Sinhala Speech to Text Library using Sphinx

WTVL Gunarathne[#1], TK Ramasinghe[2], DGJB Wimalarathne[3], BMSH Balasuriya[4] and B. Hettige[5]

[1,2,3,4,5]*General Sir John Kotelawala Defence University*
Corresponding author: vlgunarathne@gmail.com

Abstract *– Speech-to-text has generated a tremendous interest in the field of Natural Language Processing where the ultimate goal is to build applications and systems that has the capability to respond to the natural languages that us humans use in a daily basis. Converting speech to text using European languages has emerged in the world and can be found in most modern electronic devices. But a speech-to-text tool for the native language of Sri Lanka which is "Sinhala" is rare to find. So, developing a STT algorithm for Sinhala to implement an application would help most of the local workforce for their day-today tasks and for the development of the software industry with a unique Sri Lankan touch. STT has become one of the best used technologies in the modern world. It has caught the attention of the public as it is an excellent user-friendly feature that has been embedded to electronic devices in the modern world. It too has caught the attention of the organizations as they can develop much better and attractive devices which can rise in the competing market. Personal Computers, Hand-held smart devices and even modern automobiles come pre-integrated with STT for the convenient of the user and give them a better experience.*

Keywords – *Speech-to-Text, Sinhala, Sphinx4*

I. INTRODUCTION

Speech-to-text has become an interesting phenomenon in the modern world. Various types of end devices such as Personal Computers, Hand held smart devices and even modern automobiles come integrated with this feature for the convenience of the end user. One of the basic stages of these advance voice recognition systems is the recognition of words spoken buy a human being or in other words, natural language recognition. These algorithms can identify words spoken by a user, cross reference the words through a word dictionary for validation and generate an output to the user. It is an alternative to typing on a keyboard and this is something that exists today in smartphones where one of the most known application is "Siri"("Apple - iOS - Siri," 2015) for Apple products and "Google Now"("Google now," 2015) for devices running on Android OS. Speech recognition will be more and more common in the future as the amount of data grows and devices contain more features.

Speech-to-Text software programs work by analyzing sounds and converting them to text. This software has been developed in a way to provide a faster method of writing on a computer and can help people with a variety of disabilities. It is useful for people with physical disabilities who often find typing difficult, painful or impossible. Speech-to-Text software can also help those with spelling difficulties, including users with dyslexia, because recognized words are almost always correctly spelled. There some major researches in the field. The first one is a CMU SPHINIX-4 SPEECH RECOGNITION SYSTEM done by eight personals in different four institutions. They are Carnegie Mellon University, USA, University of California, Santa Cruz, USA, Sun Microsystems Laboratories, USA and Mitsubishi Electric Research Labs, USA. They have done a marvelous research as joint development. There are some significant features of the Sphinix-4 decoder. It is highly modular and flexible, supporting all types of HMM-based acoustic models, all standard types of language models, and multiple search strategies. Algorithmic innovations in the system enable the concurrent use of multiple information streams. The Sphinx-4 system is an open source project. The code has been publicly available at SourceForge™ since its inception. The design, results, and team meeting notes are also publicly available.

VIETNAM NATIONAL UNIVERSITY, THE INTERNATIONAL UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING have introduced a good method of identifying the words using the phonetics. The words to be identified should be broken down to its phonetics. As an example, for the following two Sinhala words can be encoded as,

e.g.: □□□ - M AH G EH

□□ - N AH M AH

So, by using this concept we have made a gram file for Sinhala language. We have put some frequently used words in the Sinhala language, generated their

phonetic scheme as above and created a grammar file. There is a need of a good accurate application for Speech Recognition regarding Sinhala language. So, we thought of developing a library for Sinhala language that can be edited for many purposes. We thought of making it as an open source library so that various people can make contributions to developing it and thereby make use of it to address Sinhala speech-to-text requirements.

III. PROBLEM AND OBJECTIVE

The problem at hand is that there is a need for a good and accurate Speech-to-text application for Sinhala language. And there is a need for a text library for Sinhala language so that it can be used inside other applications to convert spoken Sinhala words to Sinhala text. Our main objective in this research was to study and develop a language model and an acoustic model for the Sinhala language which results in a language model file and a dictionary file which includes the phonetics of the Sinhala word pronunciation which can be used as an externally pluggable library to any application.

IV. SOLUTION

The solution to the above-mentioned problem is to develop a library package which can convert spoken Sinhala words to Sinhala text. This can be accomplished by making modifications to the opensource library Sphinx4("Sphinx-4 - A speech recognizer written entirely in the Java(TM) programming language," 2015) and there by compiling a bunch of library files that can be directly imported an application development environment.

Speech is a complex phenomenon of which people rarely understand how it is generated and perceived. The general conception is that speech consists of words and each word consists of phones which happen to be the building blocks (phonetic schemes) of the word pronunciation.

Most modern speech recognition systems are mainly based on probability theories. That means that there are no certain boundaries between units, or between words. Speech to text translation and other applications of speech are never completely correct. Speech consists of a continuous wave form comprising stable states mixed with dynamically changing states. Within these sequences, we can define similar components of sounds called **phones**. A single word is said to be built from phones. Context, speaker, style of speech, dialect can be factors which would change the acoustic properties of a waveform. Transitions between words are more informative than stable regions, developers often talk about **diphones** - parts of phones between two consecutive phones. Sometimes sub phonetic units (different sub-states of a phone) are also considered.

In a considered phone there can be identifiable three states, namely **preceding phone**, **middle part** & **subsequent phone**. Phones that are considered in context are called **triphones** or **quinphones**.

The recognition process is undertaken by taking the waveform, split it on utterances by silences then try to recognize what's being said in each utterance.

V. DESIGN

Speech-to-text modules and libraries have been proliferating in the technological arena and they tend to gain traction much more as it is a user-friendly mode of interaction between the user and a computer. The method that these efficient algorithms use is a statistical analysis of the probability of a word that can appear in a recorded voice based on the array of words spoken before that specific word. There was no considerable success in the field of speech recognition until Lenny Baum of Princeton University invented the Hidden Markov Model (HMM) which provided a statistical based model to generate text from speech. From that point, onwards many corporations have used this model to develop their own speech recognition systems.

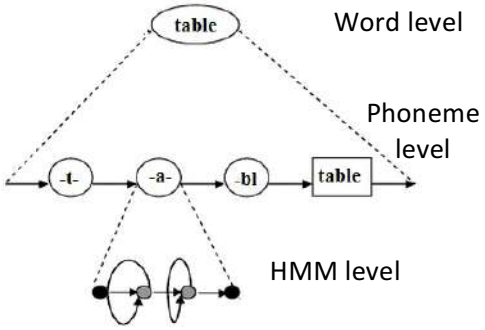

Figure 1: Breakdown of words to phonemes

A group of scientists from the Carnegie Mellon University, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs, Hewlett Packages and contributors from University of California and Massachusetts Institute of Technology have built a well-known speech recognition framework written in Java programming language called the Sphinx4 Framework.("CMU Sphinx," 2015, p. 4) Sphinx4 is a

speech recognition system based on HMM which is a precise mathematical framework. The ability to use pluggable modules in the Sphinx4 framework makes it more flexible and easy to adopt. Hidden Markov Model (HMM) is a statistical model in which the system being modeled assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from an observation parameters. In speech recognition process, after the user's voice is recorded, it will be divided into many frames that we need to process to generate the sentence in text form. Each frame is represented as a state, group of some states is represented as phoneme, and group of some phonemes is represented as word that we need to recognize. In database known as linguist model, we store the reference value of state, phoneme, and word to compare with the observed data (voice).

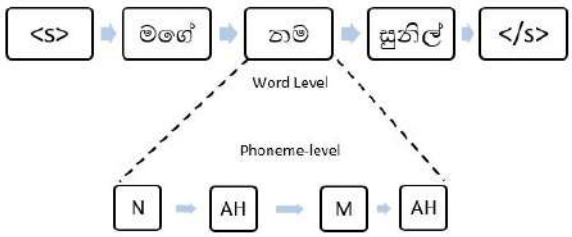

Figure 2: Phoneme breakdown for Sinhala

Word	Phoneme
???	M AH G EH
??	N AH M AH
?????	S UH N IH L

Table 1: Breakdown of Sinhala words to phonetic scheme

The above illustration depicts how a word can be sub-divided into its phones. The Sinhala word "??" is basically comprised of 4 phones according to which a human being pronounces it. The table shows the phone building blocks for some other simple Sinhala words. Notice the tags "<s> </s>" which are the opening and closing tags of a series if Words in the dictionary (.dict) file.("Vietnamese Language Recognition with Sphinx4 | Khai Tran - Academia.edu," 2015)

VI. PHONETIC SCHEME ALGORITHM

For the Speech-to-text conversion using the Sphinx4 library, the phonetic schemes of the Sinhala words should be recorded in a dictionary file which resides inside the compiled jar file of the Sphinx4 library. These phonetic schemes are the basic auditory units that buildup a single word. The Sphinx4 library already has a built-in dictionary file that comprises of English words and their corresponding phonetic schemes. But for Sinhala words to be recognized, the Sinhala words along with the corresponding phonetic scheme (as shown in Table 1). Through a thorough analysis and testing of the dictionary file for the English words, we identified the phonemes in the lower level that corresponds to various pronunciations. There by, we could map the English phonemes to Sinhala words depending on how the words are pronounced.

- ????? S IH NG
- ??? M AH G EH
- ?? N AH M AH
- ??????? S UH N IH L
- ?? R AH T AH
- ????? OW B AH N N AH
- ??? HH AH R IY
- ????????? N IH W AH R AH D IY
- ???? Y AH N N AH
- ???? EH N N AH
- ?? OW B AH
- ????????? P IH T UH W AH HH AH L

The phonetic scheme algorithm that we have developed, will decompose a given Sinhala word to its phonetic scheme as follows.

Figure 3: Phonetic scheme algorithm architecture

Figure 4: Algorithm implementation

The below figure will show an example for the Sinhala letter "□" and all its transformations and the relative phonetic scheme for each of those combinations.

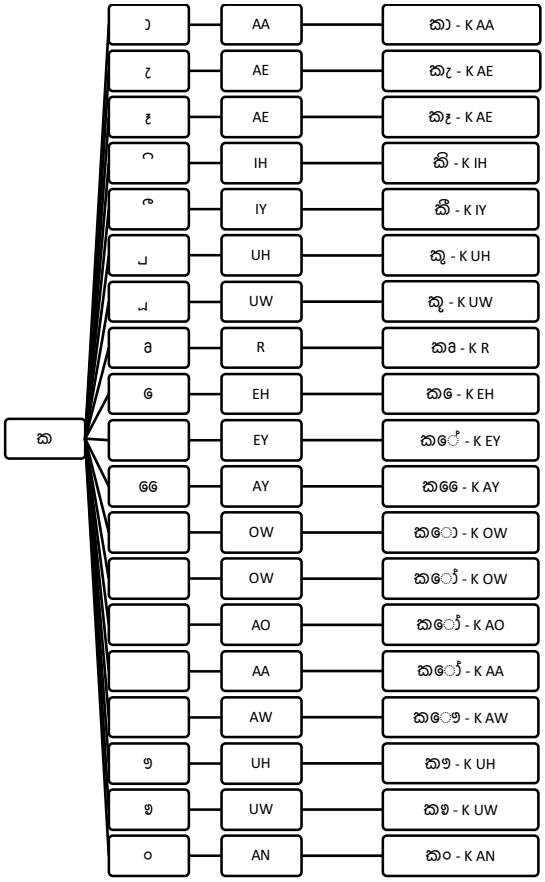

Figure 5: Formations of phonemes

VII. IMPLEMENTATION

i. Sinhala Speech-to-Text sample application

Figure 6: Sinhala speech-to-text application

This application takes the input from the microphone and using the phonetic scheme above, recognizes the word and displays it to the user. This above shown is an example of the recognized word "□□□□□□□□□" spoken through the microphone.

ii. The Web Browser

Figure 7: Sinhala voice search browser

This is a browser powered by the Google Search API[10] and completely works with Sinhala words recognized using voice, which are passed to as a search phrase to the google search engine which in turn, returns the search results as a Json object. The google Gson helps to decode the Json object to separate the titles and the URLs of the returned results. Those results can be seen in the large textarea in the above figure.

VIII. EVALUATION

The recognition performance evaluation of the system must be measured on a corpus of data. A separate test corpus, with new Sinhala words, was created from the main corpus. The test corpus was made of 200 recorded and labelled data. In order to test for speaker independency of the system, some of the subjects who participated in creation of the testing corpus had not participated in creation of the training corpus.

We have analyzed 200+ words and the data statistics are as follows:

The average word identification rate for the tested data set is 65.84% with a variance of 1000.961 and a standard deviation of 31.63797.

IX. CONCLUSION

With the proliferation of Speech-to-text technology, it's clear that it has become an essential technology in almost every electronic device. So here our attempt was to develop a system and a library to handle voice recognition in Sinhala language. We used the CMU Sphinx4 library to achieve this task. The Sphinx4 library consists of a grammar file which contains English words and their corresponding phonetic schemes. Those phonetic schemes are mapped to various word pronunciation blocks. This research used those blocks and mapped them to the pronunciation patterns of the Sinhala words available in the selected data set. There by we could develop a

Sinhala speech-to-text library which can detect a spoken Sinhala word, cross refer it with the existing data base and if the word is available then will give a text output of the spoken word. The final output of the research is a java library that can be downloaded and used to any other application development process. The issue that we faced was to improve the word detection percentage to a much higher level. The reason for that would be the fact that Sphinx4 library was originally designed to adhere to the dialect of Europeans. Since the pronunciation of Asians and specifically Sri Lankans is different, the decoder finds it difficult to identify the dialect. Therefore the word recognition rate is lower.

References

Apple - iOS - Siri [WWW Document], 2015. URL https://www.apple.com/ios/siri/ (accessed 7.16.15).

CMU Sphinx, 2015.

Developer's Guide | Google Web Search API (Deprecated) | Google Developers [WWW Document], 2015. URL https://developers.google.com/web-search/docs/?hl=en (accessed 11.26.15).

Dragon NaturallySpeaking - world's best-selling speech recognition software | Nuance [WWW Document], 2015. URL http://www.nuance.com/dragon/index.htm (accessed 7.8.15).

FreeTTS 1.2 - A speech synthesizer written entirely in the Java(TM) programming language [WWW Document], 2015. URL http://freetts.sourceforge.net/docs/index.php (accessed 7.8.15).

Google now [WWW Document], 2015. URL https://www.google.com/landing/now/ (accessed 7.16.15).

google/gson · GitHub [WWW Document], 2015. URL https://github.com/google/gson (accessed 11.26.15).

How does voice recognition software work? - Explain that Stuff [WWW Document], 2015. URL http://www.explainthatstuff.com/voicerecognition.html (accessed 8.29.15).

Microsoft Speech API (SAPI) 5.4 [WWW Document], 2015. URL https://msdn.microsoft.com/en-us/library/ee125663(v=vs.85).aspx (accessed 7.15.15).

Q6.2: How is speech recognition performed? [WWW Document], 2015. URL http://www.speech.cs.cmu.edu/comp.speech/Section6/Q6.2.html (accessed 8.29.15).

Sphinx-4 Application Programmer's Guide [CMUSphinx Wiki] [WWW Document], 2015. URL http://cmusphinx.sourceforge.net/wiki/tutorialsphinx4 (accessed 10.21.15).

Vietnamese Language Recognition with Sphinx4 | Khai Tran - Academia.edu [WWW Document], 2015. URL http://www.academia.edu/1501651/Vietnamese_Language_Recognition_with_Sphinx4 (accessed 8.29.15).

APPENDIX A

Word	Phoneme
□	AH
□	AA
□	AE
□	AE
□	IH
□	IY
□	UH
□	UW
□	EH
□	EY
□	AY
□	OW
□	OW
□	AW
□	K
□	G
□	CH
□	CH
□	JH
□	T
□	D
□	N
□	N
□	TH
□	DH
□	P
□	B
□	BH
□	M
□	Y
□	R
□	L
□	W
□	SH
□	SH
□	S
□	HH
□	L
□	F
□	
□	AA

◌	AE
◌	AE
◌	IH
◌	IY
◌	UH
◌	UW
◌	R
◌	EH
◌	EY
◌	AY
◌	OW
◌	OW
◌	AW
◌	UH
◌	UW
◌	AN