Indian Language Transliterator

-IMT2017047, IMT2017042, IMT2017045

This document explains about the working and conceptual details of the Indian Language Transliterator. There are two parts in this application. The first part renders a keyboard friendly (ASCII) input of the local language into the actual native script (in this case Kannada). The second part takes this converted string in UTF-8 format, and renders it in the indianized roman version of it. To achieve this, we have created two transliteration schemes (maps in C++), where the first one maps between ASCII and native language alphabets (in UTF-8 format) and the second one maps between the native language alphabets (in UTF-8 format) and indianized roman letters (in UTF-8).

Algorithm:

Rendering from ASCII to Native Language (Kannada):

As explained above, we have a C++ map between combinations of ASCII letters, which makes the respective sense in the native language, and the UTF-8 code of the native language. The maximum number of characters of all the keys of the map is four. The algorithm, at each iteration, starts with a window of length four characters, checking the obtained sub-string in the map and continues if found else checking for the substring of length of one less till it reaches a window of length one character, while doing the same procedure each time. If it still does not find this character then it will break, throwing an error message.

Rendering from Native Language (UTF-8 String) to Indianized Roman Version:

A similar approach is followed for this as well, but with a window length of twelve and six (because the length of string representation of a native alphabet in UTF-8 is 6, implicitly taking 2 or 1 length window in the native language). If it still does not find this character then it will break, throwing an error message.

Procedure to Execute this Program:

1.) `g++  ilt.cpp -o ilt`
2.) `./ilt`
3.) Stdin (in ascii) (eg. amma or akka)
4.) This creates an “output.txt” file where the strings are rendered in Native language(Kannada) and Indianised Roman.