Urdu Word Segmentation

Nadir Durrani
Institute for NLP
Universität Stuttgart
durrani@ims.uni-stuttgart.de

Sarmad Hussain
Center for Research in Urdu Language Processing
National University of Computer and Emerging Sciences
sarmad.hussain@nu.edu.pk

Abstract

Word Segmentation is the foremost obligatory task in almost all the NLP applications where the initial phase requires tokenization of input into words. Urdu is amongst the Asian languages that face word segmentation challenge. However, unlike other Asian languages, word segmentation in Urdu not only has space omission errors but also space insertion errors. This paper discusses how orthographic and linguistic features in Urdu trigger these two problems. It also discusses the work that has been done to tokenize input text. We employ a hybrid solution that performs an n-gram ranking on top of rule based maximum matching heuristic. Our best technique gives an error detection of 85.8% and overall accuracy of 95.8%. Further issues and possible future directions are also discussed.

1 Introduction

All language processing applications require input text to be tokenized into words for further processing. Languages like English normally use white spaces or punctuation marks to identify word boundaries, though with some complications, e.g. the word "e.g." uses a period in between and thus the period does not indicate a word boundary. However, many Asian languages like Thai, Khmer, Lao and Dzongkha do not have word boundaries and thus do not use white space to consistently mark word endings. This makes the process of tokenization of input into words for such languages very challenging.

Urdu is spoken by more than 100 million people, mostly in Pakistan and India[1]. It is an Indo-Aryan language, written using Arabic script from right to left, and Nastalique writing style (Hussain, 2003). Nastalique is a cursive writing system, which also does not have a concept of space. Thus, though space is used in typing the language, it serves other purposes, as discussed later in this paper. This entails that space cannot be used as a reliable delimiter for words. Therefore, Urdu shares the word segmentation challenge for language processing, like other Asian languages.

This paper explains the problem of word segmentation in Urdu. It gives details of work done to investigate linguistic typology of words and motivation of using space in Urdu. The paper then presents an algorithm developed to automatically process the input to produce consistent word segmentation, and finally discusses the results and future directions.

2 Urdu Writing System

Urdu is written in cursive Arabic script. Characters in general join with the neighbors within a word and in doing so acquire different shapes. Logically, a character can acquire up to four shapes, i.e. initial, medial, final position in a connected sequence or an isolated form. The characters having this four-way shaping are known as *joiners*. However, another set of characters only join with characters before them but do not join with character after them, and are termed as *non-joiners*. The non-joiners only have final and isolated forms. For example Arabic Letter Farsi Yeh ی is a joiner and has four shapes ﻳ, ﻴ, ﻰ and ﻯ respectively and Arabic letter Dal د is a non-joiner and has two forms د and ﺪ only. The shape that these characters acquire depends upon the context.

Table 1 lists the orthographic rules that Urdu characters follow. For example, the table shows that in the middle of a word, if the character is a non-joiner, it acquires final shape when following a

[1] Ethnologue.com
http://www.ethnologue.com/14/show_language.asp?code=URD

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 528–536,
Los Angeles, California, June 2010.

joiner and isolated shape when following a non-joiner. This joining behavior results in formation of multiple connected portions within a word, each called a *ligature*.

Word	J-Shape	Example	NJ-Shape	Example
Start	I	مسجد	I_s	دجال
Middle	M after J	نمره	F after J	بندر
	I after NJ	دمبا	I_s after J	نادر
End	F after J	عجم	F after J	بند
	I_s after NJ	کام	I_s after NJ	رد
J = Joiners, NJ = Non-Joiners I = Initial, I_s = Isolated, M = Medial, F = Final Underlined = Shape in Consideration				

Table 1: Orthographic Rules for Urdu

The concept of space as a word boundary marker is not present in Urdu writing. As an Urdu learner, a person is not taught to leave a space between words, but only to generate correct shaping while writing. Thus, the concept of space is only learnt later on when the person learns how to use a computer. However, space is introduced as a tool to control the correct letter shaping and not to consistently separate words. For example, the native speaker learns to insert a space within the word ضرورت مند ("needy") to generate the correct shape of ت. Without space it appears as ضرورتمند which is visually incorrect. On contrary, the user finds it unnecessary to insert a space between the two words اردومرکز ("Urdu Center"), because the correct shaping is produced automatically as the first word ends with a non-joiner. Therefore اردومرکز and اردو مرکز look identical.

Though space character is not present in Urdu, with increasing usage of computer it is now being used, both to generate correct shaping (as discussed above) and also to separate words (a habit being carried over to Urdu from English literate computer users). This makes space an unreliable cue for word boundary. The problem is further obfuscated by the lack of a clear definition of a work in Urdu in some contexts. The next section discusses these issues.

3 Segmentation Issues in Urdu Text

The segmentation challenges can be divided into two categories, challenges caused due to joiner and non-joiner characters.

3.1 Space Omission

As discussed, 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. Though there is no visible implication, from the perspective of computational processing not typing a space merges current word with the next word. Figure 1 below illustrates an example, where the phrase has eight words (or tokens) each ending with a non-joiner and thus the whole phrase can be written without a space and is still visibly same and equally readable.

قافلے کے صدر احمد شیر ڈوگر نے کہا
(a)
قافلےکےصدراحمدشیرڈوگرنےکہا
(b)

Figure 1: All Words Ending with Non-Joiners (a) with Spaces, (b) without Spaces between Words ("Troop Leader Ahmed Sher Dogar Said")

Another frequent set of space omissions are caused due to variation in the definition of a word in Urdu. There are certain function words in Urdu which may be combined with other function words and content words by some writers but may be written separately by others. Shape variation may also occur in some of these cases, but is overlooked by the writers. Table 2 gives some examples of such cases. Though the merged form is not considered correct diction, it is still frequently used and thus has to be handled. It is not considered spelling error but a writing variation.

POS	Combined	Separated	Translation
P_n+CM	آپکا	آپ کا	Yours
D+ NN	اسوقت	اس وقت	at that time
CM+ NN	کیطرف	کی طرف	Towards
V+TA	کریگی	کرے گی	will do
CM + P	کیلیے	کے لیے	For
P_n = Pronoun, D = Demonstrative, NN = Noun, CM = Case Marker, V=Verb, P = Particle			

Table 2: Multiple Words Written in Connected Form Causing Shaping Changes

Due to reasonable frequency of such cases, these may be considered as acceptable alternatives, and thus Urdu word segmentation system would need to deal with both forms and consider them equivalent. This process is productively applicable and

not limited to a few pre-determined cases. Additional complication in the process arises from the fact that in some cases (last two cases in Table 2) the spellings also change when two words are written in combined form, due to the way these characters are encoded. Urdu considers ی and ے both logically same characters at a certain level, though with different shapes to indicated different vowels (Hussain, 2004). In combined form they render the same shape. However, Unicode terms ے as a non-joiner with no medial shape. Thus, the Urdu writers use ی to generate the medial position of ے in combined form.

3.2 Space Insertion

When multiple morphemes are juxtaposed within a word, many of them tend to retain their shaping as separate ligatures. If ending characters are joiners, space is usually inserted by writers to prevent them from joining and thus to retain the separate ligature identity. This causes an extra space within a word. Though this creates the visually acceptable form, it creates two tokens from a single word in the context of its processing. If the writers do not type a space between these two morphemes within a word they would join and create a visually incorrect shape. Such examples are common in Urdu[2]. Few of these cases are given in Table 3.

Class	A	B	Translation
i	شادی شدہ	شادیشدہ	Married
ii	موم بتی	مومبتی	Candle
iii	خواہ مخواہ	خوابمخواہ	Unnecessarily
iv	ٹیلی فون	ٹیلیفون	Telephone
v	پی ایچ ڈی	پیایچڈی	PhD
i= Affixation, ii = Compounding , iii = Reduplication, iv = Foreign Word, v = Abbreviations			

Table 3: (A) Separated Form (Correct Shaping, but Two Tokens), (B) Combined Form (Erroneous Shaping, with one Token)

As categorized in Table 3, the space insertion problem is caused due to multiple reasons. Data analyzed shows that space is inserted (i) to keep affixes separate from the stem, (ii) in some cases, to keep two words compounded together from visually merging, (iii) to keep reduplicated words from combining, (iv) to enhance readability of some foreign words written in Urdu, and (v) to keep English letters separate and readable when English abbreviations are transliterated.

3.3 Extent of Segmentation Issues in Urdu

In an earlier work on Urdu spell checking (Naseem and Hussain, 2007) report that a significant number of spelling errors[3] are due to irregular use of space, as discussed above. The study does a spelling check of an Urdu corpus. The errors reported by the spelling checker are manually analyzed. A total of 975 errors are found and of which 736 errors were due to irregular use of space (75.5%) and 239 errors are non-space-related errors (24.5%). Of the space related errors, majority of errors (672 or 70% of total errors) are due to space omission and 53 errors (5%) were due to space insertion. Thus irregular use of space causes an extremely high percentage of all errors and has to be addressed for all language processing applications for Urdu.

A study of Urdu words was also conducted as part of the current work. Text was used from popular Urdu online news sites (www.bbc.co.uk/urdu and http://search.jang.com.pk/). A data of 5,000 words from both corpora was observed and space insertion and omission cases were counted. These counts are given in Table 4. Counts for Space Insertion are sub-divided into the four categories identified earlier.

Problem	BBC	Jang	Total
Space Omission	373	563	936
Space Insertion			
Affixation	298	467	765
Reduplication	52	76	128
Compounding	133	218	351
Abbreviation	263	199	462
Total	1119	1523	2642

Table 4: Space Omission and Insertion Counts from Online BBC and Jang Urdu News Websites

The data shows that a significantly high percentage of errors related to space, with significant errors

[2] Though Unicode recommends using Zero Width Non-Joiner character in these context, this is not generally known by Urdu typists and thus not practiced; Further, this character is not available on most Urdu keyboards.

[3] Errors based on tokenization on space and punctuation markers

related to both omission and insertion. Within insertion errors, affixation, compounding and abbreviation related errors are more significant (because reduplication is a less frequent phenomenon).

In summary, the space related errors are significant and must be addressed as a precursor to any significant language and speech processing of the language

3.4 Ambiguity in Defining Urdu Word

Another confounding factor in this context it that there is no clear agreement on word boundaries of Urdu in some cases.

Compound words are hard to categorize as single or multiple words. Urdu forms compounds in three ways: (i) by placing two words together, e.g. ماں باپ ("parents", literally "father mother"), (ii) by putting a combining mark between them[4], e.g. وزیر اعظم ("prime minister"), and (iii) by putting the conjunction و between two words, e.g. نظم و ضبط ("Discipline").

Similarly certain cases of reduplication are also considered a single word by a native speaker, e.g. فرفر ("fluently") and برابر ("equal"), while others are not, e.g. آہستہ آہستہ ("slowly"). There are also cases which are ambiguous, as there is no agreement even within native speakers.

Moreover, certain function words, normally case markers, postpositions and auxiliaries, may be written joined with other words in context or separately. The words like کے لیے may also be written in joined form کیلیے, and the different forms may be perceived as multiple or single words respectively.

This is demonstrated by the results of a study done with 30 native speakers of Urdu (including university students, language researchers and language teachers). The subjects were asked to mark whether they considered some text a single word or a sequence of two words. Some relevant results are given in Table 5. The table indicates that for the types of phenomena in Table 4, the native speakers do not always agree on the word boundary, that certain cases are very ambiguous, and that writing with or without space also changes the perception of where the word boundary should lie.

Word(s)	# of Words		Category
	1	2	
وزیرمملکت	24	6	Compounding with conjunctive diacritic
حکومتِ پاکستان	17	13	-do-
صورتِ حال	28	2	-do-
صورتحال	28	2	-do-
امن وامان	25	5	Compounding with conjunctive character و
نشو ونما	29	1	-do-
عقیدت مندی	30	0	Suffixation
جرائم پیشہ	22	8	-do-
جگہ جگہ	3	27	Reduplication
ساتھ ساتھ	3	27	-do-
ہوگی	15	15	Space omission between two auxiliaries
جائیگا	18	12	Space omission between verb and auxiliary
جائے گا	5	25	Same as above but without space omission

Table 5: Survey on Word Definition

As the word boundary is ambiguously perceived, it is not always clear when to mark it. To develop a more consistent solution, the current work tags the different levels of boundaries, and it is left up to the application provider using the output to decide which tags to translate to word level boundaries. Free morphemes are placed and identified at first level. At second level we identify strings that are clearly lexicalized as a single word. Compounds, reduplication and abbreviations are dealt at third level.

4 Summary of Existing Techniques

Rule based techniques have been extensively used for word segmentation. Techniques including longest matching (Poowarawan, 1986; Rarunrom, 1991) try to match longest possible dictionary look-up. If a match is found at n^{th} letter next look-up is performed starting from n+1 index. Longest matching with word binding force is used for Chinese word segmentation (Wong and Chang, 1997). However, the problem with this technique is that it consistently segments a letter sequence the same way, and does not take the context into account.

[4] The diacritics (called *zer-e-izafat* or *hamza-e-izafat*) are optional, and are not written in the example given.

Thus, shorter word sequences are never generated, even where they are intended.

Maximum matching is another rule based technique that was proposed to solve the shortcomings of longest matching. It generates all possible segmentations out of a given sequence of characters using dynamic programming. It then selects the best segmentation based on some heuristics. Most popularly used heuristic selects the segmentation with minimum number of words. This heuristic fails when alternatives have same number of words. Some additional heuristics are then often applied, including longest match (Sornlertlamvanich, 1995). Many variants of maximum matching have been applied (Liang, 1986; Li et al., 1991; Gu and Mao, 1994; Nie et al., 1994).

There is a third category of rule based techniques, which also use additional linguistic information for generating intermediate solutions which are then eventually mapped onto words. For example, rule based techniques have also been applied to languages like Thai and Lao to determine syllables, before syllables are eventually mapped onto words, e.g. see (Phissamy et al., 2007).

There has been an increasing application of statistical methods, including n-grams, to solve word segmentation. These techniques are based at letters, syllables and words, and use contextual information to resolve segmentation ambiguities, e.g. (Aroonmanakul, 2002; Krawtrakul et al., 1997). The limitation of statistical methods is that they only use immediate context and long distance dependencies cannot be directly handled. Also the performance is based on training corpus. Nevertheless, statistical methods are considered to be very effective to solve segmentation ambiguities.

Finally, another class of segmentation techniques applies several types of features, e.g. Winnow and RIPPER algorithms (Meknavin et al., 1997; Blum 1997). The idea is to learn several sources of features that characterize the context in which each word tends to occur. Then these features are combined to remove the segmentation ambiguities (Charoenpornsawat and Kijsirikul 1998).

5 Segmentation Model for Urdu

Although many other languages share the same problem of word boundary identification for language processing, Urdu problem is unique due to its cursive script and its irregular use of space to create proper shaping. Though other languages only have space omission challenge, Urdu has both omission and insertion problems further confounding the issue.

We employ a combination of techniques to investigate an effective algorithm to achieve Urdu segmentation. These techniques are incorporated based on knowledge of Urdu linguistic and writing system specific information for effective segmentation. For space omission problem a rule based maximum matching technique is used to generate all the possible segmentations. The resulting possibilities are ranked using three different heuristics, namely min-word, unigram and bigram techniques.

For space insertion, we first sub-classify the problem based on linguistic information, and then use different techniques for the different cases. Space insertion between affixes is done by extracting all possible affixes from Urdu corpus. Some affixes in Urdu are also free morphemes so it is important to identify in segmentation process whether or not they are part of preceding or following word. For example ناک is also a free morpheme ("nose") and a suffix that makes adjective from noun, e.g. in word خطر ناک ("dangerous"). This is done based on the part of speech information of the words in the context.

Reduplication is handled using edit distance algorithm. In Urdu the reduplicated morpheme is either the same or a single edit-distance from the base morpheme, e.g. فرفر has same string repeated, برابر has one insertion, and ٹھیک ٹھاک has one substitution. Thus, if a string is less than two edits from its neighbor it is an instance of reduplication[5]. As the examples suggest, the reduplication may not only be limited to word initial position and may also occur word medially. However, if the length of base word is less than four, it is further to avoid function words (case markers, postpositions, aux-

[5] Insertion, deletion and substitution are all considered contributing a single edit distance here.

iliaries, etc.) from being mistakenly identified as a case of reduplication, e.g. کیا گیا ("was done") has two words with a single edit distance but is not a reduplicated sequence.

Urdu does not abbreviate strings, but abbreviations from English are frequently transliterated into Urdu. This sequence can be effectively recognized by developing a simple finite automaton. The automaton treats marks all such co-occurring morphemes because they are likely to be an English abbreviation transliterated into Urdu, e.g. پی ایچ ڈی ("PhD"). If such morphemes are preceding proper names then these are not combined as they are more likely to be the initials of an abbreviated name, e.g. این ڈی شاکر ("N. D. Shakir"). This approach confuses the morpheme کے (genitive case marker) of Urdu with the transliteration of English letter "k". If we write پی ایچ ڈی کے بعد ("after PhD"), it is interpreted as "P H D K after". This has to be explicitly handled.

As classification of compounds into one or two word sequences is unclear, unambiguous cases are explicitly handled via lexical look-up. An initial lexicon of 1850 compound words has been developed for the system based on a corpus of Urdu. Common foreign words are also included in this list.

5.1 Algorithm

The segmentation process starts with pre-processing, which involves removing diacritics (as they are optionally used in Urdu and not considered in the current algorithm because they are frequently incorrectly marked by users[6]) and normalizing the input text to remove encoding ambiguities[7]. Input is then tokenized based on space and punctuation characters in the input stream. As has been discussed, space does not necessarily indicate word boundary. However presence of space does imply word or morpheme boundary in many cases, which can still be useful. The tokenization process gives what we call an Orthographic Word (OW). OW is used instead of "word" because one OW may eventually give multiple words and multiple OWs may combine to give a single word. Keeping space related information also keeps the extent of problem to be solved within a reasonable computational complexity. For example input string نادر خان درانی (the name of the first author) with spaces giving three OWs, creates 2 x 1 x 7 = 14 possible segmentations when sent separately to the maximum matching module (space omission error removal - see Figure 2). However, if we remove the spaces from the input and send input as a single OW نادرخاندرانی to maximum matching process, we get 77 possible segmentations. This number grows exponentially with the length of input sentence. Throwing away space character means we are losing important information so we keep that intact to our use.

After pre-processing a series of modules further process the input string and convert the OWs into a sequence of words. This is summarized in Figure 2 and explained below.

Each OW is sent to a module which deals with space omission errors. This module extracts all possible morpheme segmentations out of an OW. Ten best segmentations of these are selected based on minimum-word heuristic. This heuristic prefers segmentations with minimum number of morphemes. Such a heuristic is important to prevent the search space to explode. We observed that using 10-best segmentations proved to be sufficient in most cases as OW normally encapsulates two or three Urdu words but as a heuristic we also added a feature which increases this number of 10-best segmentations to 15, 20, 25-best and so on depending upon number of characters in an OW. Ten best segmentations for each OW are merged with the extracted segmentations of other OWs. Up till here we have successfully resolved all space omission errors and the input sentence has been segmented into morphemes. The 10^n (where 'n' is No. of OWs) segmentations are then passed on to space insertion error removal module. This module has several sub-modules that handle different linguistic phenomena like reduplication, affixation, abbreviations and compounding.

[6] The word اعلیٰ is written with the super-script Alef placed on Lam and Yay characters. The latter variation is correct but the former incorrect variation is also common in the corpus.

[7] Unicode provides multiple codes for a few letters, and both composed and decomposed forms for others. These have to be mapped onto same underlying encoding sequence for further processing. See http://www.crulp.org/software/langproc/urdunormalization.htm for details.

The reduplication identification module employs single edit distance algorithm to see if adjacent morphemes are at single edit-distance of each other. If the edit distance is less than two, then the reduplication is identified and marked.

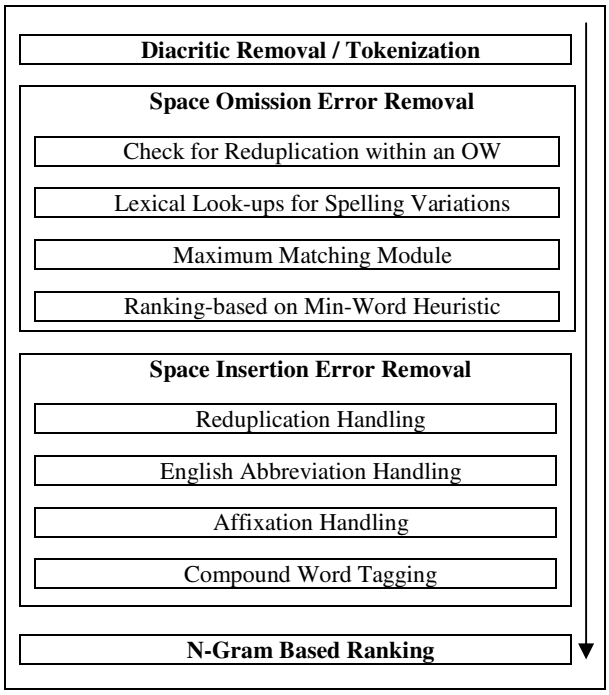

Figure 2: Urdu Word Segmentation Process

For example the module will correctly recognize consecutively occurring OWs بھولا and بھالا as a case of reduplication. Reduplication is also applied earlier in space omission error module as there may also be a case of reduplication within a single OW. This module handles such cases, by dividing words in halves and identifying possible reduplications. Thus, if the words are written without space, e.g. بھولابھالا (innocent) they are still identified and tagged as reduplicated words بھولا and بھالا.

This list of words is then fed into an automaton which recognizes the sequence of abbreviations generated by transliterating English letters.

A complete affix list is compiled, and in the next stage the short listed word sequences are processed through a process which looks through this list to determine if any of the OWs may be combined. Part of speech information of stem is also used to confirm if OWs can be merged.

Urdu compounds are finally identified. This is done by using a compound list generated through the corpus. As compounding is arbitrary, where speakers are not certain in many cases that a sequence of morphemes form a single compound or not, the segmentation process leaves this level to the discretion of the user. Whichever compounds are listed in a compound lexicon are treated as a single compound word. Those not listed are not tagged as compounds. User may enhance this list arbitrarily. The lexicon is initialized with a list of non-controversial compound, as verified from published dictionaries.

Eventually, all the segmentations are re-ranked. We used three different re-ranking methods namely minimum-word heuristic, unigram and bi-gram based sequence probabilities, comparative analysis.

Based on the segmentation process, the output sequence contains the following tagging. As discussed earlier, the word segmentation may be defined based on this tagging by the individual application using this process.

Phenomenon	Tags	Examples
Word	<W></W>	<W>اعلان</W>
Root	<R></R>	<W><R>ضرورت</R> <S>مند</S></W>
Suffix	<S></S>	<W><R>حیرت</R> <S>انگیز</S></W>
Prefix	<P></P>	<W><P>بد</P> <R>تہذیبی</R></W>
XY Compounds	<C1></C1>	<C1><W>انشاء</W> <W>اللہ</W></C1>
X-e-Y Compounds	<C2></C2>	<C2><W>وزیر</W> <W>اعلی</W></C2>
X-o-Y Compounds	<C3></C3>	<C3><W>گرد</W> <W>و</W> <W>نواح</W></C3>
Reduplication	<Rd></Rd>	<Rd><W>ٹھیک</W> <W>ٹھاک</W></Rd>
Abbreviations	<A></A>	<A><W>پی</W> <W>سی</W> </A>

Figure 3: Urdu Word Segmentation Tag Set

A regular word is tagged using <w> …</w> pair. Roots, suffixes and prefixes are also tagged within a word. Reduplication, compounding and abbreviations are all considered to be multi-word tags and relevant words are grouped within these tags. Three different kind of compounding is separately tagged.

6 Results

The algorithm was tested on a very small, manually segmented corpus of 2367 words. The corpus we selected contained 404 segmentation errors with 221 cases of space omissions and 183 cases of space insertions. In space insertion category there were 66 cases of affixation, 63 cases of compounding, 32 cases of reduplication and 22 cases of abbreviations. The results for all three techniques are shown below:

Categories	Errors	%ages
Affixation	59/66	89.39
Reduplication	27/32	84.37
Abbreviations	19/22	86.36
Compounds	28/63	44.44
Total	133/183	72.67

Table 6: Percentages of Number of Errors Detected in Different Categories of Space Insertion Error

There were 221 cases of space omission errors where multiple words were written in a continuum. Given below is a table that shows how many of these were correctly identified by each of the used techniques. Clearly, statistical techniques outperform a simple minimum number of words heuristic. Bigrams are likely to produce better results if the training corpus is improved. Our training corpus contained manually segmented 70K words. The bigram probabilities are obtained using SRILM-Toolkit (Stolcke, 2002).

Categories	Errors	%ages
Maximum Matching	186/221	84.16
Unigram	214/221	96.83
Bigram	209/221	94.5

Table 7: %age of No. of Errors Detected in Space Omission with Different Ranking Techniques

Following table gives cumulative results for correctly identified space omission and insertion errors.

Categories	Errors	%ages
Maximum Matching	323/404	79.95
Unigram	347/404	85.8
Bigram	339/404	83.9

Table 8: %age of No. of Errors Detected Cumulatively

Final table counts total number of words (reduplication, compounds and abbreviations cases are inclusive) in test corpus and total number of correctly identified words after running the entire segmentation process.

Categories	Detected	%ages
Maximum Matching	2209/2367	93.3
Unigram	2269/2367	95.8
Bigram	2266/2367	95.7

Table 9: Percentage of Correctly Detected Words

7 Future Work

This work presents a preliminary effort on word segmentation problem in Urdu. It is a multi-dimensional problem. Each dimension requires a deeper study and analysis. Each sub-problem has been touched in this work and a basic solution for all has been devised. However to improve on results each of these modules require a separate analysis and study. Statistics is only used in ranking of segmentations. In future work bi-gram statistics can be used to merge morphemes. More data can be tagged to find out joining probabilities for the affixes that occur as free morpheme. Such analysis will reveal whether an affix is more inclined towards joining or occurs freely more frequently. Similarly a corpus can be tagged on compounds. For each morpheme its probability to occur in compound can be calculated. If two or more morphemes with higher compounding probabilities co-occur they can be joined together. Similarly corpus can be tagged for abbreviations.

Ranking of segmentations and affix merging can be improved if POS tags are also involved with bigram probabilities. Use of POS tags with n-gram technique is proven to be very helpful in solving unknown word problems. Our model does not explicitly handle unknown words. Currently the maximum matching module splits an unknown OW into smaller Urdu morphemes. For example کولیسینکوف(Kolesnikov) is erroneously split into کولی،سین،کو،ف. More serious problems occur in cases when OW is a mixture of known and unknown words. For example in case فریزرکوجانا ہے (“Fraser must go”). All these are to be addressed in future work.

References

Andreas, S. 2002. SRILM - an extensible language modeling toolkit. In Intl. Conf. Spoken Language Processing, Denver, Colorado.

Aroonmanakul, W. 2002. Collocation and Thai Word Segmentation. In *proceeding of SNLPOriental* COCOSDA.

Blum, A. 1997. Empirical Support for Winnow and Weighted-Majority Algorithm: Results on a Calendar Scheduling Domain, Machine Learning, 26:5-23.

Charoenpornsawat, P., Kijsirikul, B. 1998. Feature-Based Thai Unknown Word Boundary Identification Using Winnow. In *Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS'98).*

Gu, P. and Mao, Y. 1994. The adjacent matching algorithm of Chinese automatic word segmentation and its implementation in the QHFY Chinese-English system. In *International Conference on Chinese Computing*, Singapore.

Hussain, S. 2003. www. LICT4D . asia / Fonts / Nafees_Nastalique. In the *Proceedings of 12th AMIC Annual Conference on E-Worlds: Governments, Business and Civil Society, Asian Media Information Center,* Singapore. Also available at http://www.crulp.org/Publication/papers/2003/www.LICT4D.asia.pdf.

Hussain, S. 2004. Letter to Sound Rules for Urdu Text to Speech System. In the *Proceedings of Workshop on Computational Approaches to Arabic Script-based Languages,* COLING 2004, Geneva, Switzerland, 2004.

Krawtrakul, A., Thumkanon. C., Poovorawan. Y. and Suktarachan. M. 1997. Automatic Thai Unknown Word Recognition. In *Proceedings of the natural language Processing* Pacific Rim Symposium.

Li, B.Y., S. Lin, C.F. Sun, and M.S. Sun. 1991. A maximum-matching word segmentation algorithm using corpus tags for disambiguation. In *ROCLING IV*, pages: 135-146, Taipei. ROCLING

Liang, N. 1986. A written Chinese automatic segmentation system-CDWS. In *Journal of Chinese Information Processing*, 1(1):44-52.

Meknavin. S., Charenpornsawat. P. and Kijsirikul. B. 1997. Feature-based Thai Words Segmentation. NLPRS, Incorporating SNLP.

Naseem, T., Hussain, S. 2007. Spelling Error Trends in Urdu. In the *Proceedings of Conference on Language Technology (CLT07),* University of Peshawar, Pakistan.

Nie, J., Jin W., and Hannan, M. 1994. A hybrid approach to unknown word detection and segmentation of Chinese. In *International Conference on Chinese Computing,* Singapore.

Phissamay, P., Dalolay,V., Chanhsililath, C., Silimasak, O. Hussain, S., and Durrani, N. 2007. Syllabification of Lao Script for Line Breaking. In *PAN Localization Working Papers 2004-2007.* .

Poowarawan, Y., 1986. Dictionary-based Thai Syllable Separation. In *Proceedings of the Ninth Electronics Engineering Conference*

Rarunrom, S., 1991. Dictionary-based Thai Word Separation. Senior Project Report.

Sornlertlamvanich, V. 1995. Word Segmentation for Thai in a Machine Translation System (in Thai), *Papers on Natural Language Processing, NECTEC, Thailand*

Wong, P., Chan, C. 1996. Chinese Word Segmentation based on Maximum Matching and Word Binding Force. In *Proceedings of COLING 96*, pp. 200-203.