Implementation of Urdu Probabilistic Parser

Neelam Mukhtar *sameen_gul@yahoo.com*
Department of Computer Science
University of Peshawar, Pakistan

Mohammad Abid Khan *abid_khan1961@yahoo.com*
Department of Computer Science
University of Peshawar, Pakistan

Fatima Tuz Zuhra *fateeshah@yahoo.com*
Department of Computer Science
University of Peshawar, Pakistan

Nadia Chiragh *nadiachiragh@yahoo.com*
College of Home Economics
University of Peshawar, Pakistan

Abstract

The implementation of Urdu probabilistic parser is the main contribution of this research work. In the beginning, a lot of Urdu text was collected from different sources. The sentences in the text were subsequently tagged. The tagged sentences were then parsed by a chart parser to formulate the rules. In the next step, probabilities were assigned to these rules to get a Probabilistic Context Free Grammar. For Urdu probabilistic parser, the idea of shift-reduce multi-path strategy is used. The developed software performs the syntactic analysis of a sentence, using a given set of probabilistic phrase structure rules. The parse with the highest probability is selected, as the most suitable one from a set of possible parses produced by this parser. The structure of each sentence is represented in the form of successive rules. This parser parses sentences with 74% accuracy.

Keywords: Urdu Probabilistic Parser, Urdu PCFG, Results of Urdu Probabilistic Parser.

1. INTRODUCTION

A lot of information about words and syntactic constructions are considered in parsing a human language [1]. Syntactic parsing has been widely studied with the help of different methods, including statistical parsing [2, 3, and 4] and linguistic-based methods [5, 6].

Statistical parsers are gaining popularity every day due to their noticeable accuracy and efficiency. A number of different statistical parsers are already developed by the natural language processing community [7, 8 and 9]. The main idea is to assign probabilities to the grammatical rules. “However, in practice, the probability of a parse tree being the correct parse of a sentence depends not just on the rules which are applied, but also on the words which appear at the leaves of the tree” [10].

Apart from Perso-Arabic script, the morphological system of Urdu is also making this language a highly challenging language because it has inherent grammatical forms and it has borrowed vocabulary from different languages such as Arabic, Persian, Turkish and the native languages of South Asia [11]. It is having a complex grammar with a complex script. The increasing use of Unicode characters and internationalization of software provides opportunities and ways to make research possible in this field [12].

In Urdu, research is going on from different point of views such as creating an Urdu corpus [13, 14] and tagging the Urdu corpus [15]. Researchers have proposed different tagsets for Urdu whose number of tags is ranging from 10 [16] to 350 [17].Now, one of the demanding areas is the parsing of an Urdu

corpus. Considerable amount of work has not yet been done in this direction. An efficient and accurate parser is therefore needed to parse Urdu corpus of natural text. A probabilistic parser is needed to parse Urdu sentences for adequate efficiency and accuracy compared to traditional rule-based parsers. Before developing such a parser, a Probabilistic Context Free Grammar (PCFG) is a pre-requisite.

Recently, a PCFG is developed for Urdu by taking Urdu tagged sentences from different sources [18]. After completing the pre-requisites of our objective, a new algorithm for Urdu probabilistic parser is created [19]. This algorithm is based on the idea of multi-path shift-reduce-strategy [20]. The algorithm is successfully implemented here thus resulting in an efficient Urdu probabilistic parser. This parser is tested by providing different Part-of-Speech tagged Urdu sentences as input. The parser successfully parses most of the sentences. The output from the parser is generated in the form of phrase based successive rules (resulting in a successful parse thus showing the structure of the sentence) with the highest probability. These rules clearly show the structure of the input sentence. Work in different sections of the research paper is organized as follows:

In section 2, the already developed PCFG is discussed briefly. Section 3 provides a view of the developed algorithm for Urdu Probabilistic parser and shows the implementation steps. Section 4 confirms the success of the parser by providing the results of the parser. Section 5 throws light on conclusion and future work.

2. URDU PROBABILISTIC CONTEXT FREE GRAMMAR

The sentences in the tagged text, available on the website of Center for Research in Urdu Language Processing (CRULP) under Urdu-Nepali-English Parallel Corpus project, are mostly long and ambiguous. These sentences are thus complex from parsing point of view. Therefore, apart from taking some complex sentences from the tagged corpus by CRULP (www.crulp.org), some additional data was also collected. Specifically, the focus was on Urdu aqwaal-e-zareen, mazameen and mini-kahanian written by famous authors such as Saadat Hassan Minto, Ibne Inshah and Pitras Bukhari.

Text was POS tagged by utilizing the annotator provided by CRULP. The tagset with 46 tags developed by CRULP as part of a project for developing Urdu-Nepali-English parallel corpus was used for text tagging. After acquiring the tagged data, rules were developed by utilizing the chart parser [21]. Context Free Grammar (CFG) for Urdu was thus developed.

A PCFG is a CFG where each production is assigned a probability. This probability is assigned to each rule by the ratio of the number of occurrences of a rule to the total number of occurrences of that particular phrase. Probabilities were assigned to the rules in the CFG to obtain Urdu PCFG [18].

Part of the table showing Urdu PCFG, with 127 rules is given below:

Rules	**Probabilities**
S →NP VP	0.9637
S →PP VP	0.0167
S →PP NP	0.0019
S →VP NP	0.0109

3. THE ALGORITHM AND ITS IMPLEMENTATION

The two most common types of parsing are top-down and bottom-up, though there are parsing algorithms that are of other types and there are some that are a combination of these two [22]. One such hybrid type is left-corner parsing. In left-corner parsing, top-down processing is combined with bottom-up processing to avoid going in a wrong way that may result (sometimes) when purely top-down or bottom-up technique is used.

A top-down parser starts its processing with the ‘start symbol’ S (sentence) and expands it by applying the productions until the desired string is reached. In bottom-up parsing, the sequence of symbols are

taken and compared to the right hand side of the rules. So the parser starts with the words (from the bottom) and attempts to build up a tree from the words until S is obtained [23].

Multi-path shift-reduce parsing (bottom-up parsing) keeps multiple transition paths during decoding [20]. It allows several best derived states after each expansion. The idea of multi-path shift-reduce parsing is used in developing algorithm for Urdu probabilistic parser. This idea is developed as part of this work. The above mentioned authors have only discussed theoretical concepts in their paper. There is no clue about the type of data structure(s) that is used in the implementation of the parser. In this research work, different data structures are used and the algorithm is implemented. Two different methodologies for rule storage are compared.

The special features of this work are:

a. A structured array is used for storing the results of multiple parses of the same string, where at each location there is:
    1. Identification whether to take shift action or reduce action.
    2. A separate input buffer.
    3. A separate processing stack.
    4. A separate output stack.
    5. A variable for calculating the total probability of the parse in the cell of the array.
b. The output goes to output stacks.

The algorithm for Urdu probabilistic parser [19], implemented here has four parts, i.e. a, b, c and d. Part a is the main algorithm whereas parts b, c and d are sub-algorithms. Algorithm a is the main algorithm for Urdu probabilistic parser. Algorithm b shows the procedure for checking/comparing the input on the top of the processing stack with the right hand side of the rules. Algorithm c is used for copying items onto a new processing stack. Algorithm d is used for finding the highest probability.

The algorithm discussed above is implemented using C# and XML. The parser produces a single representation, if the sentence is syntactically unambiguous. First, each configuration is represented as a rule and each rule is assigned a weight according to how often that configuration appears. These weighted rules are then used in parsing by calculating the total probability of the rules that are used in parsing. The parse with the highest probability is selected.

A user-friendly environment is provided for communication where a user inputs an Urdu file in the form of a collection of Part-of-Speech tagged sentences. The parser parses the text (by creating 200 stacks for each sentence) and produces the structure of each sentence in the form of successive rules as an output. In the beginning, rules were read from the text file but the system was unable to restrict the already used rules in previous stacks to be copied again in the new stacks within the same step. To solve this problem, rules are now read from the database table instead of a text file. Here identifiers are assigned to each rule so that the next stack will not use the already used rule again within the same step.

4. RESULTS

The structure of a sentence is shown by displaying the sequence of phrases that are used one after another in that particular sentence. A total of 100 sentences 22 long (having more than 10 words in a sentence) and 78 short are given as input to the parser. The parser parsed successfully 74 sentences. It failed to parse the remaining 26 sentences. These 26 sentences were kept separately and were carefully examined. Out of these 26 sentences, 15 sentences are long sentences. The remaining 11 sentences are short but ambiguous which is the reason for their unsuccessful parsing. The results obtained here cannot be compared with some other research in Urdu, because up to the knowledge of the authors, it is the first probabilistic parser developed for Urdu. The results from the parser, after processing the following POS tagged sentence, are shown in FIGURE 1:

```
<s>
 دوستوں<w POS= “NN”/>
 سے<w POS= “CM”/>
 محروم<w POS= “JJ”/>
 ہو<w POS= “VB”/>
 گیا<w POS= “AUXA”/>
 ہوں<w POS= “AUXT”/>
</s>
```

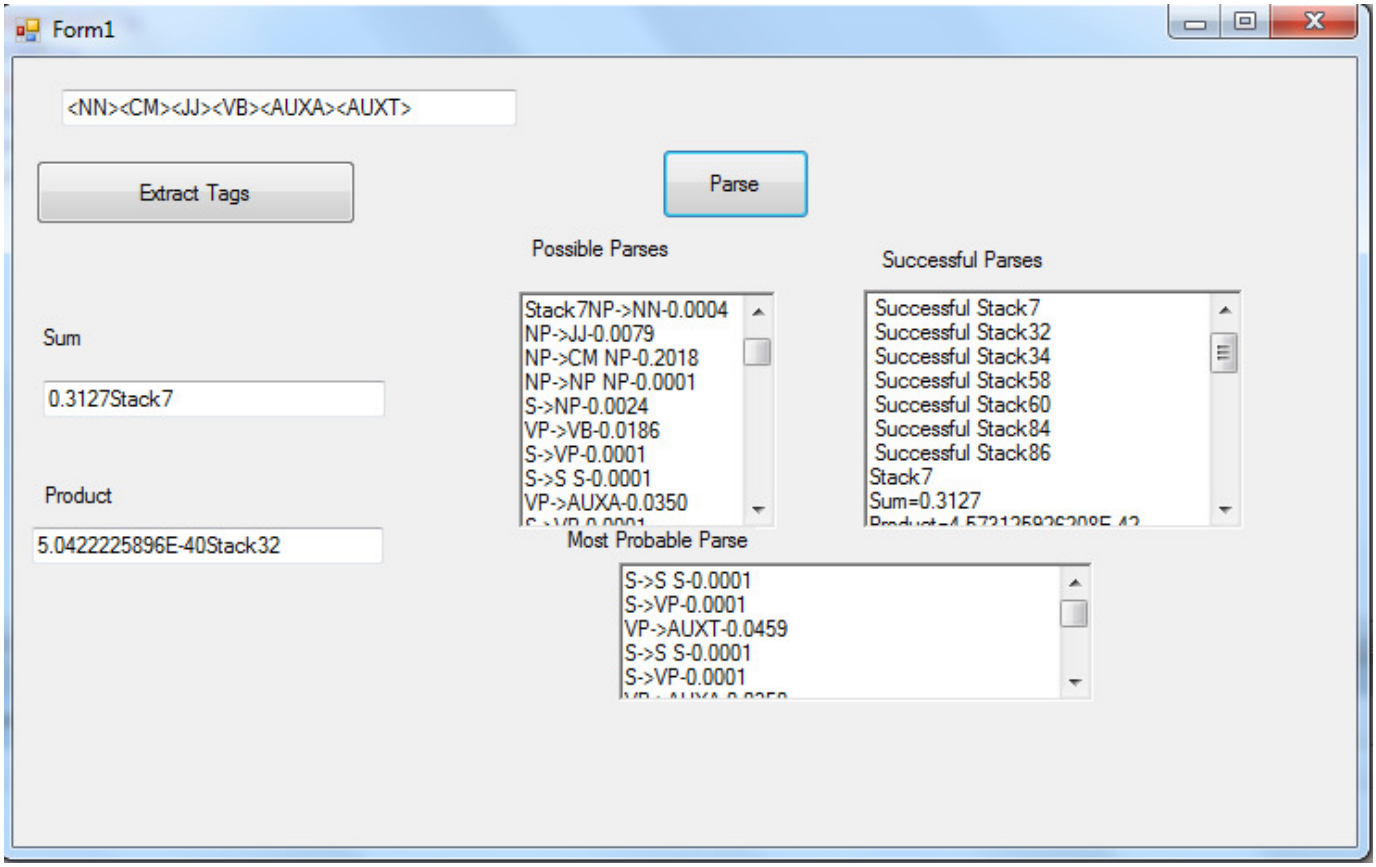

FIGURE 1: Complete output of a successfully parsed sentence.

Tags can be extracted from the sentence, when the button named “Extract Tags” is clicked (shown in FIGURE 1). When the button “Parse” is clicked, this parser provides structure of the sentence in the form of rules in four steps. Firstly, all possible parses of the processed sentence are shown under the heading “Possible Parses”. Here even unsuccessful parses are displayed. The stacks so far created and rules used are displayed. In case of a successful parse “Sc” is displayed at end of the rules in a stack showing that this particular stack is having a successful parse. Secondly, successful parses are shown under the heading “Successful Parses”. A complete list of successful parses with the stack number is provided here along with figures for the sum of probability and product of probability of the rules used. Finally, stack with the highest probability is considered as the correct parse of the sentence. The successive rules are displayed under the heading “Most Probable Parse”. One can easily draw a parse tree for the sentence from these successive rules. An example of one such sentence based, on the output by the Urdu probabilistic parser, using the POS tagged text below, is provided in FIGURE 2.

Successive rules provided by Urdu probabilistic parser.

S →NP-0.0024
NP →NNP NP-0.0688
NP →NNP NP-0.0688
NP →NNPC- 0.0051

<s>
<w POS= “NNP”>محمّد</w>
<w POS= “NNP”>اسد</w>
<w POS= “NNPC”>اللہ</w>
</s>

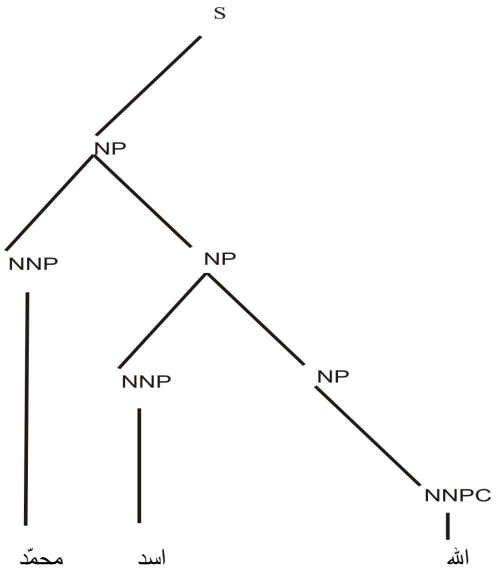

FIGURE 2: Parse tree for the POS tagged text.

In FIGURE 2, the parse tree is showing the structure of the sentence by utilizing the successive rules provided as output by Urdu probabilistic parser. If a sentence cannot be parsed successfully then “No successful parse” message is displayed in “Most Probable Parse” section as shown in FIGURE 3.

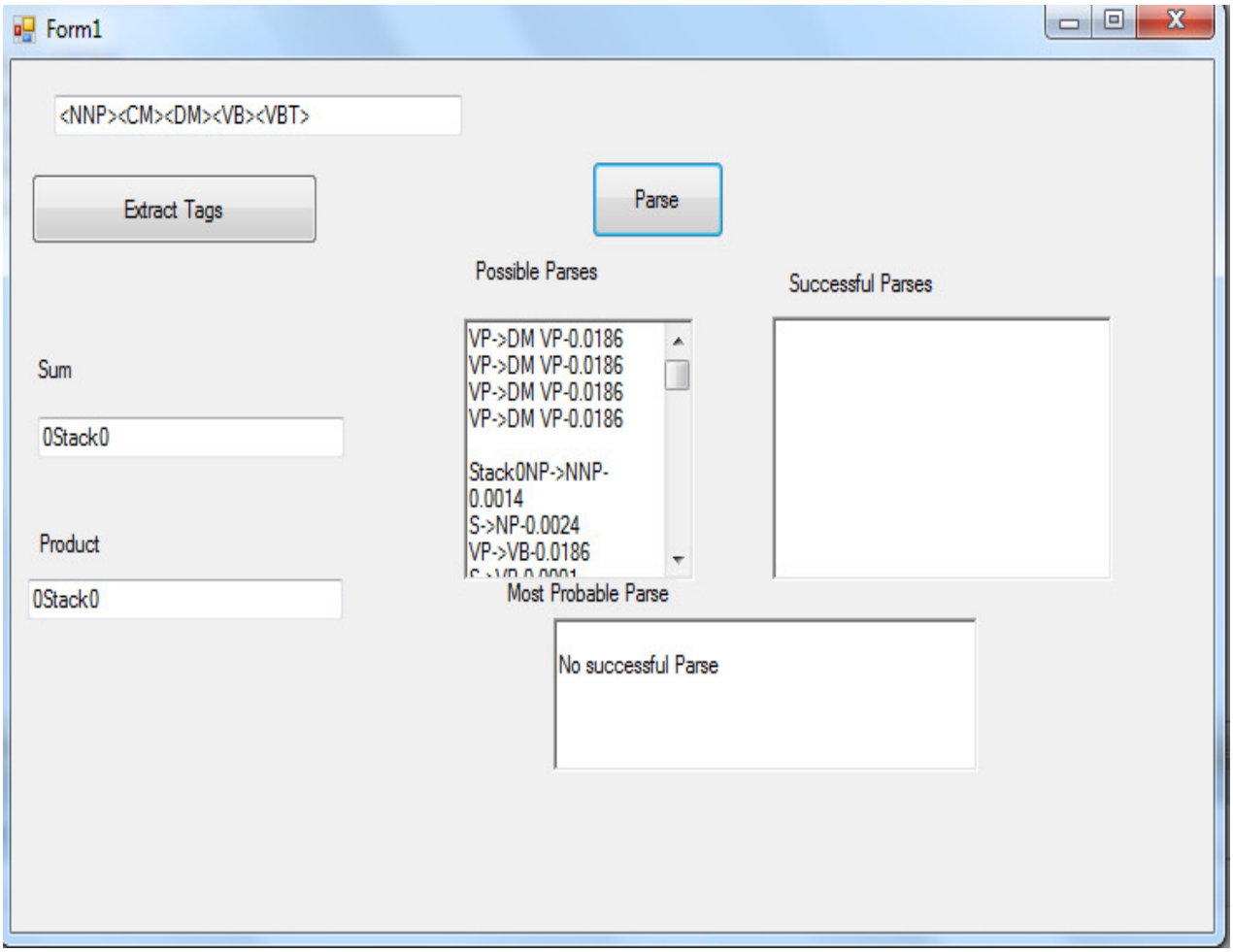

FIGURE 3: Sentence is not parsed successfully.

A section of the table showing the successfully parsed sentences by the Urdu probabilistic parser is provided in TABLE 1.

Sentence	Successful parse
<s> <w POS="NNP">محمّد</w> <w POS="NNP">اسد</w> <w POS="NNPC">اللہ</w> </s>	NP →NNPC- 0.0051 NP →NNP NP-0.0688 NP →NNP NP-0.0688 S →NP-0.0024 Highest probability= 0.1451
<s> دوستوں<w POS="NN"/> سے<w POS="CM"/> محروم<w POS= "JJ"/> ہو<w POS="VB"/> گیا<w POS="AUXA"/> ہوں<w POS="AUXT"/> </s>	NP →NN-0.0004 NP →JJ-0.0079 NP →CM NP-0.2018 NP →NP NP-0.0001 S →NP-0.0024 VP →VB-0.0186 S →VP-0.0001 S →S S-0.0001 VP →AUXA-0.0350 S →VP-0.0001 S →S S-0.0001 VP →AUXT-0.0459 S →VP-0.0001 S →S S-0.0001 Highest probability= 0.3127

TABLE 1: Output of the parser providing the rules

5. CONCLUSIONS AND FUTURE WORK

The parser mostly parses short sentences correctly. While parsing a long sentence (when the number of words exceeds 10), the parser usually fails. The parser takes a lot of time while parsing a highly ambiguous sentence. The parser consumes more memory by creating a number of stacks for highly ambiguous sentences. Sometimes, it fails to parse an ambiguous sentence. The reason is ambiguous grammar. By disambiguating the grammar the performance of the parser can be improved a lot.

Mostly this Urdu probabilistic parser can parse ambiguous sentences successfully. The chart parser, that was used in testing phase, usually failed to parse a sentence when a rule with left recursion (NP →NP NP) was required for parsing. This Urdu probabilistic parser can successfully parse a sentence even by using the rules with left recursion.

While considering the 100 tested sentences by Urdu probabilistic parser, the success rate of the parser is 74%. Usually short sentences (i.e. less than 10 words in a sentence) are parsed successfully. It may be concluded that to a large extent the success of this parser is dependent on the length of the sentence. The shorter the sentence, the higher is the chance of being successfully parsed by the parser.

For further improvement, the number of stacks is increased from 200 to 500. The performance of the parser in both the cases is compared. It is observed that increase in the number of stacks has a very small effect on the accuracy (almost negligible) of the parser. The speed of the software is decreased a lot as the software has to create 500 stacks now for each sentence. Considering speed as an important factor, parser with 200 stacks is more suitable.

The use of an Urdu tagger (so that it will tag the sentences automatically at the moment they are entered) will increase the efficiency of the parser. The use of a tree generator that will create a parse tree from the rules will make the software more efficient. The use of different data structures such as linked lists, instead of stacks and arrays will decrease the memory requirement of the algorithm.

REFERENCES

[1] B. Sagot and E. de la Clergerie. "Error Mining in Parsing Results". Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, Sydney, 2006, pp. 329–336.

[2] E. Charniak. "Statistical Parsing with a Context-Free Grammar and Word Statistics". Proceedings of the 14th National Conference on Artificial Intelligence, MIT Press, 1997.

[3] M. J. Collins. "A New Statistical Parser Based on Bigram Lexical Dependencies". Proceedings of ACL 96, 1996.

[4] D. M. Magerman. "Statistical Decision- Tree Model for Parsing". Proceedings of the 33rd Annual Meeting of the ACL, 1995.

[5] S. Abney. "Partial Parsing via Finite-State Cascades". John C. Ed. Workshop.Robust Parsing (ESSLLI'96), 1996, pp. 08-15.

[6] S. A'it-Mokhtar and J-P. Chanod. "Incremental Finite-State Parsing". Proceedings of the 5th Conference on Applied Natural Language Processing, 1997.

[7] M .J. Collins. "Head-driven statistical models for natural language parsing". Ph.D. thesis, 1999.

[8] E. Charniak."A maximum-entropy inspired parse", Proceedings of the First Meeting of The North American Chapter of the Association for Computational Linguistics, Seattle, WA, 2000, pp. 132–139.

[9] S. Petrov, L. Barrett, R. Thibaux. and D. Klein. "Learning accurate, compact and interpretable tree annotation". Proceedings of ACL, 2006.

[10] C. Lakeland and A. Knott. "Implementing a Lexicalized Statistical Parser". Proceedings of the Australasian Language Technology Workshop, Macquarie University, Sydney, 2004.

[11] M. Humayoun, H. Hammarström and Ranta. "Urdu Morphology, Orthography and Lexicon Extraction". CAASL-2, the Second Workshop on Computational Approaches to Arabic Script-based Languages, LSA 2007, Linguistic Institute, Stanford University, 2007.

[12] M. Humayoun. "Urdu Morphology, Orthography and Lexicon Extraction". Master thesis, Department of Computer Science and Engineering, Chalmers University of Technology and Goteborg University, 2006.

[13] H. Samin, S. Nisar and S. Sehrai. Project: "Corpus Development". BIT thesis, Department of Computer Science, University of Peshawar, Peshawar, Pakistan, 2006.

[14] D. Becker and K. Riaz. "A Study in Urdu Corpus Construction". Proceedings of the 3rd Workshop on Asian language resources and international standardization, 2002.

[15] W. Anwar, X. Wang, Luli and Wang. "Hidden Markov Model Based Part of Speech Tagger for Urdu". Information Technology, 2007, *Vol.6*, pp. 1190-1198.

[16] R. L. Schmidt. "Urdu: an essential grammar". Rout-ledge, London, UK, 1999.

[17] A. Hardie. “The computational analysis of morph syntactic categories in Urdu”. Ph.D thesis, Lancaster University, 2003a.

[18] N. Mukhtar, M. A. Khan and F. Zuhra. “Probabilistic Context Free Grammar for Urdu”, Linguistic and Literature Review (LLR), 2011, 1(1).

[19] N. Mukhtar, M. A. Khan and F. Zuhra. “Algorithm for developing Urdu Probabilistic Parser”, International journal of Electrical and Computer Sciences IJECS-IJENS 12(3), pp. 57-66, 2012.

[20] W. Jiang, H. Xiong and Q. Liu. "Multi-Path Shift-Reduce Parsing with Online Training".CIPS ParsEval, Beijing, November, 2009.

[21] F. Zuhra. "Pashto Chart parser". Unpublished paper, Department of Computer Science, University of Peshawar, Pakistan, 2010.

[22] G. Sandstrom. Survey paper, "Parsing and Parallelization", 2004.

[23] B. M. Bataineh and E. A. Bataineh. "An Efficient Recursive Transition Network Parser or Arabic Language".Proceedings of the World Congress on Engineering, London, U.K, 2009.