Hybrid Approach for Transliteration of Odia Named Entity to English

Suprava Das, Rakesh Ch. Balabantaray

Abstract— Transliteration of NAMED ENTITIES plays an important role for cross language information retrieval processes. This paper shows the design of a hybrid (rule based + fuzzy based) transliteration system for named entities of person, location, organizations written in Odia script to English (Roman Script). For this purpose, we have also created a database of specialized spelling i.e. location names, organization names...etc. which helped for performance improvement with accuracy 87%.

Index Terms— Transliteration, Odia named entity, hybrid approach

I. INTRODUCTION

Now a day, the web is considered as universal resource of information which is growing very rapidly and contains information in many languages. The users may pose their query written in one language but retrieve the relevant document written in another language. For this purpose several machine translation system and bilingual dictionaries are also frequently used to convert the text of one language to another language. But translation cannot translate some of the text, because some text may not have corresponds to translation word in bilingual dictionary. Transliteration overcomes the problem of out of vocabulary words (OOV) problems (Fujii & Ishikawa, 2001; Lin & Chan, 2002).

Transliteration system converts a string written in one script so that phonetic property of the string remains unchanged (Arbabi et al. 1994). Transliteration should not be confused with translation, which involves a change in language while preserving meaning. For example:

ଆମ୍ବ → Translation → WE

ଆମ୍ବ →Transliteration → AME

The remainder of the section is organized as follows. Section 2 describes the previous work on transliteration, Odia and Roman script is discussed in section 3. In section 4, I have described the implementation of our system. Evaluation and result is discussed in section 5 followed by conclusion in section 6.

II. LITERATURE SURVEY

Building transliteration system for Indian languages to Roman language is a significant research challenge.

Many researchers have used different approaches for transliteration that involves either designing a model of direct mapping between two orthographies or designing the phonetic representation for transforming strings into each other or combination of both[Oh et al. 2006].

Asif et al [1] have developed Bengali to English transliteration system and used supervised training set to obtain direct orthographic mapping. Vijaya, VP, Shivpratap and KP CEN [2] have developed a Tamil transliteration system named as WEKA. This rule based system has been tested with 1000 English names with an accuracy of 84.82%. Malik [3] has developed Shahmukhi to Gurumukhi transliteration system named as Punjab transliteration system (PMT) which is rule based. Haque et al [4] have developed English to Hindi phrase based statistical transliteration system. Monika Bhargava, M.Kumar, Sujoy Das [5] has developed Hindi to English rule based transliteration system. Lehal and saini [6] have developed a Hindi to Urdu transliteration system with an accuracy of 99.46%. In this paper, we first used a set of handcrafted rules by adding or dropping proper phonemes to normalize Odia syllables into consonant vowel format. Then we used spell check algorithm followed by cross lingual phonetic based soundex algorithm.

III. OVERVIEW OF ODIA AND ENGLISH SCRIPTS

All Odia is written in Kalinga script, one of the many descendants of Brahmi script of ancient India. Odia language uses 50 symbols for representing 10 vowels, 36 consonants and 4 modifiers. The vowels are transcribed in two forms i.e. independent and dependant form. Dependent form is also known as matraa. Former is used when vowel letter appears alone at the beginning of word or is immediately followed by another vowel. Latter is used when vowel followed consonant. English Language is written in Roman script. English is a West Germanic language that arose in the Anglo-Saxon kingdoms of England. There are 26letters in English. Out of which 21 are consonants and 5 are Vowels.

Mapping from Odia to English

There is no one to one correspondence from Odia to English script. Tables 1 and 2 show the mapping between source language [Odia] to target language [English] and Table 3 shows mapping between target language to source language. We have used these mappings to transliterate proper names in Odia to English language.

TABLE -1: Mapping of consonant from Odia to English

Sl. No.	Odia	English	Sl. No.	Odia	English
1	କ	ka	19	ଝ	Dha
2	ଖ	kha	20	ଞ	Nia
3	ଗ	ga	21	ଟ	Ta
4	ଘ	gha	22	ଠ	Tha
5	ଙ	na	23	ଡ	Da
6	ଚ	cha	24	ଢ	Dha
7	ଛ	cha	25	ଣ	Ma
8	ଜ	ja	26	ତ	Ja
9	ଝ	jha	27	ଥ	Ra
10	ଞ	nja	28	ଦ	La
11	ଟ	ta	29	ଧ	la
12	ଠ	tha	30	ନ	Sa
13	ଡ	da	31	ଫ	Sha
14	ଢ	dha	32	ବ	Ssa
15	ଣ	na	33	ଭ	Ha
16	ତ	ta	34	ମ	Kha
17	ଥ	tha	35	ଯ	Gra
18	ଦ	da	36	ୟ	Ya

Manuscript received on June 2013.

Suprava Das, CSE Department, International Institute of information technology, Bhubaneswar, India.

Rakesh ch. Balabantaray, CSE Department, International Institute of information technology, Bhubaneswar, India.

TABLE -2: Mapping of consonant from Odia to English

Sl. No.	Odia Dependent vowel	Odia Independent vowel	English vowel
1		ଅ	a
2	ା	ଆ	aa
3	ି	ଇ	i
4	ୀ	ଈ	ii
5	ୁ	ଉ	u
6	ୂ	ଊ	uu
7	େ	ଏ	e
8	ୈ	ଐ	Ae
9	ୋ	ଓ	O
10	ୌ	ଔ	Au
11	ୃ	ଋ	Ru
12		ୠ	Rru
13		ଌ	L
14		ୡ	Ll

IV. EXPERIMENTAL SET UP

Abbreviation forms an important class of named entities. So first we check whether the Odia string is an abbreviation in which English characters are spelled individually. The String of Odia language is first searched in database file created for abbreviation and month name, if the string matching found then its transliteration equivalent is substituted in place of Odia string. If the string matching not found in database, it is then transliterated using transliterated rule, which we are implementing using JDK, version-7. The system architecture is shown below in fig-1.

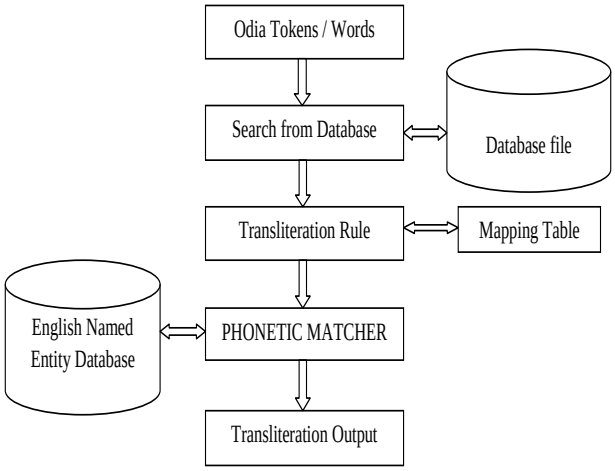

FIGURE – 3: System Architecture

Basic Transliteration Rule:

1. RULE-1

Start with an empty string. When a consonant or independent vowel (not as 'matraa') is encountered append the set of characters returned by mapping function. E.g. when we encounter ର we append 'ra'.

2. RULE -2

When a consonant is followed by a vowel the preceding 'a' should be removed and the character set for the vowel should be appended. E.g. ବି consists of two characters ବ + ି. Once we encounter ବ we append 'ba' and when ି is encountered next we remove the 'a' and append 'i'.

TABLE - 4

Odia NE	English NE	Transliteration Output	Solved Output
ଆର୍.ଭିନୀଲ କୃଷ୍ଣା	R.Vineel Krishna	ar.bhineel krishna	R.bhineel Krishna
ଏଲ୍. ଯାଦବ	L. Jadav	el. Jadaba	L. jadab
ଡି.କେ.ପାଣ୍ଡେ	D.K.Pandey	di.ke.pande	D.K.Pande
ୟୁ.ଏସ.ଏ	U.S.A	yu.esa.e	U.S.A

3. RULE -3

The observation of TABLE -4 gives the rule that if Odia string contains the symbol ".", then first the corresponding Odia string is matched with 3rd column in Table – 3. If match found then append corresponding character set returned by mapping function as given in Table – 3.

TABLE – 5

Odia NE	English NE	Transliteration Output	Solved Output
ଜୟରାମ	Jayaram	jayarama	Jayaram
ରଘୁନାଥ	Raghunath	raghunatha	Raghunath
ଅରବିନ୍ଦ	Arabinda	arabinda	Arabinda
ବଦ୍ରିନାରାୟଣ ପାତ୍ର	badrinarayan patra	badrinarayana patra	badrinarayan patra
ଜଗବନ୍ଧୁ ସିଂହ	jagabandhu singh	jagabandhu singha	jagabandhu singh
ରାମଚନ୍ଦ୍ର	Ramachandra	ramachandra	Ramachandra

4. RULE – 4

The observation of TABLE - 5 gives the rule that if the Odia Named Entity ends with a consonant is not following virama symbol '୍' then 'a' should be removed from last position in English spelling. Otherwise 'a' should be there.

TABLE -3: Mapping of English character to Odia

Sl. No.	English	Odia
1	A	ଏ
2	B	ବି
3	C	ସି
4	D	ଡି
5	E	ଇ
6	F	ଏଫ୍
7	G	ଜି
8	H	ଏଚ୍
9	I	ଆଇ
10	J	ଜେ
11	K	କେ
12	L	ଏଲ୍
13	M	ଏମ୍
14	N	ଏନ୍
15	O	ଓ
16	P	ପି
17	Q	କିଉ
18	R	ଆର୍
19	S	ଏସ୍
20	T	ଟି
21	U	ୟୁ
22	V	ଭି
23	W	ଡବ୍ଲ୍ୟୁ
24	X	ଏକ୍ସ
25	Y	ୱାଇ
26	Z	ଜେଡ୍

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

TABLE - 6

Odia NE	English NE	Transliteration Output	Solved output
ଝାଡ଼ଖଣ୍ଡ	Jharkhand	jharakhanda	Jharkhand
ଖୋର୍ଦ୍ଧା	khurdha	khorddha	Khurdha
ଭୁବନେଶ୍ୱର	Bhubaneswar	bhubaneshwar	Bhubaneswar

From the observation of TABLE - 6, most of the location and organization names are misspelled by transliteration rule. So we are making use of spellchecker algorithm i.e. jaccard coefficient combined with edit distance algorithm. A spellchecker works by searching for a given string in its **dictionary** of known strings. To correct a misspelling, the spellchecker assumes that your misspelling must be a mutation of one of the strings in its dictionary. To suggest a correction, the spellchecker searches its dictionary for strings similar ***to*** the misspelling.

To apply algorithm i have created the English NE database that contains around 1000 location name and 300 organization names. So finding edit distance of transliterated string with all entries may take a large time. So to improve the time complexity we combined Jaccard coefficient with edit distance algorithm.

Jaccard Coefficient

Given a set of elements in sets S1 and S2. Both sets may contain fraction of elements. The higher the fraction, the more similar the sets.

The similarity between two sets S1 and S2 is given by

$$\frac{|S1 \cap S2|}{|S1 \cup S2|} = \frac{|S1 \cap S2|}{|S1| + |S2| - |S1 \cap S2|}$$

Edit Distance Algorithm

Given two character strings S1 and S2, the edit distance between them is the minimum number of edit operations required to transform S1 into S2. The edit operations allowed for this purpose are:

a. Insert a character into a string
b. Delete a character from a string
c. Replace a character of a string by another character

For these operations edit distance is sometimes known as Levenshtein distance. E.g. edit distance between cat and dog is 3. For two strings S1 and S2 of length m and n respectively, the edit distance can be computed in Θ (mn) time using dynamic programming.

Spell Check algorithm we used

1. Calculate the Jaccard Coefficient of the misspelling with each string in the dictionary.
2. Collect those Strings with a jaccard score higher than threshold value **0.6**.
3. Then compute the edit distance algorithm on those set of strings.
4. Pick up the string with minimum edit distance (in this experiment threshold < **3**) and replace it in place of misspelled string if the starting character of transliterated output and string with minimum edit distance is same.

Some NEs even after applying edit distance algorithm remain unchanged i.e. misspelled. Most of such named entities are names of organization as shown in TABLE - 7. For such cases we are making use of soundex algorithm with little modification.

Soundex Algorithm

Soundex is the best-known phonetic matching scheme, developed by Odell and Russell, and patented in 1918 [Hall and Dowling, 1980]. Soundex uses codes based on the sound of each letter to translate a string into a canonical form of at most four characters, preserving the first letter.

Algorithm We Used

If word length of current NE > 1
1. Collect the entire target NEs from database whose word length matches with word length of current transliterated NE.
2. Compute soundex code for each word of current NE as well as all target NEs.
3. Select the target NE and substitute in place of current NE

If more than 50% words in target NE has same soundex code as corresponding soundex code of words in current NE
4. Otherwise current NE will be transliterated output.
Else
1. Compute soundex code of current NE and all one word length target NE
2. Select target NE and substitute in place of current NE if it is same otherwise not.

II. RESULT AND DISCUSSION

The main aim of our system is to translate named entity from Odia to English effectively. To evaluate the performance of transliteration system, word accuracy is calculated by using the following equation:

$$Accuracy = \frac{W}{T} * 100$$

Where W= Number of transliterated words that are correct and T= Total number of test words

Following Table-8 shows step by step evaluation of Odia to English Transliteration system.

TABLE – 7

Odia NE	Transliterated output	Solved Output
ଫୁଟବଲ ଆସୋସିଏସନ ଅଫ ଓଡିଶା	futabal asosiesan af odisha	Football association of odisha
ନ୍ୟାସନାଲ ଷ୍ଟକ ଏକ୍ସଚେଞ୍ଜ	nyasanal shtak eksacheng	National stock exchange
ଟ୍ୱିଟର	tyutar	twitter
ସାମସାଙ୍ଗ ଇଲେକ୍ଟ୍ରୋନିକ୍ସ କମ୍ପାନୀ	samasang ilektroniksa kampani	Samsung electronics company

TABLE - 8

Transliteration Steps	Evaluation Stages			
	1st	2nd	3rd	4th
Rule-based approach	☑	☑	☑	☑
Dictionary		☑	☑	☑
Use of Modified Edit Distance with Jaccard Coefficient			☑	☑
Use of Modified SoundEX algorithm				☑
Transliteration Accuracy (%)	46%	58%	72%	87%

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Following are some reasons for error in output.

a. Not actual Odia NE

In Odia document some words are written which actually corresponds to English NE. In these cases system fails to transliterate that particular word correctly.

b. Type mistake of words

Sometimes a word typed wrongly in document which results in incorrect transliteration output. Example- ସାରଦା ନାଯକ, ଯ is used instead of ୟ. So the transliteration output is najak instead of nayak.

c. One-to-multi mapping problem

Some Odia script has multiple characters mapping in English Script, which causes ambiguities. The multi mapping problem is associated with following characters-ଫ –FA, PHA ୱ – VA, BHA.

III. CONCLUSIONS

In this paper we also have addressed the problem of transliterating Odia to English language using rule based approach. The system is giving promising results and this can further used by researcher working on Odia and English NLP task**.**

REFERENCES

1. Ekbal Asif, Sudip Kumar Naskar and Sivaji Bandyopadhyay, "A Modified Joint Source-Channel Model for Transliteration", *Proceedings of ACL 2006*, pp 191-198, 2006.
2. Vijaya ,VP, Shivapratap and KP CEN(2009) "*English to Tamil Transliteration using WEKA system*‖ International Journal of Recent Trends in Engineering, May 2009, Vol. 1, No. 1, pages: 498-500.
3. Malik, "Punjabi Machine Transliteration System", In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL (2006), pages: 1137-1144.
4. Haque, Dandapat, Srivastava, Naskar and Way (2009) "*English—Hindi Transliteration Using Context-Informed PB-SMT: the DCU System for NEWS 2009*‖ Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 104–107,Suntec, Singapore, 7 August 2009. ACL and AFNLP.
5. Monika Bhargava, M.Kumar, Sujoy Das, "Rule Based Hindi to English Transliteration System for Proper Names", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 10, No. 8, August 2012.
6. Lehal G.S and Saini T.S., "A Hindi to Urdu Transliteration System", *Proceedings of ICON*, pp 235-240, 2010

AUTHORS PROFILE

Suprava Das, MTech from International Institute of Information Technology, BBSR, BTech from Ajay Binay Institute of Technology, Cuttack, "CASESTUDY of NER in ODIA using CRF++ tool" paper accepted in "IJACSA", Research Area – Natural language processing