# Web based Automated Speech-to-Text Translator for the Sinhala Language

*[1]M. Punchimudiyanse, [2]R.G.N. Meegama*

[1]Department of Mathematics and Computer Science, The Open University of Sri Lanka,

[2]Department of Computer Science, University of Sri Jayewardenepura

ABSTRACT

*Modern automated speech recognition (ASR) applications are built for English and the ASRs for other languages are emerging. We present a web portal which functions as a large vocabulary Sinhala ASR. In the proposed approach, a flash-based voice recorder is used to obtain voice recordings at the client side at the sampling rate of 16 kHz. A hidden Markov model (HMM) based Sphinx toolset is used as the ASR decoder and the language model trainer at the server side. The ASR output is converted to Unicode Sinhala and displayed at client through AJAX calls which can also be downloaded as a text file.*

*Speaker adaptation is performed using 125 Sinhala voice recordings of a registered user. The current speech model has a recognition accuracy of 83% when using a close-talk headset. This system also encourages registered users to participate in building a speaker independent ASR for Sinhala.*

## 1.0 INTRODUCTION

Automatic speech recognition (ASR) is used in many applications from simple voice commands to continuous dictation of sentences. Speech researchers are thriving to add voice recognition services to improve recognition accuracy for their respective native languages. A majority of ASR applications is developed for English while ASRs are being developed for other languages as well.

Hidden Markov models (HMM), Gaussian mixture models and deep neural networks are the mostly adopted techniques in popular open source ASR tool sets such as Julius, Sphinx and hidden Markov modeling toolkit (HTK)[1-3].

This paper presents a web portal, which is developed to function as an ASR for Sinhala utilizing Sphinx toolset. An Adobe Flash-based customized voice recorder captures the voice at the web browser and the portal produces the voice recognized output in the Sinhala Unicode text.

## 2.0 FUNCTIONALITY OF AN ASR

An ASR with a large vocabulary has three common building blocks, namely, an acoustic model built from the voice samples of one or more speakers, a language model (LM) derived from processing a large number of legitimate sentences (text corpus) and a pronunciation dictionary of distinct words of LM presenting how each word is phonetically pronounced. In addition, the set of phonetic sounds that are used in building the words are also required.

The LM of a particular language is a collection of two elements, namely, combinations of the words and the occurrence probabilities of such combinations in a large list of syntactically correct sentences. Word combinations are called N-grams where 1-gram (N=1) is a single word, 2-gram (N=2) is two adjoining words and 3-gram is three adjoining words in a sentence. For example, the sentence "මම ගෙදර යනවා" (I am going home) has three 1-grams (මම, ගෙදර, යනවා), two 2-grams (මම ගෙදර, ගෙදර යනවා) and one 3-gram (මම ගෙදර යනවා). 3-gram LMs are the most common amongst speech recognition applications. They are built by dividing a text corpus into 1, 2, 3 grams of words with respective conditional probabilities of the word occurrences. The built-in Unicode support in speech recognizers is still in development and some LM building tools require sentence corpus of a given language to be in phonetic English. Hence, we have opted to convert the Sinhala Unicode sentence corpus of this research to phonetic English.

Building an acoustic model is called the training phase of a typical large vocabulary continuous speech recognizer (LVCSR). A set of voice recordings (voice corpus) and the exact text transcription of each recording are required to train

an acoustic model of a given speaker or speakers. First, voice clips are split into fragments of two to five milliseconds and voice features are extracted. Those extracted feature values are sent through an HMM to obtain statistical probabilities for sequences of phonetic sounds that are required to pronounce words in entire text corpus [4]. In this case, phonetic sound is either a character or a character combination of Sinhala expressed in phonetic English. If a given voice corpus covers all the combinations of characters with enough samples to have the statistical probabilities of different phonetic sequences, then the trained acoustic model is said to have a good coverage of phonetic sounds of an alphabet. Mel-frequency spectral coefficient (MFCC) is the feature extraction technique used in this research [5].

At the usage phase (recognition phase), the trained acoustic model is used to identify character sequences of an unknown voice clip by splitting the clip into pieces, extracting features and subsequently feeding them through the same HMM used at the training. The feature values calculated from an unknown clip can be compared with the known probabilities of phonetic sounds of the trained acoustic model to derive a set of phonetic sound sequences (characters/character combinations) that comes together to form a word.

Then the phonetic dictionary of distinct words is used by the speech decoder to identify words from the sequence of identified characters of the unknown voice sample. Then the LM is used to identify possible sequence of words having the best probability from the possible word combinations identified through the phonetic dictionary. Subsequently, the word sequence having the best probability of adjoining words is produced as the speech recognition output.

Different topologies of HMMs and Gaussian mixture models (GMMs) are used in the acoustic model training by the ASR toolsets. Those toolsets have provisions for the user to change the topology, number of training iterations and several other parameters to train an acoustic model with highest ASR accuracy for a given native language through trial and error basis.

Two methods are commonly used by speech researchers to recognize voices of multiple users. First method is using speaker independent acoustic model which is a acoustic model trained from a voice corpus having voices of more than 200 people who represent different gender and age groups. Assumption is that any person's voice may closely match a voice included in the training corpus of speaker independent model. The second method uses an acoustic model trained from voice of single or few speakers and adapting that model to a voice of a new speaker through voice adaptation techniques such as maximum a posteriori (MAP) and maximum likelihood linear regression (MLLR) [6,7]. For this task, voice samples amounting to 10-15 minutes of voice data of a new speaker is required.

In this research, the second technique is used to make the system multi user by requesting a user to record 125 Sinhala sentences to obtain 15 minutes of voice data for the speaker adaptation. Users are encouraged to participate in building a voice corpus to go for the first method of training a speaker independent acoustic model to reduce the overhead of the speaker adaptation process.

## 3.0 METHODOLOGY

To build a web based Sinhala ASR, we have deployed an ASR toolset in a Linux server with custom build Sinhala LM, single user acoustic model and pronunciation dictionary for Sinhala. Then an interactive portal has been developed using HTML/PHP to access command line ASR functions. The web portal is designed to allow registered users to build their own acoustic models matching one's voice through their profiles and perform ASR simultaneously.

### 3.1 Choosing ASR Toolset for Sinhala ASR

T. Nadungodage and R. Weerasinghe [8] have shown that building a Sinhala speech recognizer is possible with HMM based HTK toolkit [3]. We have opted to use sphinx toolset provided by the Carnegie Mellon University because it contains a compact ASR decoder named "pocket sphinx" and a complete set of tools to build a LM. Pocket sphinx is a lightweight open source speech decoder which could be used out of the box for both Windows, Linux and easily portable for mobile platforms [9]. Guidelines given in Sphinx tutorial

[10] are used in producing a Sinhala LM, pronunciation dictionary and training the single user acoustic model.

### 3.2 Building Sinhala Language Model

Sinhala LM requires conversion of a large set of Sinhala sentences to phonetic English and vice versa, when training and using an ASR. To accomplish this task, we have tested several phonetic tag sets published in [11,12] and found that the conversion of phonetic English back to Unicode Sinhala produced errors as they are not designed to work with ASR when tags are written in sequence without spaces. For example, a phonetically written word kannaadi (කන්නාඩි, meaning : spectacles), is decoded as k+a+n+n +aa+d+i (ක් අ න් න් ආ ඩ් ඉ) when the tag "aa" is searched before the tag "a" otherwise produce erroneous result of k+a+n+n+a+a+d+i (ක් අ න් න් අ අ ඩ් ඉ = කන්නඅඩ්) when the tag "a" is searched first. We have modified tag set presented in [11] to have different tags for similar sounded characters like න(na) and ණ(Na), and introduced tags for ් (al) character and zero width joiner (U+200D). The phonetic tag set we have used in this research covers 18 vowels 2 half vowels and 40 consonants in the modern Sinhala alphabet [13] as well as few additional tags for several other characters/modifiers in Sinhala Unicode Standard [14] as given in the Table 1.

| Vowels and half vowels | | | | | | | |
|---|---|---|---|---|---|---|---|
| ch | tag | ch | tag | Ch | tag | ch | Tag |
| අ | a | ආ | axa | ඇ | xca e | ඈ | Aea e |
| ඉ | i | ඊ | ixi | උ | u | ඌ | Uxu |
| ඍ | zri | ඎ | zrii | ඏ | zilu | ඐ | ziluu |
| එ | e | ඒ | eze | ඓ | ai | ඔ | O |
| ඕ | oxo | ඖ | xau | ං | xon | ඃ | Zkf |
| Consonants, modifiers and special characters | | | | | | | |
| ද | dh | ඵ | zp | ක | k | ඥ | jhcn |
| ග | g | ට | t | ඩ | d | ජ | xjhx |
| න | n | ප | p | බ | b | ඩ | zndx |
| ඳ | qnd h | ඹ | zch | භ | zng | ෂ | zsh |
| ඩ | zon | ත | txh | ඣ | zjh | ඪ | zdx |
| ථ | zth | ධ | zd h | ඹ | xmb | ණ | zn |
| ළ | zl | ඛ | zt | ඤ | cn | ම | M |
| ය | jh | ශ | sh | ච | ch | ඃ | Y |
| ස | zg | ඛ | zk | හ | zb | ර | r |
| ල | l | ව | v | ෂ | s | ෆ | f |
| හ | h | ෙ ෘ | zau | ොෝ | zai k | ෘෘ | zruu |
| ්a | Zru | ෟ | w | zwj | qx | - | - |

Table 1 – Phonetic tag set used in the research

Two algorithms are developed to do regular expression based search and replace to convert phonetic English to Sinhala Unicode and vice versa. Phonetic tags are searched from the large tag to small tag. The character sequences that are required to display rakaransha, yansaya and repaya modifiers [14] are not uniform across different keyboard/font drivers. Therefore support to decode such character sequences in Sinhala online news papers are included in the above conversion algorithms.

We have used two text transcriptions (text corpuses) in this research. First transcription is the set of 3195 sentences which is used for recording voice of a single speaker. The sentences are chosen carefully to accommodate all the phonetic tags of the Sinhala alphabet. LM built from the 3195 sentences has yielded only 12000+ distinct words for phonetic dictionary.

It was observed that a large LM could be used if both acoustic model and LM has all the phonetic sounds of the alphabet. Hence a second transcription (text corpus) of 26000+ sentences is compiled from the articles published in the Sinhala online news papers and online blogs by manually removing the HTML tags and the numbers written in digits in the text. Text corpus is then converted into phonetic English and fed to the sphinx LM building tools. Second transcription has produced a Sinhala LM with 50000+ distinct Sinhala words and it is used in the web portal.

### 3.3 Training a Single User Acoustic Model

A text transcription of 3195 sentences (training set) is converted to phonetic English and each sentence is enclosed with <s> and </s> tags. Voice clips of those sentences are recorded at the sampling rate of 16 kHz by a single speaker. Then sphinx train binaries [ref] are used to train an acoustic model for

a single speaker ASR. The acoustic voice model trained from 3195 voice clips (6.3 hours of voice data) is used as the core acoustic model of the portal. Training parameters such as number of senons and number of training iterations are changed to train the best acoustic model for the current voice dataset.

### 3.4 Developing a Portal with Sinhala ASR

An HTML template and a PHP script [16] are used to develop home page (Fig. 1) and user management system of the portal. Registered users are presented with six web pages, namely, profile home page (Fig. 2), perform live Sinhala ASR (Fig. 3), Record voice to make their own acoustic model (Fig. 4), initiate voice model training (speaker adaptation) (Fig. 5), assigning their own acoustic model for ASR (Fig. 6) and participate in building speaker independent acoustic model (Fig. 7) respectively.

The portal needs to create audio files in the web server using the privileges of the system user "apache" when a recording request receives from user's web browser. A PHP script, which only creates files when a user has logged on and called through a Flash-based voice recorder is developed for the task. The permissions for files and folders are set on the server side to prevent directory browsing and execution rights on audio file folder to prevent rogue users from creating executable scripts on the web server.

### 3.5 Recording Audio in the Web Browser

Our requirements are to record audio captured from user's microphone via web browser, send it to the server for ASR and subsequently display Sinhala output back in the browser. Moreover it is required to obtain voice recordings at the sample rate of 16 kHz for sphinx ASR.

We have used the WAMI recorder [15] which is a Flash-based voice recorder that can be embedded in an HTML page with several extendable exposed functions. WAMI recorder can record voice at the web browser and send chunks of data to an audio file in the server side. The WAMI recorder gives the advantage of adjusting the recording sampling rate of the client pc at the initialization stage of recorder, a facility which HTML5 still does not support.

### 3.6 Playback of Audio in the Web Browser

WAMI does not support playback of audio having a sample rate less than 22 kHz due to a limitation of Adobe Flash player. Hence the recorded voice samples are played back using an HTML5 based audio player. One problem associated with this playback is that if voice is recorded to the same filename for the second time, browser caching plays the old file instead of the new file. To avoid this issue, dynamic generation of file names for each recording is used and a cleanup CRON job is scheduled on the server to prevent temporary audio file accumulation at the server side.

### 3.7 Calling ASR Functions with WAMI

The exposed functions of the WAMI recorder, namely, initialization, "play start", "play end", "record start" and "record end" are modified to have interaction with PHP pages used for the Live ASR, recording voice for speaker adaptation, produce ASR output and in the page for participation in voice recording for speaker independent model. AJAX is used to call PHP scripts at the server side which executes sphinx binaries to perform ASR on voice recording of the user at the server side and to send generated ASR output back to the browser of a client PC.

A PHP function is developed to convert ASR output of Sphinx speech decoder in phonetic English back to Sinhala Unicode using the algorithm given below.

/* Phonetic English to Sinhala Unicode conversion algorithm. Numbers written as words are allowed e.g. 2 - is not allowed, two - is allowed. CC = current character, M - modifier corresponding to a vowel, OT = Unicode Sinhala output */

```
While not end of CA
  if current character is a space then append it to OT
  else if CC is a consonant followed by a vowel then
    append CC and  M to OT
  else if C is a half vowel then
    append half vowel to OT
  else if next 3 characters are "wqx" then
       go to smodifier section
  else append al modifier (" ්") to OT
smodifier:
// rakaranshaya section
  if next four characters are "wqxේ" then
  if the fifth character is "ෆ" then
     append CC and " ්ේ" to OT
  else if the fifth character is a vowel then
     append CC, " ්ේ," and M to OT
  end if
//end of the rakaranshaya section
Repeat the rakaranshaya section by replacing ේ to ෛ for
yansaya and conjuncts of ෙ, ෝ respectively
//repaya section
  if next four characters are "ේwqx" then
    if the fifth character is "ෆ" then
       append "ර්," zwj, and CC to OT
    else if the fifth character is a vowel then
      append "ර්," zwj, CC and M to OT
  end if
 end if
end while
return OT // OT has the Unicode Sinhala output
```

### 3.8 Speaker Adaptation

The single acoustic model deployed at the server is copied to each user profile. When a user wants to create their own acoustic model, he/she has given a list of 125 sentences to record their voice. These sentences are selected to contain all the alphabetical characters in the Sinhala alphabet. Once a person completes the set of 125 recording, he/she can initiate the acoustic model adaptation process. This will use MAP and MLLR adaptation commands of the sphinx toolset to convert single user acoustic model to a model customized to new user's voice.

It is important to note that the voice clarity of user recordings, correctness of the pronunciation of sentences and the minimal background noise improves the success of the speaker adaptation process.

## 4.0 RESULTS AND DISCUSSION

The command line functionality of sphinx ASR toolset in the Linux server is tested first. Then the ASR execution and result output processes through PHP scripts are tested. Subsequently, the portal is tested for the proper functionality in user registration / login process, Single user live ASR, audio recording/playback, acoustic model assignment for ASR and recording of audio for the speaker independent model.

This portal is tested for the proper functionality in Microsoft Internet explorer 10, Microsoft Edge, Mozilla Firefox and Google Chrome browsers with the Adobe flash player 20 or above in Windows 10, 8 and 7 environments. Recording errors were observed only in the Firefox browser in the Windows 10 environment.

### 4.1 Sinhala Language Model

Sinhala language model, which is generated with CMU-Cambridge language modeling toolkit has the following characteristics as given in Table 2.

| | |
|---|---|
| Number of sentences | 26721 |
| Vocabulary | 50224 |
| Number of Phones | 62 |
| 1-gram | 50224 |
| 2-gram | 251174 |
| 3-gram | 327590 |

Table 2 – Characteristics of Sinhala LM

We have observed that Sinhala LM can be further enhanced by adding more sentences and words while keeping the same acoustic model when an acoustic model is built with all the phonetic character sounds of the Sinhala alphabet. Spell checking the words and grammar checking the sentences of the text corpus produces a language model which leads to a correctly spelled and grammatically correct recognition output.

### 4.2 Sinhala Acoustic Model

Voice recordings of 3195 sentences by a single speaker are used to train an acoustic model which generally takes over 40 minutes for a single training iteration. Test set of 100 sentences (895

words) recorded by the same speaker is used to check the accuracy of the model. Multiple training iterations are carried out to minimize the word error rate (WER) [16] on trial and error basis by changing the training parameters. The training parameters are altered based on the reference values given in [10].

It is observed that 1750 senons, 16 training iterations provides the lowest WER of 17.2% for the test set of 100 sentences. Hence the accuracy of the single user acoustic model stands at 82.8%. It is important to note that the number of senons and the number of training iterations are not universal values for any voice corpus in Sinhala. Those values need to be adjusted on trial and error basis to accommodate any addition to the voice corpus. Sphinx developers have recommended using 10 hours of voice data for a single user acoustic model for continuous dictation type ASR, but 6 hours of voice data used in this research has provided adequate recognition accuracy.

### 4.3 Results of the Speaker Adaptation

Voices of five registered users who have recorded at least 50 sentences using the web portal are adapted using MAP and MLLR adaptation commands available in the Sphinx ASR toolset. The WER is calculated by downloading recorded files, adapted acoustic model and subsequently running pocket sphinx batch processing binary for a sample of 15 sentences. A total of 180 words were present in the test sample and the results are given in Table 3.

| User | Gender | WER | Accuracy |
|---|---|---|---|
| 1 | M | 17.78% | 82.22% |
| 2 | M | 15.00% | 85.00% |
| 3 | M | 23.33% | 76.67% |
| 4 | F | 24.44% | 75.56% |
| 5 | F | 22.14% | 77.86% |

Table 3 – Results of the speaker adaptation

The results indicate that the speaker adaptation process performs accurately for the given test sample. The limited user participation in recording sentences indicated the need of reducing the overhead of recording 125 sentences.

### 4.4 Performance Constraints of ASR

The Internet bandwidth constraints between server and the client PC affects the response time of record / playback buttons of the portal because those buttons get enabled only after the completion of uploading recorded audio file to the server. Hence, the sentences with more than 10 seconds of audio length will have delays of over one second in getting record/playback buttons enabled.

The speaker adaptation process performs as expected when there is a clear demarcation between user's voice and background noise and when having an amplification of user's voice closer to 0 dB. The system can tolerate background noise such as noise generated from a fan. We have observed that the acoustic model building process has failed for two users registered in the portal because they have not recorded their voice with enough amplification.

## 5.0 CONCLUSIONS

Web based Sinhala ASR presented in this research is the first reported attempt for Sinhala language and it also supports multi-user speaker adaptation. The system performs well with the short sentences, but for lengthy sentences, Internet bandwidth constraints between the user's PC and the server induce delays over one second in activating record/playback buttons.

This portal allows users to build acoustic models matching their own voice without special software installation on their PCs. Such models can be re-used with a compatible desktop ASR. The initial overhead of training one's voice to the system makes it difficult for the "off the shelf" use of the web portal. Hence, the voice corpus building system provided in this portal can be used as an audio recording platform to build a speaker independent acoustic model for Sinhala language.

This portal is a valuable tool for applications such as dictation and word processing. Also, it can be used as a tool to write in Sinhala by the users who can speak in Sinhala but unable to write in Sinhala language.

## 6.0 ACKNOWLEDGEMENTS

This project is funded by the National science foundation of Sri Lanka under the technology grant number TG/2014/Tech-D/02.

The authors also wish to acknowledge all the users of the ASR web portal who have recorded their voices to build their own acoustic models through the portal, making it possible for us to test the speaker adaptation.

## 7.0 REFERENCES

1) Lee K. F., et al., "The SPHINX speech recognition system." 1989.

2) Lee A., Kawahara T., and Shikano K., "Julius - An open source real-time large vocabulary recognition engine," in EUROSPEECH2001, Denmark, 2001, pp. 1691–1694.

3) Young S., "The HTK hidden Markov model toolkit: Design and philosophy," Cambridge Univ. Eng. Dept, Tech. Rep. CUED/F-INFENG/TR152, UK, 1994.

4) Gales M. and Young S., "The application of hidden Markov models in speech recognition," Foundations and Trends in Signal Processing, vol. 1, no. 3, 2007, pp. 195–304,.

5) Shrawankar U. and Thakare V. M., "Techniques for feature extraction in speech recognition system: A comparative study," arXiv preprint arXiv:1305.1145, 2013.

6) Lee C. H., and Gauvain J., "Speaker adaptation based on MAP estimation of HMM parameters," in ICASSP-93, vol. 2, 1993, pp. 558–561.

7) Tamura M, Masuko T, Tokuda K, Kobayashi T., "Speaker adaptation for HMM-based speech synthesis system using MLLR." In Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis, 1998.

8) Nadungodage T., and Weerasinghe R., "Continuous Sinhala speech recognizer," in Conference on Human Language Technology for Development, Egypt, 2011, pp. 2–5.

9) Huggins-Daines D., et al., "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in ICASSP 2006, vol. 1, France, 2006, pp. 1185–1188.

10) (2015). CMUSphinx Tutorial for Developers. [Online]. Available: http://cmusphinx.sourceforge.net/wiki/tutorial

11) Wasala A., and Gamage K., "Research report on phonetics and phonology of Sinhala," Univ. of Colombo School of Computing, Working Papers 2004-2007, Sri Lanka, 2007. [Online]. Available: http://www.columbia.edu/~kf2119/SPLTE1014/Day%203%20slides%20and%20readings/SinhalaPhoneticsandPhonology.pdf

12) Language Technology Research Laboratory, Univ. of Colombo School of Computing (UCSC), Sri Lanka. Sinhala Syllabification Tool. [Online]. Available: http://ucsc.cmb.ac.lk/ltrl/?page=downloads&lang=en&style=default

13) Sinhala Character Code for Information Interchange - Revision 3, Sri Lanka Standards SLS1134:2011, 2011.

14) The Unicode Standard Version 7.0, Sinhala Range: 0D80–0DFF, 2014.

15) (2011), WAMI-recorder. [Online]. Available : https://code.google.com/archive/p/wami-recorder/

16) Khodke P., "Login Registration with Email Verification, Forgot Password using PHP", [Online]. Available : http://www.codingcage.com/2015/09/login-registration-email-verification-forgot-password-php.html

17) Levenshtein V. I., "Binary codes capable of correcting deletions, insertions, and reversals," Soviet Physics Doklady, vol. 10, no. 8, 1966, pp. 707–710.