android can interact with their devices faster and more easily with *JustSpeak*. On Android, *JustSpeak* does not interfere with existing screen readers but work with them. We have observed that blind users often spend a large amount of time looking for an object on mobile interface. The main reason is that screen readers are designed as a linear screen explorer, although blind users usually have strategies to speed up interaction with screen readers, it still takes more time and efforts for them to locate a target. *JustSpeak* can reduce the time and efforts needed to access application controls for blind users. A simple case that blind users of Android benefit greatly from *JustSpeak* is

launching applications. It is usually a nightmare to fumble through pages of application icons. With the assistance of *JustSpeak*, this operation can be as easy as saying an application name. After the release of *JustSpeak*, we have noticed many users, both blind and sighted, use it as their regular app launcher.

For Sighted Users

Sighted users often find them inevitably placed under situations where non-visual or hands-free interaction is required. *JustSpeak* is an effective alternative to complete urgent tasks under those circumstances. Under normal conditions, *JustSpeak* can also save them hassles to click through several levels to find commonly used functions by chaining commands. After releasing *JustSpeak*, we have observed many chains of commands developed by and shared between users. For instance, the switch of TalkBack is hidden in the last page of a 3-layer setting app, one of our users created a shortcut of the page and was able to launch the shortcut and turn on/off TalkBack with one sentence. Many other users adapted to his strategy and appraised that it was much easier.

From the user feedback we collected, we discovered another unexpected user group which also benefited from *JustSpeak*. One of our users pointed out that *JustSpeak* is also a great tool for “*anyone with dexterity issues*”. It is often a hard task for them to point to objects accurately with fingers on multi-touch screens, *JustSpeak* can alleviate their burden to move the device so that they can just put it at a place where they can activate *JustSpeak*, and use spoken commands to perform interactions.

Discussion

As discussed above, we have released *JustSpeak* on Play Store as a beta version³. Since October 2013 we have had hundreds of users, they have given us many meaningful suggestions⁴, the overall feedback is positive.

As an accessibly service, *JustSpeak* is dependent on the labeling of on-screen controls. Unfortunately, application developers often assign lower priority to accessibility than to other features. Their carelessness often causes issues and obstacles for users with disabilities. We believe that by designing and promoting *JustSpeak* and other services that benefit larger user groups, we can make developers more aware of accessibility needs.

System designs similar to *JustSpeak* can be easily applied on platforms other than Android. The essential support required is a delegate layer between system and application levels which plays the same role that accessibility APIs has in *JustSpeak*. With users’ permissions, this layer needs to be capable of invoking interface and system commands. It also needs to monitor interface contents and events and pass updates to registered applications. For operating systems on which this layer is present or can be built, the same three modules architecture can be used to enable universal voice control.

³Download *JustSpeak*, <http://goo.gl/9Pg4DN>

⁴*JustSpeak* email threads, <http://goo.gl/l9cBAF>

FUTURE WORK

We have updated *JustSpeak* several times to add new features and fix issues according to user feedback. We plan to continue listening to our users and conduct systematic studies of their usage behaviors. In addition to maintaining and improving *JustSpeak* to increase its usability, we would like to reach out to more users by adding more language supports as well.

We are also exploring other techniques to activate *JustSpeak*, most recently some mobile devices have been equipped with always-listening hot words recognition, for example, on Motorola X, users can always launch Google voice search by speaking “*OK, Google*”. We are looking into the possibility of using same technology in *JustSpeak* to enable real voice-only interaction.

CONCLUSION

In this paper, we have presented the system designs and use cases of *JustSpeak*, a universal voice control assistant on Android. The contributions of *JustSpeak* are twofold. First, it provides enhancements to all applications running on a mobile system by synthesizing commands set from on-screen context. Secondly, it supports chaining of multiple commands which enables more natural and seamless interaction experience. As an application released to public, *JustSpeak* can benefit large number of users with universal eyes-free and hands-free voice control of their mobile devices. User feedback shows *JustSpeak* is welcomed by real world users. Its framework may help to shape future voice control devices.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation Awards IIS-1149709, IIS-1116051, and by Google.

REFERENCES

1. Apple Inc. VoiceOver. <http://goo.gl/crav6r>
2. Apple Inc. Siri. <http://www.apple.com/ios/siri/>
3. Azenkot, S., Nicole B. L. Exploring the use of speech input by blind people on mobile devices. In *Proc. of ASSETS 2013*.
4. Bigham, J. P., Craig M. P., and Richard E. L. WebAnywhere: a screen reader on-the-go. In *Proc. of W4A 2008*, pp. 73-82.
5. Borodin, Yevgen, et al. More than meets the eye: a survey of screen-reader browsing strategies. In *Proc. of W4A 2010*.
6. Google Inc. Google Now. <http://goo.gl/ohPzMt>
7. Google Inc. TalkBack: An Open Source Screenreader For Android. <http://goo.gl/D3ck4M>
8. Guerreiro, T., et al. Towards accessible touch interfaces. In *Proc. ASSETS 2010*.
9. JAWS Screenreader. <http://goo.gl/XwWVNI>
10. Kane S., Jeffrey P. B., and Jacob O. W. Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proc. ASSETS 2008*.
11. Lei X., Andrew S., Alexander G., and Jeffrey S. Accurate and Compact Large Vocabulary Speech Recognition on Mobile Devices. *INTERSPEECH*, 2013.
12. Mackenzie, I. S. Fitts’ law as a performance model in human-computer interaction. (1992).
13. Raman T.V., Charles L. C., Tim C. Leveraging Android accessibility APIs to create an accessible experience. *Google I/O*, May 2011.
14. Vigo M., and Simon H. Coping tactics employed by visually disabled users on the web. *International Journal of Human-Computer Studies* 71.11 (2013): 1013-1025.
15. Voice Command Device. <http://goo.gl/x5HjvM>