UNL Nepali Deconverter

Birendra Keshari Sanat Kumar Bista

Abstract

This paper discusses about the Interlingua approach of machine translation, especially the Nepali generator part of Interlingua based machine translation in which the Interlingua used is UNL (Universal Networking Language). Nepali is the national language of Nepal, a country in Indian sub continental region. UNL is an Interlingua proposed by United Nations University/Institute Of Advanced Studies, Tokyo, Japan to remove language barrier and digital divide in the World Wide Web. This paper describes about the architecture and design of UNL Nepali Deconverter (Generator), that has been implemented using a tool called DeCo*, a language neutral generator. Nepali sentences are generated using information present in Nepali language at different linguistic levels. Information like case relations, case markers etc. in Nepali sentences can be generated from morphological level itself since Nepali is a morphologically rich language.*

Keywords : Universal Networking Language, Machine Translation, Nepali Language.

0. Introduction

There are several trends in Machine Translation Systems. Interlingua approach is one of them. In this approach the source language sentences are first analyzed and converted to an intermediate form called Interlingua, which is an equivalent semantic form of the source language. The Interlingua representation is then analyzed using source-target language dictionary and grammar to generate the target language sentences. In UNL based system, Enconverter analyzes the source language to produce UNL and Deconverter generates the target language. UNL as such has been designed as a standard Interlingua (Uchida and Zhu., 2002). Enconverter and Deconverter provide language neutral framework for source language analysis and target language generation. UNL is going to be the future language for computer (Uchida and Zhu., 2002) . This paper describes the Deconverter module for Nepali.

While English follows SVO pattern, Nepali follows SOV pattern. Nepali is a free word order language. This is due to the reason that in Nepali, thematic case relations of nouns and pronouns, number, tense, gender and honor markers of verbs are conveyed by suffixes.

1. Universal Networking Language (UNL)

UNL is an artificial digital language that represents meaning sentence by sentence. The representation is in logical form. Such logically formed expressions can be viewed as a semantic net or an acyclic directed hyper graph where a node can be a graph itself. Each node represents UW (Universal Word) or concepts. The arc represents relation between the two concepts. So, UNL can also be viewed as a set of binary relations between the concepts.

For example the Nepali sentence in transliterated form, ‘Ram kaathmaandu yunivarsithimaa padhcha’, which means ‘Ram reads in kathmandu university’, can be expressed using following UNL expressions:

{unl}
[S]
agt(read.@entry.@present,Ram)
plc(read.@entry.@present,university)
mod(university,kathmandu)
[/S]
{/unl}
The above UNL expressions can be represented as a graph in the following way.

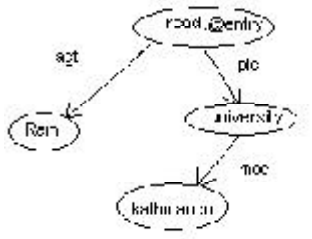

Figure 1. Graph representation of above UNL expressions.

In the above example read, student etc. are UWs and agt, plc and mod are UNL relations. The symbols starting with '@' character like @entry and @present are called UNL attributes.

UWs are based on English words but they are made unambiguous by their position in UNL Knowledge Base (KB). UNL KB maintains the hierarchy of concepts that are universal i.e. the concepts is not any language, culture or tradition specific (Uchida and Zhu., 2002) . Furthermore, restricted UW is used to avoid ambiguity by restricting the concept. For example the word "book" can represent two concepts; 'a thing' or 'act (of booking)'. So, it is disambiguated by using restricted UW like book(icl>do) or book(icl>thing).

Relations represent the semantic role such as agent, object, condition, and, place, co-agent etc. that UWs play. Attributes are attached to UWs to express the objectivity of the sentence. There are several such relations and attributes specified *UNL Specification of UNL Center* (UNL Center, 2003). UW's and relations express subjectivity of the sentence. More information about UNL can be found in Deconverter Specification of UNL Center (UNL Center, 2000), UNL Specification of UNL Center (UNL Center, 2003), (Uchida and Zhu., 2002) and many others.

1.1 UNL Benefits

Once the information is converted to UNL form, it becomes language neutral and it can be converted to other different languages. Thus, it can be used for information exchange between languages. Information in a source language can be converted to UNL using source language Deconverter and then using Enconverter of target language, UNL can be enconverted in to that language.

Since, UNL is in logical form, knowledge processing can be done unambiguously to produce useful and desired results.

2. UNL Nepali Deconverter

A tool called *DeCo* has been designed by UNU/IAS as a language independent generator that provides synchronously a framework for morphological and syntactic generation and word selection for natural collocation. Its structure has been shown in figure 2.

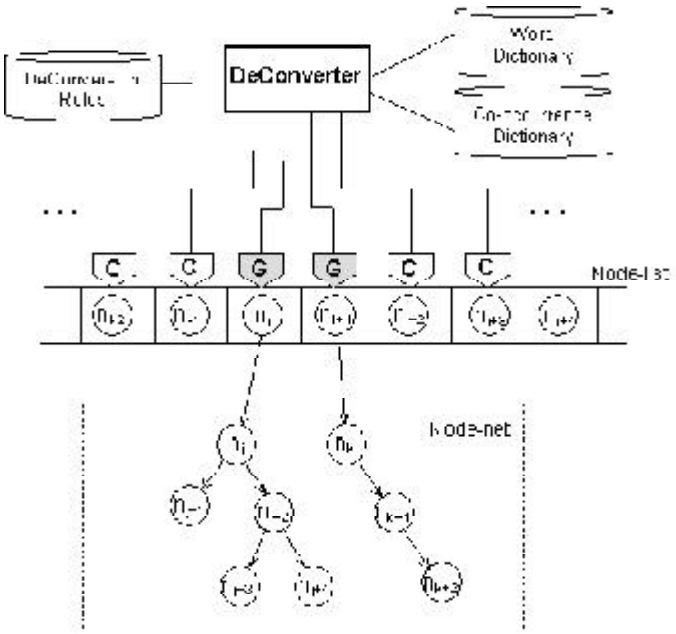

Figure 2. Deconverter Structure.

It can deconvert both context-sensitive and context-free languages. It uses target language specific Word Dictionary, Co-occurence Dictionary and Deconversion rules to generate the target language. So, developing a Deconverter for a language means developing dictionaries and writing deconversion rules, which are understood by the *DeCo* and these are language dependent. The structure of a *DeCo* has been shown in figure 2.

Each entry in Word Dictionary includes native language Head Word, corresponding UW, and the attributes. Attributes include grammatical and semantic attributes. An example of an entry in Nepali Language Word Dictionary Attributes can be:

[kitaaba] “book(icl>thing)” (N,C,INANI,PHY)<N,0,0>;

In the above example [kitaaba] is the Nepali Head Word, book(icl>thing) is UW and (N,C,INANI,PHY) is the attribute list.

First, the deconversion rules are converted into binary format and then binary format rules are loaded. The UNL expressions are converted in to semantic net called Node-net. The UWs are replaced with corresponding native language Head Words. If it is not possible to unambiguously decide the correct Head Word for a given UW, Co-occurence dictionary is used. Co-occurence dictionary contains more semantic information for proper word selection without the ambiguity. But the use of Co-occurence dictionary is optional. We have not used Co-occurence dictionary for UNL Nepali Deconverter.

Node-net represents the hyper graph (a representation of UNL expressions) that has not yet been visited. Each node contains certain attributes initially loaded from the Language Dictionary and sometime generated by *DeCo* during runtime. These attributes can be read or deleted or new attributes can be added. This is governed by deconversion rules. Each node in the Node-net is traversed and inserted in to the Node-list.

Node-list shows the current list of nodes that the Deconverter can look at through its windows. Node-list includes two generation windows circumscribed by condition windows. At the initial stage before any deconversion rule application there are three nodes in the Node-list, *Sentence Head* node, *Entry* node and *Sentence Tail* node. This is explained in *Deconverter Specification of UNL Center* (UNL Center, 2000). The generation occurs at the generation windows, when the conditions in the condition windows are satisfied.

The result of rule application is operation on the nodes in Node-list like changing attributes, copy, shift, delete, exchange etc. and/or insertion of nodes from Node-net to Node-list. The rule application halts when either Left Generation Window reaches the *Sentence Tail* node or Right Generation Window reached the *Sentence Head* node. If post-editing is required the Deconverter will start applying post editing rules. Post editing rule has not been used for UNL Nepali Deconverter. At the end, the nodes in the Node-list represent the generated sentence. More information about *DeCo* can be found in *Deconverter Specification of UNL Center* (UNL Center, 2000).

3. Architectural Design

There are basically two modules for UNL Nepali Deconversion; Syntax Planning Module and Morphology Generation Module. The overall architecture and structure of Nepali Deconverter has been shown in figure 3.

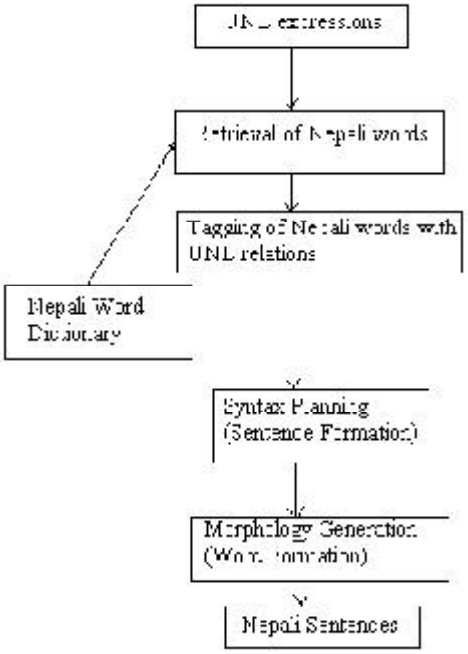

Figure 3. Nepali Deconverter Structure.

3.1 Syntax Planning Module

This module is responsible for Nepali sentence formation by syntax planning. In UNL relation *r*el(UW1,UW2), UW1 is the parent node and UW2 is the child node. We plan the syntax, by deciding which child to insert first and at what position (left or right) with respect to other child of its parent. This is done by creating a (M+1)x(M+1) priority matrix where M is the total number of relations. We write the relation labels in the first row and first columns. Each Mij can be 'L', 'R' or nothing (we represented it by '-'), where i is the row number and j is the column number.

Mij = 'L' means that the child labelled with relation label in row *i* is to be inserted in to the Node-List to the left of the child labelled with relation label in column*j*. Similarly, Mij = 'R' means that the child labelled with relation label row*i* is to inserted in to the Node-List to the right of the child labelled with relation in column *j*. Mij = - means the position with respect to each other is not applicable. A rank for each relation label is calculated by adding the number of 'R' in the row of each relation label. The higher the value of the rank the further right from the main verb is the corresponding word.

	agt	obj	ben	Rank
agt	-	L	L	0
obj	R	-	R	2
ben	R	L	-	1

Table 1. Priority Matrix

The above priority matrix, table 1, considers only three relations and suggests that child of relation agt is the leftmost element, child of ben is the middle element and child of obj is the rightmost element. Let's plan the syntax of the following UNL expression according to the rule from above table.

(Ram bought an apple for you)
{unl}
[S]
agt(buy.@entry.@past,Ram)
obj(buy.@entry.@past,apple.@def)
ben(buy.@entry.@past,you)
[/S]
{/unl}

According to above table the child of agt is 'Ram'. So, it will be the leftmost node. The child of ben is 'you' so, it will be the middle node and similarly, the child of obj, 'apples' will be the last node. So, the syntax generated will look like; *Ram(le) timi(rolaagi) syaauu kin(yo).* The morphemes, which are later generated during morphology generation phase, are shown inside "()".

Since, the syntax depends upon the sentence type, there are set of syntax planning rules as described above, for each sentence type. The sentence type is determined by checking the attributes attached to the *entry node.*

4. Morphology Generation Module

This module is responsible for proper word formation though morphology generation. This module generates most of the words. This module handles noun, verb and adjective morphology generation. This module not only inflects the root words, but also introduces conjunctions, case markers and any other new words if necessary.

The morphological rules are governed by UNL relations and attributes. Morphological rules due to UNL relations are called relation label morphology. Some relation label morphology rules have been shown in table 2. These rules introduce affixes. For example; relation ben appears in relation, suffix 'kolaagi' is added to the child. Sometimes new words are introduced. For example; if two UWs are related by relation and, new Nepali word 'ra' is introduced which has same meaning as 'and' in English.

Relation	**Definition**	**Word to be introduced**
agt	a thing that initiates an action	"le"
and	conjuctive relation between two concepts	"ra"
bas	thing used as basis for expressing degree	"bhandaa"
ben	indirectly related beneficiary	"kolaagi"
cao	thing not in focus	"sita"
con	non-focused event or state that conditions a focused event or state	"yadi"
fmt	range between two things	"samma" "dekhi"
gol	final state of an object	"laaii"
ins	instrument to carry out an event	"le"
met	means to carry out an event	"sita" "le"
opl	a place in focus affected by an event	"maa"
or	Disjunctive relation between two concepts	"athawaa"
per	basis or unit or proportion	"prati"
plc	Place where an event occurs	"maa"
pos	possessor of a thing	"ko" "kaa"
pof	concept of which a focused thing is a part	"ko
rsn	reason why an event or a state happens	"legardaa"
src	initial state of an object or an event	"baata"
tmt	Time an event ends or a state becomes false	"samma"
tmf	Time an event occurs or a state becomes true	"dekhi"
via	intermediate place or state of an event	"bhaera"

Table 2. UNL relations and Nepali affixes/words to be introduced.

UNL attributes, which expresses information like aspect, tense, number, gender, speaker's view point etc. also play an important role in morphology generation. For example, the attribute @pl means plural. When a noun has an attribute @pl, suffix 'haruu' is added to the stem (noun/pronoun). Similarly, if @not is attached to a verb, the verb needs to be negated. Suffix 'na'is added at the end of the main predicate verb to negate it.

4. Conclusion

This paper has described the development of UNL Nepali Deconverter, a Nepali language generator. Techniques of syntax planning and morphology generation have been used. Syntax planning has been done by studying the syntactic structure of the Nepali sentences. Morphology has been generated by the effect of UNL relations and attributes on Nepali word morphology. Most of the information has been generated at morphological level. The current Nepali Deconverter can deconvert moderately complex UNL expressions. Due to lack of standard UNL test data the system has yet not been formally evaluated. The size of the dictionary is small (only about 500 entries). However the size of the dictionary can be increased in the similar manner.

Nepali Deconverter can be coupled with other language Enconverter to develop a complete Machine Translation system. It can be used for future UNL Nepali viewer.

5. References

1. Dave S., Parikh J. and Bhattacharya P. (2002). Interlingua Based English Hindi Machine Translation and Language Divergence. Journal of Machine Translation, Volume 17.
2. Dhanabalan T. and Geetha T.V.(2003). UNL Deconverter For Tamil.
3. Uchida H. and Zhu M.(2001). The UNL Beyond MT. United Nations University, Tokyo.
4. UNL Center.(2000). Deconverter Specification. UNDL Foundation.
5. UNL Center.(2000). Enconverter Specification. UNDL Foundation.
6. UNL Center. (2003). UNL Specification. UNDL Foundation.
7. Uchida H. and Zhu M. (1999).A gift for a millennium . United Nations University.

About Authors

Birendra Keshari is a graduate in Computer Engineering from Kathmandu University. He is currently employed as a Teaching and Research Assistant in the Department of Computer Science and Engineering, Kathmandu University. Mr. Birendra is involved in Language Computing Research from past one and half year and is also a member of Language Processing Research Unit at Kathmandu University (www.ku.edu.np/cse/unl). He is also a member of the Nepali Language Computing Project at Madan Puraskar Pustakalaya. His general research interest is in Natural Language processing (especially Nepali Language Computing), Artificial Intelligence and Logic Programming.
E-mail : birendra@ku.edu.np

Sanat Kumar Bista, is an Assistant Professor in the Department of Computer Science and Engineering at Kathmandu University, Dhulikhel, Nepal, where he has been involved in teaching and research related to Computer Science and Information Technology. He currently leads the Language Processing Research Unit (LPRU) at Kathmandu University. He is a project leader from Kathmandu University for "Nepali Language Computing Project", being carried out in collaboration with Madan Puraskar Pustakalaya, Nepal (http://mpp.org.np) as a part of the PAN localization project(http://www.panl10n.net). Sanat's main research interests lie in the area of Localization, Multilingual Computing and Digital Libraries.
E-mail : nepal.sanat@ku.edu.np