

Universal Networking Language Based Analysis and Generation for Bengali Case Structure Constructs

Kuntal Dey¹ and Pushpak Bhattacharyya²

¹ Veritas Software, Pune, India.
u2ckuntal@yahoo.com

² Computer Science and Engineering Department
Indian Institute of Technology, Bombay, India.
pb@cse.iitb.ac.in

Abstract. Case structure analysis forms the foundation for any natural language processing task. In this paper we present the computational analysis of the complex case structure of Bengali- a member of the Indo Aryan family of languages- with a view toward interlingua based MT. Bengali is ranked 4th in the list of languages ordered according to the size of the population that speaks the language. Extremely interesting language phenomena involving morphology, case structure, word order and word senses makes the processing of Bengali a worthwhile and challenging proposition. A recently proposed scheme called the *Universal Networking Language* has been used as the interlingua. The approach is adaptable to other members of the vast Indo Aryan language family. The parallel development of both the analyzer and the generator system leads to an insightful intra-system verification process in place. Our approach is *rule based* and makes use of authoritative treatises on Bengali grammar.

1 Introduction

Bengali is spoken by about 189 million people and is ranked 4th in the world in terms of the number of people speaking the language (ref: <http://www.harpercollege.edu/~mhealy/g101ilec/intro/elt/eltclt/top100.html>). Like most languages in the Indo Aryan family, descended from Sanskrit, Bengali has the SOV structure with some typical characteristics. A motivating factor for creating a system for processing Bengali is the possibility of laying the framework for processing many other Indian languages too.

Work on Indian language processing abounds. *Project Anubaad* [1] for machine translation from English to Bengali in the newspaper domain uses the *direct translation approach*. *Angalabharati* [2] system for English Hindi machine translation is based on pattern directed rules for English, which generates a *pseudo-target-language* applicable to a group of Indian Languages. In MATRA [3], a web based MT system for English to Hindi in the newspaper domain, the input text is transformed into case-frame like structures and the the target

language is generated by parameterized templates. The *MANTRA* MT system for official documents uses Tree Adjoining Grammar (TAG) to achieve English Hindi MT (ref: <http://www.cdacindia.com/html/about/success/mantra.asp>). Project *Anusaaraka* [4] is a language accessor system rather than an MT system and addresses multiple Indian languages. Interlingua based MT for English, Hindi and Marathi [5] [6], that uses the UNL, transforms the source text into the *UNL representation* and generates target text from this intermediate representation. References to most of these works can also be found at <http://www.tdil.mit.gov.in/mat/ach-mat.htm>. Other famous MT systems are *Pivot* [7], *Atlas* [8], *Kant* [9], *Aries* [10], *Geta* [11], *SysTran* [12] etc.

The Universal Networking Language (UNL) (<http://www.unl.ias.unu.edu>) has been defined as a digital meta language for describing, summarizing, refining, storing and disseminating information in a machine independent and human language neutral form. The information in a document is represented sentence by sentence. Each sentence is converted into a directed hyper graph having concepts as nodes and relations as arcs. Knowledge within a document is expressed in three dimensions:

1. Word Knowledge is expressed by Universal Words (UWs) which are language independent. These UWs are tagged using restrictions describing the sense of the word in the current context. For example, *drink(icl > liquor)* denotes the noun sense of *drink* restricting the sense to a type of *liquor*. Here, *icl* stands for inclusion and forms an *is-a* relationship like in semantic nets [13].
2. Conceptual Knowledge is captured by relating UWs through a set of UNL relations [14]. For example,

Humans affect the environment

is described in the UNL as

```
agt(affect(icl>do).@present.@entry, human(icl>animal).@pl)
obj(affect(icl>do).@present.@entry, environment(icl>abstract thing).@pl)
```

agt means the *agent* and *obj* the *object*. *affect(icl > do)*, *human(icl > animal)* and *environment(icl > abstract thing)* are the UWs denoting concepts.

3. Speaker's view, aspect, time of event, etc. are captured by UNL attributes. For instance, in the above example, the attribute *@entry* denotes the main predicate of the sentence, *@present* the present tense and *@pl* the plural number.

The above discussion can be summarized using the example below

John, who is the chairman of the company, has arranged a meeting at his residence

The UNL for the sentence is

```

;===== UNL =====
mod(chairman(icl>post).@present.@def,company(icl>institution).@def)
aoj(chairman(icl>post).@present.@def, John(icl>person))
agt(arrange(icl>do).@entry.@present.@complete, John(icl>person))
pos(residence(icl>shelter), John(icl>person))
obj(arrange(icl>do).@entry.@present.@complete, meeting(icl>event).@indef)
plc(arrange(icl>do).@entry.@present.@complete, residence(icl>shelter))
[/S]
;=====

```

In the expressions above, *agt* denotes the *agent* relation, *obj* the *object* relation, *plc* the *place* relation, *pos* is the *possessor* relation, *mod* is the *modifier relation* and *aoj* is the *attribute-of-the-object* (used to express constructs like *A is B*) relation. The detailed specification of the Universal Networking Language can be found at <http://www.unl.ias.unu.edu/unlsys>.

Our work is based on an authoritative treatise on Bengali grammar [15]. The strategies of analysis and generation of linguistic phenomena have been guided by rigorous grammatical principles.

2 EnConverter and DeConverter machines

The EnConverter (henceforth called *EnCo*) [16] is a language-independent parser, a multi-headed Turing machine [17] providing a framework for morphological, syntactic and semantic analysis synchronously using the UW dictionary and analysis rules. The structure of the machine is shown in the figure 1.

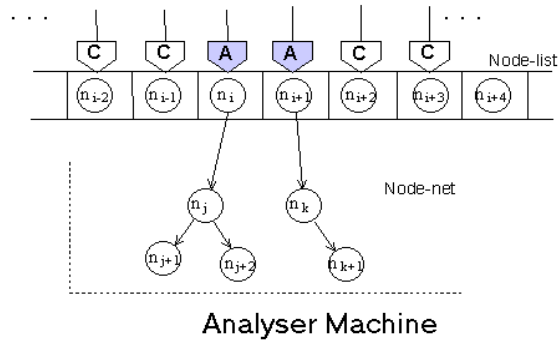


Fig. 1. The EnCo machine

The machine has two types of *heads- processing heads* and *context heads*. The processing heads (2 nos.) are called *Analysis Windows (AW)* and the

context heads are called *Condition Windows (CW)*. The machine traverses the sentence back and forth, retrieves the relevant universal words from the lexicon and, depending on the *attributes* of the nodes under the AWs and those under the surrounding CWs, generates semantic relations between the UWs and/or attaches speech act attributes to them. The final output is a set of UNL expressions equivalent to a UNL graph.

The DeConverter (henceforth called the *DeCo*) [18] is a language-independent generator that produces sentences from UNL graphs (figure 2).

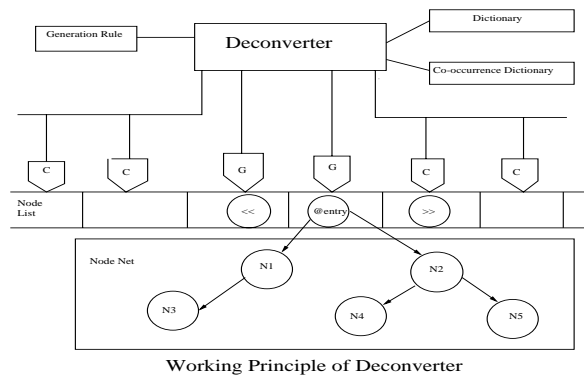


Fig. 2. The DeCo machine

Like EnCo, DeCo too is a multi-headed Turing Machine. It does syntactic and morphological generation synchronously using the lexicon and the set of generation rules.

3 Rule theory

EnCo and DeCo are driven by *analysis rules* and *generation rules* respectively. These rules are *condition-action structures* that can be looked upon as *program* written in a specialized language to process various complex phenomena of a natural language, both for analysis and generation. They have the following format:

```
< TYPE >
[" (" < PRE > ") ["*"]...
" {" || "" "" ["< COND1 >"] ":" ["< ACTION1 >"] ":" ["< RELATION1 >"] ":" ["< ROLE1 >"] } " || "" ""
[" (" < MID > ") ["*"]...
" {" || "" "" ["< COND2 >"] ":" ["< ACTION2 >"] ":" ["< RELATION2 >"] ":" ["< ROLE2 >"] } " || "" ""
[" (" < SUF > ") ["*"]...
"P (" < PRIORITY > ") :
```

Characters between double quotes are the predefined delimiters of the rule. The rules mean that

- **IF**
under the *left processing window* there is a node satisfying <COND1> and under the *right processing window* a node satisfying <COND2> attributes, and there are nodes that fulfill the conditions in <PRE>, <MID> and <SUF> in the order of left, middle and right sides of processing windows respectively,
- THEN**
the lexical attributes in processing windows are rewritten according to the <ACTION1> and <ACTION2> as specified in rule, and new attributes added if necessary. (By *processing window*, *analysis window* is meant for the enconversion process and *generation window* for the deconversion process).
- The operations are done on the node-list depending on the <TYPE> of the rule. <RELATION1> describes the semantic relation of the node on right processing window to the node on left processing window and <RELATION2> describes the reverse [6].
- <PRIORITY> describes the interpretation order of the rules, whose value lies between 0-255. Larger number indicates higher priority. Matching rule with the highest priority is selected for multiple matching rules.

A sequence of such rules get activated depending on the sentence situation (the conditions of the nodes under the analysis/generation windows). These are the lexico-morpho-grammatical-semantic attributes of the words under processing. For example, for a sentence like *John laughs*, the *animate* attribute of *John*, the *verb* attribute of *laugh* and the *adjacency* of these two words under the analysis windows dictate with high probability establishing the *agt (agent)* relation between the corresponding two nodes in the UNL graph.

In order to adapt the UNL engines to enconvert the Bengali sentences into the UNL interlingua and to deconvert the UNL interlingua/graph into Bengali sentences, an enconverter rule-base and a deconverter rule-base have been written. The rules within the rule-base are compliant with the corresponding UNL engines and are focused to deal with the Bengali language structure.

4 Case Structure in Bengali: *Kaaraks*

In the Indian linguistic system- descended from Sanskrit- the *case constructs* are called *kaaraks* [19]. As in the traditional understanding, they denote the relationship of the nominals with the main verb of the clause except in the *genitive case* where two nominals are related to each other. The case structure in Bengali is complex. The *kaaraks* are broadly classified into 6 types [15], each having a finer categorization into sub-types. The correspondence between the Bengali *kaarak* system and the traditional linguistic concept of case [20] is shown by means of table 1. The *Bibhakti signs* are the case markers. An exhaustive

study of the *kaarak* system with a view to analyzing Bengali into UNL has been carried out. The foundation of this work is the *kaarak* theory [15]. Due to the word limitation, we exemplify the work with only the first *kaarak*, viz., the *kartri kaarak*.

Table 1. Case-*kaarak* correspondence

Classical Case	Corresponding Bengali kaarak	Bibhakti signs (Case Marker)
Nominative case	<i>Kartri kaarak</i>	None
Accusative case	<i>Karma kaarak</i>	<i>ke, re, ere</i>
Instrumental case	<i>Karan kaarak</i>	<i>dwaaraa, diye, diya, kartri</i>
Dative case	<i>Sampradaan kaarak</i>	<i>janya, nimitta, ke</i>
Ablative case	<i>Apaadaan kaarak</i>	<i>theke, haite</i>
Genitive case	<i>Sambandha pad</i>	<i>r, er</i>
Case of time and place	<i>Adhikaran kaarak</i>	<i>e, te, ete</i>

4.1 *Kartri kaarak*

Kartri kaarak denotes the *agent* of the action stated by the verb. The *kaarak* is divided into the following classes:

1. **Projojok karta** (প্রয়োজক কর্তা): Here the agent *causes* some event to take place, with an inclination towards compelling the event to happen. The morphology of the verb is exploited and the extracted knowledge has the *causative* feature marked.

Example:

টম জনকে খেলাবে
tama janake khelaabe.
 Tom John-to will-make-play.
 Tom will make John play.

2. **Nirapekkha karta** (নিরপেক্ষ কর্তা): Here there are more than one verb in the sentence with at least one অসমাপিকা (finite) verb and one সমাপিকা (non-finite) verb, and the *kartas*, i.e., *agents* for these verbs are different or not related. The *karta* associated with the non-finite verb is called the *nirapekkha karta* (*nominative absolute* in English). As there is an অসমাপিকা verb involved, a *con* or *seq* etc. relation is generated, also there is a possible generation of compound UW.

Example:

টম খেলে জন খাবে
tama khele jana khaabe.
 Tom if-eats John will-eat.
 If Tom eats John will eat.

salient point to note is that the *e* bibhakti can be used with all other *kaaraks* as well, so appropriate analysis has to be done to identify its functionality. Often the context of occurrence of the word and the grammatical attributes available with the word from the lexical dictionary guide in identifying the *kaarak* in case of *e* *bibhakti*.

Example:

ছাগলে	ঘাস	খায়
<u>chaagale</u>	<u>ghaash</u>	<u>khaay</u> .
Goat	grass	eat.
Goat eats grass.		

(UNL relations generated for *kartri kaarak*: *agent (agt)*, *co-agent (cag)*, *partner (ptn)* etc.).

4.2 Other *kaaraks*

Five other *kaaraks* have been analyzed exhaustively as above.

1. *Karma kaarak* (6 subcategories): *Karma kaarak* is the person or thing on which the *kartri kaarak* executes the action stated by the sentence.
(UNL relations for *karma kaarak*: *object (obj)*, *beneficiary (ben)*, *co-object (cob)*).
2. *Karan kaarak* (5 subcategories): *Karan kaarak* is the thing or tool or method by which the *kartri kaarak* of the sentence executes the specified action.
(UNL relations for *karan kaarak*: *instrument (ins)*, *method (met)*).
3. *Sampradaan kaarak* (2 subcategories): *Sampradaan kaaraks* are cases where the agent (*kartri kaarak*) does something for someone or gives away something to someone.
(UNL relations for *sampradaan kaarak*: *beneficiary (ben)*, *goal (gol)*, *purpose (pur)*, *reason (rsn)*).
4. *Apaadaan kaarak* (6 subcategories): This stands for the concept of sources of creation, location, position etc. All types of relations bearing the concept of *source* in some sense are eligible to come into this category.
(UNL relations for *apaadaan kaarak*: *place-from (plf)*, *time-from (tmf)*, *from (frm)*, *source (src)*).
5. *Sambandha pad* (4 subcategories): If related to the next noun or pronoun, then the term having a *r* (র) or *er* (এর) *bibhakti* is called a *sambandha pad*. *Sambandha pad* always has some *bibhakti* with it (never *sunya bibhakti*).
(UNL relations for *sambandha pad*: *modifier (mod)*, *possession (pos)*, *part-of (pof)*.)
6. *Adhikaran kaarak* (8 subcategories): *Adhikaran kaaraks* are the ones that describe the place, time and topic of the action performed by the sentence.
(UNL relations for *adhikaran kaarak*: *place (plc)*, *time (tim)*, *place-to (plt)*, *time-to (tmt)*, *to (to)*, *goal (gol)*, *virtual-place (scn)*, *objectified-place (opl)*.)
7. *Sambodhan* (3 subcategories): *Sambodhan* (সম্বোধন) is the case where someone hails some other person and says something to this person. This act of hailing

- Finally, a *met* relation gets resolved when the node having the *MET* attribute and the verb becomes juxtaposed.

Salient rules:

- $+ \{N, Na, ABS, \hat{PLACE}, \hat{CONCRETE}, \hat{SCN}, \hat{RSN}, \hat{TIME}, \hat{BLKINSERT}:+MET, +MORADD, +eADD, +BLKINSERT::\}$
 $\{[[e]], NMOR, BLKINSERT::\}P30;$
- $> \{N, MET, ABS, \hat{V}::met:\} \{V, \hat{METRES}, :+METRES::\}P20;$

UNL:

```
met(enchant(icl>do):0T.@entry.@future,:01)
agt(enchant(icl>do):0T.@entry.@future,I(icl>person):0P)
and:01(song(icl>song):0K.@entry,kirwana(icl>song):00)
mod:01(song(icl>song):0K.@entry,bAula(icl>song):0E)
```

This example gives a flavor of the procedure involved. Similar procedure has been applied all the various categories and subcategories. (Note: *Kirtan* and *baaul* are two Indian blends of songs.)

6 Verification

An exhaustive verification of the system has been carried out by writing a **UNL to Bengali Deconverter** (*i.e.* generator). This uses the same lexicon as the *Bengali enconversion* system and a set of *Bengali generation rules*. The enconverted input sentences have been re-generated from the UNL graphs and manually matched for conceptual equivalence. This is a form of intra-platform verification, which verifies both the preservation of information and meaning during enconversion and its wholesome retrieval during deconversion using the appropriate rule-bases. Some examples follow. Many of the output sentences map back exactly to the same set of words and sentence structure as the input, without any divergences. However, to provide a more interesting delineation (within this short span of space) of the challenges faced, we mainly give the instances of input output divergence.

1. **Projojak karta** (প্রযোজক কর্তা):

Input to enco: tama janake khelaabe
 Equivalent: টম জনকে খেলাবে
 Gloss: Tom John-to will-make-play

Meaning: Tom will make John play.

Output of deco: tama janake khelaabe
 Equivalent: টম জনকে খেলাবে
 Gloss: Tom John-to will-make-play

Remark: Exact match between input and output sentences.

Equivalent: কী কী চাও বলি
 Gloss: What what you-want I-say

Meaning: (I)/(Let me) say what (you) want.

Output of deco: aami bali tomraa kii kii caao
 Equivalent: আমি বলি তোমরা কী কী চাও
 Gloss: I say you what what want

Remark: The input to enco has no default number information associated with the person, so the output generates (by default implementation as per the rule base) a singular number output for the first person and a plural number output for the second person. As it can be seen, an *aami*, which means *I* (first person singular number) and a *tomraa*, which means *you* (second person plural number), have been explicitly added to the output.

6. **Karaner bipsaa** (করণের বীঙ্গা) (Repetition in Karan):

Input to enco: taaraay taaraay bharaa raater aakaash
 Equivalent: তারায় তারায় ভরা রাতের আকাশ
 Gloss: Star-with star-with filled night's sky

Meaning: (The) night's sky is filled with stars.

Output of deco: raater aakaash taaraader diye bhareche
 Equivalent: রাতের আকাশ তারাদের দিয়ে ভরেছে
 Gloss: Night's sky stars-with has-filled

Remark: The structural and morphological differences in the input and output here is noticeable, although the conceptual meanings are the same in both the cases.

7. **Sunya bibhakti (no bibhakti) in karan** (করণে শূন্যবিভক্তি):

Input to enco: gaadhaake haajaar caabuk maarleo se
 ghozDaa hay naa
 Equivalent: গাধাকে হাজার চাবুক মারলেও সে ঘোড়া
 হয় না
 Gloss: Donkey-to thousand whiplash in-spite-of-beating-with it
 horse become not

Meaning: In spite of thousand beatings with whiplashes a donkey does not become a horse.

Output of deco: jadi tomraa haajaar caabuk diye gaadhaake maaro tabuo
 se ghozDaa hay naa
 Equivalent: যদি তোমরা হাজার চাবুক দিয়ে গাধাকে মারো তবুও সে
 ঘোড়া হয় না

Output of deco: futbale ostaad ebang se taase pokta
 Equivalent: ফুটবলে ওস্তাদ এবং সে তাসে পোক্ত
 Gloss: In-football expert and he in-cards solid

Remark: This is an instance of free-format input natural language, where the output structure has significantly varied from the input structure, in spite of having the same meaning and hence being correct.

7 Conclusion

Systematic analysis of the case structure forms the foundation for any natural language processing system. In this paper, we have described a system for the computational analysis of the Bengali case structure for the purpose of interlingua based MT using UNL. The complementary generator system too has been implemented, which provides the platform for intra system verification. Verification via cross system generation is being done using the Hindi generation system (also under development.) Apart from the case structure, computational analysis based on authoritative grammatical treatise, addressing complex phenomena involving verbs, adjectives and adverbs is under way.

References

1. Dey, K.: Project Anubaad: an English-Bengali MT system. Jadavpur University, Kolkata (2001)
2. Sinha, R.: Machine translation: The Indian context. AKSHARA'94, New Delhi (1994)
3. Rao, D., Mohanraj, K., Hedge, J., Mehta, V., Mahadane, P.: A practical framework for syntactic transfer of compound-complex sentences for English-Hindi machine translation. (2000)
4. Bharati, A., Chaitanya, V., Sanyal, R.: Natural Language Processing: A Paninian Perspective. Prentice Hall India Private Limited (1996)
5. Dave, S., Bhattacharya, P., Girishbhai, P.J.: Interlingua based English-Hindi machine translation and language divergence. Journal of Machine Translation, Volume 17 (2002)
6. Monju, M., Sachi, D., Bhattacharyya, P.: Knowledge extraction from Hindi texts. Knowledge Based Computer Systems, Proceedings of the International Conference KBCS2000 (2000)
7. Muraki, K.: Pivot: Two-phase machine translation system. MT Summit Manuscripts and Program, pp. 81-83 (1987)
8. Uchida, H.: Atlas. MT Summit II, pp. 152-157 (1989)
9. Lonsdale, D.W., Franz, A.M., Leavitt, J.R.R.: Large-scale machine translation: An interlingua approach. (www.lti.cs.cmu.edu/Research/Kant/PDF/aei94.pdf)
10. Gonzalez, J.C., Go, J.M., Nieto, A.F.: Aries: A ready for use platform for engineering Spanish-processing tools. Digest of the Second Language Engineering Convention, pages 219-226 (1995)
11. Vauquois, B., Boitet, C.: Automated translation at Grenoble University. (acl.ldc.upenn.edu/J/J85/J85-1003.pdf)

