Sangam: A Perso-Arabic to Indic Script Machine Transliteration Model

Gurpreet Singh Lehal
Department of Computer Science
Punjabi University, Patiala
147002 Punjab, India
gslehal@gmail.com

Tejinder Singh Saini
Advanced Center for Technical Development of
Punjabi Language Literature and Culture
Punjabi University, Patiala Punjab, India
tej@pbi.ac.in

Abstract

Indian sub-continent is one of those unique parts of the world where single languages are written in different scripts. This is the case for example with Punjabi, written in Indian East Punjab in Gurmukhi script (a Left to Right script based on Devnagri) and in Pakistani West Punjab, it is written in Shahmukhi (a Right to Left script based on Perso-Arabic). This is also the case with other languages like Urdu and Hindi (whilst having different names, they are the same language but written in mutually incomprehensible forms). Similarly, Sindhi and Kashmiri languages are written in both Persio-Arabic and Devanagri scripts. Thus there is a dire need for development transliteration tools for conversion between Perso-Arabic and Indic scripts. In this paper, we present Sangam, a Perso-Arabic to Indic script machine transliteration system, which can convert with high accuracy text written in Perso-Arabic script to one of the Indic script sharing the same language. Sangam is a hybrid system which combines rules as well as word and character level language models to transliterate the words. The system has been designed in such a fashion that the main code, algorithms and data structures remain unchanged and for a adding a new script pair only the databases, mapping rules and language models for the script pair need to be developed and plugged in. The system has been successfully tested on Punjabi, Urdu and Sindhi languages and can be easily extended for other languages like Kashmiri and Konkani.

1 Introduction

Indian sub-continent is one of those unique parts of the world where single languages are written in different scripts. This is the case for example with Punjabi, spoken by tens of millions of people, but written in Indian East Punjab (20 million) in Gurmukhi script (a Left to Right script based on Devnagri) and in Pakistani West Punjab (80 million), it is written in Shahmukhi (a Right to Left script based on Perso-Arabic). Whilst in speech, Punjabi spoken in the Eastern and the Western parts is mutually comprehensible in the written form it is not. This is also the case with other languages like Urdu and Hindi (whilst having different names, they are the same language but written, as with Punjabi, in mutually incomprehensible forms). Hindi is written in the Devnagri script from left to right, Urdu is written in a script derived from a Persian modification of Arabic script written from right to left. A similar problem resides with the Sindhi language, which is written in a Persio-Arabic script in Pakistan and both in Persio-Arabic and Devanagri in India. Similar is the case with Kashmiri language too. Konkani is probably the only language in India which is written in five scripts Roman, Devnagri, Kannada, Persian-Arabic and Malayalam (Carmen Brandt. 2014). The existence of multiple scripts has created communication barriers, as people can understand the spoken or verbal communication, however when it comes to scripts or written communication, the number diminishes, thus a need for transliteration tools which can convert text written in one language script to another script arises. A common feature of all these lan-

D S Sharma, R Sangal and J D Pawar. Proc. of the 11th Intl. Conference on Natural Language Processing, pages 232–239,
Goa, India. December 2014.

guages is that, one of the script is Perso-Arabic (Urdu, Sindhi, Shahmukhi etc.), while other script is Indic (Devnagri, Gurmukhi, Kannada, Malayalam). Perso-Arabic script is a right to left script, while Indic scripts are left to right scripts and both the scripts are mutually incomprehensible forms. Thus is a dire need for development of automatic machine transliteration tools for conversion between Perso-Arabic and Indic scripts.

Machine Transliteration is an automatic method to generate characters or words in one alphabetical system for the corresponding characters in another alphabetical system. The transformation of text from one script to another is usually based on phonetic equivalencies. Transliteration is usually categorized as forward and backward transliteration. Forward transliteration refers to transliteration from the native language to foreign language, while the process of recalling a word in native language from a transliteration is defined as back-transliteration. Forward transliteration plays an important role in natural language applications such as information retrieval and machine translation, especially for handling proper nouns, technical terms and out of vocabulary words. While back transliteration is popularly used as an input mechanism for certain languages, where typing in the native script is not very popular. In such cases, the user types the native language words and sentences (usually) in Roman script, and a transliteration engine automatically converts the Roman input back to the native script. This input mechanism is popularly used for all Indian languages including Hindi, Punjabi, Tamil, Telugu, etc., and also, Arabic, Chinese etc.

In this paper, we present Sangam, a Perso-Arabic to Indic script machine transliteration system, which can convert with high accuracy text written in Perso-Arabic script to one of the Indic script sharing the same language. The system has been successfully tested on Punjabi (Shahmukhi-Gurmukhi) , Urdu (Urdu-Devnagri) and Sindhi(Sindhi Perso Arabic - Sindhi Devnagri) languages and can be easily extended for other languages like Kashmiri and Konkani. One should note that the transliteration model presented in this paper can neither be categorized as forward nor as backward since it is concerned with script conversion in same language, so the usual techniques for forward or backward transliteration cannot be applied here and we have to develop a special methodology to handle the transliteration issues related to conversion between scripts of same language.

2 Related Work

The first transliteration system for a Perso-Arabic to Indic script was presented by Malik (2006), where he described a Shahmukhi to Gurmukhi transliteration system with 98% accuracy. But the accuracy was achieved only when the input text had all necessary diacritical marks for removing ambiguities, even though the process of putting missing diacritical marks is not practically possible due to many reasons like large input size, manual intervention, person having knowledge of both the scripts and so on. Saini et al. (2008) developed a system, which could automatically insert the missing diacritical marks in the Shahmukhi text and convert the text to Gurmukhi. The system had been implemented with various research techniques based on corpus analysis of both scripts and an accuracy of 91.37% at word level had been reported.

Durrani et al. (2010) presented an approach to integrate transliteration into Hindi-to-Urdu statistical machine translation. They proposed two probabilistic models, based on conditional and joint probability formulations and have reported an accuracy of 81.4%. Lehal and Saini (2012) presented an Urdu to Hindi transliteration system and had claimed achieving an accuracy of 97.74% at word level. The various challenges such as multiple/zero character mappings, missing diacritic marks in Urdu, multiple Hindi words mapped to an Urdu word, word segmentation issues in Urdu text etc. have been handled by generating special rules and using various lexical resources such as n-gram language models at word and character level and Urdu-Hindi parallel corpus. Recently Malik et al. (2013) have analysed the application of statistical machine translation for solving the problem of Urdu-Hindi transliteration using a parallel lexicon. The authors reported a word level accuracy of 77.8% when the input Urdu text contained all necessary diacritical marks and 77% when the input Urdu text did not contain all necessary diacritical marks, which is much below the accuracy reported in earlier works.

A rule based converter for Kashmiri language from Persio-Arabic to Devanagari script has been

developed by Kak et al. (2010) and authors have claimed 90% conversion accuracy.

Leghari and Rehman (2010) have discussed the different issues, complexities and problems of Sindhi transliteration and presented a model for transliteration between Perso-Arabic and Devanagari scripts of Sindhi language, which is based on an intermediate Roman script.

Malik et al. (2010) described a finite-state scriptural translation model based on Finite State Machines to convert the scripts for Urdu, Punjabi and Seraiki languages. But the transliteration results for Urdu-Hindi, Punjabi Shahmukhi-Gurmukhi and Seraiki Shahmukhi-Gurmukhi have not been very encouraging, with transliteration accuracy at word level ranging from 31.2% to 58.9% for Urdu-Devnagri script pair and 67.3% for Shahmukhi-Gurmukhi.

3 Challenges in Perso-Arabic to Indic Script Transliteration

Transliteration is not trivial to automate, but transliteration of Perso-Arabic script to Indic scripts is even more challenging problem. Since the language does not change, so it becomes important the correct spellings and context of the words is maintained in target script. The major challenges of transliteration of languages using Perso-Arabic script to Indic scripts are as follows:

3.1 Missing Diacritical marks and short Vowels

Diacritical marks are critical for correct pronunciation and sometimes even for disambiguation of certain words. The diacritical marks are also used for gemination (doubling of a consonant) and mark the absence of a vowel following a base consonant. But the diacritical marks and short vowels are sparingly used in Perso-Arabic script writings. These missing diacritical marks and short vowels create substantial difficulties for transliteration systems, as the missing diacritic marks and vowels have to be guessed by the system and added for correct transliteration. For example in Table 1, we see how the words in Perso-Arabic script, which are commonly written without diacritic marks, will be transliterated in Indic script, if we go in for character by character substitution and do not put the missing short vowels.

Perso-Arabic script	Word	Indic Script	Indic Transliteration	Actual transliteration
Urdu	دنیا	Devnagri	दनया	दुनिया
Shahmukhi	وچ	Gurmukhi	ਵਚ	ਵਿੱਚ
Sindhi	سنڌ	Devnagri	सनध	सिंधु

Table 1. Transliteration without diacritical marks

3.2 Filling the Missing Script Maps

There are many characters which are present in the Perso-Arabic script, corresponding to those having no character in Indic script, e.g. *Hamza* ء, *Do-Zabar* ً *Aen* ع, ٰ (*Khadi Zabar*) etc.

3.3 Multiple Mappings for Perso-Arabic Characters

It is observed that corresponding to many Perso-Arabic characters there are multiple mappings into Indic script as shown in Table 2. Additional information such as grammar rules and context are needed to select the appropriate Indic script character for such Perso-Arabic characters.

Perso-Arabic Script	Char	Indic script	Equivalent Mappings
Urdu	و	Devnagri	व, ो, ौ, ु, ू, ऊ, ओ, औ
Shahmukhi	ن	Gurmukhi	ं, ੰ, ਨ, ਣ

Table 2. Multiple Mappings of Perso-Arabic characters

3.4 Transliteration Ambiguity at Word level

Due to multiple character mappings and missing short vowels, many words in Perso-Arabic script get mapped to multiple Indic words as shown in Table 3. Higher level language information will be needed to choose the most relevant word in Indic script.

Perso-Arabic script	Word	Indic script	Equivalent words in Indic script
Urdu	کیا	Devnagri	क्या, किया
Shahmukhi	ہن	Gurmukhi	ਹਨ, ਹੁਣ
Sindhi	جان	Devnagri	जां, जान, जानि

Table 3. Multiple Mappings of Perso-Arabic words

3.5 Word-Segmentation Issues

Space is not consistently used in Perso-Arabic words, which makes word segmentation a non-trivial task. Many times the space is deleted resulting in many Perso-Arabic words being jumbled together and many other times extra space is put in word resulting in over segmentation of that word. This problem is more pronounced in Urdu and Shahmukhi scripts as compared to Sindhi script. We see in Table 4, samples of Urdu and Shahmukhi words containing multiple merged words and their transliterations if the words are transliterated as such without splitting them at proper positions.

Word Perso-Arabic	**Transliteration without splitting**	**Actual transliteration**
انکارکردیاہے (Urdu script)	अनकारकरदयाहे (Devnagri script)	इन्कार कर दिया है (Devnagri script)
پہلاشکار (Shahmukhi script)	ਪਹਲਾਸ਼ਕਾਰ (Gurmukhi script)	ਪਹਿਲਾ ਸ਼ਿਕਾਰ (Gurmukhi script)

Table 4. Merged Perso-Arabic Words and their Transliterations without splitting words

4 System Architecture

The system architecture of the general Perso-Arabic - Indic transliteration model developed by us is shown in Figure 1. The system has been designed in such a fashion that the main code, algorithms and data structures remain unchanged while depending on the script pair, the databases, mapping rules and language models need to be plugged in. The source text is in S1 script while the target text is in script S2. For example if text in Urdu script has to be converted to Devnagri, then we need to plug in word frequency list of Urdu and Urdu-Devnagri dictionary along with n-gram language models at word and character level for Devnagri script and mapping tables for Urdu to Devnagri transliteration. The system has been successfully tested on three script pairs(Urdu-Hindi, Sindhi-Devnagri and Shahmukhi-Gurmukhi) and has been able to successfully handle most of the issues raised in the previous section. We have developed the lexical resources for all the scripts and depending on our need, the relevant data is used. In case a new script pair has to be added, only the lexical resources have to be created As can be seen in figure 1, the complete transliteration system is divided into three stages: pre-processing, processing and post-processing. In the pre-processing stage, the text in S1 script is cleaned and prepared for transliteration by normalizing and joining the broken Perso-Arabic words. In the processing stage, corresponding to each word in S1 script, one or several possible words in S2 are generated. If only one word is produced, then that word is finalised. Otherwise for multiple alternatives, the final decision is taken in the post processing stage.

In the post-processing stage, the final decision about choosing from multiple S2 alternatives is made using language models for S2. The three stages are discussed in detail in the following sections.

4.1 Pre-Processing

In the pre-processing stage, the Urdu words are cleaned and prepared for transliteration by normalizing the Urdu words as well as joining the broken Urdu words. The two main stages in pre-processing are:

4.1.1 Normalizing Perso-Arabic words

Two kinds of normalization are required for Perso-Arabic words. First, a letter may be represented by multiple Unicode points, and thus the redundancy in encoding has to be cleaned in raw text before further processing. As for example, from transliteration point of view, ى(0649), ي(064a) and ى(06cc) represent the same character in Perso-Arabic script. Secondly, a letter or a ligature is sometimes encoded in composed form as well as decomposed form. Thus, the two equivalent representations must also be reduced to same underlying form before further processing. For example, آ (0622) can be also represented by the combination ا (0627) + ◌ٓ (0653). All such forms are normalized to have only one representation.

4.1.2 Joining the broken Perso-Arabic words

The transliteration system faces many problems related to word segmentation of Perso-Arabic script, as in many cases space is not properly put between words. Sometimes it is deleted resulting in many Perso-Arabic words being jumbled together and many other times extra space is put in word resulting in over segmentation of that word. The space insertion problem is handled in pre-processing stage, while the space deletion problem is handled in the processing stage. The space inser-

tion problem usually occurs due to conventional way of writing in Perso-Arabic script or due to extra space being inserted during typing. The typing related space insertion problems are handled by using the word frequency list of script S1 (Lehal, 2009). If the product of probability of occurrence of two adjacent words in S1 is lesser than the probability of occurrence of the word formed by joining the two, then the two words are joined together.

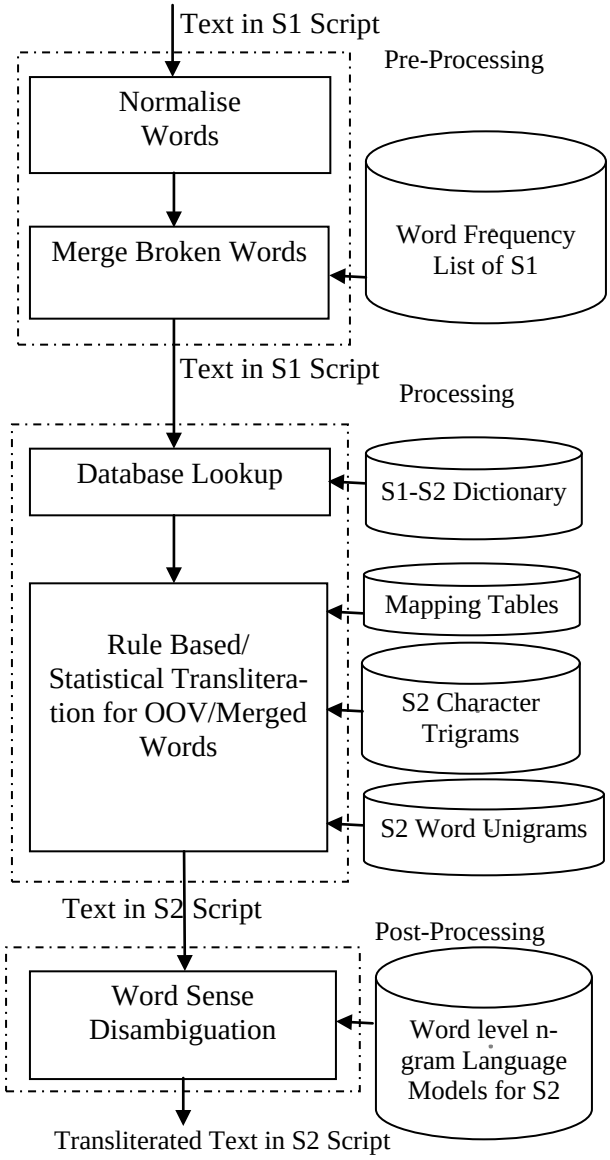

Figure 1. System Architecture

4.2 Processing Stage

This is the main stage and in this stage, corresponding to each word in S1 script, one or several possible words in S2 script are generated. For multiple alternatives, the final decision is taken in the post processing stage. First the word is searched in the S1-S2 dictionary and if it is found, then all its alternatives are passed onto post processing stage. In case the word is not found, then it is fed to a multi-stage transliteration engine. In the first stage a Hybrid-wordlist-generator (HWG) is used to convert the word. The HWG uses the mapping rules and a trigram character language model to generate a set of words in S2. A unigram word language model is then used to rank these words, after dropping words with zero probability. If there is no word with non-zero probability, then the word is inspected for presence of merged words which can be transliterated to non empty sets of words in S2 words. If no such sets can be generated then we use the simple character mapping rules to convert the word to S2.

We now discuss in detail, the main modules used in the multi-stage transliteration engine. These modules are:

4.2.1 Hybrid-wordlist-generator (HWG)

This is the major module in the transliteration engine. It generates multiple transliterations for a word in S1. The multiple outputs are produced due to ambiguity both at character and word level as already mentioned in above sections.

The sequence of probable Indic words is produced by a hybrid system, which uses rule based character mapping tables and a trigram character Language Model. The Perso-Arabic word is processed character by character, which are mapped directly to their corresponding similar sounding Indic characters based on their position in word and syntax rules (snippet shown in the Table 5). In most of the cases, there is a 1-1 mapping, but a few characters such as ن ,و ,ى, which have multiple mappings and also some character combinations in Perso-Arabic script such as تھ (062A+06BE) have single representation in some of Indic scripts such as in Devnagri (थ) or Gurmukhi (ਥ), while in Sindhi(Devnagri) we have character combination तह. Similarly the character ۾ (U06FE) gets mapped to word में in Sindhi(Devnagri) script, while it has no equivalent mapping in Devnagri and Shahmukhi scripts. In some cases the mapping is dependent on the position of the character. As for example, the character ا (U0627) is mapped to character अ if it is in beginning of the word else it gets mapped to अ and ◌ा in Devnagri script. Similarly, the character ن

(U0646) gets mapped to character न in Sindhi(Devnagri), if it is in the starting of the word, and gets mapped to न, ं and ँ otherwise. The same character, ن (U0646), gets mapped to न and ण in Devnagri, if it comes in beginning or ending of a word, else it gets mapped to न, ण, ं and ँ .

Char At	Perso Arabic	Devnagri	Gurmukhi	Sindhi (Dev)
Any	ب	ब	ਬ	ब
	تھ	थ	ਥ	तह
	ہٕ	-	-	में
Start	و	व, ऊ, ओ, औ	ਵ, ਊ, ਓ, ਔ	व, ऊ, ओ, औ
Start	ن	न, ण	ਨ, ਣ	न
Start	ا	अ	ਅ	अ
Mid	ن	न, ण, ं, ँ	ਨ, ਣ, ਂ, ਁ	न, ं, ँ
Mid/ end	ا	अ, ा	ਅ, ਾ	अ, ा
Mid /end	و	व, ो, ौ, ु, ू, ऊ, ओ, औ	ਵ, ੋ, ੌ, ੁ, ੂ, ਊ, ਓ, ਔ	व, ो, ौ, ु, ू, ऊ, ओ, औ
End	ن	न, ण	ਨ, ਣ	न, ं, ँ

Table 5. Portion of Perso Arabic - Indic character mapping Tables

As already mentioned, the short vowels and diacritical marks are usually omitted in Perso-Arabic text and there are no half characters in Perso-Arabic script, so the result is that the resultant text in Indic script has poor accuracy. For example, the Urdu word in Perso-Arabic script, قسمت, gets transliterated to क़समत, while the actual word in Devnagri should be क़िस्मत. This is because the character स is written as half character in Devnagri while the short vowel ि which is missing in original Perso-Arabic word has to written in Devnagri to maintain proper spellings in Devnagri. To fill these missing diacritical marks and put half characters at appropriate locations in Indic word, we consider all its possible mappings in Indic which include the missing short vowels and half characters. So we modify our mapping table to include all such forms for all the Perso-Arabic characters resulting in multiple mappings. Thus for example the character combination تھ (062a+06be) and character و (0648) get mapped to as shown in in Table 6.

We form all possible combinations, which could be generated from these multiple mappings and the top N combinations are retained. The character based trigram language model for Indic script is used to select the top N combinations. To avoid processing exponential number of candidates, we process the input characters one at a time and use the character trigram probability for pruning the partially generated candidates at each step.

Char At	Perso Arabic	Devnagri	Gurmukhi	Sindhi (Dev)
Any	تھ	थ, थ्, थि, थु	ਥ, ਥਿ, ਥੁ	तह, तहि, तिहि, तिह, तुह, तुहु, तह्
Start	و	व, व्, वु, वि, व्व, व्वु, व्वि, ऊ, ओ, औ	ਵ, ਵਿ, ਵੁ, ਊ, ਓ, ਔ	व, व्, वु, वि, व्व, व्वु, व्वि, ऊ, ओ, औ
Mid/ end	و	व, व्, वु, वि, व्व, व्वु,, व्वि, ों, ौ, ु, ू, ऊ, ओ, औ	ਵ, ਵਿ, ਵੁ, ੋ, ੌ, ੁ, ਊ, ਓ, ਔ	व, व्, वु, वि, व्व, व्वु,, व्वि, ों, ौ, ु, ू, ऊ, ओ, औ

Table 6. Portion of Modified Perso Arabic - Indic character mapping Tables

It should be noted that not all the suggestions generated by the character language model are valid words in Indic script. To further rank these words, we use the Unigram Word Model for Indic script. Words with zero probability in the Unigram word model are ignored and rest of the words are ranked on their probabilities. It could also happen that all the top N alternatives suggested by the character level trigram may be having zero probabilities, in which case no alternative is returned by HWG.

4.2.2 Merged word segmentation

As already discussed above, space is not consistently used in Perso-Arabic, which gives rise to both space omission and space insertion errors. Due to the space deletion problem, a sequence of words is jumbled together as a single word and when the HWG tries to generate the equivalent Indic alternatives it fails. The sequence of Perso-Arabic words written together without space is still readable because of the character joining property in Perso-Arabic. We have used the space deletion algorithm presented by Lehal (Lehal, 2010) to split the Perso-Arabic words. The algorithm makes use of unigram wordlist of S2 and statistical word disambiguation techniques to first detect if the Perso-

Arabic word contains multiple words. And in case multiple words are present, the algorithm splits them at appropriate positions. The Indic Script alternatives for these individual Perso-Arabic words are then generated using the HWG module.

4.2.3 Handling Out of Vocabulary Words

For out of vocabulary words, no possible suggestions will be generated by the dictionary or HWG. So if after passing through all the modules, still no transliteration alternatives are generated, it implies that the word is out of vocabulary. For such words the Indic Script word is generated by using the mapping rules and trigram character language model and the top most alternative is selected for further processing.

4.3 Post Processing Stage

The main task of post processing is to select the best alternative amongst the various transliteration options. The HWG module presents a set of ranked transliterations instead of a single transliteration, due to multiple character mappings as shown in Table 5 and 6. Up to this point, we were only considering the Indic words in isolation, without any consideration to their neighbouring words. Now we consider the whole sentence instead of isolated words.

$$\Pr(w_i \mid w_{i-2} w_{i-1}) = \lambda_3 \frac{c(w_{i-2} w_{i-1} w_i)}{c(w_{i-2} w_{i-1})} + \lambda_2 \frac{c(w_{i-1} w_i)}{c(w_{i-1})} + \lambda_1 \frac{c(w_i)}{N} + \lambda_0 \frac{1}{V}$$

Where

N = Number of words in the training corpus,
V = Size of the vocabulary

To choose between the different alternatives we have used the word trigram probability. To take care of the sparseness in the trigram model, we have used deleted interpolation, which offers the solution of backing away from low count trigrams by augmenting the estimate using bigram and unigram counts. The deleted interpolation trigram model assigns a probability to each trigram which is the linear interpolation of the trigram, bigram, unigram and uniform models. The weights are set automatically using the Expectation-Maximization (EM) algorithm.

5. Experimental Results

We have tested our system on text in Perso-Arabic script in Urdu, Punjabi and Sindhi languages and converted it to respective Indic scripts. The transliterated text has been manually evaluated. The results are tabulated in Table 7. We can see from the table, that the transliteration accuracy for the three scripts ranges from 91.68% to 97.75%, which is the best accuracy reported so far in literature for script pairs in Perso-Arabic and Indic scripts. As can be observed the transliteration accuracy for Sindhi language is much lesser as compared to Urdu and Punjabi languages. The main reasons for this are:

a) Lack of linguistic resources and digital text in Sindhi(Devnagri).

b) High level of ambiguity at word level in Sindhi(Perso-Arabic) words, which is much more pronounced than Shahmukhi and Urdu words.

A sample of the output for the three scripts is shown in Figure 2.

Script Pair	Words	Transliteration Accuracy
Urdu-Devnagri	30,248	97.75%
Shahmukhi-Gurmukhi	26,141	97.02%
Sindhi (Perso Arabic) to Sindhi (Devnagri)	29,131	91.68%

Table 7. Word level Transliteration Accuracy of different script pairs

سب کام اپنے وقت پر ہی ہوتے ہیں۔ ہمیں کام کرنا چاہیے۔ پھل کی فکر نہیں کرنی چاہیے۔

सब काम अपने वक़्त पर ही होते हैं। हमें काम करना चाहिए।फल की फ़िक्र नहीं करनी चाहिए।

a) Urdu-Devnagri

سارے کم اپنے سمیں سر ہی ہندے ہن۔ سانوں کم کرنا چاہیدا ہے، پھل دی چنتا نہیں کرنی چاہیدی۔

ਸਾਰੇ ਕੰਮ ਆਪਣੇ ਸਮੇਂ ਸਿਰ ਹੀ ਹੁੰਦੇ ਹਨ। ਸਾਨੂੰ ਕੰਮ ਕਰਨਾ ਚਾਹੀਦਾ ਹੈ, ਫਲ ਦੀ ਚਿੰਤਾ ਨਹੀਂ ਕਰਨੀ ਚਾਹੀਦੀ।

b) Shahmukhi-Gurmukhi

سڀ ڪم پنهنجي وقت تي ئي ٿيندا آهني۔ اسان کي ڪم ڪرڻ گهرجي، ڦل جِي چنتا نه ڪرڻ گهرجي ۔

सभु कम पंहिंजे वक़्त ते ई थींदा आहनी। असां खे कमु करणु घुर्जे, फल जी चिंता न करणु घुर्जे ।

c) Sindhi (Perso-Arabic) - Sindhi (Devnagri)

Figure 2. Samples of Transliteration output of text in three languages (Urdu, Punjabi and Sindhi) by Sangam

Conclusion

In this paper, we have presented Sangam, a Perso-Arabic to Indic script machine transliteration model, which can convert with high accuracy text written in Perso-Arabic script to one of the Indic script sharing the same language. The system has been successfully tested on Punjabi, Urdu and Sindhi languages and can be easily extended for other languages like Kashmiri and Konkani. The transliteration accuracy for the three languages ranges from 91.68% to 97.75%, which is the best accuracy reported so far in literature for translateration from Perso-Arabic to Indic script. The system has been designed in such a fashion that the main code, algorithms and data structures remain unchanged and for a adding a new script pair only the databases, mapping rules and language models for the script pair need to be developed and plugged in.

Acknowledgments

The authors would like to acknowledge the support provided by PAN ASIA Grants Singapore and ISIF grants Australia for carrying out this research. The Sindhi language support provided by Dr. Bharat Ratanpal and Ms. Madhuri Wardey, Faculty of Technology & Engineering, MSU Baroda is also duly acknowledged.

References

Aadil Amin Kak, Nazima Mehdi and Aadil Ahmad Lawaye. 2010. Building a Cross Script Kashmiri Converter: Issues and Solutions, In *Proceedings of Oriental COCOSDA (The International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques)*. web access : http://desceco.org/O-COCOSDA2010/proceedings/paper_38.pdf

Carmen Brandt. 2014. Script as a Potential Demarcator and Stabilizer of Languages in South Asia Language, *Documentation & Conservation Special Publication No. 7 in Language Endangerment and Preservation in South Asia*, ed. by Hugo C. Cardoso, pp. 78-99

Durrani, N., Sajjad, H., Fraser, A. and Schmid, H. 2010. Hindi-to-Urdu Machine Translation through Transliteration. *In Proceedings of the 48th Annual Conference of the Association for Computational Linguistics*, pp 465–474, Uppsala, Sweden.

G. S. Lehal, 2009. A Two Stage Word Segmentation System For Handling Space Insertion Problem In Urdu Script, *Proceedings of World Academy of Science, Engineering and Technology*, Bangkok, Thailand, Vol. 60, pp 321-324.

G. S. Lehal, 2010. A Word Segmentation System for Handling Space Omission Problem in Urdu Script, *Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP)*, the 23rd International Conference on Computational Linguistics (COLING), pp. 43–50, Beijing.

Gurpreet S. Lehal and Tejinder S. Saini. 2012. Development of a complete Urdu-Hindi transliteration system. In *Proceedings of the 24th International Conference on Computational Linguistics*, pp 643–652, Mumbai, India.

Leghari, M., and Rahman, M. U. 2010. Towards Transliteration between Sindhi Scripts by using Roman Script. *Conference on Language and Technology*. Islamabad: National Language Authority, Pakistan. http://www.cle.org.pk/clt10/papers/Towards%20Transliteration%20between%20Sindhi%20Scripts%20by%20using%20Roman%20Script.pdf

M. G. Abbas Malik, Christian Boitet, and Pushpak Bhattacharyya. 2010. Finite-state scriptural translation. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (COLING '10). ACL, pp 791-800, Stroudsburg, PA, USA.

Malik, M. G. Abbas. 2006. Punjabi Machine Transliteration. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pp 1137-1144.

Malik. M. G. Abbas, Boitet. Christian, Besacier. Laurent, Bhattcharrya. Pushpak. 2013 Urdu Hindi Machine Transliteration using SMT, *The 4th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), a collocated event at International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 43-57, Nagoya, Japan.

T. S. Saini, G. S. Lehal and V. S. Kalra. 2008. Shahmukhi to Gurmukhi Transliteration System, *Coling: Companion volume: Posters and Demonstrations*, pp. 177-180, Manchester, UK.