2006－NL－176（15）
2006／11／23

An efficient and user-friendly Sinhala input method based on phonetic transcription

Sandeva Goonetilleke †　Yoshihiko Hayashi ‡　Yuichi Itoh †　Fumio Kishino †
†Graduate School of Information Science and Technology, Osaka University
‡Graduate School of Language and Culture, Osaka University

We propose an application independent Sinhala character input method called *sri shell* with a principled key assignment that is based on phonetic transcription of Sinhala characters. We argue that a good character input method should fulfill two criteria, which are efficiency and user-friendliness. We have introduced several quantification methods to quantify the efficiency and user-friendliness of Sinhala character input methods. Experimental results are shown to prove the efficiency and user-friendliness of the proposed method.

音声表記・音韻表記に基づく効率的で使い易いシンハラ語入力方式

グナティラカ　サンデーワ †　林　良彦 ‡　伊藤　雄一 †　岸野　文郎 †
†大阪大学大学院情報科学研究科
‡大阪大学大学院言語文化研究科

われわれはシンハラ語文字の音声・音韻標記に基づくシンハラ文字入力方式*sri shell*を提案する．*sri shell*はアプリケーションに依存しない入力手段として利用可能である．良い文字入力手段は使いやすく，効率的であることが必要がある．われわれは，シンハラ語文字入力システムの使いやすさと効率性を定量化する方式を提案し，被験者実験の結果から，提案手法は従来手法に比べて同等以上の使いやすさを有しつつ，効率的であることを示す．

Keywords

Sinhala, Character input methods.

1. Introduction

Sinhala Language The mother language of 13,190,000 (66% of the total Sri Lankan population of 19,905,165) Sri Lankans is Sinhala [1]. Sri Lanka has three official languages. They are Sinhala, Tamil and English. Most of the governmental work in Sri Lanka is done in Sinhala. The Education system also uses Sinhala language up to high school or university level.

Computer Usage in Sri Lanka In Sri Lanka the usage of computers have begun to spread rapidly, due to the reduction of the price and high performance of computers. But there is no well established method to input Sinhala scripts into the computer. Even though various kinds of Sinhala font and Sinhala input applications have been proposed, still it is hard to say that Sinhala language is well supported by computer systems.

Our Objective The objective of this research is to propose an efficient and user-friendly Sinhala input method and to evaluate the efficiency and the user-friendliness comparing to other input methods.

2. Sinhala Alphabet

Śuddha Sinhala Hōḍiya (Pure Sinhala Alphabet) The pure Sinhala alphabet has thirty-seven characters (twelve vowels and twenty-five consonants). Most of the Sinhala words can be written using only these thirty-seven characters. After thirteenth century[2] Sinhala language was very strongly influenced by Sanskrit and Pāli languages. As a result of these influences lot of Sanskrit characters were added to the Sinhala alphabet.

Miśra Sinhala Hōḍiya (Mixed Sinhala Alphabet) The revised alphabet is called Miśra Sinhala Hōḍiya (Mixed Sinhala Alphabet). Mixed Sinhala alphabet consists of fifty-nine characters (eighteen vowels and forty-one consonants). The occurrence rate of these newly added twenty-two characters is lower than the thirty-seven pure Sinhala characters. However these new characters are frequently used in formal sentences. Thus these new characters are also an indispensable part of Sinhala alphabet.

Sammata Sinhala Hōḍiya (Standard Sinhala Alphabet) In the nineteenth and twentieth centuries Sinhala language was strongly influenced by Portuguese, Dutch and English languages. As a result the

Vowels

අ	ආ	ඇ	ඈ	ඉ	ඊ	උ	ඌ	ඍ	ඎ
a	ā	æ	ǣ	i	ī	u	ū	ṛ	ṝ
ඏ	ඐ	එ	ඒ	ඓ	ඔ	ඕ	ඖ		
ḷ	ḹ	e	ē	ai	o	ō	au		

(අ)ං	(අ)ඃ
(a)ṃ	(a)ḥ

Consonants

ක	ඛ	ග	ඝ	ඞ	ඟ
ka	kha	ga	gha	ṅa	ṅga
ච	ඡ	ජ	ඣ	ඤ	ඦ
ca	cha	ja	jha	ña	ña
ට	ඨ	ඩ	ඪ	ණ	ඬ
ṭa	ṭha	ḍa	ḍha	ṇa	ṇḍa
ත	ථ	ද	ධ	න	ඳ
ta	tha	da	dha	na	nda
ප	ඵ	බ	භ	ම	ඹ
pa	pha	ba	bha	ma	mba

ය	ර	ල	ව	ශ	ෂ	ස	හ	ළ	ෆ
ya	ra	la	va	sha	ṣa	sa	ha	ḷa	fa

Figure 1: Sammata Sinhala H̄ōḍya (Standard Sinhala Alphabet)

modern Sinhala alphabet includes also the 'f' sound. Modern standard Sinhala alphabet consists of eighteen vowels and forty-two consonants (altogether sixty characters), as shown in Figure 1.

2.1. Consonant vowel combinations

If a vowel appears at the beginning of a word, then that vowel is written as a independent character using "Vowel characters" shown in Figure 1. But when a vowel appears after a consonant, a special vowel sign (which is called *pilla* or *pili*) is used instead of the normal vowel character. The consonants shown in Figure 1 already include the vowel අ (=a). To get a pure consonant, a *consonant sign* or *hal-lakuṇa* should be added to the specific character. There are two shapes for the *hal-lakuṇa*. The selection of the *hal-lakuṇa* depends on the shape of the consonant.

Figure 2 shows a few examples of consonant-vowel combinations. Most of the vowel signs do not take different shapes depending on the consonant. But the vowel sign for u (p ̄apilla) takes various shapes according to the consonant.

2.2. Conjunct consonants

Traditionally when a consonant is preceded by another consonant, they are written as one conjunct consonant.

	Modern Writing	Traditional Writing
nda	න්ද	ඳ
ndha	න්ධ	ඬ
ttha	ත්ථ	ඵ

This combining process is prominently observed especially in Sanskrit or P ̄ali texts. However nowadays this is very rare except in some special cases. In the modern Sinhala text this combination is made only when the followeing consonant is "y" or "r".

Vowel		a	ā	æ	ǣ	i	ī	u	ū
Vowel Signs	්hal-lakuṇa	< *null* >	ාælapilla	ැketi ædaya	ෑdiga ædaya	ිketi ispilla	ීdiga ispilla	ුketi p ̄apilla	ූdiga p ̄apilla
ṭ sound	ට් ṭ	ට ṭa	ටා ṭ ̄a	ටැ ṭæ	ටෑ ṭ ̄æ	ටි ṭi	ටී ṭ ̄i	ටු ṭu	ටූ ṭ ̄u
p sound	ප් p	ප pa	පා p ̄a	පැ pæ	පෑ p ̄æ	පි pi	පී p ̄i	පු pu	පූ p ̄u
k sound	ක් k	ක ka	කා k ̄a	කැ kæ	කෑ k ̄æ	කි ki	කී k ̄i	කු ku	කූ k ̄u

ṛ	ṝ	e	ē	ai	o	ō	au
ෘgætapilla	ෲdiga gætapilla	ෙkombuva	ේkombuva & ්hal-lakuṇa	ෛkombu deka	ොkombuva & ාælapilla	ෝkombuva & ාælapilla & ්al-lakuṇa	ෞkombuva & ුgayanukitta
ඍ	ඎ	එ	ඒ	ඓ	ඔ	ඕ	ඖ
ටෘ ṭṛ	ටෲ ṭṝ	ටෙ ṭe	ටේ ṭ ̄e	ටෛ ṭai	ටො ṭo	ටෝ ṭ ̄o	ටෞ ṭau
පෘ pṛ	පෲ pṝ	පෙ pe	පේ p ̄e	පෛ pai	පො po	පෝ p ̄o	පෞ pau
කෘ kṛ	කෲ kṝ	කෙ ke	කේ k ̄e	කෛ kai	කො ko	කෝ k ̄o	කෞ kau

Figure 2: A few examples of Consonant Vowel Combinations

	Incorrect	Correct
mya	ම්ය	ම්‍ය
tya	ත්ය	ත්‍ය
pra	ප්ර	ප්‍ර
tra	ත්ර	ත්‍ර

Figure 3 shows all the characters (including conjunct consonants) derived from Sinhala letter ක(=ka). All other consonants also produce derivatives similarly. As a result Sinhala language has hundreds of characters.

3. Sinhala Input Systems

This section reviews the representative Sinhala input systems proposed so far.

3.1. Direct Input Method

Sinhala fonts assign Sinhala character to the ASCII character code. For example Sinhala අ (=a) was assigned to 0x61 (=ASCII 'a'). In the direct input method user has to input the character codes as assigned in a specific Sinhala font as shown in Figure 4. Some of the Sinhala fonts use character codes between 0x80 $\sim$ 0xFF. As there are no keys assigned for these character codes in the normal English keyboard, users have to refer a character code table when they want to input Sinhala text.

Later some improved versions of Sinhala fonts have been introduced. These fonts assign character code to the vowels, consonants and vowel signs. With this method it was able to reduce the binary range used between 0x20 $\sim$ 0x7F. Thanks to these fonts it became possible to type Sinhala text using a normal English keyboard. Although some rare Sinhala characters are

cons + vow		cons + *r* + vow		cons + *y* + vow	
ක් k		කර kr	කෘ kṛ		
ක ka	කා kā	ක්‍ර kra	ක්‍රා krā	ක්‍ය kya	ක්‍යා kyā
කැ kæ	කෑ kǣ	ක්‍රැ kræ	ක්‍රෑ krǣ		
කි ki	කී kī	ක්‍රි kri	ක්‍රී krī		
කු ku	කූ kū	ක්‍රු kru	ක්‍රූ krū	ක්‍යු kyu	ක්‍යූ kyū
කෙ ke	කේ kē	ක්‍රෙ kre	ක්‍රේ krē	ක්‍යෙ kye	ක්‍යේ kyē
කො ko	කෝ kō	ක්‍රො kro	ක්‍රෝ krō	ක්‍යො kyo	ක්‍යෝ kyō
කෛ kai	කෞ kau	ක්‍රෛ krai	ක්‍රෞ krau		

Figure 3: Conjunct consonants derived from ක(=ka)

missing, these fonts are used very widely. The typical example of this kind of fonts is ”kaputadotcom”[3] font. Most of the online Sinhala sites including news sites use these kinds of fonts.

3.2. Natural Singlish

Even though it is possible to type Sinhala text using the direct input method, there is just a key for each Sinhala character (or a part of a character). For this reason this key assignment is very far apart from the intuitive sense. To resolve this problem the ”Natural SinGlish” [4] typing method was introduced by A.D.R.Sasanka. This system converts the input sequence which is more natural for the user, into the character codes as shown in Figure 4. English spellings and the English pronunciations are the base of this system. For example *shree la\nkaa* → ශ්‍රී ලංකා(=Sri Lanka). However Sinhala has much more characters than English. To avoid the ambiguousness this system has introduced several techniques, such as:

- Capitals

a	→	අ(=a)		ta	→	ට(=ṭa)	
A	→	ඇ(=æ)		Ta	→	ඨ(=ṭha)	

- Key combinations

ea	→	ඒ(=ē)		KNa	→	ඤ(=ña)	
oe	→	ඕ(=ō)		Sha	→	ෂ(=sha)	

- Dead keys: ”\” is used as a dead key

  \n → ං(=ŋ)
  \h → ඃ(=h)

This system is simply based on the English spellings, making the system quite complex. The sounds which have phonetic similarities cannot be typed in similar manner.

ka	→	ක(=ka)	and	kha	→	ඛ(=kha)	
ta	→	ට(=ṭa)	but	tha	↛	ඨ(=ṭha)	
da	→	ඩ(=da)	and	nnda	→	ඬ(=nḍa)	
ba	→	බ(=ba)	but	nnba	↛	ඹ(=mba)	

This system is not very efficient in some cases, because it uses a lot of upper case letters in the middle of the words, where the user needs to press and release the shift-key repeatedly.

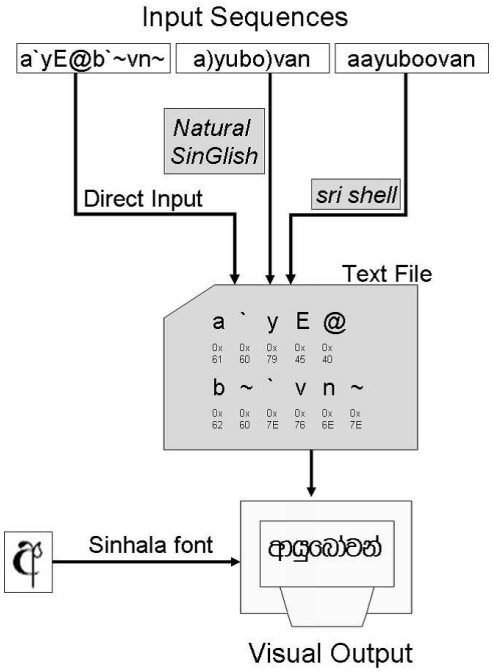

Figure 4: Sinhala Character Input Systems

4. Proposed system

We have proposed a Sinhala typing system called *sri shell*. *sri shell* assigns a key combination to each Sinhala character. The basic of this system is the phonetic notation of the Sinhala characters. Figure 5 shows the Sinhala characters, traditional phonetic notation, phonetic notation using IPA (*International Phonetic Alphabet*), and the key assignment by *sri shell*.

Unlike *Natural SinGlish*, *sri shell* has been implemented as an independent module, which allows the input of Sinhala text into any application program.

Principles of the proposed system

- It is based on phonetic notation of the characters:
  - All aspirated consonants can be produced by adding an ”h” to the unaspirated consonants.
  - Nasals can be produced by voiceless vowel proceeded by ”/”.
  - Nasal+voiced can be produced by voiced vowel proceeded by ”/”.
- It is consistent:
  - All long-vowels can be produced by doubling the last character of the short-vowel.
  - If two Sinhala characters map to the same roman character, then those Sinhala characters are differentiated by adding a ”x”. ”x” is added to the one which has less occurrence rate.
    (for example retroflex & dental, Figure 5)
- It is complete:
  Most of the Sinhala input systems introduced up

Vowels

අ	a	[a,ə]	a	ආ	ā	[aː,a]	aa
ඇ	æ	[æ]	ae	ඈ	ǣ	[æː]	aee
ඉ	i	[i]	i	ඊ	ī	[iː]	ii
උ	u	[u]	u	ඌ	ū	[uː]	uu
ඍ	ṛ	[ṛ]	rx	ඎ	ṝ	[ṛː]	rxx
ඏ	ḷ	[ḷ]	lxx	ඐ	ḹ	[ḷː]	lxxx
එ	e	[e]	e	ඒ	ē	[ɐ]	ee
ඔ	o	[o]	o	ඕ	ō	[ɑ]	oo
ඓ	ai	[ai]	ai	ඖ	au	[au]	au
ං	ṃ	[ŋ]	/n	ඃ	ḥ	[h]	hx

Consonants

	Unaspirated Voiceless	Aspirated Voiceless	Unaspirated Voiced	Aspirated Voiced	Nasal	Nasal+ Voiced
Velar	ක	ඛ	ග	ඝ	ඞ	ඟ
	ka	kha	ga	gha	ṅa	ṅga
	[ka]	[k^{h}a]	[ga]	[g^{h}a]	[ŋa]	[^{ŋ}ga]
	ka	kha	ga	gha	/ka	/ga
Palatals	ච	ඡ	ජ	ඣ	ඤ	ඦ
	ca	cha	ja	jha	ña	ñja
	[ca]	[c^{h}a]	[ɟ] a	[ɟ^{h}a]	[ɲa]	[^{ɲ}ɟa]
	ca	cha	ja	jha	/ca	/ja
Retroflex	ට	ඨ	ඩ	ඪ	ණ	ඬ
	ṭa	ṭha	ḍa	ḍha	ṇa	ṅḍa
	[ʈa]	[ʈ^{h}a]	[ɖa]	[ɖ^{h}a]	[ɳa]	[^{ɳ}ɖa]
	ta	tha	da	dha	nxa	/da
Dentals	ත	ථ	ද	ධ	න	ඳ
	ta	tha	da	dha	na	ṅda
	[ta]	[t^{h}a]	[da]	[d^{h}a]	[na]	[^{n}da]
	txa	txha	dxa	dxha	na	/dxa
Labials	ප	ඵ	බ	භ	ම	ඹ
	pa	pha	ba	bha	ma	ṁba
	[pa]	[p^{h}a]	[ba]	[b^{h}a]	[ma]	[^{m}ba]
	pa	pha	ba	bha	ma	/ba

ය	ya	[ja]	ya	ර	ra	[ra]	ra
ල	la	[la]	la	ළ	ḷa	[ɭa]	lxa
ව	va	[ʋa]	va,wa				
ස	sa	[sa]	sa	ෂ	sha	[ʃa]	sha
				ශ	ś	[ʂa]	sxa,za
හ	ha	[ha]	ha	ෆ	fa	[fa]	fa

Figure 5: Sinhala chacaters, phonetics, IPA & *sri shell*

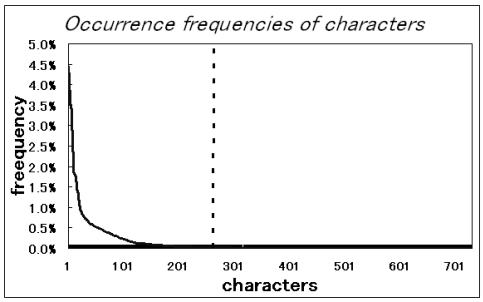

Figure 6: Occurrence frequencies of characters

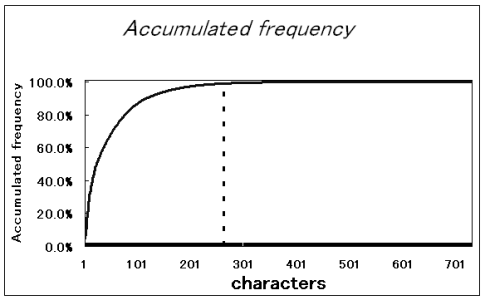

Figure 7: Accumulated frequency

to now have several missing characters, especially the rare characters such as ඐ ,ඐෘ ,ඔ ,ඔෟ are missing in most systems. The proposed system supports all the characters even though some of them cannot be displayed with most of the fonts.

5. Evaluation

To say that a specific character input method is a good one, it should fulfill two criteria. First it should be user-friendly, second it should be efficient. We believe that if it is easy to remember the input key sequences, it will lead to high accuracy and less learning time. In this section the efficiency is quantified by typing cost and the user-friendliness is quantified by the ease of remembering. The most popular Sinhala input methods have been taken into account. *kaputadotcom* (direct input), *Natural SinGlish*, and *sri shell* (see Figure 4). First of all we should know the occurrence rates of each Sinhala character, because an efficient input system must be more efficient with frequent characters. Divaina online Sinhala newspaper [5] from January, 2005 to May, 2006 (about 50MB of kaputadotcom font text) was used as a corpus to calculate the occurrence rate of each Sinhala character.

#	character		occurrence frequency	accumulated frequency
1	ය	(=ya)	4.4434%	4.4434%
2	ව	(=va)	4.1540%	8.5975%
⋮	⋮		⋮	⋮
274	ෆ්	(=f)	0.0161%	99.0472%
275	ඖෟ	(=pau)	0.0155%	99.0627%

In our evaluation 275 Sinhala characters were used, and this covers more than 99% of the characters occurred in the corpus, and all the characters have more than 0.0155% occurrence rate.

Input Method	Average edit distance
kaputadotcom	1.42
Natural SinGlish	0.35
sri shell	0.44

Table 1: Average edit distances

5.1. User-friendliness

In order to make the test more natural for the test subjects, we used a word list which includes all the 275 characters mentioned above, instead of using the characters separately. We tried to minimize the number of the words in order to reduce the test subjects' load. However the word list ended up with 106-words. The difference between the input sequences and test subjects' Romanization proposals is taken as a measure of how hard is it to remember the input sequence for each Sinhala character.

Romanizing Test Test subjects are asked to Romanize the above Sinhala word list. This test was carried out on a group of 30 subjects between 14 to 60 years old, which includes 14 males and 16 females. The Romanized word lists which we got from the subjects are split into characters. Then the difference between the input key sequence of each input method and, the proposed Romanized sequence of each test subject is measured by the edit distance between the two strings.

Edit Distance The **Levenshtein distance** or **edit distance** between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character.[6]

$$avg\_edit\_dist = \frac{1}{\#\_Subs}\sum_{subject=1}^{\#\_Subs} edit\_dist(input\_sequence(chr), proposal(subject, chr)) \quad (1)$$

$$average = \sum_{char=1}^{\#\_Chars} freq(chr) \bullet avg\_edit\_dist(chr) \quad (2)$$

Results As a measurement of the user-friendliness we have calculated the average edit distance between the input key sequence and proposed Romanization of each character. The average edit distances of each input method are calculated using Equation (1) & (2) and shown in Table 1. The results show that there is a big difference between the subjects' proposals and input sequence proposed by *kaputadotcom*. In *Natural SinGlish* and in *sri shell* the difference between the input sequences and the proposal are very small. However the *Natural SinGlish* is a little more user-friendly than *sri shell*. This happens because the test subjects always try to make the Romanization of the Sinhala word more like English. So they try to avoid key combinations such as "aa", "uu" and "ii", which are very rare in English. But *sri shell* uses these as long vowels because repeated keys are more efficient in typing. The other reason is, *Natural SinGlish* has adopted a lot of English-like input sequences, where *sri shell* emphasizes more on phonetic transcription.

5.2. Efficiency

The efficiency can be modeled using the typing speed. However, instead of the actual typing speed we used typing cost, which represents the normalized typing speed. We define the weight of average time taken to input one single key stroke as 1. The weight of shifted keys and the repeated keys may differ from 1. As a measurement of efficiency we have calculated the average typing cost of the input key sequences for each input method. We have defined the typing cost of the input sequence as Equation.3. **Exp. 1** and **Exp. 2** are carried out in order to calculate the weights of shifted keys and repeated keys.

$$typing\_cost = w_{shift} \times shifts + w_{repeat} \times repeats + 1 \times normal\_keys \quad (3)$$

$$w_{shift} = \frac{t_{xY} + t_{Xy}}{t_{xy}} - 2 \quad (4)$$

$$w_{repeat} = \frac{t_{xx}}{t_{xy}} \quad (5)$$

where,

t_{xy} = average time lapse between two alpha key strokes
t_{xx} = average time lapse to repeat an alpha key stroke
t_{xY} = average time lapse between an alpha key and a shifted alpha key
t_{Xy} = average time lapse between a shifted alpha key and an alpha key

Exp. 1 The test subjects are asked to type a set of character pairs. Some pairs consist of two different characters and in the others the two characters are the same. Then t_{xy} and t_{xx} are calculated by averaging them. This test was carried out on a group of 12 subjects (3 Female and 9 Male, Age 18-46).

Exp. 2 The test subjects are asked to type a set of common English words, but some characters of the word are capitalized. Then t_{xy}, t_{xY} and t_{Xy} are calculated by averaging them. This test was carried out on a group of 11 subjects (7 Female and 4 Male, Age 20-31).

Least Square Method The trend of the above test data is estimated using the least square method. The trend is approximated into a line (Equation 6). b and m calculated which minimize the $\sum (y - actual\_data)^2$.

$$y = mx + b \quad (6)$$

$$m = \frac{\sum (x - \overline{x})(y - \overline{y})}{\sum (x - \overline{x})} \quad (7)$$

$$b = \overline{y} - m\overline{x} \quad (8)$$

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{n\sum x^2 - (\sum x)^2}\sqrt{n\sum y^2 - (\sum y)^2}} \quad (9)$$

The test results are shown in Figure 8 and Figure 9. X-axis shows t_{xy}, the average time lapse between two alpha key strokes. Y-axes show the weights of repeated keys and the shift key.

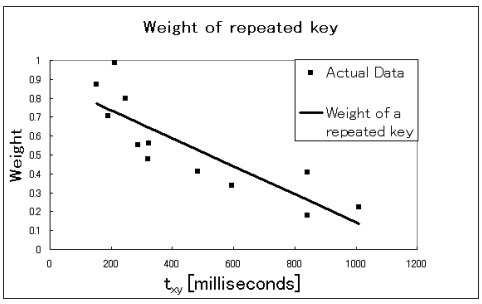

Figure 8: Weight of repeated keys

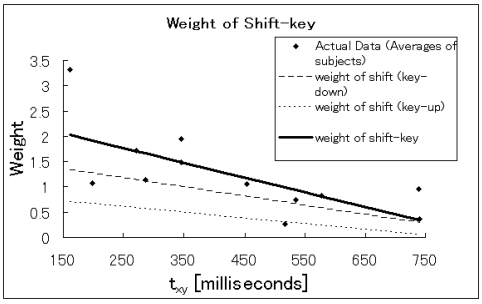

Figure 9: Weight of the shift key

The equations of the approximation lines, and the coefficient correlations are shown in Equation 10 and Equation 11.

Results The average typing cost for each input method is calculated using Equation 12, and the results are shown in Table 2. This results show that *sri shell* has the lowest typing cost among the three input methods except for t_{xy}=600ms. Even though our results show that *kaputadotcom* has the lowest typing cost for t_{xy}=600ms, *kaputadotcom* is not recommendable even to the slow typists, because *kaputadotcom* is not user-friendly. This means that *sri shell* is the most efficient input method. *sri shell* has best results because *sri shell* uses lowercase alpha characters and "/" only, where the other methods use a lot of uppercase characters and a lot of symbols (for example "*),@,#,$*"). There are a lot of demerits in using uppercase characters and symbols. This makes the users load heavier and the error rates also grow up. As our target is the average computer users in Sri Lanka, who are quite familiar with English typing, they do not feel conceptual difference on case differences. The other problem is a mixture of symbols, uppercases and lowercases, make the input sequence unreadable. (for example *kuruNA)gala* or *kOr#N$gl*) One may argue that this is just a input method and there is no necessity of readability. But if it is readable it will be easier to memorize, and when you consider about an application like latex you have to type without any output feedback, so it is better if you can read what you have typed.

$$w_{repeat} = 0.87 - 0.73t_{xy}(|r| = 85\%) \quad (10)$$

$$w_{shift} = 2.50 - 2.92t_{xy}(|r| = 69\%) \quad (11)$$

t_{xy}	*kaputadotcom*	*Natural SinGlish*	*sri shell*
200	2.60	2.26	2.18
400	2.30	2.22	2.16
600	1.99	2.16	2.13

Table 2: Average typing cost

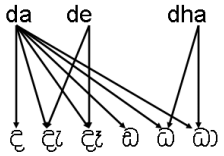

Figure 10: some many-to-many relationship in test subjects proposals

$$average = \sum_{char=1}^{\#\_Chars} freq(chr) \bullet weight(chr) \quad (12)$$

6. Conclusions and Future Work

We have proposed a Sinhala input method *sri shell* which is based on Sinhala phonetic transcription, we evaluated the user-friendliness and efficiency of the method by comparing it with other Sinhala character input methods such as *kaputadotcom* and *Natural SinGlish*.

All the Sinhala input methods proposed up to now have a one-to-one (or many-to-one) relationship between input sequence and output characters. This is the simplest way to design an input method and, these kinds of input systems require very less resources (less memory or disk space). Because of this reason, these input methods can be implemented even on mobile terminal etc. However the Romanization results by the test subjects revealed that there were certain character contexts that require many-to-many correspondences, even the subject may not recognize them. Figure 10 exemplifies such an example. In order to improve naturalness of the proposed method, we plan to incorporate these correspondences into the method, which require us to develop a context-sensitive character conversion algorithm.

References

[1] http://www.ethnologue.com /show_country.asp?name=Sri+Lanka

[2] sinhala akshara m$\bar{a}$l$\bar{a}$va by D.A.Indras$\bar{e}$na.

[3] http://www.info.lk/slword/news.htm

[4] http://www.geocities.com /naturalsinglish/

[5] http://www.divaina.com/

[6] Robert A. Wagner and Michael J. Fischer (1974). "The String-to-String Correction Problem". Journal of the ACM 21 (1): 168-173.