

Universal Language Translator Using Raspberry Pi

^[1]Ravinder Kumar^[2]Prem Kumar Ch.

University of Petroleum and Energy Studies,

P.O. Bidholi via Prem Nagar, Dehradun, Uttarakhand, India

^[1]yravinderkumar33@gmail.com ^[2]bharathkumar30@gmail.com

Abstract- This paper describes the development of a Raspberry pi Universal Language Translator and will demonstrate to you proper methodologies to transform a \$35 little PC (Raspberry Pi) into a component rich language translator which not only just backs up your voice acknowledgment and local speaker playback, but on the other hand is capable of translating your voice into many different languages (about 60 different languages). The Universal translator works by using a headset which is connected to a Raspberry pi and then it converts the voice into text by using the Google Speech Recognition tool and back into the desired language using Microsoft API (Application Program Interface). Project runs on some open source soft wares, therefore its runtime cost is negligible.

Index Terms- GoogleSpeech Recognition Tool, Google Translate Tool, Microsoft translation API, Raspberry Pi.

II. LITERATURE REVIEW

I. INTRODUCTION

If you've ever tried to communicate with someone who speaks a different foreign language, you know that it can be extremely difficult— even with the assistance of cutting edge translation sites where we need to pay a lot sums of money to fulfill our task. This project will demonstrate to you proper methodologies to transform a \$35 little PC -Raspberry Pi [1] into a component rich language translator which not only just backs up your voice acknowledgment and local speaker playback, but on the other hand is equipped translating your voice into many different language. The incredible part is that it is possible at little to no cost by utilizing reasonable equipment, free translation APIs [2], and some open source programming done on Raspberry Pi which runs on Linux Command line [3] platform. Upon successful completion of this project, mysteriously talk in another language, whatever you say gets translated into another desired language.



Fig. 1 -The Raspberry Pi is a tiny and affordable computer that you can use to learn programming through fun, practical projects.

There are numerous language translators available where we have to manually enter the content and the application do whatever is left of the work of making a translation of the content into the desired language. Some translation platforms even charges for this process. For example the most popular Google translate API charges calculated as per the usage. [4]

Usage fees for Google Translate API:

A. Translation:

\$20 are charged as per 1 M characters of text.

B. Language Detection:

\$20 per 1 M characters of text.

Moreover there is a limit too in the number of characters that are being translated in a day and the application demands a token to convey the undertaking forward.

Google translator utilizes Google translator's AJAX API's [5] in order to translate text and retrieves the text by parsing the returned JSON content.

But this project is quite free since all the API's that are being used in the project are open source, their source codes are available on the internet and anyone can buy a Raspberry pi and can make use out of it.

There is another example of a translator which uses java on Linux platform to translate from one language to another. Example based machine translation, EBMT [6], is such an example which uses java on Linux platform for translating from one language to another. The rule of deciphering in EBMT is basic: it analysis the pre-translated data in the data base and further translates the input. Along these lines, the bigger the database of pre-translated

sentences, more prominent and efficient will be the output. Moses decoder[7] and Giza++ [8] are some of the tools or say soft wares which assist with the translation part. More or less all these soft wares are paid where the user needs to pay some amount of money to translate into their desired language.

III. SYSTEM MODEL

This project aims at changing over a basic raspberry pi into a translation gadget fit for translating into more than 60 languages. It bolsters speech recognition and playback, and it runs on free open sources soft wares whose source codes can be downloaded completely free of cost.

A. Hardware Requirements [9]

1. Raspberry pi(Raspberry Pi Model B+ with 700 MHz Broadcom BCM2835 CPU with 512MB RAM, 4 USB ports and 40 pins GPIO header).
2. USB Keyboard compatible with Linux.
3. SD Card(at least 4 Gb-class 4)
4. Micro USB cable
5. Headset with microphone
6. Power Supply cord



Fig. 2 -This Raspberry PI language translator is capable of recognizing over 60 different languages

Whole process is divided into three different sections [10]

1. Google's speech recognition API
2. Microsoft's translation API
3. Text to speech.

B. Google Speech Recognition Tool

There are a couple of alternatives for speech recognition tool, however I thought the best answer for this instructional exercise was to utilize Google's Speech recognition Tool. This API permits us to transfer the voice we simply recorded and changes it to text content.

C. Microsoft Translation API[11]

Since we can record our voice and then convert it into text using the Google's speech recognition, we have to make a translation of that text to our desired outside language. I would have loved to utilize Google's Translate API for this, since tragically there is a 20\$ sign up expense for the utilization of this API. Because I needed to make this project free so everyone had a chance to attempt it and get benefited, thereforeas an alternative, I will be utilizing Microsoft's translation API which at present is free of cost for utilization.

D. Google text to Speech Tool[12]

Google text to speech API again converts back the output text content into voice which can be heard on headset.

E. Use the Universal Translator

After setting up all the API's and assembling the source code .You'll hear the translation in the headset and you'll also see feedback on the command line. By default, it translates from English to Spanish. You can easily change your origin and destination languages by making some small changes in the source code file (text-to-translate.py) and the script will do the rest! There are literally thousands of language pairs supported.



Fig.3 -Language translation step by step cycle.

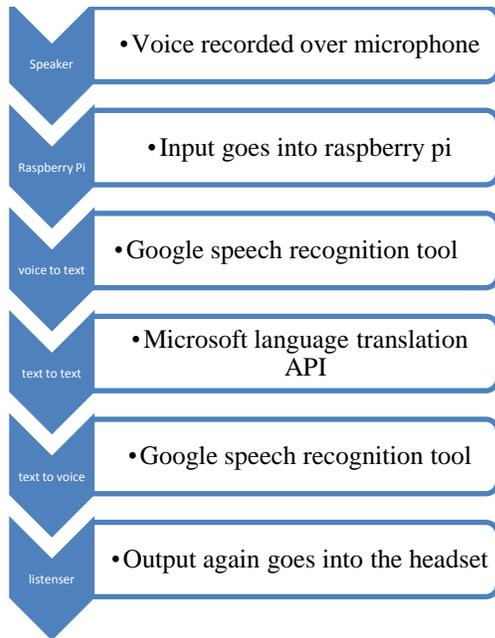


Fig. 4 -Different steps involved in language translation process

IV. CONCLUSIONS

With the aid of Google speech recognition and Microsoft language translation API, the methodology to manufacture the language interpreter resembles this:

1. Record the voice using microphone and Google speech recognition tool changes the voice into text.
2. Using Microsoft's translation service to translate the text into desired language.
3. Use Google's Speech tool again to change the output text into the voice again to be played back on the headset.

This process requires constant internet connection since all the API's functions online. After all the API's have been assembled, you can hear the translation over the headset and the required feedback on the command line on the screen.

REFERENCES

[1] What Is Raspberry Pi?. Retrieved from <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>

[2] Application Programming Interface. n.d. In *Wikipedia*. Retrieved December 17, 2015 from https://en.wikipedia.org/wiki/Application_programming_interface/

[3] Shotts, W.E.. Learning the shell. Retrieved from http://linuxcommand.org/lc3_learning_the_shell.php

[4] Pricing. Retrieved from <https://cloud.google.com/translate/v2/pricing>

[5] Badger, B..(2008, March 20). New Google AJAX Language API - Tools for translation and language detection. Retrieved from <https://googleblog.blogspot.in/2008/03/new-google-ajax-language-api-tools-for.html>

[6] Example-based machine translation. n.d. In *Wikipedia*. Retrieved October 18, 2015 from https://en.wikipedia.org/wiki/Example-based_machine_translation

[7] Moses (machine translation). n.d. In *Wikipedia*. Retrieved August 26, 2015 from [https://en.wikipedia.org/wiki/Moses_\(machine_translation\)](https://en.wikipedia.org/wiki/Moses_(machine_translation))

[8] Franz Josef Och, Hermann Ney. "A Systematic Comparison of Various Statistical Alignment Models", *Computational Linguistics*, volume 29, number 1, pp. 19-51 March 2003. Retrieved from <http://www.statmt.org/moses/giza/GIZA++.html>

[9] RASPBERRY PI 1 MODEL B+. Retrieved from <https://www.raspberrypi.org/products/model-b-plus/>

[10] Upton,L..(2013, November 6).Build a universal translator. Retrieved from <https://www.raspberrypi.org/blog/build-a-universal-translator/>

[11] Microsoft Translator API. Retrieved from <https://www.microsoft.com/en-us/translator/translatorapi.aspx>

[12] Google Text-to-Speech. n.d. In *Wikipedia*. Retrieved December 02, 2015 from https://en.wikipedia.org/wiki/Google_Text-to-Speech