Hindi Generation from Interlingua (UNL)

Smriti Singh, Mrugank Dalal, Vishal Vachhani, Pushpak Bhattacharyya, Om P. Damani

Indian Institute of Technology, Bombay (India)
{smriti, mrugank1711, vachhani_vishal, pb, damani}@cse.iitb.ac.in

Abstract

This paper reports our work on generating Hindi sentences from an interlingua representation called Universal Networking Language (UNL). UNL represents knowledge in semantic net like graphs which contain disambiguated words, binary semantic relations, and speech act like attributes associated with the words. Assisted by a semantically rich lexicon, a priority-matrix of syntax plan, and elaborate morphology synthesis rules, we produce fluent Hindi sentences which also meet the adequacy requirement with respect to the reference sentences, and the faithfulness requirement with respect to the semantic graphs. The system has been tested on an agricultural corpora, and the system generated sentences were scored by a team of evaluators. The *BLEU* scores against the reference sentences have been computed. The results show that our system is able to generate slightly flawed but easy to understand sentences that convey most of the meaning. We observe strong correlation between the fluency scores and the *BLEU* scores, as well as between fluency and the adequacy scores. Since fluency evaluation does not require reference translation, this correlation facilitates large scale evaluation of our system without translating large number of UNL sentences. This system is a step towards machine translation involving Hindi as the target language. Our approach is also adoptable to the generation of other languages, in particular Indian languages.

Keywords: Interlingua, UNL, Deconversion, Syntax Planning, Morphotactics, *BLEU* Scores, Generation, Fluency, Adequacy

1. Introduction

Generation of natural language from a machine processable, precise knowledge representation has to grapple with the problem of redundancy and impreciseness inherent in any natural language. An additional challenge is the requirement of keeping the generated language natural and native speaker acceptable. In this paper, we present *HinD*- a Hindi Deconverter (*i.e.,* generator) from Universal Networking Language (UNL), which is an Interlingua for knowledge representation in the context of machine translation. We exploit the common features of many Indian languages to generate acceptable sentences.

Contributions of this paper are the following:

1) We present the design and implementation of a Hindi Deconverter. Our thrust is on the simplicity of specification while maintaining the fluency of the generated sentences.

2) We observe strong correlation between the fluency and the BLEU scores, as well as between fluency and adequacy scores. Since fluency evaluation does not require reference translations, this correlation facilitates large scale evaluation of generation systems without translating large number of UNL sentences.

2. Universal Networking Language (UNL): The Framework

UNL is an electronic language for computers to express and exchange information (Uchida *et. al.,* 1999). The three building blocks of UNL are (i) **Semantic Relations,** (ii) **Attributes** and (iii) **Universal Words.** The UNL representation of a sentence is expressed in the form of a semantic net called ***UNL graph***. Consider sentence (1).

(1) John ate rice with a spoon.

The UNL expression for (1) is given below:

(2) [UNL:1]
agt(eat(icl>do).@entry.@past, John(iof>person))
obj(eat(icl>do).@entry.@past, rice(icl>food))
ins(eat(icl>do).@entry.@past, spoon(icl>artifact))
[\UNL]

In this expression, *agt* (agent), *obj* (object) and *ins* (instrument) are the **semantic relations**. The relatas *eat(icl>do), John(iof >person), rice (icl>food),* and *spoon (icl>artifact)* are the ***Universal Words*** (***UW***). These are language words with *restrictions* mentioned in parentheses for the purpose of denoting a unique sense. *icl* stands for *inclusion* and *iof* stands for *instance of.* UWs can be annotated with **attributes** like *number*, *tense, etc.*, which provide further information about how the concept is being used in the specific sentence. Of special significance is the @*entry* attribute, typically attached to the main predicate.

2.1 UNL Scopes: Representing Embeddings

UNL represents coherent sentence parts (like clauses and phrases) through *Compound UWs* also called *scope nodes.* These scope nodes are like graphs within graphs. These sub graphs have their own *environment* and the @*entry* node. For example, the UNL expression for sentence (3) is given in (4) and the graph illustrating the UNL relations is given in Figure 1.

(3) For this, you contact the farmers of Manchar region or of Khatav taluka.

(4) [UNL]
obj(contact(icl>communicate(agt>person,obj>person)):0W.@imperative.@entry,farmer(icl>creator):1T.@pl.@def)
pur(contact(icl>communicate(agt>person,obj>person)):0W.@imperative.@entry,this:04)
agt(contact(icl>communicate(agt>person,obj>person)):0W.@imperative.@entry,you(icl>persons):0J)
plc(farmer(icl>creator):1T.@pl.@def,:01)

```
or:01(region(icl>location):38.@entry,   taluka(icl>geographical
area):4A)
nam:01(region(icl>location):38.@entry,
            Manchar(icl>geographical place):2R)
nam:01(taluka(icl>geographical area):4A,
            Khatav(icl>geographical area):3U)
 [\UNL]
```

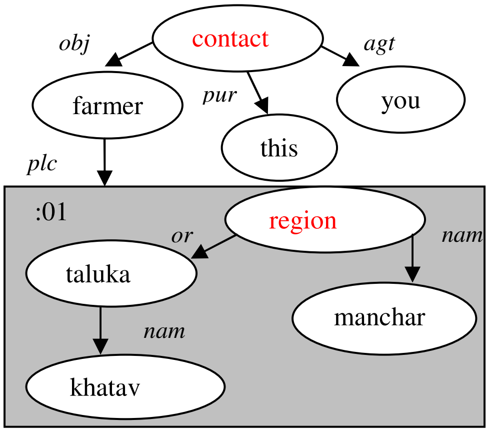

Figure 1: The UNL Graph for UNL Expression 4

The phrase ‘Manchar region or of Khatav taluka’ is considered as being within a scope. Note that the scope is given a compound UW ID:01 to denote a separate environment of knowledge representation.

UNL relations help representing the argument frame of the sentence and also draw a distinction between the argument and the non-argument links of a predicate. The information for number, tense, aspect, mood, negation, *etc.*, are represented using UNL attributes while gender and language specific morphological attributes like-vowel ending of nouns, adjectives, verbs, *etc.,* are stored in the UNL-Target language dictionary.

3. Why UNL?

Contrasted to the more popular *transfer* approach (Hutchins and Somers 1992), the Interlingua approach admits of parallel development of various knowledge resources for analyzing source language sentences and generating target language sentences. Being at the top of the Vauquois Triangle (Hutchins and Somers 1992), elaborate knowledge bases and tools are needed for morphological, syntactic, and semantic processing, both for analysis and generation.

The UNL representation has the right level of expressive power and granularity. UNL has 45 semantic relations and 87 attributes (which can be augmented with the user defined ones) to express the semantic content of a sentence.

In 1992, Interlingua KANT (Nyberg and Mitamura 1992) was designed for large scale MT of technical documentation. However, KANT is a sublanguage system, and handles only *constrained technical English*. Many phenomena are left out of consideration, which are handled by UNL. UNITRAN- the Interlingua and the eponymous MT- is too detailed a framework for meaningful practical implementation (Dorr 1993). ULTRA (Farwel and Wilks, 1991) uses Prolog based grammar for the intermediate representation, and is necessarily restricted in its scope for handling language phenomena.

UNL has been influenced by a number of linguistics-heavy Interlingua based Japanese MT systems in the 1980s- notably the ATLAS-II system [Uchida 1989]. However, the presence of researchers from Indo-Iranian, Germanic and Baltic-Slavic language families in the committee for UNL specifications (UNL Specifications 2005, www.undl.org) since 2000, has lent UNL a much more universal character compared to the interlingua used in ATLAS-II.

Comparing and contrasting UNL with primitive based interlingua like Conceptual Dependency (Schank 1972) and Conceptual Structures (Sowa 2000), we observe that like UNITRAN, they too are too detailed to admit of practical implementations.

4. Language Generation

Though traditionally, language analysis has held sway over language generation- as it involves various disambiguation tasks- early 90s saw the reemergence of Natural Language Generation (NLG) problem, mainly because of the fluency and adequacy requirement in the output produced (Reiter and Dale 2000). Add to it the need for discourse preservation, and the task becomes a real challenge.

NLG research in recent times is witnessing a flurry of activities in Dialogue Systems in which the generation component addresses the problems of sentential fluency, text planning and discourse coherence (SIGGEN conferences 2003-06). We, however, have concentrated on single sentence generation. The reasons for traversing a trodden path are- (i) the gradual re-emergence of knowledge based machine translation that needs generating target language output from an interlingua (ii) the viability of interlingua based MT for Indian languages which number many, but are closely knit in terms of kinship relations and finally (iii) the absence of a generalized framework for Indian languages generation from semantic representation.

Several UNL Deconversion (NLG) systems (Dhanbalan T. and Geetha T. 2003; Daoud D. 2005), including an earlier effort by us, used the universal deconverter tool Deco, provided by the UNL foundation (www.undl.org). Similar to experiences reported by Manati project (Pelizzoni J. and Nunes M. 2005); we too were unsatisfied with Deco. The source code for Deco is not available and its rule-format is abstruse, since it aims to be Turing complete. Manati, while being simpler than Deco, is still a complex framework since it also is a universal deconverter. In contrast, our design is considerably simpler since our scope is a subset of Indian languages only and we aim to exploit their common features.

The Chinese Deconverter reported in (Shi and Chen 2005) makes assumptions stronger than our system (discussed in Section 6.5), and mentions that for Chinese, they only have to deal with case marker insertion, but not with morphology generation in general. The French Deconverter reported in (Blanc E. 2005) also converts the graph to the tree and feeds the tree to an existing transfer program.

5. Stages in the Generation Process

The generation process consists of three main stages- morphological generation of lexical words, function words insertion, and Syntax Planning. For example, in order to translate the sentence (1) into Hindi, a machine has to generate the form *'khaayaa'* (ate) from *'khaa'* (eat) using the information for tense (past), number (singular), and gender (masculine) associated with '*khaa*'. The case markers *'ne' and 'se' also* need to be inserted after the subject 'John' and the object 'rice' respectively. All the words can finally be arranged to construct a valid sentence in Hindi- '*jaun ne cammac se caawal khaayaa',* for (1).

5.1 Morphological Generation of Lexical Words

5.1.1 Noun

Hindi nouns inflect for number and case, and can be described as having major categories of the forms based on the oppositions **direct-oblique** and **singular-plural**. They can be categorized into masculine and feminine gender in terms of their agreement with adjectives and verbs. In UNL, plural nouns are represented using the attribute @*pl,* and singular ones remain unspecified (absence of @*pl* refers to a singular noun). Direct or oblique case is identified using the relation a noun has with a verb or with another noun in a sentence (typically the genitive case). Gender and vowel endings are stored in the UNL-Hindi dictionary. The morphological rules based on word paradigms generate a noun form using all this information, *viz.*, lexical, relational, and UNL attributes. A noun that carries an attribute *NOTCH* (not changeable form) in its dictionary entry remains unchanged, and does not inflect for number or case.

5.1.2 Adjective

Like nouns, adjectives in Hindi also inflect for case, number, and gender, and exhibit concordance with their head nouns (few adjectives, *e.g.*, *sundar* (beautiful), *bhaarii* (heavy) do not inflect to agree with their head nouns). Their heads are identified using relation labels. A form in agreement with the head noun is generated using morphological rules.

5.1.3 Verb

Hindi verbs inflect based on *GNPTAM* information, voice, and vowel ending. Inflections are marked either on the main verb or on its auxiliaries that appear as free morphemes. The information for *number, tense, aspect, mood, negation, etc.*, is represented using the UNL attributes like @*pl,* @*present,* @*past,* @*possible,* @*must, etc.,* while vowel ending is stored in the UNL-Target language dictionary. A verb takes passive morphology if the noun it is related to has the attribute- @*TOPIC* in its UW. Gender information of the noun a verb agrees with is gathered using the UNL relation which dictates whether the situation is *subject controlled (kartrari prayoga)* or *object controlled (karmaNi prayoga).*

Agreement with noun

Hindi verbs always agree with their nominative subjects or with the object, in case the subject is oblique. They take the default form- *singular, masculine* when all nouns are oblique. In order to generate a verb form that is in concordance with the unmarked noun (subject or object), the noun's gender and number values are passed on to the verb's list of attributes. Rest of the information, *i.e.*, for *tense, aspect, mood, vowel ending, etc.*, is provided either by UNL attributes or by UNL-Hindi dictionary. Morphological rules generate morphemes (verbal inflection as well as auxiliaries) for a verb that correspond to the value of these attributes. For example, a verb with UNL attributes- @*present* and @*progress*, the dictionary attribute for vowel ending @*VA*, and with the attributes *F* (feminine) and @*pl* of the noun it agrees with, will be generated as- *khel rahii hain* (are playing-feminine).

Non-finite verbs that do not inflect for tense are of three kinds- *gerunds, participles and infinitives*.

Gerunds are nominal verbs that take the position of nouns but retain their verbal traits like- taking an object or adverbial qualifiers. A verb is identified as a gerund if in a UNL expression it has the attribute @*progress,* and it appears as a child of the *aoj* relation with a noun or of the *obj* relation with a verb. Gerund forms are generated by attaching ***–naa*** suffix to a verbal root.

Verb participles act as verbal adjectives or verbal adverbs in a sentence. In Hindi, verbal adjectives are formed by using *–taa huaa* to denote progressive aspect, *e.g.*, *ug**taa huaa** sooraj* (rising sun) and *–aa/yaa huaa* to denote perfective aspect, *e.g.*, *thak**aa huaa** aadmii* (tired man). Verbal adverbs are formed by attaching *–kar* or *–te huye* to verb root, *e.g.*, *'khaa**kar** aayaa'* (came after eating) and *'khaa**te huye** aayaa'* (came eating). In UNL, verbal adjectives can be identified if the verb has the attributes @*progress* or @*complete* and also appears as a child in the *mod* (modifier of) relation with a noun. Likewise, a verbal adverb appears as child in a relation with another verb. Infinitives are identified as those verbs which do not have @*progress* or @*complete* and always appear as child in an *obj* relation with another verb. Infinitives are generated by attaching ***–naa*** suffix to a verbal root.

Conjunct verb

Expressing a single word concept in one language may require two or more words in another language. Many verbs in English can only be translated into Hindi by

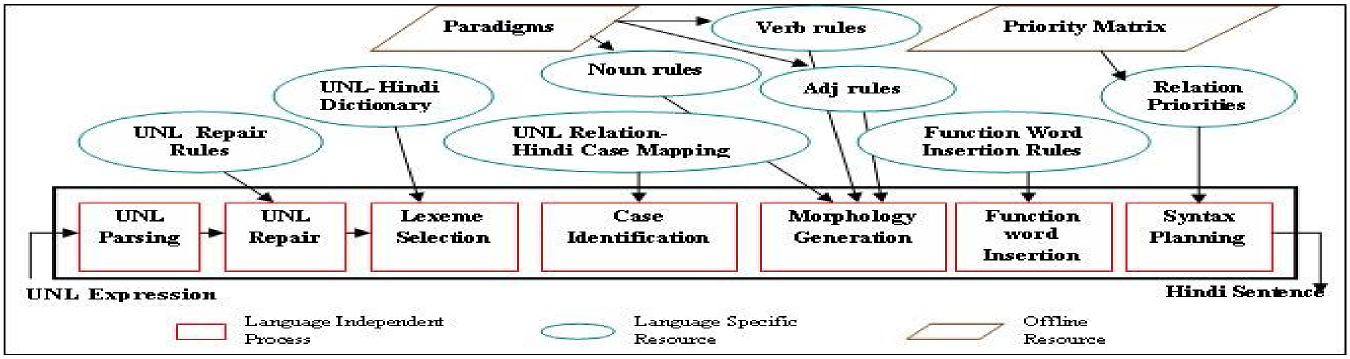

Figure 2: The Architecture of the Deconversion System

using a noun-verb or an adjective-verb sequence (Chakrabarti D. 2006). Such verbs are called conjunct verbs. The UW translations of these verbs are stored in the dictionary as a noun-verb or as an adjective-verb sequence. The morphological attributes of these verbs remain the same as other verbs. All inflections are marked only on the verb, and the noun or the adjective in the sequence remains uninflected. Many of these verbs are formed by adding nouns or adjectives to the verbs- *kar* (do) (*e.g.*, *shaadi kar* (marry), *snaan kar* (bathe) *etc.*) or *ho* (be) (*e.g. samaapt ho* (finish), *laagu ho* (promulgate) *etc.*) Such verbs carry additional attributes- @*link* and @*lnk* respectively in their dictionary entries.

5.2 Function word insertion

UNL encodes case information by using relation labels assigned as per the properties of the connected nodes. Consider, for example, the translation of sentence (1).

जॉन ने चम्मच से चावल खाया।

jaun ne cammac se caawal khaayaa--- **(7)**

Here, the case markers *ne* and *se* are inserted to derive the relation *jaun* and *cammac* have with the verb '*eat*'. Given a node along with all its lexical attributes from the UNL-Hindi dictionary, an appropriate case marker is inserted. Similarly, other function words like- conjunctions, disjunctions, particles, *etc.*, are also inserted to represent clausal information.

5.3 Syntax Planning

Syntax Planning is the process of linearizing the Semantic hyper-graph. The use of overt case-markers makes the word-order in these languages flexible. But, some orders are considered more natural than others, and hence, we assign relative positions to various words based on the relations they share with the head-word in a clause.

6. System Architecture

The previous section described the linguistic foundations of our Deconverter *HinD*. This section concentrates on the architecture of *HinD*, shown in Figure 2.

6.1 UNL Parsing and Graph Repair

The input UNL expression is parsed into a graph-structure. Based on our error analysis, we observed that some errors are common in the input UNL expressions (discussed in Section 6.7), in particular related to *Scopes*. Currently, we handle these errors using some heuristic rules for graph repair. For example, any two nodes having 'cnt' (content) relation are put into a *Scope*.

6.2 Lexeme Selection

Each UW along with its restrictions is looked up in the language specific dictionary, and the corresponding lexeme is obtained. Fortunately, the Deconverter does not have to deal with the WSD problem. It is handled during source language to UNL enconversion by associating restrictions with a UW to uniquely represent a sense. For example, the following two UWs entries correspond to two different senses of the word '*water*':

[paanii]{}"water(icl>liquid)"(N,INANI,OBJCT,PHSCL,FRM,LQD,M,NOTCH,UNCNT,NI)
[paani de]{}"water(icl>do(agt>person,obj>thing))" (V,VOA, VLTN,CJNCT,N-V,Ve)

The entries in parentheses are morpho-syntactic and semantic attributes of *Hindi* words which control various generation decisions like choosing specific case markers.

6.3 Case Identification and Morphological Generation

As discussed earlier, Hindi morphology is decided by *GNPTAM* and e*nding vowels*. We next show some sample rules for noun morphology generation in Table 1.

Suffix	Attribute values
uoM	*@N,@NU,@M,@pl,@oblique*
U	*@N,@NU,@M,@sg,@oblique*
I	*@N,@NI,@F,@sg,@oblique*
iyoM	*@N,@NI,@F,@pl,@oblique*
oM	*@N,@NA,@NOTCH,@F,@pl,@oblique*

Table 1: Sample Noun Morphology Rules

Noun inflections are handled using attribute values mainly for gender, number, case, and vowel ending. Inflections are added to a word stem to generate a desired form. For example, an *'U'* ending masculine noun- '*aaluu*' (potato)- which is stored as '*aal-*' in the dictionary along with the attributes like *N, NU, M,* and also has UNL attributes *@pl* and *@oblique*- will match the first rule of the sample rules given above, and will be outputted as '*aaluoM*'.

Suffix	*Tense*	*Aspect*	*Mood*	*N*	*Gen*	*P*	*V E*
-e rahaa thaa	@past	@progress	-	@sg	@male	3rd	e
-taa hai	@present	@custom	-	@sg	@male	3rd	-
-iyaa thaa	@past	@complete	-	@sg	@male	3rd	I
saktii hain	@present	-	@ability	@pl	@female	3rd	A

Table 2: Sample Verb morphology rules

Verbs, as mentioned previously, inflect for GNPTAM, vowel ending and voice. A few rules for verb morphology generation are given in Table 2. For example, the first rule in the table is read as- attach *-e rahaa thaa* to a verb root (*e.g., 'de' and 'le' which are stored as 'd-' and 'l-' in the UNL-Hindi dictionary*) which has the attributes- *@past* for tense, *@progress* for aspect, mood unspecified, shows agreement with a singular (@sg), masculine (@male), 3rd person noun, and ends with the vowel *'e'*. The forms generated using this rule would be '*de rahaa thaa'* (was giving) or '*le rahaa thaa*' (was taking).

6.4 Function Word Insertion

Having inflected the words as per morphological rules, function words like case markers, conjunctions, relative pronouns *etc.*, need to be inserted. The rules for inserting function words depend on UNL relations and the restrictions specified with the parent and child nodes. A rule has the following five components:

1. Relation name
2. Necessary Conditions for Parent node
3. Negative Conditions that should not be present at Parent node
4. Necessary Conditions for Child node
5. Negative Conditions that should not be present at Child node

Based on these components, a decision is made about inserting a function word before or after parent and child nodes.

Consider sentence (1) and its UNL again. The Case marker rule applicable for this sentence is:

agt : @past#V : VINT : N : null => null : null : null : ने

This rule says that in the '***agt***' relation, if the parent UW is a verb with *@past* attribute, and is not an intransitive verb, and if the child UW is a noun, insert the case marker '**ने**' after the child UW, *e.g., after John* in Sentence 7.

Similarly, the rule for inserting the conjunction लेकिन(*but)* is: *and:null:null:@contrast:null=>null*:लेकिन*:null:null*

6.5 Syntax Planning

Syntax Planning is the process of linearizing the Semantic hyper-graph, *i.e.*, it decides the word-order in the generated sentence. To make this process rule driven, we make several important assumptions:

Semantic Independence: The relative word order of a UNL relation's relata does not depend on the semantic properties of the relata.

Context Independence: The relative word order of a relation's relata does not depend on the rest of the expression.

Local Ordering: The relative word order of various relations sharing a relata does not depend on the rest of the expression.

Note that the last two assumptions are weak in that, in theory, they help us avoid making the strong Compositionality assumption [Shi X. and Chen Y. 2005], which states that the sentence for a whole tree can be composed from the sentences of its sub trees. Say, a tree is of the form A->B->C. Then, the compositionality assumption states that A can only be either at the beginning or at the end of the generated sentence. Whereas, *HinD* allows A to occur in between B and C.

In practice, we found that whenever Compositionality assumption is violated, it is due to the improper use of *Scope*, *i.e.*, if our system generates BAC then A->B should have been a *Scope* in the first place. However, given that imprecise UNLs are a fact of life, it is important that our system should be able to handle them.

Based on these assumptions, we break down the graph linearization problem into following subcomponents:

- For a given node, decide whether each of its untraversed parents (there can be multiple parents) and children nodes should be ordered before or after the current node.
- For nodes in each of the 'before' and 'after' group, decide their relative orderings.

Both of these ordering decisions are done based on the UNL relation between the node under consideration, and the parent or the child node.

6.5.1 Parent-Child Positioning

For each UNL relation, a rule-file states whether the parent should be ordered before or after the child. Currently, '*aoj*', *'seq'*, *'and'*, 'or', *'fmt'*, and '*cnt'* relations place the parent first, and the rest of the relations place the child first.

6.5.2 Prioritizing the Relations

In our system, a Priority-Matrix describes the Left-or-Right relative position of two UNL relations when they have a common relata. Consider Sentence 1 and its Hindi translation- Sentence 9. In English, the order of the arguments in the sentence is agent-object-instrument. On the other hand, the default order for its Hindi equivalent is agent-instrument-object. Table 3 (*L: towards left, R: towards right*) shows a subsection of the Priority-Matrix for Hindi.

Treating this matrix as an adjacency list representation of a directed graph, where **L** (**R**) indicates incoming (outgoing) edge, graph vertices are topologically sorted. The sorted output is ranked in descending order, *i.e.*, the relation that should appear leftmost gets the highest rank. In case a cycle is found in the graph during sorting, the user is requested to break the cycle.

	Agt	Aoj	Obj	Ins
Agt		**L**	**L**	**L**
Aoj			**L**	**L**
Obj				**R**
Ins				

Table 3: A subsection of the Priority Matrix

6.5.3 Syntax Planning Algorithm

The following algorithm does Syntax Planning by using the Parent-Child Positioning rules and the Relation Priorities.

```
Initialization: Mark the Entry node and put it on Stack.
Begin-Algo
While Stack is non-empty:
1. Pop the top node from the Stack and make it Current.
2. If the current node has unmarked relata
2.1. Divide the unmarked relata of the Current node in 'Before-Current' and 'After-Current' groups based on the Parent-Child Positioning Rules, and mark all of them.
2.2. Sort each group in ascending order based on their ranks in the topological sort output.
2.3 Push them on the stack in sorted 'After-Current', Current, sorted 'Before-Current' order.
3. If the Current node has no unmarked relata:
3.1 If the Current node is a Scope node, then recurse.
3.2 Else, output the Current node.
End-Algo
```

Table 4 shows a step-through algorithm for the UNL shown in Sentence 4 (corresponding to English Sentence 3). Step number X-Y.Z means iteration X, algorithm step Y.Z. Only some of the steps and some of the variables are shown. Note that for '*or*' relation, the parent is placed before the child and for all other relations, the child is placed first.

6.6 Language Specific and Language Independent Components

As described so far, all components of HinD use *language independent algorithms* with *language dependent data*. UNL parsing and lexeme selection are algorithmic processes independent of language. The Syntax Planning component can be applied to any language by just adopting the priority matrix for the specific language. Case marker generation and morph-synthesis too are, *engines* that make use of Hindi specific *configuration files*.

6.7 Limitations of Generating from UNL

Unlike Deco and Manati, simplicity is one of the explicit aims of *HinD*, even at the expense of some Fluency. That is, given that the hard part of analyzing a sentence is already done during the *enconversion* process, we hope that a user for a given Indian language should be able to use *HinD* by writing some simple rule files without having to worry about complicated interaction between word-forms, semantic relations, and Syntax Planning. In practice, we face several obstacles in generating high quality sentences from such a simple scheme:

a) UNL Expressiveness: In certain situations, UNL has limited expressive power. This issue is discussed in detail in (Boguslavsky I. 2005). Here we give just one example: '*aoj*' relation is used both for attributive and predicative adjectives. Hence, the same UNL expression can give rise to 'red leaf' as well as 'leaf is red'.

Step	**State**
1-1	**Stack = {},Current = contact, Output= {}**
1-2.1	**Before-Current = {farmer,this,you}**
1-2.2	**Sorted-Before-Current = {farmer,you,this}**
1-2.3	**Stack = {contact,farmer,you,this}**
2-1	**Stack = {contact,farmer,you},Current={this}**
2-5.2	**Stack = {contact,farmer,you},Output={this}**
3-5.2	**Stack = {contact,farmer},Output={this,you}**
4-2.3	**Stack = {contact,farmer,:01}**
5-3.1	**Stack = {contact,farmer},Recurse{:01}**
6-1	**Stack = {contact,farmer,region}**
6-2.1	**Before-Current = {manchar}, After-Current = {taluka}**
6-2.3	**Stack = {contact,farmer,taluka,region,manchar}**
7-1	**Current = {manchar}**
7-5.2	**Stack = {contact,farmer,taluka,region}, Output = {this,you,manchar}**
8-5.2	**Stack = {contact,farmer,taluka }, Output = {this,you,Manchar,Region }**
9-2.3	**Stack = {contact,farmer,taluka,khatav}**
10-5.2	**Stack = {contact,farmer,taluka }, Output = {this,you,manchar,region ,khatav}**
13-5.2	**Stack={},Output={this,you, manchar,region,khatav,taluka, farmer,contact}**

Table 4*: An example of Syntax Planning*

b) Imprecise UNL Expressions: Whether manual or automatic, semantic graph creation from a natural language sentence is an error-prone process. We find that many a times, scopes are not handled properly, or some relations are confused with each other, say '*obj*' and '*plc*'.

c) Syntax Planning Assumptions: To keep the system simple, *HinD* makes several assumptions, discussed in Section 6.5. For example, in case of 'X seq Y', HinD always generates 'X before Y' and never 'Y after X'.

d) Word Properties: HinD is guided by UNL relations and the attributes associated with UWs. Sometimes, two semantically similar Hindi words show different morpho-syntactic behavior. For example, *subah* (morning) and *raat* (night) can be substituted for *shaam* in - *vah shaam ko aayaa* (He came in the evening). It is only *subah* that does not take the case marker *ko* while others do. HinD does not handle this properly. Similarly we generate को in the example in Table 5 instead of से because we do not consider all the properties of the Hindi word संपर्क (contact).

This concludes our discussion of the Generation system. Table 5 shows an example illustrating various stages of the generation (* in the table shows a stem on which a suffix is to be attached).

Module	Output
Original English Sentence	For this, you contact the farmers of Manchar region or of Khatav taluka
UNL Expression	See Sentence 4 and Figure 1
Lexeme Selection	संपर्क किस□न् यह आप क्षेत्र् त□लुक् मंचर खट□व contact farmer this you region taluka manchar khatav
Case Identification	संपर्क किस□न्* यह आप क्षेत्र्* त□लुक्* मंचर खट□व contact farmer* this you region* taluka* manchar khatav
Morphology Generation	संपर्क कीजिए किस□नों यह आप क्षेत्र contact .@imperative farmer.@pl this you region त□लुके मंचर खट□व taluka manchar Khatav
Function Word Insertion	संपर्क कीजिए किस□नों को इसके लिए आप क्षेत्र contact farmers this for you region य□ त□लुके के मंचर खट□व or taluka of Manchar Khatav
Syntax Planning	इसके लिए आप मंचर क्षेत्र य□ खट□व This for you manchar region or khatav त□लुके के किस□नों को संपर्क कीजिए । taluka of farmers contact

Table 5: An example output at various generation stages

7. Evaluation

The problem being tackled in this work is the generation of NL sentences from semantic graphs which represent *meaning*. What is important is the **faithful** capturing and the rendering of this meaning in the generated sentences. Measuring this faithfulness requires careful comparison of the generated sentences with UNL expressions. However, finding evaluators outside our project, who are native Hindi speakers and also expert in UNL, is a tall task. In any case, this would be highly time-consuming and a subjective process.

Hence, we compromise by generating reference Hindi sentences from original English sentences, and measuring the adequacy of the machine generated sentences with respect to reference Hindi sentences. Assuming that the reference sentences are faithful to the UNL expressions, we indirectly measure the faithfulness of the generated sentences in addition to directly measuring fluency, the 'syntactic quality' of the generation sentence.

7.1 Input Preparation

We evaluated the generation of 901 Hindi sentences from Agricultural domain. These sentences are taken from the script of Question-Answer threads between farmers and Agriculture experts. The original sentences were in Marathi, which were manually translated to English and then to UNL. Single reference Hindi translations were generated from English sentences. *BLEU* scores (Papineni *et al.*., 2002) were computed using single reference translations. Median sentence length was 14 words with a Standard Deviation of 7.5.

7.2 Manual Evaluation Guidelines

We adapt the evaluation guidelines from (LDC 2004) and (Sumita E. *et al.*. 1999). After some trial evaluations with various schemes, and discussions with evaluators, we decided to convert the 5 point scale in (LDC 2004) to a 4 point scale, since too fine-grained a distinction may result in evaluators worrying a lot about making an accurate call, and intuitive judgment may get affected. It also makes the evaluation even more subjective. Our final evaluation guidelines are shown in Figure 3.

Fluency of the given translation is:
(4) Perfect: Good grammar
(3) Fair: Easy-to-understand but flawed grammar
(2)Acceptable: Broken - understandable with effort
(1) Nonsense: Incomprehensible
Adequacy: How much meaning of the reference sentence is conveyed in the translation?
(4) All: No loss of meaning
(3) Most: Most of the meaning is conveyed
(2) Some: Some of the meaning is conveyed
(1) None: Hardly any meaning is conveyed

Figure 3: The Evaluation Guidelines

As per (LDC 2004), the evaluators were asked to provide their intuitive reaction to the output and to work as quickly as comfortable. Adequacy judgments were taken after the fluency judgments, and the judges were asked to look at the reference Hindi translations only after the fluency judgment was over.

	BLEU	Fluency	Adequacy
Geometric Average	0.34	2.54	2.84
Arithmetic Average	0.41	2.71	3.00
Standard Deviation	0.25	0.89	0.89
Correlation *BLEU*	1.00	0.59	0.50
Correlation Fluency	0.59	1.00	0.68

Table 6: Average Scores

7.3 Evaluation Results

All three matrices were computed separately for all 901 sentences. Various statistics are shown in Table 6. From these results we conclude that our system is able to generate slightly flawed but easy to understand sentences that convey most of the meaning.

Our *BLEU* score also seems impressive, until one realizes that our system does not deal with the WSD problem, and the use of *UNL Scope* makes the handling of clauses and phrases easy.

We observe that there is good correlation between Fluency and the BLEU scores, and strong correlation between Fluency and Adequacy scores. The relation between adequacy and fluency is explored further in Figure 4. Figure 4 shows the distribution of Adequacy scores for various values of Fluency.

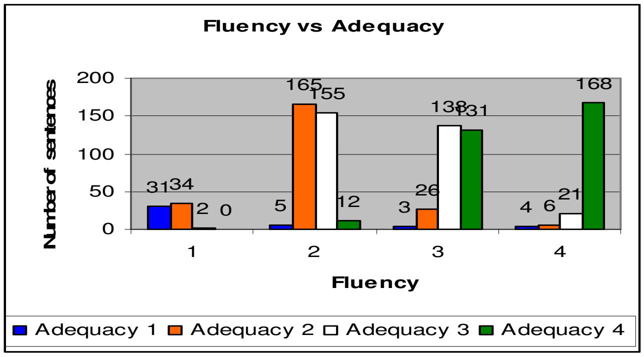

Figure 4: Fluency vs. Adequacy

This implies that we can do large scale evaluation using fluency alone. Given that the generation of reference translations is the bottleneck in very large scale MT evaluations, this finding is significant, and since fluency evaluation does not require generation of any reference translation. Note that our conclusion is applicable only to the deconversion process and not to general MT systems.

7.4 Cautionary Remarks

While our initial results are encouraging, there are several concerns that we need to worry about:

a) Domain Diversity: We have evaluated our system only for agricultural domain, that too, in a very particular setting in Maharashtra, India.

b) Speaker Diversity: Typically, in a Question-Answer thread, the questions are small and the answers are long. All our answers have been generated by a small number of experts, thus losing somewhat on stylistic, topicalization, and emphasis variations.

c) Enconversion Automation: One of the hardest parts in MT is analyzing the source sentences. In our system, this process is semi-automatic with lot of manual intervention, making it non-scalable.

8. Conclusions and Future Work

We reported work on Hindi generation from the UNL graphs with the satisfactory average *BLEU* score of approximately 0.34 which correlates well with the human evaluators' scores. The UNL phenomena have been meticulously handled, relation by relation, and attribute by attribute. The system, thus, is an example of rule-based NL generation. The linguistic concerns have been clearly separated from the computational ones, and so the system promises to be extendable to the generation of other Indian languages too, by simply changing the linguistic knowledge bases.

Future work consists in plugging the system in an Interlingua based MT system with Hindi as the target language. Dialogue- which is the modern trend in NLG- has been left out of concern. This will necessitate investigating discourse phenomena deeply (co-reference, topicalization, *etc.*). One of the main challenges is the naturalness of the output and native speaker acceptability. High fluency score is, thus, of crucial importance.

Acknowledgements

We would like to thank *Salil Badodekar, Gajanan K. Rane* and many others for the vital roles they have played in this project. This work was supported in part by the TCS sponsored project Laboratory for Intelligent Internet Research.

References

(UNL Book 2005*: Universal Network Language: Advances in Theory and Applications. Research on Computing Science)

Blanc E. (2005). About and Around the French Enconverter and the French Deconverter. UNL Book*.

Boguslavsky I. (2005). Some Controversial Issues of UNL: Linguistic Aspects. UNL Book*

Chakrabarti D. *et al.* (2006), Hindi Verb Knowledge Base and Noun Incorporation in Hindi, 3rd Global Wordnet Conference (GWC 06), Jeju Island, Korea, January.

Dhanbalan D. and Geetha T. (2003). UNL. Deconverter for Tamil, Intl. Conf. on the Convergence of Knowledge, Culture, Language and Information Technologies.

Dorr B. 1993, Dorr, B. J. (1992/1993). The use of lexical semantics in interlingual machine translation. Machine Translation, 4/3.

Farwell D. and Wilks Y. (1991). ULTRA, a Multilingual Machine Translator, MT Summit III , Washington, DC, USA

Hutchins W., Somers H. (1992). An Introduction to Machine Translation, Academic Press, New York

LDC. Linguistic Data Consortium (2004). Specifications for human assessment of translation quality.

Leavitt *et al.*. (1993). The KANT Interlingua specification Technical Report CMU-CMT-93-143, Center for Machine Translation, Carnegie Mellon University.

Nirenburg *et al.*. (1992) Nirenburg, S., Carbonell, J., Tomita, M., and Goodman, K. (1992). Machine Translation: A Knowledge-based Approach. Morgan Kaufman, San Mateo, CA.

Nyberg, E. and Mitamura, T. (1992). The KANT system: Fast, accurate, high-quality translation in practical domains. COLING.

Okamura A. *et al.*. (1991). Multilingual Sentence Generation from the PIVOT Interlingua, MT Summit III, USA.

Papineni, K., *et al.*. (2002). *BLEU*: a method for automatic evaluation of machine translation. ACL.

Pelizzoni J., Nunes M.. (2005). Flexibility, Configurability and Optimality in UNL Deconversion via Multiparadigm Programming. UNL Book*.

Reiter E. and Dale R. (2000), Building natural language generation systems, Cambridge University Press, New York.

Schank, R. (1972). Conceptual Dependency: A Theory of Natural Language Understanding. Cognitive Psychology, Vol. 3, No. 4

Schubert K. (1988). The Architecture of DLT- interlingual or double-dialect, in New Directions in Machine Translation, Floris Publications, Holland.

Shi X., Chen Y. (2005). A UNL Deconverter for Chinese. UNL Book*.

Sumita E., Yamada S. *et al.*. (1999). Solutions to Problems Inherent in Spoken-Language Translation: the ATR-MATRIX Approach. In Proc. of MT Summit VII.

Uchida H. (1989). ATLAS. Proc. (MT Summit), Munich.

Uchida, H., Zhu, M. *et al.*. (1999). Universal Networking Language: A gift for a millennium. The United Nations University,Tokyo,Japan.