Python implementation of a Parse Tree Model for Machine Translation for English to Marathi

Arnav Doifode and Akhilesh Wase
Undrer The guidence of Dr. Pooja Jain
Department of Computer Science and Engineering
Indian Institute of Information Technology

Abstract
This paper describes the model that is used for Machine Translation in Natural Language Processing. The approach this model takes is Rule Base Machine Translation also known as RBMT. This model takes simple assertive English sentences as input to give most probable Marathi equivalent sentences. The main ethos of this model is construction of Parse tree of the source language. After the construction there is rearrangement of the parse tree such that it is compatible with the target language's structure of the sentences.Generally English language contains Subject Verb Object (SVO) format while Marathi language is considered to be having Subject Object Verb(SOV) format. The model that this paper presents can be extended to several Dravidian languages which posses the same sentence structure as Marathi language sentences.

Keywords
Natural Language Processing, Rule Based Machine Translation, Parse Tree

Introduction

Humans have been using languages since 150,000 years to communicate ideas and thoughts . One of the oldest languages that humans have known is English Language. Since most of the people know how to speak English ,attempts have been made by scientist to convert English language to the language that is understood by the native people of various countries. One such attempt is to convert the English language to Marathi Language. After the evolution of neural networks, machines are becoming smarter in every aspect.Hence scientists are now trying to automate the process of communication. The subject which deals with this process is called as Natural Language Processing. Under this the conversion of source language to target language is known as Machine Translation.

There are various techniques that have been evolved for machine translation. One such technique is called as Rule Based Machine Translation which is also known as "Classical Approach to Machine Translation ". This technique defines some set of rules subject of which the language conversion takes place. Further the Machine translation is mainly divided into three sub categories called as Dictionary Based Machine Translation(Direct Systems), Transfer Rule Based Machine Translation and Inter lingual machine translation. In this paper we have used the Transfer Rule Based Machine Translation.

Related Work

Although the Natural Language Processing is not novel practice, there is not much literature regarding the English to Marathi Machine Translation. However there are several attempts for this. Salunkhe and Kadam[1] have tried to compare the different appraoches that can be used for Machine Translation and have proposed a hybrid model for machine translation. In the same context Sreelekha S.[2] have studied the Statistical and Rule based Machine translation for the same pair of languages. Apart from these there has been attempt made for analysing challenges of English to Marathi Machine Translation by Kharate and Patil[3]. There has also been an attempt in Java for the same by Garje and Khrate[4] called as Transmutter. We have tried to implement the similar model using different tools in Python language using different parsers as well. Tidke , Binayakya,Patil, Sugandhi[5] have given rules for handling inflections in Marathi.

System Architecture and Implementation

The main programming language that we have used is Python 3.7.6. We have used some open source tools such as Nltk[6]

library, Stanfordnlp[7] , Google translate API[8] , Inflection [9] library ,and other common library which comes along with standard python installation. For working environment we have used Anaconda[10] , also we have used Jupyter Notebook 6.0[11] for implementing this model.

1. Database Creation

For bilingual database creation we have taken English sentences . We then tokenised these sentences with the help of Nltk tokeniser. After these, the tokens are passed to Google Translate API to give the literal meaning of the token. This token and meaning pair is stored in the form of dictionary in notebook.

2. Prepossessing

For prepossessing we have used Nltk library. We initially take input as a string. After that we convert all the characters to lower cases. We also remove punctuaion symbols such as ”!”, ”(” , ”]” , ”?” etc.

3. Parsing

For purpose of parsing we have use Stanfordnlp parser model for parsing. This operation creates a parse tree which can be traversed further by some rules given in the next section. This parser also gives Parts of Speech tagging(POS) to the string.

4. Reshuffling

After obtaining the parse tree we now need to reshuffle the tree such that it is similar to target language . For this purpose the from 2nd level on wards of parse tree we have to symmetrically interchange the nodes of tree in such way that the left node now becomes right node and vice versa.As shown in Fig 2.

4. Traversal

After obtaining parse tree data structure we can further traverse this parse tree to get a sequence of English words which is similar to Marathi sentence structure. For this we have to first traverse in post order and in order fashion based upon POS tag and start collecting the words. For each POS tag we have to traverse in order except when it is root node that is ”S” tag and ”VP” ie verb phrase tag. The sequence is now stored in list having words. As shown in Fig 3.

5. Literal Translation

For translating words we have to now use the dictionary that we have created earlier. Each word is now taken in sequence which is formed after traversals. This word is now searched in dictionary for its equivalent Marathi translation. If the word is not present the Google Translate API is summoned to give the literal meaning of the particular word.

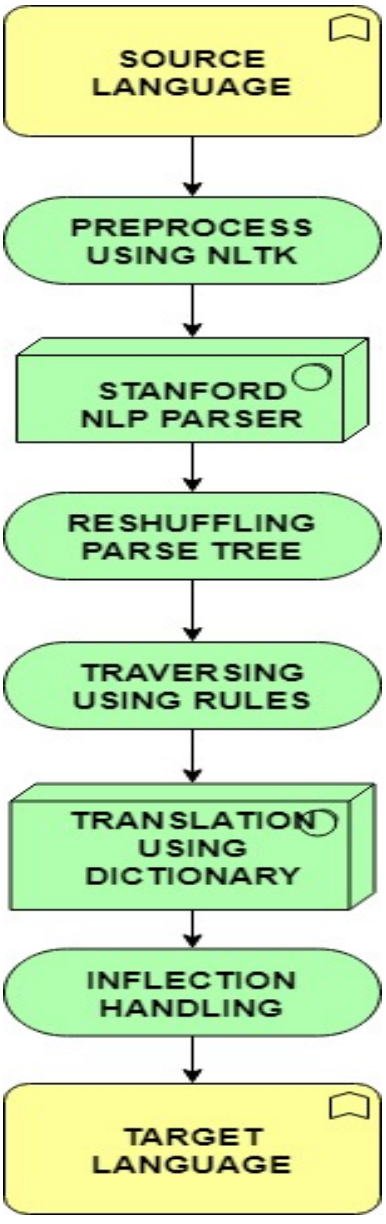

Fig 1. Flowchart of Model

6. Removal of Stop words

In this case we only remove 3 stop words as of now which do not have significance is Marathi language . These stop words are determiners that is ”a” ,”an”,”the”. We have to remove these stop words because the meaning of these words is not literally defined in Marathi Language by Google Translate API.

7. Inflection Handling

Inflection handling a tedious process and requires more knowledge about each and every word. In Marathi, there are only 4 parts of speech that requires inflection handling these are

1. Noun
2. Pronoun
3. Adjective
4. Verb

However we are able to only handle Pronoun and noun in these paper because the rest of them requires heavy knowledge of gender ,tense and singularity of the words.

Example

Let us take an English Example:
”this sentence is created for testing”

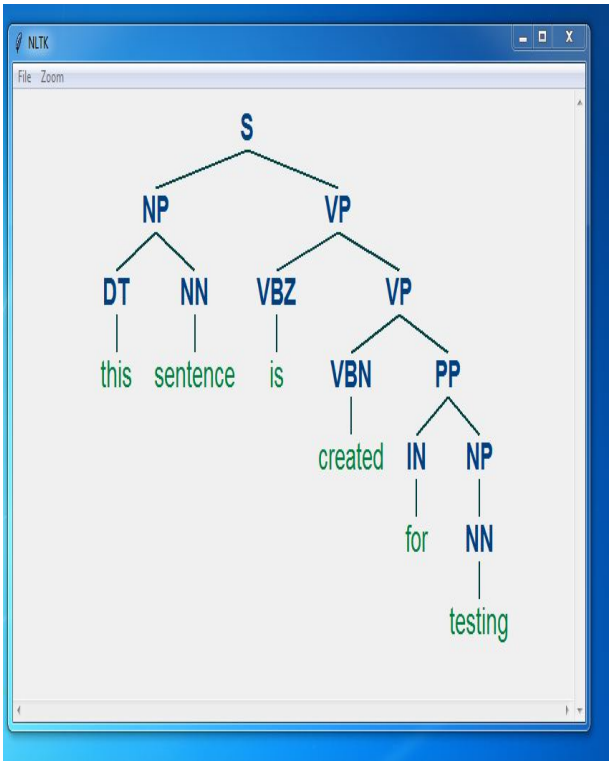

Fig 2.Parse Tree

After parsing and Reshuffling:(Fig 3)
”sentence this testing for created is”
After Literal translation and inflection handling:

प्रेषण हे चाचणी च्या साठी तयार केले आहे

Testing And Results

A well known tool for measuring accuracy of the results of Machine Translation is done using calculating BLEU[11] score . We have calculated BLEU score for 100 sentences considering Google Translated sentences as reference sentences. After testing these sentences we found the average BLEU to be 0.3225535919364329.

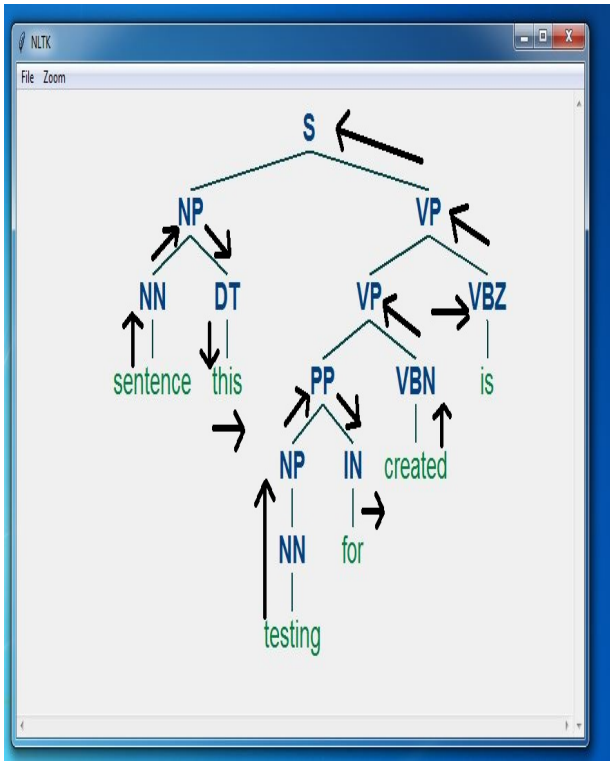

Fig 3. Traversing

Future Scope

The future scope of this paper would be handling inflections by identifying the gender, multiplicity ,tense of the particular word. Also there is a reason to believe that after expanding the database we will be able to achieve more accuracy. We could also focus on word sense disambiguation for the words which are having multiple meaning. Predicting the perfect can be achieved by context disambiguation.

Conclusion

In this paper we are able to draw conclusion that RBMT is very tough process. Although after defining some set of there are always exceptions. The process of RBMT is different for each pair of Source and Target language hence there is no single fixed approach for all the RBMT.Rules are unique for every pair of Source and Target Language. We can also conclude that accuracy of RBMT heavily depends upon data set under consideration.

Acknowledgement

We would like to thank our mentor Dr Pooja Jain for making us understand Natural Language Processing. We would also like to extend our thanks to Stanfornlp team for providing the best parser. Also we would like to acknowledge the hard work of all the researchers that have cited in this paper.

References

1. Pramod Salunkhe,Aniket. D. Kadam,Shashank Joshi Shuhas Patil,Devendrasingh Thakore,Shrikant Jadhav
   ”Hybrid machine translation for English to Marathi: A research evaluation in Machine Translation: (Hybrid translator)”,2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)
   https://ieeexplore.ieee.org/abstract/document/7754822

2. Sreelekha S.
   ”Statistical Vs Rule Based Machine Translation; A Case Study on Indian Language Perspective”
   Dept. of Computer Science & Engineering, Indian Institute of Technology Bombay, India
   https://arxiv.org/ftp/arxiv/papers/1708/1708.04559.pdf

3. Namrata G Kharate, Dr. Varsha H. Patil
   ”CHALLENGES IN RULE BASED MACHINE TRANSLATION FROM MARATHI TO ENGLISH”
   http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.676.8017&rep=rep1&type=pdf

4. G V Garje,G K Kharate,Harshad Kulkarni
   ”Transmuter: An Approach to Rule-based English to Marathi Machine Translation”
   International Journal of Computer Applications (0975 – 8887) Volume 98 – No.21, July 2014
   http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.676.8017&rep=rep1&type=pdf

5. Charugatra Tidke, Shital Binayakya, Shivani Patil, Rekha Sugandhi
   ”Inflection Rules for English to Marathi Translation”
   IJCSMC, Vol. 2, Issue. 4, April 2013, pg.7 – 18
   https://www.ijcsmc.com/docs/papers/April2013/V2I4201301.pdf

6. NLTK library documentation home page
   https://pypi.org/project/nltk/

7. Stanfordnlp documentation home page.
   https://nlp.stanford.edu/software/

8. Google Translate API home page.
   https://pypi.org/project/googletrans/

9. Anaconda Documentation
   https://docs.anaconda.com/

10. Jupyter Notebook Documentation.
    https://jupyter-notebook.readthedocs.io/en/stable/

11. 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.
    https://www.aclweb.org/anthology/P02-1040.pdf