Three Kinds of Text Input Methods for Languages that Use Indic Scripts: A Comparison

Kiran Pala, Anil Kumar Singh and Suryakanth V Ganagshetty

IIIT, Hyderabad, India. KIIT University, Bhubaneswar, India. IIIT, Hyderabad, India.
kiranap@gmail.com, svg@iiit.ac.in and aiklavya@gmail.com

Abstract content

1. Introduction

Many of the major Indian languages such as Hindi, Bengali, Telugu, Tamil use scripts which are derived from the ancient Brahmi script. They are also known as Indic scripts. Over the last few years, processing and displaying text in languages using these scripts has become easier and more standardized, something that was lacking earlier. However, text input in most of these languages is still problematic and in an unstable state (Shanbhag et al., 2002; Sowmya and Varma, 2009; Ahmed et al., 2011). The main problem is that there is no universally accepted keyboard layout for most of these languages.

To overcome this problem, the first question that needs to be decided is which text input method, using a normal PC keyboard (as different from handheld devices), is the best for Indian languages using Indic scripts. Since efforts for creating standards for localization of computing technology have intensified in recent years[1], this question needs urgent attention.

There might be different ways of categorizing the possible input methods. We will present here one broad three-way categorization. All three kinds are used for input of text in Indic languages at present even on personal computers and laptops. We discuss each of these categories and then we will present a perceptual/performance evaluation of three specific input methods, one from each of the three categories. We also note how such evaluation has to be interpreted, keeping in mind the category to which an input method belongs.

2. Indic Languages and Writing Systems

The scripts used for Indic languages are called 'complex scripts' in software development terminology. Linguistically, they are often called alpha-syllabaries or syllabaries, but the term most suited for them is Abugidas, i.e., scripts where a combinations of consonant (C) and vowel (V) has a single symbol. This is related to the linguistic fact that words in Indic language either earlier had or still have a vowel at the end and the default vowel was schwa. However, the Indic writing systems are more complicated than what this characteristic implies. Written words are formed by a process of composition of symbols, not all of which are of the type CV. There may be symbols only for C, only for V (in fact, different symbols for V depending upon whether it is a part of the previous syllable or not) and even symbols for complex combinations such as CCV etc. Even though the number of complex combinations having their own symbols is not very large, the process of composition of such symbols is one of the major characteristics that makes these scripts relatively complex, which in turn makes the design of a good input method a difficult task, especially when the same Latin based keyboard has to be used for these scripts.

A very important element in the composition of text for these languages is something that can be called schwa deletion. A symbol called 'halant' is used to remove the inherent vowel from the core consonant symbols and to then form complex combinations. This is one of the major things to be learned while learning to type Indic text using some of the input methods. For the other input methods, this process is not explicit and therefore reduces the load to some extent.

3. Three Kinds of Input Methods for Indic Languages

The various input methods used for typing Indic text can be categorized into the following three broad categories:

- **Script Based**: These methods are designed specifically and independently for a given script (or a set of similar scripts) and are not in any way based on the Latin based scripts. An example from this category is the Inscript input method for Indic languages, which is increasingly being adopted as the standard input method for Indic languages. However its use is not very prevalent so far because few people have learned it.

- **Latin Derived**: These are designed as mappings from Latin characters to Indic characters and therefore they can be used both as input methods as well as (standard or non-standard) encodings for storing and displaying the text. An example from this category is the WX input method, which was designed by some prominent Indian NLP researchers and it is used more commonly among the community of NLP researchers.

- **Transliteration Based**: In this case, the user types Latin characters as if he was (phonetically) typing the text in English, but it gets converted to text in Indic scripts on the screen through an intermediate step of transliteration. The RTS (Rice Transliteration

[1] `http://www.w3cindia.in`

Scheme) as well as Google Transliteration are examples from this category.

For our evaluation, we consider Inscript, WX and RTS. All three of these have advantages and disadvantages, which make them suitable in different scenarios. For example, Inscript has a higher learning curve because its design is directly based on the characteristics of the Indic scripts. Inscript is also not 'phonetic', in the sense that what you type is not a phonetic approximation (in terms of Latic characters) of what you intended to type, e.g. there is no such thing as the letter (and the key) 'k' being mapped to /k/ and the letter for /k/ in Indic scripts. However, once the user has learnt it, the performance load might become less because this method is based on the phonetic and orthographic characteristics of Indic scripts. Even the learning curve can be significantly reduced if the user is given a little training or is able to understand the principles on which its design is based. For example (roughly speaking), the left side of the keyboard is used for vowels, the right one for consonants. Similarly, the Shift key is used for getting the aspirated forms of consonants or the syllabic forms of vowels (as opposed to modifier or diacritic forms, which can be typed without Shift). The middle row has the unvoiced consonants, while the upper row has the voiced forms. For vowels, the upper row has the 'longer' forms of vowels on the middle row. The lower row is used for frequently used non-core consonants, vowel modifiers and punctuation marks. This leaves some special but less frequently used symbols which are elsewhere on the keyboard.

Intuitively, it seems that WX should have a learning curve comparable to Inscript and it has the additional advantage of being used as an encoding or transliteration scheme for storing and displaying Indic data using basic Latin characters. WX is 'phonetic' because the mapping from Latin characters to Indic characters tries to maximize phonetic correspondence, e.g. 'k' is for /k/, 'K' is for /k^h/, 'a' is for schwa etc. The most notable exceptions are 'w' (used for /t/) and 'x' (used for /d/), both pronounced as in French, thus giving the scheme its name. The learning curve for WX is expected to be high (though perhaps lower than that for Inscript) because you need to frequently press the Shift key to type roughly half of the characters. Roughly speaking, capital letters are used for aspirated consonants and 'long' vowels like /a:/ ('A'), /i:/ ('I') and /u:/ ('U').

RTS is the most directly 'phonetic' because here the text input process happens through phonetic approximation using Latin characters. For example, you type 'aa' or 'A' for /a:/ and 'b' for /b/. Since this is not really an input method in the conventional programming terms, there is some flexibility in typing the intended text, i.e., you can type the same thing in more than one ways, which, when combined with its direct phonetic nature, makes the learning curve the shortest. It is almost like typing in English. However, RTS has rare, if any, support on computers and operating systems. It can only be used as an additional plugin or a tool to type Indic text.

4. Experiments and Results

In our short experiment, we had 12 participants (8 male and 4 female, in the average age group of 22-26) who each had

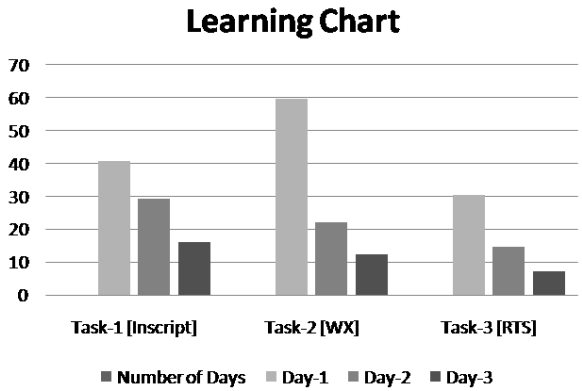

Figure 1: Learning Chart

to type two Telugu (different) sentences each for three consecutive days for all three input methods. Thus, each participant had to type two sentences each using three input methods on three days. The sentences were manually selected to be representative in terms of characters and complex combinations. All of the participants had good familiarity with computers and the Internet (the evaluation interface was web based). Out of the 12, only two had previous exposure to any of these input methods.

The evaluation had an objective (quantitative) aspect as well as a subjective (perceptual) aspect. This was a controlled experiment in the sense that we provided some basic training to the participants in a short trial period where they were given a document to type in the three input methods. The purpose was to ensure that all of them had some (and almost equal) exposure to all the three input methods. This was done because the input methods belong to three different categories and it is unrealistic to expect someone to suddenly start typing in Inscript or WX. Without this initial exposure, a phonetic transliteration scheme like RTS would have had an unfair advantage.

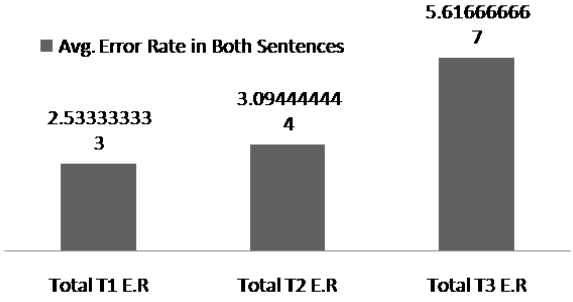

Figure 2: Average Errors

The data that we recorded from the experiment included the time taken in typing each sentence and the errors made in each sentence. From this data, as shown in a learning chart (Figure 1), we can see that the time taken was reduced over the period of three days. There was a decline in all cases, but it was highest in the case of WX, which means, contrary to expectations, WX was actually found to be harder

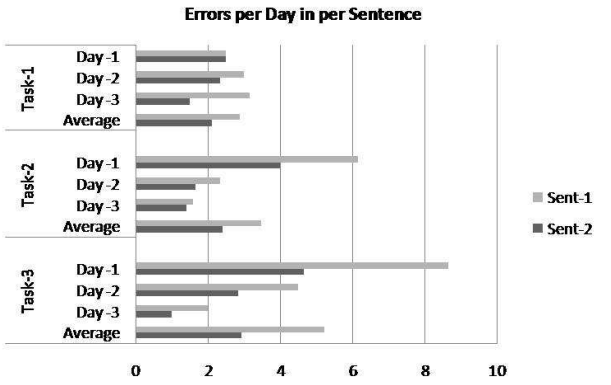

Figure 3: Daywise Errors

to learn initially in our case where the user had already been exposed to all three input methods. This seems to imply that either Inscript is not hard to learn or that it becomes easy to learn once a little training is provided. In Figure 2, we show the average errors over the three days. The number of errors were the least (as expected) for Inscript and were the highest for RTS, which reflects the nature of these kinds of input methods, with Inscript being a proper input method whereas RTS being a flexible transliteration scheme. In Figure 3, we show the errors made on each of the days. This figure suggests that Inscript is very stable as far as error rate is concerned and it also has the lowest error rate. The converse in true is for RTS and WX is somewhere in between.

Apart from this objective evaluation, we also asked the participants about their experience and the responses that we received supported the quantitative data represented by the figures here. They also made many comments that can be used to build better training modules to allow users to quickly learn to type using the concerned input method. The same can be said about the kind of errors that they made.

5. Conclusion and Implication

While the results of this short experiment are far from conclusive, they do seem to verify our comments about the learning curves and ease of use of the three kinds of input methods for Indic languages. Specifically, the experiments contradict a supposition put forward by those who are reluctant to adopt Inscript as the standard input method for Indic languages. Since there is widespread support for it on various platforms, we argue that there is no convincing case about the impracticality of adopting Inscript. And because Inscript seems to have the lowest error rate, it is also important to adopt it from the point of view of language resource quality. But since our experiment was only on a small amount of data, it will have to be confirmed by further experiments on larger data.

6. References

Umair Z. Ahmed, Kalika Bali, Monojit Choudhury, and Sowmya V. B. 2011. Challenges in designing input method editors for indian languages: The role of word-origin and context. In *Proceedings of the Workshop on Advances in Text Input Methods, IJCNLP*. AFNLP.

Shrinath Shanbhag, Durgesh Rao, and R. K. Joshi. 2002. An intelligent multi-layered input scheme for phonetic scripts. In *Proceedings of the 2nd international symposium on Smart graphics*. ACM.

V. B. Sowmya and Vasudeva Varma. 2009. Transliteration based text input methods for telugu. In *Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages*. Springer-Verlag.