A Word Segmentation System for Handling Space Omission Problem in Urdu Script

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

Abstract

Word Segmentation is the foremost obligatory task in almost all the NLP applications, where the initial phase requires tokenization of input into words. Like other Asian languages such as Chinese, Thai and Myanmar, Urdu also faces word segmentation challenges. Though the Urdu word segmentation problem is not as severe as the other Asian language, since space is used for word delimitation, but the space is not consistently used, which gives rise to both space omission and space insertion errors in Urdu. In this paper we present a word segmentation system for handling space omission problem in Urdu script with application to Urdu-Devnagri Transliteration system. Instead of using manually segmented monolingual corpora to train segmenters, we make use of bilingual corpora and statistical word disambiguation techniques. Though our approach is adapted for the specific transliteration task at hand by taking the corresponding target (Hindi) language into account, the techniques suggested can be adapted to independently solve the space omission Urdu word segmentation problems. The two major components of our system are : identification of merged words for segmentation and proper segmentation of the merged words. The system was tested on 1.61 million word Urdu test data. The *recall* and *precision* for the merged word recognition component were found to be 99.29% and 99.38% respectively. The words are correctly segmented with 99.15% accuracy.

1 Introduction

Word segmentation is the foremost obligatory task in all NLP application, where the initial phase requires tokenization of 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 such as Chinese, Thai and Myanmar, 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 (Papageorgiou, 1994; Nie et al, 1995; Wang et al, 2000; Xu et al, 2005).

Though the Urdu word segmentation problem is not as severe as some of the other Asian language, since space is used for word delimitation, but the space is not consistently used, which gives rise to both space omission and space insertion errors in Urdu. Durrani(2007) and Durrani and Hussain(2010) have discussed in detail the various Urdu word segmentation issues while Jawaid and Ahmed(2009) and Abbas et al(2009) have discussed the Hindi-Urdu transliteration issues. A word segmentation system for handling space insertion problem in Urdu script has been presented by Lehal(2009).

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,

Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), pages 43–50,
the 23rd International Conference on Computational Linguistics (COLING), Beijing, August 2010

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 we present a word segmentation system for handling space omission problem in Urdu script with application to Urdu-Devnagri Transliteration system. Instead of using manually segmented monolingual corpora to train segmenters, we make use of bilingual corpora and statistical word disambiguation techniques. Though our approach is adapted for the specific transliteration task at hand by taking the corresponding target (Hindi) language into account, the techniques suggested can be adapted to independently solve the space omission Urdu word segmentation problems.

2 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 figures 1-2.

آ ا د ڈ ذ ر ڑ ز ژ و ے

Figure 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 ہ

Figure 2. Joiners in Urdu

The space character is used in Urdu both to generate correct shaping and also to separate words. Though for words ending with non-joiners correct shaping is generated even when space is not typed and thus, many times a user omits the 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 انکارکردیاہے , 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.

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 in-depth statistical analysis.

An Urdu word is a combination of ligatures (characters which join together) and isolated characters. For example انکار is composed of isolated characters ا and ر and ligature نکا . A ligature or isolated character will be called as Urdu character cluster (UCC) in this paper. A Urdu word is thus a combination of UCCs . As for example, the word انکار is composed of three UCCs ا , نکا and ر . We borrow the term, Orthographic Word used by Durrani and Hussain(2010) to define our segmentation process. An Orthographic Word (OW) is a combination of UCCs separated by spaces or punctuation marks. An OW may contain single or multiple Urdu words. Our task is to identify if an OW contains multiple words and in that case properly segment the words.

As for example, consider the sentence:

میزبان ٹیم کی جانب سے رام نریش نے ہیروکاکرداراداکیا

It contains nine OWs

1. میزبان
2. ٹیم
3. کی
4. جانب
5. سے
6. رام
7. نریش
8. نے
9. ہیروکاکرداراداکیا

The first eight OWs contain single Urdu words, while the last OW contains 5 Urdu words(ہیرو, کا, کردار, ادا and کیا)

3 Segmentation Model for Urdu

There are three major issues in the automatic Urdu word segmentation. The first problem is to decide if the orthographic word represents a single word or a multiple word cluster. The second is the ambiguity issue. Since a word cluster can be segmented into words in multiple ways, the correct word boundary detection becomes a challenge. As for example the OW گیاناسے can be segmented as گیان + اسے or گیا + نا + سے. The third problem is the segmentation of unknown word. Unknown word refers to word that does not exist in the dictionary or corpus. Unknown words can be categorized into the different types such as error words, abbreviation, proper names, derived words, foreign words, compounds etc. The unknown word causes segmentation error since the word does not exist in the dictionary, it could be incorrectly segmented into shorter words. For example, the word, ڈرمیٹالوجی , which is a foreign word, gets segmented into four words (ڈر , میٹا , لو and جی) after dictionary look-up as the word ڈرمیٹالوجی is not present in the corpus.

The input is an Urdu Orthographic Word and the system first makes the decision if the OW contains single or multiple Urdu words. In case the OW contains multiple words, the individual Urdu words are extracted from the OW. These different stages are discussed in detail in following sections. As can be seen from the figure, at each stage we make use of lexical resources both from Urdu and Hindi languages. The details of the resources used are in Table 1.

The system architecture is shown in Fig. 3.

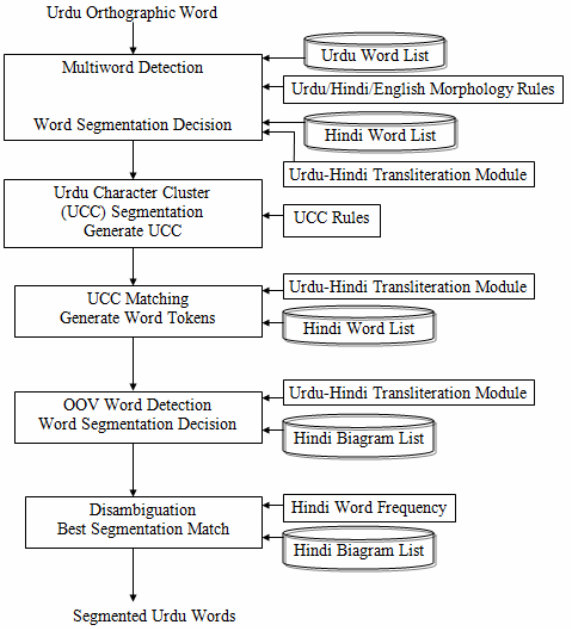

Figure 3. System Architecture

Table 1. Lexical resources used in system

Resource	**Count**
Urdu Word Frequency List	121,367 words
Hindi Word Frequency List	159,426 words
Hindi Word Bigram List	2,382,511 bigrams

4 Decision Stage

In the decision stage, the system decides if the OW contains single or multiple Urdu words. It could so happen that the OW contains single word only and we may break up into smaller words. The decision is based on Urdu and Hindi, word frequency lists analysis as well as Urdu/English/Hindi Morphological rules. To decide if the word cluster is containing multiple words, we first search for OW in the Urdu word list. If it is found then it means that the OW is a valid Urdu word and does not need any further segmentation and quit over there.

It could happen that the OW could be an inflection, whose root form maybe present in the

Urdu word list. Even though the Urdu word list contains inflected forms, but for many words all the inflections may not be present. This problem is more pronounced for English terms, which have become part of Urdu language. For such words, the inflections could follow both rules of English and Urdu. For example plural of یونیورسٹی (university) could be both *universitiyon* وںیورسٹیوںی as well as *universities* یونیورسٹیز. The first form follows the Urdu infection rules while the second form follows the English inflection rules. Similarly we found both the Urdu and English inflections for the English word secretary in Urdu text (سیکرٹریوں and سیکرٹریز) . Thus if the OW is not found in the Urdu word list, we use both Urdu and English morphological rules to generate its root form and search for the root form in the Urdu word list. If the root form is found, we assume the word to be a valid Urdu word and quit there.

It is widely reported in word segmentation papers, that the greatest barrier to accurate word Segmentation is in recognizing words that are not in the lexicon of the segmenter. Thus if a word or its root form is not present in the Urdu word list it will be wrongly presumed to be a multi word cluster. To alleviate this problem, the Urdu corpus has been supplemented with Hindi corpus, which has helped in increasing the word segmentation as well as multi-word recognition accuracy. It was found many times that the Urdu word may be a proper noun, foreign word or some valid out of vocabulary word, which is not present in Urdu corpus but present in the Hindi word list. Another advantage of checking in the Hindi corpus is that many of the Hindi words, which are written as single word are usually written as two words in Urdu. For example, کرےگا (करेगा), کھیلتے (खेलते), ایمانداری (ईमानदारी), چئیرمین (चेयरमैन) etc. These Urdu words are many times written as a single word and in that case if passed to Hindi word list would still report as correct. For checking the OW in Hindi word list, we first transliterate it to Hindi and then search for it in the Hindi wordlist. If the transliterated word is found, then the OW is not considered for segmentation. Like Urdu, it may also happen that the root word of OW may be present in the Hindi word list. So like Urdu, we use both Urdu and English morphological rules to generate its root form and search for the root form in the Hindi word list. If the root form is found, we assume the word to be a valid Urdu word and quit there. If the OW passes all the above stages, then it is considered a candidate for segmentation.

The steps in brief are :

- Search for OW in Urdu List. If OW is present in the list then quit. example : مطابق
- Determine the root form of OW using Urdu Morphological rules and search for the root form in Urdu List. If found then quit. example : سیکرٹریوں
- Determine the root form of OW using English Morphological rules and search for the root form in Urdu List. If found then quit. example : ٹورنامنٹس
- Let HW = Transliteration of OW in Hindi. Search for HW in the Hindi Word List. If HW is present in the list then quit. example : ایمانداری
- Determine the root form of HW using Hindi Morphological rules and search for the root form in the Hindi List. If found then quit. example : چیئرمینوں
- Determine the root form of HW using English Morphological rules and search for the root form in the Hindi List. If found then quit. example : ہولڈرز
- Go to the segmentation stage. example : تھااس

5 Segmenting the Orthographic Word

The Urdu orthographic word is next broken into Urdu Character Combinations (UCC) using Urdu orthographic rules. Unlike word segmentation that is a difficult task, segmenting a text into UCCs is easily achieved by applying the set of rules. These adjacent UCCs are then combined to form a sequence of Urdu words. We need to list all possible segmentations and design a strategy to select the most probable correct segmentation from them.

As for example, consider the OW توجواب: It is segmented into four UCCs : تو, جو. ا and ب . The adjacent clusters can be combined to form 6 word segmentations:

- جواب + تو
- اب + توجو

- ب + توجوا
- اب + جو + تو
- ب + ا + توجو
- ب + ا + جو + تو

5.1 Longest Matching

The method scans an input sentence from left to right, and select the longest match with a dictionary entry at each point. In case that the selected match cannot lead the algorithm to find the rest of the words in the sentence, the algorithm will backtrack to find the next longest one and continue finding the rest and so on. This algorithm fails to find the correct segmentation in many cases because of its greedy characteristic.

5.2 Maximum Matching

This method first generates all possible segmentations for a sentence and then selects the one that contain the fewest words, which can be done efficiently by using dynamic programming technique. When the alternatives have the same number of words, the algorithm cannot determine the best candidate and some other heuristics have to be applied.

We tried both longest matching and maximum matching and the smallest unit taken for combining is UCC. But we found shortcomings in both the matchings. For example the OW کرارہاہے gets segmented as کرار +ہا +ہے using longest matching, while it should be کرا+رہا+ہے . Similarly the OW بروزاتوارکودن gets segmented as بروز+اتوار+کودن using maximum matching while it should be بروز+اتوار+کو+دن.

Thus we see that both longest string match and smallest words fail sometimes. If these algorithms are supplemented by statistical information such as frequency analysis and n-grams then these failures can be avoided. So in our present work, we apply maximal matching algorithm along with these statistics. Initially we used unigram frequency of occurrence for deciding the best word combination. Each Urdu word in the combination is formed by joining adjacent UCCs. In each of the combination, we first convert each of the Urdu word to Hindi. The combination with highest combined product of the unigram frequency of occurrences is finally selected. Thus in the above example, the OW توجواب: will be segmented as تو + جواب, as shown in Table 2.

Table 2. Product of Frequency of Occurrence

Urdu Combination	Hindi Combination (Frequency of occurrence)	Frequency Product
تو جواب	तो (0.005161) जवाब (0.00026)	1.34221E-06
توجو اب	तोजो (4.16E-07) अब (0.001623)	6.75557E-10
توجوا ب	तोजवा (0) ब (4.48E-05)	0
تو جو اب	तो (0.005161) जो (0.002602) अब (0.001623)	2.18028E-08
توجو ا ب	तोजो (4.16E-07) अ (3.6E-05) ब (4.48E-05)	6.69866E-16
تو جو ا ب	तो (0.005161) जो (0.002602) अ (3.6E-05) ब (4.48E-05)	2.16191E-14

It is interesting to see that for segmentation of Urdu words, we used Hindi language statistical analysis instead of Urdu language statistical analysis. Since the current system is part of

Urdu-Hindi transliteration system, we prefer the output to be segmented according to Hindi rules. There are many words which are otherwise joined in Hindi but written as separate words in Urdu. So if we use the Urdu language modeling for segmentation, the word gets broken. Some of the examples are:
اغواکار is written as combination of two words اغوا+ کار in Urdu but its equivalent Hindi word अग़वाकार is written as a single word. Similarly, in Hindi text the verbs are concatenated with the future auxiliaries "gaa", "gii" and "ge", while they are written separately in Urdu. Thus کرین +گے are written separately, but their equivalent Hindi form करेंगे is written as single word. So the advantage of using Hindi training data is that the words get segmented according to the desired Hindi rules. Another problem with Urdu training data was that the Urdu training itself contains merged words. So the words had to be manually separated, though fortunately the Urdu corpus compiled by CRULP (www.crulp.org) has been quite clean, but many words were missing particularly English ones. Another problem is that the words are broken even in the cleaned Urdu corpus. On the other hand when we used the Hindi training data for word segmentation, the problems of merged or broken words in the training text were not encountered. Also the Hindi corpus compiled by us had much larger vocabulary coverage, while the Urdu corpus we used for training purpose had many common words such as اوباما, خطرے , گاندھی, جیکسن etc. missing. Thus the word segmentation algorithm which used the Hindi training set had much better segmentation accuracy as compared to the Urdu training set.

We observed that though the above scheme worked fine in majority of the cases, but in a few cases it failed to segment properly as it did not take care of the context or adjacent words. As for example consider the OW : مردیاعورت. It contains six CCs: مر, د, یا, عو, ر and ت. The word combination selected by above methodology is :

مر + دیا + عورت , though the correct combination is مرد + یا + عورت. It was observed that as we did not take care about adjacent words, thus wrong combination was selected. If the bigram information is added, then such problems were reduced.

We thus use both unigram and bigram frequency analysis for deciding the best word combination. Each Urdu word in the combination is formed by joining adjacent UCCs. In each of the combination, we first convert each of the Urdu word to Hindi. Next we find the unigram and bigram frequency of occurrence of each Hindi word and Hindi word pair in the combination. The bigram frequencies are normalized to avoid multiplication by zero. The combination with highest combined product of the unigram and bigram frequencies of occurrences is finally selected. Using this methodology we were able to generate the sequence combination is مرد + یا + عورت in above example.

As we are using Hindi training data, it was observed that sometimes we had merged words which did not had equivalent transliterated words in our Hindi frequency list. As example, the OW ترازابلیس had to be segmented as تراز + ابلیس, but the equivalent transliterated Hindi terms of تراز and ابلیس, were not found in the Hindi frequency list. As a result, the OW is not segmented. To take care of such situations, if we cannot segment using the Hindi frequency list, our system then goes for maximal matching using the Urdu training data. Thus in above example, after search fails in Hindi training set, the system searches for the minimum word combination and on finding the above two words in the Urdu training set segments the OW into these words.

6 Over Segmentation

For wrongly spelled or OOV (out of vocabulary) Urdu words, the system may forcibly break the word into smaller words. As for example, our system forcibly broke the OW گردوہر into گر + دو +ہر . This problem proved difficult to tackle, though we were able to partially solve it. It was found that usually the OOV words were broken into small unrelated words. So we put the condition on the system to accept only those word segments which contained at least one word of length greater than three or at least one bigram pair was present in the Hindi bigram list. The presence of at least one bigram pair ensured that all the words were not unrelated. Thus in the

above example, the OW gets split into three words, all of length two. These words when transliterated to Hindi get converted to गिर + दो + हर. On searching the bigram list, it was found that neither of the bigram pair < गिर, दो > and < दो , हर > was present and thus this word segmentation was rejected.

7 Experiments

We tested our system on a test data of 1,613,991 Urdu words. In the decision stage, it was found that 116,078 words, which make 7.19% of original text were not found in the Urdu corpus and were considered candidates for segmentation. After morphological analysis of these words, 2851 Urdu words were found to be valid Uru words and were removed from the segmentation candidate list. After converting the remaining Urdu words to Hindi and checking them in Hindi corpus, only 35,226 words were left which were not present in Hindi corpus. Therefore from original 16,13,991 only 35,226 (2.19%) were passed onto segmentation stage for checking for merged words.

In the segmentation stage it was found that out of 35,226 words, 24,001 words (68.13%) had merged words. The number of merged words varied from 2 to 6. Table 3 show the frequency of number of merged words found in word clusters. As can be seen from the table 96.71% of merged word clusters had two merged words.

Table 3. Frequency of Merged Words

Number of merged words	Frequency Percentage
2	96.71%
3	2.99%
4	0.25%
5	0.037%
6	0.004%

The *recall* and *precision* for the decision stage, which decides if the OW needs to be segmented, were found to be 99.29% and 99.38% respectively.

The word segmentation algorithm was able to correctly segment the words with 99.15% accuracy.

8 Conclusions

In this paper, we have presented a system for solving the space omission problem in Urdu text. This system is part of the larger system designed for transliteration of Urdu text to Hindi. We have combined statistical language modeling of both Urdu and Hindi languages in development of the system. We have presented a new scheme of using Hindi for segmenting Urdu text after transliteration, because Hindi uses spaces consistently versus Urdu which has both space omission and insertion problems. This is the first time such a segmentation scheme for handling Urdu space omission problem has been presented. The word segmentation algorithm was able to correctly segment the words with 99.15% accuracy.

Acknowledgements

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

References

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

Durrani N. and Hussain Sarmad. 2010. Urdu Word Segmentation.http://www.crulp.org/Publication/papers/2010/Urdu_Word_Segmentation_NAACL.pdf (accessed on 5th July 2010).

Jawaid Bushra and Ahmed Tafseer. 2009. Hindi to Urdu Conversion: Beyond Simple Transliteration. *Proceedings of the Conference on Language & Technology*, Lahore,.Pakistan, 24-31.

Lehal G. S. 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, 60: 321-324.

Malik Abbas, Besacier Laurent, Boitet Christian and Bhattacharyya Pushpak. 2009. A hybrid Model for Urdu Hindi Transliteration. *Proceedings of the*

2009 Named Entities Workshop, ACL-IJCNLP 2009, Singapore, 177-185.

Nie, J.Y., Hannan, M.L. & Jin, W. 1995. Combining dictionary, rules and statistical information in segmentation of Chinese. *Computer Processing of Chinese and Oriental Languages*, 9(2): 125-143.

Papageorgiou Constantine P. 1994. Japanese word segmentation by hidden Markov model. *Proc. of the HLT Workshop*, 283–288.

Wang Xiaolong, , Fu Guohong, Yeung Danial S., Liu James N.K., and Luk Robert. 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.

Xu Jia, Matusov Evgeny, Zens Richard, and Ney. 2005. Hermann.Integrated Chinese word segmentation in statistical machine translation. *Proceedings of the International Workshop on Spoken Language Translation*, Pittsburgh, PA, 141-147.