International Journal of Ocean System Engineering 2(1) (2012) 24-31
http://dx.doi.org/10.5574/IJOSE.2012.2.1.024

A Methodology for Urdu Word Segmentation using Ligature and Word Probabilities

Yunus Khan [1*], Chetan Nagar [2] and Devendra S Kaushal [3]

[1]M.E Researcher JIT Borawan Khargone M.P. India

[2]Professor & Head Department of CSE JIT Borawan M.P. India

[3]M.Tech. Researcher SATI Vidisha M.P. India

(Manuscript Received December 10, 2011; Revised January 4, 2012; Accepted February 20, 2012)

Abstract

This paper introduce a technique for Word segmentation for the handwritten recognition of Urdu script. Word segmentation or word tokenization is a primary technique for understanding the sentences written in Urdu language. Several techniques are available for word segmentation in other languages but not much work has been done for word segmentation of Urdu Optical Character Recognition (OCR) System. A method is proposed for word segmentation in this paper. It finds the boundaries of words in a sequence of ligatures using probabilistic formulas, by utilizing the knowledge of collocation of ligatures and words in the corpus. The word identification rate using this technique is 97.10% with 66.63% unknown words identification rate.

Keywords: Handwritten Urdu OCR System; Urdu Ligature; Word language model; Ligature language model; Word Segmentation.

1. Introduction

In Urdu language space is not used to split two words in a single sentence. For this readers us the ligature boundaries to distinguish them. In Urdu script space is prefer to get appropriate character shapes and sometimes used as a ligature. Therefore, for Urdu language processing, word segmentation or word tokenization is primary technique for understanding meanings of the sentences [1] [2] [3] [4] [5]. It has applications in many areas like handwritten recognition, spell checking, POS, speech synthesis, information retrieval and text categorization [2], however, this paper gives the solution of word segmentation problem from the point of view of handwritten text Optical Character Recognition (OCR) System. The word segmentation model for handwritten Urdu OCR system takes input either in the form of characters or in form of ligatures to construct words from them. This paper assumes that word segmentation model obtains input in form of ligatures from the handwritten OCR recognizer.

2. Litrature Review

Text in the Latin based languages such as English, French and Spanish is easily segmented into words by using word delimiters such as space, comma and semi colon etc., but many Asian languages like Urdu, Persian, Arabic, Chinese and Thai have problem of word segmentation since text is written continuously without separators in these languages. The techniques used previously for word segmentation in other languages are categorized into: (i) dictionary/lexicon based

[*] Corresponding author. Tel.: +91-7285-277862, Fax.: +91-7285-277710.
E-mail address: callyunuskhan@gmail.com

approaches, (ii) linguistic knowledge based approaches and (iii) machine learning based approaches/statistical approaches [4]. Longest matching approach [14] [19] and maximum matching approach [16] [5] are dictionary/lexicon based approaches. These techniques segment text using the dictionary or lexicon [5].Their accuracy depends on the quality and size of the dictionary. Out Of Vocabulary (OOV) may occur in these approaches [4].

N-Grams [17] [18] [19] [20] [21] and Maximum collocation approach [21] are Linguistic knowledge based approaches which also rely very much on the lexicon. These approaches select most likely segmentation from the set of possible segmentations using a probabilistic or cost-based scoring mechanism [5].

Word Segmentation Using Decision Trees Approach [22] [1] and Word Segmentation Using Lexical Semantic Approach [3] fall in the third category of word segmentation techniques. These approaches use a corpus in which word boundaries are explicitly marked. These approaches do not require dictionaries. In these approaches ambiguity problems are handled by providing a sufficiently large set of training examples to enable accurate classification [23].

3. Methodology

The methodology followed for the solution of Urdu words segmentation problem is similar to building a language model. It uses the ligature co-occurrence information along with words collocation information to construct a language model. In order to execute this methodology, we have built a proper segmented training corpus. The whole process is completed in three phases. In the first phase, information data a necessary for the Urdu word segmentation model is gatherd. Using this collected data, ligature and word probabilities are calculated. In the second phase, all sequences of words are generated from input set of ligatures and ranking of these sequences is performed using the lexicon lookup. According to a selected beam value top k sequences are selected for further processing. It uses valid words heuristic for selection process. In the third phase, maximum probable sequence, from these k word sequences is selected. Details of above three phases are described in subsequent sections.

3.1 Information Gathering and Probabilities Calculations (First Phase):

This step involves gathering of data to be used for the word segmentation model. Most of the data is gathered from the Center for Research in Urdu Language Processing (CRULP). The complete data is used for different processes in the word segmentation model. This data includes,

- Information for developing a word dictionary:

For constructing a dictionary we have gathered Urdu words from domains of word-affixes, person's names, country names, city names and company names. For the cleanup process of above data, firstly from 50170 distinct words, a word list of 49635 unique words was obtained by removing non dictionary words [6]. For example words like اب ڑا،ادب ا etc are removed from the word list. Secondly, the word–affixes list is modified by insertion of the zero-width-non-joiner. This list is also maintained without zero-width-non-joiner (ZWNJ) for further processing in data word grams. For example م ند احسان with space is replaced with م ند احسان which is without space. Thirdly, person names and company names are tokenized on space and added as words in the dictionary.

- Information for ligature grams: The Corpora

used for developing ligature grams consist of half million words. For this project, from 18-million word corpora [7], 300,000 words are taken from Society, Consumer Information and Culture/Entertainment domains. 100,000 words are obtained from Urdu corpus available at [24] from the project of Urdu-Nepali-English Parallel Corpus. 100,000 words are obtained from Hassan's POS tagged Corpus [25] .Tags of this corpus are removed before further processing.

- Information for word grams: For the compu-

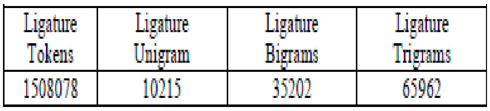
tation of word grams, a corpus is obtained which comprises of 18 million words of Urdu text [7]. This corpora is taken from the domains of sports/games, news, finance, culture and entertainment, consumer information and personal communications [7].

The next step after Information collection is probability estimation. For this model the ligature grams and word grams probabilities are estimated. For estimating the ligature grams, a cleaned properly segmented ligature corpus is required. Therefore before converting the word corpus to ligature corpus, a half million words corpus is cleaned for proper segmentation. As corpus cleaning is very monotonous and time consuming task and cleaning merely with manual effort is very slow, therefore, the corpus cleaning for ligature grams included some automated tasks but most of the work is done manually. Since the basic source for Sports, Consumer Information and Culture/Entertainment corpora files is newspaper so these files are cleaned to remove hypertext markups and English characters. Also, since "space character" in Urdu script has been used between two words to correct glyph shaping, therefore collected Urdu corpora have problem of space insertion, space removal and insertion of ZWNJ to maintain the correct shape of words. For Examples Urdu Corpora has words like , ، کش خود ، ذمہداری ,اق بالا سٹاف غرضی خود ، اسلامآباد پوسٹمارٹم etc. Ligature is a sequence of characters in a word separated by non-joiner characters or the Unicode ZWNJ character. These Non-joiners appear at only isolated and final position. The algorithm of converting the word corpora to the ligature corpora is as follows.

```
For(Character input)
  If(character=”NON JOINER”)
    Do
        Add this character to output text file
        with blank space
    Else
        Add this character to output text file
    End if
 End for
  “Algorithm for word to ligature conversion”
```

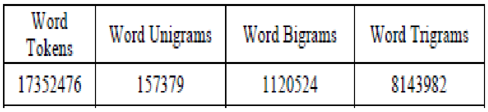
Using the above pseudo code the word corpora collected for ligature grams is converted to ligature corpora. A ligature unigram is a distinct ligature in a corpus. For the word grams probabilities calculation, first frequencies are computed and then cleaning of corpus through these frequencies is performed using some heuristics because the corpus used for word grams is very huge and it is not possible to clean 18-million word corpus before these calculations. Table I and Table II give the count frequencies and probabilities for unigram, bigrams and trigram of the ligature and word corpora respectively.

Table 1. Count of frequencies and probabilility for unigram, bigram and trigram of the ligature corpus



Ligature Tokens	Ligature Unigram	Ligature Bigrams	Ligature Trigrams
1508078	10215	35202	65962

Table 2. Count of frequencies and probabilility for unigram, bigram and trigram of the ligature corpus



Word Tokens	Word Unigrams	Word Bigrams	Word Trigrams
17352476	157379	1120524	8143982

After calculation of word unigram, bigram, and trigram counts, the following cleaning issues of corpus are handled with the help of these calculations.

• In the word corpus, certain words are combined without space and need to exist as separate words. These words occur with very high frequency in corpora for example “ ہوگا ” exists as single word rather than two individual words. To solve this space insertion problem, a list of about 700 words with frequency greater than 50 is obtained from the word unigrams. Each word of the list is manually viewed and space is inserted, where required, in each space insertion error word. Then these error words are removed from the word unigram and added to the word unigram frequency list as two or three individual words with frequency of the respective error word. For the space insertion problem in word bigrams, each error word in joined-word list (700-word list) is checked. If any of these error words occurs in a bigram word frequency list, for example “ ہوگا کیا ” exists in the bigram list and contain " ہوگا "error word, then this bigram entry “ ہوگا کیا ” is removed from the bigram list and frequencies of ہو گا“ ” and “ ہو کیا ” are increased by the frequency of “ ک یا ”. If these words do not exist in the word bigram frequency list then these are added as a new bigram word with the frequency of “ ہوگ ا ک یا ”. Same procedure is performed for the word trigrams.

- The second main issue is the word-affixes. These are treated as separate words and exist as bigram entries in the list rather than a unigram entry. For example " م ند صحت " exists as a bigram entry but in Urdu it is treated as a single word. To cope with this problem the list of word - affixes (used in making dictionary) is used. If any entry of word bigram matches with an affix word, then this word is combined by removing space from it and inserting zero-width-non-joiner (ZWNJ), if required to maintain its glyph shape. Then this affix word is inserted in the unigram list with its original bigram frequency. Same procedure is performed if a trigram word matches with an affix.

After resolving cleaning issues word and ligature unigram, bigram and trigram probability calculations are performed. The following formulas are used respectively,

$$P = \frac{C(w_i)}{\text{total Number of Words/Ligatures(N)}} \tag{1}$$

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})} \tag{2}$$

$$P(w_i|w_{i-2}w_{i-1}) = \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})} \tag{3}$$

To avoid data sparseness One Count smoothing described in [9] is applied on calculated probabilities. Using this technique estimated probabilities are calculated with the following equation

,

$$\text{Pone}\left(w_i \middle| w_{i-n+1}^{i-1}\right) = \frac{C\left(w_{i-n+1}^{i}\right) + \alpha\,\text{Pone}\left(w_i \middle| w_{i-n+2}^{i-1}\right)}{C\left(w_{i-n+1}^{i-1}\right) + \alpha} \tag{4}$$

where

$$\alpha = \gamma\left[n_1\left(w_{i-n+1}^{i-1}\right) + \beta\right], n_1\left(w_{i-n+1}^{i-1}\right) = \left|w_i : C\left(w_{i-n+1}^{i}\right) = 1\right|$$

and β and γ are constants

This Pone Smoothing technique merges two perceptions. Firstly, Pone probability technique is a reasonable form of smoothed distribution as argued by MacKay and Peto [10] that is, the parameter α represents the number of counts being added to the given distribution and new counts are distributed to the lower order distributions by recursive part of (4). Secondly, from the Good-Turing estimate [11] it can be inferred that the number of these extra counts that is denoted by α should be proportional to the number of words with exactly one frequency in the given distribution. This inference of the Good-Turing works well in (7).

3.2 Generating Words Sequences

After obtaining input in form of sequence of ligatures separated by space, all possible word segments are generated and then ranking of them is performed. In this process a tree of ligatures is built. The first ligature is added as a root of tree and at each level of the tree maximum three or minimum two child nodes are added to each node of tree. For example the second level of ligature sequence tree contains the following tree nodes.

- The string of the first node (Left Child of root) is composed of parent (root) string and next input ligature (which is second ligature here) combined with space.
- The string of the second node (Middle Child of root) is composed of parent (root) string and next ligature combined without space.
- The string of third node (Right Child of root) is composed of parent (root) string and next ligature, combined with ZWNJ if the node string of the parent node ends with a non joiner. Otherwise this node (Right child of the root) is not added in the current level of tree.

At each level of the tree, for each node a numeric value is assigned to each node-string of that node. For assigning these numeric values, firstly, all the space separated words are obtained from the node-string. For each word of the nodestring, if this word exists in the dictionary then a numeric value is assigned to this word. This numeric value is equal to the square of number of ligature this word is composed of. Otherwise if this word does not exist in dictionary then its count value is zero. The total numeric value of the node-string is the sum of the numeric values of each word of node-string which are separated by space.

If a node-string has only one word and this word does not occur in the dictionary as a valid word then it is checked that this word may occur at the start of any dictionary entry. In this case numeric value is also assigned.

After assignment of numeric value to each node at current level, node-strings are ranked according to these counts/values and best k (beam value) nodes with respected node-strings are selected. These selected nodes are further explored for processing and remaining lower ranked nodes and their respected strings are ignored.

3.3 Selection of the Best Word Segmentation Sequence

For selection of the most probable word segmentation sequence, firstly word language models and ligature language models are used. Secondly by using the relationship between words and ligatures, a model named as word bigram ligature bigram is derived and thirdly the variations of this model are obtained. The word language model [12] can be stated as

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} P(w_1^n) \qquad (5)$$

Using chain rule of probability for the decomposition of probability(PW1n) in (5) as

$$P(w_1, w_2, w_3, w_4, \_\_\_ w_n) = \prod_1^n P(w_k | w_1^k) \qquad (6)$$

To reduce the complexity of computing, Markov assumption is used and bigram approximation and trigram approximations [12] are taken and (6) results in (7) and (8) as

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} \prod_1^n P(w_i | w_{i-1}) \qquad (7)$$

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} (\prod_{k=1}^n P(w_k | w_{k-1} w_{k-2}) \quad (8)$$

Similarly the ligature language models can be built by taking assumption that sentences are made up of sequences of ligatures rather than words and space is also a valid ligature. By taking the Markov bigram and trigram assumption for ligature grams we have,

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} (\prod_1^m (P(l_i | l_{i-1})) \qquad (9)$$

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} (\prod_1^m (P(l_i | l_{i-1} l_{i-2})) \qquad (10)$$

Now by utilizing the ligature sequences and relationship among these ligatures to make words, (5) can be enhanced as

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} P(w_1^n | l_1^m) \qquad (11)$$

where $w_1^n = w_1, w_2, w_3, w_4, \_\_\_ w_n$ and $l_1^m = l_1, l_2, l_3, l_4, \_\_\_ l_m$ *n* represents number of words and m represents the number of ligatures. This equation also represent that m number of ligatures can be assigned to n number of words. By applying the Bayesian theorem on (11),

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} \frac{P(l_1^m | w_1^n).P(w_1^n)}{P(l_1^m)} \qquad (12)$$

Since P (l_1^m) remain constant for all w_1^n , so can be ignored as.

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} P(l_1^m | w_1^n). P(w_1^n) \qquad (13)$$

Where

$$P(l_1^m | w_1^n) = P(l_1, l_2, l_3, l_4, \_\_\_ l_m | w_1^n)$$

$$= P \quad (\quad l_1 | w_1^n) * P(l_2 | w_1^n l_1) * P(l_3 | w_1^n l_1 l_2) * P(l_4 | w_1^n l_1 l_2 l_3) * \ldots P(l_m | w_1^n l_1 l_2 l_3 \ldots l_{m-1})$$

Let’s assume that a ligature I1 depends only on the word sequence W1n and its previous ligature Ii-1, not all the previous ligature history so above equation can be written as,

$$P(l_1^m | w_1^n) = P(l_1 | w_1^n) * P(l_2 | w_1^n l_1) * P(l_3 | w_1^n l_2) * P(l_4 | w_1^n l_3) * \ldots P(l_m | w_1^n l_{m-1})$$

$$= \prod_1^m P(l_i | w_1^n l_{i-1}) \qquad (14)$$

Here another assumption is taken that l□ depends on the word in which it appears not whole word sequence. A word in which l□ appears it always gives a value of 1 and does not contribute .So (11) can be written as,

$$P(l_1^m | w_1^n) = \prod_1^m P(l_i | l_{i-1}) \qquad (15)$$

Now putting values in (13) we have,

$$P(W) = \mathrm{argmax}_{w_1^n \in S} (\prod_1^m (P(l_i | l_{i-1})) (\prod_{k=1}^n P(w_k | w_{k-1})) \qquad (16)$$

Equation (16) gives the maximum probable word sequence among all the alternative word sequences in set S. The next step is obtaining the variations of word bigram ligature bigram technique which is stated as The variations of (16) are as follow Ligature trigram

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} (\prod_1^m (P(l_i | l_{i-1} l_{i-2})) * (\prod_{k=1}^n P(w_k | w_{k-1})) \qquad (17)$$

- Ligature bigram and word trigram based technique

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} (\prod_1^m (P(l_i | l_{i-1})) * (\prod_{k=1}^n P(w_k | w_{k-1} w_{k-2})) \qquad (18)$$

- Ligature trigram and word trigram based technique

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} (\prod_1^m (P(l_i | l_{i-1} l_{i-2})) * (\prod_{k=1}^n P(w_k | w_{k-1} w_{k-2})) \qquad (19)$$

- Normalized ligature bigram and word bigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}(\Pi_1^m(P\,(l_i|l_{i-1}))^{1/NL} * (\Pi_{k=1}^n P(w_k|w_{k-1})\ ^{1/NW} \quad (20)$$

- Normalized ligature trigram and word bigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}\left((\Pi_1^m(P\,(l_i|l_{i-1}l_{i-2}))^{1/NL}\right) * (\Pi_{k=1}^n P(w_k|w_{k-1}))\ ^{1/NW} \quad (21)$$

- Normalized ligature bigram and word trigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}(\Pi_1^m(P\,(l_i|l_{i-1}))^{1/NL} * (\Pi_{k=1}^n P(w_k|w_{k-1}w_{k-2})\ ^{1/NW} \quad (22)$$

- Normalized ligature trigram and word trigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}(\Pi_1^m(P\,(l_i|l_{i-1}l_{i-2}))^{1/NL} * (\Pi_{k=1}^n P(w_k|w_{k-1}w_{k-2})\ ^{1/NW} \quad (23)$$

- Optimal technique

  In this technique firstly all the word sequences with highest probabilities are found using techniques presented by equations (7), (8), (9), (10), (16), (17), (18), (19), (20), (21), (22), (23). Then only one word sequence is selected which is the most occurring in the output of these techniques.

All above techniques give a most probable sequence of words given a set S of word sequences w_1^n and a fix set of ligature sequence l_1^m. Where in these techniques, probability $P(l_i|l_{i-1})$ and probability $P(l_i|l_{i-1}|l_{i-2})$ are estimated ligature bigram and trigram Pone probabilities respectively calculated from ligature corpora. Probability $P(W_k|W_{k-1}|)$ and $P(W_k|W_{k-1}|W_{k-2})$ are estimated Pone word bigram and word corpora. NL represents the number of ligature bigrams/trigrams that exist in the corpus and NW represents the number of word bigram. trigrams that exist in the corpus for the given sentence.

3.4 Results and Discussion

The model is tested on a corpus of 150 sentences composed of 2156 words and 6075 ligatures. In these sentences, 62 words are unknown and 2092 are known words. Unknown words mean here, the words that do not exist in our dictionary. The average length of the sentence is 14 in terms of words and 40.5 in terms of ligatures. The average length of word is 2.81 in terms of ligatures. All the techniques presented previously are for the beam value of 10, 20,30,40,50. The results can be viewed in two perspectives

- Sentence Identification rate
- Word identification rate

Optimal technique gives the sentence identification rate of 76% which is highest among all techniques with the beam value of 30 but from point view of word's identification rate, Normalized Ligature Trigram Word Trigram Technique (23) outperforms then all other techniques and gives 96.10% words identification rate and 65.3 % unknown words identification rate for the beam value of 50 which is highest among all techniques.

First type of errors is sentence identification errors. A sentence is considered incorrect even if one word of the sentence is identified wrongly. This type of errors depends on the other two types of errors. For example for the beam value of 30 we have 38 sentences incorrect. In the 38 sentences 25 sentences are identified in the wrong way due to unknown words errors and remaining 13 errors are due to known word identification errors. So improvement in recognition of other two types of errors results in the improvement of sentence identification rate.

Second type of errors is known word's identification errors. Most of the errors in this category are of space insertion which means two words are joined together and space is deleted from them. The reason of these errors is insufficient cleaning of word grams. The words with frequency greater than 50 in the unigram list, which covers 18962196 words, are find out and cleaned. Other low frequency words cause these errors for example errors " ب ن یادپ ر","سے ت قس یم "are space insertion errors and these error words exist in word corpora with frequency of 40 and 5 respectively, which falsifies our results. If low frequency words are also cleaned from the word grams lists then error rate for the space insertion errors will become low and results of known word recognition errors will definitely improve. Other errors in this category are due to incorrect selection of beam value.

Third type of errors is unknown word's recognition errors. These words do not exist in the dictionary. Most of these errors are recognized as real word errors. Real words are not the words that the writer intends but these words are correctly spelled words in the dictionary, For example a word " کارت ک " is a proper noun and does not exist in dictionary. This system recognizes it as two words " ت ک کار " which are valid words of dictionary. Other unknown words which are incorrectly identified are diacritized words. So the unknown words rate can be further improved by enhancing dictionary with diacritize words along with the words without diacritics.

4. Conclusion

This work presents a initial effort on statistical solution of word segmentation problem for Urdu OCR systems and simultaneously for the Urdu language. Other South Asian languages, like Chinese, have only space insertion problem. Here Urdu language differs from these languages as it also faces space removal and zero-width-non joiner insertion problems with the space insertion problem. All these problems have their own dimension and require intensive cleaned data. This work tries to solve all these problems and effectively resolve space removal problems but space insertion problem requires more detailed analysis and cleaning.

Results of ligature grams are poor than word grams techniques, for the effectiveness of the ligature gram techniques huge amount of cleaned data for ligature grams is required.

5. Future Work and Scope

This work used the knowledge of ligature grams and word grams. This work can be further enhanced by using the character grams information. In this work Statistics are only tool used for the word segmentation so the Urdu rules for the formation of words or rule based techniques can also be used along with the statistics information to improve the results.

We have tried to clean the corpus with respect to space removal, space insertion and ZWNJ insertion. These lists are need to be improved as well as abbreviations and English words are needed to handle more effectively.

The unknown word detection rate can be increased efficiently by applying POS tagging techniques or word net based techniques with the minimum distance which results in the improvement of the real word detection errors.

Other issues are related to memory as the loading of the word trigram requires huge memory. This problem can be handled by reducing the amount of trigrams by using some grammatical trigram techniques.

References

[1] Thanaruk Theeramunkong and Sasiporn Usanavasin , “Non-Dictionary- Based Thai Word Segmentation Using Decision Trees ”, Human Language Technology Conference, Proceedings of the first international conference on Human language technology research, (2001).

[2] Xin-Jing Wang,Wen Liu,Yong Qin ,“A Search-based Chinese Word Segmentation Method”, International World Wide Web Conference, Proceedings of the 16th international conference on World Wide Web, (2007).

[3] Krisda Khankasikam and Nuttanart Muansuwan , “Thai Word Segmentation a Lexical Semantic Approach”, (2008).

[4] Choochart Haruechaiyasak, Sarawoot Kongyoung and Matthew N. Dailey, “A Comparative Study on Thai Word Segmentation Approaches”, (2008)

[5] Li Haizhou and Yuan Baosheng , “Chinese Word Segmentation”. In Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation, PACLIC-12, (1998) 212-217.

[6] http://www.crulp.org/oud/default.aspx

[7] Sarmad Hussain, "Resources for Urdu Language Processing", In Proceedings of the Sixth Workshop on Asian Language Resources, (2008).

[8] Sajjad, H."Statistical Part-of-Speech for Urdu", MS thesis , Centre for Research in Urdu Language Processing , National University of Computer and Emerging Scientist, Lahore, Pakistan , (2007).

[9] Stanley F. Chen and Joshua T. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling", In Proceedings of the 34th Annual Meeting of the Association for

[10] MacKay, David J. C. and Linda C. Peto ," A hierarchical Dirichlet language model ", Natural Language Engineering, 1 (3) (1995) 1-19.

[11] Church, Kenneth W. and William A. Gale. 1991," A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams" , Computer Speech and Language, (5) 19-54.

[12] Daniel Jurafsky, James H. Martin. “Speech and Language Processing”.

[13] Pak-kwong Wong and Chorkin Chan, “Chinese Word Segmentation based on Maximum Matching and Word Binding Force”. In Proceed-ings of the 16th conference on Computational linguistics, (1996).

[14] Poowarawan, Y., “Dictionary-based Thai Syllable Separation”, Proceedings of the Ninth Elec-tronics Engineering Conference, (1986).

[15] Fung Pascale and Wu Dekai, "Statistical augmentation of a Chinese machine readable dic-tio-nary", (1994).

[16] Richard Sproat, Chilin Shih, William Gale and Nancy Chang, "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese", Computational Linguistics, 3 (22) (1996).

[17] Chang, Jyun-Shen, Shun-De Chen, Ying Zhen, Xian-Zhong Liu and Shu-Jin Ke, "Large-corpus-based methods for Chinese personal name recognition", Journal of Chinese Information Processing, 6 (3) (1992) 7-15.

[18] Li Haizhou et al, "Pinyin Streamer: Chinese pinyin to hanzi translater", Apple-ISS technical report, (1997).

[19] Richard Sproat, Chilin Shih, William Gale and Nancy Chang, "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese", Com-putational Linguistics, 3 (22) (1996).

[20] Jian-Cheng Dai and Hsi-Jian Lee, "Paring with Tag Information in a probabilistic generalized LR parser", International Conference on Chinese Computing, Singapore, (1994).

[21] Wirote Aroonmanakun ,”Collocation and Thai Word Segmentation”, In Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop.

[22] Virach Sornlertlamvanich, Tanapong Potipiti and Thatsanee charoenporn, “Automatic Corpus-Based Thai Word Algorithm Extraction with the C4.5 Learning”, Proceedings of the 18th conference on Computational linguistics, (2000).

[23] Alexander Clark1 and Shalom Lappin2, “Grammar Induction through Machine Learning”.

[24] http://www.crulp.org/software/ling_resources/ UrduNepaliEnglishParallelCorpus.htm

[25] Sajjad, H."Statistical Part-of-Speech for Urdu", MS thesis , Centre for Research in Urdu Language Processing , National University of Computer and Emerging Sciencies, Lahore, Pakistan, (2007).