MACHINE AND WEB TRANSLATOR FOR ENGLISH TO BANGLA USING NATURAL LANGUAGE PROCESS

Mohammad Hasibul Haque, Md. Fokhray Hossain and A. N. M. Fauzul Hossain
Department of Computer Science and Engineering, Daffodil International University
E-mail: hasibul2363@gmail.com, drfokhray@daffodilvarsity.edu.bd

Abstract: *The modern web contents are mostly written in English and developing a system with the facility of translating web pages from English to Bangla that can aid the massive number of people of Bangladesh. It is very important to introduce Natural Language Processing (NLP) and is required to developing a solution of web translator. It is a technique that deals with understanding natural languages and natural language generation. It is really a challenging job to building a Web Translator with 100% efficiency and our proposed Web Translator basically uses Machine Translator as its mother concern. This paper represents an optimal way for English to Bangla machine and the Web translation & translation methods are used by translator. Naturally there are three stages for MT but here we propose a translation system which includes 4 stages, such as, POS tagging,Generating parse tree, Transfer English parse tree to Bengali parse tree and Translate English to Bangla and apply AI. An innovation initiative has scope of being upgraded in future and hopefully this work will assist to develop more improved English to Bangla Web Translator.*

Keywords*: Machine Translator, Web Translator, POS Tagging, Parsing, HTML Parsing, Verb Mapping*

1 Introduction

Machine Translation has a mechanism of translating languages from one to another and it is one of the important areas of Natural Language Processing (NLP). And Web Translation is a procedure of translating contents of web pages from one language to another.

Our objective is to present a transfer methodology for both of the English sentences to corresponding Bangla sentences and English web pages to corresponding Bangla web pages and that is presenting both the Machine and Web Translation procedure. Here we propose a methodology for Machine Translation which has 4 stages. Here we have used Peen Treebank algorithm [14] which has minimum parsing steps and MIL HTML [1] parser for parsing web pages.

Yahoo Babel Fish, Google Free Online Language Translator and some others highly developed solutions provide support for multi language translation like Danish, English, Chinese, Italia, Japanese, French, Greek, Korean and so on. But we were shocked to find that there is no existing system that provides the facility for Bangla translation. However some Bangladeshi and Indian developers develop English to Bangla translation methodology using Natural Language Processing [5, 6, 7, 9, 11, 12, 13] but most of them deals only with Machine Translation. From the best of our knowledge we are one of the pioneer of developing web based English to Bangla translator.

Difference between English and Bengali sentences structures:

Direct interpretation isn't possible because of structural difference between English and Bengali. There are many situations where direct interpretation causes problem. In this case we used Artificial Intelligence(AI). This type of interpretation is possible for human beings but quite difficult for computer. Consider some examples.

I have to go. "আমার যেতে হবে।" Here direct interpretation isn't possible. Let's see another example He is too weak to walk. "সে এত দূবর্ল যে হাটতে পারে না।" here direct mapping or interpretation isn't possible.

2 Proposed Methodology/Architecture

We have divided the whole system into two parts:

1. Machine Translator
2. Web Translator

Fig. 2.1 shows the block diagram of overall architecture for machine and web translation.

2.1 Machine Translator

Machine Translator translates source language to target language using Machine Translation (MT). This architecture has 4 steps. (1) POS tagging (2) Generating parse tree (3) Transfer English parse tree to Bengali parse tree (4) Translate English to Bangla using AI.

2.1.1 POS Tagging

POS tagging is the process of assigning a parts of speech for each word in a sentence. Here we have used Penn Treebank, the linguistic corpus developed by the University of Pennsylvania [14]. The POS tagger returns array of tags and tokens. A sentence is splinted into tokens. Here we have used OPEN NLP for POS tagging.

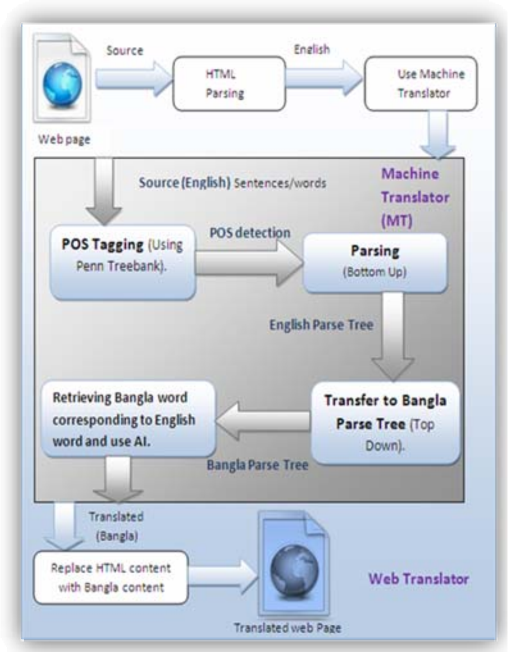

Source
HTML Parsing
English
Use Machine Translator
Web page
Source (English) Sentences/words
Machine Translator (MT)
POS Tagging (Using Penn Treebank).
POS detection
Parsing (Bottom Up)
English Parse Tree
Retrieving Bangla word corresponding to English word and use AI.
Transfer to Bangla Parse Tree (Top Down).
Bangla Parse Tree
Translated (Bangla)
Replace HTML content with Bangla content
Web Translator
Translated web Page

Fig. 1 Proposed architecture (step by step)

Following Fig. shows the tag set used by OPEN NLP [2, 3].

CC	Coordinating conjunction
RP	Particle
CD	Cardinal number
SYM	Symbol
DT	Determiner
TO	to
EX	Existential there
UH	Interjection
VBN	Verb, past participle
JJS	Adjective, superlative
VBP	Verb, non-3rd ps.sing.present
LS	List item marker (and so on)

Fig. 2 Tag set used by open NLP [2,3]

2.1.2 Generating Parse Tree (Parsing)

It is one of the complex steps and consists of chunking & parsing. The chunkier returns phrase name based on POS tagging. Producing a full parse tree is a task that builds on the NLP algorithms, which goes in grouping the chunked phrases into a tree diagram that illustrates the structure of the sentence. The full parsing algorithm is implemented by the OpenNLP library [2].

Based on chunk string parse tree is generated. Fig. 2 and Fig. 3 shows parse tree for English and Bangla sentence.

Open NLP can detect following phrases and based on these parse tree is generated.

ADJP	Adjective Phrase
PP	Prepositional Phrase
ADVP	Adverb Phrase
PRT	Particle
CONJP	Conjunction Phrase
SBAR	Clause introduced by a subordinating conjunction
INTJ	Interjection
UCP	Unlike Coordinated Phrase
LST	List marker
VP	Verb Phrase
NP	Noun Phrase

Fig. 3 Chunk set used by the OpenNLP [2, 3]

Consider following example.

English Sentence:

He gave me a book.

After Chunking:

[NP He/PRP] [VP gave/VBD] [NP me/PRP] [NP a/DT book/NN]. /.

After Parsing:

(TOP (S (**NP** (PRP He)) (**VP** (VBD gave) (**NP** (PRP me)) (**NP** (DT a) (NN book))) (. .)))

Fig. 4 shows the parse tree for "He gave me a book."

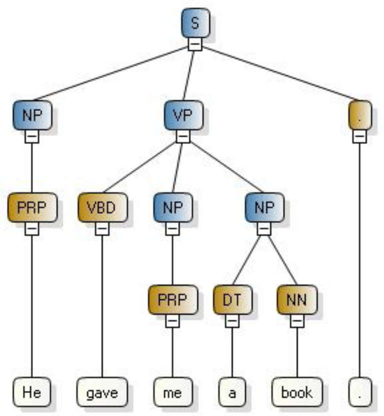


Fig. 4 English parse tree for "He gave me a book."

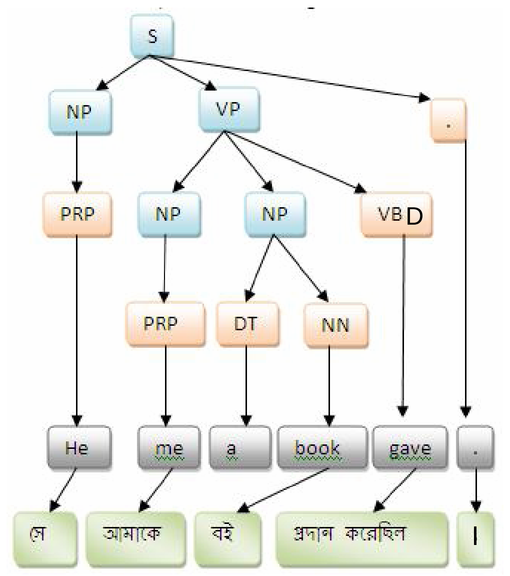


Fig. 5 Bangla parse tree for “সে অমাকে একটি কলম প্রদান করেছিল। ”

2.1.3 Transfer English to Bangla Parse Tree

Most of the Machine Translation uses Chomsky Normal Form (CNF) for defining grammar. Here we have used a sorting method for defining Bangla grammar. Let’s see the structure of English sentence and Bangla sentence according to CNF. After that we will describe our sorting method that means English to Bangla parse tree transfer algorithm.

English grammar in CNF

1. S = NP + VP + PP + NP
2. S = NP + AP+ NP+PP

And corresponding Bangla grammar in CNF

1. S = NP + NP + PP + VP
2. S = NP + PP + NP + AP

NP	=	DET + NOUN + PRONOUN
DET	=	“a”|”an”|”The”
VP	=	AUXILIRAY + PRINCIPLE

From the above example, for each rule in English there must be a Bangla rule for generating Bangla parse tree. In Context Free Grammar (CFG) we have to define grammar for different sentences.

But here we have used a **sorting method** for generating Bangla parse tree. After generating English parse tree we found whole sentence with phrase name and also tag name.

We have defined index value for each phrase/chunk and also defined index value for each tag.

Fig. 6 and 7 shows index value for phrases and tags.

PhraseName	**PhraseIndex**
NP	0
ADVP	1
ADJP	2
VP	3
..	..

Fig. 6 Phrase Indexing

TagSorting		
TagName	**TagIndex**	**AlterTagIndex**
NNP	0	
PRP$	1	
PRP	1	
DT	1.5	
NN	2	2
WP	2	
JJ	3	3

Fig. 7 Tag Indexing

These index values are used for sorting.

Now consider English parse tree (Fig. 2.1.2-b) and corresponding Bangla parse tree (Fig. 2.1.2-c) which are generated from following chunk string.

Chunk string for English Sentence:

NP (PRP)+ VP(VBD) + NP(PRP) +NP(DT,NN). (1)

Chunk string for Bangla sentence:

NP(PRP) + NP(PRP) + NP(DT,NN)+ VB. *……………..(2)*

Here we need to generate equation (2) from equation (1). If we can generate equation (2), then easily we can generate Bangla sentence. Now we will use our sorting method to get Bangla parse tree that means equation (2).

Consider following example.

Step 1: English Sentence

S = He + gave + me +a pen.

Step 2: Tagging and Parsing

S =NP (PRP) + VP (VBD) + NP (PRP) + NP (DT, NN).

Step 3: After Indexing

S = 0(1) + 3 + 0(1) + 0(1.5, 2)

After getting index value we will perform sorting based on index.

Step 4: After Sorting
BS = 0(1) + 0(1) + 0(1.5, 2) + 3
BS= NP (PRP) + NP (PRP) + NP (DT, NN) +VP (VBD).

BS= He + me + a pen + gave.

Step 5: After dictionary search
BS= সে + আমাকে + কলম + প্রদান করেছিল।

Consider another example

Step 1: English Sentence
S = what + is + your + name?
Step 2: Tagging and Parsing
S=NP (WP) + VP (VBZ) + NP (PRP$, NN)?
Step 3: After Indexing
S= 0(2) + 3 + 0(1.2, 2)?

After getting index value we will perform sorting based on index.
After Sorting we will get.

Step 4: After Sorting
BS = 0(1.2, 2) + 0(2) + 3?
BS = 0(1.2, 2, 2) + 3?
BS=NP (PRP$, NN, WP) + VP (VBZ)?
BS= your, name, what + is?
Step 5: After dictionary search
BS= আপনার নাম কি ?

By considering above steps we may transfer English sentence to Bangla sentence. We have also used following algorithm for transforming

```
BanglaParseTreeGeneration(English chunk set)
{
    1. Searching for conjunctional phrase or tag named CC.
        a. If found split sentence in two segment
        b. Else Don't need to segment.
    2. searching for Preposition.
        a. If found
            i. At the end of the sentence or segment then place it before first phase of the sentence.
            ii. Not at the end of the sentence or segment then place it before next noun and pronoun.
    3. Retrieve index no corresponding to Phrase and Tag name.
    4. For each index find out corresponding Phrase name and then search that phrase from chunk set
        a. If found then put in SortedChunk = SortedChunk +searched Phrase(Chunk)
    5. Convert SortedChunk to SortedChunkArray[] by Splitting
    6. For each chunk perform tag sorting
        a. SortedChunk[..] = TagSorting(SortedChunk[..]'s tags)
    7. Retrun SortedChunk; // This is the sorted sentence with tag and chunk used to generate Bangla parse tree.
}
GetBanglaSen(BangSortedChunkList){
    String Banglasen ="";
    For each tag in sorted chunk
    Banglasen := Banglasen + GetBanlaWar(ward/tag);
    Return BanglaSen;
}
GetBanglaWar(English ward/tag)
{
    If (tag == IN||TO) // for preposition
    {
        Return PrepositionMapping(English ward, reference object);
    }
    If (tag == VB||VBD||VBG||MD||VBN)
    {
        Return VerbMapping(...., EnglishWard);
    }
    Search Dictionary for Enlish Ward.
    If found then set BanglaWard := searched ward.
    else if tag == NN then
    set BanglaWard := Phonatic(Engward)
    else BanglaWard = EngWard;
    return Banglaward;
}
```

Fig. 8 Algorithm for Bangla parses tree generation

2.1.4 Translate English to Bengali using AI

The next process is performing dictionary lookup. After getting Bangla Parse tree means Bangla chunk string we perform dictionary look. If the word is found then we write the corresponding Bangla meaning on the contrary if not then return that English word. But the technique is different for verbal, prepositional phrases. For translating verb and preposition we have used verb mapping and preposition mapping.

2.1.4.1 Verb Mapping

Verb mapping is the process of creating Bangla verb corresponding to English verb. Bangla has come from an origin of Sanskrit. In Bangla final form of verb can be generated by knowing the base form of the root verb. More preciously construction the final verb in Bangla can be viewed as method of multiplication between a scalar quantity and matrix.

Our Verb Mapping takes four parameters

1. Reference object (Noun/Pronoun)
2. Auxiliary verb
3. Been
4. Main verb

Based on these parameters **we** generate Bangla verb.

Consider following sentence.

I am doing math. ←→ আমি গণিত করছি।

Here Reference object = I.

Auxiliary verb = am

Been =null

And Main verb = doing.

The above example has two verbs ***am*** with lemma be and ***doing*** with lemma do. Here verb is acting actively. To construct the final verb we first performs a dictionary lookup based on verb tag whose base form in Bangla is "কর". Then it looks up for the element in the verb mapping table corresponding to person and tense. Here person is generated from reference object and tense is generated from combination of auxiliary and main verb. Let's see once more.

Here "I" is first person. Auxiliary verb "am" represents present tense. Main verb "doing" represents continuous.

We hope this following table will be more helpful for realizing the Verb mapping based on above information.

Table 1 Verb modifier suffix table for 1st person when verb is active

	Indefin ite	Continu ous	Perfect	Perfect Continu ous
Present	ি	ছি	েছি	ছি
Past	েছিলাম	ছিলাম	েছিলাম	ছিলাম
Future	ব	ব	ব	ব

From this table we generate suffix. After generating suffix, it will add with base verb. Here "করছি" is generated as

কর * ছি = করছি[5]

For this example, the rule of joining is rather trivial. However, for many situations the joining rules can be quite involved. Generations of final verbs for all tense forms with subject being first person and verb being active, are illustrated below.

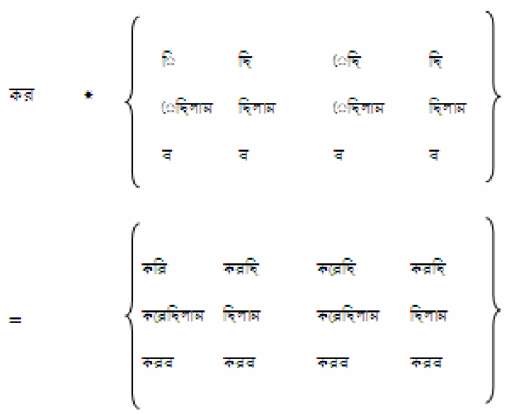


Fig. 9 Verb mapping (Suffix table)

Above verb form generating equation can be abstractly written as

Root verb form * Universal suffix matrix = Final verb forms matrix

Where a part of "Universal suffix matrix", relevant for *first person*, is illustrated in the Table 1. This aptly demonstrates the structural advantages for machine synthesis of Bengali sentences. These underlying structures owes to the fact that Bengali derives its origin from *Sanskrit*. [10]

2.1.4.2 Preposition Mapping

Preposition is usually placed before noun/pronoun and shows the relationship among other nouns, pronouns and verbs in the sentence. The noun that follows a preposition, i.e., the reference object is in the accusative case and is governed by the preposition. Prepositions can also be defined as words that begin prepositional phrases (PP). A PP is a group of words containing a preposition, an object of the preposition, and any modifiers of the object.

The translation of the three English prepositions 'in', 'on', and 'at' has involved for identifying the possible inflection to be attached to the head noun of the PP. No postpositional words are placed after the head noun for these prepositions. The three prepositions 'in', 'on', and 'at' (which are both spatial and temporal in nature) can be translated into the Bengali inflections ে (-*e*), তে (-*te*), -েতে (-*ete*) and য় (-*y*). Any of these 4 Bengali inflections can be placed after the reference object for any of these 3 English spatial and temporal prepositions. The choice depends on the spelling of the translated reference object. The rule is:

If the last letter of the Bengali representation of the reference object is a consonant, ে (-*e*) or - েতে (-*ete*) is added to it (e.g., at/in market বাজারে [*bazar-e* / *bazar-ete*]), else if the last letter of the Bengali word is a *matra* (vowel modifier) and if the *matra* is া (-*a*), any of তে (-*te*), or য় (-*y*) can be added to the Bengali reference word (e.g., in evening → সন্ধ্যাতে / সন্ধ্যায় [*sandhya-te* / *sandhya-y*]), otherwise 'তে' (-*te*) is added to it (e.g., at home → বাড়িতে [*badi-te*]). When translating the temporal expressions, if '*on*' is followed by a day (like Sunday, Monday etc.) or by a date in English, null inflection is added. To translate this type of PPs, we take the help of an example base, which contains bilingual translation examples. Here are some translation examples from the example base (TLR – target language representation of the reference object). [6, 7, 8]

1) at / in (place) ←→(TLR) - ['ে'/ 'তে' - (*e* / *te*)]

(2) of (NP)←→ (TLR) - ['র'/ 'ের'- (*r* / *er*)]

Now we consider following examples;

He reached before evening. ←→ সে সনধ্যার পূর্বে পৌছাল।

I am waiting for him. ←→ আমি তাহার জন্য অপেক্ষা করছি।

Rahim is going to school. ←→ রহিম বিদ্যালয়ে যাচ্ছে।

Here "before" is a preposition and evening represents Noun. Consider following table for preposition mapping.

Considering Preposition Mapping Table.

PP = Preposition
B1 = Represents Bangla Preposition when last latter of noun is vowel.
B2 = Represents Bangla Preposition when last latter of noun isn't vowel.
Person = Represent objects person.

Table 2 Preposition Mapping Table

PP	b1		person
after	র পরে	ের পরে	3
around	র চারিদিকে	ের চারিদিকে	3
at	তে	ে	3
before	র পূর্বে	ের পূর্বে	3
beside	র পাশে	ের পাশে	3
by	দ্বারা		3
for	জন্য	জন্য	1
for	র জন্য	ের জন্য	3
for	ার জন্য	ার জন্য	2
..	..	..	..

2.2 Web Translator

Web Translator uses Web Translation (WT) and it is the process of translating web pages. Web translation uses the service of Machine translation. Following Fig. shows the block diagram of web translation.

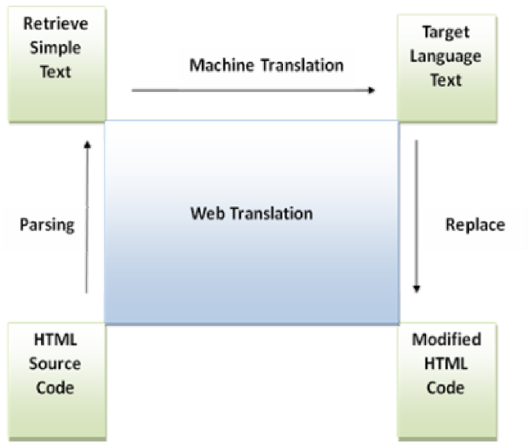


Fig. 10 Block diagram for web translation.

Web translation performs following operations.

1. HTML Parsing
2. Send to machine translator
3. Replace source text to target text

2.2.1 HTML Parsing

Web pages are shown in browser and web pages are viewed as an HTML page and consist of HTML code. Here we have used HTML parser [1] for parsing the HTML code. In this stage we have filtered the text from HTML source code. For filtering the

HTML source we considered the HTML document as a tree.
Consider following HTML code

```
<html>
   <head>
       <title>This is tested by
       ER</title>
   </head>
   <body>
       <h1>Hi I am hasibul.</h1>
         <a href="#" title = "ha
         ha">doing test.</a>
   </body>
</html>
```

Fig. 11 HTML source code

If we format this source code according to tree then we will find following tree.

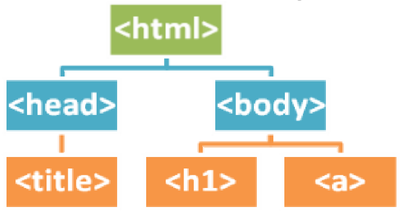


Fig. 12 Tree view of HTML source code

Our algorithm finds those nodes which have no child. Then those nodes are sending to machine translator for translating.
We have used following algorithm for finding the text node.

```
Step 1: Retrieve HTML code into a string
Step 2: While(HTML string !=EOF)
            {
            Searching for initial tag If found
            {
              If(tag is node or element)
                Nodes[].element := element
                Nodes[].parent  := parent element
                Nodes[].child   := child element

      If(tag is script)
        Put into Script[]:= script

      If(tag is style)
        Put into Style[]: = style
            }
Step 3: foreach(node in nodes)
            {
                        If(node.child == false)
                        {
                                    Text[] = node.text;
                        }
            }
```

Fig. 13 Algorithm for retrieving text from HTML source.

2.2.2 Send to Machine Translator

This is simple stage. Here we just send each text to machine translator as an input parameter of machine translator and machine translator returns target language.

2.2.3 Replace source text to target language text

This is the last stage of web translation. Now we have source text, corresponding target text and source HTML code. For replacing source text to target text we need to perform find and replace operation. Find operation searches the source text in HTML source code and replace operation replaces that source text to target text. After replacing source code is written in memory as a HTML file and shown in a browser. We have used algorithm in Fig. 14 for finding and replacing.

```
// we have a string with
source HTML code  strSource
// we have a array of text
text[]
Foreach (string strText in
text[])
{
      Set MSource :=
      replace(strSource,
      strText, MT(strText));
}

// this function write on disk
Write(MSource);
//MT() returns target text
that means
  It takes English text as a
  parameter  and returns
  Bangla text
```

Fig. 14 Algorithm for replacing source text to Target text

3 Results

As we described our translator perform machine and web translation so we can divide result finding in two parts.

3.1 Machine Translation

For every translation it requires minimum 2.40 sec because it requires 2.40 sec for initializing service.

3.1.1 Performance

Following table shows performance based on above (Table 3) input and output.

3.2 Web Translation

Web translator receives an URL of an English web page and produce corresponding Bangla page temporarily.

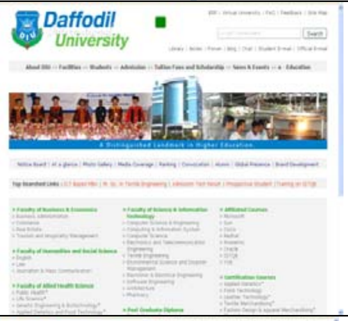
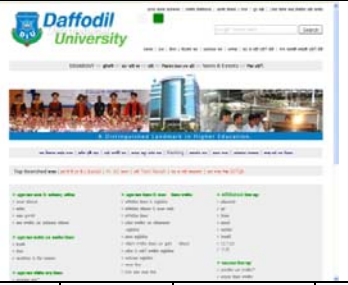
Table 3 Input and corresponding output.

Case1	*Input*	*The boy is playing.*
	Output	বালকটি খেলা করছে।
Case2	*Input*	*This is test. I am waiting for him.*
	Output	এইটি পরীক্ষা। আমি তাহার জন্য অপেক্ষা করছি।
Case3	*Input*	*This is the boy who came yesterday. Who are you? I am waiting for you. I love you.*
	Output	এইটি বালক গতকাল যে এসেছিল। আপনি কে? আমি আপনার জন্য অপেক্ষা করছি। আমি আপনাকে ভালবাসি।

Table 4 Performance of sample input and output

Case	*Words/ Sentences*	*Time*	*Accuracy*
1	*W: 4, S:1*	*2.40 Sec*	*100%*
2	*W:8, S:2*	*2.41 Sec*	*100%*
3	*W:18, S:4*	*2.45 Sec*	*94%*

Table 5 Input and output page generated by the system

Input URL	*http://daffodilvarsity.edu.bd/*
Output URL	*http://203.190.10.142 /eranubadok/output/result.htm*
Input Page (English)	
Output Page (Bangla)	

Case	*Words/ Sentences*	*Time*	*Accuracy*
1	*W:531*	*16.21 S*	*89%*

3.2.1 Performance

It is quite difficult to create a translator which provides 100% efficiency. Our translator unable to translate compound and big sentences because lack of AI, rule and also lexical resource. If we reach our lexical resource and introduce more grammatical rule then it will show good performance.

4 Conclusion

Though a complicated work is done, our proposed system requires more improvement in some cases like detecting multiple Bangla meaning for an English word and improving the artificial intelligence to detect phrases and idioms. Moreover the sentence structure of Bangla and English varies for different sentences. As why the sorting method we have proposed fails to translate all sentences with 100% efficiency. In future we would like to improve our proposed system for detecting multiple meanings, phrase and idioms. Besides we will develop a strong data dictionary.

In this paper we tried to provide transfer architecture in MT and also WT (web translation) which is quite efficient. The grammars and examples we used here are simple. We didn't consider semantic analysis or other disambiguation related with Bengali. It has been implemented in such a way to make it possible for extending in the future successfully and efficiently.

References

[1] MIL HTML Parser. [Online]. Available at: http://www.codeproject.com/KB/dotnet/apmilhtml.aspx [accessed 25 November 2008]

[2] Open NLP Class Library Help. [Online]. Available at: http://opennlp.sourceforge.net/ [accessed 15 February 2009]

[3] Statistical parsing of English sentences. [Online]. Available at: http://www.codeproject.com/KB/recipes/englishparsing.aspx [accessed 15 November 2008]

[4] SharpNLP - open source natural language processing tools. [Online]. Available at: http://www.codeplex.com/sharpnlp [accessed 18 Octobor 2008]

[5] Golam Murtuza Hossain, "Anubadok The Bengali Machine Translator". [Online]. Available at :http:// bengalinux.sourceforge.net/cgi-bin/anubadok/index.pl [accessed September 18 2008]

[6] Sudip Kumar Naskar and Sivaji Bandyopadhyay, 2006. "Hadling of in English to Bengali Machine Translation". In, third asl-sigsem workshop on preposition, Trento, Italy.

[7] Diganta Saha and Sivaji Bandyopadhyay, "A semantic based English-Bengali EBMT system for Translating News Headlines".

[8] Kevin Collins, 2005. "Microsoft Jet 3.5 Performance Overview and Optimization Techniques".

[9] Mortoza Ali and Mohammad Ali, 2002. “Development of machine translation Dictionaries for Bangla language”. In 5th ICCIT, p. 272-276.

[10] Bengali language. [Online]. Available at: http:// www.answers.com/topic/bengali-language [accessed 15 September 2008]

[11] Sajib Dasgupta, Abu Wasif, Sharmin Azam, “An optimal way of machine translation from English to Bengali”. In IICT 4.

[12] Md. Hanif Siddique, A K Mohammed Sohel Rana, Abdullah Al Mamun, 2003. “A new approach of Bangla machine translation considering Allomorph”. In 6th IICT, p. 308-312.

[13] Saha Goutam Kumar, “ The EB ANUBAD translator: A hybrid scheme”. Journal of Zhejiang University Science, [Online]. Available at: www.zju.edu.cn/jzus [accessed 12 April 2009]

[14] The Penn Treebank Project. [Online]. Available at: http://www.cis.upenn.edu/~treebank/ [accessed 12 October 2008]

Mohammad Hasibul Haque is working as a software developer, Software Department, Daffodil International University. He has obtained his Bachelor from Daffodil International University. His research interests are natural language processing, Software architecture.

Dr. Md. Fokhray Hossain was born on 12th November 1965 at Rangpur, Bangladesh. He obtained his B.Sc. (Hons) and M.Sc. in Physics from and subsequently appointed as a research fellow at ***Dhaka University*** 1991- ***Jahangirnagar University*** 1993. Dr. Hossain obtained his Ph.D. from ***University of Glamorgan***, UK back 1998 through Overseas Development Administration Shared Scholarship Scheme (ODASSS). He worked for Daffodil Institute of IT (DIIT) as a faculty as well as an Academic Director from 1999 to 2008. Currently, he has joined Daffodil International University (DIU) as Registrar on 3rd November 2009. Earlier, he was the Head of the Department of Computer Science and Engineering of DIU. Before that he had also served at various local and foreign universities as academician and researcher including ***London Metropolitan University***, UK.
He also achieved different professional certificates such as *Time Management and Introduction to HRM* (2008) in collaboration with Netherlands Management Programme (PUM) and DIU; *Academic Standards and Quality* (2005) from National Computing Center (NCC), UK; *e-business Model and Strategy* (2004) from S. Korea through DGF-KTC scholarship; *Class Management & English Language Training* (2003) from British Council; *Digital Telecommunication Technology* (2001) through AOTS scholarship, Japan.

A.N.M Fauzul Hossain is working in International Organization for Migration (IOM). He has obtained his Bachelor from Daffodil International University.