International Journal of Research in Engineering, Science and Management
Volume-2, Issue-6, June-2019
www.ijresm.com | ISSN (Online): 2581-5792

English to Marathi Translator using Anusaaraka

Manisha S. Otari
Assistant Professor, Department of Computer Science and Engineering, Nagesh Karajagi Orchid College of Engineering &Technology, Solapur, India

***Abstract*: In India there are many spoken languages. Many of the states have their own regional language which is either Hindi or one of the other constitutional languages. In addition, English is very widely used for media, commerce, science and technology and education only 5% of the world's population speaks English as a first language. In such a situation, there is a large market for translation between English & the various Indian languages. Proposed system will be able to translate appropriate meaning of English sentence to Marathi sentence by using Anusaaraka tool by inputting text file.**

***Keywords*: Anusaraka, Machine Translation, Morphology**

1. Introduction

A majority of human languages including Indian and other languages have relatively free-word order. In free-word order languages, order of words contains only secondary information such as emphasis etc. Primary information relating to 'gross' meaning (e.g., one that includes semantic relationships) is contained elsewhere. Most existing computational grammars are based on context free grammars which are basically positional grammars. Thus finding appropriate meaning of words in such languages while translating to other languages becomes a very difficult task. Anusaaraka is a language accessing software. With insights from Panini's Ashtadhyayi (Grammar rules), Anusaaraka is a machine translation tool being developed by the Chinmaya International Foundation (CIF), International Institute of Information Technology, Hyderabad (IIIT-H) and University of Hyderabad (Department of Sanskrit Studies).

Anusaaraka derives its name from the Sanskrit word '*Anusaran'* which means *'to follow'*. It is so called, as the translated Anusaaraka output appears in layers – i.e. a sequence of steps that follow each other till the final translation is displayed to the user.

Morphology is a part of linguistic that deals with study of words, i.e. internal structure and partially their meanings. A morphological analyzer is a program for analyzing morphology for an input word; it detects morphemes of any text. Many morphological analyzers have been developed before for various languages. These are mostly based on position of words in sentence hence are only useful for positional languages such as English.

In order to develop a morphological analyzer which helps to improve translation, from one language to other, information such as group word information, verb suffix etc. along with inflectional rules should be taken into account. This information can help to find correct meaning of word in the context of the given sentence.

Machine translation has different architectures such as Direct, Transfer-Based, Interlingua, Statistical, Example-platform for making Rule-Based machine translation system. Each of them has its advantages and disadvantages and selection of the approach can be made based on the domain of the application.

2. Silent features of Anusaraka

A. Faithful representation of text in source language

Throughout the various layers of Anusaaraka output there is an effort to ensure that the user should be able to understand the information contained in the English sentence. This is given greater importance than giving perfect sentences in Marathi, for it would be pointless to have a translation that reads well but does not truly capture the information of the source text.

The layered output is unique to Anusaaraka. Thus, source language text information and how the Marathi translation is finally arrived at, can be accessed by the user. The important feature of the layered output is that the information transfer is done in a controlled manner at every step thus, making it possible to revert back without any loss of information. Also, any loss of information that cannot be avoided in a translation process is then done in a gradual way. Therefore, even if the translated sentence is not as 'perfect' as human translation, with some effort and orientation on reading Anusaaraka output, an individual can understand what the source text is implying by looking at the layers and and context in which that sentence appears.

B. Reversibility

The feature of gradual transference of information from one layer to the next, gives Anusaaraka an additional advantage of bringing reversibility in the translation process – a feature which cannot be achieved by a conventional machine translation system. A bi-lingual user of Anusaaraka can, at any point, access the source language text in English, because of the transparency in the output. Some amount of orientation on how to read the Anusaaraka output would be required for this.

C. Transperancy

Display of step-by-step translation layers gives an increased

level of confidence to the end-user, as he can trace back to the source and get clarity regarding translated text by analysis of the output layers and some reference to context.

3. Proposed system and design

A sentence first enters the morphological analyzer which finds each word in the dictionary of indeclinable words and returns its grammatical features. If the word is not found then morphing refers to word paradigms to find whether it is possible to derive the word from root and its paradigm. if it cannot be derived then its passed to the sandhi package as it may be a compound word and analyzed again. The output of morphological analyzer is passed to local word grouper which groups words based on the local information available. After grouping sentential analysis can be done if a large database is available.

In the next stage using various dictionaries, Anusaaraka finds root and vibhakti for each word in target language. This is the first step in translation. Before this mapping stage the system was trying to understand the meaning of the uttered sentence. The word groups formed by the local word grouper are now split back by the local word splitter. In the last stage thesynthesizer takes the output of splitter and generates words from root and grammatical features.

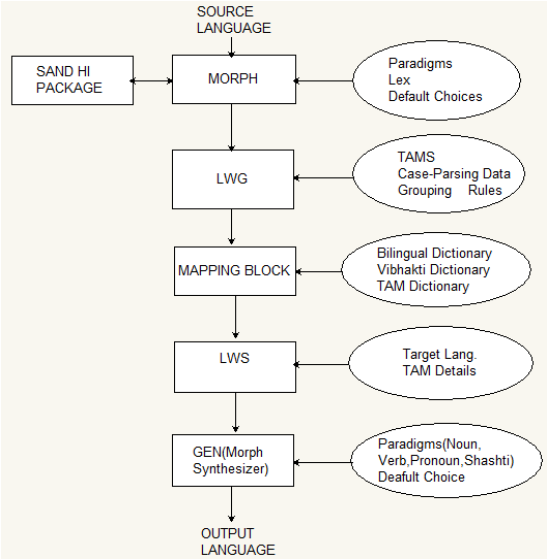

Fig. 1. Block schematic of Anusaraka

4. Implementation methodology

A. Commands to download and Install Anusaaraka

1. sudo apt-get install git

(Note: if git package is not found then check network settings and enter command sudo apt-get update Then proceed to install git)

2. Run the following commands in $HOME

git clone https://bitbucket.org/anusaaraka/anusaaraka.git

OR

git clone https://code.google.com/p/anusaaraka

gitclone https://bitbucket.org/anusaaraka/provisional_wsd_rules.git

sudo apt-get install perl python flex bison apertium xsltproc libgdbm3 libgdbm-dev libicu-dev gcc g++ ant ssmtp apache2 php5

3. Download oracle jdk from

http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

4. Extract above in home folder

vi ~/.bashrc

export HOME_anu_test=$HOME/anusaaraka

export HOME_anu_output=$HOME/anu_output

export HOME_anu_tmp=$HOME/tmp_anu_dir

export HOME_anu_provisional_wsd_rules=$HOME/provisional_wsd_rules

export PATH=$HOME/<oracle jdk_directory_name >/bin:$HOME_anu_test/bin:$PATH

export JAVA_HOME=$HOME/<oracle_jdk_directory_name >

export LD_LIBRARY_PATH=/usr/local/lib/

export http_proxy=http://proxy.iiit.ac.in:8080

{ Proxy Setting depends upon your internet connection change above proxy configuration accordingly if proxy uses password authentication it should be in form of

export http_proxy="http://usrname:passwrd@host:port"}

source ~/.bashrc

Download stanford parser latest version from following link:

http://nlp.stanford.edu/software/stanford-parser-full-2014-08-27.zip

(Note: Current version is 3.4.1)

. Copy the above downloaded zip file in the following path:

$HOME_anu_test/Parsers/stanford-parser/

Run:

cd $HOME_anu_test/Parsers/stanford-parser/

sh get_latest_version_stanford_parser.sh <stanford-parser-latest-version.zip>

Ex: sh get_latest_version_stanford_parser.sh stanford-parser-full-2014-08-27.zip

sudo cp $HOME_anu_test/miscellaneous/e-mail/mail.php /var/www/

sudo cp $HOME_anu_test/miscellaneous/e-mail/mail.php /var/www/ html/

sudo cp $HOME_anu_test/miscellaneous/e-mail/ssmtp.conf /etc/ssmtp/

sudo service apache2 restart

(Note: If apache doesnt start then, add the following line in sudo vi /etc/apache2/httpd.conf and the save the file ServerName localhost If this also doesn't work add the following line in sudo vi /etc/apache2/apache2.conf then save the file ServerName localhos)

cd $HOME_anu_test

shell_scripts/remove_out-files.sh

shell_scripts/anu_compile.sh

B. Commands to Run Anusaaraka

vi sample
Copy below code in sample file.
This is a sample file for Anusaaraka.
Anusaaraka_stanford.sh <filename><parsenum><True>
<filename> : Name of file to be given as input
<parsenum> : Number of Parser to use /(if you don't know use 0 here)
<True> : True if anusaaraka is running in server mode else leave empty
ex : Anusaaraka_stanford.sh sample 0 True
sudo apt-get install git

C. To view layered o/p

firefox $HOME_anu_output/sample_frame.html
To send email if any of the word translation is wrong:
firefox $HOME_anu_output/sample_sample2.html

D. To view debug information in layered o/p

firefox $HOME_anu_output/sample_sample2.html

5. Anusaaraka for English to Marathi

1. Created following files into anu_data.
   marathi-dic.txt
   marathi_tam.txt
   marathi_multiword.txt

Fig. 2. Files created in anu_data folder

2. Created folder marathi_wsd_rules in anusaaraka/WSD folder.

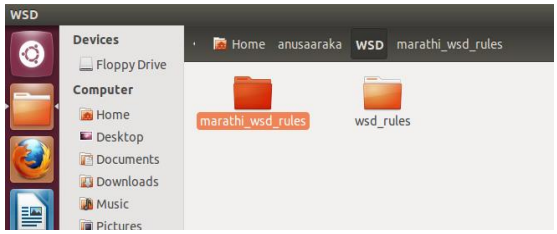

Fig. 3. Created WSD folder into Anusaaraka

3. Created marathi_compile.sh same like as anu_compile.sh in shell_scripts folder.
4. Following files are created in anusaaraka/bin folder
   - marathi_anusaaraka_stanford.sh
   - run_marathi_sentence_stanford.sh
   - marathi_generationv1.bin
   - marathi_morph.bin

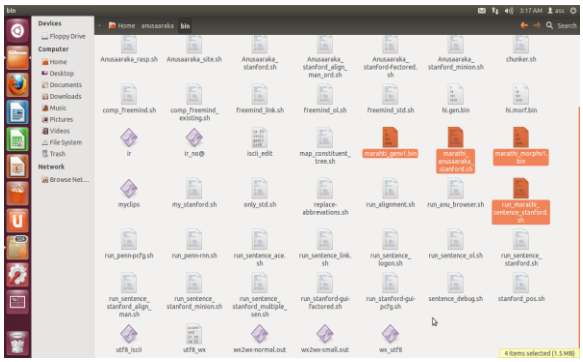

Fig. 4. Files created in anusaraka/bin folder

5. Created file run_marathi_modules_std.bat in anusaaraka/Anu_clp_files.

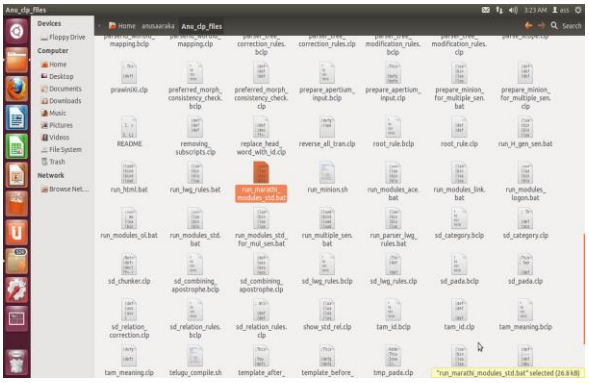

Fig. 5. Files created inAnusaaraka/Anu_clp_files Folder

6. Created file marathi_multiword_expression.c in Multifast/src folder.

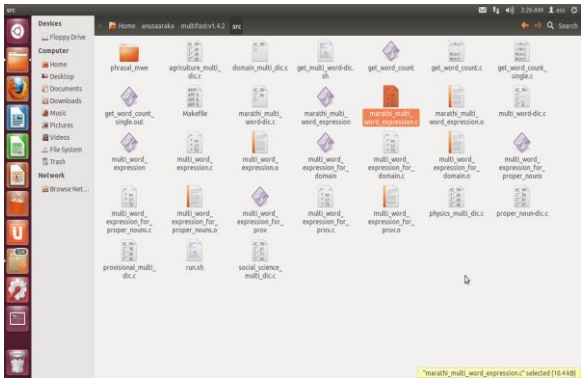

Fig. 6. Files Created in multifast/src directory

7. Created marathi_multiword_expresssion.txt in anu_data/compound_matching folder.
8. Prepared marathi-dic.txt
   - First we have prepared dictionary of English to Marathi word meaning. After that we have converted that

International Journal of Research in Engineering, Science and Management
Volume-2, Issue-6, June-2019
www.ijresm.com | ISSN (Online): 2581-5792

dictionary into form of internal representation of computer. Following screenshots shows the overview of marathi-dic.txt

- Command for converting dictionary utf8-wxinput_file>output_file

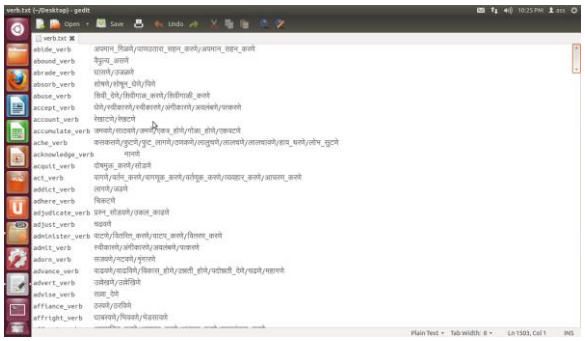

Fig. 7. English to Marathi word meaning

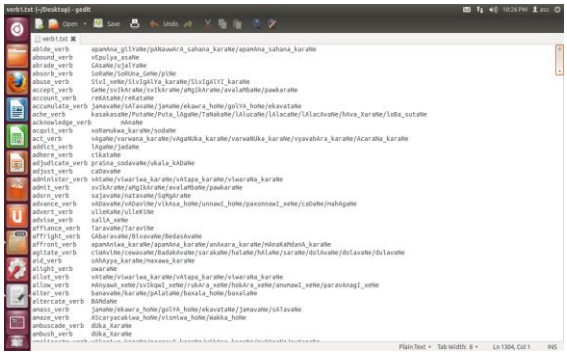

Fig. 8. English to Marathi dictionary into form of internal representation of system

9. Created file marathi_AllTam.txt in anusaaraka/Anu_data

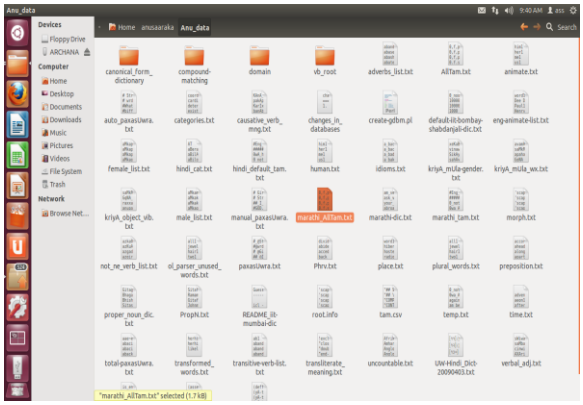

Fig. 9. Files created inanusaaraka/Anu_data folder

10. Add file name (marathi_AllTam.txt) in following files
    - Anusaaraka/Anu_data/Canonical_Form/list_Anu_data
    - Anusaaraka/Anu_data/Canonical_Form/list_two_side_hindi.txt

Fig. 10. Files to add name of file marathi_AllTam.txt

11. Specify the path of file in following shell script files.
    - Shell Scripts/marathi_compile.sh
    - bin/run_marathi_sentence_stanford.sh

6. Commands used for Anusaaraka

$shell_scripts/marathi_compile.sh
$marathi_anusaaraka_stanford.sh sample 0 true
$firefox $Home_anu_output/sample_frame.html

A. Commands used to get output in text file

$cd anu_output
$sh rm_tags_from_trns_file.sh sample_trnsltn.html

B. Commands used for Apertium

$cd Desktop/marathi_apertium_morph
$ lt-proc -c marathi_morphv1.bin
rAmAne
^rAmAne/rAma<pos:n><gender:nm><number:eka><parsarg:ne>

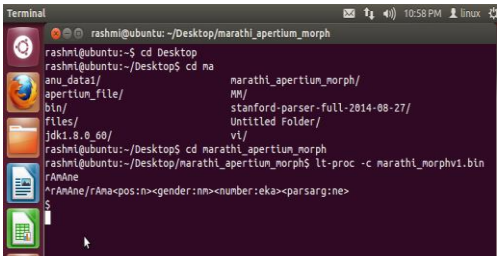

Fig. 12. Output of command $ lt-proc -c marathi_morphv1.bin

$ lt-comp rl marathi_morphv1.dict new1.bin
main@standard 45738 161895

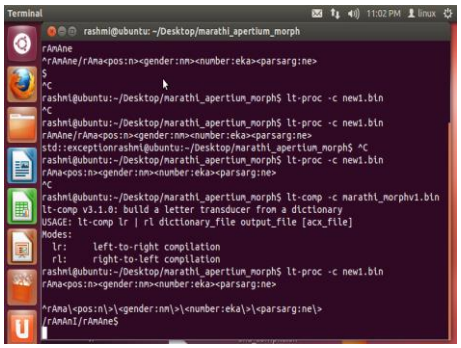

Fig. 13. Output of command $ lt-comp rl marathi_morphv1.dict new1.bin

```
$ lt-proc -*c new1.bin
rAma<pos:n><gender:nm><number:eka><parsarg:ne>
^rAma\<pos:n\>\<gender:nm\>\<number:eka\>\<parsarg:ne\>/
rAmAnI/rAmAne$
```

7. Result

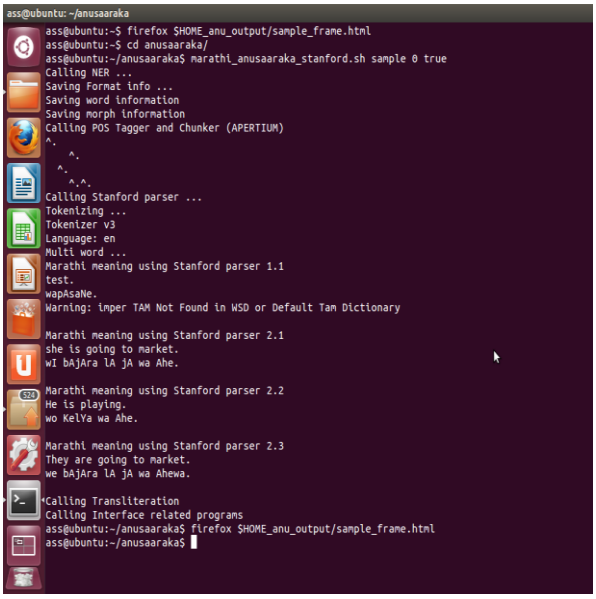

Fig. 14. Running sample file

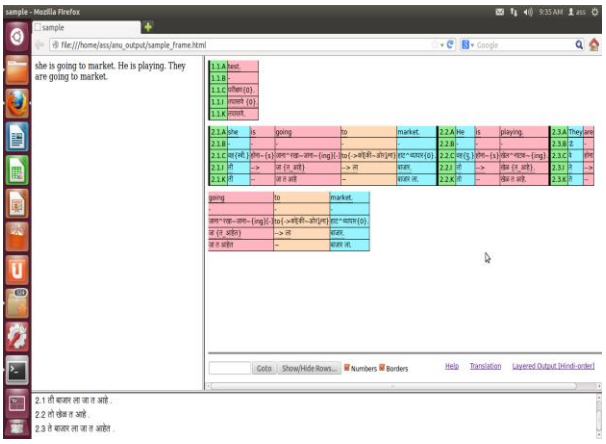

Fig. 15. Layered Output of sample file in firefox

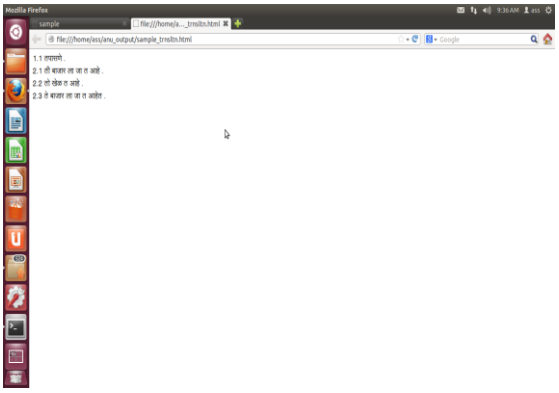

Fig. 16. Output of sample file in txt format

8. Conclusion

This paper presents an overview on English to Marathi Translator using Anusaaraka.

References

[1] Bharati Akshar, Vineet Chaitanya, Rajeev Sanghal, Natural Language Processing: A Paninian Perspective 1995.Publication Prentice-Hall of India, New Delhi.

[2] Pratiksha Gawade, Deepika Madhavi, Jayshree Gaikwad, Sharvari Jadhav, Rahul Ambekar, “Morphological Analyzer for Marathi using NLP,” International Journal of Engineering Research and Applications, Vol. 3, Issue 2, March-April 2013.

[3] Mugdha Bapat, Harshada Gune, Pushpak Bhattacharyya, A Paradigm-Based Finite State Morphological Analyzer for Marathi August 2010, Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), pages 26–34, The 23rd International Conference on Computational Linguistics (COLING), Beijing,

[4] Shalmalee Pitale, Vaijayanthi Sarma, Plural Problems in the Nominal Morphology of Marathi, 25th Pacific Asia Conference on Language, Information and Computation, pages 178–185, December 2011.

[5] Ganesh Bhosale, Subodh Kembhavi, Archana Amberkar, Supriya Mhatre, Lata Popale, Pushpak Bhattacharyya, Processing of Kridanta (Participle) in Proceedings of ICON-2011: 9th International Conference on Natural Language Processing Macmillan Publishers, India.

[6] Koskenniemi Kimmo, Hallituskatu, “Two-level Morphology: a general computational model for word-form recognition and production,” 11-13, SF-00100, Helsinki 10, Finland, Publications, No. 11, 1983.

[7] Ashwini Vaidya and Dipti Misra Sharma,, “Using Paradigms for certain morphological phenomenon in Marathi,” Proceedings of ICON-2009: 7thInternationalConference on Natural Language Processing, Macmillan Publishers, India.

[8] Harshada Gune, Mugdha Bapat, Mitesh M. Khapra, and Pushpak Bhattacharyya, “Verbs are where all the action lies: Experiences of Shallow Parsing of a Morphologically Rich Language,” Proceedings of COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Pages 347-355, Association for Computational Linguistics Stroudsburg, PA, USA 2010.