

Ethiopic Keyboard Mapping and Predictive Text Inputting Algorithm in a Wireless Environment

Shiferaw Abebe⁺, Tewodros Seyum⁺⁺, Solomon Atnafu⁺, Samuel Kinde Kassegne⁺⁺⁺¹

⁺ Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia

⁺⁺ Department of Information Science, Addis Ababa University, Addis Ababa, Ethiopia

⁺⁺⁺ Henri Samueli School of Engineering, University of California at Irvine, Irvine, CA 92697. External Faculty, Addis Ababa University.

Abstract- As the wireless revolution continues to make fast in-roads in telecommunication in developing countries such as Ethiopia, the availability of wireless local content and application programs in local writing systems becomes a necessity. However, a few fundamental technical barriers have to be removed before wireless content in such writing systems such as Ethiopic writing system becomes a possibility. In this paper, we address the issues of keyboard mapping and predictive text inputting - two such issues of significant challenge - for Ethiopic applications in wireless environments.

Keywords: Wireless, Ethiopic, Unicode, Predictive text inputting, ITU key-pad, Keyboard mapping, Feedel.

1. Introduction

Over the fast few decades, wireless devices have revolutionized the communication industry giving rise to unprecedented level of service penetration for telecom carriers. For developing countries such as Ethiopia, where the huge expense of laying physical fixed communication lines coupled with the cost of maintenance kept the growth of the telecom industry to the bare minimum, the advent of the wireless communication revolution is a welcome development. With encouraging signs being seen in Ethiopia in the possible expansion of the current mobile telephony service to benefit as much as a total of 350,000 users in the near future alone [1], it is increasingly becoming apparent that the demand for local wireless content is poised to grow significantly. For users comfortable in non-Latin script contents, however, a few fundamental technical challenges need to be met before wireless content is available in their script of choice. The first challenge is that the 12-key ITU standard keyboard used by the hundreds of million of wireless devices around the globe was designed to meet the demands of Latin script and its users. For non-Latin script usages, however, the issue of keyboard mapping to a native script has to be addressed by providing a mapping solution. Second, - even if the mapping issue is addressed - convenient and effortless composing of contents (such as in text messages) in the 12-keys of wireless devices will remain to be an issue demanding a solution for scripts such as Chinese, Japanese, and Ethiopic. The number of taps for entering a single character as well as composing whole words and sentences in these scripts could be impractically high thereby discouraging widespread and effortless use. In the Latin script arena and a number of Near and Far Eastern scripts, the solution for minimizing key taps has already been commercially

¹ Address correspondence to Samuel Kinde Kassegne • University of California at Irvine, Department of Mechanical and Aerospace Engineering, Henri Samueli School of Engineering, 4200 Engineering Gateway Building, Irvine, CA 92697. E-mail: samk@digitaladdis.com • Tel: (760) 402-7162.

implemented through novel predictive text inputting algorithms. However, in the case of Ethiopic, the solutions to the technical challenges of the mapping of the 12-key ITU standard keyboard and predictive text inputting have not been reported yet.

This work is, therefore, an attempt to enable a practical and efficient composing of message texts with Ethiopic on wireless devices such as wireless phones. Two aspects of this work, namely, keyboard mapping and predictive text inputting are addressed separately. The Ethiopic keyboard mapping for wireless devices is addressed first where we investigate various keyboard mapping schemes for Ethiopic script and propose the most efficient one (**ETWirelessKeyBoard**) as a model mapping scheme. Then an algorithm (**EasyET**) that predicts the word so as to decrease the number of tapping required to type the word is developed. As a bundle, - we argue - the addressing of these two technical barriers lays the ground-work for developing Ethiopic wireless applications such as Ethiopic SMS (Simple Messaging Service). This new and novel Ethiopic SMS application, we suggest, demonstrates that Ethiopic writing system has now entered the Wireless revolution.

2. Ethiopic Key Board Mapping for Wireless Devices

The Ethiopic writing system (EWS) – in the traditional non-Unicode environment - has 33 base characters with their 6 forms, which gives a total of 231 core symbols. It also has 7 special symbols, 44 labialized symbols, 8 punctuation marks and 20 numerals that raise the total number of characters to 310. In the latest Unicode version adopted (i.e., Unicode 3.0) [2], the ‘extended’ Ethiopic writing system has the range 1200-137F with 345 number of characters². In both the ‘basic’ and ‘extended’ EWS – therefore – the number of characters is considerably high making Ethiopic text composition on wireless devices a very challenging task.

For wireless applications, two significant technical barriers arise due to the excessively large number of characters in the Ethiopic writing system. The first is that the number of keys available in the typical ITU-T keyboard is limited to 12 out of which only 9 are dedicated to characters. Mapping 345 characters to just only 9 keys is, then, a formidable task. The second technical barrier is that most cell phone manufacturers have a 64KB memory limit (in some cases less) on the size of the wireless application file [3]. For example, for wireless applications written in Java, this limit suggests that size of the ‘jar’ file – including fonts - should not exceed 64 KB. Most Ethiopic font files commonly used in desk-top applications alone have sizes in the ranges of 175-250 KB [4,5]. This further puts an additional constraint on the number of characters in EWS that could efficiently be employed for a wireless application. The keyboard mapping to be employed, therefore, requires that the number of characters in the font set selected should be optimized.

For this required font optimization, we have looked at four sets of characters as candidates for elimination. These are the so-called ‘superfluous’ characters [4], replaceable characters, Ethiopic numerals and labialized characters. The determination to remove the ‘superfluous’ characters is made based on the many researches reported over the years regarding the issue of these so-called redundant (superfluous) Ethiopic characters. The works of Abraham Demoz [6], Imirru Haile Selassie *et al* [7], Hadiss Alemayehu [8], Getachew Haile [9], Yonas Admassu, Habte Mariam Markos, Yohannes Admassu and Hailu Fulas [10], and Worku Alemu [11] are some of the important works we have consulted to support this decision. In most of these cited works, it has been argued that these characters could be neglected in the

² For discussions in this paper, the term ‘basic EWS’ will refer to the traditional Feedel shown in Figure 1 whereas the term ‘extended EWS’ will refer to the version of Feedel adopted in the Unicode standards where additional sounds from Afaan Oromo, Chaha and other languages are included (cf. Figure 2).

Ethiopic writing system. The work of Worku [11] which is based on a statistical analysis of a sample of written text from the print media provides the first quantitative approach to this rather historically contentious issue. Figure 5 shows the results obtained by Worku [11] regarding the frequency of these sets of characters. While Worku’s study is based on a statistically inadequate sample, we submit that these results could serve as a reasonable starting point until comprehensive and detailed studies are available.

The replaceable characters are characters that are used infrequently and can, therefore, – for most practical purposes – be substituted by other characters. They are distinct from the ‘superfluous’ characters in a sense that the sounds they represent are indeed unique – albeit mostly in a subtle fashion. 12A7 and 1368 are some of these characters. The labialized characters which are the ones represented in Unicode as 120F, 121F, 122F, 1237, 123F, 126F, 127F, 1297, 129F, etc. could easily be replaced by typically a combination of two characters. The Ethiopic numerals represented by 1369 to 137C have commonly been substituted by Arabic numerals in desk-top applications. The same practice is adopted here.

With these optimizations, a compact keyboard mapping is then formulated. Table 1 shows such a keyboard mapping. In this mapping, the number of Feedel characters in our scheme will reduce by 40% to 210 from the original set of 345 in the extended EWS. Table 2 shows the expansion of the wireless keyboard (ITU –T) mapping.

Once the keyboard mapping is determined, the issue of character entry (or input method) has, then, to be addressed. Character entry determines the sequences of key strokes or taps to compose a character. In the following sections, we investigate two of such entry methods and then outline the cons and pros for each. Subsequently, we outline why one of them is selected as a preferred way of character entry. These two methods are the Multi-Press input method and the Three-Key input method.



Figure 1. The basic Feedel (Ethiopic base characters and their six forms)

	120	121	122	123	124	125	126	127	128	129	12A	12B		12C	12D	12E	12F	130	131	132	133	134	135	136	137
0	ሀ	ሐ	ሠ	ሰ	ቀ	ቁ	በ	ተ	ኀ	ነ	አ	ኩ	ኸ	ዐ	ዠ	ዡ	ዢ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	
1	ሁ	ሑ	ሡ	ሱ	ቁ	ቁ	ቡ	ቱ	ኀ	ኑ	አ			ዑ	ዡ	ዣ	ዤ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	
2	ሂ	ሐ	ሢ	ሲ	ቁ	ቁ	ቢ	ቲ	ኀ	ኒ	አ	ኰ	ኸ	ዒ	ዢ	ዣ	ዤ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	
3	ሃ	ሐ	ሣ	ሳ	ቁ	ቁ	ቢ	ቲ	ኀ	ና	አ	኱	ኸ	ዣ	ዣ	ዤ	ዥ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	
4	ሄ	ሐ	ሣ	ሳ	ቁ	ቁ	ቢ	ቲ	ኀ	ኔ	አ	ኲ	ኸ	ዣ	ዣ	ዤ	ዥ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	
5	ህ	ሐ	ሥ	ሰ	ቁ	ቁ	ቢ	ቲ	ኀ	ኔ	አ	ኰ	ኸ	ዣ	ዣ	ዤ	ዥ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	
6	ሆ	ሐ	ሥ	ሰ	ቁ	ቁ	ቢ	ቲ	ኀ	ኖ	አ			ዣ	ዣ	ዤ	ዥ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	
7		ሐ	ሢ	ሲ			ቢ	ቲ		ኖ	አ				ዣ	ዣ	ዤ	ዥ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።
8	ለ	መ	ረ	ሸ	ቁ	ቁ	ቢ	ቲ	ኀ	ኘ	ከ	ኸ	ዐ	ዠ	ዡ	ዢ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	፣	
9	ሉ	መ	ሩ	ሸ			ቢ	ቲ		ኘ	ከ	ኸ	ዐ	ዠ	ዡ	ዢ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	፣	
A	ሊ	ሚ	ረ	ሸ	ቁ	ቁ	ቢ	ቲ	ኀ	ኙ	ከ	ኸ	ዐ	ዠ	ዡ	ዢ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	፣	
B	ላ	ማ	ረ	ሸ	ቁ	ቁ	ቢ	ቲ	ኀ	ኚ	ከ	ኸ	ዐ	ዠ	ዡ	ዢ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	፣	
C	ሌ	ሚ	ረ	ሸ	ቁ	ቁ	ቢ	ቲ	ኀ	ኛ	ከ	ኸ	ዐ	ዠ	ዡ	ዢ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	፣	
D	ል	ም	ረ	ሸ	ቁ	ቁ	ቢ	ቲ	ኀ	ኜ	ከ	ኸ	ዐ	ዠ	ዡ	ዢ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	፣	
E	ሎ	ም	ረ	ሸ			ቢ	ቲ		ኜ	ከ	ኸ	ዐ	ዠ	ዡ	ዢ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	፣	
F	ሊ	ሚ	ረ	ሸ			ቢ	ቲ		ኞ	ከ	ኸ	ዐ	ዠ	ዡ	ዢ	ጐ	ጠ	ጸ	ፀ	ፐ	፡	።	፣	

Figure 2. Feedel (extended EWS) as adopted by Unicode 3.0 (Feb 2000).

1	2 abc አበቸ	3 def ደፈ
4 ghi ኀሀ	5 jkl ጸበለ	6 mno መ?
7 p q r s አቀረሰ	8 t u v ተሀ	9 w x y z ወ የከገ
*	0 ስኘገፀ	#

Table 1. Proposed Ethiopic keyboard mapping for wireless devices. Refer to table 2 for details.

	E	u	I	a	Y		o
H	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
L	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
M	መ	ሙ	ሚ	ማ	ሚ	ም	ሞ
S	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
R	ረ	ሩ	ሪ	ራ	ራ	ራ	ራ
Sh	ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
q	ቀ	ቁ	ቁ	ቁ	ቁ	ቁ	ቁ

b	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
t	ተ	ቱ	ቲ	ታ	ቲ	ቶ	ቶ
ch	ቸ	ቹ	ቺ	ቻ	ቼ	ች	ች
n	ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
Gn	ጎ	጑	ጒ	ጓ	ጔ	ጕ	጖
a	አ	አ	አ	አ	አ	አ	አ
k	ከ	ከ	ከ	ካ	ከ	ከ	ከ
w	ወ	ወ	ወ	ወ	ወ	ወ	ወ
z	ዘ	ዘ	ዘ	ዘ	ዘ	ዘ	ዘ
Zh	ዘ	ዘ	ዘ	ዘ	ዘ	ዘ	ዘ
y	የ	የ	የ	የ	የ	የ	የ
d	ደ	ደ	ደ	ደ	ደ	ደ	ደ
D	ደ	ደ	ደ	ደ	ደ	ደ	ደ
j	ጆ	ጆ	ጆ	ጆ	ጆ	ጆ	ጆ
g	ገ	ገ	ገ	ገ	ገ	ገ	ገ
T	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ
CH	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ
P	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ
Ts	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ
f	ፈ	ፈ	ፈ	ፈ	ፈ	ፈ	ፈ
p	ፕ	ፕ	ፕ	ፕ	ፕ	ፕ	ፕ

Table 2. This table summarizes the keyboard mapping indicated in Table 1.

2.1 Multi-Press Input Method

A Multi-Press (or multi-tapping) method is a character entry procedure where a tapping of two or more keys is needed to compose a single character. This – we propose – can be implemented in two ways. In the first approach, when one of the base characters is pressed from the keypad a menu appears that enables the user to select one of the vowels or (the different orders of the base characters). Right arrow and right arrow key can be used to browse through the seven orders and select the order of interest.

As shown in Figure 3, when one base character is pressed, say “L”, its different orders appear on the menu of the top right corner of the screen, enabling the user to select the order of his/her choice. For example, to type the name አበበ “Abebe” the user presses the key labeled ‘2’ once and to select the first order of አ “ae”, the user utilizes the right arrow to pick order of choice (which is the first order in this case and also the default; requiring no tap of the arrow). Similarly, to compose the character በ “be”, the user presses the key labeled ‘2’ with the editor picking the first order of በ by default. The second በ is composed in a similar fashion. A total of 8 key taps are, therefore, required to compose the text አበበ.

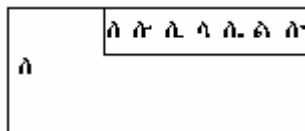


Figure 3. Orders of a character displayed on the top right most corner of the screen

It is instructive to note that Worku [11] has demonstrated that the 6th order (**sads**) is used frequently in Amharic text. The frequency of occurrence of the 6th order is greater than that of order 1 which is in turn more frequent than the 4th order. The frequency of order 4 is greater than order 2; frequency of occurrence of order 3 is greater than order 7; and frequency of occurrence of order 7 is greater than order 5. It is also indicated that the frequency of occurrence of order 2 and 3 are equal [11] (cf. Figure 4). Therefore, based on the above observations, we propose that when one of the base characters is pressed, it will be the six order (**sads**) character that will be automatically displayed. In the event that the user wants to change to the other orders, the left or right arrow could be used to select the required order. Further, the cursor will always be on the first order which is the second most used order.

In the second approach of multi-press input method, the keypad alone is used without selecting the appropriate order from the top right most corner of the screen. In this scenario the user presses the key labeled '2' once and the key labeled '3' twice to write the character for "ae". Similarly, the user again taps the key labeled '2' twice and the key labeled '3' twice to enter the character for "be". The same sets of key taps will be repeated to enter the second "be". This results in a total of 11 key taps to compose the name አበበ "Abebe". In another example, the name አሌሙ "Alemu" can be composed by tapping the key labeled '2' once and the key labeled '3' twice (to enter the character that sounds አ"ae") and then the key labeled '5' three times and the key labeled '3' twice to enter the character ለ"le" and finally the key labeled '6' once and the key labeled '8' twice to enter the character ሙ"mu". This results in a total of 11 key taps.

With regard to the so-called 'explosive' characters, Table 1 shows that a single 'sub-key' has been assigned to represent two characters (i.e., the explosive and its Geez root). For example, ዘ'ze' and ዠ'zhe' share the same key ('9') at the fourth position. The explosive characters are activated through a use of a shift (or * key).

2.1.1 Advantages and Disadvantages of Multi-Press Input Method

Advantages

- Multi-Press Input Method is simple and straight forward.
- It has adopted the de facto standard of the phonetic keyboard mapping for QWERTY keyboards which is familiar to most users of desk top applications.
- It has a choice of features for composing a text. The first option features a menu at the top right most corner of the screen that displays the different orders of the base character. The second option allows the use of the keypad alone without the need of the menu.

Disadvantages

- A user is expected to tap the keypad multiple times to enter a single character
- Loading of the seven forms of the character typed in the memory drags the system from performing efficiently.

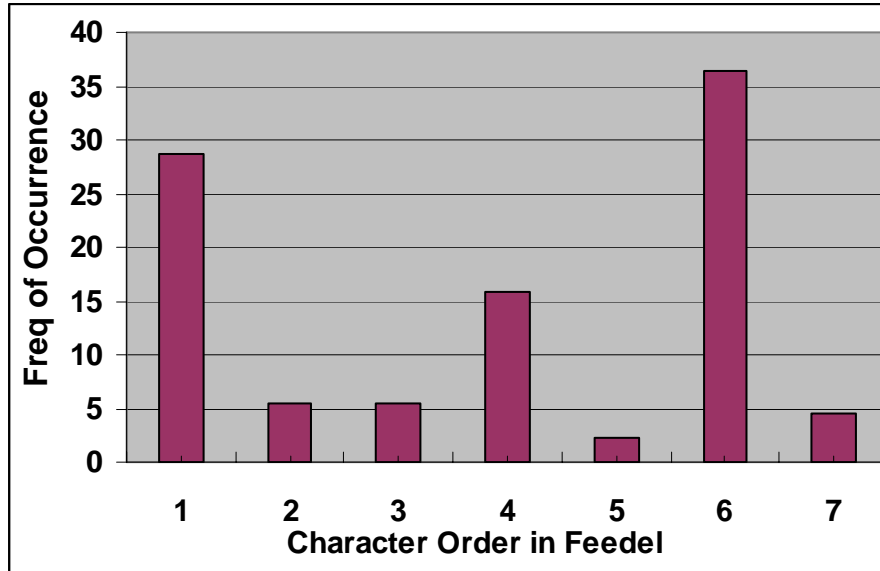


Figure 4. Worku's frequency distribution of orders of characters in Ethiopic Writing System.

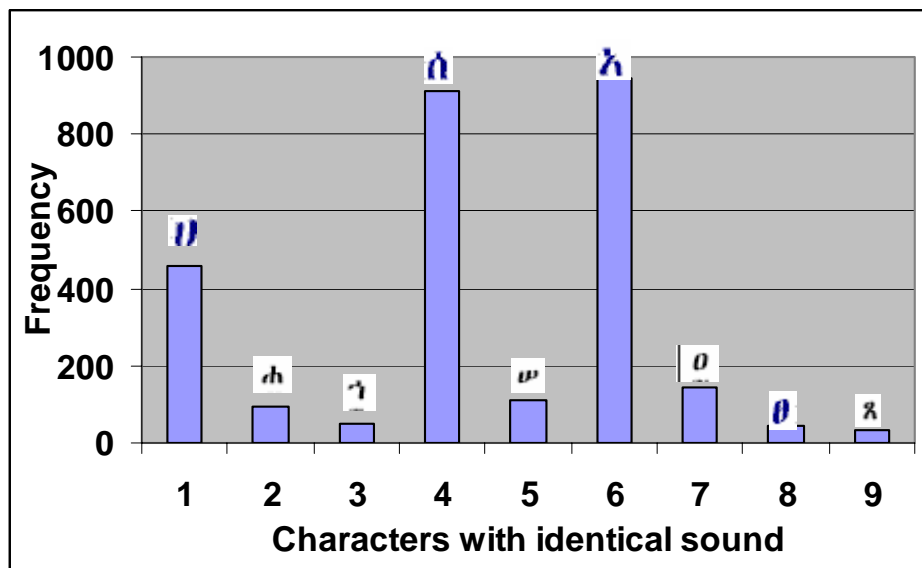


Figure 5. Worku's frequency distribution of characters with identical sounds.

2.2 Three-Key Input Method

As the name implies, the three-key input method is a character entry procedure where a tapping of exactly three keys is needed to compose a single character. The three key method is very much related to the so-called two-key input method [13]. In the two-key method, the user presses two keys successively to specify a character. The first key press, as in the multi-press method, selects the “group” of characters (e.g. key 5 for ‘JKL’). The second press is for disambiguation. One of the number keys, 1, 2, 3, or 4, is pressed to specify the position of the character within the group. For example to enter the character ‘K’, the user presses 5-2 (‘K’ is second character in ‘JKL’) [13]. The two-key method is very simple and has no timeouts. Each character A-Z is entered with exactly two key presses. SPACE is entered with a single press of the 0-key. The two-key method is not in common use for entering Roman characters,

however. In Japan, a similar method (often called the “pager” input method) is very common for entering Katakana characters [13].

Extending the concepts of the two-key input method, we have adopted the same methodology for Ethiopic text inputting using three-keys which is referred to as Three-key input method. In the Three-key input method, the first key press is used to select the group. For instance, pressing the key labeled ‘2’ selects the character group “he, le, me” [cf. Table 4]. Then pressing the key ‘3’ corresponds to selecting the third character from the selected group (in this case, “me” is the third member in the group). Then the user presses one of the keys labeled ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’ or ‘7’, to select the order of the character selected in the previous key press. This is applicable to all the root characters shown in Table 2. In the case of three-key input method the keyboard design follows the natural sequence of the Ethiopic characters as shown in Table 4.

2.2.1 Advantages and Disadvantages of Three-Key Input Method

Advantages

- In general, the Three-Key Input Method decreases the number of tapped required to compose a text compared to Multi-Press Input Method (see Table 5).
- It is easy for the user to learn.

Disadvantages

- The user needs to train on how to use this keyboard layout to compose a text.

2.2 Selection of Input Method

Based on the arguments presented above, we submit that the Multi-Press input method is more suitable and easier for Ethiopic applications in a wireless environment. Further, with the implementation of ‘dictionary’ or predictive text inputting algorithm discussed below, the Multi-Press input method proves to be easier to use and program.

1	2 ሀሰመ	3 ሰረዥ
4 ቀበተ	5 ቸጎኘ	6 ለበገ
7 ወዛገ	8 የደደደ	9 ገጠጫ
*	0 ጸፀፈገ	#

Table 4. Keyboard design for Ethiopic that follows alphabetical arrangement.

3. Predictive Text Inputting in Wireless Environment

Predictive text inputting (PTI) is the process of adoption of an intelligent algorithm that predicts words and, in some cases, phrases and sentences while the user is typing on a the keypad of a wireless device. In a typical PTI application, as the user types, the sequence of the characters is compared with possible letter combinations in the built-in dictionary. Whenever the algorithm

predicts a similarity, the user is prompted with the most likely word or - in some cases - phrase. In general, PTI application saves the user from tapping far too many keys.

The origin of PTI algorithms could be traced to the mid 1990s when new applications started to appear on handsets that tried to predict the word being typed based on a single tap of a key for each letter. Using these predictive applications reduced the number of keystrokes required to compose a message on wireless phones. Some of the commonly used proprietary algorithms in most wireless phones for PTI are T9 Text Input, I-Tap, eZiText, and tx4u.

T9 Text Input, which is marketed by Tegic and is quite widely available in many of the mobile phones shipped around the world, makes text entry more efficient by allowing users to compose text messages with just one key press letter [12]. Each key on a telephone keypad has more than one letter. A single press on the “5” key could be “J”, “K”, or “H”. T9 uses a dictionary as the basis for disambiguation. The method is based on the same key layout as the multi-press method, but each key is pressed only once. For example, when the user wants to enter “*is*”, s/he pushes the “4” key first which consists of “G”, “H”, and “I”, and then taps the “7” key which consists of “P”, “Q”, “R”, and “S”. Using the combination of “4” and “7” corresponds to various two-character combinations including “*hr*”, “*gs*”, etc., but the word “*is*” appears most frequently in English texts, and the algorithm predicts that the word “*is*” is the intended word in this case. T9 compares the word possibilities to its linguistic database to “guess” the intended word.

Naturally, linguistic disambiguation is not perfect, since multiple words may have the same key sequence. In this case, T9 gives the most common word as a default. To select an alternate word, the user presses a special SHIFT function key. For example, the key sequence 6-6 gives “on” by default. If another word was intended, the user presses “*” to view the next possible word. In this case, “no” appears. If there are more alternatives, SHIFT (*) is pressed repeatedly until the intended word appears. Pressing 0 accepts the word and inserts a SPACE character [13]. Tegic’s patented T9 Text Input software currently supports more than 14 languages.

I-Tap is marketed by Motorola’s Lexical division and is bundled in most Motorola cell phones. eZiText is another proprietary PTI product and was developed by Zi Corporation. The capability of text prediction of the three products namely T9 Text Input, I-Tap and eZiText, vary, but broadly speaking they do the same thing: predicting the current word the user is composing on a wireless phone. However, Motorola has recently announced that a new version of their I-Tap software which will predict whole sentences based on past writing [14].

More rigorous PTI products have also been introduced recently. A company called AirTx proposed a solution “that reads the users mind” through its product called tx4u [15]. AirTx anticipates an algorithm that not only predicts next words, but also next action. The algorithm accomplishes this by developing a personal language model to anticipate what the user wants to compose as he/she begins typing the message and then predicts the end of the sentence based on the users’ previous use of language. In some circumstances, a claim has been made that tx4u will even predict an entire sentence [14,15].

3.1 Predictive Text Inputting for Ethiopic - EasyET Algorithm

In the context of the Ethiopic writing system, there is a legitimate need to formulate and adopt intelligent text predicting algorithms. Text input prediction in Ethiopic script will depend on two sets of factors. The first factor is the structure of Ethiopic languages (such as Amharic) and statistical distribution of the order of characters. There is limited statistical study in this area that could guide the development of the algorithm. As a starting point, however, the work of Worku [11] in statistical distribution of order of characters in EWS (with particular emphasis on Amharic) will again be used to guide this process. The second factor is that of the word, phrase

and sentence usage habits of the individual user. Naturally, this is difficult to predict apriori and a 'dictionary' approach should be used to build a library of words, phrases and sentences commonly composed by the user.

Based on these observations and results reported in the literature [11], we propose the following algorithm called **EasyET** for predictive text inputting in Ethiopic writing system – with particular emphasis and application in the Amharic language.

Step 1: Determine which of the letters in each box is used more frequently at the beginning of Amharic words. Worku's work [11] is used as guidance here. Based on the distribution scheme (Figure 6), the algorithm will pick that character as the first one when the user taps the corresponding key. For example, in the '7' key that corresponds to the letters 'PQRS', the 'S' character (Cf. Table 1) occurs more often than the other three (refer to Worku's work and Figure 6). According to Worku's work on model sample texts, 'S' occurs 457 times, 'R' occurs 250 whereas 'Q' for "Qe" occurs 156 times. Therefore the algorithm will pick the sixth order of the 'Se' family (i.e. S) as the first character of the predicted word. Again, the reason why the 'sads' (6th) order is selected is due to the fact that it is the most frequent order (cf. Figure 4). This algorithm by itself could easily save a number of multi-taps in a given text message.

Step 2: The second level of prediction will involve picking the right order in the root character. For example, if the intended word is 'sukar', the algorithm in Step 1 had managed to pick only the sixth order. The second order that corresponds to "su" will be picked by tapping key '8' which contains the vowel 'u'. For words that have not been completed yet, the algorithm determines that the operation of picking the order of the character is in scope and forces tapping of the key '8' to pick only the vowel 'u' from the set that contains 'T U V'.

Step 3. If the word is a one-character size, the user enters a space character to separate the current word from the next word. Otherwise the user enters the second character by repeating the first two steps.

Step 4. After the first two or more characters have been entered by the user, the predictive algorithm will pop up – from its dictionary - the first n top rated candidate words that begin with the entered characters.

Step 5. The user selects one of the candidate words from the pop up list. The user can browse through the word list using asterisk "*" key. If the intended word is not in the list, the user will simply continue to tap in the next character of the word. In this case, the predictive algorithm will come up with a better list of candidate words using step 4. This process continues until the predictive algorithm correctly predicts the intended word or until the user finishes typing in the intended word. Once the word is entered either predicted by the algorithm or typed in by the user, the user will enter a space character to indicate the end of the word.

Step 6. If the predictive algorithm is not capable of predicting the intended word due to absence of the entered word in the dictionary, the algorithm will request the user to save the new word. Similarly, the user is prompted to save all new words, phrases and sentences to the dictionary for a later use. These will then serve as basis for prediction in subsequent text compositions.

Step 7. If the user opts to save the word the new word will be added to the dictionary and rated.

Table 5 summarizes the amount of tapping efforts required to compose some common Amharic phrases using non-predictive and predictive methods. This example is intended for

demonstration purposes and can not be taken as an exhaustive statistical comparison between the three approaches. Nevertheless, its demonstration of the efficiency of the predictive algorithm is useful. The number of taps for Three-key input method is made by adding three key presses for each character. The estimation for the number of taps required when the predictive algorithm is used is based on the assumption that some of the words have already been 'learnt' by the dictionary. Here, we have assumed that before a certain word is predicted, the user will enter, on average, half of its characters and the remaining will be entered by selecting the predicted word.

Text Phrase	Number of taps with different input methods			
	QWERTY	<i>Multi press</i>	<i>Three-key</i>	Predictive
Sibseba lay negn ሰብሰባ ላይ ነገን	12	32	24	16
Meleshe idewlalehu ሙሉሼ አደው ላለው	17	30	27	15
Melkam Inqutatash ሙሉአዘም አንቁጣጣሽ	19	29	30	15
Inquan des yalachihu አንቁጣጣሽ ደስ ያለኛው	17	41	33	21

Table 5. Comparison of number of taps required for composing a text message for some common Amharic phrases.

3.2 Discussion

The EasyET Algorithm will work well on a database that has fairly common Amharic words that can be used for messaging purpose. As improvement of the suggested algorithm, the actual selection of these words can be made by taking SMS messages from the SMS server itself. Algorithms that are used in information retrieval for identifying index terms and removing stop words will be applied to get the required terms that are commonly used for messaging purpose. First the stop words will be removed, then the frequency of every word will be calculated and assigned to the word. Finally a threshold value will be set and words beyond the threshold value will be included in the word database (dictionary). Once the user has started to use the messaging service (SMS) the algorithm will add new words to the dictionary. For example, when the user types a new word, if the word is not in the dictionary, the algorithm asks the user whether to save the new word in the database or not. Based on the user's response, the word can be added to the dictionary. Words in the dictionary will have a corresponding rating attribute. This default rating attribute will be given to the words by categorizing the frequency range in to three groups. For example most commonly used words will be given a rating of 2, the next common words will be given a rating of 1 and ordinary words will be given a rating of 0. This default rating will help the algorithm to predict words for a new user. However this default rating is subject to change based on the user's word usage habit. Whenever a user uses a certain word from the dictionary or a new word is added to the dictionary, the word will be rated. For new words, the rating will begin from 1 whereas for existing words the rating increases by one on the previous rating value. This gives the word a better chance to be predicted in

successive entries. The other benefit is it gives the chance for the predictive algorithm to learn the user's word usage habit.

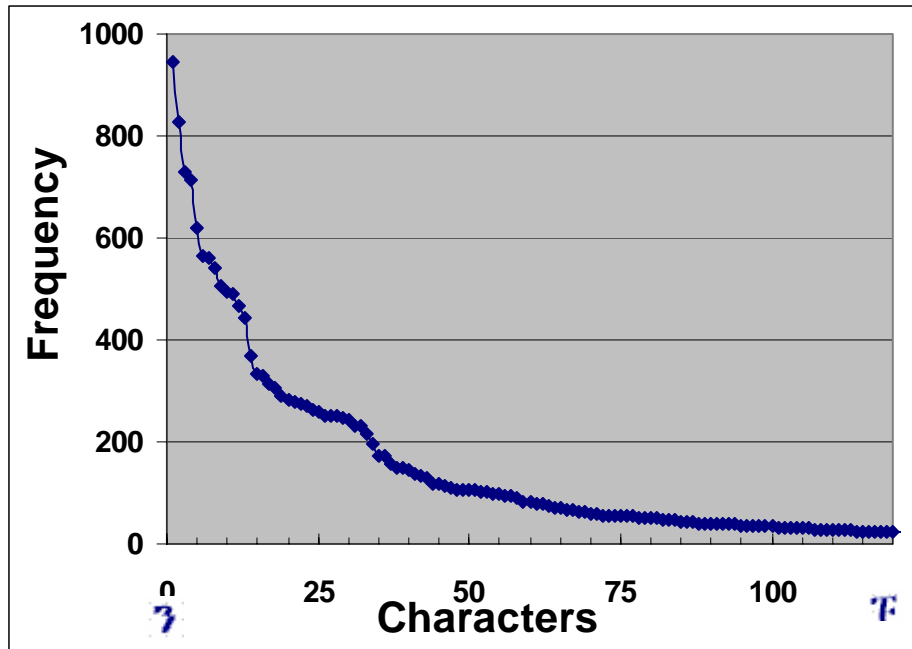


Figure 6. Worku's frequency distribution of Ethiopic characters.

4. Application Integration

The addressing of the Ethiopic keyboard mapping and predictive text inputting lays the necessary groundwork for developing Ethiopic wireless applications. The most obvious of such applications is text messaging in Ethiopic script. Text messaging or SMS (Simple Messaging Service) – as it is technically known - is a popular and high-volume wireless application that has experienced explosive growth around the globe.

Short Message Service (SMS) is the transmission of short text messages to and from a mobile phone, fax machine and/or IP address. Messages must not be longer than 160 alphanumeric characters when Latin alphabets are used and 70 characters when non-Latin alphabets such as Arabic and Chinese are used [16]. The SMS messages cannot contain images or graphics. SMS messages are supported by GSM, TDMA and CDMA based mobile phone networks currently in use. With an increasing wireless mobile phone service user base in Ethiopia, it is becoming clear that such an application will find a wide use.

By integrating these two solutions discussed in this work and other innovative approaches for some more technical barriers peculiar to wireless devices and the protocols that govern their operations, we have developed a wireless application for Ethiopic text messaging. Figure 6 shows a screen shot of such a text messaging application in Ethiopic writing system.



Figure 6. Integration of Ethiopic keyboard mapping and predictive text inputting enables the development of text messaging in Ethiopic.

Conclusions

In this paper we have attempted to discuss the two different types of keyboard mappings for wireless devices. Namely the multi press input method and the three-keys input method. We also have indicated the pros and cons of each input method. In the developed solution the multi press input method option has been implemented by considering its advantage over the three-key input method. In the Predictive Text Inputting part, we have proposed a new algorithm, EasyET that will predict the current word while the user is typing a word. This algorithm learns the user's word usage habit and predicts accordingly. We believe that the integration of these two solutions, demonstrates that Ethiopic writing system has now entered the Wireless revolution.

References

1. Ethiopian Telecommunication Corporation, http://www.telecom.net.et/publications/eNews/enews_july22_04.html.
2. The Unicode Standard, Version 4.0.1, defined by: *The Unicode Standard, Version 4.0* (Reading, MA, Addison-Wesley, 2003. ISBN 0-321-18578-1), as amended by *Unicode 4.0.1*.
3. Device Profile for Java™ 2 Micro Edition, Version 2.0, JSR 118 Expert Group, jsr-118-comments@jcp.org
4. Geez Frontier Foundation, GF Zemen Ethiopic Unicode font.
5. Ethio Systems, Jiret Ethiopic Unicode font.
6. Demoz, Abraham, "Amharic Script Reform Efforts", *Ethiopian Studies: Dedicated to Wolf Leslau on the Occasion of his 75th Birthday November 14th 1981*, Otto Harrassowitz, 1983
7. Immiru Haile Selassie, Abebe Reta, and Blatta Merse Hazen Wolde Kirkos, Ye-timihrt Wedajoch, "*Feedel'n Mashashal*", 1948.
8. Haddis Alemayehu, *Fikir iske meqabir*, Addis Ababa, Ethiopia, 1966.
9. Getachew Haile (1966-67), "*The Problem of the Amharic Writing System*", A paper Prepared in advance for the interdisciplinary seminar of the Faculty of Arts and Education, Haile Selassie Ist University.
10. Yonas Admassu, Habte Mariam Markos, Yohannes Admassu and Hailu Fulas (1974), "*Amarigna le-Kolej Dereja Yetezegaje*", Department of Ethiopian Languages and Literature, Haile Selassie Ist University.
11. Worku Alemu (1997), "*The Application of OCR Technique to the Amharic Script*", Addis Ababa University
12. Tegic Communications Extend Instant Messaging to Wireless Phones with Next Generation of T9 Text Input Software, Press Release from Tegic Communications, July 27, 1999
13. Silfverberg, M., & et al. (2000), "*Predicting Text Entry Speed on Mobile Phones*", Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2000, pp. 9-16. New York: ACM
14. Hall, Justin, Technology:[FEATURE], "*Adaptive Disambiguation*", December 2003 <http://www.thefeature.com/topic?topicid=33>.
15. <http://www.airtx.com>. Accessed on September 10, 2004.
16. Le Bodic, Gwenaël, '*Mobile Messaging Technologies and Services SMS, EMS and MMS*', Alcatel, France, Wiley & Sons, November 2002