

Algorithm to Correct Missing *Pulli*-Signs in Printed Tamil Text

Author: Muthiah Annamalai*

Corresponding email: ezhillang@gmail.com

Abstract:

A common problem in digitizing Tamil texts is the missing consonant sign *pulli* (unicode u0BCD) which may be lost in the process of OCR, or absent by convention during printing of older texts. We propose a bigram statistics based combinatorial algorithm to correct such errors in text.

1. Introduction

A common problem in digitizing Tamil texts [1] is the missing consonant sign *pulli* (unicode u0BCD [5]) which may be lost in the process of OCR, or absent by convention during re-printing of older texts; in one case the reprints contain newer errors than in original library copies. We propose a bigram statistics based combinatorial algorithm to correct such errors in text, adding to the classes of algorithms for Tamil text correction presented previously [4].

2. Methodology

A given Tamil word is split into Tamil letters and we consider to correct only the missing *pulli* or consonant sign. Since *pulli* can go with only the consonants, we find each consonant occurring in the word can be true consonant or a false consonant yielding at least one correct alternate in the 2 raised to power of number of consonants in the word.

For example, a misprinted word கண்ணன் (correct form is only 'கண்ணன்'), has 4 consonants [க, ண, ன, ன]. Therefore we may expect upto 16 variants of the word with and without the pulli, at least one of which will be the correct answer.

['கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்', 'கண்ணன்']

The algorithm for generating the combinatorial alternates is shown in code [pulligal_helper](#) in Appendix A. The complexity of this algorithm is $O(2^{\text{consonants}})$.

Algorithm:

Input: Tamil word **Wr** with potentially one or more missing pulli(s)

Output: Best alternative word **Wc**

1. Let **C** be list of all the consonants in order of occurrence in **Wr**
2. Generate set **S** of all words with consonants **C** occurring with and without pulli

3. Since there is only the binary choice - i.e. each consonant can occur with or without pulli we have $2^{|C|}$ alternatives
 - a. These alternatives can be simply generated by looking at bit pattern of enumeration of numbers from (0, .. $2^{|C|}-1$).
 - b. For each of the bit pattern we can associate 1 with presence and 0 with absence of pulli sign.
4. Sort the alternatives by their bigram probabilities; e.g. bigram probability of occurrence of a n-letter word in a language can be written as, with base bigram probabilities, $P(w_i|w_{i-1})$, picked off a standard reference - e.g. Tamil VU Dictionar or Project Madurai [2a,2b,4].
 - a. $P(W_n) = \prod_{i=2}^n P(w_i|w_{i-1})$
5. A reasonable alternative has highest probability of occurrence and can be marked as output word.
6. Optionally we can combine step 5 by filter all words **S** for a prefix occurring in a dictionary **D**.

Once we generate this list of alternates we can filter them with a dictionary or a unigram or bigram scores. The bigram score is computed using the Tamil Virtual University dictionary based corpus statistics in Open-Tamil library[2a,2b]. Bigram or unigram probabilities for the word are computed in standard manner using the sum of logarithm of individual bigram probabilities. The computational complexity of the algorithm is exponential in number of consonants occurring in the word but usually this is of order of 1-10 consonants even in a large word.

3. Results

The Sangam poetry lines by classical Tamil poet Chembulapeyalneerar [3] when misprinted (without the *pulli*) appears as follows:

Incorrect	யாயும ஞாயும யாராகியரோ எந்தையும் நுந்தையும் எம்முறைக கேளிர
Correct	யாயும் ஞாயும் யாராகியரோ எந்தையும் நுந்தையும் எம்முறைக் கேளிர்

Misprinted Word	Alternates	Bigram Scores	Correct Word
யாயும	‘யாயும்’ ‘யாயும’	20 16	யாயும்
ஞாயும	‘ஞாயும்’ ‘ஞாயும’	20 16	ஞாயும்
யாராகியரோ	‘யாராகிய்ரோ’ ‘யாராகியரோ’	36 31.94	யாராகியரோ
எந்தையும்	‘எந்தையும்’	32	எந்தையும்

	‘எந்தையும்’ ‘எந்தையும்’ ‘எந்தையும்’	28.83 28 24.83	
நுந்தையும்	‘நுந்தையும்’ ‘நுந்தையும்’ ‘நுந்தையும்’ ‘நுந்தையும்’	36 32.839 23.0 28.839	நுந்தையும்
எம்முறைக்	‘எம்முறைக்’ ‘எம்முறைக்’ ‘எம்முறைக்’ ‘எம்முறைக்’	32.18 28.54 28.18 24.54	எம்முறைக்
கேளிர	‘கேளிர’ ‘கேளிர’	20 16	கேளிர

In total we see just a bigram based scoring of the 7 misprinted words give correct choices for 6 words of 7 in error, for an **85.7%** accuracy score, showing the promise of our approach. Further improvements in the accuracy can be made by using a dictionary based approaches - for example in sub-routine [pulligal_branch_bound](#) in the **Appendix A**.

4. Conclusion

We present a simple algorithm for correcting OCR *induced* missing consonant sign by combinatoric generation and filtering by bigram scoring. This approach is useful for advanced spelling correction on lines of our work in [4].

5. References

1. W. H. Arden, “A Progressive Grammar of Common Tamil,” Soc. for Promoting Christian Knowledge (1910).
2. (a) Tamil Virtual Academy Dictionary <http://www.tamilvu.org/library/dicIndex.htm> (accessed Jun, 2020).
(b) Solthiruthi module of Open-Tamil v0.96 <https://pypi.org/project/Open-Tamil/> (accessed Jun, 2020).
3. Wikipedia entry on Sangam Poet Chembulapeyalneerar, <https://ta.wikipedia.org/s/468> (accessed Jun, 2020).
4. M. Annamalai, T. Shrinivasan, “Algorithms for certain classes of Tamil Spelling correction,” INFITT Tamil Internet Conference, Chennai (2019).
5. JD Allen, et-al, “The Unicode Standard 5.0,” Addison-Wesley Professional (2012).

Appendix - A

இந்த நிரல் துண்டு MIT உரிமத்தில் வெளியிடப்பட்டது

import tamil

import operator

from solthiruthi.scoring import bigram_scores, unigram_score

def mean(x): **return** sum(x)/float(len(x))

def pulligal_helper(prefix,letters):

if len(letters) == 0: **return** [prefix]

 letter = letters[0]

 result = []

if letter **in** tamil.utf8.agaram_letters:

 result1 = pulligal_helper(prefix + letter, letters[1:])

 mei_letter = letter + tamil.utf8.pulli_symbols[0]

 result2 = pulligal_helper(prefix + mei_letter, letters[1:])

 result.extend(result1)

 result.extend(result2)

else:

 result1 = pulligal_helper(prefix + letter, letters[1:])

 result.extend(result1)

return result

def pulligal_branch_bound(prefix,letters,அகராதி):

""" we restrict options if its not a prefix in dictionary """

if len(letters) == 0: **return** [prefix]

 letter = letters[0]

 result = []

 prefer = அகராதி.starts_with(prefix)

if letter **in** tamil.utf8.agaram_letters:

 alternate2 = prefix + mei_letter

if அகராதி.starts_with(alternate2) **or** prefer:

 mei_letter = letter + tamil.utf8.pulli_symbols[0]

 result2 = pulligal_branch_bound(alternate2, letters[1:])

 result.extend(result2)

 alternate1 = prefix + letter

if அகராதி.starts_with(alternate1) **or** prefer:

 result1 = pulligal_branch_bound(alternate1, letters[1:])

 result.extend(result1)

return result

#sort in descending order

```
chol = tamil.utf8.get_letters(input("Enter Word>>>"))
result_tpl = ["".join(sol),(-1.0*bigram_score(sol))] for sol in pulligal_helper("",chol)]
result_tpl = sorted(result_tpl,key=operator.itemgetter(1),reverse=True)
print(result_tpl)
```