NLization of Nouns, Pronouns and Prepositions in Punjabi With EUGENE

[1]Ashutosh Verma and [2]Parteek Kumar

[1]M.E (Computer Science), Thapar University Patiala, Punjab

ashutosh1556@gmail.com

[2]Computer Science & Engineering Department, Thapar University Patiala, Punjab

parteek.bhatia@thapar.edu

Abstract

Universal Networking Language (UNL) has been used by various researchers as an Interlingua approach for AMT (Automatic machine translation). The UNL system consists of two main components/tools, namely, EnConverter-IAN (used for converting the text from a source language to UNL) and DeConverter - EUGENE (used for converting the text from UNL to a target language). This paper highlights the DeConversion generation rules used for the DeConverter and indicates its usage in the generation of Punjabi sentences. This paper also covers the results of implementation of UNL input by using DeConverter-EUGENE and its evaluation on UNL sentences such as Nouns, Pronouns and Prepositions.

Keyword

UNL, AMT, IAN, EUGENE, UNLization, NLization.

1. Introduction

Universal Networking Language (UNL) is a declarative formal language specifically designed to represent semantic data extracted from natural language texts. It can be used as a pivot language in interlingual machine translation systems or as a knowledge representation language in information retrieval applications [1]. Applications of UNL have been found in many domains such as machine translation, Information retrieval, multilingual document generation, *etc*. There are basically two approaches UNL follows *i.e.* UNLization and NLization. UNLization, formerly known as EnConversion, is the process of representing the content of a natural language structure into UNL and NLization, formerly known as DeConversion, is the process of generating natural language structures corresponding to UNL graphs. The process of NLization may have different generation units, *i.e.* Sentence-driven NLization and Text-driven NLization and Word-driven NLization. In Sentence-driven UNLization, the source document is represented as a list of non-semantically related networks of individual concepts. In Text-driven UNLization, the source document is represented as a network of semantically related networks of individual concepts and in Word-driven UNLization, the sentence boundaries and the structure of the source document are ignored, and the source document is represented as a single graph, *i.e.*, as a simple network of individual concepts [2].

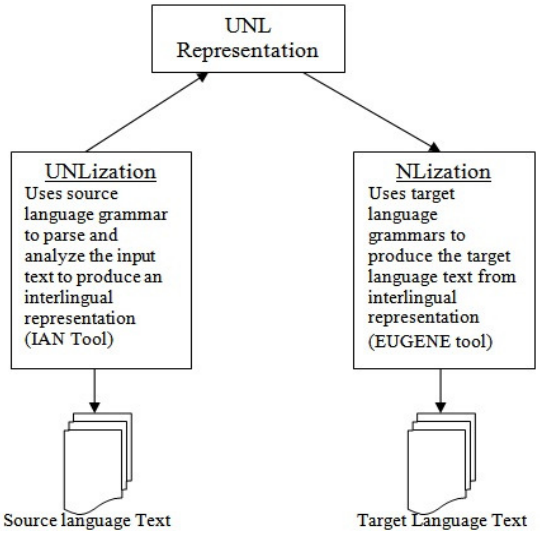

Figure 1: Classification of UNL

In Figure 1 source text language is converted to UNL using IAN tool, also known as UNLization and then UNL expression is converted to Target language text using EUGENE tool, also known as NLization.

This paper is divided into Seven sections, First section is Introduction, second is Related work, Third is EUGENE framework, Fourth is Features of Punjabi language, fifth is Implementation for NLization of Punjabi language, six is Results and Discussion, seventh is Conclusion and Future scope.

2. Related Work

There are three different approaches that have been used to design and develop the tool for DeConversion processes. In the first approach, one uses a common engine like 'DeCo' tool provided by the UNDL center to accomplish the task [4]. The other approach is integrative approach that is based on the integration of UNL into pre-existing MT systems. Third approach is followed by researchers who have noticed the drawbacks in the tools provided by the UNDL center, and have created new architectures from scratch.

Dhanabalan and Geetha (2003) 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 [5].

Blanc (2005) has performed the integration of 'Ariane-G5' to the proposed French DeConverter. 'Ariane-G5' is a generator of MT systems. Its DeConversion process also takes place in the 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 [6].

Boguslavsky *et al.* (2005) have proposed a multi-functional linguistic processor, 'ETAP-3', as an extension of 'ETAP' machine translation system to a UNL based machine translation system. The proposed system is used to build a bridge between UNL and the internal representations of 'ETAP', namely Normalized Syntactic Structure (NormSS). The system has performed the resolution of ambiguity with the linguistic knowledge base of 'ETAP-3'. They have also proposed an interactive system that helps to resolve difficult cases of linguistic ambiguity by means of a dialogue with the human [7].

Shi and Chen (2005) have proposed UNL DeConverter for Chinese language. They have highlighted the problems of 'DeCo' tool provided by the UNDL center which includes difficulty in writing the rules, its slow speed and non-availability of the source code. These issues motivated the developers to propose a new DeConverter for Chinese [8].

Pelizzoni and Nunes (2005) have introduced 'Manati' DeConversion model as a UNL mediated Portuguese-Brazilian sign language human-aided machine translation system. The system proposed by them is based on constraint programming to reduce search; while object-oriented and higher-order programming provides a basis for defining friendly primitives [9].

Daoud (2005) has proposed an Arabic DeConversion system which involves mapping of relations, lexical transfer, word ordering, and morphological generations. In the mapping of relations phase, each UNL relation has been mapped to the corresponding Arabic grammar structure which is implemented by DeConversion rules. Its word ordering phase is governed by DeConversion rules which are used during the insertion of a new node from the graph to the node-list. Arabic morphological generations is achieved by implementing a modular approach for coding the generation rules [10].

Keshari and Bista (2005) 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 [11].

Singh *et al.* (2007) have proposed a DeConverter for Hindi Language known as 'HinD', indicating the non-availability of the source code of 'DeCo' tool and its complex rule format. Their system consists of four main stages including; lexeme selection, morphological generation of lexical words, function word insertion, and syntax planning. All these components use language-independent algorithms operating on language dependent data [12].

Kumar (2012) has proposed a UNL to Punjabi DeConverter. In this DeConverter generation process is based on the predicate-centric nature of the UNL. The DeConverter transforms an input UNL expression into the directed hyper graph structure known as node-net. The root node of node-net is called as entry node and it acts as the main predicate. The traversing of node-net starts from entry node and system traverse the entire UNL graph to produce equivalent sentence in the target language [3].

UNDL foundation proposed a Punjabi DeConverter EUEGNE. EUGENE (dEep-to-sUrface GENErator) is a natural language generation system. It generates natural language sentences out of semantic networks represented in the UNL format. In its current release, it is a web application developed in Java and available at the UNLdev. It takes a UNL input and delivers an output in natural language without any human intervention. It is language-independent and has to be parameterized to the natural language input through a dictionary and a grammar, provided as separate interpretable files [13].

3. EUGENE Framework

A framework is provided by UNDL foundation, known as EUGENE. It comes under UNLdev. The UNLdev is an integrated environment for the development of UNL-driven grammars [14]. It currently contains the following systems i.e. NLizer (or deconverters) are tools for producing natural language texts out of UNL documents. EUGENE is one of the NLizer. EUGENE performs the three following movements over the input file:

- **Segmentation**, *i.e.*, the division of the input document into a series of isolated graphs, which are processed one at a time.
- **Tokenization**, *i.e.*, the identification of the tokens (UWs, relations and attributes) of each graph of the input document.

- **Transformation**, *i.e.*, the application of the transformation rules of the grammar over each tokenized graph in order to generate a natural language sentence.

EUGENE performs the functioning of DeConverter. The reverse process is carried out during natural language generation. A systematic approach is followed during this process. Firstly Network Processing (NN) is performed, in this UNL graph is preprocessed by the NN rules in order to become a more easily tractable semantic network. Second Network-to-tree (NT) processing performed; in this resulting network structure is converted, by the NT rules, into a syntactic structure, which is still distant from the surface structure, as it is directly derived from the semantic arrangement. Third Tree-to-Tree (TT) processing performed, in this deep syntactic structure is subsequently transformed into a surface syntactic structure by the TT rules. Fourth Tree-to-List (TL) processing is performed, in this surface syntactic structure undergoes many other changes according to the TL rules, which generate a NL-like list structure. Fifth List Processing (LL) processing is performed; in this list structure is finally realized as a natural language sentence by the LL rules. All these processing modules displayed in Figure 2.

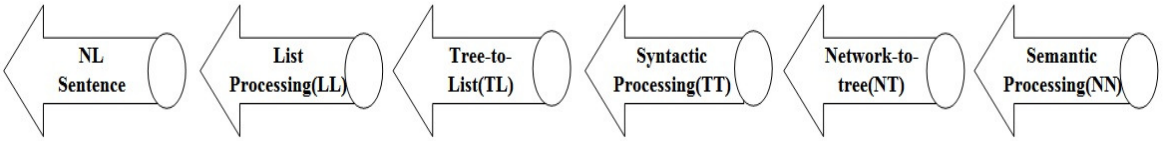

Figure 2: Modules of EUGENE

4. Features of Punjabi Language

Punjabi has word classes in the form of noun, pronoun, adjective, cardinal, ordinal, main verb, auxiliary verb, adverb, postposition, conjunction, interjection and particle. In this paper NLization of Nouns, Pronouns and Prepositions have been presented. Punjabi nouns change the form for number (singular or plural) and case in the sentences. Punjabi nouns have assigned gender (masculine or feminine). For example, ਕੰਧ kandh ‘fence’, ਕੁਰਸੀ kurasī ‘chair’, ਸੜਕ saṛak ‘road’ *etc* are used in feminine gender, and ਮੇਜ਼ mēja◌ ‘table’, ਟਰੱਕ ṭarakk ‘truck’, ਦਿਨ din ‘day’ *etc*. are used in masculine gender [3].

Punjabi has six types of pronouns. These are, personal pronouns, ਮੈਂ maiṃ ‘i’, ਤੂੰ tūṃ ‘you’; reflexive pronouns, e.g., ਆਪ āp (some what equivalent to honorific form of English sec-ond person ‘you’); demonstrative pronouns, e.g., ਉਹ uh ‘that’ and ਇਹ ih ‘this’; indefinite pron-ouns, e.g., ਕੋਈ kōī, ਕੁਝ kujh, ਸਾਰੇ sārē etc.; relative pronouns (to join two clauses in a complex sentence), e.g., ਜੋ jō and ਜਿਹੜਾ jihṛā and interrogative pronouns, e.g., ਕੌਣ kauṇ ‘who’, ਕੀ kī ‘what’etc.

In Punjabi, postpositions follow the noun or pronoun unlike English, where these precede the noun or pronoun, and thus termed prepositions. In Punjabi, the postpositions can be classified into two types, namely, inflected postpositions and uninflected postposition. For example, ਦਾ dā ‘of’ (marker of possessive case) postposition is an inflected postposition because it changes forms for gender, number and case. There are a large number of postpositions, which do not change forms at all, e.g., ਨੇ nē (instrumental or agentive case marker), ਨੂੰ nūṃ (generally used with objects), and ਤੋਂ tōṃ ‘from’ are known as uninflected postpositions [3].

5. Implementation for NLization of Punjabi Language

The proposed NLization system for Punjabi has been tested on three types of part of speech sentences, which are Nouns, Pronouns, and Prepositions. Table 1 depicts the number T-rules and Dictionary words written corresponding to each type of part of speech.

Table 1: Details of NLization for each Part of Speech

Type	Sentences Processed	Dictionary Entries	T-rules Used
Nouns	36	15	30
Pronouns	33	10	25
Prepositions	40	25	60

NLization process works in sequence: First it Identifies all nodes - all UWs, relations, attributes are identified in this phase. Secondly Dictionary lookup - meanings of UWs are extracted from dictionary. Third firing all the relevant Rules- T-rules and D-rules Rule are fired to produce the intermediate and the final Output. Final result- when all the relevant rules got fired then the final result is get formed in Punjabi natural language.

5.1 NLization of Nouns

The process of NLization of input UNL sentence containing Noun to Punjabi language sentence is illustrated with an example given below

Example 1:
an almost beautiful car

UNL expression:
{unl}
mod (car:07.@indef, beautiful:05.@almost)
{/unl}

Equivalent Punjabi sentence:
ਲਗਭਗ ਸੋਹਣੀ ਇਕ ਗੱਡੀ

lagbhag sōhṇī ik gaḍḍī

As given in UNL expression it contains two root words, first ‘beautiful’ and second ‘car’. The detail process of UNL sentence given in example 1 is shown in Table 2.

Table 2: NLization of Noun given in Example 1

Rule Fired	**Description**	**Action Taken**
(%x, M3) :=(%x,-M3, +FLX (SNG: =0>""; PLR: =0>"ਆਂ";));	This paradigm M3 has been defined to attach user word "ਆਂ" in attribute node. Here attribute *‘FLX’* is indicates inflections.	**To:** [car:07.@indef] **Result:** ["ਗੱਡੀ":07.@indef] **To:** [beautiful:05.@almost] **Result:** ["ਸੋਹਣੀ":05.@almost]
(N,@indef,%a):=(("ਇਕ")(" ")(%a, @indef),%a,+NA,-@indef);	This rule creates two new nodes *i.e.* "ਇਕ" and blank space for '@indef' attribute	**To:** [car:07.@indef] **Result:** [sc:02(#L:02(ਇਕ:06, -:08); #L:02(-

	associated with Noun.	:08, car:07))] Output: ਇਕ ਗੱਡੀ
({N|V|D|J},FLX,^inflected,%x):=(!FLX,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the root word.	**To:** [car:07] **Result:** ["ਗੱਡੀ":07] **To:** [beautiful:05.@almost] **Result:** ["ਸੋਹਣੀ":05.@almost]
mod(%a,N,@indef;%b,J,@almost):=(("ਲਗਭਗ")(" ")(%b)(" ")(%a));	This rule creates five nodes for a relation 'mod' between node %a and %b. Here first node is "ਲਗਭਗ", second is blank space, third is node %b, fourth is blank space and fifth is node %a.	Now the Sentence becomes →ਲਗਭਗ ਸੋਹਣੀ ਇਕ ਗੱਡੀ

5.2 NLization of Pronouns

The process of NLization of input UNL sentence containing Pronoun to Punjabi language sentence is illustrated with an example sentence given below

Example 2:
my books

UNL expression:
{unl}
pos (book:03.@pl, 00:07.@1)
{/unl}

Equivalent Punjabi sentence:
ਮੇਰੀਆਂ ਕਿਤਾਬਾਂ

mērīāṃ kitābāṃ

As given in UNL expression it contains two root words, first ‘my’ and second ‘book’. The detail process of UNL sentence given in example 2 is shown in Table 3.

Table 3: NLization of Pronoun given in Example 2

Rule Fired	Description	Action Taken
(%x, M3) :=(%x,-M3, +FLX (SNG:=0>""; PLR: =0>"ਆਂ";));	This paradigm M3 attaches user word “ਆਂ” with the root word.	**To:** [00:07.@1] **Result:** ["ਮੇਰੀ":07.@1]
(%x, M2) :=(%x,-M2, +FLX (SNG:=0>""; PLR: =0>"ਾਂ";));	Paradigm M2 attach word “ਾਂ” with the root word when root word has *‘PLR’* attribute associated with it.	**To:** [book:03.@pl] **Result:** ["ਕਿਤਾਬ":03.@pl]
pos(%a,N,@pl;%b,D,^PLR):=pos(%a;%b,+NUM=PLR);	This rule creates a node i.e. node "%b" followed by node "%a" and update the NUM (number) value of the node to	**To:** pos(book:03.@pl, 00:07.@1) **Result:** pos(book:03.@pl, 00:07.@1)

	PLR (plural).	
(N,@pl,%a):=(%a,-NUM,NUM=PLR,-@pl);	This rule creates a node and update its NUM i.e. information to PLR i.e. plural.	**To:** [book:03.@pl] **Result:** ["ਕਿਤਾਬ":03]
pos(%a,N;%b,POD):=(%b)(" ")(%a);	This rule creates three nodes, first is node "%b", second is blank space and third is node "%a".	**To:** pos(book:03, 00:07.@1) **Result:** #L(00:07.@1 , -:01); #L(-:01, book:03) **Output:** ਮੇਰੀ ਕਿਤਾਬ
({N|V|D|J},FLX,^inflected,%x):=(!FLX ,-FLX,+inflected,%x);	It fires the corresponding paradigm rule to inflect the associated root word.	**To:** [00:07.@1] **Result:** ["ਮੇਰੀਆਂ":07. @1] To: [book:03] **Result:** ["ਕਿਤਾਬਾਂ":03] **Output:** ਮੇਰੀਆਂ ਕਿਤਾਬਾਂ

5.3 NLization of Prepositions

The process of NLization of input UNL sentence containing Preposition to natural language sentence is illustrated with an example sentence given below

Example 3:
the book on the table about Paris without pictures

UNL expression:
{unl}
plc (book:03.@def, table:09.@def.@on)
cnt (book:03.@def, Paris:0D.@about)
man (book:03.@def, picture:0H.@pl.@without)
{/unl}

Equivalent Punjabi sentence:
ਮੇਜ ਉੱਤੇ ਪੈਰਿਸ ਬਾਰੇ ਤਸਵੀਰ ਤੋਂ ਬਿਨਾਂ ਕਿਤਾਬ

mēj uttē pairis bārē tasvīr tōṃ bināṃ kitāb

As given in UNL expression it contains four root words, first is 'book' and second 'table', third is 'Paris' and fourth is 'picture'. The detail process of UNL sentence given in example 3 is shown in Table 4.

Table 4: NLization of Preposition given in Example 3

Rule Fired	**Description**	**Action Taken**
(%x,@def):=(%x,-@def);	It resolves '@def' attribute to remove keyword "the" from the English sentence.	**To:** [book:03.@def] **Result:** ["ਕਿਤਾਬ":03] **To:** [table:09.@def. @on]

		Result: ["ਮੇਜ":09.@on]
(N,NOU,@on,%b):=((%b,-@on)(" ਉੱਤੇ"),N,NOU,NUM=%b,DONE,%f)	This rule creates two nodes i.e. first is node %b followed by second node "ਉੱਤੇ".	**To:** [table:09.@on] **Result:** [sc:01(#L:01(table:09, ਉੱਤੇ :02))] Output: ਮੇਜ ਉੱਤੇ
(N,NOU,@without,%b):=((%b,-@without)(" ਤੋਂ ਬਿਨਾਂ",%x),N,NOU,NUM=%b,DONE,%f);	It resolves the attribute '@without' with noun. It creates two nodes i.e. node %b followed by node "ਤੋਂ ਬਿਨਾਂ".	**To:** [picture:0H.@pl. @without] **Result:** [sc:02(#L:02(picture:0H.@pl, ਤੋਂ ਬਿਨਾਂ :05))] Output: ਤਸਵੀਰ ਤੋਂ ਬਿਨਾਂ
(N,PPN,SNGT,@about,%a):=((%a,-@about)(" ਬਾਰੇ "),N,PPN,SNGT,DONE,%f);	It resolves the attribute '@about' with noun. This rule creates two nodes i.e. first node is %a and second is "ਬਾਰੇ".	**To:** [Paris:0D.@abou t] **Result:** [sc:03(#L:03(Paris:0D, ਬਾਰੇ :07))] Output: ਪੈਰਿਸ ਬਾਰੇ
plc(N,NOU,%a;N,{PPN|NOU},%b):=((%b)(%a),%f);	This rule resolves 'plc' i.e. place relationship between two nodes, first is node '%b', second is node '%a'.	**To:** plc(book:03, :01) **Result:** [sc:04(#L:04(:01, book:03))]
cnt(N,NOU,%a;N,PPN,%b):=((%b)(%a),%f);	It resolves the 'cnt' i.e. content relationship between two nodes, first is node '%b', second is node '%a'.	**To:** cnt(book:03, :03) **Result:** [sc:05(#L:05(:03, book:03))]
man(N,NOU,%a;N,{NOU|PPN},{SNG|SNGT},%b):=((%b)(%a),%f);	This rule resolves the 'man' *i.e.* manner relationship between two nodes, first is node '%b', second is node '%a'.	**To:** man(book:03, :02) **Result:** [sc:06(#L:06(:02, book:03))] **Output:** ਮੇਜ ਉੱਤੇ ਪੈਰਿਸ ਬਾਰੇ ਤਸਵੀਰ ਤੋਂ ਬਿਨਾਂ ਕਿਤਾਬ

6. Result and Discussion

In this paper, a Punjabi DeConverter (EUGENE) has been discussed, implemented and tested. EUGENE uses generation rules for UNL relation resolution and generation of attributes from the input UNL sentences. More than one hundred thirty DeConversion generation rules and sentences are processed. This proposed DeConverter has been tested for its performance via F-measure a tool available in public domain on the UNL Web Server. F-measure is the measure of a grammar's accuracy. It considers both the precision and the recall of the grammar to compute the score, according to the formula F-measure = 2 x ((precision x recall) / (precision + recall))

In this paper multi-lingual to cross-lingual generation and emerging problems for the definition of an Interlingua discussed. The results of testing are encouraging and the outputs of our system are very good. Figure 2 depicts the F-score for different part of speech sentences.

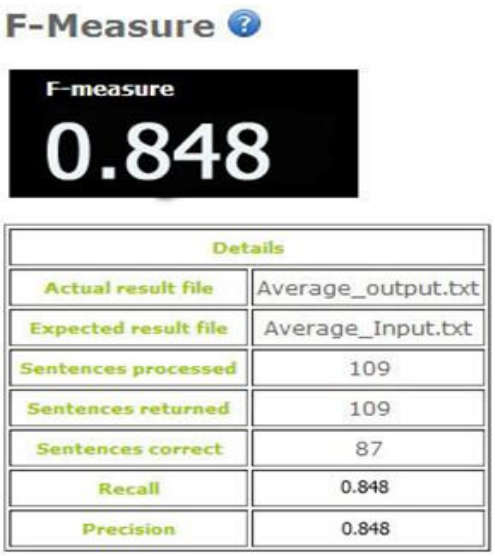

Figure 3: F-Measure for different Part of Speech Sentences

7. Conclusion and Future Scope

This conclusion is based on experience gained from implementing natural language generation components. The system still needs many improvements like handling parser errors, improving rule base, compiling verb properties, *etc*. The coverage and accuracy of system can further be improved by expanding the Punjabi-UW dictionary and enriching it with more semantic information. The rule base can further be improved to increase the accuracy of the system and to handle very large and complex sentences. The system can further be tested on other languages by using corresponding language's rule base and lexicon.

References

[1] Uchida H 2005 Universal Networking Language (UNL): Specifications version 2005, UNDL Foundation.

[2] "NLization","http://www.unlweb.net/wiki/NLization#Units, December, 2012.

[3] Kumar P, "UNL Based Machine Translation System for Punjabi Language" PhD thesis, Thapar University, Patiala, 2012.

[4] UNDL Foundation,"http://www.undl.org", December, 2012.

[5] Dhanabalan T, Geetha T V 2003 UNL DeConverter for Tamil. *Proc. Int. Conf. on the Convergence of Knowledge, Culture, Language and Information Technologies,* Alexandria, Egypt, 1-6.

[6] Blanc É 2005 about and around the French EnConverter and the French DeConverter. *Universal Network Language: Advances in Theory and Applications, Ed(s) Cardenas J, Gelbukh A, Tovar E,* México, Research on Computing Science: 157-166.

[7] Boguslavsky I M, Iomdin L, Sizov V G 2005 Interactive EnConversion by means of the ETAP-3 system. *Universal Network Language: Advances in Theory and Applications, Ed(s) Cárdenas J, Gelbukh A, Tovar E,* México, Research on Computing Science: 230-240.

[8] Shi X, Chen Y 2005 A UNL DeConverter for Chinese Universal Network Language. *Universal Network Language: Advances in Theory and Applications, Ed(s) Cardenas J, Gelbukh A, Tovar E,* México, Research on Computing Science: 167-174.

[9] Pelizzoni J M, Nunes M V 2005 Flexibility, configurability and optimality in UNL deconversion via multiparadigm programming. *Universal Network Language: Advances in Theory and Applications, Ed(s) Cardenas J, Gelbukh A, Tovar E,* México, Research on Computing Science: 175-194.

[10] Daoud D M 2005 Arabic Generation in the Framework of the Universal Networking Language. *Universal Network Language: Advances in Theory and Applications, Ed(s) Cárdenas J, Gelbukh A, Tovar E,* México, Research on Computing Science: 195-209.

[11] Keshari B, Bista S K 2005 UNL Nepali Deconverter. *Proc. 3rd International CALIBER*, Cochin, 70-76.

[12] Singh S,"Parsing and syntax planning for UNL Punjabi DeConverter" M.E. Thesis, Thapar University, Patiala, 2007.

[13] UNDL foundation "EUGENE","http://dev.undlfoundation.org/generation/login.jsp", January 2013.

[14] UNDL foundation "UNLdev", "http://www.unlweb.net/wiki/UNLdev", January 2013.

Authors

Ashutosh Verma is pursuing his M.Tech in Computer Science from Thapar University, Patiala. He is undergoing the thesis of his M.Tech entitled "NLization of different part-of-speech sentences" in the supervision of Parteek Kumar. His current research interest includes UNL Based DeConversion,UNL Dictionary Specs, UNL Grammer Specs.

Parteek Kumar is Assistant Professor in the Department of Computer Science and Engineering at Thapar University, Patiala. He has more than fifteen years of academic experience. He has earned his B.Tech from SLIET and MS from BITS Pilani. He has completed his Ph.D on "UNL Based Machine Translation System for Punjabi Language" from Thapar University. He has published more than 50 research papers and articles in Journals, Conferences and Magazines of repute. His research work with UNDL foundation, Geneva, Switzerland provides international exposure to him and he was a member for Advanced UNL School at Alexandria, EGYPT in 2012. He has also undergone various faculty development programme from industries like Sun Microsystems, TCS and Infosys. He has authored six books including Simplified Approach to DBMS. He is acting as Co-PI for the research Project on Development of Punjabi WordNet sponsored By Department of Information Technology, Ministry of Communication and Information Technology, Govt. of India.