Syracuse University

SURFACE

Electrical Engineering and Computer Science - Technical Reports

College of Engineering and Computer Science

3-29-2011

Voice Commands to Control Recording Sessions

J. Marty Goddard

EE CS CE
EECS@syr

Department of Electrical Engineering
and Computer Science

Technical Report

SYR-EECS-2011-03 March 29, 2011

Voice Commands to Control Recording Sessions

J. Marty Goddard jmgoddar@syr.edu

Advisor: Jim W. Fawcett

ABSTRACT: In this report, the music recording workflow is described, with support for voice commands. Natural command grammars are proposed, allowing the user to name items, and issue commands on items identified by name. Recognition accuracy is examined within the contexts of single-phrase commands, and of versatile command grammars which enable the referring to items by name.

KEYWORDS: Voice Commands, Keyword Spotting Recognition, Dictation Speech Recognition, Named Entity, Recording Workflow

Syracuse University - Department of EECS,
4-206 CST, Syracuse, NY 13244
(P) 315.443.2652 (F) 315.443.2583
http://eecs.syr.edu

Voice Commands to Control Recording Sessions

J. Marty Goddard
Department of Computer Science and Electrical Engineering
Syracuse University
Syracuse New York USA
jmgoddar@syr.edu

Abstract -- In this report, the music recording workflow is described, with support for voice commands. Natural command grammars are proposed, allowing the user to name items, and issue commands on items identified by name. Recognition accuracy is examined within the contexts of single-phrase commands, and of versatile command grammars which enable the referring to items by name.

Keywords – Voice Commands, Keyword Spotting Recognition, Dictation Speech Recognition, Named Entity, Recording Workflow.

I. INTRODUCTION

Using voice commands to control music recording has been in the minds of recording musicians for many years [11], and yet there are no published studies on its effectiveness for this purpose. The primary issue with music recording is that it must be a hands-free process, allowing the musicians to perform on their instruments. Reaching for the mouse is detrimental to workflow, both practically and creatively. With a baseline of voice commands to control recording transport functions (Play, Stop, Record, etc.), a user could be more productive.

A set of Single-Phrase commands was implemented, and it was soon discovered that this was not sufficient for allowing hands-free workflow, as the user would have to frequently select tracks which were not visible onscreen. The operation of assigning names to tracks, by which they are subsequently referred, became an important addition to basic commands.

Naming tracks requires large vocabulary dictation speech recognition, since a name cannot be limited to a specific set. Once a name is correctly assigned, the name can be added to the speech dictionary, and loaded into the grammar as choice among other names for a particular command, to improve recognition accuracy.

II. BACKGROUND INFORMATION

Controlling music software by voice commands was described as early as 1990 [11]. However, speech recognition accuracy and flexibility prevented its widespread adoption in music recording. It was just easier to use the keyboard and mouse to arm tracks for recording and controlling the track functions of Mute, Solo, Pan, etc.

Hands-free operation by voice commands is used by radiologists in making X-rays [5], and automobile drivers interacting with GPS navigation devices [3] and smart phones [4]. It is also allowed computer programmers with a disability such as carpal-tunnel syndrome to continue to write code hands-free [7].
And this paper shows how it can help musicians record themselves.

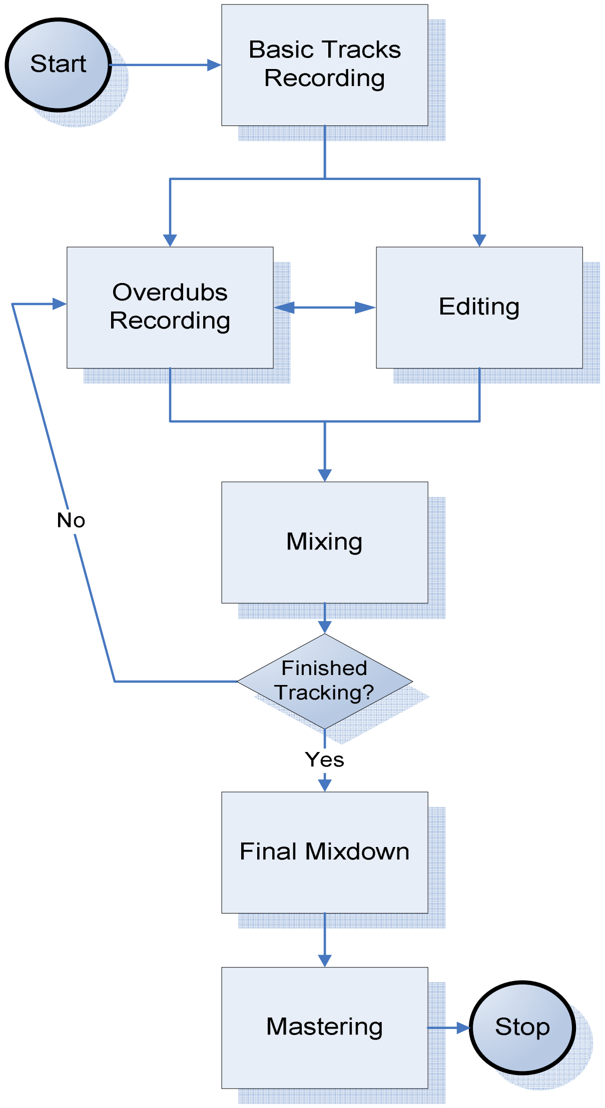


Figure 1- Recording Workflow

Figure 1 shows the process of recording music. If an instrumentalist such as a pianist or guitarist wants to make a recording using a computer, they must reach for the mouse or keyboard. If the hands-free computing paradigms used for other applications could be used for audio recording, then the musicians would not have to reach away from their instruments to start and stop a recording. A producer wouldn't need a tape operator to record an ensemble.

This paper focuses on the recording of Overdubs, selecting the best and alternate takes, naming them, and adjusting their Volume, Pan, and Solo/Mute states. Figure 2 shows a detailed flow diagram of this sub-process of the entire music recording workflow.

Audio playback is for the purpose of deciding whether or not to keep a recorded track, and what to include in the mix that is monitored during the recording process.

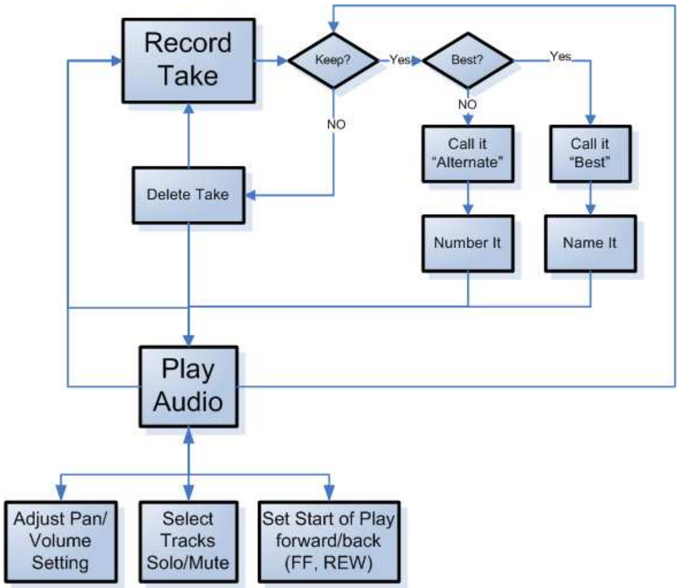


Figure 2- Workflow of Recording Overdubs.

III. Experimental Software System

Audacity [1] is chosen for audio recording and Microsoft Speech Recognition [2] is used for voice command detection. These choices are based on the following benefits:

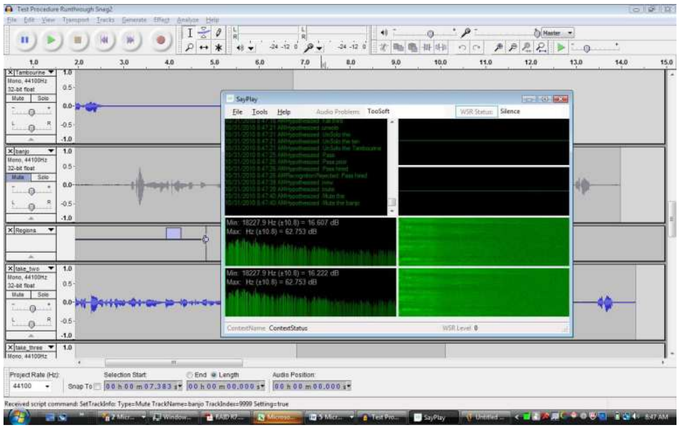
1) Audacity is open source: This means that changes can be made to surmount obstacles to progress, if necessary.

2) Audacity supports remote scripts [1]. Remote scripting allows the voice command event handler to be a separate program from the audio recording program, minimizing the invasiveness of any changes that have been necessary.

3) The SayPlay program was written for this research, and it provides the voice command handling functions by utilizing the speech recognition functions made available in the .NET environment. Within SayPlay, voice commands can be fielded, formatted, and dispatched to the audio recorder software (Audacity) described above.

Figure 3 shows the SayPlay voice command handling program running in the foreground, with a list of recognized phrases, while Audacity is running in the background.

It is perhaps more convenient to leave SayPlay in the background, unless the status of the speech recognition engine must be monitored, for example, if it seems to have stopped working.



Min: 18227.9 Hz (±10.8) = 16.607 dB
Max: Hz (±10.8) = 62.753 dB

Min: 18227.9 Hz (±10.8) = 16.222 dB
Max: Hz (±10.8) = 62.753 dB

Received script command: SetTrackInfo: Type=Mute TrackName=banjo TrackIndex=9999 Setting=true

Figure 3- SayPlay and Audacity.

IV. Flexible Grammar Structures

The overdub recording workflow shown in Figure 2 is supported by the flexible grammars shown in Figures 4 and 5, in addition to some single word commands such as "Record", and "Stop".

Note that in the diagrams, the Trapezoid (keystone) shape represents a Choice of one box within. Curly braces {} represent an Optional word. Thus, referring to Figure 4, the command to assign a track name can be spoken as "Name this track 'piano'", or "Name the recorded track 'piano".

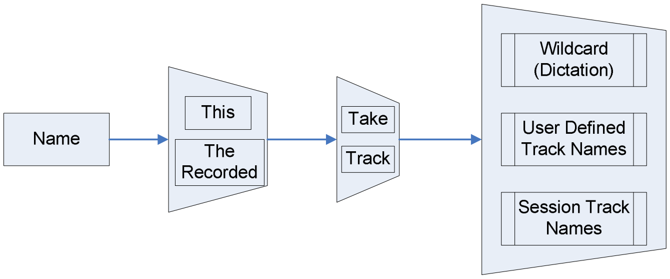


Figure 4- Flexible Grammar for Creating Names for Tracks.

Using the assigned track names to control track parameters, such as Solo, Mute, and the Pan setting, is handled by a grammar structure shown in Figure 5.

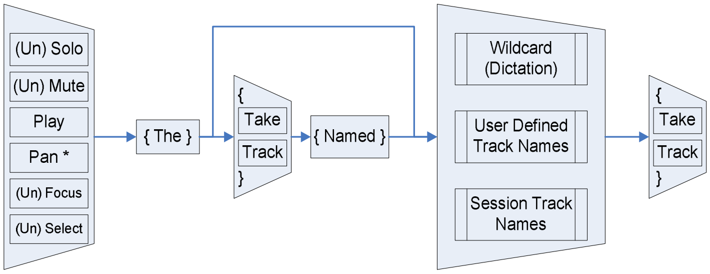


Figure 5- Flexible Grammar to Refer to Tracks by Name.

Issuing a command on a named track can be accomplished by saying "Mute the track named 'piano'", or "Mute the piano track", or "Mute the piano", or merely "Mute piano". Note that the "Pan" command, requires a setting to be stated at the end (not shown in the diagram), as follows: "Pan the piano {hard, medium, soft} {left, center, right}".

V. Improving Recognition Accuracy

Improving accuracy is focused on commands that involve names: both assigning names and using names.

A. Techniques Used When Assigning Names

Three methods help improve accuracy when assigning names. They are: 1. **Elaboration**, 2. **Quoting a phrase**, and 3. **Spelling out the name.** Note that these 3 techniques can also be used subsequently in the commands that refer to items by their assigned names.

1. Elaboration

Elaboration allows the user to add descriptive words to the name, providing the speech recognition engine's N-Gram analysis with extra keywords that can tip the balance toward the correct response. Elaboration is made using the phrases "Like" or "As In". Examples of elaboration are "Name this track 'bass' as in bass guitar", or "Name this track 'Theremin' like the science fiction movie sound". The name string (including the elaboration) is recognized using Dictation speech recognition, because it the user can use any word or phrase as a name. A simple rule is applied to any name string returned from the speech recognition engine containing "Like" or "As In". Applying the rule simply truncates the string, starting with "Like" or "As In". It and everything after is removed, leaving only the desired name.

2. Quoting a Phrase to use as a Name

Elaboration, as described above, deletes the words "Like" and "As In" from a name. It is still possible to create a name which contains "Like" or "As In", by using "Quote/Unquote" or "As Follows" to specify an entire phrase. For example, if the user wants to name a track "Scream like a banshee", they would say: "Name this track as follows: 'Scream like a banshee'", and the entire phrase "Scream like a banshee" becomes the new name. Alternately, they could say: "Name this track quote scream like a banshee, unquote", and the string is parsed to extract the desired name. It is not possible, however, to create a name which contains the words "quote", "unquote" or "as follows". This is just as well.

3. Spelling it Out

Spelling out the name is a last-resort technique for getting a new name to stick. The phrasing of the command is slightly unnatural, in the interest of consistency with the other naming commands. It is as follows: "Name this track spelled W O W". It is possible to say "apostrophe" in contractions, and space between words.

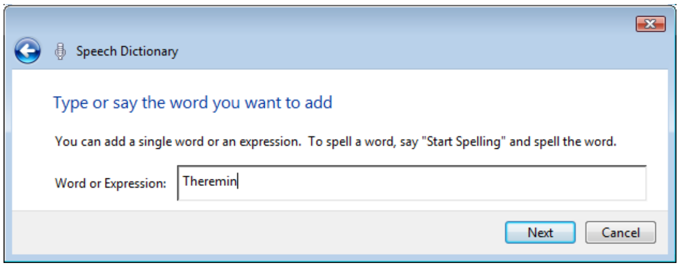
3 techniques for simply getting the correct name to stick have been described. Recognizing it correctly afterward, when issuing commands, can be another matter, however. The user can refer to a track by continuing to elaborate, or to spell it out again, but that becomes tiresome. Next, I describe 3 techniques for improving recognition accuracy when referring to tracks, once names are correctly assigned.

B. Techniques Used to Refer to Named Entities

Once a name is correctly assigned to an item, the user should expect that subsequent references to it will work. This is not always the case, requiring the user to resort to the same techniques as used in assigning the name, described above. Needless to say, having to spell out a track name that was already assigned by spelling it out can be tiresome. There are actions that can be taken to greatly improve the recognition accuracy of commands which refer to named entities. These actions are: 1. **Add the name to the Speech Dictionary**, 2. **Prevent dictation of mistaken words**, and 3. **Load the name into the grammar** so that it is among the choices available to the recognizer, rather than having to rely on dictation speech recognition.

1. Add a Name to the Speech Dictionary

This technique is very helpful when words, such as "Theremin" do not seem to be present in the dictionary. The ability to add a word to the dictionary is a standard part of the Windows Speech Recognition feature (in Windows Vista and Windows 7).



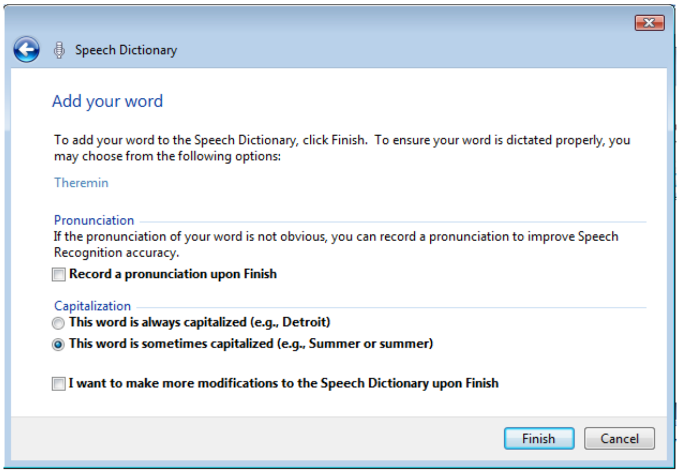
Type or say the word you want to add

You can add a single word or an expression. To spell a word, say "Start Spelling" and spell the word.

Word or Expression: Theremin

Figure 6- Adding a word to the Windows Speech Dictionary.

Near the end of the process, the user can provide a proper pronunciation of the word, by recording a spoken example.



Add your word

To add your word to the Speech Dictionary, click Finish. To ensure your word is dictated properly, you may choose from the following options:

Theremin

Pronunciation

If the pronunciation of your word is not obvious, you can record a pronunciation to improve Speech Recognition accuracy.

☐ Record a pronunciation upon Finish

Capitalization

○ This word is always capitalized (e.g., Detroit)

● This word is sometimes capitalized (e.g., Summer or summer)

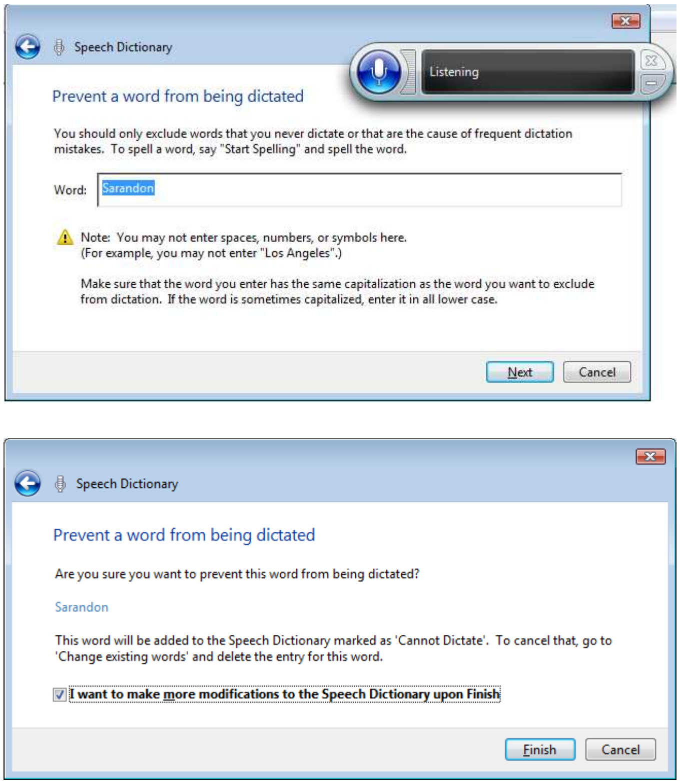
☐ I want to make more modifications to the Speech Dictionary upon Finish

Figure 7- Record a pronunciation of new word added.

Ideally, this would be done automatically whenever a new name is not already in the dictionary, and has been deemed correct. However, we wouldn't want to add every mistaken name to the dictionary; only once the correct name is assigned. It is for this reason that the user must issue the command to get all the assigned names to load them into the actual grammar structure that the engine is comparing phrases to.

2. Prevent a Name from Being Recognized

Sometimes a particular name is repeatedly misrecognized as a different word. For example, even after naming a track “Wow” and adding “Wow” to the Speech Dictionary, it gets repeatedly mistaken for the word “While”. The user can manually request that “While” not be recognized, so that “Wow” is correctly recognized. Figures 8 and 9 show how to prevent a word being dictated. Ideally this would be done (temporarily) when the user says “Wrong” after repeating a command using the same name, and getting the same wrong name.



Prevent a word from being dictated

You should only exclude words that you never dictate or that are the cause of frequent dictation mistakes. To spell a word, say "Start Spelling" and spell the word.

Word: Sarandon

Note: You may not enter spaces, numbers, or symbols here. (For example, you may not enter "Los Angeles".)

Make sure that the word you enter has the same capitalization as the word you want to exclude from dictation. If the word is sometimes capitalized, enter it in all lower case.

Prevent a word from being dictated

Are you sure you want to prevent this word from being dictated?

Sarandon

This word will be added to the Speech Dictionary marked as 'Cannot Dictate'. To cancel that, go to 'Change existing words' and delete the entry for this word.

☑ I want to make more modifications to the Speech Dictionary upon Finish

Figure 8- Prevent a word from being dictated.

3. Adding the Name to the Loaded Grammar

Once the user has successfully named a track, it helps to add the name to the grammar for track commands, so that the name doesn’t have to be recognized by the Dictation Recognition engine. This should be done under user control, rather than whenever creating a name, because the name created isn’t always correct. Only once names are correct, would a user wish to load them into the grammar.

VI. Experimental Results

A. Single Word/Single Phrase Commands

Commands for which Audacity provides a scripting interface are handled by SayPlay and sent to Audacity if the recognition event has a Confidence value which is above a threshold. Figure 9 shows many of these commands uttered by an experienced user, along with the average Confidence value returned by the Windows Speech Recognition engine. Most commands averaged well above the threshold for deciding to act on the command (0.93).

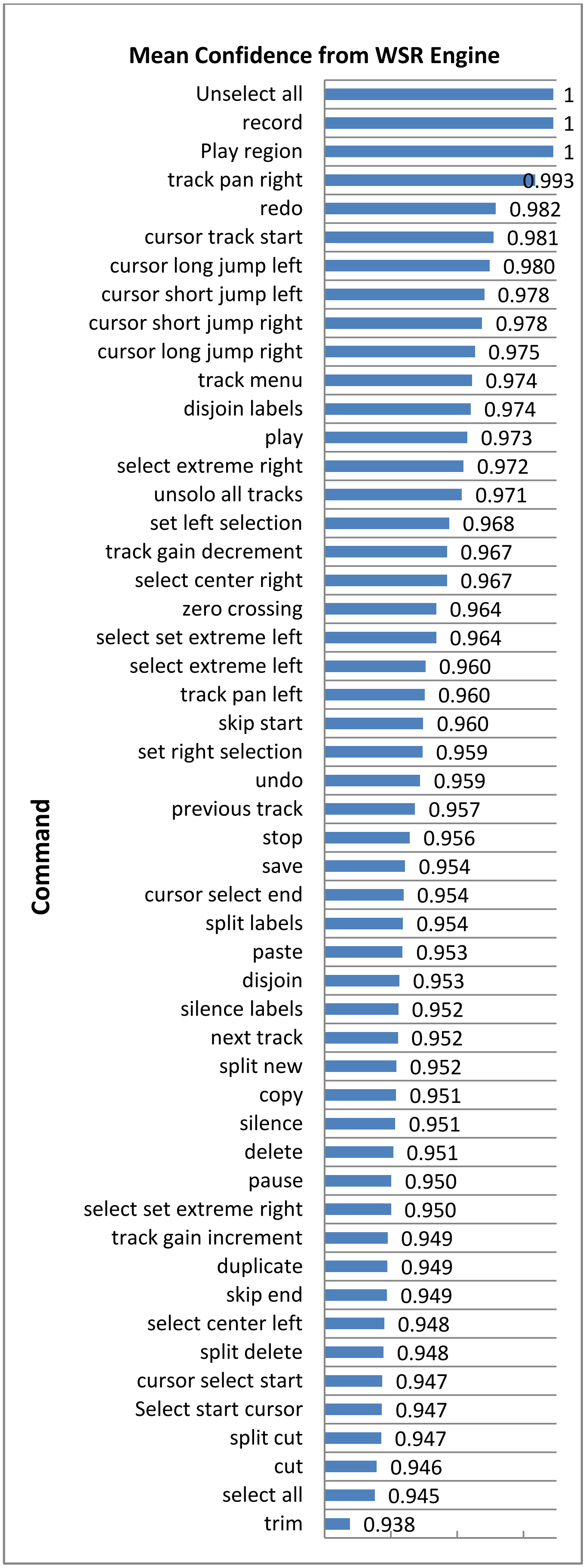


Figure 9 – Average Confidence for Basic Commands

B. Tricky or Difficult Names

During the course of exploration for this research, several names were found to be particularly difficult to recognize using the Dictation Speech Recognition required for the open-ended nature that any word can be used for a name. Two reasons for this difficulty are: 1. The name is sometimes simply not in the Speech Dictionary, either because it is rather esoteric (jargon or slang), or it is an entirely made-up word, and 2. The names were frequently confused with similar sounding names. In the case of “Wow”, both are true (it is slang and easily confused with “While”). There were some surprises in this regard too, as some non-words were reliably recognized: “Whoosh” and “Vox” (an abbreviation of “vocals”).

Figure 10 shows cumulative successes for some tricky names, before and after adding the word to the Speech Dictionary.

Prior to adding Crotales, it was frequently misrecognized as “Croat Olives”, and “Wow” was frequently misrecognized as “While”. After adding “Wow” and ”Crotales” to the Speech Dictionary, both were successful for the remainder of the experiment.

Gambales is a made-up name, and thus, was not recognized until after it was added to the Speech Dictionary. However, success thereafter was somewhat erratic, having low recognition confidence on several occasions.

Further testing of “While” showed continued problems with low confidence, or being confused with the newly added “While”.

Recognizing these words is very dependent upon pronunciation. As slight variation can cause misrecognition.

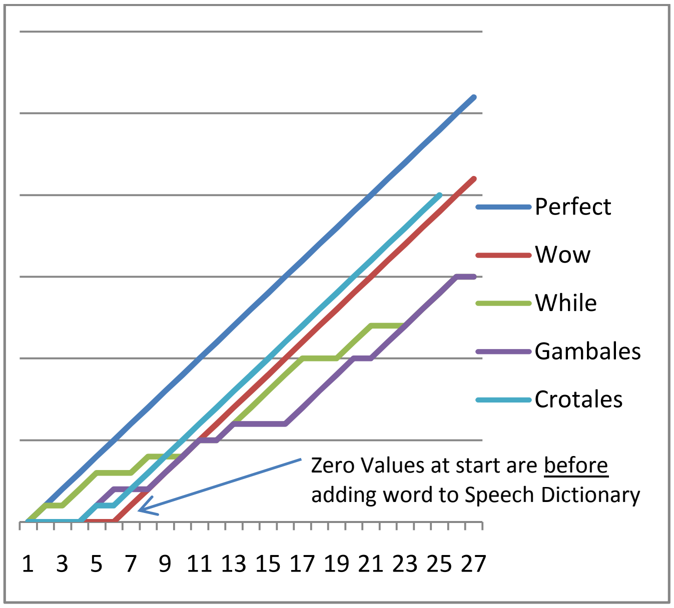


Figure 10 - Cumulative Success Rate of Track Commands, Before and After Adding Name To Speech Dictionary.

C. Result of Adding Names to Dictionary, Preventing Recognition of Mistaken Names, and Loading Names into a WSR Grammar

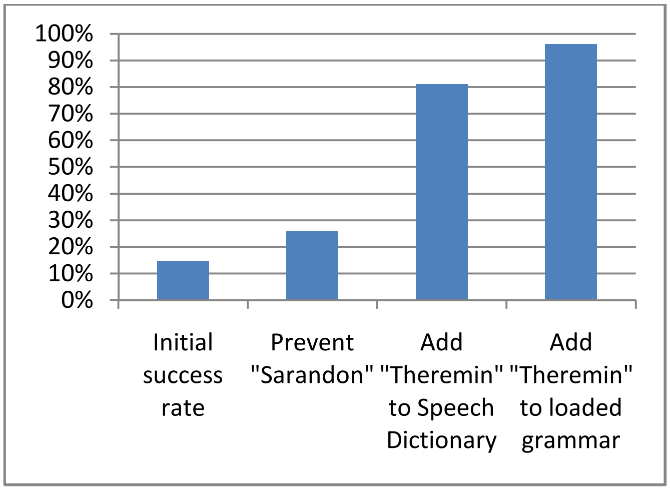


Figure 11 – Recognition accuracy of track commands for “Theremin”, based on employing successive techniques.

Each technique for enhancing recognition of track names was employed in succession on the word “Theremin”, resulting in a jump in recognition accuracy as it was employed, as shown in Figure 11 above.

VII. Conclusion

I show in this paper that voice commands can be reliable and effective in controlling audio recording software. This is particularly true when techniques are employed to flexibly allow audio tracks to be named, so that subsequent commands can be issued referring to them by name. Performance improvements were shown as a result of adding track names into the Windows Speech Dictionary, preventing recognition of incorrect names, and by making the track names part of the command grammar that is loaded into the Speech Recognition Engine. This makes the names part of the set of possible commands, rather than relying on dictation speech recognition for recognizing the names.

References

[1] Audacity Scripting Interface online WIKI/FAQ, http://manual.audacityteam.org/index.php?title=Scripting

[2] Brown, Robert; 2006 “Exploring New Speech Recognition and Synthesis APIs in Windows Vista”. MSDN Magazine, Microsoft Speech Recognition. MSDN online resources: http://msdn.microsoft.com/en-us/magazine/cc163663.aspx

[3] Chang, J. 2009, "Usability evaluation of a Volkswagen Group in-vehicle speech system", Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications, Automotive UI 2009, , pp. 137.

[4] Cohen, J. 2008, "Embedded speech recognition applications in mobile phones: Status, trends, and challenges", 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, March 31,2008 - April 04Institute of Electrical and Electronics Engineers Inc, Las Vegas, NV, United states, pp. 5352.

[5] Ernst, R. 2001, "Combining speech recognition software with digital imaging and communications in medicine (DICOM) workstation software on a Microsoft windows platform", Journal of Digital Imaging, vol. 14, no. 2, pp. 182.

[6] Gorniak, P. 2003, "Augmenting user interfaces with adaptive speech commands", ICMI'03: Fifth International Conference on Multimodal Interfaces, pp. 176.

[7] Hubbell, T.J., Langan, D.D. & Hain, T.F. 2006, "A voice-activated syntax-directed editor for manually disabled programmers", *Eighth International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2006, October 23,2006 - October 25* Association for Computing Machinery, Portland, OR, United states, pp. 205.

[8] Lee, K. 1988, "Large-vocabulary speaker-independent continuous speech recognition: The SPHINX system", ProQuest Dissertations and Theses, PhD. Dissertation, Carnegie-Mellon University

[9] Rudnicky, Alexander I. 1989 *"The Design of Voice-Driven Interfaces*", Human Language Technology Conference archive. Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics Morristown, NJ, USA 120 - 124.

[10] Nakano, T. 2008, "Flexible shortcuts: Designing a new speech user interface for command execution", 28th Annual CHI Conference on Human Factors in Computing Systems, April 05,2008 - April 10Association for Computing Machinery, Florence, Italy, pp. 2621.

[11] Yavelow, Christopher, "Voice Navigation for the Macintosh Musician" Articulate Systems, Inc. 1990 http://chrisyavelow.com/articles/voicenavigation.pdf