Using an Open-Source Speech Recognition Software, CMU Sphinx's Pocketsphinx, to interface with a web-based Avatar for a Smart Home

Kira Curry
Computer Science
Rhodes College
Memphis, TN, 38112
Email: curkn-17@rhodes.edu

Dr. Monica Anderson
Assistant Professor
Department of Computer Science
University of Alabama
Email: anderson@cs.us.edu

***Abstract*— Speech recognition applications are useful for enabling convenient interaction with technology, especially for the physically challenged. This project was designed to make daily tasks easier by creating a speech-recognition app using the open-source speech software, Pocketsphinx (compared to Windows Cortana and IBM Bluemix) that connects to a web-based avatar that interfaces with a smart home. Fifty voice samples were collected to test Pocketsphinx's accuracy. Although, it had the lowest accuracy, it is more platform independent than Cortana and Bluemix.**

***Index Terms*—Smart Home, Speech Recognition, Pocketsphinx**

I. Introduction

As people grow older and live longer, the availability of long-term healthcare facilities is expected to decrease and there will not be enough facilities to support the growing aging population [1]. This highlights the need for an alternative way to care for the elderly population aside from long-term healthcare facilities. One idea that is gaining popularity is the idea of "aging in place". This is a practice that would allow an individual to remain in their homes, neighborhood, and community with some level of independence as they grow older. Smart Homes are a new approach to this issue that allows the elderly individual(s) to age in place while using technology to assist and monitor their living situation [1]. Smart Homes are seen as an enhancement to home automation by incorporating an element of artificial intelligence to allow for more complex functionality and guaranteed security and safety [2].

Another field of study that has gained traction in recent years is the use speech recognition software, such as Siri and Cortana in smart phones. This technology allows for new ways to interact with mechanical technology. It also provides a more natural and simpler interaction with traditional push-button or touch-screen interfaces which to many especially the elderly population find complicated and difficult to use. Integrating speech recognition and natural language processing software into Smart Homes will help to make them more user-friendly for the elderly who are possibly visually or physically impaired.

II. Related Works

As a whole there is not much research in the integration of Speech recognition software with Smart Homes, and the research that does exist usually involve very expensive systems that would be hard for the average person to afford.

A. Smart Homes

In the field of home automation, Smart Home research mostly has been focused in four areas: care for elderly, energy efficiency, comfort and entertainment, and safety and security [2]. Research in care for the elderly generally focuses on healthcare and monitoring. Some services that Smart Home research with regard to the elderly aim to provide is health monitoring, emergency assistance prevent injury, fall detection, medicine reminders, administration, and monitoring and more [1].

B. Voice Recognition

In the field of voice recognition software, there are many commercial speech recognition systems and open source automatic speech recognition system for professional and individuals to use depending on their needs [3]. For open source models, they mostly follow the format of training an acoustic model with Gaussian mixture models and depending on how they implement the model, dictates the difference in performance accuracy of the model [3].

III. Approach

This project is different from previous approaches to interaction with Smart Home Environments because it aimed to use speech recognition software in order to facilitate the interaction between the elderly home owners and the Smart Home (Figure 2). This approach specifically uses an open source speech recognition software, CMU Pocketsphinx, in this interaction as an alternative to corporately owned speech recognition software such as Microsoft Cortana and IBM Bluemix, and paid speech recognition software. This project

uses a command-based UNIX app to interact with the web-based avatar (in Figure 1) which when completed will interface with an A.I. system that controls the Smart Home. The app is based in the UNIX command line, written in python, and utilized Pocketsphinx along with Gstreamer and gtk+ for audio input and the visual feedback. To increase the accuracy of the speech recognition an FSG grammar is used. This app is very flexible and can be used to easily build a web-based app.

Fig 1. The web-based avatar [4].

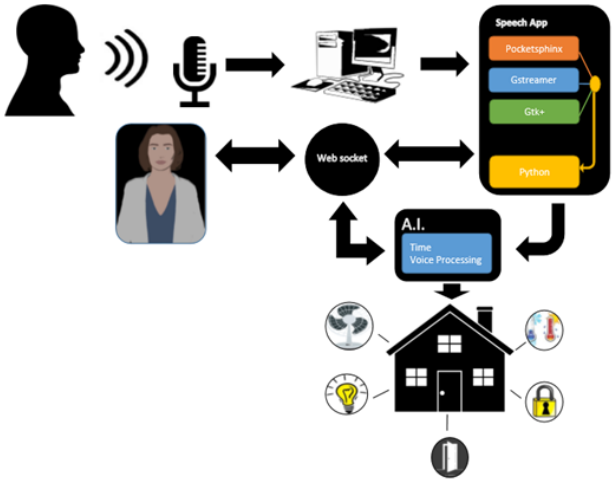

Fig 2. This is a diagram of how the speech app takes spoken word, translates it using Pocketsphinx, sends it to the web-based Avatar that communicates with the smart home.

IV. Experiment

A. Methods

Three students collected a total of fifty voices from predominantly the Eastern and Southern parts of the United States, the majority of which were female. Each of the fifty voices were recorded saying five common phrases used when interacting with a Smart Home to test the accuracy of the speech recognition software. The five phrases were:

- Phrase 1: “Turn on the light”
- Phrase 2: “Turn off the light”
- Phrase 3: “Turn off/on the bedroom light”
- Phrase 4: “Open the door”
- Phrase 5: “What is the status of the security system?”

The phrases one, two, and four are simple phrases to tests the rate of recognition of simple commands containing an action word such as “on”, “off”, or “open” and the intended appliance such as “light” or “door”. Phrase three is the same except that is incorporates the idea of aliasing which is specifying which object in the house to apply the action to (i.e. the “bedroom light” instead of just the “light”). Phrase five is unlike the other four phrases because it is not command but is instead an inquiry of status.

The voice were collected via voice recordings on smart phones (only iPhones and Androids were used). These recordings were then played and recognized by Pocketsphinx and it was documented if the phrases were recognizable (Yes or No) and how many trials until it was recognized (5 trials max until it was listed as unrecognizable) to calculate the accuracy of the software.

In other related experiments, by other researchers, the same process was executed with other speech recognition software: Microsoft Cortana and IBM Bluemix.

B. Calculations

- The accuracy of each phrase
  - The total number of phrase recognized/ Number of participants
- The average number of trials per phrase
  - Trial for yeses + trials for noes/Number of Participants*Number of phrases
- The average number of trials per person
  - Trial for yes’ + trials for no’s/Number of Participants
- The average number of phrase recognized per person
  - Number of yes total/Number of Participants
- The accuracy per phrase
  - Number of yeses per phrase/Number of people
- The overall accuracy of the speech recognition software
  - Number of yeses/ Total number of phrases

C. Results

TABLE I. Voice Recognition Test: Pocketsphinx

Participants	Phrase 1: “Turn off the Light.”	Phrase 2: “Turn on the Light.”	Phrase 3: “Turn on/off the bedroom light.”	Phrase 4: “Open the door.”	Phrase 5: “What is the status of the security system?”
1	Yes (1)	Yes (1)	Yes (1)	Yes(1)	Yes (1)
2	Yes (3)	Yes (1)	Yes (1)	Yes(1)	Yes (1)
3	Yes (1)	Yes (1)	Yes (1)	Yes(1)	Yes (1)
4	Yes (1)	Yes (1)	No	Yes(3)	No

5	No	No	Yes (2)	No	Yes (2)
6	Yes (2)	Yes (1)	Yes (1)	Yes(1)	No
7	No	No	No	Yes(1)	Yes (1)
8	Yes (1)	Yes (1)	Yes (2)	Yes(1)	No
9	No	No	Yes (3)	Yes(1)	Yes (1)
10	Yes (2)	Yes (1)	Yes (3)	Yes(1)	Yes (3)
11	Yes (1)	Yes (1)	Yes (2)	No	Yes (1)
12	No	Yes (1)	No	Yes(3)	Yes (1)
13	Yes(2)	Yes(1)	Yes(1)	Yes(1)	Yes(1)
14	No	Yes (2)	Yes (3)	Yes(2)	No
15	Yes (2)	Yes (4)	Yes (1)	No	No
16	Yes(2)	Yes(2)	No	No	Yes(2)
17	Yes(4)	Yes(4)	Yes(5)	Yes(2)	Yes(1)
18	Yes(1)	Yes(3)	No	Yes(3)	Yes(3)
19	No	Yes(3)	Yes(3)	Yes(1)	Yes(2)
20	No	Yes(1)	No	Yes(2)	No
21	No	No	Yes(3)	Yes(2)	No
22	No	No	No	Yes(1)	No
23	No	Yes(1)	No	Yes(2)	No
24	No	No	Yes(1)	Yes(3)	No
25	Yes(1)	Yes(2)	Yes(1)	No	No
26	Yes(2)	Yes(1)	No	Yes(1)	No
27	No	Yes(1)	No	Yes(1)	No
28	Yes(2)	Yes(2)	No	No	Yes(1)
29	Yes(2)	Yes(2)	Yes(1)	Yes(1)	Yes(1)
30	No	No	No	No	No
31	Yes(1)	Yes(1)	Yes(4)	Yes(3)	Yes(2)
32	No	Yes(1)	No	Yes(1)	No
33	Yes(3)	No	Yes(4)	No	Yes(1)
34	No	Yes(2)	Yes(1)	Yes(4)	Yes(1)
35	Yes(4)	No	Yes(3)	Yes(1)	Yes(1)
36	Yes(3)	Yes(2)	Yes(2)	No	Yes(1)
37	Yes(3)	No	No	Yes(1)	Yes(1)
38	No	Yes(2)	No	Yes(2)	Yes(1)
39	No	No	No	Yes(1)	Yes(1)
40	Yes(3)	Yes(2)	No	Yes(1)	Yes(4)
41	No	Yes(4)	No	Yes(2)	Yes(1)
42	No	No	No	Yes(3)	No
43	Yes(1)	Yes(1)	Yes(1)	Yes(1)	Yes(1)
44	Yes(1)	Yes(1)	Yes(1)	Yes(1)	Yes(1)
45	No	No	No	Yes(1)	No
46	Yes(3)	Yes(2)	Yes(2)	Yes(1)	Yes(1)
47	Yes(1)	Yes(2)	No	Yes(1)	No
48	No	Yes(2)	Yes(1)	Yes(1)	Yes(1)
49	No	Yes(1)	Yes(3)	Yes(1)	No
50	Yes(1)	Yes(2)	Yes(1)	Yes(2)	Yes(1)

Fig 3. This table shows the results of the trials of the 50 participants when their voices were tested in Pocketsphinx.

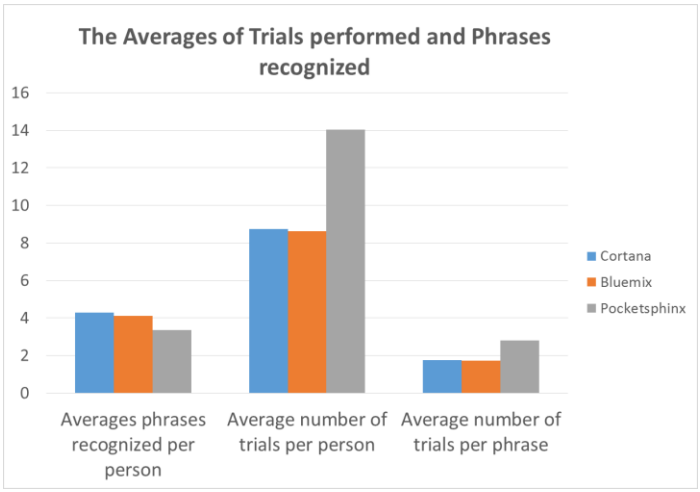

Fig 4. This graph show the averages of phrases recognized, trials per person and trials per phrase of Pocketsphinx (Grey) compared to Cortana (Blue) and Bluemix (Orange).

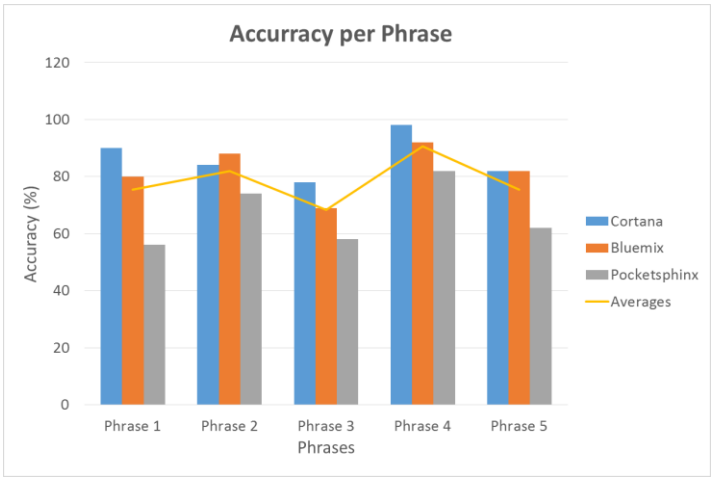

Fig 5. This graph shows the accuracy per phrase of Pocketsphinx in comparison to Cortana and Bluemix.

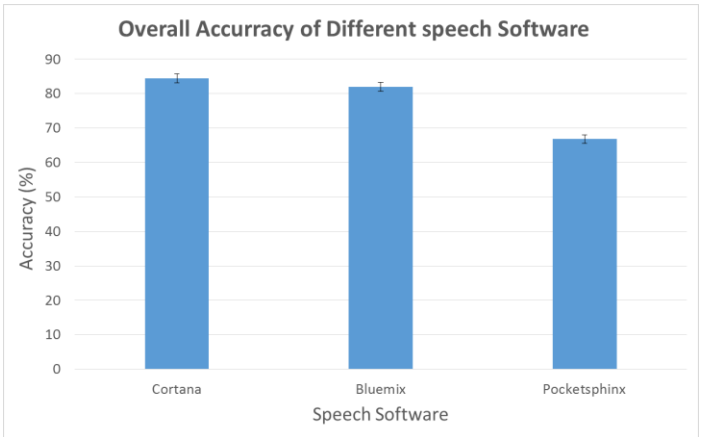

Fig 6. This graph depicts the average accuracy of Pocketsphinx compared with Cortana and Bluemix overall.

V. Analysis

VI. Between the three different voice recognition software, Pocketsphinx was the least accurate overall as shown in Figure 6. There was no significant difference between Cortana and Bluemix in overall accuracy although Cortana had a slightly higher average (Figure 6). Pocketsphinx required more trials per person and per phrase than Cortana or Bluemix and it also had a lower average of phrases recognized than both as shown in Figure 5. Table 1 shows the results of the trials of the fifty participants with whether or not they were able to recognize the phrase and how many trials it took. .(Yes or No if phrase is recognized, and if Yes, the trials it took to be recognized are in parenthesis).

VII. Of the five phrases, Pocketsphinx consistently scored lower in accuracy while Cortana and Bluemix fluctuated between which was more accurate except on phrase 5 where they were not significantly different.

VIII. Conclusion

The goal of this project was to create a speech application that would interface with a web-based avatar to enable a conversational interaction between user and smart home in order to facilitate aging in place for our growing elderly population. The project focused on the open-source software Pocketsphinx, but it was compared with similar speech recognition applications using Windows Cortana and IBM Bluemix to see which software would work the most accurately and the most efficiently. Of the three, Pocketsphinx was the least accurate and took the most trials. Between Cortana and Bluemix there was only a 2% difference in accuracy and no significant difference in the number of trials it took.

Therefore, Windows Cortana is the best of the tested software, however, it is limited to Windows products and Windows technology, and the only way to develop for it is locally in Visual Studios. Bluemix is on equal grounds with Cortana for the most part. Although, it is cloud based, so everything is done in the cloud which makes it a more portable software since it can be used anywhere with an internet connection. However, in order to use Bluemix, you must have an IBM account and for the average user it cannot be run locally and must be done online. Pocketsphinx is not the best; however, it is open source and platform independent. It can be run of Windows, Mac, and Linux/Unix machines and it supports several different programing languages. Plus it can be easily incorporated into a web (as well as mobile, desktop, and local) app.

Future extensions to this project would be to complete the A.I. engine to enable NLP (natural language processing) in order to make the interaction between the user and Smart Home more conversational and less command based. Another idea would be to actually deploy and test the speech recognition interface into an actual Smart Home and see how well it performs outside of a lab setting. Also, one other extension to consider would be to incorporate voice recognition (identifying the owner of the voice speaking) with the speech recognition to enable the smart home to know who is speaking to it which would allow for more complex alias concerning possessive adjectives such as “my”, “his”, “her”, etc.

Acknowledgment

I would like to thank Denson Ferrell who programmed the voice recognition app using IBM Bluemix and provided the results from his trials. I would also like to thank Morgan Hood who programmed the voice recognition app using Windows Cortana and provided the results from her trials. Lastly, I would like to thank Dr. Monica Anderson the experience and opportunity to work with her research and helping me to learn many valuable skills.

References

[1] P. Cheek, L. Nikpour, and H. D. Nowlin, “Aging well with smart technology,” *Nursing administration quarterly*, vol. 29, no. 4, pp. 329-338, 2005.

[2] C. Badica, M. Brezovan, and A. Badica . "An Overview of Smart Home Environments: Architectures, Technologies and Applications,” in Proceedings of the Sixth Balkan Conference in Informatics, ser. BCI 2013, C. K. Georgiadis, P. Kefalas,and D. Stamatis, Eds., vol. 1036, pp. 78 - 85, 2013.

[3] Gaida, C., Lange, P., Proba, P., Malatawy, A., Suendermann-Oeft, D.: Comparing open-source speech recognition toolkits.

[4] Bickmore, T., Schulman, D. and Shaw, G. (2009) DTask & LiteBody: Open Source, Standards-based Tools for Building Web-deployed Embodied Conversational Agents Proceedings of Intelligent Virtual Agents, Amsterdam