# PHRASE STRUCTURE BASED ENGLISH TO KANNADA SENTENCE TRANSLATION

**SHARANBASAPPA HONNASHETTY[1], MALLAMMA REDDY[2], DR. M HANUMANTHAPPA[3]**

[1]Doddappa Appa Institute of MCA, Gulbarga, India
[2,3]Bangalore University, Bangalore, India
E-mail: [1]sharan.sjce@yahoo.com, [2]mallamma_vreddy@yahoo.co.in, [3]hanu6572@hotmail.com

**Abstract-**In order to build a natural language processing system first the words are placed into a structured form that leads to a syntactically correct sentence. Syntactic analysis of a sentence is performed by parsing technique. This paper explores the novel approach that how the shift reduce parsing technique is used for translating English sentences into a grammatically correct Kannada sentences by reordering of English parse tree structure, generating and implementing phrase structure grammar(PSG) for kannada sentences. Recursive Descent Parsing technique is used to generate English phrase tree structure and terminal symbols are tagged with Kannada equivalent words then Shift-Reduce Parsing technique is used to construct a Kannada sentence. Part-of-Speech (POS) tagger is used to tag Kannada words to English words. It is implemented by using supervised machine learning approach

***Keywords****- Natural Language Processing (NLP), Phrase Structure Grammar (PSG), Part-of-Speech (POS), Shift-Reduce Parsing (Tagging).*

## I. INTRODUCTION

Perception and communication are the essential components of intelligent behavior; they provide the ability to interact effectively. Humans who reside at different regions have same perception but lack of communication language. To achieve effective interaction a system is required that translates sentences from one native language to another. Developing a program that understands natural language is difficult task due to large number of different sentences and the ambiguity in a natural language. The native languages have distinctive structural differences at phonological, morphological, lexical, syntactical and semantic levels. In this paper an attempt is made to translate English sentences into a syntactically correct sentences by using shift reduce parsing technique. Kannada is a language spoken in India predominantly in the state of Karnataka, Making it the 25th most spoken language in the world and Kannada use the “UTF-8” western windows encode and decode from their vocabulary word charecters.

## II. REVIEW OF LITERATURE

### *A.* shift-reduce parsering

A shift-reduce parser starts out with the entire input string and looks for a substring that matches the right-hand side of a production. If one is found, the substring is replaced by the left-hand side symbol of the production. Reductions occur until the string is reduced to just the start symbol. A shift reduce parsing begins with actual words appearing in the sentence therefore it is a data driven, in a shift-reduce parser it tries to find sequences of words and phrases that correspond to the right hand side of a grammar production, and replace them with the left-hand side, until the whole sentence is reduced to an S.

### *B.* Phrase structure of Kannada Language

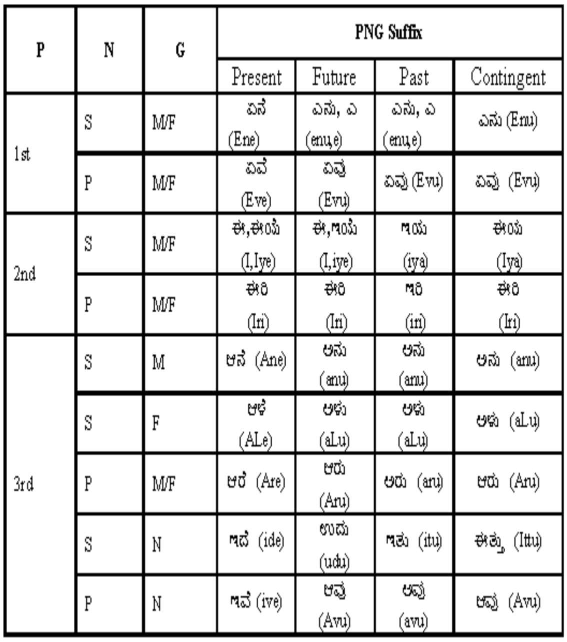
Kannada language is highly agglutinative language with three gender forms namely masculine, feminine and neutral and Word order plays an important role in positional languages like English which normally follow right-branching with Subject-Verb-Object orders where as In Kannada language is verb final language and all the noun phrases in the sentence normally appear to the left of the verb, hence it is ‘Left branching language’ and the adjectives, genitive and relative clauses precede their head nouns in a sentence.

The subject noun phrase may also appear in many different positions relative to other noun phrases in the sentence. E.g. in a sentence “Rama went to school” depict the structure of English language (i.e. Subject-verb-object) and its equivalent Kannada sentence is “rAma SAlEge hOdanu” ( ರಾಮ ಶಾಲೇಗೆ ಹೋದನು ) which depicts the structure of Kannada language (i.e. Subject-Object-Verb).

### *C.* Verb morphology in south Dravidian Languages

The PNG and the tense marker concatenated to the verb stems are the two aspect of verb morphology in South Dravidian languages. The verbal inflectional morphemes attach to the verbs providing information about the syntactic aspects like number, person, case-ending relation and tense. The PNG features of the head noun of the subject NP determine the agreement marker of the verb. The table 2.1 shows the various PNG suffixes [3] that can be attached to any Kannada verb root word

TABLE I. PNG- SUFFIXES IN KANNADA

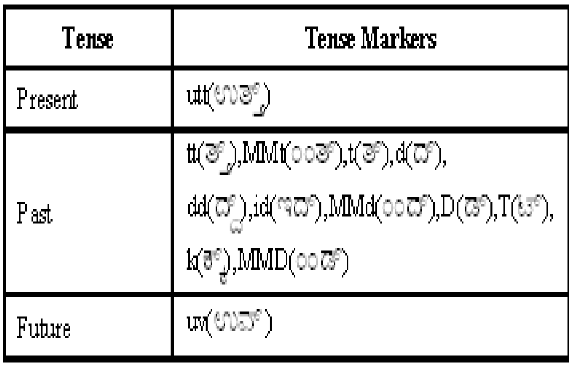


| P | N | G | PNG Suffix | | | |
|---|---|---|---|---|---|---|
| | | | Present | Future | Past | Contingent |
| 1st | S | M/F | ಏನೆ<br>(Ene) | ಎನು, ಎ<br>(enu,e) | ಎನು, ಎ<br>(enu,e) | ಎನು (Enu) |
| | P | M/F | ಏವೆ<br>(Eve) | ಏವು<br>(Evu) | ಏವು (Evu) | ಏವು (Evu) |
| 2nd | S | M/F | ಈ,ಈಯೆ<br>(I,Iye) | ಈ,ಇಯೆ<br>(I,iye) | ಇಯ<br>(iya) | ಈಯ<br>(Iya) |
| | P | M/F | ಈರಿ<br>(Iri) | ಈರಿ<br>(Iri) | ಇರಿ<br>(iri) | ಈರಿ<br>(Iri) |
| 3rd | S | M | ಆನೆ (Ane) | ಅನು<br>(anu) | ಅನು<br>(anu) | ಅನು (anu) |
| | S | F | ಆಳೆ<br>(ALe) | ಅಳು<br>(aLu) | ಅಳು<br>(aLu) | ಅಳು (aLu) |
| | P | M/F | ಆರೆ (Are) | ಆರು<br>(Aru) | ಅರು (aru) | ಆರು (Aru) |
| | S | N | ಇದೆ (ide) | ಉದು<br>(udu) | ಇತು (itu) | ಈತ್ತು (Ittu) |
| | P | N | ಇವೆ (ive) | ಆವು<br>(Avu) | ಅವು<br>(avu) | ಆವು (Avu) |

P: Person N:Number G:Gender
S: Singular P:Plural M:Masculine F:Feminine N:Neuter

All the verb words use the same present and future tense markers but all the South Dravidian languages uses different past tense markers based on the types of verb paradigms. The table 2 shows the different tense markers that are used in Kannada language [3]

TABLE II. TENSE MARKERS IN KANNADA



| Tense | Tense Markers |
|---|---|
| Present | utt(ಉತ್ತ್) |
| Past | tt(ತ್ತ್),MMt(ಂಂತ್),t(ತ್),d(ದ್),<br>dd(ದ್ದ್),id(ಇದ್),MMd(ಂಂದ್),D(ಡ್),T(ಟ್),<br>k(ಕ್ಕ್),MMD(ಂಂಡ್) |
| Future | uv(ಉವ್) |

## III. METHODOLOGY

Sentences are composed of groups of words making up phrases. Phrases may contain other phrases. Phrases fall into a small set of types, the most important of which are NP (noun phrase), PP (prepositional phrase) and VP (verb phrase). Every phrase has a 'head' word which defines its type. A simple English sentence S is composed of a noun phrase Followed by a verb phrase. In this paper a novel approach is given to translate English sentences which comprise PP to equivalent Kannada sentences.

The below architecture show the flow of operations performed during the translation of a sentences

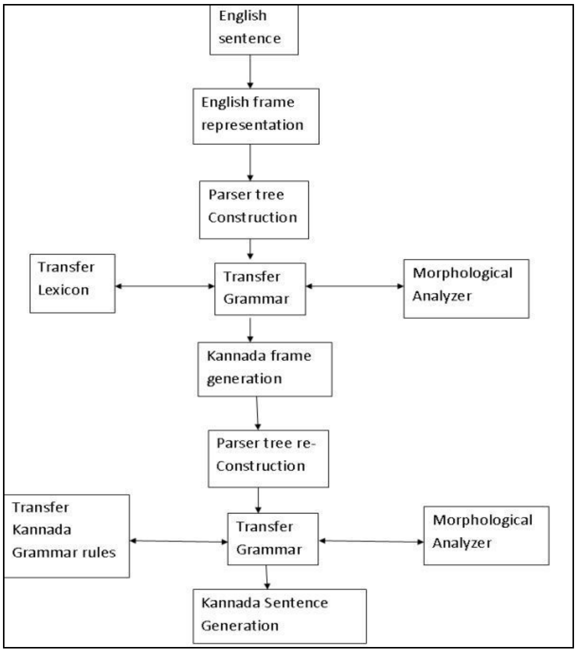


**Figure 3.1 Architecture of translation system**

The following steps are designed to produce a novel approach in translating sentences from English to Kannada by using rule-based method. The outline of the our methodology is as follows

- Parsing and structural representation.
- Morphological analysis.
- Lexicon and mapping.
- Translation.

### *A.* Parsing and Structural Representation

In this method we use shift reduce parsing technique to construct a parse tree [9] where a shift-reduce parser tries to find sequences of words and phrases that correspond to the left hand side of a grammar production, and replace them with the right-hand side, until the whole sentence is reduced to an S and the compound sentences are split with respect to conjunction by generating individual simple sentences later build constituent structures for all individual sentences.

The syntax of a language can be described by a 'formal grammar' which consists of

- A set of non-terminal symbols
- A start symbol
- A set of terminal symbols (words)
- A set of productions (also called re-write rules)

Nonterminal symbols:

S = sentence PP = prepositional phrase
NP = noun phrase DET = determiner
SNP = simple noun phrase ADJ = adjective noun
VP = verb phrase PV = preposition verb

V = Verb PRN = pronoun
VC = verb complements N=Noun

The following production rules used by the parser to make the parser tree structure for English sentences

S -> NP VP
NP -> DET SNP | PRN
SNP -> ADJ N
VP -> V VC
VC-> NP PP
PP -> PV NP

Finally the transfer rule was used to change the structure of English sentence according to Kannada chunk order as shown below

S -> NP VP
NP -> DET SNP| PRN
SNP -> ADJ N
VP -> VC V
VC-> NP PP
PP -> NP PV

After applying the transfer rules, the reordered parse tree look as shown in figure

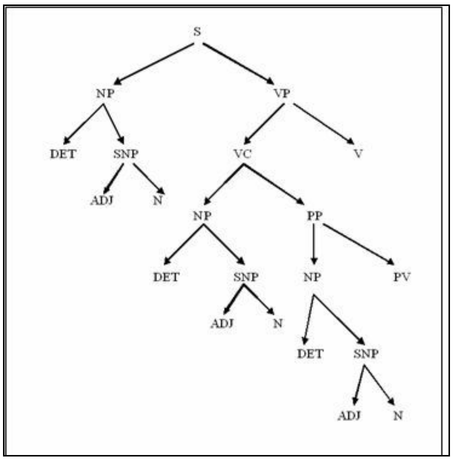


**Figure 1 Tree Structure after reordering**

*B.* Morphological analysis

Morphological analyzer and morphological generator are two essential and basic tools for building any language processing application. Morphological Analysis is the process of providing grammatical information of a word given its suffix. Morphological analyzer is a computer program which takes a word as input and produces its grammatical structure as output. A morphological analyzer will return its root/stem word along with its grammatical information depending upon its word category. For nouns it will provide gender, number, and case information and for verbs, it will be tense, aspects, and modularity

e.g. - children Child + n + s (pl) (English)

Grammatical structure–morpheme order, feature values, suffixes.

Feature value-gender, number, person etc.

Morphology deals with all combinations that form words or parts of words. Two broad classes of morphemes, stems and affixes: The stem is the “main morpheme” of the word, supplying the main meaning.

e.g., eat in eat + ing.

Affixes: an affix is a bound morph that is realized as a sequence of phonemes. Concatenative morphology (since a word is composed of a number of morphemes concatenated together) uses the following types of affixes

Prefixes: A Prefix is an affix that is attached in front of a stem. e.g.-admission- re in readmission

Suffixes: A Suffix is an affix that is attached after the stem. Suffixes are used derivationally and inflectionally.

E.g.–ing in telling

Circumfixes: A Circumfixis the combination of a prefix and a suffix which together express some feature.

Circumfixes can be viewed as really two affixes applied one after the other.

E.g. German ge--tinge + sag + t ([have] said)

In non-concatenative morphology (morphemes are combined in more complex ways) the stem morpheme is split up. The following types of affixes are used:

Infixes: Infixes are attached in between some phonemes of a stem.

Transfixes: Transfixes are a special kind of infix involves not only discontinuous affixes but also discontinuous bases.

In this method a porter stemming algorithm can be used to reduce the English words to morphemes with the some additional rules to remove suffixes [1] which are listed in Table III

TABLE III. MORPHOLOGICAL RULES

| Noun Rules | Verb Rules | Adjective Rules |
|---|---|---|
| s->null | s->null | er->e or null |
| ses ->s | ies->y | est->e or null |
| xes->x | es->e or null | |
| zes->z | ed->e or null | |
| ches->ch | ing->e or null | |
| shes->sh | | |

### i. The Role Of Morphology In Different Languages

Morphology is not equally prominent in all spoken languages. What one language expresses morphologically [12] may be expressed by a separate word or left implicit in another language. For example, English expresses the plural nouns by means of morphology (the forms like boys, spies, vehicles where the morpheme, with its variant forms expresses the plurality) but Yoruba (a language of south-western Nigeria) use separate word expressing the same meaning. Thus, 'ookunrin' means the man, and 'a won' can be used to express the plural: 'the men'. Quite generally, we can say that English makes more use of morphology than Yoruba. But there are many languages that make more use of morphology than English. For instance Sumerian uses Morphology to distinguish between 'he went' and 'I went', and between 'he went' and 'he went to him', where English must use separate words. The terms analytic and synthetic are used to describe the degree to which morphology is made use in a language. Languages like Yoruba, Vietnamese or English, where morphology plays a relatively modest role are called analytic. Traditionally, linguists discriminate between the following types of languages types of languages with regard to morphology:

- Isolating languages (e.g. Mandarin Chinese): there are no bound forms. E.g., no affixes that can be attached to a word. The only morphological operation is composition.

- Agglutinative languages (e.g.Ugro-Finnicand Turkic languages): all bound forms are either prefixes or suffixes, i.e., they are added to a stem like beads on a string. Every affix represents a distinct morphological feature. Every feature is expressed by exactly one affix.

- Inflectional languages (e.g. Indo-European language): distinct features are merged into a single bound form (a so called portmanteau morph). The same underlying feature may be expressed differently, depending on the paradigm.

- Polysynthetic languages (e.g. Limit language): these languages express more of syntax in morphology than other languages, e.g., verb arguments are incorporated into the verb. This classification is quite artificial. Real languages rarely fall cleanly into one of the above classes, e.g., even Mandarin has a few suffixes. Moreover, this classification mixes the aspect of what is expressed morphologically and the means expressing it.

### *C.* Lexicon and mapping

In our approach lexicon is arranged in alphabetical order and Keep a separate lexicon that contains frequently used words as well as a domain specific components of words can be kept in different lexicon to increase the efficiency of search process [6]. Once the word is matched in the lexicon, it is tagged to respective Kannada equivalent word which is defined in the lexicon which is in the same language using one to one mapping technique and the following algorithm [1] can be used to perform tagging

```
Input: Untagged English Sentence
Output: Tagged Translated Kannada Sentence
Tag<= First Word in Sentence
For each word in the Sentence Do
If the Word is tagged
Stop
Else
Tag<=Word
End if
End For
Return Tagged Translated Kannada Sentence
```

### *D.* Translation

Once mapping (tagging) is completed, the resultant Kannada sentence is obtained which is not in proper meaning so next step is to apply some rules to make it a meaningful Kannada sentence. If the input English sentence is a simple sentence then these rules can be ignored. A few rules are listed here that are applied when an English sentence consists of Prepositional Phrase and these are known to be inflections to the noun stem.

Rule 1: add suffix "annu" (C£ÀÄß ) to noun phrase of the VP root.

Rule 2: add suffix "da" (CzÀ) to noun phrase of the PP root when "with" preposition encountered.

Rule 3: add suffix "ige" (EUÉ) to noun phrase of the PP root when "to" preposition encountered.

Rule 4: add suffix "alli" (C°è) to noun phrase of the PP root when "in" preposition encountered.

E.g. A simple translation of an English Sentence "The dog saw a man in the park" to equivalent Kannada sentence "nAyi oMdu manuShyanannu pArkinalli nODitu"

(ನಾಯಿ ಒಂದು ಮನುಷ್ಯನನ್ನು ಪಾರ್ಕಿನಲ್ಲಿ ನೋಡಿತು)

## IV. CONCLUSION AND FUTURE WORK

In this paper we have discussed how Machine translation from English to Kannada can be achieved with the use of prepositional phrase(PP). The different NLP techniques are described for Machine translation such as Morphological analysis for stemming which will increase the information retrieval accuracy and minimizes the dictionary words for lexicon mapping. Part of Speech Taggers are used for resolving the ambiguity in the Sentence for translation for English to Kannada and it also help for resolving the problem of Sentence format such as English have Subject-Verb-Object Format where as Kannada have Subject-

Object- Verb format. In future this will help to develop the other modules for Different Indian Languages and this approach can be used to build the module which convert entire document from english to kannada.

## ACKNOWLEDGMENT

I owe my sincere feelings of gratitude to Dr. M. Hanumanthappa, Associate Professor Department of Computer Science and Applications for his valuable guidance and suggestions which helped me a lot to write this paper.

## REFERENCES

[1] Mallamma V Reddy, Dr.M. Hanumanthappa, "POS Tagger for Kannada Sentence Translation" , International Journal of Emerging Trends & Technology in Computer Science(IJETTCS) Volume 1, Issue 1, May-June 2012

[2] Natural Language Processing K.R. Chowdhary Professor & Head CSE Dept. M.B.M. Engineering College, Jodhpur, India April 29, 2012

[3] Unnikrishnan P, Antony P J, Dr Soman K P, "A Novel Approach for English to South Dravidian Language Statistical Machine Translation System", International Journal on computer Science and engineering (IJCSE) Vol 02, No. 08, 2010,2749-2759.

[4] K. Narayana Murthy, "Computer Processing of Kannada Language", University of Hyderabad.

[5] R.M.K. Sinha and Anil Thakur, "Synthesizing Verb Form in English to Hindi Translation", Case of Mapping Infinitive and Gerund in English to Hindi, Proceedings of International Symposium on Machine Translation, NLP and Translation Support System (iSTRANS- 2004), November 17-19, 2004, Tata Mc Graw Hill, New Delhi, pp: 52-55.

[6] Amitabha Mukerjee, Achla Raina, Pankaj Goyal,and Pushpraj Shukla, A unified Computational Lexicon for Hindi-English code-switching Proceedings International Conference on Natural Language Processing (ICON), Hyderabad, India, December 19-22, 2004.

[7] Blaheta, Don, and Eugene Charniak. 2000. Assigning function tags to parsed text. In Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics, pages 234–240, Seattle.

[8] Hai Zhao, Yan Song, Chunyu Kit, and Guodong Zhou. 2009. Cross language dependency parsing using a bilingual lexicon. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1 (ACL '09), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 55-63.

[9] B.S. Baker 1979. Composition of top-down and bottom-up tree transductions Inform. and Control, 41(2):186–213

[10] K. Knight. 2007. Capturing practical natural language transformations. Machine Translation 21, 121–133.

[11] Takuya Matsuzaki, Yusuke Miyao and Junichi Tsujii. 2007. Efficient HPSG Parsing with Supertagging and CFG-filtering. Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)

[12] Ritchie, Graeme. The Lexicon. In Whitelock, eds.1985.p. 225-256

◈◈◈