ENGLISH TO SANSKRT MACHINE TRANSLATOR
LEXICAL PARSER AND SEMANTIC MAPPER

Ms. Vaishali M. Barkade*, Prof. Prakash R. Devale*, Dr. Suhas H. Patil*

*Information Technology Department, Information Technology Department, Computer Department, Bharati Vidyapeeth University College of Engineering Pune-43, Maharashtra, India

Abstract—Here we propose to develop a converter which converts English Sentence to Sanskrit sentence. The Proposed modules are as follows:

MODULE 1: LEXICAL PARSER
MODULE 2: SEMANTIC MAPPER
MODULE 3: ITTRANSLATOR
MODULE 4: COMPOSER

Here we would concentrate only on the first two modules. The first module i.e. Lexical parser which parses an English sentence and the second module i.e. Semantic Mapper which maps the English semantic word with Sanskrit semantic word.

Keywords: machine translation, lexical parser, word order, grammar, tree, rule based, semantic mapper.

I. INTRODUCTION

Machine Translation has been defined as the process that utilizes computer software to translate text from one natural language to another. It is one of the most important applications of Natural Language Processing. It helps people from different places to understand an unknown language without the aid of a human translator. The language to be translated is the Source Language (SL). The language to which source language translated is Target Language (TL). The major machine translation techniques are Rule Based Machine Translation Technique [1], Statistical Machine Translation Technique (SMT) and Example-based machine translation (EBMT). One of the effective techniques for machine translation is Rule Based Machine Translation. In India, different machine translation systems are implemented. AnglaUrdu (AnglaHindi based) Machine Translation System for English to Urdu [2], HindiAngla Machine Translation Systems form Hindi to English, English-Assamese Machine Translation System (Machine Translation System from English to Assamese, MaTra: Human Aided Machine Translation System, AnglaHindi: An English to Hindi Machine-Aided Translation System [3] and AnglaBharti Technology for machine aided translation from English to Indian Languages[4], these are some of the machine translation works implemented in India. Here we are describing about Machine Translation Technique for translating English sentence to Sanskrit sentence. English is a well known language and Sanskrit is an ancient language. Machine translation in Sanskrit is never an easy task because of structural vastness of its grammar but the grammar is well organized and least ambiguous compared to other natural language. The proposed methodology uses a Rule based parser. The English sentence which is the input for our first module i.e. lexical Parser it generates a Parse tree that is generated by using semantic relationships. This parse tree acts as an input to the Second module i.e. Semantic mapper where the English semantic word is mapped to the Sanskrit semantic word (Sanskrit word in English).

II. APPROACH USED: RULE BASED MACHINE TRANSLATION

Major approaches of Machine Translation are rule-based machine translation (RBMT, also known as the Rational approach).

Rule based translation consists of

1. Process of analysing input sentence of a source language syntactically and or semantically
2. Process of generating output sentence of a target language based on internal structure each process is controlled by the dictionary and the rules.
The strength of the rule based method is that the information can be obtained through introspection and analysis.

The weakness of the rule based method is the accuracy of entire process is the product of the accuracies of each sub stage.

III. PARSER

A parser breaks data into smaller elements, according to a set of rules that describe its structure. Parsing is the process of analyzing a text, made of a sequence of tokens (for example, words), to determine its grammatical structure with respect to a given grammar.

Following are the Steps to generate a Parse Tree

Step 1: Input is a English sentence.
Step 2: Lexical Analyzer
   - Creates Tokens
Step 3: Tokens generated acts as an input to Semantic analyzer
Step 4: Semantic analyzer
   - Creates a parse tree
Step 5: Output is a parse tree

![Figure 1 Stages of creating a Parse Tree](image)

IV. LEXICAL PARSER

The semantic standard representation was designed to provide a simple description of the grammatical relationships in a sentence that can easily be understood and effectively used by people without linguistic expertise who want to extract textual relations. The sentence relationships are represented uniformly as semantic standard relations between pairs of words. For the sentence: Bell, based in Los Angeles, makes and distributes electronic, computer and building products.

A. The Semantic representation is:

<table>
<thead>
<tr>
<th>Relation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>nsubj (makes-8, Bell-1)</td>
<td></td>
</tr>
<tr>
<td>nsubj (distributes-10, Bell-1)</td>
<td></td>
</tr>
<tr>
<td>partmod (Bell-1, based-3)</td>
<td></td>
</tr>
<tr>
<td>nn (Angeles-6, Los-5)</td>
<td></td>
</tr>
<tr>
<td>prep in (based-3, Angeles-6)</td>
<td></td>
</tr>
<tr>
<td>conj and (makes-8, distributes-10)</td>
<td></td>
</tr>
<tr>
<td>amod (products-16, electronic-11)</td>
<td></td>
</tr>
<tr>
<td>conj and (electronic-11, computer-13)</td>
<td></td>
</tr>
<tr>
<td>amod (products-16, computer-13)</td>
<td></td>
</tr>
<tr>
<td>conj and (electronic-11, building-15)</td>
<td></td>
</tr>
<tr>
<td>amod (products-16, building-15)</td>
<td></td>
</tr>
<tr>
<td>dobj (makes-8, products-16)</td>
<td></td>
</tr>
<tr>
<td>dobj (distributes-10, products-16)</td>
<td></td>
</tr>
</tbody>
</table>

The above relations maps straightforwardly onto a directed graph representation, in which

- NODES - words in the sentence
- EDGES - grammatical relations are edge labels.

Figure 2 gives the graph representation for the example sentence above.
B. Algorithm for Parsing an English sentence

Step 1: Tokenize the sentence into various tokens i.e. token list

Step 2: To find the relationship between tokens we are using dependency grammar and binary relation for our English language. Token list acts as an input to semantic class to represent the semantic standard.

Step 3: Semantic class generates a tree we have a class Tree Transform which will create a tree.

Step 4: Sentence is splitted into words that are nouns, verbs etc.

V. COMPARISON OF ENGLISH AND SANSKRIT WORD ORDER

English is well known language and Sanskrit is an ancient language [5]. The English sentence always has an order of Subject-Verb-Object, while Sanskrit sentence has a free word order. A free order language is a natural language which does not lead to any absurdity or ambiguity, thereby maintaining a grammatical and semantic meaning for every sentence obtained by the change in the ordering of the words in the original sentence. For example, the order of English sentence (ES) and its equivalent translation in Sanskrit sentence (SS) is given as below.

ES: Ram reads book.
(Subject) (Verb) (Object)

SS: Raamah pustakam pathati.
(Subject) (Object) (Verb)

or
Pustakam raamah pathati.
(Object) (Subject) (Verb)

or
Pathati pustakam raamah
(Verb) (Object) (Subject)

Thus Sanskrit sentence can be written using SVO, SOV and VOS order.

A. The comparative view of English and Sanskrit on different basis is as below in table I.

<table>
<thead>
<tr>
<th>Basis</th>
<th>English</th>
<th>Sanskrit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabet</td>
<td>26 character</td>
<td>42 character</td>
</tr>
<tr>
<td>Number of vowel</td>
<td>Five vowels</td>
<td>Nine vowels</td>
</tr>
<tr>
<td>Number of consonant</td>
<td>Twenty one</td>
<td>Thirty consonant</td>
</tr>
<tr>
<td>Number of singular and plural</td>
<td>consonant</td>
<td>Three: singular, dual and plural</td>
</tr>
<tr>
<td>Sentence Order</td>
<td>SVO(Subject-Verb-Object)</td>
<td>Free word order</td>
</tr>
</tbody>
</table>
Tenses | Three: present, past and future | Six: present, aorist, imperfect, perfect. 1st future and

Mood | Five: indicative, imperative, interrogative, conditional | Four: imperative, potential, benedictive and conditional

Table 1
Comparative views of English and Sanskrit

B. Word based Matching

The matching is not only word by word but it will be semantic (meaningful) matching based on the relationship been established.

(1) (a) ES: A man eats vegetables.
SS: Narah shaakam khaadati.
   (A) (man) (vegetables) (eats)

(b) ES: Acids eats metal.
SS: Aambat dhautum nashyati.
   (Acids) (metal) (eats)

(2) (a) ES: He eats potatoes.
SS: Sah sukantham khaadati.
   (He) (potatoes) (eats)

(b) ES: Sulphuric acid eats iron.
SS: Gandhak lauham nashyati.
   (Sulphuric acid) (iron) (eats)

C. Algorithm for Semantic mapper

Step 1: The output from the first module i.e. Lexical parser acts as input to the semantic Mapper

Step 2: The tokens generated from the first module are stored in Data Structure i.e. Collection. These tokens have grammatical relations which are represented with various Symbols e.g. conj, nn, nsubj, det, dobj etc.

Step 3: Look up in Sanskrit dictionary we are matching English semantic word with the dictionary Sanskrit word. This matching is not word by word but it will be semantic (meaningful) matching based on the relationship been established.

Step 4: After matching the selected words from the Sanskrit dictionary are kept as another data structure.

Step 5: Identify the relationships among the various Sanskrit words from these Data Structures.

D. Basic Writing Rules

1. Keep sentences short
2. Make sure sentences are grammatical
3. Avoid complicated grammatical constructions
4. Avoid words which has several meaning
VI. WORK DONE BASED ON THIS

In the first module we have design our own lexical parser for getting the typed dependency information POS tag information and context-free phrase structure grammar representation of source structure. Translation based on above method is implemented with the help of Java codes. The nouns, verb, subject etc are stored in a Data structure i.e. Collection (Array List). This parser is used at lower level of our application.

The sentence:
Bell, based in Los Angeles, makes and distributes electronic, computer and building products.
is broken into tokens.

<table>
<thead>
<tr>
<th>Noun (nsubj)</th>
<th>Token 1</th>
<th>Bell</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participial</td>
<td>Token 2</td>
<td>based</td>
</tr>
<tr>
<td>modifier (partmod)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preposition (prep)</td>
<td>Token 3</td>
<td>in</td>
</tr>
<tr>
<td>Noun (nn)</td>
<td>Token 4</td>
<td>Los</td>
</tr>
<tr>
<td>Noun (nn)</td>
<td>Token 5</td>
<td>Angeles</td>
</tr>
<tr>
<td>Verb</td>
<td>Token 6</td>
<td>makes</td>
</tr>
</tbody>
</table>

Conjunction (conj) | Token 7 | and |
Verb              | Token 8 | distributes |
Adjective (amod)  | Token 9 | electronic |
Adjective(amod)   | Token 10 | computer |
Conjunction (conj) | Token 11 | and  |
Adjective (amod)  | Token 12 | building |
Directobject (dobj) | Token 13 | products |

Table 2 Shows how a sentence is broken into tokens

In the second module Semantic mapper we have to look up in Sanskrit dictionary in this we are mapping English word with the Sanskrit word i.e. Sanskrit word in English semantically (meaningful). The mapping is based on the relationship been established in the first module i.e. Lexical Parser.

Different Rules are considered for Mapping:

1. Equality Rule
   English word directly mapping to Sanskrit word
   e.g.
   ES : A man eats vegetables.
   SS : Naraḥ saakam khaadati.
   (A) (man) (vegetables) (eats)

2. Synonyms Rule (word having same meaning)
   English words mapping to Sanskrit word
   e.g.
   ES : He is a good /fine/excellent man
   SS : Sah sajjana

3. Antonyms Rule (word having opposite meaning)
   English word not directly mapping to Sanskrit word
   e.g.
   ES : He is not good man
   He is a bad man
   SS : Sah Durjana
VII. CONCLUSION

Here more emphasis is on Module 1 LEXICAL PARSER and Module 2 SEMANTIC MAPPER. It shows how an English statement is parsed into tokens and then finds the relationship between tokens using dependency grammar and by using the semantic representation we generate a tree then mapping of English word to Sanskrit (Sanskrit in English) word that to semantically is done

REFERENCES


