NLization of Adjectives, Conjunctions, Determiners and Verbs with EUGENE

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

Master of Engineering
in
Software Engineering

Submitted By
Harinder Singh
(801131010)

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 which is being presented in the thesis entitled, "*NLization of Adjectives, Conjunctions, Determiners and Verbs in Punjabi with EUGENE*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software 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.

Harinder Singh
(Harinder Singh)

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

Parteek Bhatia
(Dr. Parteek Bhatia)
Assistant 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

Acknowledgement

Above all, I would like to bow and salute to the destroyer of hard times and embodiment of Mercy Lord, who is ever present with all, indestructible and Glorious Lord. The moment of completing thesis work and receiving degree, that I am enjoying is due to His Grace and Blessings.

Secondly, I would also like to thank my family members who are dearest and precious to me for their love, encouragement, blessings and support in all respects.

I am extremely thankful to my respectful Guide Dr. Parteek Bhatia Asst. Prof. Computer Science and Engineering Department of Thapar University for his valuable advice, motivation, guidance, encouragement, moral support, sincere efforts and attitude with which he solved my queries and provide delightful ambiance for learning, exploring and making this thesis possible. It has been a great pleasure and experience to work under his sanctuary.

I would like to deeply express my gratitude towards Dr. Maninder Singh Associate Prof. Head of Computer Science and Engineering Department, Thapar University, Patiala, who has always been a source of motivation and hope.

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

I would like to thank my colleagues Ashutosh Verma and Vaibhav Agarwal for their continuous support and coordination by which research work involved in thesis has been progressed towards completion and helped in achieving the desired objectives. I would like to specially thank my dear friend Sukhpal Singh for his acts of kindness in achieving various milestones in my journey of ME degree and doing the things in righteous way. I would also like to say thank to my juniors whose charming faces always give motivation to be good and make this period of two years memorable for me.

Harinder Singh

Harinder Singh

Abstract

In 21st century, information and information access is a key to building prosperity and economic wealth. This is true not only in the world at large, for countries that wish to build knowledge-based economies, but is also true increasingly for enterprises involved in building global businesses. It is useful to look more closely at the forces that are driving the change as it helps us to understand where these forces are leading and also better understand the impact on the business of professional translation. Access to knowledge is one of the keys to economic prosperity. Automated translation is one of those technologies that offer a way to reduce the digital divide and raise living standards across the world. So, in this age, Machine Translation has been an area of immense interest in Natural Language Processing domain among the researchers.

Further, the most challenging task is design a system for multi-lingual machine translation environment, where large number of languages are to be translated between one-another. So, instead of following Statistical approach, an Interlingua based approach for MT is more preferable in multi-lingual machine translation environment. UNL system designed and implemented by UNDL Foundation, in Geneva, Switzerland, is an internationally standardized interlingua. The UNL is an effort to achieve a simple basis for representing the most central aspects of information and meaning in a human-language-independent form. As a knowledge representation language, the UNL aims at coding, storing, disseminating and retrieving information independently of the original language in which it was expressed. UNL approach comprises two basic different movements: UNLization and NLization. UNLization is the process of representing/mapping/analysing the information conveyed by natural language utterances into UNL; NLization, conversely, is the process of realizing/manifesting/generating a natural language document out of a UNL graph. These processes are completely independent.

The project work that has been carried out in the thesis is focused on implementation of NLization framework. EUGENE tool is a freeware online tool, provided by UNDL Foundation, which has been used for implementing NLization framework for generating Punjabi natural language from UNL expression. Implementation of NLization for

adjectives, conjunctions, determiners and verbs, in Punjabi using EUGENE has been elaborated with three examples (easy, medium and tough levels) in each case. At the end F-measure, which is verification mechanism for measuring the correctness of the resulting NL output, has been calculated for generated sentences.

Table of Content

List of Figures

List of Tables

Chapter 1

Introduction to UNL System

Universal Networking Language (UNL) is an artificial language that can be used as a pivot language in interlingual machine translation systems or as a knowledge representation language in information retrieval applications. It was created at the Institute of Advanced Studies of the United Nations University, in Tokyo, and it has been developed at the UNDL Foundation, in Geneva, Switzerland, along with a large community of researchers all over the world [1]. This system is based on Interlingual approach, the source language, *i.e.*, the text to be translated is transformed into an Interlingua, *i.e.*, an abstract language-independent representation, which is UNL knowledge representation and the target language is then generated from this representation, as shown in Figure 1.1.

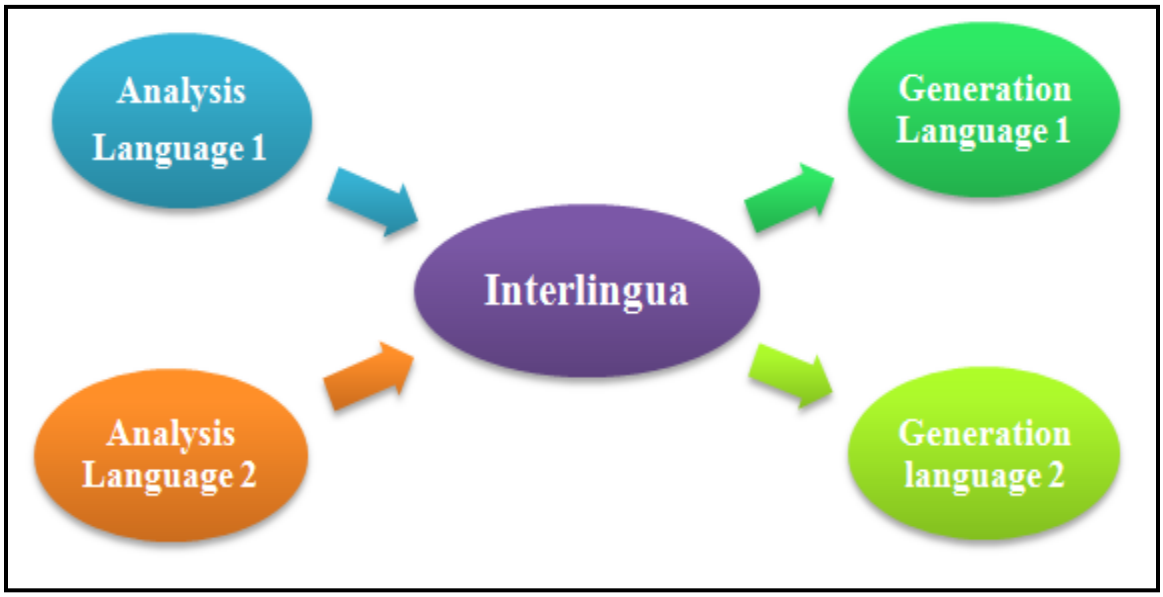


Figure 1.1: Model of Interlingual based Machine Translation

1.1 History

The UNL project came into being as a reaction to a question. Back in 1995, at the Institute of Advanced Studies (IAS) at the United Nations University in Tokyo, someone asked: "in what language should the world's first virtual university be established?" There

were the three individuals who were to become the founding members of the UNDL Foundation who proffered the reply: "in a universal language that does not yet exist". That was the commencement of the development of UNL within the setting of the IAS, a universal digital language not derived from any natural language or any culture, but there to serve all of them. For five years, the UNL system was developed in Tokyo, centered on a small founding team and benefiting from the creation of an impressive network of scientists specialized in data processing and linguistics which was brought about through international symposiums held around the world [2].

In 1999, the patent applications were filed for UNL. That was done to protect the very idea of UNL as well as the system of EnConversion and DeConversion from and to natural languages. That was when the inventors of UNL and the administrators of the IAS gave the UNL intellectual property to the United Nations Organization to make sure that their invention would benefit the entire world. In order to simplify liaison between the inventors of UNL and the owners of the rights to the invention (the UNO), it was decided to set up a non-profit private-law foundation in Geneva, Switzerland. Since 2001, Geneva has thus been the centre from where the UNL and its network have been growing [2].

Throughout these years, the UNL system has gradually been taking shape and developing. Once the UNL was patented, the UNDL Foundation was able to organize international conferences every year to increase still further the community of scientists and impassioned visionaries working on UNL. In 2005, a contract was signed between the UNO and the UNDL Foundation, authorizing the latter to keep on developing and promoting UNL under license. The Foundation is in a position to pursue its research, whilst looking for economic outlets for UNL and exploring the multiple beneficial applications of this revolutionary technology. In order to demonstrate the efficiency of its project, the UNDL Foundation has been involved in an extremely ambitious project: the translation of a 250000-pages online encyclopedia for UNESCO [2].

1.2 The UNLweb

The UNLweb is the UNL workplace; it has been created to bring together the community of researchers and developers of UNL as well as the general people interested in UNL in particular and Natural Language Processing (NLP) in general in order to share their

expertise, documentation and resources. It also aims at collecting all information and training related to the development of UNL in one place for those who are interested in joining the effort. It contains several working platforms including the UNLarium, the UNL^{dev}, the UNL^{wiki}, the UNL EDGES and VALERIE (Virtual Learning Environment) which is introduced to teach those wishing to be part of the project. Successful candidates are granted a permission to work within the UNLarium. VALERIE comprises two series of certificates; CLEA (CLEA250, CLEA450 and CLEA700) and CUP (CUP500). These certificates are required in order to be an active user and participate in the development of UNL resources. No previous knowledge in UNL or NLP is required in order to create an account in the UNLweb environment. The UNLweb is entirely free and all its results are released under an Attribution Share Alike (CC-BY-SA) Creative Commons license [3].

1.3 Language Representation Using UNL

The information conveyed by natural languages can be formally represented through a semantic network made of three different types of discrete semantic units, as given below.

- UW's
- Relations
- Attributes

1.3.1 UWs

Universal Words constitute the vocabulary of UNL. They are used to express the meaning of any concept. The UW's are the nodes in the graph, to be interlinked by relations and specified by attributes [3]. As the name indicates, Universal Words are expected to be "universal". This does not mean that they represent a sort of common lexical denominator to all languages or a semantic primitive. The concept of "universality", in UNL, must be understood in the sense of "capable of being used and understood by all", and Universal Words depict concepts that may range from absolutely global to absolutely local and even temporary. They are "universal" in the sense that they are uniform identifiers to the entities defined in the UNL Knowledge Base, which is expected to map everything that we know about the world, and that is used to assign translatability to any concept [4].

UW's may represent concepts that are believed to be lexicalized in most languages (such as "cause to die"); concepts that are lexicalized only in a few languages (such as "to execute someone by suffocation so as to leave the body intact and suitable for dissection"); concepts that are lexicalized in one single language (such as "a person who is ready to forgive any transgression a first time and then to tolerate it for a second time, but never for a third time"); and concepts that are not lexicalized in any language (such as "women that normally wear red hats and white shoes in big theaters"). The universality of a UW does not come from the type of concept that it represents, but from the way it does that: the UW provides a method for processing the concept, so that any natural language would be able to deal with it, either as a single node, if lexicalized, or as a hyper-node, *i.e.*, a sub-graph, otherwise [4].

1.3.2 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 such as agent, object, instrument *etc.* between UWs. The repertoire of relations is defined in the UNL specifications and it is not open to frequent additions [5].

Relations constitute the syntax of UNL. Relations are three-letter symbols that signify the kind of semantic relationship that ties two UWs in a natural language utterance, as shown in (1.1), as general relation.

rel(UW1, UW); …(1.1)

Examples are the agt (agent) relation, pos (possession) relation, and tim (time) relation. These relations are demonstrated in details as follow.

- **Example Sentence**: He arrived. …(1.2)

  The UNL relation for sentence (1.2) is given in (1.3):

  agt(arrive:03.@past, 00:01.@3.@male) …(1.3)

- **Example Sentence**: a lot of their books. …(1.4)

  The UNL relation for sentence (1.4) is given in (1.5):

  pos(book:05.@multal, 00:03.@3.@pl) …(1.5)

1.3.3 Attributes

Attributes are additional tags that encode the contextual and/or subjective knowledge present in the original sentence into the UNL graph. They are used to further modify the semantic network and add information that is not expressed via UWs or Relations [3].

Attributes are arcs linking a node to it. In opposition to relations, they correspond to one-place predicates, *i.e.*, functions that take a single argument. In UNL, attributes have been normally used to represent information conveyed by natural language grammatical categories, such as tense, mood, aspect, number *etc*. The set of attributes, which is claimed to be universal, is defined in the UNL specifications and is not open to frequent additions. Attributes are annotations made to nodes or hypernodes of a UNL hypergraph. They denote the circumstances under which these nodes (or hypernodes) are used [6]. Table 1.1, describes some of the UNL attributes used to represent the knowledge extracted from an input sentence.

Table 1.1: UNL Attributes [7]

Concept	**UNL Sentence**
Time with respect to the speaker	@past, @present, @future
Speaker's view of Reference	@generic, @def, @indef
Speaker's view of Quantity	@multal, @extra
Speaker's view of Reference	@pl
Speaker's view of Reference	@male, @female
Speaker's view of Reference	@habitual, @perfective and @progressive

1.3.4 UNL-Knowledge Base

The UNL Knowledge Base, or simply UNL^{KB}, constitutes a network structure where UWs are interconnected through relations of UNL. Differently from the UNL Ontology, which deals only with hierarchical relations ("icl" and "iof"), the UNL^{KB} comprises any relation necessary to define a given UW. In that sense, the UNL^{KB} contains and extends the UNL Ontology, and it is expected to include all the information normally available in ordinary dictionaries. The UNL^{KB} is also a part of the UNL Memory, a network structure that includes not only the essential information about a concept but also the accidental information extracted from corpora [8].

The UNL^{KB} is a semantic network comprising every directed binary relation between UWs. All the binary relations of the UNL^{KB} are in the format shown in (1.6).

relation(UW1, UW2)=c; …(1.6)

Here 'c' is the degree of certainty, which has the value 0 (impossible) or 1 (certain). This binary relation means "UW1 takes UW2 as the relation in certainty value c", or "UW2 plays the role of relation for UW1 in certainty value c" [9]. The format of knowledge-base entries is given in (1.7).

<knowledge-base entry> ::= <binary relations> "=" <degree of certainty>
<binary relation> ::= <relation label> "(" <UW1> "," <UW2> ")"
<degree of certainty> ::= "0" | "1" | ... | "255" …(1.7)

Here, "::=" implies that left part can be replaced by the right part and "0" degree of certainty implies that the relation between two UWs is false and degree of certainty equal to or more than "1" implies that the relation between two UWs is true, and larger numbers represent higher credibility. Following this scheme, the knowledge-base of sentence given in (1.8) will be represented in UNL^{KB} as illustrated in (1.9).

A boy can run. …(1.8)
icl(boy(icl>male person),human(icl>living thing))=1
agt(run(agt>person),human(icl>living thing))=1 …(1.9)

UNL^{KB} makes the use of three main categories of relations, namely, *'icl* (sub class of); *iof'* (element/instance of); and *'equ'* (equivalent to). With the use of these three relations, UNL^{KB} is arranged hierarchically. This hierarchical structure allows implementation of the principle of inheritance in the definition of concepts. All information assigned to a parent node can be subsumed as inherited by the children nodes. For instance, in defining *'cat(icl>feline)'*, there is no need to repeat all properties of felines, as they are defined under all mammals of the same species [10].

1.3.5 UNL Sentence

UNL sentence is the basic unit of representation inside the UNL framework. It consists of nodes (UWs) interlinked with binary semantic relations and modified by attributes. UNL

representation of a sentence is expressed in the form of a semantic net called UNL graph [7]. Consider the following English sentence in (1.10) for its UNL representation.

Example Sentence: He ate apples. ...(1.10)

The UNL sentence for sentence given in (1.10) is shown in (1.11).

{unl}

agt(eat:03.@past, 00:01.@3.@male)

obj(eat:03.@past, apple:02.@pl)

{/unl} …(1.11)

Equivalent Punjabi sentence: ਉਸ ਨੇ ਸੇਬ ਖਾਧੇ । …(1.12)

Transliterated Punjabi sentence: *us ne seb khadhe*

Here, *'agt'* and *'obj'* are the relation labels, *'eat:03'*, *'00:01'* and *'apple:03'* are the UWs, first UW in a relation is always head of whole expression, *'@3.@male'* is used to indicate third person male pronoun, *'@past'* is used to indicate past tense of the sentence and *'@pl'* is for plurality of object.

As indicated earlier, a UNL sentence can also be represented in the form of a graph called as UNL graph. The UNL graph of the example sentence given in (1.9) is depicted in Figure 1.2.

Figure 1.2: UNL graph for example sentence given in (1.9)

1.3.6 Compound UWs in UNL

Compound UWs are a set of binary relations that are grouped together to express a compound concept. Complex sentences are represented in UNL with the help of compound UWs. Compound UWs denote compound concepts that are to be interpreted as a whole so that one can talk about their parts all at the same time. A compound UW is expressed by a scope in UNL expressions. A Compound UW is defined by placing a compound UW-ID immediately after the UNL relation label in all of the binary relations

that are to be grouped together [7]. Compound UW-ID starts with ':' followed by two numbers (each between 0 and 9), *e.g.*, ':01'. The concept of compound universal word is illustrated below with the help of one example sentence is given in (1.13).

Example Sentence: Ram, who lives in Delhi, eats rice. …(1.13)

One can see that this is a clausal sentence involving one main clause '[Subject] eats rice'. Here, Subject is a complete sentence in itself represented by 'Ram, who lives in Delhi' acting as sub-clause of the main sentence. These, sub-clauses are represented by the compound UW as shown in following UNL expression (1.14).

{unl} agt(eat:03.@present, :01)
obj(eat:03.@present, rice:02)
agt:01(live:04.present, Ram:05)
plc:01(live:04.present,Delhi:06) {/unl} …(1.14)

Here, ':01' indicates the compound UW. Figure 1.3, indicates the UNL graph of (1.14) UNL expression.

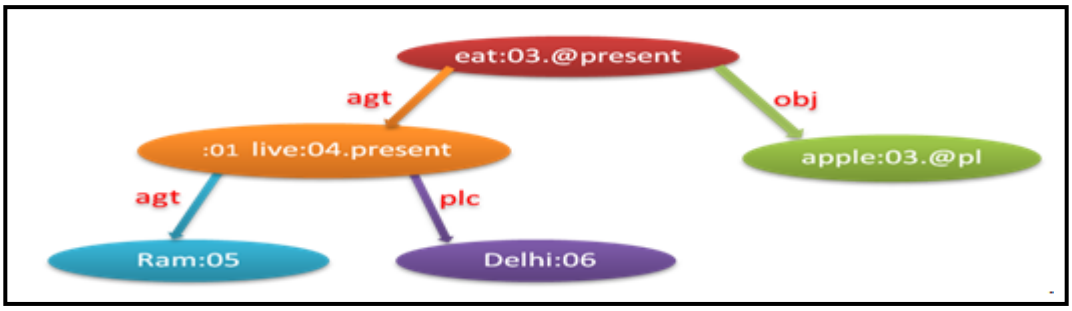


Figure 1.3: UNL graph for example sentence given in (1.14)

1.4 UNL Tools

The UNLdev is the wrapper application for the development of various UNL tools and applications to achieve the aims of the Universal Networking Language. These tools implement different tasks; they perform data transformations, analysis, search and generation. Secondary tools have been developed to aid the basic ones in the completion of their tasks. These secondary tools help in building the resources (such as dictionaries and grammars) that are later used by the basic tools and engines. There are tools for professional (linguists, computational linguists) and non-professional users [11].

Most of these tools are shareware, in the sense that the source code is not provided and users are supposed to sign the UNL Development Set agreement in order to have access to the software. UNLdev includes three UNLization softwares; IAN, the UNL Editor and SEAN take a natural language input and deliver output in UNL. On the other hand, the UNLdev also contains one NLization software, EUGENE, its input is in UNL while the output is in natural language [3].

1.4.1 IAN (Interactive ANalyzer)

It is an open-source interactive UNLizer, developed by UNDL Foundation. 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 [11]. IAN performs the three following movements over the input file.

- **Segmentation**, *i.e.*, the division of the input document into a series of processing units (sentences), which are processed one at a time.
- **Tokenization**, *i.e.*, the identification of the tokens (lexical items) of each sentence of the input document.
- **Transformation**, *i.e.*, the application of the transformation rules of the grammar over each tokenized sentence in order to represent it as a UNL graph.

1.4.2 EUGENE (dEep-to-sUrface GENErator)

It is an open-source interactive NLizer, developed by UNDL Foundation. 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 [11]. Similarly, EUGENE performs the three following movements over the input file.

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

Chapter Summary

Introduction to the UNL system has been given in this chapter, *i.e.*, UNL is an artificial language that can be used as a pivot language in interlingual machine translation. It was created at the Institute of Advanced Studies of the United Nations University, in Tokyo, and it has been developed at the UNDL Foundation, in Geneva, Switzerland. In 2005, a contract was signed between the UNO and the UNDL Foundation, authorizing the latter to keep on developing and promoting UNL under license. The UNL workplace UNLweb is described, which has been created to bring together the community of researchers and developers of UNL as well as the general people interested in UNL and Natural Language Processing (NLP) in general in order to share their expertise, documentation and resources. Later in this chapter, basic terms like, UWs, attributes, relations, UNL^{KB}, and UNL-sentence, compound UNL sentence are explained for using UNL as knowledge representation. Tools in the UNL system are also explained briefly.

Chapter 2

Literature Reviews

In the 21st century, emerging digital communication technologies and the World Wide Web facilitated an exponential growth in the volume of information in different languages and blurred geographic boundaries, making once hard-to-find data easily accessible online. Although English was the bridge language of the Web in its early days, due to rapid information and communication technologies (ICT) growth today less than one-third of Internet users are native English-speakers [12]. There are more than 6,700 languages in the world today [13]. The growing number of users from non-Anglophone countries creates unappeasable demand for cross-cultural interaction and understanding of online content in the era of globalization that speeds up every aspect of modern social interactions, including consumption of information and communication. This social necessity has spurred demand for instant machine translation services. Translation is actually the communication of the meaning of a source-language text by means of an equivalent target-language text. Machine Translation is sub-field of computational linguistics that investigates the use of software to translate text or speech from one natural language to another. It performs simple substitution of words in one natural language for words in another, but that alone usually cannot produce a good translation of a text, because recognition of whole phrases and their closest counterparts in the target language is needed [14].

2.1 Approaches of Machine Translation

The Task of Machine translation can be performed by various techniques. In this section, all these approaches are discussed briefly.

2.1.1 Rule-Based Approach

It is a general term that denotes machine translation systems based on linguistic information about source and target languages basically retrieved from bilingual dictionaries and grammars covering the main semantic, morphological, and syntactic regularities of each language respectively. Having input sentences (in some

source language), an RBMT system generates them to output sentences (in some target language) on the basis of morphological, syntactic, and semantic analysis of both the source and the target languages involved in a concrete translation task [15] . It is further divided into following three categories.

- **Transfer-Based Machine Translation:** In a rule-based machine translation system the original text is first analyzed morphologically and syntactically in order to obtain a syntactic representation. This representation can then be refined to a more abstract level putting emphasis on the parts relevant for translation and ignoring other types of information. The transfer process then converts this final representation (still in the original language) to a representation of the same level of abstraction in the target language. These two representations are referred to as "intermediate" representations. From the target language representation, the stages are then applied in reverse [16].

- **Interlingua-Based Machine Translation:** In this approach, the source language, *i.e.*, the text to be translated is transformed into an Interlingua, *i.e.*, an abstract language-independent representation. The target language is then generated from the interlingua. The Figure 2.1 depicts the model of interlingua based MT system.

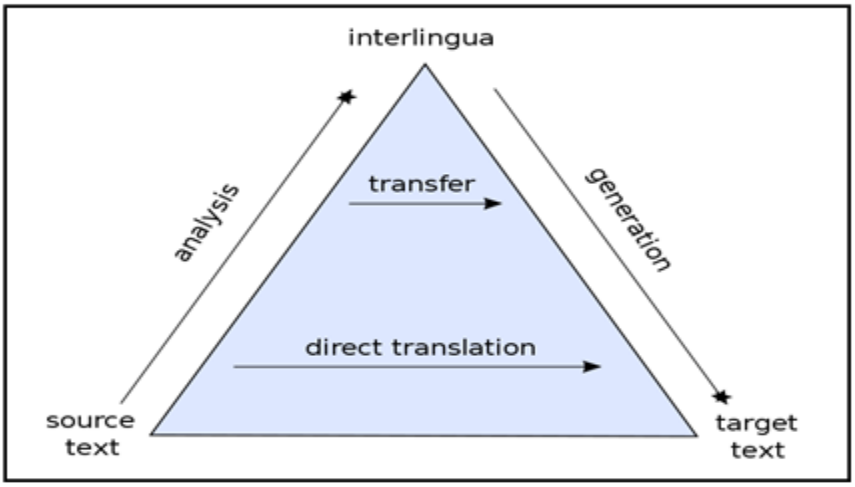


Figure 2.1 Interlingua based MT system [17]

- **Dictionary-Based Machine Translation:** This approach uses a method based on dictionary entries, which means that the words will be translated as a dictionary does – word by word, usually without much correlation of meaning between them.

Dictionary lookups may be done with or without morphological analysis or lemmatization. While this approach to machine translation is probably the least sophisticated, dictionary-based machine translation ideally suits the translation of long lists of phrases on the sub sentential (*i.e.*, not a full sentence) level [18].

2.1.2 Statistical–Based Machine Translation

Statistical machine translation (SMT) is a machine translation paradigm where translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpus (a large and structured set of texts). It translates a document according to the probability distribution $p(e|f)$ that a string e in the target language (for example, English) is the translation of a string f in the source language [14].

2.1.3 Example-Based Machine Translation

Example-based machine translation is the idea of translation by analogy. When applied to the process of human translation, the idea that translation takes place by analogy is a rejection of the idea that people translate sentences by doing deep linguistic analysis. Instead it is founded on the belief that people translate firstly by decomposing a sentence into certain phrases, then by translating these phrases, and finally by properly composing these fragments into one long sentence. Phrasal translations are translated by analogy to previous translations [19]. The principle of translation by analogy is encoded to example-based machine translation through the example translations in Table 2.1 that are used to train such a system.

Table 2.1: Example of bilingual English-Punjabi corpus [7]

English corpus	**Punjabi corpus**
How much is that **red umbrella**?	ਲਾਲ ਛਤਰੀ ਕਿੰਨੇ ਦੀ ਹੈ? *lal chtari kinne di hai?*
How much is that **book**?	ਕਿਤਾਬ ਕਿੰਨੇ ਦੀ ਹੈ ? *lāl kitab kinne di hai?*

Example-based machine translation system would learn three units of translation.

- How much is that X? **corresponds to** X ਕਿੰਨੇ ਦੀ ਹੈ? *X kinne di hai?*

- red umbrella **corresponds to** ਲਾਲ ਛਤਰੀ *lal chtari*.
- book **corresponds to** ਕਿਤਾਬ *kitab.*

2.1.4 Hybrid Machine Translation

Hybrid machine translation (HMT) leverages the strengths of statistical and rule-based translation methodologies. The approaches differ in a number of ways.

- **Rules post-processed by statistics:** Translations are performed using a rules based engine. Statistics are then used in an attempt to adjust/correct the output from the rules engine [20].

- **Statistics guided by rules:** Rules are used to pre-process data in an attempt to better guide the statistical engine. Rules are also used to post-process the statistical output to perform functions such as normalization. This approach has a lot more power, flexibility and control when translating [20].

2.2 Machine Translation for Indian Languages

As per 2001 census, Indians use 122 languages, including 22 as official languages. The top six languages such as Hindi, Bengali, Telugu, Marathi, Tamil, and Urdu are spoken by approximately 850 million people worldwide [21]. Only about 10% of Indian population speaks English. In the age of Internet, most of information is generated in English. Automatic translation of information available in English has immerged as an important topic of research during recent times. Department of Information Technology, MoCIT, Government of India, has initiated 'Technology Development for Indian Languages' (TDIL) project in the year 1990. The main objective of this project is to support and fund research and development efforts in the area of information processing in Indian languages including machine translation [22]. History of machine translation in India has witnessed a number of research projects covering a good number of Indian languages. Prominent among these are the projects at IIT Kanpur, IIT Bombay, IIIT Hyderabad, University of Hyderabad, National Center for Software Technology (NCST) Mumbai (now, Center for Development of Advanced Computing (CDAC) Mumbai), CDAC Pune and Jadavpur University, Kolkata [7].

2.3 Evolution of UNL-Based Machine Translation

The UNL Program started in 1996, as an initiative of the Institute of Advanced Studies of the United Nations University in Tokyo, Japan. 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, has an independent identity from the United Nations University, although it has special links with the UN. It inherited from the UNU/IAS the mandate of implementing the UNL Program so that it can fulfill its mission. Its headquarters are based in Geneva, Switzerland [23]. From the very beginning, a consortium of university departments from all regions of the world has been engaged in developing the UNL. That's the UNL Society, a global-scale network of research and development teams, involving about 200 specialists in computer science and linguistics, who are at work creating the linguistic resources and developing the web structure of the UNL System. The UNL Centre provides technological support and co-ordinates the implementation of the Program [23].

This program has already crossed important milestones. The overall architecture of the UNL System has been developed with a set of basic software and tools necessary for its functioning. These are being tested and improved. A vast amount of linguistic resources from the various native languages already under development, as well as from the UNL expression, has been accumulated in the last few years. Moreover, the technical infrastructure for expanding these resources is already in place, thus facilitating the participation of many more languages in the UNL system from now on. A growing number of scientific papers and academic dissertations on the UNL are being published every year [24].

The most visible accomplishment so far is the recognition by the Patent Co-operation Treaty (PCT) of the innovative character and industrial applicability of the UNL, which was obtained in May 2002 through the World Intellectual Property Organization (WIPO). Acquiring the patent for the UNL is a completely novel achievement within the United Nations [23].

In the UNL approach, from the very beginning there are two basic different movements: UNLization and NLization. UNLization is the process of representing/mapping/analyzing

the information conveyed by natural language utterances into UNL; NLization, conversely, is the process of realizing/manifesting/generating a natural language document out of a UNL graph. These processes are completely independent. For the time being, the NLization process has been already fully automatic, whereas the UNL-ization process is still mostly human, even though machine-aided [2]. As far as, NLization which is the prime concern of the thesis, it can be accomplished by NLizers, or formerly known as deconverters, *i.e.*, tools for producing natural language texts out of UNL documents [11]. There are two deconverters. One is DeCo, it is UNL centre's official deconverter engine, developed by UNL Tokyo. It is a shareware licensed tool and available in two versions, DeConverter version 2006 C deals with two-byte code languages like, Chinese, Korean, Thai language *etc*. and DeConverter version 2006 L which deals with ASCII codes, any one-byte code languages such as Arabic, Latin language, and Hindi [25]. Other NLization tool is EUGENE, which is an acronym for dEep-to-sUrface GENErator. EUGENE is online freeware licensed tool by UNDL Foundation [26]. It is standard NLizer has been used nowadays by many researchers of different natural languages.

Chapter 3
Problem Statement

All over the world, more than 91 million people speak Punjabi language, which makes it approximately the 12th most widely spoken language in the world [27]. Therefore, there is an immense need to develop a machine translation system for Punjabi language. In multi-lingual machine translation environment, where large number of languages are to be translated between one-another, it would require a separate system for every language, *i.e.*, $n*(n-1)$ language systems will be required for n number of languages. An interlingua approach based machine translation system is one solution for this problem, because here, only $2n$ language systems are required. UNL system designed and implemented by UNDL Foundation, in Geneva, Switzerland, is an internationally standardized interlingua that has the potential to bridge the gaps across all the languages of the world. UNL system constitutes two tools. One is IAN, for natural language to UNL conversion, and second is EUGENE, for UNL to natural language generation. For Punjabi language, no system has been yet developed for generating Punjabi language text from UNL using EUGENE. By developing such system, the task of translating any other language to Punjabi language becomes possible.

3.1 Objectives

The main objective of research work is to generate Punjabi from UNL using EUGENE. In order to perform this task, following objectives were proposed to be carried out.

i) To study the UNL Framework for NLization.

ii) To create UNL-NL Dictionary for Corpus500, UC-A1 and UC-A2 given by UNDL Foundation for testing the system for NLization using EUGENE.

iii) To create T-rules and Paradigms for NLization of adjectives, conjunctions, determiners and verbs.

iv) To achieve the F-measure score of 0.8 or above for ensuring correctness of Punjabi language sentences generated from UNL.

3.2 Methodology

To achieve the objectives discussed in section 3.1, a tool EUGENE developed by UNDL

Foundation has been used. A step-by-step methodology is followed for writing the T-rules and Dictionary for generating the Punjabi language sentences out of given UNL sentences.

i) To achieve the first objective, CUP500 certification through VALERIE, for proficiency in UNL, has been cleared.

ii) To make a UNL-NL dictionary for given UNL document, dictionary entries of all the UWs in the sentence have to be made first, *e.g.*, dictionary entry of UW "picture" as shown in (3.1) has been explained in below.

[ਤਸਵੀਰ]{}"picture"

(LEX=N,POS=NOU,NUM=SNG,GEN=FEM,PAR=M1) …(3.1)

Here, "picture" is the UW, 'ਤਸਵੀਰ' *tasveer* is its meaning in Punjabi. Its lexical category is noun, number and gender information is given to be singular and female respectively. *"PAR=M1"* specifies the paradigm number is *"M1"*, which is useful to generate other inflection of 'ਤਸਵੀਰ' *tasveer* to 'ਤਸਵੀਰਾਂ' *tasveeran* "picturs", which is explained in subsequent section.

iii) To write transformation rules based on the characteristics of the UWs included in the UNL sentence.

Example of *'aoj'* relation between a verb and noun is given in (3.2) for explaining the concept of resolving relation using T-rules. This relation is resolved into three nodes, *i.e.*, noun "%b", a blank space " " and verb "%a".

aoj(%a,V; %b,N):=(%b)(" ")(%a); …(3.2)

T-rules should be written such that they provide single translation for each UNL sentence. Rules should be as general as possible, so that, either lesser rules could be written or only slight changes could be made for processing the subsequent UNL sentences.

Example of inflection paradigm is shown in (3.3), for generating different inflections of words, whose paradigm information is mentioned to be *"PAR=M5"* in their dictionary

entries. This paradigm inflects Punjabi word like, 'ਗੱਡਾ' *gadda* "cart", 'ਸੋਹਣਾ' *sohna* "handsome" to 'ਗੱਡੀ' *gaddi* "car", 'ਗੱਡੇ' *gadde* "carts", 'ਸੋਹਣੀ' *sohni* "handsome" (for female), 'ਸੋਹਣੇ' *sohne* "handsome" (for plural) respectively, based on the attributes associated with the UWs.

(%x,M5):=(%x,-M5,+FLX(SNG&FEM:=1>"ੀ"; PLR&MCL:=1> "ੇ";
PLR&FEM:=1>"ੀਆਂ";)) ; …(3.3)

If UW has *"SNG"* and *"FEM"* attributes simultaneously then one character is removed from right end and "ੀ" is appended to it, for *"PLR"* and *"MCL"* attributes one character is removed from right end and "ੇ" is appended to it, similarly, for *'PLR'* and *'FEM'* attributes one character is removed from right end and "ੀਆਂ" is appended to it.

iv) After writing Dictionary entries and T-rules for the UNL sentences, the sentences are processed in a single go, and then these output natural language sentences are compared with the expected natural language sentences for obtaining the F-measure.

Chapter 4

NLization Framework with EUGENE

The EUGENE is a language independent generator, which provides a NLization framework for syntactic and morphological generation synchronously. It can convert UNL expressions into a variety of natural languages, by using respective word dictionaries and sets of grammar rules.

NLization plays a core role in the UNL system. It is very significant that NLizer is capable of expressing UNL information with very high accuracy. It follows that information, once composed in UNL, can be understood in any language as far as there be a NLizer of the language.

4.1 Components of NLizer in UNL Framework

NLization of UNL-Document to NL-Document is performed using various components of NLizer, as shown in Figure 4.1. Function of different components is discussed in this section.

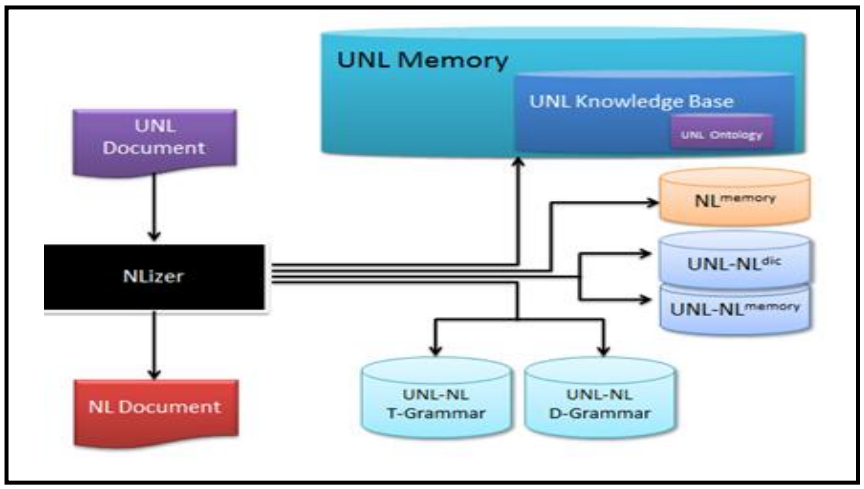


Figure 4.1: NLization of UNL-Document to NL-Document [11]

4.1.1 UNL Document

UNL documents are documents written in UNL. They are plain text files that include UNL sentences and some special tags which are discussed in section 4.2.1. These are the output of the UNLization process and the input of the NLization process.

4.1.2 NL Document

NL document contains the sentences of the natural language, which are generated as output of NLization process.

4.1.3 UNL Memory

The UNL Memory, also known as UNL Example Base, is a network structure where UW's are interconnected through any semantic relation of UNL. Differently from the UNL Ontology, which deals only with monotonic relations ("icl" and "iof"), and the UNL Knowledge Base, which deals only with relations used to define a concept, the UNL Memory contains any relation that is likely to link two concepts, including those that are not necessary but only frequent, according to their use in a given corpus [28]. In that sense, the UNL Memory comprises and extends the UNL^{KB} and UNL Ontology, formerly known as the UW System, a tree-like structure where UWs are interconnected through hierarchical relations, *i.e.*, icl (is-a-kind-of) and iof (is-an-instance-of). Differently from the UNL^{KB}, which comprises any relation necessary to define a given UW, the UNL Ontology contains only monotonic relations, *i.e.*, relations that preserve the features of their arguments, and that may be used for inheritance [29].

4.1.4 NL-Memory

The NL-Memory constitutes a list of syntactic (sub-categorization) frames between natural language words or terms that co-occur more often than would be expected by chance. These are used to represent collocations, *i.e.*, partly or fully fixed expressions that become established through repeated context-dependent use [30]**.**

4.1.5 UNL-NL Dictionary

$UNL\text{-}NL^{dic}$ or Generation Dictionary is list of mappings between entries from the UNL Dictionary and entries from the NL Dictionary, with their corresponding features. In other words, the $UNL\text{-}NL^{dic}$ aims at mapping UW's into natural language entries [31].

4.1.6 UNL-NL Memory

The UNL-NL Memory is a set of mappings between a given natural language and UNL. It may be unidirectional (UNL-NL Memory or NL-UNL Memory) or bidirectional (UNL←→NL Memory). It is used to improve and normalize the results of the UNLization and the NLization, as it contain segments that have been previously UNLized or NLized [30].

4.1.7 UNL-NL T Grammar and D Grammar

UNL-NL T-Grammar, or Transformation Grammar, is a set of T-rules used to transform to UNL into natural language. Whereas, UNL-NL D-Grammar contains the disambiguation rules used for improving the results of the UNL-NL T-Grammar.

4.2 EUGENE tool for NLization

EUGENE (the dEep-to-sUrface natural language GENErator) is a tool designed and implemented by UNDL. It uses UNL resources (UNL-NL Dictionaries, UNL Transformation and Disambiguation Rules) to process UNL documents and to produce an equivalent text in a target natural language [26]. The fully automated web based EUGENE tool developed by UNDL provides following tabs to accomplish desired task of NLization:

4.2.1 UNL Input

In this tab, list of already existing UNL document can be viewed, user can load a particular document, edit any UNL expression in the document, delete a document, add new UNL document or share among different user *etc.* using various options as shown in Figure 4.2.

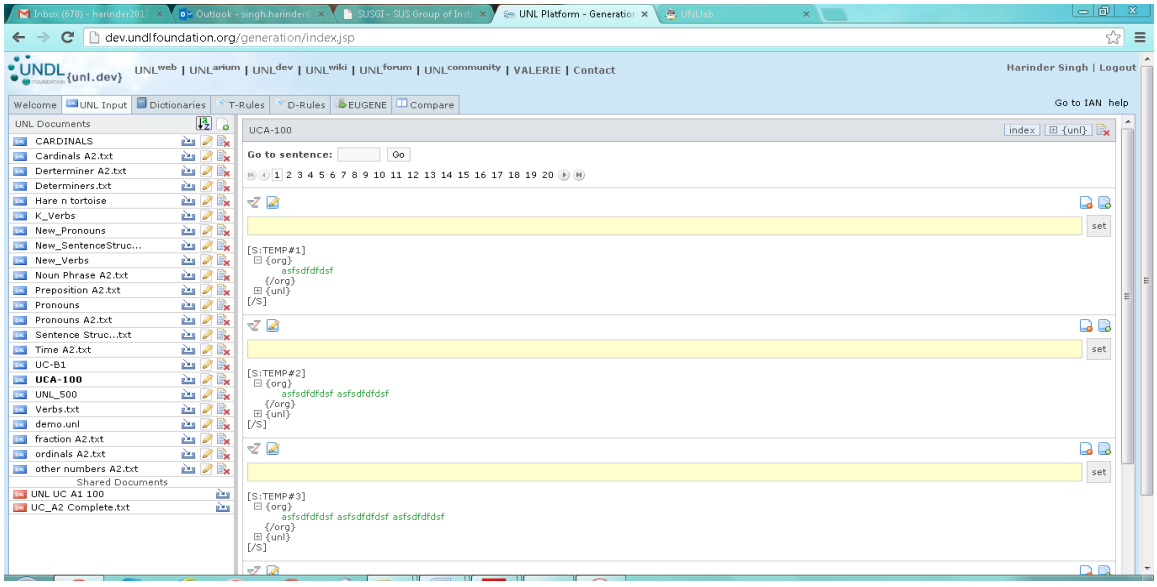


Figure 4.2: Snapshot to indicate UNL Input EUGENE

A UNL document is enclosed with tags "[D:<id>]" and "[/D]". Within these tags, each paragraph is enclosed with a pair of tags "[P:<id>]" and "[/P]", and each sentence is enclosed with a pair of tags "[S:<id>]" and "[/S]". Inside a sentence, the text of original

sentence is enclosed with "{org:<lang>}" and "{/org}", its UNL expression is enclosed with "{unl:<id>}" and "{/unl}". Sentences of target languages can also be stored in the UNL document [7]. Each target sentence is enclosed with a pair of language tags "{<lang>}" and "{</lang>}" following the UNL expression of each sentence. All tags used with UNL expression and final sentence of target are listed in Table 4.1.

Table 4.1: Tags used in a UNL document [7]

Tags	**Description**
[D:<id>]	To represent beginning of a document
[/D]	To represent the end of a document
[P:<id>]	To represent the beginning of a paragraph
[/P]	To represent the end of a paragraph
[S:<id>]	To represent the beginning of a sentence
[/S]	To represent the end of a sentence
{org:<lang>=<code>}	To represent the beginning of an original/source sentence
{/org}	To represent the end of an original sentence
{unl:<id>}	To represent the beginning of the UNL expressions of a sentence
{/unl}	To represent the end of the UNL expressions of a sentence
{<lang>}	To represent the beginning of a target sentence of the language indicated by <lang>
{/<lang>)	To represent the end of a target sentence of the language indicated by <lang>

Here, ':<id>' (generally optional), which is normally represented by an integer, may be any sequence of characters used to identify the document, the sentence, the paragraph or the UNL expression. '<lang>' , which is optional in case of {org} tag and it corresponds to the language code in ISO639-2 or ISO639-3. '=<code>' (generally optional) corresponds to the character encoding [7].

4.2.2 Dictionary

The UNL-NL dictionary contains the information of words that correspond to UWs included in the input of UNL expressions and grammatical attributes (features) that describe the behaviors of the words [32]. Figure 4.3 shows the Dictionary tab of EUGENE for manipulating dictionary files and their contents.

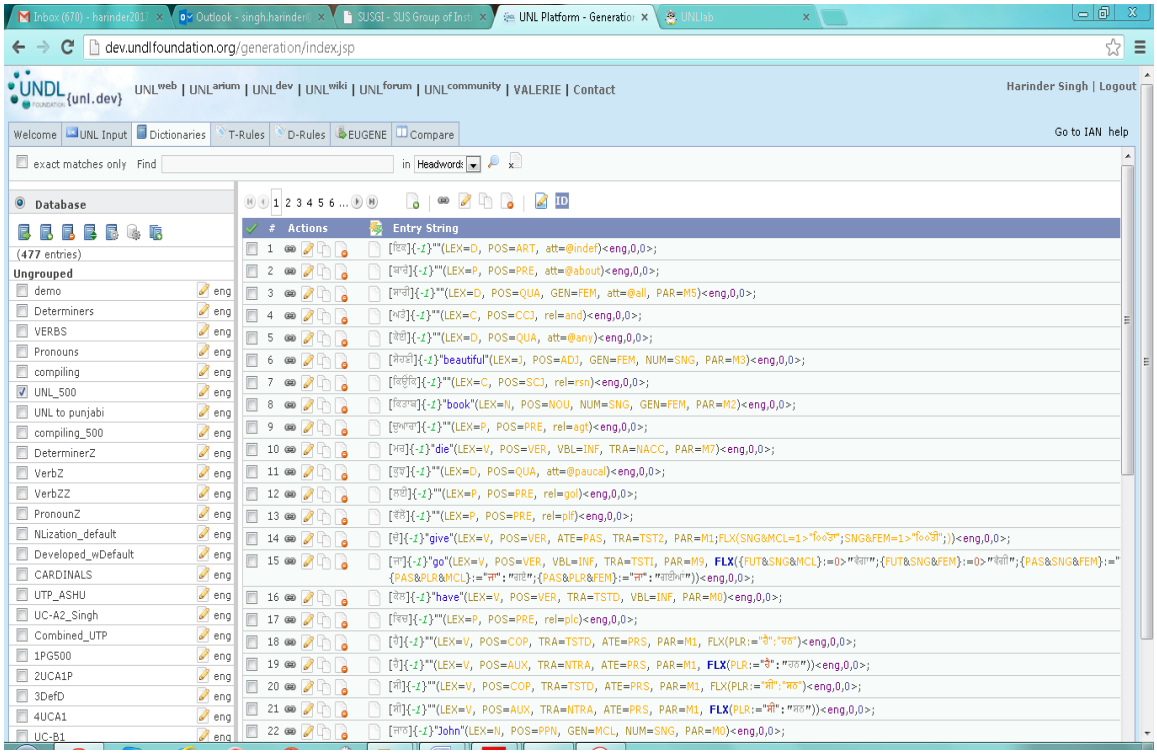


Figure 4.3: Snapshot showing Dictionary tab in EUGENE tool

Authorized user has to provide the UNL-NL dictionary. 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. In any case, the dictionary must be provided according to the UNL Dictionary specifications. Once, a user created/uploaded a dictionary, he/she have to select it by clicking the corresponding check box and load it by pressing the load button at the top menu. One may have several different dictionaries, and may load many of them to process the same corpus, but be sure that they are loaded in the correct order (because the order of the entries in the dictionary does matter for tokenization) [31]. User may reorder the dictionaries through the option "reorder dictionaries" at the top menu.

4.2.3 T-Rules

T-rules describe how to construct a sentence using the information from the input of UNL expressions and defined in a word dictionary. The EUGENE converts UNL expressions into sentences of a target language following the descriptions of T-rules. Figure 4.4 shows all options available in T-rule tab of EUGENE.

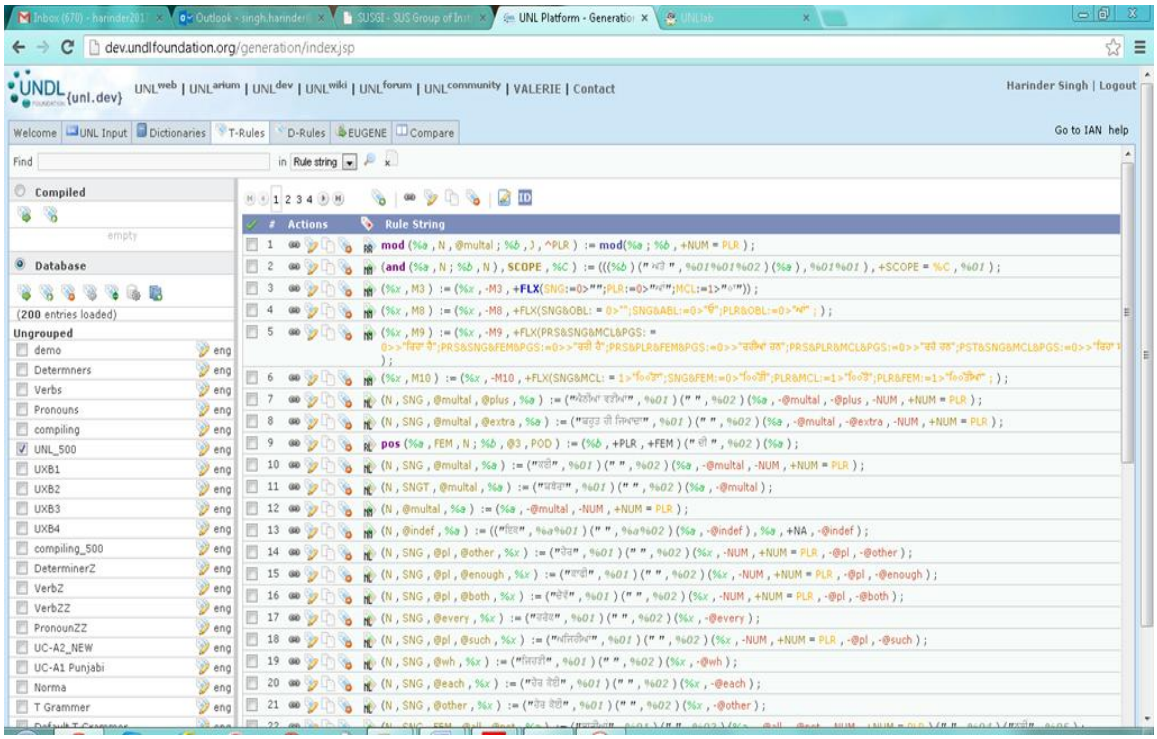


Figure 4.4: Snapshot showing T-Rules tab of EUGENE tool

Authorized user has to provide the UNL-NL T-grammar for processing a particular UNL document and generating natural language sentences out of it. A user can load a particular rule set, either create a new set or upload an existing one, delete a rule set, edit a rule or create new rule, enable or disable a rule and also share file with different authentications among different users.

4.2.4 EUGENE Console

Here, the user gets the final results of generation of natural language sentence from UNL input. The EUGENE console brings the list of sentences appearing in the UNL input. The processing of the documents can be done sentence by sentence or for the whole document at once. EUGENE is language-independent, it simply uses the target language grammar rules and UNL-NL dictionary in order to decode the incoming UNL document and generate it in natural language format [3] [26].

Providing relevant T-rule and dictionary EUGENE console then produces the natural language text from UNL as shown in Figure 4.5.

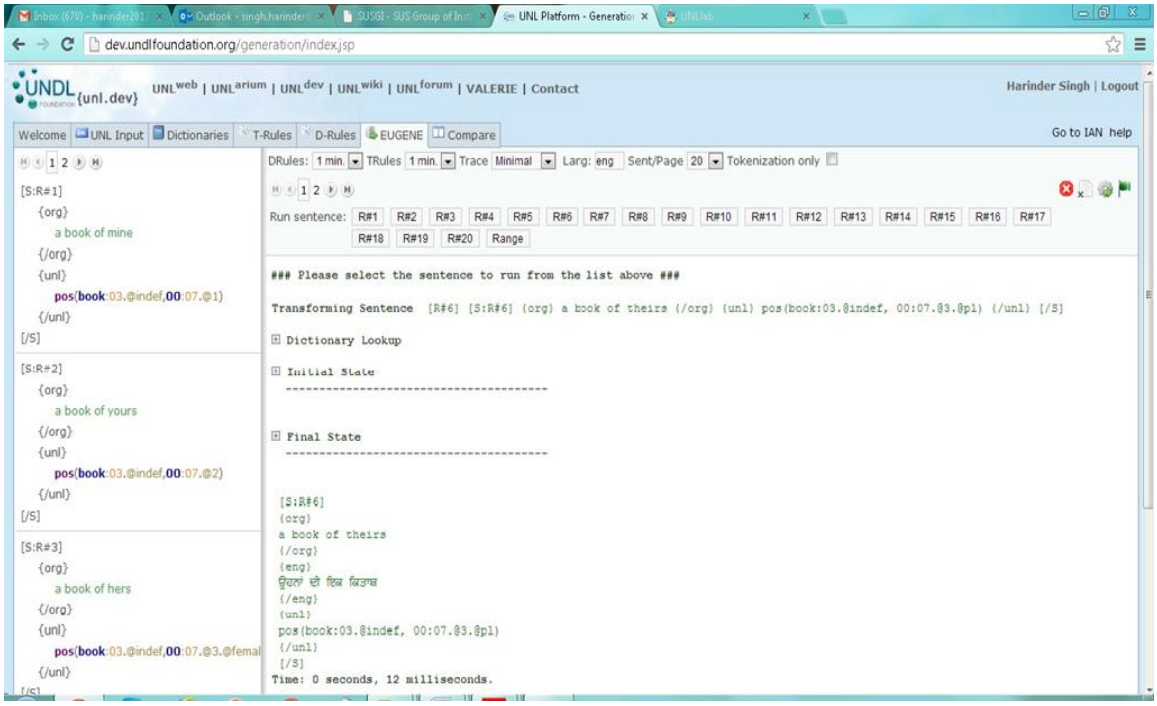


Figure 4.5: Snapshot of processing a UNL sentence one at a time by EUGENE

4.2.5 D-Rules

These rules (disambiguation rules) are used to prevent wrong lexical choices, to provoke best matches and to check the consistency of graphs, trees and lists. Differently from T-rules, they do not provoke transformations, but induce or block them, by assigning priorities to natural language phenomena. The set of D-rules form the disambiguation grammar [26].

4.3 Processing a UNL Document

In this section, the navigations are given for using EUGENE tool, starting from first step of loading or adding UNL input file, to finally getting the natural language output.

- Firstly the UNL document containing the UNL sentences to be NLized is loaded (if already existed in the list) or new document file is added using first tab. Figure 4.6 including navigation box showing where to click for adding a new file.

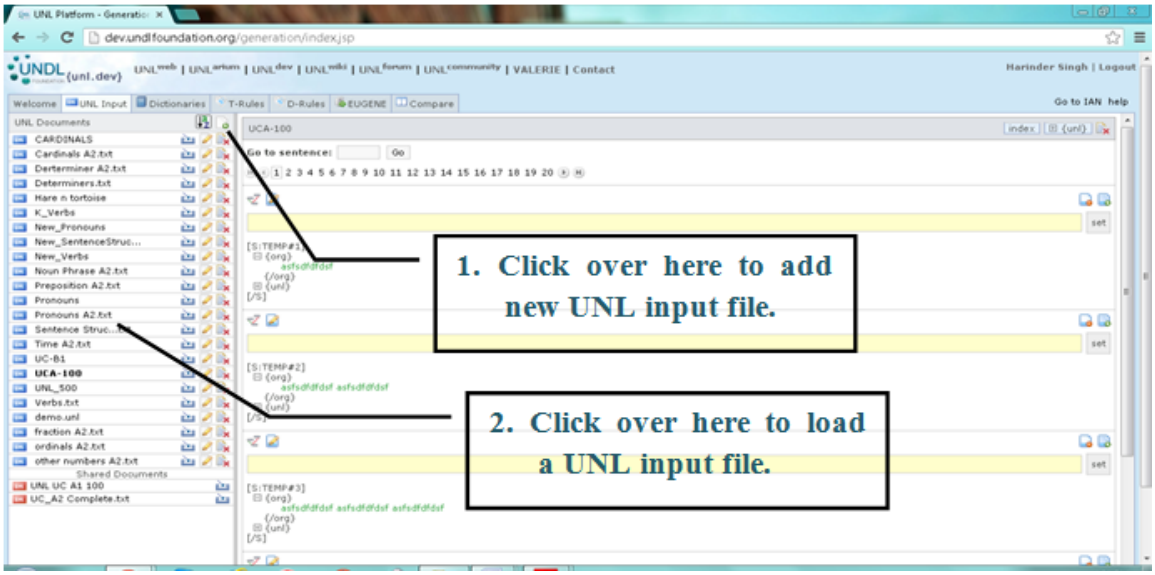


Figure 4.6: Navigation for adding new UNL document

Figure 4.7 shows the prompt window appearing after clicking the button shown in Figure 4.6. Using this window firstly suitable name is given to the file and file containing the UNL sentences in text format is browsed, selected and uploaded from the system (personal computer or shared drive).

Figure 4.7: Choosing and uploading the UNL file

- After uploading and loading UNL document next step is to make the dictionary entries, starting from the UWs appeared in the first UNL sentence. User can either add a new dictionary file in the list or add entry to the already loaded file. Figure 4.8 shows how to add new dictionary file. After clicking our the navigated icon same sort of prompt window as shown in Figure 4.7 will appear for choosing and uploading the new dictionary file.

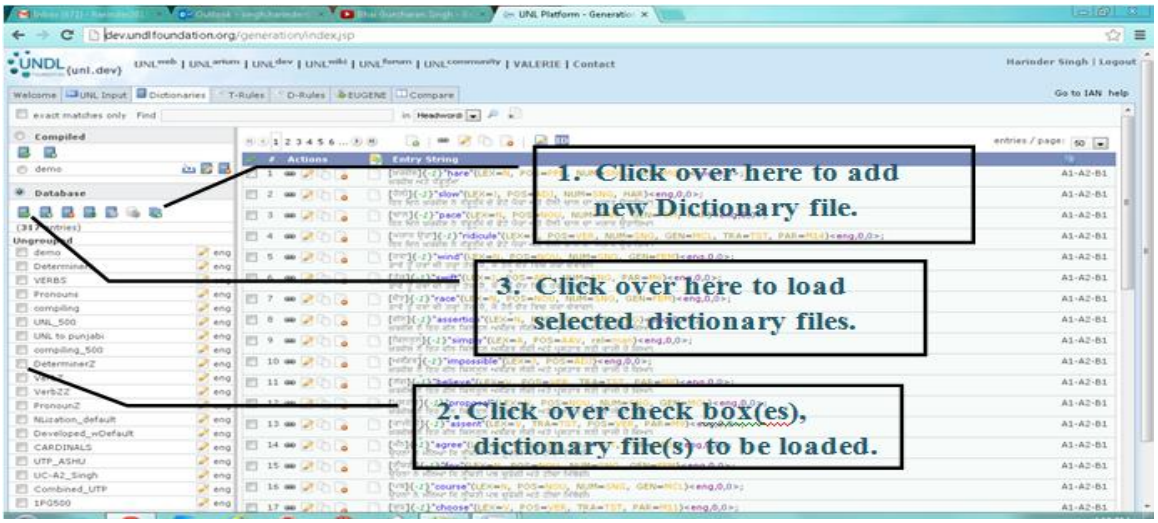


Figure 4.8: Navigation for adding/loading dictionary files

User may add new dictionary entry in the already loaded file. Figure 4.9 navigates to the icon on interface for making new entry in existing file.

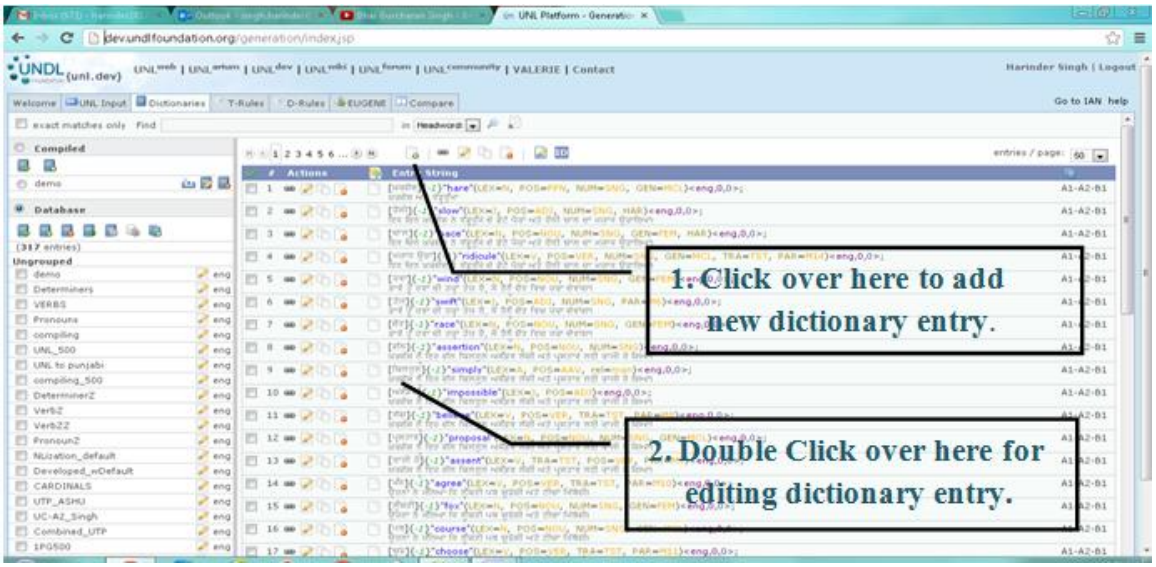


Figure 4.9: Navigation for making new entry in the loaded file

After clicking over the icon navigated in Figure 4.9, a prompt window shown in Figure 4.10 appears. User has to firstly write the dictionary entry, *i.e.*, UW 'book' is added and after that order of entry is to be specified in Figure 4.10.

Also the user can double click over any dictionary entry in already loaded dictionary file as navigated in Figure 4.9 for editing the dictionary entry or changing its order.

After double clicking the over the dictionary entry, same prompt window as shown in Figure 4.10 appears where user can make the desired changes.

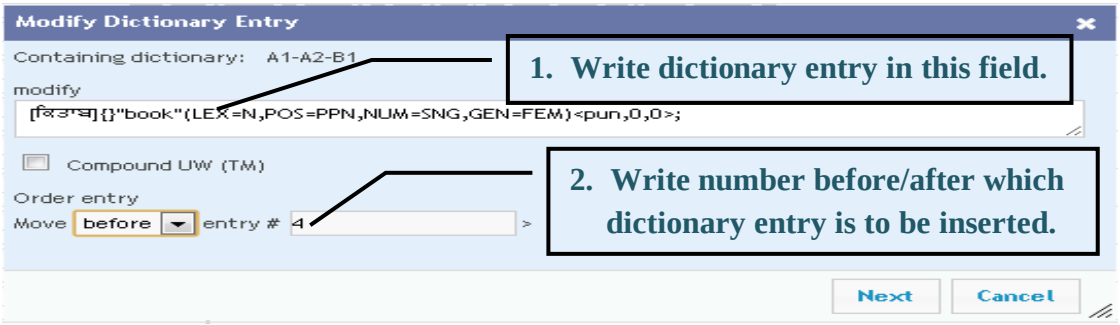


Figure 4.10: Making new/editing already existing entry in the loaded file

- After making dictionary entries of UWs relevant to UNL sentences the next step is to write the T-rules for processing the UNL sentence. Figure 4.11 shows where to click for adding new rule set and adding/editing the T-rule in the already loaded rule set.

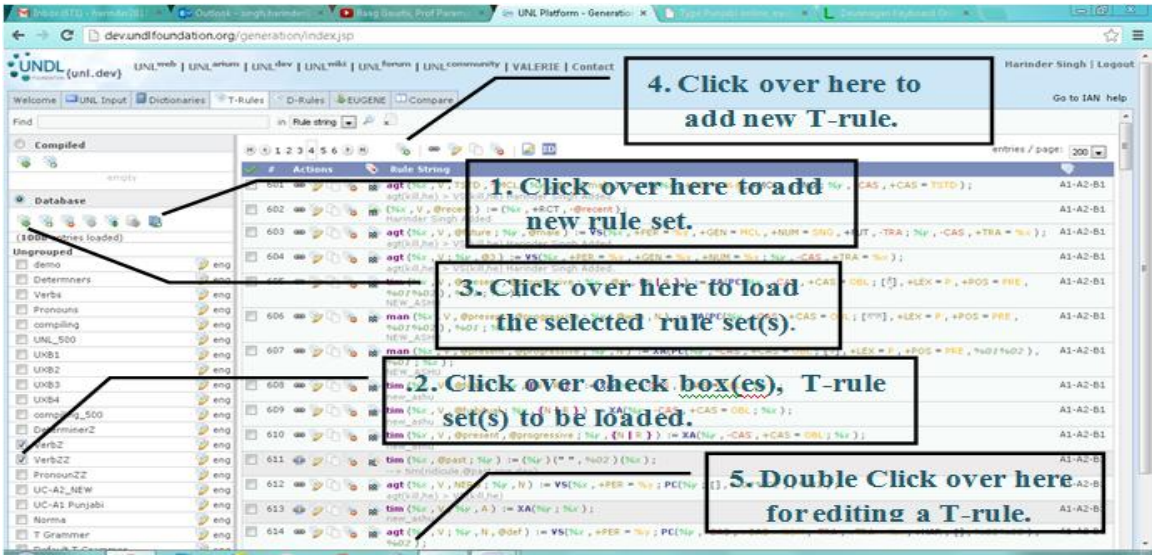


Figure 4.11: Navigation for adding/loading a rule set or add/edit a rule

After clicking over the navigated icon for adding new rule set, same sort of prompt window as shown in Figure 4.7 for choosing and uploading the new rule set will appear. Using that prompt window user may add new rule set. If user chooses to edit/add a T-rule in already loaded rule set then a prompt window as shown in Figure 4.12 will appears.

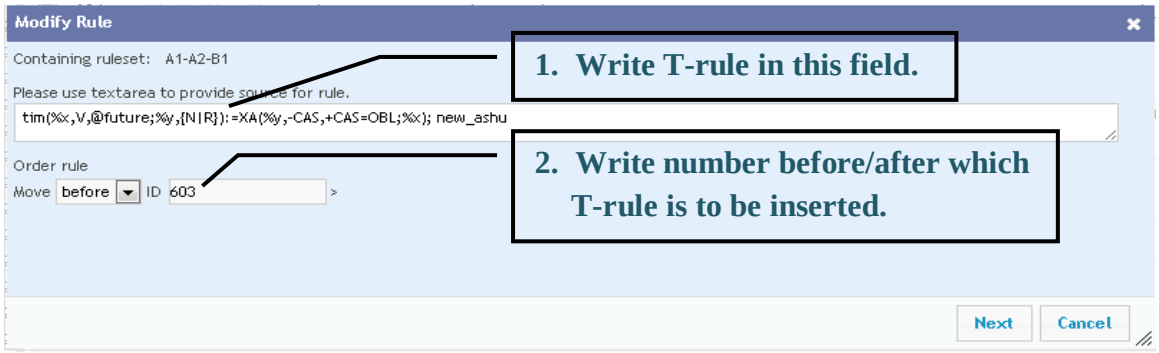
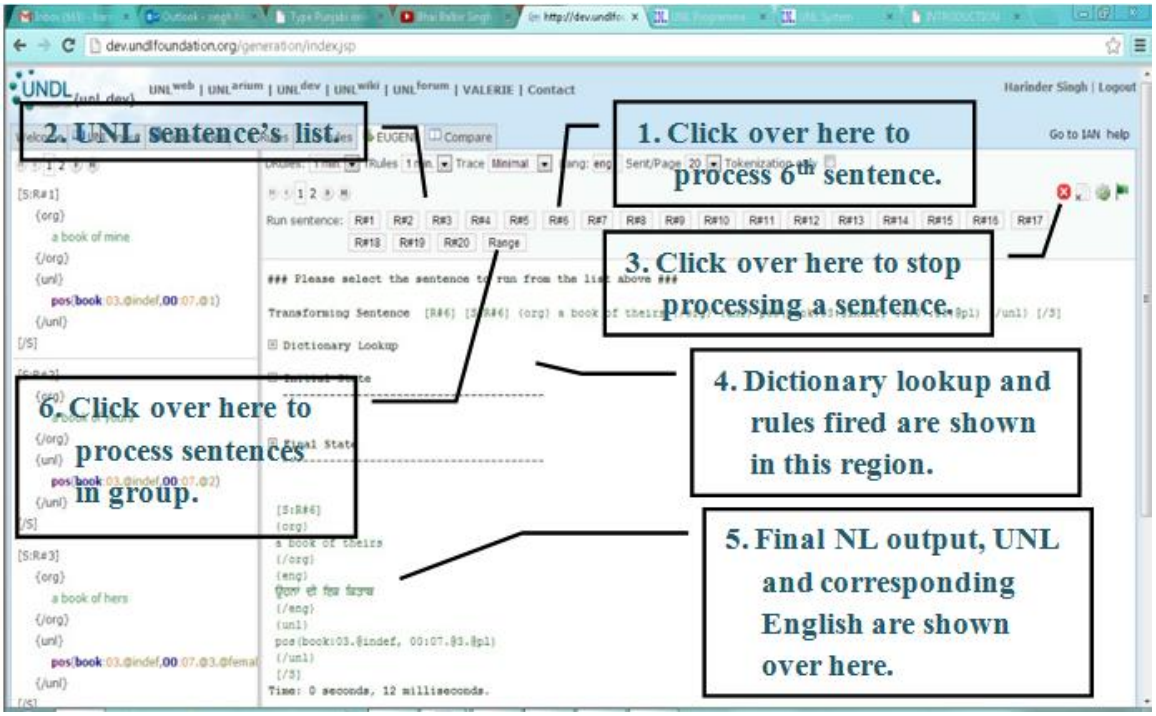


Figure 4.12: Making new/editing already existing T-rule in the loaded rule set

- UNL sentences in the selected/loaded UNL input file appear as a list of buttons on the EUGENE console tab as shown in Figure 4.13. By clicking over any button from the list, the dictionary lookup, relevant rules fired and natural language output generated can be seen in respective fields. In Figure 4.13 sixth button for processing sixth UNL sentence is clicked and output is generated for it.



4.13: Navigation for adding/loading a rule set or add/edit a rule

Chapter Summary

Various components involved in NLization framework have been described in this chapter, *i.e.*, every component like, input UNL documents comprising of UNL

expressions, UNL-NL dictionary, UNL-NL grammar, output NL document comprising of natural language sentences *etc.* have been elaborated. Later, the functionalities provided in EUGENE including creating/loading the input UNL file(s), dictionary file(s), T-rule file(s), editing and reordering of dictionary words in a loaded dictionary(s) and in T-rules in loaded rule sets *etc.* and finally getting the NL output sentence all have been explained with the appropriate navigations on the screenshots of GUI of EUGENE.

Chapter 5

Implementation of NLization of Conjunctions, Determiners,

The NLization framework has been implemented with EUGENE for generating natural language output for some experimental corpuses. UNDL foundation has prepared three corpuses, which are Corpus500, UC-A1 and UC-A2, comprising of 500, 100 and 300 UNL sentences respectively, for testing the framework for generation of natural language sentences containing different parts of speech. These corpuses are used to prepare the initial versions of the grammar for sentence-based NLization. The parts of speech contained in these corpuses are listed below.

1. Adjectives
2. Conjunctions
3. Determiners
4. Noun Phrase and sentence structures
5. Numbers and ordinals
6. Preposition
7. Pronouns
8. Time
9. Temporary entries
10. Verbs

The scope of research work encompasses NLization of adjectives, determiners, conjunctions and verbs using EUGENE. In the subsequent section, implementation of NLization framework using EUGENE is explained with three examples (easy, medium, and difficult) for each part of speech.

5.1 NLization of Adjectives

In Punjabi language, adjectives usually precede the nouns but follow the pronouns. For example, ‘ਸੋਹਣਾ’ *sohan* “handsome”, ‘ਕਾਲਾ’ *kala* “black” are functioning as adjectives and they precede nouns in ‘ਸੋਹਣਾ ਮੁੰਡਾ’ *sohna munda* “handsome boy”, ‘ਕਾਲਾ ਘੋੜਾ’ *kala ghora* “black horse”, respectively. The examples of adjectives following pronouns are, ‘ਮੈਂ ਸੋਹਣਾ ਹਾਂ’ *main sohna han* “I am handsome”, ‘ਤੂੰ ਸੋਹਣਾ ਹੈਂ’ *tun sohna hain* “you are handsome”. Punjabi adjectives can be classified into two categories, inflected and uninflected. For example, ‘ਸੋਹਣਾ’ *sohna* “handsome” inflections are ਸੋਹਣਾ *sohna* “handsome” (direct

singular), ਸੋਹਣੇ *sohne* "handsome" (oblique singular and direct plural), and 'ਸੋਹਣੀਆਂ *sohnian* "handsome" (oblique plural).The uninflected adjectives are 'ਲਾਲ' "red", 'ਮਿਹਨਤੀ' *mehanti* "hardworking" , ਮਸ਼ਹੂਰ *mashoor* "famous" *etc.*[7].

Generation process for phrases containing adjectives form UNL sentences is usually dependent on the following things, *i.e.*, relative placements of UWs, resolution of morphology according to Punjabi grammar are done on the basis of these things.

- Adjective(s)
- Subject or Object
- Gender information of subject
- Number information of subject

The process of NLization of input UNL sentence containing adjectives to natural language sentence is illustrated with an example sentence given in (5.2).

Example 1: English sentence is: the beautiful car. …(5.1)

{unl}

mod(car:03.@def, beautiful:01)

{/unl} …(5.2)

Expected Punjabi sentence: ਸੋਹਣੀ ਗੱਡੀ …(5.3)
Transliterated Punjabi sentence: *sohni gaddi*

All nodes (or UWs) identified in UNL sentence given in (5.2), are mapped with their corresponding dictionary entries, and are listed below.

beautiful:01 -

[ਸੋਹਣੀ]{}"beautiful"(LEX=J,POS=ADJ,GEN=FEM,NUM=SNG,PAR=M3);

car:03 - [ਗੱਡੀ]{}"car" (LEX=N,POS=NOU,GEN=FEM,PAR=M3);

Here, *LEX* represents lexical category, *J* represents adjective, *POS* represents part-of-speech, *"ADJ"* represents adjective, *"GEN"* represents gender which can be either *MCL* for male and *"FEM"* for female, *NUM* represents number whose value could be either *"SNG"* for singular or *"PLR"* for plural, *PAR* represents the paradigm number which is

"M3" in above cases, it specifies which inflection paradigm has to be fired for inflecting the root words. *N* represents noun. The process of NLization of example UNL sentence given in (5.2) has been illustrated in Table 5.1. Here, transliteration of each Punjabi word is also shown.

Table 5.1: NLization process of first example of adjectives

Rule Fired	**Description**	**Action Taken**
(%x,M3):=(%x,-M3, +FLX(SNG:=0>"";PLR:=0>"ਆਂ"; MCL:=1>"ਾਂ"));	This paradigm M3 has been defined to attach corresponding postfix to verbs in Punjabi. Here, attribute *'FLX'* indicates that word has been processed for inflection.	**To:** [car:03] **Result:** ["ਗੱਡੀ":03] Nothing is appended to UW 'ਗੱਡੀ' *gaddi*, because this root word does not have any combination of attributes as are involved in this rule.
(%x,M3):=(%x,-M3, +FLX(SNG:=0>"";PLR:=0>"ਆਂ"; MCL:=1>"ਾਂ"));	This paradigm M3 has been defined to attach corresponding postfix to verbs in Punjabi. Here, attribute *'FLX'* indicate that word has been processed for inflection.	**To:** [beautiful:01] **Result:** ["ਸੋਹਣੀ":01] Nothing is appended to UW 'ਸੋਹਣੀ' *sohni*, because this root word does not have any combination of attributes as are involved in this rule.
mod(%a,N;%b,J):= ((" ")(%b)(" ")(%a),%a,+NA);	This T-rule has been defined to resolve the "*mod*" relation between noun and adjective.	**To:** mod(car:03, beautiful:01) **Result:** [sc:01(#L:01(-:02, beautiful:01); #L:01(beautiful:01, - :04); #L:01(-:04, car:03))] **Intermediate Result:** ਸੋਹਣੀ ਗੱਡੀ As a result, this relation is resolved by placing adjective before noun with space between them.
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm to inflect the root word "ਸੋਹਣੀ" *sohani* based on number and gender information associated with the word.	**To:** [:01] **Result:** [:01] Nothing is appended to UW 'ਸੋਹਣੀ' *sohni*, because this root word neither have *'PLR' nor 'MCL'* attribute.
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm to inflect the root word "ਗੱਡੀ" *gaddi* based on number and gender information associated with the word.	**To:** [car:03] **Result:** ["ਗੱਡੀ":03] **Final Output:** ਸੋਹਣੀ ਗੱਡੀ Nothing is appended to UW 'ਗੱਡੀ' *gaddi*, because this root word

		neither have *'PLR' nor 'MCL'* attribute.

Natural language output generated by EUGENE is given in (5.4), correspond to UNL sentence given in (5.2).

{org}

The beautiful car

{/org}

{pun}

ਸੋਹਣੀ ਗੱਡੀ {/pun}

{unl}[W]

mod(car:03.@def, beautiful:01)

[/W] {/unl} …(5.4)

NLization of adjectives has further been explained with the help of bit tougher example UNL sentence given in (5.6).

Example 2: English sentence is: the most expensive car. …(5.5)

{unl}

mod(car:07.@def, expensive:05.@most)

{/unl} …(5.6)

Expected Punjabi sentence: ਬਹੁਤ ਮਹਿੰਗੀ ਗੱਡੀ …(5.7)

Transliterated Punjabi sentence: *bhot mehangi gaddi*

All nodes (or UWs) identified in UNL sentence given in (5.6), are mapped with their corresponding dictionary entries, and are listed below.

expensive:05.@most -

[ਮਹਿੰਗੀ]{}"expensive"(LEX=J,POS=ADJ,NUM=SNG,PAR=M3);

car:07.@def - [ਗੱਡੀ]{}"car" (LEX=N,POS=NOU,GEN=FEM,PAR=M3)

The process of NLization of example UNL sentence given in (5.6) has been illustrated in Table 5.2. Transliteration of each Punjabi word is also shown in the table.

Table 5.2: NLization process of second example of adjectives

Rule Fired	Description	Action Taken
(%x,M3):=(%x,-M3, +FLX(SNG:=0>"";PLR:=0>"ਆਂ";MCL :=1>"ਾਂ"));	This paradigm M3 has been defined to attach corresponding postfix to verbs in Punjabi. Here, attribute *'FLX'* indicates that word has been processed for inflection.	**To:** [car:07.@def] **Result:** ["ਗੱਡੀ":07.@def] Nothing is appended to UW 'ਗੱਡੀ' *gaddi*, because this root word does not have any combination of attributes as are involved in this rule.
(%x,M3):=(%x,- M3,+FLX(SNG:=0>"";PLR:=0>"ਆਂ"; MCL:=1>"ਾਂ"));	This paradigm M3 has been defined to attach corresponding postfix to verbs in Punjabi.	**To:** [expensive:05.@most] **Result:** ["ਮਹਿੰਗੀ":05.@most] Nothing is appended to UW 'ਮਹਿੰਗੀ' *mehangi*, because this root word does not have any combination of attributes as are involved in this rule.
mod(%a,N;%b,J):=((" ")(%b)(" ")(%a),%a,+NA);	This T-rule has been defined to resolve the *'mod'* relation between noun and adjective.	**To:** mod(car:07.@def, expensive:05.@most) **Result:** [sc:01(#L:01(expensive:05.@most, -:01); #L:01(-:01, car:07.@def))] **Intermediate form:** ਮਹਿੰਗੀ ਗੱਡੀ *mehangi gaddi* As a result, *'mod'* relation is resolved by placing adjective before noun with space between them.
	This T-rule has been defined to attach special word "ਬਹੁਤ" *bhot* before a adjective word having attribute "*@most*".	**To:** [expensive:05.@most] **Result:** #L:01(ਬਹੁਤ:02, -:03); #L:01(-:03, expensive:05) **Intermediate form:** ਬਹੁਤ ਮਹਿੰਗੀ ਗੱਡੀ *bhot Mehangi gaddi* Attribute "*@most*" is removed from adjective and a special word 'ਬਹੁਤ' *bhot* is appended before 'ਮਹਿੰਗੀ' *Mehangi.*
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਮਹਿੰਗੀ" *mehangi*	**To:** [expensive:05] **Result:** ["ਮਹਿੰਗੀ":05] **Intermediate form:** ਬਹੁਤ ਮਹਿੰਗੀ ਗੱਡੀ *bhot Mehangi gaddi* Nothing is appended to UW 'ਮਹਿੰਗੀ' *mehangi*, because this root word neither have *'PLR' nor 'MCL'* attribute.

({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਗੱਡੀ" *gaddi*.	**To:** [car:03] **Result:** ["ਗੱਡੀ":03] **Final Output:** ਬਹੁਤ ਮਹਿੰਗੀ ਗੱਡੀ *bhot Mehangi gaddi* Nothing is appended to UW 'ਗੱਡੀ' *gaddi*, because this root word neither have *'PLR' nor 'MCL'* attribute.

Natural language output generated by EUGENE is given in (5.8), correspond to UNL sentence given in (5.6).

{org}

the most expensive car

{/org}

{pun}

ਬਹੁਤ ਮਹਿੰਗੀ ਗੱਡੀ {/pun}

{unl}[W]

mod(car:07.@def, expensive:05.@most)

[/W] {/unl} …(5.8)

NLization of adjectives has further been explained with the help of bit tougher example UNL sentence given in (5.10).

Example 3: English sentence is: a quite beautiful car …(5.9)

unl}

mod(car:07.@indef, :01)

man:01(beautiful:05, quite:03)

{/unl} …(5.10)

Expected Punjabi sentence: ਇੱਕ ਕਾਫੀ ਸੋਹਣੀ ਗੱਡੀ …(5.11)

Transliterated Punjabi sentence: *ikk kafi sohni gaddi*

All nodes (or UWs) identified in UNL sentence given in (5.10), are mapped with their corresponding dictionary entries, and are listed below.

quite:03 - [ਕਾਫੀ]{}"quite"(LEX=A,POS=SAV);

beautiful:05 -

[ਸੋਹਣੀ]{}"beautiful"(LEX=J,POS=ADJ,GEN=FEM,NUM=SNG,PAR=M3);

:01 - []{-1}""(SCOPE)<xxx,0,0>;

car:07.@indef - [ਗੱਡੀ]{}"car" (LEX=N,POS=NOU,GEN=FEM,PAR=M3);

The process of NLization of example UNL sentence given in (5.10) has been illustrated in Table 5.3. Transliteration of each Punjabi word is also shown in the table.

Table 5.3.: NLization process of third example of adjectives

Rule Fired	**Description**	**Action Taken**
(%x,M3):=(%x,-M3,+FLX(SNG:=0>"";PLR:=0>"ਆਂ";MCL:=1>"ਾਂ"));	This paradigm M3 has been defined to attach corresponding postfix to words in Punjabi. Here, attribute *'FLX'* indicate that word has been processed for inflection.	**To:** [car:07.@indef] **Result:** ["ਗੱਡੀ":07.@indef] Nothing is appended to UW 'ਗੱਡੀ' *gaddi*, because this root word does not have any combination of attributes as are involved in this rule.
(%x,M3):=(%x,-M3,+FLX(SNG:=0>"";PLR:=0>"ਆਂ";MCL:=1>"ਾਂ"));	This paradigm M3 has been defined to attach corresponding postfix to words in Punjabi.	**To:** [beautiful:05] **Result:** ["ਸੋਹਣੀ":05] Nothing is appended to UW 'ਸੋਹਣੀ' *sohni*, because this root word does not have any combination of attributes as are involved in this rule.
mod(%a,N,@indef;SCOPE,%b):=(("ਇੱਕ ")(%b)(" ")(%a,-@indef),%c);	This T-rule has been defined to resolve the *'mod'* relation between noun and scope node, which is a complete sentence in itself and acting as an object in this sentence.	**To:** mod(car:07.@indef, :01) **Result:** [sc:02(#L:02(ਇੱਕ :02, :01); #L:02(:01, -:04); #L:02(-:04, car:07))] As a result, *'mod'* relation is resolved by placing scope node before noun with space between them. Also, attribute *"@indef"* is removed from noun and a special word "ਇੱਕ " *ikk* is placed before scope node.
man(%a,J;%b,A):=((%b)(" ")(%a));	This T-rule has been defined to resolve the *'man'* relation between adjective and an article.	**To:** man:01(beautiful:05, quite:03) **Result:** [sc:03(#L:03(quite:03, -:08); #L:03(-:08, beautiful:05))] **Intermediate Result:** ਇੱਕ ਕਾਫੀ ਸੋਹਣੀ ਗੱਡੀ *ikk kafi sohni gaddi* As a result, *'man'* relation is resolved by placing article before

		adjective with space between them.
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm to inflect the root word "ਗੱਡੀ" *gaddi* based on number and gender information associated with the word.	**To:** [car:07] **Result:** ["ਗੱਡੀ":07] **Final Output:** ਇੱਕ ਕਾਫੀ ਸੋਹਣੀ ਗੱਡੀ *ikk kafi sohni gaddi* Nothing is appended to UW 'ਗੱਡੀ' *gaddi*, because this root word neither have *'PLR' nor 'MCL'* attribute.

Natural language output generated by EUGENE is given in (5.12), correspond to UNL sentence given in (5.10).

{org}
a quite beautiful car
{/org}
{pun}
ਇੱਕ ਕਾਫੀ ਸੋਹਣੀ ਗੱਡੀ {/pun}
{unl}[W]
mod(car:07.@indef, :01)
man:01(beautiful:05, quite:03)
[/W] {/unl} …(5.12)

5.2 NLization of Conjunctions

Conjunctions are used to join words, phrases, or independent clauses in compound sentences. For example, 'ਅਤੇ/ਤੇ' *ate/te* "and", 'ਜਾਂ' *jaan* "or" are acting as co-ordinate conjunctions. Sub-ordinate conjunctions are typically used to introduce dependent clauses in complex sentences [7]."For example, 'ਜੇ' *je* "if", ਉਂਞ *unj* "though", ਤਾਂ *tan* "then" *etc.*

Generation process for phrases containing conjunctions from UNL senetences is usually dependent on the following things.

- Conjunction
- Subject or Object

The process of NLization of input UNL sentence containing conjuctions to natural language sentence is illustrated with an example sentence given in (5.14).

Example 1: English sentence is: John and Mary …(5.13)

unl}

and(Mary, John)

{/unl} …(5.14)

Expected Punjabi sentence: ਜਾਨ ਅਤੇ ਮੈਰੀ …(5.15)

Transliterated Punjabi sentence: *jan ate mari*

Input UNL sentence contains *"and"* relation between two nouns. While generating Punjabi NL from it, second noun get placed before the first one, with "and" as conjunction between them.

All nodes (or UWs) identified in UNL sentence given in (5.14), are mapped with their corresponding dictionary entries, and are listed below.

Mary:01 - [ਮੈਰੀ]{}"Mary"(LEX=N,POS=NOU,GEN=FEM,NUM=SNG);

John:02 - [ਜਾਨ]{}"John"(LEX=N,POS=NOU,GEN=MCL,NUM=SNG);

The process of NLization of example UNL sentence given in (5.14) has been illustrated in Table 5.4. Transliteration of each Punjabi word is also shown in the table.

Table 5.4.: NLization process of first example of conjunction

Rule Fired	**Description**	**Action Taken**

`and(%x;%y):=((%y,+>BLK)([and],LEX=C,POS=CCJ,+>BLK)(%x,+>BLK),+LEX=%x);`	This T-rule has been defined to resolve the *'and'* relation between two UWs. It adds *'>BLK'* attribute with each word for inserting blank spaces between the words later on. Also adds "and" between the words which can be translated to Punjabi later on with direct mapping.	**To:** `and(Mary:01, John:02)`
		Result:
		`[sc:01(`
		`#L:01(John:02, and:04);`
		`#L:01(and:04, Mary:01))]`
		Intermediate Result:
		ਜਾਨandਮੈਰੀ
		janandmari
		As a result, relation *'and'* is resolved; "John" is placed before "Mary" with word "and" as conjunction between them.

`(%x,>BLK)(%y,^BLK,^PUT,^STAIL):=(%x,->BLK)(" ",+BLK)(%y);`	This T-rule has been defined to add blank space after a node having '>BLK' attribute and its subsequent node does not have *'BLK'*,	**To:** `#L(John:02, and:04)` **Result:** `#L(John:02, -:05);` `#L(-:05, and:04)` **Intermediate Result:** ਜਾਨ andਮੈਰੀ

	'PUT' and *'STAIL'* attributes.	*jan andmari* As a result, a blank space is inserted between "John" and "and".
`(%x,>BLK)(%y,^BLK,^PUT,^STAIL):=(%x,->BLK)(" ",+BLK)(%y);`	This T-rule has been defined to add blank space after a node having *'>BLK'* attribute and its subsequent node does not have *'BLK'*, *'PUT'* and *'STAIL'* attributes.	**To:** `#L(and:04, Mary:01)` **Result:** `#L(and:04, -:06); #L(-:06, Mary:01)` **Intermediate Result:** ਜਾਨ and ਮੈਰੀ *jan and mari* As a result, a blank space is inserted between "and" and "Mary".
`([and]):=([ਅਤੇ]);`	This T-rule generates Punjabi for word "and" with direct mapping.	**To:** `[and:04]` **Result:** `["ਅਤੇ":04]` **Final Output:** ਜਾਨ ਅਤੇ ਮੈਰੀ *jan ate mari* "and" get translated to 'ਅਤੇ' *ate* directly.

Natural language output generated by EUGENE is given in (5.16), correspond to UNL sentence given in (5.14).

{org}

John and Mary

{/org}

{pun}

ਜਾਨ ਅਤੇ ਮੈਰੀ {/pun}

{unl}[W]

and(Mary, John)

[/W] {/unl} …(5.16)

NLization of conjunctions has further been explained with the help of bit tougher example UNL sentence given in (5.18).

Example 2: English sentence is: John, Mary, Peter and Paul …(5.17)

{unl}

and(Pual, :02)

and:02(Peter, :01)

and:01(Mary, John)

{/unl} …(5.18)

Expected Punjabi sentence: ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ ਅਤੇ ਪਾਲ …(5.19)

Transliterated Punjabi sentence: *jan, mari, peetar ate pal*

In UNL sentence given in (5.18), last *"and"* relation has two nouns, which are "Mary" and "John". This *"and"* relation is a scope node ':01' and acts as the object in another *"and"* relation, which itself is a scope node ':02' and acts as the object in another *"and"* relation to form a complete and meaningful sentence.

All nodes (or UWs) identified in UNL sentence given in (5.18), are mapped with their corresponding dictionary entries, and are listed below.

Mary:01 - [ਮੈਰੀ]{}"Mary"(LEX=N,POS=NOU,GEN=FEM,NUM=SNG);

John:03 - [ਜਾਨ]{}"John"(LEX=N,POS=NOU,GEN=MCL,NUM=SNG);

Peter:02 - [ਪੀਟਰ]{}"Peter"(LEX=N,POS=NOU,GEN=MCL,NUM=SNG);

Paul:04 - [ਪਾਲ]{}"Paul"(LEX=N,POS=NOU,GEN=MCL,NUM=SNG);

The process of NLization of example UNL sentence given in (5.18) has been illustrated in Table 5.5. Transliteration of each Punjabi word is also shown in the table.

Table 5.5: NLization process of second example of conjunction

Rule Fired	**Description**	**Action Taken**
`and(%x;%y):=((%y,+>BLK)([and],LEX=C,POS=CCJ,+>BLK)(%x,+>BLK),+LEX=%x);`	This T-rule has been defined to resolve the *'and'* relation between two UWs. It adds *'>BLK'* attribute with each word for inserting blank spaces between the words later on. Also adds "and" between the words which can be translated to Punjabi, later on with direct mapping.	**To:** `and(Paul:04, :02)` **Result:** `[sc:03(#L:03(:02, and:06); #L:03(and:06, Paul:04))]` **Intermediate Result:** :02andਪਾਲ *:02andpal* As a result, first relation *'and'* is resolved; scope node ":02" to be resolved is placed before "Paul" with word "and" as conjunction between them.
`and(%x;%y):=((%y,+>BLK)([and],LEX=C,POS=CCJ,+>BLK)(%x,+>BLK),+LEX=%x);`	This T-rule has been defined to resolve the *'and'* relation between two UWs.	**To:** `and:02(Peter:02, :01)` **Result:** `[sc:04(#L:04(:01, and:08); #L:04(and:08,Peter:02))]y);` **Intermediate Result:** :01andਪੀਟਰandਪਾਲ *:01andpeetarandpal*

		As a result, second relation *'and'* is resolved; scope node ":01" to be resolved is placed before "Peter" with word "and" as conjunction between them.
`and(%x;%y):=((%y,+>BLK)([and],LEX=C,POS=CCJ,+>BLK)(%x,+>BLK),+LEX=%x);`	This T-rule has been defined to resolve the *'and'* relation between two UWs.	**To:** `and:01(Mary:01, John:03)` **Result:** `[sc:05(#L:05(John:03, and:0A); #L:05(and:0A, Mary:01))]` **Intermediate Result:** ਜਾਨandਮੈਰੀandਪੀਟਰandਪਾਲ *janandmariandpeetarandpal* As a result, relation *'and'* is resolved; "John" is placed before "Mary" with word "and" as conjunction between them.
`([and],%c1)(%x,^ADJ)({[and]|COMMA},%c2):=([, ],",",+PUT=COMMA,+>BLK,$c3)(%x)(%c2);`	This T-rule has been defined to add ", " (comma and a blank space) before a node which does not have *'ADJ'* attribute and its subsequent node is either "and" or "comma". This rule will keep on firing itself till it finds three nodes in this form "([and],%c1)(%x) ({[and]|COMMA},%c2)" for transforming "and" to ", ". So this rule will be fired for all pairs of UWs having "and" as conjunction between them, except for last pair.	**To:** `#L(and:0A, Mary:01); #L(Mary:01, ,:0B)` **Result:** `#L(, :0C, Mary:01); #L(Mary:01, , :0B)` **Intermediate Result:** ਜਾਨ, ਮੈਰੀandਪੀਟਰandਪਾਲ *jan, mariandpeetarandpal* As a result, ", " (comma and a blank space) is inserted between "John" and "Mary".
`([and],%c1)(%x,^ADJ)({[and]|COMMA},%c2):=([, ],",",+PUT=COMMA,+>BLK,$c3)(%x)(%c2);`	This T-rule has been defined to add ", " (comma and a blank space) before a node which does not have *'ADJ'* attribute and its subsequent node is either "and" or "comma".	**To:** `#L(and:09, Peter:02); #L(Peter:01, ,:0D)` **Result:** `#L(, :0E, Peter:02); #L(Peter:02, , :0D)` **Intermediate Result:** ਜਾਨ, ਮੈਰੀ, ਪੀਟਰandਪਾਲ *jan, mariandpeetarandpal* As a result, ", " (comma and a blank space) is inserted between "Mary" and "Peter".
`(%x,>BLK)(%y,^BLK,^PUT,^STAIL):=(%x,->BLK)(" ",+BLK)(%y);`	This T-rule has been defined to add blank space after a node having *'>BLK'* attribute and its subsequent node does not have *'BLK'*, *'PUT'* and *'STAIL'* attributes. Also, *'>BLK'* attribute is removed from first node, So that this relation should not get	**To:** `#L(Peter:02, and:06)` **Result:** `#L(Peter:02, -:0P); #L(-:0P, and:06)` **Intermediate Result:** ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ andਪਾਲ *jan, mari, peetar andpal* As a result, a blank space is inserted between "Peter" and

	execute for this UWs.	"and".
`(%x,>BLK)(%y,^BLK,^PUT,^STAIL) :=(%x,->BLK)(" ",+BLK)(%y);`	This T-rule has been defined to add blank space after a node having '>BLK' attribute and its subsequent node does not have *'BLK'*, *'PUT'* and *'STAIL'* attributes.	**To:** `#L(and:06, Paul:04)` **Result:** `#L(and:06, -:0Q); #L(-:0Q, Paul:04)` **Intermediate Result:** ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ and ਪਾਲ *jan, mari, peetar and pal* As a result, a blank space in inserted between "and" and "Paul".
`([and]):=([ਅਤੇ]);`	This T-rule generates Punjabi for word "and" with direct mapping.	**To:** `[and:04]` **Result:** `["ਅਤੇ":04]` **Final Output:** ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ ਅਤੇ ਪਾਲ *jan, mari, peetar ate pal* "and" get translated to 'ਅਤੇ' *ate* directly.

Natural language output generated by EUGENE is given in (5.20), correspond to UNL sentence given in (5.18).

{org}
John, Mary, Peter and Paul
{/org}
{pun}
ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ ਅਤੇ ਪਾਲ {/pun}

{unl}[W]
and(Pual, :02)
and:02(Peter, :01)
and:01(Mary, John)
[/W] {/unl} …(5.20)

NLization of conjunctions has further been explained for 'ਜਾਂ' *jaan* "or", with the help of example UNL sentence given in (5.22).

Example 3: English sentence is: John, Mary, Peter, Paul or Bill. …(5.21)
{unl}
or:(Bill, :03)

or:03(Pual, :02)

or:02(Peter, :01)

or:01(Mary, John)

{/unl} …(5.22)

Expected Punjabi sentence: ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ ਜਾਂ ਪਾਲ …(5.23)

Transliterated Punjabi sentence: *jan, mari, peetar jaan pal*

In UNL sentence given in (5.22), last *"or"* relation has two nouns, which are "Mary" and "John". This *"or"* relation is a scope node ':01' and acts as the object in another *"or"* relation, which is a scope node ':02' and acts as the object in another *"or"* relation, which itself is a scope node ':03' and acts as the object in another *"or"* relation to form a complete and meaningful sentence.

All nodes (or UWs) identified in UNL sentence given in (5.22), are mapped with their corresponding dictionary entries, and are listed below.

Mary:01 - [ਮੈਰੀ]{}"Mary"(LEX=N,POS=NOU,GEN=FEM,NUM=SNG);

John:04 - [ਜਾਨ]{}"John"(LEX=N,POS=NOU,GEN=MCL,NUM=SNG);

Peter:03 - [ਪੀਟਰ]{}"Peter"(LEX=N,POS=NOU,GEN=MCL,NUM=SNG);

Paul:05 - [ਪਾਲ]{}"Paul"(LEX=N,POS=NOU,GEN=MCL,NUM=SNG);

Bill:02 - [ਬਿਲ]{}"Bill"(LEX=N,POS=NOU,GEN=MCL,NUM=SNG);

The process of NLization of example UNL sentence given in (5.22) has been illustrated in Table 5.6. Transliteration of each Punjabi word is also shown in the table.

Table 5.6: NLization process of third example of conjunction

Rule Fired	**Description**	**Action Taken**
`or(%x;%y):=((%y,+>BLK)([or], LEX=C,POS=CCJ,+>BLK)(%x,+>BLK),+LEX=%x);`	This T-rule has been defined to resolve the *'or'* relation between two UWs. It adds *'>BLK'* attribute with each word for inserting blank spaces between the words later on. Also adds "or" between the words which can be translated to Punjabi, later on with direct mapping or may replaced be comma.	**To:** `or(Bill:02, :03)` **Result:** `[sc:04(#L:04(:03, or:07); #L:04(or:07, Bill:02))]` **Intermediate Result:** :03orਬਿਲ *:03orbil* As a result, first relation *'or'* is resolved; scope node ":03" to be resolved is placed before "Bill" with word "or" as conjunction

		between them.
or(%x;%y):=((%y,+>BLK)([or],LEX=C,POS=CCJ,+>BLK)(%x,+>BLK),+LEX=%x);	This T-rule has been defined to resolve the *'or'* relation between two UWs. It adds *'>BLK'* attribute with each word for inserting blank spaces between the words later on. Also adds "or" between the words.	**To:** or:03(Paul:05, :02) **Result:** [sc:05(#L:05(:02 , or:09); #L:05(or:09, Paul:05))] **Intermediate Result:** :02orਪਾਲorਬਿਲ *:02orpalorbil* As a result, first relation *'or'* is resolved; scope ":02" to be resolved is placed before "Paul" with word "or" as conjunction between them.
or(%x;%y):=((%y,+>BLK)([or],LEX=C,POS=CCJ,+>BLK)(%x,+>BLK),+LEX=%x);	This T-rule has been defined to resolve the *'or'* relation between two UWs. It adds *'>BLK'* attribute with each word for inserting blank spaces between the words later on. Also adds "or" between the words.	**To:** or:02(Peter:03, :01) **Result:** [sc:06(#L:06(:01, or:0B); #L:06(or:0B, Peter:03))] **Intermediate Result:** :01orਪੀਟਰorਪਾਲorਬਿਲ *:01orpeetarorpalorbil* As a result, first relation *'or'* is resolved; scope ":01" to be resolved is placed before "Peter" with word "or" as conjunction between them.
or(%x;%y):=((%y,+>BLK)([or],LEX=C,POS=CCJ,+>BLK)(%x,+>BLK),+LEX=%x);	This T-rule has been defined to resolve the *'or'* relation between two UWs. It adds *'>BLK'* attribute with each word for inserting blank spaces between the words later on. Also adds "or" between the words.	**To:** or:01(Mary:01, John:04) **Result:** [sc:07(#L:07(John:04, or:0D); #L:07(or:0D, Mary:01))] **Intermediate Result:** ਜਾਨorਮੈਰੀorਪੀਟਰorਪਾਲorਬਿਲ *janorpeetarorpalorbil* As a result, relation *'or'* is resolved; "John" is placed before "Mary" with word "or" as conjunction between them.
([or],%c1)(%x,^ADJ)({[or]|COMMA}, %c2):=([,],",",+PUT=COMMA,+>BLK,$c3)(%x)(%c2);	This T-rule has been defined to add "," (comma) before a node which does not have *'ADJ'* attribute and its subsequent node is either "or" or "comma". This rule will keep on firing itself till it found three nodes in this form "([or],%c1)(%x) ({[or]|COMMA},%c2)" for transforming "or" to ",". So this rule will be fired for all pairs UWs having "or" as conjunction between them, except for last pair.	**To:** #L(or:09, Paul:05); #L(Paul:05, or:07) **Result:** #L(,:0E, Paul:05); #L(Paul:05, or:07) **Intermediate Result:** ਜਾਨ,ਮੈਰੀorਪੀਟਰorਪਾਲorਬਿਲ *jan,mariorpeetarorpalorbil* As a result, "," (comma) is inserted between "John" and "Mary".

([or],%c1)(%x)({[or]|COMMA},%c2):=([,],",",+PUT=COMMA,+>BLK,$c3)(%x)(%c2);	This T-rule has been defined to add "," (comma) before a node which does not have *'ADJ'* attribute and its subsequent node is either "or" or "comma".	**To:** #L(or:0B, Peter:03); #L(Peter:03, ,:0E) **Result:** #L(,:0F, Peter:03); #L(Peter:03, ,:0E) **Intermediate Result:** ਜਾਨ,ਮੈਰੀ,ਪੀਟਰorਪਾਲorਬਿਲ *jan,mari,peetarorpalorbil* As a result, "," (comma) is inserted between "Mary" and "Peter".
([or],%c1)(%x)({[or]|COMMA},%c2):=([,],",",+PUT=COMMA,+>BLK,$c3)(%x)(%c2);	This T-rule has been defined to add "," (comma) before a node which does not have *'ADJ'* attribute and its subsequent node is either "or" or "comma".	**To:** #L(or:0D, Mary:01); #L(Mary:01, ,:0F) **Result:** #L(,:0G, Mary:01); #L(Mary:01, ,:0F) **Intermediate Result:** ਜਾਨ,ਮੈਰੀ,ਪੀਟਰ,ਪਾਲorਬਿਲ *jan,mari,peetar,palorbil* As a result, "," (comma) is inserted between "Peter" and "Paul".
(%x,>BLK)(%y,^BLK,^PUT,^STAIL):=(%x,->BLK)(" ",+BLK)(%y);	This T-rule has been defined to add blank space after a node having *'>BLK'* attribute and its subsequent node does not have *'BLK', 'PUT'* and *'STAIL'* attributes. Also, *'>BLK'* attribute is removed from first node, So that this relation should not get execute for this UWs.	**To:** #L(,:0G, Mary:01) **Result:** #L(,:0G, -:0H); #L(-:0H, Mary:01) **Intermediate Result:** ਜਾਨ, ਮੈਰੀ,ਪੀਟਰ,ਪਾਲorਬਿਲ *jan, mari,peetar,palorbil* As a result, a blank space is inserted between "," and "Mary".
(%x,>BLK)(%y,^BLK,^PUT,^STAIL):=(%x,->BLK)(" ",+BLK)(%y);	This T-rule has been defined to add blank space after a node having *'>BLK'* attribute and its subsequent node does not have *'BLK', 'PUT'* and *'STAIL'* attributes.	**To:** #L(,:0F, Peter:03) **Result:** #L(,:0F, -:0I); #L(-:0I, Peter:03) **Intermediate Result:** ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ,ਪਾਲorਬਿਲ *jan, mari, peetar,palorbil* As a result, a blank space is inserted between "," and "Peter".
(%x,>BLK)(%y,^BLK,^PUT,^STAIL):=(%x,->BLK)(" ",+BLK)(%y);	This T-rule has been defined to add blank space after a node having *'>BLK'* attribute and its subsequent node does not have *'BLK', 'PUT'* and *'STAIL'* attributes.	**To:** #L(,:0E, Paul:05) **Result:** #L(,:0E, -:0J); #L(-:0J, Paul:05) **Intermediate Result:** ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ, ਪਾਲorਬਿਲ *jan, mari, peetar, palorbil* As a result, a blank space is inserted between "," and "Paul".
(%x,>BLK)(%y,^BLK,^PUT,^STAIL):=(%x,->BLK)(" ",+BLK)(%y);	This T-rule has been defined to add blank space after a node having *'>BLK'* attribute and its subsequent node does not have *'BLK', 'PUT'* and *'STAIL'* attributes.	**To:** #L(Paul:05, or:07) **Result:** #L(Paul:05, -:0K); #L(-:0K, or:07) **Intermediate Result:** ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ, ਪਾਲ orਬਿਲ *jan, mari, peetar, pal orbil*

		As a result, a blank space is inserted between “Paul” and “or”.
`(%x,>BLK)(%y,^BLK,^PUT,^STAIL) :=(%x,->BLK)(" ",+BLK)(%y);`	This T-rule has been defined to add blank space after a node having *‘>BLK’* attribute and its subsequent node does not have *‘BLK’, ‘PUT’* and *‘STAIL’* attributes.	**To:** `#L(or:07, Bill:02)` **Result:** `#L(or:07, -:0L); #L(-:0L, Bill:02)` **Intermediate Result:** ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ, ਪਾਲ or ਬਿਲ *jan, mari, peetar, pal or bil* As a result, a blank space is inserted between “or” and “Bill”.
`([or]):=([ਜਾਂ]);`	This T-rule generates Punjabi for word “or” with direct mapping.	**To:** `[or:07)` **Result:** `["ਜਾਂ":07]` **Final Output:** ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ, ਪਾਲ ਜਾਂ ਬਿਲ *jan, mari, peetar, pal jaan bil* “or” get translated to ‘ਜਾਂ’ *jaan* directly.

Natural language output generated by EUGENE is given in (5.24), correspond to UNL sentence given in (5.23).

{org}

John, Mary, Peter, Paul or Bill

{/org}

{pun}

ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ, ਪਾਲ ਜਾਂ ਬਿਲ {/pun}

{unl}[W]
or:(Bill, :03)
or:03(Pual, :02)
or:02(Peter, :01)
or:01(Mary, John)
[/W] {/unl} …(5.24)

5.3 NLization of Determiners

In Punjabi language, determiners usually precede the nouns. They indicate the kind of reference which the noun has. The determiner ‘ਇੱਕ’ *ikk* “a/an” is the indefinite article. It is used when the noun is singular. In UNL it is represented as *‘@indef’* attribute attached to the noun. The determiner “the” is known as the definite article. It is used before both

singular and plural nouns. In UNL it is represented as *'@def'* attribute attached to the noun. In Punjabi definite article is replaced by none, *e.g.*, "the papers" 'ਕਾਗਜ' *kagj*, "the apple" 'ਸੇਬ' *seb*. The articles "the" and "a/an" are the most common determiners, but there are many other, *e.g.*, 'ਇਹ ਚਾਕੂ' *eh chakku* "this knif", 'ਕੋਈ ਪੱਖਾ' *koi pakha* "any fan" *etc.* Many determiners express quantity, *e.g.*, 'ਦੋਵੇਂ ਮੁੰਡੇ' *doven munde* "both boys", 'ਹਰ ਰਾਤ' *har raat* "every night", 'ਕੁਝ ਬਹਾਨੇ' *kujh bahane* "few excuses" *etc.*

Generation process for the phrases containing determiners from UNL sentences is usually dependent on the following things.

- Determiner
- Subject or Object

The process of NLization of input UNL sentence containing determiners to natural language sentence is illustrated with an example sentence given in (5.26).

Example 1: English sentence is: a book. …(5.25)

{unl}

(book:03.@indef)

{/unl} …(5.26)

Expected Punjabi sentence: ਇੱਕ ਕਿਤਾਬ …(5.27)

Transliterated Punjabi sentence: *ikk kitab*

All nodes (or UWs) identified in UNL sentence given in (5.26), are mapped with their corresponding dictionary entries, and are listed below.

book:03.@indef -

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

The process of NLization of example UNL sentence given in (5.26) has been illustrated in Table 5.7. Transliteration of each Punjabi word is also shown in the table.

Table 5.7: NLization process of first example of Determiners

Rule Fired	**Description**	**Action Taken**

(%x,M1):=(%x,-M1, +FLX(SNG:=0>"";PLR:=0>"ਾਂ";));	This paradigm M1 has been defined to attach corresponding postfix to words in Punjabi. Here attribute *'FLX'* indicate that word has been processed for inflection.	**To:** [book:01.@indef] **Result:** ["ਕਿਤਾਬ":01.@indef] Nothing is appended to UW 'ਕਿਤਾਬ' *kitab*, because this root word does not have any combination of attributes as are involved in this paradigm.
(%x,N,@indef):=("ਇੱਕ ")((%x,-@indef);	This T-rule has been defined to attach special word "ਇੱਕ" *ikk* before a noun word having attribute *'@indef'*.	**To:** [book:01.@indef] **Result:** #L(ਇੱਕ :03, -:02); #L(-:02, book:01.@other) **Intermediate Result:** ਇੱਕ ਕਿਤਾਬ *ikk kitab* Attribute *'@indef'* gets removed and a special word "ਇੱਕ " *ikk* is added before noun 'ਕਿਤਾਬ' *kitab*.
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਕਿਤਾਬ" *kitab* .	**To:** ["ਕਿਤਾਬ":01] **Result:** ["ਕਿਤਾਬ":01] **Final Output:** ਇੱਕ ਕਿਤਾਬ *Ikk kitab* Nothing is appended to UW 'ਕਿਤਾਬ' *kitab*, because this root word does not have any combination of attributes as are involved paradigm.

Natural language output generated by EUGENE is given in (5.28), correspond to UNL sentence given in (5.26).

{org}
a book
{/org}
{pun}
ਇੱਕ ਕਿਤਾਬ {/pun}
{unl}[W]
(book:03.@indef)
[/W]{/unl} …(5.28)

NLization of determiners has further been explained with the help of bit tougher example UNL sentence given in (5.30).

Example 2: English sentence is: Many other books …(5.29)

{unl}

(book:05.@other.@multal)

{/unl} …(5.30)

Expected Punjabi sentence: ਕਈ ਹੋਰ ਕਿਤਾਬਾਂ …(5.31)

Transliterated Punjabi sentence: *kai hor kitaban*

All nodes (or UWs) identified in UNL sentence given in (5.30), are mapped with their corresponding dictionary entries, and are listed below.

book:05.@other.@multal -

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

The process of NLization of example UNL sentence given in (5.30) has been illustrated in Table 5.8. Transliteration of each Punjabi word is also shown in the table.

Table 5.8: NLization process of second example of Determiners

Rule Fired	**Description**	**Action Taken**
`(%x,M1):=(%x,-M1, +FLX(SNG:=0>"";PLR:=0>"ਾਂ"));`	This paradigm M1 has been defined to attach corresponding postfix to verbs in Punjabi. Here attribute *'FLX'* indicate that word has been processed for inflection.	**To:** `[book:01.@indef]` **Result:** `["ਕਿਤਾਬ":01.@indef]` Nothing is appended to UW 'ਕਿਤਾਬ' *kitab*, because this root word does not have any combination of attributes as are involved in this rule.
`(%x,N,SNG,@multal):=("ਕਈ")(" ")(%x,-@multal,-NUM,+NUM=PLR);`	This T-rule has been defined to attach special word "ਕਈ" *kai* before a noun word having attribute *'@multal'* and add "PLR" attributes.	**To:** `["book":01.@other.@multal]` **Result:** `#L(ਕਈ:01, -:02); #L(-:02, "book":01.@other)` **Intermediate Result:** ਕਈ ਕਿਤਾਬ Attribute *'@multal'* gets removed and a special word "ਕਈ" *kai* is added before noun 'ਕਿਤਾਬ' *kitab*.

(%x,N,PLR,@other):=("ਹੋਰ")(" ")(%x,-@other);	This T-rule has been defined to attach special word "ਹੋਰ" *hor* before a noun word having attribute *'@other'*.	**To:** #L["ਕਿਤਾਬ":01.@other] **Result:** #L(ਹੋਰ:05, -:04); #L(-:04, "ਕਿਤਾਬ":01) **Intermediate Result:** ਕਈ ਹੋਰ ਕਿਤਾਬ *kai hor kitab* Attribute *'@other'* gets removed and a special word "ਹੋਰ" *hor* is added before noun 'ਕਿਤਾਬ' *kitab*.
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਕਿਤਾਬ" *kitab* to make it "ਕਿਤਾਬਾਂ" *kitaban*.	**To:** ["ਕਿਤਾਬ":01] **Result:** ["ਕਿਤਾਬਾਂ":01] **Final Output:** ਕਈ ਹੋਰ ਕਿਤਾਬਾਂ *kai hor kitaban*

Natural language output generated by EUGENE is given in (5.32), correspond to UNL sentence given in (5.30).

{org}
Many other books
{/org}
{pun}
ਕਈ ਹੋਰ ਕਿਤਾਬਾਂ {/pun}
{unl}[W]
(book:05.@other.@multal)
[/W] {/unl} …(5.32)

NLization of determiners with a noun in a relation has been explained with the help of example UNL sentence given in (5.34).

Example 3: English sentence is: a lot of their books …(5.33)
{unl}
pos(book:05.@multal, 00:03.@3.@pl)
{/unl} …(5.34)
Expected Punjabi sentence: ਉਹਨਾਂ ਦੀਆਂ ਕਈ ਕਿਤਾਬਾਂ …(5.35)
Transliterated Punjabi sentence: *uhnan dian kai kitaban*

All nodes (or UWs) identified in UNL sentence given in (5.34), are mapped with their corresponding dictionary entries, and are listed below.

00:03.@3.@pl - [ਉਹ]{}"00.@3.@pl" (LEX=R,POS=POD,PAR=M7);

book:05.@multal –

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

The process of NLization of example UNL sentence given in (5.34) has been illustrated in Table 5.9. Transliteration of each Punjabi word is also shown in the table.

Table 5.9: NLization process of third example of Determiners

Rule Fired	**Description**	**Action Taken**
(%x,M1):=(%x,-M1, +FLX(SNG:=0>"";PLR:=0>"ਾਂ";)) ;	This paradigm M1 has been defined to attach corresponding postfix to nouns in Punjabi. Here attribute *'FLX'* indicates that word has been processed for inflection.	**To:** [book:01.@indef] **Result:** ["ਕਿਤਾਬ":01.@indef] Nothing is appended to UW 'ਕਿਤਾਬ' *kitab*, because this root word does not have any combination of attributes as are involved in this rule.
(%x,M7,+FLX(SNG&BEN&PAS:=1>"ਸ ਨੂੰ"; SNG&AGT&PAS:=1>"ਸ ਨੇ"; SNG&POS&MCL&^DET:=1>"ਸ ਦਾ"; SNG&POS&FEM&^DET:=1>"ਸ ਦੀ"; PLR &POS&MCL&^DET:=1>"ਸ ਦੇ"; PLR&POS& FEM&^DET:=1>"ਸ ਦੀਆਂ"; PLR&BEN&PAS&DET:=0>"ਨਾਂ ਨੂੰ"; PLR&POS&DET:=0>"ਨਾਂ";))	This paradigm M7 has been defined to attach corresponding postfix to pronoun 'ਉਹ' *uh* in Punjabi. Here attribute *'FLX'* indicates that word has been processed for inflection.	**To:** [00:03.@3.@pl] **Result:** ["ਉਹ":03.@3.@pl] Nothing is appended to UW 'ਉਹ' *uh*, because this root word does not have any combination of attributes as are involved in this rule.
pos(%a,FEM,N,@multal;%b,@3,@pl ,POD):= (%b,+PLR,+FEM,+POS,+DET,- @pl)(" ਦੀਆਂ ")(%a);	This T-rule has been defined to resolve the *'pos'* relation between noun and pronoun. It places pronoun before the noun and insert a special word "ਦੀਆਂ" *dian* between them. Also it removes *'@pl'* and adds *'PLR', 'POS', 'DET'* attributes from second UW.	**To:** pos(book:01.multal;%b, 00.@3.@pl) **Result:** #L(00.@3, ਦੀਆਂ :02); #L(ਦੀਆਂ :02 , book:01.@multal) **Intermediate Result:** ਉਹ ਦੀਆਂ ਕਿਤਾਬ *uhnan dian kitab* *'pos'* relation get resolved by placing pronoun before the noun and special word "ਦੀਆਂ" *dian* get inserted between them.

(%x,N,SNG,@multal):=("ਕਈ")(" ")(%x,-@multal,-NUM,+NUM=PLR);	This T-rule has been defined to remove an attribute *'@multal'* from a noun, add a special word "ਕਈ" *kai* before it and update the number information of noun to "PLR"	**To:** ["ਕਿਤਾਬ":01.@multal] **Result:** #L(ਕਈ:01, -:03); #L(-:03, "ਕਿਤਾਬ":01) **Intermediate Result:** ਉਹ ਦੀਆਂ ਕਈ ਕਿਤਾਬ *uhnan dian kai kitab* Attribute *'@multal'* gets removed and a special word "ਕਈ" *kai* is added before noun 'ਕਿਤਾਬ' *kitab*.
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding inflection paradigm to inflect the root word "ਉਹ" *uh* to "ਉਹਨਾਂ" *uhnan* based on the combination *'PLR&POS& DET'* of attributes it has.	**To:** [00:03.@3.@pl] **Result:** ["ਉਹਨਾਂ":03.@3] **Final output:** ਉਹਨਾਂ ਦੀਆਂ ਕਈ ਕਿਤਾਬ *uhnan dian kai kitab* Attribute *'PLR'* was associated with 'ਕਿਤਾਬ' *kitab*, thus, it gets inflected to 'ਕਿਤਾਬਾਂ' *kitaban*
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਕਿਤਾਬ" *kitab* to make it "ਕਿਤਾਬਾਂ" *kitaban* based on the number information of the word	**To:** ["ਕਿਤਾਬ":01] **Result:** ["ਕਿਤਾਬਾਂ":01] **Final output:** ਉਹਨਾਂ ਦੀਆਂ ਕਈ ਕਿਤਾਬਾਂ *uhnan dian kai kitaban* Attribute *'PLR'* was associated with 'ਕਿਤਾਬ' *kitab*, thus, it gets inflected to 'ਕਿਤਾਬਾਂ' *kitaban*

Natural language output generated by EUGENE is given in (5.36), correspond to UNL sentence given in (5.34).

{org}

a lot of their books

{/org}

{pun}

ਉਹਨਾਂ ਦੀਆਂ ਕਈ ਕਿਤਾਬਾਂ {/pun}

{unl}[W]

pos(book:05.@multal, 00:03.@3.@pl)

[/W] {/unl} …(5.36)

5.4 NLization of Verbs

In a Punjabi sentence, verbs must agree with the subject or object of the sentence in terms of gender, number, and person. Punjabi verbs change forms for gender, number, person and tense. The verbs have assigned transitivity and causality. In Punjabi, there are two auxiliary verbs – 'ਹੈ' *hai* for present tense (*e.g.,* 'ਰਾਮ ਅੰਬ ਖਾਂਦਾ ਹੈ' *ram amb khanda hai* "Ram eats mango") and 'ਸੀ' *si* for past tense (*e.g.,* 'ਰਾਮ ਨੇ ਅੰਬ ਖਾਧਾ ਸੀ' *'ram ne amb khadha si* "Ram had eaten mango"). All the forms of these two auxiliary verbs can equally be used for both the genders. For future tense in sentences, *'EGA'* form of main verb is used and in those sentences auxiliary verb is thus not used (*e.g.,* 'ਰਾਮ ਅੰਬ ਖਾਵੇਗਾ' *ram amb khawega* "Ram will eat mango") [7]. Thus, Generation process for the phrases containing verbs from UNL sentences is usually dependent on the following things.

- Person
- Tense of sentence
- Gender of Subject or Person

The process of NLization of input UNL sentence containing verbs to natural language sentence is illustrated with an example sentence given in (5.38).

Example 1: English sentence is: He arrived …(5.37)

{unl}

agt(arrive:03.@past, 00:01.@3.@male)

{/unl} …(5.38)

Expected Punjabi sentence: ਉਹ ਪਹੁੰਚ ਗਿਆ ਸੀ …(5.39)

Transliterated Punjabi sentence: *uh phunch giya si*

Input UNL sentence contains *"agt"* relation between verb and a pronoun. While generating Punjabi NL from it, pronoun gets placed before the verb.

All nodes (or UWs) identified in UNL sentence given in (5.38), are mapped with their corresponding dictionary entries, and are listed below.

"arrive:03" - [ਪਹੁੰਚ]{}"arrive"(LEX=V,POS=VER,TRA=TSTI,PAR=M3);

"00:01" - [ਉਹ]{}"00.@3.@male" (LEX=D,POS=POD,PAR=M7);

The process of NLization of example UNL sentence given in (5.38) has been illustrated in Table 5.10. Transliteration of each Punjabi word is also shown in the table.

Table 5.10: NLization process of first example of verbs

Rule Fired	Description	Action Taken
(%x,M3):=(%x,-M3,+FLX(PST&SNG&MCL&^PGS&^ANT:=0>" ਗਿਆ ਸੀ";PST&SNG&FEM&^PGS&^ANT:=0>" ਗਈ ਸੀ";PST&PLR&FEM&^ANT:=0>" ਗਈਆਂ ਸਨ";PST&PLR&MCL:=0>" ਗਏ ਸਨ";PRS&SNG&MCL&^PGS&^PER:=0>" ਦਾ ਹੈ";PRS&SNG&FEM&^PGS&^PER:=0> "ਦੀ ਹੈ";PRS&PLR&FEM:=0>" ਦਿਆ ਹਨ"; PRS&PLR&MCL:=0>" ਦੇ ਹਨ"; PRS&SNG&MCL&PGS:=0>" ਰਿਹਾ ਹੈ"; PRS&PLR&MCL&PGS:=0>" ਰਹੇ ਹਨ"; PST&SNG&MCL&PGS:=0>" ਰਿਹਾ ਸੀ"; PST&SNG&FEM&PGS:=0>" ਰਹੀ ਸੀ"; PST&PLR&FEM&PGS:=0>" ਰਹੀਆਂ ਸਨ"; PST&PLR&MCL&PGS:=0>" ਰਹੇ ਸਨ"; {PST&MCL&SNG&ANT}:=0>" ਚੁੱਕਾ ਸੀ"; {PST&FEM&SNG&ANT}:=0>"ਚੁੱਕੀ ਸੀ"; {PST&MCL&PLR&ANT}:=0>" ਚੁੱਕੇ ਸਨ"; {PST&FEM&PLR&ANT}:=0>" ਚੁੱਕੀਆਂ ਸਨ"; {PER&PRS&MCL&SNG}:=0>" ਚੁੱਕਾ ਹੈ"; {PER&PRS&FEM&SNG}:=0>" ਚੁੱਕੀ ਹੈ"; {PER&FUT&MCL&SNG}:=0>" ਚੁੱਕਾ ਹੋਵੇਗਾ"; {PER&FUT&FEM&SNG} :=0>" ਚੁੱਕੀ ਹੋਵੇਗੀ"; {FUT&MCL&PGS& SNG}:=0>" ਰਿਹਾ ਹੋਵੇਗਾ";{FUT&FEM& PGS&SNG}:=0>" ਰਿਹਾ ਹੋਵੇਗੀ";{FUT& MCL&SNG}:=0>"ੇਗਾ";FUT&FEM& SNG&^PGS&^PER&^RES:=0>"ੇਗੀ";));	This paradigm M3 has been defined to attach corresponding postfix to verbs in Punjabi. Here attribute *'FLX'* indicates that word has been processed for inflection.	**To:** [arrive:03.@past] **Result:** ["ਪਹੁੰਚ":03.@past] Nothing is appended to UW 'ਪਹੁੰਚ' *phunch*, because this root word does not have any combination of attributes as are involved in this rule.
(%x,M7,+FLX(SNG&BEN&PAS:=1>"ਸ ਨੂੰ"; SNG&AGT&PAS:=1>"ਸ ਨੇ"; SNG&POS&MCL&^DET:=1>"ਸ ਦਾ"; SNG&POS&FEM&^DET:=1>"ਸ ਦੀ"; PLR &POS&MCL&^DET:=1>"ਸ ਦੇ"; PLR&POS& FEM&^DET:=1>"ਸ ਦੀਆਂ";	This paradigm M7 has been defined to attach corresponding postfix to pronoun 'ਉਹ' *uh* in Punjabi. Here attribute *'FLX'* indicates that word has been processed for inflection.	**To:** [00:03.@3.@pl] **Result:** ["ਉਹ":03.@3.@pl] Nothing is appended to UW 'ਉਹ' *uh*, because this root word does not have any combination of attributes as are involved in this rule.

PLR&BEN&PAS&DET:=0>"ਨਾਂ ਨੂੰ"; PLR&POS&DET:=0>"ਨਾਂ";))		
agt(%a,V,@past,^PAS;%b,R):= agt(%a ,+PAS,-@past;%b) ;	It adds *'PER'* attribute to the verb depending upon attributes associated with the node, because *'PAS'* combination is necessary for appending appropriate postfix to Punjabi root word verb.	**To:** agt(arrive:03.@past, 00:01.@3.@male) **Result:** agt(arrive:03, 00:01.@3) Nothing is appended to any Punjabi root word as postfix, *'@past'* attribute is removed from second node *'%b'* and *'PAS'* is added to first node *'%a'*.
agt(%a,^MCL;%b,R,@male):= agt(%a ,+MCL,-NUM,+NUM=SNG;%b,-@male);	It adds *'MCL'* and *'SNG'* attributes to the verb because of *'@male'* attribute associated with the second node '%b', as *'PAS&MCL&SNG'* combination is necessary for appending appropriate postfix to Punjabi root word.	**To:** agt(arrive:03 , 00:01.@3.@male) **Result:** agt(arrive:03, 00:01.@3) Nothing is appended to any Punjabi root word as postfix, just *'@male'* attribute is removed from second node *'%b'* and *'MCL'* and *'SNG'* are added to first node *'%a'*.
agt(%a,V;%b,R):=(%b)(" ")(%a);	It resolves the *'agt'* relation, places pronoun before verb and introduces new node of single space (" ") between two nodes.	**To:** agt(arrive:03, 00:01.@3) **Result:** #L(00:01.@3, -:02); #L(-:02, arrive:03) **Intermediate Result:** ਉਹ ਪਹੁੰਚ *uh phunch* As a result, *'agt'* relation is resolved by placing the pronoun 'ਉਹ' *uh* before verb 'ਪਹੁੰਚ' *phunch*.
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਉਹ" *uh*.	**To:**[00:01.@3] **Result:**["ਉਹ":01.@3] Nothing is appended to UW 'ਉਹ' *uh*, because this root word does not have any combination of attributes as are involved in this rule.
{N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding inflection paradigm to inflect the root word "ਪਹੁੰਚ" *phunch* to make it "ਪਹੁੰਚ ਗਿਆ ਸੀ" *phunch giya si*	**To:**[arrive:03] **Result:**["ਪਹੁੰਚ ਗਿਆ ਸੀ":03] **Final Output:** ਉਹ ਪਹੁੰਚ ਗਿਆ ਸੀ *uh phunch giya si* Root word "ਪਹੁੰਚ" *phunch* get inflected to "ਪਹੁੰਚ ਗਿਆ ਸੀ" *phunch giya si*

Natural language output generated by EUGENE is given in (5.40), correspond to UNL sentence given in (5.38).

{org}
He arrived
{/org}
{pun}
ਉਹ ਪਹੁੰਚ ਗਿਆ ਸੀ {/pun}
{unl}[W]
agt(arrive:03.@past, 00:01.@3.@male)
[/W] {/unl} …(5.40)

NLization of determiners has further been explained with the help of bit tougher example UNL sentence given in (5.42).

Example 2: English sentence is: He will not arrive …(5.41)
{unl}
agt(arrive:03.@not.@future, 00:01.@3.@male)
{/unl} …(5.42)
Expected Punjabi sentence: ਉਹ ਨਹੀਂ ਪਹੁੰਚੇਗਾ …(5.43)
Transliterated Punjabi sentence: *uh nahin phunchega*

All nodes (or UWs) identified in UNL sentence given in (5.42), are mapped with their corresponding dictionary entries, and are listed below.

"00:01" - [ਉਹ]{}"00.@3.@male" (LEX=D,POS=POD,PAR=M7);
arrive:03.@not.@future -
[ਪਹੁੰਚ]{}"arrive"(LEX=V,POS=VER,TRA=TSTI,PAR=M3)**;**

The process of NLization of example UNL sentence given in (5.42) has been illustrated in Table 5.11. Transliteration of each Punjabi word is also shown in the table.

Table 5.11: NLization process of second example of verbs

Rule Fired	**Description**	**Action Taken**
(%x,M3):=(%x,-M3,+FLX(PST&SNG&MCL&^PGS&^ANT:=0>" ਗਿਆ ਸੀ";PST&SNG&FEM&^PGS&^ANT:=0>" ਗਈ ਸੀ";PST&PLR&FEM&^ANT:=0>"	This paradigm M3 has been defined to attach corresponding postfix to verbs in Punjabi. Here attribute *'FLX'* indicates that	**To:** [arrive:03.@future.@not] **Result:** ["ਪਹੁੰਚ":03.@future.@not] Nothing is appended to UW 'ਪਹੁੰਚ'

ਗਈਆਂ ਸਨ";PST&PLR&MCL:=0>" ਗਏ ਸਨ";PRS&SNG&MCL&^PGS&^PER:=0>" ਦਾ ਹੈ";PRS&SNG&FEM&^PGS&^PER:=0> "ਦੀ ਹੈ";PRS&PLR&FEM:=0>" ਦਿਆ ਹਨ"; PRS&PLR&MCL:=0>" ਦੇ ਹਨ"; PRS&SNG&MCL&PGS:=0>" ਰਿਹਾ ਹੈ"; PRS&SNG&FEM&PGS:=0>" ਰਹੀ ਹੈ"; PRS&PLR&FEM&PGS:=0>" ਰਹੀਆਂ ਹਨ"; PRS&PLR&MCL&PGS:=0>" ਰਹੇ ਹਨ"; PST&SNG&MCL&PGS:=0>" ਰਿਹਾ ਸੀ"; PST&SNG&FEM&PGS:=0>" ਰਹੀ ਸੀ"; PST&PLR&FEM&PGS:=0>" ਰਹੀਆਂ ਸਨ"; PST&PLR&MCL&PGS:=0>" ਰਹੇ ਸਨ"; {PST&MCL&SNG&ANT}:=0>" ਚੁੱਕਾ ਸੀ"; {PST&FEM&SNG&ANT}:=0>"ਚੁੱਕੀ ਸੀ"; {PST&MCL&PLR&ANT}:=0>" ਚੁੱਕੇ ਸਨ"; {PST&FEM&PLR&ANT}:=0>" ਚੁੱਕੀਆਂ ਸਨ"; {PER&PRS&MCL&SNG}:=0>" ਚੁੱਕਾ ਹੈ"; {PER&PRS&FEM&SNG}:=0>" ਚੁੱਕੀ ਹੈ"; {PER&FUT&MCL&SNG}:=0>" ਚੁੱਕਾ ਹੋਵੇਗਾ"; {PER&FUT&FEM&SNG} :=0>" ਚੁੱਕੀ ਹੋਵੇਗੀ"; {FUT&MCL&PGS& SNG}:=0>" ਰਿਹਾ ਹੋਵੇਗਾ";{FUT&FEM& PGS&SNG}:=0>" ਰਿਹਾ ਹੋਵੇਗੀ";{FUT& MCL&SNG}:=0>"ੇਗਾ";FUT&FEM& SNG&^PGS&^PER&^RES:=0>"ੇਗੀ";));	word has been processed for inflection.	*phunch*, because this root word does not have any combination of attributes as are involved in this rule.
(%x,M7,+FLX(SNG&BEN&PAS:=1>"ਸ ਨੂੰ"; SNG&AGT&PAS:=1>"ਸ ਨੇ"; SNG&POS&MCL&^DET:=1>"ਸ ਦਾ"; SNG&POS&FEM&^DET:=1>"ਸ ਦੀ"; PLR &POS&MCL&^DET:=1>"ਸ ਦੇ"; PLR&POS& FEM&^DET:=1>"ਸ ਦੀਆਂ"; PLR&BEN&PAS&DET:=0>"ਨਾਂ ਨੂੰ"; PLR&POS&DET:=0>"ਨਾਂ";))	This paradigm M7 has been defined to attach corresponding postfix to pronoun 'ਉਹ' *uh* in Punjabi. Here attribute *'FLX'* indicates that word has been processed for inflection.	**To:** [00:03.@3.@pl] **Result:** ["ਉਹ":03.@3.@pl] Nothing is appended to UW 'ਉਹ' *uh*, because this root word does not have any combination of attributes as are involved in this rule.
agt(%a,V,@future, ^FUT;%b,R):= agt(%a ,+FUT,-@future;%b) ;	It adds *'FUT'* attribute to the verb depending upon attributes associated with the node, because *'FUT'* combination is necessary for appending appropriate	**To:** agt(arrive:03.@not, 00:01.@3.@male) **Result:** agt(arrive:03, 00:01.@3) Nothing is appended to any Punjabi root word as postfix, *'@future'* attribute is removed

	postfix to Punjabi root word verb.	from second node *'%b'* and *'FUT'* is added to first node *'%a'*.
agt(%a,^MCL;%b,R,@male):= agt(%a ,+MCL,-NUM,+NUM=SNG;%b,-@male);	It adds *'MCL'* and *'SNG'* attributes to the verb because of *'@male'* attribute associated with the second node '%b', as *'FUT&MCL&SNG'* combination is necessary for appending appropriate postfix to Punjabi root word.	**To:** agt(arrive:03.@not, 00:01.@3.@male) **Result:** agt(arrive:03.@not, 00:01.@3) Nothing is appended to any Punjabi root word as postfix, just *'@male'* attribute is removed from second node *'%b'* and *'MCL'* and *'SNG'* are added to first node *'%a'*.
agt(%a,V;%b,R):=(%b)(" ")(%a);	It resolves the *'agt'* relation, places pronoun before verb and introduces new node of single space (" ") between two nodes.	**To:** agt(arrive:03.@not, 00:01.@3) **Result:** #L(00:01.@3, -:02); #L(-:02, arrive:03.@not) **Intermediate Result:** ਉਹ ਪਹੁੰਚ *uh phunch* As a result, *'agt'* relation is resolved by placing the pronoun 'ਉਹ' *uh* before 'verb' 'ਪਹੁੰਚ' *phunch*.
(%x,V,@not):=)(" ਨਹੀਂ ")(%x,-@not;	This T-rule has been defined to remove an attribute *'@not'* from a verb, add a special word "ਨਹੀਂ" *nahi* before it.	**To:** [arrive:03.@not] **Result:** **Result:** #L(00:01.@3, -:04); #L(-:04 ਨਹੀਂ :05, arrive:03) **Intermediate Result:** ਉਹ ਨਹੀਂ ਪਹੁੰਚ *uh nahin phunch* Attribute *'@not'* gets removed and a special word , "ਨਹੀਂ" *nahi* is added before verb 'ਪਹੁੰਚ' *phunch*.
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਉਹ" *uh*.	**To:** [00:01.@3] **Result:** ["ਉਹ":01.@3] Nothing is appended to UW 'ਉਹ' *uh*, because this root word does not have any combination of attributes as are involved in this rule.
{N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਪਹੁੰਚ" *phunch* to make it "ਪਹੁੰਚੇਗਾ" *phunchega*	**To:** [arrive:03] **Result:** [" ਪਹੁੰਚੇਗਾ ":07] Final **Output:**

		ਉਹ ਨਹੀਂ ਪਹੁੰਚੇਗਾ *uh nahin phunchega* Root word "ਪਹੁੰਚ" *phunch* get inflected to "ਪਹੁੰਚੇਗਾ" *phunchega*.

Natural language output generated by EUGENE is given in (5.44), correspond to UNL sentence given in (5.42).

{org}
He will not arrive
{/org}
{pun}
ਉਹ ਨਹੀਂ ਪਹੁੰਚੇਗਾ {/pun}
{unl}[W]
agt(arrive:03.@not.@future, 00:01.@3.@male)
[/W]{/unl} …(5.44)

NLization of determiners has further been explained with the help of bit tougher example UNL sentence given in (5.46).

Example 3: English sentence is: He should have not arrived …(5.45)
{unl}
agt(arrive:03.@present.@perfect.@not.@progressive.@possibility, 00:01.@3)
{/unl} …(5.46)
Expected Punjabi sentence: ਉਹ ਨਹੀਂ ਪਹੁੰਚ ਚੁੱਕਾ ਹੈ …(5.47)
Transliterated Punjabi sentence: *uh nahin phunch chukka hai*

All nodes (or UWs) identified in UNL sentence given in (5.46), are mapped with their corresponding dictionary entries, and are listed below.

00:01" - [ਉਹ]{}"00.@3.@male" (LEX=D,POS=POD,PAR=M7);
arrive:09.@not.@present.@perfect.@probability -
[ਪਹੁੰਚ]{}"arrive"(LEX=V,POS=VER,TRA=TSTI,PAR=M3);

The process of NLization of example UNL sentence given in (5.46) has been illustrated in Table 5.12. Transliteration of each Punjabi word is also shown in the table.

Table 5.12: NLization process of second example of verbs

Rule Fired	Description	Action Taken
(%x,M3):=(%x,-M3,+FLX(PST&SNG&MCL&^PGS&^ANT:=0>" ਗਿਆ ਸੀ";PST&SNG&FEM&^PGS&^ANT:=0>" ਗਈ ਸੀ";PST&PLR&FEM&^ANT:=0>" ਗਈਆਂ ਸਨ";PST&PLR&MCL:=0>" ਗਏ ਸਨ";PRS&SNG&MCL&^PGS&^PER:=0>" ਦਾ ਹੈ";PRS&SNG&FEM&^PGS&^PER:=0> "ਦੀ ਹੈ";PRS&PLR&FEM:=0>" ਦਿਆ ਹਨ"; PRS&PLR&MCL:=0>" ਦੇ ਹਨ"; PRS&SNG&MCL&PGS:=0>" ਰਿਹਾ ਹੈ"; PRS&SNG&FEM&PGS:=0>" ਰਹੀ ਹੈ"; PRS&PLR&FEM&PGS:=0>" ਰਹੀਆਂ ਹਨ"; PRS&PLR&MCL&PGS:=0>" ਰਹੇ ਹਨ"; PST&SNG&MCL&PGS:=0>" ਰਿਹਾ ਸੀ"; PST&SNG&FEM&PGS:=0>" ਰਹੀ ਸੀ"; PST&PLR&FEM&PGS:=0>" ਰਹੀਆਂ ਸਨ"; PST&PLR&MCL&PGS:=0>" ਰਹੇ ਸਨ"; {PST&MCL&SNG&ANT}:=0>" ਚੁੱਕਾ ਸੀ"; {PST&FEM&SNG&ANT}:=0>"ਚੁੱਕੀ ਸੀ"; {PST&MCL&PLR&ANT}:=0>" ਚੁੱਕੇ ਸਨ"; {PST&FEM&PLR&ANT}:=0>" ਚੁੱਕੀਆਂ ਸਨ"; {PER&PRS&MCL&SNG}:=0>" ਚੁੱਕਾ ਹੈ"; {PER&PRS&FEM&SNG}:=0>" ਚੁੱਕੀ ਹੈ"; {PER&FUT&MCL&SNG}:=0>" ਚੁੱਕਾ ਹੋਵੇਗਾ"; {PER&FUT&FEM&SNG} :=0>" ਚੁੱਕੀ ਹੋਵੇਗੀ"; {FUT&MCL&PGS& SNG}:=0>" ਰਿਹਾ ਹੋਵੇਗਾ";{FUT& MCL&SNG}:=0>"ੇਗਾ";FUT&FEM& SNG&^PGS&^PER&^RES:=0>"ੇਗੀ";));	This paradigm M3 has been defined to attach corresponding postfix to verbs in Punjabi. Here attribute *'FLX'* indicates that word has been processed for inflection.	**To:** [arrive:03.@future.@not] **Result:** ["ਪਹੁੰਚ":03.@future.@not] Nothing is appended to UW 'ਪਹੁੰਚ' *phunch*, because this root word does not have any combination of attributes as are involved in this rule.
(%x,M7,+FLX(SNG&BEN&PAS:=1>"ਸ ਨੂੰ"; SNG&AGT&PAS:=1>"ਸ ਨੇ"; SNG&POS&MCL&^DET:=1>"ਸ ਦਾ"; SNG&POS&FEM&^DET:=1>"ਸ ਦੀ"; PLR &POS&MCL&^DET:=1>"ਸ ਦੇ"; PLR&POS& FEM&^DET:=1>"ਸ ਦੀਆਂ"; PLR&BEN&PAS&DET:=0>"ਨਾਂ ਨੂੰ"; PLR&POS&DET:=0>"ਨਾਂ";))	This paradigm M7 has been defined to attach corresponding postfix to pronoun 'ਉਹ' *uh* in Punjabi. Here attribute *'FLX'* indicates that word has been processed for inflection.	**To:** [00:03.@3.@pl] **Result:** ["ਉਹ":03.@3.@pl] Nothing is appended to UW 'ਉਹ' *uh*, because this root word does not have any combination of attributes as are involved in this rule.

agt(%a,V,@present,@perfect, ^PRS,^PER,^SNG,^MCL;%b,R,@3,@male):= agt(%a ,+PRS,+PER,+SNG,+MCL;%b); 	It adds *PRS, PER, MCL, SNG* attribute to the verb depending upon attributes associated with the node, because *'PRS&PER&MCL& SNG'* combination is necessary for appending appropriate postfix to Punjabi root word verb.	**To:**agt(arrive:03.@not.@present.@perfect.@probability, 00:01.@3.@male) **Result:** agt(arrive:03.@not.@present.@perfect.@probability, 00:01.@3.@male) Attributes and *PRS, PER, MCL, SNG* are added to first node *'%a'*.
agt(%a,V;%b,R):=(%b)(" ")(%a);	It resolves the *'agt'* relation, places pronoun before verb and introduces new node of single space (" ") between two nodes.	**To:** agt(arrive:03.@not.@present.@perfect.@probability, 00:01.@3.@male) **Result:** [sc:01#L(00:01.@3.@male, -:02); #L:01(-:02, arrive:03.@not.@present.@perfect.@probability)] **Intermediate Result:** ਉਹ ਪਹੁੰਚ *uh phunch* As a result, *'agt'* relation is resolved by placing the pronoun 'ਉਹ' *uh* before verb 'ਪਹੁੰਚ' *phunch*.
(%x,V,@not):=)(" ਨਹੀਂ ")(%x,-@not);	This T-rule has been defined to remove an attribute *'@not'* from a verb, add a special word "ਨਹੀਂ" *nahi* before it.	**To:** [arrive:03.@not] **Result:** **Result:** #L(00:01.@3, -:04); #L(-:04 ਨਹੀਂ :05, arrive:03) **Intermediate Result:** ਉਹ ਨਹੀਂ ਪਹੁੰਚ *uh nahin phunch* Attribute *'@not'* gets removed and a special word , "ਨਹੀਂ" *nahi* is added before verb 'ਪਹੁੰਚ' *phunch*.
({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਉਹ" *uh.*	**To:**[00:01.@3] **Result:**["ਉਹ":01.@3] Nothing is appended to root word 'ਉਹ' *uh* because '*AGT'* was not associated with it.
{N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word "ਪਹੁੰਚ ਚੁੱਕਾ ਹੈ" *phunch* to make it "ਪਹੁੰਚ ਚੁੱਕਾ ਹੈ" *phunch chukka hai*	**To:**[arrive:03] **Result:**["ਪਹੁੰਚ ਚੁੱਕਾ ਹੈ":07] **Final Output:** ਉਹ ਨਹੀਂ ਪਹੁੰਚ ਚੁੱਕਾ ਹੈ *uh nahin phunch chukka hai* Root word "ਪਹੁੰਚ" *phunch* get inflected to "ਪਹੁੰਚ ਚੁੱਕਾ ਹੈ" *phunch chukka hai*

Natural language output generated by EUGENE is given in (5.48), correspond to UNL sentence given in (5.46).

{org}
He should have not arrived
{/org}
{pun}
ਉਹ ਨਹੀਂ ਪਹੁੰਚ ਚੁੱਕਾ ਹੈ{/pun}
{unl}[W]
agt(arrive:03.@present.@perfect.@not.@progressive.@possibility, 00:01.@3)
[/W] {/unl} …(5.48)

Chapter Summary

The proposed NLization of adjectives, conjunctions, determiners and verbs using EUGENE, for generating Punjabi language sentence from a given UNL expression has been discussed, in detail with easy, medium and tough examples in each case, in this chapter. Processing of each UNL sentence is done by traversing through three stages, *i.e.*, identifying all nodes, relations, attributes, dictionary lookup for extracting meanings of UWs from dictionary, firing all the relevant rules. Punjabi natural language sentence acquire several intermediate forms with execution of relevant T-rule in sequence. Some of the attributes get removed from UWs and some attributes get attached to them based on the attributes of other UW or relation. After that, the nodes become ready for generation of morphology according to the target language. Morphology rule base or paradigms are used for Punjabi language to handle attribute label resolution morphology; relation label resolution morphology; adjective and verb morphology.

Chapter 6

Result and Discussion

Universal Networking Language is an internationally standardized interlingua that has the potential to bridge the language barriers across all the languages of the world. It is a natural-language independent system. EUGENE tool is an effective online tool developed by UNDL foundation for UNL to natural language generation.

The system is tested on UNL sentences containing adjectives, conjunctions, determiners and verbs in Corpus500, UC-A1 and UC-A3 for generating sentences of Punjabi language.

About 200 UWs involved in these corpuses and their respective information has been stored in a word dictionary using EUGENE. An entry of the word dictionary contains basically three parts: a headword of Punjabi natural language, a UW and a set of grammatical attributes (features) which defines behavior of word in a sentence. UW appears in the UNL expression and is used as a trigger to obtain an appropriate word to form the target sentence of Punjabi natural language.

About 300 T-rules has been written using EUGENE for NLization of these corpuses. These rules control the process of NLization. The ability of a rule is designed to be able to describe on what condition to perform what operation using the grammatical features both or either defined by the rules and/or given in a word dictionary.

Also, F-measure / F1-score has been calculated for these sentences with the help of online tool developed by UNDL foundation available at UNL^{arium}. F-measure is the measure of accuracy of grammar [33]. Two parameters described below are utilized for calculating F-measure.

i.) Precision

Precision is ratio of number of correct results to the number of all returned results.

ii.) Recall

Recall is the ratio of correct results to the number of results that should have been returned.

This tool considers both the precision and the recall of the grammar to compute the score [34], according to the formula shown in (6.1).

F-measure =2*{(precision*Recall) / (precision + Recall)} …(6.1)

6.1 F-measure for Corpus500

There are total of 143 sentences in Corpus500 containing adjectives, determiners and verbs. The F-measure comes out to be 0.993, as shown in Figure 6.1. Details of results, extracted from calculation of F-measure are listed below.

Sentences processed	= 143
Sentences returned	= 143
Sentences correct	= 142
Precision	= 0.993
Recall	= 0.993

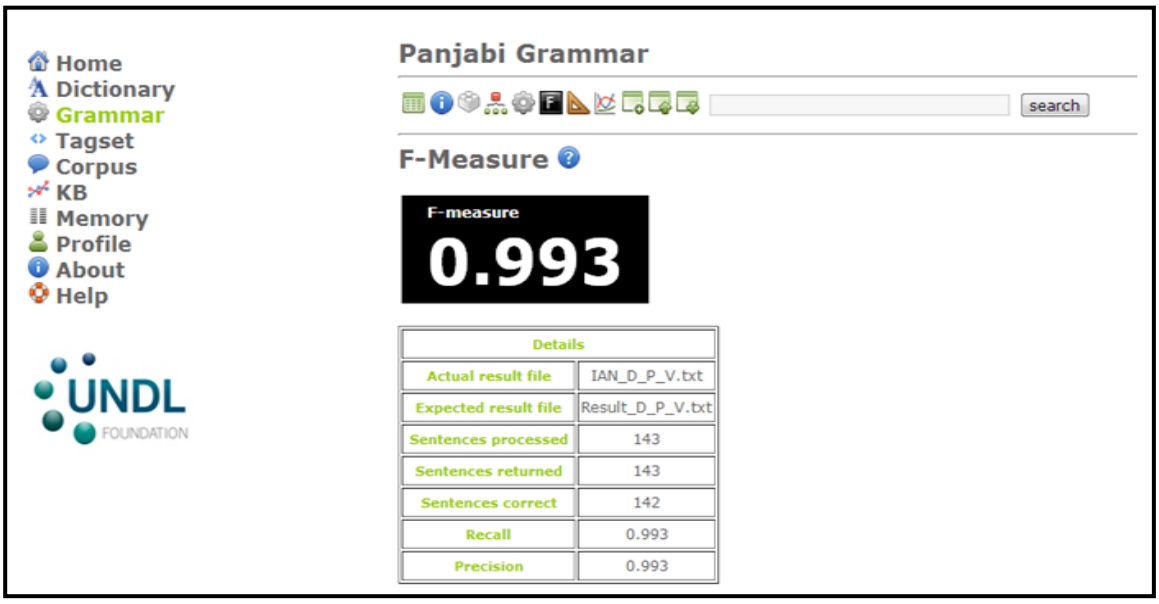


Figure 6.1: Snapshot of F-measure of Corpus500

6.2 F-measure for UC-A1

The UC-A1 contains total of 100 sentences. The combined F-measure for all sentences comes out to be 0.900, as shown in Figure 6.2. Details of results, extracted from calculation of F-

measure are listed below.

Sentences processed	= 100
Sentences returned	= 100
Sentences correct	= 90
Precision	= 0.900
Recall	= 0.900

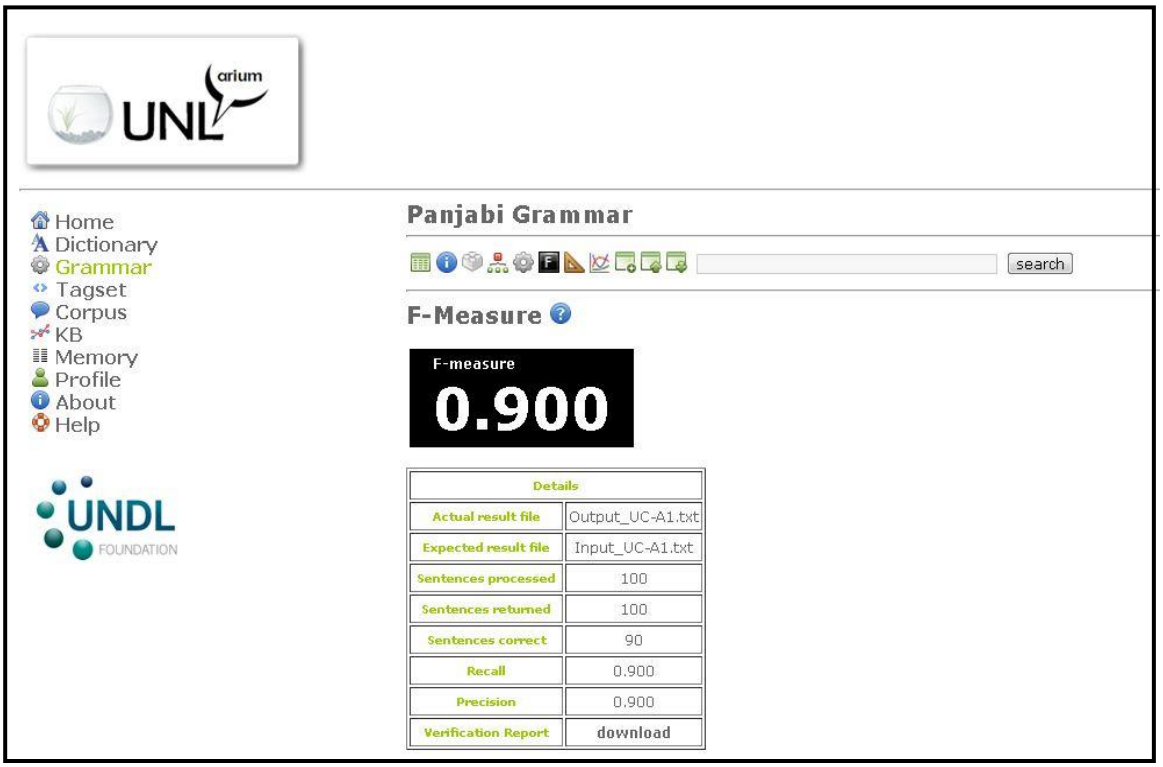


Figure 6.2: Snapshot of F-measure of UC-A1

6.3 F-measure for UC-A2

There are total of 135 sentences in UC-A2 containing adjectives, determiners and verbs. The combined F-measure for all sentences comes out to be 0.900, as shown in Figure 6.3. Details of results, extracted from calculation of F-measure are listed below.

Sentences processed	= 135
Sentences returned	= 135
Sentences correct	= 110
Precision	= 0.815
Recall	= 0.815

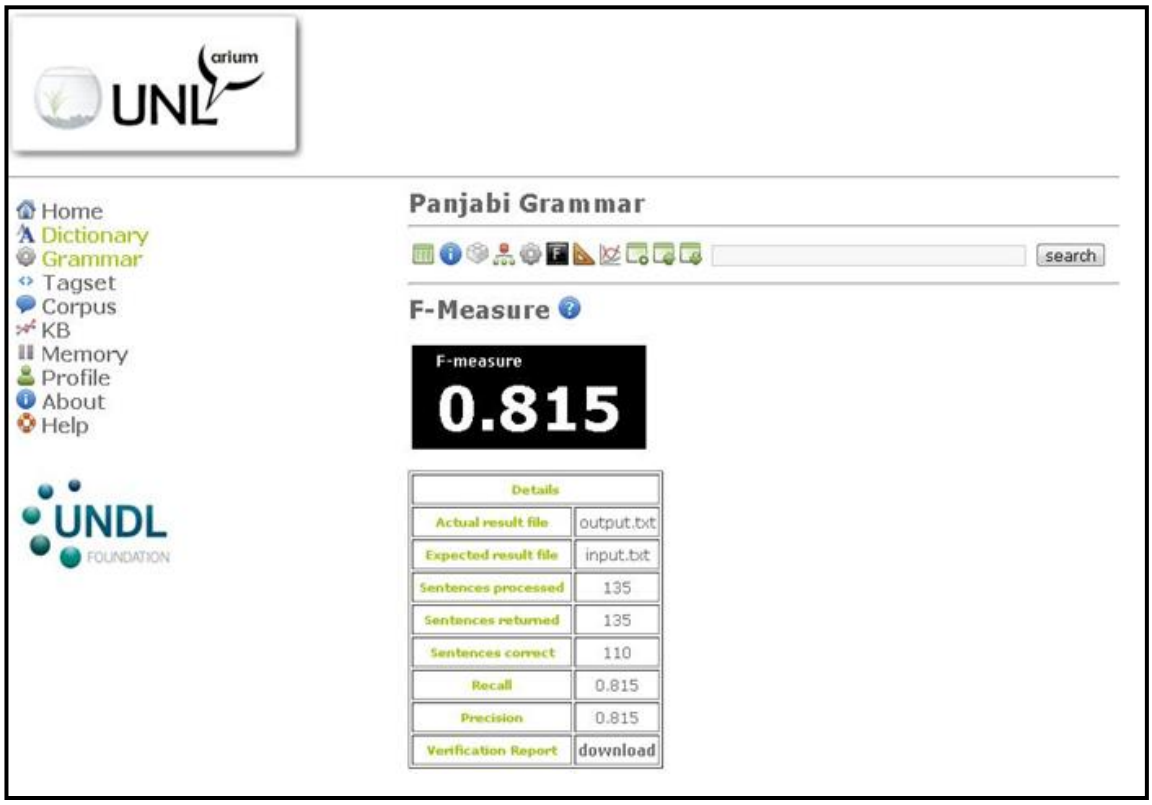


Figure 6.3: Snapshot of F-measure of UC-A2

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

Universal Networking Language is an internationally standardized interlingua that has the potential to bridge the gap across all the languages of the world. The UNL Program was come in existence as by efforts of the Institute of Advanced Studies of the United Nations University in Tokyo, Japan in 1996. The United Nations University set up an autonomous organization in January 2001, named as UNDL Foundation, to be responsible for UNL Programme, for its development and management [2]. In multi-lingual machine translation environment, where large number of languages are to be translated between one-another, it would require a separate system for every language, *i.e.*, $n*(n-1)$ language systems will be required for n number of languages, So, interlingua approach is far more better approach than other approach, because here, only $2n$ language systems are required.

EUGENE tool is an effective online tool for NLization provided by UNDL Foundation and released in 2012. Three corpuses, Corpus500, UC-A1, and UC-A2 are provided by UNDL foundation for testing the system. Generation of Punjabi natural language sentences, containing adjectives, conjunctions, determiners and verbs from interlingua like UNL, with EUGENE using T-rules and UNL-NL dictionary, has been described in this thesis. NLization of these corpuses has been done, using a word dictionary containing the information of about 200 words that correspond to UWs included in the input UNL expressions with their grammatical attributes (features) that describe the behaviors of the words, and about 300 T-rules for describing how to construct a sentence using the information from the input UNL expressions and defined word dictionary. F-measure for Corpus500, UC-A1, and UC-A2 came out to be 0.993, 0.900, and 0.815 respectively.

7.2 Future Scope

- NLization will be done on real Corpuses at paragraph level. UNDL Foundation has provided such corpus UC-B1 which contains different stories like, The Hare and The Tortoise, The Bat and The Weasels, Father and his sons, The Ants and the Grasshopper and The Man and the Lion.
- Since UNL captures semantics of the natural language so semantic based searching system can be developed for Punjabi.
- Work can be extended to carry out NLization of numbers and ordinals.
- System needs to be improved so as to achieve F-Measure of 1.000.
- The proposed system can be tested for other natural languages spoken worldwide.

Appendix A

Some of the Dictionary Words

S. No.	Dictionary Words
1.	[ਉਹ]{}"00.@3.@female" (LEX=R,POS=PPR,CAS=OBL,NUM=SNG,GEN=FEM,PAR=M7) <pun,0,0>;
2.	[ਉਹ]{}"00.@3.@pl" (LEX=R,POS=POD,PAR=M7)<pun,0,0>;
3.	[ਮੈਂ]{}"00.@1" (LEX=R,POS=PPR,CAS=OBL,PER=1PS)<pun,0,0>;
4.	[ਤੁਹਾਡੀ]{}"00.@2" (LEX=D,POS=POD)<pun,0,0>;
5.	[ਮੇਰੀ]{}"00.@1" (LEX=D,POS=POD,PAR=M3)<pun,0,0>;
6.	[ਗੱਡੀ]{}"car" (LEX=N,POS=NOU,FEM,PAR=M4)<pun,0,0>;
7.	[ਕਿਤਾਬ]{}"book"(LEX=N,POS=NOU,NUM=SNG,PAR=M2)<pun,0,0>;
8.	[ਰੋਮ]{}"Rome"(LEX=N,POS=PPN,NUM=SNGT)<pun,0,0>;
9.	[ਬਿਲ]{}"Bill"(LEX=N,POS=PPN,NUM=SNGT)<pun,0,0>;
10.	[ਪੀਟਰ]{}"Peter"(LEX=N,POS=PPN,NUM=SNGT)<pun,0,0>;
11.	[ਪੁਅਲ]{}"Paul"(LEX=N,POS=NOU,NUM=SNGT)<pun,0,0>;
12.	[ਕਾਫੀ]{}"quite"(LEX=A,POS=SAV)<pun,0,0>;
13.	[ਸੋਹਣੀ]{}"beautiful" (LEX=J,POS=ADJ,GEN=FEM,NUM=SNG,PAR=M5)<pun,0,0>;
14.	[ਮਹਿੰਗੀ]{}"expensive"(LEX=J,POS=ADJ,NUM=SNG,PAR=M5)<eng,0,0>;
15.	[ਨਵੀਂ]{}"new" (LEX=J,GEN=FEM,NUM=SNG,POS=ADJ,PAR=M5)<pun,0,0>;
16.	[ਪਿਆਰ]{}"love" (LEX=V,POS=VER,TRA=TSTD,ATE=INF,PAR=M3)<pun,0,0>;
17.	[ਪਹੁੰਚ]{}"arrive"(LEX=V,POS=VER,TRA=TSTI,PAR=M3)<pun,0,0>;

Appendix B

Some of the T-rules for NLization

S. No.	T-rules
1.	mod(%a,N,@multal;%b,J):=((%b,+NUM=PLR)(" ")(%a,+NUM=PLR,-@multal),+@multal,+LEX=%a,+NUM=%a, +GEN=%a);
2.	mod(%a,N,@all;%b,J):=((%b)(" ")(%a,+NUM=PLR,-@all),+@all,+LEX=%a,+NUM=%a, +GEN=%a);
3.	(mod(%a;%b),SCOPE,%C):=(((%b)(" ")(%a)),+SCOPE=%C);
4.	(and(%a,N;%b,N),SCOPE,%C):=(((%b)(" ਅਤੇ ")(%a)),+SCOPE=%C);
5.	(%a,N,@multal):=("ਕਈ ")(%a,-@multal);
6.	(N,SNG,@multal,@extra,%a):=("ਬਹੁਤ ਹੀ ਜਿਆਦਾ")(" ")(%a,-@multal,-@extra,-NUM,+NUM=PLR);
7.	pos(%a,FEM,N;%b,@3,@pl,POD):=(%b,+PLR,+FEM)(" ਦੀ ")(%a);
8.	(N,SNG,FEM,@both,%a):=("ਦੋਵੇਂ")(" ")(%a,-@both,-NUM,+NUM=PLR);
9.	(N,@indef,%a):=(("ਇਕ")(" ")(%a,-@indef),%a,+NA,-@indef);
10.	(N,SNG,@every,%x):=("ਹਰੇਕ")(" ")(%x,-@every);
11.	(N,SNG,@other,%x):=("ਹੋਰ ਕੋਈ")(" ")(%x,-@other);
12.	(N,SNG,@less,%x):=("ਘੱਟ")(" ")(%x,-@less);
13.	(N,PLR,FEM,@both,%a):=("ਦੋਵੇਂ")(" ")(%a,-@both);
14.	(and(%a,N;%b,SCOPE),SCOPE,%C):=(((%b)(" , ")(%a)),+SCOPE=%C);
15.	(N,SNG,@no,%x):=(%x,+@any,-@no)(" ")("ਨਹੀਂ");
16.	(N,SNGT,@paucal,%a):=("ਥੋੜਾ ਕੁ")(" ")(%a,-@paucal);

17.	pos(%a,@all;%b,^PLR):=pos(%a;%b,-NUM,+NUM=PLR);
18.	mod(%a,N,@def;%b,J,@least):=((("ਸਭ ਤੋ ਘਟ")(" ")(%b)(" ")(%a)));
19.	(N,@proximal,%x):=("ਇਹ")(" ")(%x,-@proximal);
20.	pos(%a,N,FEM;%b,N):=(%b)(" ")("ਦੀ")(" ")(%a);
21.	plc(N,NOU,%a;N,NOU,^DONE,%b):=((%b)(" ਵਿਚ ")(%a),%f);
22.	(V,@reflexive,%x):=("ਆਪਣੇ ਆਪ ਨੂੰ")(" ")(%x,-@reflexive);
23.	mod(N,NOU,%a;%b,N,PPN,^DONE,SNGT):=((%b)(" ਦੀ ")(%a),%f);
24.	agt(%a,V,@future,@progressive,^PGS,^FUT,^SNG,^MCL;%b,@3,@male):=agt(%a,+PGS,+FUT,+MCL,-@future,-@progressive,+SNG;%b);
25.	agt(%a,V,@present,@perfect,^PER,^PRS,^SNG,^MCL;%b,@3,@male):=agt(%a,+PER,+PRS,+MCL,+SNG;%b);
26.	agt(%a,V,@future,^FUT,^SNG,^MCL;%b,R,@3,@male):=agt(%a,+FUT,+MCL,+SNG;%b);
27.	agt(%a,V,@possibility,^PBS,^SNG,^MCL;%b,@male):=agt(%a,+PBS,+MCL,+SNG;%b);
28.	(N,SNG,@no,%x):=(%x,@any,-@no)(" ")("ਨਹੀਂ")(%x,-@other);
29.	aoj(%a,J,^PLR;%b,R,@pl):=aoj(%a,-NUM,+NUM=PLR;%b,-@pl);
30.	(N,SNG,FEM,@all,%a):=("ਸਾਰੀਆਂ")(" ")(%a,-@all,-NUM,+NUM=PLR);
31.	([and]):=([ਅਤੇ]);
32.	([or]):=([ਜਾਂ]);
33.	([already]):=([ਪਹਿਲਾਂ ਹੀ]);
34.	([after]):=([ਤੋਂ ਬਾਅਦ]);
35.	([in]):=([ਵਿਚ]);

Appendix C

Some of the Inflection Paradigms

S. No.	Inflection Paradigms
1.	(%x,M1):=(%x,-M1,+FLX(SNG:=0>"";PLR:=0>"ਾਂ";));
2.	(%x,M2):=(%x,- M2,+FLX(SNG&MCL:=1>"ਾ";PLR&FEM:=0>"ਆਂ";PLR&NOM:=1>"ੇ";OB L:=3>"ੈਨੂੰ";EXP&PLR:=4>"ਅਸੀਂ";CNT:=2>"ਨੂੰ";)); for ਸਾਡੀ
3.	(%x,M3):=(%x,-M3,+FLX(IMT:=0>"ਣ ਹੀ ਵਾਲਾ ਹੈ"; PRS&SNG&MCL&^PGS&^TSTD&^HAB:=0>"ਦਾ ਹੈ"; PRS&SNG&MCL&^PGS&^TSTD&HAB:=0>"ਦਾ ਹੁੰਦਾ ਹੈ"; PRS&TSTD&^SNG&^PLR:=0>" ਕਰਦੇ ਹੋ"; PRS&TSTD&PLR&^EXP:=0>" ਕਰਦੇ ਹਾਂ"; PRS&TSTD&PLR&EXP:=0>" ਕਰਦੇ ਹਨ"; PRS&TSTD&SNG&MCL:=0>" ਕਰਦਾ ਹੈ"; PRS&TSTD&SNG&FEM:=0>" ਕਰਦੀ ਹੈ"; PAS&NOT&SNG&MCL&^TSTD:=1>"ਚਿਆ"; PAS&TSTD&^EMARK:=0>" ਦਿੱਤਾ";));
4.	(%x,M4):=(%x,-M4,+FLX(SNG:=0>"";PLR&^HAR&^CNT:=0>"ਆਂ"; HAR:=4>"ਅਸੀਂ";CNT:=2>"ਨੂੰ";));
5.	(%x,M5):=(%x,-M5,+FLX(SNG:=0>"";PLR:=0>"ਆਂ";MCL:=1>"ਾ";));

6.	(%x,M6):=(%x,-M6,+FLX(SNG:=0>"";PLR:=0>"ਆਂ";MCL:=2>"ਾਂ";));
7.	(%x,M7):=(%x, - M7,+FLX(TSTI:=1>"ਹ";SNG&OBL&^TST:=0>"ਨੂੰ"; SNG&TSTD:=0>"ਨੇ";SNG&AOJ&FEM&^PLR&^DET:=0>"ਦੀ"; AOJ&PLR&HAR:=0>"ਦੇ";AOJ&PLR&FEM:=0>" ਦੀਆਂ"; AOJ&FEM&DET:=0>"ਨਾਂ ਦੀ";POS&PLR&FEM&^AOJ:=0>"ਨਾਂ ਦੀਆਂ"));
8.	(%x,M8):=(%x, -M8,+FLX(AOJ&FEM&PLR:=1>"ਸ ਦੀਆਂ"; CNT&SNG:=1>"ਸ ਨੂੰ";));
9.	(%x,M10):=(%x, -M10,+FLX(OBL:=3>"ੈਨੂੰ";));
10.	({N|V|D|J|R},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);

References

[1] J. Ollie, Machine translation [Online]. Available: http://en.wikipedia.org/wiki/Machine_translation [Accessed 26 March 2013].

[2] "Universal Networking Language (UNL): History of UNL," [Online]. Available: http://www.unlweb.net/unlweb/index.php?option=com_content&view=article&id=58:unl&catid=54:unlweb [Accessed 26 March 2013].

[3] S. Alansary *et al.*, "A Suite of Tools for Arabic Natural Language Processing: A UNL Approach," in *IEEE Transactions*, vol. 33, 2013 pp. 56-99.

[4] "Universal Networking Language (UNL): Universal Word," [Online]. Available: http://www.unlweb.net/wiki/UW [Accessed 26 March 2013].

[5] "Universal Networking Language (UNL): Relations," [Online]. Available: http://www.unlweb.net/wiki/Relations [Accessed 26 March 2013].

[6] "Universal Networking Language (UNL): Attributes," [Online]. Available: http://www.unlweb.net/wiki/Attributes [Accessed 26 March 2013].

[7] P. Kumar, R. K. Sharma, "UNL Based Machine Translation System for Punjabi Language," PhD thesis, Thapar University, Patiala, India, 2012.

[8] H. Uchida, "Universal Networking Language (UNL) Specifications 2005," UNDL Foundation, Geneva, Switzerland, Tech. Rep. version 2005, JAN. 2005.

[9] "Universal Networking Language (UNL): UNL Knowledge Base," [Online]. Available: http://www.unlweb.net/wiki/UNL_Knowledge_Base [Accessed 26 March 2013].

[10] M. Nagi, T. D. Senta, "Sharing knowledge across language barriers: a universal approach for online books," in *Proc. ACM workshop on Research advances in large digital book repositories*, California, USA, 2008, pp. 49-52.

[11] "Universal Networking Language (UNL): Tools," [Online]. Available: http://www.unlweb.net/wiki/Tools [Accessed 27 March 2013].

[12] Internet World Stats. "Internet World Users by Language," [Online]. Available: http://www.internetworldstats.com/stats7.htm [Accessed 27 March 2013].

[13] "CQ Researcher: Future of language," [Online]. Available: http://library.cqpress.com.offcampus.lib.washington.edu/cqresearcher/cqresrre2000111700 [Accessed 15 May 2013].

[14] "The Evolution of Machine Translation: Supervening Social Necessity, Hype Cycle, Disruptive Innovation," [Online]. Available: http://comdj.wordpress.com/research/the-evolution-of-machine-translation-supervening-social-necessity-hype-cycle-disruptive-innovation/introduction/ [Accessed 15 May 2013].

[15] "Systran Language Translation Softwares. Machine Translation," [Online]. Available: http://www.systransoft.com/systran/corporate- profile/ translation-technology /what-is-machine-translation [Accessed 15 May 2013].

[16] S. Reshef, W. Shuly, "Transfer-based Machine Translation between morphologically-rich and resource-poor languages: The case of Hebrew and Arabic," MA thesis, Tel Aviv University, Israel, 2011.

[17] "An Overview of Automated Machine Translation – The Ruled-Based Approach," [Online]. Available: http://kuiwon.wordpress.com/ 2013/ 05/24/an-overview-of-automated-machine-translation-the-ruled-based-approach/ [Accessed 15 May 2013].

[18] A. Okumura, E. Hovy, "Building Japanese-English Dictionary based on Ontology for Machine Translation," in *Proc. of the ARPA Human Language Technology Conference* Princeton, NJ, 1994.

[19] K. Tretyakov, "Example-Based Machine Translation of Short Phrases Using the Context Equivalence Principle," University of Tartu, Estonia, Tech. Rep. 2005.

[20] S. Hunsicker *et al.*, "Machine Learning for Hybrid Machine Translation," in *Proc. of the 7th Workshop on Statistical Machine Translation,* Montreal, Canada, 2012, pp. 312–316.

[21] G. Anthes, "Automated translation of Indian languages." in *Communications of the ACM*, New York, USA, 2010, pp. 24-26.

[22] Srikanth, D. Kapoor, "Machine translation set for quantum leap in India," *Express Computer,* 2001.

[23] "History of UNL Programme," [Online]. Available: http://www.undl.org/index.php? option=com_content&view=article&id=47&Itemid=67&lang=en#History [Accessed 17 March 2013].

[24] "Universal Networking Language," [Online]. Available: https://www.princeton.edu /~achaney/tmve/wiki100k/docs/Universal_Networking_Language.html [Accessed 17 March 2013].

[25] "UNL Development Set," [Online]. Available: http://www.undl.org/unlsys/ds.html# howToStartOnEnconverting [Accessed 17 March 2013].

[26] "Universal Networking Language (UNL): EUGENE Specifications," [Online]. Available: http://www.unlweb.net/wiki/EUGENE [Accessed 12 June 2013].

[27] Lewis, M. Paul, "*Ethnologue: Languages of the World*", 16th ed, Dallas: SIL International, 2009.

[28] "Universal Networking Language (UNL): UNL-Memory," [Online]. Available: http://www.unlweb.net/wiki/UNL_Memory [Accessed 28 May 2013].

[29] "Universal Networking Language (UNL): UNL Ontology" [Online]. Available: http://www.unlweb.net/wiki/UNL_Ontology [Accessed 28 May 2013].

[30] “Universal Networking Language (UNL): Training and Research,” [Online]. Available: http://www.unlweb.net/unlweb/index.php?option=com_content&view=article&id=65:road-map&catid=1:latest-news&Itemid=60 [Accessed 28 May 2013].

[31] “Universal Networking Language (UNL): UNL-NL Dictionary,” [Online]. Available: http://www.unlweb.net/wiki/UNL-NL_Dictionary [Accessed 28 May 2013].

[32] “Universal Networking Language (UNL): DeConverter,” [Online]. Available: http://www.undl.org/index.php?option=com_content&view=article&id=49&Itemid=70&lang=en#DeConverter [Accessed 1 June 2013].

[33] “Universal Networking Language (UNL): UNL-arium,” [Online]. Available: http://www.unlweb.net/unlarium/index.php?grammar=fmeasure [Accessed 19 June 2013].

[34] “Universal Networking Language (UNL): F-measure,” [Online]. Available: http://www.unlweb.net/wiki/F-measure [Accessed 19 June 2013].

Certifications

- CUP500 - Certificate of Proficiency in UNL

UNDL FOUNDATION

Universal Networking Language

CUP 500

Certificate of Proficiency in UNL

The UNDL Foundation certifies that

Harinder Singh

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

Research Publications

Research Paper Published

- Harinder Singh, Parteek Kumar, "Analysis of Noun, Pronoun and Adjective Morphology for NLization of Punjabi with EUGENE" in *International Journal of Computational Linguistics and Natural Language Processing (IJCLNLP),* vol. 2 Issue 7, July 2013, pp. 436-442.

Research Paper Communicated

- H. Singh, P. Kumar, "Implementation of NLization framework for Verbs, Pronouns and Determiners with EUGENE," in Inderscience Publishers of distinguished academic, scientific and professional journals.