IJCAT International Journal of Computing and Technology, Volume 1, Issue 2, March 2014
ISSN : 2348 - 6090
www.IJCAT.org

Relative Study on Malayalam-English Translation using Transfer Based Approach

[1]Shahana.I.L, [2]Sharafudheen.K.A

[1] Mtech Student,Dept of CSE, Malappuram, Kerala, India

[2] Asst. Professor ,Dept of CSE, Malappuram, Kerala, India

Abstract - Machine Translation is the process of translating the sentences from source language into target language, by the use of computers, with or without the influence of human assistance. Here describes a relative study on Malayalam to English and English to Malayalam translation by using transfer based approach. For Malayalam to English translation, we need a system comprises of a preprocessor for splitting the compound words, a morphological parser for context disambiguation and chunking, a syntactic structure transfer module and a bilingual dictionary. All the modules are morpheme based to reduce dictionary size. For English to Malayalam translation, the English sentence as input and parse it with the help of Stanford Parser. The system takes the parsed input and separates the source text word by word with POS category and searches for their corresponding target words in the bilingual dictionary. In this stage, the set of Malayalam words with their POS category are the output. For named entities, we need an English to Malayalam transliterator. The system processes through the FST model, which has been developed for incorporating Malayalam morphology. Then the English sentence syntactically reordered to suit Malayalam language by applying the reordering rules.

***Keywords* - POS, FST, Stemming , Transliterator.**

1. Introduction

Translation is the process of translating the sentences from source language into target language, by the use of computers, with or without the influence of human assistance. The major goals in translation system development work are accuracy and speed. Accuracy-wise, smart tools for handling transfer grammar and translation standards including equivalent words, expressions, phrases and styles in the target language are to be developed. The grammar should be optimized with a view to obtaining a single correct parse and hence a single translated output. Speed-wise, innovative use of corpus analysis, efficient parsing algorithm, design of efficient Data Structure and run-time frequency-based rearrangement of the grammar which substantially reduces the parsing and generation time are required.

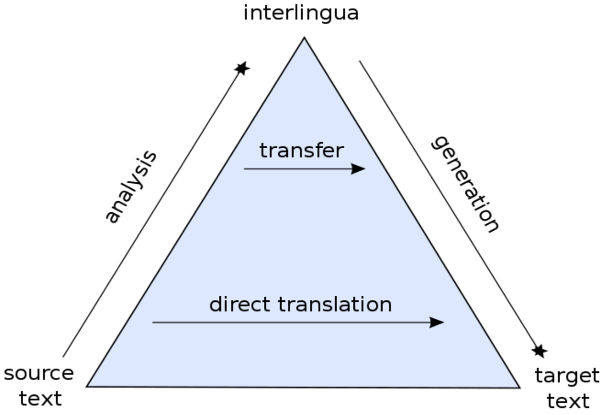


Fig1: Various approaches for machine translation

The various approaches used in the MT systems are: Direct machine translation systems, Rule based systems and Corpus based systems. The rule-based machine translation paradigm includes transfer-based machine translation, interlingual machine translation and dictionary-based machine translation. Transfer-based machine translation is similar to interlingual machine translation in that it creates a translation from an intermediate representation that simulates the meaning of the original sentence. Unlike interlingual MT, it depends partially on the language pair involved in the translation. Machine translation can use a method based on dictionary entries, which means that the words will be translated as they are by a dictionary. Dictionary usually followed by some syntactic arrangement.

2. Literature Survey

2.1 English-Malayalam translation: transfer based approach [3]

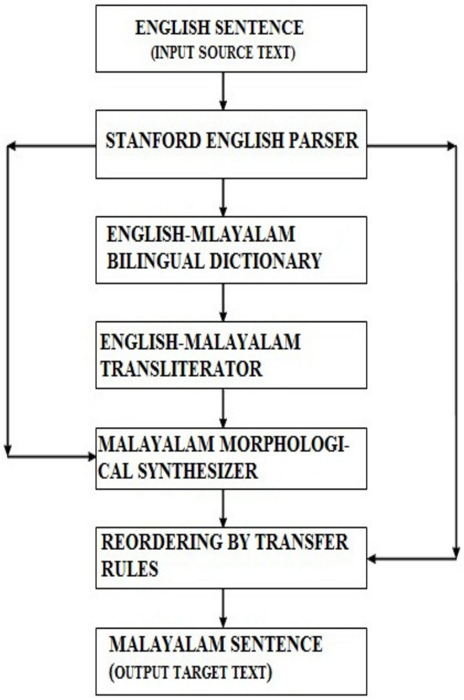

ENGLISH SENTENCE
(INPUT SOURCE TEXT)
STANFORD ENGLISH PARSER
ENGLISH-MLAYALAM BILINGUAL DICTIONARY
ENGLISH-MALAYALAM TRANSLITERATOR
MALAYALAM MORPHOLOGI-CAL SYNTHESIZER
REORDERING BY TRANSFER RULES
MALAYALAM SENTENCE
(OUTPUT TARGET TEXT)

Fig2: Block diagram for translation English-Malayalam

Parser: First phase of this model is a statistical Stanford parser, which is based on probabilistic context free grammar (PCFG). This is used here for mainly 4 purposes. They are:

1. Syntax analysis
2. POS tagging
3. Stemming
4. Morpholgical analysis of given English sentence

English-Malayalam Bilingual Dictionary: This dictionary contains a set of words in English and their corresponding meaning in Malayalam. It is used here for word by word translation. Each entry of the dictionary is preprocessed. Preprocessing includes Font-converting, Aligning, Lexicalizing, Adding synonyms, removing duplicates. Thus dictionary contains mainly 5 fields. They are source, target, category, feature and synonym.

English to Malayalam Transliteration: In English to Malayalam transliteration, the English text is replaced with the corresponding Malayalam text by preserving the spell, which is done by English-Malayalam mapping.

Malayalam Morphological Generator: A bi-directional Morphological Generator cum Morphological analyzer has been used for Malayalam, for synthesizing morph to the corresponding Malayalam words. Finite State Transducer (FST) is used to model the morphology and some orthographic rules of Malayalam are written. This FST based Malayalam morphological synthesizer is used in the machine translation system. The Stanford Parser is used to stem the English words from the input sentence and also to get the morphologically analyzed information. The equivalent Malayalam words are extracted from the English to Malayalam Bilingual Dictionary.

The dependency information from the parser is transferred to the required format for morphological synthesis. So these dependency transfer information are separately stored for nouns and verbs in the database. The input to the FST is considered as the Regular Expression. For example, the plural marking for noun maraM is written as [   ]:
FST input: maraM+NOUN+PLURAL → maraM + N + PL

Corresponding FST is:

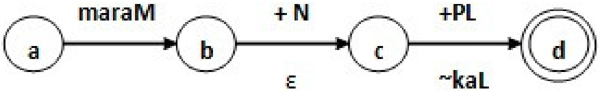
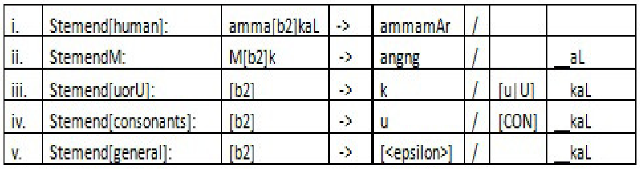


Fig3:FST

Output of FST: maraM+N+PL →maraM~kaL
Orthographic rules for plural is:



i.	Stemend[human]:	amma[b2]kaL	->	ammamAr	/		
ii.	StemendM:	M[b2]k	->	angng	/		__aL
iii.	Stemend[uorU]:	[b2]	->	k	/	[u|U]	kaL
iv.	Stemend[consonants]:	[b2]	->	u	/	[CON]	__kaL
v.	Stemend[general]:	[b2]	->	[<epsilon>]	/		__kaL

Fig4:Orthographic rules

Reordering by Transfer Rules: The reordering can be machine learned or executed by rules or both. English is the Subject-Verb-Object (SVO) language, whereas Malayalam is Subject-Object-Verb (SOV) language. So reordering is necessary in machine translation. The reordering by rules is followed here to reorder the English sentence in the order of Malayalam sentence.

Eg: I am eating an apple, this sentence is reordered with respect to Malayalam by executing two re-ordering rules:
Rule 1: VP (VBP: VP) → VP(V P : V BP)
Rule 2: VP (VBG: NP) → VP(NP : V BG)

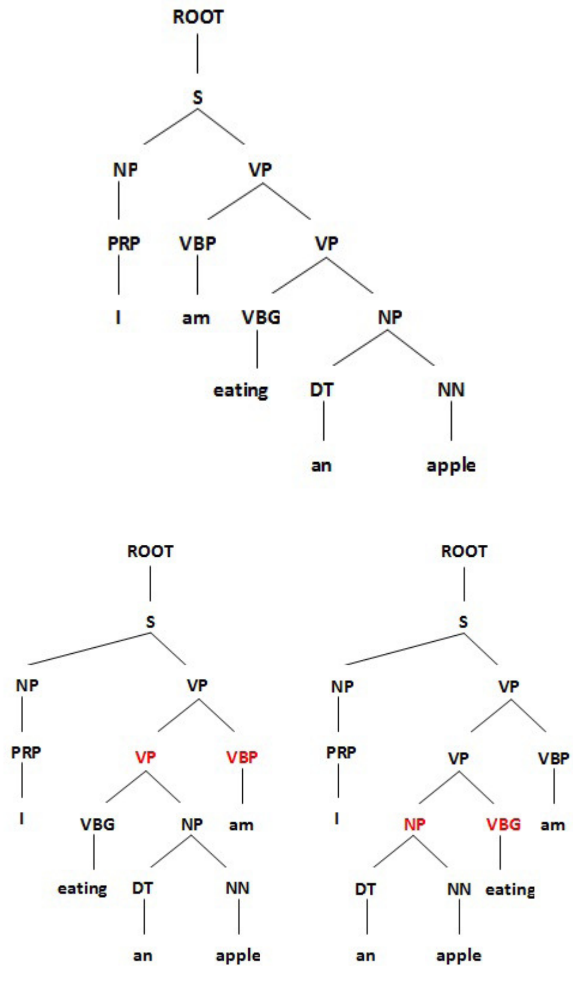


Fig5: An Eg:

2.2 Malayalam-English Translation: Transfer Based Approach [4]

A transfer based MT system has been developed with the following system modules:

[1] A preprocessor for splitting the compound words
[2] A morphological parser for context disambiguation and chunking
[3] A transfer module which transfers the source language structure representation to a target language representation.
[4] A generation module which generates target language text using target language structure.

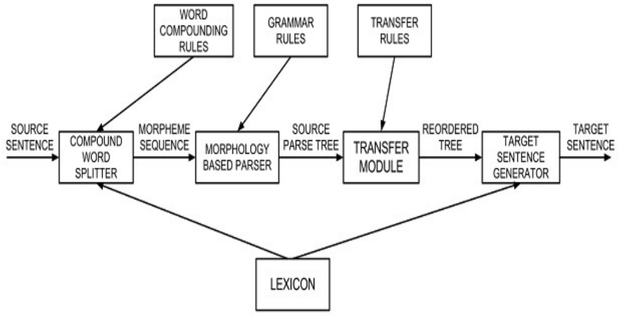


Fig 6: Block diagram for Malayalam-English translation

Compound Word Splitter Module [2]: Morphological variations for words occur in Malayalam due to infections, derivations and word compounding. Malayalam is an agglutinative language i.e, words of different syntactic categories are combined to form a single word. The word may be formed by combining a noun and a noun, noun and adjective, verb and noun, adverb and verb, adjective and noun and in some cases all the words of an entire sentence to reflect the semantics of the sentence are very common. Such a compound statement is shown below:

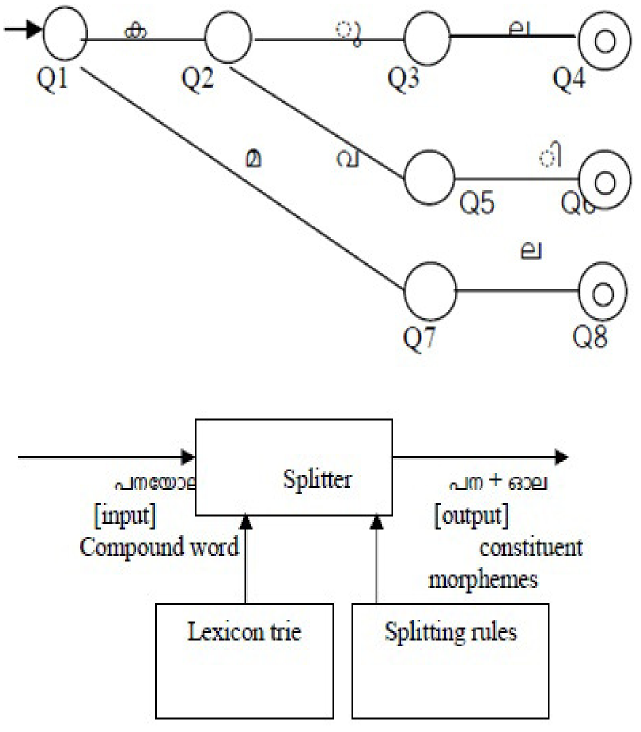
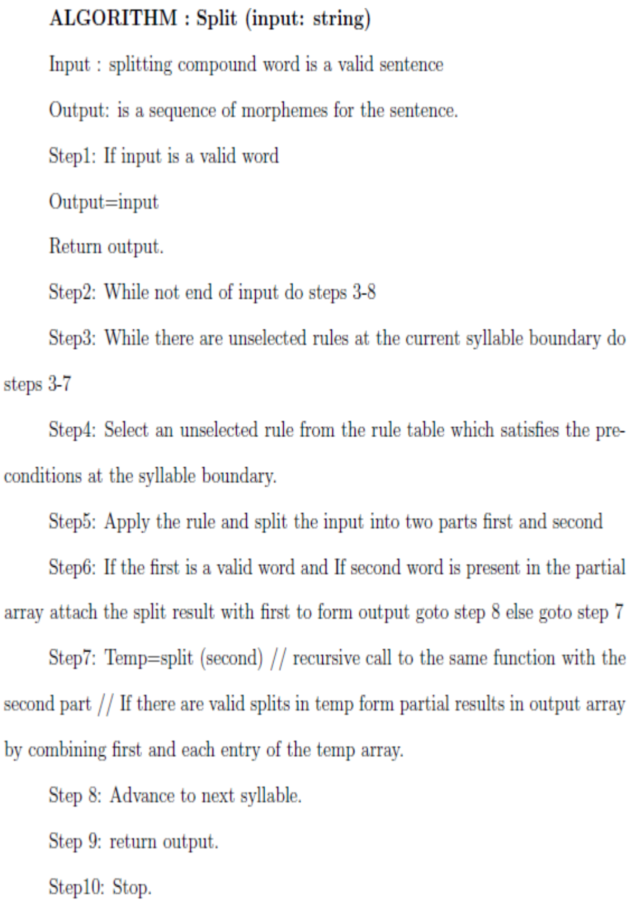


Fig7:a).Splitting rules, b).Splitter



ALGORITHM : Split (input: string)

```
Input : splitting compound word is a valid sentence
Output: is a sequence of morphemes for the sentence.
Step1: If input is a valid word
Output=input
Return output.
Step2: While not end of input do steps 3-8
Step3: While there are unselected rules at the current syllable boundary do steps 3-7
Step4: Select an unselected rule from the rule table which satisfies the pre-conditions at the syllable boundary.
Step5: Apply the rule and split the input into two parts first and second
Step6: If the first is a valid word and If second word is present in the partial array attach the split result with first to form output goto step 8 else goto step 7
Step7: Temp=split (second) // recursive call to the same function with the second part // If there are valid splits in temp form partial results in output array by combining first and each entry of the temp array.
Step 8: Advance to next syllable.
Step 9: return output.
Step10: Stop.
```

Parser:

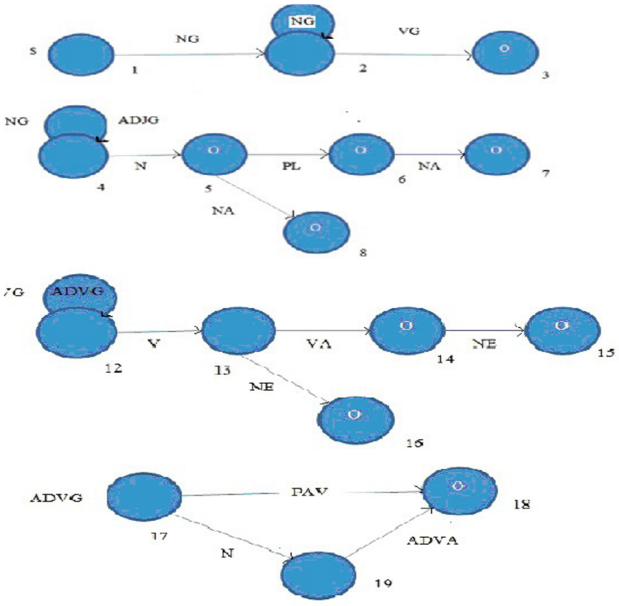


Fig 8: The transition diagrams for the chunks.

For parsing chunker is used [1]. The chunker uses sentence level grammatical rules for finding the morpheme groups. The rules are implemented as finite state cascades. A few of the chunks and their corresponding rules are given in following figure. Here the chunk tag NG refers to the group of morphemes which function as a noun in the sentence.

The parsing algorithm proceeds in a depth first approach and when it cannot move forward since there is no transition on the input tag, it backtracks to the previous state and tries other options.

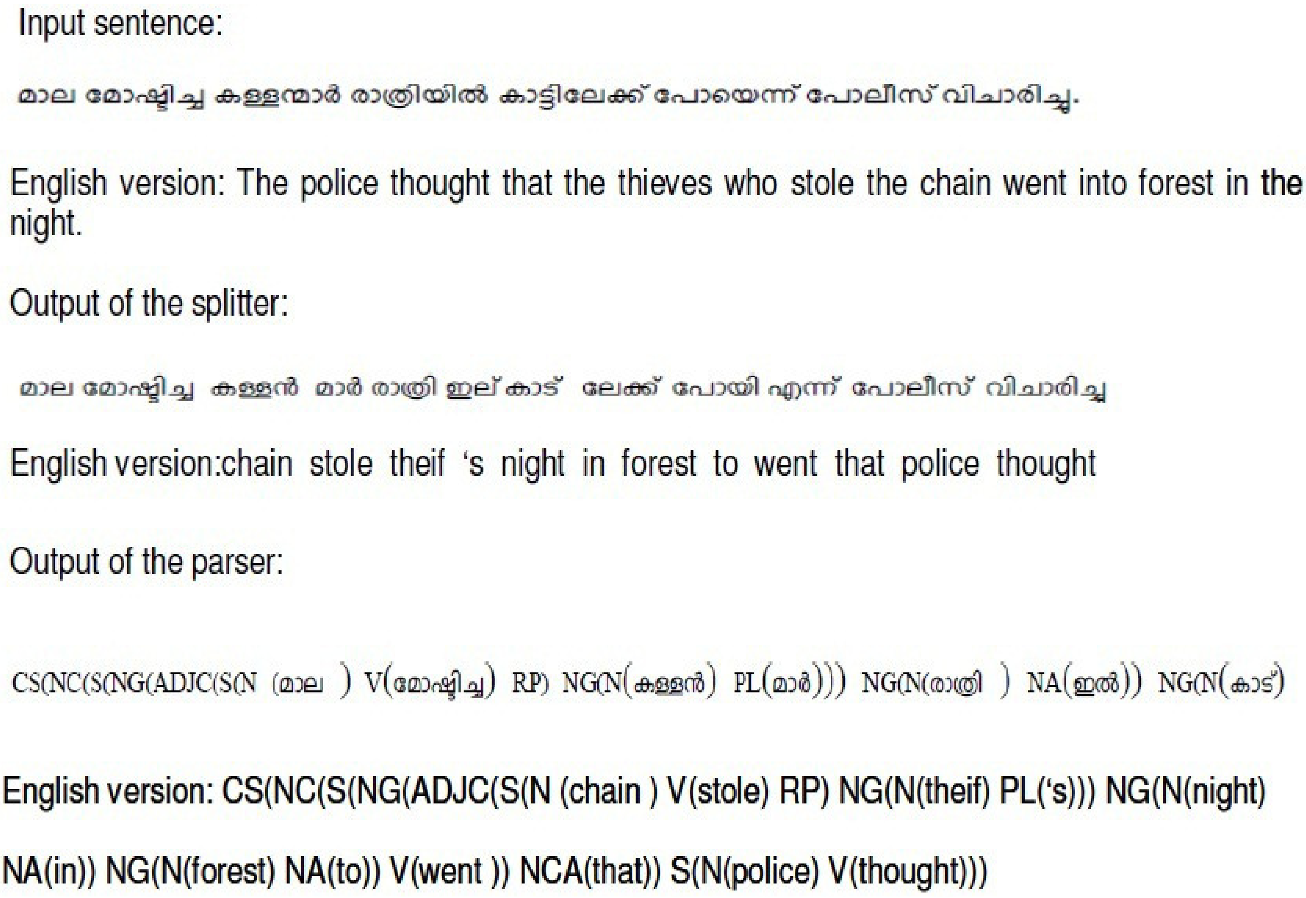
Algorithm for detecting a chunk:

1. Begin from the starting state of the finite state hierarchy.
2. Initialize an array CL(chunk list) to store the sequence of chunks identified.
3. For each possible untried transition on the input symbol do steps 4, 5, 6 and 7. Longer matches are tried first before shorter ones.
4. If the transition is based on a lexical tag symbol then move to the next state if it matches with the morpheme tag in input and advance the input pointer goto step3.
5. Else if the transition is on a chunk tag then search for the sequence of tags corresponding to the chunk recursively. Add the chunk returned if it returns a list and go to next state. Proceed to step 3.
6. Else if the current state is a final state then the Chunk name, sequence of words for the chunk in CL and the current input pointer are returned.
7. Return failure.
8. Stop

Syntactic Structure Transfer Module: The transfer module transfers the source language structure representation to a target language representation. This module needs the sub tree rearrangement rules by which the source language sentence syntax tree can be transformed into target language sentence syntax tree. The system performs most of the commonly needed reordering for Malayalam to English translation.

Target Sentence Generator Module: The generation module generates target language text using target language structure. This uses inter chunk dependency rules and intra chunk dependency rules. It involves lexical transfer of verbs, transfer of auxiliary verb for tense, aspect and mood and transfer of gender, number and person information. It also uses a cross lingual dictionary.

English to Malayalam translator is 53% accurate.



Input sentence:

മാല മോഷ്ടിച്ച കള്ളന്മാർ രാത്രിയിൽ കാട്ടിലേക്ക് പോയെന്ന് പോലീസ് വിചാരിച്ചു.

English version: The police thought that the thieves who stole the chain went into forest in the night.

Output of the splitter:

മാല മോഷ്ടിച്ച കള്ളൻ മാർ രാത്രി ഇല് കാട് ലേക്ക് പോയി എന്ന് പോലീസ് വിചാരിച്ചു

English version:chain stole theif ‘s night in forest to went that police thought

Output of the parser:

CS(NC(S(NG(ADJC(S(N (മാല) V(മോഷ്ടിച്ച) RP) NG(N(കള്ളൻ) PL(മാർ))) NG(N(രാത്രി) NA(ഇൽ)) NG(N(കാട്)

English version: CS(NC(S(NG(ADJC(S(N (chain) V(stole) RP) NG(N(theif) PL(‘s))) NG(N(night) NA(in)) NG(N(forest) NA(to)) V(went)) NCA(that)) S(N(police) V(thought)))

Fig 9: An eg: for Malayalam-English translation

3. Performance Analysis

The performance of Malayalam to English translator is only 20 %. Reasons are:

- The positioning of articles is not considered.
- Many inter chunk and intra chunk dependencies are not considered.
- The lexicon stores only the common translation for polysemous words.

The system takes care of word sense disambiguation based on lexical category successfully. The compound nouns are also not handled by the system as the shallow parser cannot group them using the current set of rules.
The system output can be enhanced including rules which can take care of the above shortcomings.

4. Conclusion

Various MT groups have used different formalisms best suited to their applications. Of them transfer based systems are more flexible and it can be extended to language pairs in a multilingual environment. A transfer based MT system has been developed for converting Malayalam to English, which comprises of a preprocessor for splitting the compound words, a morphological parser for context disambiguation and chunking, a syntactic structure transfer module and a bilingual dictionary. Performance of this system is improved by using a transliterator. For converting English to Malayalam splitting of compound word is not required. This system consists of parser, English Malayalam bilingual dictionary, English Malayalam transliterator, Malayalam morphological synthesizer and reordering by transfer rules.

References

.

[1] L.R.Nair, D.S. Peter, “Shallow parser for Malayalam Language using finite state cascades”,IEEE 2011

[2] L.R.Nair, D.S. Peter, “Development of a rule based learning system for splitting compound words in Malayalam language”IEEE 2011

[3] R. Harshawardhan, “A Rule Based Machine Translation from English to Malayalam”, 2011

[4] Latha R Nair, David Peter & Renjith P Ravindran, “Design and Development of a Malayalam to English Translator- A Transfer Based Approach”,,IJCL, 2012

[5] S. Sumaja, R. Loganathan, and Soman.K.P, \English to malayalam translit-eration using sequence labeling approach." International Journal of Recent Trends in Engineering, May 2009. Vol. 1, No. 2.

First Author: Shahana.I.L obtained her BTech in computer science and engineering from university of Kerala in the year 2012. Presently she is pursuing her MTech program in computer science and engineering at MESCE kuttippuram. Her research interests covers areas of Natural language processing, pattern recognition etc.

.

Second Author : **Sharafudheen.K.A** was born in Thodupuzha, India, in 1987. He received the B.Tech. degree in Computer Science and Engineering from the GEC Idukki, Mahatma Gandhi University, Kottayam, India, in 2009, and the M.Tech. degrees in Computer Science and Engineering from the MES CE,University of Calicut, Calicut, India, in 2011 respectively. In 2012, he joined in the Department of Computer Science and Engineering, MES CE, Malappuram as a Lecturer, and in 2013 he joined at AACET, Thodupuzha, India as Assistant Professor in Dept. of CSE and currently(from 20014 onwards) he is in ISSAT Muvattupuzha as Assistant Professor in Dept. of CSE. His current research interests include Compiler Construction, Data Mining and Natural Lamguage Processing. Prof. Sharafudheen is a Fellow of Association of Compluter Electric and Elctronics Engineers(ACEEE) and has Life Member of the Indian Society for Technical Education (ISTE).