Shahmukhi to Gurmukhi Transliteration System: A Corpus based Approach

Tejinder Singh Saini[1] and Gurpreet Singh Lehal[2]

[1] Advanced Centre for Technical Development of Punjabi Language, Literature & Culture, Punjabi University, Patiala 147 002, Punjab, India
tej@pbi.ac.in
http://www.advancedcentrepunjabi.org
[2] Department of Computer Science, Punjabi University, Patiala 147 002, Punjab, India
gslehal@yahoo.com

Abstract. This research paper describes a corpus based transliteration system for Punjabi language. The existence of two scripts for Punjabi language has created a script barrier between the Punjabi literature written in India and in Pakistan. This research project has developed a new system for the first time of its kind for Shahmukhi script of Punjabi language. The proposed system for Shahmukhi to Gurmukhi transliteration has been implemented with various research techniques based on language corpus. The corpus analysis program has been run on both Shahmukhi and Gurmukhi corpora for generating statistical data for different types like character, word and n-gram frequencies. This statistical analysis is used in different phases of transliteration. Potentially, all members of the substantial Punjabi community will benefit vastly from this transliteration system.

1 Introduction

One of the great challenges before Information Technology is to overcome language barriers dividing the mankind so that everyone can communicate with everyone else on the planet in real time. South Asia is one of those unique parts of the world where a single language is written in different scripts. This is the case, for example, with Punjabi language 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 Devanagari*) and in Pakistani West Punjab (80 million), written in Shahmukhi script (*a right to left script based on Arabic*), and by a growing number of Punjabis (2 million) in the EU and the US in the Roman script. While in speech Punjabi spoken in the Eastern and the Western parts is mutually comprehensible, in the written form it is not. The existence of two scripts for Punjabi has created a script barrier between the Punjabi literature written in India and that in Pakistan. More than 60 per cent of Punjabi literature of the medieval period (500-1450 AD) is available in Shahmukhi script only, while most of the modern Punjabi writings are in Gurmukhi. Potentially, all members of the substantial Punjabi community will benefit vastly from the transliteration system.

Advances in Natural Language Processing and Applications
Research in Computing Science 33, 2008, pp. 151-162

Received 25/10/07
Accepted 07/12/07
Final Version 22/01/08

2 Related Work

Most of the available work in Arabic-related transliteration has been done for the purpose of machine translation. In the paper titled "Punjabi Machine Transliteration (PMT)" Malik A. 2006 [1] has demonstrated a very simple rule-based transliteration system for Shahmukhi to Gurmukhi script. Firstly, two scripts are discussed and compared. Based on this comparison and analysis, character mappings between Shahmukhi and Gurmukhi scripts have been drawn and transliteration rules formulated. Along with this only dependency rules have been formed for special characters like aspirated consonants, non-aspirated consonants, Alif ا[ə], Alif Madda آ[ɑ], Vav و[v] , Choti Ye ى[j] etc. The primary limitation of this system is that this system works only on input data which has been manually edited for missing vowels or diacritical marks (*the basic ambiguity of written Arabic script*) which practically has limited use. Some other transliteration systems available in literature are discussed by Haizhou et al (2004) [3], Youngim et al (2004) [4], Nasreen et al (2003) [5] and Stalls et al (1998) [9].

3 Major Challenges

The major challenges of transliteration of Shahmukhi to Gurmukhi script are as follows:

3.1 Recognition of Shahmukhi Text without Diacritical Marks

Shahmukhi script is usually written without short vowels and other diacritical marks, often leading to potential ambiguity. Arabic orthography does not provide full vocalization of the text, and the reader is expected to infer short vowels from the context of the sentence. Like Urdu, in the written Shahmukhi script it is not mandatory to put short vowels below or above the Shahmukhi character to clear its sound. These special signs are called "Aerab" in Urdu. It is a big challenge in the process of machine transliteration or in any other process to recognize the right word from the written text because in a situation like this, correct meaning of the word needs to be distinguished from its neighboring words or, in worst cases, we may need to go into deeper levels of n-gram.

3.2 Filling the Missing Script Maps

There are many characters which are present in the Shahmukhi script, corresponding to those having no character in Gurmukhi, e.g. Hamza ء [ɪ], Do-Zabar ً [ən], Do-Zer ٍ [ɪn], Aen ع[ʔ] etc.

3.3 Multiple Mappings

It is observed that there is multiple possible mapping into Gurmukhi script corresponding to a single character in the Shahmukhi script as shown in Table 1.

Table 1. Multiple Mapping into Gurmukhi Script

Name	Shahmukhi Character	Unicode	Gurmukhi Mappings
Vav	و [v]	0648	ਵ [v], ੋ [o], ੌ [ɔ], ੁ [ʊ], ੂ [u], ੋ [o]
Ye Choti	ی[j]	0649	ਯ [j], ਿ [ɪ], ੇ [e], ੈ[æ], ੀ[i], ੀ [i]

3.4 Word-Boundary Mismatch

Urdu Zabata Takhti (UZT) 1.01 [2] has the concept of two types of spaces. The first type of space is normal space and the second type of space is given name Hard Space (HS). The function of hard space is to represent space in the character sequence that represents a single word. In Unicode character set this Hard Space is represented as Zero Width Non Joiner (ZWNJ). But it is observed that in the written text normal space is used instead of hard space. Therefore, transliterating a single word of Shahmukhi with space in between will generate two tokens of the same word in Gurmukhi script.

4 Script Mappings

4.1 Gurmukhi Script

The Gurmukhi script, derived from the Sharada script and standardised by *Guru Angad Dev* in the 16th century, was designed to write the Punjabi language. The meaning of "Gurmukhi" is literally "*from the mouth of the Guru*". As shown in Table 2 the Gurmukhi script has forty one letters, including thirty eight consonants and three basic vowel sign bearers (*Matra Vahak*). The first three letters are unique because they form the basis for vowels and are not consonants. The six consonants in the last row are created by placing a *dot* at the foot (pair) of the consonant (*Naveen Toli*). There are five nasal consonants (ਙ[ɲə], ਞ[ɲə], ਣ[ɳ], ਨ[n], ਮ[m]) and two additional nasalization signs, bindi ਂ [ɲ] and tippi ੰ [ɲ] in Gurmukhi script. In addition to this, there are nine dependent vowel signs (ੁ[ʊ], ੂ [u], ੋ[o], ਾ[ə], ਿ[ɪ], ੀ[i], ੇ[e], ੈ[æ], ੌ[ɔ]) used to create ten independent vowels (ਉ [ʊ], ਊ [u], ਓ [o], ਅ [ə], ਆ [ɑ], ਇ [ɪ], ਈ [i], ਏ [e], ਐ [æ], ਔ [ɔ]) with three bearer characters: Ura ੳ[ʊ], Aira ਅ [ə] and Iri ੲ[ɪ]. With the exception of Aira ਅ [ə] independent vowels are never used without additional vowel signs. Some Punjabi words require consonants to be written

in a conjunct form in which the second consonant is written under the first as a subscript. There are only three commonly used subjoined consonants as shown here Haha ਹ[h] (usage ਨ[n] +੍+ਹ[h] = ਨ੍ਹ [nʰ]), Rara ਰ[r] (usage ਪ[p] +੍+ਰ[r] =ਪ੍ਰ [prʰ]) and Vava ਵ[v] (usage ਸ[s] +੍+ਵ[v] = ਸ੍ਵ [sv]).

Table 2. Gurmukhi Alphabet

ੳ	ਅ[ə]	ੲ				*Matra Vahak*
			ਸ[s]	ਹ[h]		*Mul Varag*
ਕ[k]	ਖ[kʰ]	ਗ[g]	ਘ[kʰ]	ਙ[ŋə]		*Kavarg Toli*
ਚ[ʧ]	ਛ[ʧʰ]	ਜ[ʤ]	ਝ[ʤʰ]	ਞ[ɲə]		*Chavarg Toli*
ਟ[ʈ]	ਠ[ʈʰ]	ਡ[ɖ]	ਢ[ɖʰ]	ਣ[ɳ]		*Tavarg Toli*
ਤ[t]	ਥ[tʰ]	ਦ[d]	ਧ[dʰ]	ਨ[n]		*Tavarg Toli*
ਪ[p]	ਫ[pʰ]	ਬ[b]	ਭ[bʰ]	ਮ[m]		*Pavarg Toli*
ਯ[j]	ਰ[r]	ਲ[l]	ਵ[v]	ੜ[ɽ]		*Antim Toli*
ਸ਼[ʃ]	ਖ਼[x]	ਗ਼[ɣ]	ਜ਼[z]	ਫ਼[f]	ਲ਼[ɭ]	*Naveen Toli*

4.2 Shahmukhi Script

The meaning of "Shahmukhi" is literally "from the King's mouth". Shahmukhi is a local variant of the Urdu script used to record the Punjabi language. It is based on right to left Nastalique style of the Persian and Arabic script. It has thirty seven simple consonants, eleven frequently used aspirated consonants, five long vowels and three short vowel symbols.

4.3 Mapping of Simple Consonants

Unlike Gurmukhi script, the Shahmukhi script does not follow a *'one sound-one symbol'* principle. In the case of non-aspirated consonants, Shahmukhi has many character forms mapped into single Gurmukhi consonant. This has been highlighted in Table 3 below.

4.4 Mapping of Aspirated Consonants (AC)

In Shahmukhi script, the aspirated consonants are represented by the combination of a simple consonant and HEH-DAOCHASHMEE ھ[h]. Table 4 shows 11 frequently used aspirated consonants in Shahmukhi corresponding to which Gurmukhi script has unique single character except the last one ੜ੍ਹ [ɽʰ] having compound characters.

Table 3. Shahmukhi Non-Aspirated Consonents Mapping

Sr.	Char	Code	Gurmukhi	Code	Sr.	Char	Code	Gurmukhi	Code
1	ب[b]	0628	ਬ [b]	0A2C	20	ع[ʔ]	0639	ਅ [ə]	0A05
2	پ[p]	067E	ਪ [p]	0A2A	21	غ[ɣ]	063A	ਗ਼ [ɣ]	0A5A
3	ت[t̪]	062A	ਤ [t̪]	0A24	22	ف[f]	0641	ਫ਼ [f]	0A5E
4	ٹ[ʈ]	062B	ਸ [s]	0A38	23	ق[q]	0642	ਕ [k]	0A15
5	ج[ʤ]	062C	ਜ [ʤ]	0A1C	24	ک[k]	06A9	ਕ [k]	0A15
6	چ[ʧ]	0686	ਚ [ʧ]	0A1A	25	گ[g]	06AF	ਗ [g]	0A17
7	ح[h]	062D	ਹ [h]	0A39	26	ل[l]	0644	ਲ [l]	0A32
8	خ[x]	062E	ਖ਼ [x]	0A59	27	م[m]	0645	ਮ [m]	0A2E
9	د[ɖ]	062F	ਦ [ɖ]	0A26	28	ن[n]	0646	ਨ [n], ◌ੰ [ɲ]	0A28, 0A70
10	ذ[z]	0630	ਜ਼ [z]	0A5B	29	ڻ[ɳ]	06BB	ਣ [ɳ]	0A23
11	ر[r]	0631	ਰ [r]	0A30	30	و[v]	0648	ਵ [v]	0A35
12	ز[z]	0632	ਜ਼ [z]	0A5B	31	ہ[h]	06C1	ਹ [h]	0A39
13	ژ[ʒ]	0698	ਜ਼ [z]	0A5B	32	ی[j]	06CC	ਯ [j]	0A2F
14	س[s]	0633	ਸ [s]	0A38	33	ے[j]	06D2	ਯ [j]	0A2F
15	ش[ʃ]	0634	ਸ਼ [ʃ]	0A36	34	ھ[h]	06BE	◌੍ਹ [h]	0A4D +0A39
16	ص[s]	0635	ਸ [s]	0A38	35	ٹ[ʈ]	0679	ਟ [ʈ]	0A1F
17	ض[z]	0636	ਜ਼ [z]	0A5B	36	ڈ[ɖ]	0688	ਡ [ɖ]	0A21
18	ط[t̪]	0637	ਤ [t̪]	0A24	37	ڑ[ɽ]	0691	ੜ [ɽ]	0A5C
19	ظ[z]	0638	ਜ਼ [z]	0A5B					

Table 4. Aspirate Consonants (AC) Mapping

Sr.	AC ھ[h]	Code (06BE)	Gurmukhi	Code	Sr.	AC ھ[h]	Code (06BE)	Gurmukhi	Code
1	بھ[b]	0628	ਭ [b]	0A2D	7	ڈھ[ɖ]	062F	ਧ [ɖ]	0A27
2	پھ[p]	067E	ਫ [p]	0A2B	8	ٹھ[ʈ]	0679	ਠ [ʈ]	0A20
3	تھ[t̪]	062A	ਥ [t̪]	0A25	9	کھ[k]	06A9	ਖ [k]	0A16
4	ڈھ[ɖ]	0688	ਢ [ɖ]	0A22	10	گھ[g]	06AF	ਘ [g]	0A18
5	جھ[ʤ]	062C	ਝ [ʤ]	0A1D	11	ڑھ[ɽ]	0691	ੜ੍ਹ [ɽ]	0A5C+ 0A4D+ 0A39
6	چھ[ʧ]	0686	ਛ [ʧ]	0A1B					

Table 5. Shahmukhi Long Vowels Mapping

Sr.	Vowel	Code	Mapping	Code	Sr.	Vowel	Code	Mapping	Code
1	ا [ə]	0627	ا →ਅ [ə]	0A05	4	و [o]	0648	و→ਵ [v]	0A35
			ا →ਾ [ə]	0A3E				و→ੋ [o]	0A4B
2	آ [ɑ]	0622	آ→ ਆ [ɑ]	0A06				و →ੌ[ɔ]	0A4C
3	ى[i]	0649	ى→ ਈ [i]	0A08				و→ੁ [ʊ]	0A41
			ى→ ਯ [j]	0A2F				و→ੂ [u]	0A42
			ى→ ਿ [ɪ]	0A3F				و→ੋ [o]	0A13
			ى→ ੀ [i]	0A40	5	ے[e]	06D2	ے→ੲ [e]	0A0F
			ى→ ੇ [e]	0A47				ے→ਯ [j]	0A2F
			ى→ ੈ [æ]	0A48				ے→ੇ [e]	0A47
								ے→ੈ [æ]	0A48

Table 6. Shahmukhi Short Vowels Mapping

Sr.	Vowel	Unicode	Name	Gurmukhi	Unicode
1	ِ [ɪ]	0650	Zer	ਿ [ɪ]	0A3F
2	ُ [ʊ]	064F	Pesh	ੁ [ʊ]	0A4B
3	َ [ə]	064E	Zabar	-	-

Table 7. Mapping of other Diacritical Marks or Symbols

Sr.	Shahmukhi		Unicode	Gurmukhi	Unicode
1	Noon ghunna	ں [ɲ]	06BA	ਂ [ɲ]	0A02
2	Hamza	ء [ɪ]	0621	positional dependent	-
3	Sukun	ْ	0652	ੁਨ [un]	0A42, 0A28
4	Shad	ّ	0651	ੱ	0A71
5	Khari Zabar	ٰ [ə]	0670	ਾ [ə]	0A3E
6	do Zabar	ً [ən]	064B	ਨ [n]	0A28
7	do Zer	ٍ [ɪn]	064D	ਿਨ [ɪn]	0A3F, 0A28

4.5 Mapping of Vowels

The long and short vowels of Shahmukhi script have multiple mappings into Gurmukhi script as shown in Table 5 and Table 6 respectively. It is interesting to observe that Shahmukhi long vowel characters Vav و[v] and Ye ی,ے [j] have vowel-vowel multiple mappings as well as one vowel-consonant mapping.

4.6 Mapping other Diacritical Marks or Symbols

Shahmukhi has its own set of numerals that behave exactly as Gurmukhi numerals do with one to one mapping. Table 7 shows the mapping of other symbols and diacritical marks of Shahmukhi.

5 Transliteration System

The transliteration system is virtually divided into two phases. The first phase performs pre-processing and rule-based transliteration tasks and the second phase performs the task of post-processing. In the post-processing phase bi-gram language model has been used.

5.1 Lexical Resources Used

In this research work we have developed and used various lexical resources, which are as follows:

Shahmukhi Corpus: There are very limited resources of electronic information of Shahmukhi. We have created and are using a Shahmukhi corpus of 3.3 million words.

Gurmukhi Corpus: The size of Gurmukhi corpus is about 7 million words. The analysis of Gurmukhi corpus has been used in pre and post-processing phases.

Shahmukhi-Gurmukhi Dictionary: In the pre-processing phase we are using a dictionary having 17,450 words (most frequent) in all. In the corpus analysis of Shahmukhi script we get around 91,060 unique unigrams. Based on the probability of occurrence we have incorporated around 9,000 most frequent words in this dictionary. Every Shahmukhi token in this dictionary structure has been manually checked for its multiple similar forms in Gurmukhi e.g. token اس [əs] has two forms with weights[1] as ਇਸ{59998} [ɪs] (this) and ਉਸ{41763} [ʊs] (that).

Unigram Table: In post-processing tasks we are using around 163,532 unique weighted unigrams of Gurmukhi script to check most frequent (MF) token analysis.

Bi-gram Tables: The bi-gram queue manager has around 188,181 Gurmukhi bi-grams resource to work with.

[1] Weights are unigram probabilities of the tokens in the corpus.

All Forms Generator (AFG): Unigram analysis of Gurmukhi corpus is used to construct AFG Component having 86,484 unique words along with their similar phonetic forms.

5.2 Pre-Processing and Transliteration

In pre-processing stage Shahmukhi token is searched in the Shahmukhi-Gurmukhi dictionary before performing rule-based transliteration. If the token is found, then the dictionary component will return a weighted set of phonetically similar Gurmukhi tokens and those will be passed on to the bi-gram queue manager. The advantage of using dictionary component at pre-processing stage is that it provides more accuracy as well as speeds up the overall process. In case the dictionary lookup fails then the Shahmukhi token will be passed onto basic transliteration component. The Token Converter accepts a Shahmukhi token and transliterates it into Gurmukhi token with the help of Rule Manager Component. Rule Manager Component has character mappings and rule-based prediction to work with. Starting from the beginning, each Shahmukhi token will be parsed into its constituent characters and analyzed for current character mapping along with its positional as well as contextual dependencies with neighboring characters. Shahmukhi script has some characters having multiple mappings in target script (as shown in Table 1 and 5).
Therefore, to overcome this situation extra care has been taken to identify various dependencies of such characters in the source script and prediction rules have been formulated accordingly to substitute right character of target script. Ultimately, a Gurmukhi token is generated in this process and that will be further analyzed in the post- processing activities of transliteration system. Figure 1 shows the architecture of this phase.

5.3 Post-Processing

The first task of this phase is to perform formatting of the Gurmukhi token according to Unicode standards. The second task in this phase is critical and especially designed to enable this system to work smoothly on Shahmukhi script having missing diacritical marks. The input Gurmukhi token has been verified by comparing its probability of occurrence in target script with predefined threshold value. The threshold value is minimum probability of occurrence among most frequent tokens in the Gurmukhi corpus. If the input token has more probability than the threshold value, it indicates that this token is most frequent and acceptable in the target script. Therefore, it is not a candidate for AFG routine and is passed on to the bi-gram queue manager with its weight of occurrence.
On the other hand, a token having probability of occurrence less than or equal to the threshold value becomes a candidate for AFG routine. In AFG routine input Gurmukhi token is examined by All Forms Generator (AFG) with the help of AF manager. AF Manager will generate a phonetic code corresponding to the characters of input Gurmukhi token. This phonetic code will be used by Similar Forms Generator (SFG) routine for producing a list of weighted Gurmukhi tokens with similar phonetic similarities. The suggestion rules will be used to filter out undesired

tokens from the list. This final list of Gurmukhi tokens will then pass on to bi-gram queue manager. The phonetic code generation rules along with suggestion rules play a critical role in the accuracy of this task.

5.4 Bi-gram Queue Manager

The system is designed to work on bi-gram language model in which the bi-gram queue of Gurmukhi tokens is maintained with their respective unigram weights of occurrence. The bi-gram manager will search bi-gram probabilities from bi-gram table for all possible bi-grams and then add the corresponding bi-gram weights. After that it has to identify and mark the best possible bi-gram and pop up the best possible unigram as output. This Gurmukhi token is then returned to the Output Text Generator for final output.

The Output Text Generator has to pack these tokens well with other input text which may include punctuation marks and embedded Roman text. Finally, this will generate a Unicode formatted Gurmukhi text as shown in Figure 2.

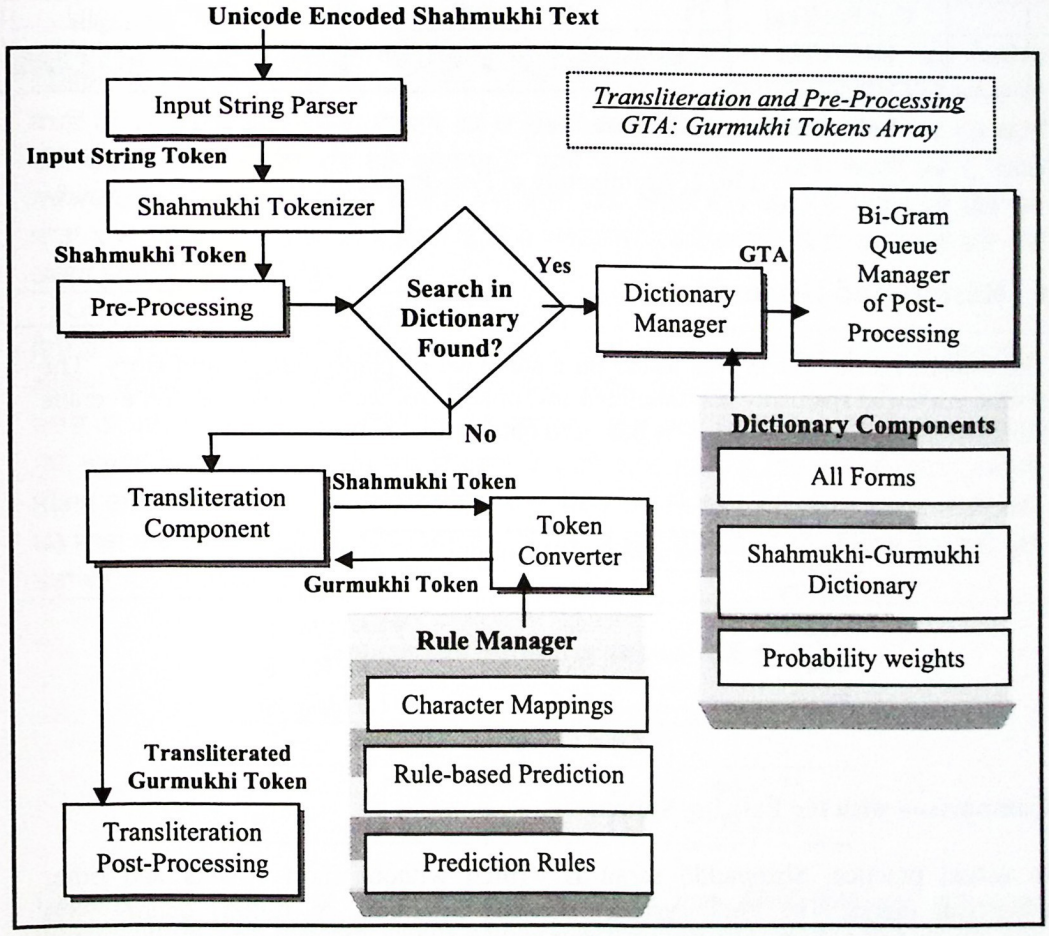

Fig. 1. Architecture of Transliteration and Pre-Processing

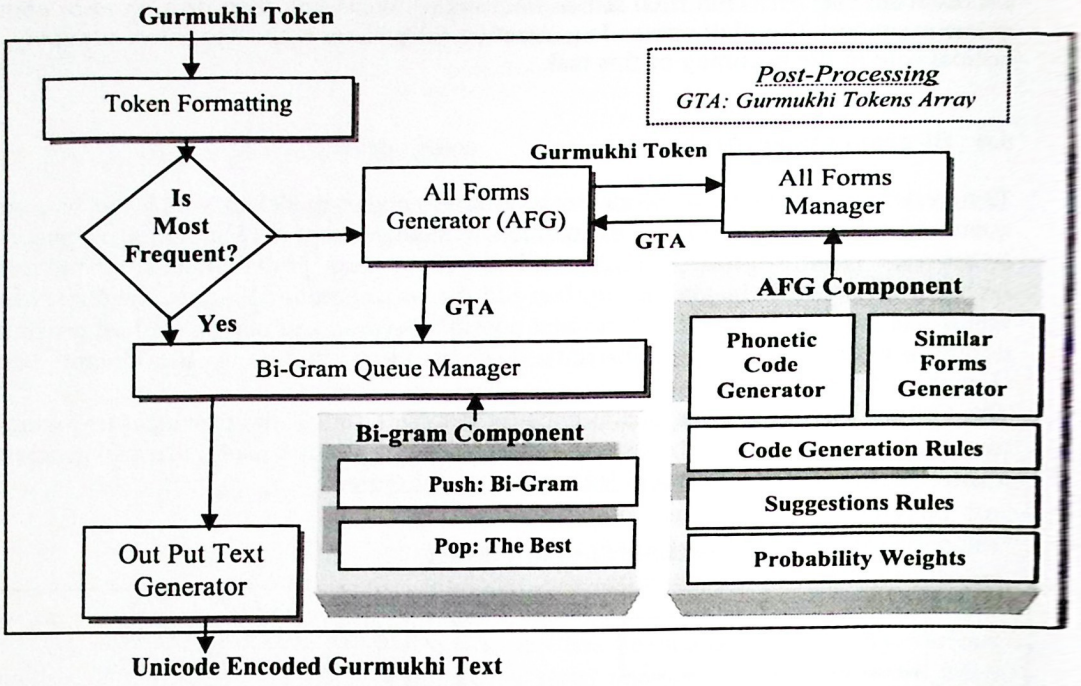

Fig. 2. Architecture of Post-Processing

6 Results and Discussion

The transliteration system was tested on a small set of poetry, article and story. The results reviewed manually are tabulated in Table 8. As we can observe, the average transliteration accuracy of 91.37% has been obtained.

Table 8. Transliteration Results

Type	Transliterated Tokens	Accuracy
Poetry	3,301	90.63769 %
Article	584	92.60274 %
Story	3,981	90.88043 %
Total	7,866	91.37362 %

Comparison with the Existing System

In actual practice, Shahmukhi script is written without short vowels and other diacritical marks. The PMT system discussed by Malik A. (2006) claims 98% accuracy only when the input text has all necessary diacritical marks for removing

ambiguities. But this 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. We have manually evaluated PMT system against the following Shahmukhi input published on a web site and the output text is shown as output-A in table 9.The output of proposed system on the same input is shown as output-B. The wrong transliteration of Gurmukhi tokens is shown in bold and italic and the comparison of both outputs is shown in table 10.

Table 9. Input/Output of PMT and Proposed Systems

Input text (right to left)
اس گل وچ جدوں اسیں بہتے پنجابیاں نوں ویکھدے ہاں تاں پرنسپل تیجا سنگھ دے لیکھ وچ بیانیاں گئیاں کوڑیاں سچائیاں ہور وی شدت نال محسوس ہندیاں ہین۔ اسیں دیس نوں پیار کرن دا دعویٰ کردے ہاں پر اپنے صوبے نوں وساری بیٹھے ہاں۔ اس دا سبہ توں وڈا ثبوت ایہہ ہے کہ بھارت دے لگ بھگ بہتے صوبے اپنے اپنے ستھاپنا دوس بڑے اتشاہ تے جذبے نال مناؤندے ہین۔ اپنی زبان، اپنے سبھیاچار، اپنے پچھوکڑ تے اپنے ورثے تے مان کردے ہین۔ اپنی قومی پچھان تے مان کردے ہین۔ پر ساڈا بابا آدم ہی نرالا ہے۔ سرکاراں توں لے کے عام لوکاں تک پنجابی صوبے دے بنن دن بارے پوری طرحاں اویسلے ہی رہندے ہین۔
Output-A of PMT system (left to right)
ਅਸ ਗਲ ਵਚ ਜਦੋਂ ***ਅਸੋਂ ਬਹਤੇ ਪਨਜਾਬੇਆਂ ਨੋਂ*** ਵੇਖਦੇ ਹਾਂ ਤਾਂ ***ਪਰਨਸਪਲ*** ਤੇਜਾ ***ਸਨਘ*** ਦੇ ਲੇਖ ***ਵਚ ਬੇਅਨੇਆਂ*** ਗਈਆਂ ***ਕੋੜੇਆਂ*** ਸਚਾਈਆਂ ਹੋਰ ਵੀ ***ਸ਼ਦਤ*** ਨਾਲ ***ਮਹਸੋਸ ਹਨਦੇਆਂ ਹੇਨ।*** ***ਅਸੋਂ*** ਦੇਸ ***ਨੋਂ ਪੇਆਰ*** ਕਰਨ ਦਾ ਦਾਵਾ ਕਰਦੇ ਹਾਂ ਪਰ ***ਅਪਨੇ ਸੋਬੇ ਨੋਂ*** ਵਸਾਰੀ ***ਬੇਠੇ*** ਹਾਂ। ***ਅਸ*** ਦਾ ਸਭ ਤੋਂ ***ਵਡਾ ਸਬੋਤ ਇਹਾ ਹੇ ਕਹ*** ਭਾਰਤ ਦੇ ਲਗ ਭਗ ***ਬਹਤੇ ਸੋਬੇ ਅਪਨੇ ਅਪਨੇ*** ਸਥਾਪਨਾ ***ਦੋਸ*** ਬੜੇ ***ਅਤਸ਼ਾਹ*** ਤੇ ਜਜ਼ਬੇ ਨਾਲ ***ਮਨਾਈਵਨਦੇ ਹੇਨ। ਅਪਨੀ*** ਜ਼ਬਾਨ, ***ਅਪਨੇ ਸਭੇਅਚਾਰ, ਅਪਨੇ ਪਛੋਕੜ*** ਤੇ ***ਅਪਨੇ ਵਰਸੇ*** ਤੇ ***ਮਾਨ*** ਕਰਦੇ ***ਹੇਨ।*** ***ਅਪਨੀ ਕੋਮੀ ਪਛਾਨ*** ਤੇ ***ਮਾਨ*** ਕਰਦੇ ***ਹੇਨ।*** ਪਰ ਸਾਡਾ ਬਾਬਾ ਆਦਮ ਹੀ ***ਨਰਾਲਾ ਹੇ।*** ਸਰਕਾਰਾਂ ਤੋਂ ***ਲੇ*** ਕੇ ਆਮ ਲੋਕਾਂ ***ਤਕ ਪਨਜਾਬੀ ਸੋਬੇ*** ਦੇ ***ਬਨਨ ਦਨ*** ਬਾਰੇ ***ਪੋਰੀ ਤਰਹਾਂ ਉੇਅਸਲੇ*** ਹੀ ***ਰਹਨਦੇ ਹੇਨ।***
Output-B of proposed system (left to right)
ਇਸ ਗੱਲ ਵਿਚ ਜਦੋਂ ਅਸੀਂ ਬਹੁਤੇ ਪੰਜਾਬੀਆਂ ਨੂੰ ਵੇਖਦੇ ਹਾਂ ਤਾਂ ਪ੍ਰਿੰਸੀਪਲ ਤੇਜਾ ਸਿੰਘ ਦੇ ਲੇਖ ਵਿਚ ਬਿਆਨੀਆਂ ਗਈਆਂ ਕੌੜੀਆਂ ਸਚਾਈਆਂ ਹੋਰ ਵੀ ਸ਼ਿੱਦਤ ਨਾਲ ਮਹਿਸੂਸ ਹੁੰਦੀਆਂ ਹੈਨ। ਅਸੀਂ ਦੇਸ ਨੂੰ ਪਿਆਰ ਕਰਨ ਦਾ ਦਾਵਾ ਕਰਦੇ ਹਾਂ ਪਰ ਆਪਣੇ ਸੂਬੇ ਨੂੰ ***ਵਸਾਰੀ*** ਬੈਠੇ ਹਾਂ। ਇਸ ਦਾ ਸਭ ਤੋਂ ਵੱਡਾ ਸਬੂਤ ਇਹ ਹੈ ਕਿ ਭਾਰਤ ਦੇ ਲਗ ਭਗ ਬਹੁਤੇ ਸੂਬੇ ਆਪਣੇ ਆਪਣੇ ਸਥਾਪਨਾ ***ਦੋਸ*** ਬੜੇ ਉਤਸ਼ਾਹ ਤੇ ਜਜ਼ਬੇ ਨਾਲ ਮਨਾਉਂਦੇ ਹੈਨ। ਆਪਣੀ ਜ਼ਬਾਨ, ਆਪਣੇ ਸਭਿਆਚਾਰ, ਆਪਣੇ ਪਿਛੋਕੜ ਤੇ ਆਪਣੇ ਵਿਰਸੇ ਤੇ ਮਾਣ ਕਰਦੇ ਹੈਨ। ਆਪਣੀ ਕੌਮੀ ਪਛਾਣ ਤੇ ਮਾਣ ਕਰਦੇ ਹੈਨ। ਪਰ ਸਾਡਾ ਬਾਬਾ ਆਦਮ ਹੀ ਨਿਰਾਲਾ ਹੈ। ਸਰਕਾਰਾਂ ਤੋਂ ਲੈ ਕੇ ਆਮ ਲੋਕਾਂ ਤੱਕ ਪੰਜਾਬੀ ਸੂਬੇ ਦੇ ਬਣਨ ਦਿਨ ਬਾਰੇ ਪੂਰੀ ਤਰ੍ਹਾਂ ਅਵੈਸਲੇ ਹੀ ਰਹਿੰਦੇ ਹੈਨ।

Table 10. Comparison of Output-A & B

Output Type	Transliteration Tokens			Accuracy %
	Total	Wrong	Right	
A	116	64	52	44.8275
B	116	02	114	98.2758

Clearly, our system is more practical in nature than PMT and we got good transliteration with different inputs having missing diacritical marks. But we are still having erroneous transliterations by the system. The main source of error is the

existence of vowel-consonant mapping between the two scripts as already shown in table 5. In some of the cases the bi-gram approach is not sufficient and we need some other contextual analysis technique. In other cases, system makes errors showing deficiency in handling those tokens which do not belong to common vocabulary domain. These observations point to places where the system can be improved and we hope to study them in the near future.

Acknowledgments. This research work is sponsored by *PAN ASIA ICT R&D Grants Programme* for Asia Pacific http://www.apdip.net and the Beta version of this program is available online at http://s2g.advancedcentrepunjabi.org.We would like to thank *Sajid Chaudhry* for providing us data for Shahmukhi corpus.

References

1. Malik, M. G. A.: Punjabi Machine Transliteration. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL (2006) 1137-1144.
2. Afzal, M., Hussain S.: Urdu Computing Standards: Urdu Zabta Takhti (UZT) 1.01. In proceedings of the IEEE INMIC, Lahore (2001).
3. Haizhou, L., Min, Z., and Jian S.: A Joint Source-Channel Model for Machine Transliteration. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (2004) 159-166.
4. Youngim, J., Donghun, L., Aesun, Y., Hyuk-Chul, K.: Transliteration System for Arabic-Numeral Expressions using Decision Tree for Intelligent Korean TTS, Vol. 1. 30th Annual Conference of IEEE (2004) 657-662.
5. Nasreen Abdululjaleel, leah S. Larkey: Statistical Transliteration for English-Arabic Cross Language Information Retrieval. Proceedings of the 12th international conference on information and knowledge management (2003) 139-146.
6. Yan, Q., Gregory, G., David A. Evans: Automatic Transliteration for Japanese-to-English Text Retrieval. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (2003) 353-360.
7. Arbabi, M., Fischthal, S. M., Cheng, V. C., and Bart E.: Algorithms for Arabic Name Transliteration. IBM Journal of research and Development (1994) 183-193.
8. Knight, K., and Graehl, J.: Machine Transliteration. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (1997) 128-135.
9. Stalls, B. G. and Kevin K.: Translating Names and Technical Terms in Arabic Text. COLING ACL Workshop on Computational Approaches to Semitic Languages (1998) 34-41.