

**GENERATION OF UNL ATTRIBUTES AND RESOLVING RELATIONS FOR PUNJABI ENCONVERTER**

Parteek Kumar\*, R. K. Sharma+

\*Assistant Professor, Department of Computer Science & Engineering, Thapar University, Patiala-147001, India,  
parteek.bhatia@thapar.edu+Professor, School of Mathematics & Computer Applications, Thapar University, Patiala-147001, India,  
rksharma@thapar.edu**ABSTRACT**

Automated Machine Translation (AMT) has a potential to digitally unite the whole world. Universal Networking Language (UNL) has been used by various researchers as an interlingual approach for AMT. The UNL system consists of two main components, namely, *EnConverter* (used for converting the text from a source language to UNL) and *DeConverter* (used for converting the text from UNL to a target language). This paper discusses the framework for designing *EnConverter* for Punjabi language with a special focus on generation of UNL attributes and relations from Punjabi source text. The architecture of Punjabi *EnConverter*, algorithm for understanding the Punjabi input sentence and resolution of UNL relations and attributes is also discussed in this paper. The paper highlights the *EnConversion* analysis rules for the *EnConverter* and indicates its usage in the generation of UNL expressions. This paper also covers the results of implementation of Punjabi *EnConverter* and compares these with the system available at Spanish Language Server.

**Keywords:** *EnConverter*; *DeConverter*; Machine Translation; Universal Networking Language

**1.0 INTRODUCTION**

Automated Machine Translation (AMT) is an important research area that can help to digitally unite the world. A good number of societies all over the world are lagging behind in this age of Information Technology. One of the reasons for this process is the language barrier in information distribution. There is a great need to translate web pages and electronic mail messages *etc.* into the native language for overcoming the language barrier. This paper focuses on one such technology and includes the work carried out in this direction for Punjabi Language. Punjabi is an Indo-Aryan language spoken by the Punjabi people in India, Pakistan, Canada and other parts of the world. It has approximately 57.1 million speakers in India and it is in the list of top ten speaking languages of India.

In order to design a multilingual machine translation system for Punjabi Language, interlingua approach is best suited as it requires only  $n$  interlingua transfer modules for  $n$  languages [1]. A transfer module of each language requires two components: one for conversion from source language to interlingua and other for interlingua to target language. As such, in this approach we require  $2n$  components, while direct or transfer based approach requires  $(n*(n-1))$  components [2]. We have used Universal Networking Language (UNL) as the interlingua for this task, because, the UNL representation has the right level of expressive power and granularity. UNL has 46 semantic relations and 87 attributes (which can be augmented with the user defined ones) to express the semantic content of a sentence [ ]. UNL has been developed and is managed by the Universal Networking Digital Language (UNDL) foundation, an international organization of Institute of Advanced Studies of United Nations University, Tokyo, Japan [3]. UNL System has *EnConverter* and *DeConverter* as two important components. The *EnConverter* converts source language sentences into UNL expressions. The *DeConverter* converts UNL expressions to target language sentences. With the development of *EnConverter* for Punjabi language, the Punjabi text is converted to UNL expressions. It has a potential to translate Punjabi language text to any language that has its own *DeConverter* for conversion of UNL expressions generated by Punjabi *EnConverter* to the target language. This will certainly help to develop a multilingual machine translation system for Punjabi Language.

In this paper, the framework for designing the *EnConverter* for Punjabi is discussed with special focus on generation of UNL attributes and relations from Punjabi source text. The paper is divided into nine sections. Section 2 describes the UNL system, its format for representation of the information and the design issues of the Punjabi *EnConverter*. The architecture of Punjabi *EnConverter* is given in Section 3. The algorithm for processing the input sentence is discussed in Section 4. Section 5 contains the working of Punjabi *EnConverter* and its analysis rules. The algorithm for the resolution of relations and generation of attributes is discussed in the Section 6. Section 7 of this paper elaborates the working of the *EnConverter* with a sample input sentence. The experimentation and testing of the system is given in Section 8. Section 9 concludes the work done for this paper and also contains the work to be done in future.

## 2.0 INTRODUCTION TO UNL

The *EnConverter* and *DeConverter* are the core software in the UNL system. The *EnConverter* converts source language sentences into UNL expressions. The *DeConverter* converts UNL expressions to target language sentences. UNL language servers can be connected to the Internet to carry out the conversions between natural languages and UNL expressions. Each Language Server will thus contain an *EnConverter* and a *DeConverter* of a language as shown in Fig. 1. The UNL system basically consists of UNL language servers, UNL Editors and UNL Viewers [4]. Anybody with the access to the Internet will be able to *EnConvert* source language text into UNL expressions. Also, a UNL expression can be *DeConverted* into a variety of native languages.

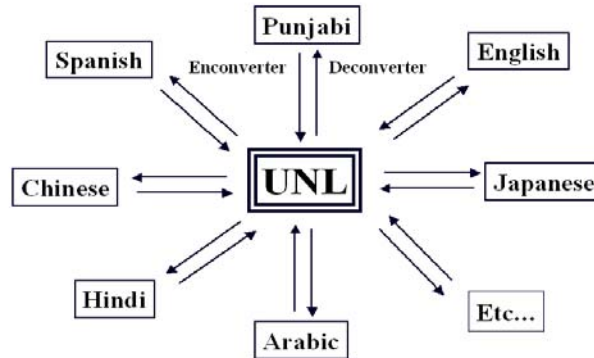


Fig. 1. UNL system

The concept of UNL as extended to the World Wide Web is illustrated in Fig. 2. UNL Editor for Punjabi language shall recognize the Punjabi contents and shall send a request to Punjabi language server to *EnConvert* these to UNL expressions. These expressions when embedded into a web page can be *DeConverted* by the target language server and one shall be able to view the web page in target language using UNL viewer [5].

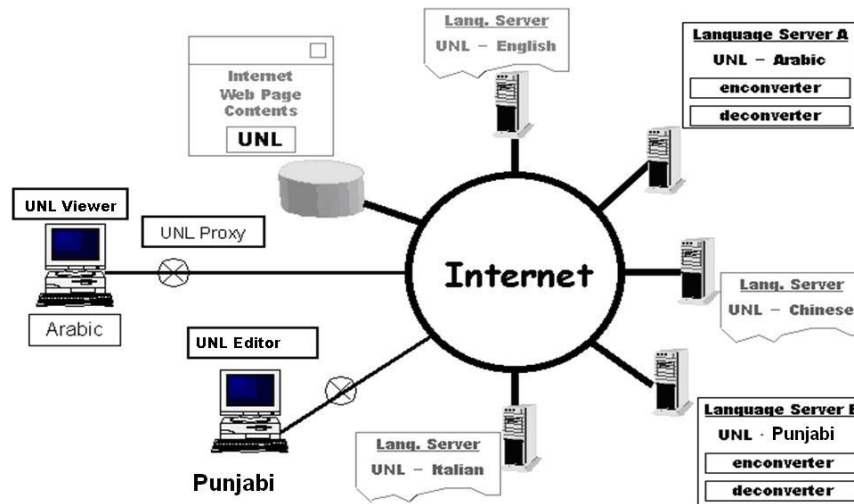


Fig. 2. UNL over web

### 2.1 UNL format for representation of information

UNL representation consists of UNL relations, UNL attributes and Universal Words (UWs). UWs are represented by their English equivalents. These words are listed in the Universal Word Lexicon of UNL knowledge base [6]. Relations are the building blocks of UNL sentences. The relations between the words are drawn from a set of predefined relations

[3, 7]. The attribute labels are attached with universal words to provide additional information like tense, number *etc.* For example, the English sentence “Girls eat mangoes” can be represented into UNL expression as:

```
{unl}
agt(eat(icl>event).@entry.@present, girl(icl>person).@pl)
obj(eat(icl>event).@entry.@present, mango(icl>food).@pl)
{/unl}
```

UNL expressions can also be represented in the form of UNL graphs. The UNL graph of the above mentioned sentence is given in Fig. 3.

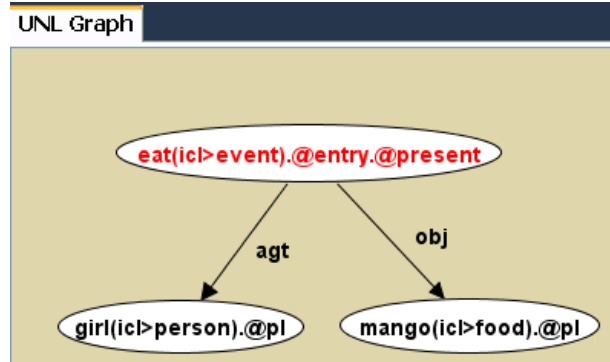


Fig. 3. UNL graph

We can here note that *agt* is the UNL relation which indicates “a thing which initiates an action”; *obj* is another UNL relation which indicates “a thing in focus which is directly affected by an event”; @entry and @present are UNL attributes which indicate the main verb and tense information; and @pl is UNL attribute which indicates the number information.

**2.2 Issues involved in conversion of Punjabi to UNL**

The important issues involved in this conversion are creation of Punjabi to Universal Word dictionary, generation of UNL attributes, and resolution of UNL relations. In the process of conversion, we first search the lexicon to have a map between Punjabi words and UWs, so that we have the root of word in order to establish the relevant relations and attributes. After finding the root of word, we have to generate the relevant attributes that would describe the subjective information of the sentence. These attributes are to be generated on the basis of the morphology of the source language [8].

The relations between UWs are resolved according to the different roles they play. A number of factors are to be considered in resolving a relation. The case markers of a language play a vital role in the resolution of relations among UWs [9].

**3.0 ARCHITECTURE OF PUNJABI ENCONVERTER**

The architecture of Punjabi-UNL *EnConverter* is given in Fig. 4. In the chunking phase which is the first phase of the module, the input sentence is divided into chunks. These chunks are created on the basis of Punjabi-UW dictionary. The chunks are then processed in Punjabi Morphological analyzer to get the root and grammatical information of each chunk in the form of tag set. The UWs for each chunk are retrieved from Punjabi-UW dictionary. Some chunks may return more than one tag set. In order to resolve the ambiguity among these chunks, Part-of-Speech (POS) tagging is performed [10, 11]. The UNL relation resolution and attribute generation rule base is used to fire the appropriate rule according to the dictionary attributes of each tag [12]. After applying the actions mentioned in the rule base, final UNL expression is generated for the input Punjabi sentence.

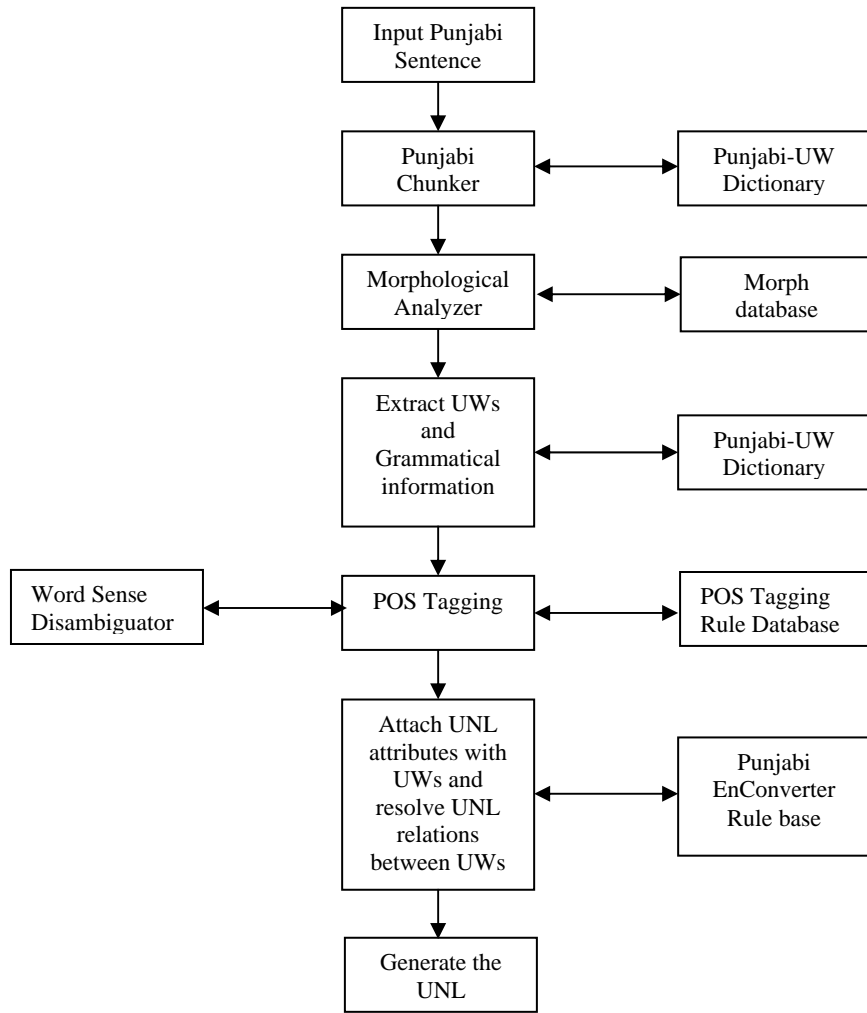


Fig. 4. Architecture of Punjabi-UNL *EnConverter*

#### 4.0 ALGORITHM FOR PROCESSING INPUT PUNJABI SENTENCE

Algorithm for the processing of input Punjabi sentence involves the following steps:

- i. Accept the input Punjabi sentence.
- ii. Chunk the input sentence with the help of Punjabi-UW dictionary.  
The UWs from Punjabi-UW dictionary that contain the largest set of consecutive words from input sentence are considered as chunks. If we are not able to find a UW corresponding to a word in the sentence, this word is considered as a chunk. The process of chunking is described below the help of one example.

Punjabi text: ਮੈਂ ਥੋੜੀ ਦੇਰ ਦੇ ਲਈ ਤੇਰੇ ਨਾਲ ਜਾਵਾਂਗਾ।

Transliterated text: *main thodi der de lai tere naal javanga.*

Equivalent English text: I will accompany you for a short while.

This Punjabi sentence will be chunked as:

(ਮੈਂ)	(ਥੋੜੀ ਦੇਰ ਦੇ ਲਈ)	(ਤੇਰੇ)	(ਨਾਲ)	(ਜਾਵਾਂਗਾ)।
( <i>main</i> )	( <i>thodi der de lai</i> )	( <i>tere</i> )	( <i>naal</i> )	( <i>javanga</i> ).
(I)	(for a short while)	(you)	(with)	(will go).

- iii. Process each chunk obtained from step (ii) using Punjabi Morphological Analyzer in order to get the root and suffix for each chunk.

For example, ਜਾਵਾਂਗਾ (*javanga*) has the root word as ਜਾ (*ja*) and suffix ਵਾਂਗਾ (*vanga*).

- iv. Replace the each chunk in the input sentence with the root and suffix obtained from step (iii).

For example, now the final chunks from the input sentence given in step (ii) will be:

(ਮੈਂ) (ਥੋੜੀ ਦੇਰ ਦੇ ਲਈ) (ਤੇਰੇ) (ਨਾਲ) (ਜਾ) (ਵਾਂਗਾ)।  
 (*main*) (*thodi der de lai*) (*tere*) (*naal*) (*ja*) (*vanga*).  
 (I) (for a short while) (you) (with) (go).

- v. Construct the string from the modified chunks.

The string for the given sentence will become,

ਮੈਂ ਥੋੜੀ ਦੇਰ ਦੇ ਲਈ ਤੇਰੇ ਨਾਲ ਜਾ ਵਾਂਗਾ।

(*main thodi der de lai tere naal ja vanga*).

- vi. Perform step (ii) in order to find new chunks in the string.

This step will give the following chunks for the string in step (v).

(ਮੈਂ) (ਥੋੜੀ ਦੇਰ ਦੇ ਲਈ) (ਤੇਰੇ) (ਨਾਲ ਜਾ) (ਵਾਂਗਾ)।  
 (*main*) (*thodi der de lai*) (*tere*) (*naal ja*) (*vanga*).  
 (I) (for a short while) (you) (accompany).

In this step, (ਨਾਲ) (ਜਾ) are grouped as a single chunk (ਨਾਲ ਜਾ) and this results into English equivalent “accompany” as desired chunk instead of “go”.

- vii. Search the case marker file and verb ending suffix file for the words that are not available in Punjabi-UW dictionary. The largest set of consecutive words that is found in case marker file or verb ending suffix file is considered as a chunk.

For example, ਵਾਂਗਾ (*vaanga*) is considered as chunk as this has an entry in verb ending suffix file. This, however, does not have an entry in Punjabi-UW dictionary.

- viii. Retrieve UWs and grammatical attributes corresponding to each chunk from Punjabi-UW dictionary.

- ix. Apply Part of Speech (POS) Tagging to a chunk to remove the ambiguity.

- x. Create a linked list for the input sentence.

This linked list shall have nodes as chunks of the input sentence with attributes like punjabi word, universal word, and others.

### 5.0 WORKING OF PUNJABI ENCONVERTER

The Punjabi *EnConverter* processes the given input sentence from left to right. It uses two types of windows [1,13], namely, analysis window and condition window in the processing. The currently focused analysis windows are circumscribed by condition windows as shown in Fig. 5.

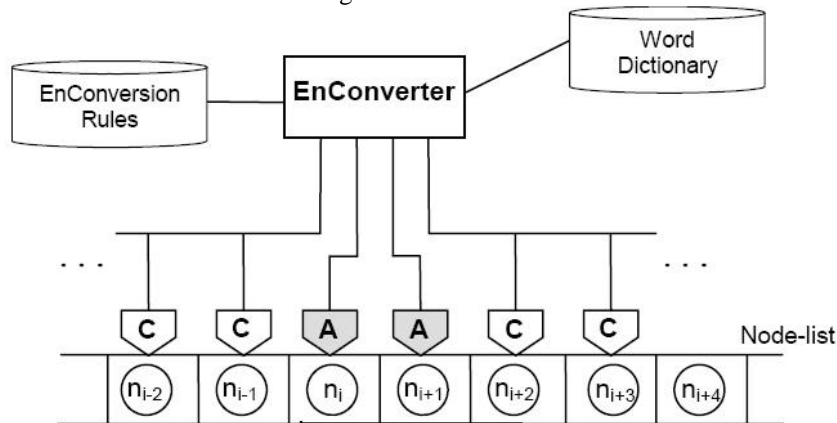


Fig. 5. Working of Punjabi *EnConverter*

Here, in Fig. 5, ‘A’ indicates an analysis window, ‘C’ indicates a condition window, and ‘n<sub>i</sub>’ indicates an analysis node.

### 5.1 *EnConversion* analysis rules

The *EnConversion* rule base is created on the basis of case markers and morphology of the Punjabi language [9, 13]. These rules are designed on the guidelines given in the UNDL *EnConverter* specifications [14].

The rule format used in designing of the system is as follows.

{COND1:ACTION1:REL1} {COND2:ACTION2:REL2}

Here,

- <COND1> indicates Condition1, it contains the lexicon attributes of left analysis window.
- <COND2> indicates Condition2, it contains the lexicon attributes of right analysis window.
- <ACTION1> and <ACTION2> are used to indicate the actions performed if the corresponding condition is true.
- The <REL1> and <REL2> fields indicate the possible relation between two analysis windows.

## 6.0 ALGORITHM FOR RELATION RESOLUTION AND GENERATION OF ATTRIBUTES

According to the algorithm discussed in Section 4, a node-net is created from the input Punjabi sentence. Then, Punjabi *EnConverter* System invokes the following algorithm for UNL relation resolution and generation of attributes.

- i) Process each node of linked list by considering the first node as left analysis window and next node as right analysis window.
  - ii) Search the required rule from *EnConverter* analysis rules. This depends upon the dictionary attributes of left and right analysis windows.
  - iii) Modify the linked list to resolve the UNL relations and generate UNL attributes according to the fired rule. If no rule is fired, then go to step (v).
  - iv) Consider first node of modified linked list as left analysis window and next node as right analysis window. Go to step (ii) with new analysis windows.
- If the modified linked list contain only single node, then consider that node as “entry node” and stop further processing. It means that all the nodes are successfully processed by the system.
- v) If no rule is fired in step (ii), then shift the window to right. This effectively means that right analysis node will become the left analysis window and next node will become the right analysis window. Go to step (ii) with new analysis windows.

## 7.0 ENCONVERSION OF A PUNJABI SENTENCE TO UNL REPRESENTATION: AN EXAMPLE

Let us understand the working of the Punjabi *EnConverter* with the help of one example Punjabi sentence.

Punjabi sentence: ਰਾਮ ਚੱਲਦਾ ਅਤੇ ਖਾਂਦਾ ਹੈ।

Transliterated sentence: *ram chalda ate khanda hai.*

Equivalent English sentence: Ram walks and eats.

The input sentence is processed according to the algorithm discussed in Section 4. The chunks obtained from the input sentence are given below.

(ਰਾਮ) (ਚੱਲ) (ਦਾ) (ਅਤੇ) (ਖਾ) (ਦਾ ਹੈ)।

(*ram*) (*chal*) (*da*) (*ate*) (*kha*) (*da hai*).

There are six nodes in the input sentence, with the following details.

Node1 ਰਾਮ: Universal Word: "ram(icl>person)", Dictionary Attributes: (*N*, *ANIMT*)

Node 2 ਚੱਲ: Universal Word : "walk(icl>event)", Dictionary Attribute: (*V*)

Node 3 ਦਾ: Universal Word : null, Dictionary Attribute: (*CASE*)

Node 4 ਅਤੇ: Universal Word : null, Dictionary Attribute: (*CONJ*)

Node 5 ਖਾ: Universal Word : "eat(obj>food)", Dictionary Attribute: (*V*)

Node 6 ਦਾ ਹੈ: Universal Word : null, Dictionary Attribute: (*NULL*).

Here, *N* indicates noun, *ANIMT* indicates animate, *V* represents verb, *CASE* is used to represent case markers of Punjabi, *CONJ* represents the words used for conjunction and *NULL* represents no dictionary attribute.

These nodes are processed by the algorithm given in Section 6, for UNL relation resolution and generation of attributes as explained in the following steps.

i) Initially, the system has following left and right analysis windows for the given input sentence.

Left Node is ਰਾਮ: Universal Word: "ram(icl>person)", Dictionary Attributes: (*N*, *ANIMT*).

Right Node is ਚੱਲ: Universal Word: "walk(icl>event)", Dictionary Attribute: (*V*).

Both left and right analysis windows satisfy the following *EnConversion* rule.

>{N,ANIMT:null:agt}{V:+**AGTRES**:null}

Here, **AGTRES** means that right analysis node is used in resolving *agt* relation with the left analysis node.

The above rule is preceded by the ">" sign, this means that it as a right modification rule. This rule is applicable, when left node modifies the right node. It deletes the left node from the node-list, while the right node remains in the node-list.

It results in the resolution of *agt* relation between two nodes as shown below.

agt(walk(icl>event), ram(icl>person))

The attributes of right analysis window are modified owing to the presence of "+" sign in ACTION2 of the fired rule (indicated bold in the rule). Its presence in action part of rule results into addition of attribute(s) to the concerned node, while the presence of "-" sign results into deletion of attribute(s) from the concerned node.

Thus, the modified attributes of node ਚੱਲ; "walk(icl>event)"; become *V*, **AGTRES** and the linked list after left node deletion becomes,

[ਚੱਲ] [ਦਾ] [ਅਤੇ] [ਖਾ] [ਦਾ ਹੈ].  
(*chal*) (*da*) (*ate*) (*kha*) (*da hai*).

ii) Now, ਚੱਲ (*chal*) becomes the left analysis window with dictionary attributes *V*, **AGTRES** and ਦਾ (*da*), becomes the right analysis window with dictionary attributes *CASE*.

The rule fired between left and right analysis windows is given below.

+{V:+.@custom.@sg.@male:null}{[ਦਾ]:null:null}

This rule is preceded by the "+" sign, which indicates that it as a left composition rule. It results into concatenation of right node to the left node as a single composition node and the attributes of left node are inherited for further processing. The presence of "+" sign in the action part of left analysis window results into the addition of attributes. Since, these attributes are preceded by "@" sign, they are concatenated to corresponding UW as UNL attributes. The UW "walk(icl>event)" is modified as "walk(icl>event).@custom. @sg.@male". The linked list after application of left composition rule becomes,

[ਚੱਲ ਦਾ] [ਅਤੇ] [ਖਾ] [ਦਾ ਹੈ].  
(*chal da*) (*ate*) (*kha*) (*da hai*).

iii) Now, ਚੱਲ ਦਾ (*chal da*) becomes the left analysis window with dictionary attributes *V*, **AGTRES** and ਅਤੇ (*ate*) becomes the right analysis window with dictionary attribute *CONJ*.

The rule fired between left and right analysis windows is given below.

+{V:+CONJ:null}{CONJ:null:null}

It is again a left composition rule which results into concatenation of right node to the left node and addition of *CONJ* as dictionary attribute for left analysis window ਚੱਲ ਦਾ (*chal da*); "walk(icl>event)". The linked list after application of left composition rule becomes,

[ਚੱਲ ਦਾ ਅਤੇ] [ਖਾ] [ਦਾ ਹੈ].  
(*chal da ate*) (*kha*) (*da hai*).

iv) Here, ਚੱਲ ਦਾ ਅਤੇ (*chal da ate*) becomes the left analysis window with dictionary attributes *V*, **AGTRES**, *CONJ* and ਖਾ (*kha*) becomes the right analysis window with dictionary attribute *V*.

The rule fired between left and right analysis windows is given below.

>{V,CONJ:null:null}{V:+ANDRES:and}

It is a right modification rule and it results into deletion of left node from the node-list. Here, *and* relation is resolved between two analysis windows (due to presence of *and* in the relation part of right analysis window) as shown below.  
 and (eat(obj>food), walk(icl>event))

The modified attributes of right analysis window ਖਾ (*kha*) are *V, ANDRES*. The linked list after application of right modification rule becomes,

[ਖਾ] [ਦਾ ਹੈ]  
 (*kha*) (*da hai*).

v) Here, ਖਾ (*kha*) becomes the left analysis window with dictionary attributes *V, ANDRES* and ਦਾ ਹੈ (*da hai*) becomes the right analysis window with dictionary attribute *NULL*.

The rule fired between left and right analysis nodes is given below.

+{V:+.@custom.@present.@sg.@male:null}{[ਦਾ ਹੈ]:null:null}

It is a left composition rule which results into concatenation of right node to the left node. The presence of “+” sign and “@” sign in the action part of left analysis window results into the concatenation of “.@custom.@present.@sg.@male” as UNL attributes with the UW "eat(obj>food)".

Finally, the modified linked list contains only one node as given below.

[ਖਾ ਦਾ ਹੈ]  
 (*kha da hai*)

Being the last node, this is considered as root node and “.@entry” attribute is concatenated to its UW as given below, eat(obj>food).@custom.@present.@sg.@male.@entry.

vi) The details of UNL expression generated by the Punjabi *EnConverter* system for input Punjabi Sentence is as follows.

Punjabi sentence: ਰਾਮ ਚੱਲਦਾ ਅਤੇ ਖਾਂਦਾ ਹੈ।

Transliterated sentence: *ram chalda ate khanda hai*.

Equivalent English sentence: Ram walks and eats.

UNL generated by the system:

```
{unl}
agt(walk(icl>event) .@custom.@sg.@male,ram(icl>person))
and(eat(obj>food).@custom.@present.@sg.@male.@entry,walk(icl>event).@custom.@sg.@male)
{/unl}
```

## 8.0 EXPERIMENTATION AND TESTING OF THE SYSTEM

We have tested our system on several Punjabi sentences. It has been seen that the system successfully handles the resolution of UNL relations and generation of attributes for these sentences. The system has been tested with the help of English sentences available at Spanish UNL language server with URL

[http://opera.dia.fi.upm.es/cle/demo\\_interactiva/interactive\\_main\\_page.htm](http://opera.dia.fi.upm.es/cle/demo_interactiva/interactive_main_page.htm).

We have manually translated the given English sentences at Spanish language server into equivalent Punjabi sentences and then inputted those equivalent Punjabi sentences to the designed Punjabi-UNL *EnConverter* system. We have compared the UNL expressions generated by our system with the UNL expressions generated by Spanish UNL language server. This comparative analysis is given in Table 1 for ten sentences.



Table 1. Comparative analysis of UNL expressions generated by Punjabi *EnConverter*

S. No.	Input Punjabi Sentence	Equivalent English Sentence	Relations Resolved	Rules Fired	UNL expressions generated by the Punjabi <i>EnConverter</i>	UNL expressions generated by the Spanish UNL language server
1.	ਧਮਾਕੇ ਨੇ ਖਿੜਕੀਆਂ ਤੇੜੀਆਂ।  <i>tamake ne kidkeaan todiyaan.</i>	The explosion broke the windows.	obj, agt	>{N,INANI, ^WICH,^PLC, ^SRCRES, ^TOON:null :obj}{V: +OBJRES:null} >{N, CASE, ^NOO, ^CAG: null:agt}{V: +AGTRES: null} +{N:+. @pl. @fe male:null}{[ਆਂ]: null:null} +{V:+. @past. @ pl. @female:null} {[ੀਆਂ]:null :null}	{unl} obj(break(icl>do) . @past. @pl. @female. @e ntry, window(icl>frame). @pl. @female) agt(break(icl>do) . @past. @pl. @female. @e ntry,explosion) {/unl}	agt(break(icl>do). @e ntry. @past, explosion. @def) obj(break(icl>do). @e ntry. @past, window(icl>thing). @ def. @pl)
2.	ਮੇਰੀ ਜਾਨ ਦੇ ਨਾਲ ਚੱਲਦੀ ਹੈ।  <i>mary john de naal chaldi hai.</i>	Mary walks with John.	cag, agt	+{N,ANIMT:+C AG:null} {cag/ins:null :null} >{N,ANIMT, CAG:null:cag} {V:+CAGRES: null} +{V:+. @present. @sg. @female :null}{[ਦੀ ਹੈ]: null:null}	{unl} cag(walk(icl>do) . @present. @sg . @female. @entry, John(icl<person)) agt(walk(icl>do) . @present. @sg. @female @entry, Mary(icl<person)) {/unl}	agt(walk(icl>do) . @entry. @present, Mary) cag(walk(icl>do). @e ntry. @present, John)
3.	ਕਾਗਜ਼ ਲਾਲ ਤੋਂ ਨੀਲੇ ਵਿੱਚ ਬਦਲ ਗਿਆ।  <i>kagaj laal toon neele wich badal gaya.</i>	The paper changed from red to blue.	gol, src, obj	+{ADJ:+TOON: null}{[ਤੋਂ]:null :null} >{ADJ,TOON:n ull:src}{V: +SRCRES:null} +{ADJ:+WICH: null}{[ਵਿੱਚ]:null :null} >{ADJ,WICH, ^PLC:null:gol} {V:+GOLRES :null} >{ADJ,SRCRES ,^PLC:null:gol} {V:+GOLRES :null}	{unl} gol(change(icl>do). @past . @male. @sg. @entry, blue(icl>thing)) src(change(icl>do) . @past. @male. @sg. @ent ry, red(icl>thing)) obj(change(icl>do) . @past. @male. @sg . @entry, paper(icl>thing)) {/unl}	obj(change(icl>do). @entry. @past, paper(icl>thing) . @topic. @def) src(change(icl>do) . @entry. @past, red(icl>thing)) gol(change(icl>do). @entry. @past, blue(icl>thing))

				+{V:+. @past . @male. @sg :null} {[ਗਿਆ] : :null:null}		
4.	ਉਸਨੇ ਚਾਕੂ ਨਾਲ ਡੱਬਾ ਖੋਲਿਆ। <i>usne chaku naal diba kolya.</i>	He opened the box with a knife.	obj, ins, agt	+{N,INS: +NAAL:null} {[ਨਾਲ]:null:null} >{N,INS,NAAL: null:ins}{V: +INSRES:null}	{unl} obj(open(icl>do). @past . @entry, box(icl>thing)) ins(open(icl>do). . @past. @entry, knife(icl>thing)) agt(open(icl>do) . @entry. @past, he(icl<person)) {/unl}	agt(open(icl>do) . @entry. @past, he) obj(open(icl>do). @e ntry. @past, box(icl>thing) . @def) ins(open(icl>do) . @entry. @past, knife(icl>thing) . @indef)
5.	ਮੈਰੀ ਨੇ ਟੋਕੀਓ ਟਾਵਰ ਦਾ ਦੌਰਾ ਕੀਤਾ। <i>Mary ne tokyo tower da doura kita.</i>	Mary visited the Tokyo tower.	nam, agt	>{N,PROPER :null:nam}{N: +NAMRES :null}	{unl} nam( tower(icl>thing), Tokyo (icl>place)) obj(visit(icl>do) . @past. @entry, tower(icl>thing)) agt(visit(icl>do) . @past. @entry, Mary(icl>person)) {/unl}	agt(visit(icl>do) . @entry. @past, Mary) obj(visit(icl>do) . @entry. @past, tower(icl>thing) . @def) nam( tower(icl>thing) . @def, Tokyo)
6.	ਜਾਨ ਨੇ ਨੌਂ ਘੰਟੇ ਕੰਮ ਕੀਤਾ। <i>John ne noon ghante kam kita.</i>	John worked for 9 hours	qua, dur, agt	>{NUM:null :qua}{N: +QUARES:null} >{N,TIME:null: dur}{V: +DURRES:null}	{unl} qua(hour. @pl,9) dur(work(icl>do) . @past. @entry, hour. @pl) agt(work(icl>do) . @past. @entry, john(icl>person)) {/unl}	agt(work(icl>do). @e ntry. @past, John) dur(work(icl>do) . @entry. @past, hour. @pl) qua(hour. @pl, 9)
7.	ਮੈਂ ਗੇਰੇਜ ਵਿੱਚ ਕਾਰ ਧੋਂਦਾ ਹਾਂ। <i>main garage wich car tonda haan.</i>	I wash the car in the garage	plc, obj, agt	+{N,PLC: +WICH:null}{ [ਵਿੱਚ]:null:null} >{N,PLC,WICH :null:plc}{V:null :null}	{unl} plc(wash(icl>do) . @present. @entry, car(icl>thing)) Obj(wash(icl>do) . @present. @entry, car(icl>thing)) Agt(wash(icl>do) . @present. @entry, i(icl>person)) {/unl}	agt(wash(icl>do) . @entry. @present, i) obj(wash(icl>do) . @entry. @present, car(icl>thing) . @def) plc(wash(icl>do) . @entry. @present, garage(icl>thing) . @def)
8.	ਮੈਂ ਜਾਪਾਨੀ ਆਦਮੀ ਨੂੰ ਮਿਲਿਆ। <i>main japoni aadmi noo</i>	I met a man from Japan.	agt, obj, frm	+{N,PLC:+FRM :null} {[ੀ]:null :null} >{N,PLC,FRM: null:frm}{N: +FRMRES:null} +{N:+NOO:null}	{unl} frm(man(icl>person), Japan(icl>place)) obj(meet(icl>do) . @past. @entry, man(icl>person)) agt(meet(icl>do)	agt(meet(icl>do) . @entry. @past, i) obj(meet(icl>do) . @entry. @past, man. @indef) frm(man. @indef, Japan(icl>thing))

	<i>milaya.</i>			{[ਨੂੰ]:null :null} >{N,NOO:null: obj}{V: +OBJRES:null}	.@past.@entry, i(icl>person)) {/unl}	
9.	ਜਾਨ ਨਿਊਯਾਰਕ ਵਲੋਂ ਸ਼ਿਕਾਗੋ ਤਕ ਜਾਂਦਾ ਹੈ। <i>john new york valon chicago tak janda hai.</i>	John goes to Chicago via New York.	plt, via, agt	+{N,PLC:+VIA: null}{[ਵਲੋਂ]:null: null} +{N,PLC:+TAK :null}{[ਤਕ]:null: null} >{N,PLC,TAK:n ull:plt}{V: +PLTRES:null} >{N,PLC,VIA :null:via}{V: +VIARES:null}	{unl} plt(go(icl>do) .@present.@sg .@male.@entry, Chicago(icl>place)) via(go(icl>do) .@present.@sg .@male, New York(icl>place) ) agt(go(icl>do) .@present.@sg .@male, John(icl<person)) {/unl}	agt(go(icl>do) .@entry.@present, John) plt(go(icl>do) .@entry.@present, Chicago) via(go(icl>do) .@entry.@present, New York)
10.	ਜਹਾਜ਼ ਗੱਡੀਆਂ ਨਾਲੋਂ ਜਿਆਦਾ ਤੇਜ਼ ਹਨ। <i>jahaaz gadian naalon jayada tej han.</i>	Aeroplanes are faster than trains.	man, bas, aoj	>{ADJ:null :man}{ADJ: +MANRES :null} +{N:+BAS:null} {[ਨਾਲੋਂ]:null:null} >{N,BAS:null :bas}{ADJ: +ADJRES:null} >{N:null:aoj} {ADJ,ST :+AOJRES:null}	{unl} man( fast(mod<thing) .@present.@entry, more) bas(more, train(icl>thing).@pl) aoj( fast(mod<thing) .@present.@entry, aeroplane(icl>thing) .@pl) {/unl}	aoj( fast(mod<thing) .@entry.@present, aeroplane(icl>thing) .@generic.@pl) man( fast(mod<thing) .@entry.@present, more) bas(more, train(icl>thing) .@generic.@pl)

We have also tested the UNL expressions generated by our system with UNL Parser and Viewer available at Spanish Language Server with URL <http://opera.dia.fi.upm.es/vvunl/vvunl.html>.

The UNL graph generated by the system for the following input Punjabi sentence is given in Fig. 6.

Punjabi sentence: ਰਾਮ ਮੈਦਾਨ ਵਿੱਚ ਮਾਨ ਦੇ ਨਾਲ ਫੁੱਟਬਾਲ ਖੇਡ ਰਿਹਾ ਹੈ।

Transliterated sentence: *ram maidan wich maan de naal football khel riha hai.*

Equivalent English sentence: Ram is playing football in the field with Maan.

UNL expression generated by the Punjabi *EnConverter* is given below.

```
{unl}
obj(play(agt>human,con>player,ppl>game).@progress.@present.@sg.@male.@entry, football(icl>game))
cag(play(agt>human,con>player,ppl>game).@progress.@present.@sg.@male.@entry, maan)
plc(play(agt>human,con>player,ppl>game).@progress.@present.@sg.@male.@entry, field(icl>stadium))
agt(play(agt>human,con>player,ppl>game).@progress.@present.@sg.@male.@entry, ram)
{/unl}
```

All ten sentences given in Table 1 have also successfully been parsed and viewed with the help of Spanish Language Server.

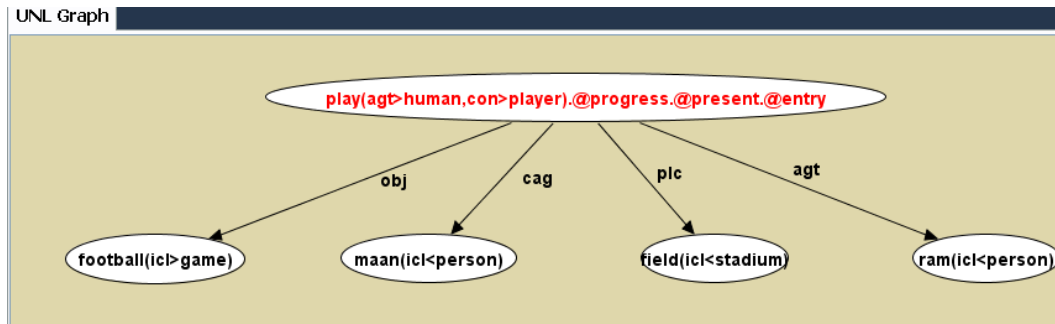


Fig. 6. UNL graph for a UNL expression generated by Punjabi *EnConverter*

## 9.0 CONCLUSION AND FUTURE SCOPE

In this paper, the Punjabi *EnConverter* has been discussed, developed, implemented and tested. Punjabi *EnConverter* uses the *EnConversion* analysis rules for UNL relation resolution and generation of attributes from the input Punjabi sentence. We have here developed approximately one thousand *EnConversion* analysis rules for the development of Punjabi *EnConverter*. This *EnConverter* has been tested for its performance vis-à-vis an *EnConverter* available in public domain on the Spanish Language Server. The results of testing are encouraging and the outputs of our system and the Spanish Language Server are in a very good agreement.

Presently, the Punjabi *EnConverter* handles the simple sentences. We are working on extending the scope of *EnConverter* to include clausal, interrogative and long sentences. We are also working on implementing the word sense disambiguation module in the proposed Punjabi *EnConverter*.

## REFERENCES

- [1] Hiroshi Uchida & Meiyong Zhu, Interlingua for Multilingual Machine Translation, *Proc. Machine Translation Summit IV, Japan*, July 20-22, 1993.
- [2] Kuntal Dey & P. Bhattacharya, UNL based analysis and generation of Bengali case structure constructs, *Proc. International conference on the Convergence of Knowledge, Culture, Language and Information Technologies. Library of Alexandria, Egypt*, December 2003.
- [3] Hiroshi Uchida & Meiyong Zhu, The Universal Networking Language beyond machine translation, *Proc. Third International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, Spain*, 2002.
- [4] Hong Munpyo & Streiter Oliver, Overcoming the language barriers in the Web: The UNL-Approach, *Proc. GLDV, Deutschland*, 1999.
- [5] Universal Networking Language (UNL) Specifications, Version 2005. <http://www.unl.org/unlsys/unl/unl2005/>
- [6] Hiroshi Uchida, Meiyong Zhu & Senta Della, A gift for a Millennium. *The United Nations University, Japan*, 1999. <http://www.unl.org/publications/gm/index.htm>
- [7] J. S. Parikh & P. Bhattacharya, Interlingua based English Hindi Machine Translation and language divergence, *Journal of Machine Translation(JMT) (17)*, 2002.
- [8] Yousuf Ali Nawab, Krishna Das Jugal, S. M. Abdullah Al-Mamun & Mohammad Nurannabi Abu, Morphological Analysis of Bangla words for Universal Networking Language, *Proc. Third International Conference on Digital Information Management (ICDIM 08), London, England*, pp 532-537.
- [9] T. Dhanabalan, K. Saravanan & T.V. Geetha, Tamil to UNL EnConverter, *Proc. International Conference on Universal Knowledge and Language, Goa, India*, 2002.

- [10] Hiroshi Uchida, ATLAS: Fujitsu Machine Translation System, *Proc. Machine Translation Summit, Japan*, September 17-19, 1987.
- [11] Manoj Jain & O. P. Damani, English to UNL (Interlingua) Enconversion, *Proc. Second Conference on Language and Technology, (CLT), 2009*.
- [12] P. Bhattacharya, Knowledge extraction from Hindi text, *Journal of Institution of Electronic and Telecommunication Engineers*, July, 2001.
- [13] P. Bhattacharya, Multilingual information processing through Universal Networking Language, *Proc. Indo UK Workshop on Language Engineering for South Asian Languages (LESAL), Mumbai, India*, April, 2001.
- [14] Universal Networking Language (UNL) Center: EnConverter Specifications. UNDL Foundation, 2000.