[ICUKL November 2002, Goa, India]

Tamil to UNL EnConverter

T.Dhanabalan
K.Saravanan
T.V.Geetha
Resource Center for Indian Language Technology Solutions (RCILTS) – Tamil,
School of Computer Science and Engineering,
Anna University,
Chennai – 600 025,
India.

dhanabalan@cs.annauniv.edu
k_saravan76@hotmail.com
tvgee@cs.annauniv.edu

Abstract

This paper discusses the interlingua approach to machine translation. Here Universal Networking Language (UNL) has been used as the intermediate representation. In this paper the EnConverter from Tamil to UNL has been described. The information needed to construct the UNL structure is available at different linguistic levels. Tamil being a morphologically rich language allows a large amount of information including syntactic categorization, and thematic case relation to be extracted from the morphological level itself. Information about relating concepts like verbs to thematic cases, adjectival components to nouns and adverbial components to verbs is available through syntactic functional grouping that has been done by the specially designed parser taking into consideration the requirements of the UNL structure.

1. Introduction

There are many possible approaches to machine translation. One approach is the interlingual machine translation system where Source Language (SL) text is analyzed using source language dictionary and grammatical information and converted into an interlingual representation. This interlingual representation along with SL to Target Language (TL) dictionary, and grammar is synthesized to generate TL text [8]. In this paper the interlingua structure used to translate from source language to target language is the semantically biased language independent UNL representation. Any translation system using UNL as intermediate representation needs to have an EnConverter from the source language to UNL and a DeConverter from UNL to target language. The UNL has been designed to act as a standard interlingua structure and enables automatic translation from source language to target language in case an EnConverter is designed for the source language and deconverters are available from UNL to target languages.

In this paper the EnConverter from Tamil to UNL has been described. Tamil is a free word order language basically because of its rich morphological characteristics. The attachment of constituents to other constituents of the sentence is achieved through morphology and grouping across words through syntactic functional attachments and not through fixed word ordering of words. In this work morphological features obtained from words using an existing morphological analyzer and syntactical information obtained from a specially designed parser has been used in analyzing the source language and converting it into UNL. But the language divergence between Hindi and English can be considered representative of the divergence between the SOV and SVO classes of languages [9].

2.Universal Networking Language (UNL)

2.1 Introduction

The Universal Networking Language (UNL) is an electronic language in the form of a semantic network for computers to express and exchange every kind of information developed at United Nations University [2]. This language is assumed to express meanings in the same standardized way as HTML presents its layout.

The UNL represents information, i.e. meaning, sentence by sentence. Sentence information is represented as a hyper-graph having Universal Words (UWs) as nodes and relations as arcs. This hyper-graph is also represented by a set of directed binary relations, each between two of the UWs present in the sentence. The UNL expresses information classifying objectivity and subjectivity. Objectivity is expressed using UWs and relations. Subjectivity is expressed using attributes by attaching them to UWs. Nodes, or Universal Words (UWs) are words loaned from English and disambiguated by their positioning in a knowledge base (KB) of conceptual hierarchies [1]. Function words, such as determiners and auxiliaries are represented in the form of attributes to UWs, provided that these function words contribute to the meaning.

Binary relations are the building blocks of UNL sentences. They are made up of a relation and two UWs. Each relation is labeled with one of the possible label descriptors. Relations that link UWs are labeled with semantic roles of the type such as *agent, object, experiencer, time, place, cause*, which characterize the relationships between the concepts participating in the events or states a natural language sentence may denote. The complete specifications of UNL has been described in a number of papers [2], [3], [5], [6], [10] and [11].

2.2 UNL Features

The text – once converted into UNL – can be converted to many different languages [3]. For example, home pages can be designed in one's native language and then enconverted to UNL before being uploaded. Once a home page is expressed in UNL, it can be read in a variety of natural languages. As a result of this standardized meaning of UNL representation, documents no need to be multiplied in order to represent the content in different natural languages. The meaning representation is directly available to retrieval and indexing mechanisms and tools for automatic summarizing and knowledge

extraction, and it will be converted to a natural language only when communicating with a human user.

The process of the presentation of UNL format to natural language text will be taken over by UNL viewer. It uses UNL DeConverter that deconverts the UNL format into natural language text. So UNL-viewer represents a new generation of web-browser, which, in addition to their capabilities to handle scripts, that are equipped with one or more national UNL-DeConverter in order to display the meaning of the content in a national language.

The UNL greatly reduces the costs of developing knowledge or contents necessary for knowledge processing by sharing knowledge and contents. Furthermore, if the type of knowledge required for doing something through software is described in a language for computers, such as the UNL, the software only needs to interpret instructions written in the language to be able to perform its functions. And other software can share these instructions. It is then feasible to accumulate such knowledge for computers in the same way as a library for humans.

3. Tamil to UNL EnConverter

An EnConverter is a software that automatically or interactively enconverts natural language text into UNL. As it generates UNL from natural languages, enables peoples to make UNL documents without any knowledge about UNL. It means that users of the UNL system do not need to learn UNL.

EnConverter is a language independent parser that provides synchronously a framework for morphological, syntactic and semantic analysis [4]. It would be impossible to solve an ambiguity in morphological analysis without the use of syntactic or semantic information. Also, it would be impossible to solve an ambiguity in syntactic analysis without the use of semantic information.

EnConverter generates UNL expressions from sentences (or lists of words of sentences) of a Tamil language by applying enconversion rules. In addition to the fundamental function of enconversion, it checks the formats of rules, and outputs the messages for any errors. It also outputs the information required for each stage of enconversion in different levels. With these facilities, a rule developer can easily develop and improve rules by using EnConverter.

EnConverter loads the enconversion rules and the rule checker works while converting rules. Once the rules are made, they are stored automatically and can be used directly the next time without rule conversion.

- Convert or load the rules

Secondly, it inputs a string or a list of morphemes/words of a sentence of Tamil language.

- Input a Tamil sentence

Then, it starts to apply rules to the Node-list from the initial state. EnConverter applies enconversion rules to the Node-list. The process of rule application is to find a suitable rule and to take actions or operate on the Node-list in order to create a syntactic functionalities and UNL network using the nodes in the Analysis Windows. If a string appears in a window, the system will retrieve the Word Dictionary and apply the rule to the candidates of word entries. In this case, if a word satisfies the conditions required for the window of a rule, this word is selected and the rule application succeeds. This process will be continued until the syntactic functions and UNL network are completed and only the entry node remains in the Node-list.

- Apply the rules and retrieve the Word Dictionary

Finally, it outputs the UNL network (Node-net) to the output file in the binary relation format of UNL expression.

- Output the UNL expressions

with the exception of the first process of rule conversion and loading, once EnConverter starts to work, it will repeat the other processes for all input sentences.

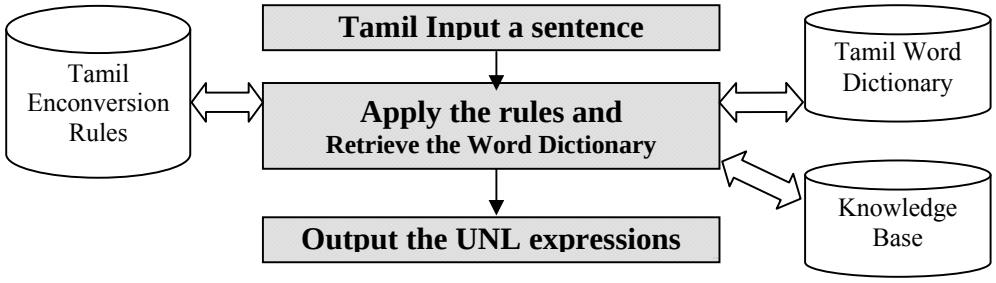

Fig 1 References between each process and the dictionary and rule files.

4. Tamil Language specific and language independent information

EnConverter analyses a sentence using the Word Dictionary, Knowledge Base, and Enconversion Rules [4]. It retrieves relevant dictionary entries from the Word Dictionary, operates on nodes in the Node-list by applying Enconversion Rules, and generates semantic networks of UNL by consulting the Knowledge Base.

The word entries of Tamil language are stored in the Word Dictionary. Each entry of the Word Dictionary is composed of three kinds of elements: the Headword, the Universal Word (UW) and the Grammatical Attributes. A headword is a notation/surface of a word of a Tamil language that composes the input sentence, and it is to be used as a

trigger for obtaining equivalent UWs from the Word Dictionary in enconversion. An UW expresses the meaning of the word and it is to be used in creating UNL networks (UNL expressions) of output. Grammatical Attributes are the information on how the word behaves in a sentence and they are to be used in enconversion rules.

All possible relations between each pair of UWs are defined in the UNL Knowledge Base (KB) using the UW system, a kind of hierarchy of UWs, with certainty values. When a relation is being established between two UWs by applying an enconversion rule, EnConverter consults with the UNL KB. If the relationship is approved, EnConverter will establish the relation between the two UWs (i.e., it will connect the two UWs using the relation label) and the rule application succeeds. If the relationship is not approved, no relation will be established between the two UWs and the rule application fails. To utilize the KB function, all the UWs used in a native language must be linked in the UNL KB.

For example binary relation **agt** in Knowledge base
UW1 – should be a action (do) and
UW2 – should be a thing
E.g. 'avan ooTinaan' (he ran)

'avan' - a thing
'ooTinaan' – do (action)

An Enconversion Rule is composed of Conditions for the nodes placed on the Analysis and Actions and/or Operations for the nodes placed on the Analysis Windows [4]. Such enconversion rules describe the kind of actions and/or operations that should be carried out for all phenomena of a language, and under what conditions. EnConverter will find the most suitable rule every time, and create a UNL expression. A set of UNL expressions of a sentence will finally be completed after having applied a set of all the necessary rules.

Basically the Enconverter needs certain information from the input sentences. The information is available at various linguistic levels either the morphological, syntactic or semantic levels. The amount and type of information available at each level is largely dependent on the characteristics of the language. This means the design of the Enconverter is decided by what information is needed by UNL and the nature of the language, which decides on the type of information that can be extracted from the various linguistic levels. UNL has separate concepts for noun, verb, adjective and adverb in other words there is a need to syntactically categorize the words of the sentence. For a noun the attributes to be included in the concept definition in UNL is number. For the verb the concept definition tense marker is required. The next important part of UNL is the definition of relations. These include case relations of noun concepts with the corresponding verbs, association of adverbial components with verb definitions and the association of adjectival components with the noun definitions.

In this paper the extraction of the information needed in building the UNL structures from the various linguistic levels of Tamil language has been discussed.

5. Morphologically generated information from word level analysis

Tamil is a morphologically rich language and hence a large amount of information can be obtained by morphologically analyzing the input sentences. The word level Morphological analyzer for Tamil analyzes the given derivative word and determines the root word and the various features conveyed by the suffixes. In the case of nouns the feature normally conveyed by attached suffixes is number.

- Example showing number attachment to nouns
  'malarkaL' meaning flowers → 'malar ' (root) + 'kaL' (number - plural)

However in the Tamil language the case suffix is also attached with the noun. This essentially means that morphologically analyzing the noun would enable one to determine the case relation of that noun with the associated verb. This information in languages that do not allow case attachment to nouns at the morphological level can be often obtained only through syntactic analysis of prepositional phrases where prepositions act as case indicators as in example English.

- Example showing case attachment to nouns

'avanukkaaka' meaning for him → 'avan' (root) + 'ukkaaka' (case - for)

The morphological analysis of a verb is much more complex in Tamil. A verb can be attached with simple tense information and person, number and gender (PNG) information.

- Example showing verb with simple tense and PNG information
  'ooTinaan' meaning He ran → 'ooTu' (root) + 'n' (past tense) + 'aan' (third – person, singular - number, masculine - gender)

The attachment of number and gender with verb enables sentences in Tamil to be subject absent. Attached to the verb are also suffixes that indicate aspect and modality. This means that a verb phrase such as 'may have been going' can be expressed as a single word 'cenrukonTiruntirukkalaam'. The splitting of this verb will yield all features conveyed by the English version. In Tamil adjectives and adverbs are indicated by special suffixes such as 'aana' and 'aaka' respectively. A few number of adjectives and adverbs occur in root form alone.

- Example showing complex verb with tense, aspect and modality
  'cenRukonTiruntirukkalaam' → 'cel' (root) + 'Ru' (verbal participle) + 'koL' (auxillary verb) + 'Tu' (verbal participle) + 'iru' (auxillary verb) + 'nttu' (verbal participle) + 'iru' (auxiliary verb) + 'kka' (infinitive) + 'aam'

- Example showing UNL formation from a given Tamil sentence
  'avan ooTukiRaan'

For the verb phrase ‘ooTukiRaan’, the morphological analyzer gives the information as root word, present tense, and third person singular masculine and from noun phrase ‘avan’ it will give as noun
So equivalent UWs

he(icl>person).@generic : 0
run(icl>do).@ present. @entry : 1

This equivalent UWs are collected from word dictionary with the above morphological analyzer results.

In most of the languages preposition comes as separate word. But in Tamil, there is no preposition and it comes as case marker along with noun and person, number, gender marker along with verb. This case ending marker and PNG marker information are used for syntactic functional grouping of words and finding out the correct binary relation. There are several case markers in Tamil like nominative case(‘o’), accusative case (‘ai’), instrumental case (‘aal’), dative case (‘kku’), locative case (‘il’, ‘iTam’), genitive case (‘in’, ‘uTaiya’, ‘atu’), ablative case (‘iliruntu’, ‘iTamiruntu’),…

The letter ‘t’, ‘nt’, ‘in’, ‘tt’, ‘T’ are represents past tense marker, ‘kiRu’ and ‘kinRu’ represents present tense marker and ‘p’, ‘v’, and ‘pp’ are represents future tense marker.

The verb takes Person, Number, Gender markers according to the subject of the verb. A single word (verb) contains all these information’s. Let us consider the word ‘seikiRaan’. Morphological output of this word is

‘Sei’ + ‘kiRu’ + ‘aan’
sei - verb
kiRu - present tense marker
aan - third person masculine

Let us consider the sentence ‘avan kaTaikku ooTinaan’. When this sentence given to morphological analyzer, that will gives the information such as noun, verb, past tense marker, case marker and gender marker (third person singular).

So the equivalent UWs in UNL format is

he(icl>person).@generic : 0
store(icl>place).@generic : 1
run(icl>do).@past.@entry : 2

Here @past information is collected from the morphological analyzer. Other attributes are collected from word dictionary.

6. Syntactic functional grouping based on knowledge base of UNL

Tamil as already discussed is a relatively free word order language. The free orderness of the language is achieved by two mechanisms namely richness in morphology and cues available that enable syntactic grouping of words. Though morphological endings allows thematic cases of the nouns to be determined, the complete binary relation can be constructed only with help of syntactic functional groupings.

The syntactic grouping or the job of the parser in the case of Tamil EnConverter is obtaining information to fill the binary relations obtained from the morphological analyzer phase. Though the morphological level does provide information about the possible case relations between the main verb and noun components there is ambiguity in some cases. This ambiguity is solved using a combination of information obtained at the syntactic and semantic levels. In this paper the parser has been designed to extract the required syntactic and semantic information in order to build the UNL relations. The parser mainly helps in deciding one among the thematic cases suggested by the morphological analyzer and to form the corresponding case based UNL binary relation.

Take the example of the sentence '*avan kaTaikku ooTinaan*' (he ran to store).

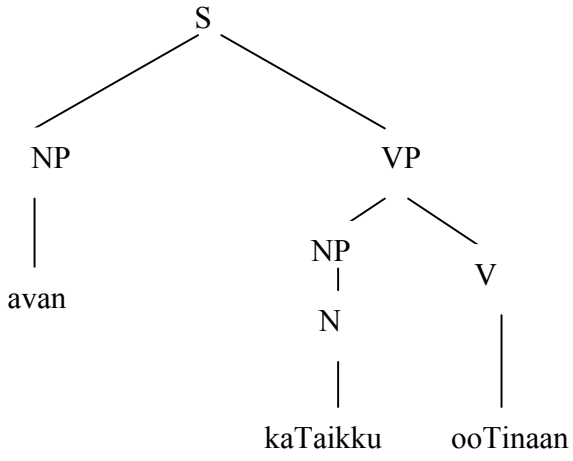

Fig 2. Syntactical structure of the sentence '*avan kaTaikku ooTinaan*'

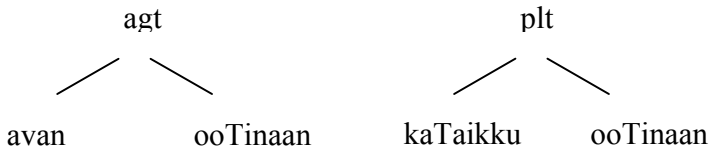

Fig 3. Possible binary relation for the sentence '*avan kaTaikku ooTinaan'*

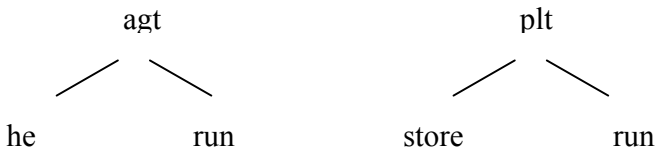

Fig 4. Equivalent Possible binary relation in UNL format

The noun without case ending is normally the agent in this case 'avan'. The word 'ooTinaan' is the action verb (run), which has been morphologically analyzed and obtained from the morphological analyzer. So the two words 'avan' and 'ooTinaan' are linked with the binary relation 'agt'. The binary relation 'agt' can be assigned in many ways. If that sentence contains only one nominative case or UNL dictionary contains person information for a word or subject matches with the verb then the 'agt' relation is assigned. But a noun with case ending 'ukkaaka' is a 'ben' (beneficiary) binary relation.
The suffix 'kku' specifies it to be a locative, temporal or beneficiary case at the morphological level. But in the word dictionary 'kaTai' specifies it to be a location. The words 'kaTaikku' and 'ooTinaan' are linked with the binary relation *'plt'*.

relation for *plt* (final place). Here
UW1 - is an event or state, and
UW2 - a place or thing defining a place (locative case)

The equivalent UNL format of the above sentence as follows.

```
[S]
{unl}
[W]
he(icl>person).@generic : 0
store(icl>place).@generic : 1
run(icl>do).@past.@entry : 2
[/W]
[R]
2 agt 0
2 plt 1
[/R]
{/unl}
[/S]
```

The morphological analyser detects adverbs and adjectives, however the intensifiers corresponding to these are determined by their position in the sentence. Thus the parser is needed to identify the adjectives or adverbs to which the intensifiers are attached in order to form the appropriate UNL modifier relation.

Take the example sentence '*avan mika azakaaka ooTinaan*' (he ran very beautifully). Here the intensifier 'mika' comes just before the adverb 'azakaaka'. So this intensifier modifies the adverb. These words are linked with the 'mod' relation. In a sentence a intensifier normally occurs before adjective or adverb.

UNL equivalents of the above sentence is

[S]
{unl}
[W]
he(icl>person).@generic : 0
very(icl>concept).@generic : 1
run(icl>do).@past.@entry : 2
beauty(aoj>thing) : 3
[/W]
[R]
2 agt 0
3 man 2
1 mod 3
[/R]
{/unl}
[/S]

Here the word 'avan' is the nominative case in this sentence and it matches with verb 'ooTu'. So these two words are linked with 'agt' relation. The word 'azakaaka' is adverb and it defines the characteristics of the event 'ooTu'. So these words are linked with the relation 'man'. And also the word 'mika' modifies the adverb 'azakaaka'. So these two words are linked with the relation 'mod'.

Participles in Tamil may function as postpositions when occurring with noun or as adverbial participles when occurring with verbs. At the morphological level if the participle is attached to the word then the function of the participle is identified. However in Tamil it is possible for the participle to occur in isolation. In this case the parser is needed to decide on the functionality of the participle and to form the corresponding UNL relation.

'mazai niir veeTu varai vantatu' (rain water came up to house)

In this sentence the participle 'varai' acts as a postposition, since it succeeds the noun 'veeTu'.

The word 'iruntu'(from) and 'varai' (to) specifies it to be a binary relation 'plf' (starting place) or 'tmf' (starting time) and 'plt' (final place) or 'tmt' (final time) respectively that depends upon the context.

[S]
{unl}
[W]
water(icl>thing.@generic) : 1

rain(icl>thing.@adj.@generic) : 2
come(icl>do.@past.@entry) : 3
house(icl>occur.@generic) : 4
[/W]

[R]
3 agt 1
1 mod 2
3 plt 4
[/R]
{/unl}
[/S]

Here the action ‘vantatu’ is agent of ‘niir’ and these two words are linked with the ‘agt’ relation. The word ‘mazai’ is adjective and it modifies the word ‘niir’. So these words are linked with the relation ‘mod’. The word ‘varai’ is post position and it mentioning a place. So these words are linked with the relation ‘plt’ (final place).
Take the example sentence ‘mazai niRkum varai avan ooTinaan’ (he ran till rain stops)
In this sentence same participle ‘varai’ acts as an adverbial participle since it succeeds the verb ‘niRkum’.

[S]
{unl}
[W]
he(icl>person).@generic) : 1
run(icl>do.@past.@entry) : 2
stop(icl>occur.@generic : 3
rain(icl>thing.@generic :4
[/W]

[R]
2 agt 1
2 tmt 3
3 mod 4
[/R]
{/unl}
[/S]

Here the words ‘avan’ and ‘ooTinaan’ are linked with ‘agt’ relation. The word ‘niRkum’ is adverb and it mentioning time. So these words are linked with the relation ‘tmt’ (final time). The words ‘mazai’ and ‘niRkum” are linked with ‘mod’ relation.

In complex sentences sentence components or clauses can function as adjectival or adverbial or noun components. One of the main functions of the parser is to group the words and detect the functions of the clauses in order to form the appropriate UNL equivalent.
Take the example sentence

‘mazai vanta pootu avaL paaTinaaL’ (when the rain came she sang)
In this sentence the word ‘mazai vanta pootu’ is an adverbial phrase, because of it modifies the verb ‘paTu’

[S]
{unl}
[W]
she(icl>person.@generic) : 0
sing (icl>do.@past.@entry) : 1
come(icl>do.@adv.@generic) : 2
rain(icl>thing.@generic):3
[/W]

[R]
1 agt 0
1 tim 2
2 mod 3
[/R]
{/unl}
[/S]

Take the example sentence
‘neeTRu paaTiya manitan oorukku cenRaan’ (the man who sang yesterday went to village)
The UNL format of the above sentence is as follows.

[S]
{unl}
[W]
[W]
man(icl>person.@generic) : 0
go (icl>do.@past.@entry) : 1
village (icl >place.@generic) : 2
yesterday(icl>time.@generic) : 3
sing(icl>adj.@generic) : 4

[/W]

[R]
1 agt 0
1 plc 2
0 mod 4
4 tim 3
[/R]

{/unl}
[/S]

This like for all the binary relation, the UNL Knowledge Base (KB) stores all the information about the relation.

Endings in Tamil words are used to find out most of the binary relation. Endings with syntactic functional grouping and semantic information are used for identify the correct binary relation. The following table shows some binary relation and its definition and associated endings in Tamil words.

Binary Relation	**Definition**	**Endings in Tamil**
aoj	defines a thing which is in a state or has an attribute	" aaka "
ben	defines not directly related beneficiary or victims of an state or event	" ukkaaka " " ukku "
cag	defines a thing not in focus which initiates an implicit event which is done in parellel	" uTan" and " ooTu "
con	defines a non focused event or state which conditioned a focus event or state	" aal "
coo	defines a co occurred event or state for a focused event or state	" konTu "
dur	defines a period of time during an event occur or a state exist	" pootu "
fmt	defines a range between two things	" iliruntu " " mutal " " varai "
frm	defines a origin of a thing	" iliruntu "
gol	defines the final state of object or the thing finally associated with object of an event.	" ukku "
ins	defines the instrumental to carry out an event	" aal " " uTan "
met	defines the means to carry out an event	" aal " " upayookittu "
mod	defines a thing which restrict a focused thing	" aaka "
opl	defines a place in focus where an event affects	" il " " miitu "
plc	place an event occurs or a state is true or a thing exist	" il "
plf	defines a place an event begins or a state becomes true	" iliruntu " " mutal "
pof	defines of concept of which a focussed thing is a part	" in " " uTaiya "
pos	defines a possessor of a thing	" uTaiya "

ptn	defines non focused initiator of an action	" uTan "
pur	purpose of an event	" ukkaaka " " kku "
qua	defines quantity of a thing or unit	" niRaiya", atikamaana , “kuRaivaana "
tmf	defines a time of an event that starts	" iliruntu " " mutal "
tmt	defines a time of an event that ends	" varai "
to	defines a destination of a thing	" kku"
via	defines intermediate place or state of an event	" vaziyaaka" , "vaziyee"

7. Ambiguity Handling

Generally there are two ambiguities named as Local ambiguity and Global ambiguity when dealing with natural language sentences. Local ambiguity means that part of a sentence can have more than one interpretation, but not the whole sentence. Global ambiguity means that the whole sentence can have more than one interpretation. Local ambiguity can sometimes be resolved by syntactic analysis. In a syntactic ambiguity, a sentence can have more than one parse interpretation. Each parse could have different meanings.

In some languages the agent of the action is determined by its position in the sentence. In Tamil the agent of the main action is determined by its person, number and gender agreement with the main verb. Thus the way to determine the agent of the verb in case of ambiguity,

‘NaaTakam avarkaL paarttaarkaL’

Here the two nouns (‘NaaTakam’ and ‘avarkaL’) do not have case attachment. Hence there is an ambiguity in determining the agent. Here the information obtained from the morphological analyzer for

‘avarkaL’ → ‘avar’ (root) + ‘kaL’ (plural marker)
‘paarttaarkaL’ → ‘paar’ (root) + ‘tt’ (past tense marker) + ‘aarkaL’ (plural marker)

Shows that two words have person, number and gender agreement. Hence the word ‘avarkaL’ is determined to be the agent. Thus the morphological information enables the disambiguation of agent case.

There are some sentences in Tamil where case and number do not indicate the thematic case. Take the example ‘avanukku raaman koTuttaan’. Here ‘raman’ is the agent and ‘avan’ + ‘kku’ is the recipient. The dative case marker indicates recipient. A sentence with a similar syntactic structure,

Example 2

‘avanukku pazam piTikkum’. Here ‘avan’ is the agent and ‘pazam’ is object. In example 2 ‘piTikkum’ is a special type of mental verb where dative case marker indicates

agent. Here knowledge about the semantics of the verb is necessary for creating the binary relation of UNL.

8. Conclusion

In this paper the development of EnConverter for Tamil to UNL network is described. With this UNL format, any user can see that contents in their own native language with the help of UNL DeConverter. In Tamil most information required for conversion to UNL is obtained from morphological level and from the syntactic function grouping at the syntactic level. Only in case of ambiguity is there need to go for complete semantic processing. The semantic component can be interpreted as part of the parser to solve these ambiguities. The conversion to UNL enables conversions to other languages for which deconverts from UNL to the corresponding language are available. The use of UNL as the intermediate representation makes translation of Tamil available worldwide using a standardized format.

References

[1] Munpyo Hong & Oliver Streiter (1999): *Overcoming the Language Barriers in the Web: The UNL-Approach*, 11. Jahrestagung der Gesellschaft für Linguistische Datenverarbeitung (GLDV'99), Frankfurt am Main, Deutschland.

[2] Hiroshi Uchida, Meiying Zhu, Tarcisio Della Senta (1999) : *A gift for a Millennium.* The United Nations University

[3] UNL Center (2000) : *The Universal Networking Language (UNL) Specification,* UNDL Foundation

[4] UNL Center (2000) : *Enconverter Specification*, UNDL Foundation

[5] Serrasset, G. and Boitet, C. (2000). *On UNL as the Future "html of the linguistic content" & the Reuse of Existing NLP Components in UNL-related Applications with the Example of a UNL-French Deconverter.* Proceedings of the 18 th International Conference on Computational Linguistics, pp. 76□□□□□771.

[6] Bouguslavsky, I., Frid, N. and Iomdin, L. (2000). *Creating a Universal Networking Module within an Advanced NLP System*. Proceedings of the 18 International Conference on Computational Linguistics, pp. 83-89.

[7] John Hutchins (1993). *Latest Developments in Machine Translation Technology: Beginning a New Era in MT research*, University of East Anglia, Norwich, England.

[8] Arnold, D. et al. (1994) *Machine translation: an introductory guide.* Manchester/Oxford: NCC/Blackwell

[9] Shachi Dave, Jignashu Parikh and Pushpak Bhattacharyya, *Interlingua Based English Hindi Machine Translation and Language Divergence* , to appear in Journal of Machine Translation, vol 17, 2002.

[10] Pushpak Bhattacharyya, *Many Languages on the Net*, PCQuest Magazine, September 2002.

[11] P. Bhattacharyya *Knowledge Extraction into Universal Networking Language Expressions*, in Universal Networking Language Workshop, Geneva, Switzerland, January, 2001.