

# English to Malayalam Transliteration Using Sequence Labeling Approach

Sumaja Sasidharan, Loganathan R, and Soman K P  
Amrita Vishwa Vidyapeetham/CEN, Coimbatore, India  
sumaja.sasi@gmail.com  
{kp\_soman, r\_logu}@ettimadai.amrita.edu

**Abstract-** Transliteration is the mapping of a word or text written in one writing system into another writing system. Transliteration maps the letters of the source language to the letters in the target language for a specific pair of source and target language. Transliteration must preserve sound. Transliteration can be used for encryption also. Here the source language is English and the target language is Malayalam. In some cases the letters in the source script may not match exactly with the target language. Transliteration usually defines some conventions for dealing with that. The source string is segmented in to transliteration units and related with the target language units. Thus transliteration problem can be viewed as a sequence labeling problem. Here the classification is done using Support Vector Machine (SVM).

**Index Terms-**Transliteration, Sequence Labeling Approach, Support Vector Machine

## I. INTRODUCTION

Transliteration performs a mapping from one alphabet into another. Transliterations can be used to write words in some old scripts with good precision. For example, traditional or cheap typesetting with a small character set; editions of old texts in scripts not used any more; some library catalogues. The transliteration process is quite close to phonetic mapping of Indian language characters to the letters of the Roman alphabet; hence it should preserve phonetic structure of words. Transliteration can be used in situations where we want to express words or concepts in a language with another script.

## II. THE SEQUENCE LABELING APPROACH

Transliteration maps the letters from the source script to the letters of the goal script. The process of transliteration mainly involves two steps:

- Segmentation of the source string into transliteration units.
- Mapping the source language transliteration into the target language.

Thus the transliteration problem can be viewed as a sequence labeling problem [1] from one language alphabet to another.

Here the source language is English and target language is Malayalam. An English name, for example, X is segmented in to  $x_1, x_2, \dots, x_n$  where  $x_i$  corresponds to the alphabet in the name. Let the equivalent Malayalam name be Y and Y is segmented as  $y_1, y_2, \dots, y_n$  where each  $y_i$  is treated as a label in the label sequence. Each  $x_i$  is now aligned with its phonetically equivalent  $y_i$ .

$$\begin{array}{ccccccc} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & & y_n \end{array}$$

To generate an efficient model the phonetically equivalent segments should be properly aligned. The valid target language alphabet ( $y_i$ ) for a source language alphabet ( $x_i$ ) in the given source language input word depends on the, alphabet ( $x_i$ ), alphabets ( $x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}$ ) surrounding source language alphabet ( $x_i$ ), alphabets ( $y_{i-2}, y_{i-1}, y_{i+1}, y_{i+2}$ ) surrounding target language alphabet ( $y_i$ ). These features are used to train the model using support vector machine. This transliteration model then used to predict a target language word for new source language word.

## III. SUPPORT VECTOR MACHINE

SVMs [2] [3] [4] belong to the class of supervised learning algorithms in which the learning machine is given a set of examples (or inputs) with the associated labels (or output values). The goal of supervised learning is to identify an optimal mapping from some input variables to some output variables, which is solely based on a sample of observations of the values of the variables. SVMs construct a hyperplane that separates two classes and tries to achieve maximum separation between the classes. Separating the classes with a large margin minimizes a bound on the expected generalizations error.

Let  $\{x_1 \dots x_n\}$  be our data set and let  $y_i \in \{1, -1\}$  be the class label of  $x_i$ . The decision boundary should classify all points correctly.

$$c_i (w^T x_i - \gamma) \geq 1; \quad 1 \leq i \leq n \quad (1)$$

The decision boundary can be found by solving the following constrained optimization problem

$$\text{Minimize } \frac{1}{2} \|w\|^2$$

$$\text{Subject to } c_i (w^T x_i - \gamma) \geq 1; \quad 1 \leq i \leq n$$

Some applications allow a certain degree of misclassifications where a certain amount of error can be tolerated. In such cases SVM with Soft Margin is used by where slack variables are introduced. This slack variable measures the degree of misclassification of each data.

$$c_i (w^T x_i - \gamma) \geq 1 - \xi_i; \quad 1 \leq i \leq n \quad (2)$$

The objective function becomes,

$$\text{Minimize } \frac{1}{2} \|w\|^2 + C \sum \xi_i$$

$$\text{subject to } c_i (w^T x_i - \gamma) \geq 1 - \xi_i; \quad 1 \leq i \leq n$$

In situations where the data is not linearly separable, one prefer non-linear mapping of data in a high dimensional space. This new space is called feature space where it is linearly separable. We use a non-linear kernel function. The parameters are obtained by solving the following non linear SVM formulation (in Matrix form),

$$\text{Minimize } L_D(u) = \frac{1}{2} u^T Q u - e^T u$$

$$d^T u = 0; \quad 0 \leq u \leq Ce$$

where  $Q = DKD$  and  $K$  - the Kernel Matrix. The kernel function  $K(AA^T)$  (polynomial or Gaussian) is used to construct hyperplane in the feature space, which separates two classes linearly, by performing computations in the input space.

SVM is basically a binary classifier. We can use SVM to classify multiclass data where there are  $K$  classes. The interpretations will differ from classical binary SVMs.

- class labels are vectors instead of +1s and -1s in the binary SVM. Thus class labels in binary SVM belong to one-dimensional subspace where as in the case of multi-class SVM, the class label belongs to multi-dimensional sub-space.
- the separating hyperplane in binary SVM is a vector. In multiclass, it is a matrix.

In multi-class SVM, the natural extension of this function is then the mapping of data/feature space into vector label space whose defining bases are vectors. In other words, multi-class learning may be viewed as vector-labeled learning or vector value- learning.

#### IV. TRANSLITERATION USING SVM

Transliteration problem can be viewed as a multiclass classification problem. Training is done for every class to distinguish the examples of each class from the rest. The most probable class labels are selected. For each alphabet in the source script a dictionary is created from the training samples.

The transliteration process consists of three phases:

- Preprocessing phase
- Training phase
- Transliteration phase

##### A. Preprocessing phase

In this phase the source language names are romanized, segmented and aligned with the corresponding segmented target language words. It is then converted in to the format required by the SVMTool.

*Romanization:* During romanization all the English words are converted in to lowercase and converted into the corresponding Malayalam words. These Malayalam words are then Romanized using the mapping rules that defines English alphabet for each Malayalam alphabet.

The following table shows the Romanized output

TABLE I  
ROMANIZED MALAYALAM NAMES

| English | Malayalam | Romanized Malayalam |
|---------|-----------|---------------------|
| sahdev  | സഹദേവ്    | sahdEv              |
| dheeraj | ധീരജ്     | dhEraj              |
| sadik   | സാദിക്    | sAdik               |
| ekachit | ഏകചിത്ത്  | Ekacit              |

Long vowels are represented using uppercase letters.

*Segmentation:* After Romanization the English names and Romanized Malayalam names are segmented. Segmentation is done based on vowels, consonants, digraphs like th, dh, sh and trigraphs like ksh, ngg.

The following table shows the segmented output

TABLE II  
SEGMENTED ENGLISH AND ROMANIZED MALYALAM NAMES

| English       | Romanized Malayalam |
|---------------|---------------------|
| s a h d e v   | s a h d E v         |
| dh ee r a j   | dh E r a j          |
| s a d i k     | s A d i k           |
| e k a c h i t | E k a c i t         |

*Alignment:* After segmentation the English names and the corresponding Malayalam names are aligned. If the number of units in both English and Malayalam names are equal they are properly aligned. If the number of units, for a particular name, is different in English and Romanized Malayalam a mismatch will occur.

Consider the example,

s a h d e v ( 6 units )  
s a h d E v ( 6units )

In the above example the number of units is the same and they can be properly aligned.

A mismatch can be resolved by introducing an empty symbol or combining the adjacent units.

s a d y o j a t a (9 units)  
s a d y o j a a t a (10 units)

The above mismatch can be resolved by combining the two adjacent symbols.

s | a | d | y | o | j | a | t | a (9 units)  
s | a | d | y y | o | j | a a | t | a (9 units)

Consider another example,

s a m a t h (6 units)  
s a m a t ^ (5 units)

One alphabet is less in the Romanized Malayalam name. So here introduce an empty symbol ^.

s | a | m | a | t | h (6 units)  
s | a | m | a | t | ^ (6 units)

The labels are the target language n-grams. After alignment the names are converted into SVM format

*B. Training Phase*

After converting in to SVM format the source language names and target language names are

given to the SVMTool for training. The source language names are the input sequence and the target language names are the label sequence.

*C. Transliteration Phase*

The SVMTool is a simple and effective generator of sequential taggers based on Support Vector Machines. SVMTool uses SVMlight for training. SVM learning uses linear kernel. The learning time remains linear with respect to the number of examples. The trained model is used for transliterating English words into Malayalam words. The model is evaluated using the same SVMTool.

V. RESULTS AND CONCLUSIONS

The model produced the Malayalam transliteration of English words with an accuracy of 90%. The corpus is includes 20,000 names for training and 1,000 names for testing.

REFERENCES

- [1] M.S. Vijaya, R. Loganathan, G.Shivapratap, V.P. Ajith, and K.P. Soman, "English to Tamil Transliteration Using Sequence Labeling Approach." International Conference on Asian Language Processing (IALP),Chiang Mai ,Thailand, November 12-14 2008.
- [2] V.N. Vapnik, "Statistical Learning Theory,." J.Wiley & Sons, Inc., New York, 1998.
- [3] O.L.Mangasarian, and R. David Musicant, "Lagrangian Support Vector Machines", Journal of Machine Learning Research, Vol 1, March 2001, pp 161-177.
- [4] O.Mangasarian, "Data Mining with support vector machines." Data mining institute report, University of Wisconsin, Madison, May 2001.