A Transliteration based Word Segmentation System for Shahmukhi Script

Gurpreet Singh Lehal and Tejinder Singh Saini
Punjabi University, Patiala 147 002 Punjab, India
{gslehal, tej74i}@gmail.com

Abstract. Word Segmentation is an important prerequisite for almost all Natural Language Processing (NLP) applications. Since word is a fundamental unit of any language, almost every NLP system first needs to segment input text into a sequence of words before further processing. In this paper, Shahmukhi word segmentation has been discussed in detail. The presented word segmentation module is part of Shahmukhi-Gurmukhi transliteration system. Shahmukhi script is usually written without short vowels leading to ambiguity. Therefore, we have designed a novel approach for Shahmukhi word segmentation in which we used target Gurmukhi script lexical resources instead of Shahmukhi resources. We employ a combination of techniques to investigate an effective algorithm by applying syntactical analysis process using Shahmukhi Gurmukhi dictionary, writing system rules and statistical methods based on n-grams models.

Keywords: Shahmukhi, Gurmukhi, Word Segmentation, Transliteration.

1 Introduction

Segmentation of a sentence into words is one of the necessary preprocessing tasks of NLP. Word segmentation can be split into two main processes: word candidate generation and word candidate selection. The first process aims at constructing all possible word candidates from a given input text. While, the latter process aims at choosing the most suitable candidate. For languages like English, French, and Spanish etc. tokenization is considered trivial because the white space or punctuation marks between words is a good approximation of where a word boundary is. Whilst many Asian languages like Urdu, Persian, Arabic, Chinese, Dzongkha, Lao and Thai have no explicit word boundaries [5-7]. Therefore, one must resort to higher levels of information such as: information of morphology, syntax, and statistical analysis to reconstruct the word boundary information [1-4]. In general the problem of segmenting word can be classified into dictionary based and statistical based methods. Statistical methods are considered to be very effective to solve segmentation ambiguities. Durrani [5] and Durrani and Hussain [6] have discussed in detail the various Urdu word segmentation issues. A word segmentation system for handling space insertion problem in Urdu script has been presented by Lehal [9].

In this paper, Shahmukhi word boundary issues have been discussed in detail. The word segmentation module is part of Shahmukhi-Gurmukhi transliteration system and

the novel approach presented in this paper, mainly uses target script lexical resources instead of Shahmukhi resources because Shahmukhi script is usually written without short vowels leading to potential ambiguity. We employ a combination of techniques to investigate an effective algorithm by applying syntactical analysis process using Shahmukhi Gurmukhi dictionary, writing system rules and statistical methods, including n-grams to solve word segmentation.

1.1 Shahmukhi Script

Shahmukhi is a local variant of cursive Urdu script used to record the Punjabi language in Pakistan. It is based on right to left Nastalique style of the Persian and Arabic script. Shahmukhi script has thirty eight letters, including four long vowel signs Alif ا[ə], Vao و[v], Choti-ye ی[j] and Badi-ye ے[j]. Shahmukhi script in general has thirty seven simple consonants and eleven frequently used aspirated consonants. There are three nasal consonants (ں[ŋ], ن[n], م[m]) and one additional nasalization sign, called Noon-ghunna ں [ɲ]. In addition to this, there are three shot vowel signs called Zer ِ[ɪ], Pesh ُ[ʊ] & Zabar َ[ə] and some other diacritical marks or symbols like hamza ء [ɪ], Shad ّ, Khari-Zabar ٰ[ə], do-Zabar ً[ən], do-Zer ٍ[ɪn] etc.

Shahmukhi characters change their shapes depending upon neighboring context. But generally they acquire one of these four shapes, namely isolated, initial, medial and final. 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. Any machine transliteration or text to speech synthesis system has to automatically guess and insert these missing symbols. This is a non-trivial problem and requires an in depth statistical analysis [6]

2 Word Boundary Issues in Shahmukhi text

Shahmukhi is written in cursive Urdu script. The concept of space as a word boundary marker is not present in Urdu script but with the increasing usage of computer it is now being used, both to generate correct shaping and also to separate words [6]. The word boundary identification for Shahmukhi text is not simple. Due to cursive script and irregular use of space, Shahmukhi word segmentation has both space omission and space insertion problems as discussed below. Space insertion refers to insertion of extra spaces in a word, while space omission refers to deletion of spaces between adjacent words.

2.1 Space Insertion problem

There are two basic reasons for space insertion in a Shahmukhi word.

- The space within a word is also used to generate correct shaping while writing Shahmukhi words. Therefore, space is introduced as a tool to control the correct letter shaping and not to consistently separate words. For Example consider a

word ات واد /att vād/ and گنجل دار /guñjhal dār/ having a space to generate the correct shape of ت [t] and ل [l] respectively. Without space both are having visually incorrect forms as اتواد /attvād/ and گنجلدا /guñjhaldār/ respectively. Presence of this type of space in Shahmukhi text leads to space insertion problem in Shahmukhi word which needs to be handled accordingly while processing the Shahmukhi text.

- Many Shahmukhi words which are written as combination of two words are written as single word in Gurmukhi script. So if the two words are as such transliterated to Gurmukhi, they cannot be read properly and in some cases their meaning also gets changed. For example, if the Shahmukhi word ذمےواری /zimmē vārī/ is as such transliterated to Gurmukhi, then it will be read as ਜ਼ਿੰਮੇ ਵਾਰੀ /zimmē vārī/ while it should be written as single word ਜ਼ਿੰਮੇਵਾਰੀ/zimmēvārī/. Thus, the two Shahmukhi words had to be combined before transliteration so that the correct Gurmukhi word is generated. Similarly the city names like حیدر آباد /haidar ābād/, جیکب آباد /jaikab ābād/, جعفر آباد /jāfar ābād/ after transliteration produce unacceptable names in Gurmukhi script as ਹੈਦਰ ਆਬਾਦ /haidar ābād/, ਜੈਕਬ ਆਬਾਦ/jaikab ābād/, ਜਾਫ਼ਰ ਆਬਾਦ/jāfar ābād/. To produce correct transliteration the extra space between the names should be removed to combine them as a single word as ਹੈਦਰਾਬਾਦ /haidrābād/, ਜੈਕਬਾਬਾਦ /jaikbābād/, ਜਾਫ਼ਰਾਬਾਦ /jāfrābād/.

2.2 Space Omission problem

While writing in Urdu/Arabic script a common user finds that it is unnecessary to insert space between the two Urdu words because the correct shape is produced automatically when the first word ends with a non-joiner Urdu character [6]. The same case is observed in Shahmukhi text that many times the user omits word boundary space between the consecutive words where the first word ends with a non-joiner character. This is because the absence of space after non-joiner character has no visible implication and do not affect the readability of the Shahmukhi text. But during computational processing where space is used as a word boundary delimiter, these two or more words are found to be merged together. This gives rise to space omission problem in Shahmukhi text.

Table1. Space Omission Problem with Multiple Merged Words

Word			Merged Words		Romanized			
w	w4	w3	w2	w1	w1	w2	w3	w4
انسپیکٹرمہمدخان		خان	مہمد	انسپیکٹر	iṃspaikṭar	muhmmad	k̲h̲ān	
رشتےدےمقام		مقام	دے	رشتے	rishtē	dē	mukām	
دابےایہدےوچ	وچ	ایہدے	ہے	دا	dā	hai	Ihdē	vic

For example, consider the following Shahmukhi words آ گیا /ā giā/ and ہو سکدا /hō sakdā/ having the first word token ends with a non-joiner character. We can see that they will retain same shape after deleting word boundary space as آگیا /āgiā/ and ہوسکدا /hōsakdā/. Therefore, user can easily skip word boundary space because it does not

affect the readability of the Shahmukhi words. More examples of Shahmukhi words having space omission problem with multiple merged words is shown in table 1.

3 Algorithm for Handling Space Insertion Problem

Rule based techniques like longest matching, maximum matching and statistical methods including n-grams have been extensively used for word segmentation. We employ a combination of both rule based and statistical n-gram techniques for Shahmukhi word segmentation, as proposed by Lehal [9] for Urdu space insertion problem. Based on the idea presented by Lehal [9] we have divided the whole process into two stage architecture as shown in fig.1. In the first stage, writing system rules have been applied to decide if the adjacent Shahmukhi words have to be joined. The rule based analyzer is incorporated based on the knowledge of the writing system specific information for instance some characters such as ں [ɲ] and ً come at the end of a word only, certain characters such as (ؤ, ئ ء ى ً ّ ٰ ں and ۂ), cannot come at the beginning of a word and the presence or absence of *hamza*(ء) before the second vowel gives a indication of joining or not joining of words. Along with these rules there are some typical words in Shahmukhi for example یا /yā/, یاں /yāṃ/ and نہ /nā/ which need special care while processing.

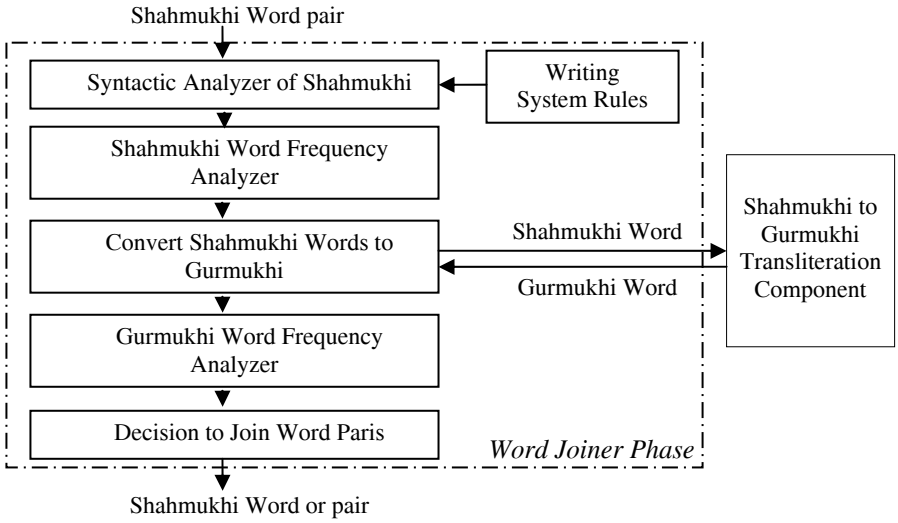

Fig.1. Word Joiner Phase of Transliteration

In case these rules give a definite answer, then we do not move to the second stage. Otherwise, after rule based analyzer the word pairs are analyzed for statistical analysis. In this stage, we have made use of Gurmukhi corpus resources to make the final decision. We use Shahmukhi resources only if the Gurmukhi resources are not sufficient to make a decision for example in case of out-of-vocabulary words (OOV) and unknown cases where the corresponding Gurmukhi transliteration is not present. The algorithm of the statistical analysis is as follows:

Step1: We have to first transliterate the individual (w1, w2) Shahmukhi tokens and their joined form (w1 concatenated with w2) into Gurmukhi say g1, g2 and g3 respectively and then look for the probability of occurrence in Gurmukhi corpus p(g1),p(g2) and p(g3).

Step2: If the probability of occurrence of Joined Gurmukhi form p(g3) is greater than the individual Gurmukhi tokens then the words are joined else not.

Step3: If the joining decision at step2 is to join the word tokens then we additionally look for the existence of the bigram (g1, g2) in Gurmukhi corpus. If the bigram is present, then the two Shahmukhi words are not joined. This is to overcome the situation when the product of probabilities p(g1).p(g2) becomes much more small. As a result many times step2 give the decision to join the words even though they were not to be joined.

Consider the five outputs provided in table 2 to understand the detailed processing of statistical analysis. The system evaluated the unigram probabilities and found that at step 2 the condition is true for all the cases except the first case and the decision is to join them. But at step 3 system found that the last two cases are not joined because the corresponding bigrams (ਚੰਨ/cann/, ਵਲੀ/valī/) and (ਗੁਣ/guṇ/, ਗਾ/gā/) are present in the bigram lexicon.

Table 2.Processing Steps of Statistical Analysis

Input tokens		Transliteration			Unigram Probability		Decision	
w2	w1	g1	g2	g3	p(g3)	p(g1).p(g2)	Step2	Step3
اج	کول	ਕੋਲ	ਅੱਜ	ਕੋਲਾਜ	0.00003919	0.00240909	No	-
شائن	سن	ਸਨ	ਸ਼ਾਇਨ	ਸਨਸ਼ਾਇਨ	0.00001120	0.00000039	Join	Join
سلو	ہن	ਹਨ	ਸਲੂ	ਹੰਸਲੋ	0.00004478	0.00000387	Join	Join
ولی	چن	ਚੰਨ	ਵਲੀ	ਚਨੈਲੀ	0.00003639	0.00000060	Join	No
گا	گن	ਗੁਣ	ਗਾ	ਗੰਗਾ	0.00172694	0.00001642	Join	No

4 Algorithm for Handling Space Omission Problem

We employ a combination of both rule based and statistical n-gram techniques for handling space omission problem. This is a challenging task to predict the correct combination of words from the merged word string. Firstly, Input multi-word has to be broken up into character combinations (CC) as per defined rules. The position of non-joiner characters in the multi-word and the position of ں, ے[e] and ً characters is a good broken point with in a multi-word. Then each adjacent CC's are combined to form a list of the purposed Shahmukhi words. After which, each CC in all the purposed words is transliterated using the transliteration component. Next, we have to design a strategy to select the most probable correct segmentation from the purposed word list. In this stage, the Shahmukhi and Gurmukhi lexical resources are used to make the final decision. For example consider the merged token تیلاتیلااکٹھاکرکے /tīlātīlāikṭ̣hākarkē/ which is broken into تیلا، تیلا، اکٹھا، کر، کے five CCs using the CC

rules. Then each pair of adjacent CC's are combined to form a list of 16 purposed Shahmukhi words. After transliteration and statistical analysis of all the purposed words, the best probable word is selected as an output by the system. To handle over segmentation of out-of-vocabulary (OOV) or unknown words we have imposed the condition that the system will accept only those purposed word combinations which contain at least one character combination of length greater than three or at least one valid bigram character combination exist. For example, consider the Shahmukhi word خانسامیاں /k̲h̲ānsāmīāṃ/ which is out-of-vocabulary and it can be broken down into three valid Gurmukhi CCs ਖਾ/khā/, ਨੱਸਾ/nassā/ and ਮੀਆਂ/mīāṃ/ by this algorithm. Clearly, these CCs qualify the first condition but they do not have existence of valid bigram. Hence, this word will not be broken down by the system due to imposed condition and transliterated into Gurmukhi script as ਖ਼ਾਨਸਾਮੀਆਂ /k̲h̲ānsāmīāṃ/ which is correct transliteration. The system architecture is shown in fig. 2.

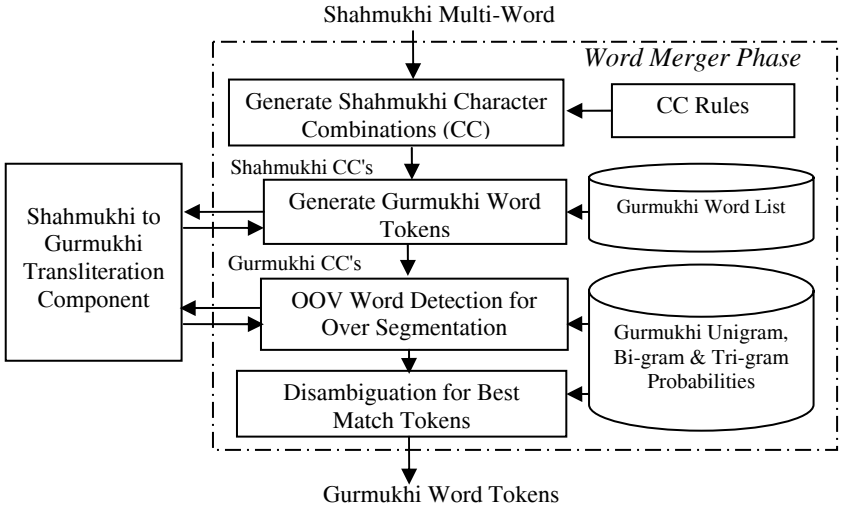

Fig. 2. Word Merger Phase of Transliteration

Experiments and Results

A study of segmentation analysis of Shahmukhi text is conducted on a Shahmukhi corpus of size 3 million words. This corpus is a collection of data like news, articles, short stories, books, novels, poetry etc. collected from Pakistan and downloaded from popular Shahmukhi Unicode website http://www.wichaar.com. It is observed that the Shahmukhi corpus has 1.49% words with space omission and 1.05% of words with space insertion problem. The algorithm for space insertion problem was tested on this corpus and after manual evaluation we found that this algorithm works at 95.23% of accuracy. The system has shown good performance except some over joining cases are also observed. The main cases for consideration and improvement are those

Shahmukhi tokens having no bi-gram in Gurmukhi lexicon as a result they are over joined. This type of situation can be improved by increasing the size of lexicon.

Table 3 shows the observed occurrence of space omission cases which are broken up with respect to number of merged words. It is observed that the maximum number of merged words in a multi-word ligature is five and their occurrence in the corpus is 0.037%. The percentage of occurrence of four merged words is observed to be 0.23% which is also very less in number. After that, relatively high occurrence 3.83% of three merged words is observed. The most frequent space omission cases are two merged words having maximum coverage 96.99% of the corpus.

Table 3. Occurrence of Merged Words in Shahmukhi Corpus

Number of Merged words (n)	Occurrence (%)	Segmentation Accuracy (%)
n=5	0.036778	75
n=4	0.229864	77.5
n=3	3.83413	76.11
n=2	96.99338	93.77

The overall segmentation accuracy of space omission algorithm is 92.97%. The system has shown highest accuracy 93.77% when two merged words are found in the multi-word ligatures. The accuracy of the system decreases when the number of merged word is more that two.

Table 4. Failure Cases of Space Omission Algorithm

SN	Merged words	Error Type	Incorrect Form	Correct Form	Romanized
1	تےفراق	OOV	ਤੇ ਫ਼ਰ ਇਕ تے فر اق	ਤੇ ਫ਼ਿਰਾਕ تے فراق	tē firāk
2	اورکٹ	OOV	ਔਰ ਕੱਟ اور کٹ	ਔਰਕੁਟ اورکٹ	aurkuṭ
3	وینزیلاوچ	OOV	ਵੇਨਜ਼ ਯੁਲਾ ਵਿਚ وینز یلا وچ	ਵੇਂਜੁਏਲਾ ਵਿਚ وینزیلا وچ	vēñjuēlā vic
4	آسٹرولوجی	OOV	ਆਸਟਰ ਵੱਲੋ ਜੀ آسٹر ولو جی	ਆਸਟ੍ਰੋਲੋਜੀ آسٹرولوجی	āsṭraulōjī
5	ناصرخان	Prob.	ਨਾ ਸਿਰ ਖ਼ਾਨ نا صر خان	ਨਾਸਿਰ ਖ਼ਾਨ ناصر خان	nāsir ḵẖān
6	پرتانوالی	Prob.	ਪਰ ਤਾਂ ਵਾਲ਼ੀ پر تاں والی	ਪਰਤਾਂ ਵਾਲ਼ੀ پرتاں والی	partāṃ vāḷī
7	ونڈداربیا	Prob.	ਵੰਡ ਦਾ ਰਿਹਾ ونڈ دا ربیا	ਵੰਡਦਾ ਰਿਹਾ ونڈدا ربیا	vaṇḍdā rihā
8	خدانخواسطہ	Izafat	ਖ਼ੁਦ ਅਣਖਵਾ ਸੱਤਾ	ਖ਼ੁਦਾ-ਨ-ਖ਼ਾਸਤਾ	ḵẖudā-na-ḵẖāstā

9	دورِفاروقی	Izafat	خد انخوا سطہ ਦੋਰ ਫ਼ਾਰੂਕੀ دور فاروقی	خدانخواسطہ ਦੌਰ-ਏ-ਫ਼ਾਰੂਕੀ دورِفاروقی	daur-ē-fārūkī
10	سیدمحمودالحسن	Izafat	ਸੱਯਦ ਮਹਿਮੂਦ ਅਲਹਸਨ سید محمود الحسن	ਸੱਯਦ ਮਹਿਮੂਦ- ਉਲ-ਹਸਨ سید محمودالحسن	sayyad mahimūd-ul-hasan

The analysis of system errors shows that there are three types of errors that the system had made with the current input. As shown in table 4 first type of words are those which are out of vocabulary and system performed over segmentation. The second type of error words are those in which the joined word ligature (unigram) has less probability then the probability of individual word tokens (bi-gram) e.g. the unigram ਪਰਤਾਂ/partāṃ/ has very less probability of occurrence where as the probability of bi-gram ਪਰ/par/ and ਤਾਂ/tāṃ/ is much more. The third type of error words are special unknown Izafat or compound words from Urdu domain which need to be handled. We can produce better results in the future with the scope to increase the size of the training corpus.

References

1. Papageorgiou, C. P.: Japanese Word segmentation by hidden Markov model. In: Proceedings of the workshop on Human Language Technology, pp. 283-288 (1994)
2. Nie, J.Y., Hannan, M.L., Jin, W.: Combining dictionary, Rules and Statistical Information in Segmentation of Chinese. In: Computer Processing of Chinese and Oriental Languages, vol. 9, pp. 125-143 (1995)
3. Wang, X., Fu G., Yeung, D. S., Liu, J. N. K., Luk, R.: Models and Algorithms of Chinese Word Segmentation. In: Proceedings of the International Conference on Artificial Intelligence (IC-AI'2000), pp. 1279-1284. Las Vegas, Nevada, USA (2000)
4. Xu, J., Matusov, E., Zens, R., Ney, H.: Integrated Chinese Word Segmentation in Statistical Machine Translation. In: Proceedings of the International Workshop on Spoken Language Translation, pp. 141–147. Pittsburgh, PA (2005)
5. Durrani, N.: Typology of Word and Automatic Word Segmentation in Urdu Text Corpus. MS Thesis, National University of Computer and Emerging Sciences, Lahore, Pakistan (2007)
6. Durrani, N., Hussain S.: Urdu Word Segmentation. In: The 2010 Annual Conference of the North American Chapter of the ACL, pp. 528-536, Los Angeles, California (2010)
7. Akram, M., Hussain, S.: Word Segmentation for Urdu OCR System. In: Proceedings of the 8th Workshop on Asian Language Resources, pp. 88-94. Beijing, China (2010)
8. Sproat, R., Shi, C., Gale W., Chang N.: A stochastic finite state word segmentation algorithm for Chinese. Computational Linguistics, vol. 22, pp. 377-404. (1996)
9. Lehal G. S.: A Two Stage Word Segmentation System for Handling Space Insertion Problem in Urdu Script. In: World Academy of Science, Engineering and Technology, vol. 60, pp. 321-324. Bangkok, Thailand (2009)
10. Lehal G. S.: A Word Segmentation System for Handling Space Omission Problem in Urdu Script, In: 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP) 23rd COLING, pp. 43-50. Beijing, China (2010)