A Novel Approach to Morphological Analysis for Tamil Language

(தமிழ் உருபனியல் ஆய்வில் ஒரு புதிய அணுகுமுறை)

Anand kumar M[1], Dhanalakshmi V[1], Rajendran S[2], Soman K P[1]
{m_anandkumar, v_dhanalakshmi, kp_soman} @ettimadai.amrita.edu, raj_ushus@yahoo.com

[1]Amrita Vishwa Vidyapeetham, Ettimadai, Coimbatore, Tamilnadu, India .
[2]Tamil University, Thanjavur, Tamilnadu, India .

Abstract

This paper presents the morphological analysis for complex agglutinative Tamil language using machine learning approach. Morphological analysis is concerned with retrieving the structure, syntactic rules, morphological properties and the meaning of a morphologically complex word. The morphological structure of an agglutinative language is unique and capturing its complexity in a machine analyzable and generatable format is a challenging job. Generally rule based approach is used in building morphological analyzer. In rule based approach what works in the forward direction may not work in the backward direction. The Novel approach to morphological analyzer is based on sequence labeling and training by kernel methods. It captures the non-linear relationships and various morphological features of Tamil language in a better and simpler way. The efficiency of our system is compared with the existing morphological analyzers which are available in net. Regarding the accuracy our system significantly outperforms the existing morphological analyzer and achieves a very competitive accuracy of 95.65% for Tamil language.

1 Introduction

Morphological analysis is the process of segmenting words into morphemes and analyzing the word formation. It is a primary step for various types of text analysis of any language. Morphological analyzers are used in search engines for retrieving the documents from the keyword (Daelemans Walter et al., 2004). Morphological analyzer increases the recall of search engines. It is also used in speech synthesizer, speech recognizer, lemmatization, noun decompounding, spell and grammar checker and machine translation.

Tamil language is morphologically rich and agglutinative. Each root word is affixed with several morphemes to generate word forms. Generally, Tamil language is postpositionally inflected to the root word. Computationally each root word can take a few thousand inflected word forms, out of which only a few hundred will exist in a typical corpus. For the purpose of analysis of such inflectionally rich languages, the root and the morphemes of each word have to be identified. Generally rule based approaches are used for building morphological analyzer (Rajendran.S et al., 2001). We have implemented a novel method for the morphological analysis of the Tamil language using machine learning approach.

2 Challenges in Morphological Analyzer for Tamil

The morphological structure of Tamil is quite complex since it inflect to person, gender, and number markings and also combines with auxiliaries that indicate aspect, mood, causation, attitude etc in verb. Noun inflects with plural, oblique, case, postpositions and clitics suffixes. For the purpose of analysis of such inflectionally rich languages, the root and the morphemes of each word have to be identified. The structure of verbal complex is unique and capturing this complexity in a machine analyzable and generatable format is a challenging job. The formation of the verbal complex involves arrangement of the verbal units and the interpretation of their combinatory meaning. Phonology also plays its part in the formation of verbal complex in terms of morphophonemic or sandi rules which account for the shape changes due to inflection. Classification of Tamil verbs based on tense inflections is evolved. The inflection includes finite, infinite, adjectival, adverbial and conditional forms of verbs (Rjendran.S et al., 2001). For the computational need we have made thirty two paradigms of verb.

Compared to verb morphological analysis noun morphological analysis is less challenging. Noun can occur separately or with plural, oblique, case, postpositions and clitics suffixes. We have classified seventeen noun paradigms to resolve the challenges in noun morphological analysis. Based on the paradigm we classified the root words into its group. We have prepared the corpus with all morphological feature information. So the machine by itself captures all morphological rules. Finally the morphological analysis is redefined as a classification task which is solved by using sequence labeling approaches.

3 Creating Data for supervised Learning

Nowadays machine learning approaches are directly applied to all the natural language processing tasks machine learning approaches can be supervised or unsupervised. In supervised learning set of input and output examples are used for training. So, data creation plays the key role in supervised machine learning approaches.

The first step involved in the corpora development for morphological analyzer is classifying paradigms for verbs and nouns. The classification of Tamil verbs and nouns are based on tense markers and case markers respectively. Each paradigm root word will inflect with the same set of inflections. The second step is to collect the word list for each noun and verb paradigm. Figure 1 explains the preprocessing steps involved in the development of morphological corpus.

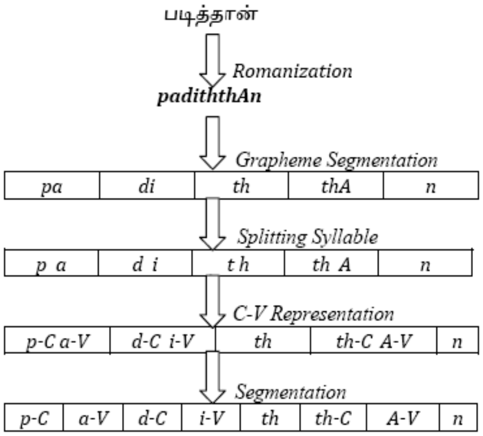

Figure 1: Preprocessing steps

Morphological corpus which is used for machine learning is developed by following steps:

Romanization: The set of most commonly used noun and verb forms are generated manually for input structure and similarly the output structure is developed. These data are converted to Romanized forms using the Unicode to Roman mapping file.

Segmentation: After Romanization each and every word in the corpora is segmented based on the Tamil grapheme and additionally each syllable in the corresponding word is further segmented into consonants and vowels. In segmented syllable append “*–C*” and “*–V*” to the consonant and vowel respectively. We name it as C-V representation i.e. Consonant – Vowel representation. Morpheme boundaries are indicated by “*” symbol in output data.

Alignment and mapping: The segmented words are aligned vertically as segments using the gap between them. And the input segments are consequently mapped with output segments. Sample data format is given in the fig.2 .First one represents the input data and the latter one represents output data.”*” indicates the morpheme boundaries.

Input	p-C a-V d-C i-V th th-C A-V n
Output	p a d i* th th* A n*

Figure 2: Sample Data Format

Mismatching: It is the key problem in mapping between the input and output data. Mismatching occurs in two cases i.e., either the input units are larger or smaller than that of the output units. This problem is solved by inserting null symbol “$” or combining two units based on the morph-syntactic rules to the output data. And the input segments are mapped with output segments. After mapping machine learning tool is used for training the data.

Case 1:

Input Sequence:

P-C | a-***V*** | d-***C*** | i-***V*** | k | k-***C*** | a-***V*** | y-***C*** | i-***V*** | y-***C*** |a-***V*** | l-***C*** | u-***V*** | m (14 segments)

Mismatched Labels:

p | a | d | i* | k | k | a* | i | y | a | l* | u | m* (13 segments)

Corrected labels:

p | a | d | i* | k | k | a* | **$** | i | y | a | l* | u | m* (14 segments)

In case 1 input sequence is having more number of segments than the output sequence. For the Tamil verb *padikkayiyalum* is having 14 segments in input sequence but in output it has only 13 segments. The second occurrence of “*y*” in the input sequence becomes null due to the morphosyntactic rule. So there is no segment to map with “y”. For this reason, in training data “*y*” is mapped with “$” symbol (“$” indicates null).Now the input and the output segments are equalized.

Case 2:

Input Sequence:

O | d-***C*** | i-***V*** | n-***C*** | A-***V*** | n (6 segments)

Mismatched Labels:

O | d | u* | i | n* | A | n (7 segments)

Corrected labels:

O | **du*** | i | n* | A | n (6 segments)

In case 2 the input sequence is having less number of segments than the output sequence. Tamil verb *OdinAn* is having 6 segments in input sequence but output has 7 segments. Due to morphosyntactic change the segment “*d-C”* in the input sequence is mapped to two segments “*d” &”’u*”* in output sequence. For this reason, in training “*d-C”* is mapped with “*du**”.Now the input and the output segments are equalized and thus the problem of sequence mismatching is solved.

4 Implementation of Morphological Analyzer

Using machine learning approach the morphological analyzer for Tamil is developed. We have developed separate engines for noun and verb. Morphological analyzer is redefined as a classification task using the machine learning approaches. Three phases are involved in our morphological analyzer.

- Preprocessing
- Segmentation of morphemes
- Identifying morpheme

Fig.3 gives an outlook of the morphological analyzer system. In this machine learning approach two training models are created for morphological analyzer. These two models are represented as model-I and model-II. First model is trained using the sequence of input characters and their corresponding output labels. This trained model-I is used for finding the morpheme boundaries. Second model is trained using sequence of morphemes and their grammatical categories. This trained model-II is used for assigning grammatical classes to each morpheme.

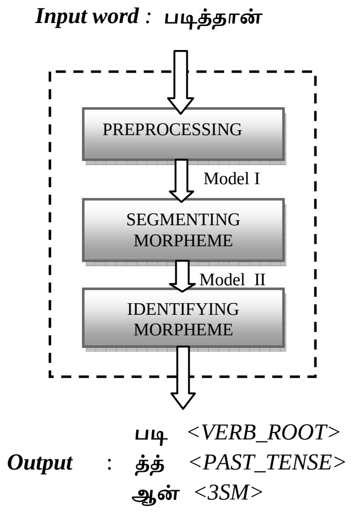

Figure 3: Schematic Representation

Preprocessing: The surface form is converted into sequence of units which is given as the input for the morphological analyzer tool. The input word is converted into input segments and syllables are identified. Using C-V representation consonants and vowels are represented to the syllable.

Segmentation of Morphemes: Preprocessed words are segmented into morpheme according to the morpheme boundary. The input sequence is given to the trained model-I. The trained model predicts each label to the input segments.

Identifying Morpheme: The Segmented morphemes are given to the trained model-II. It predicts grammatical categories to the segmented morphemes. We have trained the system to give multiple outputs to handle the compound words.

5 Morphological Analyzer Using Machine Learning

In morphological analysis a complex word form is transformed into root and suffixes. Generally rule based approaches are used for morphological analysis which are based on a set of rules and dictionary that contains root and morphemes. For example a complex word is given as an input to the morphological analyzer and if the corresponding morphemes are missing in the dictionary then the rule based system fails (Daelemans Walter et al., 2004). In rule based approaches every rule is depends on the previous rule. So if one rule fails, it will affect the entire rule that follows. In machine learning all the rules including complex spelling rules are also handled by the classification task. Machine learning approaches don’t require any hand coded morphological rules (Daelemans Walter et al., 2004). It needs only corpora with linguistical information. These morphological or linguistical rules are automatically extracted from the annotated corpora. Here input is a word and output is root and inflections. Input word is denoted as ‘W’ root and inflections are denoted by ‘R’ and ‘I’ respectively.

$[W]_{Noun/Verb} = [R]_{Noun/Verb} + [I]_{Noun/Verb}$

5.1 Machine learning using SVM

Support vector approaches have been around since the mid 1990s, initially as a binary classification technique, with later extensions to regression and multi-class classification. Here Morphological problem is converted into classification problem. These classifications can be done through supervised machine learning approach.

Support Vector Machine is a new approach to supervised pattern classification which has been successfully applied to a wide range of classification problems. SVM is based on strong mathematical foundations and results in simple yet very powerful algorithms. SVMs are learning systems that use a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory.

5.2 SVMTool

SVMTool is an open source generator of sequential taggers based on Support Vector Machine. Generally SVMTool is developed for POS tagging but here this tool is used in morphological analyzer for classification task. The SVMTool software package consists of three main components, namely the model learner (SVMTlearn), the tagger (SVMTagger) and the evaluator (SVMTeval). SVM models (weight vectors and biases) are learned from a training corpus using the SVMTlearn component (Jes´us Gim´enez and Llu´ıs M`arquez, 2006). Different models are learned for the different strategies. Finally, given a correctly annotated corpus, and the corresponding SVMTool predicted annotation, the SVMTeval component displays tagging results. SVMTeval evaluates the performance in terms of accuracy.

6 System Evaluation

Efficiency of the system is compared. The morphological analyzer system for verb and noun are trained with 130,000 and 70,000 words respectively. This system is also tested with 40,000 verb forms and 30,000 nouns from an Amrita POS Tagged corpus (Dhanalakshmi.V et al., 2008). The SVM based machine learning tool affords better results compare to MBT and CRF++. Training time is very less in MBT compare to SVM and CRF++. But in testing SVM holds good. The outputs which are incorrect are noticed and its corresponding input words are added in the training file and trained again. This increases the efficiency of our system. This is the main advantage of using machine learning approach to rule based approach.

7 Conclusion

This paper has described the morphological analyzer based on the new and state of the art machine learning approaches. We have demonstrated a new methodology adopted for the preparation of the data which was used for the machine learning approaches. We have not used any morpheme dictionary but from the training model our system has identified the morpheme boundaries. The accuracy obtained from the different machine learning tools shows that SVM based machine learning tool gives better result than other machine learning tools. A GUI to enhance the user friendliness of the morphological analyzer engine was also developed using Java Net Beans. We are currently implementing the same methodology for the other Dravidian languages like Malayalam, Telugu, and Kannada. Preliminary experimentation gave promising results. We are confident that the proposed method is general enough to be applied for any languages.

References

Anandan. P, Ranjani Parthasarathy, Geetha T.V.2002. *Morphological Analyzer for Tamil*, ICON 2002, RCILTS-Tamil, Anna University, India.

Daelemans Walter, G. Booij, Ch. Lehmann, and J. Mugdan (eds.)2004 , *Morphology. A Handbook on Inflection and Word Formation*, Berlin and New York: Walter De Gruyter, 1893-1900

Dhanalakshmi V, Anandkumar M, Vijaya M.S, Loganathan R, Soman K.P, Rajendran S,2008, *Tamil Part-of-Speech tagger based on SVMTool*, Proceedings of the COLIPS International Conference on Asian Language Processing 2008 (IALP), Chiang Mai, Thailand. 2008: 59-64.

Jes´us Gim´enez and Llu´ıs M`arquez,2006 *SVMTool:Technical manual v1.3*, August 2006.

John Goldsmith. 2001. *Unsupervised Learning of the Morphology of a Natural Language*. Computational Linguistics, 27(2):153–198.

Rajendran, S., Arulmozi, S., Ramesh Kumar, Viswanathan, S. 2001. *Computational morphology of verbal complex*. Paper read in Conference at Dravidan University, Kuppam, December 26-29, 2001.