1.5em 0pt

Dependency Parsing for Bangla Text

*Note: Sub-titles are not captured in Xplore and should not be used

Dr. K. M. Azharul Hasan
Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh
az@cse.kuet.ac.bd

Shaikh Prothomaa Binte Minhaz
Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh
prothomaaminhaz@gmail.com

***Abstract*—A fundamental task in natural language processing (NLP) called dependency parsing involves analyzing the grammatical structure of sentences by detecting the syntactic connections between words. This study examines the difficulties and methods for performing dependency parsing for Bangla (Bengali) text, a language with complex morphology and distinctive syntactic properties. The article addresses the value of dependency parsing in capturing the linguistic subtleties of Bangla sentences and their applications in various NLP tasks. In this work, a method is proposed for developing dependency parsing on Bangla text by using a graph-based approach. Here the parsing tree is generated from a directed graph of Bangla text input. This method helps to enrich the Bangla language linguistic resources and annotated corpora that help to use language globally. The resultant tree is also evaluated using evaluation metrics.**

***Index Terms*—Bangla, dependency parsing, maximum spanning tree, probability**

I. Introduction

Dependency parsing is a crucial component of natural language processing (NLP) that helps understand the syntactic structure of sentences by identifying word relationships. However, its application to languages with complex morphologies like Bangla presents unique challenges and opportunities. Bangla, spoken by millions worldwide, has a rich linguistic heritage with intricate grammatical rules and flexible word order, making it an intriguing subject for computational linguistics research. The importance of dependency parsing in Bangla text processing is significant, as it forms the basis for various NLP applications such as machine translation, sentiment analysis, information retrieval, and question answering. However, the development of efficient and accurate dependency parsers tailored specifically for Bangla text has been hindered by the lack of annotated corpora, linguistic resources, and research focus compared to more widely studied languages. This thesis work aims to contribute to this emerging field by providing a comprehensive analysis and implementation of dependency parsing techniques for Bangla text. By leveraging advances in computational linguistics and machine learning, the researchers aim to develop robust and accurate dependency parsers that can effectively handle the complexities of Bangla syntax. This research not only advances dependency parsing for Bangla but also paves the way for broader applications of NLP in the Bangla-speaking world.

Identify applicable funding agency here. If none, delete this.

II. Related Works

Utpal Garain et al. (2014) have described a Grammar-driven dependency parsing for Bangla. Bangla language has a free-word order nature which makes parsing a sentence difficult. The Paninian grammatical model is used to solve this difficulty by simplifying the complex and compound sentences and then parsing the simple sentences by satisfying the Karaka demands of the Demand Groups (Verb Groups), and finally rejoining such parsed structures with appropriate links and Karaka labels. The parser has been trained with a Treebank of 1000 annotated sentences and then evaluated with un-annotated test data of 150 sentences. Arnab Dhar et al. (2012) described a two-stage dependency parser for Bangla. In the first stage, authors build a model using a Bangla dependency Treebank released in ICON 2009 and subsequently, this model is used to build a data-driven Bangla parser [1]. In the second stage, constraint-based parsing has been used to modify the output of the data-driven parser. This implements the Bangla-specific constraints with the help of demand frames of Bangla verbs. In the data-driven module, authors use Covington's algorithm as implemented in MaltParser by Nivre (2006, 2007, 2009) for statistically annotating the dependency relations in Bangla sentences.

Urmi Ghosh et al. (2019) develop a code-mixing (CM) NLP system that has significantly gained importance in recent times due to an upsurge in the usage of CM data by multilingual speakers [5]. The authors present a rule-based system to computationally generate a synthetic code-mixing treebank for Bengali and English (Syn-BE) which is used to further improve the accuracy of dependency parser. They use a dataset of 500 Bengali-English tweets annotated under the Universal Dependencies scheme.

Sanjay Chatterji et al. in the paper "Grammar Driven Rules for Hybrid Bengali Dependency Parsing" describe a hybrid approach for parsing Bengali sentences based on the dependency tagset and Treebank released in the ICON 2009 tool contest. A data-driven dependency parser is considered a baseline system. Some handcrafted rules are identified based on the error patterns in the output of the baseline system.

Phani Gadde et al. present a data-driven dependency parsing strategy that makes use of sentence-specific clause information to enhance the parser efficiency. A partial parser is used to automatically incorporate the clausal information. We use a modified version of MSTParser to demonstrate the experiments on Hindi, a morphologically rich free-word-order language. We conducted all of our studies using the parsing contest data from ICON 2009. Our unlabeled attachment and labeled attachment accuracy improvements over the baseline parsing accuracy were 0.87% and 0.77%, respectively.

III. Dependency Parser Generation

A. Introduction

The technique of a dependency parsing algorithm describes the sequential steps the algorithm takes to examine a sentence and build a dependency tree that represents the syntactic relationships between words. Dependency parsing algorithms can be categorized into transition-based and graph-based approaches. Here is a general description of the proposed methodology of the graph-based approach and graphical representation of proposed methodology in figure 1.

Step 1: Take user input as a paragraph or a sentence.
Step 2: If the input is a paragraph then sentence separation occurs according to ending symbols.
Step 3: Each word of a sentence converts into tokens and identifies the POS of each token in a sentence.
Step 4: Create a fully connected directed graph using the tokens with a root node that has only outgoing edges to each node of the graph.
Step 5: Assign weights to each edge using transition and emission probabilities added with some factor.
Step 6: Execute Edmond's algorithm to get the max spanning tree. It provides nodes with a single parent.
Step 7: According to the transition probability dictionary calculate the accuracy of each sentence.

B. Terminologies

Separation of sentences: In this work, the user can input a paragraph. So each sentence needed to be separated. Sentences are separated by using " ।", "! ", "? " symbols for Bangla text and English text ".", "!", "?" symbols are used. For example,
Input paragraph: প্রাকৃতিক রূপবৈচিত্রে ভরা আমাদের এই বাংলাদেশ। এই দেশে পরিচিত অপরিচিত অনেক পর্যটক-আর্কষক স্থান আছে। এর মধ্যে প্রত্নতাত্ত্বিক নিদর্শন, ঐতিহাসিক মসজিদ এবং মিনার, পৃথিবীর দীর্ঘতম প্রাকৃতিক সমুদ্র সৈকত, পাহাড়, অরণ্য ইত্যাদি অন্যতম। এদেশের প্রাকৃতিক সৌন্দর্য পর্যটকদের মুগ্ধ করে। বাংলাদেশের প্রত্যেকটি এলাকা বিভিন্ন স্বতন্ত্র বৈশিষ্ট্যে বিশেষায়িত।
After separating into sentence,
প্রাকৃতিক রূপবৈচিত্রে ভরা আমাদের এই বাংলাদেশ।
এই দেশে পরিচিত অপরিচিত অনেক পর্যটক-আর্কষক স্থান আছে।
এর মধ্যে প্রত্নতাত্ত্বিক নিদর্শন, ঐতিহাসিক মসজিদ এবং মিনার, পৃথিবীর দীর্ঘতম প্রাকৃতিক সমুদ্র সৈকত, পাহাড়, অরণ্য ইত্যাদি অন্যতম।
এদেশের প্রাকৃতিক সৌন্দর্য পর্যটকদের মুগ্ধ করে।
বাংলাদেশের প্রত্যেকটি এলাকা বিভিন্ন স্বতন্ত্র বৈশিষ্ট্যে বিশেষায়িত।
Lexical analysis: Lexical analysis, also known as tokenization is a crucial phase because it divides the input phrase into individual words (tokens) and gives each one linguistic information that will be used to establish the syntactic relationships between them in the dependency tree. Parts of speech (POS) detection: Define the POS of each word in a sentence. Bangla language is a rich morphological language. So to detect the POS of each word while considering morphological analysis. For Bangla text POS tagging and tokenization, BNLP is used. For English text POS tagging and tokenization, NLTK is used. The tokenization and POS detection are done using BNLP. For example,
Sentence 1:প্রাকৃতিক রূপবৈচিত্রে ভরা আমাদের এই বাংলাদেশ।
After tokenization and POS detection: [(প্রাকৃতিক, 'JJ'), (রূপবৈচিত্রে, 'NC'), (ভরা, 'NC'), (আমাদের, 'PPR'), (এই, 'DAB'), (বাংলাদেশ, 'NC'), (।, 'PU')]
Edge labeling: Make a fully connected directed graph with words as the nodes and edges G = (V, E) standing in possible word dependency relationships. Assign the weights to the edges based on linguistic characteristics, transition and emission probabilities which are from training annotated corpus. The transition probability refers to the probability of transitioning from one hidden state to another. The emission probability refers to the probability of observing a particular output (emission) symbol from a given hidden state. Here also add a 'root' node which only has outgoing edges to all nodes. In table 1 weights of outgoing edges of "root" is shown and in figure 2 the edges are shown for 'root' and 'রূপবৈচিত্রে' nodes. Like this, all other edges for all nodes need to be created for generating a fully connected graph.

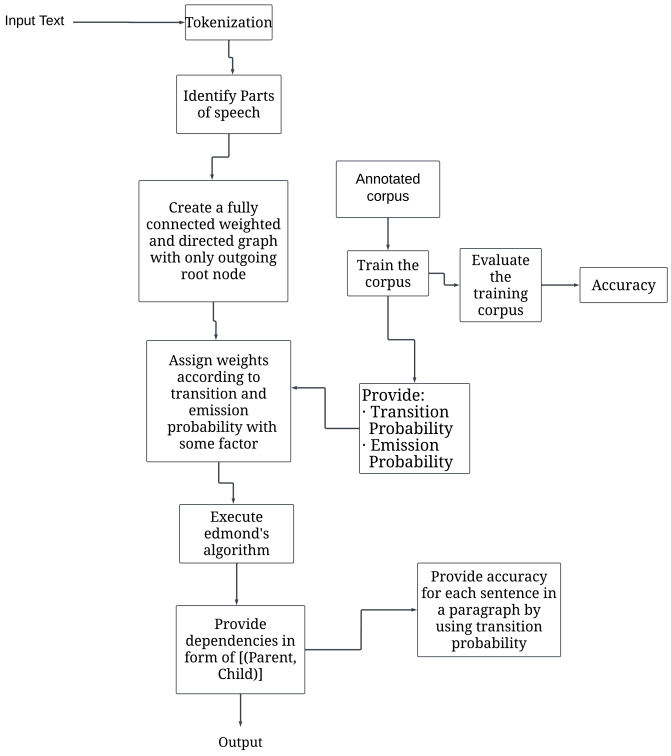

Fig. 1. Proposed Methodology

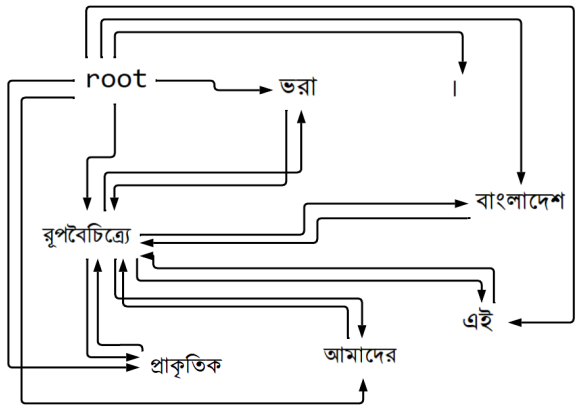

Fig. 2. Graph by tokens

TABLE I
GRAPH AFTER ASSIGNING WEIGHT

From	*To*	*Weight*
root	প্রাকৃতিক	0.0784313725490196
	রূপবৈচিত্র্যে	0.7209302325581395
	ভরা	0.31794871794871793
	আমাদের	7.564102564102565
	এই	0.275
	বাংলাদেশ	0.625
	।	0.6377171215880894

Tree generation: Implement the maximum spanning tree (MST) from the directed graph. To implement MST, we use the Chu-Liu-Edmonds algorithm. In figure 3, there is a graphical representation of output.

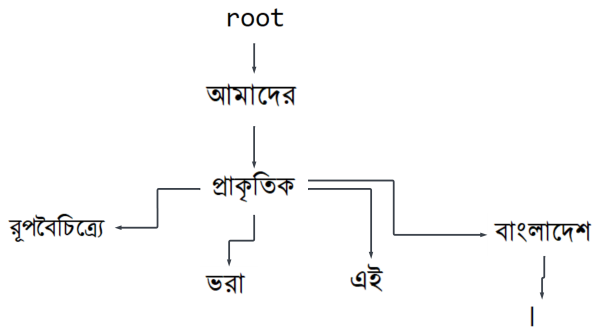

Fig. 3. Graphical representation of dependencies in Bangla sentence

Flask application framework: Flask is a lightweight WSGI web application framework in Python. It is used for user input and generates output for users.

C. Proposed Algorithm

The proposed algorithm is Edmond's algorithm which is used for finding the maximum spanning tree in a fully connected directed graph G=(V, E) where V is the set of vertices (nodes) representing words in a sentence and E is the set of directed edges representing possible syntactic dependencies between words. This graph has a root node with only outgoing edges. Each edge in the graph is associated with a weight that represents the likelihood or cost of the dependency relation between the connected words. In this work, the weight is determined using transition and emission probability with some factor multiplication. The output of the Edmonds algorithm is the maximum spanning tree (MST) of the input graph, which represents the most likely syntactic structure or dependency parse tree of the sentence.

Algorithm 1 Max Spanning tree Algorithm

function MST $(G = (V, E), \text{root}, \text{score})$
F $\leftarrow$ []
T$'$ $\leftarrow$ []
score$'$ $\leftarrow$ []
for each $v \in V$ **do**
  bestInEdge $\leftarrow \arg\max_{(e=(u,v)\in E)}$ score$[e]$
  F $\leftarrow$ F $\cup$ bestInEdge
  for each $e = (u, v) \in E$ **do**
    score$'[e] \leftarrow$ score$[e]$ $-$ score[bestInEdge]
    if $T = (V, F)$ is a spanning tree **then**
      return T
    else
      C $\leftarrow$ a cycle in F
      G$'$ $\leftarrow$ Contract(G, C)
      T$'$ $\leftarrow$ MST(G$'$, root, score$'$)
      $T \leftarrow$ Expand(T$'$, C)
      return T
    end if
  end for
end for
function Contract (G, C) **return** contracted graph
function Expand (T, C) **return** expanded graph

IV. Implementation, Results and Discussions

A. Experimental Setup

The computer is installed Windows 10 OS. Python programming language is used to implement Edmond's algorithm, greedy decoding. Jupyter Notebook is used as IDE. Python module version is python 3.10.4. For the NLP task NLTK is used and for Bangla POS tagging BNLP is used. Flask which is a Python web framework used to create web applications. Both user input and output trees were shown using Flask. Flask 3.0.2 is used.

B. Evaluation

Evaluation is the measurement that is used to assess the performance or effectiveness of a system, model, algorithm

or process. For the evaluation of this thesis work, the estimated accuracy of each generated tree is measured. For this computation, the transition probability is used. At first sort the transition probabilities in descending order of the same parent node that occurs in a current sentence of the paragraph. Then accuracy of each edge is summed up to get the accuracy of a sentence. If any transition probability is absent in the dataset that is held in the current sentence, this system adds a factor for the accuracy of that edge.
Estimated accuracy: [0.5718649499155655, 0.5633383105308238, 0.7518373783079666, 0.3970933828076685, 0.4707541478129713]
In figure 2, (আমাদের, প্রাকৃতিক), ('root', আমাদের), (প্রাকৃতিক, রূপবৈচিত্রে), (বাংলাদেশ, ।) these 4 edges are correct and (প্রাকৃতিক, ভরা), (প্রাকৃতিক, এই), (প্রাকৃতিক, বাংলাদেশ)these 3 edges are incorrect. So estimated accuracy is 0.5718649499155655.

C. Dataset

The annotated corpus is used. The dataset is self-developed and there are fifty unique sentences with 33 unique states. States are 'RDF', 'NC', 'JJ', 'CCD', 'NV', 'PU', 'NP', 'VM', 'JQ', 'PP', 'PPR', 'VAUX', 'NST', 'PRL', 'AMN', 'CCL', 'CSB', 'DAB', 'CX', 'EX', 'MD', 'VB', 'NNS', 'VBG', 'WP', 'PRP', 'VBD', '.', '"', 'ALC', 'PWH', 'PRF', 'NNP'. Accuracy Of Dataset 0.7692307692307693 (77%). Another dataset is designed which store the estimated maximum spanning tree of few sentences.

D. Implementation and Results

Bangla corpus which is self-developed is used for assigning weights of directed edges of the graph, this system uses transition and emission probability with some factor. These transition and emission probabilities are acquired from the corpus. Then the graph passes to the Edmonds algorithm that provides the maximum spanning tree of the graph where the sum of weights of all edges is maximum than other spanning trees of the graph. The resultant tree indicates the dependencies of the words in a sentence. The output segment shows the dependencies as a parent-child relationship in the list i.e. [(Parent, Child)] in table 2.

Output dependencies: [[(আমাদের, প্রাকৃতিক), (root, আমাদের), (প্রাকৃতিক, রূপবৈচিত্রে), (প্রাকৃতিক, ভরা), (প্রাকৃতিক, এই), (প্রাকৃতিক, বাংলাদেশ), (বাংলাদেশ, '।')], [(root, আছে), (root, এই), ('।', পরিচিত), ('।', অপরিচিত), ('।', অনেক), ('।', '-'), ('।', আর্কষক), (আছে, '।'), (এই, পর্যটক), (এই, স্থান), (এই, দেশে)], [(root, এর), (root, এবং), (এর, নিদর্শন), (এর, মসজিদ), (এর, মিনার), (এর, পৃথিবীর), (এর, সমুদ্র), (এর, পাহাড়), (এর, অরণ্য), (অরণ্য, ইত্যাদি), (ইত্যাদি, মধ্যে), (ইত্যাদি, প্রত্নতাত্ত্বিক), (ইত্যাদি, ','), (ইত্যাদি, ঐতিহাসিক), (ইত্যাদি, ','), (ইত্যাদি, দীর্ঘতম), (ইত্যাদি, প্রাকৃতিক), (ইত্যাদি, সৈকত), (ইত্যাদি, ','), (ইত্যাদি, ','), (ইত্যাদি, অন্যতম), (অন্যতম, '।')], [(root, এদেশের), (root, করে), ('।', প্রাকৃতিক), (এদেশের, সৌন্দর্য), (এদেশের, পর্যটকদের), (এদেশের, মুগ্ধ), (করে, '।')], [(root, প্রত্যেকটি), (এলাকা, বিভিন্ন), (বাংলাদেশের, এলাকা), (বাংলাদেশের, স্বতন্ত্র), (বাংলাদেশের, বৈশিষ্ট্যে), (প্রত্যেকটি, বাংলাদেশের), (বৈশিষ্ট্যে, বিশেষায়িত), (বিশেষায়িত, '।')]]

TABLE II
DEPENDENCIES OF THE SENTENCE: প্রাকৃতিক রূপবৈচিত্রে ভরা আমাদের এই বাংলাদেশ।

Token	*Head*	*Children*
প্রাকৃতিক	আমাদের	[রূপবৈচিত্রে, ভরা, এই, বাংলাদেশ]
রূপবৈচিত্রে	প্রাকৃতিক	[]
ভরা	প্রাকৃতিক	[]
আমাদের	root	[প্রাকৃতিক]
এই	প্রাকৃতিক	[]
বাংলাদেশ	প্রাকৃতিক	[।]
।	বাংলাদেশ	[]

1) Quantitative Evaluation : This paragraph represents the quantitative evaluation of this work. The accuracy of this work is measured by the self-developed corpus of 51 sentences. The accuracy of the procedure is 0.679950435052476 which means the accuracy is about 68%.

2) Qualitative Evaluation : Here the best-case and worst-case results are shown below:
In best-case example, the sentence is "কচুরিপানা একটি জলজ উদ্ভিদ।". Figure 4 is the best-case example. Here the generated dependencies of a sentence are completely similar to the targeted dependencies.

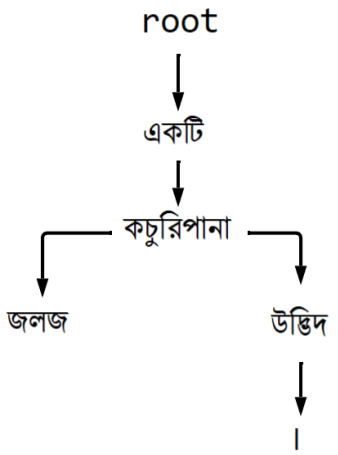

Fig. 4. Best-case example

In worst-case example, the sentence is "আমার পছন্দের খেলা ফুটবল।". Figure 5 is the generated tree by the proposed method and figure 6 is the targeted tree for the sentence আমার পছন্দের খেলা ফুটবল।. Here only (ফুটবল, '।') this single edge is the same for both generated and target tree. So in this particular case accuracy is 0.2 only that is 20%.

V. CONCLUSION

In conclusion, the intricate details of adapting dependency parsing techniques to the distinctive linguistic features of the

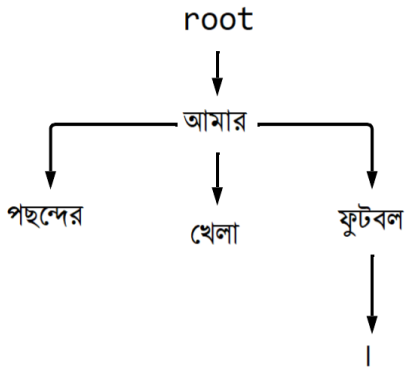

Fig. 5. Generated tree in worst-case

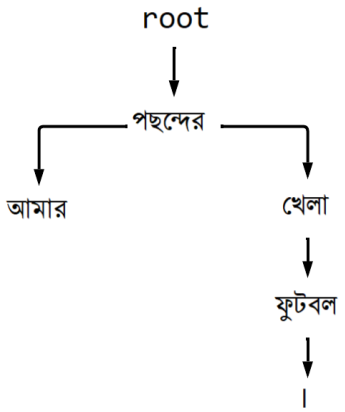

Fig. 6. Targeted tree in worst-case

Bangla language have been examined in the thesis research on ”Dependency Parsing for Bangla text”. In addition to revealing word relationships and improving our comprehension of the grammar and syntax of the Bangla text, the study has made significant contributions to the syntactic analysis of the Bangla language. The results of this study provide insight into the importance of dependency parsing in Bangla and its wider implications for linguistics and natural language processing. The development of annotated corpora and linguistic resources intended specifically for Bangla dependency parsing has been aided by the thesis research. These resources provide a foundation for future research endeavors and facilitate the advancement of Bangla language technology. The research's outcomes place a strong emphasis on the usefulness of dependency parsing in Bangla. The proper study of syntactic links improves the performance of several natural language processing tasks, including sentiment analysis and machine translation.

REFERENCES

[1] Dhar, Arnab, et al. ”A hybrid dependency parser for Bangla.” Proceedings of the 10th Workshop on Asian Language Resources. 2012.
[2] Das, Arjun, Arabinda Shee, and Utpal Garain. ”Evaluation of two bengali dependency parsers.” Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages. 2012.
[3] Chatterji, Sanjay, et al. ”Grammar driven rules for hybrid bengali dependency parsing.” ICON09 NLP TOOLS CONTEST: INDIAN LANGUAGE DEPENDENCY PARSING (2009): 38.
[4] McDonald, Ryan, et al. ”Non-projective dependency parsing using spanning tree algorithms.” Proceedings of human language technology conference and conference on empirical methods in natural language processing. 2005.
[5] Ghosh, Urmi, Dipti Misra Sharma, and Simran Khanuja. ”Dependency parser for bengali-english code-mixed data enhanced with a synthetic treebank.” Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019). 2019.
[6] Nivre, Joakim. ”Parsing indian languages with maltparser.” Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing (2009): 12-18.
[7] Mannem, Prashanth. ”Bidirectional dependency parser for hindi, telugu and bangla.” ICON09 NLP TOOLS CONTEST: INDIAN LANGUAGE DEPENDENCY PARSING (2009): 49.
[8] Husain, Samar. ”Dependency parsers for indian languages.” Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing (2009).
[9] Kosaraju, Prudhvi, et al. ”Experiments on indian language dependency parsing.” Proceedings of the ICON10 NLP Tools Contest: Indian Language Dependency Parsing (2010): 40-45.
[10] Husain, Samar, et al. ”The ICON-2010 tools contest on Indian language dependency parsing.” Proceedings of ICON-2010 Tools Contest on Indian Language Dependency Parsing, ICON 10 (2010): 1-8.
[11] Das, Amitava, and Sivaji Bandyopadhyay. ”Phrase-level Polarity Identification for Bangla.” Int. J. Comput. Linguistics Appl. 1.1-2 (2010): 169-182.
[12] Khatun, Ayesha, and Mohammed Moshiul Hoque. ”Statistical parsing of Bangla sentences by CYK algorithm.” 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE). IEEE, 2017.
[13] Abhilash, Aswarth, and Prashanth Mannem. ”Bidirectional dependency parser for indian languages.” Proceedings of the ICON10 NLP Tools Contest: Indian Language Dependency Parsing (2010).