International Journal of Computing Science and Information Technology, 2018,
ISSN: 2278-9669, April 2018(http://ijcsit.org)

A Survey On The Language Transliteration From English To Kannada Using Application

Megha. H.R [1].
S.E.A College of Science, Commerce & Arts,
Bangalore, India;

Bharathi.A [2].
S.E.A College of Science, Commerce & Arts,
Bangalore, India,

Received January 2018

Abstract

It is challenging to translate names and technical terms across languages with different alphabets and sound inventories. These items are commonly transliterated, i.e., replaced with approximate phonetic equivalents. In this paper, we represent the construction of language transliteration Application. we also implement the translation from English to Kannada language using two methods WEKA and SVM in general Language transliteration Application.

Keywords: Language Transliteration Application, WEKA method, SVM method

I. Introduction

Language transliteration is one of important area in Natural Language Processing (NLP). Language Transliteration Application will do the conversion of a character or word from one language to another without losing its phonological characteristics. In other word we can say machine transliteration is an Orthographical and phonetic converting process. Therefore, both grapheme and phoneme information should be considered. The transliteration model must be designed in such a way that the phonetic structure of words should be preserved as closely as possible.

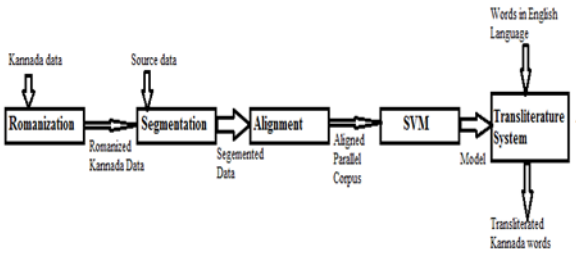
Language Transliteration Application are BARAHA, NUDI, QUILLPAD, GOOGLE TRANSLITERATURE. Etc..

Some Concepts

Natural Language: A system of communication among humans with sound.
Script: A system of symbols for representing language in writing.
Phoneme: Basic unit of sound in a language that is meaningful.
Grapheme: basic distinct unit of a script.

II. LANGUAGE TRANSLITERATION MODEL CREATION

In the proposed work, the English to Kannada transliteration problem was modelled as classification problem using two different approaches. The first transliteration model was based on a rule based approach using WEKA"s C4.5 Decision tree classifier with features extracted from a parallel corpus. The second model was based on statistical approach using SVM. The model was trained with the same aligned parallel corpus which consists of 40,000 words containing names of various places in India.



1. Romanization

WEKA"s C4.5 Decision tree classifier and SVM support only Roman (ASCII) character code but Dravidian language like Kannada does not support this code format and support only Unicode character. Unicode or officially called the Unicode Worldwide Character Standard is an entirely new idea in setting up binary codes for text or script characters. Unicode is an industry standard whose goal is to provide the means by which text of all forms and languages can be encoded for use by computers. So in order to map training and testing target data from Unicode to Roman and vice versa, mapping files were created. Using the mapping rules that defines Eng-

lish alphabet for each Kannada alphabet, Romanizes all the Kannada words.

English names	Kannada	Romanized Kannada
Megha	□□□	mEgha
Bharathi	□□□□□	BArati
Bombay	□□□□□	bAMbe

2. Segmentation

An important phase in machine transliteration process is segmentation and alignment. Efficiency of the transliteration model mainly depends on segmentation of source language and target language words into transliteration units (n-grams) and aligning the source language n-grams with corresponding target language n-grams. So before training the transliteration model, the transliteration units are obtained by segmenting the source and the target language words. The rules for segmentation have been derived to suit phonetic reproduction of English names into Kannada. The English Names are segmented based on vowels, consonants, digraphs and trigraphs into English transliteration units. The segments or units can be synonymously called as English n-grams.

Vowels : a, e, i, o, u

Consonants : b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z

Digraphs: bh, ch, dh, gh, kh, ph, rh, sh, th, wh, zh, ng, nj

Trigraphs: ksh

When more than one vowel occur together, they are combined like aa, ae, ai, ao, au, ia, ie, io, etc., to form a single unit.Similarly Romanized Kannada names are segmented based on vowels, consonants, digraphs and trigraphs into Kannada transliteration units. The segments or units can besynonymously called as Kannada n-grams.

English	Romanized Kannada
M e g h a	m E g h a
B h a r a t h i	B A r a t i
Bombay	bAMbe

3. Alignment

Alignment is the final step in the preprocessing phase. Alignment is a most important phase in the transliteration process in which the one to one mapping between English language n-grams and the Kannada language n-grams is performed. Proper alignment of source language n-grams with phonetically equivalent target language n-grams is required to generate an efficient transliteration model. Alignment is based on the number of transliteration units in the segmented English and Romanized Kannada place names. The corresponding transliteration units in English and Romanized Kannada words are aligned if the number of units in the corresponding English and Romanized Kannada words are equal. Otherwise inserting an empty symbol „^" or combining the adjacent units in the Romanized Kannada words, the units in the source place name are properly align to the unit in the target place name. Examples below shows, how the alignments of source and target words take place under different situation. Where „S" and „T" denotes source and target language words respectively.

Case 1: When the number of units are same:

Before alignment	After alignment
M e g h a (S)	M | e | g | h | a
(5 units)	(5 units)
m | E | g | h |a (T)	m | E | g | h |a
(5 units)	(5 units)

Case 2: Alignment of words by combining adjacent units:

Before alignment	After alignment
B h a r a t h i (S)	Bh | a | r | a | th | i
(8 units)	(6 units)
B A r a t i (T)	B | A | r | a | t | i
(6 units)	(6 units)

Case 3: Alignment of words by inserting empty symbol:

Before alignment	After alignment
b o m b a y (S)	b | o | m | b | a | y
(6 units)	(6 units)
b A M b e (T)	b | A | M | b | e | ^
(5units)	(6 units)

4. Support Vector Machine (SVM) Tool

Convert these aligned source and target names in a column format based on the sequence labelling approach and SVM training input data format. The token is expected to be the first column of the line. The tag to predict takes the second column in the output. The column separator is the blank space. A sequence of tokens forms a word and each word is marked with boundary as shown below. The features required for training are defined with a window size of 5 elements and the core being the third position.

M m
e E
g g
h h
a a

IV. MAPPING ANALYSIS

From the results of segmentation and alignment, it is noted that an English n-gram can be mapped into one or more Kannada n-grams. A dictionary consisting of all English n-grams and their corresponding mapping Kannada n-grams (Class labels) was created from the training corpus. The frequency of an each English n-gram, i.e., number of occurrences of an English n-gram in the training corpus, along with the corresponding mapping label frequency is also maintained in the dictionary. The dictionary is referred during the training and the prediction process of transliteration.

Conclusion and Future Work:

In these survey paper, we have shown the construction of Language Transliteration Application (LTA) using the two methods such as WEKA and SVM. And we have discussed the working of LTA by transliterating English to Kannada.

In our future work we are going to develop the LTA using some Programming Language. And also we will implement the language transliteration of all Indian Language.

REFERENCES:

[1] Karimi, Sarvnaz, Falk Scholer, and Andrew Turpin. "Machine transliteration survey." ACM Computing Surveys. 2011.

[2] Kumar Sourabh. “An Extensive Literature Review on CLIR and MT activities in India”. International Journal of Scientific & Engineering Research, Feb 2013.

[3] Pushpak Bhattacharyya. “Machine Translation”, CRC Press. 2015.

[4] Philipp Koehn, Franz Josef Och, Daniel Marcu. “Statistical Phrase-Based Translation”. 2003.

[5] Balajapally, P., Bandaru, P., Ganapathiraju, M., Balakrishnan, N., & Reddy, R. Multilingual Book Reader: “Transliteration, Word-to-Word Translation and Full-text Translation”. 2006.

[6] http://www.google.com/