International Journal of Information Technology and Knowledge Management
July-December 2011, Volume 4, No. 2, pp. 459-463

PUNJABI TO HINDI STATISTICAL MACHINE TRANSLITERATION

Gurpreet Singh Josan[1] & Jagroop Kaur[2]

Handling out of vocabulary (OOV) words in any MT application particularly in machine translation is a vital activity. Transliteration is the general choice for such words. This paper presents empirical results for statistical Punjabi to Hindi transliteration system. Experimental results show that statistical approach effectively improves the Transliteration accuracy rate and average Levenshtein distance of the various categories by a large margin.

Keywords: Transliteration, Statistical Approach, Transliteration Accuracy Rate.

1. Introduction

A quite common requirement for every NLP application is to deal with out-of-vocabulary words (OOV). Technical terms, proper names of person, places, objects etc. all occur frequently in everyday text and for a robust system it is necessary to identify such occurrences and process them differently. Generally, these words are transliterated as such in target script. In Transliteration, system converts an input string to a string in target alphabet, usually based on the phonetics of the original word. Thus the process is dependent on the availability of all the phonemes in target language. The transliteration is straightforward if all the phoneme representations are present in both languages e.g. the Hindi transliteration of Punjabi word "ਘਰ" [ghar] (home) is "घर" which is essentially pronounced in the same way. But in real world, this barely happens. Generally the two scripts vary and some of the sounds are missing or are extra in the target language. We have to map the missing or extra phonemes to the most phonetically similar letter, e.g., in Hindi we have alphabet "घ" but no such letter is present in English. So generally a similar sounding letter or combination of letters is used to denote such sounds as in the above case we use letter combination "gh" for representing above alphabet.

For several decades now, Roman transliteration has been used to represent in English Indian language texts. Extensive research has been carried out on methodologies for transliterating Indian scripts to and from the Romanized counterpart. Transliteration among Indian scripts is rather a neglected area. One of the simplest methods for transliteration is to use a dictionary that contains transliter ated text in target language for every entity in source language. Obviously, this is not a good option due to the doubt on the exhaustiveness of the dictionary. Another way is to perform the task by machine automatically. In this paper we will discuss the Punjabi to Hindi Machine transliteration system. We will use letter to letter mapping as baseline and try to find out the improvements by statistical methods.

[1]Department of Information Technology, RBIEBT, Mohali, INDIA
[2]Department of Computer Engineering, UCOE, Punjabi University, Patiala, INDIA
E-mail: [1]josangurpreet@rediffmail.in, [2]jagroop_80@rediffmail.in

2. Present Work

The topic of Machine transliteration has been studied extensively for several different language pairs, and many techniques have been proposed. Grapheme based and Phoneme Based are the two approaches found in literature. For instance, noisychannel model (NCM) (Lee & Chang, 2003; Virga et.al. 2003), HMM (Jung, Hong and Paek, 2000), statistical machine transliteration model (Lee et.al. 2003), and rulebased approach (Oh and Choi, 2002; Van and Verspoor, 1998). The phoneme-based approach has received remarkable attention in various works (Lee et.al. 2003; Oh et.al., 2002; Virga et. al. 2003; Jung et,al. 2000; Al-Onaizan and Knight, 2002). For Indian Languages, as mentioned earlier, Roman transliteration has been used to represent texts of Indian languages in English. ITRANS (Chopde A., 2001), RIT (Kanneganti and Kishore, from website http://www.teluguworld.org/RIT/rit.html), ADHAWIN (Srinivasan, 1995), MYLAI (KalyanaSundram K. from website http://tamilelibrary. org/teli/mylai1.html) etc., are the examples that uses above scheme. For Punjabi language, a Gurmukhi to Roman transliteration system using transliteration scheme based on ISO: 15919 transliteration and ALA-LC is developed at Punjabi University Patiala (Sharma, R. K., from website http:// www.advancedcentrepunjabi.org/). All these transliteration schemes are either from Roman to Indian languages or vice-versa. Transliteration among Indian languages is rather ignored. (Malik, M.G.A., 2006) has developed Machine transliteration system from Hindi to Urdu. Corpus based transliteration systems for Shahmukhi to Gurmukhi (Saini & Lehal, 2008) and from Gurmukhi to Shahmukhi (Lehal, 2009) have also been developed. A rule based machine transliteration system has also been developed for

transliterating from Hindi to Punjabi (Goyal & Lehal, 2008). The system is not reversible owe to the dependency on language specific rules. various steps used in the present face recognition system are discussed below.

3. GURMUKHI AND DEVANAGRI SCRIPTS

Gurmukhi, meaning "from the mouth of the Guru" is the most commonly used script in India for writing in Punjabi. Gurmukhi has descended from the Brahmi script of Ashoka. Gurmukhi was introduced by the second Guru of the Sikhs, Guru Angad Dev Ji, in the sixteenth century (Bhatia T.K., 1993). Gurmukhi has 38 consonants, 10 vowels alphabets (Independent vowels), 9 vowel symbols (Dependent vowels), 2 symbols for nasal sounds and 1 symbol that duplicates the sound of consonants (Bhatia 1993, Malik 2006). The N ̄ (lit. 'of the city') or Devan ̄ ('divine Nagari') alphabet descended from the Brahmi script some time around the 11th century AD. Devanagri has 65 consonants, 18 full vowel alphabets, 17 vowel symbols, 2 symbols for nasal sounds. Hindi uses only 11 vowel alphabets. In Hindi, there are thirty four consonantal syllables and thirteen vowels. Except minor differences, most of the alphabets are same in both the scripts. There are three aksharas of consonent clusters in Hindi which are written as a single atomic grapheme, i.e. त्र, ज्ञ, क्ष, but no such alphabets or consonant clusters are available in Gurmukhi. Punctuation in Punjabi is similar to Hindi.

4. APPROACH TO TRANSLITERATION FROM PUNJABI TO HINDI

4.1 Baseline Method (Letter to Letter Mapping)

Both Punjabi and Hindi are phonetic languages and their scripts represent the phonetic repository of their respective languages. These phonetic sounds are used to determine the relations between the alphabets of the two scripts. On the basis of this idea, character mappings are determined. Taking into account the similarity of both the scripts, letter to letter mapping is the obvious choice for the baseline computation. Alphabets are mapped using table 3.1.

Table 1
Alphabet Set of Gurmukhi & Devnagri Script

Gurmukhi	Devanagari	Gurmukhi	Devanagari	Gurmukhi	Devanagari	Gurmukhi	Devanagari	Gurmukhi	Devanagari	Gurmukhi	Devanagari	Gurmukhi	Devanagari
ੳ	-	ਾ	ा	ਕ	क	ਟ	ट	ਨ਼	ऩ	ਲ	ल	ਗ਼	ग़
ਅ	अ	ਿ	ि	ਖ	ख	ਠ	ठ	ਪ	प	ਲ਼	ळ	ਜ਼	ज़
ੲ	-	ੀ	ी	ਗ	ग	ਡ	ड	ਫ	फ	-	ऴ	ੜ	ड़
ਆ	आ	ੁ	ु	ਘ	घ	ਢ	ढ	ਬ	ब	ਵ	व	ਫ਼	फ़
ਇ	इ	ੂ	ू	ਙ	ङ	ਣ	ण	ਭ	भ	ਸ਼	श	ਢ਼	ढ़
ਈ	ई	ੇ	े	ਚ	च	ਤ	त	ਮ	म	-	ष	ਯ਼	य़
ਉ	उ	ੈ	ै	ਛ	छ	ਥ	थ	ਯ	य	ਸ	स	ੜ੍	ऱ
ਊ	ऊ	ੋ	ो	ਜ	ज	ਦ	द	ਰ	र	ਹ	ह	-	ऩ
ਏ	ए	ੌ	ौ	ਝ	झ	ਧ	ध	ੜ	ऱ	ਕ਼	क़	ਹ਼	ह़
ਐ	ऐ	-	ऋ	ਞ	ञ	ਨ	न	-	ऱ्	ਖ਼	ख़	ਵ਼	व़
ਓ	ओ	-	ॠ	-	ॡ								
ਔ	औ	-	ॐ	ੰ	॰								

4.2 Problems in Letter to Letter Mapping

The foremost problem is for the alphabets that have no mapping in target language. They never get mapped using this baseline system. Next is the multiple representation of the source alphabet in target character set e.g. e.g., ਸ may be mapped to श or ष. Another problem is the use of conjunct consonant forms in Hindi. In Hindi a syllable may consist of a vowel, a consonant followed by vowel or a consonant cluster followed by a vowel. The last form i.e., when two or more consonants are used within a word with no intervening vowel sound, is known as conjunct consonant. Use of conjunct consonants is limited in Punjabi. Only three letters can be used as conjuncts i.e., ਹ, ਰ, and ਵ. Their representation is also unique. It is not a trivial task to find out which combinations of alphabets in Punjabi will take conjunct consonant form in Hindi. For example, why, the word (ਨਿਊ [niū] (new) in Punjabi takes the conjunct consonant form in Hindi न्यू, is not clear. Also the mapping of nasal consonants is not clear. Nasal consonants in initial place in a conjunct may be expressed using the anusvara over the previous vowel, rather than as a half-glyph attached to the following consonant. It is written above the headstroke, at the right-hand end of the preceding character. In the list below, both spellings are correct and equivalent, although anusvara is preferred in the case of the first two: रंग = रङ्ग, पंजाबी = पञ्जाबी, हिंदी = हिन्दी, लंबा = लम्बा. Anusvara is still applied when previous character has its own vowel sign. If the vowel sign is [aa], the anusvara appears over the [aa], eg. फ्रांसीसी or आंदोलन. Also there is no rule to fine out when a sequence of alphabets in Punjabi is going to map in consonent cluster in Hindi, e.g., consider the two names written in Punjabi viz. ਸ਼ਿਤਿਜ {shitij} क्षितिज and ਸ਼ਿਕਾਕਾਈ {shikakai} शिकाकाई. In first name the consonant ਸ਼ is mapped to क्ष while in second name same is mapped to श.

4.3 Statistical Machine Transliteration

Assume that given a word, represented as a sequence of letters of the source language s = s1...sj...sJ, needs to be transcribed as a sequence of letters in the target language, represented as t = t1...ti...tI . The problem of finding the best target language letter sequence among the transliterated candidates can be represented as:

$$t_{best} = \text{argmax}\ t\{Pr\ (t \mid s)\} \quad (1)$$

We model the transliteration problem based on the noisy channel model. Reformulating the above equation using Bayes Rule:

$$t_{best} = \text{argmax}\ t\ p(s \mid t)\ p(s) \quad (2)$$

This formulation allows for a target language letters' n-gram model p(t) and a transcription model p(s | t). Given a sequence of letters s, the argmax function is a search function to output the best target letter sequence. SMT has already been tried for various language pair. Work in the field of Indian Language was done by Jaleel and Larkey (Larkey et al., 2003). They did this based on their work in English-Arabic transliteration for CLIR (Nasreen and Larkey, 2003). Their approach was based on HMM using GIZA++ (Och and Ney, 2000). We use GIZA++, SRILM and Moses toolkit, which are freely available, for developing language and transliteration model.

5. EVALUATION METHODOLOGY

Following combinations of approaches are tested for Punjabi—Hindi transliteration Task.

Baseline: As a Baseline for our experiments, we used a simple letter to letter based approach which maps Punjabi letters to the most likely letter in Hindi. We call it CASE-I.

Statistical Machine Transliteration: A statistical model is developed and used for transliterating the Punjabi text into Hindi text. This is termed as CASE II. The training data and development data consisted of a parallel corpus having entries in both Punjabi and Hindi. The training data and development data had 8000 entries and 1125 entries respectively. From the training and development data we have observed that the words can be roughly divided into following categories, Punjabi origin, Hindi Origin and other (includes English and other languages). The test data consisted of 1000 entries.

Human: For the purpose of comparison, we allowed an independent human subject (fluent in Punjabi but native speaker of Hindi) to perform the same task. The subject was asked to transliterate the Punjabi words in the test set without any additional context. No additional resources or collaboration were allowed. This output is used as gold standard for checking the system's performance.

Transliteration accuracy rate and Levenshtein distance is used for evaluation to capture the performance at word level and character level. Accuracy Rate is the percentage of correct transliteration from the total generated transliterations by the system. Average Levenshtein Distance is the average of Levenshtein distances between the transliterated word and reference word.

4.1 Results

Following are the results of this experiment.

Table 2
Transliteration Accuracy Rate and Avg Lev Dist.

	CASE I	CASE II
TAR	73.13%	87.72%
ALD	0.61	0.19

The baseline model produce 73.13% accuracy rate. The statistical method shows the improvements in performance by producing 87.72% accuracy rate. The breakup of figures is shown in following tables.

Table 3
TAR and ALD for Person Names, Location Names and Foreign Words

	Person Name		Location Name		Foreign Words	
	TAR	ALD	TAR	ALD	TAR	ALD
CASE I	75.85	0.59	67.10	0.66	63.50	0.67
CASE II	87.5	0.37	77.7	0.41	89.4	0.24

The Transliteration accuracy for words whose origin is also Punjabi is 86.6 in baseline model and shows the similar trend in other cases as shown in table 4. The improvement in all cases is registered with maximum for Hindi and other Languages.

Table 4
Transliteration Accuracy Rate and Avg. Lev Dist According to Origin of Source Language of Input Word

	Person Name		Location Name		Foreign Words	
	TAR	ALD	TAR	ALD	TAR	ALD
CASE I	86.6	0.41	63.54	0.76	70.1	0.79
CASE II	88.3	0.16	89.9	0.23	85.05	0.21

The accuracy in baseline model for Hindi is quite low because baseline model can not capture the half form representation of alphabets. As there is no half form in Punjabi so a Hindi name when spelt in Punjabi uses the full form of character instead of its half form. E.g. the name ਇ□ □ (Inder) when spelt in Punjabi will look as ਇਂਦਰ [indar]. Here half form of न and र in Hindi name are represented by (tippi-a character for nasal sound) and ਰ respectively. When this form is transliterated by baseline model it will produce इंदर which is wrong. Similar is the case with English and other foreign words. So due to the character gap in Punjabi and other languages, the Transliteration accuracy rate for baseline is low. It is also interesting to note that when words, which are originally from Hindi, are used in Punjabi are transliterated back to Hindi, the accuracy rate is lower than other types of words. The reason is that the Hindi words written in Punjabi is an approximate transliteration of original Hindi word in Punjabi. Depending upon the perception of transliterator, a Hindi word may have more than one representations in Punjabi, e.g., the name माइक्रोसौफ्ट (Microsoft) written in Hindi can be represented in Punjabi in number of ways. Some of them are ਮਾਇਕਰੋਸੋਫਟ, ਮਾਇਕ੍ਰੋਸੋਫਟ ਮਾਈਕਰੋਸੋਫਟ and ਮਾਈਕ੍ਰੋਸੋਫਟ. Only last representation, when again transliterated back into Hindi, convert to the correct representation.

Other representations are converted into such strings by baseline method that is incorrect. Normalization of spellings at the source may improve the results.

6. CONCLUSION

In this paper, we have described our transliteration system build on statistical techniques. This system can be developed with minimum efforts. All that is required is a parallel word list of source and target languages. There are many issues left for further improvement. The system itself could be improved by e.g. defining a better syllable similarity score, performing tuning of language model on various parameters like alignment heuristics, maximum phrase length etc. Comparing with other potential algorithms is also on future agenda.

REFERENCES

[1] Al-Onaizan Y. and Knight K., 2002, "Translating Named Entities Using Monolingual and Bilingual Resources", Proceedings of the 40th ACL, Philadelphia, 2002, pp. 400-408.

[2] Bhatia, T. K. 1993, "Punjabi: "A Cognitive-descriptive Grammar", Descriptive Grammars, Routledge, London.

[3] Chopde A. 2001, "Printing Transliterated Indian Language Documents", ITRANS, from website http://www. aczoom. com/itrans/idoc/idoc.html (Accessed on 22 Aug 2006)

[4] F.J. Och and H. Ney. 2003, "A Systematic Comparison of Various Statistical Alignment Models", Computational Linguistics, 29(1):19–51.

[5] Goyal V., Lehal G. S. 2009, "Hindi-Punjabi Machine Transliteration System (For Machine Translation System)", George Ronchi Foundation Journal, Italy, 64, n.1, 2009.

[6] Jung S. Y., Hong S. L. and Paek E., 2000. "An English to Korean Transliteration Model of Extended Markov Window", Proceedings of COLING 2000.

[7] Larkey, Connell, Abdul Jaleel. 2003. Hindi CLIR in Thirty Days. ACM Transactions on Asian Language Information Processing (TALIP) 2(2), 130–142 (2003)

[8] Lee C. and Chang J. S., 2003, "Acquisition of English-Chinese Transliteration Word Pairs from Parallel-Aligned Texts using a Statistical Machine Translation Model", Proceedings of HLT-NAACL Workshop: Building and Using parallel Texts Data Driven Machine Translation and Beyond, 2003, Edmonton, pp. 96-103.

[9] Lehal, Gurpreet Singh, "A Gurmukhi to Shahmukhi Transliteration System", In proceedings of ICON-2009: 7th International Conference on Natural Language Processing. Hyderabad. Pp 167 – 173.

[10] Malik M.G.A.. 2006. Punjabi Machine transliteration. Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL, pages 1137–1144.

[11] N. Abdul Jaleel and L.S. Larkey. 2003. "Statistical Transliteration for English-arabic Cross Language Information Retrieval", Proceedings of the Twelfth International Conference on Information and Knowledge Management, November 03-08, 2003, New Orleans, LA, USA.

[12] Och Franz Josef and Hermann Ney. 2000. "Improved Statistical Alignment Models, Proc. of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440-447, Hong Kong, China.

[13] Oh J.H. and Choi K.S. 2002, "An English-Korean Transliteration Model Using Pronunciation and Contextual Rules", Proceedings of the 19th International Conference on Computational Linguistics-1, pages 1–7.

[14] Saini T. S. and Lehal G. S. 2008, "Shahmukhi to Gurmukhi Transliteration System: A Corpus Based Approach", Research in Computing Science (Mexico), 33, pp. 151-162 (2008).

[15] Srinivasan 1995, "ADHAWIN", in "Transliteration Schemes" from website http://acharya.iitm.ac.in/multi_sys/transli/schemes.php.

[16] Virga P. and Khudanpur S. 2003. "Transliteration of proper names in crosslanguage applications", In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp: 365 –366.