# UNL-ization of Punjabi with IAN

**Vaibhav Agarwal**
M.E(S.E), Thapar University
Patiala, India
vaibhavagg123@gmail.com

**Parteek Kumar**
Assistant Professor, Thapar University
Patiala, India
parteek.bhatia@gmail.com

## Abstract

UNL-ization is the process of converting Natural Language resource to Universal Natural Language (*i.e.,* UNL). UNL is based on Interlingua approach, specifically designed by UNDL foundation for storing, summarizing, representing and describing information in a format which is independent to a natural language. This paper illustrates UNL-ization of Punjabi language with the help of IAN (*i.e., I*nteractive *AN*alysis) tool. UNL-ization of major part-of-speeches of a Natural language *viz* Preposition, Conjunction, Determiner, Verb, Noun, Adjective, Time, Numbers and Ordinals has been done. In this paper UNL-ization process is explained with the help of three example sentences. Total 257 TRules and 623 Dictionary entries have been created, and the system has been tested successfully for Corpus500 (provided by UNDL Foundation) for Five hundred Punjabi sentences, comprising of all the major part-of-speeches and its F-Measure comes out to be 0.936 (on a scale of 0 to 1).

## 1 Introduction

In UNL, UNL-ization and NL-ization are the two approaches that are being followed. UNL-ization is the process of converting the given Natural Language resource to UNL whereas NL-ization is the reverse process. Both UNL-ization and NL-ization are independent to each other. *I*nteractive *An*alyzer (*i.e.,* IAN), and d*E*ep-to-s*U*rface *GENE*rator (*i.e.,* EUGENE) are two online tools provided by UNDL foundation used for UNL-ization and NL-ization, respectively. With the help of TRules and Analysis Grammar for that particular Natural language, the resource of that Natural Language can be UNL-ized using IAN. TRules and Analysis Grammar is user made, in accordance with specifications provided by UNDL Foundation [13][12]. Universal Networking language is based on the concept of Universal words, Relations and Attributes, each having its predefined specifications as given by UNDL foundation [16].UNL-ization should not be compared with Machine Translation or interlanguage conversion. UNL can be used for summarizing, representing, storing, and describing information in a natural language independent format. In case of translation of Natural languages using Rule based UNL approach it has an advantage as explained below.

Assume there are '*n*' number of different natural languages which needs to be translated into one another. Now if we are using the approach of UNL for converting those '*n*' natural languages into each other then only '*2*n*' components needs to be developed. This is because now only *2* conversions needs to be done for every natural language *viz* natural language to UNL and then from UNL to that natural language. Now in order to convert our source language into other '*n-1*' languages only its UNL representation is required because the system for conversion of UNL to those '*n-1*' natural languages has already been developed by computational linguist experts of those '*n-1*' languages. 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. Therefore the proposed UNL system for Punjabi language will certainly be very helpful for more than 91 million Punjabi language users [4]. In Figure 1 below NL *1*, NL *2*, ….., NL *n* represents *n* different natural languages.

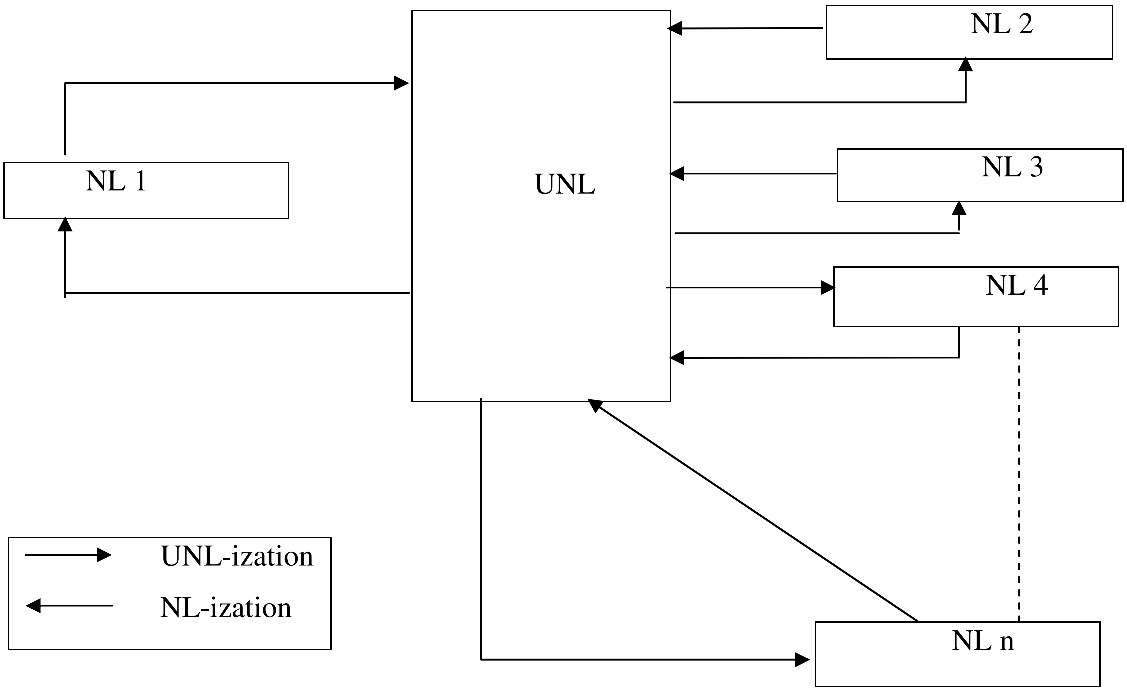

Figure 1. Approach for UNL-ization and NL-ization of *n* natural languages

## 2 Related Work

A prototype system for converting Brazilian Portuguese into UNL and deconverting UNL expressions into Brazilian Portuguese with *'EnCo'* and *'DeCo'* tools, respectively have been proposed by Martins *et al.* (1997) [10]. Their system consists of three important sub-modules, namely, the lexical, the syntactic and the semantic modules. Martins *et al.* (2005) 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 [11]. 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.

For developing a UNL based MT system Semantically Relatable Sequence (SRS) based approach have been used by Mohanty *et al.* (2005) [9]. Kumar and Sharma (2012) have proposed an Enconversion system to convert Punjabi language to UNL [8]. Dey and Bhattacharyya (2005) have presented the computational analysis of complex case structure of Bengali for a UNL based MT System [3]. 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. Blanc (2005) has performed the integration of 'Ariane-G5' to the proposed French EnConverter and French DeConverter. *'Ariane-G5'* is a generator of MT systems [1]. 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.* (2005) have proposed a multifunctional linguistic processor, *'ETAP-3'*, as an extension of 'ETAP' machine translation system to a UNL based machine translation system [2]. Choudhury *et al.* (2005) have proposed a framework for converting Bangla to UNL and have also proposed a procedure to construct Bangla to UNL dictionary [5]. The system developed by Lafourcade (2005) uses ant colony algorithm for semantic analysis and fuzzy UNL graphs for EnConversion process [6].

## 3 Features of Punjabi Language

Gill (2008) has explained the features of Punjabi language [7]. Punjabi has word classes in the form of noun, pronoun, adjective, cardinal, ordinal, main verb, auxiliary verb, adverb, postposition, conjunction, interjection and particle. For example ਸੜਕ *saṛak 'road'* is used as feminine gender while ਟਰੱਕ *ṭarakk 'truck'* is used as masculine gender. Punjabi has six types of pronouns. These are: personal pronouns, *e.g.*, ਮੈਂ maiṃ *'i'*; reflexive pronouns, *e.g.*, ਆਪ *āp*; demonstrative pronouns, *e.g.*, ਉਹ *uh 'that'*; indefinite pronouns, *e.g.*, ਕੋਈ *kōī*, ਕੁਝ *kujh*, *etc.*; relative pronouns, *e.g.*, ਜੋ *jō* and ਜਿਹੜਾ *jihṛā* and interrogative pronouns, *e.g.*, ਕੌਣ *kauṇ 'who' etc.* In Punjabi language, adjectives usually precede the nouns but follow the pronouns. The examples of adjectives following pronouns are, 'ਉਹ ਸੋਹਣੀ ਹੈ' *'uh sōhṇī hai' 'She is beautiful'.* Punjabi verbs change forms for gender, number, person, and tense. The verbs have assigned transitivity and causality. In Punjabi, there are two auxiliary verbs – ਹੈ *hai* for present tense and ਸੀ *sī* for past tense. Adverbs can indicate manner, time, place, condition *etc.* For example, ਉੱਪਰ *uppar 'upon'*, ਉੱਤੇ *uttē 'over'*, *etc.* are some Punjabi adverbs. Postpositions are similar to prepositions in English. These link noun, pronoun, and phrases to other parts of the sentence. For example ਉੱਤੇ *uttē 'over'*, ਦਾ *dā 'of'* *etc.* 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) [7].

## 4 Implementation

UNL-ization of Five Hundred Punjabi sentences has been done with the help of 257 TRules and 623 Dictionary entries. Apart from these one thousand sentences, all the numbers and ordinals upto fourteen digits can be UNL-ized with same TRules and Analysis Dictionary, while any number of similar Punjabi sentences can also be UNL-ized with same TRules and few more Dictionary entries as required in those sentences.

The UNL-ization process of prepositions, conjunctions, Nouns and adjectives is explained in subsequent subsections with the help of Example sentences.

### 4.1 UNL-ization of Prepositions

In UNL Prepositions are represented by either relations or by relations and attributes. The UNL-ization process for prepositions has been illustrated with the help of a simple example sentence (1).

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

*mēj uttē pairis bārē tasvīrāṃ tōṃ bināṃ kitāb*
*The book on the table about Paris without pictures*

After tokenization of example sentence (1) with IAN tool thirteen lexical items are identified as given in (2).

[ਮੇਜ]{}"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=man,att=@withou t)<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>; …(2)

The process of UNL-ization of example sentence (1) has been illustrated in Table 1. In Description column of table 1, English translation of nodes is not shown because the order of appearance of those translated nodes is not same as in natural language input sentence.

Table 1. UNL-ization process for example sentence (1)

| Sno | TRule fired | Description |
|---|---|---|
| 1 | (%a,BLK):=; | Here, *%a* refers to blank node. This rule is fired six times and deletes all the blank spaces. |

| | | |
|---|---|---|
| 2 | (N,%a)(P,PRE,rel,att,%b):=(%a,+att=%b,+rel=%b,+N); | Here, *%a* refers to node **[ਮੇਜ]** *[mēj]*, *%b* refers to node **[ਉੱਤੇ]** *[uttē]*. This rule deletes the node *%b* and gives its attributes to node *%a*. |
| 3 | (N,%a)(P,PRE,rel,att,%b):=(%a,+att=%b,+rel=%b,+N); | Here, *%a* refers to node **[ਤਸਵੀਰਾਂ]** *[tasvīrāṃ]*, *%b* refers to node **[ਤੋਂ ਬਿਨਾਂ]** *[tōṃ bināṃ]*. As above, this rule deletes the node *%b* and gives its attributes to node *%a*. |
| 4 | (N,rel=man,att,%a)(N,%b):=(NA(%b;%a),+MAN,+N); | Here, *%a* refers to node **[ਤਸਵੀਰਾਂ]** *[tasvīrāṃ]*, *%b* refers to node [ਕਿਤਾਬ] *[kitāb]*. 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. |
| 5 | (N,%a)(P,cnt,%b)(N,%c):=(NA(%c;%a,+att=%b),%d,+N,+CNT); | Here, *%a* refers to node **[ਪੈਰਿਸ]** *[pairis]*, *%b* refers to node [ਬਾਰੇ] *[bārē]*, and *%c* refers to node **[ਤਸਵੀਰਾਂ@without]** *[tasvīrāṃ*@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*. |
| 6 | (N,rel=plc,att,%a)(N,^rel,%b):=(NA(%b;%a),+PLC,+N); | Here, *%a* refers to node **[ਮੇਜ@on]** *[mēj*@on*]*, *%b* refers to node **[NA(NA(**ਕਿਤਾਬ**;ਤਸਵੀਰਾਂ@without);ਪੈਰਿਸ@about)]** *[NA(NA(kitāb;tasvīrāṃ@without);pairis@about)]*. This rule results into a relation 'NA' with first, second arguments as *%b* and *%a* respectively. |
| 7 | (N,PLR,^@pl,%a):=(%a,+@pl); | Here, *%a* refers to node **[ਤਸਵੀਰਾਂ@without]** *[tasvīrāṃ*@without*]*. This rule adds attribute '@pl' to node *%a*. |
| 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])]** *[NA([kitāb];[tasvīrāṃ@without@pl])]*, *%b* refers to node **[ਪੈਰਿਸ@about]** *[pairis@about]*, *%w* refers to **[NA([NA([**ਕਿਤਾਬ**];[ਤਸਵੀਰਾਂ@without@pl])];[ਪੈਰਿਸ@about])]** *[NA([NA([kitāb];[tasvīrāṃ@without@pl])];[pairis@about])]*, *%c* refers to node [ਮੇਜ@on] *[mēj*@on*]*, *%r* refers to original node. 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. |
| 9 | (NA(NA(%a;%b),MAN,%w;%c),CNT,%r):=(%w),(NA(%a;%c),+CNT); | Here, *%a* refers to node **[ਕਿਤਾਬ]** *[kitāb]*, *%b* refers to node **[ਤਸਵੀਰਾਂ@without@pl]** *[tasvīrāṃ@without@pl]*, *%c* refers to **[ਪੈਰਿਸ@about]** *[pairis@about]*, *%w* refers to node **[NA([**ਕਿਤਾਬ**];[ਤਸਵੀਰਾਂ@without@pl])]** *[NA([kitāb];[tasvīrāṃ@without@pl])]*, and *%r* refers to node **[NA([NA([**ਕਿਤਾਬ**];[ਤਸਵੀਰਾਂ@without@pl])];[ਪੈਰਿਸ@about])]** *[NA([NA([kitāb];[tasvīrāṃ@without@pl])];[pairis@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* respec- |

| | | |
|---|---|---|
| | | tively. |
| 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])];[ਮੇਜ@on])]** *[NA([NA([kitāb];[tasvīrāṃ@without@pl])];[mēj@on])]*, *%c* refers to node **[ਮੇਜ@on]** *[mēj@on]*, *%w* refers to node **[NA([ਕਿਤਾਬ];[ਤਸਵੀਰਾਂ@without@pl])]** *[NA([kitāb];[tasvīrāṃ@without@pl])]*, *%a* refers to node **[ਕਿਤਾਬ]** *[kitāb]*, *%b* refers to node **[ਤਸਵੀਰਾਂ@without@pl]** *[tasvīrāṃ@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. |
| 11 | (NA(%a;%b),CNT) :=cnt(%a;%b); | Here, *%a* refers to node **[ਕਿਤਾਬ]** *[kitāb]*, *%b* refers to node **[ਪੈਰਿਸ@about]** *[pairis@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. |
| 12 | (NA(%a;%b),PLC): =plc(%a;%b) | Here, *%a* refers to node **[ਕਿਤਾਬ]** *[kitāb]*, *%b* refers to node **[ਮੇਜ@on]** *[mēj@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. |
| 13 | (NA(%a;%b),MAN ):=plc(%a;%b) | Here, *%a* refers to node **[ਕਿਤਾਬ]** *[kitāb]*, *%b* refers to node **[ਤਸਵੀਰਾਂ@without@pl]** *[tasvīrāṃ@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.<br>Now all the natural language words are replaced by their universal words and final output is generated by IAN as shown in (3). |

The UNL generated is given in (3).

```
{org}
ਮੇਜ 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}                                          ...(3)
```

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

## 4.2 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 (4).

Example 1: ਇਕ ਸੋਹਣੀ ਗੱਡੀ, ਇਕ ਮਹਿੰਗੀ ਗੱਡੀ ਅਤੇ ਇਕ ਨਵਾਂ ਪਿਆਲਾ ...(4)

*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 (4) with IAN tool, twenty lexical items are identified as shown in (5).

```
[ਸੋਹਣੀ]{}"beautiful"(LEX=J,POS=ADJ,GEN=FEM)<pan,0,0>;
[ਗੱਡੀ]{}"car"(LEX=N,POS=NOU,NUM=SNG)<pan,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)

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 (4) has been illustrated in Table 2.

Table 2. UNL-ization process for example sentence (4)

| Sno | TRule fired | Description |
|---|---|---|
| 1 | (%a,BLK):=; | Here, *%a* refers to blank node. This rule is fired nine times and deletes all the blank spaces. |
| 2 | (D,att,%a)(J,%b)(N,%c):=(NA(%c,+att=%a;%b),+N,+NOU,+MOD); | Here, *%a* refers to node **[ਇਕ]** *[ik] [a]*, *%b* refers to node **[ਸੋਹਣੀ]** *[sōhṇī] [beautiful]*, and node *%c* refers to **[ਗੱਡੀ]** *[gaḍḍī] [car]*. 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'. |
| 3 | (D,att,%a)(J,%b)(N,%c):=(NA(%c,+att=%a;%b),+N,+NOU,+MOD); | Here, *%a* refers to node **[ਇਕ]** *[ik] [a]*, *%b* refers to node **[ਮਹਿੰਗੀ]** *[mahiṅgī] [expensive]*, and node *%c* refers to **[ਗੱਡੀ]** *[gaḍḍī] [car]*. 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'. |
| 4 | (N,NOU,%a)(C,%b)(N,NOU,%c):=(NA(%c;%a),+N,+NOU,+AND); | Here, *%a* refers to node **[NA([ਗੱਡੀ@indef];[ਸੋਹਣੀ])]** *[NA([gaḍḍī@indef];[sōhṇī])] [NA([car@indef];[beautiful])]*, *%b* refers to node **[,]** and *%c* refers to node **[NA([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])]** *[NA([gaḍḍī@indef];[mahiṅgī])] [NA([car@indef];[expensive])]*. 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'. |
| 5 | (D,att,%a)(J,%b)(N,%c):=(NA(%c,+att=%a;%b),+N,+NOU,+MOD); | Here, *%a* refers to node **[ਇਕ]** *[ik] [a]*, *%b* refers to node **[ਨਵਾਂ]** *[navāṃ] [new]*, and node *%c* refers to **[ਪਿਆਲਾ]** *[piālā] [mug]*. 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'. |
| 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])])]** *[NA([NA([gaḍḍī@indef];[mahiṅgī])];[NA([gaḍḍī@indef];[sōhṇī])])] [NA([NA([car@indef];[expensive])];[NA([car@indef];[beautiful])])]*, *%b* refers to node **[ਅਤੇ]** *[atē] [and]* and *%c* refers to node **[NA([ਪਿਆਲਾ@indef];[ਨਵਾਂ])]** *[NA([piālā@indef];[navāṃ])] [NA([mug@indef];[new])]*. 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’. |
| 7 | (NA(%a;%b ),MOD):=mod(%a;%b); | Here, *%a* refers to node **[ਪਿਆਲਾ@indef]** *[piālā@indef] [mug@indef]*, *%b* refers to **[ਨਵਾਂ]** *[navāṃ] [new]*. This rule changes the name of relation from ‘NA’ to ‘mod’ keeping same arguments as in original node, as required in the final UNL. |
| 8 | (NA(%a;%b ),MOD):=mod(%a;%b); | Here, *%a* refers to node **[ਗੱਡੀ@indef]** *[gaḍḍī@indef] [car@indef]*, *%b* refers to **[ਸੋਹਣੀ]** *[sōhṇī] [beautiful]*. This rule changes the name of relation from ‘NA’ to ‘mod’ keeping same arguments as in original node. |
| 9 | (NA(%a;%b ),MOD):=mod(%a;%b); | Here, *%a* refers to node **[ਗੱਡੀ@indef]** *[gaḍḍī@indef] [car@indef]*, *%b* refers to **[ਮਹਿੰਗੀ]** *[mahiṅgī] [expensive]*. This rule changes the name of relation from ‘NA’ to ‘mod’ keeping same arguments as in original node, as required in the final UNL. |
| 10 | (NA(%a;%b ),AND):=and(%a;%b); | Here, *%a* refers to node **[mod([ਪਿਆਲਾ@indef];[ਨਵਾਂ])]** *[mod([piālā@indef];[navāṃ])] [mod([mug@indef];[new)],* and *%b* refers to node **[NA([mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])];[mod([ ਗੱਡੀ@indef];[ਸੋਹਣੀ])])]** *[NA([mod([gaḍḍī@indef];[mahiṅgī])];[mod([gaḍḍī@indef];[sōhṇī])])] [NA([mod([car@indef];[expensive])];[mod([car@indef];[beautiful])])].* This rule changes the name of relation from ‘NA’ to ‘and’ keeping same arguments as in original node, as required in the final UNL. |
| 11 | (NA(%a;%b ),AND):=and(%a;%b); | Here, *%a* refers to node **[mod([ਗੱਡੀ@indef];[ਮਹਿੰਗੀ])]** *[mod([gaḍḍī@indef];[mahiṅgī])] [mod([car@indef];[expensive])],* and *%b* refers to node **[mod([ਗੱਡੀ@indef];[ਸੋਹਣੀ])]** *[mod([gaḍḍī@indef];[sōhṇī])] [mod([car@indef];[beautiful])].* This rule changes the name of relation from ‘NA’ to ‘and’ keeping same arguments as in original node, as required in the final UNL.<br>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 (6). |

The UNL generated is given in (6).

```
{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}                                   ...(6)
```

Here, :06, :09, :0L, :0J, :07, :08, :0D, :0B, :05, :03, are all scopes internally generated by IAN.

## 5 Results and Discussions

Universal Networking Language is a natural-language-independent language which can be used for refining, describing, and semantic searching. Interactive Analyser (*i.e.* IAN) tool is an effective online tool developed by UNDL foundation used for UNL-ization of any Natural Language. With the help of 257 TRules and 623 Dictionary entries, the system is tested on Corpus500 (provided by UNDL Foundation) for Punjabi Language, and their F-Measure is calculated with the help of online tool developed by UNDL foundation available at UNL-arium [15] as shown in Table 3.

Table 3. Category wise F-Measure of Corpus500

| Category | Number of sentences processed | Number of sentences returned | Number of sentences correct | Precision | Recall | F-Measure |
|---|---|---|---|---|---|---|
| Numbers and Ordinals | 150 | 150 | 150 | 1.000 | 1.000 | 1.000 |
| Preposition | 40 | 38 | 36 | 0.947 | 0.900 | 0.923 |
| Conjunctions | 10 | 10 | 10 | 1.000 | 1.000 | 1.000 |
| Determiners | 60 | 59 | 58 | 0.983 | 0.966 | 0.884 |
| Verbs | 50 | 45 | 40 | 0.888 | 0.800 | 0.842 |
| Nouns and Adjectives | 155 | 149 | 135 | 0.906 | 0.870 | 0.888 |
| Time | 20 | 20 | 18 | 0.900 | 0.900 | 0.900 |
| Temporary words | 15 | 15 | 15 | 1.000 | 1.000 | 1.000 |
| **TOTAL** | **500** | **486** | **462** | **0.9506** | **0 .924** | **0.936** |

F-Measure is calculated by the following formulae [14]:

F-Measure =2*{(precision*recall ) / (precision+recall)} ...(7)

where, Precision is the number of correct results divided by the number of all returned results [14]. Recall is the number of correct results divided by the number of results that should have been returned [14]. A result is considered returned when the output is a graph made of only Universal Words [14]. 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 [14]. 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) [14]. The distribution of F-Measure for various part-of-speeches is depicted in Figure 2.

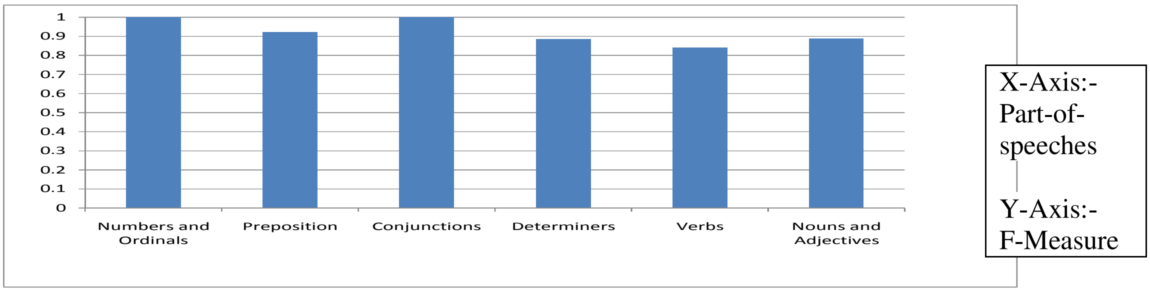

Figure 2. Distribution of F-Measure of Corpus500 for Punjabi Language of different part-of-speeches

## 6 Future Scope

UNL captures semantics of the natural language so semantic based searching system can be developed based on UNL-ization. Sentence level UNL-ization for Punjabi language is yet to be carried out. Work can be extended to carry out UNL-ization of Numbers and ordinals of more than fourteen digits. System needs to be improved so as to achieve F-Measure of 1.000.

## References

1. E. 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. Gel-

bukh , and E. Tovar, Ed(s). México, Research on Computing Science, 2005, pp. 157-166.

2. I.M. Boguslavsky, L. Iomdin, and V.G. Sizov, "Interactive EnConversion by means of the ETAP-3 system," in *Universal Network Language: Advances in Theory and Applications*, vol. 12, J. Cardeñosa , A. Gelbukh , and E. Tovar, Ed(s). México, Research on Computing Science, 2005, pp. 230-240.
3. K. Dey, and 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 , and E. Tovar, Ed(s). México, Research on Computing Science, 2005, pp. 215-229.
4. Lewis, M. Paul (Eds.) *Ethnologue: Languages of the World.*, 16th ed., SIL International, Dallas, 2009.
5. M. Choudhury, H. Ershadul, Y.A. Nawab, Z.H.S. Mohammad, and R.M. Ahsan, "Bridging Bangla to Universal Networking Language- a human language neutral meta-language," *Proc. 8th Int. Conf. on Computer and Information Technology*, Dhaka, Bangladesh, 2005, 104-109.
6. 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 , and E. Tovar, Ed(s). México, Research on Computing Science, 2005, pp. 125-137.
7. M.S. Gill, "Development of a Punjabi grammar checker," Ph.D. dissertation, Punjabi University, Patiala, 2008.
8. P. Kumar, and R.K. Sharma , "Punjabi Enconversion System" *Sadhana*,*Part 2*, April 2012, pp. 299–318.
9. R. Mohanty, A. Dutta , and P. Bhattacharyya,"Semantically relatable sets: building blocks for representing semantics," *Proc. 10th MT Summit*, Phuket, 2005, pp. 1-8
10. R. T. Martins, L.H.M. Rino, O.N. Osvaldo, R. Hasegawa, and M.G.V. Nunes, "Specification of the UNL-Portuguese enconverter-deconverter prototype," *Relatórios Técnicos do ICMC-USP*, 1997, pp.1-10.
11. R.T. Martins, R. Hasegawa, M. Graças, and V. Nunes, "Hermeto: A NL–UNL Enconverting Environment," in *Universal Network Language: Advances in Theory and Applications*, vol. 12, J. Cardeñosa , A. Gelbukh , and E. Tovar, Ed(s). México, Research on Computing Science, 2005, pp. 254-260.
12. UNDL Foundation. (2012, Jul 23). Dictionary Specifications [Online]. Available: http://www.unlweb.net/wiki/UNL_Dictionary_Specs
13. UNDL Foundation. (2012, Oct 23). Grammar Specifications [Online]. Available: http://www.unlweb.net/wiki/UNL_Grammar_Specs
14. UNDL Foundation. (2012, Sep 18). F- Measure [Online]. Available: http://www.unlweb.net/wiki/F-measure
15. UNDL Foundation. UNL-arium [Online]. Available: http://www.unlweb.net/unlarium/index.php?lang=pa
16. UNDL Foundation. (2012, Sep 21). UWs, Relations and Attributes Specs [Online]. Available: http://www.unlweb.net/wiki/Specs