UNLization of Punjabi text for natural language processing applications

VAIBHAV AGARWAL* and PARTEEK KUMAR

Thapar Institute of Engineering and Technology, Patiala 147001, India
e-mail: vaibhavagg123@gmail.com; parteek.bhatia@thapar.edu

MS received 12 October 2015; revised 2 January 2018; accepted 20 February 2018; published online 26 May 2018

Abstract. During the last couple of years, in the field of Natural Language Processing, UNL (i.e., Universal Networking Language) immense research activities have been witnessed. This paper illustrates UNLization of Punjabi Natural Language for UC-A1, UGO-A1, and AESOP-A1 with IAN (i.e., Interactive Analyzer) tool using X-Bar approach. This paper also discusses the UNLization process in depth, step-by-step with the help of tree diagrams and tables.

Keywords. Universal networking language; Punjabi analysis grammar; Punjabi generation grammar; UNL; Punjabi natural language processing; X-bar; IAN; EUGENE.

1. Introduction

The exponential growth of the internet has made its content increasingly difficult to find, access, present and maintain for a wide variety of users. Mainly because of this reason the concept of communicating with non-human devices was further emerged as an area of research and investigation in the field of Natural Language Processing (NLP). These are the corpus provided by UNDL Foundation [1]. Prior to X-Bar approach, no standard approach had ever been followed for UNLization. Paper also highlights the errors/discrepancies in UNLization system. The proposed system has been tested with the help of online tool developed by UNDL (i.e., Universal Networking Digital Language) foundation available at UNL-arium [2], and their F-measure / F1-score (on a scale of 0 to 1) comes out to be 0.970, 0.990, and 1.00 for UC-A1, UGO-A1, and AESOP-A1, respectively. The system proposed in this paper had won UNL Olympiad II, III, and IV conducted by UNDL Foundation [3–5].

There are many applications of Natural Language Processing developed over the years. They can be broadly divided into two parts, i.e., text-based applications, and dialogue based applications [6]. Text-based applications include applications like searching for a certain topic or a keyword in a database, extracting information from a large document, translating one language to another or summarizing text for different purposes, and sentiment analysis, etc. Dialogue based application includes applications like answering systems that can answer questions, services that can be provided over a telephone without an operator,

teaching systems, voice controlled machines (that take instructions by speech) and general problem solving systems. In order to develop various NLP applications many techniques like Artificial Intelligence, Natural Language Processing, Machine Translation, and Information Retrieval strategy, etc. are being used since many years.

Artificial Intelligence techniques work on inferential mechanism and logic. Natural Language Processing strategy involves question/document analysis, information extraction, and language generation. Information Retrieval strategy technique involves query formulation; document analysis and retrieval; and relevancy feedback. Machine Translation techniques are used solely for the purpose of translation from one language to another. Since Natural Language Processing involves deep semantic analysis of the language, therefore out of these approaches Natural Language Processing approach is having wider scope for being used in developing various NLP applications. UNL is one such approach for Natural Language Processing.

This paper reports the work for UNLization of Punjabi language. Punjabi language is world's 9th most widely spoken language [7]. There are relatively less efforts in the field of computerization and development of this language.

This paper has been divided into 9 sections. Features and advantages of UNL over other traditional approaches for delivering solutions to various NLP applications are described in sub section 1.1. UNL Framework and its building blocks have been covered in subsection 1.2. Subsection briefly describes our contribution. Various advancements and previous work in UNL has been covered in section 2. Complete UNLization process and steps involved in UNLization have been described in section 3. Role of X-Bar and UNLization using X-Bar along with an

*For correspondence

Figure 1. Building blocks of UNL.

example sentence is explained in section 4. Implementation of the proposed system is covered in section 5. Section 6 illustrates the usage of the proposed system in large NLP tasks. Section 7 describes the evaluation mechanism used for calculating the accuracy of the proposed system. Results of the proposed system are given in section 8. Section 9 covers the future scope and conclusion.

1.1 Features of UNL

UNL is an artificial language to summarize, describe, represent, and store information in a natural-language-independent format [8]. UNL is expected to be used in several other different tasks such as Machine Translation, Speech to Speech Machine Translation, Text Mining, Multilingual Document Generation, Summarization, Text Simplification, Information Retrieval and Extraction, Sentiment Analysis, etc. Key features of UNL which makes it a better approach than other existing approaches are UNL represent ‘what was meant’ (and not ‘what was said’), UNL is computable, UNL representation does not depend on any implicit knowledge, i.e., it is self sufficient, UNL is not bound to translation, UNL is non ambiguous, UNL is non-redundant, UNL is compositional, UNL is declarative, and UNL is complete. Uchida *et al* [9] have provided a general idea of the UNL and its first version specifications. They have also presented the UNL system with all its components.

1.2 UNL framework

The UNL programme was launched in 1996 at Institute of Advanced Studies (IAS) of United Nations University (UNU), Tokyo, Japan and it is currently supported by Universal Networking Digital Language foundation, an autonomous organization [10]. The main aim of UNL is to capture semantics of the Natural Language resource. In UNL, UNLization and NLization are the two approaches that are being followed. UNLization is the process of converting the given Natural Language resource to UNL whereas NLization is the reverse process. UNLization and NLization are independent to each other. UNLization is done with the help of an online tool IAN while NLization uses dEep-to-sUrface GENERator (EUGENE). IAN and EUGENE tools are developed by UNDL Foundation and is

available to any user for free who registers on their portal. Universal Words (UW’s), relations and attributes are the three building blocks of UNL as shown in figure 1.

UNL represents information sentence by sentence [11]. 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 has formally defined the specifications of UNL [12].

Consider a simple example given in (I) [13].

The boy eat potatoes in the kitchen. (I)

In this example there are four UWs, three relations, and three attributes. UWs are ‘eat’, ‘boy’, ‘potato’, and ‘kitchen’. Relations are ‘agt’, ‘obj’, and ‘plc’. Attributes are ‘@entry’, ‘@def’, and ‘@pl’. The UWs ‘eat’, ‘boy’, ‘potato’, and ‘kitchen’ are restricted with constraint list to represent these concepts unambiguously as given below.

‘icl>do’ represents that UW ‘eat’ is a kind of action, ‘icl>person’ represents that UW ‘boy’ is a kind of person, ‘icl>food’ represents that UW ‘potato’ is a kind of food, and ‘icl>facilities’ represents that UW ‘kitchen’ is used as a facility provided to the UW ‘boy’.

UNL of English sentence given in (I) is given in (II):

```
{unl}
agt(eat(icl>do)@entry, boy(icl>person)@def)
obj(eat(icl>do)@entry, potato(icl>food)@pl)
plc(eat(icl>do)@entry, kitchen(icl>facilities)@def)
{/unl}
```

(II)

UWs, relations, and attributes of example sentence given in (I) are shown with the help of UNL graph in figure 2.

In the given example ‘agt’ relation specifies relation between agent ‘boy’ (who did work), and verb ‘eat’. The ‘obj’ relation specifies relation between object ‘potato’, and verb ‘eat’. The ‘plc’ relation specifies relation between verb ‘eat’, and place ‘kitchen’ where action took place. Finally, attributes represent the circumstances under which the node is used. These are the annotations made to nodes. In the given example sentence (II), ‘@pl’ signifies that UW is used as a plural. The attribute ‘@def’ is a kind of specifying attribute used in case of general specification (normally conveyed by determiners). In the given example sentence (II) it represents node ‘the’. The ‘@entry’ attribute represents sentence head. Attributes give additional information that is not expressed via UW or Relations. UW’s, Relations and Attributes, each have its predefined specifications given by UNDL Foundation [14]. UNL uses the concept of KCIC (Key Concept in Context) to link every UW of the UNL ontology to the UNL documents where the UW is included. Every UW must be registered in the UNL ontology for realizing this inter-linkage of UWs crossing UNL documents [15].

Figure 2. UNL graph of sentence ‘The boy eat potatoes in the kitchen’ [13].

1.3 Problem statement

During the last couple of years, in the field of Natural Language Processing, UNL (i.e., Universal Networking Language) has been an area of immense research among researchers. Punjabi language is world’s 9th most widely spoken language [7]. There are relatively less efforts in the field of computerization and development of this language. This paper illustrates UNLization of Punjabi Natural Language for UC-A1, UGO-A1, and AESOP-A1 with IAN (i.e., Interactive Analyzer) tool using X-Bar approach. These are the corpus provided by UNDL Foundation [1]. Prior to X-Bar approach, no standard approach had ever been followed for UNLization.

Previously none of the articles have discussed the errors/ discrepancies in UNLization system with the help of example sentences. Section 8.3 of this paper highlights errors/ discrepancies in UNLization system. This paper also discusses the UNLization process in depth step by step with the help of tree diagrams and tables.

1.4 Our contribution

For UNLization of Punjabi language 24 NRules, 48 DRules, 982 TRules and 771 Dictionary entries are created, and the proposed system has been tested on UC-A1, UGO-A1, and AESOP-A1. The UNLization module for Punjabi language has been built using X-Bar theory (later described in section 4).

Since proposed system uses X-Bar therefore it is generic and can be reused for similar languages (only dictionary and UWs needs to be replaced for the target language). For example, Hindi natural language to UNL system was developed using this proposed system and it won GOLDEN medal for IV Olympiad held by UNDL foundation [5]. With the help of these Punjabi language resources, Agarwal and Kumar [16] have developed a multilingual cross-domain client application prototype for UNLization and NLization for NLP applications. Additionally, on top of this proposed system, a public platform for developing language-independent applications has been developed and tested by Agarwal and Kumar [17].

The work presented in this paper had been submitted for UNL Olympiad II, III, and IV conducted by UNDL Foundation in July 2013, March 2014, and November 2014 for UC-A1, UGO-A1, and AESOP-A1, respectively. The proposed UNLization module for Punjabi language was selected for top 10 best UNLization Grammars. The results are available at UNDL Foundation’s website [3–5].

2. Related work

UNL aims at coding, storing, disseminating and retrieving information independently of the original language in which it was expressed. In addition to translation, the UNL has been exploited for several other different tasks in natural language engineering, such as multilingual document generation, summarization, text simplification, information retrieval and semantic reasoning. Sérasset and Boitet [18] have viewed UNL as the future ‘html of the linguistic content’. This section aims to cover the important work that has been done in the field of NLP using UNL.

Multilingual information processing through UNL has been proposed by Bhattacharyya [19] and Dave and Bhattacharyya [20]. Their system performs sentence level encoding of English, Hindi and Marathi into the UNL form and then decodes this information into Hindi and Marathi.

Lafourcade and Boitet [21] have found that during DeConversion process there are some lexical items, called UWs, which are not yet connected to lemmas.

Choudhary and Bhattacharyya [22] have performed the text clustering using UNL representation. Martins *et al* [23] have analyzed unique features of UNL taking inferences from Brazilian Portuguese-UNL EnConverting task. 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* [24] have proposed an EnConversion tool for Tamil. 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.

Tomokiyo and Chollet [25] have proposed a VoiceUNL to represent speech control mechanisms within the UNL framework. The proposed system has been developed to support Speech to Speech Machine Translation (SSMT).

Dhanabalan and Geetha [26] have proposed a DeConverter for Tamil language. It is a language-independent generator that provides synchronously a framework for word selection, morphological generation, syntactic generation and natural collocation necessary to form a sentence. The proposed system involves the use of language-specific, linguistic-based DeConversion rules to convert the UNL structure into natural language sentences.

Martins [27] addressed the color categorization problem from the multicultural knowledge representation perspective.

Surve *et al* [28] have proposed an 'Agro-Explorer' as a meaning based, interlingua search engine designed specifically for the agricultural domain covering English, Hindi and Marathi languages. 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.

Boguslavsky *et al* [29] have proposed a multi-functional linguistic processor, 'ETAP-3', as an extension of 'ETAP' machine translation system to a UNL based machine translation system.

Jiang *et al* [30] have explored UNL as a facilitator for communication between languages and cultures. 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.

Cardeñosa *et al* [31] have proposed an extended Markup Language (XML) UNL model for knowledge-based annotation.

Hajlaoui and Boitet [32] have proposed a pivot XML based architecture for multilingual, multiversion documents through UNL.

Montesco and Moreira [33] have proposed Universal Communication Language (UCL) derived from UNL.

Iyer and Bhattacharyya [34] have proposed the use of semantic information to improve case retrieval in case-based reasoning systems. They have proposed a UNL based system to improve the precision of retrieval by taking into account semantic information available in words of the problem sentence. The proposed system makes use of WordNet to find semantic similarity between two concepts.

Martins *et al* [35] 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. 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.

Blanc [36] has performed the integration of 'Ariane-G5' to the proposed French EnConverter and French DeConverter. '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. Its DeConversion process also takes place in two steps. The first step is lexical and structural transfer from the UNL graph to an equivalent dependency tree and second step is the generation of the French sentence.

Shi and Chen [37] have proposed UNL DeConverter for Chinese language. Pelizzoni and Nunes [38] have introduced 'Manati' DeConversion model as a UNL mediated Portuguese-Brazilian sign language human-aided machine translation system.

Keshari and Bista [39] have proposed the architecture and design of UNL Nepali DeConverter for 'DeCo' tool. The proposed system has two major modules, namely, syntax planning module and morphology generation module.

Ramamritham [40] have further improved 'Agro-Explorer' to develop 'aAQUA' an online multilingual, multimedia agricultural portal for disseminating information from and to rural communities. The 'aAQUA' makes use of novel database systems and information retrieval techniques like intelligent caching, offline access with intermittent synchronization, semantic-based search, etc.

Alansary *et al* [41] have proposed the concept of language-independent Universal Digital Library within UNL framework. A UNL based Library Information System (LIS) has been implemented as a proof of concept.

Karande [42] has proposed a multilingual search engine with the use of UNL. Before building index terms or list of keywords for implementing multilingual search engine using UNL, the conversion of contents from any language to UNL is required. Spider is responsible for input to the convertor. Convertor converts these native language pages into UNL. The language of web pages is identified by the convertor and then convertor sends this page to that corresponding language server. The native language page is translated into UNL by language server. Convertor also converts the query into UNL. Now, since query as well as index terms are available in UNL form, all searching operations are performed on UNL. Finally, result is now converted into native language in which the query was asked.

Adly and Alansary [43] had introduced a prototype of Library Information Systems that uses the Universal Networking Language (UNL) as a means for translating the metadata of books. This prototype is capable of handling the bibliographic information of 1000 books selected from the catalogs of Bibliotheca Alexandrina (B.A.).

Rouquet and Nguyen [44] have proposed an interlingual annotation of texts. They have explored the ways to enable multimodal multilingual search in large collections of images accompanied by texts.

Boudhh and Bhattacharyya [45] have proposed the unification of UW dictionaries by using WordNet ontology. 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. Kumar and Sharma [46, 47] have proposed an EnConversion and DeConversion system to convert Punjabi language to UNL and vice versa. However, their system uses adhoc rules and is limited to only particular set of corpus. The database of these EnConversion rules is created on the basis of morphological, syntactic and semantic information of Punjabi language as recommended by Uchida [48], Dave and Bhattacharyya [20], Dey and Bhattacharyya [49].

Jadhav and Bhattacharyya [50] have proposed an unsupervised rule-based approach using deep semantic processing to identify only relevant subjective terms.

Agarwal and Kumar [16] have developed a multilingual cross-domain client application prototype for UNLization and NLization for NLP applications. Additionally, on top of this proposed system, a public platform for developing language-independent applications has been developed and tested by Agarwal and Kumar [17].

3. UNLization process

UNLization is a rule based approach. UNLization process aims to convert NL document to UNL document. UNLization is done using IAN that involves creation of

Normalization Grammar (N-Grammar or NRules), Dictionary, Disambiguation Grammar (D-Grammar or DRules), and Transformation Grammar (T-Grammar or TRules) as shown in figure 3 in accordance with specifications provided by UNDL Foundation [14]. In order to do UNLization for a given natural language, the corpus is first converted to that particular natural language manually. Each of the step shown in figure 3 has been explained in the following subsections.

3.1 Normalization

It is like a pre-processing phase. Before applying Transformation rules or Disambiguation rules on natural language document, the document is first of all normalized. Normalization is done so that the original sentence can be converted into more refined form. Some of the steps are given below.

- Replacing contractions
Don't > do not, he'll > he will
- Replacing abbreviations
U > you
- Reordering
Would you > you would
- Filling gaps and ellipses
Next week > in the next week
- Removing extra content
, say, > $\emptyset$

Normalization is done with the help of Normalization Grammar (N-Grammar). An example N-Grammar is given in (IV).

(%a, [don't]):= (%c,[do not]);
(%b, [dr.]):= (%d,[doctor]); (IV)

Here, '%a' refers to node [don't], and '%b' refers to node [dr.]. These rules will replace the node '%a' with [do not], and the node '%b' with [doctor].

Let us consider an example paragraph given in (V) as an input text to IAN.

Figure 3. UNLization process overview.

Dr. Peter H. Smith isn't coming on July 1st. He'll be in another meeting in N.Y. I'll check with him another date asap. Would u be available next week, say, around 2 PM? (V)

Normalized form of example sentence given in (V) is given in (VI).

Doctor Peter H Smith is not coming on 01/07. He will be in another meeting in New York. I will check with him other date as soon as possible. You would be available next week around 14:00:00? (VI)

3.2 Tokenization

In UNLization, Tokenization refers to splitting the natural language input into nodes, i.e., the tokens or processing units of the UNL framework [51]. During Tokenization the string like 'hare and tortoise' is split into 5 tokens viz. [hare][][and][][tortoise], if these entries are provided in the dictionary. However, if any word is not found in dictionary, then that word is considered as temporary word. An attribute 'TEMP' is assigned to that word.

The following tokens are created by default by IAN [51].

- i. SCOPE – Scope
- ii. SHEAD – Sentence head (the beginning of a sentence)
- iii. STAIL – Sentence tail (the end of a sentence)
- iv. CHEAD – Scope head (the beginning of a scope)
- v. CTAIL – Scope tail (the end of a scope)
- vi. TEMP – Temporary entry (entry not found in the dictionary)
- vii. DIGIT – Any sequence of digits (i.e.: 0,1,2,3,4,5,6,7,8,9)

Tokenization is done by IAN on the basis of following rules:

- i. The system matches first the longest entry in the dictionary, from left to right.
- ii. The highest frequent entry comes first in case of entries with the same length.
- iii. The first to appear in the dictionary comes first in case of entries with the same length and same frequency.
- iv. The feature TEMP (temporary) is assigned to the strings that are not found in the dictionary.
- v. The feature DIGIT is assigned to the strings exclusively formed by digits.
- vi. The feature SHEAD (Sentence head) is automatically assigned to the beginning of the paragraph, and the feature STAIL (Sentence tail) is assigned to the end of the paragraph.
- vii. No other tokenization and punctuation is done by the system (e.g.: blank spaces and punctuation signs are not automatically recognized).

3.3 Disambiguation rules (D-rules)

Tokenization is also controlled with the help of D-Rules (Disambiguation Grammar). There can be several scenarios where a single natural language word has several dictionary entries. In such cases D-Rules helps in tokenization by selecting the desired dictionary entry. Consider an example sentence given in (VII).

This is necessary (VII)

Assume that the dictionary entries are as:

```
[[]{}]"(BLK)<eng,0,0>;
[this][{}]"(LEX = D, POS = DEM, att=
@proximal)<eng,0,0>;
[this][{}]"(LEX = R, POS = DEP, att=
@proximal)<eng,0,0>;
[is][{}]"(LEX = I, POS = AUX)<eng,0,0>;
[is][{}]"(LEX = V, POS = COP)<eng,0,0>;
[necessary][{}]"(LEX = N, POS = NOU)<eng,0,0>;
[necessary][{}]"(LEX = J, POS = ADJ)<eng,0,0>;
```

By default Tokenization will happen like:
(this,D)(BLK)(is,I)(BLK)(necessary,N)

Consider the scenarios in which Tokenization will modify if following D-Grammars are written:

Scenario #1:

- i. D-Rules:
(D)(BLK)(I):=0;
According to this rule the probability of occurrence of a node having 'D' as a feature, followed by blank space and a node having 'I' as a feature is zero.
- ii. Tokenization:
(this,D)(BLK)(is,V)(BLK)(necessary,N)

Scenario #2:

- i. D-Rules:
(D)(BLK)(I):=0; (D)(BLK)(V):=0;
- ii. Tokenization:
(this,R)(BLK)(is,V)(BLK)(necessary,N)

Scenario #3:

- i. D-Rules:
(D)(BLK)(I):=0; (D)(BLK)(V):=0; (V)(BLK)(J):=1;

ii. Tokenization:

(this,R)(BLK)(is,V)(BLK)(necessary,J)

All the three processes namely Normalization, Tokenization, and Transformation are done on each of the sentence in the corpus and we get the final UNL of the corpus.

3.4 Transformation (UNLization)

After Normalization and Tokenization, UNLization process starts. UNLization is done with the help of Transformation Grammar (T-Rules). Transformation using X-Bar approach is explained in section 1.6. Simple Transformation process is explained with the help of an example sentence given in (VIII).

Hare and Tortoise (VIII)

After Tokenization of example sentence given in (VIII) with IAN tool, five lexical items are identified as given in (IX).

[Hare]{} "Hare"(LEX = N, POS = NOU, GEN = MCL)<eng,0,0>;
 []{} ""(BLK)<eng,0,0>;
 [and]{} ""(LEX = C, POS = CCJ, rel = and)<eng,0,0>;
 []{} ""(BLK)<eng,0,0>;
 [Tortoise]{} "Tortoise"(LEX = N, POS = NOU, GEN = MCL)<eng,0,0>; (IX)

The process of UNLization of example sentence (VIII) is given in table 1.

UNL of example sentence given in (VIII) generated by IAN is given in (X).

{unl}
 and(tortoise;hare)
 {/unl} (X)

4. Role of X-bar in UNLization

Ever since the UNL programme was launched UNLization and NLization had been done by various computational linguists and other experts based on their own understanding. No systematic or standardized approach had been followed by anybody. It was realized that as the natural language sentences become more and more complex, the number of TRules increases significantly and conflicts arise

with the previously made TRules [52]. Thus, there was a need to follow a more systematic approach for UNLization. So, X-Bar approach was followed by computational linguists working under UNL programme.

The X-Bar theory postulates that all human languages share certain structural similarities, including the same underlying syntactic structure, which is known as the 'X-Bar' [53]. The X-bar abstract configuration is depicted in figure 4 [52]. Here,

Table 1. UNLization of example sentence (VIII).

Input Sentence: Hare and Tortoise		
1.	TRule	(%a,BLK):=;
	Description	Here, '%a' refers to blank node [] having attribute 'BLK'. This rule is fired twice consecutively and it removes all the blank spaces.
	Action	Original nodes :
	Taken	[Hare][][and][][Tortoise] Resultant nodes: [Hare][and][Tortoise]
2.	TRule	(([SHEAD CHEAD], %01) (N, {NOU PPN }, %a) (C, and, CCJ, %b) (N, {NOU PPN }, %c) ([STAIL CTAIL], %02) := (NA(%c; %a), +N, +AND,%d);
	Description	Here, '%01', refers to scope head, '%a' refers to node [Hare], '%b' refers to node [and], '%c' refers to node [tortoise], and '%02' refers to scope end. 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 'AND' and 'N' are assigned to this new node.
	Action	Original nodes : [Hare][and][Tortoise]
	Taken	Resultant nodes: NA([Tortoise], [Hare])
3.	TRule	(NA (%a; %b), AND, %01) := and(%a; %b);
	Description	Here, '%a' refers to node [Tortoise], and '%b' refers to node [Hare]. This rule renames 'NA' relation to actual 'and' relation. This is the final output generated by IAN.
	Action	Original nodes :
	Taken	NA([Tortoise], [Hare])
		Resultant nodes: and([Tortoise], [Hare])

Figure 4. X-Bar abstract configuration [52].

- **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.

Consider an example sentence given in (XI).

The beautiful tortoise won the race. (XI)

X-Bar configuration of example sentence given in (XI) is shown in figure 5.

In example sentence given in (XIV), first determiner 'the' is promoted up to its maximal projection 'DP', adjective beautiful is promoted to its maximal projection 'JP', and the noun 'tortoise' is promoted to its intermediate projection 'NB'. 'JP' and 'NB' combines to form the intermediate noun projection 'NP' which later combines with 'DP' to form a noun phrase 'NP'.

In example sentence given in (XIV) consider the substring 'won the race'. Here, verb 'won' is promoted up to its intermediate projection 'VB', determiner 'the' is promoted to its maximal projection 'DP', and the noun 'race' is promoted to its intermediate projection 'NB'. 'DP' and 'NB' combines to form the maximal noun projection 'NP' which later combines with 'VB' to form a verbal phrase 'VP', which is in its intermediate projection. 'VB' combines with the maximal projection 'NP' ('NP' is the maximal projection of substring 'the beautiful tortoise' as shown in

Figure 5. X-Bar structure of example sentence (XI).

figure 5) to form the verbal phrase ‘VP’ which is in its maximal projection.

4.1 Transformation (UNLization) using X-bar approach

While using X-Bar approach, UNLization is performed in five steps, i.e., Parsing, Transformation, Dearborization, Interpretation, and Rectification. These are shown in figure 6.

Each of these five processes is explained below.

A. Parsing

When an input document is tokenized by IAN then it is in list form. In Parsing, initial list structure is converted to tree structure as shown in figure 7. In parsing, syntactic analysis of the normalized input is performed.

Consider an example sentence given in (XII).

John did not kill Mary. (XII)

Figure 6. UNLization steps.

Figure 7. Conversion of list structure to tree structure by Parsing.

After tokenization and removing blank spaces the list structure is like [John][did][not][kill][Mary]. After parsing this list structure gets converted to tree structure as shown in figure 8.

B. Transformation

The tree which is obtained after Parsing is in its surface structure. Some dependency relations that are not represented directly inside the list and which are important in the UNLization process are not present in the surface structure. For instance, in case of ‘John did not kill Mary’, the NP ‘John’ will be represented at the position of specifier of the IP ‘did not kill Mary’, but it is important to move it to the position of specifier of the VP ‘kill Mary’. In order to do that, we have to convert the surface structure into a deep structure. The deep syntactic structure is supposed to be more suitable to the semantic interpretation. In transformation phase, this surface tree structure is converted into a modified tree in order to expose its inner organization, i.e., the deep syntactic structure as shown in figure 9.

Consider the same example sentence given in (XII). The surface structure which is obtained after Parsing is converted to deep structure as shown in figure 10.

C. Dearborization

The UNL graph is a network rather than a tree. In order to be converted to UNL, the deep syntactic structure obtained after transformation, must be ‘dearborized’, i.e., transformed into a network structure. This is done by rewriting X-Bar relations (XP, XB) as head-driven syntactic relations (XS, XC, XA). In Dearborization, tree structures are converted into head-driven structures. These head driven structures are further converted into intermediate semantic relations like ‘VS’, ‘VC’, ‘VA’, etc. In these relations, the first character of ‘VS’, ‘VC’, and ‘VA’, i.e., ‘V’ indicates that the first argument is a verb, while second character ‘S’, ‘C’, and ‘A’ indicates that the second argument of a relation is specifier, complement or an adjunct, respectively. The Network structure of example sentence given in (XII) obtained after Dearborization is shown in figure 11.

D. Interpretation

In Interpretation, syntactic network obtained after dearborization is simply mapped to a semantic network by analyzing the arguments of each relation as shown in figure 12.

In the above example, node ‘not’ assigns the attribute ‘@not’ to the Universal word ‘kill’. The node ‘not’ does not form any relation with any node.

E. Rectification/ Post-processing

In post-processing, the resulting graph is adjusted according to UNL standards in order to eliminate contradictions and redundancies. For example, consider the rule given in (XIII).

Figure 8. Parsing (List to Tree Structure).

Figure 9. Conversion of surface structure to deep structure.

Figure 10. Transformation (surface structure to deep structure).

(@pl, {@multal|@paucal|@all|@both});=(-@pl); (XIII)

This rule eliminates the redundancy of '@pl'. The idea of plural is already being conveyed by '@multal',

'@paucal', '@all', or '@both'. Therefore '@pl' is redundant and should be fixed. So here '@pl' is removed with the help of Post-processing rules.

Figure 11. Dearborization (Tree Structure to Network Structure).

Figure 12. Interpretation (Mapping Syntactic Network to Semantic network).

5. Working of the proposed system with an example sentence

The working of the proposed system has been explained with the help of an example sentence taken from AESOP-A1 corpus. AESOP-A1 is a latest corpus provided by

5.1 Normalization

Since the corpus is in a paragraph form hence with the help of N-Grammer as given in (XIV), the paragraph is broken down to 13 sentences as shown in table 2 as an input to IAN.

(%a,“|”)(%b,^STAIL):=(%a)(STAIL)(%b) (XIV)

Here, ‘%a’ refers to node ‘|’ which indicates sentence end in Punjabi language, similar to the punctuation ‘.’ in English language. This N-Grammer adds the tag <STAIL> after every sentence end. The tag <SHEAD> is assigned automatically after <STAIL>. So because of this N-Grammer the paragraph is broken into sentences as given in table 2.

Out of the sentences given in table 2, UNLization of Punjabi natural language is explained with the help of an example sentence given in (XV).

Example 1: ਖਰਗੋਸ਼ ਨੇ ਇਕ ਦਿਨ ਕੱਢੂਕੁੰਮੇ ਦੇ ਛੋਟੇ ਪੈਰਾਂ ਅਤੇ ਹੌਲੀ ਚਾਲ ਦਾ ਮਜ਼ਾਕ ਉਡਾਇਆ (XV)

k̥hargōsh nē ik din kacchūkummē dē chōtē pairāṁ atē haulī cāl dā mazāk uḍāiā
The hare one day ridiculed the short feet and slow pace of the tortoise

UNDL Foundation. AESOP-A1 contains the famous story of ‘The Tortoise and the Hare’ from Aesop’s Fables. For UNLization, AESOP-A1 is manually converted into Punjabi language and uploaded to ‘NL-Input’ tab of IAN. The subsections below gives detailed explanation of UNLization of Punjabi natural language.

5.2 Tokenization

During tokenization of example sentence given in (XV) with IAN tool, twenty two lexical items are identified as given in (XVI).

Table 2. Sentences of AESOP-A1.

Sl. No.	Punjabi sentences of AESOP-A1
1.	ਖ਼ਰਗੋਸ਼ ਅਤੇ ਕੱਛੂਕੁੰਮਾ <i>kḥargōsh atē kacchūkummā</i> <i>The Hare and the Tortoise</i>
2.	ਖ਼ਰਗੋਸ਼ ਨੇ ਇਕ ਦਿਨ ਕੱਛੂਕੁੰਮੇ ਦੇ ਛੋਟੇ ਪੈਰਾਂ ਅਤੇ ਹੌਲੀ ਚਾਲ ਦਾ ਮਜ਼ਾਕ ਉਡਾਇਆ <i>kḥargōsh nē ik din kacchūkummē dē chōṭē pairām atē haulī cāl dā mazāk uḍāiā</i> <i>The Hare one day ridiculed the short feet and slow pace of the Tortoise</i>
3.	ਕੱਛੂਕੁੰਮੇ ਨੇ ਜਵਾਬ ਦਿੱਤਾ <i>kacchūkummē nē javāb dittā</i> <i>The Tortoise replied</i>
4.	ਭਾਵੇਂ ਤੂੰ ਹਵਾ ਦੀ ਤਰ੍ਹਾਂ ਤੇਜ਼ ਹੈ, ਮੈਂ ਤੈਨੂੰ ਦੌੜ ਵਿਚ ਹਰਾ ਦੇਵਾਂਗਾ <i>bhāvēm tūm havā dī tarhām tēz hai, maiṁ tainūm daur vic harā dēvāṅgā</i> <i>Though you be swift as the wind, I will beat you in a race</i>
5.	ਖ਼ਰਗੋਸ਼ ਨੂੰ ਇਹ ਗੱਲ ਬਿਲਕੁਲ ਅਸੰਭਵ ਲੱਗੀ ਅਤੇ ਪ੍ਰਸਤਾਵ ਲਈ ਰਾਜੀ ਹੋ ਗਿਆ <i>kḥargōsh nūm ih gall bilkul asambhav laggī atē prastāv lai rājī hō giā</i> <i>The Hare believed her assertion to be simply impossible and assented to the proposal</i>
6.	ਉਹਨਾਂ ਨੇ ਮੰਨਿਆ ਕਿ ਲੁੰਬੜੀ ਪਥ ਚੁਣੇਗੀ ਅਤੇ ਟੀਚਾ ਮਿੱਥੇਗੀ <i>uhnām nē manniā ki lūmbṛī path cūṇēgī atē ṭicā mitthēgī</i> <i>They agreed that the Fox should choose the course and fix the goal</i>
7.	ਦੌੜ ਲਈ ਨਿਰਧਾਰਿਤ ਦਿਨ ਦੋਵਾਂ ਨੇ ਇਕੱਠੇ ਸ਼ੁਰੂ ਕੀਤਾ <i>daur lai nirdhārit din dauvām nē ikṭhē shurū kītā</i> <i>On the day appointed for the race the two started together</i>
8.	ਕੱਛੂਕੁੰਮਾ ਰੁਕਿਆ ਨਹੀਂ <i>kacchūkummā rukiā nahīm</i> <i>The Tortoise did not stop</i>
9.	ਉਹ ਹੌਲੀ ਪਰ ਲਗਾਤਾਰ ਪਥ ਦੇ ਅਖੀਰ ਤਕ ਸਿੱਧਾ ਜਾਂਦਾ ਗਿਆ <i>uh haulī par lagātār path dē akhīr tak siddhā jāndā giā</i> <i>She went on with a slow but steady pace straight to the end of the course</i>
10.	ਖ਼ਰਗੋਸ਼ ਰਸਤੇ ਵਿਚ ਥੱਲੇ ਲੰਮਾ ਪੈ ਗਿਆ ਅਤੇ ਇਕ ਰੁੱਖ ਹੇਠ ਝਪਕੀ ਲੈ ਗਿਆ <i>kḥargōsh rastē vic thallē lammā pai giā atē ik rukkh hēṭh jhapkī lai giā</i> <i>The Hare laid down by the wayside and took a nap under a tree</i>
11.	ਆਖਰ, ਉਹ ਉੱਠਿਆ ਅਤੇ ਦੌੜਿਆ ਜਿੰਨੀ ਤੇਜ਼ ਉਹ ਦੌੜ ਸਕਦਾ ਸੀ, ਪਰ ਇਹ ਬਹੁਤ ਦੇਰ ਸੀ <i>ākhar, uh uṭṭhiā atē dauṛiā jinnī tēz uh daur sakdā sī, par ih bahut dēr sī</i> <i>At last, he woke up and ran as fast as he could, but it was too late</i>
12.	ਕੱਛੂਕੁੰਮਾ ਪਹਿਲਾਂ ਹੀ ਦੌੜ ਨੂੰ ਜਿੱਤ ਚੁੱਕਿਆ ਸੀ <i>kacchūkummā pahilām hī daur nūm jitt cukkiā sī</i> <i>The Tortoise had already won the race</i>
13.	ਹੌਲੀ ਪਰ ਲਗਾਤਾਰ ਤਰੱਕੀ ਦੌੜ ਨੂੰ ਜਿੱਤਦੀ ਹੈ <i>haulī par lagātār tarkkī daur nūm jittdī hai</i> <i>Slow but steady progress wins the race</i>

[ਖਰਗੋਸ਼] {} "hare"(LEX=N,POS=NOU,GEN=MCL,NUM=SNG)<pan,0,0>;
 [ਨੇ] {-} {} ""(LEX=P,POS=PPS,rel=agt)<pan,0,0>;
 [ਇਕ ਦਿਨ] {} "one day"(LEX=N,POS=NOU,rel=tim)<pan,0,0>;
 [ਕੱਛੂਰੁੰਮੇ] {} "tortoise"(LEX=N,POS=NOU,NUM=SNG)<pan,0,0>;
 [ਦੇ] {} ""(LEX=P,POS=PPS,rel=mod)<pan,0,0>;
 [ਛੋਟੇ] {} "small"(LEX=J,POS=ADJ)<pan,0,0>;
 [ਪੈਰਾਂ] {} "foot"(LEX=N,POS=NOU,NUM=PLR)<pan,0,0>;
 [ਅਤੇ] {} "and"(LEX=C,POS=COO,rel=and)<pan,0,0>;
 [ਹੌਲੀ] {} "slow"(LEX=J,POS=ADJ,NUM=SNG)<pan,0,0>;
 [ਚਾਲ] {} "pace"(LEX=N,POS=NOU,NUM=SNG)<pan,0,0>;
 [ਦਾ] {} ""(LEX=D,POS=ART,att=@def)<pan,0,0>;
 [ਮਜ਼ਾਕ ਉਡਾਇਆ] {} "ridicule"(LEX=V,POS=VER,att=@past,NUM=SNG)<pan,0,0>;

Eleven blank spaces are also identified as :-

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

Here, 'LEX' represents lexical category, 'N' represents noun, 'P' represents preposition, 'J' represents adjective, 'C' represents conjunction, 'D' represents determiner, 'V' represents verb, 'POS' represents part-of-speech, 'NOU' represents common noun, 'PPS' represents postposition, 'ADJ' represents adjective, 'COO' represents coordinating conjunction, 'ART' represents article, 'VER' represents full verb, 'GEN' represents gender, 'MCL' represents masculine, 'rel' represents relation, 'agt' represents agent relation, 'tim' represents time relation, 'mod' represents modifier relation, 'and' represents and relation, 'att' holds the attribute value of a node, '@def' represents definite, '@past' represents past attribute, 'NUM' represents number whose value could be

either 'SNG' for singular or 'PLR' for plural, and 'BLK' is the attribute given to the blank space. 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 NLization.

5.3 Parsing

After Parsing, the example sentence given in (XV) is converted to tree structure as shown in figure 13.

The UNL generated after Parsing phase is given in (XVII).

NP:01(foot;small)
 NP:02(pace;slow)
 and:03(02;01)
 NP:04(03;tortoise)
 VB:05(ridicule;04)
 VB:06(05;one day)
 VP(06;hare)

(XVII)

5.4 Transformation

In Transformation phase, the surface tree structure is converted into a modified tree in order to expose its inner organization, i.e., the deep syntactic structure. However, in the given example there is no need to convert the surface

Figure 13. Tree structure for example sentence given in (XV).

Figure 14. UNL graph of example sentence given in (XV).

tree structure as given in figure 14 into deep syntactic structure.

5.5 Dearborization

In Dearborization phase the example sentence given in (XV) is converted to network structure.

The UNL generated after Dearborization phase is given in (XVIII).

NA:01(foot,small)
 NA:02(pace,slow)
 and:03(02:01)
 VA(ridicule,one day)
 VS(ridicule,hare)
 NA:04(03,tortoise)
 VC(ridicule,04) (XVIII)

5.6 Interpretation

In Interpretation phase the example sentence given in (XV) is converted to semantic network.

The UNL generated after Parsing phase is given below in equation (XIX).

agt(ridicule.@past,hare.@def)
 tim(ridicule.@past,one day)
 obj(ridicule.@past,:04)
 mod:04(:03,tortoise.@def)
 and:03(02,01)
 mod:02(pace,slow)
 mod:01(foot.@pl,short) (XIX)

The UNL generated after Interpretation phase does not require any rectification or post processing because there are no contradictions and redundancies. The final UNL graph of example sentence given (XV) is shown in figure 14.

6. Usage of the proposed system in large NLP tasks

Unlike the traditional approaches and techniques in natural language processing, scope and use of UNL is not limited to one domain. . How UNL can be exploited for other NLP tasks has been covered under sub-sections 6.2 to sub-section 6.6. Sub-section 6.1 below explains the differences and advantages of UNL.

6.1 Differences and advantages of UNL over other traditional approaches

Unlike a particular technique/ method/ algorithm/ approach, UNL can be exploited for several other goals like machine translation, text to speech systems, question answering systems, sentiment analysis, text summarization, etc. UNL is not limited to only one goal. In subsections below the scope of the proposed system for these applications has been explored.

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

Figure 15. Approach for UNL-ization and NL-ization of n natural languages.

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. This is shown in figure 15.

UNLization and NLization modules can be used to develop UNL based applications like question answering system, machine translation, sentiment analysis, text summarization, etc. In order to support these applications, it is important to analyze UNL from these applications point of view.

6.2 UNL based multilingual cross domain client application prototype

In order to utilize the IAN and EUGENE resources for all the languages which are a part of UNL programme, a ‘Multilingual Cross Domain Client Application Prototype’ has been developed [16]. The proposed client application prototype is successfully able to use IAN and EUGENE resources of Punjabi natural language without actually logging into the account on UNL web and perform UNL-ization and NL-ization. Thus, their proposed system is 100% accurate. The correctness of results depends on the F-Measure (described in next section) of UNL-ization and NL-ization module of the selected language.

6.3 UNL for question-answering system

QA system provides the exact answer instead of providing the listing of all relevant documents containing the answer of query. With the capability of UNL, multilingual QA answering system can be built with the use of IAN and EUGENE tools. The general architecture of UNL based multilingual QA system for Punjabi language has been depicted in Figure 16.

As shown in figure 16, the analysis module is used to convert the Punjabi natural language sentence to UNL. In order to UNLize any given Punjabi corpus and question inputted by the user; Dictionary, Transformation Rules (TRules) and Disambiguation Rules (DRules) need to be created for IAN (Interactive Analyzer) tool. UNL crawler would be responsible for searching the UNL repository for the UNL document of input question. It attempts to find for an exact match and gives the answer. Optimizer would eliminate the superfluous or extra information retrieved by the previous module. Depending on full match/partial match, it will give the most likely answer among all the possible solutions. UNL representation generated by Optimizer will be modified so as answer can be generated by the Generation Module developed using EUGENE (dEp-to-sUrface GENERator) engine of the native language. The proposed system can be used to create the UNL Repository which can be used by the UNL Crawler.

For such a web-based system to be used by global audience, a public platform for developing language-independent applications has been developed and made available online [17]. An initial prototype of proposed QA system has been integrated with this platform and is available online for some sample sentences available at <http://www.punjabinlp.com>.

6.4 UNL for machine translation

UNL is language-independent and machine-tractable database, with the help of many relations, and attributes. This feature of UNL can be exploited for machine translation. With the help of IAN module, any corpus can be converted to UNL and using EUGENE module UNL can be converted back to natural language. Using the proposed system, we can convert any Punjabi corpus to UNL. This Punjabi corpus (now in the form of

Figure 16. UNL based multilingual QA system for Punjabi language.

UNL) can be converted to any foreign language if analysis module of that foreign language has been developed. Similarly when we have analysis module for Punjabi language, we can convert any language corpus (given in the form of UNL) to Punjabi natural language corpus.

A UNL based Machine Translation system had been developed by Kumar [13]. The fluency score of 3.61 (on a 4-point scale), adequacy score of 3.70 (on a 4-point scale), and BLEU score of 0.72 was achieved by their proposed system.

6.5 UNL for sentiment analysis

Sentiment Analysis (SA) plays a vital role in decision making process. The decisions of the people get affected by the opinions of other people. Researchers heavily rely on supervised approaches to train the system for SA. These systems take into account all the subjective words and/or phrases. But not all of these words and phrases actually contribute to the overall sentiment of the text. The proposed architecture UNL based SA system has been depicted in figure 17.

Jadhav and Bhattacharyya [50] have proposed an unsupervised rule-based approach using deep semantic processing to identify only relevant subjective terms. Their UNL rule-based system had an accuracy of 86.08% for English Tourism corpus whereas for English Product corpus an accuracy of 79.55% was achieved. In the proposed approach, UNL graph had been generated for the input text. Rules are applied on the graph to extract relevant terms. The sentiment expressed in these terms is used to figure out the overall sentiment of the text. Results on binary

sentiment classification have shown promising results. It has been observed that relations like *agt*, *obj*, *aoj*, *and*, *mod* and *man* relations play vital role in SA of text. The system proposed in this paper can be used as a ‘UNL Generator’ on the input text of the given language.

6.6 UNL for text summarization

Most of existing works on text summarization rely on surface information of documents. Employing the surface information, these approaches select the best sentences and list them together to summarize the whole text. Without employing the semantic information, these approaches have a great drawback. The generated summaries are often not much readable and contain a lot of redundancies. However, for UNL documents, the UNL semantic information is very useful to summarize and generate high quality summaries.

The process of UNL based text summarization system is shown in figure 18. The natural language input document need to be summarized is converted to UNL document with

Figure 17. UNL based sentiment analysis system.

Figure 18. Process of UNL based text summarization system.

the use IAN engine of corresponding language. Then, a sentence score is calculated by the weight of each word constituting the sentence. Weight of each word is computed according to its term frequency and inverted document frequency. After applying the scoring process to a UNL document, the sentences with the highest scores are selected. In the next step, redundant words are removed, as the selected sentences still contain redundant words. Most of the redundant words are the modifiers. These modifiers are easily identified by considering the UNL semantic relations. The semantic relations such as *man*, *mod* and *ben* imply the modifying relationship. If an auxiliary node does not help in clarifying the head node, the auxiliary node can then be removed without distorting the total meaning. In order to improve readability and naturalness, some selective sentences are combined in next stage of processing. Sentences that employ the same UW can be merged to reduce the sentential redundancy [54]. Finally, to generate the natural language from summarized UNL document, EUGENE engine of native language is used to NLize the UNL document. The document summarization system developed by Sornlertlamvanich *et al* [54] showed very

promising results. For a sample corpus they compared plain text summarization results with their developed UNL document summarization system. After Plain text summarization the result consists of 5 sentences and 67 words whereas UNL document summarization results in 4 sentences and 47 words. The system proposed in this paper can be used in the above architecture for UNLizing natural language text.

After analyzing these various NLP applications from UNL perspective (subsections 6.3. to 6.6), and understanding the advantages of UNL over other traditional approaches (subsection 6.1), UNL can be viewed as an alternative techniques for various NLP applications like sentiment analysis, machine translation, question answering system, etc. If any of these above mentioned NLP applications are UNL based, then that will be language independent, could integrate other NLP UNL based applications, and would be available online for worldwide audience. There will not be any change in the architecture/codebase of such a UNL based NLP application in order to support any other new language.

7. Evaluation metrics

F-measure is the measure of a grammar's accuracy. The F-measure / F1-score is calculated with the help of online tool developed by UNDL foundation available at UNL-arium [2]. Two parameters required for the calculation of F-measure are Precision and Recall. F-measure is calculated by the formulae given in (XX) [14].

$$F\text{-measure} = 2 * \{ (Precision * Recall) / (Precision + Recall) \} \quad (XX)$$

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

8. Results and discussions

The proposed system was tested on UC-A1, UGO-A1, and AESOP-A1 corpus. Dataset details and results are described in subsections 8.1 and 8.2.

8.1 Dataset details

UC-A1, UGO-A1, and AESOP-A1 are the corpus provided by UNDL Foundation. These corpus were provided in English language to all the participant languages. These corpora were downloaded and manually converted to Punjabi natural language for UNLizing. UC-A1 contains 100 Natural Language sentences, and UGO-A1 comprises of 250 sentences, both covering all the major part-of-speeches. Table 3 shows the categorization of UC-A1, and UGO-A1.

Table 3. Categorization of UC-A1 and UGO-A1.

Sl. No.	Type	Number of sentences in UC-A1	Number of sentences in UGO-A1	Number of sentences in AESOP-A1
1.	Temporary words	5	14	-
2.	Numbers and Ordinals	10	25	-
3.	Nouns	15	35	-
4.	Adjectives	7	15	-
5.	Determiners	9	20	-
6.	Prepositions	6	25	-
7.	Pronouns	5	20	-
8.	Time	5	12	-
9.	Verb	9	24	-
10.	Conjunctions	9	20	-
11.	Sentence structures	20	40	13

AESOP-A1 contains the famous story of ‘The Tortoise and the Hare’ from Aesop’s Fables. For testing these corpus, the Punjabi natural language corpus was uploaded on unlweb and UNLization of entire corpus was done in one go using IAN (separately for each corpus). The output UNL of the entire was copied from IAN console and saved in UTF-8 format. This is the actual output file. Unl web also provides the expected output file (in UNL format). These actual and expected UNL files are then uploaded at UNL-arium and the F-Measure is then calculated by the system as described in section 7.

8.2 Results and testing details

The values of Precision, Recall, number of processed, returned and correct sentences for UC-A1, UGO-A1, and AESOP-A1 is given in table 4.

The work presented in this paper had been submitted for UNL Olympiad II, III, and IV conducted by UNDL Foundation in July 2013, March 2014, and November 2014 for UC-A1, UGO-A1, and AESOP-A1, respectively. The proposed UNLization module for Punjabi language was selected for top 10 best UNLization Grammars. The results are available at UNDL Foundation’s website [3–5].

Since proposed system uses X-Bar therefore it is generic and can be reused for similar languages (only dictionary

and UWs needs to be replaced for the target language). For example, Hindi natural language to UNL system was developed using this proposed system and it won GOLDEN medal for Olympiad IV held by UNDL foundation [5] for corpus AESOP-A1. The F-Measure for this Hindi system was 0.923.

For the corpora UC-A1 and UGO-A1 that were tested on the proposed system, none of them had an accuracy/ F-Measure of 1.00. This was because of the ‘overall discrepancy’ (refer table 5) was found in the proposed system which occurred due to less accurate TRules, and DRules. These resulted in the incorrect attributes getting assigned to few nodes in the actual output. These discrepancies cannot be justified because they are valid and they should not be a part of any UNLization module. Work is still going on for making the proposed UNLization module more refined and accurate.

8.3 Errors/ discrepancies in the UNLization system

As explained above, the accuracy of the proposed system is calculated using an online tool (provided by UNDL foundation) available at UNL-arium. The F-Measure of a corpus calculated by the tool depends upon the following discrepancies:

Table 4. Testing details of UC-A1, UGO-A1, and AESOP-A1.

Sl. No.	Parameters	UC-A1 Value	UGO-A1 Value	AESOP-A1 Value
1.	Sentences processed	100	250	13
2.	Sentences returned	100	250	13
3.	Sentences correct	97	247	13
4.	Precision	0.970	0.988	1.00
5.	Recall	0.970	0.992	1.00
6.	F-Measure	0.970	0.990	1.00

Table 5. Formulas to calculate discrepancies and their meaning.

Value to be calculated	Formulae	Meaning of keywords
Discrepancy of relations	$(\text{exceeding_relations} + \text{missing_relations}) / \text{total_relations}$	<p>exceeding_relations is the number of relations present in the actual output but absent from the expected output</p> <p>missing_relations is the number of relations absent from the actual output but present in the expected output</p> <p>total_relations is the sum of the total number of relations in the actual output and in the expected output</p> <p>exceeding_UW is the number of UW's present in the actual output but absent from the expected output</p>
Discrepancy of UW's	$(\text{exceeding_UW} + \text{missing_UW}) / \text{total_UW}$	<p>missing_UW is the number of UW's absent from the actual output but present in the expected output</p> <p>total_UW is the sum of the total number of UW's in the actual output and in the expected output</p>
Overall Discrepancy	$((3 * (\text{exceeding_relations} + \text{missing_relations})) + (2 * (\text{exceeding_UW} + \text{missing_UW})) + (\text{exceeding_attribute} + \text{missing_attribute})) / ((3 * \text{total_relations}) + (2 * \text{total_UW}) + (\text{total_attribute}))$	<p>exceeding_attribute is the number of attributes present in the actual output but absent from the expected output</p> <p>missing_attribute is the number of attributes absent from the actual output but present in the expected output</p> <p>total_attribute is the sum of the total number of attributes in the actual output and in the expected output</p>

Table 6. Scores calculated based on above formulas for sample sentences by online tool developed by UNDL foundation available at UNL-arium.

Sl. No.	Sample sentence	Actual output	Expected output	Status	Calculated score	Explanation
1.	The book on the table	plc(book:05@def, table :01.@on.@def)	plc(book.@def, table.@def.@on)	CORRECT	1.00	Actual and Expected results match
2.	The book under the table	agt(book:05@def, table:01.@def.@under)	plc(book.@def, table.@def.@under)	WRONG	0.00	The discrepancy of relations is above the limit
3.	The books	book:01	book.@pl.@def	CORRECT	0.80	There are missing attributes
4.	His book	agt(kill;ram)	pos(book:03,00 :01.@3.@male)	NON-RETURNED	-1.00	The actual output does not correspond to any sentence in the expected output

Table 7. Comparison of proposed system with other UNLization modules for UNL Olympiad II [3].

Sl. No.	Language	Rank	F-Measure	Corpus Used
1.	Slovenian	1	1.000	UC-A1
2.	German	2	1.000	
3.	Serbian	3	1.000	
4.	Croatian	4	1.000	
5.	Swedish	5	1.000	
6.	Latin	6	1.000	
7.	Turkish	7	1.000	
8.	Afrikaans	8	1.000	
9.	<u>Punjabi (Proposed System)</u>	9	0.975	
10.	Khmer	10	0.950	

1. Discrepancy of relations
2. Discrepancy of UW's
3. Overall Discrepancy

A result is considered “RETURNED” when the output is a graph (i.e., all the nodes are interlinked), and the output is made only of UW's (i.e., without natural language words).

A result is considered “CORRECT” when the discrepancy of relations between the actual and the expected output is less than 0.3, and the discrepancy of UW's between the actual and the expected output is less than 0.3, and the overall discrepancy is less than 0.3. Meaning of these discrepancies and formulae to calculate these are given in table 5.

Table 6 shows the calculated scores by online tool (available at UNL-arium) of some of the sample sentences by using the above mentioned formulas.

8.4 Comparison of proposed system with other systems

Ever since the Universal Networking Language (UNL) programme started in 1996, researchers and computational linguists across the globe have actively participated in this.

English, Chinese, Georgian, Greek, Hindi, Punjabi, Portuguese, Russian, Ukrainian, Vietnamese, Slovenian, etc., are some of the languages which are part of the UNL programme. The 37 languages which took part in UNL Olympiad II (UNLization) are: Afrikaans, Armenian, Assamese, Bengali, Chinese, Croatian, Estonian, German, Greek (Modern), Gujarati, Hindi, Indonesian, Japanese, Kannada, Kashmiri, Khmer, Laotian, Latin, Malay, Malayalam, Manipuri, Marathi, Nepali, Persian, Punjabi, Sanskrit, Serbian, Sindhi, Sinhala, Slovenian, Swahili, Swedish, Tamil, Telugu, Thai, Turkish, and Vietnamese.

Table 7 compares the proposed system with other UNLization modules which were identified as best UNLization modules by UNDL foundation for UNL Olympiad II [3].

The 20 languages which took part in UNL Olympiad III (UNLization) are: Afrikaans, Bulgarian, Chinese, Estonian, Georgian, Greek (Modern), Hindi, Kannada, Khmer, Malay, Nepali, Panjabi, Persian, Portuguese, Russian, Sinhala, Slovenian, Telugu, Ukrainian, and Vietnamese.

Table 8 compares the proposed system with other UNLization modules which were identified as best UNLization modules by UNDL foundation for UNL Olympiad III [4].

Table 8. Comparison of proposed system with other UNLization modules for UNL Olympiad III [4].

Sl. No.	Language	Rank	F-Measure	Corpus used
1.	Slovenian	1	1.000	UGO-A1
2.	<i>Punjabi (Proposed System)</i>	2	0.990	
3.	Bulgarian	3	0.976	
4.	Ukrainian	4	0.946	
5.	Russian	5	0.942	

Table 9. Comparison of proposed system with other UNLization modules for UNL Olympiad IV [5].

Sl. No.	Language	Rank	F-Measure	Corpus used
1.	Ukrainian	1	1.000	AESOP-A1
2.	Russian	2	1.000	
3.	<i>Punjabi (Proposed System)</i>	3	1.000	
4.	German	4	1.000	
5.	Bulgarian	5	0.923	
6.	Hindi	6	0.923	

The 25 languages which took part in UNL Olympiad IV (UNLization) are: Armenian, Baatonum, Bengali, Bosnian, Bulgarian, Czech, Estonian, French, Georgian, German, Greek (Ancient), Greek (Modern), Hindi, Khmer, Latin, Oriya, Panjabi, Persian, Portuguese, Russian, Sinhala, Tamil, Telugu, Ukrainian, and Vietnamese.

Table 9 compares the proposed system with other UNLization modules which were identified as best UNLization modules by UNDL foundation for UNL Olympiad IV [5].

9. Conclusion and future scope

With the help of 24 NRules, 48 DRules, 982 TRules and 771 Dictionary entries, the proposed system has been tested on UC-A1, UGO-A1, and AESOP-A1 for Punjabi language, and their F-measure comes out to be 0.970, 0.990, and 1.00, respectively.

The proposed system can further be improved to achieve F-Measure 1.000 for all Punjabi corpuses. The work can be extended to develop NLization module for Punjabi language. The proposed system has the potential to be used in applications like multilingual information service, information retrieval system, question and answering system, UNL based search engine, and machine translation system, etc.

Acknowledgement

This Publication is an outcome of the R&D work undertaken in the project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology,

Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

References

- [1] UNDL Foundation 2016 [Online]. Available: <http://www.undlfoundation.org/undlfoundation/>
- [2] Calculation of F-Measure 2013 [Online]. Available: <http://www.unlweb.net/unlarium/index.php?tools=fmeasure>
- [3] Second Olympiad 2013 [Online]. Available: http://www.unlweb.net/wiki/II_UNL_Olympiad
- [4] Third Olympiad 2014 [Online]. Available: http://www.unlweb.net/wiki/III_UNL_Olympiad
- [5] Fourth Olympiad 2014 [Online]. Available: http://www.unlweb.net/wiki/IV_UNL_Olympiad
- [6] NLP Applications. [Online]. Accessed 2014, Available: http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol1/hks/intro.html
- [7] Punjabi Language [Online]. Accessed 2015, Available: http://en.wikipedia.org/wiki/Punjabi_language
- [8] Agarwal V and Kumar P 2013a UNL-ization of Numbers and Ordinals in Punjabi with IAN. *Int. J. Nat. Lang. Comput. (IJNLC)* 2(3): 61–69
- [9] Uchida H, Zhu M and Senta T D 1999 *The UNL, A gift for a millennium*. Tokyo: Institute of Advanced Studies, The United Nations University
- [10] Introduction to UNL [Online]. Accessed 2015, Available: http://www.unlweb.net/wiki/Introduction_to_UNL
- [11] Uchida H and Zhu M 2001 The universal networking language beyond machine translation. In: *Proc. Int. Symposium on Language in Cyberspace*, Seoul, Korea, pp. 1–15
- [12] Uchida H 2005 *Universal Networking Language (UNL): Specifications Version 2005*, UNDL Foundation
- [13] Kumar P 2012 *UNL Based Machine Translation System for Punjabi Language*. Ph.D. thesis, Thapar University, Patiala

- [14] Agarwal V and Kumar P 2013b UNL-ization of Punjabi with IAN. In: *Proc. of the 11th Workshop on Asian Language Resources*, Nagya, Japan, pp. 61–69
- [15] Uchida H and Zhu M 2005 UNL 2005 for providing knowledge infrastructure. In: *Proc. Semantic Computing Workshop*, Chiba, Japan, pp. 1–12
- [16] Agarwal V and Kumar P 2016 A multilingual cross-domain client application prototype for UNL-ization and NL-ization for NLP applications. *Dig. Scholarsh. Humanit.* doi:<http://dx.doi.org/10.1093/lilc/fqw022>
- [17] Agarwal V and Kumar P 2017 A public platform for developing language-independent applications. *Dig. Scholarsh. Humanit.* doi:<https://doi.org/10.1093/lilc/fqw063>
- [18] Sérasset G and Boitet C 2000 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 UNL French deconverter. In: *Proc. Int. Conf. on Computational Linguistics*, Saarbruecken, Germany, pp. 768–774
- [19] Bhattacharyya P 2001 Multilingual information processing through Universal Networking Language. In: *Indo UK Workshop on Language Engineering for South Asian Languages*, pp. 1–10
- [20] Dave S and Bhattacharyya P 2001 Knowledge extraction from Hindi text. *J. Inst. Electron. Telecommun. Eng.* 18(4): 1–12
- [21] Lafourcade M and Boitet C 2002 UNL Lexical selection with conceptual vectors. In: *Proc. Int. Conf. on Language Resources and Evaluation*, Spain, pp. 1958–1964
- [22] Choudhary B and Bhattacharyya P 2002 Text clustering using Universal networking language representation. In: *Proc. 11th Int. Conf. on World Wide Web*, Hawaii, USA, pp. 1–7
- [23] Martins R T, Rino L H M, Nunes M G V and Oliveira O N 2002 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, pp. 8–13
- [24] Dhanabalan T, Saravanan K and Geetha T V 2002 Tamil to UNL EnConverter. In: *Proc. Int. Conf. on Universal Knowledge and Language*, Goa, India, pp. 1–16
- [25] Tomokiyo M and Chollet G 2003 VoiceUNL : a proposal to represent speech control mechanisms within the Universal Networking Digital Language. In: *Proc. Int. Conf. on the Convergence of Knowledge, Culture, Language and Information Technologies*, Alexandria, Egypt, pp. 1–6
- [26] Dhanabalan T and Geetha T V 2003 UNL DeConverter for Tamil. In: *Proc. Int. Conf. on the Convergence of Knowledge, Culture, Language and Information Technologies*, Alexandria, Egypt, pp. 1–6
- [27] Martins R 2003 HOW many colors should be in the rainbow? Representing color names in knowledge bases. In: *International Conference on the Convergence of Knowledge, Culture, Language and Information Technologies*, Alexandria, Egypt.
- [28] Surve M, Singh S, Kagathara S, Venkatasivaramasastry K, Dubey S, Rane G, Saraswati J, Badodekar S, Iyer A, Almeida A, Nikam R, Perez C G and Bhattacharyya P 2004 AgroExplorer: a meaning based multilingual search engine. In: *Proc. Int. Conf. on Digital Libraries*, Delhi, India, pp. 1–13
- [29] Boguslavsky I M, Iomdin L and Sizov V G 2005 Interactive EnConversion by means of the ETAP-3 system. *Universal Network Language: Advances in Theory and Applications*. Cardeñosa J, Gelbukh A, Tovar E, (Eds) México, Research on Computing Science, pp. 230–240
- [30] Jiang C, Tissiani G, Falquet G and Rodolfo P 2005 Facilitating communication between languages and cultures: a computerized interface and knowledge base. *Universal Network Language: Advances in Theory and Applications*, Cardeñosa J, Gelbukh A, Tovar E, (Eds) México, Research on Computing Science, pp. 358–368.
- [31] Cardeñosa J, Gallardo C and Iraola L 2005 An XML-UNL model for knowledge based annotation. *Universal Network Language: Advances in Theory and Applications*, Cardeñosa J, Gelbukh A, Tovar E, (Eds) México, Research on Computing Science, pp. 300–308
- [32] Hajlaoui N and Boitet C 2005 A Pivot XML-based architecture for multilingual, multiversion documents: parallel monolingual documents aligned through a central correspondence descriptor and possible use of UNL. *Universal Network Language: Advances in Theory and Applications*, Cardeñosa J, Gelbukh A, Tovar E, (Eds) México, Research on Computing Science, pp. 309–325
- [33] Montesco C E and Moreira D A 2005 UCL-universal communication language. *Universal Network Language: Advances in Theory and Applications*, Cardeñosa J, Gelbukh A, Tovar E, (Eds) México, Research on Computing Science, pp. 326–336
- [34] Iyer J A and Bhattacharyya P 2005 Using semantic information to improve case retrieval in case-based reasoning systems. *Universal Network Language: Advances in Theory and Applications*, Cardeñosa J, Gelbukh A, Tovar E, (Eds) México, Research on Computing Science, pp. 347–358
- [35] Martins R T, Hasegawa R, Graças M and Nunes V 2005 Hermeto: A NL-UNL enconverting environment. *Universal Network Language: Advances in Theory and Applications*, Cardeñosa J, Gelbukh A, Tovar E, (Eds) México, Research on Computing Science, pp. 254–260
- [36] Blanc É 2005 About and around the French Enconverter and the French Deconverter. *Universal Network Language: Advances in Theory and Applications*, Cardeñosa J, Gelbukh A, Tovar E, (Eds) México, Research on Computing Science, pp. 157–166
- [37] Shi X and Chen Y 2005 A UNL Deconverter for Chinese Universal Network Language. *Universal Network Language: Advances in Theory and Applications*, (Eds) Cardeñosa J, Gelbukh A, Tovar E, México, Research on Computing Science, pp. 167–174
- [38] Pelizzoni J M and Nunes M V 2005 Flexibility, configurability and optimality in UNL deconversion via multi-paradigm programming. *Universal Network Language: Advances in Theory and Applications*, (Eds) Cardeñosa J, Gelbukh A, Tovar E, México, Research on Computing Science, pp. 175–194
- [39] Keshari B and Bista S K 2005 UNL Nepali Deconverter. In: *Proc. 3rd International CALIBER*, Cochin, pp. 70–76
- [40] Ramamritham 2006 aAqua: a database-backed multilingual, multimedia community forum. In: *Proc. Int. Conf. on Management of Data*, Chicago, USA, pp. 784–786
- [41] Alansary S, Nagi M and Adly N 2006 Towards a language-independent Universal Digital Library. In: *Proc. 2nd Int.*

- Conf. on Universal Digital Library, Alexandria, Egypt*, pp. 1–10
- [42] Karande J B 2007 Multilingual search engine: implementation using UNL. In: *Proc. Int. Conf. on Semantic Web and Digital Libraries*, Bangalore, India, pp. 1–7
- [43] Adly N and Alansary S 2009 Evaluation of Arabic machine translation system based on the Universal Networking Language. *Nat. Lang. Proc. Inf. Syst. Lect. Notes Comput. Sci.* 5723: 243–257
- [44] Rouquet D and Nguyen H 2009 Interlingual annotation of texts in the OMNIA project. In: *Proc 4th Int. Conf. on Language and Technology*, Poland, pp. 1–5
- [45] Boudhh S and Bhattacharyya P 2009 Unification of Universal Word dictionaries Using WordNet ontology and similarity measures. In: *Proc 7th Int. Conf. on Computer Science and Information Technologies*, Armenia, pp. 271–283
- [46] Kumar P and Sharma R K 2012 Punjabi to UNL enconversion system. *Sadhana Acad. Proce. Eng. Sci.* 37(2): 299–318
- [47] Kumar P and Sharma R K 2013 Punjabi DeConverter for generating Punjabi from Universal Networking Language. *J. Zhejiang Univ. Sci. C* 14(3): 79–196
- [48] Uchida H 1987 ATLAS: Fujitsu Machine Translation system. *Proc. MT Summit*, Japan, pp. 129–134
- [49] Dey K and Bhattacharyya P 2005 Universal Networking Language based analysis and generation of Bengali case structure constructs. *Universal Network Language: Advances in Theory and Applications*, Cardeñosa J, Gelbukh A, Tovar E, (Eds) México, Research on Computing Science, pp. 215–229
- [50] Jadhav N K and Bhattacharyya P 2014 Dive deeper: Deep Semantics for Sentiment Analysis. In: *5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA 2014)*, Baltimore, USA
- [51] Tokenization [Online]. Accessed 2015, Available: <http://www.unlweb.net/wiki/Tokenization>
- [52] Agarwal V 2013 UNL-ization of Punjabi with IAN. M.E Thesis, Thapar University, Patiala.
- [53] X-Bar theory 2014 [Online]. Available: http://www.unlweb.net/wiki/X-bar_theory
- [54] Sornlertlamvanich V, Potipiti T and Charoenporn T 2001 UNL Document Summarization. In: *Proc. First International workshop on Multimedia Annotation*, Tokyo