Transmuter: An Approach to Rule-based English to Marathi Machine Translation

G V Garje
Pune Vidyarthi Girh's COET,
Pune, India

G K Kharate
Matoshri COE and Research
Center, Eklahare, Nashik

Harshad Kulkarni
3dplm Software Solutions Ltd,
Pune, India

ABSTRACT

This paper describes the architecture of a Machine Translation System with source language as English and target language as Marathi. The basic approach used for the development of this system is Rule Based Machine Translation. The basic algorithm for obtaining the correct word order in the target language was developed based on specific traversals of the parse tree. One of the special features of the system is a Word Sense Disambiguation model. Presently only prepositions will be disambiguated and work is going on for verbs and nouns. The model is a generalized approach based on the categories/domains a word belongs to. Another feature is the target language generation module. The focus is on the grammar structure of the target language that will produce better and smoother translations. The architecture though developed specifically for English – Marathi language pair, may be extended to other language pairs with similar structure. The architecture is partially implemented in the form of Machine Translation system. A lexicon is built for morphological and semantic properties. The results, even at partial implementation stage, are really encouraging.

General Terms

Artificial intelligence, Natural Language Processing, Grammar, Source language, Target language, inflections

Keywords

Machine Translation, Word Sense Disambiguation, Parser, Transliteration, Marathi, Case-suffixes

1. INTRODUCTION

Machine Translation (MT) has always been a dream of Computer Scientists. Due to large variations in the language structures, this dream is still away from reality. The variation in languages ranges from entirely different grammatical structure in different language families to very minute differences in grammar rules in closely related families. These subtleties make machine translation a challenging problem for both computer scientists and linguists. Development of a machine translation system has been approached in many ways in the past. Rule-based Machine translation, Statistical machine Translation, Example based Machine Translation are the major approaches.

This paper deals with a machine translation system with source language as English and target language as Marathi. Marathi belongs to the family of Indian languages, which originate from Sanskrit. It is spoken mainly in the central – western part of India and 68 million people speak Marathi in India. Grammatically the sentence structure of Marathi is Subject - Object - Verb (S-O-V) whereas English is Subject - Verb - Object (S-V-O) [1]. Further, the language is highly dominated by inflections and case-suffixes. Syntactically, the script for Marathi language is Devanagari, similar to that of Sanskrit or Hindi.

Being a low resource language, Marathi has not been worked upon heavily by the Computer Linguistic community.

Anuvadaksha [2], developed by the Technology Development of Indian Languages (TDIL), and Saakava [3] are the tools available on the World Wide Web, for English to Marathi machine translation. The work on these tools is still in progress.

Though, Google Translate, Bing Translate do not perform translations from English to Marathi, it works with Hindi, Punjabi which are closer in structure to Marathi.

In this paper, we propose architecture for a Machine translation system from English to Marathi. The focus of the architecture is on the following points:

- Rearrangement of the sentence structure
- Word Sense disambiguation approach
- Inflections in Marathi
- Addition of case-suffixes, postpositions to various words after translation

2. SYSTEM ARCHITECTURE

Figure 1 depicts the overall architecture of the proposed MT system. The details of the various components of architecture of the system are as below.

2.1 Pre-translation processor

In this component the input sentence is analyzed according to grammatical structure of the source language and made fit for further word to word processing.

2.2 Parser

Initially the sentence is parsed using an English Language Parser. Here a grammar based tree structure of the English sentence is obtained as shown in figure 2. The words in the tree structure are tagged according to their parts of speech. The dependencies between words of the sentence are also obtained in this phase of translation process [4]. The Stanford parser [5] is used for analyzing source language i.e. English.

Example

Rhinoceros can be seen closely in the Pavitra Sancturay

(S (NP (NNS Rhinoceros)) (VP (MD can) (VP (VB be) (VP (VBN seen) (ADVP (RB closely)) (PP (IN in) (NP (DT the) (NNP Pavitra) (NNP Sanctuary)))))))

2.3 Named Entity Tagger

Further, the named entities in the sentence are recognized using a Named Entity Tagger so that the categories of all the words are defined properly. The words can thus be segregated into persons, locations, time, etc. using these categories.

Example

(S (NP (NNS Rhinoceros/1/O)) (VP (MD can/2/O) (VP (VB be/3/O) (VP (VBN seen/4/O) (ADVP (RB closely/5/O)) (PP (IN in/6/O) (NP (DT the/7/O) (NNP Pavitra/8/LOCATION) (NNP Sanctuary/9/LOCATION)))))))

2.4 Rearrangement Generator

The tree structure is then traversed in a specific way to obtain a proper sequence of words in target language (Marathi) based on its grammar. The traversal takes place in two stages. In the first stage, at the root the traversal is *post-order* traversal. In the second stage, the sub-trees of the root are first *mirrored*. The further traversal is based on the type of current node and its children as shown in figure 3.

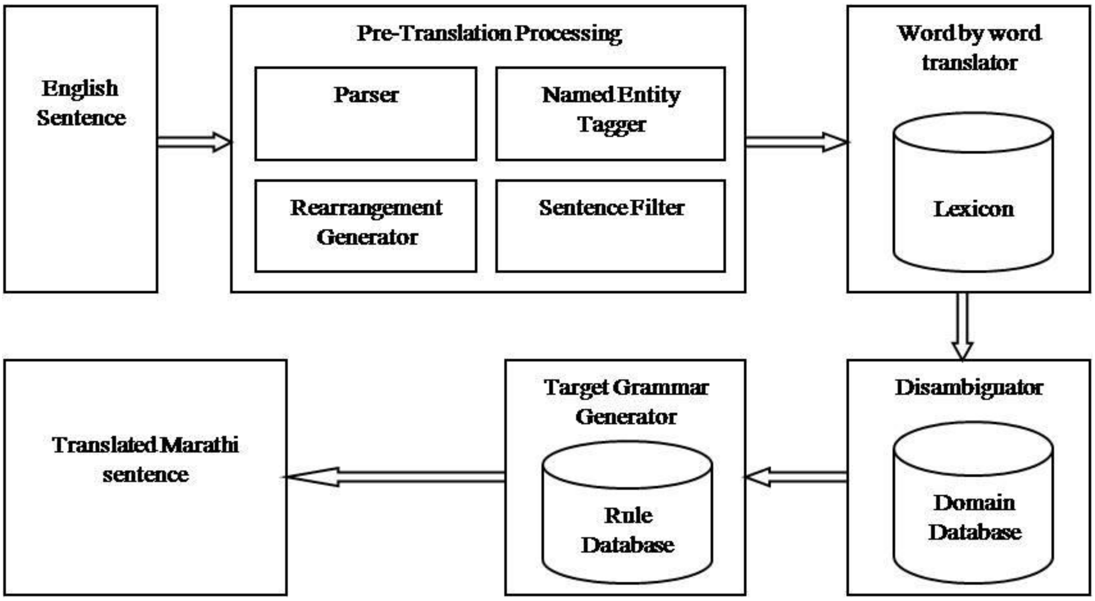

Figure 1. Overall System Architecture

For a

- NP node : in-order
- PP node : in-order
- VP node : post-order
- ADJP : in-order
- ADVP : in-order

Example
The simple assertive sentence taken is- *"The atmosphere of the city gives pleasure to tourists"*

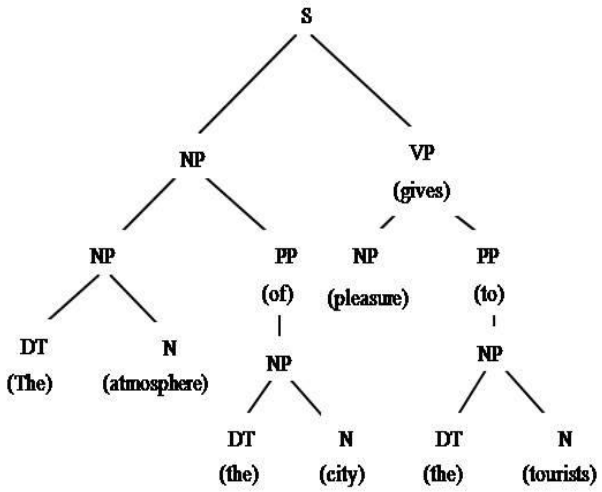

Figure 2. Parse Tree

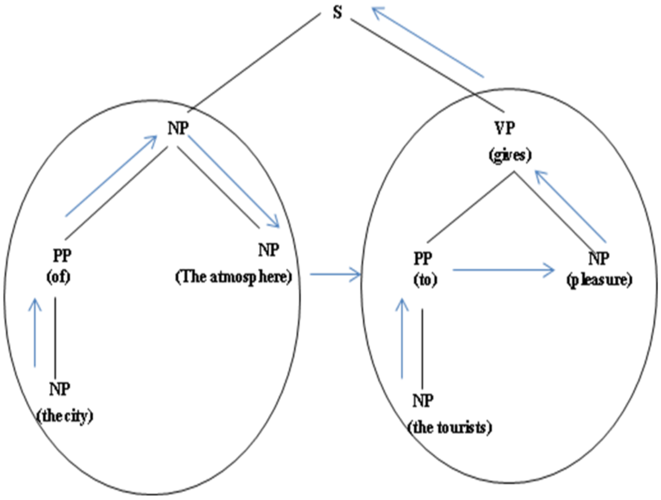

Figure 3. Rearrangement of words

The traversal of the tress shown in figure 3 will produce the following sequence of words which maps to target language grammar structure.

[The city] of [the atmosphere] [the tourists] to pleasure gives

शहराचे वातावरण यात्रेकरू ना/ला आनंद देते

2.5 Filter

A few adjustments have to be made for achieving accuracy in the Marathi structure. For example, most of the times, articles do not play a major role as far as meaning of the sentence is concerned in the Marathi sentence structure and thus may be removed or it can be grouped with next word in the sentence and meaning of the group is taken. Such changes are made by the Filter.

2.6 Word by Word Translator

This component includes a bilingual lexicon of English and corresponding Marathi words. A sentence of source language from the pre-translation process is split into words. For each word its parallel Marathi meanings are obtained from the lexicon.

The lexicon also contains various grammatical attributes of the Marathi word which are then assigned according to nature of the sentence. The attributes include base forms for noun inflections, base forms of verbs for conjugations, etc.

2.7 Disambiguator

Some words of the sentence obtained from the *word by word Translator* possess ambiguities due to presence of multiple meanings to single word. These ambiguities must be resolved to obtain better translations. Few algorithms have been proposed for Word Sense Disambiguation like Lesk Algorithm [6], Walker Algorithm [7], and HyperLex Algorithm [6].

We here propose a derived approach for ambiguity resolution. The resolution is based upon a set of pre-defined categories C. The set C may contain coarse categories (like PERSON, LOCATION, and ORGANIZATION) as well as finer categories (like HISTORICAL PERSON, LANDMARK PLACE, and FINANCIAL ORGANIZATION). Here the finer categories are considered to be subcategories of their respective coarse categories.

Let us consider the sentence as a set of 'n' words W = $\{w_0, w_1 \ldots w_{n-1}\}$.

Each w_i in W has a set of categories C_i such that $C_i \subseteq C$. All the categories $c_{ik} \in C_i$ are assigned strength s_{ik} which is the strength of the semantic connection between the word and the pre-defined category. This strength s_{ik} is determined by the common world knowledge. Another strategy for deciding the strength is based on the statistical analysis of an existing corpus.

Consider an ambiguous word $w_a \in W$. If c_{ak} is the k^{th} category, such that $c_{ak} \in C_a$ then, all such categories belonging to C_a, are the candidates of the election.

For all words $w_i \in W$ such that $i \neq a$, with category set C_i, w_i will vote for the candidate category $c_{ak} \in Ca$ if $c_{im} = c_{ak}$ or if c_{im} is a subcategory of c_{ak} where $c_{im} \in C_i$.

Each vote is weighted according to the strength s_{im} of the word w_i and the category c_{im}. Further, we need to consider another factor in finalizing the vote.

This factor is the grammatical relationship between w_a and w_i in the sentence under consideration. This consideration is important to ensure the fact that only the vote of the related words counts during word sense disambiguation.

Finally, the category c_{ak} of the word w_a, getting the maximum votes wins the election. In case of a tie at the maximum vote, the category c_{ak} having the higher strength s_{ak}, among the tied categories, wins the election. The sense belonging to this category is selected as the final sense for the ambiguous word.

2.8 Target Grammar Generator

This component is responsible for attaching case-suffixes to words whenever and wherever necessary. The case-suffixes are attached in case of three grammatical structures [8],[9]:

2.8.1 Karta-Karma Processing

Karta is the doer of the action described by the verb while *Karma* is the object. In Marathi, case suffixes (*vibhakti pratyay*) are attached to either *Karta* or *Karma* or both depending on the case of the sentence. Before attachment of any case suffix, the *Karta* and *Karma* have to undergo inflections. The Target Grammar Generator contains a database of rules which elaborate the use of appropriate case suffixes and necessary inflections.

Example

Ram gave Bheem a mango.

रामाने भीमाला आंबा दिला.

(doer) (receiver)

2.8.2 Verb Processing

Marathi sentences are categorized to be in three *prayog - kartari, karmani, bhave.* Each prayog has its own effective word. In *Kartari prayog* the verb is conjugated according to *Karta* (doer) while in *Karmani prayog* it is according to *Karma* (object). In *Bhave prayog* the verb conjugation is independent of *Karta* and *Karma*.

The Target Grammar Generator decides the prayog of the sentence and determines the suffix to be added to the base form according to the attributes of the effective word and overall tense of the sentence. The attributes of the effective word required for verb conjugation include gender, number and person. The Target Grammar Generator also handles the inflections of verb forms according to tense.

Example

i) A crowd of devotees engulfs Haridwar during the time of daily prayer in the evening

भक्तांची गर्दी संध्याकाळी दररोजच्या प्रार्थनेच्या वेळेदरम्यान हरिद्वारला घेरते

ii) One line has gone from Luxar to Haridwar

एक मार्ग लक्सरहून हरिद्वारला गेलेला आहे

2.8.3 Preposition Processing

Prepositions in English occur as postpositions in Marathi. Thus preposition processing involves attachment of suffix (appropriate postposition) to the base form of prepositional object of the preposition. Postposition also undergoes inflections according to the word about which the preposition is providing information.

Example

One line has gone from Luxar to Haridwar

एक मार्ग लक्सरहून हरिद्वारला गेलेला आहे

3. IMPLEMENTATION

The system architecture shown in figure 1 is a proposed architecture. We have implemented a system on a reduced scale.

The system takes simple assertive sentences as input. These sentences must be of the following regular expression structure:

((ADV)?(PP)*(NP)+(PP)*(ADV)?(V)*(ADV)?(V)*(ADV)?(PP)*(NP)*(PP)*) where ADV – Adverb

NP – Noun Phrase

PP – Prepositional Phrase

V – Verb

The implementation language used is JAVA. The output of the system is translated Marathi sentence. The Marathi sentence is represented in the Devanagri script using UNICODE.

3.1 Pre-Translation Generator

Pre-translation processing is done using standard tools available.

The parser used is Stanford Parser [10],[11], which is a Lexicalized PCFG parser. The parser provides Stanford Dependencies output as well as phrase structure trees.

For tagging the Named Entities, we have used Stanford Named Entity Recognizer [12]. In a 7 class model, it tags following categories: Person, Location, Time, Organization, Money, Percent, and Date.

3.2 Word by Word Generator

The bilingual Lexicon is built in a flat file format. It has a hierarchical structure. Each entry has corresponding Marathi words, and attributes. The lexicon is not a full-fledged dictionary. It contains around 2000 root word entries. Some groups of words like 'hundreds of' have a corresponding single Marathi word. Such groups of words are also included in the lexicon.

3.3 Disambiguator

In this system, we only handle ambiguities occurring in prepositions. The algorithm for disambiguation is based on the generalized algorithm mentioned above. In the scaled down version, the algorithm only requires the two connected words of the preposition to vote, in order to disambiguate the meaning of the preposition [13].

3.4 Target Grammar Generator

The system implements the Target Grammar Generator completely using Paninian Grammar approach. [8],[9].

4. TESTING AND RESULTS

The system is tested on a corpus of 1000 parallel sentences. This corpus, provided by the Technology Development for Indian Languages (TDIL) [2], contains sentences related to Indian Tourism domain. Along with the corpus, we also obtained translations of these 1000 sentences from some people proficient in English and Marathi. These translations form the reference of our testing.

As no standard is available for testing Marathi translations, the testing is done using BLEU scoring [14].

The system is also tested for 100 sentences against Saakava and Google Translator and results are shown in fig. 5.

Table 1. BLUE Scores

	Transmuter(%)	Anuvadaksha(%)
Corpora by TDIL	44.29	30.66
Human Translation	49.78	26.29

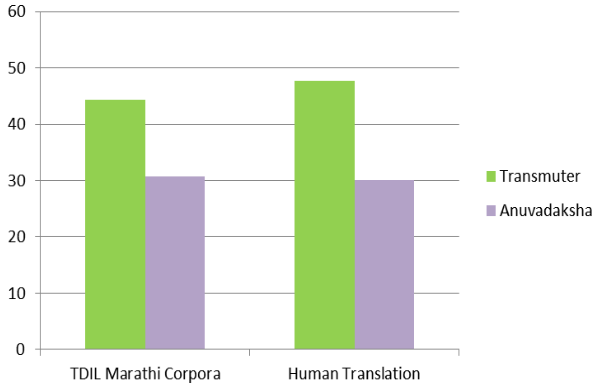

Figure 4. Performance copmarision of Transmuter

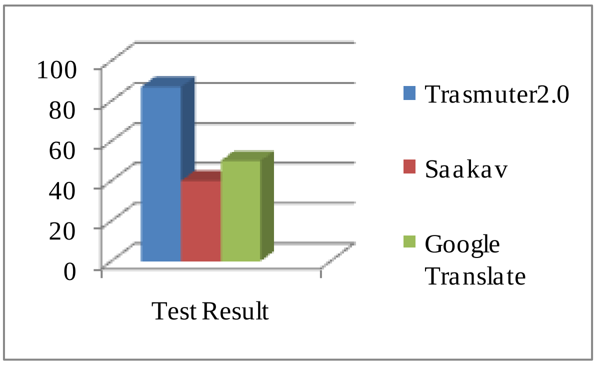

Figure 5. Performance comparison of Transmuter with Saakava and Google Translate

5. CONCLUSION

Rule Based Machine Translation is a tedious approach but it produces better quality language translations. The number of rules formed is large for target language generation to achieve batter quality translations. However, there exist many exceptions in the language which do not conform to these rules. The translation quality of this approach is dependent on the size of the grammar knowledge base. The exceptions can be handled but it will increase the size and the complexity of the knowledge base. Increase in the size of the knowledge base will improve the quality of translation up to a certain threshold. If the size crosses this threshold, the quality may reduce due to conflicting rules. We are able to achieve far better translation quality compared to existing systems such as Anuvadaksha, Saakava and Google translate for 1000 simple assertive sentences from tourism domain and 100 general simple assertive sentences we have tested.

This translation system can be extended for any domain by making necessary changes in rule-base. The quality can be improved by developing algorithms for disambiguation of verbs and nouns.

6. ACKNOWLEGEMENTS

We would also like to extend our thanks to Technology Development for Indian Languages (TDIL) and Stanford NLP group for the tools that we used. We also appreciate the support provided by Dr. Raymond and Mr. Chandrakant Dhutadmal from Center for Development in Advanced Computing (CDAC), Pune for sparing their valuable time and providing advice and support, and the people who helped us in manually translating the sentences and for testing our results of the system. We would also extend our sincere gratitude and thanks to Dr. M. L. Dhore, Vishwakarma

Institute of Technology, Pune for his valuable support and guidance.

7. REFERENCES

[1] Wren P. and Martin H. High School English Grammar and Composition. S Chand Publication

[2] Technology Development for Indian Languages, DIT, Government of India

Also available at: http://www.tdil-dc.in/index.php

[3] http://www.saakava.com

[4] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.

[5] http://www.nlp.**stanford**.edu/software/lex-**parser**.shtml

[6] Esha Palta. 2006-07. Word Sense Disambiguation. Master of Technology First Stage Report, IIT Bombay.

[7] Walker D. and Amsler R. 1986. The Use of Machine Readable Dictionaries in Sublanguage Analysis. In Analyzing Language in Restricted Domains, Grishman and Kittredge (eds), LEA Press, pp. 69-83

[8] Tarkhadkar Dwarakanath. Tarkhadkar Bhashantar Pathmala. Raj Prakashan, 1st edition

[9] Walimbe M. R. 2013. Sugam Marathi Vyakran Lekhan. Nitin Prakashan, revised edition

[10] Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

[11] Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In proceedings of LREC 2006.

[12] Malarkodi C.S, Pattabhi RK Rao and Sobha Lalitha Devi, 2012. Tamil NER – Coping with Real Time Challenges. In proceedings of Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012), 24th International Conference on Computer Linguistics

[13] Jayan V., Sunil R., Bhadran V. K. 2012. Disambiguation of pre/post positions in English-Malyalam Text Translation Proceedings of Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012), 24th International Conference on Computer Linguistics

[14] Papineni, K. Roukos, S. Ward, T.; Zhu, W. J., 2002. BLEU: a method for automatic evaluation of machine translation. ACL-2002: 40th Annual meeting of the Computational Linguistics. pp. 311–318.