information

Article

Bangla DeConverter for Extraction of BanglaText from Universal Networking Language

Md. Nawab Yousuf Ali [1], Md. Lizur Rahman [1,*] and Golam Sorwar [2]

[1] Department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh; nawab@ewubd.edu

[2] School of Business and Tourism, Southern Cross University, Lismore, QLD 4225, Australia; Golam.Sorwar@scu.edu.au

* Correspondence: lizur.sky@gmail.com

Received: 25 August 2019; Accepted: 12 October 2019; Published: 21 October 2019

Abstract: The people in Bangladesh and two states (i.e., Tripura and West Bengal) in India, which is about 230 million of the world population, use Bangla as their first dialect. However, very few numbers of resources and tools are available for this language. This paper presents a Bangla DeConverter to extract Bangla texts from Universal Networking Language (UNL). It explains and illustrates the different phases of the proposed Bangla DeConverter. The syntactic linearization, the implementation of the results of the proposed Bangla DeConverter, and the extraction of a Bangla sentence from UNL expressions are presented in this paper. The Bangla DeConverter has been tested on UNL expressions of 300 Bangla sentences using a Russian and English Language Server. The proposed system generates 90% syntactically and semantically correct Bangla sentences with a UNL Bilingual Evaluation Understudy (BLEU) score of 0.76.

Keywords: Bangla DeConverter; UNL; UNL expression; generation rules; syntactic linearization

1. Introduction

The Universal Networking Language (UNL) [1] is a digital language in the form of a network of semantic words that performs as an intermediate representation to expose and interchange all types of knowledge and information. EnConverter (EnCo) and DeConverter (DeCo) are two vital components of UNL. EnCo changes a native language text into UNL expressions and DeCo transforms them into a target language. Therefore, a UNL system bridges the gap between languages around the world. This paper develops a Bangla DeConverter in producing Bangla texts from UNL. Syntactic linearization, a process of ascertaining a proper order of lexicons/words in generated texts, acts a significant role in the quality of generated output.

Unlike English, Bangla is a free word order language known for its affluent semantical and morphological features similar to Hindi and Punjabi. English is a fixed word order language that follows the subject, verb, and object (SVO) pattern. While the Bangla language is patterned with a subject, object, and verb (SOV) structure, it can also be arranged with VSO and OSV structure.

For conversion of a source language to UNL and from UNL to a target language, EnCo and DeCo tools need to be developed. These tools execute their tasks based on a word dictionary and a set of analysis and generation rules for a given language [1]. UNL provides knowledge and information based on the structure of universal words (UW), attributes of UNL, and relations of UNL. The role of each word is represented by the concepts of UWs and UNL relations. UNL attributes represent the subjective meaning of a sentence [1]. For example, consider the following UNL expressions shown in (1) for the sentence 'The color of the screen has changed from red to green' and the corresponding

UNL graph shown in Figure 1. Here, *obj*_indicates object relation, *src*_for source (initial state) relation, and *gol*_for goal (final state), respectively.

{unl}
obj(change(icl>occur,src>thing,obj>thing,gol>thing).@entry.@present.@complete,colour(icl
>kind>thing,equ>color).@def)
obj(colour(icl>kind>thing,equ>color).@def,screen(icl>surface>thing).@def) (1)
src(change(icl>occur,src>thing,obj>thing,gol>thing).@entry.@present.@complete,red(icl>adj,equ>crimson))
gol(change(icl>occur,src>thing,obj>thing,gol>thing).@entry.@present.@complete,green(icl>adj))
{/unl}

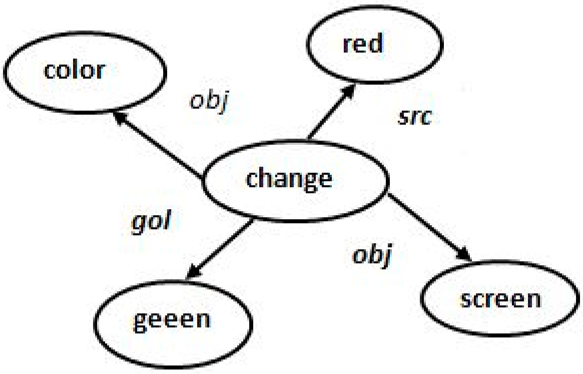


Figure 1. Universal Networking Language (UNL) expression and UNL graph.

In the UNL system [1], the linguistic communication process is administered by two tools: EnConverter (EnCo) [2,3] and DeConverter (DeCo) [3,4]. EnCo interprets a linguistic communication text into UNL expressions and DeCo translates/extracts UNL expressions into a good form of native languages. Each of the tools area unit connected with a word dictionary of linguistic communication and a group of language specific analysis and generation rules. Our paper focuses on extracting Bangla texts from UNL. Thus, we attempted to develop a group of generation rules for attaining our goals.

The paper is organized as follows. Section 2 presents literature review if related areas. The architecture of the Bangla DeConverter is discussed in Section 3. Different phases of Bangla DeConverter along with syntactic linearization issues are detailed in Sections 4 and 5. Syntactic linearization of simple and compound sentences are demonstrated in Section 6. In Section 7, we illustrate experimental results and discussions by extracting a sentence from a UNL expression using some generation rules. Finally, Section 8 includes a summary of the paper with some concluding remarks.

2. Related Works

The structure of UNL–Russian DeCo has been presented by [5]. A DeCo has been developed by [6] for extracting Brazilian Portuguese text from UNL Expression. DeCo for Marathi and Hindi has been proposed by [7]. DeCo for converting UNL to Panjabi has been presented by [8]. A system 'ARIANE-G5' has been proposed by [9]. A DeCo has been proposed by [10] for converting UNL to Chinese. They addressed the drawbacks of the DeCo developed by the UNL center. An Arabic DeCo has been introduced by [11] involving the mapping of morphological analysis, lexical generation, word order, and the relation of the words for semantic meanings. In [12], the authors have designed the architecture of a Nepali DeCo. They have highlighted two major modules, morphological generation and syntactic linearization in their architecture. A DeCo for Hindi text 'HinD' has been presented by [13]. They have indicated the complex rule format in writing analysis and generation rules, non-availability of source codes, and slow speed of DeCo tools provided by the UNDL

Foundation. Their system includes word selection, morphological analysis of the lexicons, word insertion, and syntactic linearization. So far, no attempt has been taken for designing the architecture for Bangla DeCo. These concerns motivated us to exhibit a tool for Bangla DeConverter.

3. Architecture of Bangla DeConverter

Figure 2 shows the Architecture of the Bangla DeCo. The structure of a Bangla sentence is similar to that of Hindi and Punjabi sentences [8]. The architecture is based on language-dependent and language-independent components during the text extraction process. A parser is a tool that converts the UNL expressions, and, based on the output, it creates a node. During the selection of the language unit, the word unit starts from words to express the UNL in the input taken for the universal words (UW) unit of Bangla keywords and their properties. After that, morphological analysis is to be performed in the morphology phase based on the Bangla language. In this phase, the Bangla roots words are changed by adding the inflexions/morphemes to obtain the full meaning of the words. In the case of the insertion phase of the maker, the case maker is inserted into the morphed word, e.g., ইতেছি, ইব, ই. These case makers are integrated into the extracted sentence. Finally, to determine the lexicon order for the extracted Bangla sentence, syntactic linearization is used to match the output with the Bangla native language sentence [13,14].

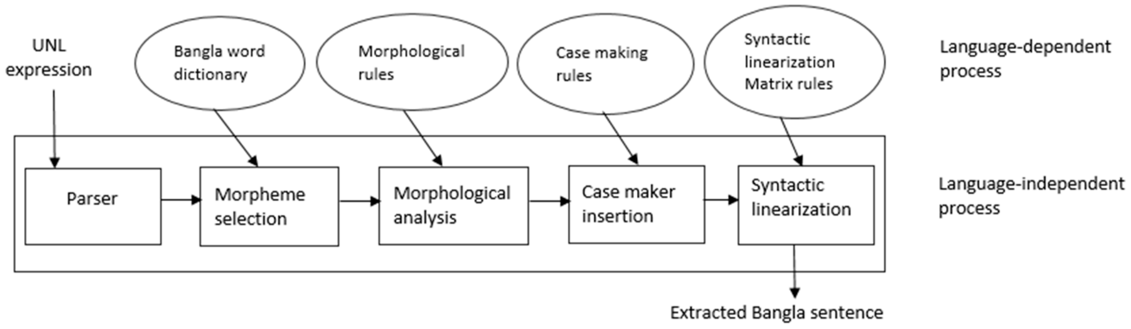


Figure 2. Architecture of Bangla DeConverter for Universal Networking Language.

The execution procedure of the Bangla DeCo is depicted with a given Bangla sentence as follows.
Bangla sentence: বালিকাটি মঞ্চে গান গেয়েছিল।
Transliterated sentence: Balikati monche gan geyechhilo.
Equivalent English sentence: The girl sang a song on a stage.
The UNL expression for the above sentence is presented in (2).

{unl}
agt(sing(icl>do).@entry.@past,girl(icl>child>person,ant>boy))
obj(sing(icl>do).@entry.@past,song(icl>musical_composition)) (2)
plc(sing(icl>do).@entry.@past,stage(icl>place))
{/unl}

The Bangla DeConverter is used to convert the above UNL expression (2) into the Bangla language text. This expression is the input for the Bangla DeCo. The parser verifies the input expression for faults and creates the UNL graph or node-net as illustrated in Figure 3.

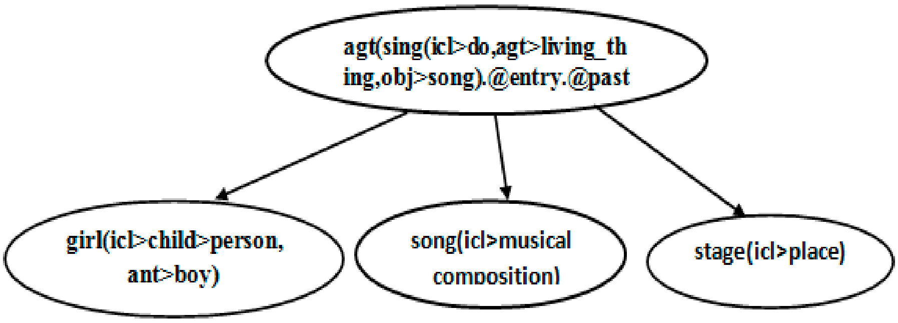


Figure 3. UNL graph generated by the parser for the input UNL expression.

The morpheme selection phase selects the node list for the UWs provided in the inserted UNL phase with corresponding Bangla phases. The settled node list is set out in (3).

Node1: Bangla word: 'গাওয়া' pronounce as *gawa*
UW: sing(icl>do,agt>living_thing,obj>song.@entry.@past)
Node2: Bangla word: 'বালিকা' pronounce as *balika*
UW: girl(icl>child>person,ant>boy))
Node3: Bangla word: 'গান' pronounce as *gan*
UW: song(icl>musical_composition))
Node4: Bangla word: 'মঞ্চ' pronounce as *moncho*
UW: stage(icl>place) (3)

In the morphology stage, morphological analysis is performed by applying morphological rules to amend the Bangla words in the nodes based on the UNL attributes in the inserted UNL expressions and lexicons retrieved from the word dictionary. The morphological analysis is to be performed on the nodes given in (3) using morphological rules. The processed nodes are provided in (4) after evaluation.

Node1: Bangla word: **'গেয়েছিল'**pronounce as *geyechhilo*
UW: sing(icl>do,agt>living_thing,obj>song.@entry.@past)
Node2: Bangla word: 'বালিকা' pronounce as *balika*
UW: girl(icl>child>person,ant>boy)
Node3: Bangla word: **'গান'** pronounce as *gan*
UW: song(icl>musical_composition)
Node4: Bangla word: 'মঞ্চ' pronounce as *moncho*
UW: stage(icl>place) (4)

From the nodes given in (4) morphological analysis has been performed on verbal noun 'গাওয়া'. Firstly, it has been changed from the main verb root 'গা'(ga) to alternative verb root 'গে'(ge). Then the morphological rule is applied to integrated verb root 'গে' (ge) and verbal inflexion'য়েছিল''echhilo' to from 'গেয়েছিল' (geyechhilo). It means in this phase, sing (gawa) is changed to its past form 'sang' by morphology rule. Hence, the case maker is inserted in the morphed lexicon in the case maker insertion phase. The nodes processed during the insertion phase of the case maker are shown in (5).

Node1: Bangla word: **'গেয়েছিল'**pronounce as *geyechhilo*
UW: sing(icl>do,agt>living_thing,obj>song.@entry.@past)
Node2: Bangla word: 'বালিকাটি'pronounce as *balikati*
UW: girl(icl>child>person,ant>boy)
Node3: Bangla word: **'গান'**pronounce as *gan*
UW: song(icl>musical_composition)
Node4: Bangla word: 'মঞ্চে' pronounce as *monche*
UW: stage(icl>place) (5)

Here, case makers ‘টি’(ti) and ‘এ(e)’ are combined with Node2 and Node4, respectively based on the case maker insertion rule. The string of the processing nodes is shown in (6), and the Bangla text produced by this sequence is shown in (7).

Node2 Node4 Node3 Node1 (6)

বালিকাটি মঞ্চে গান গেয়েছিল। (7)

Balikati Monche gan geyechhilo.

From the extracted Bangla sentence, it is apparent that the proposed system can accurately translate a UNL expression to the Bangla sentence.

4. Phases of Bangla DeConverter

A Bangla sentence is produced by the Bangla DeConverter from UNL expressions through different phases.

4.1. Parser Phase

The parser phase is the first phase of the Bangla DeCo. It parses the UNL-expression input to respond to errors in the expression if any. If the input expression is free from error, it then builds a semantic network called node-net structure for the expression. The node-net is also known as the UNL graph consisting of nodes and edges. The nodes of the UNL graph represent a concept in the form of UNL universal word (UWs). The edge in the node-net represents the UNL binary relations between two nodes. From the parent node to the child node, the edges in the UNL graph (Figure 3) are indicated. The system allows accessing from children to their parents for backtracking purposes.

4.2. Morpheme Selection Phase

Morpheme selection is the way of choosing Bangla lexicons for universal words (UWs) in the UNL expressions given as input. During morpheme selection, UWs are searched in the word dictionary along with constraints specified in the expression. A Bangla word lexicon containing around 10,000 entries has been developed. The dictionary consists of the Bangla root words as the headword (HW) of the UW and a set of grammatical, morphological, syntactic, and semantic attributes as its entries. The format of some generation rules that are to be used for the insertion of selection words/morphemes for the Bangla word dictionary is given below.

Format 1: Subjective pronoun insertion rules
where, *HPR* indicates human pronoun, *SUB* is a topic, *agt* is an agent relationship, *VR* is a verb root, *VEN* is a vowel ended root, ^*AL* is an alternative root, *P* is an individual, and *p* is a temporary attribute for an individual to avoid recursive activities. Example of rule,
:”HPR,P2,SUB,^@respect,@contempt,^RES,NGL::agt:”{RT,VEN,^AL,#AGT,^ p2:p2::}P9;
Format 2: Noun insertion rule before verb root
:”N,[^]@pl,^SUB:SUB:agt:”{RT,VEN,#AGT,^p3,[^]sg|pl:p3,sg|pl::}P7;
Example:
○ :”N,^@pl,^SUB:SUB:agt:”{VR,VEN,#AGT,^p3,^sg:p3,sg::}P7;

4.3. Morphological Analysis Phase

Morphology is the field of linguistics that studies the structure of words. It focuses on the patterns of words formation within and across languages and attempts to formulate rules that model the knowledge of the speakers of those languages.

A morpheme is defined as the minimal meaningful unit of a language. For example, in a word like ‘independently’, the morphemes are in-, depend, -ent, and –ly. In this case, the word ‘depend’ is the root and the other morphemes are the derivational affixes. Consider the following Bangla sentence,

Bangla sentence: আমি রুটি খাইতেছি

Pronounce as: Ami ruti khaitechhi

English meaning: I am eating bread.

The verb of the sentence is ‘খাইতেছি’ (eating). The verb is the combination of two words root ‘খা’ and morpheme ‘ইতেছি’. The construction of the verb ‘খাইতেছি’ is depicted below.

Root	+	**Verbal Inflexion**	=	**Verb**
খা	+	ইতেছি	ইতেছি	খাইতেছি

The root ‘খা’ (eat) and verbal inflexion (morpheme) ‘ইতেছি’ are the headwords of dictionary entries as follows.

[খা] {} “eat(icl>consume>do)” (|R, VR, VEN, VEG1)

[ইতেছি] {} “ “ (VI, VEN, P1, PRS)

where, R stands for root, VR for verb roots, VEN for vowel ended, VEG for vowed group 1, VI for verbal inflexion (morpheme), P1 for the first person, and PRS for the present tense.

The following morphological rule is to be used to combine verb root and verbal inflexion to form the verb khaitechhi

+ {R, VR, VEN:::} {VI:+V,-VI::}

The rule illustrates that verb root ‘খা’ (kha) is inserted into the left analysis window (LAW) and verbal inflexion ‘ইতেছি’ (itechhi) is inserted into the RAW (right analysis window) of the DeConverter to form a verb ‘খাইতেছি’ (khaitechhi).

Therefore, the application of morphological rules results in changing, the headword’s *root* and *verbal inflexion* into a *verb*.

These rules are developed according to the analysis of Bangla morphology. Three categories of morphology are defined for the conversion of UNL expressions into equivalent Bangla sentences [15]. Attribute level resolution morphology, relation level resolution morphology, and word level morphology.

The first label of morphology forms Bangla based on UNL attributes appended to a UW, and its specifications recovered dictionary entries. The root word fetched from the word dictionary is changed based on persons, numbers, gender, tenses, aspects, modality (PNGTAM), and vowel and consonant ended roots.

The relation label resolution morphology handles the postpositions in Bangla or prepositions in English as the prepositions in latter resemble the postpositions in Bangla. These postpositions connect to nouns, pronouns, verbs, and other parts of a sentence.

For example,

verb	+	suffix	=	verbal noun
চল	+	অন্ত	=	চলন্ত

In this case, most of the UNL relations set up postpositions or case makers or function words between the parent and child node during the text extraction process. Generation of a target language texts/words depends on the UNL relations and the conditions entailed on the child and parent nodes’ specifications of UNL relation. With UNL relation and attribute label morphology the Bangla DeConverter generates a sentence close to its original form.

4.4. Case Maker Insertion Phase

The case maker insertion phase inserts case makers such as conjunction and postposition in Bangla e.g., ‘হইতে’(from), ‘র’(of), ‘পেরিয়ে’ (over) to the words produced at the morphology phase.

The insertion of case makers in the rendered output depends on child and parent nodes' features in a relation [13]. For each of the 46 UNL relations [1], various types of case makers are used based on the grammatical structures of a target language for each of the UNL relations [16]. The rule which is to be prepared for the insertion of case makers includes nine columns. The explanation of each column is as follows.

1. The first column (UNL relation name): This column stores the UNL relations' name in which the rule is being made. For example; agt (agent relation), obj (object relation), etc.
2. The second column (The case maker preceding the parent node): This column stores the case maker that can be inputted before the parent node of a given UNL relation in produced output.
3. The third column (The case maker following the parent node): This column stores the case maker that can be inputted after the parent node of a given UNL relation in produced output.
4. The fourth column (The case maker preceding the child node): This column stores the case maker that can be inputted before the child node of a given UNL relation in produced output.
5. The fifth column (The case maker following the child node): This column stores the case maker that can be inputted after the child node of a given UNL relation in produced output.
6. The sixth column (Positive condition for the parent node): This column stores attributes that need to be declared on the parent node for firing the rule.
7. The seventh column (Negative conditions for the parent node): This column stores attributes that need to be declared on the parent node for firing the rule.
8. The eighth column (Positive condition for the child node): This column stores attributes that need to be declared on the child node for firing the rule.
9. The ninth column (Negative conditions for the child node): This column stores attributes that need to be asserted on the child node for firing the rule.

Formats of some generation rules used for insertion of the case makers from the Bangla word dictionary are given below.

Format 1: Verbal inflexion insertion rules for first person
:{RT,VEN,p(x),[^]@present,[^]@progress,[^]@complete,^vi:vi::}"[[VI]],VI,VEN,P(x),PRE|PAS|FUT,[^]PRG,[^]CMPL:::"P7;
Examples:
:{RT,VEN,p1,@present,^@progress,^@complete,^vi:vi::}"[[VI]],VI,VEN,P1,PRE,^PRG,^CMPL:::"P9;

Format 2: Verbal inflexion insertion rules for second person
:{RT,VEN,p(x),[^]@present,[^]@progress,[^]@complete,[^]res,[^]ngl,^vi:vi::}"[[VI]],VI,VEN,P(x),PRE|PAS|FUT,[^]PRG,[^]CMPL,[^]RES,[^]NGL:::"P7;
Examples:
:{RT,VEN,2p,@present,^@progress,^@complete,^hon,^ngl,^kbiv:kbiv::}"[[VI]],VI,VEN,P2,PRE,^PRG,^CMPL,^RES,^NGL:::"P7;

Format 3: Verbal inflexion insertion rules for third person
:{RT,VEN,p(x),[^]@present,[^]@progress,[^]@complete,^res,|^ngl,^vi:vi::}"[[VI]],VI,VEN,P(x),PRE|PAS|FUT,[^]PRG,[^]CMPL,^RES,^NGL:::"P7;
Examples:
:{RT,VEN,p3,@present,^@progress,^@complete,^resn,^vi:vi::}"[[VI]],VI,VEN,P3,PRE,^PRG,^CMPL,^RES:::"P7;

4.5. Syntactic Linearization Phase

This is the process of linearizing the words/morphemes of the sentence in the semantic hypergraph. As such, it determines the word order of produced texts. Syntactic linearization handles the systematic organization of lexicons (words/morphemes) in generated output to match the target language sentence. It allocates relative positions to different lexicons based on the bond they keep with the headwords of a sentence [15]. Constructive variety between languages Bangla (subject–object–verb, SOV) and English (subject–verb–object, SVO) enforces the stage of semantic linearization in order to improve the Bangla DeCo.

5. Issues in Syntactic Linearization

Two significant issues in syntactic linearization are parent–child relation and matrix-based priority of relation.

5.1. Parent–Child Relation

Binary relation between two words is represented as rel (uw1,uw2) in UNL, where uw1 works as a parent and uw2 works as a child. The system manages whether parent should be stated before or after the child in generated output [17]. In most of the UNL relations, child nodes appear before their parent nodes in the produced output. To demonstrate this conception, we take a sentence say, 'He is writing a letter'. UNL expression of the sentence is given below.

$$\begin{array}{c}\{\text{unl}\}\\ \text{agt(write(icl>do,equ>compose).@entry.@present.@progress,he(icl>person))}\\ \text{obj(write(icl>do,equ>compose).@entry.@present.@progress,letter(icl>text).@indef)}\\ \{\text{/unl}\}\end{array} \quad (8)$$

In this expression, there are two UNL relations: agt(write, he) or agt(uw1,uw2) and obj(write, letter) or obj(uw1,uw2). In the first relation, write (uw1) is a verb and he(uw2) is the subject, and in the second relation, uw1 is the same verb as in first relation and letter (uw2) is the object or noun.

Since, Bangla is an SOV type language, in case of UNL relation both children i.e., 'he(icl>person)' and 'letter(icl>document)' will be set to the left of the verb 'write'. Matrix-based priority will decide which child is to be placed first in the produced output.

5.2. Matrix-Based Priority of Relations

When a parent node has two or more children in a UNL relation, a matrix-based priority of relations is necessary to decide the relative position of children with respect to each other in the produced output [8].The relative positions of children in a sentence are decided in the proposed Bangla DeConverter using a matrix $M * M$. The matrix has 46 columns and 46 rows, publishing 46 UNL relations [1]. The matrix $M = [m_{ij}]$, where $i = 1, 2, \ldots 46$ and $j = 1, 2, \ldots .46$. The elements of the matrix are '*L*' denotes toward left, '*R*' denotes toward right and '-' denotes no action.

If m_{ij} = '*L*', it indicates that when two children share the same parent, the position of the child of *i*th relation is to the left of the child of *j*th relation. Again, if m_{ij} = '*R*', then the place of the child of *i*th relation is to the right of the child of *j*th relation, sharing the common parent. If m_{ij} = '-', no action is to be taken, as it is not possible to share a common parent by the children of *i*th and *j*th relations [16]. The following UNL graph illustrates the relationships of two nodes with the same parent.

In this Figure 4, the children nodes are 'N1' and 'N2' and the parent node is 'N3'. The UNL binary relations between these three nodes 'N1', 'N2', and 'N3' are Ri(N3,N1) and Rj(N3,N2). If we consider a Bangla sentence, say, আমিভাতখাই, '*ami vat khai*' meaning 'I eat rice', then according to the graph in Figure 4, the representation of the first word of the sentence আমি will be the node N1, the second word 'ভাত' will be the node N2, and the third word 'খাই' will be the node N3, respectively.

According to the structure of Bangla language, if 'N1' places to the left of 'N2' in the produced sentence denoted by '(N1 L N2)' then the priority matrix given in Figure 5 should be followed for its syntactic linearization.

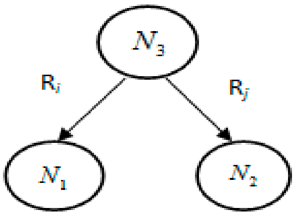


Figure 4. UNL graph of two nodes with the common parent.

	Ri	Rj
Ri	-	L
Rj	R	-

Figure 5. Representation of matrix for (N1 L N2) structure.

Precedence of a child in a binary relation relies on the frequency of 'L' in its row. The child in a relationship will have the highest priority and will be placed at the remaining output if the UNL binary relationship has all 'L' in its line. Similarly, if the UNL binary relationship has all 'R' in its line shared by the same parent, the child of the relationship will be the smallest priority and position at the extreme right of all child nodes in the output produced [14,15].

6. Syntactic Linearization of Simple and Compound Sentences

Syntactic linearization of simple and compound sentences is described in this section.

6.1. Syntactic Linearization of Simple Sentence

A simple sentence says, 'She has earned 100 dollars'. The UNL expression of this sentence is given in (9).

```
{unl}
agt(earn(icl>do).@entry.@present.@complete,she(icl>person))
qua(dollar(icl>monetary_unit).@pl,100)                                        (9)
obj(earn(icl>do).@entry.@present.@complete,dollar(icl>monetary_unit>thing).@pl)
{/unl}
```

The UNL graph of the UNL expression in (9) is depicted in Figure 6.

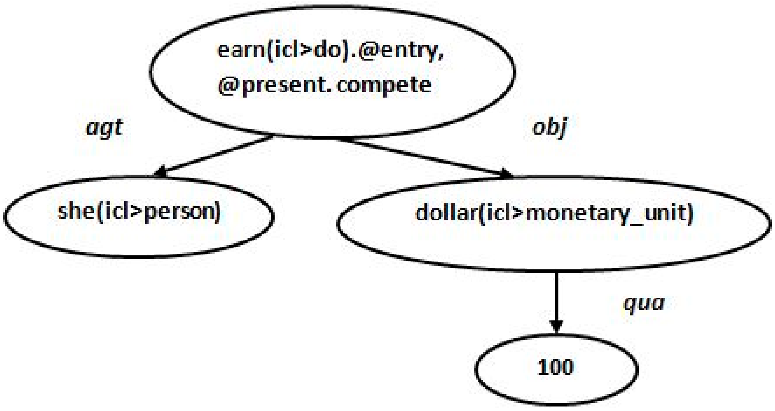


Figure 6. UNL graph for the UNL expression (9).

After morphological and case maker insertion phase the node list in the UNL expression is presented in (10).

Node1: Bangla word: আয়করা, pronounce as 'aye kora'
UW: earn(icl>do,agt>person,obj>thing,ins>uw)
Node2: Bangla word: সে (মহিলা), pronounce as 'se (mohila)
UW: she(icl>person)
Node3: Bangla word: ডলার, pronounce as dollar
UW: dollar(icl>monetary_unit>thing)
Node4: Bangla word: ১০০, pronounce as 100
UW: 100 (10)

In Figure 6, the parent node 'earn(icl>do).@entry.@present.@complete' is the entry node as it is the main predicate and includes UNL attribute @entry. The labels on the two edges denote UNL relation. The relation 'agt' has a higher priority than 'obj' relation as shown in Figure 6. Hence, the node 'She' will traverse first. The priority matrix of UNL relations 'agt' and 'obj' is given in Figure 7.

	agt	obj
agt	-	L
obj	R	-

Figure 7. Precedence matrix of UNL relations for 'agt' and 'obj'.

The node 'She' does not have any child and its parent node 'earn' has traversed. This node 'She' will be processed, and its Bangla attributes will appear to the final string in the produced output, i.e., the produced output will be 'সে' (She). At present, the parent node 'earn' of the child 'she' will be the active node. The 'earn' node has one unprocessed child, i.e., the node 'dollar'. Thus, the node 'dollar' will traverse next and will mark as traversed. The node 'dollar' has one unprocessed child, i.e., '100' and it will traverse next and mark as traversed. The node '100' has no child and it will be processed and its Bangla word specification will appear to its final string. Therefore, the last sentence will be১০০.

Now, the node 'earn' will be the active node. It is the main predicate of the sentence, i.e., the entry node, and it has no unprocessed child. Hence, it will be processed, and its Bangla word appears in the final sentence; i.e., the final sentence will be 'সে১০০ডলারআয়করেছে।'. Since, the main predicate is processed, the produced output will be available in the string based on the syntactic linearization as shown in node produced sequence below.

Node sequence: Node2 Node4 Node3 Node1 (11)

Bangla sentence: সে ১০০ ডলার আয় করেছে (12)

Pronounced as: Se 100 dollar aye korechhe.
Equivalent English sentence: She has earned 100 dollars.

6.2. Syntactic Linearization of Compound Sentence

A compound sentence is represented in UNL expression as a compound concept. A compound concept or a compound universal word is defined using a scope-node. A scope-node is a set of UNL binary relations combined jointly to denote a complex concept. This complex concept is named by an identifier, UW-ID, which appears after the relation label. In UNL expression, a compound UW is referred to with its UW-ID. The syntactic linearization of compound sentences is a bit different from that of simple sentences. Consider a sentence say, 'I will go to university after 40 minutes'. The UNL expression of this sentence is given in (13).

I will go to university after 40 min
{/org}
{unl}
agt(go).@entry.@future,i(icl>person))
plt(go.@entry.@future,university(icl>body))
tim(go.@entry.@future,:01))
man:01(after,@entry, minutes)
qua:01(minute.@pl,40)
{/unl}
[/S] (13)

The UNL graph of the UNL expression (13) is illustrated in Figure 8. In the UNL graph, 'go' is the main predicate, the entry node. It has three children: 'i(icl>person)', with 'agt' relation between 'go' and 'i', 'university(icl>body)' with 'plt' relation between 'go'and 'university' and scope node '01' with 'tim' relation between 'go' and '01'. The UNL graph shown in Figure 8 shows that the relation 'agt' has the highest precedence followed by the relation 'tim' and then by the relation 'plt' as given in the precedence matrix in Figure 9.

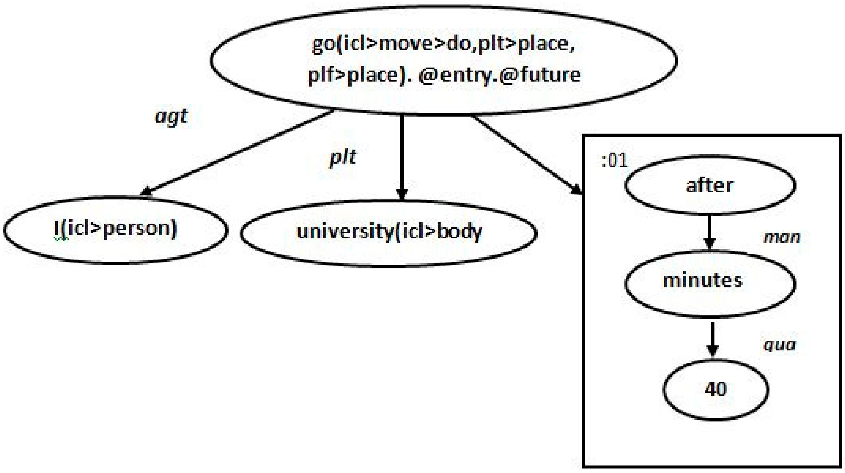


Figure 8. UNL graph for the UNL expression of the sentence with a scope node.

	agt	plt	tim
agt		L	L
plt	-	-	R
tim	R	L	-

Figure 9. Precedence matrix for the UNL relation 'agt', 'plt', and 'tim'.

In the UNL expression given in (13), when we consider scope node: 01 as single nodes, the syntactic linearization of the UNL graph considering the UWs will be

I: 01 University go (14)

This scope node is then replaced by the scope nodes of the UNL graph from the generated output of syntactic linearization. So, the UNL graph of the scope node for UWs of syntactic linearization is

after 40 min (15)

The output of the UNL expression (Figure 8) is produced by replacing the scope-node in (1) combining with UWs given in (2). So, the final syntactic linearization of UWs without restrictions of the UNL graph illustrated in Figure 8 will be produced as follows.

I 40 min after university go (16)

A Bangla sentence produced by the Bangla DeConverter after morphology, case maker insertion, and syntactic linearization phases is given as follows.

আমি ৪০ মিনিট পর ইউনিভার্সিটি যাবো (17)

Pronounced as: Ami 40 min por bisshabiddaloy jabo.

7. Experimental Results and Discussions

This section shows the experimental results by extracting a Bangla sentence using generation rules. It also presents the evaluation of the proposed system and discussions of results.

7.1. Extraction of a Bangla Sentence

This section explains the steps of transformation and the experimental outcomes of extracting a Bangla sentence from UNL phrases. The UNL expression of the sentence, *I will go to office after eating rice* is shown in (18) using Russian English Language Server. In this expression, *agt*(agent), *plt*(place to), *tim* (time), and *obj* (object) are the semantic relations. The relaters go(icl>move>do), i(icl>person), office(icl>body>thing), after(icl>how, tim<uw,obj>uw), eat(icl>consume>do), rice(icl>grain> thing) are the UWs [1]. Typically connected to the primary predicate attribute @entry and @future denotes future tense sentence. We use our proposed DeConverter tool for our experiment. UNL expression (18), a dictionaries file (Table 1), and a set of generation rules (Table 2), are the inputs of the tool for extracting a sentence from the given UNL expression.

agt(go(icl>move>do,plt>place).@entry.@future,i(icl>person))
plt(go(icl>move>do,plt>place).@entry.@future,office(icl>body>thing))
tim(go(icl>move>do,plt>place).@entry.@future,after(icl>how,tim<uw,obj>uw)) (18)
obj:01.@entry,rice(icl>grain>thing))
obj(after(icl>how,tim<uw,obj>uw),:01)

Table 1. Dictionary entries for respective Bangla sentence.

[আমি] { }“i(icl>person)”(PRO, HPRO,SUB,P1,SG)
[ভাত]{ } “rice(icl>grain>thing)”(N, OBJ, NCOM,CEN)
[খে]{ } “eat(icl>consume>do,agt>living_thing,obj>concrete_thing)” (ROOT,VEN,AGT,OBJ,VEG1)
[য়ে]{ }“VI” (VI,VEN,P1,PRE)
[ইউনিভার্সিটি]{ } “office(icl>body>thing)”(N,
[যা]{ } “go(icl>move>do,plt>place,plf>place,agt>thing)”(VR, VEN,AGT,PLT,PLF,P1)
[বো]{ }“VI” (VI,VEN,P1,FUT)

Table 2. Generation rules for extracting Bangla sentence from UNL expression.

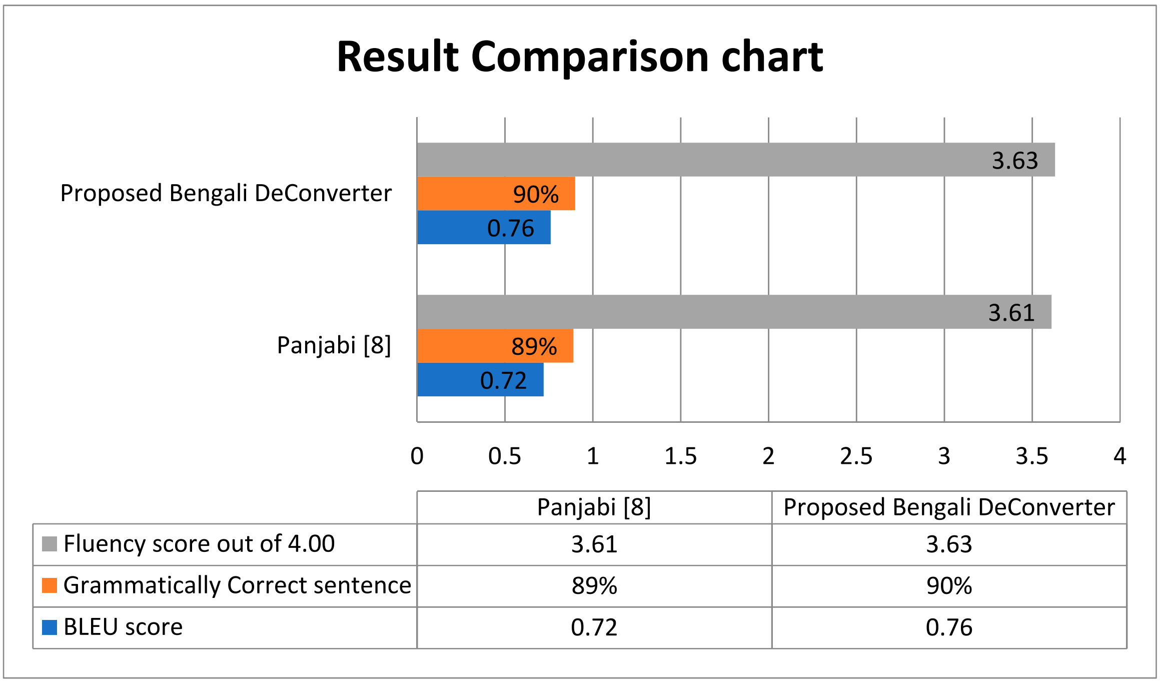
Rule 1: (Pronoun insertion) :”HPRO,P1,SUB,^@pl,::agt:”{ROOT,VEN,^AL,#AGT,^p1:p1::}
Rule 2: R{:::}{SUB:::}
Rule 3: {SUB,^blk:blk::}”[],BLK:::”}
Rule 4: -R{:::}{SUB:::}
Rule 5: -R{SUB:::} {:::}
Rule 6: “N,^OBJ:OBJ::”{ROOT,VEN,#OBJ::}
Rule 7: R{PRO:::}{N:::}
Rule 8: R{N:::} {ROOT:::}
Rule 9: {ROOT,VEN,p1,@present,^@progress,^@complete,^vi:vi::}”[[VI]],VI,VEN,P1,PRE,^PRG,^CMPL:::”}
Rule 10: R{PRO:::}{N:::}
Rule 11: “N,^OBJ:OBJ::”{ROOT,VEN,#OBJ::}P7;
Rule 12: {SUB,^blk:blk::}”[],BLK:::”}
Rule 13: R{N:::}{:::}
Rule 14: {N:::} {ROOT,VEN,#OBJ::}
Rule 15: R{ROOT:::} {:::}
Rule 16: {ROOT,VEN,p1,@future,^@progress,^@complete,^vi:vi::}”[[VI]],VI,VEN,P1,FUT,^PRG,^CMPL:::”}
Rule 17: R{:::} {V:::}

Rule 1 describes when root “খে” is in LCW (left condition window) the pronoun “আমি” is to be inserted in the RGW (right generation window). Rule 2 is used to move the DeConverter windows to the right. The blank insertion rule (rule 3) is applied to insert a blank space between pronoun and root. After applying right shift rules (rules 4 and 5), rule 6 is to be used to insert noun “ভাত” in the RGW. Rules 7 and 8 are now being implemented to move the windows two steps right. Now the root “খে” is in LGW. The verbal inflexion insertion rule (rule 9) is to be applied to inset “য়ে” in the RGW. Now both root “খে” and verbal inflexion “য়ে” are to be combined to make verb “খেয়ে” and will be in the RGW. Noun insertion rule (rule 11) is to be applied to insert noun “ইউনিভার্সিটি” followed by a blank insertion rule (rule 12) to insert a blank between the verb “খেয়ে” and noun “ইউনিভার্সিটি” after applying right shift rule (rule 10). Again, right shift (rule 13) is to be applied followed by root insertion rule (rule 14). Finally, the right shift (rule 15) is to be applied followed by a verbal inflexion insertion rule (rule 16). The right shift (rule 17) completes the text extraction process of DeConverter. After completing the extraction procedures, DeCo generates the following Bangla sentence, আমি ভাত খেয়ে ইউনিভার্সিটি যাবো।

We have experimented for various types of Bangla simple and complex texts (sentences) with diverse subjects, persons, and tenses. Our result shows that Bangla native language texts are extracted acceptably by the proposed Bangla DeCo.

7.2. Results and Discussions

Our projected system was tested by converting a set of 300 Bangla sentences into their corresponding set of UNL expressions using a Russian–English Language server [18]. The server includes English sentences along with their corresponding UNL expressions. For comparison with the output generated by our Bangla DeConverter, these English phrases were manually converted into equivalent Bangla phrases. The UNL expressions were used as input to the Bangla DeConverter for producing corresponding Bangla sentences. The output of the Bangla DeConverter was compared with the corresponding manually translated Bangla sentences from English sentences. The projected system has been applied to subjective tests such as fluency and adequacy tests. To ensure the quality of output, the Bilingual Evaluation Understudy(BLEU) score has been calculated. Some Bangla phrases generated with their respective UNL phrases by the proposed Bangla DeConverter are shown in Table 3. Our proposed system achieved a BLEU score of 0.76. Since, there are no other Bangla DeConverters proposed yet, we compared our system with a Punjabi DeConverter [8] to test the efficiency of our work. A comparison of the results, shown in Figure 10, was conducted based on the BLEU score, fluency score, and the percentage of a grammatically correct sentence.



	Panjabi [8]	Proposed Bengali DeConverter
■ Fluency score out of 4.00	3.61	3.63
🟧 Grammatically Correct sentence	89%	90%
🟦 BLEU score	0.72	0.76

Figure 10. Result comparison with Punjabi DeConverter [8].

We have evaluated the proposed Bangla DeConverter with only one other DeConverter (Punjabi DeConverter) because, no DeConverters have been proposed for Bangla language yet. The sentence structure of Bangla language and Punjabi language are almost similar. Unlike English, Bangla is a free word order language known for its affluent semantical and morphological features similar to the Punjabi language. The English language is patterned with Subject, Verb, and Object (SVO), while both Bangla and Punjabi languages follow Subject, Object, and Verb (SOV) pattern. Therefore, we have evaluated the proposed DeConverter with the Punjabi DeConverter.

In is well known that a rule-based machine translation system provides good accuracy in written and plainly structured documents such as simple article, weather report etc. But it cannot work efficiently for real-world documents. The main reason is that a human language does not follow a fixed set of rules. Human languages are full of regional variations, special cases, and new rules. New rules are continuously evolving and old rules are continually changing in most, if not all languages. Therefore, a slight improvement may play importance roles.

Table 3. Bangla sentences produced by the proposed Bangla DeCo with their corresponding input UNL expressions.

Sentence No.	Input UNL Expressions	Bangla Sentences Produced by Bangla DeConverter

1	{unl}
	agt(spend(icl>pass>do,com>time,).@entry.@past,i(icl>person))
	pos(holiday(icl>leisure>thing,equ>vacation),i(icl>person))
	obj(spend(icl>pass>do,com>time).@entry.@past,holiday(icl>leisure>thing,equ>vacation))
	plc(holiday(icl>leisure>thing,equ>vacation),paris(iof>national_capital>thing))
	{/unl}
		আমি ছুটির দিন পারিসে কাটিয়েছি
		Ami chhutir din parishe katiechhi
		I spent my holiday in Paris.

2	{unl}
	agt(perform_an_action(icl>do).@entry.@present,we(icl>group).@pl)
	pos(work(icl>activity>abstract_thing),we(icl>group).@pl)
	obj(perform_an_action(icl>do).@entry.@present,work(icl>activity>abstract_thing))
	man(perform_an_action(icl>do).@entry.@present,perfectly(icl>how,equ>absolutely))
	{/unl}
		আমরা আমাদের কাজ সঠিকভাবে করি।
		Amra amader kaj shothikvabe kori
		We do our work perfectly

3	{unl}
	aoj(city(icl>administrative_district).@entry.@present,tokyo(iof>national_capital>thing))
	man(beautiful(icl>adj,ant>ugly),very(icl>how,equ>extremely))
	mod(city(icl>administrative_district).@entry.@indef.@present,beautiful(icl>adj,ant>ugly))
	{/unl}
		টকিও একটি সুন্দর শহর।
		Tokyo ekti shundor shohor
		Tokyo is a very beautiful city.

4	{unl}
	aoj(have(icl>be,equ>possess,obj>thing,aoj>thing).@entry.@present,i(icl>person))
	obj(have(icl>be,equ>possess,).@entry.@present,tomorrow(icl>time,ant>yesterday))
	aoj(meet(icl>join>be,cao>thing,aoj>thing).@progress,tomorrow(icl>time,,ant>yesterday))
	{/unl}
	[/S]
		আগামিকাল আমার একটা মিটিং আছে।
		Agamikal amar ekti meeting achhe
		I have a meeting tomorrow.

5	{unl}
	agt(go(icl>move>do,plt>place,plf>place,agt>thing).@entry.@present,they(icl>group).@pl)
	plt(go(icl>move>do,plt>place).@entry.@present,office(icl>organization,icl>place))
	met(go(icl>move>do,plt>place).@entry.@present,car(icl>motor_vehicle>thing))
	{/unl}
		তারাএকটিগাড়িতেকরেঅফিসজায়।
		Tara ekti garite kore office jaye.
		They go to office by a car.

6	{unl}
	aoj(admit(icl>icl>give_access>be,plt>place).@entry.@past,he(icl>person))
	plc(admit(icl>give_access>be,plt>place).@entry.@past,hospital(icl>medical_institution>))
	rsn(admit(icl>give_access>be,plt>place).@entry.@past,illness(icl>ill_health>thing))
	{/unl}
		সে অসূস্ত হওয়ায় হাসপাতাল ভর্তি হয়েছে।
		Se oshusto howaye haspatale vorti hoyeche
		He admitted in a hospital due to illness

7	{unl}
	agt(go(icl>move>do,plt>place,plf>place,agt>thing).@entry.@present,i(icl>person))
	plt(go(icl>move>do,plt>place).@entry.@present,australia(iof>country>thing))
	via(australia(iof>country>thing),singapore(iof>island>thing))
	{/unl}
		আমি সিঙ্গাপুর হয়ে অস্ট্রেলিয়া জাই।
		Ami Singapore hoye Australia jai
		I go to Australia via Singapore.

8. Conclusions

This research paper has proposed a Bangla DeConverter. Syntactic linearization is a significant part of the proposed system for the extraction of quality Bangla language texts. Syntactic linearization of simple and compound sentences with scope-node and matrix-based priority of relations have been discussed in this paper. The proposed Bangla DeCo system has been tested for 300 UNL expressions. The system attained a fluency score of 3.63 on a four-point scale, and a BLEU score of 0.76. The proposed Bangla DeCo can successfully convert a UNL expression to resemble Bangla texts. Researchers of other native languages can explore our system to develop DeCo for their respective native languages. Currently our system can convert simple Bangla sentences accurately. But for complex and compound sentences, sometimes the system does not provide efficient results. In our future work, we will address those issues and make our system more accurate.

Author Contributions: The authors contributions is as follows-"Conceptualization, M.N.Y.A. and M.L.R.; Methodology, M.N.Y.A. and M.L.R.; Validation, M.N.Y.A. and G.S.; Formal analysis, M.N.Y.A. and M.L.R.; Resources, M.N.Y.A.; Data curation, M.L.R.; Writing—original draft preparation, M.N.Y.A. and M.L.R.; Writing—review and editing, G.S.; Visualization, G.S.; Supervision, M.N.Y.A.; Project administration, M.N.Y.A.; Funding acquisition, M.N.Y.A. and G.S.".

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Uchida, H.; Zhu, M.; Senta, T.C.D. Universal Networking Language, UNDL Foundation. *Int. Environ. House* **2005**, *6*.
2. *EnConverter Specification*, Version 3.3; UNL Center/UNDL Foundation: Tokyo, Japan, 2002.
3. Boguslavsky, I.; Frid, N.; Iomdin, L.; Kreidlin, L.; Sagalova, I.; Sizov, V. Creating a Universal Networking Language module within an advanced NLP system. In Proceedings of the 18th conference on Computational Linguistics, Saarbrücken, Germany, 31 July–4 August 2000; Volume 1, pp. 83–89.
4. *DeConverter Specification*, Version 2.7; UNL Center, UNDL Foundation: Tokyo, Japan, 2002.
5. Martins, R.T.; Hasegawa, R.; Rino, L.H.M.; Oliveira Junior, O.N.D.; Nunes, M.D.G.V. Specification of the UNL-Portuguese enconverter-deconverter prototype. 1997. Available online: https://bdpi.usp.br/item/000951455 (accessed on 21 October 2019).
6. Dave, S.; Parikh, J.; Bhattacharyya, P. Interlingua-based English–Hindi Machine Translation and Language Divergence. *Comput. Transl.* **2001**, *16*, 251–304.
7. Kumar, P.; Sharma, R.K. Punjabi DeConverter for generating Punjabi from Universal Networking Language. *J. Zhejiang Univ. Sci. C* **2013**, *14*, 179–196. [CrossRef]
8. Blanc, E. About and around the French Enconverter and the French Deconverter. *Univers. Netw. Lang. Adv. Theory Appl.* **2005**, *12*, 157–166.
9. Shi, X.; Chen, Y. *A Unl Deconverter for Chinese*; UNL Book: Instituto Politécnico Nacional, Mexico, 2005.
10. Daoud, D.M. Arabic generation in the framework of the Universal Networking Language. *Univers. Netw. Lang. Adv. Theory Appl.* **2005**, *12*, 195–209.
11. Keshari, B.; Bista, K. UNL Nepali DeConverter. In Proceedings of the 3rd International Conference on CALIBER, Cochin University of Science and Technology, Kochi, India, 2–4 February 2005; pp. 70–76.
12. Singh, S.; Dalal, M.; Vachhani, V.; Bhattacharyya, P.; Damani, O.P. Hindi generation from Interlingua (UNL). In Proceedings of the Machine Translation Summit XI, Copenhagen, Denmark, 10–14 September 2007.
13. Nalawade, A. Natural Language Generation from Universal Networking Language. Master's Thesis, Indian Institute of Technology, Bombay/Mumbai, India, 2007.
14. Vachhani, V. UNL to Hindi DeConverter. Bachelor's Thesis, Dharamsinh Desai Institute of Technology, Nadiad, India, 2006.
15. Dey, K.; Bhattacharyya, P. Universal Networking Language based analysis and generation of Bangla case structure constructs. *Univers. Netw. Lang. Adv. Theory Appl.* **2006**, *12*, 215–229.
16. Vora, A. Generation of Hindi sentences from Universal Networking Language. Bachelor's Thesis, Dharamsinh Desai Institute of Technology, Nadiad, India, 2002.

17. Hrushikesh, B. Towards Marathi Sentence Generation from Universal Networking Language. Master's Thesis, Indian Institute of Technology, Bombay/Mumbai, India, 2002.
18. Ru: Russian and English Language Server. Available online: http://www.unl.ru (accessed on 18 August 2019).