Facilitating simpler programming for N T P by devising newer codes

N.D.Logasundara Muthali
Chennai - India

Digital Tamil has come a long way since 20+ years from initial mere display features to practical useful operator-friendly apps. For a popular hand held devises of 2014 Today we have advanced levels of workhorse tools in Text processing inlaid with Spell checkers & morphological analyzers etc created by Ganesan and Deyvasundaram

Steep growth in Social media brought strong stimulants to Internet users. Being one of its associate member as INFITT got synchronized with Unicode philosophy for future developments performing its best part. Propelled by indisputable universal thumb rule as 'continuous improvement' is must for advancement in any walks of life and in this niche area as NTP, ways and means from basics were thought of and identified the following as simpler solutions. New codes with a focus on analytical engines that lie in higher semantic layers as translation etc and we know well that Programs, Application Tools are always a deal with Digital processing of continuous stream of Binary Dumps generated and handled in and through every machine and devises, every moment.

Multifaceted nature in contents created from world with wide open freedom need complex programs to resolve them in NLP. By providing new codes for pinpointed specific feature & keying in by content authors, can made to preempt them of complex programs and algorithms to considerable extent.

(1) Identification of Language = Language (font) marking code
This is an indisputable necessary in analytical engines that operate in semantic level as N T P, voice2text text2voice Machine Translation and A.I. The claim as with Unicode it is very simple to find the languagae from few strings of codes, it is not really so when we delve deep inside users universe it can be commented as a myth.

There are 20+languages share Devanagari range. Nothing less with Latin origins. Now there are two letter codes (not as Unicode codes) for programming and for users as EN for English TA for Tamil etc. But they cannot be used straightaway inside in a single stroke by a command of an author into content, like tag at start any file as text or image or else. and make the doc features more flexible and connect to suitable display font which will be more apt than a indirectly deduced one by complex and voluminous software.

There shall be many instances in which more than one language can present in contents which are sharing the same range or instances of contents that toggle between different languages than he major content by quote/unquote.

By keying in indicators for selection of more suitable font that can be identified away from fonts of unsupported ranges and more so if the content warrant special user-defined fonts which can lie in Private use area. Few custom packaged font range can be created for actual and popular use and unto the taste or mental comfort of the user for display / print thro' his / her popular special fonts, out of a list, that can be by option, instead of a blanket long range, memory gulping font that being imposed by OS installed as default by the vendors.
Initially this new proposed code range can cover major languages that are in CLDR maintained by Unicode and also from threshold of their dynamism if not for all those identified by Ethnologue. Contents using PUA also may need to call its own earmarked font for special scripts if any.

In some case the content may be in transliterated format with different script than its native script. Here identifying the real language is a necessary before handling by many analytical engines that lie in higher semantic planes. Here the language markers are indispensable and auto identifiers are of hindrance than a helping hand.

(2) **Text & Verse marking codes.**
Identification of contents in TEXT or VERSE modes are in need for parsing inside NTP. In the semantic layer as there are differences bound to come into play in the Contents whether they are in which mode as text or verse. It has to be clarified by an indicative operators for their functional semantic differences to analytical engines. Provision of dedicated codes to indicate start and end of verse modes from an in situ operatives will ease out tailor made algorithmic solutions that identify content stream is in verse or text mode.

(3) **Fine tuned Verse mode needs in Tamil = Codes for verse components**
Not only in the case of Tamil all other Indic language along with many international languages use verse mode in contents (with different kinds of prosodies and connected with their own well defined canonical grammar)

In a Tamil verse
(1) Start & end of a verse's line, in any meter ** >> ** அமிக்கி புதுமு முழுமு
(2) Identify first and second letter of a verse line, for மூக்கு / மூக்கு / கா / கா
(see below) ஆி / ஆி / ஆி / ஆி
(3) Blank space that identify the 'acai' break # = உண்த உட
(4) Identification of Special hyphen inside a verse ( - ) மன்னித்திதிர / சைத்திதிர / மன்னிதித
(5) Long tab like white spaces to indicate non breaking extra long verse line from usual line break in the verse, that is used in formatting that accommodate width of the print media

All are required to be identified for understanding a verse for NLP easily without complex algorithms

Examples for 1 ,2 , 3 & 5
**அமிக்கி புதுமு முழுமு # திராஸ்பர்னே # ம்லாலிய பார்த்தே # அமிக்கி புதுமு முழுமு

Examples for 1 ,2 , 3 & 5
**அமிக்கி புதுமு முழுமு # திராஸ்பர்னே # ம்லாலிய பார்த்தே # அமிக்கி புதுமு முழுமு

Examples for 4
ஏமைல மைல ஏமைல மைல
With above indicators a verse, its meter, type and components can be identified accurately

(4) Acronym or text mode = Acronym marker code
There are millions of acronyms are in use inside contents By simple code special semantic operators can trigger applications as Voice2text, Text2Voice, translation etc to identify the Digital streams nature as acronym or a standard word of Lexicon. Shortened words as Mr. Dr. Prof. etc shall fall in this slot if they are not in Lexicon in ref.

(5) Texts in Head lines mode = Head lines marker code
Contents in Headlines are mostly written with short circuited sentences sans full syntax by dropping some canonical grammar components using fewer discrete words. Indicative markers can help in identifying and translate when full meaning of the words as a normal sentence with full syntax Translation tool must identify them correctly

these words are to be translated, If in need, for conveying the contents with completed syntax for full explanation as

(6) Texts found as phrases, idioms and 'old saying' modes
= A special marker for all three flavors
Every major language have these richness and mark them with special codes will help in identifying the boundary unto which the special core line semantics lies and help in translate in one stroke to a equivalent in another language

In these three pieces of examples the core sense conveyed by these group of words can be presented in another language with different words that need not be a one to one equivalent but the sense can be maintained presented in different ambiance suitable to that culture maintaining underlying basics with minimum distortion and ambiguity.

In higher layers of coding schemes it is possible to provide codes for each these (in all the three modes) core theme sense/items and and showing equivalents of different languages under same codes so that translation can be easy
Part 2 Proposal for new letters to Tamil Range
Proposal as new codes into existing Tamil range in BMP as they are not provided in it now. These will ease out complex programs NLP Voice2text Text2voice Translation etc in differentiating their canonical features.

(a) aLapaTaikaL >>> uyiraLpaTaikaL >>>>>>>> 5 codes
Now the same letter a, i, u, e, o are written to indicate very long voicing of uyirmey with a, i, u, e, o combine with the preceding long vowel modifiers as of A, I, U, E, O for 3 units of voicing time (to the respective uyirmey letters)

(b) Indicator for oRRaLapadakaL /Aytha aLapaTaikaL >>>>> 1 (One code)
In these letters their voicing time of doubled consonant or Aytham is increased from half unit to full one

(c) Common Modifiers to indicate kuRRiyaikaram kuRRiyalikaram >> 1
Though the practice of denoting these with a dot above is not in use today, but they are deemed very much reside latent and unavoidable in prosody of verses. In old Tamil texting the same dot is used as that of a consonant. Devising a new kind of dot will differentiate these from normal consonant dot above a letter in the proposed code

(d) Common modifier for kuRukkangkaL as aikArak kuRukkam aukArak kuRukkam >>>>>>>> 1 code
aukArak kuRukkam which will come only when the letter au is the first letter of the word
Though in the present verse scenario these are not taken as serious features but these are very much in need to teach them as old grammatical feature once existed. New suitable glyphs will be devised different from others avoiding confusables

(e) AythakuRukkam >>>>>>>> 1 code
The aytham inside the verse its voicing time is reduced from half unit to a quarter unit
Though at present this is not taken as serious features, but very much in need in pedagogy of old grammatical feature once existed.
(f) makarak kurukkam >>>> 1 code

Though in the present scenario these are not taken as serous features but these are very much in need to teach them as old grammatical feature once existed. As per the nURpA in tolkAppiyam

From this nURpA it is found that there can be a second puLLi in use. As the makara mey already have one puLLi on it the second one to denote further shortening has to be denoted by second pulli inserted inside the makara glyph

(g) letters of ancient Tamil Musical notes >>>> 7 codes

sa, ri, ka, ma, pa, ta, ni, or other equivalents These Tamil letters indicating 7 paNNcinai kuRikaL (of music) They must be present as marks i.e. different from normal Tamil letters in the text ૃ, ૅ, ે, ૊, ્ ૒ Newer improved sylphs will be devised with consideration to find older marks if any existed in stone inscriptions etc,

Conclusion

All these will be proposed to utilize from the 50 "reserved" slots exist in the Tamil Range as 0B80 through 0BFF