NLization of Nouns, Pronouns, Prepositions and Sentence Structures with EUGENE

Thesis Submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Ashutosh Verma
(Roll No. 801132008)

Under the supervision of:
Dr. Parteek Bhatia
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2013

CERTIFICATE

I hereby certify that the work is being presented in this thesis entitled, "*NLization of Nouns, Pronouns, Prepositions and Sentence Structures with EUGENE*", in partial fulfillment of the requirement for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Parteek Bhatia* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other university.

(Ashutosh Verma)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Parteek Bhatia)
Asst. Professor, CSED

Countersigned By

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

i

Acknowledgement

No volume of words is enough to express my gratitude towards my guide Dr. Parteek Bhatia, Asst. Professor, Department of Computer Science & Engineering, Thapar University, Patiala, who has been very concerned and has aided for all the materials essentials for the preparation of this thesis report. He has helped me to explore this vast topic in an organized manner and provided me all the ideas on how to work towards a research-oriented venture.

I am also thankful to Dr. Maninder Singh, Head of Computer Science & Engineering Department for the motivation and inspiration that triggered me for the thesis work.

I would like to thank Mr. Karun Verma (PG Coordinator) Asst. Professor, Department of Computer Science & Engineering, Thapar University and all other respected faculty members for their assistance and suggestions during research period.

I would also like to thank the staff members and my colleagues Harinder Singh and Vaibhav Agarwal who were always there at the need of hour and provided with all the help and facilities, which I required, for the completion of my thesis work.

Most importantly, I would like to thank my parents and the almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

(Ashutosh Verma)

Abstract

Universal Networking Language (UNL) has been used by various researchers as an Interlingua approach for MT (machine translation). Researchers have proposed different paradigms for machine translation across natural languages with reasonable success. Universal Networking Language (UNL) based MT is also an effort in this direction. The UNL program was launched in 1996 in Institute of Advanced Studies (IAS) of United Nations University (UNU), Tokyo, Japan. The approach in UNL revolves around two main components, namely, EnConverter-IAN (used for converting the text from a source language to UNL) and DeConverter-EUGENE (used for converting the text from UNL to a target language).

The work carried out in this thesis is divided into seven chapters. These chapters are: Introduction to UNL; Literature Review; Problem statement; NLization Process; Implementation of NLization of Nouns, Pronouns, Prepositions and Sentence Structures for Punjabi language; Results and Discussion; Conclusion and Future scope.

In first chapter, basic introduction of UNL system has been presented.The second chapter discusses about the evolution of the UNL alongwith the existing DeConverter systems. The problem statement has been discussed inthe third chapter. This chapter also describes the objectives of the proposed system andthe methodologies that have been followed to develop the proposed system.

The fourth chapter discusses about the NLization process. Components used in this process such asGrammar specifications and natural language generation rules has been described. EUGENE tool in NLization framework has also been presented in this chapter.

The fifth chapter discusses about the implementation of the proposed system. Requirements for the development of the proposed system have been described. DeConversion process has been explained with the help of example sentences of Nouns, Pronouns, Prepositions and Sentence Structures.

The Sixth chapter presents the results and discussion on the developed system. In the seventhchapter, the conclusion and future scope has been discussed.

Table of Contents

List of Figures

List of Tables

Chapter-1

Introduction to UNL

Universal Networking Language (UNL) is a declarative formal language specifically designed to represent semantic data extracted from natural language texts [1]. It can be used as a pivot language in Interlingua machine translation systems or as a knowledge representation language in information retrieval applications [2].

UNL is a computer language that enables computers to process information and knowledge. It is designed to replicate the functions of natural languages. Using UNL, people can describe all information and knowledge conveyed by natural languages for computers [3]. As a result, computers can intercommunicate through UNL and process information and knowledge using UNL, thus providing people with a Linguistic Infrastructure (LI) in computers and Internet for distributing, receiving and understanding multi-lingual information. Such multilingual information can be accessed by natural languages through UNL System [4].

Universal Networking Language is an artificial language created to represent and process information across language barriers. Universal Networking Language is an interlingua-based framework aimed to facilitate semantic processing of natural language by a computer. Its main applications cover not only in machine translation and other natural language processing tasks, but also in a wide variety of applications ranging from e-learning platforms to management of multilingual document bases [5].

UNL system uses EnConversion/UNLization and DeConversion/NLization as its core software components [6]. UNLization, formerly known as EnConversion, is the process of representing the content of a natural language structure into UNL and NLization, formerly known as DeConversion is the process of generating natural language structures corresponding to UNL graphs. For EnConversion or UNLization, IAN tool has been developed by UNDL foundation. IAN converts natural language sentences into UNL expression. For DeConversion or NLization, EUGENE tool has been developed by UNDL foundation. EUGENE converts UNL expression to natural language sentences. Both IAN and EUGENE perform their functions based on a set of grammar rules and a word dictionary of target language. Conversion process through EnConverter and DeConverter is shown in Figure 1.1.

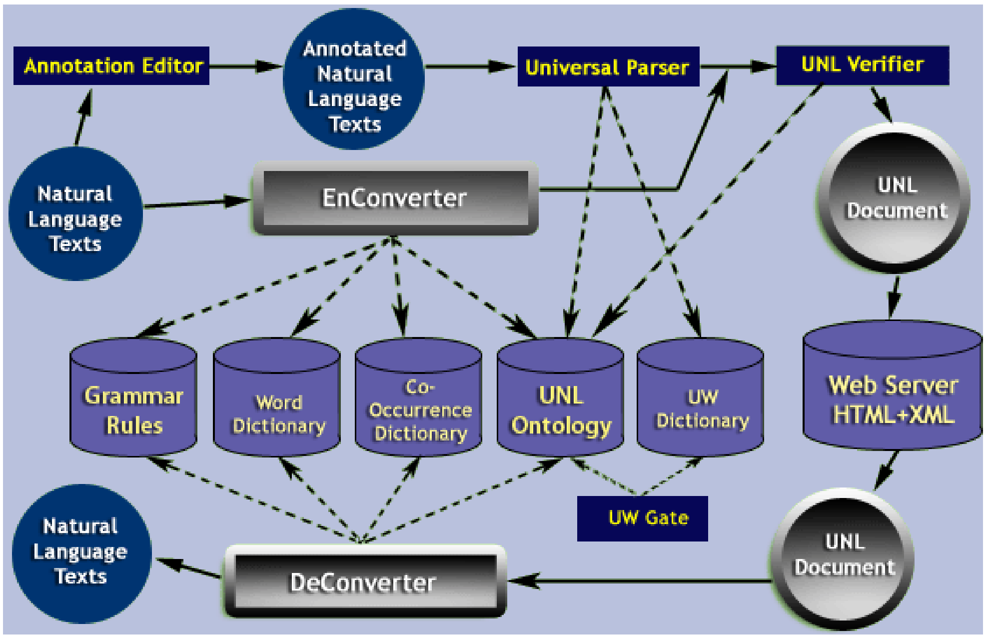

Figure 1.1: Conversion process in a UNL system [4] [6]

The Universal Parser (UP) is a specialized version of the EnConverter. It generates UNL Expressions from annotated sentences with referring to the UW dictionary without using grammatical features. All UNL Expressions are verified by the UNL Verifier, and then to be stored in the format of UNL Document. The DeConverter converts UNL Expressions to natural language sentences. Both the EnConverter and DeConverter perform their functions based on a set of grammar rules and a word dictionary of a target language. Whether consulting the UNL Ontology or a co-occurrence dictionary in EnConverter or DeConverter is optional [4].

1.1 History of UNL

The UNL is an ambitious initiative that was launched in 1996, as a program of the Institute of Advanced Studies in Tokyo, Japan [7] [8]. From the very beginning, a global-scale network of Research and Development teams, involving specialists in computer science and linguistics, have been at work creating the linguistic resources and developing the software of the UNL System. Under the direction of the UNL Center, a consortium of the university department and research institutes around the world have been developing UNL Language Centers (LC) for the respective native languages [4].

UNL program has already crossed important milestones. The overall structure of the UNL System has been developed with a set of basic software and tools necessary for the development and functioning of UNL. A vast amount of linguistic resources of the UNL as well as the various native languages has been accumulated in the last few years [2].

In recent years, UNL-based applications have been gradually developed. The purposes of current UNL-based applications are mainly to provide intelligent search of information utilizing UNL technology and infrastructure, to assist or guide people in creating or improving UNL environment, to facilitate use of NLized information and technology, *etc.* [4].

In January 2001, the United Nations University set up an autonomous organization, the UNDL Foundation, to be responsible for the development and management of the UNL Program. The Foundation, a non-profit international organization based in Geneva, Switzerland, inherited from the UNU/IAS the mandate implementing the UNL Program and of bringing it to fulfill its mission. The UNL remains a long term endeavor, and invites a broad participation of people from all languages. UNL is designed as modular and open endeavor, in which many people can work together, individually or in teams. It is accessible to all languages, with all their variations. To benefit from it, it is required to develop the linguistic resources, such as dictionaries, and the EnConversion and DeConversion rules for the natural language [2] [4] [8].

1.2 Structure of UNL

The structure of UNL is defined by the UNL specification. It specifies the structure of a UNL document; the syntax of Universal Words; the set of attributes; the set of relations; and all the information concerning UNL as a formalism [5]. The description of these are given below.

1.2.1 Universal Words (UW)

Universal Words are usually abbreviated to “UWs”. A UW is not only a unit of the UNL syntactically and semantically for expressing a concept but also a basic element for constructing a UNL expression of a sentence or a compound concept. Such a Universal Word is represented as a node in the semantic network of UNL expression [6].

They are labels for concepts, syntactic and semantic units to form UNL Expression. A combination of a set of UWs - linked with each other through relations and modified by attributes - expresses the meaning of a sentence [7]. Format of a universal word of UNL is given in (1.1).

<uw> =:: <headword>[<constraint list>] ... (1.1)

Here, a headword of a UW is an English expression, a word, a compound word, a phase or a sentence of English. If the meaning of a headword is unique, the headword itself becomes a UW. Otherwise, constraints are attached to the headword to make more specific UWs. If a UW consists of a headword only, it is called a "Basic UW" [8].

1.2.2 Attributes

Attributes are mainly for the purpose to describe subjectivity information. It includes time, aspect, emphasis, focus, topic, attitude, feeling and judgment. Attributes are also used to specify qualities of concepts such as the genericity, the specificity and the logicality of UWs. They are attached to a UW or a scope to specify the information [4]. Examples of attributes are shown in Table 1.1.

Table 1.1: Examples of Attributes of UNL

Describing logicality	@transitive,@symmetric,@identifiable,@disjointed
Describing times	@future,@past,@present
Describing aspects	@begin,@complete,@continue,@end,@progress,@state,…
Describing genericity and specificity	@generic,@def,@indef,@not,@ordinal
Describing emphasis, focus and topic	@emphasis,@entry,@focus,@topic,…
Describing attitudes	@affirmative,@imperative,@interrogative,@request,…
Describing feelings and judgments	@ability,@grant,@wish,@will,@obligation,@possible,@regret,…

Example

“I can hear a dog barking outside”

UNL expression of above example is given in (1.2).

{unl}

agt(hear(icl>perceive(agt>person,obj>thing)):06.@ability.@entry,I(icl>person):00.

@ topic)

obj(hear(icl>perceive(agt>person,obj>thing)):06.@ability.@entry,:01)

agt:01(bark(agt>dog):0H.@progress.@entry,dog(icl>mammal):0D.@indef)

plc:01(bark(agt>dog):0H.@progress.@entry,outside(icl>area):0P)

{/unl} ...(1.2)

In the UNL Expression given in (1.2), has ‘*agt*’, ‘*obj*’ and ‘*plc*’ are relations. ‘I(icl>person)’,‘hear(icl>perceive(agt>person,obj>thing))’,‘dog(icl>mammal)’,‘bark(agt>dog)’ and ‘outside(icl>area)’ are UWs. ‘@ability’, ‘@entry’, ‘@indef’, ‘@progress’ and ‘@topic’ are attributes. The part “a dog barking outside” is expressed in a scope, ‘01’ is assigned as a Scope-ID to the scope. Binary relations with the same Scope-ID appearing following the relation labels constitute the UNL Expression of a scope. In (1.2), the fourth and the fifth lines are the binary relations of the scope ‘01’. A Scope-ID together with a colon in front (for example ‘:01’ in the third line of (1.2)) appearing in the position of a UW is called a scope node. A scope node is used as a way to refer to a scope [4]. A graph of UNL expression corresponding to (1.2) is shown in Figure 1.2.

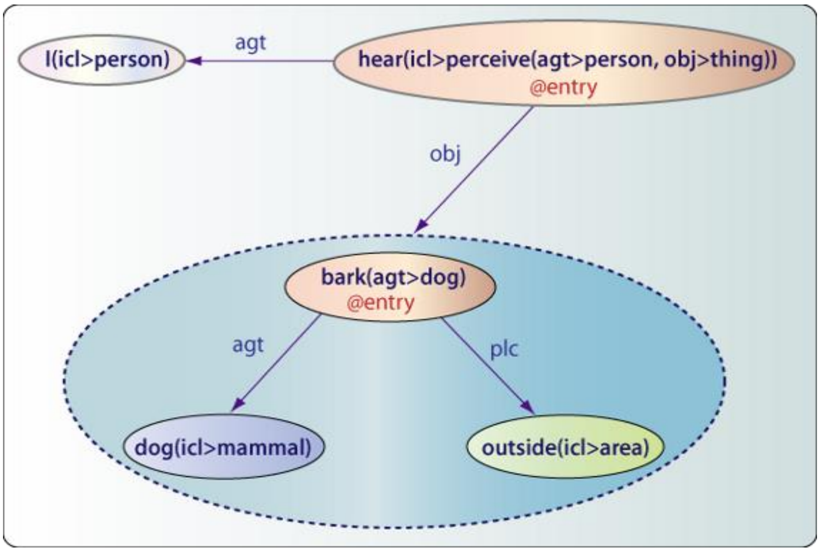

Figure 1.2: UNL graph of expression given in (1.2) [12]

1.2.3 Relations

Relations formerly known as "links", are labeled arcs connecting a node to another node in a UNL graph. They correspond to two-place semantic predicates holding between two Universal Words. In UNL, relations have been normally used to represent semantic cases or thematic roles between UWs [4].

There are 46 relations in the UNL, such as '*agt', 'gol', 'obj*', *etc*. They are used to connect every two UWs or scopes to construct the semantic networks of UNL expression. The relations are edges in the UNL graphs, or functions of the directed binary relations that constitute UNL expression [4]. A relation denotes a semantic role of a UW or scope for others. Relations in UNL is given in Table 1.2.

Table 1.2: Relations of UNL

agt	and	aoj	bas	ben	cag	cao	cnt	cob	con
coo	dur	euq	fmt	frm	gol	icl	ins	int	iof
man	met	mod	nam	obj	opl	or	per	plc	plf
plt	pof	pos	ptn	pur	qua	rsn	scn	seq	shd
src	tim	tmf	tmt	to	via				

1.2.4 UNL Sentence Structure

UNL expression are sentences of UNL. They are hyper graphs made out of universal words interlinked by binary semantic relations and modified by attributes. UNL sentences have been the basic unit of representation inside the UNL framework. According to the UNL specifications, there are two different ways of representing UNL sentences, *i.e.*, the list format and the Table format. In the list format, UWs and relations are represented separately; in the Table format, they constitute a single structure [9].

List Format

The syntax for UNL sentences in the list format is given in (1.3).

<UNL sentence> ::= "[W]" <list of UWs> "[/W]" ["[R]" <list of relations> "[/R]"]

<list of UWs> ::= <UW+attributes> [<UW+attributes>...]

<UW+attributes> ::= <UW>{":"<Scope-ID>}}[<attribute list>]":"<UW-ID>

<list of relations> ::= <binary relation>[<binary relation>...]

<binary relation> ::= <source node><relation[":"<Scope-ID>]<target node>

<source node> ::= <UW-ID>

<target node> ::= <UW-ID> …(1.3)

Table Format

The syntax for UNL sentences in the Table format is given in (1.4).

<UNL sentence> ::= <list of relations>

<list of relations> ::= <binary relation>[<binary relation>...]

<binary relation> ::= <relation> [":"<Scope-ID>] "(" <source node> , <target node> ")"

<source node> ::= <UW+attributes>

<target node> ::= <UW+attributes>

<UW+attributes> ::= <UW>{":"<Scope-ID>}[<attribute list>]":"<UW-ID>...(1.4)

Here, "and" indicate a predefined delimiter, < and > indicate a non-terminal symbol, {and} indicate a range, [and] indicate an omissible part, ... indicates more than 0 times repetition of the front part, ::= indicates the left part can be replaced by the right part.

1.3 Applications of UNL

UNL applications are application software that allow end users to accomplish natural language tasks, such as translating, summarizing, retrieving or extracting information, *etc*. They differ from UNL tools in that they do not require any expertise or even knowledge on UNL from their users [1] [10]. Some of the applications are given below.

1.3.1 LILY (Language-to-Interlingua-to-Language sYstem)

LILY is an interlingua-based human-aided multilingual machine translation web service. It is expected to provide end-to-end high quality translations through semi-automatic (human-interactive) analysis of the source text into the UNL and fully-automatic generation from the resulting UNL document onto several different target languages [10].

1.3.2 TUT (Text-to-Text through UNL)

TUT is a web service for translating, rephrasing, summarizing and simplifying documents. TUT is a digital library of texts represented in the UNL. It comprises links to the integral version of more than 30,000 titles and, whenever available, the UNL version of the text, along with three possible realizations (summarized, simplified and rephrased), in any of the languages available in the UNL System [10].

1.3.3 KEYS (Knowledge Extraction sYStem)

KEYS is a web service for extracting knowledge bases from multi documents. KEYS is an information retrieval and extraction system. It searches for information inside documents represented in UNL, *i.e.,* in semantic hyper-graphs. This allows for retrieval and extraction practices that are language-independent and semantically-oriented [10].

1.4 Tools for UNL

UNL tools are programming software, developed to assist linguists in producing UNL resources (such as dictionaries and grammars). They differ from UNL applications in that they are not tailored to non-specialists and require expertise in UNL [11].

1.4.1 NLizer

NLizers (formerly known as DeConverter) are tools for producing natural language texts out of UNL documents. One of the NLizer is described below.

- EUGUENE (dEep-to-sUrface GENErator)

  EUGENE is a natural language generation system. It generates natural language sentences out of semantic networks represented in the UNL format. In its current release, it is a web application developed in Java and available at the UNL development (a wrapper application for developing various UNL tools and applications). EUGENE is a fully automatic natural language generator. It takes a UNL input and delivers an output in natural language without any human intervention. Similarly to the UNLization tools, it is language-independent and has to be parameterized to the natural language input through a dictionary and a grammar, provided as separate interpretable files [11].

1.4.2 UNLizer

UNLizers (formerly known as EnConverter) are tools for producing UNL documents out of natural language texts. One of the UNLizer is described below.

- IAN (Interactive ANalyzer)

  IAN is a natural language analysis system. It represents natural language sentences as semantic networks in the UNL format. In its current release, it is a web application developed in Java and available at the UNL dev. (a wrapper application for developing various UNL tools and applications). Differently from the UNL editor, it includes a grammar for natural language analysis and operates semi-automatically. The word sense disambiguation is still carried out by the language specialist, but the system can filter the candidates using an optional set of disambiguation rules. The syntactic processing is done automatically through the natural language analysis grammar, but syntactic ambiguities are signaled to the user, who may backtrack and choose different syntactic paths. In any case, human interaction is always optional, and is used to improve the results. In case of no human intervention, the system simply outputs the most likely alternative, which is the one corresponding to the highest priority in the lexicon and in the grammar [11].

Chapter 2
Literature Review

Today, machine translations are becoming a fact of business life whether used by consumers or small business for ad-hoc translations, or by large global enterprises whose success depends on communicating with millions of customers, employees and partners in their native language [12]. Development of translating systems using machines has been a long cherished dream of human beings. People had been working for the realization of this dream even before the invention of computers. Researchers around the world have worked towards developing a system that would overcome language barriers. While lots of different systems have been developed by various organizations, each has their special representation of a given language. This results in incompatibilities between systems. Then, it is impossible to break language barriers in all over the world, even if we get together all the results in one system. Against this backdrop, the concept of Universal Networking Language (UNL) as a common language for all computer systems was born [13]. With the approach of UNL, the results of the past research and development can be applied to the present development, and make the infrastructure of future research and development. This chapter contains the review of literature on the efforts carried out in the area of UNL. Research activities in UNL have been discussed in next section.

2.1 Research activities in Universal Networking Language

In 1996, Institute of Advanced Studies of United Nations University, Tokyo, proposed an intermediary language, known as UNL. Initially, it was proposed for six official languages of the United Nations and other widely spoken languages, involving 15 countries. Uchida *et al.* have illustrated general idea of the UNL and its first version specifications. They have also presented the UNL system with all its components [13] [14].

Research on UNL has three distinct divisions. These divisions are: development of EnConversion and DeConversion modules; applications of UNL in other contexts such as knowledge representation and knowledge management, multilingual search engines, language-independent Universal Digital Library *etc.* and use of external lexical and ontological resources like Word Net to enhance the processes of UNL [15].

Dhanabalan and Geetha have proposed a DeConverter for Tamil language. It is a language-independent generator that provides synchronously a framework for word selection, morphological generation, syntactic generation and natural collocation necessary to form a sentence. The proposed system involves the use of language specific, linguistic based DeConversion rules to convert the UNL structure into natural language sentences [16].

Blanc has performed the integration of 'Ariane-G5' to the proposed French DeConverter. 'Ariane-G5' is a generator of MT systems. Its DeConversion process also takes place in the two steps. The first step is lexical and structural transfer from the UNL graph to an equivalent dependency tree and second step is the generation of the French sentence [17].

Boguslavsky *et al.* have proposed a multi-functional linguistic processor, 'ETAP-3', as an extension of 'ETAP' machine translation system to a UNL based machine translation system. The proposed system is used to build a bridge between UNL and the internal representations of 'ETAP', namely Normalized Syntactic Structure (NormSS). The system has performed the resolution of ambiguity with the linguistic knowledge base of 'ETAP-3'. They have also proposed an interactive system that helps to resolve difficult cases of linguistic ambiguity by means of a dialogue with the human [18].

Shi and Chen have proposed UNL DeConverter for Chinese language. They have highlighted the problems of 'DeCo' tool provided by the UNDL center which includes difficulty in writing the rules, its slow speed and non-availability of the source code. These issues motivated the developers to propose a new DeConverter for Chinese [19].

Pelizzoni and Nunes have introduced 'Manati' DeConversion model as a UNL mediated Portuguese-Brazilian sign language human-aided machine translation system. The system proposed by them is based on constraint programming to reduce search; while object-oriented and higher-order programming provides a basis for defining friendly primitives [20].

Daoud has proposed an Arabic DeConversion system which involves mapping of relations, lexical transfer, word ordering, and morphological generations. In the mapping of relations phase, each UNL relation has been mapped to the corresponding Arabic grammar structure which is implemented by DeConversion rules. Its word ordering phase is governed by DeConversion rules which are used during the insertion of a new node from the graph to the node-list. Arabic morphological generations is achieved by implementing a modular approach for coding the generation rules [21].

Keshari and Bista have proposed the architecture and design of UNL Nepali DeConverter for 'DeCo' tool. The proposed system has two major modules, namely, syntax planning module and morphology generation module [22].

Singh *et al.* have proposed a DeConverter for Hindi Language known as 'HinD', indicating the non-availability of the source code of 'DeCo' tool and its complex rule format. Their system consists of four main stages including; lexeme selection, morphological generation of lexical words, function word insertion, and syntax planning. All these components use language-independent algorithms operating on language dependent data [23].

Hameed *et al.* have proposed DeConverter framework for the Malayalam Language. It focuses on the linguistic aspects of Malayalam required for the DeConversion process. The case marking module handles the different UNL relations and their corresponding language features in Malayalam. A Morphological Generator takes care of generation of Malayalam words with the appropriate case endings and features, with respect to the cases and UNL attributes present in the UNL expression. The words are rearranged by the Syntax Planning Module to produce proper Natural Language sentences. These language dependent features for Malayalam have been successfully incorporated into a DeConverter Framework that handles English and Tamil [24].

Kumar P. has proposed a Punjabi DeConverter that generates natural language Punjabi sentence from a given input UNL expression. The architecture of Punjabi DeConverter has been divided into five phases, namely, (i) UNL parser phase, (ii) Lexeme selection phase, (iii) Morphology generation phase, (iv) Function word insertion phase and (v) Syntax planning phase. The first stage of a DeConverter is UNL parser which parses the input UNL expression to build a node-net from the input

UNL expression. During lexeme selection phase, Punjabi root words and their dictionary attributes are selected for the given UWs in the input UNL expression from the Punjabi- UW dictionary. After that, the nodes are ready for generation of morphology according to the target language in the morphology phase. The proposed system makes use of morphology rule base for Punjabi language to handle attribute label resolution morphology; relation label resolution morphology; and noun, adjective, pronoun and verb morphology. In function word insertion phase, the function words are inserted to the morphed words. These function words are inserted in the generated sentence based on nine column rule base. Finally, the syntax planning phase is used to define the word order in the generated sentence so that output matches with a natural language sentence [13].

Chapter 3
Problem Statement

Punjabi is one of the major languages spoken in India. So the problem of Punjabi language generation has its own importance. It is impossible for human to manually translate huge number of documents. A UNL based machine translation system is one of the solution for this problem. If we connect Punjabi language with all other languages across the world, *i.e.*, with *n* number of languages, then it requires *n*(n-1)* components to be developed, but if UNL approach is used then only *2n* components need to be develop [13]. One for Punjabi to interlingua and other for interlingua to any target language. At present no work has been done using EUGENE framework. In this thesis, processing of Punjabi sentences from UNL has been performed with EUGENE. In order to deal with the problem statement, objectives have been framed, discussed in next section.

3.1 Objectives

Following objectives have been framed for this thesis work:

1. To study the framework of UNL and EUGENE.
2. Creation of Punjabi language generation dictionary of Nouns, Pronouns, Prepositions and Sentence Structures.
3. To create Transformation rules and Paradigms for NLization of different part-of-speech.
4. To achieve F-measure of at least 0.8 for NLization of Nouns, Pronouns, Prepositions and Sentence Structures.

3.2 Methodology

To achieve all the objectives discussed in above section, the following methodology has been used.

1. To perform the study and analysis of framework of UNL, CUP500 certification has been performed. CUP500 is a document certifying that a person has reached a standard of knowledge concerning the UNLization of natural language documents.
2. By analyzing CORPUS500, UC-A1 and UC-A2, UNL-NL dictionary has been created, in which Nouns, Pronouns, Prepositions and Sentence Structures of Punjabi has been considered.

3. By analyzing generation features of Punjabi language, different transformation rules and paradigms have been created.
4. System generated output has been compared with expected output and F-measure has been calculated which is available at UNL development page. F-measure (or F1-score) is the measure of a grammar's accuracy [25].

The above mentioned methodologies have been followed in the NLization of Nouns, Pronouns, Prepositions and Sentence Structures to achieve the objectives defined in thesis.

Chapter 4

NLization Process

NLization, formerly known as DeConversion, is the process of generating natural language structures corresponding to UNL graphs [26]. The input for the NLization is a UNL document. The input document, which is already segmented according to the UNL document structure, is tokenized, according to the UNL-NL dictionary and the UNL-NL memory. The tokenization process may be controlled by the UNL-NL D-grammar (disambiguation Grammar). The resulting tokenized string is then syntactically and semantically analyzed with the UNL-NL T-grammar (transformation Grammar), which may be improved by the UNL-NL D-grammar, the NL memory, the UNL memory, the UNL knowledge base and the UNL ontology. The output of the process is a natural language document, which may reflect either the deep or the surface structure of the whole or part of the UNL input document [27]. NLization of UNL-document to NL-document is shown in Figure 4.1. The components used for NLization explained in next section.

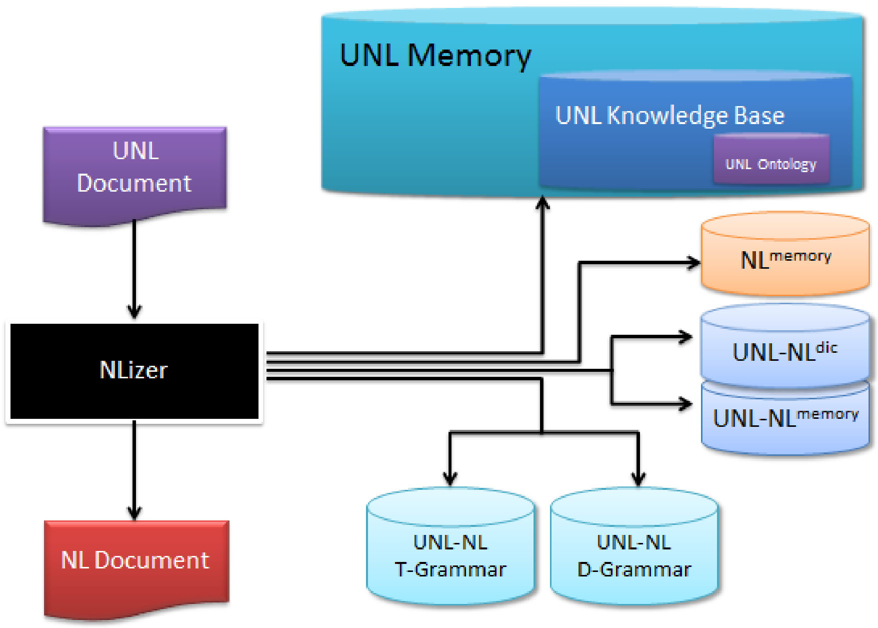

Figure 4.1: NLization of UNL-Document to NL-Document [27]

4.1 Components used in NLization

In the process of NLization, the components carried out are, UNL document, NL document, UNL memory, UNL knowledge base, NL memory, UNL-NL dictionary, UNL-NL memory, UNL-NL T-grammar, and UNL-NL D-grammar [27]. The description of these are given below.

- UNL document - UNL document is simply a collection of UNL sentences. However, it can also be treated as a hyper graph itself, comprising several inter-related sub hyper graphs.
- NL document - NL document contains the sentences of the natural language, which are generated as result of NLization process.
- UNL memory - UNL memory is a network structure where UW's are interconnected through any semantic relation of UNL.
- The UNL knowledge base (KB) - UNL knowledge base is a semantic network comprising every directed binary relation between UWs.
- NL memory – NL memory constitutes a list of syntactic frames between natural language words. They are used to represent partly or fully fixed expressions.
- UNL-NL dictionary - UNL-NL dictionary is a lexical database where UWs are mapped into natural language entries, along with the corresponding features.
- UNL-NL memory - UNL-NL memory is a set of mappings between a given natural language and UNL. It is used to improve and normalize the results of the UNLization and the NLization.
- UNL-NL T-grammar - UNL-NL T-grammar is a set of transformation rules used to convert the UNL graphs into natural language sentences.
- UNL-NL D-grammar - UNL-NL D-grammar is a set of disambiguation rules used to improve the results of the tokenization and of the transformation.

4.2 Methods of NLization

Humans and machines may play different roles in NLization methods [13]. These are given below.

- Fully automatic NLization:

  The whole process is carried out by the machine, without any intervention of the human user [27].
- Human-aided machine NLization:

  The process is carried mainly by the machine, with some intervention of the human user, either as a pre-editor or as a post-editor, or during the NLization itself [27].

- Machine-aided human NLization:

  The process is carried mainly by the human user, with some help of the machine [27].
- Fully human NLization:

  The whole process is carried by the human user, without any intervention of the machine [27].

4.3 Grammar in NLization

In the UNL framework, a grammar is a set of rules that are used to generate UNL out of natural language, and natural language out of UNL. Along with the UNL<->NL dictionaries, they constitute the basic resource for UNLization and NLization. Natural language sentences and UNL graphs are supposed to convey the same amount of information in different structures: whereas the former arranges data as an ordered list of words, the latter organizes it as a network. In that sense, going from natural language into UNL and from UNL into natural language is ultimately a matter of transforming lists into networks and vice-versa [28] [29].

The UNL framework assumes that such transformation can be carried out progressively, *i.e.*, through a transitional data structure: the tree, which could be used as an interface between lists and networks. Accordingly, there are five different types of rules given in Table 4.1.

Table 4.1: Different types of Generation Rules

S.No.	Generation Rules	Description
1.	NN	Semantic Processing (network-to-network)
2.	NT	Deep-Structure Formation (network-to-tree)
4.	TT	Syntactic Processing (tree-to-tree)
4.	TL	Surface-Structure Formation (tree-to-list)
5.	LL	List processing(list-to-list)

The details of above mentioned rules are discussed in next section.

4.4 Natural Language Generation System

During natural language generation the UNL graph is preprocessed by the NN rules in order to become a more easily traceable semantic network. The resulting network structure is converted, by the NT rules, into a syntactic structure, which is still distant from the surface structure, as it is directly derived from the semantic arrangement. This deep syntactic structure is subsequently transformed into a surface syntactic structure by the TT rules. The surface syntactic structure undergoes many other changes according to the TL rules, which generate a NL-like list structure. This list structure is finally realized as a natural language sentence by the LL rules [30] [31]. The process of natural language generation has been illustrated in Figure 4.2.

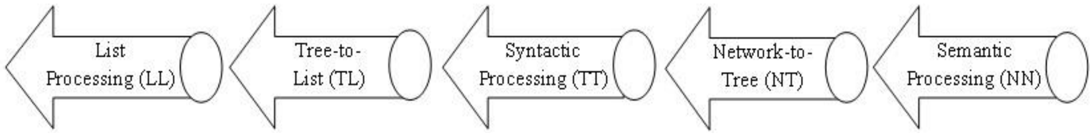

Figure 4.2: Process of natural language generation

As sentences are complex structures that may contain nested or embedded phrases, both the analysis and the generation processes may be interleaved rather than pipelined. This means that the natural flow described above is only normal and not necessary. During natural language generation, a LL rule may apply prior to a TT rule, or a NN rule may be applied after a TL rule. Rules are recursive and must be applied in the order defined in the grammar as long as their conditions are true, regardless of the state [30]. The description of rules used during natural language generation has been given in next sub-sections.

4.4.1 List-to-List Rules (LL rules)

These rules are used for processing lists. They are used for post-editing the output of the syntactic module and generating the natural language sentence [30]. There are five different subtypes of LL rules as given in Table 4.2.

Table 4.2: List of LL rules

ACTION	RULE	DESCRIPTION
ADD	(%x):=(%x)(%y);	The node %y is added to the right of the node %x
	(%x):=(%y)(%x);	The node %y is added to the left of the node %x
DELETE	(%x):=-(%x);	The node %x is deleted.
	(%x):=;	
REPLACE	(%x):=(%y);	All the instances of the node %x will be replaced by the node %y

MERGE	(%x)(%y):=(%x&%y);	The nodes %x and %y will be merged. Here, %x and %y are nodes.

4.4.2 Tree-to-Tree Rules (TT rules)

These are used for processing trees. They are used for transforming the deep into the surface syntactic structure. Syntactic relations are *n-ary*, *i.e.*, they can have as many arguments (nodes) as necessary [30]. There are four different subtypes of TT rules given in Table 4.3.

Table 4.3: List of TT rules

ACTION	RULE	DESCRIPTION
ADD RELATION	SYN1(%x;%y):=+SYN2(%w;%z)	The relation SYN2 between the nodes %w and %z will be added to the graph containing the relation SYN1 between the nodes %x and %y
DELETE RELATION	SYN(%x;%y):=-SYN(%x;%y); SYN(%x;%y)=;	The relation SYN between the nodes %x and %y will be deleted (the nodes %x and %y will not be deleted)
REPLACE RELATION	SYN1(%x;%y):=SYN2(%w;%z);	The relation SYN1 between the nodes %x and %y will be replaced by the relation SYN2 between the nodes %w and %z here SYN is a syntactic relation, and %x, %y, %z and %w are nodes.
ADD NODE	SYN(%x;%y):=SYN(%x;%y;%z)	The binary relation SYN between the nodes %x and %y is replaced by a ternary relation SYN between the nodes %x, %y and %z
DELETE NODE	SYN(%x;%y):=SYN(%y);	The binary relation SYN between the nodes %x and %y is

		replaced by a unary relation SYN with the node %y here SYN is a syntactic relation, and %x, %y and %z are nodes.

4.4.3 Network-to-Network Rules (NN rules)

These are used for processing networks. They are used for pre-editing the UNL graph, transforming it into a semantic network that would be more appropriate for sentence generation [30]. There are four different subtypes of NN rules given in Table 4.4.

Table 4.4: List of NN rules

ACTION	RULE	DESCRIPTION
ADD RELATION	SEM1(%x;%y):=+SEM2(%w;%z);	The relation SEM2 between the nodes %w and %z will be added to the graph containing the relation SEM1 between the nodes %x and %y
DELETE RELATION	SEM(%x;%y):=-SEM(%x;%y);	The relation SEM between the nodes %x and rules %y will be deleted (the nodes %x and %y will not be deleted)
	SEM(%x;%y)=;	
REPLACE RELATION	SEM1(%x;%y):=SEM2(%w;%z);	The relation SEM1 between the nodes %x and %y will be replaced by the relation SEM2 between the nodes %w and %z here SEM is any of the existing UNL relations, and %x, %y, %z and %w are nodes.

4.4.4 List-to-Tree Rules (LT rules)

These rules are used to parse the list structure into a tree structure [30]. There are two different subtypes of LT rules given in Table 4.5.

Table 4.5: List of LT rules

ACTION	RULE	DESCRIPTION
ADD	(%x)(%y):=+SYN(%x;%y);	The relation SYN is created between the nodes %x and %y if there is a linear relation between them (the linear relation is not deleted)
REPLACE	(%x)(%y):=SYN(%x;%y);	The linear relation between %x and %y is replaced by the relation SYN between the same nodes, *i.e.*, the linear relation is deleted here SYN is a syntactic relation, and %x and %y are nodes.

4.4.5 Tree-to-List Rules (TL rules)

These rules are used to linearize the tree structure into a list structure [30]. There is one single type of TL rule given in Table 4.6.

Table 4.6: List of TL rules

ACTION	RULE	DESCRIPTION
REPLACE	SYN(%x;%y):=(%x)(%y);	The relation SYN between %x and %y is replaced by a linear relation between %x and %y here SYN is a syntactic relation and %x and %y are nodes.

4.4.6 Tree-to-Network Rules (TN rules)

These rules derive a semantic network out of a syntactic tree [30]. There are two types of TN rules given in 4.7.

Table 4.7: List of TN rules

ACTION	RULE	DESCRIPTION
ADD	SYN(%x;%y):=+SEM(%w;%x);	The semantic relation SEM between the nodes %w and %x is created if there is a syntactic relation SYN

		between the nodes %x and %y
REPLACE	SYN(%x;%y):=SEM(%x;%y);	The syntactic relation SYN between the nodes %x and %y is replaced by the semantic relation SEM between the same nodes. Here, SYN is a syntactic relation, SEM is a semantic relation, and %x, %y, %w and %z are nodes.

4.4.7 Network-to-Tree Rules (NT rules)

These rules reorganizes the network structure as a deep tree structure [30]. There are two types of NT rules given in Table 4.8.

Table 4.8: List of NT rules

ACTION	RULE	DESCRIPTION
ADD	SEM(%x;%y):=+SYN(%w;%x);	The syntactic relation SYN between the nodes %w and %x is created if there is a semantic relation SEM between the nodes %x and %y
REPLACE	SEM(%x;%y):=SYM(%x;%y);	The semantic relation SEM between the nodes %x and %y is replaced by the syntactic relation SYN between the same nodes. Here, SYN is a syntactic relation, SEM is a semantic relation, and %x, %y, %w and %z are nodes.

4.5 EUGENE Framework

EUGENE is fully automatic natural language generator. It takes a UNL input and delivers an output in natural language without any human intervention. It is language-independent and has to be parameterized to the natural language input through a dictionary and a grammar, provided as separate interpretable files [31] [32]. Processing of UNL sentence is given in Figure 4.3.

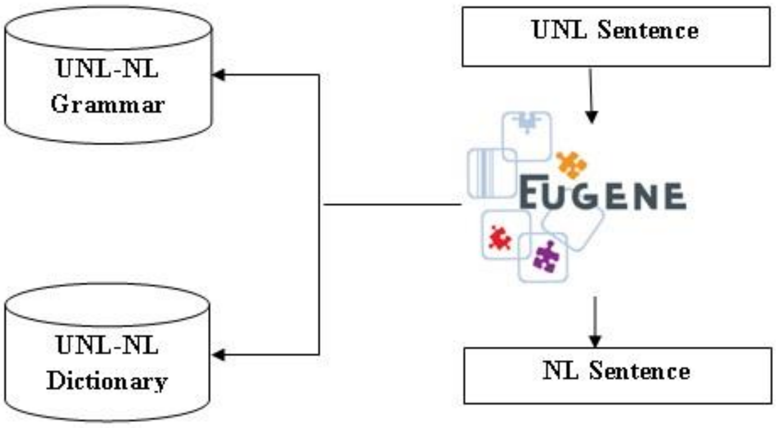

Figure 4.3: Processing of UNL sentence

4.5.1 Requirements for EUGENE

As a universal engine, EUGENE must be parameterized to the target languages with the following files, to be provided through EUGENE's interface [33].

- The input document in the UNL document structure, *i.e.*, the universal semantic network to be generated in natural language
- The UNL-NL (generation) dictionary, *i.e.*, a lexical database where UWs are mapped into natural language entries, along with the corresponding features, to be provided according to the UNL dictionary specifications.
- The UNL-NL (generation) transformation grammar, *i.e.*, a set of transformation rules used to convert the UNL graphs into natural language sentences, to be provided according to the UNL grammar specifications.
- The UNL-NL (generation) disambiguation grammar, *i.e.*, a set of disambiguation rules used to improve the results of the tokenization and of the transformation to be provided according to the UNL grammar specifications.

4.5.2 Functioning of EUGENE

EUGENE performs the three following movements over the input file [34].

- Segmentation - The division of the input document into a series of isolated graphs, which are processed one at a time.
- Tokenization - The identification of the tokens (UWs, relations and attributes) of each graph of the input document.
- Transformation - The application of the transformation rules of the grammar over each tokenized graph in order to generate a natural language sentence.

4.6 EUGENE Tabs

The fully automated web based EUGENE tool developed by UNDL foundation provides five tabs to accomplish desired task of NLization, *i.e.*, UNL input, Dictionary Tab, T-rule Tab, D-rule Tab, and EUGENE Console. Functions and snapshots of these tabs are given in next sub-sections.

4.6.1 UNL input

In this tab, a user can view UNL documents, load UNL documents, edit UNL expressions, delete existing documents, and can share files with multiple users. Snapshot of UNL input tab is given in Figure 4.4.

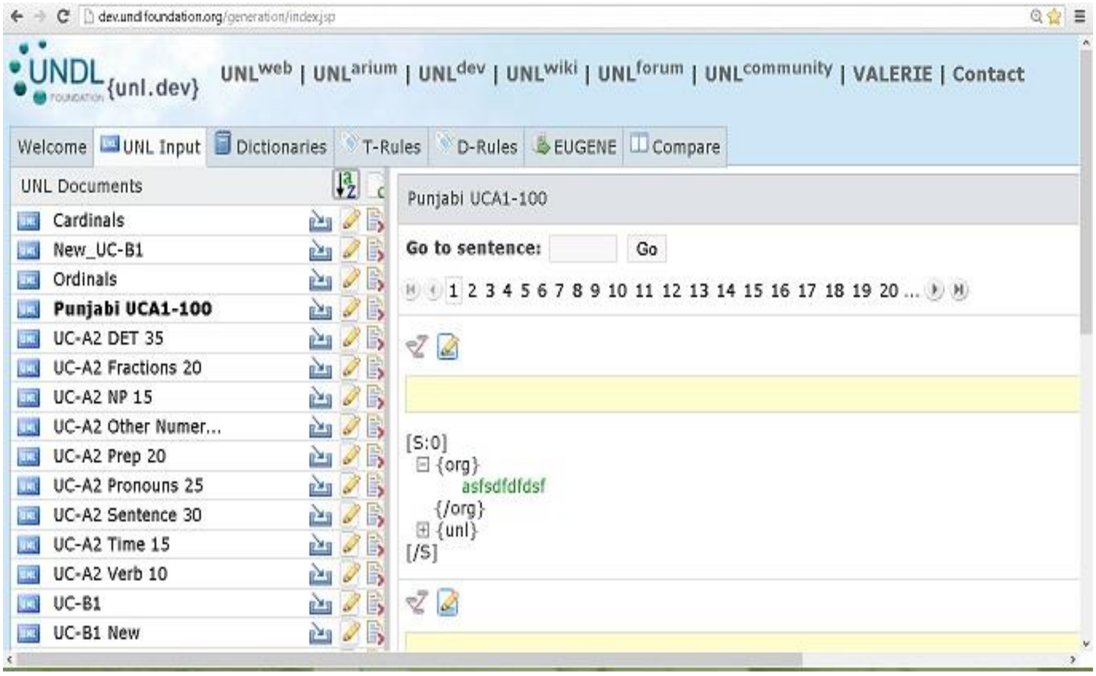

Figure 4.4: Snapshot of UNL Input tab in EUGENE

4.6.2 Dictionary tab

In this tab, UNL-NL dictionaries has been created, *i.e.*, the dictionaries to be used in natural language generation. User may either create a new file or upload an existing file. Use the default option "Database", instead of "Compiled", which are used for very big dictionaries. Once a dictionary is created/uploaded, it is ready for the use. There may have several different dictionaries, and may load many of them to process the same corpus. Snapshot of dictionary tab is shown in Figure 4.5.

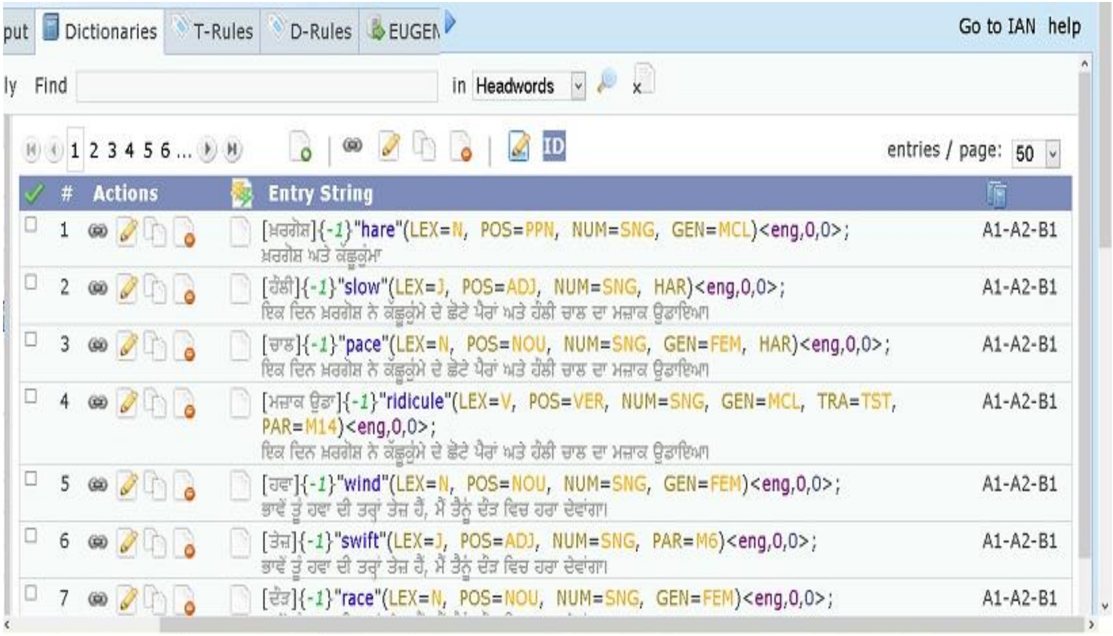

Figure 4.5: Snapshot of Dictionary tab in EUGENE

4.6.3 T-Rules tab

In this tab, UNL-NL transformation grammar rules has been created, *i.e.*, the grammar rule to be used to convert the UNL input into the NL output. User may either create a new file or upload an existing file. In this tab, the grammar must be provided according to the UNL Grammar Specifications, and must contain only transformation rules. User may have several different grammars, and may load many of them to process the same corpus. Snapshot of dictionary tab is given in Figure 4.6.

Figure 4.6: Snapshot of Transformation Rules in EUGENE

4.6.4 D-Rules tab

In this tab, UNL-NL disambiguation grammar has been created, *i.e.*, the grammar to be used to control the tokenization and improve the results of the transformation grammar. User may have several different grammars, and may load many of them to process the same corpus.

4.6.5 EUGENE console

This tab displays the final output in the form of natural language generated sentence. It brings the list of sentences, which may be processed one at a time, or in a range. The results are displayed in five different trace levels. Snapshot of EUGENE console is given in Figure 4.7.

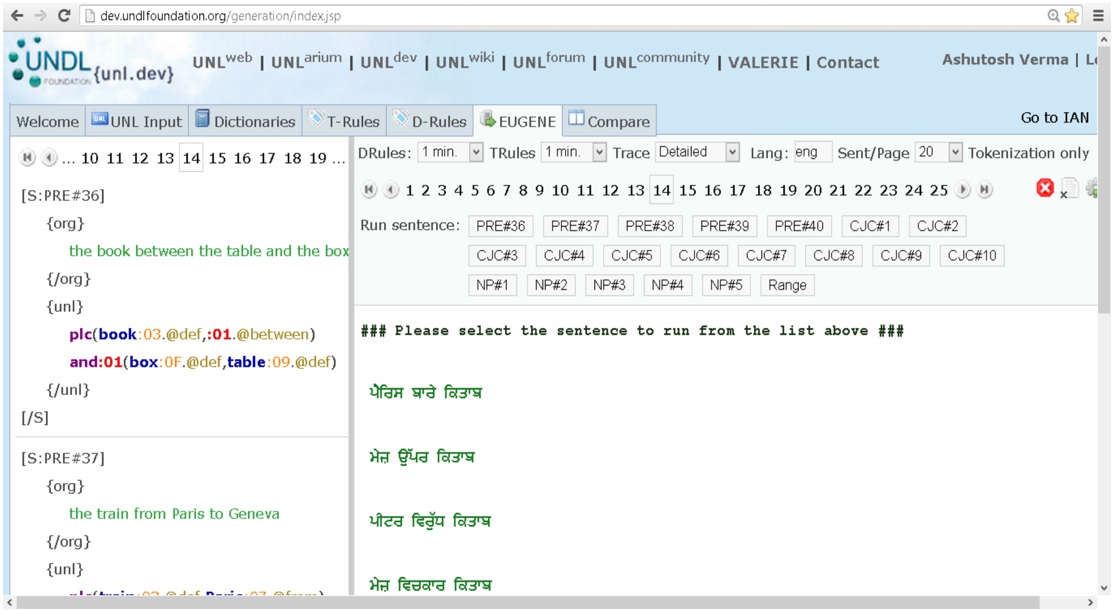

Figure 4.7: Snapshot of getting NL sentence, by processing UNL sentence one at a time

Chapter 5

Implementation of NLization of Nouns, Pronouns, Prepositions and Sentence Structures for Punjabi Language

CORPUS500, UC-A1, and UC-A2 has been NLized to Punjabi language using EUGENE tool. These corpus covers all the major part-of-speech of a natural language, from which NLization of Nouns, Pronouns, Prepositions and Sentence Structures are covered in this thesis work. In order to process UNL document with EUGENE the system requires three types of input, *i.e.*, UNL input document, dictionary of UWs which carries the semantic information of the all the UW's and Transformation rules(T-Rules) for the processing of input UNL sentences. The process of NLization has been explained with an example UNL expression given in (5.1). The Punjabi output generated by EUGENE tool is given in (5.2).

Input UNL sentence

{unl}

mod(car:03,beautiful:01)

{/unl} … (5.1)

Output Punjabi sentence

ਸੋਹਣੀ ਗੱਡੀ … (5.2)

In the proposed system, entire processing of UNL sentence involves many phases. The description of these phases are given below.

- Identifying nodes - In this phase, all the universal words, relations, and attributes are identified.
- Dictionary lookup – In this phase, meanings of universal words are extracted from dictionary, which shows the lexical category and other semantic information of the words.
- Execution of T-Rules – In this phase, all the relevant T-rules are fired, processing of the sentence has been performed and shows the intermediate results.
- Final result – After processing of all the above phases final output is generated.

5.1 NLization of CORPUS500, UC-A1 and UC-A2

CORPUS500, UC-A1 and UC-A2 are the corpuses provided by UNDL Foundation comprising of approximate 950 sentences covering all the major part-of-speech. These corpus has been categorized into different categories. These are given in Table 5.1.

Table 5.1: Different part-of-speech

S. No.	Part-of-speech
1.	Adjectives
2.	Conjunctions
3.	Determiners
4.	Nouns
5.	Numbers and Ordinals
6.	Prepositions
7.	Pronouns
8.	Sentence structures
9.	Time
10.	Verbs

In this thesis the scope has been limited to Nouns, Pronouns, Prepositions and Sentence Structures. All the Nouns, Pronouns, Prepositions and Sentence Structures available in CORPUS500, UC-A1 and UC-A2 has been processed in this work. The details of NLization of these sentences has been given in next section.

5.2 NLization of Nouns

Nouns are classified in two different categories Common nouns (NOU) which describe a class of entities such as "city", "planet", "person", "language", *etc*. and Proper nouns (PPN) represent unique entities such as "London", "Jupiter", "Johnny", "English", *etc*. The process of NLization of Noun has been illustrated with the help of simple example sentence given in (5.3).

Example 1: John's Car

UNL expression:

{unl}

pos (car:04, John:01)

{/unl} … (5.3)

After the tokenization of example sentence given in (5.3) with EUGENE tool, two lexical items are identified as shown below.

[ਜਾਨ]{}"John"(LEX=N,POS=PPN,GEN=MCL,NUM=SNG)<pun,0,0>;

[ਗੱਡੀ]{}"car" (LEX=N,POS=NOU,GEN=FEM,PAR=M3)<pun,0,0>;

Here, LEX represents lexical category, N represents noun, POS represents part-of-speech, its value could be either PPN or NOU, PPN indicates proper noun, NOU indicates common noun. GEN represents gender whose value could be either MCL for male or FEM for female, NUM represents number whose value could be either SNG for singular or PLR for plural, PAR represents paradigm which is used to generate the inflected forms out of the base form. In <pun,0,0> 'pun' refers to the three-character language code for Punjabi according to ISO 639-3. First '0' represents the frequency of Natural Language Word (NLW) in natural texts. The second '0' refers to the priority of the NLW. The process of NLization of example sentence given in (5.3) has been illustrated in Table 5.2.

Table 5.2: NLization of example sentence given in (5.3)

S.No	Rule Fired	Description	Action Performed
1.	(%x,M3):=(%x,-M3,+FLX(SNG:=0>"";PLR:=0>"ਆਂ";));	This paradigm M3 has been defined to attach word "ਆਂ" with node %x if it is plural, otherwise it remains same.	No action Performed
2.	({N|V|D|J},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule M3 to inflect the root word. Inflectional paradigms are sets of rules used to generate the inflected forms out of the base form.	**UW View:** pos(car:04,John:01) **String View:** pos("ਗੱਡੀ":04,"ਜਾਨ" :01) There is no change in the UW "car" because it is singular.
3.	pos(%a,N,FEM;%b,N):=(%b)(" ")("ਦੀ")(" ")(%a);	This rule places the node %b before node %a and insert the word "ਦੀ" betw-een them. *"pos"* indicates the possessor of a thing where UW2 is the	**UW View:** #L(John:01, -:02) #L(-:02, ਦੀ:03) #L(ਦੀ:03, -:05) #L(-:05, car:04)

		possessor of UW1.	**String View:** #L("ਜਾਨ":01," ":02) #L(" ":02, "ਦੀ":03) #L("ਦੀ":03, " ":05) #L(" ":05,"ਗੱਡੀ":04) **Generated output:** ["ਜਾਨ"][" "]["ਦੀ"][" "]["ਗੱਡੀ"] As a result, relation *"pos"* is resolved; "John" is placed before "car" with word "ਦੀ" between them.

Natural language output generated by EUGENE is given in (5.4), corresponding to UNL sentence given in (5.3).

{org}

John's car

{/org}

{pun}

ਜਾਨ ਦੀ ਗੱਡੀ

{/pun}

{unl}

pos(car:04, John:01)

{/unl} … (5.4)

NLization of Nouns has further been explained with the help of bit tougher example given in (5.5).

Example 2: All the books available

UNL expression:

{unl}

mod(book:05.@def.@all,available:07)

{/unl} … (5.5)

After the tokenization of example sentence given in (5.5) with EUGENE tool, two lexical items are identified as shown below.

[ਉਪਲਬਧ]{}"available"(LEX=J,POS=ADJ)<pun,0,0>;

[ਕਿਤਾਬ]{}"book"(LEX=N,POS=NOU,NUM=SNG,GEN=FEM,PAR=M2)<pun,0,0>;

Here, LEX represents lexical category, J represents adjective, POS represents part-of-speech, its value could be either ADJ or NOU, ADJ indicates adjective, NOU indicates common noun. GEN represents gender whose value could be either MCL for male or FEM for female, NUM represents number whose value could be either SNG for singular or PLR for plural, PAR represents paradigm which is used to generate the inflected forms out of the base form. The process of NLization of example sentence given in (5.5) has been illustrated in Table 5.3.

Table 5.3: NLization of example sentence given in (5.5)

S.No	Rule Fired	Description	Action Performed
1.	mod(%a,N,@all;%b,J):=((%b)(" ")(%a,+NUM=PLR,-@all),+@all,+LEX= %a,+NUM=%a,+GEN=%a);	Modification indicates a thing that restricts a focused thing. Here, UW1 is the focused thing to be restricted by UW2, and UW2 is a restriction or a thing that restrict UW1 in some way. This rule resolves the *"mod"*, *i.e.*, modification relation between the nodes. It creates two nodes and place node %b before node %a.	**UW View:** L:01(available:07, -:02) #L:01(-:02, book:05.@def) **String View:** #L:01("ਉਪਲਬਧ":07, " ":02) #L:01(" ":02, "ਕਿਤਾਬ":05.@def) **Generated output:** sc:01{["ਉਪਲਬਧ"][" "]["ਕਿਤਾਬ"]} Relation *"mod"* is resolved; "available" is placed before "book" with [" "] as blank space between them. Node: 01 indicates scope node internally generated by EUGENE.
2.	(N,SNG,FEM,@all,%a):=("ਸਾਰੀਆਂ")(" ")(%a,-@all ,-NUM,+NUM =PLR);	This rule resolves the attribute "@all" to insert word "ਸਾਰੀਆਂ"	**UW View:** #L:01(available:07, -:02) #L:01(-:02, book:05.@def)

		before the node %a, which is having a lexical category noun and number value singular.	**String View:** #L("ਸਾਰੀਆਂ":03, " ":04) #L(" ":04, :01) #L:01("ਉਪਲਬਧ":07, " ":02) #L:01(" ":02, "ਕਿਤਾਬ":05.@def) **Generated output:** ["ਸਾਰੀਆਂ"][" "]["ਉਪਲਬਧ"][" "] ["ਕਿਤਾਬ"] As a result, "@all" attribute is resolved, and blank space [" "] is inserted between UWs.
3.	(%a,@def):=(%a,-@def);	This rule resolves the attribute "@def" and removes the word "the" from english sentence, *i.e.*, "the books".	**UW View:** #L(-:04, :01) #L:01(available:07, -:02) #L:01(-:02, book:05.@def) **String View:** #L("ਸਾਰੀਆਂ":03, " ":04) #L(" ":04, :01) #L:01("ਉਪਲਬਧ":07, " ":02) #L:01(" ":02, "ਕਿਤਾਬ":05) **Generated output:** ["ਸਾਰੀਆਂ"][" "]["ਉਪਲਬਧ"][" "] ["ਕਿਤਾਬ"] As a result, "@def" attribute is resolved with UW "book", and blank space [" "] is inserted between UWs.
4.	(%x,M2):=(%x,-M2,+FLX (SNG:=0>"";PLR:=0>"	This paradigm M2 has been defined to attach	

	ਾਂ";));	word "ਾਂ" with node %x if it is plural, otherwise it remains same.	No action performed
5.	({N|V|D},FLX,^inflected, %x):=(!FLX,-FLX,+inflec ted,%x);	It fires the corresp-onding paradigm rule M2, to inflect the root word “book”. Inflecti-onal paradigms are sets of rules used to generate the inflected forms out of the base form.	**Generated output:** ["ਸਾਰੀਆਂ"][" "]["ਉਪਲਬਧ"][" "] ["ਕਿਤਾਬਾਂ"] There is a change in the UW "book" because it is plural.

Natural language output generated by EUGENE is given in (5.6), corresponding to UNL sentence given in (5.5).

{org}

All the books available

{/org}

{eng}

ਸਾਰੀਆਂ ਉਪਲਬਧ ਕਿਤਾਬਾਂ

{/eng}

{unl}

pos(car:04, John:01)

{/unl} … (5.6)

NLization of Nouns has further been explained with the help of bit complex example given in (5.7).

Example: Many new beautiful expensive cars

UNL expression:

{unl}

mod(:02, new:03)

mod:02(:01, beautiful:05)

mod:01(car:09.@multal,expensive:07)

{/unl} … (5.7)

After the tokenization of example sentence given in (5.7) with EUGENE tool, four lexical items are identified as shown below.

[ਨਵੀਂ]{}"new"(LEX=J,GEN=FEM,NUM=SNG,POS=ADJ,FLX(SNG:=0>"";PLR:=0>"ਆਂ";))<pun,0,0>;

[ਗੱਡੀ]{}"car" (LEX=N,POS=NOU,FEM,PAR=M3)<pun,0,0>;

[ਸੋਹਣੀ]{}"beautiful"(LEX=J,POS=ADJ,GEN=FEM,NUM=SNG,FLX(SNG:=0>"";PLR:=0>"ਆਂ";))<pun,0,0>;

[ਮਹਿੰਗੀ]{}"expensive"(LEX=J,POS=ADJ,NUM=SNG,FLX(SNG:=0>"";PLR:=0>"ਆਂ";))<pun,0,0>;

Here, LEX represents lexical category, J represents adjective, POS represents part-of-speech, its value could be either ADJ or NOU, ADJ indicates adjective, NOU indicates common noun. GEN represents gender whose value could be either MCL for male or FEM for female, NUM represents number whose value could be either SNG for singular or PLR for plural, PAR represents paradigm which is used to generate the inflected forms out of the base form. Here FLX, *i.e.*, inflection is merged with the dictionary entry of new, beautiful and expensive. FLX modify the universal word if it is a plural, otherwise it remains the same. The process of NLization of example sentence given in (5.7) has been illustrated in Table 5.4.

Table 5.4: NLization of example sentence given in (5.7)

S.No.	Rule Fired	Description	Action Performed
1.	mod(%a,N,@multal;%b,J):=((%b,+NUM=PLR)("")(%a,+NUM=PLR,-@multal),+@multal,+LEX=%a,+NU M=%a, +GEN=%a);	This rule resolves the *mod, i.e.,* modification relation between the nodes. It creates two nodes and place node %b before node %a. Modification indicates a thing that restricts a focused thing. Here, UW1 is the focused thing to be restricted by UW2, and UW2 is	**UW View:** #L:03(expensive:07, -:02) #L:03(-:02, car:09) **String View:** #L:03("ਮਹਿੰਗੀ":07, " ":02) #L:03(" ":02, "ਗੱਡੀ":09) **Generated output:** sc:03{["ਮਹਿੰਗੀ"][" "]["ਗੱਡੀ"]} Relation *"mod"* is resolved; "expensive" is placed before "car" with [" "] as space

		a restriction or a thing that restricts UW1.	between them. Node: 03 indicates scope node internally generated by EUGENE.
2.	mod(%a,N,@multal;%b,J):=((%b,+NUM=PLR)("")(%a,+NUM=PLR,-@multal),+@multal,+LEX=%a, +NUM=%a, + GEN=%a);	This rule resolves the mod, *i.e.*, modification relation between the nodes. It creates two nodes and place node %b before node %a and make it a single node.	**UW View:** #L:04(beautiful:05, -:06) #L:04(-:06, :03) #L:03(expensive:07, -:02) #L:03(-:02, car:09) **String View:** #L:04("ਸੋਹਣੀ":05, " ":06) #L:04(" ":06, :03) #L:03("ਮਹਿੰਗੀ":07, " ":02) #L:03(" ":02, "ਗੱਡੀ":09) **Generated output:** sc:04{ ["ਸੋਹਣੀ"][" "]["ਮਹਿੰਗੀ"][" "] ["ਗੱਡੀ"]} Relation *"mod"* is resolved; "beautiful" is placed before "expensive car" with [" "] as space between them. Node: 04 indicates scope node internally generated by EUGENE.
3.	mod(%a,N,@multal;%b,J):=((%b,+NUM=PLR)("")(%a,+NUM=PLR,-@multal),+@multal,+LEX=%a, +NUM=%a, +GEN=%a);	This rule resolves the mod, *i.e.*, modification relation between the nodes. It creates two nodes and place node %b before node %a. Modification indicates a thing that restricts a focused thing. Here,	**UW View:** #L:05(new:03, -:0A) #L:05(-:0A, :04) #L:04(beautiful:05, -:06) #L:04(-:06, :03) #L:03(expensive:07, -:02) #L:03(-:02, car:09) **String View:** #L:05("ਨਵੀਂ":03, " ":0A)

		UW1 is the focused thing to be restricted by UW2, and UW2 is a restriction or a thing that restricts UW1.	#L:05(" ":0A, :04) #L:04("ਸੋਹਣੀ":05, " ":06) #L:04(" ":06, :03) #L:03("ਮਹਿੰਗੀ":07, " ":02) #L:03(" ":02, "ਗੱਡੀ":09) **Generated output:** sc:05{ ["ਨਵੀਂ"] [" "] ["ਸੋਹਣੀ"] [" "] ["ਮਹਿੰਗੀ"] [" "] ["ਗੱਡੀ"] Relation "*mod*" is resolved; "new" is placed before "beautiful expensive car" with [" "] as space between them. Node: 05 indicates scope node internally generated by EUGENE.
4.	(%a,N,@multal):=("ਕਈ") (%a ,-@multal);	It resolves "@multal" attribute with the node %a having lexical category Noun. It inserts node "ਕਈ" before node %a.	**UW View:** #L(ਕਈ :0B, :05) #L:05(new:03, -:0A) #L:05(-:0A, :04) #L:04(beautiful:05, -:06) #L:04(-:06, :03) #L:03(expensive:07, -:02) #L:03(-:02, car:09) **String View:** #L("ਕਈ ":0B, :05) #L:05("ਨਵੀਂ":03, " ":0A) #L:05(" ":0A, :04) #L:04("ਸੋਹਣੀ":05, " ":06) #L:04(" ":06, :03) #L:03("ਮਹਿੰਗੀ":07, " ":02)

			#L:03(" ":02, "ਗੱਡੀ":09) **Generated output:** ["ਕਈ"] ["ਨਵੀਂ"] [" "] ["ਸੋਹਣੀ"] [" "] ["ਮਹਿੰਗੀ"] [" "] ["ਗੱਡੀ"] As a result, "@multal" attribute is resolved, "many" is placed before "new beautiful expensive car with [" "] as blank space between them.
5.	(%x,M3):=(%x,-M3,+FLX(SNG:=0>"";PLR:=0>"ਆਂ";MCL:=1>"ਾ"));	This paradigm M3 has been defined to attach word "ਆਂ" with node %x if it is plural, and attach word "ਾ" if it is a masculine, otherwise it remains same.	No action performed
6.	({N|V|D|J},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule M3, to inflect the root words. Inflectional paradigms are sets of rules used to generate the inflected forms out of the base form.	**UW View:** [new:03] [beautiful:05] [expensive:07] [car:09] **String View:** ["ਨਵੀਂਆਂ":03] ["ਸੋਹਣੀਆਂ":05] ["ਮਹਿੰਗੀਆਂ":07] ["ਗੱਡੀਆਂ":09] **Generated output:** ["ਕਈ"] ["ਨਵੀਂਆਂ"] [" "] ["ਸੋਹਣੀਆਂ"] [" "] ["ਮਹਿੰਗੀਆਂ"][" "]["ਗੱਡੀਆਂ"] As a result, UWs "new", "beautiful", "expensive", and

			“car” are effected by paradigm M3.

Natural language output generated by EUGENE is given in (5.8), corresponding to UNL sentence given in (5.7).

{org}
many new beautiful expensive cars
{/org}
{eng}
ਕਈ ਨਵੀਂਆਂ ਸੋਹਣੀਆਂ ਮਹਿੰਗੀਆਂ ਗੱਡੀਆਂ
{/eng}
{unl}
mod(:02, new:03)
mod:02(:01, beautiful:05)
mod:01(car:09.@multal, expensive:07)
{/unl} … (5.8)

5.3 NLization of Pronouns

Pronouns are Lexical Realization Unit (LRUs) that substitute other LRUs. LRU is the natural language counterpart to a UW. It can be a sub word, a simple word or a multiword expression. Pronouns are classified as personal pronoun (PPR) ("I", "me"), interrogative (IPR) (“who”, in "who is there?"), and reflexive (FPR) (“myself”, in "I saw myself"). The process of NLization of Pronoun has been illustrated with the help of simple example sentence given in (5.9).

Example 1: A book of mine
UNL expression:
{unl}
pos (book:03.@indef, 00:07.@1)
{/unl} … (5.9)

After the tokenization of example sentence given in (5.9) with EUGENE tool, two lexical items are identified as shown below.

[ਮੇਰੀ]{}"00.@1" (LEX=D,POS=POD,PAR=M3)<eng,255,0>;

[ਕਿਤਾਬ]{}"book"(LEX=N,POS=NOU,NUM=SNG,GEN=FEM,PAR=M2)<pun,0,0>;

Here, LEX represents lexical category, D represents determiner, POS represents part-of-speech whose value is POD which indicates possessive determiner. GEN represents gender value whose value could be either MCL for male or FEM for female, NUM represents number whose value could be either SNG for singular or PLR for plural, PAR represents paradigm which is used to generate the inflected forms out of the base form. The process of NLization of example sentence given in (5.9) has been illustrated in Table 5.5.

Table 5.5: NLization of example sentence given in (5.9)

S.No.	Rule Fired	Description	Action Performed
1.	(%x,M2):=(%x,-M2, + FLX(SNG:=0>"";PLR :=0>“◌ਾ◌ਂ";));	This paradigm has been defined to attach word "◌ਾ◌ਂ" with node %x if it is plural, otherwise it remains same.	No action performed
2.	(%x,M3):=(%x,-M3,+ FLX(SNG:=0>" "; PL R: =0>"ਆਂ";));	This paradigm M3 has been defined to attach word "ਆਂ" with node %x if it is plural, otherwise it remains same.	No action performed
3.	({N|V|D|J},FLX,^infle cted,%x):=(!FLX,-FL X,+inflec ted,%x);	It fires the corresponding paradigm M2 and M3 to inflect the root word.	There is no change in the UW “00:07.@1” and "book" because these are singular.
4.	pos(%a,N;%b,POD):= (%b)("") (%a);	This rule places the node %b before node %a. *“pos”* indicates the possessor of a thing where UW2 is the possessor of UW1.	**UW View:** #L(00:07.@1,-:01) #L(-:01,book:03. @indef) **String View:** #L("ਮੇਰੀ":07.@1," ":01) #L(" ":01,"ਕਿਤਾਬ":03. @indef) **Generated output:**

<table>
<tr><td></td><td></td><td></td><td>["ਮੇਰੀ"] [" "] ["ਕਿਤਾਬ"]

As a result, "pos" relation is resolved; UW "00:07.@1" is placed before "book". [" "] indicates a blank space between UWs.</td></tr>
<tr><td>5.</td><td>(N,@indef,%a):=(("ਇ ਕ")("")(%a,-@indef), %a,+NA, -@indef);</td><td>This rule inserts the word "ਇਕ" before node %a which is a noun and it also resolves the attribute "@indef".</td><td>UW View:
#L(00:07.@1, -:01)
#L(-:01, :01)
#L:01(ਇਕ:02, -:04)
#L:01(-:04, book:03)
String View:
#L("ਮੇਰੀ":07.@1,"":01)
#L(" ":01, :01)
#L:01("ਇਕ":02,"":04)
#L:01(" ":04,"ਕਿਤਾਬ" :03)
Generated output:
["ਮੇਰੀ"] [" "] ["ਇਕ"] [" "]
["ਕਿਤਾਬ"]
As a result, "@indef" attribute is resolved; "00:07.@1" is placed before "book" with "ਇਕ" between them.</td></tr>
</table>

Natural language output generated by EUGENE is given in (5.10), corresponding to UNL sentence given in (5.9).

{org}
a book of mine
{/org}
{eng}

ਮੇਰੀ ਇਕ ਕਿਤਾਬ

{/eng}

{unl}

pos(book:03.@indef, 00:07.@1)

{/unl} … (5.10)

NLization of Pronouns has further been explained with the help of bit tougher example given in (5.11).

Example 2: They love one another

UNL expression:

{unl}

pos (book:03.@indef, 00:07.@1)

{/unl} … (5.11)

After the tokenization of example sentence given in (5.11) with EUGENE tool, three lexical items are identified as shown below.

[ਉਹ]{}"00.@3.@pl" (LEX=R,POS=POD,PAR=M7)<pun,255,0>;

[ਇਹ]{}"00.@3"(LEX=R,POS=PPR,CAS=NOM,PER=3PS,NUM=SNG)<pun,255,0>;

[ਪਿਆਰ]{}"love"(LEX=V,POS=VER,TRA=TSTD,ATE=INF,PAR=M3)<pun,0,0>;

Here, LEX represents lexical category, D represents determiner, POS represents part-of-speech whose value is POD which indicates possessive determiner. GEN represents gender value whose value could be either MCL for male or FEM for female, NUM represents number whose value could be either SNG for singular or PLR for plural, PAR represents paradigm which is used to generate the inflected forms out of the base form. The process of NLization of example sentence (5.11) has been illustrated in Table 5.6.

Table 5.6: NLization of example sentence given in (5.11)

S.No.	Rule Fired	Description	Action Performed
1.	(%x,M3):=(%x,-M3,+FLX(SNG:=0>"";PLR:=0>"ਆਂ";));	This paradigm has been defined to attach word "ਆਂ" with node %x if it is plural, otherwise it remains same.	No action performed
2.	(%x,M7):=(%x,M7,+FLX(PAS&^ASPA&^WHEN:=0	Paradigm M7 has been defined to attach word	

	>"ਗਿਆ"; GENA:=1>"ਸਨੇ"; PRS&^ASPA:=0>"ਦਾ ਹੈ"; FUT:=0>"ੇਗਾ"; OBL:=0> "ੇ";PAS&ASPA :=0>"ਰਿਹਾ ਸੀ";PAS&WHE N&^ASPA :=0>"ਚੁੱਕਿਆ ਸੀ"; PRS&AS PA:=0>" ਰਿਹਾ ਹੈ"; PST& PLR&MCL:=0>"ਗਏ";));	corresponding to its attributes, *i.e.*, if node %x has PAS attribute then "ਗਿਆ" word is attached, if node %x has FUT attribute then "ੇਗਾ" word is attached.	No action performed
3.	agt(%a,V,@reciprocal;%b, @3,@pl):=(%b)(" ")(%a, PER=3PS);	This T-rule has been defined to resolve the *"agt"*, *i.e.*, agent relation between two UWs. It places node %b before node %a.	**UW View:** #L(00:01.@3.@pl, -:02) #L(-:02, love:03.@pres ent.@reciprocal) **String View:** #L("ਉਹ":01.@3.@pl,"":0 2 #L("":02,"ਪਿਆਰ":03.@pr esent.@reciprocal) **Generated output:** ["ਉਹ"] [" "] ["ਪਿਆਰ"] As a result, *"agt"* relation is resolved; "00:01.@3" is placed before "love" and [" "] as space is inserted between them.
4.	(V,@reciprocal,%x):=("ਇਕ ਦੂਜੇ ਨੂੰ")(" ")(%x,-@reciproc al);	This rule inserts the word "ਇਕ ਦੂਜੇ ਨੂੰ" before node %x which is a verb and it also resolves the attribute "@reciprocal".	**UW View:** #L(00:01.@3.@pl, -:02) #L(-:02, ਇਕ ਦੂਜੇ ਨੂੰ:04) #L(ਇਕ ਦੂਜੇ ਨੂੰ:04, -:05) #L(-:05,love:03@present

			String View: #L("ਉਹ":01.@3.@pl,"":0 2) #L(" ":02,"ਇਕ ਦੂਜੇ ਨੂੰ":04) #L("ਇਕ ਦੂਜੇ ਨੂੰ":04," ":05) #L(" ":05,"ਪਿਆਰ":03. @present **Generated output:** ["ਉਹ"] [" "] ["ਇਕ ਦੂਜੇ ਨੂੰ"] [" "] ["ਪਿਆਰ"] As a result, “00:01.@3” is placed before “love” with "ਇਕ ਦੂਜੇ ਨੂੰ" between them.
5.	(V,@present,ATE=INF,PER=3PS,%x):=(%x,-@present)(" ")("ਕਰਦੇ ਹਨ");	It resolves the attribute "@present" .This rule inserts the word "ਕਰਦੇ ਹਨ" before node %x which is a verb.	**UW View:** #L(00:01.@3.@pl,-:02) #L(-:02,ਇਕ ਦੂਜੇ ਨੂੰ:04) #L(ਇਕ ਦੂਜੇ ਨੂੰ:04,-:05) #L(-:05,love:03) #L(love:03,-:06) #L(-:06,ਕਰਦੇ ਹਨ:07 **String View:** #L("ਉਹ":01.@3.@pl,"":0 2) #L(" ":02,"ਇਕ ਦੂਜੇ ਨੂੰ":04) #L("ਇਕ ਦੂਜੇ ਨੂੰ":04," ":05) #L(" ":05, "ਪਿਆਰ":03)

			#L("ਪਿਆਰ":03, " ":06) #L(" ":06, "ਕਰਦੇ ਹਨ":07) **Generated Output:** ["ਉਹ"] [" "] ["ਇਕ ਦੂਜੇ ਨੂੰ"] [" "] ["ਪਿਆਰ"] [" "] ["ਕਰਦੇ ਹਨ"] As a result, “00:01.@3” is placed before "ਇਕ ਦੂਜੇ ਨੂੰ" then “love” followed by "ਕਰਦੇ ਹਨ"

Natural language output generated by EUGENE is given in (5.12), corresponding to UNL sentence given in (5.11).

{org}
they love each other
{/org}
{eng}
ਉਹ ਇਕ ਦੂਜੇ ਨੂੰ ਪਿਆਰ ਕਰਦੇ ਹਨ
{/eng}
{unl}
agt(love:03.@present.@reciprocal, 00:01.@3.@pl)
{/unl} … (5.12)

5.4 NLization of Prepositions

Prepositions are the words that shows the relationship between a noun or pronoun and other words in a sentence. The process of NLization of input UNL sentence containing Preposition to natural language sentence is illustrated with an example sentence given in (5.13).

Example 1: The book for John

UNL expression:

{unl}

ben(book:03.@def,John:07.@for)

{/unl} … (5.13)

After the tokenization of example sentence given in (5.13) with EUGENE tool, three lexical items are identified as shown below.

[ਜਾਨ]{}"John"(LEX=N,POS=PPN,GEN=MCL,NUM=SNG)<pun,0,0>;

[ਕਿਤਾਬ]{}"book"(LEX=N,POS=NOU,NUM=SNG,GEN=FEM,PAR=M1)<pun,0,0>;

Here, LEX represents lexical category, D represents determiner, POS represents part-of-speech whose value is POD which indicates possessive determiner. GEN represents gender value whose value could be either MCL for male or FEM for female, NUM represents number whose value could be either SNG for singular or PLR for plural, PAR represents paradigm which is used to generate the inflected forms out of the base form. The process of NLization of example sentence given in (5.13) has been illustrated in Table 5.7.

Table 5.7: NLization of example sentence given in (5.13)

S.No.	Rule Fired	Description	Action Taken
1.	(%x,@def):=(%x,-@def);	This rule resolves “@def” attribute to remove keyword "the" from English sentence.	**UW View:** ben(book:03.@def,John:07. @for) **String View:** ben("ਕਿਤਾਬ":03,"ਜਾਨ":07.@ for) "@def" attribute is resolved with UW "book".
2.	(N,PPN,@for,%b):=((%b ,-@for)("ਲਈ"),N,PPN, NOU=%b,DONE,%f);	This rule resolves “@for” attribute to insert word "ਲਈ" after the node %b which is having a lexical category noun and part of speech as proper noun.	**UW View:** #L:01(John:07, ਲਈ :02) **String View:** #L:01("ਜਾਨ":07, " ਲਈ ":02) **Generated output:** sc:01{["ਜਾਨ"] ["ਲਈ"]} As a result, "@for" attribute is resolved; "ਲਈ" is placed

			after "john".
3.	ben(N,NOU,%a;N,PPN, %b):=((%b)(%a),%f);	This rule resolves "*ben*", *i.e.*, beneficiary relation between the nodes. It creates two nodes and place node %b before node %a.	**UW View:** #L:02(:01, book:03) #L:01(John:07, ਲਈ :02) **String View:** #L:02(:01, "ਕਿਤਾਬ":03) #L:01("ਜਾਨ":07, " ਲਈ ":02) **Generated output:** sc:02{ ["ਜਾਨ"] ["ਲਈ"] ["ਕਿਤਾਬ"] } As a result, "ben" relation is resolved; "john" is placed before "book" with node "ਲਈ" between them.

Natural language output generated by EUGENE is given below, correspond to UNL sentence given in (5.13).

{org}

the book for John

{/org}

{eng}

ਜਾਨ ਲਈ ਕਿਤਾਬ

{/eng}

{unl}

ben(book:03.@def, John:07.@for)

{/unl} … (5.14)

NLization of Preposition has further been explained with the help of bit complex example given in (5.15).

Example 2: The train from Rome to Paris through Geneva

UNL expression:

{unl}
plc(train:03.@def, Rome:07.@from)
plc(train:03.@def, Paris:0B.@to)
plc(train:03.@def, Geneva:0F.@through)
{/unl} … (5.15)

After the tokenization of example sentence given in (5.15) with EUGENE tool, three lexical items are identified as shown below.

[ਪੈਰਿਸ]{}"Paris"(LEX=N,POS=PPN,NUM=SNGT,PAR=M0)<pun,0,0>;

[ਰੇਲ ਗੱਡੀ]{}"train" (LEX=N,POS=NOU,NUM=SNG)<pun,0,0>;

[ਰੋਮ]{}"Rome"(LEX=N,POS=PPN,NUM=SNGT,PAR=M0)<pun,0,0>;

[ਜਨੇਵਾ]{}"Geneva"(LEX=N,POS=PPN,NUM=SNGT)<pun,0,0>;

Here, LEX represents lexical category, D represents determiner, POS represents part-of-speech whose value is POD which indicates possessive determiner. GEN represents gender value whose value could be either MCL for male or FEM for female, NUM represents number whose value could be either SNG for singular or PLR for plural, PAR represents paradigm which is used to generate the inflected forms out of the base form. The process of NLization of example sentence (5.15) has been illustrated in Table 5.8.

Table 5.8: NLization of example sentence given in (5.15)

S.No.	Rule Fired	Description	Action Performed
1.	(%x,@def):=(%x,-@def);	This rule resolves "@def attribute to remove keyword "the" from English sentence.	**UW View:** plc(train:03.@def,Rome:07.@from) plc(train:03.@def,Paris:0B.@to) plc(train:03.@def,Geneva:0F.@through) **String View:** plc("ਰੇਲ ਗੱਡੀ":03, "ਰੋਮ":07.@from) plc("ਰੇਲ ਗੱਡੀ":03, "ਪੈਰਿਸ":0B.@to) plc("ਰੇਲ ਗੱਡੀ":03, "ਜਨੇਵਾ"

			:0F.@through) As a result, "@def" attribute is resolved with UW "train".
2.	(N,PPN,SNGT,@from, %a):=((%a,-@from)("ਤੋਂ "),N, PPN,SNGT, DONE,%f);	This rule resolves "@from" attribute to insert word "ਤੋਂ" after the node %a, which is having a lexical category noun and part of speech as proper noun.	**UW View:** #L:01(Rome:07, ਤੋਂ :02) **String View:** #L:01("ਰੋਮ":07, "ਤੋਂ":02) **Generated output:** sc:01{ ["ਰੋਮ"] ["ਤੋਂ"] } As a result, "@from" attribute is resolved. UW "Rome" is placed before word "ਤੋਂ".
3.	(N,PPN,SNGT,@to,%a):=((%a,@to)("ਨੂੰ"),N,PPN, SNGT,DONE,%f);	This rule resolves "@to" attribute to insert word "ਨੂੰ" after the node %a which is having a lexical category noun and part of speech as proper noun.	**UW View:** #L:02(Paris:0B, ਨੂੰ :05) **String View:** #L:02("ਪੈਰਿਸ":0B,"ਨੂੰ":05) **Generated output:** sc:02{["ਪੈਰਿਸ"] ["ਨੂੰ"]} As a result, "@to" attribute is resolved. UW "Paris" is placed before word "ਨੂੰ".
4.	(N,PPN,SNGT,@through,%a):=((%a,@through)("ਵਿਚੋਂ"),N,PPN,SNGT,DONE,%f);	This rule resolves "@through" attribute to insert word "ਵਿਚੋਂ" after the node %a which is having a lexical category noun and part of speech as proper noun.	**UW View:** #L:03(Geneva:0F,ਵਿਚੋਂ:08) **String View:** #L:03("ਜਨੇਵਾ":0F,"ਵਿਚੋਂ" :08) **Generated output:** sc:03{ ["ਜਨੇਵਾ"]["ਵਿਚੋਂ"]} As a result, "@through" attribute is resolved. UW

			"Geneva" is placed before word "ਵਿਚੋਂ".
5.	plc(N,NOU,%a;N,{PPN|NOU},%b):=((%b)(%a),%f);	This rule resolves the plc, *i.e.*, place relation between the nodes. It creates two nodes and place node %b before node %a. Place indicates a place where an event occurs. Here, UW1is an event, a state, or a thing, and UW2 is a place or thing understood as a place.	**UW View:** #L:04(:01, :02) #L:01(Rome:07, ਤੋਂ :02) #L:02(Paris:0B, ਨੂੰ :05) #L:06(:03, train:03) #L:03(Geneva:0F, ਵਿਚੋਂ :08) **String View:** #L:04(:01, :02) #L:01("ਰੋਮ":07,"ਤੋਂ":02) #L:02("ਪੈਰਿਸ":0B,"ਨੂੰ":05) #L:06(:03,"ਰੇਲ ਗੱਡੀ":03) #L:03("ਜਨੇਵਾ":0F,"ਵਿਚੋਂ ":08) **Generated output:** sc:04{ ["ਰੋਮ"] ["ਤੋਂ"] ["ਪੈਰਿਸ"] ["ਨੂੰ"] ["ਜਨੇਵਾ"] ["ਵਿਚੋਂ"] ["ਰੇਲ ਗੱਡੀ"] As aresult "plc" relation is resolved; and the output is generated.

Natural language output generated by EUGENE is given in (5.16), corresponding to UNL sentence given in (5.15).

{org}

the train from Rome to Paris through Geneva

{/org}

{eng}

ਰੋਮ ਤੋਂ ਪੈਰਿਸ ਨੂੰ ਜਨੇਵਾ ਵਿਚੋਂ ਰੇਲ ਗੱਡੀ

{/eng}

{unl}

plc(train:03.@def, Rome:07.@from)

plc(train:03.@def, Paris:0B.@to)

plc(train:03.@def, Geneva:0F.@through)

{/unl} … (5.16)

NLization of Preposition has further been explained with the help of bit complex example given in (5.17).

Example 3: The book on the Table about Paris without picture

UNL expression:

{unl}

plc (book:03.@def, Table:09.@def.@on)

cnt (book:03.@def, Paris:0D.@about)

man (book:03.@def, picture:0H.@without)

{/unl} … (5.17)

After the tokenization of example sentence given in (5.17) with EUGENE tool, four lexical items are identified as shown below.

[ਤਸਵੀਰ]{}"picture" (LEX=N,POS=NOU,NUM=SNG,PAR=M2)<pun,0,0>;

[ਪੈਰਿਸ]{}"Paris"(LEX=N,POS=PPN,NUM=SNGT,PAR=M0)<pun,0,0>;

[ਕਿਤਾਬ]{}"book"(LEX=N, POS=NOU, NUM=SNG, GEN=FEM, PAR=M2)<pun,0,0>;

[ਮੇਜ਼]{}"Table"(LEX=N, POS=NOU, NUM=SNG, PAR=M2) <pun,0,0>;

Here, LEX represents lexical category, N represents noun, D represents determiner, POS represents part-of-speech, its value could be either PPN or NOU, PPN indicates proper noun, NOU indicates common noun. GEN represents gender value whose value could be either MCL for male or FEM for female, NUM represents number whose value could be either SNG for singular or PLR for plural, PAR represents paradigm which is used to generate the inflected forms out of the base form. The process of NLization of example sentence (5.17) has been illustrated in Table 5.9.

Table 5.9: NLization of example sentence given in (5.17)

S.No.	Rule Fired	Description	Action Performed
1.	(%x,M2):=(%x,-M2,+FLX(SNG:=0>"";PLR:=0	This paradigm has been defined to attach word	

	> "◌ਾ◌ਂ";));	"◌ਾ◌ਂ" with node %x if it is plural, otherwise it remains same.	No action performed
2.	(%x,@def):=(%x,@def) ;	This rule resolves “@def” attribute to remove keyword "the" from English sentence.	**UW View:** plc(book:03.@def,table:09.@def.@on) cnt(book:03.@def,Paris:0D.@about) man(book:03.@def,picture:0H.@without) **String View:** plc("ਕਿਤਾਬ":03,"ਮੇਜ਼":09.@on) cnt("ਕਿਤਾਬ":03,"ਪੈਰਿਸ":0D.@about) man("ਕਿਤਾਬ":03,"ਤਸਵੀਰ":0H.@without) As a result, "@def" attribute is resolved with UW "book" and "train".
3.	(N,NOU,@on,%b):=((%b,-@on)("ਉੱਤੇ"),N,NOU,NUM=%b,DONE,%f);	This rule resolves "@on attribute to insert word "ਉੱਤੇ" after the node %b which is having a lexical category noun and part of speech as common noun.	**UW View:** #L:01(table:09, ਉੱਤੇ :02) **String View:** #L:01("ਮੇਜ਼":09, " ਉੱਤੇ ":02) **Generated output:** sc:01{["ਮੇਜ਼"] ["ਉੱਤੇ"]} As a result, "@on" attribute is resolved. UW "Table" is placed before word "ਉੱਤੇ".
4.	(N,NOU,@without,%b):=((%b,-@without)("ਤੋਂ	This rule resolves "@without" attribute to	**UW View:** #L:02(picture:0H, ਤੋਂ ਬਿਨਾਂ

	ਬਿਨਾਂ",%x),N,NOU, NUM=%b,DONE,%f);	insert word "ਤੋਂ ਬਿਨਾਂ" after the node %b which is having a lexical category noun and part of speech as common noun.	:05) **String View:** #L:02("ਤਸਵੀਰ":0H, " ਤੋਂ ਬਿਨਾਂ ":05) **Generated output:** sc:02{ ["ਤਸਵੀਰ"] ["ਤੋਂ ਬਿਨਾਂ"] } As a result, "@without" attribute is resolved. UW "Picture" is placed before word "ਤੋਂ ਬਿਨਾਂ".
5.	(N,PPN,SNGT,@about, %a):=((%a,@about)("ਬਾਰੇ"),N,PPN,SNGT, DONE,%f);	This rule resolves "@about" attribute to insert word "ਬਾਰੇ" after the node %b which is having a lexical category noun and part of speech as proper noun.	**UW View:** #L:03(Paris:0D, ਬਾਰੇ :07) **String View:** #L:03("ਪੈਰਿਸ":0D, " ਬਾਰੇ ":07) **Generated output:** sc:03{ ["ਪੈਰਿਸ"] ["ਬਾਰੇ"] } As a result, "@about" attribute is resolved. UW "Paris" is placed before word "ਬਾਰੇ".
6.	plc(N,NOU,%a;N,{PPN|NOU},%b):=((%b)(%a),%f);	This T-rule has been defined to resolve the *"plc"* relation between two UWs. It creates two nodes and places the node %b before node %a.	**UW View:** #L:04(:01, book:03) #L:01(table:09, ਉੱਤੇ :02) #L:03(Paris:0D, ਬਾਰੇ :07) #L:02(picture:0H, ਤੋਂ ਬਿਨਾਂ :05) **String View:** #L:04(:01, "ਕਿਤਾਬ":03)

			#L:01("ਮੇਜ਼":09, " ਉੱਤੇ ":02) #L:03("ਪੈਰਿਸ":0D, " ਬਾਰੇ ":07) #L:02("ਤਸਵੀਰ":0H, " ਤੋਂ ਬਿਨਾਂ ":05) **Generated output:** sc:04 {["ਮੇਜ਼"] ["ਉੱਤੇ"] ["ਕਿਤਾਬ"] ["ਪੈਰਿਸ"] ["ਬਾਰੇ"] ["ਤਸਵੀਰ"] ["ਤੋਂ ਬਿਨਾਂ"]} As a result, "*plc*" relation is resolved. It indicates a place where an event occurs, or a state that is true, or a thing that exists.
7.	cnt(N,NOU,%a;N,PPN, %b):=((%b)(%a),%f);	This T-rule has been defined to resolve the *"cnt"*, *i.e*, content relation between two UWs. It creates two nodes and places the node %b before node %a.	**UW View:** #L:04(:01, :03) #L:01(table:09, ਉੱਤੇ :02) #L:03(Paris:0D, ਬਾਰੇ :07) #L:02(picture:0H, ਤੋਂ ਬਿਨਾਂ :05) #L:05(:03, book:03) **String View:** #L:04(:01, :03) #L:01("ਮੇਜ਼":09, " ਉੱਤੇ ":02) #L:03("ਪੈਰਿਸ":0D, " ਬਾਰੇ ":07) #L:02("ਤਸਵੀਰ":0H, " ਤੋਂ ਬਿਨਾਂ ":05) #L:05(:03, "ਕਿਤਾਬ":03)

			Generated output: ["ਮੇਜ਼"] ["ਉੱਤੇ"] ["ਪੈਰਿਸ"] ["ਬਾਰੇ "] ["ਕਿਤਾਬ"] ["ਤਸਵੀਰ"] ["ਤੋਂ ਬਿਨਾਂ"] As a result, "cnt" relation is resolved. It indicates the content of a concept, *i.e.*, UW2 is the content or explanation of UW1.
8.	man(N,NOU,%a;N,{NOU|PPN},{SNG|SNGT},%b):=((%b)(%a),%f);	This T-rule has been defined to resolve the "*man*", *i.e*, manner relation between two UWs. It creates two nodes and places the node %b before node %a.	**UW View:** #L:04(:01, :03) #L:01(table:09, ਉੱਤੇ :02) #L:03(Paris:0D, ਬਾਰੇ :07) #L:06(:02, book:03) #L:02(picture:0H, ਤੋਂ ਬਿਨਾਂ :05) #L:05(:03, :02) **String View:** #L:04(:01, :03) #L:01("ਮੇਜ਼":09, " ਉੱਤੇ ":02) #L:03("ਪੈਰਿਸ":0D, " ਬਾਰੇ ":07) #L:06(:02, "ਕਿਤਾਬ":03) #L:02("ਤਸਵੀਰ":0H, " ਤੋਂ ਬਿਨਾਂ ":05) #L:05(:03, :02) **Generated output:** ["ਮੇਜ਼"] ["ਉੱਤੇ"] ["ਪੈਰਿਸ"] ["ਬਾਰੇ"] ["ਤਸਵੀਰ"] ["ਤੋਂ

			ਬਿਨਾਂ"] ["ਕਿਤਾਬ"] As a result, "man" relation is resolved. It indicates a way to carry out an event or the characteristics of a state.

Natural language output generated by EUGENE is given in (5.18), corresponding to UNL sentence given in (5.17).

{org}

the book on the table about Paris without picture

{/org}

{eng}

ਮੇਜ਼ ਉੱਤੇ ਪੈਰਿਸ ਬਾਰੇ ਤਸਵੀਰ ਤੋਂ ਬਿਨਾਂ ਕਿਤਾਬ

{/eng}

{unl}

plc(book:03.@def, table:09.@def.@on)

cnt(book:03.@def, Paris:0D.@about)

man(book:03.@def, picture:0H.@without)

{/unl} … (5.18)

5.5 NLization of Sentence Structures

Information conveyed by natural language is represented sentence by sentence as a hyper graph composed of a set of directed binary labeled links. The process of NLization of input UNL sentence containing sentence structure to natural language sentence is illustrated with an example sentence given in (5.19).

Example 1: The book for John

UNL expression:

{unl}

aoj(beautiful:05.@present, 00:01.@3)

{/unl} … (5.19)

After the tokenization of example sentence given in (5.19) with EUGENE tool, three lexical items are identified as shown below.

[ਇਹ]{}"0.@3"(LEX=R,POS=PPR,CAS=NOM,PER=3PS,NUM=SNG)<pun,0,0>;

[ਸੋਹਣੀ]{}"beautiful"(LEX=J,POS=ADJ,GEN=FEM,NUM=SNG,PAR=M5)<pun,0,0>;

Here, LEX represents lexical category, R represents pronoun, J represents adjective, and POS represents part-of-speech whose value is PPR which indicates proper noun. GEN represents gender value whose value could be either MCL for male or FEM for female, NUM represents number whose value could be either SNG for singular or PLR for plural, PAR represents paradigm which is used to generate the inflected forms out of the base form. The process of NLization of example sentence given in (5.19) has been illustrated in Table 5.10.

Table 5.10: NLization of example sentence given in (5.19)

S.No.	Rule Fired	Description	Action Performed
1.	(%x,M5):=(%x,-M5,+FLX(SNG:=0>"";PLR:=0>"ਆਂ";)	This paradigm has been defined to attach word "ਆਂ" with node %x if it is plural, otherwise it remains same.	No action performed
2.	aoj(%a,J;%b):=(%b)(" ")(%a);	This rule resolves "*aoj*" relation between two nodes. "*aoj*" indicates a thing that is in s state or has an attribute. Here, UW1 is an attribute or state of UW2, or UW1 is a state associated with UW2.	**UW View:** #L(00:01.@3, -:02) #L(-:02,beautiful:05.@present) **String View:** #L("ਇਹ":01.@3, " ":02) #L(" ":02,"ਸੋਹਣੀ":05.@present) **Generated output:** ["ਇਹ"] [" "] ["ਸੋਹਣੀ"] As a result, *"aoj"* relation is resolved; and *UW2-*"00:01.@3" is placed before UW1-"beautiful".
3.	(J,@present,%x):=(%x,-@present)(" ")("ਹੈ");	This rule resolves "@present" attribute with node %x having lexical category adje-	**UW View:** #L(00:01.@3, -:02) #L(-:02, beautiful:05) #L(beautiful:05, -:03)

		ctive. It inserts the node "ਹੈ" after corresponding UW .	#L(-:03, ਹੈ:04) **String View:** #L("ਇਹ":01.@3, " ":02) #L(" ":02, "ਸੋਹਣੀ":05) #L("ਸੋਹਣੀ":05, " ":03) #L(" ":03, "ਹੈ":04) **Generated output:** ["ਇਹ"] [" "] ["ਸੋਹਣੀ"] [" "] ["ਹੈ"] As a result, "@present" attribute is resolved; word "ਹੈ" is attached with UW "beautiful".

Natural language output generated by EUGENE is given in (5.20), corresponding to UNL sentence given in (5.19).

{org}

It is beautiful

{/org}

{eng}

ਇਹ ਸੋਹਣੀ ਹੈ

{/eng}

{unl}

aoj(beautiful:05.@present, 00:01.@3)

{/unl} … (5.20)

Chapter 6

Results and Discussion

This chapter presents the results and discussions on the work carried out in this thesis. This work, as mentioned earlier has focused on the NLization of Nouns, Pronouns, Prepositions and Sentence Structures. Apart from the NLization process that was followed, the outcome of NLization has also been discussed.

The testing of NLization process is an important aspect in order to establish the usefulness of the system. The evaluation of Nouns, Pronouns, Prepositions and Sentence Structures have been performed for CORPUS500, UC-A1 and UC-A2. The results of the system are very encouraging and the outputs of system are very good.

Evaluation of these sentences is performed by using F-measure. In the UNL System, the F-measure (or F1-score) is the measure of a grammar's accuracy. F-measure rates the T-rules and Dictionary written for the given sentences on the scale of (0.1 to 1.0). It considers both the precision and the recall of the grammar to compute the score, according to the formula given in (6.1).

$$\text{F-measure} = 2 \times ((\text{precision} \times \text{recall}) / (\text{precision} + \text{recall})) \qquad \ldots (6.1)$$

Precision is the number of correct results divided by the number of all returned results. Recall is the number of correct results divided by the number of results that should have been returned [36].

For calculating F-measure we require two files. First is the expected output file and second is the actual output file. Actual output file contains the result that is generated by EUGENE and expected output file contains those sentences which were manually translated earlier from English to Punjabi. The documents must be provided in plain text format (.txt) with UTF-8 encoding. The actual and the expected results must have the same number of sentences and must be aligned, the first sentence in the actual result will be compared to the first sentence in the expected result, the second with the second, and so on [25]. F-measure for CORPUS500, UC-A1 and UC-A2 is given in next sub-sections.

6.1 F-measure for CORPUS500

CORPUS500 is a set of 500 sentences that cover the most frequent semantic structures of UNL. A UNL corpus is a collection of documents written in UNL according to the UNL document structure. F-measure for CORPUS500 is given in Figure 6.1.

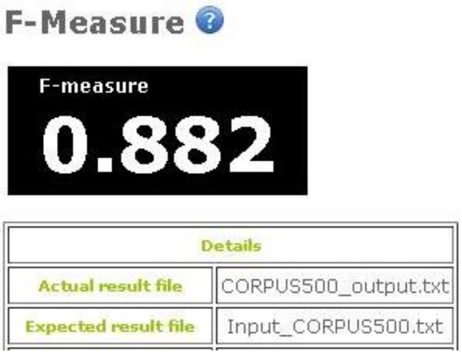

Figure 6.1: F-Measure for CORPUS500

The detailed testing results of CORPUS500 is shown in Table 6.1.

Table 6.1: Testing results of CORPUS500

Details	Count
Sentences processed	195
Sentences returned	195
Sentences correct	172
Precision	0.882
Recall	0.882

6.2 F-measure for UC-A1

The UC-A1 is an experimental corpus used to prepare the initial versions of the grammar for sentence-based NLization, using EUGENE tool. It comprises a list of 100 sentences in UNL. F-measure for UC-A1 is given in Figure 6.2.

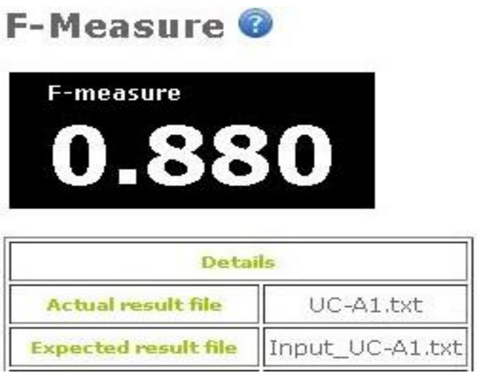

Figure 6.2: F-Measure for UC-A1

The detailed testing results of UC-A1 is shown in Table 6.2.

Table 6.2: Testing results of UC-A1

Details	Count
Sentences processed	100
Sentences returned	100
Sentences correct	88
Precision	0.880
Recall	0.880

6.3 F-measure for UC-A2

The UC-A2 is the second part of an experimental corpus used to prepare the initial versions of the grammar for sentence-based NLization, using EUGENE tool. It comprises a list of 300 sentences in UNL. F-measure for UC-A2 is given in Figure 6.3.

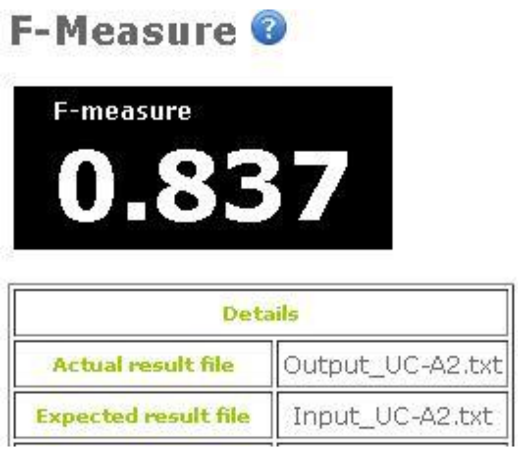

Figure 6.3: F-Measure for UC-A2

The detailed testing results of UC-A2 is shown in Table 6.3.

Table 6.3: Testing results of UC-A2

Details	Count
Sentences processed	135
Sentences returned	135
Sentences correct	113
Precision	0.837
Recall	0.837

Chapter 7
Conclusion and Future scope

7.1 Conclusion

This thesis explains the generation of natural language sentences from input UNL document. UNL based machine translation (MT) systems follows the interlingua approach of MT [13]. UNL is a computer language that enables computers to process information and knowledge. It is designed to replicate the functions of natural languages [37]. The generation process in UNL revolves around DeConverter-EUGENE. EUGENE is used to convert a given UNL expression to an equivalent natural language sentence. UNL system has the potential to bridge the language barriers across the world with the development of *2n* components, while traditional approaches require the *n*(n-1)* components, where *n* is the number of languages. UNL represents the information sentence by sentence in the form of Universal Words (UWs), UNL relations and UNL attributes. UNL-NL dictionaries and transformation grammar rules have been created, *i.e.*, the dictionaries to be used in natural language generation and grammar rule to be used to convert the UNL input into the NL output.

CORPUS500, UCA1, and UCA2 provided by UNDL foundation. With the help of transformation rules and Dictionary entries CORPUS500, UC-A1 and UC-A2 has been processed. UC-A1 and UC-A2 was done using X-Bar approach. The accuracy of the proposed system was calculated with the help of F-measure (a tool developed by UNDL Foundation) for CORPUS 500, UCA1, and UCA2. F-measure of CORPUS500, UCA1, and UCA2 came out to be 0.882, 0.880, and 0.837 respectively.

7.2 Future Scope

Some of the work that can be carried out in future includes:

- The system can further be tested on other languages by using corresponding language's rule base and lexicon.
- The rule base can further be improved linguistically to increase the accuracy of the system and to handle very large and complex sentences.
- The coverage and accuracy of system can further be improved by expanding the Punjabi-UW dictionary and enriching it with more semantic information.
- The system can further be improved to get the maximum score up to 1.000.

Appendix-A

Some of the Dictionary Words

S.No.	Dictionary Entries
1.	[ਇਕ]{}"one"(LEX=D,POS=ART,att=@indef)<pun,0,0>;
2.	[ਬਾਰੇ]{}"about"(LEX=P,POS=PRE,att=@about)<pun,0,0>;
3.	[ਸਾਰੀ]{}"all"(LEX=D,POS=QUA,GEN=FEM,att=@all,PAR=M5)<pun,0,0>;
4.	[ਅਤੇ]{}"and"(LEX=C,POS=CCJ,rel=and)<pun,0,0>;
5.	[ਇਹ]{}"it"(LEX=D,POS=DEM,NUM=SNG,att=@proximal)<pun,0,0>;
6.	[ਕੋਈ]{}"any"(LEX=D,POS=QUA,att=@any)<pun,0,0>;
7.	[ਕਿਉਂਕਿ]{}"because"(LEX=C,POS=SCJ,rel=rsn)<pun,0,0>;
8.	[ਦੁਆਰਾ]{}"through"(LEX=P,POS=PRE,rel=agt)<pun,0,0>;
9.	[ਮਰ]{}"die"(LEX=V,POS=VER,VBL=INF,TRA=NACC,PAR=M7)<pun,0,0>;
10.	[ਕੁਝ]{}"some"(LEX=D,POS=QUA,att=@paucal)<pun,0,0>;
11.	[ਲਈ]{}"for"(LEX=P,POS=PRE,rel=gol)<pun,0,0>;
12.	[ਵੱਲੋਂ]{}"from"(LEX=P,POS=PRE,rel=plf)<pun,0,0>;
13.	[ਕੋਲ]{}"have"(LEX=V,POS=VER,TRA=TSTD,VBL=INF,PAR=M0)<pun,0,0>;
14.	[ਵਿਚ]{}"in"(LEX=P,POS=PRE,rel=plc)<pun,0,0>;
15.	[ਜਾਨ]{}"John"(LEX=N,POS=PPN,GEN=MCL,NUM=SNG,PAR=M0)<pun,0,0>;
16.	[ਮਾਰ]{}"kill"(LEX=V,POS=VER,TRA=TSTD,VBL=INF,PAR=M10)<pun,0,0>;
17.	[ਰਸੋਈ]{}"kitchen"(LEX=N,POS=NOU,NUM=SNG,PAR=M6)<pun,0,0>;
18.	[ਚਾਕੂ]{}"knife"(LEX=N,POS=NOU,NUM=SNG,PAR=M8)<pun,0,0>;
19.	[ਘੱਟ]{}"less"(LEX=D,POS=QUA,att=@less)<pun,0,0>;
20.	[ਕਈ]{}"many"(LEX=D,POS=QUA,att=@paucal)<pun,0,0>;

21.	[ਮੈਰੀ]{}"mary"(LEX=N,POS=NOU,GEN=FEM,NUM=SNG)<pun,0,0>;
22.	[ਜ਼ਿਆਦਾ]{}"more"(LEX=D,POS=QUA,att=@multal)<pun,0,0>;
23.	[ਨਵੀਂ]{}"new"(LEX=J,GEN=FEM,NUM=SNG,POS=ADJ,PAR=M3)<pun,0,0>;
24.	[ਪੈਰਿਸ]{}"Paris"(LEX=N,POS=PPN,NUM=SNGT,PAR=M0)<pun,0,0>;
25.	[ਪੀਟਰ]{}"Peter"(LEX=N,POS=PPN,NUM=SNGT,PAR=M0)<pun,0,0>;
26.	[ਬਹੁਤ ਹੀ]{extra}""(LEX=D,POS=QUA,att=@extra)<pun,0,0>;
27.	[ਦੋ]{}"2"(LEX=U,POS=CDN,NUM=PLR)<pun,0,0>;
28.	[ਨਾਲ]{}"with"(LEX=P,POS=PRE,rel=ins)<pun,0,0>;
29.	[ਕਲ੍ਹ]{}"yesterday"(LEX=A,POS=ADV,SEM=TME)<pun,0,0>;
30.	[ਮੇਰੀ]{}"00.@1" (LEX=D,POS=POD,PAR=M3)<eng,255,0>;
31.	[ਤੁਹਾਡੀ]{}"00.@2" (LEX=D,POS=POD)<eng,255,0>;
32.	[ਉਹ]{}"00.@3.@female" (LEX=D,POS=POD)<eng,255,0>;
33.	[ਉਸ]{}"00.@3.@male" (LEX=D,POS=POD,R)<eng,255,0>;
34.	[ਸਾਡੀ]{}"00.@1.@pl" (LEX=D,POS=POD)<eng,255,0>;
35.	[ਉਹਨਾਂ]{}"00.@3.@pl" (LEX=D,POS=POD)<eng,255,0>;

Appendix-B

Some of the T-rules for NLization

S.No.	Transformation Rules
1.	mod(%a,N,@multal;%b,J,^PLR):=mod(%a;%b,+NUM=PLR);
2.	pos(%a,N,@pl;%b,D,^PLR):=pos(%a;%b,+NUM=PLR);
3.	(and(%a,N;%b,N),SCOPE,%C):=(((%b)(" , ")(%a)),+SCOPE=%C);
4.	mod(%a,N,NOU;SCOPE,%b):=((%b)(" ")(%a),%c);
5.	mod(%a,N;%b,@ordinal):=(("ਦੂਜੀ")(" ")(%a));
6.	qua(%a,N;%b,DIGIT):=(%b)(" ")(%a);
7.	mod(%a,N,@indef;%b,J,@almost):=((("ਲਗਭਗ")(" ")(%b)(" ")(%a)));
8.	man(%a,A;%b,J):=((%b)(" ")(%a));
9.	mod(%a,N,@def;%b,J,@least):=((("ਸਭ ਤੋ ਘਟ")(" ")(%b)(" ")(%a)));
10.	mod(%a,N,@def;%b,J,@most):=((("ਸਭ ਤੋ")(" ")(%b)(" ")(%a)));
11.	man(J,%a;A,%b):=((%b)(" ")(%a),%f);
12.	man(N,NOU,%a;N,{NOU|PPN},{SNG|SNGT},%b):=((%b)(%a),%f);
13.	(N,@pl,%a):=(%a,-NUM,NUM=PLR,-@pl);
14.	(and(%a;%b),%x):=((%b)(" ਅਤੇ ")(%a),AND,SCOPE=%x,%f);
15.	(N,@multal,%a):=(%a,-@multal,-NUM,+NUM=PLR);
16.	(N,@indef,%a):=(("ਇਕ")(" ")(%a,-@indef),%a,+NA,-@indef);
17.	(N,SNG,@pl,@other,%x):=("ਹੋਰ")(" ")(%x,-NUM,+NUM=PLR,-@pl,-@other);
18.	(N,SNG,@pl,@both,%x):=("ਦੋਵੇਂ")(" ")(%x,-NUM,+NUM=PLR,-@pl,-@both);
19.	(N,SNG,@every,%x):=("ਹਰੇਕ")(" ")(%x,-@every);
20.	(N,SNG,@wh,%x):=("ਜਿਹੜੀ")(" ")(%x,-@wh);
21.	(N,SNG,@each,%x):=("ਹੋਰ ਕੋਈ")(" ")(%x,-@each);
22.	(N,SNG,@other,%x):=("ਹੋਰ ਕੋਈ")(" ")(%x,-@other);
23.	(N,SNG,@same,%x):=("ਉਹੀ")(" ")(%x,-@same);
24.	(N,@proximal,%x):=("ਇਹ")(" ")(%x,-@proximal);

25.	(N,SNG,@distal,%x):=("ਉਹ")(" ")(%x,-@distal);
26.	(N,SNG,FEM,@all,%a):=("ਸਾਰੀਆਂ")(" ")(%a,-@all,-NUM,+NUM=PLR);
27.	(N,SNG,@any,%x):=("ਕੋਈ")(" ")(%x,-@any);
28.	(N,SNG,@paucal,%a):=("ਕੁਝ")(" ")(%a,-@paucal,-NUM,+NUM=PLR);
29.	pos(%a,@all;%b,^PLR):=pos(%a;%b,-NUM,+NUM=PLR);
30.	pos(%a,@both;%b,^PLR):=pos(%a;%b,-NUM,+NUM=PLR);
31.	pos(%a,FEM,N;%b,@3,POD):=(%b)(" ਦੀ ")(%a);
32.	pos(%a,N;%b,POD):=(%b)(" ")(%a);
33.	(N,PLR,FEM,@both,%a):=("ਦੋਵੇਂ")(" ")(%a,-@both);
34.	({N|V|D|J},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);
35.	or(%a,N;%b,N):=(%b)(" ਜਾਂ ")(%a);
36.	and(%a,N;%b,N):=(%b)(" ਅਤੇ ")(%a);
37.	(and(%a,N;%b,SCOPE),SCOPE,%C):=(((%b)(" , ")(%a)),+SCOPE=%C);
38.	(N,SNGT,@multal,%a):=("ਵਧੇਰੇ")(" ")(%a,-@multal);
39.	(N,SNG,@no,%x):=(%x,@any,-@no)(" ")("ਨਹੀਂ")(%x,-@other);
40.	(N,SNGT,@paucal,%a):=("ਥੋੜਾ ਕੁ")(" ")(%a,-@paucal);
41.	obj(%a,V;%b,N):=((%b)(" ਨੂੰ ")(%a,+NUM=%b,+GEN=%b),%a,+V);
42.	(N,SNG,@each,%x):=("ਹੋਰ ਕੋਈ")(" ")(%x,-@each);

References

[1] S. Ripon, S. M. Allayear and N. Y. Ali, "UNL Based Bangla Natural Text Conversion - Predicate Preserving Parser Approach," *International Journal of Computer Science Issues*, vol. 9, no. 3, pp. 1-7, 2012.

[2] "Universal Networking Language: Introduction to UNL," [Online]. Available: https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Universal_Networking_Language.html. [Accessed 11 May 2013].

[3] M. Z. H. Sarker and M. N. Y. Ali, "Dictionary Entries for Bangla Consonant Ended Roots in Universal Networking Language," *International Journal of Computational Linguistics*, vol. 3, no. 1, pp. 79-87, 2012.

[4] "Universal Networking Language (UNL)," [Online]. Available: http://www.undl.org/index.php?option=com_content&view=article&id=46&Itemid=63&lang=en. [Accessed 11 May 2013].

[5] A.Martins, G. Tissiani and R. M. Barcia, "A Framework for the Development of Universal Networking Language E-Learning User Interface," in Universal Networking Language: Advances in Theory and Applications, vol. 12. C. Jesus, G. Alexander and T. Edmundo, Mexico: Centre for Computing Research of IPN, 2005, pp. 268-275.

[6] P. Kumar, "UNL Based Machine Translation System for Punjabi Language," Ph.D. dissertation, Dept. Comput. Sci. Eng., Thapar University, Patiala, 2013.

[7] P. Kumar and R. K. Sharma, "Generation of UNL Attributes and resolving relations for Punjabi Enconverter," *Malaysian Journal of Computer Science*, vol. 24, no. 1, pp. 34-36, 2011.

[8] "History of UNL," [Online]. Available: http://www.unlweb.net/wiki/Introduction_to_UNL#History. [Accessed 15 May 2013].

[9] "Universal Networking Language: UNL specifications," [Online]. Available: http://www.unlweb.net/wiki/Specs. [Accessed 20 May 2013].

[10] "Universal Networking Language: UNL Specification Version 2005," [Online]. Available: http://www.undl.org/unlsys/unl/unl2005/. [Accessed 1 June 2013].

[11] "Universal Networking Language: Universal Words," [Online]. Available: http://www.undl.org/publications/UW%20and%20UNLKB.htm. [Accessed 5 June 2013].

[12] M. Monju, T. Shilpa, D. Smita and P. Bhattacharya, ”Knowledge extraction from Punjabi Text,” in Knowledge Based Computer Systems, M. Sasikumar, D. Rao and P. R. Prakash, Mumbai: Allied Publishers, 2000, pp.193-204.

[13] "Universal Networking Language: UNL Sentence," [Online]. Available: http://www.unlweb.net/wiki/UNL_sentence. [Accessed 7 June 2013].

[14] "Universal Networking Language: Applications of UNL," [Online]. Available: http://www.unlweb.net/wiki/Applications. [Accessed 9 June 2013].

[15] "Universal Networking Language: UNL Tools," [Online]. Available: http://www.unlweb.net/wiki/Tools. [Accessed 9 June 2013].

[16] "Machine Translation: Machine Translation Users," [Online]. Available: http://www.machinetranslation.net/quick-guide-to-machine-translation/machine-translation-users. [Accessed 11 June 2013].

[17] H. Uchida, M. Zhu and T. D. Senta, The UNL-A Gift for a Millennium, Tokyo, 1999, pp.1-62.

[18] T. Dhanabalan and T. V. Geetha , "UNL DeConverter for Tamil," in *International Conference on the Convergence of Knowledge, Culture, Language and Information Technologies*, Alexandria, 2003, pp. 1-6.

[19] E. Blanc, “About and Around the French Enconverter and the French Deconverter,” in Universal Networking Language: Advances in Theory and Applications, vol. 12. C. Jesus, G. Alexander and T. Edmundo, Mexico: Centre for Computing Research of IPN, 2005, pp. 157-166.

[20] M. Boguslavsky , L. L. Iomdin and V. G. Sizov, "Interactive enconversion by means of the ETAP-3 system," in *Proc. International Conference on the Convergence of Knowledge, Culture, Language and Information Technologies*, 2003, pp. 1-6.

[21] X. Shi and Y. Chen, “A UNL Deconverter for Chinese,” in Universal Networking Language: Advances in Theory and Applications, vol. 12. C. Jesus, G. Alexander and T. Edmundo, Mexico: Centre for Computing Research of IPN, 2005, pp. 167-174.

[22] J. M. Pellizoni, “Flexibility, Configurability and Optimality in UNL Deconversion via Multiparadigm Programming,” in Universal Networking Language: Advances in Theory and Applications, vol. 12. C. Jesus, G. Alexander and T. Edmundo, Mexico: Centre for Computing Research of IPN, 2005, pp. 175-194.

[23] M. D. Daoud, “Arabic Generation in the Framework of the Universal Networking Language,” in Universal Networking Language: Advances in Theory and Applications, vol. 12. C. Jesus, G. Alexander and T. Edmundo, Mexico: Centre for Computing Research of IPN, 2005, pp. 195-209.

[24] B. Keshari and S. K. Bista, "UNL Nepali Deconverter," in *3rd International CALIBER*, Coachin, 2005, pp. 70-76.

[25] S. Singh, M. Dalal, V. Vachhani, P. Bhattacharyya and O. . P. Damani , "Hindi generation from interlingua," in *11th Machine Translation Summit*, Copenhagen, 2007, pp. 1-8.

[26] M. S. Hameed, C. N. Subalalitha, T. V. Geetha and R. Parthasarathi, "A Deconverter Framework for Malayalam," in *Proc. International Conference on Advances in Computing, Communications and Informatics*, Chennai, 2012. pp.847-856.

[27] "Universal Networking Language: F-measure," [Online]. Available: http://www.unlweb.net/unlarium/index.php?grammar=fmeasure. [Accessed 15 June 2013].

[28] S. AlAnsary, "Interlingua-based Machine Translation Systems: UNL versus Other Interlinguas," in *11th International Conference on Language Engineering*, Cairo, 2011, pp. 1-11.

[29] "Universal Networking Language: NLization Process," [Online]. Available: http://www.unlweb.net/wiki/UNL_System#NLization. [Accessed 17 June 2013].

[30] S. Alansary, M. Nagi and N. Adly, "Understanding Natural Language through the UNL Grammar Work-bench," in *Conference on Human Language Technology for Development*, Egypt, 2011, pp. 118-127.

[31] "Universal Networking Language: NLization Grammar," [Online]. Available: http://www.unlweb.net/wiki/Grammar. [Accessed 19 June 2013].

[32] "Transformation Rules," [Online]. Available: http://www.unlweb.net/wiki/T-rule.

[Accessed 19 June 2013].

[33] A. Verma and P. Kumar, "NLization of Nouns, Pronouns and Prepositions in Punjabi With EUGENE," *International Journal of Natural Language Computing*, vol. 2, no. 2, pp. 33-42, 2013.

[34] "EUGENE Framework," [Online]. Available: http://dev.undlfoundation.org/. [Accessed 26 June 2013].

[35] "Universal Networking Language: Requirements for EUGENE," [Online]. Available: http://www.unlweb.net/wiki/EUGENE#Requirements. [Accessed 29 June 2013].

[36] "Universal Networking Language: Funtions of EUGENE," [Online]. Available: http://www.unlweb.net/wiki/EUGENE#Functioning. [Accessed 30 June 2013].

[37] "Universal Networking Language: Snapshot of EUGENE tabs," [Online]. Available: http://dev.undlfoundation.org/generation/index.jsp. [Accessed 1 July 2013].

[38] "Universal Networking Language: Precision and Recall in F-measure," [Online]. Available: http://www.unlweb.net/wiki/F-measure. [Accessed 1 July 2013].

[39] V. T. Hung , "A Machine Translation Dictionary for Vietnamese Language in UNL System," in *International Conference on Complex, Intelligent and Software Intensive Systems*, Seoul, 2011, pp. 310-314.

Certification

CUP500 - Certificate of Proficiency in UNL is a certificate issued by the UNDL Foundation.

UNDL FOUNDATION

Universal Networking Language

CUP 500

Certificate of Proficiency in UNL

The UNDL Foundation certifies that

Ashutosh Verma

has successfully completed the exercises leading to the Certificate of Proficiency in UNL (CUP500), offered online through VALERIE, the Virtual Learning Environment for UNL.

Geneva, October 18, 2012

Tarcisio G. Della Senta, PhD
President
UNDL Foundation

Ronaldo T. Martins, PhD
Language Resources Manager
UNDL Foundation

List of Publications

Published

A. Verma, P. Bhatia, “NLization of Nouns, Pronouns and Prepositions in Punjabi with EUGENE”, Published in “International Journal of Natural Language Computing,” vol.2, no.2, pp. 33-42, 2013.

Accepted

Ashutosh Verma, Parteek Bhatia, “NLization Framework with EUGENE”, accepted in “International Journal of Computational Linguistics and Natural Language Processing”.

Communicated

Ashutosh Verma, Parteek Bhatia, “Generation of Natural Language Sentences for Different Part-of-Speech”, communicated to “International Journal of Computer Science and Information Technologies”.