Rule Based Machine Translation for Assamese-English Using Apertium
PranjalDas¹, KalyaneekanchanBaruah², AbdulHannan³, ShikharKr.Sarma⁴
Department of InformationTechnology
IST, GauhatiUniversity
Guwahatii
India

Abstract: Machine Translation is the process of using a software application to convert a source language into a target language without the intervention of human. It is an important part of Natural Language Processing. Here we have proposed a method for translating Assamese to English sentences and vice versa using a Rule Based Machine Translation (RBMT) approach. RBMT uses linguistic rules to analyze the inputted content in the source language to generate the output text in the target language. We have used Apertium toolkit, which is a free and open-source platform for RBMT. We have used two monolingual dictionaries and a bilingual dictionary for building the Assamese-English RBMT system. We wrote some transfer rules to deal with word-reordering and other grammatical stuffs.

Keywords: Apertium, Transfer Rules, Assamese, Morphology, Machine Translation, Dictionary

I. Introduction
In this paper, we have described the development of the Assamese-English (asm-eng) language pair using Apertium RBMT. Assamese is a language mainly spoken by the people of Assam, a state in the north-east of India [9]. In Assam, not much has been done in the field of NLP and Machine Translation. Assamese is a computationally underdeveloped language and research on Natural Language Processing (NLP) is still at its developing stage. Today, almost everything is web-based and majority of the information is available in English. So the purpose of our project is to enable the common Assamese people, who have lesser knowledge about the English language, to stay abreast with the recent advancements of the world by providing him an Machine Translation interface. The purpose of the project is also to let the world know about the Assamese language. Since Indian languages are morphologically very rich and agglutinative in nature, majority of the MT projects in India are hybrid or statistical based. But we believe that if we have a firm grip on the language, RBMT can be better. In this paper, we have briefly described about the design of the RBMT system using the Apertium platform. Then, we have described about the dictionaries and the rules used. Following this, we have given an evaluation of the quality of output of our system and finally we concluded our paper with a discussion and prospective for future work.

II. Related Works
Machine Translation in India is not a new process. There are many machine translation projects in India [2]. Some of the Machine Translation systems for Indian languages are:

<table>
<thead>
<tr>
<th>Machine Translation System</th>
<th>Source-Target Language</th>
<th>Developer</th>
<th>Approach Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANGLABHARATI (1991)</td>
<td>English-Indian Languages (primarily Hindi)</td>
<td>IIT, Kanpur</td>
<td>Pseudo-interlingua</td>
</tr>
<tr>
<td>UCSG-based MT</td>
<td>English-Kannada</td>
<td>University of Hyderabad</td>
<td>Transfer based</td>
</tr>
<tr>
<td>Anusaaraka (1995)</td>
<td>English to Hindi and five other Indian Languages to Hindi</td>
<td>IIT, Kanpur</td>
<td>LWG Mapping/PG</td>
</tr>
<tr>
<td>VAASAANUBAADHA (2002)</td>
<td>Bengali- Assamese</td>
<td>Kommaluri Vijayanand, S Choudhury and Pranab Ratna</td>
<td>EBMT</td>
</tr>
<tr>
<td>Anuvadaksh</td>
<td>English to Hindi, Urdu, Oriya, Bangla, Marathi, Tamil</td>
<td>EILMT consortium</td>
<td>Hybrid Approach</td>
</tr>
</tbody>
</table>

Several universities in India, along with CDAC have been working on improving Machine Translation for Indian Languages. MHRD, India is also taking several measures to make Machine Translation serve for the people of India.

We are using Apertium to develop our RBMT system. There are many language pairs in use for Apertium. Work is still going on for some language pairs. Some of the stable language pairs that are currently in use are shown in the following figure:
Figure 1: Stable language pairs of Apertium.

All the released pairs are stored in the trunk [3] of Apertium.

III. Files and Tools used for Implementation

A. Apertium
Apertium is a free and open-source platform used in Rule Based Machine Translation. Apertium uses FST (finite-state transducers) for lexical processing to treat many kinds of multi-word expressions. Hidden Markov models (HMM) are used in Apertium for part-of-speech tagging. Apertium also uses chunking, which is finite-state based used for structural transfer.

B. Lttoolbox
It is a finite state toolkit used for lexical processing, morphological analysis and generation of words [4]. Three programs are split in the lttoolbox:
- lt-comp – the compiler
- lt-proc – the processor
- lt-expand – generates all possible mappings in the dictionary between surface forms and lexical forms.

C. Monolingual Dictionaries
The monolingual dictionary or the morphological dictionary contains the rules how words in that dictionary are inflected. It is written in XML format. The monolingual dictionary establishes the correspondences between Surface Forms (SFs) and Lexical Forms (LFs) [5]. For our language pair Assamese-English, we have two monolingual dictionaries –
- apertium-asm-eng.asm.dix – for Assamese
- apertium-asm-eng.eng.dix – for English

D. Bilingual Dictionary
The bilingual dictionary contains relations between words and symbols in the two languages i.e. it establishes the correspondences between source language LFs and target language LFs [5]. For our language pair, we have the following bilingual dictionary –
- apertium-asm-eng.asm-eng.dix

E. Transfer Rules
The transfer rule file contains some grammatical rules for how a source language will be changed to a target language. These rules are mainly used because all languages do not follow the same word order. For e.g. English follows SVO construction, while Assamese follows SOV word order construction. Two modules of structural transfer can be used –
1-stage transfer – only one set of transfer rules is used (.t1x file).
3-stage transfer – transfer rules are used to group words into chunks, on which later operations such as inter-chunk and post-chunking can be performed (.t1x, .t2x and .t3x files).
1-stage transfer can be used for languages which are closely related [6], for e.g. Assamese-Bengali. For Assamese-English we have to use 3-stage transfer rules. We have used the following transfer files –
apertium-asm-eng.asm-eng.t1x, apertium-asm-eng.asm-eng.t2x, apertium-asm-eng.asm-eng.t3x for Assamese to English
apertium-asm-eng.asm-eng.t1x, apertium-asm-eng.asm-eng.t2x, apertium-asm-eng.asm-eng.t3x for English to Assamese.

IV. Methodology

A. The Apertium Architecture

The eight module architecture of Apertium [7] is shown in the following figure:

![Architecture of Apertium](image)

Deformatter

The deformatter separates text from the format information (html tags, space etc) by encapsulating it with brackets, thus the rest of the module ignores it thinking it as blank space between words [12].

For example the html text দিল্লী <em>ভাৰতৰ</em> রাজধানী। would be transformed by the deformatter as দিল্লী [<em>]<বারৰ</em>।

The deformatter allows only text to be translated from the format information not the html tags [14].

Morphological Analyser

Here text in the surface forms is tokenized and one or more LFs (lexical forms) are delivered from each SF (Surface form). For e.g.

^দিল্লী/দিল্লী/np/th<sg><nom>$ ^ভাৰত/ভাৰত/np/th<gen>$ ^রাজধানী/রাজধানী/n/m<sg><nom>$^।$^sent>$

Here ‘দিল্লী’ is analyzed into lemma ‘দিল্লী’, proper noun, toponym, singular and nominative; ‘ভাৰত’ is analyzed into lemma ‘ভাৰত’, proper noun, toponym, singular and genitive. The characters ‘^’ and ‘$’ delimit the analyses for each SF; LFs for each SF are separated by ‘/’; angle brackets ‘< . >’ are used to delimit grammatical symbols.

The string after the ‘^’ and before the first ‘/’ is the SF as it appears in the source input text [13].

Parts of Speech Tagger

This step chooses one lexical form for those words having more than one.

The output of the tagger is something like this:

^দিল্লী/np/th<top><sg><nom>$ ^ভাৰত/np/th<top><sg><gen>$ ^রাজধানী/n/m<sg><nom>$^।$^sent>$

Structural Transfer and Lexical Transfer

As the name says ‘Lexical’ changes words while ‘Structural’ changes the word order.

The output of the lexical transfer is:

^দিল্লী/np/th<top><sg><nom>/Delhi<np/th<top><sg><nom>$ ^ভাৰত/np/th<top><sg><gen>/India<np/th<top><sg><gen>$

Structural Transfer is performed in 3 stages: chunking, interchunking and postchunking.

The final output of the Structural transfer stage would be as follows:

^Delhi/np/th<top><sg>$ ^be/vbser><pres><p3><sg>$ ^capital/n/capital<n><sg>$ ^of/prp>$

Morphological Generator

Generate the surface forms of the target language words from the decomposition in lemma plus attributes obtained from the previous steps.
Post Generator

Performs some TL orthographical transformations, such as contractions (দিন্দু + স্থান $\rightarrow$ দিন্দুস্থান), inserting apostrophes (can + not $\rightarrow$ can’t) [14].

Reformatter

The reformatter restores formatted information encapsulated by the deformatter.

For e.g. “Delhi [<em>] is capital of </em> India.” is restored into “Delhi <em>is capital of</em> India.”

B. Creating Dictionaries

Apertium basically works on dictionaries and shallow transfer rules. There are 3 basic dictionaries: two monolingual dictionaries and a bilingual dictionary.

1) The Monolingual Dictionaries

We have two monolingual dictionaries, the Assamese monolingual dictionary and the English monolingual dictionary. The skeleton of our monolingual dictionary is given below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dictionary>
  <sdefs>
    <sdef n="n"/>
    <sdef n="sg"/>
    <sdef n="pl"/>
  </sdefs>
  <pardefs>
    <pardef n="অপ্রাধ_ন">
      <e><p><l/><r><s n="n"/><s n="nt"/><s n="sg"/><s n="nom"/></r></p></e>
      <e><p><l><b>ববাৰ</b></l><r><s n="n"/><s n="nt"/><s n="pl"/><s n="nom"/></r></p></e>
    </pardef>
    <pardef n="অপ্রাধ">
      <e><p><l><b>অপ্রাধ</b></l><r><s n="n"/><s n="nt"/></r></p></e>
    </pardef>
  </pardefs>
  <section id="main" type="standard">
    <e><i>অপ্রাধ</i><par n="অপ্রাধ_ন"></e>
  </section>
</dictionary>
```

Here the singular form the noun is “অপ্রাধ” and the plural form is “অপ্রাধববাৰ”. Here the noun “অপ্রাধ” is also neuter gender and nominative case.

Now we put the entry in the <section>.

```xml
<section id="main" type="standard">
  <e lm="অপ্রাধ">
    <par n="অপ্রাধ_ন"></par>
  </e>
</section>
```

This entry states the lemma of the word, अप्राध, the root, अप्राध and the paradigm with which it inflects अप्राध_न [8].

In the same way we have created the English monolingual dictionary.

2) The Bilingual Dictionary

The bilingual dictionary describes the mapping between words [8]. Lexical transfer is performed by the bilingual dictionary i.e. it gives translations between words and not tags. The skeleton of the bilingual dictionary is shown below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
```
The skeleton is very much similar to that of the monolingual dictionary. Here the paradigm section is not required. We just need add an entry to translate between the two words in Assamese and English, inside the <section> tag. This can be done as following:

```xml
<section id="main" type="standard">
  <e>
    <p>
      <l>
        অপৰাধ
        <s n="n"/>
        <s n="nt"/>
      </l>
      <r>
        crime
        <s n="n"/>
      </r>
    </p>
  </e>
</section>
```

All the nouns, pronouns, verbs etc. are added to the bilingual dictionary inside the <section> tag likewise.

**Transfer Rules**

Making rules is the toughest part in RBMT based systems. A sound knowledge in the relevant language is very much required in making the grammatical rules. The lexical transfer only gives translations between words. The reordering of words is not performed by lexical transfer. Now we will perform structural transfer, which can change the order of words and can change or add tags on a per-category (group of lemmas) basis. As we are dealing with Assamese and English languages, we have to perform the 3-stage transfer [10]. The stages are:

- **Transfer stage or chunk stage:** Here the words are translated using the bilingual dictionary and grouped and put into chunks (in the .t1x file). The tags in the words can also be added here and removed or made into 'pointers' that points to the tags in the enclosing chunk.
- **Interchunk stage:** In this stage the chunks are reordered, combined and split and chunk tags are changed (in the .t2x file).
- **Postchunk stage:** This is the final stage of structural transfer. Here the chunks are restored (in the .t3x file).

**Results and Evaluation**

Now as we have the dictionaries and the transfer rules, we can test our system with some sentences. Some translations are not perfect, because of the problem in the transfer rules. We are currently working on adding more and more rules into our system.

The current status of our work is shown in the following table:

<table>
<thead>
<tr>
<th>Table II Status of our language pair</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number Entries</strong></td>
</tr>
<tr>
<td>Monolingual Dictionary (asm)</td>
</tr>
<tr>
<td>Bilingual Dictionary (asm-eng)</td>
</tr>
<tr>
<td>Monolingual Dictionary (eng)</td>
</tr>
<tr>
<td>Transfer rules (asm→eng)</td>
</tr>
<tr>
<td>Transfer rules (eng→asm)</td>
</tr>
</tbody>
</table>

Some test results for Assamese to English and vice versa are shown in the following tables:

**Table III Assamese to English Translation**

<table>
<thead>
<tr>
<th>Assamese</th>
<th>English</th>
<th>Correct English</th>
</tr>
</thead>
<tbody>
<tr>
<td>জয়পুৰ রাজস্থানের এখন দবখযাত চিৰ।</td>
<td>Jaipur is a famous city of Rajasthan.</td>
<td>Jaipur is a famous city of Rajasthan.</td>
</tr>
<tr>
<td>তাজমিল আগ্রাত অৱদস্থত।</td>
<td>Taj Mahal is situated Agra.</td>
<td>Taj Mahal is situated in Agra.</td>
</tr>
<tr>
<td>জামা মছদজি শ্বািজািাবন দনমমাণ কদৰদছল।</td>
<td>Jama Masjid was built by Shahjahan.</td>
<td>Jama Masjid was built by Shahjahan.</td>
</tr>
<tr>
<td>দিৰি ভারতের বলদী।</td>
<td>Delhi is capital of India.</td>
<td>Delhi is the capital of India.</td>
</tr>
<tr>
<td>on bank of Ganges river situated Ujjain India of #seven sacred cities of inside #one of the.</td>
<td>Situated near the banks of the river Ganga, Ujjain is one of the seven sacred cities of India.</td>
<td></td>
</tr>
<tr>
<td>মুমতাজ মহালের পত্নীতা।</td>
<td>Mumtaz Mahal was wife of Shahjahan.</td>
<td>Mumtaz Mahal was the wife of Shahjahan.</td>
</tr>
<tr>
<td>অন্ধ্ৱপ্ৰবিশভাৰতের ঐতিহাসিক।</td>
<td>Andhra Pradesh is a state of India.</td>
<td>Andhra Pradesh is a state of India.</td>
</tr>
</tbody>
</table>

**Table IV English to Assamese Translation**

<table>
<thead>
<tr>
<th>English</th>
<th>Assamese</th>
<th>Correct Assamese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delhi is the capital of India.</td>
<td>ভারতের রাজধানী।</td>
<td>ভারতের রাজধানী।</td>
</tr>
<tr>
<td>মুমতাজ মহালের পত্নীতা।</td>
<td>তাজমহলের পত্নীর।</td>
<td>Mumtaz Mahal was the wife of Shahjahan.</td>
</tr>
</tbody>
</table>

---

**Table V Status of our language pair**

<table>
<thead>
<tr>
<th>Number Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>3983 words</td>
</tr>
<tr>
<td>4341 words</td>
</tr>
<tr>
<td>5940 words</td>
</tr>
<tr>
<td>13 rules</td>
</tr>
<tr>
<td>11 rules</td>
</tr>
</tbody>
</table>

**Table VI Assamese to English Translation**

<table>
<thead>
<tr>
<th>Assamese</th>
<th>English</th>
<th>Correct English</th>
</tr>
</thead>
<tbody>
<tr>
<td>জয়পুৰ রাজস্থানের এখন দবখযাত চিৰ।</td>
<td>Jaipur is a famous city of Rajasthan.</td>
<td>Jaipur is a famous city of Rajasthan.</td>
</tr>
<tr>
<td>তাজমিল আগ্রাত অৱদস্থত।</td>
<td>Taj Mahal is situated Agra.</td>
<td>Taj Mahal is situated in Agra.</td>
</tr>
<tr>
<td>জামা মছদজি শ্বািজািাবন দনমমাণ কদৰদছল।</td>
<td>Jama Masjid was built by Shahjahan.</td>
<td>Jama Masjid was built by Shahjahan.</td>
</tr>
<tr>
<td>দিৰি ভারতের বলদী।</td>
<td>Delhi is capital of India.</td>
<td>Delhi is the capital of India.</td>
</tr>
<tr>
<td>on bank of Ganges river situated Ujjain India of #seven sacred cities of inside #one of the.</td>
<td>Situated near the banks of the river Ganga, Ujjain is one of the seven sacred cities of India.</td>
<td></td>
</tr>
<tr>
<td>মুমতাজ মহালের পত্নীতা।</td>
<td>Mumtaz Mahal was wife of Shahjahan.</td>
<td>Mumtaz Mahal was the wife of Shahjahan.</td>
</tr>
<tr>
<td>অন্ধ্ৱপ্ৰবিশভাৰতের ঐতিহাসিক।</td>
<td>Andhra Pradesh is a state of India.</td>
<td>Andhra Pradesh is a state of India.</td>
</tr>
</tbody>
</table>

**Table VII English to Assamese Translation**

<table>
<thead>
<tr>
<th>English</th>
<th>Assamese</th>
<th>Correct Assamese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delhi is the capital of India.</td>
<td>ভারতের রাজধানী।</td>
<td>ভারতের রাজধানী।</td>
</tr>
<tr>
<td>মুমতাজ মহালের পত্নীতা।</td>
<td>তাজমহলের পত্নীর।</td>
<td>Mumtaz Mahal was the wife of Shahjahan.</td>
</tr>
</tbody>
</table>

---

**Dictionary**

<table>
<thead>
<tr>
<th>Alphabet</th>
</tr>
</thead>
<tbody>
<tr>
<td>অপৰাধ</td>
</tr>
<tr>
<td>অলংকার আড়ত শ্বািজািাবন দনমমাণ কদৰদছল তাজমিল আগ্রাত অৱদস্থত।</td>
</tr>
<tr>
<td>জামা মছদজি শ্বািজািাবন দনমমাণ কদৰদছল জামা মছদজি শ্বািজািাবন মিনার কফিলিন</td>
</tr>
</tbody>
</table>
Assam is a beautiful place.  
Kanak Vrindavan is a popular picnic spot in Jaipur.  
Hyderabad is situated in Bangalore.  
Udaipur is a famous city of Rajasthan.  
Canada is a vast country.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>WER</th>
<th>PWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tourism data</td>
<td>53.80%</td>
<td>36.96%</td>
</tr>
</tbody>
</table>

The scores are not amusing enough, which indicates that our system is still not ready for use in general. Work is still going on the improvement and addition of more transfer rules.

VI. Conclusion

We have tried to implement a bidirectional RBMT system between Assamese and English language. The results obtained from our work are encouraging but not satisfactory. We are working on the transfer rules and increasing the coverage of words in our dictionary. In the future we are looking forward to improve the translation quality of our system and make it a full-fledged MT system. The long-term plan of our project is also to make our system hybrid, taking the good qualities of both RBMT and SMT systems and see if it gives a better translation output.

References


Acknowledgement

The authors are thankful to the Department of Information Technology, Gauhati University for providing us the corpus, which helped us in making the dictionaries and building the RBMT system with Apertium.