UNL-ization of Punjabi with IAN

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

Master of Engineering
in
Software Engineering

Submitted By
Vaibhav Agarwal
(801131028)

Under the supervision of:
Dr. Parteek Bhatia
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2013

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Unl-ization of Punjabi with IAN*", 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.

(Vaibhav Agarwal)

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

(Dr. Parteek Bhatia)
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

First of all, I am thankful to God for his blessings and showing me the right direction. With his mercy, it has been made possible for me to reach so far.

It gives me great pleasure to express my gratitude towards the guidance and help I have received from Dr. Parteek Bhatia, Assistant Professor, Computer Science and Engineering Department, Thapar University, Patiala. I am thankful for his continual support, encouragement, and invaluable suggestion. It has been a great honor to work under him.

I am thankful to Dr. Maninder Singh, Head, Computer Science and Engineering Department, Thapar University, Patiala, who has been a constant source of inspiration for me throughout this work. I express my gratitude to all other staff members of Computer Science and Engineering Department for providing me all the facilities required for the completion of my thesis work.

I am also thankful to my friends Ashutosh Verma, and Harinder Singh for boosting me with their support and confidence.

I am forever indebted to my parents for their constant encouragement and support throughout the past years.

Vaibhav Agarwal

(801131028)

Abstract

In the field of Natural Language Processing, Universal Networking Language (*i.e.*, UNL) has been an area of immense interest among researchers during last couple of years. Universal Networking Language is an artificial language used for representing information in a natural-language-independent format. This thesis presents UNL-ization of Punjabi sentences with the help of different example sentences, using IAN (Interactive Analyzer) tool. In UNL approach, UNL-ization is a process of converting natural language resource to UNL and NL-ization, is a process of generating a natural language resource out of a UNL graph. IAN is an online tool provided by UNDL Foundation.

IAN processes input sentences with the help of TRules and Dictionary entries. The proposed system performs the UNL-ization of all the major Part-of-Speech, numbers and ordinals up to fourteen digits, written in words in Punjabi language, with the help of Analysis Dictionary and TRules. The system is tested on three different corpuses *i.e.*, CORPUS500, UCA1, and UCA2 provided by UNDL Foundation. A detailed description of IAN tool has also been explained.

F-Measure of the proposed system has also been calculated. F-Measure is the score used to calculate the grammar's accuracy. F-Measure of CORPUS500, UCA1, and UCA2 has been calculated and detailed description is given in Chapter 6, *i.e.,* Results and Discussion. F-Measure considers both the precision and recall of the grammar to compute the score. Its value lies between 0 to 1.

Table of Content

List of Figures

List of Tables

Chapter 1
Introduction

Machine Translation (MT) has been an area of immense interest among researchers during last couple of decades. Universal Networking Language (UNL) based MT is also an effort in this direction. The **Universal Networking Language** (UNL) is an artificial language for representing, describing, summarizing, and storing information in a natural-language-independent format. In the UNL approach, there are two basic movements: UNL-ization and NL-ization. UNL-ization is the process of representing/mapping/analyzing the information conveyed by natural language utterances into UNL; NL-ization, conversely, is the process of realizing/manifesting/generating a natural language document out of a UNL graph [47]. These processes are completely independent.

Currently, the main goal of the UNL-ization process has been to convert the given information text into the intermediate language, *i.e.*, UNL which is of course language-independent and machine-tractable database, with the help of many relations, and attributes.

Suppose there are *n* numbers of different natural languages. Now using the approach of UNL for converting those *n* natural languages into each other, *2*n* number of possible translations or mappings that needs to be done. This is because now only *2* conversions needs to be done for that particular natural language , means from that natural language to UNL and then from UNL to that natural language. Had this approach been not followed, the total number of conversions in converting every natural language to every other natural language would have been *n*(n-1)* as every language needs to be converted into the other *n-1* languages. Such a system for Punjabi language will certainly be very helpful for more than 91 million Punjabi language users [29]. The tool and methodology used for UNL-ization has been explained in chapter 4 and chapter 5. Figure 1.1 depicts the approach of UNL-ization and NL-ization process. Both these processes are completely independent.

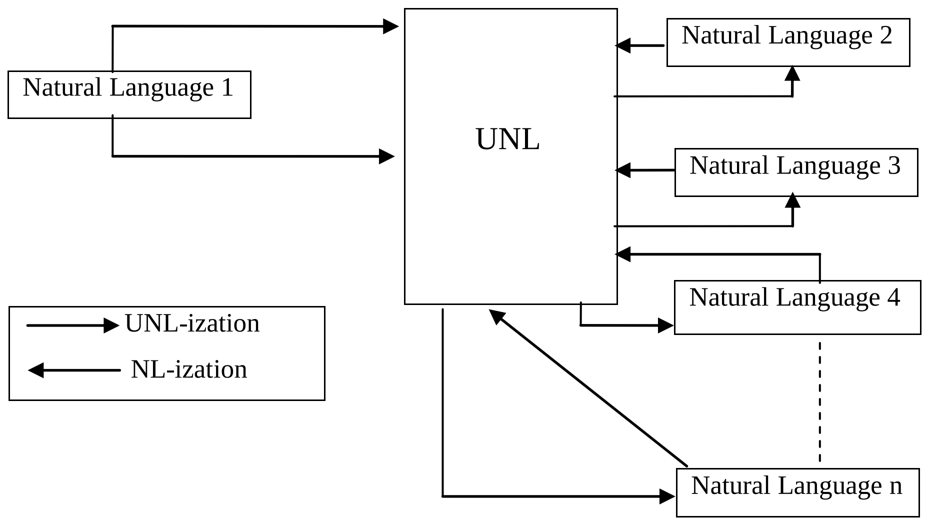

Figure 1.1: Approach for UNL-ization and NL-ization of *n* natural languages

1.1 History

The UNL Programme started in 1996, as an initiative of the Institute of Advanced Studies of the United Nations University in Tokyo, Japan [14]. 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 Programme. The Foundation, a non-profit international organisation, 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 Programme so that it can fulfill its mission. Its headquarters are based in Geneva, Switzerland. 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 several specialists in computer science and linguistics, who are at work creating the linguistic resources and developing the web structure of the UNL System. The UNDL Foundation provides technological support and co-ordinates the implementation of the Programme. The Programme 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 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.

1.2 Scope of UNL

UNL has been exploited for several different tasks in natural language engineering, such as multilingual document generation, summarization, text simplification, information retrieval and semantic reasoning. UNL is not a human language. It is not at all expected that one should speak UNL or should communicate in UNL, rather it is expected that people should use UNL and should communicate through UNL in the same invisible and spontaneous way they do with other declarative and procedural languages which are pervasive in everyday applications. As no one is required to know HTML to browse the Internet or even to create websites, similarly everyone would be able to UNL-ize documents and to extract out of them the information needed without any knowledge of UNL. UNL is therefore a formal language designed for computers, not for humans. Like other logical systems, it seeks to provide the linguistic and semiotic infrastructure for computers (and not for humans) to handle what is meant by natural languages. Indeed, it is important to note that at this point in time it would be foolish to state it is possible to represent the "full" meaning of any word, sentence or text for any language [47].

The UNL avoids the pitfalls of trying to represent the "full meaning" of sentences or texts, targeting instead the "core" or "consensual" meaning that is most often attributed to them. So poetry, metaphor, figurative language, and other complex, indirect communicative behaviours is beyond the current scope and goals of the UNL. Instead, the UNL targets direct communicative behaviour and literal meanings as a tangible, concrete basis for much or most of human communication in practical, day-to-day settings. This is the main reason why UNL has not been exactly a interlingua-based machine translation project, even though machine translation is one of the possible and more obvious and promising uses of UNL. Currently, the main goal of the UNL-ization process has been to map the information that is verbally elicited in the surface structure of written texts into a language-independent and machine-tractable database. This means that the UNL representation has not been committed to replicate the lexical and the syntactic choices of the original, but focuses in representing, in a non-ambiguous format, one of its possible readings, preferably the most conventional one. In this sense, the UNL representation has been an interpretation rather than a translation of a given text [47].

1.3 Building Blocks of UNL

UNL represents the information in three different types of semantic units, namely, Universal Words (UWs), Relations and Attributes. UNL represents information sentence by sentence. Each sentence is converted into a hyper-graph (also known as UNL graph) having concepts represented as nodes and relations as directed arcs. The concepts are represented by UWs and UNL relations are used to specify the role of each word in a sentence. The subjective meanings intended by the author are expressed through UNL attributes. UNDL Foundation has formally defined the specifications of UNL.

1.3.1 Universal Words

Universal Words form the vocabulary of UNL. These words correspond to the nodes that are interlinked by relations or modified by attributes in a UNL graph. The concepts of UWs are divided into four categories, namely, nominal concepts, verbal concepts, adjective concepts and adverbial concepts [7]. A UW is a character string (an English language word) followed by a list of constraints. UW's can be temporary or permanent [52].

1.3.1.1 Temporary UW's

Temporary UW's are words that represent concepts or entities that are still in process of lexicalization ("googlers", "twittered"); are too specific to be included in the UNL Dictionary ("Universal Networking Digital Language Foundation", "Léon Werth"); or are not translatable ("3.14159", "H_2O", "www.undlfoundation.org").

1.3.1.2 Permanent UW's

Permanent UW's are included in the UNL Dictionary and correspond to concepts that have been already lexicalized in at least one language. They can be simple, compound or complex.

1.3.2 Attributes

Attributes are annotations made to nodes or hypernodes of a UNL hypergraph. They denote the circumstances under which these nodes (or hypernodes) are used. The syntax of attributes is given in (1.1) [48].

<attribute> ::= "@"<attribute name>

<attribute name> ::= <character>+

<character> ::={"a",…,"z" , "_" } …(1.1)

where:

< > variable

" " terminal symbol

::=... is defined as ...

{ } disjunction ("or")

+ to be used one or more times

... to be repeated more than 0 times

Attribute names are always lower case words or expressions. Normally, English words ("past", "will") or mnemonic abbreviations ("def", "pl") are used for attribute labelling. No blank space is allowed inside an attribute name. Attributes may convey three different kinds of information [48]. These are given below.

- The information on the role of the node in the UNL graph (as in the case for '@entry', that indicates the main (starting) node of a UNL directed graph).
- The information conveyed by bound morphemes and closed classes, such as affixes (gender, number, tense, aspect, mood, voice, *etc*), determiners (articles and demonstratives), adpositions (prepositions, postpositions and circumpositions), conjunctions, auxiliary and quasi-auxiliary verbs (auxiliaries, modals, coverbs, preverbs) and degree adverbs (specifiers), for example '@under', '@past' *etc*.
- The information on the (external) context of the utterance, *i.e.*, non-verbal elements of communication, such as prosody, sentence and text structure, politeness, schemes, and speech acts, for example '@anger', '@surprise' *etc*.

1.3.3 Relations

Relations, formerly known as "links", are labelled 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 [39]. The repertoire of relations is defined in the UNL specifications and it is not open to frequent additions. In the UNL framework, relations describe semantic functions between two UWs. These functions are binary and directed (from a source to a target) and are claimed to be universal. Because of their similarity in name and function to syntactic relations, it may seem that the labels used for relations are different names for special grammatical functions. This is emphatically not the case. The intention is that the labels used denote specific ideas rather than grammatical structures; the idea of “something that initiates an event,” or “agent” for example, is quite different from “grammatical subject of a sentence”, even though many times the subject of a sentence will indicate the agent of the event [39]. The agent of an event may also appear as an adjective or noun modifier, with the preposition “by” or embedded in nouns with “er” suffixes.

The whole point of the conceptual relations is to have a name for these very different grammatical structures which are conceptually quite the same. Thus, the conceptual relations used in UNL are much more abstract than the grammatical relations found in sentences [39]. The relations between the two nodes is of the form: rel_name (node1,+att1;node2,+att2); where rel_name is the name of the relation , node1 and node2 are the two nodes between which the relations hold and att1 , att2 are the attributes which are added to nodes1 and nodes2 respectively. It should be kept in mind that any number of additional attributes can be added. For example UNL representation of English sentence ‘book about john’ is ‘cnt(book, john@about)’ where, ‘cnt’ is the relation name whose first argument is ‘book’ while second argument is ‘john@about’.

Chapter 2
Literature Review

There are more than 91 million native speakers of Punjabi language, which makes it approximately the 12th most widely spoken language in the world [29]. Gill has explained the features of Punjabi language [30]. Punjabi has word classes in the form of noun, pronoun, adjective, cardinal, ordinal, main verb, auxiliary verb, adverb, postposition, conjunction, interjection and particle. Punjabi phrases can be broadly classified into two types, namely, nominal phrases (built using the words of various word classes like noun, pronoun, adjective *etc.*) and verb phrases (built using primarily the words of main verb and auxiliary verb word classes).

2.1 Research activities in Universal Networking Language

Research on UNL has three distinct divisions [35]. These divisions are: development of EnConversion and DeConversion modules; applications of UNL in other contexts such as knowledge representation and knowledge management, multilingual search engines, language-independent Universal Digital Library *etc.*; and use of external lexical and ontological resources to enhance the processes of UNL [20].

2.2 Development of Enconversion and Deconversion Modules

There are three different approaches that have been used to design and develop the tools for EnConversion and DeConversion processes [35]. In the first approach, one uses a common engine like *'EnCo'* and *'DeCo'* tools provided by the UNDL center to accomplish the task [53]. The other approach is integrative approach that is based on the integration of UNL into pre-existing MT systems. Third approach is followed by researchers who have noticed the drawbacks in the tools provided by the UNDL center, and have created new architectures from scratch. Martins *et al.* have proposed a prototype system for converting Brazilian Portuguese into the UNL and DeConverting UNL expressions into Brazilian Portuguese with *'EnCo'* and *'DeCo'* tools respectively [43]. Their system consists of three important sub-modules, namely, the lexical, the syntactic and the semantic modules. Sérasset and Boitet have viewed UNL as the future 'html of the linguistic content' [13]. The French DeConverter proposed by them is based on the modular architecture. It splits the DeConversion process into transfer and generation steps. Multilingual information processing through UNL has been proposed by Bhattacharyya [34]. Their system performs sentence level encoding of English, Hindi and Marathi into the UNL form and then decodes this information into Hindi and Marathi. As such, the system has created a way of semiautomated translation from English to Hindi and Marathi and also between Hindi and Marathi. Martins *et al.* have analyzed unique features of UNL taking inferences from Brazilian Portuguese-UNL EnConverting task [42]. They have suggested that UNL should not be treated as an interlingua, but as a source and a target language owing to flexibility that EnConversion process brings to UNL making this just like any other natural language. Dhanabalan *et al.* have proposed an EnConversion tool from Tamil [50]. Their system uses existing morphological analyzer of Tamil to obtain the morphological features of the input sentence. They have also employed a specially designed parser in order to perform syntactic functional grouping. The whole EnConversion process has been driven by the EnConversion rules written for Tamil language. Martins *et al.* have noted that the *'EnCo'* and Universal Parser tools provided by UNDL foundation require inputs from a human expert who is seldom available and as such their performance is not quite adequate [44]. They have proposed the *'HERMETO'* system which converts English and Brazilian Portuguese into UNL. This system has an interface with debugging and editing facilities along with its high level syntactic and semantic grammar that make it more user friendly. Mohanty *et al.* have used Semantically Relatable Sequence (SRS) based approach for developing a UNL based MT system [41]. They have analyzed the source language using semantic graphs and used these graphs to generate target language text. Dey and Bhattacharyya have presented the computational analysis of complex case structure of Bengali for a UNL based MT System [21]. They provided the details of the rule theory of *'EnCo'* and *'DeCo'* tools which are driven by analysis rules and generation rules respectively for Bengali language. These rules are based on case structure of '*kaaraks'* used in the Bengali language. Mohanty *et al.* have investigated the problem of prepositional phrase (PP) attachment in the context of UNL based MT systems for English language [40]. They have performed the linguistic analysis of six common prepositions in English, namely*, 'for', 'from', 'in', 'on', 'to'* and *'with'*. The insights obtained from this analysis have been used to enrich a lexicon and a rule base, for the *'EnCo'* tool. Blanc has performed the integration of *'Ariane-G5'* to the proposed French EnConverter and French DeConverter [11]. *'Ariane-G5'* is a generator of MT systems. In the proposed system, EnConversion takes place in two steps; first step is analysis of the French text to produce the representation of its meaning in the form of a dependency tree and second step is lexical and structural transfer from the dependency tree to an equivalent UNL graph. Boguslavsky *et al.* have proposed a multi-functional linguistic processor, *'ETAP-3'*, as an extension of *'ETAP'* machine translation system to a UNL based machine translation system [16]. Choudhury *et al.* have proposed a framework for converting Bangla to UNL and have also proposed a procedure to construct Bangla to UNL dictionary [25]. The system developed by Lafourcade uses ant colony algorithm for semantic analysis and fuzzy UNL graphs for EnConversion process [28]. Nguyen and Ishizuka have trained UNL relations classifier using statistical techniques based feature extractor indicating the usage of statistical approach [38]. Lexicalized probabilistic parser has been employed by Jain and Damani for English to UNL conversion. The parser is used to create typed dependency tree and phase structure tree for a given English sentence [27]. Arabic MT System based on UNL has also been developed and successfully tested. In this system, Arabic generation grammar is created for the tools for MT system [45][32]. Mridha *et al.* have proposed a Bangla EnConversion system [26]. They have analyzed Bangla words morphologically in order to obtain their roots and primary suffixes.

2.3 Application of UNL in other contexts

Mukerjee *et al.* used UNL as a tool for language-independent semantics for Question Answering system [2]. This concept has further been explored by Shukla *et al.* suggesting two major tasks in the implementation of proposed Question Answering system [36]. The first task is document processing that converts the document into equivalent UNL expression. For this, the document is processed through a UNL Parser to get the complete set of semantic predicates for the document. The other task is query processing and answer extraction. It involves the tasks of question analysis and processing to generate the answer template; conversion of Natural Language (NL) answer template to the NL semantic predicates; conversion of NL semantic predicates to interlingua; search engine and the conversion of the answer from the interlingua to NL through the DeConversion process. Bértoli *et al.* have proposed *'CELTA'* showcase as a web platform for using UNL [23]. They demonstrated a multilingual business-to-business platform using UNL. Surve *et al*. have proposed an *'Agro-Explorer'* as a meaning based, interlingua search engine designed specifically for the agricultural domain covering English, Hindi and Marath languages [31]. The system involves the use of *'EnCo'* tool for the EnConversion of English query to UNL. The query in the UNL expression searches the UNL corpus of all documents. When a match is found, it sends the corresponding UNL file to DeConverter to provide the contents in the native language. Ramamritham *et al.* have further improved *'Agro-Explorer'* to develop *'aAQUA'* an online multilingual, multimedia agricultural portal for disseminating information from and to rural communities [22]. Choudhary and Bhattacharyya have performed the text clustering using UNL representation. They have generated feature vectors using UNL [3]. The clustering method used in the proposed system is the Self Organizing Maps (SOMs). They have inferred that UNL method for feature vector generation performed better than frequency based methods. Jiang *et al.* have explored UNL as a facilitator for communication between languages and cultures [6]. They designed a system to solve critical problems emerging from current globalization trends of markets and geopolitical interdependence. It facilitates the participations of people from various linguistic and cultural backgrounds to construct UNL knowledge bases in a distributed environment. Karande has proposed a multilingual search engine with the use of UNL [19]. The proposed system requires EnConverter to convert the contents of source language to UNL. Boitet *et al.* have built Hindi-French-English-UNL resources for *'SurviTra'* (Survival Translator) project [5]. It is a web service, initially being developed to help a French visitor needing to communicate with an Indian helper when English is not an option. It is a bilingual chat web service equipped with a phrasebook and a dictionary. Avetisyan and Avetisyan have proposed a *'LOOK4'* system for enhancement of web search results with Universal Words (UWs) [1]. They have used some existing resources such as online services provided by major search engines, semantic ontology of WordNet and flexibility of UNL for representation of semantic relations among concepts.

2.4 Use of external resources to enhance UNL processes

The researches have been working on the use of lexical and ontological resources for the improvement of the UNL processes. In this section, we present different activities in this direction. Sornlertlamvanich *et al.* have performed the Thai lexical semantic annotation by UWs [55]. The process proposed by them consists of word extraction, word sense disambiguation, and UW annotation. They have also proposed a computable method to find the appropriate UW to annotate a Thai word. Iraola has used WordNet for linking UWs of Spanish-UNL dictionary [24]. The proposed system has enriched UWs with semantic information. Verma and Bhattacharyya have performed automatic generation of multilingual lexicon for English, Hindi and Marathi by using English, Hindi and Marathi WordNets [33]. In addition to the WordNet, the system also makes use of word sense disambiguator, an inferencer and the knowledge base of the UNL. Ribeiro *et al.* have proposed the development of a Portuguese-UNL dictionary using the WordNet.PT, a lexical database developed under the EuroWordNet framework [8]. This system has explored the similarity between the WordNet.PT's lexical semantic relations and the UNL. Bueno *et al.* have worked on the systematization of knowledge acquisition process for its use in intelligent management systems [49]. Their efforts resulted into a knowledge engineering suite to support the construction of ontologies. Bekios *et al.* have proposed a strategy for building UW dictionary with the use of WordNet [18]. They have proposed six rules for the creation of semantic restrictions of UWs. Each Wordnet word, *i.e.*, the set of senses for the word and its lexical relations, is used as 45 input to the rule base and the system yields a semantic restriction suitable for the UW that is being created.

Rouquet and Nguyen have proposed an interlingual annotation of texts [10]. They have explored the ways to enable multimodal multilingual search in large collections of images accompanied by texts. They have used the domain ontology and UNL UWs as the interlingual lexemes for annotation. Boudhh and Bhattacharyya have proposed the unification of UW dictionaries by using WordNet ontology [46]. They have used the WordNet ontology and proposed an extension of UW dictionary in the form of U++ UW dictionary. They have used the concept of similarity measures to recognize the semantically similar context.

Chapter 3
Problem Statement

This chapter gives the details of gaps and the work that has not been done yet. Later objectives of this thesis and methodology to achieve those objectives are described.

3.1 Gaps in previous work

Punjabi language is world's 12th most widely spoken language [29]. Punjabi language is used in both parts of Punjab, in India and also in Pakistan. There are relatively less efforts in the field of computerization and development of this language. IAN, *i.e.,* Interactive Analyzer is an online tool developed by UNDL Foundation and released in 2012 for UNL-ization and no work has yet been carried out. Other tools that were used before have now become obsolete because they were very complex and their inability to be integrated on a central site. IAN has been accepted globally as a standardized tool for UNL-ization.

3.2 Objectives

The main objectives of this work are:

- To study UNL-ization and IAN (*i.e.*, *I*nteractive *A*nalyzer) Framework.
- To manually translate CORPUS500, UC-A1, and UC-A2 provided by UNDL Foundation to our local Natural language, *i.e.*, Punjabi, to be used as an input for IAN tool.
- To create Analysis Dictionary for UNL-ization of CORPUS500, UC-A1, and UC-A2.
- To create TRules for UNL-ization of CORPUS500, UC-A1, and UC-A2.
- To achieve F-Measure of 0.8 or above for UNL-ization of CORPUS500, UC-A1, and UC-A2.

3.3 Methodology

To study the objectives discussed in section 3.2, a step-by-step methodology has been followed. The detail of this is given below.

- CUP500 Certificate of Proficiency in UNL through VALERIE has been cleared so as to understand deeply the concepts of UNL-ization.
- Manual translation of CORPUS500, UC-A1, and UC-A2 into Punjabi Natural Language, which is used as an input to IAN tool, has been carried out.

- For each of the Natural Language word appearing in the resource, its corresponding dictionary entry has been made.
- TRules have been created for carrying out UNL-ization.
- F-Measure, which is the measure of accuracy of the grammar, has been calculated for CORPUS500, UC-A1, and UC-A2.

Chapter 4
IAN Framework

IAN, *i.e., Interactive Analyzer* is an online tool developed by UNDL Foundation for UNL-ization. As a universal engine, IAN must be parameterized to the source languages with the following files, to be provided through IAN's interface. These are:-

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

4.1 UNL-ization with IAN

IAN performs the following three movements over the input file [17]. First is Segmentation, *i.e.*, the division of the input document into a series of processing units (sentences), which are processed one at a time. Second movement is Tokenization, *i.e.*, the identification of the tokens (lexical items) of each sentence of the input document, and the third movement is 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.

4.2 Introduction to IAN tool

IAN tool is fully automatic tool. It takes Natural Language document as input and delivers an output in UNL without any human intervention. IAN has 6 tabs, *i.e.*, Welcome, NL Input, Dictionaries, TRules, DRules, and IAN console. First is the W**elcome** tab**,** which gives a brief introduction about the tool. Figure 4.1 shows the Welcome tab.

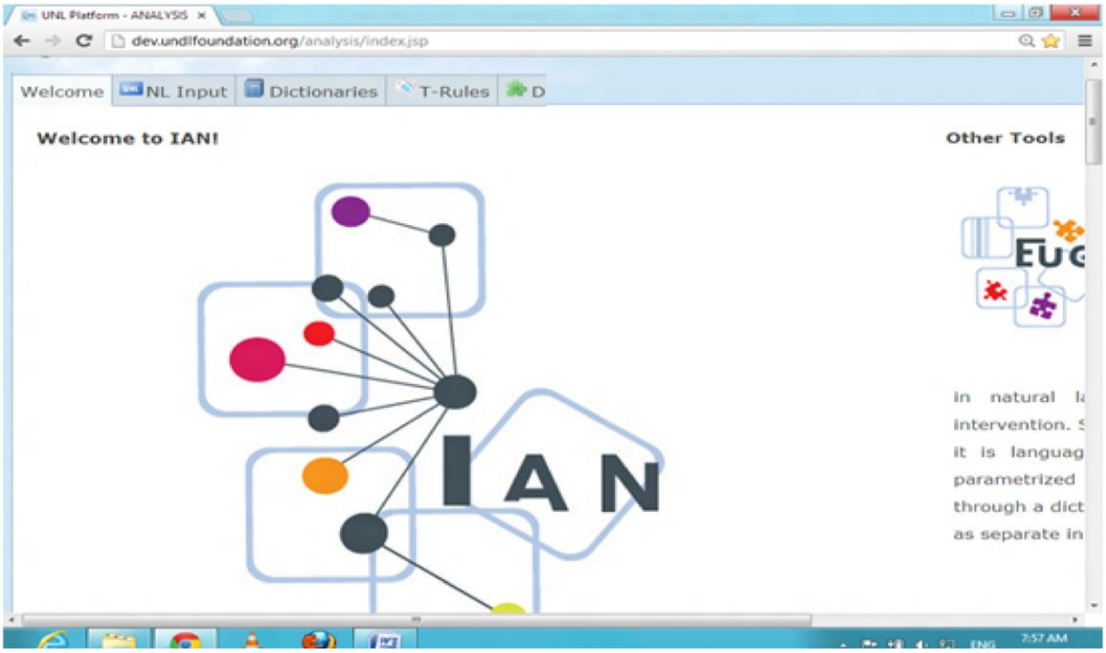

Figure 4.1: Snapshot of Welcome tab [15]

Second is the **NL Input** tab, where the natural language document to be UNL-ized is provided. Figure 4.2 shows the NL input process in the input tab.

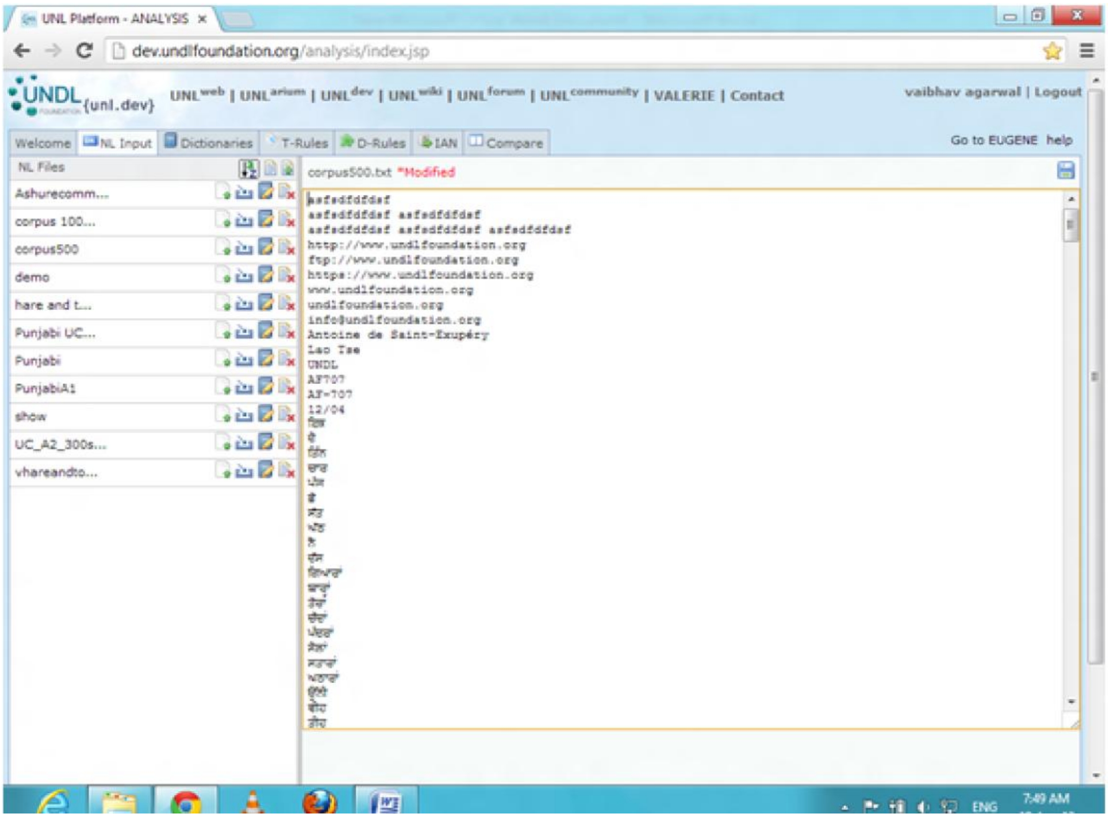

Figure 4.2: Snapshot of NL Input tab [15]

Third tab is the **Dictionaries**, where NL-UNL dictonaries (*i.e.*, the dictionaries to be used in natural language analysis) is provided. There can be several different dictionaries, and can be loaded 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). Reordering the dictionaries through the option "reorder dictionaries" at the top menu can be done. Figure 4.3 shows how to create dictionary entry.

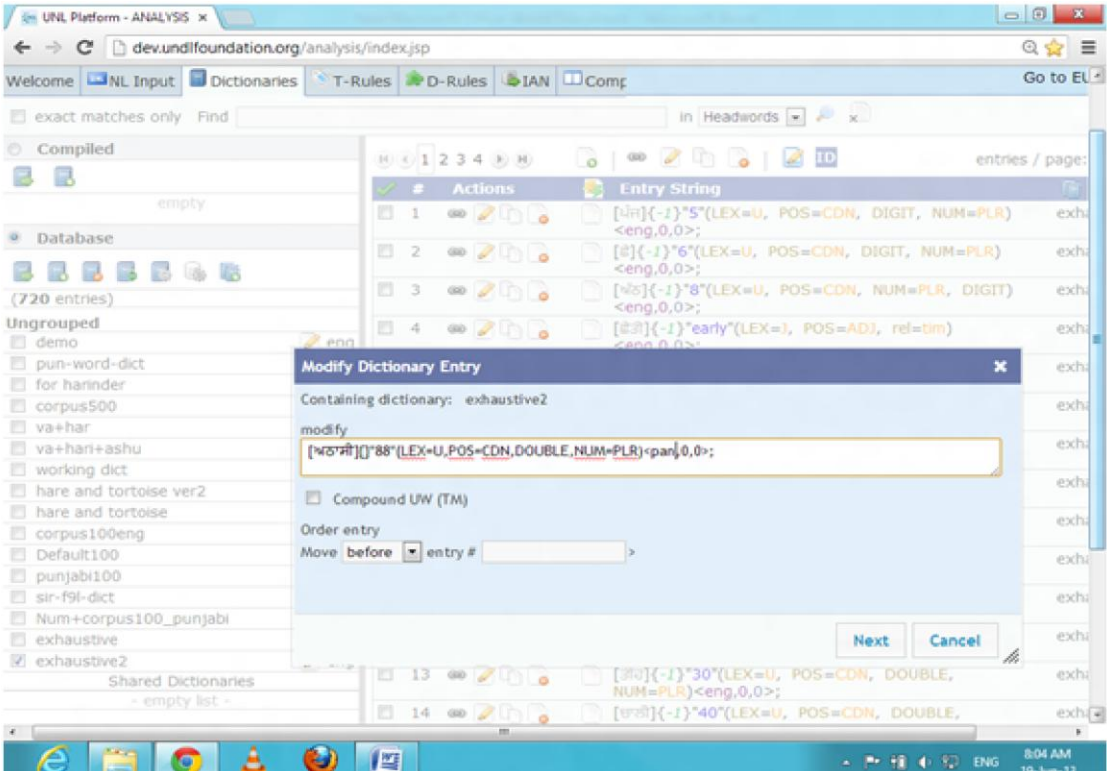

Figure 4.3: Snapshot showing creation of Dictionary under Dictionay tab [15]

Dictionaries are plain text files with a single entry per line in the following format [9].

[NLW]{ID}"UW"(ATTR)<FLG,FRE,PRI> ; COMMENTS

Here, NLW is the natural language word, UW is the corresponding Universal Word, ATTR, the list of features of the NLW, FLG, The three-character language code according to ISO 639-3, FRE, the frequency of NLW in natural texts. Used for natural language analysis (NL-UNL). It can range from 0 (less frequent) to 255 (most frequent). PRI, the priority of the NLW used for natural language generation (UNL-NL). It can range from 0 to 255.

Fourth tab is **T-Rules**, where NL-UNL transformation grammar (*i.e.*, the grammar to be used to process the natural language input) is provided. Figure 4.4 shows the snapshot of TRule tab.

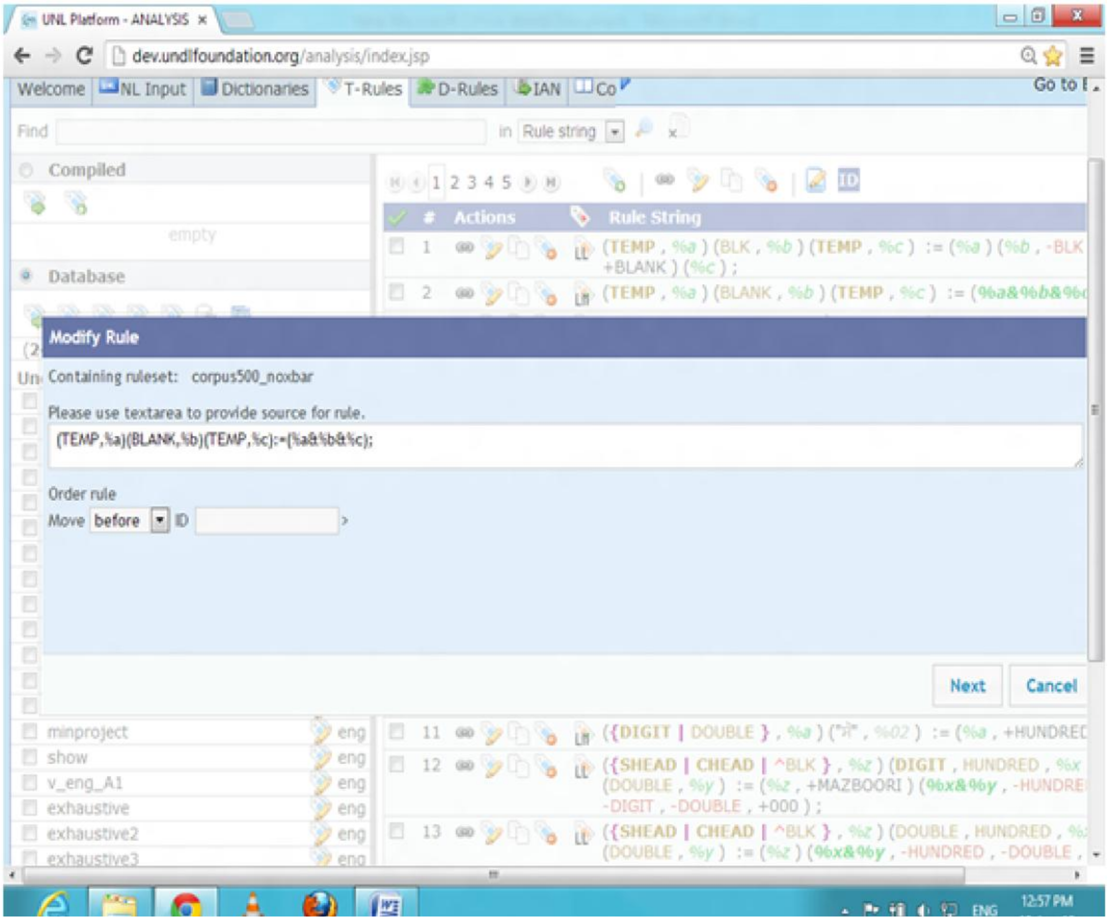

Figure 4.4: Snapshot showing creation of TRule in TRule tab [15]

T-rules may be classified according to the type of modification that they promote [51].

a.) **LL, or list-to-list**, where the initial state and the final state are list structures. There are 5 different subtypes of LL rules as shown in Table 4.1.

Table 4.1: LL rules [51]

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

		Where %x and %y are nodes.

b) **TT, or tree-to-tree,** where the initial state and the final state are tree structures. There are 3 different subtypes of TT rules as shown in Table 4.2.

Table 4.2: TT rules [51]

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

c) **NN, or network-to-network,** where the initial state and the final state are network structures. There are 3 different subtypes of NN rules as shown in Table 4.3.

Table 4.3: NN rules [51]

ACTION	RULE	DESCRIPTION
ADD RELATION	SEM1(%x;%y):=+SEM2(%w;%z);	The relation SEM2 between the nodes %w and %z will be added to the graph containing the relation SEM1 between the nodes %x and %y.
DELETE	SEM(%x;%y):=-SEM(%x;%y);	The relation SEM between the

RELATION	SEM(%x;%y)=;	nodes %x and %y will be deleted (the nodes %x and %y will not be deleted).
REPLACE RELATION	SEM1(%x;%y):=SEM2(%w;%z);	The relation SEM1 between the nodes %x and %y will be replaced by the relation SEM2 between the nodes %w and %z Where SEM is any of the existing UNL relations, and %x, %y, %z and %w are nodes.

d) **LT, or list-to-tree,** converts lists into trees. There are 2 different subtypes of LT rules as shown in Table 4.4.

Table 4.4: LT rule [51]

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

e) **TN, or tree-to-networks,** converts trees into networks. There are 2 types of TN rules as shown in Table 4.5.

Table 4.5: TN rule [51]

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

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

Fifth is the **D-Rules** tab, NL-UNL disambiguation grammar (*i.e.*, the grammar to be used to control the tokenization and improve the results of the transformation grammar) is provided. Figure 4.5 shows the snapshot of D-Rule tab.

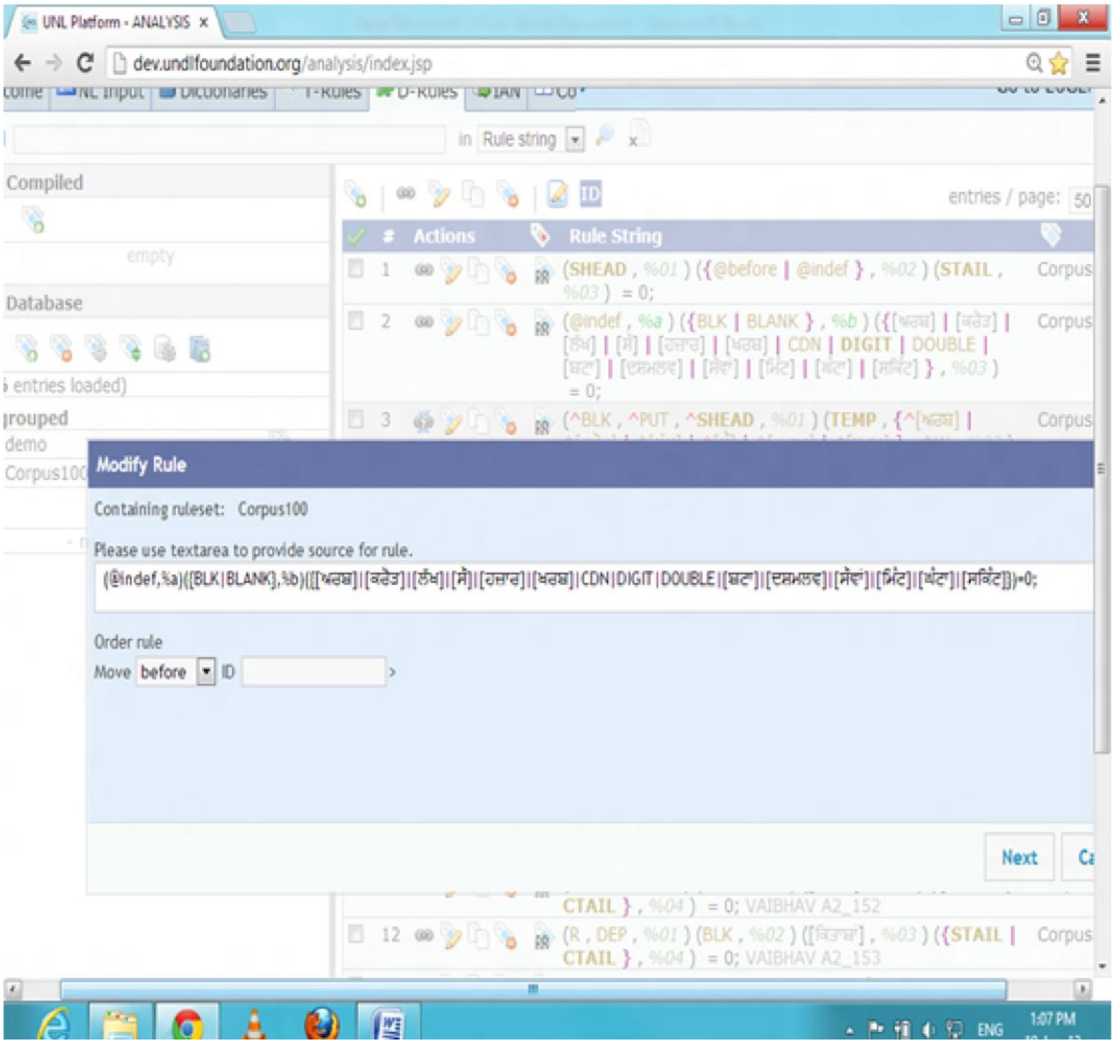

Figure 4.5: Snapshot showing creation of DRule under DRule tab [15]

Sixth tab is the **IAN console** tab, where results are seen. The IAN console brings the list of sentences appearing in the NL input, which may be processed one at a time, or in a range. IAN console shows the complete UNL-ization process of the selected natural language sentence. The results are displayed in 5 different trace levels. Figure 4.6 shows a snapshot of IAN console.

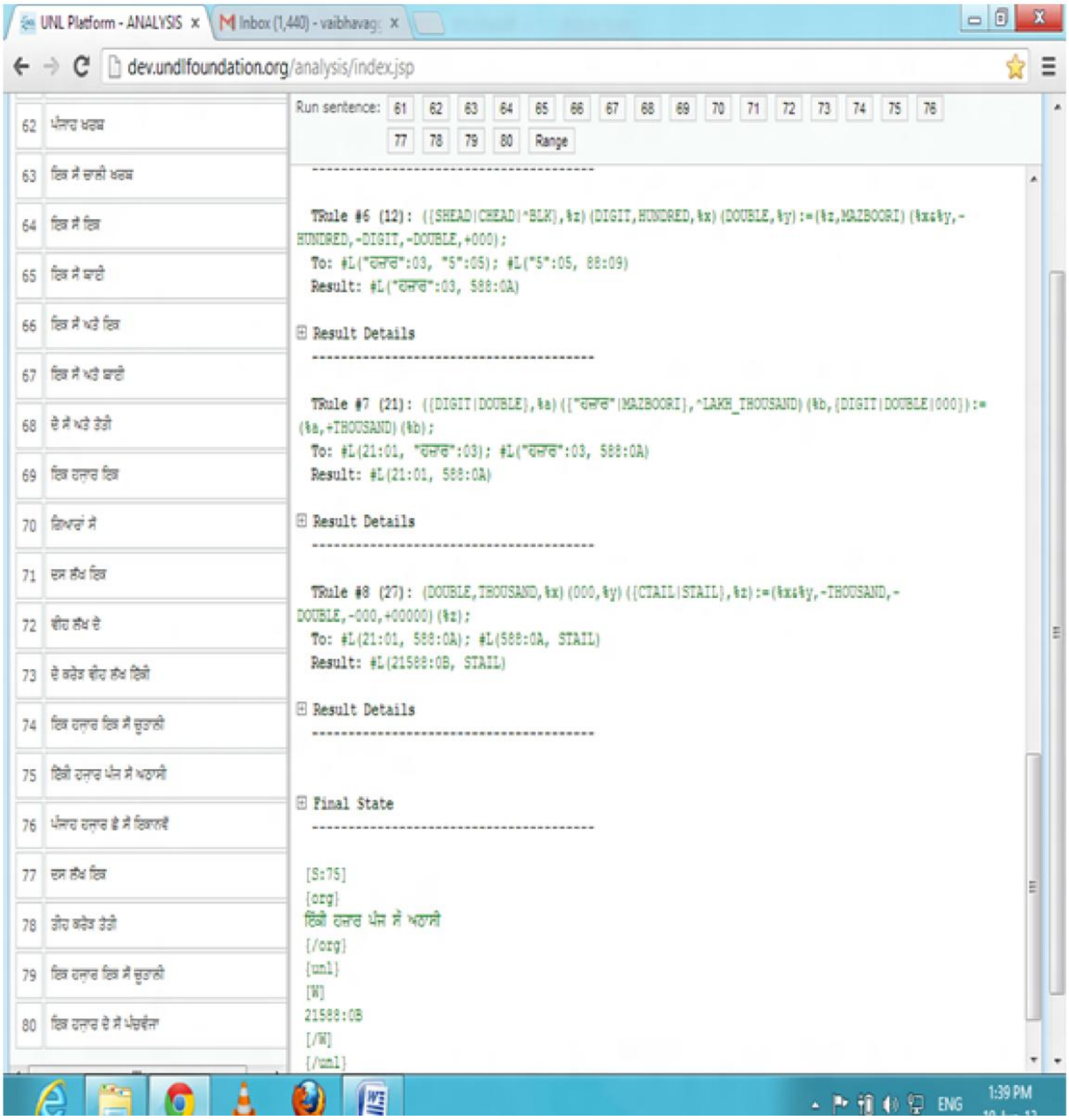

Figure 4.6: Snapshot of IAN console [15]

Chapter 5

Implementation

5.1 Overview

UNDL Foundation has given CORPUS500, UC-A1, and UC-A2. These corpuses were manually converted into our natural language, *i.e.*, Punjabi for input to IAN. The detail of these corpuses has been given in subsequent sections. These corpuses cover all the major part-of-speech of a natural language. These corpuses also include numbers, ordinals, time and temporary entries which are represented in UNL by their natural form. The UNL-ization of these corpuses are explained in the subsequent sections below.

5.2 UNL-ization of CORPUS500

CORPUS500 is a document provided by UNDL Foundation comprising of total 500 sentences covering all the major part-of-speech. CORPUS500 has been categorized into 10 categories. These are given below: -

1.) Temporary words
2.) Numbers and ordinals
3.) Prepositions
4.) Conjunctions
5.) Determiners
6.) Time
7.) Verbs
8.) Adjectives
9.) Nouns, and sentences
10.) Pronouns

UNL-ization of CORPUS500 was done with the help of total 241 TRules and 450 dictionary entries. Sections below give the detailed explanation of UNL-ization process for these corpuses.

5.2.1 UNL-ization of Numbers

UNL represents the numbers in figures, written in the form of words. In Punjabi single digit and double digit numbers cannot be generated by TRules unlike English because in Punjabi they have no repeating pattern. For example in English a double digit number let's say 'fifty-two' and 'fifty-nine' has '*fifty*' as repeating pattern while entries of '*two*' ,'*nine*' have been made earlier in Analysis dictionary. The dictionary entry of every single digit and double digit number like ਇਕ *ik* '*one'*, ਦੋ *dō 'two'*, ਚਾਰ *cār 'four'*, ਗਿਆਰਾਂ *giārāṃ 'eleven'*, ਪੰਦਰਾਂ *pandrāṃ 'fifteen' etc.* are made into the dictionary with their respective attributes as shown below in the examples. While using rule based approach the sequences of rule is very important because the rules are fired in sequence as per matching. For example, TRule 29 will be fired before TRule 34 irrespective of the fact that the nodes matching the criteria of TRule 34 are before the nodes matching the criteria of TRule 29 in the given input string. The UNL-ization process for number has been illustrated with the help of a simple example sentence (5.1).

Example 1: ਤਿੰਨ ਸੌ ਵੀਹ ...(5.1)

tinn sau vīh

three hundred twenty

After the tokenization of example sentence given in (5.1) with IAN tool, five lexical items are identified as shown in (5.2).

[ਤਿੰਨ]{}"3"(LEX=U,POS=CDN,DIGIT, NUM=SNG)<pan,0,0>;

[ਸੌ]{-1}"ਸੌ"(TEMP)<xxx,0,0>;

[ਵੀਹ]{}"20"(LEX=U,POS=CDN,DOUBLE,NUM= PLR)<pan,0,0>;

Two blank spaces are also identified as :-

[]{}" "(BLK)<pan,0,0>; ...(5.2)

Here, *LEX* represents lexical category, *U* represents numeral, *POS* represents part-of-speech, *CDN* represents cardinal, *NUM* represents number whose value could be either *SNG* for singular or *PLR* for plural, *BLK* is the attribute given to the blank space, *TEMP* represents temporary entry, '*DIGIT*' indicates single digit number like ਇਕ *ik 'one'*, ਦੋ *dō 'two'*, ਚਾਰ *cār 'four' etc*. while '*DOUBLE*' indicates two digit number like ਗਿਆਰਾਂ *giārāṃ 'eleven'*, ਪੰਦਰਾਂ *pandrāṃ 'fifteen' etc*. In <pan,0,0> 'pan' refers to the three-character language code for Punjabi according to ISO 639-3. First

‘0’ represents the frequency of Natural Language Word (NLW) in natural texts. The second ‘0’ refers to the priority of the NLW, used in case of NL-ization. The process of UNL-ization of example sentence (5.1) has been illustrated in Table 5.1. Here, transliteration of each Punjabi word is shown only in Action taken column.

Table 5.1: UNL-ization process for example sentence (5.1)

S.No	TRule fired	Description	Action Taken
1.	(%a,BLK):= ;	Here, *%a* refers to blank node [] having attribute ‘BLK’. This rule is fired twice consecutively and it removes all the blank spaces.	Original nodes : [ਤਿੰਨ][][ਸੌ][][ਵੀਹ] *[tinn][][sau][][vīh] [three][][hundred][][twenty]* Resultant nodes: [ਤਿੰਨ][ਸੌ][ਵੀਹ] *[tinn][sau][vīh] [three][hundred][twenty]*
2.	({DIGIT|DOUBLE},%a)("ਸੌ")(%b,{DIGIT|DOUBLE}):=(%a,+HUNDRED)(%b);	Here, node [ਸੌ] *[sau] [hundred]* has been deleted and attribute HUNDRED has been given to previous node [ਤਿੰਨ] *[tinn] [three]*.	Original nodes : [ਤਿੰਨ][ਸੌ][ਵੀਹ] *[tinn][sau][vīh] [three][hundred][twenty]* Resultant nodes : [ਤਿੰਨ][ਵੀਹ] *[tinn][vīh] [three][twenty]*
3.	({SHEAD|CHEAD|^BLK},%a)(DIGIT,HUNDRED,%x)(DOUBLE,%y):=(%a,ATT9)(%x&%y,-HUNDRED,-DIGIT,-DOUBLE,+000);	Here, SHEAD and CHEAD means sentence head and scope head, respectively and ‘^’ is used as negation means logical NOT. This rule concatenates nodes [ਤਿੰਨ] and [ਵੀਹ] to form a single node [ਤਿੰਨਵੀਹ] as shown in the action taken column. All attributes are removed from final node and attribute 000 is added to it, to indicate, that it is a three digit number.	Original nodes: **[ਤਿੰਨ][ਵੀਹ]** ***[tinn][vīh] [three][twenty]*** Resultant nodes: [ਤਿੰਨਵੀਹ] *[tinnvīh] [threetwenty]* Now, ਤਿੰਨ *ik* ‘*three*’ and ਵੀਹ *vīh* ‘*twenty*’ are replaced by their universal words ‘3’ and ‘20’, respectively, and the final output 320 is generated by IAN as shown in (5.3).

UNL generated by IAN is given in (5.3).

{org}

ਤਿੰਨ ਸੌ ਵੀਹ

{/org}

{unl}

[w] 320:08 [/w]

{/unl} ...(5.3)

Here, ':08' is the scope internally generated by the IAN tool.

UNL-ization of numbers has further been explained with the help of another example sentence (5.4).

Example 2: ਪੰਜਾਹ ਖਰਬ ਵੀਹ ਅਰਬ ਤਿੰਨ ਕਰੋੜ ਤੀਹ ਲੱਖ ਅੱਠ ਸੌ ਤਿੰਨ ...(5.4)

pñjāh kharab vīh arab tinn karōṛ tīh lakkh aṭṭh sau tinn

Five trillion twenty billion thirty three million eight hundred three

After tokenization of example sentence given in (5.4) with IAN tool, twenty-one lexical items are identified as given in (5.5).

[ਪੰਜਾਹ]{-1}"50"(LEX=U,POS=CDN,NUM=PLR,DOUBLE)<pan,0,0>;

[ਖਰਬ]{-1}"ਖਰਬ"(TEMP)<xxx,0,0>;

[ਵੀਹ]{}"20"(LEX=U,POS=CDN,DOUBLE,NUM=PLR)<pan,0,0>;

[ਅਰਬ]{-1}" ਅਰਬ"(TEMP)<xxx,0,0>;

[ਤਿੰਨ]{}"3"(LEX=U,POS=CDN,DIGIT,NUM=SNG)<pan,0,0>;

[ਕਰੋੜ]{-1}"ਕਰੋੜ"(TEMP)<xxx,0,0>;

[ਤੀਹ]{}"30"(LEX=U,POS=CDN,DOUBLE,NUM=PLR)<pan,0,0>;

[ਲੱਖ]{-1}"ਲੱਖ"(TEMP)<xxx,0,0>;

[ਅੱਠ]{}"8"(LEX=U,POS=CDN,DIGIT,NUM=SNG)<pan,0,0>;

[ਸੌ]{-1}"ਸੌ"(TEMP)<xxx,0,0>;

Ten blank spaces are also identified as: -

[ ]{}""(BLK)<pan,0,0>; …(5.5)

The process of UNL-ization of example sentence (5.4) is given in Table 5.2.

Table 5.2: UNL-ization process for example sentence (5.4)

S.No	TRule fired	Description	Action Taken
1.	(%a,BLK):=;	This Rule is fired ten times to deletes all four blank nodes as shown in Action taken column.	Original nodes : [ਪੰਜਾਹ][][ਖਰਬ][][ਵੀਹ][][ਅਰਬ][][ਤਿੰਨ][][ਕਰੋੜ][][ਤੀਹ][ ][ਲੱਖ][][ਅੱਠ][][ਸੌ][][ਤਿੰਨ] *[pñjāh][][kharab][][vīh][][arab][][tinn][][karōṛ][][tīh][][lakkh][][aṭṭh][][sau][ ][tinn]* *[fifty][][kharab][][twenty][ ][arab][][three][][crore][][thirty][][lakh][][eight][][hundred][][three]* Resultant nodes : [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ][ਸੌ][ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][lakkh][aṭṭh][sau][tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight][hundred][three]*
2.	({DIGIT|DOUBLE},%a)("ਸੌ")(%b,{DIGIT|DOUBLE}):=(%a,+HUNDRED)(%b);	Here, *%a* refers to node [ਅੱਠ] having attribute DIGIT and *%b* refers to node [ਤਿੰਨ] having attribute DIGIT. This rule deletes node [ਸੌ] and gives attribute HUNDRED to node *%a*.	Original nodes : [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ][ਸੌ][ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][lakkh][aṭṭh][sau][tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight][hundred][three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ] [ਤਿੰਨ] *[pñjāh][kharab][vīh][arab]*

			[tinn][karōṛ][tīh][lakkh][aṭṭh][tinn] *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight][three]*
3.	({SHEAD|CHEAD|^BLK},%z)({DIGIT|DOUBLE},HUNDRED,%x)({STAIL|CTAIL|DIGIT},^DOUBLE,^DOZEN,^00,%y):=(%z)(%x)([[0]],[0],"0",DOZEN,DIGIT)(%y);	Here, *%z* refers to node [ਲੱਖ], *%x* refers to node [ਅੱਠ], and *%y* refers to node [ਤਿੰਨ]. This rule adds a node [0] between *%x* and *%y*. It also attaches the attributes 'DOZEN' and 'DIGIT' to this newly created node.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ][ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][lakkh][aṭṭh][tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight][three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ][0][ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][lakkh][aṭṭh][0][tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight][0][three]*
4.	({SHEAD|CHEAD|^BLK|DIGIT},%z)(DIGIT,DOZEN,%x)(DIGIT,%y):=(%z)(%x&%y,-DOZEN,-DIGIT,+00);	Here, *%z* refers to node [ਅੱਠ], *%x* refers to node [0], and *%y* refers to node [ਤਿੰਨ]. This rule concatenates nodes *%x* and *%y*.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ][0][ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][lakkh][aṭṭh][0][tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight][0][three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ]

			[0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][lakkh][aṭṭh][0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight][0three]*
5.	({SHEAD|CHEAD|^BLK},%z)(DIGIT,HUNDRED,%x)(00,%y):=(%z)(%x&%y,-HUNDRED,-DIGIT,-00,+000);	Here, *%z* refers to node [ਲੱਖ], *%x* refers to node [ਅੱਠ], and *%y* refers to the node [0ਤਿੰਨ]. This rule concatenates *%x* and *%y* to form the node [ਅੱਠ0ਤਿੰਨ]. The attribute '000' is given to it and existing attributes DIGIT, 00, and HUNDRED are removed.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ][0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][lakkh][aṭṭh][0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight][0three]* Resultant nodes : [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][lakkh][aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight0three]*
6.	({DIGIT|DOUBLE|TRIPLE},%a)("ਲੱਖ")(%b,{DIGIT|DOUBLE|000|0000|00000}):=(%a,+LAKH)(%b);	Here, node *%a* refers to [ਤੀਹ], *%b* refers to [ਅੱਠ0ਤਿੰਨ]. This rule deletes node [ਲੱਖ] and gives attribute LAKH to node *%a*.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][ਲੱਖ][ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][lakkh][aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][lakh][eight0three]* Resultant nodes : [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][

			ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ] [ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][eight0three]*
7.	({SHEAD|CHEAD|^BLK},%z)({DIGIT|DOUBLE|TRIPLE},LAKH,%x)({000|DIGIT|DOUBLE|STAIL|CTAIL},^TEMP1,^00000,%y):=(%z)(%x)([[00]],[00],"00",THOUSAND,DOUBLE,LAKH_THOUSAND)(%y);	Here, *%z* refers to node [ਕਰੋੜ], *%x* refers to node [ਤੀਹ], and *%y* refers to node [ਅੱਠ0ਤਿੰਨ]. This rule adds a node [00] between *%x* and *%y*. It also attaches the attributes THOUSAND, DOUBLE and LAKH_THOUSAND to this newly created node.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ] [ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][eight0three]* Resultant nodes : [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][00][ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][00][aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][00][eight0three]*
8.	(LAKH_THOUSAND,THOUSAND,%x)(000,%y)({CTAIL|STAIL},%z):=(%x&%y,-THOUSAND,-000,-DIGIT,-DOUBLE,-LAKH_THOUSAND,+00000)(%z);	Here, *%x* refers to node [00], *%y* refers to node [ਅੱਠ0ਤਿੰਨ], *%z* refers to STAIL. This rule concatenates nodes *%x* and *%y*. Attribute 00000 is given to this node while all other attributes are removed from it.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ] [00][ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][00][aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][00][eight0three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][00ਅੱਠ0ਤਿੰ

			ਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][00aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][00eight0three]*
9.	(DOUBLE,LAKH,%x)(00000,%y)({CTAIL|STAIL},%z):=(%x&%y,-LAKH,-DIGIT,-DOUBLE,-00000,+0000000)(%z);	Here, *%x* refers to node [ਤੀਹ], *%y* refers to node [00ਅੱਠ0ਤਿੰਨ], and *%z* refers to STAIL. This rule concatenates nodes *%x* and *%y* and gives the attribute 0000000 to this concatenated node while removes all other attributes.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ][00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh][00aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty][00eight0three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh00aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty00eight0three]*
10.	({DIGIT|DOUBLE|TRIPLE},%a)("ਕਰੋੜ",%b):=(%a,+CRORE);	Here, *%a* refers to node [ਤਿੰਨ] , *%b* refers to node [ਕਰੋੜ]. This rule deletes node [ਕਰੋੜ] and gives attribute CRORE to node *%a*.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਕਰੋੜ][ਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][karōṛ][tīh00aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][crore][thirty00eight0three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][

			ਤਿੰਨ][ਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][tīh00aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][thirty00eight0three]*
11.	(DIGIT,CRORE,^ARAB_CRORE,%a)(0000000,%b):=(%a&%b,-0000000,-CRORE,-DIGIT,+00000000);	Here, *%a* refers to node [ਤਿੰਨ] and *%b* refers to node [ਤੀਹ00ਅੱਠ0ਤਿੰਨ]. This rule concatenates nodes *%a* and *%b*. Attribute 00000000 is given to this concatenated node while all other attributes are removed.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨ][ਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinn][tīh00aṭṭh0tinn]* *[fifty][kharab][twenty][arab][three][thirty00eight0three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinntīh00aṭṭh0tinn]* *[fifty][kharab][twenty][arab][threethirty00eight0three]*
12.	({DIGIT|DOUBLE|TRIPLE},%a)("ਅਰਬ")(%b,{DIGIT|DOUBLE|000|0000|00000|000000|0000000|00000000|000000000}):=(%a,+ARAB)(%b);	Here, *%a* refers to node [ਵੀਹ] , and *%b* refers to node [ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ]. This rule deletes node [ਅਰਬ] and gives attribute ARAB to node *%a*.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਅਰਬ][ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][arab][tinntīh00aṭṭh0tinn]* *[fifty][kharab][twenty][arab][threethirty00eight0three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][tinntīh00aṭṭh0tinn]* *[fifty][kharab][twenty][threethirty00eight0three]*

13.	{DIGIT|DOUBLE|TRIPLE},ARAB,%x)(00000000,%y):=(%x)([[0]],[0],"0",TEMP3,%z)(%y);	Here, *%x* refers to node [ਵੀਹ] , and *%y* refers to node [ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ]. This rule adds a node [0] between *%x* and *%y*. Attribute TEMP3 is given to this new node.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][tinntīh00aṭṭh0tinn]* *[fifty][kharab][twenty][threethirty00eight0three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][0][ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][0][tinntīh00aṭṭh0tinn]* *[fifty][kharab][twenty][0][threethirty00eight0three]*
14.	(TEMP3,%a)(00000000,%b):=(%a&%b,-TEMP3,-00000000,+000000000);	Here, *%a* refers to node [0], *%b* refers to node [ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ]. This rule concatenates nodes *%a* and *%b*. Attribute 000000000 is given to this concatenated node.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][0][ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][0][tinntīh00aṭṭh0tinn]* *[fifty][kharab][twenty][0][threethirty00eight0three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][0ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][0tinntīh00aṭṭh0tinn]* *[fifty][kharab][twenty][0threethirty00eight0three]*
15.	(DOUBLE,ARAB,%a)(000000000,%b):=(%a&%b,-000000000,-ARAB,-DOUBLE,+00000000000);	Here, *%a* refers to node node [ਵੀਹ], and *%b* refers to node [0ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] .This rule concatenates node *%a* and *%b*. Attribute 00000000000 is given to this concatenated node.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ][0ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh][0tinntīh00aṭṭh0tinn]* *[fifty][kharab][twenty][0threethirty00eight0three]* Resultant nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ0ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh0tinntī*

			h00aṭṭh0tinn] *[fifty][kharab][twenty0threethirty00eight0three]*
16.	({DIGIT|DOUBLE|TRIPLE},%a)("ਖਰਬ")(%b,{DIGIT|DOUBLE|000|0000|00000|000000|0000000|00000000|000000000|0000000000|00000000000}):=(%a,+KHARAB)(%b);	Here , *%a* refers to node [ਪੰਜਾਹ], and *%b* refers to node [ਵੀਹ]. This rule deletes node [ਖਰਬ] and gives attribute KHARAB to node *%a*.	Original nodes: [ਪੰਜਾਹ][ਖਰਬ][ਵੀਹ0ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][kharab][vīh0tinntīh00aṭṭh0tinn]* *[fifty][kharab][twenty0threethirty00eight0three]* Resultant nodes: [ਪੰਜਾਹ][ਵੀਹ0ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][vīh0tinntīh00aṭṭh0tinn]* *[fifty][twenty0threethirty00eight0three]*
17.	(DOUBLE,KHARAB,%a)(00000000000,%b):=(%a&%b,-KHARAB,-DOU BLE,-00000000000,+0000000000000);	Here, *%a* refers to node [ਪੰਜਾਹ], and *%b* refers to node [ਵੀਹ0ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ]. This rule concatenates nodes *%a* and *%b*. Attribute 0000000000000 is given to this concatenated node.	Original nodes: [ਪੰਜਾਹ][ਵੀਹ0ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāh][vīh0tinntīh00aṭṭh0tinn]* *[fifty][twenty0threethirty00eight0three]* Resultant nodes: [ਪੰਜਾਹਵੀਹ0ਤਿੰਨਤੀਹ00ਅੱਠ0ਤਿੰਨ] *[pñjāhvīh0tinntīh00aṭṭh0tinn]* *[fiftytwenty0threethirty00eight0three]* Now, ਪੰਜਾਹ *pñjāh* '*fifty*', ਵੀਹ *vīh* '*twenty*', ਤਿੰਨ *tinn* '*three*', ਤੀਹ *tīh* '*thirty*', and ਅੱਠ *aṭṭh* 'eight' are replaced by their universal words '50', '20', '3', '30'

			and ‘8’ respectively, and the final output 5020033000803 is generated by IAN as shown in (5.6).

UNL generated by IAN is given in (5.6).

{org}

ਪੰਜਾਹ ਖਰਬ ਵੀਹ ਅਰਬ ਤਿੰਨ ਕਰੋੜ ਤੀਹ ਲੱਖ ਅੱਠ ਸੌ ਤਿੰਨ

{/org}

{unl}

[w] 5020033000803:0F [/w]

{/unl} ...(5.6)

Here ‘:0F’ is the scope internally generated by the IAN tool.

5.2.2 UNL-ization of ordinals

Ordinals numbers in Punjabi language contains ਲਾਂ *lāṃ* / ਵਾਂ *vāṃ* / ਥਾ *thā* / ਜਾ *jā* as suffix as in ਪਹਿਲਾਂ *pahilāṃ* ‘*first*’, ਇੱਕੀਵਾਂ *ikkīvāṃ* ‘*twenty first*’, ਚੌਥਾ *cauthā* ‘*fourth*’, ਤੀਜਾ *tījā* ‘*third*’, respectively. In UNL ordinals are represented in figures and attribute “@ordinal” is given to it to retain its semantics. During UNL-ization of ordinals, all rules fired will be same, with the only difference that after deleting blank spaces, a new rule as given in (5.7) will be fired. This rule will add “@ordinal” attribute to the corresponding node having suffixes ਲਾਂ *lāṃ* / ਵਾਂ *vāṃ* / ਥਾ *thā* / ਜਾ *jā*.

({DIGIT|DOUBLE|CDN},%x)("ਵਾਂ"|"ਲਾਂ"|"ਥਾ"|"ਜਾ"):=(%x,ORD,att=@ordinal); …(5.7)

This rule gives an attribute “@ordinal” to node having any one of the attributes *DIGIT*, *DOUBLE*, or *CDN* (being referred as *%x*) preceding any one of the nodes [ਵਾਂ] / [ਲਾਂ] / [ਥਾ] / [ਜਾ]. The UNL-ization process for ordinals has been illustrated with the help of a simple example sentence (5.8).

Example 3: ਤਿੰਨ ਸੌ ਵੀਹਵਾਂ ...(5.8)

tinn sau vīhvāṃ

three hundred twentieth

After tokenization of example sentence (5.8) with IAN tool, five lexical items are identified as already given in (5.2) and it produce an additional lexical item as shown in (5.9).

[ਵਾਂ]{-1}"ਵਾਂ"(TEMP)<xxx,0,0>; ...(5.9)

The lexical item help in firing of rule given in (5.7) to attach the attribute '@ordinal'. UNL generated by IAN for example sentence (5.8) is given in (5.10).

{org}

ਤਿੰਨ ਸੌ ਵੀਹਵਾਂ

{/org}

{unl}

[w] 320:09.@ordinal [/w]

{/unl} ...(5.10)

Here, ':09' is the scope internally generated by the IAN tool.

5.2.3 UNL-ization of Temporary entries

Temporary entries are entries that are not expected to be UNL-ized, such as URL's, e-mail addresses, phone numbers, *etc.* They are not included in the dictionary, because the UNL is the same as the source. Temporary entries are represented, in UNL, between quotes, as natural language entries. For instance, the UNL for "asfsdfdfdsf" is "asfsdfdfdsf"; the UNL for "www.undlfoundation.org" is "www.undlfoundation.org"; *etc.* The UNL-ization process for temporary entries has been illustrated with the help of a simple example sentence (5.11).

Example 1: asfsdfdfdsf asfsdfdfdsf asfsdfdfdsf ...(5.11)

After tokenization of example sentence given in (5.11) with IAN tool five lexical items are identified, out of which three are the same temporary words and two are the blank spaces, as given in (5.12).

[asfsdfdfdsf]{-1}"asfsdfdfdsf"(TEMP)<xxx,0,0>;

[ ]{}""(BLK)<pan,0,0>; ...(5.12)

The process of UNL-ization of example sentence (5.11) is given in Table 5.3.

Table 5.3: UNL-ization process for example sentence (5.11)

S.No	TRule fired	Description	Action Taken
1.	(%a,TEMP)(BLK,%b)(%c,TEMP):=(%a)(%b,-BLK,+BLANK)(%c);	Here, *%a* refers to first temporary node [asfsdfdfdsf] having attribute 'TEMP', *%b* refers to first blank node [], and *%c* refers to second temporary node [asfsdfdfdsf]. This rule is fired twice consecutively to remove attribute 'BLK' from both blank nodes and gives them attribute 'BLANK'. This is done so that this blank node does not get deleted.	Original nodes : [asfsdfdfdsf][][asfsdfdfdsf][][asfsdfdfdsf] Resultant nodes: [asfsdfdfdsf][][asfsdfdfdsf][][asfsdfdfdsf]
2.	(%a,TEMP)(BLK,%b)(%c,TEMP):=(%a&%b&%c);	Here, *%a* refers to first temporary node [asfsdfdfdsf] having attribute 'TEMP', *%b* refers to first blank node [], and *%c* refers to second temporary node [asfsdfdfdsf]. This rule concatenates node *%a*, *%b* and %c as shown in Action Taken column.	Original nodes : [asfsdfdfdsf][][asfsdfdfdsf][][asfsdfdfdsf] Resultant nodes: [asfsdfdfdsf asfsdfdfdsf][][asfsdfdfdsf]
3.	(%a,TEMP)(BLK,%b)(%c,TEMP):=(%a&%b&%c);	Here, *%a* refers to first temporary node [asfsdfdfdsf asfsdfdfdsf], *%b* refers to second blank node [], and *%c* refers to third temporary node [asfsdfdfdsf]. This rule concatenates node *%a*, *%b* and %c as shown in Action Taken column.	Original nodes: [asfsdfdfdsf asfsdfdfdsf][][asfsdfdfdsf] Resultant nodes: [asfsdfdfdsf asfsdfdfdsf asfsdfdfdsf] Final output generated by IAN is as shown in (5.13).

UNL generated by IAN tool is given in (5.13).

{org}

asfsdfdfdsf asfsdfdfdsf asfsdfdfdsf

{/org}

{unl}

[w] "asfsdfdfdsf asfsdfdfdsf asfsdfdfdsf ":07[/w]

{/unl} ...(5.13)

Here ':07' is the scope internally generated by the IAN tool.

5.2.4 UNL-ization of Prepositions

In UNL, prepositions are represented either by only relations or by both relations and attributes. The UNL of the preposition 'for' in different context is shown in Table 5.4.

Table 5.4: Values of Preposition 'for' [37]

Value	**UNL**	**Examples**	
		English	**UNL**
Aim or purpose	pur	trained for the ministry, put the house up for sale	pur(trained, ministry)
Destination	to	headed off for town	to(headed off, town)
Object of a desire, intention or perception	obj	eager for success	obj(eager, success)
Recipient (direction, addressee) of an action	gol	prepared lunch for us	gol(prepared, us)
Beneficiary of an action	ben	He did it for us	ben(did, us)
Duration	dur	stood in line for an hour	dur(stood, hour)
On behalf of, in favor of, in place of	man	spoke for all members	man(spoke, members.@for)
Amount, extent	man	a bill for five dollars; walked for miles;	man(bill, dollars.@for)
As being	man	mistook me for the librarian	man(mistook, librarian.@for)

The UNL-ization process for prepositions has been illustrated with the help of a simple example sentence (5.14).

Example 1: ਮੇਜ਼ ਉੱਤੇ ਪੈਰਿਸ ਬਾਰੇ ਤਸਵੀਰਾਂ ਤੋਂ ਬਿਨਾਂ ਕਿਤਾਬ ...(5.14)

mēz uttē pairis bārē tasvīrāṃ tōṃ bināṃ kitāb

The book on the table about Paris without pictures

After tokenization of example sentence (5.14) with IAN tool thirteen lexical items are identified as given in (5.15).

[ਮੇਜ਼]{}"table"(LEX=N,POS=NOU,NUM=SNG)<pan,0,0>;

[ਉੱਤੇ]{}"on"(LEX=P,POS=PRE,rel=plc,att=@on)<pan,0,0>;

[ਪੈਰਿਸ]{}"Paris"(LEX=N,POS=PPN,NUM=SNG,SEM=LCT)<pan,0,0>;

[ਬਾਰੇ]{}"about"(LEX=P,POS=PRE,rel=cnt,att=@about)<pan,0,0>;

[ਤਸਵੀਰਾਂ]{}"picture"(LEX=N,POS=NOU,NUM=PLR)<pan,0,0>;

[ਤੋਂ ਬਿਨਾਂ]{}"without"(LEX=P,POS=PRE,rel=ma n,att=@without)<pan,0,0>;

[ਕਿਤਾਬ]{}"book"(LEX=N,POS=NOU,NUM=SNG)<pan,0,0>;

Six blank spaces are also identified as :-

[ ]{}" "(BLK)<pan,0,0>; …(5.15)

The process of UNL-ization of example sentence (5.14) has been illustrated in Table 5.5.

Table 5.5: UNL-ization process for example sentence (5.14)

S.No	TRule fired	Description	Action Taken
1.	(%a,BLK):=;	Here, *%a* refers to blank node []. This rule is fired six times and deletes all the blank spaces.	Original nodes : [ਮੇਜ਼][][ਉੱਤੇ][][ਪੈਰਿਸ][][ਬਾਰੇ][][ਤਸਵੀਰਾਂ][][ਤੋਂ ਬਿਨਾਂ][][ਕਿਤਾਬ] *[mēz][][uttē][][pairis][][bārē][][tasvīrāṃ][][tōṃ bināṃ][][kitāb]* Resultant nodes: [ਮੇਜ਼][ਉੱਤੇ][ਪੈਰਿਸ][ਬਾਰੇ][ਤਸਵੀਰਾਂ][ਤੋਂ ਬਿਨਾਂ] [ਕਿਤਾਬ] *[mēz][uttē][pairis][bārē][tasvīrāṃ][tōṃ bināṃ][kitāb]*
2.	(N,%a)(P,PRE,rel,att,%b):=(%a,+att=%b,+rel=%b,+N);	Here, *%a* refers to node [ਮੇਜ਼], *%b* refers to node [ਉੱਤੇ]. This rule deletes the node *%b* and gives its attributes to node *%a*.	Original nodes : [ਮੇਜ਼][ਉੱਤੇ][ਪੈਰਿਸ][ਬਾਰੇ][ਤਸਵੀਰਾਂ][ਤੋਂ ਬਿਨਾਂ] [ਕਿਤਾਬ] *[mēz][uttē][pairis][bārē][tasvīrāṃ][tōṃ bināṃ][kitāb]*

			Resultant nodes: [ਮੇਜ਼@on][ਪੈਰਿਸ][ਬਾਰੇ][ਤਸਵੀਰਾਂ][ਤੋਂ ਬਿਨਾਂ] [ਕਿਤਾਬ] *[mēz@on][pairis][bārē][tasvīrāṃ][tōṃ bināṃ][kitāb]*
3.	(N,%a)(P,PRE,rel,att,%b):=(%a,+att=%b,+rel=%b,+N);	Here, *%a* refers to node [ਤਸਵੀਰਾਂ], *%b* refers to node [ਤੋਂ ਬਿਨਾਂ]. As above, this rule deletes the node *%b* and gives its attributes to node *%a*.	Original nodes: [ਮੇਜ਼@on][ਪੈਰਿਸ][ਬਾਰੇ][ਤਸਵੀਰਾਂ][ਤੋਂ ਬਿਨਾਂ] [ਕਿਤਾਬ] *[mēz@on][pairis][bārē][tasvīrāṃ][tōṃ bināṃ][kitāb]* Resultant nodes: [ਮੇਜ਼@on][ਪੈਰਿਸ][ਬਾਰੇ][ਤਸਵੀਰਾਂ@without][ਕਿਤਾਬ] *[mēz@on][pairis][bārē][tasvīrāṃ@without][kitāb]*
4.	(N,rel=man,att,%a)(N,%b):=(NA(%b;%a),+MAN,+N);	Here, *%a* refers to node [ਤਸਵੀਰਾਂ], *%b* refers to node [ਕਿਤਾਬ]. This rule resolves a relation 'NA' whose first and second argument are *%b* and *%a* respectively. This new node so formed is given an attribute 'MAN' so that at later stages it could be resolved into the actual UNL relation 'man'. This new node is treated as Noun and hence attribute 'N' is given to this node.	Original nodes: [ਮੇਜ਼@on][ਪੈਰਿਸ][ਬਾਰੇ][ਤਸਵੀਰਾਂ@without][ਕਿਤਾਬ] *[mēz@on][pairis][bārē][tasvīrāṃ@without][kitāb]* Resultant nodes: [ਮੇਜ਼@on][ਪੈਰਿਸ][ਬਾਰੇ][NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without])] *[mēz@on][pairis][bārē][NA([kitāb];[tasvīrāṃ@without])]*
5.	(N,%a)(P,cnt,%b)(N,%c):=(NA(%c;%a,+att=%b),%d,+N,+CNT);	Here, *%a* refers to node [ਪੈਰਿਸ], *%b* refers to node [ਬਾਰੇ], and *%c* refers to node [ਤਸਵੀਰਾਂ@without]. This rule resolves a relation 'NA' whose first and second arguments are *%c* and *%a* respectively. The new node so formed is given the name *%d* and attributes 'CNT' and 'N' for same reasons as in previous rule. Second argument of the relation is given attributes of *%b*.	Original nodes: [ਮੇਜ਼@on][ਪੈਰਿਸ][ਬਾਰੇ][NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without])] *[mēz@on][pairis][bārē][NA([kitāb];[tasvīrāṃ@without])]* Resultant nodes: [ਮੇਜ਼@on][NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without])];[ਪੈਰਿਸ@about])] *[mēz@on][NA([NA([kitāb];[tasvīrāṃ@without])];[pairis@about])]*

6.	(N,rel=plc,att,%a)(N,^rel,%b):=(NA(%b;%a),+PLC,+N);	Here, *%a* refers to node [ਮੇਜ਼@on], *%b* refers to node [NA(NA(ਕਿਤਾਬ;ਤਸਵੀਰਾਂ@without);ਪੈਰਿਸ@about)]. This rule results into a relation 'NA' with first, second arguments as *%b* and *%a* respectively.	Original nodes: [ਮੇਜ਼@on][NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without])];[ਪੈਰਿਸ@about])] *[mēz@on] [NA([NA([kitāb];[tasvīrāṃ@without])];[pairis@about])]* Resultant nodes: [NA([NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without])];[ਪੈਰਿਸ@about])];[ਮੇਜ਼@on])] *[NA([NA([NA([kitāb];[tasvīrāṃ@without])];[pairis@about])];[mēz@on])]*
7.	(N,PLR,^@pl,%a):=(%a,+@pl);	Here, *%a* refers to node [ਤਸਵੀਰਾਂ@without]. This rule adds attribute '@pl' to node *%a*.	Original nodes: [NA([NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without])];[ਪੈਰਿਸ@about])];[ਮੇਜ਼@on])] *[NA([NA([NA([kitāb];[tasvīrāṃ@without])];[pairis@about])];[mēz@on])]* Resultant nodes: [NA([NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])];[ਪੈਰਿਸ@about])];[ਮੇਜ਼@on])] *[NA([NA([NA([kitāb];[tasvīrāṃ@without@pl])];[pairis@about])];[mēz@on])]*
8.	(NA(NA(%a;%b),CNT,%w;%c),PLC,%r):=(%w),(NA(%a;%c),+PLC);	Here, *%a* refers to node [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])], *%b* refers to node [ਪੈਰਿਸ@about], *%w* refers to [NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])];[ਪੈਰਿਸ@about])], *%c* refers to node [ਮੇਜ਼@on], *%r* refers to original node as	Original nodes: [NA([NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])];[ਪੈਰਿਸ@about])];[ਮੇਜ਼@on])] *[NA([NA([NA([kitāb];[tasvīrāṃ@without@pl])];[pairis@about])];[mēz@on])]* Resultant nodes: Node1:

		shown in Action Taken column. This rule splits node *%r* into nodes %w and a new node having relation 'NA' with first and second argument as *%a* and *%c* respectively.	[NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ @without@pl])];[ਪੈਰਿਸ@about])] *[NA([NA([kitāb];[tasvīrāṃ @without@pl])];[pairis@ about])]* Node2: [NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ @without@pl])];[ਮੇਜ਼@on])] *[NA([NA([kitāb];[tasvīrāṃ @without@pl])];[mēz@on])]*
9.	(NA(NA(%a;%b),MAN,%w;%c), CNT,%r) :=(%w),(NA(%a;%c), +CNT);	Here, *%a* refers to node [ਕਿਤਾਬ], *%b* refers to node [ਤਸਵੀਰਾਂ@without@pl], *%c* refers to [ਪੈਰਿਸ@about], *%w* refers to node [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])], and *%r* refers to node [NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@ without@pl])];[ਪੈਰਿਸ@about])]. This rule split node *%r* into nodes *%w* and a new node having relation 'NA' with first and second argument as *%a* and *%c* respectively.	Original nodes: Node1: [NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ @without@pl])];[ਪੈਰਿਸ@about])] *[NA([NA([kitāb];[tasvīrāṃ @w ithout@pl])];[pairis@ about])]* Node2: [NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ @without@pl])];[ਮੇਜ਼@on])] *[NA([NA([kitāb];[tasvīrāṃ @without@pl])];[mēz@on])]* Resultant nodes: Node 1: [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])] *[NA([kitāb];[tasvīrāṃ@without@pl])]* Node 2: [NA([ਕਿਤਾਬ];[ਪੈਰਿਸ@about])] *[NA([kitāb];[pairis@about])]*

			Node3: [NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])];[ਮੇਜ਼@on])] *[NA([NA([kitāb];[tasvīrāṃ@without@pl])];[mēz@on])]*
10.	(NA(NA(%a;%b),MAN,%w;%c),PLC,%r):=(%w),(NA(%a;%c),+PLC);	Here, *%r* refers to node [NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])]; [ਮੇਜ਼@on])], *%c* refers to node [ਮੇਜ਼@on], *%w* refers to node [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])], *%a* refers to node [ਕਿਤਾਬ] , *%b* refers to node [ਤਸਵੀਰਾਂ@without@pl]. This rule split node *%r* into nodes *%w* and a new node having relation 'NA' with first and second argument as *%a* and *%c* respectively. Note that node *%w* is already present and hence redundancy is removed by IAN tool and in final UNL, redundant nodes appears only once. Therefore redundant nodes are shown in Action Taken column only once.	Original nodes: Node 1: [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])] *[NA([kitāb];[tasvīrāṃ@without@pl])]* Node 2: [NA([ਕਿਤਾਬ];[ਪੈਰਿਸ@about])] *[NA([kitāb];[pairis@about])]* Node3: [NA([NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])];[ਮੇਜ਼@on])] *[NA([NA([kitāb];[tasvīrāṃ@without@pl])];[mēz@on])]* Resultant nodes: Node1: [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])] *[NA([kitāb];[tasvīrāṃ@without@pl])]* Node2: [NA([ਕਿਤਾਬ];[ਪੈਰਿਸ@about])] *[NA([kitāb];[pairis@about])]* Node3: [NA([ਕਿਤਾਬ];[ਮੇਜ਼@on])]

			[NA([kitāb];[mēz@on])]
11.	(NA(%a;%b),CNT):=cnt(%a;%b);	Here, *%a* refers to node [ਕਿਤਾਬ], *%b* refers to node [ਪੈਰਿਸ@about]. This rule changes the name of relation from 'NA' to 'cnt' keeping same arguments as in original node, as required in the final UNL.	Original nodes: Node1: [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])] *[NA([kitāb];[tasvīrāṃ@without@pl])]* Node2: [NA([ਕਿਤਾਬ];[ਪੈਰਿਸ@about])] *[NA([kitāb];[pairis@about])]* Node3: [NA([ਕਿਤਾਬ];[ਮੇਜ਼@on])] *[NA([kitāb];[mēz@on])]* Resultant nodes: Node1: [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])] *[NA([kitāb];[tasvīrāṃ@without@pl])]* Node2: [cnt([ਕਿਤਾਬ];[ਪੈਰਿਸ@about])] *[cnt([kitāb];[pairis@about])]* Node3: [NA([ਕਿਤਾਬ];[ਮੇਜ਼@on])] *[NA([kitāb];[mēz@on])]*
12.	(NA(%a;%b),PLC):=plc(%a;%b)	Here, *%a* refers to node [ਕਿਤਾਬ], *%b* refers to node [ਮੇਜ਼@on]. This rule changes the name of relation from 'NA' to 'plc' keeping same arguments as in original node, as required in the final UNL.	Original nodes: Node1: [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])] *[NA([kitāb];[tasvīrāṃ@without@pl])]* Node2: [cnt([ਕਿਤਾਬ];[ਪੈਰਿਸ@about])] *[cnt([kitāb];[pairis@about*

			])] Node3: [NA([ਕਿਤਾਬ];[ਮੇਜ਼@on])] *[NA([kitāb];[mēz@on])]* Resultant nodes: Node1: [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])] *[NA([kitāb];[tasvīrāṃ@without@pl])]* Node2: [cnt([ਕਿਤਾਬ];[ਪੈਰਿਸ@about])] *[cnt([kitāb];[pairis@about])]* Node3: [plc([ਕਿਤਾਬ];[ਮੇਜ਼@on])] *[plc([kitāb];[mēz@on])]*
13.	(NA(%a;%b),MAN):=plc(%a;%b)	Here, *%a* refers to node [ਕਿਤਾਬ], *%b* refers to node [ਤਸਵੀਰਾਂ@without@pl]. This rule changes the name of relation from 'NA' to 'man' keeping same arguments as in original node, as required in the final UNL.	Original nodes: Node1: [NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])] *[NA([kitāb];[tasvīrāṃ@without@pl])]* Node2: [cnt([ਕਿਤਾਬ];[ਪੈਰਿਸ@about])] *[cnt([kitāb];[pairis@about])]* Node3: [plc([ਕਿਤਾਬ];[ਮੇਜ਼@on])] *[plc([kitāb];[mēz@on])]* Resultant nodes: Node1: [man([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])] *[man([kitāb];[tasvīrāṃ@without@pl])]* Node2: [cnt([ਕਿਤਾਬ];[ਪੈਰਿਸ@about])] *[cnt([kitāb];[pairis@about*

			])] Node3: [plc([ਕਿਤਾਬ];[ਮੇਜ਼@on])] *[plc([kitāb];[mēz@on])]* Now all the natural language words are replaced by their universal words and final output is generated by IAN as shown in (5.16).

UNL generated by IAN is given in (5.16).

{org}

ਮੇਜ਼ ਉੱਤੇ ਪੈਰਿਸ ਬਾਰੇ ਤਸਵੀਰਾਂ ਤੋਂ ਬਿਨਾਂ ਕਿਤਾਬ

{/org}

{unl}

plc(book:0D, table:01.@on)

cnt(book:0D, paris:05.@about)

man(book.:0D, picture:09.@without.@pl)

{/unl} ...(5.16)

Here, :0D, :01, :05, :09, are the scopes internally generated by the IAN tool.

5.2.5 UNL-ization of Conjunctions

Conjunctions are those lexical items which joins two or more phrases or clauses. In UNL framework conjunctions are classified as coordinating conjunctions (COO) which joins two or more items of equal syntactic importance ("ਅਤੇ", "ਜਾਂ") and subordinating conjunctions (SCJ), which introduce a dependent clause like "ਕਿ". The UNL-ization process of conjunctions has been illustrated with the help of a simple example sentence (5.17).

Example 1: ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ ਅਤੇ ਪਾੱਲ ...(5.17)

jān, mairī, pīṭar atē pāll

John, Mary, Peter and Paul

After the tokenization of example sentence given in (5.17) with IAN tool, ten lexical items are identified as shown in (5.18).

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

[,]{}""(LEX=C,POS=COO,rel=and)<pan,0,0>;

[ਮੈਰੀ]{}"Mary"(LEX=N,POS=PPN,GEN=FEM,NUM=SNG)<pan,0,0>;

[ਪੀਟਰ]{}"Peter"(LEX=N,POS=PPN,GEN=MCL,NUM=SNG,CAS=NOM)<pan,0,0>

[ਅਤੇ]{}"and"(LEX=C,POS=COO,rel=and)<pan,0,0>;

[ਪਾੱਲ]{}"Paul"(LEX=N,POS=PPN,NUM=SNG)<pan,0,0>;

Four blank spaces are also identified as: -

[ ]{}""(BLK)<pan,0,0>; ...(5.18)

Here, GEN=MCL means that gender is masculine, CAS=NOM indicates that case is nominal, POS=COO indicates that node's part of speech is coordinating conjunction.The process of UNL-ization of example sentence (5.17) has been illustrated in Table 5.6.

Table 5.6: UNL-ization process for example sentence (5.17)

S.No	TRule fired	Description	Action Taken
1.	(%a,BLK): =;	Here, *%a* refers to blank node []. This rule is fired four times and deletes all the blank spaces.	Original nodes : [ਜਾਨ][,][][ਮੈਰੀ][,][][ਪੀਟਰ][ ][ਅਤੇ][][ਪਾੱਲ] *[jān][,][][mairī][,][][pīṭar][][atē][][pāll]* *[John][,][][Mary][,][][Pet er][][and][][Paul]* Resultant nodes: [ਜਾਨ][,][ਮੈਰੀ][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ] *[jān][,][mairī][,][pīṭar][at ē][pāll]* *[John][,][Mary][,][Peter][and][Paul]*
2.	(N,^COMM AAND,%a) (C,COO,an d,%b)(N,^C	Here, *%a* refers to node [ਪੀਟਰ], *%b* refers to node [ਅਤੇ], *%c* refers to [ਪਾੱਲ]. This	Original nodes : [ਜਾਨ][,][ਮੈਰੀ][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ]

	OMMAAND,%c):=(%a,C OMMAAND)(%b)(%c, COMMAAND);	rule gives attribute 'COMMAAND' to nodes *%a* and *%c*.	*[jān][,][mairī][,][pīṭar][atē][pāll]* *[John][,][Mary][,][Peter][and][Paul]* Resultant nodes: [ਜਾਨ][,][ਮੈਰੀ][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ] *[jān][,][mairī][,][pīṭar][atē][pāll]* *[John][,][Mary][,][Peter][and][Paul]*
3.	(N,^COMMAAND,%a)(C,COO,%b)(N,COMMAAND,%c):=(%a,+COMMAAND) (%b)(%c);	Here, *%a* refers to node [ਮੈਰੀ], *%b* refers to node [,], *%c* refers to [ਪੀਟਰ]. This rule gives attribute 'COMMAAND' to node *%a*.	Original nodes: [ਜਾਨ][,][ਮੈਰੀ][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ] *[jān][,][mairī][,][pīṭar][atē][pāll]* *[John][,][Mary][,][Peter][and][Paul]* Resultant nodes: [ਜਾਨ][,][ਮੈਰੀ][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ] *[jān][,][mairī][,][pīṭar][atē][pāll]* *[John][,][Mary][,][Peter][and][Paul]*
4.	(N,^COMMAAND,%a)(C,COO,%b)(N,COMMAAND,%c):=(%a,+COMMAAND) (%b)(%c);	Here, *%a* refers to node [ਜਾਨ], *%b* refers to node [,], *%c* refers to [ਮੈਰੀ]. This rule gives attribute 'COMMAAND' to node *%a*.	Original nodes: [ਜਾਨ][,][ਮੈਰੀ][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ] *[jān][,][mairī][,][pīṭar][atē][pāll]* *[John][,][Mary][,][Peter][and][Paul]* Resultant nodes: [ਜਾਨ][,][ਮੈਰੀ][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ] *[jān][,][mairī][,][pīṭar][atē][pāll]* *[John][,][Mary][,][Peter][and][Paul]*
5.	(N,PPN,COMMAAND,%a)(C,and,COO,%b)(	Here, *%a* refers to node [ਜਾਨ], *%b* refers to node [,], and *%c* refers to node [ਮੈਰੀ]. This rule	Original nodes: [ਜਾਨ][,][ਮੈਰੀ][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ]

	N,PPN,COMMAAND,%c):=(NA(%c ;%a),+N,+PPN,AND,COMMAAND);	resolves a relation 'NA' whose first and second arguments are *%c* and *%a* respectively. The new node so formed is given attributes 'N', 'PPN', 'AND' and 'COMMAAND'.	*[jān][,][mairī][,][pīṭar][atē][pāll]* *[John][,][Mary][,][Peter][and][Paul]* Resultant nodes: [NA([ਮੈਰੀ];[ਜਾਨ])][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ] *[NA([mairī];[jān])][,][pīṭar][atē][pāll]* *[NA([Mary];[John])][,][Peter][and][Paul]*
6.	(N,PPN,COMMAAND,%a)(C,and,COO,%b)(N,PPN,COMMAAND,%c):=(NA(%c; %a),+N,+PPN,AND,COMMAAND);	Here, *%a* refers to node [NA([ਮੈਰੀ]; [ਜਾਨ])], *%b* refers to node [,] and *%c* refers to node [ਪੀਟਰ]. This rule resolves a relation 'NA' whose first and second arguments are *%c* and *%a* respectively. The new node so formed is given attributes 'N', 'PPN', 'AND' and 'COMMAAND'.	Original nodes: [NA([ਮੈਰੀ];[ਜਾਨ])][,][ਪੀਟਰ][ਅਤੇ][ਪਾੱਲ] *[NA([mairī];[jān])][,][pīṭar][atē][pāll]* *[NA([Mary];[John])][,][Peter][and][Paul]* Resultant nodes: [NA([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])][ਅਤੇ][ਪਾੱਲ] *[NA([pīṭar];[NA([mairī];[jān])])] [atē][pāll]* *[NA([Peter];[NA([Mary];[John])])] [and][Paul]*
7.	({SHEAD|CHEAD})(N,{NOU|PPN},%a)(C,and, CCJ,%b)(N,{NOU|PPN},%c)({STAIL|CTAIL}):=(NA(%c;%a),+N,+AND);	Here, *%a* refers to node [NA([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])], *%b* refers to node [ਅਤੇ], and *%c* refers to node [ਪਾੱਲ]. This rule resolves a relation 'NA' whose first and second arguments are *%c* and *%a* respectively. The new node so formed is given attributes 'N', and 'AND'.	Original nodes: [NA([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])][ਅਤੇ][ਪਾੱਲ] *[NA([pīṭar];[NA([mairī];[jān])])] [atē][pāll]* *[NA([Peter];[NA([Mary];[John])])] [and][Paul]* Resultant nodes: [NA([ਪਾੱਲ];[NA([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])])] *[NA([pāll];[NA([pīṭar];[NA([mairī];[jān])])])]*

			[NA([Paul];[NA([Peter];[NA([Mary];[John])])])]
8.	(NA(%a;%b),AND):=and(%a;%b);	Here, *%a* refers to node [ਪਾੱਲ], *%b* refers to node [NA([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])]. This rule changes the name of relation from 'NA' to 'and' keeping same arguments as in original node, as required in the final UNL.	Original nodes: [NA([ਪਾੱਲ];[NA([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])])] *[NA([pāll];[NA([pīṭar];[NA([mairī];[jān])])])]* *[NA([Paul];[NA([Peter];[NA([Mary];[John])])])]* Resultant nodes: [and([ਪਾੱਲ];[NA([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])])] *[and([pāll];[NA([pīṭar];[NA([mairī];[jān])])])]* *[and([Paul];[NA([Peter];[NA([Mary];[John])])])]*
9.	(NA(%a;%b),AND):=and(%a;%b);	Here, *%a* refers to node [ਪੀਟਰ], *%b* refers to node [NA([ਮੈਰੀ];[ਜਾਨ])]. This rule changes the name of relation from 'NA' to 'and' keeping same arguments as in original node, as required in the final UNL.	Original nodes: [and([ਪਾੱਲ];[NA([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])])] *[and([pāll];[NA([pīṭar];[NA([mairī];[jān])])])]* *[and([Paul];[NA([Peter];[NA([Mary];[John])])])]* Resultant nodes: [and([ਪਾੱਲ];[and([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])])] *[and([pāll];[and([pīṭar];[NA([mairī];[jān])])])]* *[and([Paul];[and([Peter];[NA([Mary];[John])])])]*
10.	(NA(%a;%b),AND):=and(%a;%b);	Here, *%a* refers to node [ਮੈਰੀ], *%b* refers to node [ਜਾਨ]. This rule changes the name of relation from 'NA' to 'and' keeping same arguments as in original node, as required in the final UNL.	Original nodes: [and([ਪਾੱਲ];[and([ਪੀਟਰ];[NA([ਮੈਰੀ];[ਜਾਨ])])])] *[and([pāll];[and([pīṭar];[NA([mairī];[jān])])])]* *[and([Paul];[and([Peter];[NA([Mary];[John])])])]*

			Resultant nodes: [and([ਪਾੱਲ];[and([ਪੀਟਰ];[and([ਮੈਰੀ];[ਜਾਨ])])])] *[and([pāll];[and([pīṭar];[and([mairī];[jān])])])]* *[and([Paul];[and([Peter];[and([Mary];[John])])])]* Now all the natural language words are replaced by their universal words. Also, the inner relations are replaced by their scopes which are generated internally by IAN. Note that, here internal relations are not resolved. Final output is generated by IAN as shown in (5.19).

UNL generated by IAN is given in (5.19).

{org}

ਜਾਨ, ਮੈਰੀ, ਪੀਟਰ ਅਤੇ ਪਾੱਲ

{/org}

{unl}

and(Paul:09, :04)

and:04(Peter:05, :05)

and:05(Mary:03, John:01)

{/unl} ...(5.19)

Here :09, :04, :05, :03, :01 are the scopes internally generated by the IAN tool.

5.2.6 UNL-ization of Determiners

Determiners are Noun-modifiers which expresses the reference of a noun or noun-phrase in the context, including quantity. For example ‘my’, ‘your’, ‘this’, ‘that’ etc. The UNL-ization process for ordinals has been illustrated with the help of a simple example sentence (5.20).

Example 1: ਉਹਦੀਆਂ ਬਹੁਤ ਸਾਰੀਆਂ ਕਿਤਾਬਾਂ ...(5.20)

uhdīāṃ bahut sārīāṃ kitābāṃ

His many books

After the tokenization of example sentence given in (5.20) with IAN tool, five lexical items are identified as shown in (5.21).

[ਉਹਦੀਆਂ]{}"00.@3.@male"(LEX=D,POS=POD)<pan,0,0>;

[ਬਹੁਤ ਸਾਰੀਆਂ]{}"many"(LEX=D,POS=QUA,att =@multal)<pan,0,0>;

[ਕਿਤਾਬਾਂ]{}"book"(LEX=N,POS=NOU,NUM=PLR)<pan,0,0>;

Two blank spaces are also identified as:-

[]{}""(BLK)<pan,0,0>; ...(5.21)

Here, LEX=D implies that node's lexical category is determiner , 'POS=QUA' implies that part-of-speech is quantity, 'POS=POD' implies that part-of-speech is a possessive determiner. Universal word 'many' is associated with an attribute '@multal'. The process of UNL-ization of example sentence (5.20) has been illustrated in Table 5.7.

Table 5.7: UNL-ization process for example sentence (5.20)

S.No	TRule fired	Description	Action Taken
1.	(%a,BLK):=;	Here, *%a* refers to blank node []. This rule is fired twice and deletes all the blank spaces.	Original nodes : [ਉਹਦੀਆਂ][][ਬਹੁਤ ਸਾਰੀਆਂ][][ਕਿਤਾਬਾਂ] *[uhdīāṃ][][bahut sārīāṃ][][kitābāṃ]* *[His][][many][][books]* Resultant nodes: [ਉਹਦੀਆਂ][ਬਹੁਤ ਸਾਰੀਆਂ][ਕਿਤਾਬਾਂ] *[uhdīāṃ][bahut sārīāṃ][kitābāṃ]* *[His][many][books]*
2.	(D,att,%a)(N,%b):=(%b,+att=%a);	Here, *%a* refers to node [ਬਹੁਤ ਸਾਰੀਆਂ], *%b* refers to node [ਕਿਤਾਬਾਂ]. This rule deletes the node *%a* and gives its attributes to node *%b*.	Original nodes : [ਉਹਦੀਆਂ][ਬਹੁਤ ਸਾਰੀਆਂ][ਕਿਤਾਬਾਂ] *[uhdīāṃ][bahut sārīāṃ][kitābāṃ]* *[His][many][books]*

			Resultant nodes: [ਉਹਦੀਆਂ][ਕਿਤਾਬਾਂ@multal] *[uhdīāṃ][kitābāṃ@multal]* *[His][books@multal]*
3.	(N,PLR,^@pl, %a):=(%a, +@pl);	Here, *%a* refers to node [ਕਿਤਾਬਾਂ]. This rule gives attribute '@pl' to this node.	Original nodes: [ਉਹਦੀਆਂ][ਕਿਤਾਬਾਂ@multal] *[uhdīāṃ][kitābāṃ@multal]* *[His][books@multal]* Resultant nodes: [ਉਹਦੀਆਂ][ਕਿਤਾਬਾਂ@multal@pl] *[uhdīāṃ][kitābāṃ@multal@pl]* *[His][books@multal@pl]*
4.	(D,POD,%a)(%b):=(NA(%b;%a),+POS,%d);	Here, *%a* refers to node [ਉਹਦੀਆਂ], *%b* refers to node [ਕਿਤਾਬਾਂ@multal@pl]. This rule resolves a relation 'NA' whose first and second argument are *%b* and *%a* respectively. This new node so formed is given an attribute 'POS'.	Original nodes: [ਉਹਦੀਆਂ][ਕਿਤਾਬਾਂ@multal@pl] *[uhdīāṃ][kitābāṃ@multal@pl]* *[His][books@multal@pl]* Resultant nodes: [NA([ਕਿਤਾਬਾਂ@multal@pl];[ਉਹਦੀਆਂ])] *[NA([kitābāṃ@multal@pl];[uhdīāṃ])]* *[NA([books@multal@pl];[uhdīāṃ])]*
5.	(NA(%a;%b),POS):=pos(%a;%b);	Here, *%a* refers to node [ਕਿਤਾਬਾਂ@multal@pl], and *%b* refers to node [ਉਹਦੀਆਂ]. This rule changes the name of relation from 'NA' to 'pos' keeping same arguments as in original node, as required in the final UNL.	Original nodes: [NA([ਕਿਤਾਬਾਂ@multal@pl];[ਉਹਦੀਆਂ])] *[NA([kitābāṃ@multal@pl];[uhdīāṃ])]* *[NA([books@multal@pl];[uhdīāṃ])]* Resultant nodes: [pos([ਕਿਤਾਬਾਂ@multal@pl];[ਉਹਦੀਆਂ])] *[pos([kitābāṃ@multal@pl];[uhdīāṃ])]*

			Now all the natural language words are replaced by their universal words and final output is generated by IAN as shown in (5.22).

UNL generated by IAN is given in (5.22).

{org}

ਉਹਦੀਆਂ ਬਹੁਤ ਸਾਰੀਆਂ ਕਿਤਾਬਾਂ

{/org}

{unl}

pos(book:05.@multal.@pl, 0:01.@3.@male)

{/unl} …(5.22)

Here, :01, :05 are the scopes internally generated by the IAN tool.

5.2.7 UNL-ization of Time

UNL-ization of time is almost similar as that of numbers because like numbers, their UNL representation is realistic, *i.e.*, without the use of any relation or attribute in final output. The UNL-ization process for time has been illustrated with the help of a simple example sentence (5.23).

Example 1: ਪੰਦਰਾ ਮਿੰਟ ...(5.23)

Pandrā miṇṭ

fifteen minutes

After the tokenization of example sentence given in (5.23) with IAN tool, three lexical items are identified as shown in (5.24).

[ਪੰਦਰਾ]{}"15"(LEX=U,POS=CDN,DOUBLE)<pan,0,0>;

[]{}""(BLK)<pan,0,0>;

[ਮਿੰਟ]{-1}"ਮਿੰਟ"(TEMP)<xxx,0,0>; ...(5.24)

The process of UNL-ization of example sentence (5.23) has been illustrated in Table 5.8.

Table 5.8: UNL-ization process for example sentence (5.23)

S.No	TRule fired	Description	Action Taken
1.	(%a,BLK):=;	Here, *%a* refers to blank node []. This rule deletes the blank	Original nodes : [ਪੰਦਰਾ][][ਮਿੰਟ]
		spaces.	*[Pandrā][][miṇṭ]* *[Fifteen][][minutes]* Resultant nodes: [ਪੰਦਰਾ][ਮਿੰਟ] *[Pandrā][miṇṭ]* *[Fifteen][minutes]*
2.	(U,CDN,{MYTIME|DOUBLE},%a)({"ਮਿੰਟ"|MINUTESMZ},%b):=(%a,MINUTES);	Here, *%a* refers to node [ਪੰਦਰਾ], *%b* refers to node [ਮਿੰਟ]. This rule deletes the node *%b* and gives an attribute 'MINUTES' to node *%a*.	Original nodes : [ਪੰਦਰਾ][ਮਿੰਟ] *[Pandrā][miṇṭ]* *[Fifteen][minutes]* Resultant nodes: [ਪੰਦਰਾ] *[Pandrā]* *[Fifteen]*
3.	(MINUTES,%a)({STAIL|CTAIL},^SECONDS,^COLON1,^COLON2,%c):=(%a)([[00]],[00],"00",SECONDS)(%c);	Here, *%a* refers to node [ਪੰਦਰਾ], *%c* refers to STAIL. This rule adds a node [00] after node *%a i.e.* before STAIL as shown in Action Taken column and is given an attribute 'SECONDS'.	Original nodes: [ਪੰਦਰਾ] *[Pandrā]* *[Fifteen]* Resultant nodes: [ਪੰਦਰਾ][00] *[Pandrā][00]* *[Fifteen][00]*
4.	({SHEAD|CHEAD},^HOURS,^COLON1,^COLON2,%a)(MINUTES,%b):=(%a)([[00]],[00],"00",HOURS)(%b);	Here, *%a* refers to SHEAD, and *%b* refers to node [ਪੰਦਰਾ]. This rule adds a node [00] after SHEAD and before *%b* as shown in Action Taken column. This new node is given an attribute 'HOURS'.	Original nodes: [ਪੰਦਰਾ][00] [*Pandrā*][00] [*Fifteen*][00] Resultant nodes: [00][ਪੰਦਰਾ][00] *[00][Pandrā][00]* *[00][Fifteen][00]*
5.	(HOURS,%a)(MINUTES,%b)(SECONDS,%c):=(%a)([[:]],[:],":",COLON1,%x)(	Here, *%a* refers to node [00] having attribute 'HOURS', *%b* refers to node [ਪੰਦਰਾ], and *%c* refers to node [00] having attribute 'SECONDS'. This rule adds node [:], being	Original nodes: [00][ਪੰਦਰਾ][00] *[00][Pandrā][00]* *[00][Fifteen][00]* Resultant nodes:

	%b)([[:]],[:],":",COLON2,%y)(%c);	referred as *%x*, between *%a* and *%b*, and node [:], being referred as *%y*, between *%b* and *%c*. The node *%x* is given an attribute 'COLON1', while node *%y* is given attribute 'COLON2'.	[00][:][ਪੰਦਰਾ][:][00] *[00][:][Pandrā][:][00]* *[00][:][Fifteen][:][00]*
6.	(HOURS,%a)(COLON1,%b)(MINUTES,%c)(COLON2,%d)(SECONDS,%e):=(%a&%b&%c&%d&%e,-COLON1,-COLON2,-MINUTES,-SECONDS,-HOURS,-MYTIME,-TIME_DIGIT,-HOURSMZ,-MINUTESMZ);	Here, *%a* refers to node [00] having attribute 'HOURS', *%b* refers to node [:] having attribute'COLON1', *%c* refers to node [ਪੰਦਰਾ], *%d* refers to node [:] having attribute 'COLON2', and *%d* refers to node [00] having attribute 'SECONDS'. This rule concatenates all nodes to form a new hyper node as shown in Action Taken column. All the surplus attributes are removed from this hyper node.	Original nodes: [00][:][ਪੰਦਰਾ][:][00] *[00][:][Pandrā][:][00]* *[00][:][Fifteen][:][00]* Resultant nodes: [[00][:][ਪੰਦਰਾ][:][00]] *[[00][:][Pandrā][:][00]]* *[[00][:][Fifteen][:][00]]* Now the natural language word 'ਪੰਦਰਾ' is replaced by its universal word and final output is generated by IAN as shown in (5.25).

UNL generated by IAN is given in (5.25).

{org}

ਪੰਦਰਾ ਮਿੰਟ

{/org}

{unl}

[w] "00:15:00":08 [/w]

{/unl} ...(5.25)

Here, :08 is the scope internally generated by the IAN tool.

5.2.8 UNL-ization of Verbs

Verbs are an important constituent of any natural language. They describe an action or occurrence or indicate a state of being like 'run', 'eat', 'sleeping' *etc*. The UNL-ization process for verbs has been illustrated with the help of a simple example sentence (5.26).

Example 1: ਉਸਨੇ ਉਹਨੂੰ ਕੱਲ੍ਹ ਮਾਰ ਦਿੱਤਾ ...(5.26)

usnē uhnūṃ kallh mār dittā

He killed him yesterday

After the tokenization of example sentence given in (5.26) with IAN tool, seven lexical items are identified as shown in (5.27).

[ਉਸਨੇ]{}"00.@3.@male"(LEX=R,POS=PPR,CAS=NOM,PER=3PS)<pan,0,0>;

[ਉਹਨੂੰ]{}"00.@3.@male" (LEX=R,POS=PPR,CAS=NOM,PER=3PS)<pan,0,0>;

[ਕੱਲ੍ਹ]{}"yesterday"(LEX=A,POS=AAV,SEM=TME)<pan,0,0>;

[ਮਾਰ ਦਿੱਤਾ]{}"kill"(LEX=V,POS=VER,TRA=TSTD,GEN=MCL,ATE=PAS,NUM=SNG)<pan,0,0>;

Three blank spaces are also identified as :-

[ ]{}""(BLK)<pan,0,0>; ...(5.27)

Here, 'V', 'R' indicates that lexical category of those nodes are verb and pronoun respectively, TRA=TSTD indicates that verb is direct transitive, 'NPR' means that pronoun is an indefinite pronoun whereas 'COP' means that the verb is a linking verb which is used to link the subject of a sentence with a predicate. 'SEM=TME' means that semantics of a node is time. The process of UNL-ization of example sentence (5.26) has been illustrated in Table 5.9.

Table 5.9: UNL-ization process for example sentence (5.26)

S.No	TRule fired	Description	Action Taken
1.	(%a,BLK):=;	Here, *%a* refers to blank node []. This rule is fired thrice and deletes all the blank spaces.	Original nodes : [ਉਸਨੇ][][ਉਹਨੂੰ][][ਕੱਲ੍ਹ][] [ਮਾਰ ਦਿੱਤਾ] *[usnē][][uhnūṃ][][kallh] [][mār dittā]* Resultant nodes: [ਉਸਨੇ][ਉਹਨੂੰ][ਕੱਲ੍ਹ][ਮਾਰ ਦਿੱਤਾ] *[usnē][uhnūṃ][kallh][mār dittā]*
2.	(V,PAS,^@past,%a):=(%a,+att=@past);	Here, *%a* refers to node [ਮਾਰ ਦਿੱਤਾ]. This rule gives an attribute '@past' to node *%a*.	Original nodes : [ਉਸਨੇ][ਉਹਨੂੰ][ਕੱਲ੍ਹ][ਮਾਰ ਦਿੱਤਾ]

			[usnē] [uhnūṃ] [kallh] [mār dittā] Resultant nodes: [ਉਸਨੇ][ਉਹਨੂੰ][ਕੱਲ੍ਹ][ਮਾਰ ਦਿੱਤਾ @past] *[usnē] [uhnūṃ] [kallh] [mār dittā@past]*
3.	(A,%a)(V,TRA=TSTD,%b):=(NA(%b;%a),+V,+TRA=TSTD,+TIM);	Here, *%a* refers to node [ਕੱਲ੍ਹ], *%b* refers to node [ਮਾਰ ਦਿੱਤਾ@past]. This rule creates a relation 'NA' between nodes *%b*, and *%a* as its first and second argument respectively.	Original nodes: [ਉਸਨੇ][ਉਹਨੂੰ][ਕੱਲ੍ਹ][ਮਾਰ ਦਿੱਤਾ@past] *[usnē] [uhnūṃ] [kallh] [mār dittā@past]* Resultant nodes: [ਉਸਨੇ][ਉਹਨੂੰ][NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])] *[usnē] [uhnūṃ] [NA([mār dittā];[kallh])]*
4.	(R,PPR,%a)(V,TSTD,%b):=(NA(%b;%a),+OBJ,V,TSTD,+GEN=%b,+NUM=%b);	Here, *%a* refers to node [ਉਹਨੂੰ], *%b* refers to node [NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])] .This rule resolves a relation 'NA' with *%b* and *%a* as first and second argument respectively. The final node is given attribute 'OBJ' and this node is treated as verb.	Original nodes: [ਉਸਨੇ][ਉਹਨੂੰ][NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])] *[usnē] [uhnūṃ] [NA([mār dittā@past];[kallh])]* Resultant nodes: [ਉਸਨੇ] [NA([NA([ਮਾਰ ਦਿੱਤਾ @past];[ਕੱਲ੍ਹ])]; [ਉਹਨੂੰ])] *[usnē]* [NA([NA([*mār dittā* @past];[*kallh*])];[*uhnūṃ*])]
5.	({SHEAD|CHEAD},%z)(R,{CPR|PPR},%a)(V,TSTD,%b):=(NA(%b;%a),+AGT,V,TSTD);	Here, *%a* refers to SHEAD, *%b* refers to node [ਉਸਨੇ], and *%c* refers to node [NA([NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])]; [ਉਹਨੂੰ])] . This rule resolves a relation 'NA' with *%b* and *%a* as first and second argument respectively. The final node is given attribute 'AGT' and this node is treated as verb.	Original nodes: [ਉਸਨੇ] [NA([NA([ਮਾਰ ਦਿੱਤਾ @past];[ਕੱਲ੍ਹ])]; [ਉਹਨੂੰ])] *[usnē]* [NA([NA([*mār dittā* @past];[*kallh*])]; [*uhnūṃ*])] Resultant nodes: [NA([NA([NA([ਮਾਰ ਦਿੱਤਾ @past];[ਕੱਲ੍ਹ])];[ਉਹਨੂੰ])];[ਉਸਨੇ])] *[NA([NA([NA([mār dittā @past];[kallh])];[uhnūṃ])];[usnē])]*
6.	(NA(NA(%a;%b),OBJ,	Here, *%a* refers to node	Original nodes:

	%w;%c),AGT,%r):=(%w),(NA(%a;%c),%e,+AGT);	[NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])], *%b* refers to node [ਉਹਨੂੰ], *%c* refers to node [ਉਸਨੇ], *%w* [NA([NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])]; [ਉਹਨੂੰ])] , and *%r* refers to node [NA([NA([NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])];[ਉਹਨੂੰ])];[ਉਸਨੇ])]. This rule splits the node into two nodes *%w* and *%e*. Node *%e* holds relation NA with *%a* and *%c* as its first and second argument respectively.	[NA([NA([NA([ਮਾਰ ਦਿੱਤਾ @past];[ਕੱਲ੍ਹ])];[ਉਹਨੂੰ])];[ਉਸਨੇ])] *[NA([NA([NA([mār dittā @past];[kallh])];[uhnūṃ])];[usnē])]* Resultant nodes: Node 1: [NA([NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])];[ਉਹਨੂੰ])] *[NA([NA([mār dittā@past];[kallh])];[uhnūṃ])]* Node 2: [NA([NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])];[ਉਸਨੇ])] *[NA([NA([mār dittā@past];[kallh])];[usnē])]*
7.	(NA(NA(%a;%b),TIM,%w;%c),OBJ,%r):=(%w),(NA(%a;%c),%k,+OBJ);	Here, *%a* refers to node [ਮਾਰ ਦਿੱਤਾ@past], *%b* refers to node [ਕੱਲ੍ਹ], *%w* refers to node [NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])], *%c* refers to node [ਉਹਨੂੰ], and *%r* refers to node [NA([NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])];[ਉਹਨੂੰ])]. This rule splits node into *%w* and *%k.* Node *%k* holds relation NA with *%a* and *%c* as its first and second argument respectively.	Original Nodes: [NA([NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])];[ਉਹਨੂੰ])] *[NA([NA([mār dittā@past];[kallh])];[uhnūṃ])]* Resultant Nodes: Node 1: [NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])] *[NA([mār dittā@past];[kallh])]* Node 2: [NA([ਮਾਰ ਦਿੱਤਾ@past];[ਉਹਨੂੰ])] *[NA([mār dittā@past];[uhnūṃ])]*
8.	(NA(NA(%a;%b),TIM,%w;%c),AGT,%r):=(%w),(NA(%a;%c),%k,+AGT);	Here, *%a* refers to node [ਮਾਰ ਦਿੱਤਾ@past], *%b* refers to node [ਕੱਲ੍ਹ], *%w* refers to node [NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])], *%c* refers to node [ਉਸਨੇ], and *%r* refers to node	Original Nodes: [NA([NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])];[ਉਸਨੇ])] *[NA([NA([mār dittā@past];[kallh])];[usnē])]* Resultant Nodes: Node 1:

		[NA([NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])];[ਉਸਨੇ])]. This rule splits node into *%w* and *%k.* Node *%k* holds relation NA with *%a* and *%c* as its first and second argument respectively.	[NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])] *[NA([mār dittā@past];[kal lh])]* Node 2: [NA([ਮਾਰ ਦਿੱਤਾ@past];[ਉ ਸਨੇ])] *[NA([mār dittā@past];[us nē])]*
9.	(NA(%a;%b),AGT):=agt(%a;%b);	Here, *%a* refers to node [NA(ਮਾਰ ਦਿੱਤਾ@past)], and *%b* refers to node [ਉਸਨੇ]. This rule changes the name of relation from 'NA' to 'agt' as required in the final UNL.	Original Nodes: [NA([ਮਾਰ ਦਿੱਤਾ@past];[ਉ ਸਨੇ])] *[NA([mār dittā@past];[us nē])]* Resultant Nodes: [agt([ਮਾਰ ਦਿੱਤਾ@past];[ਉ ਸਨੇ])] *[agt([mār dittā@past];[us nē])]*
10.	(NA(%a;%b),OBJ):=obj(%a;%b);	Here, *%a* refers to node [NA(ਮਾਰ ਦਿੱਤਾ@past)], and *%b* refers to node [ਉਹਨੂੰ]. This rule changes the name of relation from 'NA' to 'obj' as required in the final UNL.	Original Nodes: [NA([ਮਾਰ ਦਿੱਤਾ@past];[ਉਹ ਨੂੰ])] *[NA([mār dittā@past];[uhnūṃ])]* Resultant Nodes: [obj([ਮਾਰ ਦਿੱਤਾ@past];[ਉ ਹਨੂੰ])] *[obj([mār dittā@past];[uhnūṃ])]*
11.	(NA(%a;%b),TIM):=tim(%a;%b);	Here, *%a* refers to node [NA(ਮਾਰ ਦਿੱਤਾ@past)], and *%b* refers to node [ਕੱਲ੍ਹ]. This rule changes the name of relation from 'NA' to 'tim' as required in the final UNL.	Original Nodes: [NA([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱ ਲ੍ਹ])] *[NA([mār dittā@past];[kallh])]* Resultant Nodes: [tim([ਮਾਰ ਦਿੱਤਾ@past];[ਕੱਲ੍ਹ])] *[tim([mār dittā@past];[kallh])]* Now all the natural

			language words are replaced by their universal words and final output is generated by IAN as shown in (5.28).

UNL generated by IAN is given in (5.28).

{org}

ਉਸਨੇ ਉਹਨੂੰ ਕੱਲ੍ਹ ਮਾਰ ਦਿੱਤਾ

{/org}

{unl}

agt(kill:07.@past, 00:01.@3.@male)

obj(kill:07.@past, 00:03.@3.@male)

tim(kill:07.@past, yesterday:05)

{/unl} ...(5.28)

Here, :07 is the scope internally generated by the IAN tool.

5.2.9 UNL-ization of Nouns and Adjectives

The main role of an adjective is to assign attributes to a noun. Adjectives are different from determiners, which express references rather than qualities. The UNL-ization process for nouns and adjectives has been illustrated with the help of a simple example sentence (5.29).

Example 1: ਇਕ ਸੋਹਣੀ ਗੱਡੀ, ਇਕ ਮਹਿੰਗੀ ਗੱਡੀ ਅਤੇ ਇਕ ਨਵਾਂ ਪਿਆਲਾ ...(5.29)

ik sōhṇī gaḍḍī, ik mahiṅgī gaḍḍī atē ik navāṃ piālā

A beautiful car, a expensive car and a new mug.

After the tokenization of example sentence given in (5.29) with IAN tool, twenty lexical items are identified as shown in (5.30).

[ਸੋਹਣੀ]{}"beautiful"(LEX=J,POS=ADJ,GEN=FEM)<pan,0,0>;

[ਗੱਡੀ]{}"car"(LEX=N,POS=NOU,NUM=SNG)<p an,0,0>;

[,]{}""(LEX=C,POS=COO,rel=and)<pan,0,0>;

[ਮਹਿੰਗੀ]{}"expensive"(LEX=J,POS=ADJ)<pan,0,0>;

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

[ਅਤੇ]{}"and"(LEX=C,POS=COO,rel=and)<pan,0,0>;

[ਨਵਾਂ]{}"new"(LEX=J,POS=ADJ)<pan,0,0>;

[ਪਿਆਲਾ]{}"mug"(LEX=N,POS=NOU,NUM=SNG)<pan,0,0>;

Three nodes are identified as :-

[ਇਕ]{}""(LEX=D,POS=ART,att=@indef)<pan,0,0>;

Nine blank spaces are also identified as :-

[]{}""(BLK)<pan,0,0>; ...(5.30)

Here, ‘J’, ‘ADJ’ represents lexical category and part of speech respectively as adjective, ‘FEM’ represents gender of the node as female, and ‘ART’ indicates that determiner is an article. Articles are used to express definiteness like ‘a’, ‘the’ *etc*. The process of UNL-ization of example sentence (5.29) has been illustrated in Table 5.10.

Table 5.10: UNL-ization process for example sentence (5.29)

S.No	TRule fired	Description	Action Taken
1.	(%a,BLK):=;	Here, *%a* refers to blank node []. This rule is fired nine times and deletes all the blank spaces.	Original nodes : [ਇਕ][][ਸੋਹਣੀ][][ਗੱਡੀ][,][][ਇਕ][][ਮਹਿੰਗੀ][][ਗੱਡੀ][][ਅਤੇ][][ਇਕ][][ਨਵਾਂ][][ਪਿਆਲਾ] *[ik][][sōhṇī][][gaḍḍī][,][][ik][][mahiṅgī][][gaḍḍī][][atē][][ik][][navāṃ][][piālā]* *[A][][beautiful][][car][,][ ][a][expensive][][car][][and][][a][][new][][mug]* Resultant nodes : [ਇਕ][ਸੋਹਣੀ][ਗੱਡੀ][,][ਇਕ][ਮਹਿੰਗੀ][ਗੱਡੀ][ਅਤੇ][ਇਕ][ਨਵਾਂ][ਪਿਆਲਾ] *[ik][sōhṇī][gaḍḍī][,][ik][mahiṅgī][gaḍḍī][atē][ik][navāṃ][piālā]*

			[A] [beautiful] [car] [,] [a] [expensive] [car] [and] [a] [new] [mug]
2.	(D,att,%a)(J,%b)(N,%c):=(NA(%c,+att=%a;%b),+N,+NOU,+MOD);	Here, *%a* refers to node [ਇਕ], *%b* refers to node [ਸੋਹਣੀ], and node *%c* refers to [ਗੱਡੀ]. This rule resolves a relation 'NA' whose first and second argument are *%c* and *%b* respectively. The attributes of node *%a* are given to first argument of 'NA' relation. This new node is given attributes 'N', 'NOU', 'MOD'.	Original nodes : [ਇਕ][ਸੋਹਣੀ][ਗੱਡੀ][,][ਇਕ][ਮਹਿੰਗੀ][ਗੱਡੀ][ਅਤੇ][ਇਕ][ਨਵਾਂ][ਪਿਆਲਾ] *[ik] [sōhṇī] [gaḍḍī] [,] [ik] [mahiṅgī] [gaḍḍī] [atē] [ik] [navāṃ] [piālā]* *[A] [beautiful] [car] [,] [a] [expensive] [car] [and] [a] [new] [mug]* Resultant nodes: [NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])][,][ਇਕ][ਮਹਿੰਗੀ][ਗੱਡੀ][ਅਤੇ][ਇਕ][ਨਵਾਂ][ਪਿਆਲਾ] *[NA([gaḍḍī@indef];[sōhṇī])] [,] [ik] [mahiṅgī] [gaḍḍī] [atē] [ik] [navāṃ] [piālā]* *[NA([car@indef];[beautiful])] [,] [a] [expensive] [car] [and] [a] [new] [mug]*
3.	(D,att,%a)(J,%b)(N,%c):=(NA(%c,+att=%a;%b),+N,+NOU,+MOD);	Here, *%a* refers to node [ਇਕ], *%b* refers to node [ਮਹਿੰਗੀ], and node *%c* refers to [ਗੱਡੀ]. This rule resolves a relation 'NA' whose first and second argument are *%c* and *%b* respectively. The attributes of node *%a* are given to first argument of 'NA' relation. This new node is given attributes 'N', 'NOU', 'MOD'.	Original nodes: [NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])][,][ਇਕ][ਮਹਿੰਗੀ][ਗੱਡੀ][ਅਤੇ][ਇਕ][ਨਵਾਂ][ਪਿਆਲਾ] *[NA([gaḍḍī@indef];[sōhṇī])] [,] [ik] [mahiṅgī] [gaḍḍī] [atē] [ik] [navāṃ] [piālā]* *[NA([car@indef];[beautiful])] [,] [a] [expensive] [car] [and] [a] [new] [mug]* Resultant nodes: [NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])

			][,][NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])][ਅਤੇ][ਇਕ][ਨਵਾਂ][ਪਿਆਲਾ] *[NA([gaḍḍī@indef];[sōhṇī])][,][NA([gaḍḍī@indef];[mahiṅgī])][atē][ik][navāṃ][piālā]* *[NA([car@indef];[beautiful])][,][NA([car@indef];[expensive])][and][a][new][mug]*
4.	(N,NOU,%a)(C,%b)(N,NOU,%c):=(NA(%c;%a),+N,+NOU,+AND);	Here, *%a* refers to node [NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])], *%b* refers to node [,] and *%c* refers to node [NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])]. This rule resolves a relation 'NA' whose first and second argument are *%c* and *%a* respectively. This new node so formed is given an attribute 'N', 'NOU', and 'AND'.	Original nodes: [NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])][,][NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])][ਅਤੇ][ਇਕ][ਨਵਾਂ][ਪਿਆਲਾ] *[NA([gaḍḍī@indef];[sōhṇī])][,][NA([gaḍḍī@indef];[mahiṅgī])][atē][ik][navāṃ][piālā]* *[NA([car@indef];[beautiful])][,][NA([car@indef];[expensive])][and][a][new][mug]* Resultant nodes: [NA([NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])][ਅਤੇ][ਇਕ][ਨਵਾਂ][ਪਿਆਲਾ] *[NA([NA([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])][atē][ik][navāṃ][piālā]* *[NA([NA([car@indef];[expensive])];[NA([car@indef];[beautiful])])][and][a][new][mug]*
5.	(D,att,%a)(J,%b)(N,%c)	Here, *%a* refers to node [ਇਕ],	Original nodes:

	:=(NA(%c, +att=%a;%b),+N,+NOU,+MOD);	*%b* refers to node [ਨਵਾਂ], and node *%c* refers to [ਪਿਆਲਾ]. This rule resolves a relation 'NA' whose first and second argument are *%c* and *%b* respectively. The attributes of node *%a* are given to first argument of 'NA' relation. This new node is given attributes 'N', 'NOU', 'MOD'.	[NA([NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])][ਅਤੇ][ਇਕ][ਨਵਾਂ][ਪਿਆਲਾ] *[NA([NA([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī @indef];[sōhṇī])])][atē][ik][na vāṃ][piālā]* *[NA([NA([car@indef];[expensive])];[NA([car@indef];[beautiful])])][and][a][new][mug]* Resultant nodes: [NA([NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])][ਅਤੇ][NA([ਪਿਆਲਾ@indef];[ਨਵਾਂ])] *[NA([NA([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])][atē][NA([piālā@indef];[navāṃ])]* *[NA([NA([car@indef];[expensive])];[NA([car@indef];[beautiful])])][and][NA([mug@indef];[new])]*
6.	(N,NOU,%a)(C,%b)(N,NOU,%c):=(NA(%c;%a),+N,+NOU,+AND);	Here, *%a* refers to node [NA([NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])], *%b* refers to node [ਅਤੇ] and *%c* refers to node [NA([ਪਿਆਲਾ@inde f];[ਨਵਾਂ])]. This rule resolves a relation 'NA' whose first and second argument are *%c* and *%a* respectively. This new node so formed is given an attribute 'N', 'NOU', and 'AND'.	Original nodes: [NA([NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])][ਅਤੇ][NA([ਪਿਆਲਾ@indef];[ਨਵਾਂ])] *[NA([NA([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])][atē][NA([piālā@indef];[navāṃ])]* *[NA([NA([car@indef];[expensive])];[NA([car@indef];[beautiful])])][and][NA([*

			mug@indef];[new])] Resultant nodes: [NA([NA([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])])] *[NA([NA([piālā@indef];[navāṃ])];[NA([NA([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])])]* *[NA([NA([mug@indef];[new])];[NA([NA([car@indef];[expensive])];[NA([car@indef];[beautiful])])])]*
7.	(NA(%a;%b),MOD):=mod(%a;%b);	Here, *%a* refers to node [ਪਿਆਲਾ@indef], *%b* refers to [ਨਵਾਂ]. This rule changes the name of relation from 'NA' to 'mod' keeping same arguments as in original node, as required in the final UNL.	Original nodes: [NA([NA([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])])] *[NA([NA([piālā@indef];[navāṃ])];[NA([NA([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])])]* *[NA([NA([mug@indef];[new])];[NA([NA([car@indef];[expensive])];[NA([car@indef];[beautiful])])])]* Resultant nodes: [NA([mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ@ indef];[ਸੋਹਣੀ])])])] *[NA([mod([piālā@indef];[navāṃ])];[NA([NA([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])])]* *[NA([mod([mug@indef];[n*

			ew])];[NA([NA([car@indef];[expensive])];[NA([car @indef];[beautiful])])])]
8.	(NA(%a;%b),MOD):=mod(%a;%b);	Here, *%a* refers to node [ਗੱਡੀ@indef], *%b* refers to [ਸੋਹਣੀ]. This rule changes the name of relation from 'NA' to 'mod' keeping same arguments as in original node, as required in the final UNL.	Original nodes: [NA([mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ @i ndef];[ਸੋਹਣੀ])])])] *[NA([mod([piālā@indef];[navāṃ])];[NA([NA([gaḍḍī @indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])])]* *[NA([mod([mug@indef];[new])];[NA([NA([car@indef];[expensive])];[NA([car @indef];[beautiful])])])]* Resultant nodes: [NA([mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ @indef];[ਸੋਹਣੀ])])])] *[NA([mod([piālā@indef];[navāṃ])];[NA([mod([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])])]* *[NA([mod([mug@indef];[new])];[NA([mod([car@indef];[expensive])];[NA([car @indef];[beautiful])])])]*
9.	(NA(%a;%b),MOD):=mod(%a;%b);	Here, *%a* refers to node [ਗੱਡੀ@indef], *%b* refers to [ਮਹਿੰਗੀ]. This rule changes the name of relation from 'NA' to 'mod' keeping same arguments as in original node, as required in the final UNL.	Original nodes: [NA([mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[NA([ਗੱਡੀ @indef];[ਸੋਹਣੀ])])])] *[NA([mod([piālā@indef];[*

			navāṃ])];[NA([mod([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])])] *[NA([mod([mug@indef];[new])];[NA([mod([car@indef];[expensive])];[NA([car@indef];[beautiful])])])]* Resultant nodes: [NA([mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[mod([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])])] *[NA([mod([piālā@indef];[navāṃ])];[NA([mod([gaḍḍī@indef];[mahiṅgī])];[mod([gaḍḍī@indef];[sōhṇī])])])]* *[NA([mod([mug@indef];[new])];[NA([mod([car@indef];[expensive])];[mod([car@indef];[beautiful])])])]*
10.	(NA(%a;%b),AND):=and(%a;%b);	Here, *%a* refers to node [mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])], and *%b* refers to node [NA([mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[mod([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])]. This rule changes the name of relation from 'NA' to 'and' keeping same arguments as in original node, as required in the final UNL.	Original nodes: [NA([mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[mod([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])])] *[NA([mod([piālā@indef];[navāṃ])];[NA([mod([gaḍḍī@indef];[mahiṅgī])];[mod([gaḍḍī@indef];[sōhṇī])])])]* *[NA([mod([mug@indef];[new])];[NA([mod([car@indef];[expensive])];[mod([car@indef];[beautiful])])])]* Resultant nodes:

			[and([mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[mod([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])])] *[and([mod([piālā@indef];[navāṃ])];[NA([mod([gaḍḍī@indef];[mahiṅgī])];[mod([gaḍḍī@indef];[sōhṇī])])])]* *[and([mod([mug@indef];[new])];[NA([mod([car@indef];[expensive])];[mod([car@indef];[beautiful])])])]*
11.	(NA(%a;%b),AND):=and(%a;%b);	Here, *%a* refers to node[mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])], and *%b* refers to node [mod([ਗੱਡੀ@indef];[ਸੋਹਣੀ])]. This rule changes the name of relation from 'NA' to 'and' keeping same arguments as in original node, as required in the final UNL.	Original nodes: [and([mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[NA([mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[mod([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])])] *[and([mod([piālā@indef];[navāṃ])];[NA([mod([gaḍḍī@indef];[mahiṅgī])];[mod([gaḍḍī@indef];[sōhṇī])])])]* *[and([mod([mug@indef];[new])];[NA([mod([car@indef];[expensive])];[mod([car@indef];[beautiful])])])]* Resultant nodes: [and([mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])];[and([mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[mod([ਗੱਡੀ@indef];[ਸੋਹਣੀ])])])] *[and([mod([piālā@indef];[navāṃ])];[and([mod([gaḍḍī@indef];[mahiṅgī])];[mod([gaḍḍī@indef];[sōhṇī])])*

			])] *[and([mod([mug@indef];[new])];[and([mod([car@indef];[expensive])];[mod([car@indef];[beautiful])])])]* Now all the natural language words are replaced by their universal words, internal hypernodes are represented by their scopes as shown in final output generated by IAN as given in (5.31).

UNL generated by IAN is given in (5.31).

{org}

ਇਕ ਸੋਹਣੀ ਗੱਡੀ, ਇਕ ਮਹਿੰਗੀ ਗੱਡੀ ਅਤੇ ਇਕ ਨਵਾਂ ਪਿਆਲਾ

{/org}

{unl}

and(:06, :09)

mod:06(mug:0L.@indef, new:0J)

and:09(:07, :08)

mod:07(car:0D.@indef, expensive:0B)

mod:08(car:05.@indef, beautiful:03)

{/unl} ...(5.31)

5.2.10 UNL-ization of Pronouns

A pronoun is a word or form that substitutes for a noun. For example in the sentence "I did it myself", 'myself' is a pronoun. The UNL-ization process for pronouns has been illustrated with the help of a simple example sentence (5.32).

Example 1: ਓੱਹ ਉਸਨੂੰ ਪਿਆਰ ਕਰਦਾ ਹੈ ...(5.32)

ōhh usnūṃ piār karadā hai

He loves her

After the tokenization of example sentence given in (5.32) with IAN tool, Seven lexical items are identified as shown in (5.33).

[ਓੱਹ]{}"00.@3.@male" (LEX=R,POS=PPR,CAS=NOM,PER=3PS)<pan,0,0>;

[ਉਸਨੂੰ]{}"00.@3.@female" (LEX=R,POS=PPR,CAS=OBL,PER=3PS)pan,0,0>;

[ਪਿਆਰ]{}"love"(LEX=V,POS=VER,TRA=TSTD)<pan,0,0>;

[ਕਰਦਾ ਹੈ]{}""(LEX=V,POS=AUX,ATE=PRS,PER=3PS,att=@present,att1=@male,

GEN=MCL)<pan,0,0>;

Three blank spaces are also identified as :-

[]{}""(BLK)<pan,0,0>; ...(5.33)

Here, 'LEX=R' represents lexical category pronoun, 'POS=PPR' represents part-of-speech as personal pronoun, 'CAS=OBL' means that case is oblique, and 'PER=3PS' represents 3rd person. Rest every other tagset has been described in previous examples.The process of UNL-ization of example sentence (5.32) has been illustrated in Table 5.11.

Table 5.11: UNL-ization process for example sentence (5.32)

S.No	TRule fired	Description	Action Taken
1.	(%a,BLK):=;	Here, *%a* refers to blank node []. This rule is fired three times and deletes all the blank spaces.	Original nodes : [ਓੱਹ][][ਉਸਨੂੰ][][ਪਿਆਰ][] [ਕਰਦਾ ਹੈ] *[ōhh][][usnūṃ][][piār][] [karadā hai]* Resultant nodes : [ਓੱਹ][ਉਸਨੂੰ][ਪਿਆਰ][ਕਰਦਾ ਹੈ] *[ōhh][usnūṃ][piār] [karadā hai]*
2.	(V,VER,TSTD,%a)(V,AUX,%b):=(%a,+att=%b,+GEN=%b,+NUM=%b);	Here, *%a* refers to node [ਪਿਆਰ], and *%b* refers to node [ਕਰਦਾ ਹੈ], This rule deletes node *%b* and gives its attributes to node *%a*.	Original nodes : [ਓੱਹ][ਉਸਨੂੰ][ਪਿਆਰ][ਕਰਦਾ ਹੈ] *[ōhh][usnūṃ][piār] [karadā hai]* Resultant nodes: [ਓੱਹ][ਉਸਨੂੰ][ਪਿਆਰ@present] *[ōhh][usnūṃ][piār@present]*

3.	(R,PPR,%a)(V,TSTD,%b):=(NA(%b;%a),+OBJ,V,TSTD,+GEN=%b,+NUM=%b);	Here, *%a* refers to node [ਉਸਨੂੰ], and *%b* refers to node [ਪਿਆਰ@present]. This rule resolves a relation 'NA' whose first and second argument are *%b* and *%a* respectively. This new node is given attributes 'V', 'OBJ'.	Original nodes: [ਓਹ][ਉਸਨੂੰ][ਪਿਆਰ@present] *[ōhh][usnūṃ][piār@present]* Resultant nodes: [ਓਹ][NA([ਪਿਆਰ@present];[ਉਸਨੂੰ])] *[ōhh][NA([piār@present];[usnūṃ])]*
4.	({SHEAD|CHEAD},%z)(R,{CPR|PPR},%a)(V,TSTD,MCL,%b):=(NA(%b;%a,+att=@male),+AGT,V,TSTD);	Here, *%z* refers to SHEAD, *%a* refers to node [ਓਹ], and *%c* refers to node [NA([ਪਿਆਰ@present];[ਉਸਨੂੰ])] . This rule resolves a relation 'NA' whose first and second arguments are *%b* and *%a* respectively. This new node is given attributes 'V', 'AGT'.	Original nodes: [ਓਹ][NA([ਪਿਆਰ@present];[ਉਸਨੂੰ])] *[ōhh][NA([piār@present];[usnūṃ])]* Resultant nodes: [NA([NA([ਪਿਆਰ@present];[ਉਸਨੂੰ])];[ਓਹ])] *[NA([NA([piār@present];[usnūṃ])];[ōhh])]*
5.	(NA(NA(%a;%b),OBJ,%w;%c),AGT,%r):=(%w),(NA(%a;%c),%d,+AGT);	Here, *%a* refers to node [ਪਿਆਰ@present], *%b* refers to node [ਉਸਨੂੰ], *%w* refers to node [NA([ਪਿਆਰ@present];[ਉਸਨੂੰ])], *%c* refers to [ਓਹ], and *%r* refers to node [NA([NA([ਪਿਆਰ@present];[ਉਸਨੂੰ])];[ਓਹ])]. This rule splits the node into two nodes *%w* and *%d*. The node *%d* holds a relation 'NA' whose first and second arguments are *%a* and *%c* respectively.	Original nodes: [NA([NA([ਪਿਆਰ@present];[ਉਸਨੂੰ])];[ਓਹ])] *[NA([NA([piār@present];[usnūṃ])];[ōhh])]* Resultant nodes: Node 1: [NA([ਪਿਆਰ@present];[ਉਸਨੂੰ])] *[NA([piār@present];[usnūṃ])]* Node 2: [NA([ਪਿਆਰ@present];[ਓਹ])] *[NA([piār@present];[ōhh])]*

6.	(NA(%a;%b),AGT):=agt(%a;%b);	Here, *%a* refers to node [ਪਿਆਰ@present], and *%b* refers to node [ਓੱਹ]. This rule chages the name of relation from 'NA' to 'agt' as required in the final output.	Original nodes: [NA([ਪਿਆਰ@present];[ਓੱਹ])] *[NA([piār@present];[ōhh])]* Resultant nodes: [agt([ਪਿਆਰ@present];[ਓੱਹ])] *[agt([piār@present];[ōhh])]*
7.	(NA(%a;%b),OBJ):=obj(%a;%b);	Here, *%a* refers to node [ਪਿਆਰ@present], and *%b* refers to node [ਓੱਹ]. This rule chages the name of relation from 'NA' to 'obj' as required in the final output.	Original nodes: [NA([ਪਿਆਰ@present];[ਉਸਨੂੰ])] *[NA([piār@present];[usnūṃ])]* Resultant nodes: [obj([ਪਿਆਰ@present];[ਉਸਨੂੰ])] *[obj([piār@present];[usnūṃ])]* Now all the natural language words are replaced by their universal words and final output is generated by IAN as given in (5.34).

UNL generated by IAN is given in (5.34).

{org}

ਓੱਹ ਉਸਨੂੰ ਪਿਆਰ ਕਰਦਾ ਹੈ

{/org}

{unl}

agt(love:05.@present, 00:01.@3.@male)

obj(love:05.@present, 00:03.@3.@female)

{/unl} ...(5.34)

Next section describes the need of following a different approach, *i.e.*, X-Bar approach for UNL-ization. UNL-ization of UC-A1 and UC-A2 is done using X-Bar approach.

5.3 UNL-ization with X-Bar Approach

During UNL-ization of CORPUS500 it was realised that as the sentences becomes more and more complex, the number of TRules increases significantly and conflicts arises with the previously made TRules. Thus there was a need to follow a more systematic approach for UNL-ization, X-Bar approach. The X-bar theory was first proposed by Noam Chomsky (1970). It postulates that all human languages share certain structural similarities, including the same underlying syntactic structure, which is known as the "X-bar" [56]. The X-bar abstract configuration is depicted in the Figure 5.1.

```
      XP
     /  \
  spec   XB
          / \
       XB   adjt
      /   \
     X    comp
     |
    Head
```

Figure 5.1: X-Bar abstract Configuration [56]

Here,

- X is the **head**, the nucleus or the source of the whole syntactic structure, which is actually derived (or projected) out of it. The letter X is used to signify an arbitrary lexical category (part of speech). When analyzing a specific utterance, specific categories are assigned. Thus, the X may become an N for noun, a V for verb, a J for adjective, or a P for preposition.
- **comp** (*i.e.*, complement) is an internal argument, *i.e.*, a word, phrase or clause which is necessary to the head to complete its meaning (*e.g.*, objects of transitive verbs)
- **adjt** (*i.e.*, adjunct) is a word, phrase or clause which modifies the head but which is not syntactically required by it (adjuncts are expected to be

extranuclear, *i.e.*, removing an adjunct would leave a grammatically well-formed sentence)

- **spec** (*i.e.*, specifier) is an external argument, *i.e.*, a word, phrase or clause which qualifies (determines) the head
- **XB** (X-bar) is the general name for any of the intermediate projections derived from X
- **XP** (X-bar-bar, X-double-bar, X-phrase) is the maximal projection of X.

Section 5.3.1 explains UNL-ization process using X-Bar approach with the help of an example sentence (5.35).

5.3.1 UNL-ization of UC-A1 and UC-A2 using X-Bar

UC-A1 and UC-A2 are the corpus provided by UNDL Foundation. UC-A1 contains 100 Natural Language sentences, and UC-A2 comprises of 300 sentences, both covering all the major part-of-speeches. UNL-ization of UC-A1 and UC-A2 is done with the help of 881 TRules and 663 Dictionary entries. UNL-ization using X-Bar approach is explained with the help of example sentences (5.35).

Example 1: ਪੈਰਿਸ ਤੋਂ ਜਨੇਵਾ ਨੂੰ ਰੇਲ ਗੱਡੀ …(5.35)

pairis tōṃ janēvā nūṃ rēl gaḍḍī
A train from pairis to geneva

After the tokenization of example sentence given in (5.35) with IAN tool, Nine lexical items are identified as shown in (5.36).

[ਪੈਰਿਸ]{}"Paris"(LEX=N,POS=PPN,NUM=SNG,SEM=LCT)<pan,0,0>;

[ਤੋਂ]{}"from"(LEX=P,POS=PRE,rel=plc,att=@from)<pan,0,0>;

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

[ਨੂੰ]{}"to"(LEX=P,POS=PRE,rel=plc,att=@to)<pan,0,0>;

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

Four blank spaces are also identified as :-
[]{}""(BLK)<pan,0,0>; ...(5.36)

The process of UNL-ization of example sentence (5.32) has been illustrated in Table 5.12.

Table 5.12: UNL-ization process for example sentence (5.35)

S.No	TRule Fired	Description	Action Taken
1.	(%a,BLK,^BLANK):=;	Here, *%a* refers to blank node []. This TRule is fired four times and deletes all the blank nodes.	Original Nodes: [ਪੈਰਿਸ][][ਤੋਂ][][ਜਨੇਵਾ][][ਨੂੰ][][ਰੇਲ ਗੱਡੀ] *[pairis][][tōṃ][][janēvā][][nūṃ][][rēl gaḍḍī]* Resultant Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ][ਨੂੰ][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā][nūṃ][rēl gaḍḍī]*
2.	(N,^proj,^XB,^XP,%a):=(+XB=NB);	Here, *%a* refers to node [ਪੈਰਿਸ]. This TRule takes node *%a* to its intermediate projection XB.	Original Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ][ਨੂੰ][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā][nūṃ][rēl gaḍḍī]* Resultant Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ][ਨੂੰ][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā][nūṃ][rēl gaḍḍī]*
3.	(N,^proj,^XB,^XP,%a):=(+XB=NB);	Here, *%a* refers to node [ਜਨੇਵਾ]. This TRule takes node *%a* to its intermediate projection XB.	Original Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ][ਨੂੰ][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā][nūṃ][rēl gaḍḍī]* Resultant Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ][ਨੂੰ][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā][nūṃ][rēl gaḍḍī]*
4.	(%a,NB,LCT,^proj,^XP)([ਨੂੰ],%b)(%c,{^D,^J,^LCT,^P|P	Here, *%a* refers to node [ਜਨੇਵਾ]. This TRule takes node *%a* to its maximal projection NP while keeping nodes *%b*	Original Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ][ਨੂੰ][ਰੇਲ ਗੱਡੀ]

	UT|STAIL|CTAIL}):=(%a,-XB,-NB,+XP=NP,+proj)(%b)(%c);	and *%c* as such.	*[pairis][tōṃ][janēvā][nūṃ][rēl gaḍḍī]* Resultant Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ][ਨੂੰ][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā][nūṃ][rēl gaḍḍī]*
5.	(XP,^DP,^PP,%y)(P,^PP,%x)({XP|VB|PUT|STAIL|CTAIL|N},%z):=(%y,-LEX,-XP,+LEX=P,+XP=PP,+adjt,+rel=%x,+att=%x)(%z);	Here, *%y* refers to node [ਜਨੇਵਾ], *%x* refers to node [ਨੂੰ] and *%z* refers to node [ਰੇਲ ਗੱਡੀ]. This rule deletes node *%x* and gives the attributes of *%x* to *%y*. Node *%y* is taken to its maximal projection.	Original Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ][ਨੂੰ][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā][nūṃ][rēl gaḍḍī]* Resultant Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ@to][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā@to][nūṃ][rēl gaḍḍī]*
6.	({NOU|PPN},^N,%a):=(-LEX,+LEX=N);	Here, *%a* refers to node [ਜਨੇਵਾ@to]. This rule assigns node *%a* lexical category Noun.	Original Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ@to][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā@to][nūṃ][rēl gaḍḍī]* Resultant Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ@to][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā@to][nūṃ][rēl gaḍḍī]*
7.	(N,^proj,^XB,^XP,%x):=(+XB=NB);	Here, *%x* refers to node [ਰੇਲ ਗੱਡੀ]. This rule takes node *%x* to its intermediate projection NB.	Original Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ@to][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā@to][nūṃ][rēl gaḍḍī]* Resultant Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ@to][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā@to][nūṃ][rēl gaḍḍī]*
8.	({^V|^P|SHEAD|CHEAD},%w)(PP,plc,%y)(NB,%x)({XP|CB|STAI	Here, *%w* refers to node [ਤੋਂ], *%y* refers to node [ਜਨੇਵਾ@to], *%x* refers to node [ਰੇਲ ਗੱਡੀ],	Original Nodes: [ਪੈਰਿਸ][ਤੋਂ][ਜਨੇਵਾ@to][ਰੇਲ ਗੱਡੀ] *[pairis][tōṃ][janēvā@to]*

	L|CTAIL},%z):=(%w)(XB(%x,+head;%y,+comp),+XB=NB,+LEX=N,%xy)(%z);	and *%z* refers to STAIL. This rule creates a relation XB whose first argument is *%x* and second argument is *%y*, and named it as *%xy*. *%x* becomes head while *%y* becomes complement. *%w* and *%z* remains as such at their respective places. *%xy* is in its intermediate projected form *i.e.*, NB.	*[nūṃ][rēl gaḍḍī]* Resultant Nodes: [ਪੈਰਿਸ][ਤੋਂ][XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])] *[pairis][tōṃ][XB([rēl gaḍḍī];[janēvā@to])]*
9.	(%ss,^J,{SHEAD|^SHEAD|CHEAD|^CHEAD})(%a,LCT,NB,^proj,^XP)(P,%b)(%c,{^D,^J,^LCT,^P|PUT|STAIL|CTAIL}):=(%ss)(%a,-XB,-NB,+XP=N P,+proj,-LCT,vswitzerland)(%b)(%c);	Here, *%ss* refers to SHEAD, *%a* refers to node [ਪੈਰਿਸ], *%b* refers to node [ਤੋਂ], and *%c* refers to node [XB(ਰੇਲ ਗੱਡੀ;ਜਨੇਵਾ@to)]. This rule takes node *%a* to its maximal projection 'NP' and removes surplus attributes.	Original Nodes: [ਪੈਰਿਸ][ਤੋਂ][XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])] *[pairis][tōṃ][XB([rēl gaḍḍī];[janēvā@to])]* Resultant Nodes: [ਪੈਰਿਸ][ਤੋਂ][XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])] *[pairis][tōṃ][XB([rēl gaḍḍī];[janēvā@to])]*
10.	(XP,^DP,^PP,%y)(P,^PP,%x)({XP|VB|PUT|STAIL|CTAIL|N},%z):=(%y,-LEX,-XP,+LEX=P,+XP=PP,+adjt,+rel=%x,+att=%x)(%z);	Here, *%y* refers to node [ਪੈਰਿਸ], *%x* refers to node [ਤੋਂ], and *%z* refers to node [XB(ਰੇਲ ਗੱਡੀ;ਜਨੇਵਾ@to)]. This rule deletes node *%x* and gives its attributes to node *%y*. Node *%y* is taken to its maximal projection PP *i.e.*, prepositional phase and is made adjunct.	Original Nodes: [ਪੈਰਿਸ][ਤੋਂ][XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])] *[pairis][tōṃ][XB([rēl gaḍḍī];[janēvā@to])]* Resultant Nodes: [ਪੈਰਿਸ@morf][XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])] *[pairis@from][tōṃ][XB([rēl gaḍḍī];[janēvā @to])]*
11.	({NOU|PPN},^N,%a):=(-LEX,+LEX= N);	Here, *%a* refers to node [ਪੈਰਿਸ@from]. This Node is assigned as lexical category Noun.	Original Nodes: [ਪੈਰਿਸ@morf][XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])] *[pairis@from][tōṃ][XB([rēl gaḍḍī];[janēvā @to])]* Resultant Nodes: [ਪੈਰਿਸ@morf][XB([ਰੇਲ

			ਗੱਡੀ];[ਜਨੇਵਾ@to])] *[pairis@from][tōṃ][XB([rēl gaḍḍī];[janēvā @to])]*
12.	({^V|^P|SHEAD|CHEAD},%w)(PP,plc,vswitzerland,%y)(NB,%x)({XP|CB|STAIL|CTAIL},%z):=(%w)(XB(%x,+head;%y,+comp),+XB=NB,+LEX=N,%xy,vswitzerland)(%z);	Here, *%w* refers to SHEAD, *%y* refers to node [ਪੈਰਿਸ@from], *%x* refers to node [XB(ਰੇਲ ਗੱਡੀ;ਜਨੇਵਾ@to)], and *%z* refers to STAIL. XB relation is formed with *%x* and *%y* as first and second argument respectively. First argument becomes head while second argument becomes complement. *%w* and *%z* are maintained as such.	Original Nodes: [ਪੈਰਿਸ@morf][XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])] *[pairis@from][tōṃ][XB([rēl gaḍḍī];[janēvā @to])]* Resultant Nodes: [XB([XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])];[ਪੈਰਿਸ@from])] *[XB([XB([rēl gaḍḍī];[janēvā@to])];[pairis @from])]*
13.	({^D,^J,^N|SHEAD|CHEAD|PUT},%a)(NB,^proj,^XP,%b)({^D,^J,^P|PUT|STAIL|CTAIL},%c):=()(-XB,-NB,+XP=NP,+proj)();	Here, *%a* refers to SHEAD, *%b* refers to node [XB([XB(ਰੇਲ ਗੱਡੀ;ਜਨੇਵਾ@to)];[ਪੈਰਿਸ@from])] , and *%c* refers to node STAIL. This rule takes node *%b* to its maximal projection XP.	Original Nodes: [XB([XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])];[ਪੈਰਿਸ@from])] *[XB([XB([rēl gaḍḍī];[janēvā@to])];[pairis@from])]* Resultant Nodes: [XB([XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])];[ਪੈਰਿਸ@from])] *[XB([XB([rēl gaḍḍī];[janēvā@to])];[pairis@from])]*
14.	XB(XB(%x;%y);%z):=XB(%x;%y)XB(%x;%z);	Here, *%x* refers to node [ਰੇਲ ਗੱਡੀ], *%y* refers to node [ਜਨੇਵਾ@to], and *%z* refers to node [ਪੈਰਿਸ@from]. This rule splits the hypernode into two XB relations with first and second arguments as shown in Action Taken column.	Original Nodes: [XB([XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])];[ਪੈਰਿਸ@from])] *[XB([XB([rēl gaḍḍī];[janēvā@to])];[pairis@from])]* Resultant Nodes: [XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])][XB([ਰੇਲ ਗੱਡੀ];[

			ਪੈਰਿਸ@from])] *[XB([rēl gaḍḍī];[janēvā@to])][XB([rēl gaḍḍī];[pairis@from])]*
15.	XB(%x,N;%y,adjt):=NA(%x;%y);	Here, *%x* refers to node [ਰੇਲ ਗੱਡੀ], and *%y* refers to node [ਜਨੇਵਾ@to]. This rule changes the name of relation from 'XB' to 'NA'.	Original Nodes: [XB([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])][XB([ਰੇਲ ਗੱਡੀ];[ਪੈਰਿਸ@from])] *[XB([rēl gaḍḍī];[janēvā@to])][XB([rēl gaḍḍī];[pairis@from])]* Resultant Nodes: [NA([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])][XB([ਰੇਲ ਗੱਡੀ];[ਪੈਰਿਸ@from])] *[NA([rēl gaḍḍī];[janēvā@to])][XB([rēl gaḍḍī];[pairis@from])]*
16.	XB(%x,N;%y,adjt):=NA(%x;%y);	Here, *%x* refers to node [ਰੇਲ ਗੱਡੀ], and *%y* refers to node [ਪੈਰਿਸ@from]. This rule changes the name of relation from 'XB' to 'NA'.	Original Nodes: [NA([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])][XB([ਰੇਲ ਗੱਡੀ];[ਪੈਰਿਸ@from])] *[NA([rēl gaḍḍī];[janēvā@to])][XB([rēl gaḍḍī];[pairis@from])]* Resultant Nodes: [NA([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])][NA([ਰੇਲ ਗੱਡੀ];[ਪੈਰਿਸ@from])] *[NA([rēl gaḍḍī];[janēvā@to])][NA([rēl gaḍḍī];[pairis@from])]*
17.	/[ACDIJNPV][ACS]/(%x;%y,plc):=plc(%x;%y);	Here, *%x* refers to node [ਰੇਲ ਗੱਡੀ], and *%y* refers to node [ਜਨੇਵਾ@to]. This rule changes the name of relation from 'NA' to 'plc' as required in the final UNL.	Original Nodes: [NA([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ@to])][NA([ਰੇਲ ਗੱਡੀ];[ਪੈਰਿਸ@from])] *[NA([rēl gaḍḍī];[janēvā@to])][NA([rēl gaḍḍī];[pairis@from])]*

			Resultant Nodes: [plc([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ @to])][NA([ਰੇਲ ਗੱਡੀ];[ਪੈਰਿਸ@from])] *[plc([rēl gaḍḍī];[janēvā @to])][NA([rēl gaḍḍī];[pairis@from])]*
18.	/[ACDIJNPV][ACS]/(%x;%y,plc):=plc(%x;%y);	Here, *%x* refers to node [ਰੇਲ ਗੱਡੀ], and *%y* refers to node [ਪੈਰਿਸ@from]. This rule changes the name of relation from 'NA' to 'plc' as required in the final UNL.	Original Nodes: [plc([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ @to])][NA([ਰੇਲ ਗੱਡੀ];[ਪੈਰਿਸ@from])] *[plc([rēl gaḍḍī];[janē vā@to])][NA([rēl gaḍ ḍī];[pairis@from])]* Resultant Nodes: [plc([ਰੇਲ ਗੱਡੀ];[ਜਨੇਵਾ @to])][plc([ਰੇਲ ਗੱਡੀ];[ਪੈਰਿਸ@from])] *[plc([rēl gaḍḍī];[janē vā@to])][plc([rēl gaḍ ḍī];[pairis@from])]* Now all the natural language words are replaced by their universal words and final output is generated by IAN as shown in (5.37).

UNL generated by IAN is given in (5.37).

{org}

ਪੈਰਿਸ ਤੋਂ ਜਨੇਵਾ ਨੂੰ ਰੇਲ ਗੱਡੀ

{/org}

{unl}

plc(train:09, Geneva:05.@to)

plc(train:09, Paris:01.@from)

{/unl} ...(5.37)

5.4 Implementation Summary

This chapter explained that how with the help of 241 TRules and 450 Dictionary entries UNL-ization of CORPUS500 has been done successfully for all the major Part-of-Speech for Punjabi Language. Further UNL-ization of UC-A1 and UC-A2 has been done using X-Bar approach with the help of 881 TRules and 663 Dictionary entries. UNL-ization has been carried out with the help of IAN tool developed by UNDL Foundation.

Chapter 6

Results and Discussion

Universal Networking Language is a natural-language independent language which can be used for refining, describing, and semantic searching. Interactive Analyser tool is an effective online tool developed by UNDL foundation used for UNL-ization of any natural language.

With the help of 241 TRules and 450 Dictionary entries, the proposed system has been tested on CORPUS500 for Punjabi Language, and their F-measure / F1-score is calculated with the help of online tool developed by UNDL foundation available at UNL-arium [54] as shown in Figure 6.1. Same tool is used for calculating F-Measure of UC-A1 and UC-A2. F-measure is the measure of a grammar's accuracy [12]. Two parameters required for the calculation of F-measure. These are:-

Precision

Precision is the number of correct results divided by the number of all returned results.

Recall

Recall is the number of correct results divided by the number of results that should have been returned.

A result is considered returned when the output is a graph made of only Universal Words. A result is considered "correct" when the Levensthein distance between the actual result and the expected result was less than 30% of the length of the expected result. The Levenshtein distance is defined as the minimal number of characters you have to replace, insert or delete to transform a string (the actual output) into another one (the expected output).

F-measure is calculated by the formulae given in (6.1) [12].

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

6.1 Testing of CORPUS500

The values of Precision, Recall, number of processed, returned and correct sentences for CORPUS500 is given in Table 6.1.

Table 6.1: Testing details of CORPUS500

S. No	Parameters	Value
1.	Sentences processed	500

2.	Sentences returned	471
3.	Sentences correct	401
4.	Precision	0.851
5.	Recall	0.802

The F-measure of CORPUS500 comes out to be 0.826 as shown in Figure 6.1.

Figure 6.1: Snapshot showing F-measure of CORPUS500 [4]

6.2 Testing of UC-A1

The values of Precision, Recall, number of processed, returned and correct sentences for UC-A1 is given in Table 6.2.

Table 6.2: Testing details of UC-A1

S. No	Parameters	Value
1.	Sentences processed	100
2.	Sentences returned	100
3.	Sentences correct	97
4.	Precision	0.970
5.	Recall	0.970

Total 881 TRules and 663 Dictionary entries were created for UNL-ization of UC-A1 and UC-A2. F-measure of UC-A1 comes out to be 0.970 as shown in Figure 6.2.

Figure 6.2: Snapshot showing F-measure of UC-A1 [4]

6.3 Testing of UC-A2

The values of Precision, Recall, number of processed, returned and correct sentences for UC-A2 is given in Table 6.3.

Table 6.3: Testing details of UC-A2

S. No	Parameters	Value
1.	Sentences processed	300
2.	Sentences returned	287
3.	Sentences correct	264
4.	Precision	0.880
5.	Recall	0.920

F-measure of UC-A2 comes out to be 0.900 as shown in Figure 6.3.

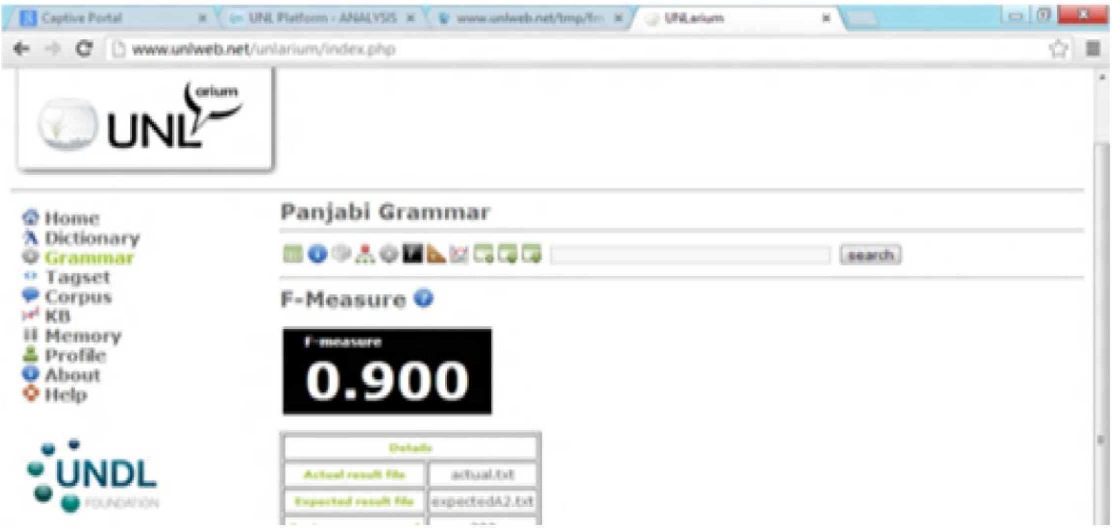

Figure 6.3: Snapshot showing F-measure of UC-A2 [4]

Chapter 7
Conclusion and Future Scope

7.1 Conclusion

UNL is a natural language independent language which can be used not only for natural language translation but also for other NLP tasks like information retrieval, text simplification and semantic reasoning *etc*. The UNL programme started in 1996, as an initiative of the Institute of Advanced Studies of the United Nations University in Tokyo, Japan [14]. 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 programme. As far as multilingual translation of the natural language is concerned, the UNL approach is far more better than other statistical approaches, because by using the approach of UNL we only need *2n* components while other methods/approaches uses *n*(n-1)* components where *n* is the number of natural languages.

UNDL Foundation had provided CORPUS500, UC-A1, and UC-A2. This thesis explains how the given natural language text like Punjabi can be converted to UNL with the help of IAN using TRules and its corresponding grammar. This tool is very efficient and the resources like TRules, Analysis Grammar *etc*, can be shared with other users. CORPUS500, UC-A1, and UC-A2 were manually converted to our natural language, *i.e.*, Punjabi for input to IAN. After that with the help of 241 TRules and 450 Dictionary entries CORPUS500 has been UNL-ized. Later UNL-ization of UC-A1 and UC-A2 has been done using X-Bar approach. For UNL-ization of UC-A1 and UC-A2 total 881 TRules and 663 Dictionary entries has been created.

The accuracy of the proposed system has been calculated using F-measure for CORPUS 500, UC-A1, and UC-A2 with the help of an online tool developed by UNDL Foundation. F-measure of CORPUS500, UC-A1, and UC-A2 came out to be 0.826, 0.910, and 0.900 respectively.

7.2 Future Scope

UNL-ization of Punjabi language has been done on sentence level using IAN. Some of the work that can be carried out in future includes following.

1.) UNL-ization will be done on real Corpuses at paragraph level. UNDL Foundation has provided UCB1 which contains different stories like hare and tortoise, ant and the grasshopper *etc*.

2.) Since UNL captures semantics of the natural language so semantic based searching system can be developed for Punjabi.

3.) Work can be extended to carry out UNLization of numbers and ordinals of more than fourteen digits.

4.) System needs to be improved so as to achieve F-Measure of 1.000.

5.) The proposed system can be tested for other natural languages spoken worldwide.

6.) The work can be extended for other UNL applications like LILY (*i.e.*, Language-to-Interlanguage-to-Language sYstem), TUT (*i.e.*, Text-to-Text through UNL), and KEYS (*i.e.*, Knowledge Extraction sYStem). LILY, TUT, and KEYS are the applications which are under development by UNDL Foundation.

References

[1] A. Avetisyan, V. Avetisyan, "LOOK4: Enhancement of web search results with Universal Words and WordNet", in *Proc. 5th Int. Conf. on Global WordNet*, Mumbai, India, 2010, pp. 1-5.

[2] A. Mukerjee *et al.*, "Universal Networking Language: a tool for language independent semantics," in *Universal Network Language: Advances in Theory and Applications,* vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 145-154.

[3] B. Choudhary, P. Bhattacharyya, "Text clustering using Universal Networking Language representation," in *Proc. 11th Int. Conf. on World Wide Web*, Hawaii, USA, 2002, pp. 1-7.

[4] "Calculation of F-Measure," [Online]. Available: http://www.unlweb.net/unlarium/index.php [Accessed 15 May 2013].

[5] C. Boitet *et al.*, "Building Hindi-French-English-UNL resources for SurviTra-CIFLI, a linguistic survival system under construction," in *Proc. 7th Int. Symp. on Natural Language Processing*, Chonburi, Thailand, 2007, pp. 1-6.

[6] C. Jiang *et al.*, "Facilitating communication between languages and cultures: a computerized interface and knowledge base," in *Universal Network Language: Advances in Theory and Applications,* vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 358-368.

[7] "Concepts of UW," [Online]. Available: http://www.undl.org/unlsys/unl/unl2005/uw.htm [Accessed 3 June 2013]

[8] C. Ribeiro *et al.*, "Semi-automatic UNL dictionary generation using WordNet.PT," in *Proc. 4th Int. Conf. on Language Resources and Evaluation,* Lisbon, 2004, pp. 279-282.

[9] “Dictionary Specifications,” [Online]. Available: http://www.unlweb.net/wiki/Dictionary_Specs [Accessed 9 June 2013].

.

[10] D. Rouquet, H. Nguyen, “Interlingual annotation of texts in the OMNIA project,” in *Proc. 4th Int. Conf. on Language and Technology*, Poland, 2009, pp. 1-5.

[11] É. Blanc, “About and around the French Enconverter and the French Deconverter,” in *Universal Network Language: Advances in Theory and Applications,* vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 157-166.

[12] “F-Measure,” [Online]. Available: http://www.unlweb.net/wiki/F-measure [Accessed 19 June 2013].

[13] G. Sérasset, C. Boitet, “On UNL as the future ‘html of the linguistic content’ & the reuse of existing NLP components in UNL-related applications with the example of a UNLFrench deconverter,” in *Proc. Int. Conf. on Computational Linguistics,* Saarbruecken, Germany, 2000, pp. 768-774.

[14] “History of UNL,” [Online]. Available: http://www.unlweb.net/wiki/Introduction_to_UNL [Accessed 3 June 2013].

[15] “IAN,” [Online]. Available: http://dev.undlfoundation.org/analysis/index.jsp [Accessed 9 June 2013].

[16] I. M. Boguslavsky *et al*., “Interactive EnConversion by means of the ETAP-3 system,” in *Universal Network Language: Advances in Theory and Applications,* vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 230-240.

[17] “Interactive Analyzer,” [Online]. Available: http://www.unlweb.net/wiki/IAN [Accessed 9 June 2013].

[18] J. Bekios *et al*., "Using Wordnet for building an interlingual dictionary," in *Proc. 5th Int. Conf. Information Research and Applications*, Varna, Bulgaria, 2007, pp. 1-8.

[19] J. B. Karande, "Multilingual search engine: implementation using UNL," in *Proc. Int. Conf. on Semantic Web and Digital Libraries*, Bangalore, India, 2007, pp. 1-7.

[20] J. Cardeñosa, A. Gelbukh, E. Tovar, *Universal Network Language: Advances in Theory and Applications.* México: Centre for Computing Research of IPN, 2005, pp. 1-466.

[21] K. Dey, P. Bhattacharyya, "Universal Networking Language based analysis and generation of Bengali case structure constructs," in *Universal Network Language: Advances in Theory and Applications*, vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 215-229.

[22] K. Ramamritham *et al*., "aAqua: a database- backended multilingual, multimedia community forum," in *Proc. Int. Conf. on Management of Data*, Chicago, USA, 2006, pp. 784-786.

[23] L. Bértoli *et al*., "A web platform using UNL: CELTA's Showcase," in *Universal Network Language: Advances in Theory and Applications*, vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 276-285.

[24] L. Iraola, "Using WordNet for linking UWs to the UNL UW system," in *Universal Network Language: Advances in Theory and Applications*, vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 369-378.

[25] M. Choudhury *et al*., “Bridging Bangla to Universal Networking Language - a human language neutral meta-language,” in *Proc. 8th Int. Conf. on Computer and Information Technology,* Dhaka, Bangladesh, 2005, pp. 104-109.

[26] M. F. Mridha *et al.,* “Development of Morphological Rules for Bangla Words for Universal Networking Language,” *Int. J. of Computer Science and Network Security,* vol. 10, pp. 231-237, 2010.

[27] M. Jain, O. P. Damani, “English to UNL (Interlingua) EnConversion,” in *Proc.2nd Conference on Language and Technology,* Lahore, Pakistan, 2009, pp. 1-8.

[28] M. Lafourcade, “Semantic analysis through ant algorithms, conceptual vectors and fuzzy UNL graphs,” in *Universal Network Language: Advances in Theory and Applications,* vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 125-137.

[29] M.P. Lewis, “*Ethnologue: Languages of the World*”, 16th ed, Dallas: SIL International, 2009.

[30] M. S. Gill, “Development of a Punjabi grammar checker,” Ph.D. dissertation, Punjabi University, Patiala, India, 2008.

[31] M. Surve *et al*., “AgroExplorer: a meaning based multilingual search engine,” in *Proc. Int. Conf. on Digital Libraries*, Delhi, India, 2004, pp. 1-13.

[32] N. Adly, S. Alansary, “Evaluation of Arabic machine translation system based on the Universal Networking Language,” in *Natural Language Processing and Information Systems, Lecture Notes in Computer Science,* vol. 5723. H. Horacek, E. Métais, R. Muñoz, M. Wolska, Ed: Springer Berlin Heidelberg, 2009, pp. 243-257.

[33] N. Verma, P. Bhattacharyya, “Automatic generation of multilingual lexicon by using Wordnet,” in *Universal Network Language: Advances in Theory and*

Applications, vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 380-391.

[34] P. Bhattacharyya, "Multilingual information processing through Universal Networking Language," *Indo UK Workshop on Language Engineering for South Asian Languages*, 2001, pp. 1-10.

[35] P. Kumar, R. K. Sharma, "UNL Based Machine Translation System for Punjabi Language," Ph.D. dissertation, Computer Science and engineering department, Thapar University, Patiala, India, 2012.

[36] P. Shukla *et al*., "Towards a language independent encoding of documents a novel approach to multilingual question answering," in *Proc. 1st Int. Workshop on Natural Language Understanding and Cognitive Science*, Porto, Portugal, 2004, pp. 116-125.

[37] "Prepositions," [Online]. Available: http://www.unlweb.net/wiki/English_prepositions [Accessed 13 June 2013].

[38] P. T. Nguyen, M. Ishizuka, "A statistical approach for Universal Networking Language-based relation extraction," in *Proc. Int. Conf. on Research, Innovation and Vision for the Future*, Ho Chi Minh City, Vietnam, 2006, pp. 153-160.

[39] "Relations," [Online]. Available: http://www.unlweb.net/wiki/Relations [Accessed 3 June 2013].

[40] R. Mohanty *et al*., "Prepositional phrase attachment and interlingua," in *Universal Network Language: Advances in Theory and Applications,* vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005a, pp. 241-253.

[41] R. Mohanty *et al*., “Semantically relatable sets: building blocks for representing semantics,” in *Proc. 10th MT Summit*, Phuket, 2005b, pp. 1-8.

[42] R. T. Martins *et al.*, “The UNL distinctive features: evidences through a NL-UNL encoding task,” in *Proc. 1st Int. Workshop on UNL, other Interlinguas and their Applications*, Las Palmas, Spain, 2002, pp. 8-13.

[43] R. T. Martins *et al*., *Specification of the UNL-Portuguese enconverter-deconverter prototype.* São Carlos: ICMSC-USP, 1997, pp. 1-10.

[44] R. T. Martins *et al.*, “Hermeto: A NL–UNL Enconverting Environment,” in *Universal Network Language: Advances in Theory and Applications,* vol. 12. J. Cardeñosa, A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 254-260.

[45] S. Alansary *et al*., “A Semantic-based approach for multilingual translation of massive documents,” in *Proc. 7th Int. Symposium on Natural Language Processing,* Chonburi, Thailand, 2007, pp. 317-323.

[46] S. Boudhh, P. Bhattacharyya, “Unification of Universal Word dictionaries Using WordNet ontology and similarity measures,” in *Proc. 7th Int. Conf. on Computer Science and Information Technologies*, Armenia, 2009, pp. 271-283.

[47] “Scope and goals of UNL,” [Online]. Available: http://www.unlweb.net/unlweb/index.php?option=com_content&view=article&id=58:unl&catid=54:unlweb [Accessed 3 June 2013].

[48] “Syntax of attributes,” [Online]. Available: http://www.unlweb.net/wiki/Attributes [Accesed 3 June 2013].

[49] T. Bueno *et al*., “Knowledge engineering suite: a tool to create ontologies for automatic knowledge representation in intelligent systems,” in *Universal Network Language: Advances in Theory and Applications,* vol. 12. J. Cardeñosa,

A. Gelbukh, E. Tovar, Ed. México: Centre for Computing Research of IPN, 2005, pp. 337-346.

[50] T. Dhanabalan *et al*., "Tamil to UNL EnConverter," in *Proc. Int. Conf. on Universal Knowledge and Language*, Goa, India, 2002, pp. 1-16.

[51] "Transformation Rules," [Online]. Available: http://www.unlweb.net/wiki/T-rule [Accessed 10 June 2013].

[52] "Types of UW," [Online]. Available: http://www.unlweb.net/wiki/UW [Accessed 3 June 2013].

[53] "UNDL Foundation," [Online]. Available: http://www.undl.org [Accessed 6 June 2013].

[54] "UNL-arium," [Online]. Available: http://www.unlweb.net/unlarium/index.php?lang=pa [Accessed 19 June 2013].

[55] V. Sornlertlamvanich *et al*., "Thai lexical semantic annotation by UW," in *Proc. 7th Int. Workshop on Academic Information Networks and Systems*, Bangkok, Thailand, 2000, pp. 1-6.

[56] "X-Bar theory," [Online]. Available: http://www.unlweb.net/wiki/X-bar [Accessed 18 June 2013].

Cleared CUP500 Certificate of Proficiency in UNL.

UNDL FOUNDATION

Universal Networking Language

CUP 500

Certificate of Proficiency in UNL

The UNDL Foundation certifies that

vaibhav agarwal

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, July 15, 2012

Tarcisio G. Della Senta, PhD	**Ronaldo T. Martins, PhD**
President	**Language Resources Manager**
UNDL Foundation	**UNDL Foundation**

Publications

Research paper Published

- V. Agarwal, P. Kumar. “UNL-ization of Numbers and Ordinals in Punjabi with IAN.” *International Journal of Natural Language Computing*, vol. 2, pp. 61-69, June 2013.

Research paper Communicated

- V. Agarwal, P. Kumar. “UNL-ization of Punjabi with IAN.” in *The 11th workshop on Asian language resources*, Nagoya, Japan, Oct 14, 2013.
- V. Agarwal, P. Kumar. “UNL-ization Framework with IAN.” *CSI journal of computing*.
- V. Agarwal, P. Kumar. “Punjabi Natural language processing with IAN: A UNL approach.” *Springer*: *CSI Transactions on ICT*.

Appendix-A

Some important TRules

<table>
<tr><th>S. No</th><th>TRules</th></tr>
<tr><td>1.</td><td>(TEMP,%a)(BLK,%b)(TEMP,%c):=(%a)(%b,-BLK,+BLANK)(%c);</td></tr>
<tr><td>2.</td><td>(TEMP,%a)(BLANK,%b)(TEMP,%c):=(%a&%b&%c);</td></tr>
<tr><td>3.</td><td>({SHEAD|CHEAD|^BLK},%z)(DIGIT,HUNDRED,%x)(DOUBLE,%y):=(%z,MAZ)(%x&%y,-HUNDRED,-DIGIT,-DOUBLE,+000);</td></tr>
<tr><td>4.</td><td>({SHEAD|CHEAD|^BLK},%z)(DOUBLE,HUNDRED,%x)(DOUBLE,%y):=(%z)(%x&%y,-HUNDRED,-DOUBLE,+0000);</td></tr>
<tr><td>5.</td><td>({SHEAD|CHEAD|^BLK},%z)({DIGIT|DOUBLE},HUNDRED,%x)({STAIL|CTAIL|DIGIT},^DOUBLE,^DOZEN,^00,%y):=(%z)(%x)([[0]],[0],"0",DOZEN,DIGIT)(%y);</td></tr>
<tr><td>6.</td><td>({SHEAD|CHEAD|^BLK|DIGIT},%z)(DIGIT,DOZEN,%x)(DIGIT,%y):=(%z)(%x&%y,-DOZEN,-DIGIT,+00);</td></tr>
<tr><td>7.</td><td>({SHEAD|CHEAD|^BLK|DIGIT},%z)(DIGIT,DOZEN,%x)(DIGIT,%y):=(%z)(%x&%y,-DOZEN,-DIGIT,+00);</td></tr>
<tr><td>8.</td><td>({SHEAD|CHEAD|^BLK},%z)(DIGIT,HUNDRED,%x)(00,%y):=(%z)(%x&%y,-HUNDRED,-DIGIT,-00,+000);</td></tr>
<tr><td>9.</td><td>({SHEAD|CHEAD|^BLK},%z)(DOUBLE,HUNDRED,%x)(00,%y):=(%z)(%x&%y,-HUNDRED,-00,-DOUBLE,+0000);</td></tr>
<tr><td>10.</td><td>({SHEAD|CHEAD|^BLK},%z)(DIGIT,^LAKH_THOUSAND,THOUSAND,%x)({DIGIT|DOUBLE|STAIL|CTAIL},%y):=(%z)(%x)([[0]],[0],"0",HUNDRED,DIGIT)(%y);</td></tr>
<tr><td>11.</td><td>(DIGIT,^LAKH_THOUSAND,THOUSAND,%x)(000,%y)({CTAIL|STAIL},%z):=(%x&%y,-THOUSAND,-000,-DIGIT,-DOUBLE,+0000)(%z);</td></tr>
<tr><td>12.</td><td>(DOUBLE,THOUSAND,%x)(000,%y)({CTAIL|STAIL},%z):=(%x&%y,-THOU SAND,-DOUBLE,-000,+00000)(%z);</td></tr>
<tr><td>13.</td><td>({DIGIT|DOUBLE|TRIPLE},%a)("ਲੱਖ"):=(%a,+LAKH);</td></tr>
<tr><td>14.</td><td>(DIGIT,LAKH,^CRORE_LAKH,%x)(00000,%y):=(%x&%y,-LAKH,-DIGIT,-00000,+000000);</td></tr>
<tr><td>15.</td><td>(DOUBLE,LAKH,%x)(00000,%y)({CTAIL|STAIL},%z):=(%x&%y,-LAKH,-DIGIT,-DOUBLE,-00000,+0000000)(%z);</td></tr>
<tr><td>16.</td><td>(TRIPLE,LAKH,%x)(00000,%y)({CTAIL|STAIL},%z):=(%x&%y,-LAKH,-DIGIT,-TRIPLE,-00000,+00000000)(%z);</td></tr>
<tr><td>17.</td><td>({SHEAD|CHEAD|^BLK},%z)({DIGIT|DOUBLE|TRIPLE},CRORE,%x)({000|0000|00000|DIGIT|DOUBLE|STAIL|CTAIL},^TEMP2,^0000000,%y):=(%z)(%x)([[00]],[00],"00",LAKH,DOUBLE,CRORE_LAKH)(%y);</td></tr>
<tr><td>18.</td><td>(DIGIT,CRORE,^ARAB_CRORE,%a)(0000000,%b):=(%a&%b,-0000000,-CRORE,-DIGIT,+00000000);</td></tr>
<tr><td>19.</td><td>(DOUBLE,CRORE,%a)(0000000,%b):=(%a&%b,-0000000,-CRORE,-DOUBLE, +000000000);</td></tr>
<tr><td>20.</td><td>(TRIPLE,CRORE,%a)(0000000,%b):=(%a&%b,-0000000,-CRORE,-TRIPLE,+0000000000);</td></tr>
<tr><td>21.</td><td>({DIGIT|DOUBLE|TRIPLE},CRORE,%a)(000000,%b):=(%a)([[0]],[0],"0",TEMP2,%c)(%b);</td></tr>
</table>

22.	(TEMP2,%a)(000000,%b):=(%a&%b,-TEMP2,-000000,+0000000);
23.	({SHEAD|CHEAD|^BLK},%z)({DIGIT|DOUBLE|TRIPLE},ARAB,%x)({000|0000|00000|000000|0000000|DIGIT|DOUBLE|STAIL|CTAIL},^TEMP3,^000000000,%y):=(%z)(%x)([[00]],[00],"00",CRORE,DOUBLE,ARAB_CRORE)(%y);
24.	({DIGIT|DOUBLE|TRIPLE},ARAB,%x)(00000000,%y):=(%x)([[0]],[0],"0",TEMP3,%z)(%y);
25.	(TEMP3,%a)(00000000,%b):=(%a&%b,-TEMP3,-00000000,+000000000);
26.	(N,^COMMAAND,%a)(C,CCJ,and,%b)(N,^COMMAAND,%c):=(%a,COMMAAND)(%b)(%c,COMMAAND);
27.	(N,^COMMAOR,%a)(C,CCJ,or,%b)(N,^COMMAOR,%c):=(%a,COMMAOR)(%b)(%c,+COMMAOR);
28.	(N,^COMMAOR,%a)(C,COO,%b)(N,COMMAOR,%c):=(%a,+COMMAOR)(%b)(%c);
29.	(N,^COMMAAND,%a)(C,COO,%b)(N,COMMAAND,%c):=(%a,+COMMAAND)(%b)(%c);
30.	(N,NOU,COMMAOR,%a)(C,and,COO,%b)(N,NOU,COMMAOR,%c):=(NA(%c;%a),+N,+NOU,OR,+COMMAOR);
31.	({SHEAD|CHEAD})(N,{NOU|PPN},%a)(C,and,CCJ,%b)(N,{NOU|PPN},%c)({STAIL|CTAIL}):=(NA(%c;%a),+N,+AND);
32.	(N,%a)(P,PRE,rel,att,%b):=(%a,+att=%b,+rel=%b,+N);
33.	(N,rel=man,att,%a)(N,%b):=(NA(%b;%a),+MAN,+N);
34.	(N,%a)(P,cnt,%b)(N,%c):=(NA(%c;%a,+att=%b),%d,+N,+CNT);
35.	(N,rel=plc,att,%a)(N,^rel,%b):=(NA(%b;%a),+PLC,+N);
36.	(NA(NA(%a;%b),GOL,%w;%c),AGT,%r):=(%w),(NA(%a;%c),+AGT);
37.	(NA(NA(%a;%b),MOD,%w;%c),MOD,%r):=(%w,-att),(NA(%a,+att=%w;%c),+ MOD);
38.	(NA(%a;%b),AOJ):=aoj(%a;%b);
39.	({^V|^P|SHEAD|CHEAD},%w)(PP,plc,%y)(NB,%x)({XP|CB|STAIL|CTAIL},%z):=(%w)(XB(%x,+head;%y,+comp),+XB=NB,+LEX=N,%xy)(%z);
40.	(XP,^DP,^PP,%y)(P,^PP,%x)({XP|VB|PUT|STAIL|CTAIL|N},%z):=(%y,-LEX,-XP,+LEX=P,+XP=PP,+adjt,+rel=%x,+att=%x)(%z);
41.	(JB,%jb)(AP,%xp):=(JB(%jb,+proj;%xp,+adjt,+proj),+XB=JB,+LEX=J,%new);
42.	({^A|SHEAD|CHEAD|PUT})(PB,^proj,^XP,%pb)({PUT|STAIL|CTAIL}):=()(-XB,-PB,+XP=PP,+proj)();
43.	/[ACDIJNPV]P/(%x;%y):=XP(%x;%y);
44.	(XP(%x;%y),%w,rel):=(XP(%x,+rel=%w;%y),%w,-rel);
45.	XB(%x,P;%y,adjt):=PA(%x;%y);
46.	XB(%x,V;%y,adjt):=VA(%x;%y);
47.	/[ACDIJNPV][ACS]/(%x;%y,aoj):=aoj(%x;%y);
48.	/[ACDIJNPV][ACS]/(%x;%y,ben):=ben(%x;%y);

Appendix-B

NL-UNL Grammar

S. No	Analysis Grammar
1.	[ਚਾਰ]{}"4"(LEX=U,POS=CDN,NUM=PLR,DIGIT)<pan,0,0>;
2.	[ਛੇ]{}"6"(LEX=U,POS=CDN,DIGIT,NUM=PLR)<pan,0,0>;
3.	[ਬੱਤੀ]{}"32"(LEX=U,POS=CDN,DOUBLE,NUM=PLR)<pan,0,0>;
4.	[ਤਿਰਤਾਲੀ]{}"43"(LEX=U,POS=CDN,DOUBLE,NUM=PLR)<pan,0,0>;
5.	[ਸੋਲਵਾਂ]{}"16"(LEX=U,POS=ORD,NUM=PLR,att=@ordinal)<pan,0,0>;
6.	[ਪੰਦਰਵਾਂ]{}"15"(LEX=U,POS=ORD,NUM=PLR,att=@ordinal)<pan,0,0>;
7.	[ਬਾਰੇ]{}""(LEX=P,POS=PST,rel=cnt,att=@about)<pan,0,0>;
8.	[ਸਾਰੀਆਂ]{}""(LEX=D,POS=QUA,GEN=FEM,NUM=MCL,att=@all)<pan,0,0>;
9.	[ਕੋਈ]{}""(LEX=D,POS=QUA,att=@any)<pan,0,0>;
10.	[ਸੋਹਣੀ]{}"beautiful"(LEX=J,POS=ADJ,GEN=FEM,NUM=SGT)<pan,0,0>;
11.	[ਕਿਉਂਕਿ]{}""(LEX=C,POS=SCJ,rel=rsn)<pan,0,0>;
12.	[ਕਿਤਾਬ]{}"book"(LEX=N,POS=NOU,NUM=SNG)<pan,0,0>;
13.	[ਕਿਤਾਬਾਂ]{}"book"(LEX=N,POS=NOU,NUM=PLR)<pan,0,0>;
14.	[ਦਿੱਤੀ]{}"give"(LEX=V,POS=VER,ATE=PAS,GEN=FEM,NUM=SNG)<pan,0,0>;
15.	[ਵਿਚ]{}"in"(LEX=P,POS=PRE,rel=plc)<pan,0,0>;
16.	[ਘੱਟ]{}""(LEX=D,POS=QUA,att=@less)<pan,0,0>;
17.	[ਕਈ]{}""(LEX=D,POS=QUA,att=@mutal)<pan,0,0>;
18.	[ਮੈਰੀ]{}"Mary"(LEX=N,POS=PPN,GEN=FEM,NUM=SNG)<pan,0,0>;
19.	[ਜ਼ਿਆਦਾ]{}""(LEX=D,POS=QUA,att=@multal)<pan,0,0>;
20.	[ਨਵੀਂ]{}"new"(LEX=J,GEN=FEM,POS=ADJ,NUM=SNG)<pan,0,0>;
21.	[ਨਵੀਂਆਂ]{}"new"(LEX=J,GEN=FEM,POS=ADJ,NUM=PLR)<pan,0,0>;
22.	[ਦੀ]{}""(LEX=P,POS=PST,GEN=FEM,rel=pos)<pan,0,0>;
23.	[ਪੈਰਿਸ]{}"Paris"(LEX=N,POS=PPN,NUM=SNG,SEM=LCT)<pan,0,0>;
24.	[ਸਾਡੀ]{}"00.@1.@pl" (LEX=D,POS=POD)<pan,0,0>;
25.	[ਉੱਤੇ]{}"on" (LEX=P,POS=PRE,rel=plc,att=@on)<pan,0,0>;