

An Optimal Way of Machine Translation from English to Bengali

Sajib Dasgupta

CSE, BUET, Bangladesh
sajib44new@bracuuniversity.net

Abu Wasif

Assistant Professor,
CSE, BUET, Bangladesh
wasif@cse.buet.ac.bd

Sharmin Azam

CSE, BUET, Bangladesh
elora@citechco.net

Abstract

In this paper we propose a transfer architecture which is used in the syntactic transfer of English to Bengali with optimal time complexity. The proposed transfer architecture has five stages: (1) Tagging (2) Parsing (3) Change CNF parse tree to normal parse tree (4) Transfer of English parse tree to Bengali parse tree. (5) Generation with morphological analysis. In parsing stage we use Cockey-Younger-Kasami (CYK) algorithm which has minimized parsing steps from exponential order (in TopDown or BottomUp parsing) to polynomial order. This algorithm requires that grammar is in Chomsky Normal Form (CNF). But problem with defining grammar in this way is that transfer of English parse tree to Bengali parse tree is not easy. Because elements of the English parse tree that are geographically distant may need to be in close proximity in the Bengali parse tree. So we used an approach where the English parse tree, which is generated via CYK parsing algorithm, is changed into another form of English parse tree, which in turn can be easily transferred into Bengali parse tree.

Keywords

MT: Machine Translation, SL: Source Language, TL: Target Language, CYK: Cockey-Younger-Kasami, CNF: Chomsky Normal Form

INTRODUCTION

Machine Translation from English to Bengali has become one of the most important tasks as far as Natural Language Processing of Bengali language is concerned. Our aim in this paper is to present a transfer architecture, which is not only successful in translating English sentences (Simple) to corresponding Bengali sentences, but also does it in most optimal way. Normally there are 3 stages for Machine Translation: (1) Parsing (2) Transfer and (3) Generation. But here we will propose a transfer architecture which has 5 stages: (1) Tagging (2) Parsing (3) Change CNF parse tree to normal parse tree (4) Transfer of English parse tree to Bengali parse tree. (5) Generation with morphological

analysis. We argue that this transfer architecture will take less time because among the different stages parsing is the most critical stage and we have done the parsing in optimal way with a dynamic Parsing algorithm [Cockey-Younger-Kasami (CYK) algorithm] which has minimized parsing steps from exponential order (in TopDown or BottomUp parsing) to polynomial order. [10][3]. This algorithm requires that grammar is in Chomsky Normal Form. But there are some problems associated with defining grammar that way. In this paper we discuss how to remove these problems by introducing a new architecture which is both complete and optimal.

First of all we will discuss about normal transfer architecture, then we will discuss about our proposed architecture.

1. NORMAL TRANSFER ARCHITECTURE

There are three stages in the normal transfer architecture. Here we discuss in short about them. [1][4][9]

1.1 Parsing Stage

Parsing is the process to form a source language dependent representation (Parse Tree) from a source language sentence by using grammar defined previously. [5]

Here is a sample Bengali Grammar (CFG):

1. S=NP+PRIN+NP
2. S=NP+AP+NP
3. S=NP+AP+OBJ1+OBJ2
4. S=NP+PRIN+OBJ1+OBJ2
5. NP=DET+NOUN
6. NP=PRON
7. AP=AUX+PRIN
8. OBJ2=DET+NOUN
9. PRON='He' | 'She'
10. DET='a' | 'the'
11. NOUN='pen' | 'boy'
12. PRIN='give' | 'gave' | 'make'
13. AUX='has' | 'is'
14. OBJ1='him' | 'her'

Given the above grammar, the parse tree for the sentence “He gave me a pen” is as follows:

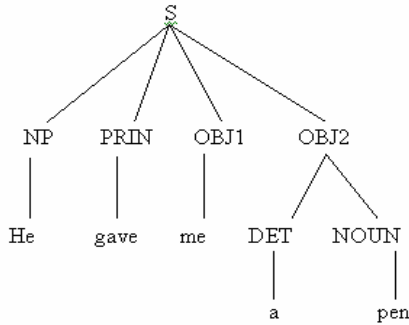


Figure 1: English Parse Tree.

Parsing can be done in many ways: for example Top Down, Bottom Up, or Dynamic Approach (like chart Parsing or CYK algorithm). [7][10]

1.2 Transfer Stage

In this stage the English parse tree is changed into their corresponding Bengali parse tree. It uses bilingual dictionary to convert English word to Bengali word and predefined syntactic transfer rules. For example the following is the generated Bengali parse tree from English parse tree shown in Figure 1.

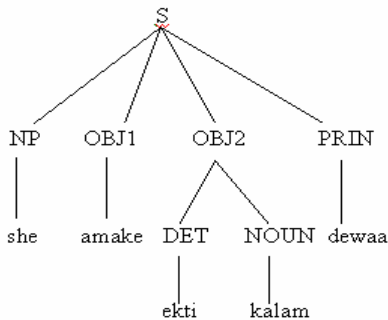


Figure 2: Bengali Parse Tree.

Syntactic transfer systems rely on mappings between the surface structures of sentences: a collection of tree-to-tree transformations is applied recursively to the analysis tree of the SL sentence in order to construct a TL analysis tree. The tree-to-tree transformation algorithm is a recursive, non-deterministic, top down process in which one side of the tree-to-tree transfer rules are matched against

the input structure, resulting in the structure on the right-hand side. [9]

Transfer Algorithm:

Let there is a rule like this in a certain grammar in English

S = NP + VP + NP (for example “He eats rice”)

Its corresponding Bengali rule will be like this

S = NP + NP + VP (for example “-p ija Mju”)

Algorithm for this rule can be:

Algorithm:

changeEnglishToBengali (engHead , bangHead)

```

{
  If ( engHead->rule = “S = NP + VP + NP” )
  BEGIN
    bangHead->childNo=3;

    for(int i=0;i<3;++i)
      bangHead->childNode[i]=new node;

    changeEnglishToBengali(
      engHead->childNode[0],
      bangHead->childNode[0]);
    changeEnglishToBengali(
      engHead->childNode[2],
      bangHead->childNode[1]);
    changeEnglishToBengali(
      engHead->childNode[1],
      bangHead->childNode[2]);

  END
}
  
```

1.3 Generation Stage

Here morphological generation takes place. A top down analysis in the parse tree will reveal what is the tense of the sentence and what is the subject’s person and number. Dictionary again provides us person, number and other information.

For the sentence: “He gave me a pen” if we transfer it as the parse tree shown in Figure 3 we will get the sentence “সে একটি কলম দেয়া”. But here morphological analysis is not taken into consideration. We can easily see that here

Subject: 3rd person, singular number
Tense: Past Indefinite
so, দেয়া --> দিয়েছিল

So the final Bengali sentence is “সে একটি কলম দিয়েছিল”.

2. OUR PROPOSED ARCHITECTURE

The proposed transfer architecture has five stages: (1) Tagging (2) Parsing (3) Change CNF parse tree to normal parse tree (4) Transfer of English parse tree to Bengali parse tree. (5) Generation with morphological analysis. The stages are shown in the following figure:

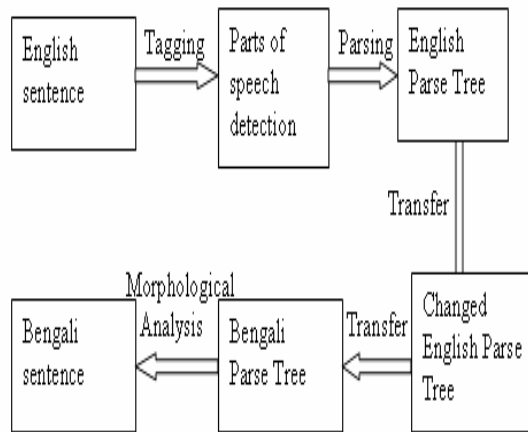


Figure 3: Transfer Architecture for MT

We can see that of the five stages two stages are new. Tagging and Transfer of English parse tree to “changed English Parse tree”. Tagging is to get the parts of speech of a particular word. For example, *ija* (vat) is tagged as NOUN. **Why there is an extra transfer stage needs some explanation.**

We argue that this transfer architecture will take less time because among the five stages Parsing is the most critical stage and in our proposed architecture we have implemented the parsing using **Cockey-Younger-Kasami (CYK)** algorithm which has minimized parsing steps from exponential order (in TopDown or BottomUp parsing) to polynomial order. It is a dynamic approach towards parsing and it makes MT time efficient. This algorithm requires that grammar is in Chomsky Normal Form (CNF). We won’t discuss how CYK algorithm works but discuss in detail how defining grammar in CNF form creates problems in terms of transfer of English parse tree to Bengali parse tree. [10][3]

2.1 Problems with CYK parsing

There are some restrictions on every grammar. For example in Regular Grammar productions are restricted to

$$A1 = a$$

Or $A1 = aA2$

Where ‘A1’ and ‘A2’ are member of non-Terminals and ‘a’ is a member of Terminals defined in grammar. Chomsky Normal Form where each production is either

$$A = BC$$

$$A = a$$

Where ‘A’, ‘B’, ‘C’ are members of non-Terminals and ‘a’ is a member of Terminals defined in grammar.

For example here is a English grammar that is defined in Chomsky Normal Form (CNF) form:

```

S=NP+VP
NP=DET+NOUN
NP= book | money
NP= He
DET= a | the
NOUN= pen | home
VP=VP+PP
VP=AP+NP
VP=PRIN+NP
VP=AP+DOBJ
VP=PRIN+DOBJ
AP=AUX+PRIN
PRIN=give | tell
AUX=has
DOBJ=OBJ1+OBJ2
OBJ1=me | him
OBJ2=DET+NOUN
PP= prepositional phrase (to go home)
----- Grammar1
  
```

But problem is that grammar defined that way is not suitable for transfer rules. **We cannot define a grammar such that transfer structure in SL and TL are markedly different.** Elements of the SL structure that are geographically distant (as they are embedded in different branches of the structure) may need to be in close proximity in the TL. We will now show some example in support of this.

2.2 Examples

In the following examples we first show some English parse trees for particular sentences and then show their expected Bengali Parse trees. The problems concerning this transfer are also described. The grammar used here is as defined above in Grammar1.

(1) "He gives me a pen" to "সে আমাকে একটি কলম দেয়":

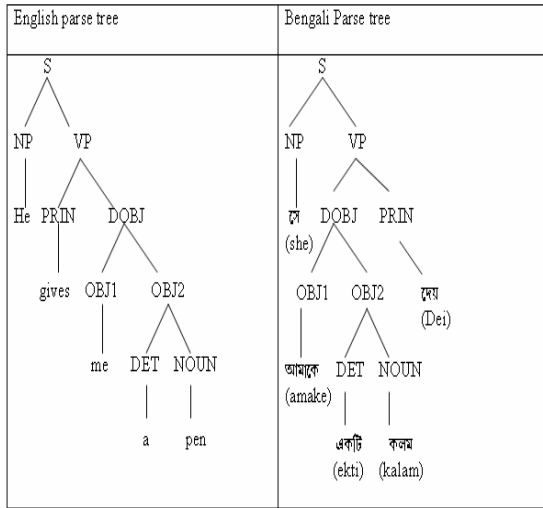


Figure 4: English to Bengali Parse Tree.

In the above example, consider the two non-terminals DOBJ and PRIN in the English parse tree. They can be swapped easily using proper transfer technique to form the Bengali Parse tree. So there should be no problem in transfer at all.

(2) "He tells me to go home" to "সে আমাকে বাড়ি যেতে বলল":

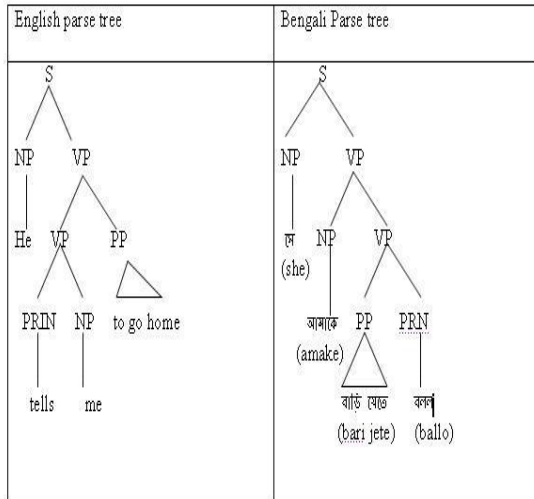


Figure 5: English to Bengali parse tree

In the above example, the non-terminals PP and PRIN of English parse tree has to be merged in Bengali parse tree, although they are quite distant. The distance factor can be even more evident in the next example.

(3) "He gave him the money from the fund" to "সে তাকে কোষাগার হতে টাকা দিয়েছিল":

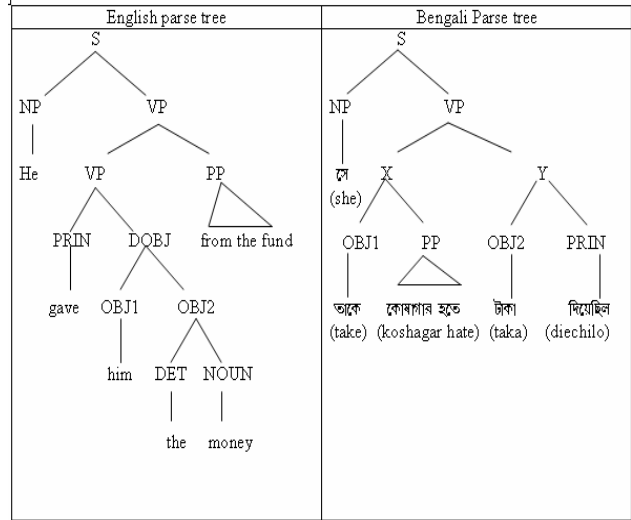


Figure 6: English to Bengali Parse Tree

In the above example, the non-terminals OBJ1 and PP as well as PRIN and OBJ2 of English parse tree has to be merged in Bengali parse tree, although they are quite distant.

So we have shown several examples where elements of the SL structure that are geographically distant may need to be in close proximity in the TL. **This happens specially when there is Prepositional Phrase included.** For these kind of examples tree to tree transfer is quite difficult and it ruins the recursive structure of the transfer algorithm shown above.

2.3 Remedy

The problem with the previous grammar is that grammatically close nodes in the Parse Tree are under different parent (which may be quite a distance away). **Basic approach to remove this problem is to make all the related nodes under the same Head/Parent.**

So if we define a grammar this way

S=NP+PRIN+NP+PP
 S=NP+AP+NP+PP
 S=NP+AP+OBJ1+OBJ2+PP
 S=NP+PRIN+OBJ1+OBJ2+PP
 -----Grammar2

-then there would have been no problems stated above and we can implement the tree to tree transfer with recursive algorithm as we shown in Section 1.2. Here is a table which shows some English rules and their corresponding Bengali rules.

Table 1: English to Bengali Transfer Rules

English Rule	Bengali Rule
(1) S=NP+PRIN+NP+PP	S=NP+PP+NP+PRIN
(2) S=NP+AP+NP+PP	S=NP+PP+NP+AP
(3) S=NP+AP+OBJ1+OBJ2+PP	S=NP+OBJ1+PP+OBJ2+AP
(4) S=NP+PRIN+OBJ1+OBJ2+PP	S=NP+OBJ1+PP+OBJ2+PRIN

But as we said earlier we need a grammar defined in CNF form to have an optimal parsing strategy (CYK algorithm). So what we did is that when parsing stage is completed we transfer the parse tree derived by CNF grammar to a tree which supports the above Grammar2 i.e. we make all the related nodes under the same parent. That's why there is an extra Transfer stage in our proposed architecture.

2.4 CNF Parse Tree to Normal Parse Tree

Here we will discuss how to transfer CNF parse tree to Normal Parse tree (From Grammar1 to Grammar2). The following figure is the parse tree transfer for the sentence "He gives me a pen"

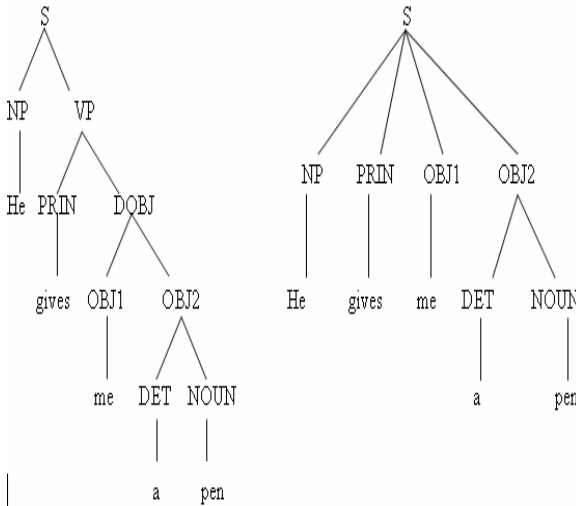


Figure 7: CNF parse tree to Normal parse tree

We can see from the above picture that there are two different types of nodes.

NODE-TYPE1: There are some nodes whose child (or children) will remain their child (or children) when we change the tree structure. For example:

OBJ2 = DET + NOUN

NODE-TYPE2: There are some nodes whose child (or children) will not remain their child (or children) when we change the tree structure. For example:

VP = PRIN + DOBJ

Here PRIN and DOBJ will be under the parent S. So, when we are at node VP, we have to know beforehand which is the parent (say S) node under which PRIN and DOBJ will be assigned as child. This can be done during the expansion or transfer of S in S = NP + VP.

To accommodate those two types of nodes we have to define 3 different kinds of rule for which transfer rule will be different.

(1) **RULE-TYPE1:** "S=NP+VP" where NP is a node of type NODE-TYPE1 and VP is a node of type NODE-TYPE2.

(2) **RULE-TYPE2:** "VP=PRIN+NP" where both PRIN and NP node is of type NODE-TYPE1.

(3) **RULE-TYPE3:** "NP=he" where child node is 1 and it is the **terminal node**.

So if all the rules and nodes are defined that way, then we can implement the tree to tree transfer in a simple pre-order top-down analysis of the source tree. Here is a skeleton program for the transfer discussed above.

2.5 Skeleton Program

```

topDownAnalysis(node head)
{
    takeAction(head, leftHead, rightHead);
    if(head.child==TERMINAL_NODE)
        return;
    topDownAnalysis(leftHead);
    topDownAnalysis(rightHead);
}
takeAction(struct node head, struct node
newLeftHead, struct node newRightHead)
{
    if (head.rule=RULE_TYPE1)
    BEGIN
        nn=create a new node at the head
        newLeftHead=nn
        newRightHead=head;
        head.childNo++;
    END

    else if (head.rule=RULE_TYPE2)
    BEGIN
        nn1=create a new node at the head
        nn2=create a new node at the head
        newLeftHead=nn1
        newRightHead=nn2
        head.childNo += 2;
    END
}

```

```

else if (head.rule=RULE_TYPE3)
BEGIN
    nn=create a new node at the head
    head.childNo++;
END
}

```

2.6 Time Complexity

So if we can define English grammar in CNF form we can parse it polynomial time order of the given number of rules using CYK algorithm. Then we can change the English parse tree by making a preorder search in the tree. After that we can get the Bengali Parse tree in another preorder search in the changed English parse tree. So whereas in Normal architecture there requires one exponential search and one pre-order search to find Bengali Parse tree, our architecture requires one polynomial order search and two pre-order search to find the Bengali parse tree, which proves that our proposed architecture is better.

3. CONCLUSION

Although Bengali is our mother tongue, there is hardly any work on complete Machine Translation of English to Bengali. In this paper we tried to provide a new transfer architecture in MT which is quite efficient. The grammars and examples we used here are simple ones. We did not consider semantic analysis or other disambiguation related with Bengali. But this paper should go down as a starting point for future implementation of a successful and complete Bengali MT engine.

ACKNOWLEDGEMENTS

We would like to thank all the teachers, employees of Bangladesh University of Engineering and Technology (BUET) for their constant support and advices.

REFEREMCES

[1]Shah Asaduzzaman, "A comprehensive study on MT towards development of Bangla-English Translation system". Thesis at CSE, Bangladesh University of Engineering and Technology, September 1999.

[2]Shah Asaduzzaman and Muhammad Masroor Ali, "Transfer Machine Translation- An Experience with Bnagla English Machine Translation System". In the Proceedings of the International Conference on Computer and Information Technology (ICCIT), Bangladesh, 2003.

[3]Berwick, R., A. Weignberg, "Parsing Efficiency, Computational Complexity and the Evaluation of Grammatical Theories". In Linguistic Inquiry, 1982.

[4]Bonnie Jean Dorr, "UNITRAN: A Principle-Based Approach to Machine Translation". MIT, Artificial Intelligence Laboratory, Cambridge, Mass. December, 1987.
<ftp://publications.ai.mit.edu/aipublications/pdf/AITR-1000.pdf>.

[5]Daniel Jurafsky and James H. Martin, "Speech and Language Processing: An Introduction to Nataural Language Processing, Computational Linguistics, and Speech Recognition", Prentice Hall, 2000.

[6]J. Earley, "An efficient context-free parsing algorithm". In Communications of the ACM 14, 453-60, 1970.

[7]Mohammed Mohisul Hoque and Muhammad Masroor Ali, "A parsing Methodology for Bangla Natural Sentences". In the Proceedings of the International Conference on Computer and Information Technology (ICCIT), Bangladesh, 2003.

[8]S. M. Shieber, "A uniform architecture for parsing and generation", In the Proceedings of COLING'88, Budaphest, Hungary, 1988.

[9]Arturo Trujillo, "Translation Engines: Techniques for Machine Translation". In Springer-Verlag London Limited, 1999.

[10]Documentation available at lingo.stanford.edu/courses/03/pg.