A Two Stage Word Segmentation System for Handling Space Insertion Problem in Urdu Script

Gurpreet Singh Lehal

Abstract- Hindi and Urdu are variants of the same language, but while Hindi is written in the Devanagari script from left to right, Urdu is written in a script derived from a Persian modification of Arabic script written from right to left. To break the script barrier an Urdu-Devnagri transliteration system has been developed. The transliteration system faced many problems related to word segmentation of Urdu script, as in many cases space is not properly put between Urdu words. Sometimes it is deleted resulting in many Urdu words being jumbled together and many other times extra space is put in word resulting in over segmentation of that word. In this paper, a two-stage system for handling the extra space insertion problem in Urdu has been presented. In the first stage, Urdu grammar rules have been applied, while a statistical based approach has been employed in the second stage. For statistical analysis, lexical resources from both Urdu and Hindi languages, including Urdu and Hindi unigram and bigram probabilities have been used. In addition the Urdu-Devnagri transliteration module is also executed in parallel to help in decision making. The system was tested on 1.84 million word Urdu corpus and the success rate was 98.57%. This is the first time such a system has been developed for Urdu script.

I. INTRODUCTION

Word segmentation is the foremost obligatory task in all NLP application. The initial phase of text analysis for any language processing task usually involves tokenization of the input into words. 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 in various Asian languages, white spaces is rarely or never used to determine the word boundaries, so 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].

Urdu also suffers from word segmentation dilemma, though the problem is not so severe as other South Asian languages, since space is being used to represent the word boundary. But there is no consistency in its usage and a single word might have space in between or alternatively multiple words are written in continuum without any space. The sequence of Urdu words written together without space is still readable because of the character joining property in Urdu. As for example, consider the word cluster انکارکردیابے , which is composed of four words انکار, کر , دیا and ہے. The Urdu readers can very easily segment and read the four words separately, but the computer will read them as a single word since there is no space in between. Similarly, the word cluster پرزوردیاگیاہے is composed of five words(پر, زور, دیا, گیا and ہے), which can be easily read as five separate words by Urdu readers but will be considered as a single word by the computer.

The second issue in Urdu word segmentation is insertion of extra spaces in an Urdu word. The space insertion problem usually occurs for words with derivational affixes(قصور وار), compound words (بول بالا). proper nouns (علی گڑھ) and English words (پلےٹ فارم) [5]. It was also found that many times extra space is inserted during typing as in (کو ئی and ویر وا). The Urdu reader will automatically join the words together and read them perfectly but the computer will treat them as different words. Both the missing space problem and extra space problem in a word, have to be resolved before the text can be used as input for machine translation, machine transliteration, speech synthesis etc.

Hindi and Urdu are variants of the same language characterized by extreme digraphia: Hindi is written in the Devanagari script from left to right, Urdu in a script derived from a Persian modification of Arabic script written from right to left. Hindi and Urdu share grammar, morphology, vocabulary, history, classical literature *etc.* Because of their identical grammar and nearly identical core vocabularies, most linguists do not distinguish between Urdu and Hindi as separate languages. The difference in the two scripts has created a script wedge as majority of Urdu speaking people in Pakistan cannot read Devnagri, and similarly the majority of Hindi speaking people in India cannot comprehend Urdu script. To break this script barrier an Urdu-Devnagri transliteration system has been developed. The transliteration system faced many problems related to word segmentation of Urdu script as discussed above.

In this paper, the extra space problem in Urdu words has been discussed in detail. The word segmentation module is part of Urdu-Hindi transliteration system. Some typical problems encountered during transliteration are also discussed. Interestingly, the solution presented in this paper, mainly uses Hindi lexical resources instead of Urdu resources. To the best of our knowledge, this is the first time the extra space problem in Urdu has been solved and presented.

II. URDU SCRIPT: A BRIEF OVERVIEW

Urdu is a Central Indo-Aryan language of the Indo-Iranian branch, belonging to the Indo-European family of languages. It is the national language of Pakistan. It is also one of the 22 scheduled languages of India and is an official language of five Indian states.

Urdu script has 35 simple consonants, 15 aspirated consonants, one character for nasal sound and 15 diacritical marks. Urdu characters change their shapes depending upon neighboring context. But generally they acquire one of these four shapes, namely isolated, initial, medial and final. Urdu characters can be divided into two groups, non-joiners and joiners. The non-joiners can acquire only isolated and final shape and do not join with the next character. On contrary joiners can acquire all the four shapes and get merged with the following character. A group of joiners and/or non-joiner joined together form a ligature. A word in Urdu is a collection

of one or more ligatures. The isolated form of joiners and non-joiners is shown in figure1-2.

آ ا د ڈ ذ ر ڑ ز ژ و ے

Fig. 1 Non-Joiners in Urdu

ب پ ت ٹ ث ج چ ح خ س ش ص ض ط ظ ع غ ف ق ک گ ل م ن ہ ی ھ

Fig. 2 Joiners in Urdu

Another unique feature of Urdu is that the Urdu words are usually written without short vowels or diacritic symbols. 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 indepth statistical analysis [6-7].

III NEED

As already discussed above, in this paper a solution for deleting the extra space inserted in Urdu words has been presented. The extra space could be because of conventional way or due to typing. It is necessary to delete the extra space before the word can be sent for transliteration to Devnagri for following reasons:

(1) If an Urdu word is broken into two words due to accidentally inserting space during typing, then it will get wrongly transliterated. The word broken into two parts is not a big issue for Urdu readers and he may still read it as a single word, but for the computer it will make a lot of difference, especially from transliteration point of view. As for example consider the words عمر ان and عمران. They both look similar but the first word has an extra space after the first ligature, which is not visible. The two words when transliterated are उम्र इन and इमरान. It is necessary to understand why the transliteration becomes so different. Since in Urdu, the diacritic symbols are not put, so the transliteration system tries to guess and insert the missing diacritic symbols. The system generates all word forms by inserting and all the combinations where the diacritic symbol could be present. The most appropriate word is selected as the word with highest frequency of occurrence in the Hindi word frequency list. Thus in the above example, the word عمر has two equivalent words उमर and उम्र in the Hindi word list and as उम्र has greater frequency of occurrence, so it gets selected. Similarly the second word has five equivalent words (उन, इन, एन, अन्न and अन) in the Hindi word frequency list and the word with highest frequency of occurrence (इन) is selected. Similarly in case of second word, the best match found is इमरान. We can see that insertion of extra space has completely changed the word after transliteration.

(2) Many Urdu words which are written as combination of two words, but they are written as single word in Hindi. So if the two words are as such transliterated to Hindi, then they cannot be read properly and in some cases their meaning also gets changed. As for example, if the Urdu word رشتے داروں is as such transliterated to Hindi, then it will read as रिश्ते दारों while it should be written as रिश्तेदारों. Thus the two Urdu words had to be combined before transliteration so that the correct Hindi word is generated. Similarly the word حیدر آباد has to combine before being sent to the transliteration module since in that case the words will be transliterated as हैदर आबाद , while the word is conventionally written as हैदराबाद, which can be generated only if the two words are combined. In some cases if the two Urdu words are not combined, then their meanings get totally changed after transliteration, as in case of word چکنا چور (completely smashed). If it is transliterated as such, then the output is चिकना चोर (slippery thief). But if the two words are combined and then transliterated, then the correct output चकनाचूर is produced.

IV SYSTEM ARCHITECTURE

The system architecture is shown in Fig. 3. The input is an Urdu word pair and the system makes the decision if the two words need to be joined before being sent for transliteration. We have a two stage architecture. In the first stage, Urdu grammar rules have been applied to decide if the adjacent Urdu words have to be joined. In case these rules give a definite answer, then we stop there only else we move to the second stage. In the second stage a hybrid approach is employed to incorporate Urdu and Devnagri unigram and bigram probabilities to make the decision. The two stages are discussed in detail in the following section.

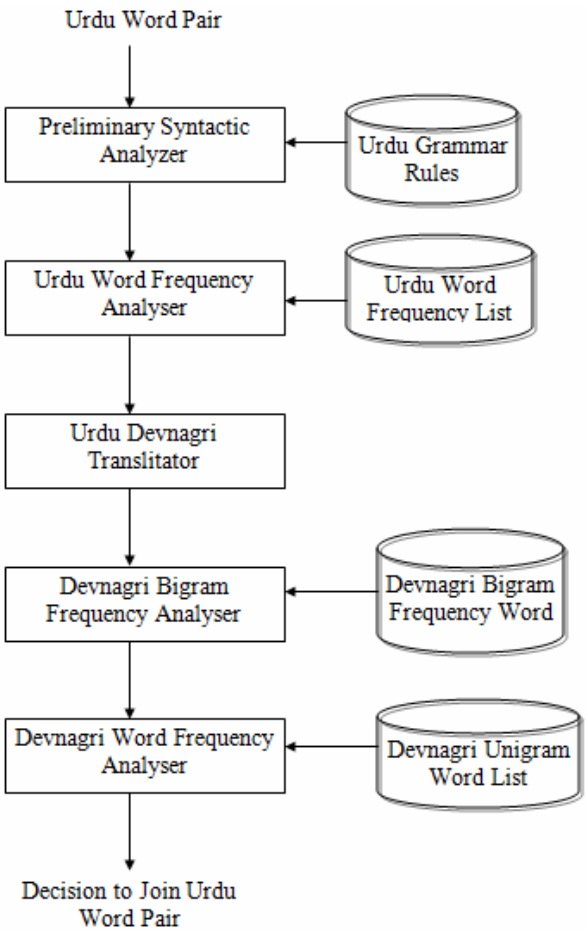

Fig. 3 System Architecture

V RULE BASED DECISION STAGE

In this stage, Urdu grammar rules have been used to generate word joining rules which can be used to decide if two adjacent words (w1 and w2) can be joined. If the rule based analyser is confident that the two words can be joined or not, then there is no need to go for further processing. This saves the next time consuming statistical analysis stage. The following rules have been developed:

A Starting character rule (r1)

There are certain characters such as (ء ئ ؤ ً ّ ٰ ں and ۂ), which cannot come at the beginning of a word. Thus if such character is present in w2, then join w1 and w2, so that the character goes to a middle position in the word. As for example if w1 = ہو and w2 = ئی, then we have to join w1 and w2 to form ہوئی , since w2 starts with a character which does not come in the beginning of a word.

B. Ending Character Rule (R2)

Some characters such as ں and ً come at the end of a word only. When any of these characters is found, we can safely assume it is the end of the word. Thus if w1 ends with any of these characters, then do not join w1 and w2, since in that case the last character will come in middle of the word, which is not allowed.

Thus for example, if w1= ہیں and w2= اور then we cannot join w1 and w2, since w1 ends the ں, which can only come at end of a word and on joining it will come in mid of the word.

C. Second Vowel Rule (R3)

When a word has two vowels, then the second vowel will be preceded with a *hamza*(ء). Hamza is a place holder between two successive vowel sounds. Thus if w1 ends with a vowel and w2 starts with a vowel, then we cannot join w1 and w2, since there is no hamza character between the two.

Thus for example, if w1= نے and w2= آنا then we cannot join since w1 ends with a vowel and w2 starts with a vowel and on combining the two words, we have two consecutive vowel sounds without intermediate hamza character.

D. Single Character Rule (r4)

If w1 or w2 is single character excepting alif madd (آ), and if w1w2 forms a valid Urdu word, then join the two words.

If the rule analyser cannot make any firm joining decision, then the two words are passed to the statistical analysis stage. During experiments, it was found that about 11.7% of the words were filtered out in this stage. Besides being accurate this saves the processing time of the next stage. The remaining 88.3% of the words, whose decision could not be made are sent to the statistical stage.

VI STATISTICAL STAGE

In this stage, the Urdu and Hindi lexical resources are used to make the final decision. The details of the resources are in Table 1.

TABLE 1 : DETAILS OF LEXICAL RESOURCES

Resource	Count
Urdu Word Frequency List	1,21,367 words
Hindi Word Frequency List	1,59,426 words
Hindi Word Bigram List	23,82,511 bigrams

As already discussed above, there are two categories of words, which need to be joined. The first category is of Urdu words containing an extra space due to typing mistake. The second category is the list of Urdu words which are conventionally written as two words, but which is actually a single word only and as such has to be written as a single word in Devnagri.

It was also found that frequently during typing an extra space was typed, which would break the Urdu word into two words. If the space is typed after a joiner character, then the shape of the word changes, as the joiner assumes its final shape, and the typing error can be detected. But if the space is typed after a non-joiner, then the shape of the Urdu word remains same and it is difficult to judge if extra space has been inserted, which has actually broken the word into two smaller words.

As for example, consider the word کر اچی . Visually it looks to be a single word, but internally it is stored as two words (کر and اچی)since an extra space has been inserted between them. Such words need to be joined together, if they have to be properly transliterated. To detect and join such broken words the Urdu frequency list, generated from Urdu corpus, is used. If the product of probability of occurrence of two adjacent Urdu words is lesser than the probability of occurrence of the word formed by joining the two, then the two words are joined together. Thus in the above example,

P(کر) = 0.00234, P(اچی) = 0 and P(کراچی)=0.000051.

Since P(کراچی) > P(کر) *P(اچی) , so the two words are joined.

For the second category of words, the Urdu corpus cannot be used to decide if the words can be joined, as the joined word may not be present in the Urdu corpus, but its equivalent form may be present in Devnagri. As for example, consider Table2. It contains Urdu word pairs, which are present in Urdu corpus, but whose combined form is not present in the corpus. On the other hand, the transliterated version of the combined pair is present in the Hindi corpus, implying that the two Urdu words have to be joined before being sent for transliteration.

TABLE 2 : URDU WORD PAIRS

First word	Second word	Combined
دہشت	گردی	دہشتگردی (not present in Urdu Corpus)
दहशत	गर्दी	दहशतगर्दी (present in Hindi corpus)
کٹھ	پتلی	کٹھپتلی (not present in Urdu Corpus)
कठ	पतली	कठपुतली(present in Hindi corpus)
سچ	مچ	سچمچ (not present in Urdu Corpus)
सच	मच	सचमुच(present in Hindi corpus)

To solve this problem, the individual words have to be first transliterated to Hindi and their probability of occurrence in Hindi corpus is determined.

If the two adjacent Urdu words are represented as u_1 and u_2. Then $u_3 = u_1.u_2$

Transliterate u_1, u_2 and u_3 to Devnagri and let the respective transliterated words be h_1, h_2 and h_3.

Let P(x) be the probability of occurrence of x in the Devnagri corpus.

If $(P(h_3) > P(h_1).P(h_2))$ then the words u_1 and u_2 are joined else not.

Thus if for example, the consider the Urdu word pair < چکنا, چور >discussed earlier:

u_1 = چکنا

u_2 = چور

$u_3=u_1.u_2$ = چکناچور

h_1 = चिकना

h_2 = चोर

h_3 = चकनाचूर

$P(h_1) = 0.00003$
$P(h_2) = 0.0000017$
$P(h_3) = 0.000002$
and $P(h_1).P(h_2) = 0.0000000051$

As $P(h_1).P(h_2) < P(h_3)$, the two Urdu words are joined together resulting in correct Devnagri transliteration.

It was observed that as the value of P(x) is much smaller than 1, so the above expression was biased for h_3, since the product of $P(h_1)$ and $P(h_2)$ becomes too small. As a result many times it gave the decision to join the words even though they were not to be joined.

As for example, consider the Urdu words

u_1 = فن
u_2 = سے
$u_3 = u_1.u_2$ = فنسے

h_1 = फ़न

h_2 = से

h_3 = फंसे

$P(h_1) = 0.0000055$
$P(h_2) = 0.01372$
$P(h_3) = 0.0000094$
and $P(h_1).P(h_2) = 0.00000007547$

Since $P(h_1).P(h_2) < P(h_3)$, the two words are joined together, even though they were not supposed to be joined.

To take care of such cases, an additional lexical resource, bigram list of Hindi words generated from Hindi corpus, is used. First the bigram pair <h_1, h_2> is searched in the Hindi bigram list. If the bigram is present, then the corresponding Urdu words are not joined. Thus in the above example, it was found that the Hindi pair < फ़न ,से > was present in the bigram list and so the corresponding Urdu word pair <فن , سے > was not joined.

VII EXPERIMENTS

The system was tested on an Urdu corpus containing 1,784,995 words. The system detected that around 2.49% (44,576) of Urdu word pairs needed to be joined before being sent for transliteration. The decision was correct in 98.57% of cases. As a sample, in table 3, consider a list of Urdu word pairs, which were joined and also the equivalent Devnagri transliteration, which was greatly improved once the Urdu words were joined. The failure case was when the joined word had a very high probability of occurrence and the bigram pair of the individual words was not present in the bigram list. As an example, the words مین and نے were not to be joined but our system gave the decision that the words had to be joined, since P(मैंने) is much higher than P(मैन).P(ने) and the bigram string <मैन, ने > is not present in the bigram list. The system also fails if a word is broken into more than two words as for example, the wordبے روز گاری ، (बे रोज़ गारी) is split in three words.

TABLE 3 : URDU WORD PAIRS CORRECTLY JOINED

Urdu word pair	Transliteration before joining	Transliteration after joining
ان پٹ	इन पिट	इनपुट
کیلی فورنیا	कीली फ़ोरनिया	कैलीफ़ोर्निया
پر امڈ	पर उमड	पिरामिड
انٹر ویو	इंटर वेव	इंटरव्यू
جان بوجھ	जान बोझ	जानबूझ
امر تسر	अमर तसर	अमृतसर
ہو نگے	हो नग्गे	होंगे
سہو اگ	सहो अग	सहवाग
اتر اکھنڈ	उतर अखंड	उत्तराखंड
شیخ چلی	शेख़ चली	शेख़चिल्ली
جیا پردا	जिया पर्दा	जयाप्रदा

VIII CONCLUSIONS

This paper presents a two-stage system for detecting the extra space problem in Urdu word segmentation. In the first stage, Urdu grammar rules have been applied to decide if the adjacent Urdu words have to be joined. In case these rules give a definite answer, then the system stops at that stage only. A hybrid approach is employed in the second stage to incorporate Urdu and Devnagri unigram and bigram probabilities to make the decision. The system was tested on 1.84 million word Urdu corpus and the success rate was 98.57%. This is the first time, that such type of system has been developed for the solving the extra space problem in Urdu.

IX ACKNOWLEDGEMENTS

The author will like to acknowledge the support provided by ISIF grants for carrying out this research.

REFERENCES

[1] Constantine P. Papageorgiou. "Japanese Word segmentation by hidden Markov model". *In Proc. of the HLT Workshop*, pages 283–288. (1994)

[2] Nie, J.Y., Hannan, M.L. & Jin, W. Combining dictionary, "Rules and Statistical Information in Segmentation of Chinese". *Computer Processing of Chinese and Oriental Languages*, Vol. 9, No. 2, pp. 125-143. (1995)

[3] Wang, Xiaolong, Fu Guohong, Danial S.Yeung, James N.K.Liu, and Robert Luk. 2000. "Models and algorithms of Chinese word segmentation". *Proceedings of the International Conference on Artificial Intelligence (IC-AI'2000)*, Las Vegas, Nevada, USA, 1279-1284. (2000)

[4] Jia Xu, Evgeny Matusov, Richard Zens, and Hermann Ney. "Integrated Chinese Word Segmentation in Statistical Machine Translation". *Proceedings of the International Workshop on Spoken Language Translation*, pages 141–147, Pittsburgh, PA. (2005)

[5] N. Durrani. "Typology of Word and Automatic Word Segmentation in Urdu Text Corpus". National University of Computer and Emerging Sciences, Lahore, Pakistan. 2007.

[6] Bushra Jawaid, and Tafseer Ahmed, 'Hindi to Urdu Conversion: Beyond Simple Transliteration". *Proceedings of the Conference on Language & Technology, Lahore*, pp.24-31 (2009).

[7] Malik, M G Abbas; Besacier, Laurent; Boitet, Christian; Bhattacharyya, Pushpak. "A Hybrid Model for Urdu Hindi Transliteration." *ACL/IJCNLP Workshop on Named Entity (NEW-09), Joint conference of the 47th Annual Meeting of the Association of Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of NLP,* August 2 - 7, 2009, Singapore. (2009)