(5) category \ 5th/ 2016

Unicode Sinhala and Phonetic English Bi-directional Conversion for Sinhala Speech Recognizer

M. Punchimudiyanse
Department of Mathematics and Computer Science
The Open University of Sri Lanka
Nawala, Sri Lanka
malinda@ou.ac.lk

R. G. N. Meegama
Department of Computer Science
University of Sri Jayawardenapura
Gangodawila, Sri Lanka
rgn@sci.sjp.ac.lk

***Abstract*—An automated speech recognizer (ASR) having a large vocabulary is yet to be developed for the Sinhala language because of the time consuming nature of gathering the training data to build a language model. The dictionary and building the language model require non-English text, in our case, Sinhala Unicode, to be transcribed in phonetic English text. Unlike text to speech conversions which only require transcribing the non-English text to phonetic English text, an ASR needs correct reproduction of the original language text when the phonetic English text is produced as the output of the speech recognizer. In the present research, newspaper articles are used to gather a large set of sentences to build a language model having thousands of words for the Sphinx ASR. We present a decoder algorithm that produces phonetic English text from Sinhala Unicode text and an encoder algorithm that produces the correct reproduction of Unicode Sinhala text from phonetic English. For a near phonetic tag set for Sinhala alphabet, results indicate 100% accuracy for the decoder algorithm while for numberless text, accuracy of the encoder algorithm stands at 98.61% for distinct phonetic English words.**

Keywords—Sinhala to Phonetic English, Phonetic English to Sinhala, Sinhala ASR, Sinhala Phonetic Tag set, Sphinx

I. INTRODUCTION

Commercial and open source products to automatically recognize spoken English is available in the market today. In terms of performance and the accuracy, ASR systems that are based on acoustic and language models utilize hidden markov model (HMM) or Gaussian mixture models (GMM) for recognition. Most popular such open source ASR systems include Sphinx, HTK and Julius [1-3]. HMM based ASR systems typically have two phases, namely, the training phase and the recognition phase [16]. Onetime training phase is used in which a single or multiple user's voice is utilized to train a proper speech model called an acoustic model. In the recognition phase, the speech model built in the previous phase is recurrently used to recognize spoken words of the speaker.

The process of training a speech model requires a collection of acoustic data, a recorded voice sample of a person, and the correct phonetic text transcription of that voice sample. Adopting this technique for systems built for non-English languages require a production of non-English words in phonetic English text for the speech trainer. In order to build the phonetic text transcription from the original Unicode text, a proper phonetic tag per character is necessary.

Sinhala Unicode text to phonetic English tagging (syllabification) is not a straightforward task because Sinhala characters have several modifiers that are typed after the actual base character but appears before, after, top and the bottom of the actual base character. In addition, a non printable zero-width joiner (Unicode U+200D) is inserted between the characters to convert certain character combinations to one of the modifiers named yansaya, rakaranshaya or repaya which are not present in Sinhala Unicode.

For an ASR application to work, both Sinhala Unicode to phonetic text conversion and proper reproduction of Sinhala Unicode from phonetic English output of the speech recognizer is necessary. Phonetic tags for Sinhala alphabet has to be searched and replaced in a specific order to prevent invalid construction of words from phonetic English. For example, if phonetic tag for the characters ඉ is i and ඊ is ii with search order i, ii is used to encode a phonetic English word giithaya (meaning song), the word is constructed as (ග් + ඉ + ඉ + ත + ය) ගිඉතය which is misspelled. Correct spelling of the word ගීතය will be constructed if the search order ii,i is followed.

The ASR applications use N-gram based techniques to construct word combinations to build a language model in phonetic English. Therefore, it is necessary for the researchers to identify suitable phonetic tag set to develop a Sinhala to phonetic text decoder algorithm and phonetic English to Sinhala Unicode encoding algorithm.

II. BACKGROUND AND RELATED WORK

Representation of Sinhala characters in digital form commenced in early 1980s to display Sinhala subtitles in television. The Wijesekara keyboard used in the type writers are extended to Sinhala fonts and keyboard drivers used in the computers. The Sinhala alphabet is first standardized by the Sri Lanka standards (SLS) institute as Sinhala Character Code for Information Interchange (SLS 1134:1996) and subsequently revised in 2004 and 2011 (SLS 1134:2004, SLS 1134:2011) [5]. The Sinhala alphabet representation in Unicode is first added to the Unicode standard version 3 based on the ISO/IEC 10646 [4].

The Sinhala language has 18 vowels, 2 half vowels and 40 consonants in modern Sinhala alphabet of 60 characters [6]. The consonant letter ඦ is present to make it 61 characters in the Unicode standard. In addition, there are several modifiers placed after a consonant which corresponds to vowel sounds [7]. The first well documented syllabification effort claims an accuracy of 99.95% and employs a rule based algorithm to produce the output in English characters using the International Phonetic Alphabet (IPA) format [9]. The IPA format for the Sinhala [12] consists of characters that are not part of the 26 letter English alphabet. A different tag set is proposed to represent phonetic sounds for the Sinhala alphabet with an accuracy of 98.21% for the purpose of text to speech (TTS) applications utilizing the festival framework [13].

A syllabification tool published by the University of Colombo School of Computing (UCSC) also employs a slightly modified tag set to the one that is proposed in [13] to convert Sinhala Unicode text to phonetic English text where the output of each character is separated by a space [14].

Although the tag set in [13] work well with TTS, it presents several problems for ASR applications as it does not suggest different tags for characters with similar sounds but different characters. Therefore, phonetic English text generated by those tag sets make wrong classifications of acoustic sounds in the training phase of the ASR and make erroneous output when building a language model as well. The reason behind this scenario is that the words spelled correctly are replaced by misspelled words during the conversion from Sinhala Unicode to phonetic English producing errors in the ASR output.

This research paper presents an improved tag set to decode Sinhala Unicode text gathered from online Sinhala news paper articles to phonetic English text with a decoder algorithm and an encoder algorithm to convert phonetic English to Sinhala Unicode text in an ASR application.

III. Methodology

The Sphinx ASR speech trainer tool (training phase) requires Sinhala Unicode sentences to be represented in phonetic English format. In addition, it requires a list of distinct words with its phonetic pronunciations separated by a space as the pronunciation dictionary.

A. Special Features Associated with Phonology of Sinhala

Technical explanation of Sinhala Unicode to phonetic English decoder algorithm is not fruitful without giving specifics of how Sinhala language words are constructed from alphabetical characters. In addition to the occurrences of vowels and base consonants, one or more modifier is typed after a consonant but the appearance of the modifier could be before, after, top and bottom of a particular consonant.

ක් ක කා කැ කෑ කි කී කු කූ කෙ කේ කො කෝ

කෘ කෲ ක්‍ර ක්‍රා ක්‍රෙ ක්‍රේ ක්‍රැ ක්‍රෑ ක්‍රි ක්‍රී ක්‍රො ක්‍රෝ

කෞ කං කඃකෛ

Fig. 1. Derived consonents from the consonent ක.

The pronunciation of the derived consonant varies with the associated modifiers. Most of the modifiers are inheriting the sounds of a vowel. The al modifier (්) does not have a vowel associated to its sound. A word in Sinhala language is constructed from a set of phonetic sounds related to the alphabet. Phonologically a base consonant (termed as pure consonant) is denoted as a consonant with a al (්) modifier.

For example the consonant ක (ka) consist of ක් + අ (k + a) sounds where ක් (k) is consonant with al (්) modifier and අ (a) is a hidden vowel according to phonetic pronunciation. Therefore, a consonant written without a modifier is having the hidden vowel sound අ (a) associated with it. The set of derived consonants attaching one or more modifiers after the consonant ක (ka) is depicted in figure 1 [9, 10].

Three special modifiers, namely, rakaranshaya, repaya, yansaya are used with consonants but do not have a corresponding character code in Unicode standard. When a specific character and modifier combination is encountered after a consonant, such sequences are immediately replaced with a respective special character or modifier by using either a font driver or a keyboard driver of the Sinhala language [17].

There are conjuncts (two or more letters combined to form a single letter) present in some of the words which are not compulsory to write in that manner. If a conjunct is detected in a word, it has to be processed by the decoder to produce correct phonetic English and reproduced in the same manner by the encoder that produces Unicode text.

Those three special modifiers, as well as the conjuncts, require a special non-printing character called a zero-width joiner (ZWJ) inserted appropriately to build a proper character sequence in Sinhala Unicode text [8]. The character combinations used in special modifiers and two common conjuncts are given in the Table I.

When designing an algorithm to decode Sinhala Unicode to phonetic English, the features present in a standard text of Sinhala language need preservation across the decoding process. A sentence is read by the decoder algorithm and Sinhala characters and modifiers are replaced with the relevant phonetic tag. The reconstruction process of the recognized phonetic English text output of ASR to Sinhala Unicode text will fail if those features are overlooked or lost at the decoding process.

TABLE I. Special Modifiers and Conjunct

Derived consonants	Type	Required Character Sequence
ක්‍ර	Modifier = rakaranshaya	ක් + ZWJ+ ර + අ
ක්‍රී	Modifier = rakaranshaya, ී	ක් + ZWJ+ ර + ඊ
ක්‍ය	Modifier= yansaya	ක් + ZWJ + ය
ර්‍ණ	Modifier= repaya	ර + ZWJ + ණ
ක්‍ෂ	Conjunct	ක් + ZWJ + ෂ
න්‍ධ	Conjunct	න් + ZWJ + ධ

B. Improved Phonetic Tag Set

Based on the initial tag set present in [13, 14], several tags are changed to eliminate ambiguity in characters with similar phonetic sounds but having different Unicode character codes. The near phonetic tag set used in this research is given in Table II.

Several new tags are introduced to prevent actual vowels being misinterpreted as modifiers. The phonetic tag used for a particular modifier for a corresponding vowel is given in Table III. The character form derived by adding al modifier (◌්) to a consonant is considered as the base form (pure consonant) for breaking down a Sinhala word to its phonetic pronunciation.

The Sinhala to phonetic English decoder algorithm is planned to generate the phonetic form automatically from an input character sequence so that it will fulfill the requirement of the Sphinx ASR trainer tools. The encoder algorithm used to reconstruct Sinhala Unicode word from phonetic English output of the speech recognizer also utilizes the same phonetic pronunciation format.

TABLE II. SINAHALA ALPHEBHET WITH IDENTIFIED PHONETIC TAGS

Character	Phonetic Tag	Character	Phonetic Tag	Character	Phonetic Tag
Vowels					
අ	a	ආ	axa	ඇ	xcae
ඈ	aeae	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
Consonants and Zero Width Joiner					
ද	dh	ඵ	zp	ක	k
ග	g	ට	t	ඩ	d
න	n	ප	p	බ	b
ඥ	jhcn	ඣ	xjhx	ඬ	zndx
ඳ	qndh	ඡ	zch	ඟ	zng
ඞ	zon	ත	txh	ඦ	zjh
ඨ	zth	ධ	zdh	ඹ	xmb
ෂ	zsh	ඪ	zdx	ණ	zn
ළ	zl	ථ	zt	ඤ	cn
ජ	jh	ශ	sh	ච	ch
ඝ	zg	ඛ	zk	භ	zb
ම	m	ය	y	ර	r
ල	l	ව	v	ස	s
ෆ	f	හ	h	zwj	qx

TABLE III. PHONETIC TAGS FOR MODIFIERS AND MATCHING VOWELS

Vowel	Matching modifier	Phonetic Tag	Vowel	Matching modifier	Phonetic Tag
-	◌්	w	ආ	◌ා	axa
ඇ	◌ැ	xcae	ඈ	◌ෑ	aeae
ඉ	◌ි	i	ඊ	◌ී	ixi
උ	◌ු	u	ඌ	◌ූ	uxu
එ	◌ෙ	e	ඒ	◌ේ	eze
ඔ	◌ො	o	ඕ	◌ෝ	oxo
ඍ	◌ෘ	zru	ඎ	◌ෲ	zruu
ඓ	◌ෛ	zaik	ඖ	◌ෞ	zau
-	◌ං	xon	-	◌ඃ	zkf

C. Sinhala Unicode to Phonetic English Decoder Algorithm

The numbers in digit form (0-9) and HTML tags present in the Sinhala Unicode sentences that are copied from the online newspaper articles are removed first. The cleaned up text is processed using the following decoder algorithm. Numbers written in textual form such as එක (one) in text are allowed.

DECODER ALGORITHM

```
//This algorithm converts Sinhala Unicode text to phonetic
//English text in order to use with sphinx based Sinhala ASR
// abbreviations: zero width joiner - ZWJ, carriage return - CR
tr = input_string
//sinhala character and phonetic tag placed in three categories
vowels = "අ-a,ආ-axa,..,◌ං-xon"
joiners = "◌ා-axa,◌ි-i,◌ී-ixi,◌ු-u,..,◌ෲ-zruu"
conson = "ඡ-zch,ඟ-zng,..,ෆ-f"
//processing section
split vowel to two lists vowel & phoneticvowel
split conson to two lists consonent & phoneticconsonent
split joiner to two  lists modifier and phoneticmodifier
remove non printable keeping ZWJ and CR
repeat until end of vowellist
tr =  tr.replace(vowel,"@"+ phoneticvowel +"#")
repeat until end of consonentlist
tr =  tr.replace(consonent,"$" & phoneticconsonent & "!")
repeat until end of modifierlist
tr =  tr.replace(modifier,"-" & phoneticmodifier & "~")
//zero width joiner replacement
tr = tr.replace(ZWJ,"$qx#")
//introduce "a" sound for non al consonents
tr = tr.replace("!$", "!-a~")
tr = tr.replace("!@", "!-a~@")
tr = tr.replace("! ", "!-a~ ")
tr = tr.replace("!" & CR, "!-a~" & CR)
tr = tr.string.replace(CR, " </s>" & CR & "<s> ")
//remove all special characters except /, < , >
tr = tr.replace("[\\!@#$`%^&*."",()_+=~|{}?;:\['\-\]]", "")
add <s> to the start of tr
remove last <s> from the tr
return tr
```

Decoder algorithm is capable of handling multiple sentences separated by linefeed. The tags <s> and </s> is a requirement to build sentence transcription of sphinx language model building tool.

D. Phonetic English to Sinhala Unicode Encoder Algorithm

A longer algorithm required to encode phonetic English to Sinhala Unicode because the features mentioned in 3A has to be considered.

```
//This algorithm converts phonetic English to Sinhala Unicode
//Phonetic English text is without numbers, but number written
//in text is possible. eg. 1 - is not allowed, one - is allowed

ra = input_phonetic_english_text

//data preparation section
vowel() = vowel list in sinhala unicode
modifier() = modifier list sinhala unicode
consonent() = consonant list sinhala unicode

csingle = unicode character list with single letter phonetic tag
cdual = unicode character list with two letter phonetic tag
ctri = unicode character list with three letter phonetic tag
clarge = unicode character list with large phonetic tag
singlephone = single letter phonetic tag list
dualphone =  two letter phonetic tag list
triphone = three letter phonetic tag list
largephone = phonetic tag list with more than 3 letters for tag

//replace tag with Unicode character in the order of tag size

repeat until eof largephonelist
ra = ra.replace(largephone,clarge)
repeat until eof triphonelist
ra = ra.replace(triphone, ctriple)
repeat until eof dualphonelist
ra = ra.replace(dualphone, ,cdual)
repeat until eof singlephonelist
ra = ra.replace(singlephone, cingle)

//make a character array to do linear search to convert
//vowels that immediately follow a consonant to modifiers
cta() = ra.ToCharArray()

//ch - current char, cc - counter, kk, kkr - tempcharacter
while not end of input_phonetic_english_text
  if current char  = space then
    sout += space
  end if
  if ch is a consonant then
    check next character cta(cc+1) for in vowel()
    if vowel found
         next character = modifier
          kkr = modifier
    else
         check next character for modifier without vowel
          if found then
               kk = current char & next char
               Increment character count by 1
          else
               if current character is not in ("ං", "ඃ" ) then
                  if next 3 characters are "wqx" Then
                     GoTo rakaransha_section
                  end If
                  kk = current char & "්"
               else
                   kk = current char
               end if
          rakaransha_section:
          //rakaranshaya handling section
          If next 4 characters are "wqxර" Then
            kk = current char + "්ර"
            If fifth character from current char = "අ" Then
               Increment character count by 5
             else // otherwise check the additional modi
                 check 5th char from current position for vowel
                 if found fifth character from current = modifier
                    kk += modifier
                    Increment character count by 3
                  end if
                  Increment character count by 1
              end If
          //end of rakaransaya handling section
          repeat rakaransha section changing  ර to ය, ෂ for
          yansaya and conjuncts of ෂ,ධ respectively

        //repaya handling
          If next 4 characters are "රwqx" Then
            kk =  "ර්" + cta(cc+4)
            If fifth character from current char = "අ" Then
               Increment character count by 5
             else // otherwise check the additional modi
                 check 5th char from current position for vowel
                 if vowel found
                    Fifth character from current char = modifier
                    kk += modifier
                    Increment character count by 4
                  end if
                  Increment character count by 1
              end If
           end if
          sout  += kk
        else
sout = sout + current char + kkr
else
  if current character is not in modifier then
     sout = sout + current char
end if
Increment current character count by 1
end while
return sout     //sout has the Sinhala Unicode output
```

IV. Results and Discussion

Sinhala to phonetic English algorithm (decoder) and phonetic English to Sinhala Unicode (encoder) algorithm given in sections 3C and 3D are implemented and tested with different test cases.

A. Sinhala Unicode to Phonetic English Decoder Algorithm

The functionality of the Sinhala Unicode to phonetic English decoder algorithm is tested with several phases. In the first phase, it is checked for accurate generation of phonetic English text from a given sentence. Test Results are described without <s> and </s> tags which is a format requirement of the Sphinx trainer.

A sentence රේල් පාරේ පයින් ගමන් කිරීම අනතුරුදායකය (meaning: It is risky to walk in the rail track) is processed using the decoder algorithm produces the output as rezelw paxareze payinw gamanw kirixima anatxhurudhaxayakaya. It correctly produces the phonetic text by ordering the phonetic tags of the test sentence in the following character sequence: ර්ඒල්w ප්ආර්ඒ ප්අය්ඉන්w ග්අම්අන්w ක්ඉර්ඊම්අ අන්අත්උර්උද්ආය්අක්අය්අ.

The word රේල් (first word of the above sentence) is constructed as "ර්ඒල්w" by decoding it to phonetic sounds of the base character and its modifier. රේ becomes ර් + ඒ and ල් becomes ල් + w in the word රේල්. The tag w implies that the consonant does not have a hidden vowel අ (a) present. Therefore, the word රේල් becomes ර්ඒල්w and subsequently, rezelw and the rest of the words are processed with the same notation.

The second phase of testing the decoder algorithm involves generating phonetic text in the presence of special modifiers (repaya, rakaranshaya and yansaya) and conjuncts (combined letters). Table IV depicts the decoding process of the words වක්‍රය (curve), අව්‍යාජ (truthful), වර්ණ (colors) and අක්ෂර (letters) tested with online Sinhala typing application "real time Unicode converter" by UCSC [15]. This algorithm converts the words to proper phonetic English format.

The initial results indicate that the decoder algorithm generates phonetic English from Sinhala Unicode in the expected format of the Sphinx ASR trainer. A test set of 500 sentences is gathered from the Divaina, Silumina and Lankadeepa online editions of Sinhala newspapers which are published in the Sinhala Unicode font to test bidirectional conversion between phonetic English and Sinhala Unicode text. The numbers present in the text in digits (0-9) are manually replaced by the word form of the numbers.

A test set of 500 Sinhala Unicode sentences is processed using the decoder algorithm to produce the phonetic English transcription within 3 seconds in a personal computer having a processing speed of 2.7 GHz.

TABLE IV. PHONETIC ENGLISH CONVERSION OF SPECIAL MODIFIERS AND CONJUNCTS

Derived consonants	Modifier	Phonetic Form	Phonetic tag separated
වක්‍රය	rakaranshaya	vakwqxraya	v a k w qx r a y a
අව්‍යාජ	yansaya	avwqxyaxajha	a v w qx y axa jh a
වර්ණ	repaya	varwqxzna	v a r w qx zn a
අක්ෂර	conjunct	akwqxzshara	a k w qx zsh a r a

B. Phonetic English to Sinhala Unicode Encoder Algorithm

The phonetic English to Sinhala encoder algorithm is tested in several ways. The first test involves carrying out proper reproduction of 61 alphabetical letters from their respective phonetic tags each separated by a space character. The results indicate an accuracy of 100% in this regard.

The next test is carried out for a sentence containing words without special modifiers. The sentence "sixonhalenw katahazndxa haqndhunaxa gxcaeniximeze mzrudhukaxaxongaya saqndhahaxa mema xcaelwgoritxhama dheka upakaxarixi veze" (meaning: these two algorithms helpful in Sinhala ASR software) is encoded using phonetic English to Sinhala Unicode algorithm. It produces the accurate output "සිංහලෙන් කටහඬ හඳුනා ගැනීමේ මෘදුකාංගය සඳහා මෙම ඇල්ගොරිතම දෙක උපකාරී වේ".

A sentence with special modifiers is tested using the same technique. The sentence "puxujhwqxya pakwzshaya chakwqxralezezkaya sahitxha dhuxmburu varwqxznayenw yutxhu kavaraya vzaikdhwqxyavarayaxata zbaxara dhunwnezeya" (meaning: The brown colored cover containing the circular handed over to the doctor by the clergy) is properly encoded by the encoding algorithm which produces the output as "පූජ්‍ය පක්ෂය චක්‍රලේඛය සහිත දුඹුරු වර්ණයෙන් යුතු කවරය වෛද්‍යවරයාට භාර දුන්නේය".

Researchers have observed that the phonetic English text generated by decoder algorithm using the Sinhala text typed from real time Unicode converter application[15] and the Google input tools provided a 100% accurate output in the encoder algorithm for the three special modifiers yansaya, rakaranshaya and repaya. The encoding algorithm accurately supports one conjunct that is the combined letter ක්ෂ which is ක් and ෂ added together. Testing with the text converted from other online sources produced errors which will be discussed in the next section.

The output of Sinhala Unicode text from phonetic English source sentence is a continuous effort of experimental comparisons in identifying a proper near phonetic tag set for the Sinhala alphabet. It is also observed that the search order of the phonetic tags has to be from the largest to the smallest. .

C. Testing the Bi-directional Conversion

At the final stage, the bidirectional capability of the two algorithms is tested using the following three steps.

Step 1: A test set of Sinhala Unicode text is processed by the decoder algorithm to obtain phonetic English text. *Step 2*: The phonetic text is fed as the input to the encoder algorithm to obtain Sinhala Unicode text. *Step 3*: The sentences in the original test set are compared with the regenerated Sinhala Unicode text.

Online editions of the Silumina, Divaina and Lankadeepa Sinhala newspapers possess the least number of typographical errors. As such, Sinhala Unicode text of the articles published in such newspapers are selected for building the test set containing 500 sentences to be used in the bidirectional conversion. Note that the test set of 500 sentences are stripped of all the HTML tags.

The regenerated Sinhala Unicode output of the encoder had 46 errors from a total of 7324 words present in the original test set and the accuracy stands at 99.37%. On the other hand, if the accuracy is taken as the percentage of errors from the number of distinct words, 38 errors out of 2743 distinct words measure up to the encoding accuracy of 98.61%.

It is observed that the encoding errors occur due to the placement of zero-width joiners in the wrong sequence or placed twice in the places with special modifiers rakaransha, yansa and repaya in the original Sinhala text, which is used to generate phonetic English text. This is due to excessive use of zero-width joiners when typing the Sinhala text by the user or the way different keyboard / font driver interprets rakaransha, yansa and repaya by the Sinhala typing application.

Having identified the cause of errors, an accuracy of 100% can be reached after running the set of raw sentence through the decoder and the encoder and replacing the words containing invalid zero-width joiners with correctly typed Sinhala words.

The Sphinx ASR trainer is used to build the Sinhala language model using the phonetic English transcription generated by decoder algorithm taking Sinhala sentences as an Input. Then, the language model, along with the acoustic model, is used with the Sphinx ASR to produce continuous Sinhala voice recognition output in phonetic English. Subsequently, the phonetic English output is fed into the encoder algorithm to generate Sinhala Unicode text via a wrapper application.

V. Conclusions

Building a language model in phonetic English from Sinhala Unicode text is a tedious task in Sphinx based Sinhala ASR. In this regard, online newspapers are a valuable source of obtaining a large set of Sinhala sentences that can be used to generate a language model.

The main contributions of the proposed technique are the decoder algorithm having an accuracy of 100% for numberless text and the encoder algorithm with an accuracy of 98.61% for distinct words in making successful bidirectional conversion between Sinhala Unicode text and phonetic English

Two future improvements are suggested. First improvement is enhancement of the decoder algorithm to support inline number to text conversion with appropriate sensing of suffixes used in specifying days, positions and numbers. Second improvement is adding a touching letter (two consonants touched together without al modifier in the first consonant) conversion support as an option which are commonly used in classical Buddhist and Pali literature.

Acknowledgment

This research is funded by the National Science Foundation of Sri Lanka research grant number TG/2014/Tech-D/02.

References

[1] K.F. Lee, H.W. Hon and R. Reddy, "An overview of the SPHINX speech recognition system", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 38, no. 1, pp. 35–45, January 1990.

[2] S. Young, "The HTK hidden markov model toolkit: design and philosophy," Cambridge University Engineering Department, UK, Tech. Rep. CUED/F-INFENG/TR152, September 1994.

[3] A. Lee, T. Kawahara and K. Shikano,"Julius - An open source real-time large vocabulary recognition engine", In Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH2001), Aalborg, Denmark, pp. 1691-1694, September 2001.

[4] V.K. Samaranayake, S.T. Nandasara, J.B. Dissanayake, A.R. Weerasinghe and H. Wijayawardhana. (2003) "An introduction to UNICODE for Sinhala characters". University of Colombo School of Computing, Department of Sinhala University of Colombo, UCSC technical report 03/01, Sri Lanka.

[5] Sri Lanka Standards Institute, "Sri Lanka Standards SLS 1134 : 2011 - Sinhala character code for information interchange - revision 3", SLSI, 2011.

[6] "Sinhala Lekhana Reethiya", National Institute of Education, Maharagama, Sri Lanka, 1989.

[7] The Unicode Consortium, "The Unicode standard version 7.0, Sinhala Range: 0D80–0DFF", 2014. [Online]. Available: http://unicode.org/charts/PDF/U0D80.pdf/, [Accessed: 30-Nov-2014].

[8] G.V. Dias,"Challenges of enabling IT in the Sinhala language",27th Internationalization and Unicode Conference, Berlin, Germany, 2005.

[9] R. Weerasinghe, A. Wasala and K. Gamage, "A rule based syllabification algtoithm for sinhala", In Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05), Jeju Island, Korea. pp. 438-449, 2005.

[10] J.B. Disanayaka,"අකුරු හා පිලි (Letters and Strokes)",Colombo, S. Godage & Bros., 2000.

[11] J.B. Disanayaka,"National Languages of Sri lanka - I, Sinhala", Colombo, Department of Cultural Affairs, 1976.

[12] B. Hettige and A.S. Karunananda,"Transliteration system for English to Sinhala machine translation", In proceedings of International Conference on Industrial and Information Systems (ICIIS 2007), pp. 209-214, 2007.

[13] A. Wasala and K. Gamage. (2007) "Research report on phonetics and phonology of Sinhala ". University of Colombo School of Computing,Working Papers 2004-2007, Sri Lanka. [Online]. Available: http://www.columbia.edu/~kf2119/SPLTE1014/Day%203%20slides%20and%20readings/SinhalaPhoneticsandPhonology.pdf, [Accessed: 17-Jul-2014].

[14] "Sinhala syllabification tool", Language Technology Research Laboratory, University of Colombo School of Computing (UCSC), Sri Lanka. [Online]. Available: http://ucsc.cmb.ac.lk/ltrl/?page=downloads &lang=en&style=default, [Accessed: 14-May-2014].

[15] "යුනිකෝඩ් එසැනින් පරිවර්තකය (Unicode real time font conversion utility)", Language Technology Research Laboratory, University of Colombo School of Computing (UCSC), Sri Lanka,2006. [Online]. Available: http://www.ucsc.cmb.ac.lk/ltrl/services/feconverter/t1.html

[16] K. Seymore et al.,"The 1997 CMU sphinx3 English broadcast news transcription system", In Proceedings of DARPA Broadcast News Transcription and Understanding Workshop,1998.

[17] A.R. Weerasinghe, D.L. Herath and K. Gamage,"The Sinhala collation sequence and its representation in UNICODE",International Journal of Localization (Localization Focus), vol 5, no.1, pp 13-19, March 2006.