Hindi to English Transfer Based Machine Translation System

Shashi Pal Singh, Ajai Kumar, Akanksha Gehlot* and Vaishali Sharma
AAIG, Centre for development of advanced computing, Pune India

Received: 06-April-2015; Revised: 08-June-2015; Accepted: 09-June-2015
©2015 ACCENTS

Abstract

In large societies like India there is a huge demand to convert one human language into another. Lots of work has been done in this area. Many transfer based MTS have developed for English to other languages, as MANTRA CDAC Pune, MATRA CDAC Pune, SHAKTI IISc Bangalore and HIT Hyderabad. Still there is a little work done for Hindi to other languages. Currently we are working on it. In this paper we focus on designing a system, that translate the document from Hindi to English by using transfer based approach. This system takes an input text check its structure through parsing. Reordering rules are used to generate the text in target language. It is better than Corpus Based MTS because Corpus Based MTS require large amount of word aligned data for translation that is not available for many languages while Transfer Based MTS requires only knowledge of both the languages (source language and target language) to make transfer rules. We get correct translation for simple assertive sentences and almost correct for complex and compound sentences.

Keywords

Parsing, transfer rules, table filling algorithm, transliteration, CFG, Part of Speech, Reordering rules.

1. Introduction

A Machine Translation system essentially takes a text in one language (Source Language) and converts it into another language (Target Language). Source and target languages are natural languages as Hindi, English, as opposed to manmade languages as C or SQL. There are various approaches of machine translation as dictionary based and rule based.

In Dictionary Based Machine Translation, translation is based on entries of a language dictionary. The word’s equivalent is used to develop the translation. Rule Based Machine Translation deals with morphological, syntactic and semantic information of source language and target language. It uses bilingual dictionaries and grammar rules. It has various approaches as Direct Approach, Transfer Based, Knowledge Based, Corpus Based, Context Based and Example Based. Here we are using Transfer Based approach in our project.

In Transfer Based System a source language is transformed into an abstract representation then its equivalent is generated for target language using bilingual dictionaries and grammar rules. This system has three major components: Analysis, Transfer and Synthesis.

Analysis of the source text is based on linguistic information such as morphology, Part of speech, syntax, semantic, etc. Algorithms are applied to parse the source language and derive

- The syntactic structure of the text to be Translated
- The semantic structure

Transfer module; transfer the source language structure into its equivalent target language structure. Some arrangement rules are used by which source language syntax structure can be transformed into target language syntax structure. Set of transfer rules are used for Hindi to English machine translation systems are [4]:

<table>
<thead>
<tr>
<th>Hindi Structure</th>
<th>English Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>S -&gt; NP VP</td>
<td>S -&gt; NP VP</td>
</tr>
<tr>
<td>NP -&gt; PRON NOUN</td>
<td>NP -&gt; PRON NOUN</td>
</tr>
<tr>
<td>VP -&gt; ADV Y</td>
<td>VP -&gt; AUX Y</td>
</tr>
<tr>
<td>Y -&gt; ADJ AUX</td>
<td>Y -&gt; ADV ADJ</td>
</tr>
</tbody>
</table>

Generation module: generate target language text using target language structure.

*Author for correspondence
Syntactic and lexical ambiguities can be resolved better than direct translation approaches. Example:

Input: यह किताब बहुत अच्छी है
Output: This book is very good

### Fig 1: Structural transfer in Hindi-English MTS

We have used two transfer rules. First transform VP- > ADV Y into VP->AUX Y. Second, transform Y->ADJ AUX into Y->ADV ADJ.

Transfer system is more realistic and flexible to meet the needs of different level of syntactic and semantic analysis.

### 2. Related Work [5]

#### Table 1: Existing transfer based MTS

<table>
<thead>
<tr>
<th>Machine Translation System</th>
<th>Developer</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANTRA-English to Hindi, Telugu, Gujarati</td>
<td>CDAC PUNE</td>
<td>1995</td>
</tr>
<tr>
<td>MATRA-English to Hindi MTS</td>
<td>CDAC PUNE</td>
<td>2004</td>
</tr>
<tr>
<td>SHAKTI-English to Hindi,Marathi,Telugu MTS (combines rule based and statistical approach)</td>
<td>IISc Bangalore and IIIT Hyderabad</td>
<td>2004</td>
</tr>
<tr>
<td>Anubaad-English-Bengali MTS(n-gram approach for pos)</td>
<td>CDAC Kolkata</td>
<td>2005</td>
</tr>
</tbody>
</table>
2.1 Mantra MT System
MANTRA (Machine assisted translation tool) translates the document from English to Hindi in a specified domain of personal administration, office order, office memorandums and circulars. Mantra uses Tree Adjoining Grammar for parsing and generation. It takes input in the form of text or output of speech recognition program [6].

2.2 MATRA MT System
MATRA, human-aided transfer based translation system for English to Hindi. The system has been applied mainly in the domains of news, technical phrases and annual reports [6].

2.3 SHAKTI MT System
This system works on three languages as Hindi, Marathi, and Telugu. If the user is not satisfied with the translation then it ask for meaning of input text components as word, phrases and even sentence level from user and retranslate it [6].

2.4 English to Kannada MT System
This system is based on Transfer Based approach and has been applied in the domain of government circulars. Now efforts are on applying it on English to Telugu [6].

3. Structural Differences between Hindi and English sentences
In Hindi sentences object appears before verb. It follows SOV structure while in English sentences object appears after verb it follows SVO structure [2].

<table>
<thead>
<tr>
<th>Tagging</th>
<th>University of Hyderabad</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>English to Kannada MTS using UCSG</td>
<td>University of Hyderabad</td>
<td>2006</td>
</tr>
<tr>
<td>English – Malayalam MTS</td>
<td>Amrita Institute of Technology</td>
<td>2009</td>
</tr>
</tbody>
</table>

Input: जवाहर लाल नेहरू भारत के प्रथम प्रधानमंत्री थे
Output: Jawahar Lal Nehru was first prime minister of India.

Adjectives appear before noun in both the languages as Hindi and English.

Input: वह बुद्धिमान लड़की है
Output: She is intelligent girl

Adverb appears before verb in Hindi sentences while in English sentences, it appears after verb.

Input: मोहन तेज दौड़ता है
Output: Mohan runs fast

Auxiliary verb appears after main verb in Hindi structure while in English structure it appears before main verb.

Input: रिया काफी पी रही है
Output: Riya is drinking coffee

4. Methodology
For parsing we have used CYK algorithm. It follows bottom up approach. It works on context free grammar. Context Free Grammar allows the construction of efficient parsing algorithm. Context Free Grammar is defined by 4 tuples.

\[ G = (V, T, P, S) \]
Where,
\[ V \rightarrow \text{set of variables} \]
\[ T \rightarrow \text{set of terminals} \]
\[ P \rightarrow \text{production rule} \]
\[ S \rightarrow \text{Start Symbol} \]

For CYK algorithm, context free grammar is converted into Chomsky Normal Form. This algorithm rules used in this algorithm are in the form

\[ A \rightarrow BC, B \rightarrow b, \text{ and } C \rightarrow c. \]

4.1 Cocke Younger Kasami Algorithm
CYK (Cocke Younger Kasami) Algorithm is a parallel dynamic syntax analysis algorithm. A two dimension matrix \( Z(p, q) \).

Among it: each element \( Z(p, q) \) according to non-terminal set of all possible forms of phrase with certain span of input sentences. [1]

Abscissa \( p \): position of the first word on the left of the span.

Ordinate \( q \): number of word span contained. For \( p = 1 \) to \( n \) (filling in the first row length is \( p \))
1) For every rule A->Rᵢ, adding the non-terminals A to set Z (m, m);

2) For q=2 to n (filling in other rows, length is q )
For p=1….q -n+1 (for all start point p)
For K=1…… q-1(for the length of all left child
cnodes is k)

Basic ideas table filling algorithm
Two dimensional matrix can be used to encode the
structure of a tree. Each cell [p, q] in this matrix
contain a set of non-terminal that represent all the
content that span position p through q of the input.
Cell that represent entire input reside in position [1, n] in the matrix. Since grammar used is in CNF, non-
terminal entries in the table have exactly two child in
parse [1].

For each content represented by an entry [ p, q] in
the table there must be a position in input k, where it
can be split into two parts as p<k<q. First content [p, k]
must lie to the left of the entry[p, q] along row p
second entry [k, q] must lie beneath it, along column
q.

\[
\begin{array}{cccc}
Z(1,5) & & & \\
Z(1,4) & Z(2,5) & & \\
Z(1,3) & Z(2,4) & Z(3,5) & \\
Z(1,2) & Z(2,3) & Z(3,4) & Z(4,5) \\
Z(1,1) & Z(2,2) & Z(3,3) & Z(4,4) & Z(5,5)
\end{array}
\]

Fig 2: Structure of table in table filling algorithm

\[
\begin{array}{ccc}
S & VP & C \\
& NP & VP \\
& PRON & NOUN & VP & AUX \\
& & \text{वह} & \text{बाजार} & \text{जाती} & \text{है}
\end{array}
\]

Fig 3: Example of CYK Algorithm

4.2 System Architecture
4.2.1 Pre-processing
In pre-processing phase, numbers of operation are
applied to input data to make it executable by
translation system. It includes treatment of
punctuation and special characters.

4.2.2 Tokenizer
Tokenizer, which is also known as lexical analyzer or
word segmenter segment a sentence into meaningful
unit known as tokens. Tokenizer takes output of pre-
processing phase as an input. Individual units or
tokens are extracted and processed to find its
corresponding meaning in target language. This
module uses space and punctuation mark as
delimiter, extract token one by one from input text
and give it to translation system till the input text is
processed.

Input: अब्दुल कलाम महान वैज्ञानिक है
Output: Token1 अब्दुल
Token2 कलाम
4.2.3 Hindi Morphological Analyzer
Morphological analyzer identifies the root word and its corresponding category for tokens that are generated by Tokenizer [3].

Input: सीता बहुत अच्छी लड़की है
Output:
सीता NOUN
बहुत ADV
अच्छी ADJ
लड़की NOUN
है AUX

4.2.4 Transliteration
For proper noun, we use transliteration instead of translation. Words or tokens that are not found in our database are transliterated.

Input: अजय
Output: ajay

For transliteration we maintain a table in database, containing Hindi letters and matras. In above example
अ -अज ज-याय -या
If character at the end of word is not equal to “ा” then delete ‘ा’ from its English meaning. English meaning of ‘य’ is ‘y’ only.

Input: रामा
Output: rama

र-रा म-मा
Input: मनाली
Output: manalee

म-मा न-नाल-ला ,ी-ее here ल्गिवेस I , ' a’ is deleted.

4.2.5 Dealing with Replicative Noun:

Input: बच्चा बच्चा गांधी जी को जानता है
Output: Every child knows Gandhi ji

4.2.6 Identify singular and plurals: [3]
If (first word== “मैं”|“तुम”|“वे”|”हम”) Then Word is plural
If (last letter of first word== “या”|“वे”|”ये”) Then word is plural
Otherwise it is considered as singular

4.2.7 Dealing Interrogative Sentences:
If sentence ends with “ता”, “ती”, “ते” Then it is considered as Present indefinite. Word next to “क्या” is checked to identify the number whether it is singular or plural.

Input क्या सीता खाती है
Output Does Sita eats food

In the above example “सीता” is taken as singular so क्या is translated as does.

Input क्या तुम प्रत्मेि भॊगरवाय पुटफॉर खेरते हो
Output Do you play football every Tuesday?

In the above example “तुम” is taken as plural so क्या is translated as do.

if sentence ends with “रहा ”, “रही”, ”रहे” Then it is considered as continuous sentences. Word next to “क्या” is checked to identify the number whether it is singular or plural.

Input क्या तुम पढ़ रहे हो
Output: are you reading?

Input क्या सीता आजार जा रही है
Output is Sita going to market?

4.2.8 Dealing with synonyms:

Input: कृपया ध्यान दे
Output: please pay attention

Meditation
In bilingual dictionary ध्यान is translated as attention and meditation. In our system these synonyms can be handled at runtime.

**4.2.9 Dealing with sentences where exact matching rules are not found:**

In our system we split the sentences by using “|”. Here we get three sentences.

Sentence1: राम, मोहन और श्याम दोस्त हैं।
Sentence2: राम पुणे में रहता है।
Sentence3: मोहन और श्याम मुंबई में रहते हैं।

For sentence 1 its corresponding target equivalent structure NOUN NOUN CONJ NOUN AUX NOUN is found in database so it is translated as

Sentence 1: Ram, Mohan and Shyam are friend.

For sentence 2, its equivalent target structure is not found in our database. In this case equivalent target structure is searched at phrase level. At phrase level, it is matched with a structure NOUN PREP VERB and its equivalent i.e. VERB PREP NOUN is found in our database. Here it is translated at phrase level.

राम \ पुणे में रहता है।
Ram lives in Pune

Similarly for sentence 3, its equivalent target structure is not found in our database then it is searched at phrase level.

मोहन और श्याम \ मुंबई में रहते हैं।
Mohan and Shyam \ lives in Mumbai

Phrases for which any equivalent target structure is not found, word to word translation is used for them.

Fig 4: Flow chart of Transfer Based Machine Translation System

**5. Conclusion**

In this paper, we have discussed a transfer based machine translation which is based on rule based approach of machine translation. We have incorporated almost all possible translation rules for different Hindi structures. As a future work we can handle the sentences consisting of idioms, complex, compound and other form of sentences in perfect and
perfect continuous tenses. We can enhance dictionary to improve translation quality.

References


