

# PAL: Personal Assistant System Using Low-Cost Computer

Melanie Louisa Khong Fui Yee and Sarah Samson Juan

Faculty of Computer Science and Information Technology, University Malaysia Sarawak,  
94300 Kota Samarahan, Sarawak, MALAYSIA.  
louisa940321@gmail.com

**Abstract**—Automatic Speech Recognition (ASR) describes the ability of a computer to capture, identify and recognize the variety of human speech. It has been applied in many technologies such as personal assistant systems. Unfortunately, many personal assistant systems has been built in a way that may not always be disability-friendly and this causes the affected users whom are blind, disabled, illiterates and those who have physical limitations unable to enjoy the benefits of operating a computer. Hence, PAL is introduced. PAL is a personal assistant system built using low-cost device called the Raspberry Pi and open source voice-controlled software called Jasper. The functionalities of PAL includes searching for information on the internet, check for unread emails, schedule events in the calendar, manage a to-do list and translate texts through voice commands. Apart from that, a friendly graphical user interface (GUI) is also designed to display the output of each of the functional modules. Lastly, a number of tests are conducted to evaluate the performance and accuracy of the functional modules, GUI output display as well as response rate of the system. These tests include GUI output display test, user acceptance testing, the Command Success Rate (CSR) and Word Error Rate (WER) tests as well as response rate test. With the development of this project, it is hoped that PAL will be able to provide users with the benefits of using a computer in a more convenient and cost efficient manner.

**Index Terms**—Automatic Speech Recognition; Artificial Intelligence; Raspberry Pi; Jasper.

## I. INTRODUCTION

Speech recognition has been applied in many voice command devices throughout the years such as personal assistant systems. Personal assistant systems has been tremendously useful and convenient in many ways in our daily life especially to users who are blind, disabled, illiterate and have physical limitations such as visual impairments. Hence, the proposed personal assistant system, PAL will act as a personal assistant that enables the affected users to search for information on the internet, check for unread emails, schedule events on the calendar, manage a task list and translate texts to a different language through voice commands. The objectives of PAL are to develop the system with lightweight device, the Raspberry Pi [1] and open-source speech recognition engine called Jasper, to design a graphical user interface (GUI) that displays the output results for each of the modules as well as to evaluate the performance and accuracy of the customized modules. The Raspberry Pi is a low-cost, credit card-sized single-board computers developed with a purpose to promote the teaching of basic computer science in schools [2]. On the other hand, Jasper is an open source voice-control software introduced by Shubhro Saha and Charlie Marsh which is basically a Siri clone running on the Raspberry Pi [3].

The following section will be covering the topic on the concept of ASR and background of personal assistant systems. After that, Section III discusses on the installation and configuration of the Raspberry Pi and Jasper, the development of the functional modules and GUI layout as well as testing and evaluations. All the results and analysis obtained through the tests and evaluations conducted on PAL are discussed in Section IV. Finally, Section V concludes our work.

## II. BACKGROUND / MOTIVATION

### A. Concept of Automatic Speech Recognition

Figure 1 shows the standard concept of Automatic Speech Recognition (ASR) system that explains how speech recognition works. The speech recognition system will first analyse the input speech signal before decoding it to find the best match between the input speech and its corresponding word string.

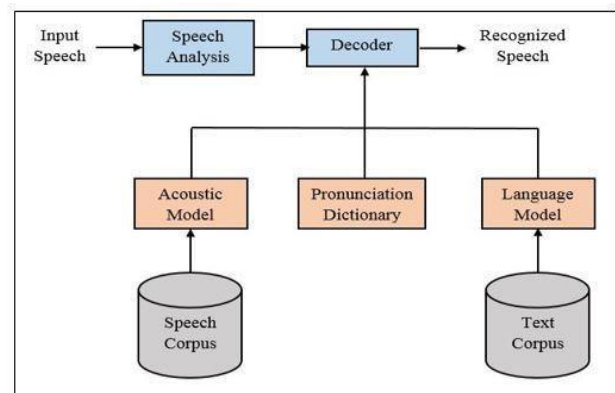


Figure 1: A standard concept of an ASR system

The three basic components, which are the acoustic model, language model and pronunciation dictionary, are responsible for finding the best match in the decoder. Acoustic models are statistical models used to estimate the probability that a certain phoneme has been uttered in a recorded input speech [4]. In other words, it converts sound to phoneme. The Hidden Markov Model (HMM) has been widely used to train acoustic models as it is an efficient algorithm for training and recognition [5]. After that, phonemes are converted into graphemes and to do so, a pronunciation dictionary is used to form valid words by combining various combinations of phonemes together. Several techniques have been used by researchers to create pronunciation dictionary with thousands of words such as rule based or statistical approach. For example, Juan and Besacier [6] built an Iban pronunciation dictionary for ASR through a semi-supervised Grapheme-to-Phoneme (G2P) bootstrapping strategy. Lastly, a language

model is required in ASR to convert words into a sentence. The simplest and most common language model used to assign probabilities to sentences and sequences of words is the  $n$ -gram model [7]. In speech recognition, the  $N$ -gram model is used to find the best possible estimate on the probability for the word to occur. It is assumed that the probability for the word depends on the  $n-1$  preceding words. The sequence of words with the highest probability will be chosen as the final output of the recognized speech.

### B. Personal Assistant Systems

A personal assistant system is a voice command system that incorporates the concept of ASR to act as an assistant and responds to whatever the user asks for. Just by using their voice, users can ask for current news, weather and many more. The intelligent system has been applied in robotics for creating ambient intelligent meeting rooms [8], studying human behaviours in human-computer interaction [9] and improving human-robot relationship by investigating the interaction between child and social robot [10]. In mobile devices, there are several personal assistant systems such as Apple's Siri, Microsoft's Cortana and Google's Google Now. Users can search for information and manage their tasks using these applications. Eventhough these applications come free with mobile device, customising commands are not available and the applications can support limited languages.

Raspberry Pi is an affordable, tiny computer that can be operated by connecting the device with an Ethernet cable to a laptop or by connecting the device with a computer monitor, keyboard and mouse. It has been used to make low-cost microcomputer for students, explore programming languages such as Python and create Science, Technology, Engineering and Mathematics (STEM) projects. Furthermore, the Raspberry Pi has been used in smart home projects, where it works as a server to collect values that are read from sensors of home appliances and it communicates with mobile phones to send reports [11, 12]. Besides that, the tiny computer has also been used for developing a voice-controlled wheelchair [13]. In the wheelchair project, authors used a speech recognition module, microcontroller and ultrasonic sensors. These solutions can help people with disabilities to take control of home applications or move about using voice commands. By default, Raspbian OS, the operating system for Raspberry Pi, does not come with a speech recognition system. Visually or physically impaired people will face difficulties interacting with the device. It needs a speech recognizer to assist them to perform tasks such as checking emails, take notes or search information from the Internet. There are several speech recognition systems that can be used in Raspberry Pi such as Jasper.

Jasper is an open source voice-control software that works on Raspberry Pi [3]. The software allows developers to customize commands by creating modules. Besides that, it supports speech recognition engines like PocketSphinx [14] by Carnegie Mellon University, which allows users to use their own acoustic models. In other words, Jasper can support any language as long as software developers have the language data to train models. Thus, we develop our system using Jasper in Raspberry Pi as it allows us to build functional modules for customizing voice commands, which could assist disabled people to use the low-cost computer.

## III. METHODOLOGY

Throughout the development of PAL, a number of procedures were done namely the preparation of the required hardware and software components, the configuration of the Raspberry Pi and Jasper, the development of the functional modules and graphical user interface (GUI) as well as the testing of the system.

### A. Hardware Requirements

The hardware requirements includes the Raspberry Pi 2 Model B, Raspberry Pi power supply, 8GB microSD card, microSD card reader, USB microphone, speakers as well as an Ethernet cable and a laptop.

### B. Software Requirements

The following are the software requirements of the system:

- Raspbian Jessie: Operating system for the Raspberry Pi.
- Win32DiskImager: To burn the Raspbian OS disk image into microSD card.
- PuTTY and Virtual Network Computing (VNC): For remote access to the Raspberry Pi's terminal and graphical desktop.
- Jasper: An open source voice-controlled software.
- Python 2.7 and Tkinter: Programming language used to code the functional modules and design GUI layout.

### C. System Design

A system architecture diagram as shown in Figure 2 is created to show the structural organization of PAL.

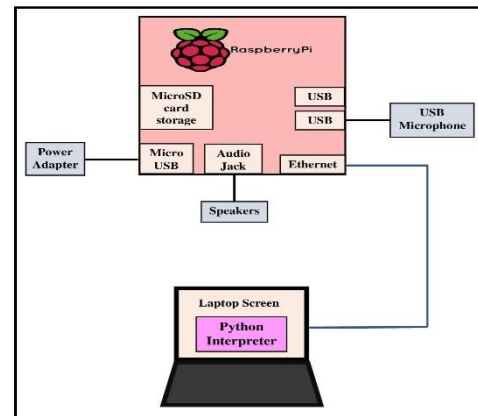


Figure 2: The system architecture of PAL

### D. Installations and Configurations

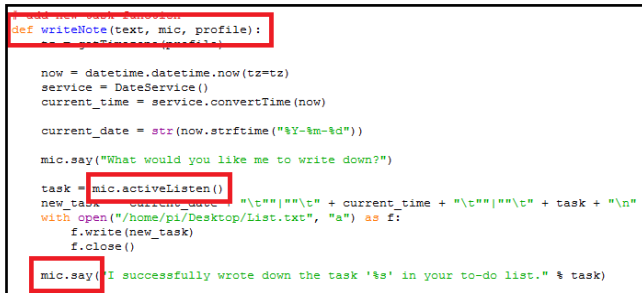
All the required software was downloaded before moving on to installations and configurations. Next, the Raspberry Pi's operating system, Raspbian Jessie is installed by downloading the disk image from the Raspberry Pi's website: <https://www.raspberrypi.org/downloads/raspbian/> and then burning it into the 8GB microSD card. The hardware components were set up as can be seen in Figure 2 in Section III.C. Then, we installed Jasper and its dependencies using the guide from Jasper project's website: <https://jasperproject.github.io/documentation/installation/>.

The dependencies that are needed in order to use Jasper are Speech-To-Text (STT) engine are Wit.Ai (<https://wit.ai/>) and Text-To-Speech engine called SVOX Pico (<https://ankiatts.appspot.com/services/pico2wave>). The STT

engine is used to convert speech to written text while the Text-To-Speech (TTS) engine is used to convert text into speech. After all that is done, Jasper can be launched in the terminal.

### E. Development of the Functional Modules

The development phase involves the development procedures of the functional modules and the GUI. All the coding were written in Python 2.7 and saved as a python file (.py). For this section, the To-Do List Module will be taken as an example to explain the coding method of the development process. The first step to develop this module was assigning keywords such as “NEW”, “DISPLAY”, “TASK” and “LIST” in the Python module.



```

def writeNote(text, mic, profile):
    now = datetime.datetime.now(tz=tz)
    service = DateService()
    current_time = service.convertTime(now)

    current_date = str(now.strftime("%Y-%m-%d"))

    mic.say("What would you like me to write down?")

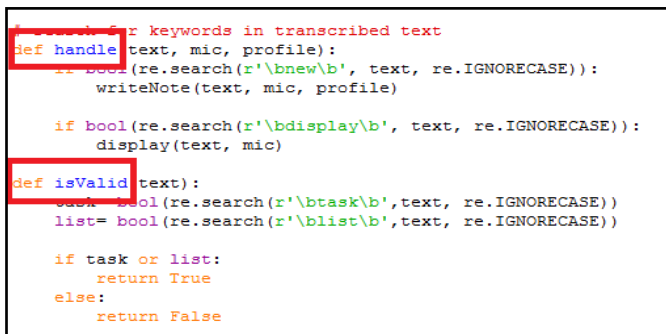
    task = mic.listen()
    new_task = current_date + "\n" + current_time + "\n" + task + "\n"
    with open("/home/pi/Desktop/List.txt", "a") as f:
        f.write(new_task)
        f.close()

    mic.say("I successfully wrote down the task '%s' in your to-do list." % task)

```

Figure 3: The writeNote function in Jasper

The To-Do List Module contains two functions which are ‘writeNote’ and ‘display’ that adds a new task into the list as well as displays the contents in the list respectively. Figure 3 shows a code snippet of the ‘writeNote’ function. The “def” command is used to define functions of the modules. The syntax for the def command starts off with the word “def” and followed by a function name which in this case is “writeNote”. After that, the “mic.listen()” command is used to capture the user’s input speech while the “mic.say(message)” command is for the system to respond back to the user.



```

def handle(text, mic, profile):
    if bool(re.search(r'\bnew\b', text, re.IGNORECASE)):
        writeNote(text, mic, profile)

    if bool(re.search(r'\bdisplay\b', text, re.IGNORECASE)):
        display(text, mic)

def isValid(text):
    check = bool(re.search(r'\btask\b', text, re.IGNORECASE))
    list = bool(re.search(r'\blast\b', text, re.IGNORECASE))

    if task or list:
        return True
    else:
        return False

```

Figure 4: The handle and isValid function in Jasper

The users’ input speech that has been captured will then be converted into written text (transcribed text). From Figure 4, the ‘isValid(text)’ function is used to determine whether the transcribed text is valid or not by finding a match between the transcribed text and the assigned keywords. If a match is found, then the valid translated text is the passed on to the ‘handle(text, mic, profile)’ function to execute a function in the module, otherwise the command is ignored. For example, the transcribed text is ‘new task list’ and since the keywords ‘task’ and ‘list’ are valid, it is then passed on to the ‘handle()’ function. In the handle function, it then detects for another

keyword where in this case the keyword is ‘new’ and thus executes the ‘writeNote’ function. The same method of coding was used to develop the Translate Module whereby the only difference was the keywords and function statements. On the other hand, the Email Module is already pre-installed in Jasper whereas the Search Module and Calendar Module are obtained from the Jasper project’s website. We configured the Calendar Module in order to link it to our own Google Calendar.

### F. Testing and Evaluation

A number of tests are done to determine the accuracy and performance of the functional module, GUI layout and the overall performance of the system. The GUI layout is tested to ensure that it displays the correct output according to the modules executed. Evaluation questionnaire surveys of PAL were distributed to respondents whom have tried using PAL during the event Innovation Technology Expo (InTEX) 2017 held in UNIMAS to obtain some feedbacks. The evaluation survey test helped in determining the drawbacks of the system and hence giving us an idea for future improvements. Functional modules were tested individually by calculating the Command Success Rate (CSR) and Word Error Rate (WER). For the CSR, the highest the percentage of the WER, the higher the success rate in executing commands whereas for the WER, the lower the percentage of the WER, the higher the accuracy in recognizing speech. Finally, a test on three chosen names for PAL were conducted to determine which name has the fastest response rate and hence the most efficient to be used for PAL.

## IV. RESULTS & ANALYSIS

### A. GUI Output Display

For this section, the Search Module is used to test on the accuracy of the GUI output display. When the keywords “SEARCH” is recognized, this will prompt Tina to ask the users what information they would like to search for as can be seen in Figure 5. Tina then dictates what the user says and use that input to do a search through Wikipedia. Figure 6 shows that the output result displayed in the GUI appeared in the center of the screen while displaying the summarized search result on the search topic requested by the users. Thus, the GUI has proven to give the correct output accordingly.

### B. User Acceptance Test

Due to page limitation constraints, we summarize our evaluation survey result obtained during the InTEX’17 instead of publishing the responses from our respondents. We have observed and concluded that the biggest drawback of PAL is that it has a slow response rate when trying to detect its name ‘Tina’ from being called out by users. Apart from that, the accuracy of the input and output, success rate of executing commands and functions, usefulness of the functional modules, accuracy of the GUI output display as well as the user-friendliness and cost of the system has all been rated as satisfied by the respondents with the percentage number of 50% and above. Overall, half from the total number of respondents are satisfied with the overall performance of PAL despite the drawback.

```

('User': [],
('User': ['TINA'])
('User': ['SEARCH'])
TINA: What would you like to learn about?
('User': ['BORNEO'])
TINA: Borneo (/ˈbɔːrniəʊ/; Malay: Pulau Borneo, Indonesian: Kalimantan) is the t
third-largest island in the world and the second-largest island in Asia. At the g
eographic centre of Maritime Southeast Asia, in relation to major Indonesian isl
ands, it is located north of Java, west of Sulawesi, and east of Sumatra.
The island is politically divided among three countries: Malaysia and Brunei in
the north, and Indonesia to the south. Approximately 73% of the island is Indone
sian territory. In the north, the East Malaysian states of Sabah and Sarawak mak
e up about 26% of the island. Additionally, the Malaysian federal territory of L
abuan is situated on a small island just off the coast of Borneo. The sovereign
state of Brunei, located on the north coast, comprises about 1% of Borneo's land
area. Antipodal to an area of Amazon rainforest, Borneo is itself home to one o
f the oldest rainforests in the world.
('User': [])
    
```

Figure 5: Search information on the internet

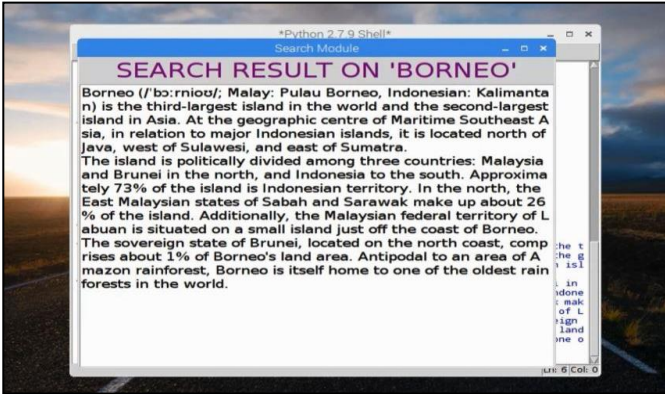


Figure 6: GUI output display of "Search" module

C. Word Error Rate (WER)

Whereas for the WER, according to Table 1, the average WER calculated for all five of the functional modules is 14%. This shows that the proposed system is able to transcribe the input speech into text accurately.

Table 1  
Summary of WER and CSR results

Modules	WER (%)	Average WER (%)	CSR (%)	Average CSR (%)
Search	17		96	
Email	8		100	
Calendar	9	14	96	96.8
To-Do List	19		92	
Translate	17		100	

D. Command Success Rate (CSR)

Based on Table 1, the average CSR for all five of the functional modules is calculated to be 96.8%, which means that the success rate of executing a command correctly and accurately is very high.

E. Response Rate Test

Table 2 shows the response rates of three names that were chosen for the system. Each name was called out repeatedly in 5 rounds to determine the response rate in seconds. From the table, it is observed that the name 'Tina' has the fastest response rate of 8.08 seconds compared to the names 'Jasper' and 'Apple'. Hence, the name 'Tina' was chosen to be used for PAL. Unfortunately, the response rate from 'Tina' is still voted as rather slow by the evaluators as observed in our survey. One of the reasons that could affect the response rate is because the online STT, Wit.Ai, is built using native English speech data. Therefore, Wit.Ai may not be able to recognize speech accurately, especially for words that sound similar such as 'one' and 'want'. Another cause is the inability of the STT to recognize speech fast as it uses Internet connection. Therefore, the response rate is the drawback of PAL.

Table 2  
Summary of response rate result

Number of rounds	Tina		Jasper		Apple	
	Response Rate (s)	Average (s)	Response Rate (s)	Average (s)	Response Rate (s)	Average (s)
1	8		5		20	
2	6.6		20		16.6	
3	13.3	8.08	12	13	5.7	14.44
4	7.1		15		13.5	
5	5.4		13		16.4	

V. CONCLUSIONS

This paper described our work in developing a personal assistant system called PAL, which could assist disabled people using Raspberry Pi as their personal computers. In our project, five functional modules namely the Search, Email, Calendar, To-Do List and Translate module were developed whereby the functionalities includes searching for information on the internet, check for unread emails, schedule events in the calendar, manage a to-do list and translate texts respectively. A simple GUI layout was also designed to display the output results on each of the functional modules upon execution. For all five of the functional modules, the modules are proven to be able to recognize words rather accurately with the average WER of 14% (Section IV.C), while the average CSR is calculated to be 96.8% (Section IV.D), which means that the success rate of executing a command correctly and accurately is very high. Based on the results obtained in Section IV.B and Section IV.E, we have concluded that the drawback of PAL is that it has a slow

response rate and this could be due to a number of reasons such as heavy accents, poor speech recognizer and poor Internet connection. Thus for future enhancement, we can use an open source STT called PocketSphinx which enables us to train our own models to further improve the accuracy of speech recognition which can improve response rate.

REFERENCES

- [1] G. Halfacree and E. Upton, *Raspberry Pi user guide*, Wiley Publishing, 2012.
- [2] S. Bush, *Dongle computer lets kids discover programming on a TV*, 2011. Retrieved June 12, 2017, from <https://www.electronicweekly.com/market-sectors/embedded-systems/dongle-computer-lets-kids-discover-programming-on-a-2011-05/>
- [3] I. Paul, *Meet Jasper, an open-source, Siri-like virtual assistant for Raspberry Pi*, 2014. Retrieved June 12, 2017, from <http://www.pcworld.com/article/2142283/jasper-voice-automation-project-brings-a-siri-like-virtual-assistant-to-raspberry-pi.html>
- [4] A. Mansikkaniemi, *Acoustic model and language model adaptation for a mobile dictation service*. Aalto University, 2010. Retrieved from

- <http://lib.tkk.fi/Dipl/2010/urn100143.pdf>
- [5] D. S. Deiv, G. K. Sharma, and M. Bhattacharya, "Development of Application Specific Continuous Speech Recognition System in Hindi," pp. 394–401, Aug. 2012. Retrieved from [http://file.scirp.org/pdf/JSIP20120300014\\_96299246.pdf](http://file.scirp.org/pdf/JSIP20120300014_96299246.pdf)
- [6] S. S. Juan and L. Besacier, "Fast bootstrapping of grapheme to phoneme system for under-resourced languages - application to the Iban language," 1–8 Oct. 2013. Retrieved from [http://ir.unimas.my/8876/1/wssanlp2013\\_sarah.pdf](http://ir.unimas.my/8876/1/wssanlp2013_sarah.pdf)
- [7] D. Jurafsky and J. H. Martin, *N-Gram. In Speech and Language Processing (2nd ed.)*, Prentice Hall, Pearson Education International, 2014. Retrieved from <https://lagunita.stanford.edu/c4x/Engineering/CS-224N/asset/slp4.pdf>
- [8] M. Nuttin, D. Vanhooydonck, H. D. Brussel, K. Buijsse, L. Desimpelaere, P. Ramon, T. Verschelden, "A robotic assistant for ambient intelligent meeting rooms," In: Aarts E., Collier R.W., van Loenen E., de Ruyter B. (eds) *Ambient Intelligence. EUSAI 2003. Lecture Notes in Computer Science*, vol. 2875. Springer Berlin Heidelberg, 2003.
- [9] E. C. Grigore, A. Pereira, I. Zhou, D. Wang, and B. Scassellati, "Talk to me: Verbal communication improves perceptions of friendship and social presence in human-robot interaction," *Intelligent Virtual Agents Lecture Notes in Computer Science*, pp. 51-63, 2016.
- [10] S. Strohkorb, E. Fukuto, N. Warren, C. Taylor, B. Berry, and B. Scassellati, "Improving human-human collaboration between children with a social robot," *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2016.
- [11] B. Davidovic and A. Labus, "A smart home system based on sensor technology," *Facta Universitatis - Series: Electronics And Energetics*, vol. 29, no. 3, pp. 451-460, 2016.
- [12] V. Vujovic and M. Maksimovic, "Raspberry Pi as a sensor web node for home automation," *Comput. Electr. Eng.*, vol. 44, pp. 153-171, 2015.
- [13] R. Chauhan, Y. Jain, H. Agarwal, and A. Patil, "Study of implementation of Voice Controlled Wheelchair," 2016 3Rd *International Conference On Advanced Computing And Communication Systems (ICACCS)*, 2016.
- [14] A. W. Black, A. Chan, D. Huggins-Daines, M. Kumar, M. Ravishankar, and A. I. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1, pp. I-I, 2006.