Natural Language Processing Tools for Tamil Grammar Learning and Teaching

Dhanalakshmi V,
Anand Kumar M, Soman K.P,
CEN, Amrita Vishwa
Vidyapeetham Coimbatore,India

Rajendran S
Dept. of Linguistics
Tamil University
Thanjavur, India

ABSTRACT
Today we are living in the world of communication. The world of communication interlinks everyone through its various media. In this aspect Computers play a major role by bringing the world under the user's finger tip. Grammar is the legal advocacy to the human art of communication. But learners get annoyed with the language rules and the old teaching methodology. Interlinking the computer to the language through Natural language Processing (NLP) paves a way to solve this problem. The innovative NLP applications are used to generate language learning and teaching tools which enhance the teaching and learning of Grammar. In this paper we present the Grammar teaching tools for analyzing and learning character, word and sentence of Tamil Language. Tools like Character Analyzer for analyzing character, Morphological Analyzer and Generator and Verb Conjugator for the word level analysis and Parts of Speech Tagger, Chunker and Dependency parser for the sentence level analysis were developed using machine learning based technology. These tools are very useful for second language learners to understand the character, word and sentence construction of Tamil language in a non-conceptual way.

General Terms
Tamil grammar, Agglutinative language, Natural Language processing, Machine Translation.

Keywords
Grammar Learning and Teaching, Machine Learning, Character Analyzer, Morphological Analyzer and Generator, Verb Conjugator, Parts of Speech Tagger and Chunker , Dependency parser.

1. INTRODUCTION
We are now living in the world of communication. Grammar plays an important role in good communication. But Learners get annoyed with the language rules and the old teaching methodology. Today the computer aided teaching technologies are widely used to increase the learning ability of the learner. Learners also get benefit from the immediate feedback provided by computers and most of them appreciate this self-paced learning environment. At its best, CAI engages learners interest, motivates them to learn, and increases their personal responsibility for learning [1].The field of Natural Language Processing is developing steadily with the well advancement in the Artificial Intelligence application. The innovative NLP applications are used to generate Language learning and teaching tools which enhance the learning and teaching of Grammar.

The major draw back of most language learning and teaching software is that it is not automated. They use a small set of predefined sentences that cannot be modified or replaced. This makes the system monotonous for the user. But here we have automated the system with the Natural Language Processing tools. The NLP tools were developed using machine learning approaches. The key thing needed for the machine learning process is the linguistically annotated data. This linguistically annotated corpus is very useful for creating the language learning and teaching tools. There is no such annotated corpus for Indian languages, especially for Tamil. We have developed an annotated corpus for POS tagger, Chunker, Morphological Analyzer and Dependency Parser. Using our machine learning based NLP Tools and the annotated corpus we have developed the Grammar learning and teaching Tools for Tamil language.

Our automatic Character Analyzer, Morphological analyzer and generator will help the learners to learn the structural construction and generation of the words and sentence in Tamil. When the user gives the input sentence, the system automatically analyzes the Parts of Speech of each word, Phrase and Dependency relationship between the words present in the sentence. It is also used to generate the word using the users morpholexical input to the morphological generator. Verb Conjugator automatically generates various word forms (lexemes) for a given root word .This non-conceptual way of study enhances the grammar learning ability of the user. User Interfaces were developed for the practical usage of the tools.

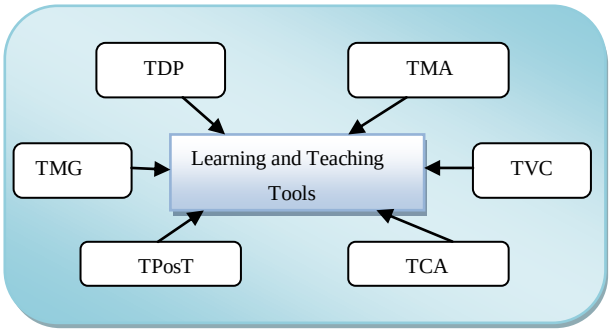
2. ANNOTATED CORPORA CREATION
In language teaching tools annotated corpus plays an important role. Annotation of corpora is needed in sentence level as well as in word level. In the sentence level, we have annotated corpus for Parts of Speech, Phrase identification and Dependency relationship between the words in a sentence. In the word level, Lemma and morpheme annotation was done. Part of speech tagging forms the basic step towards building an annotated corpus at sentence level. For creating Tamil Part of Speech Tagger, we have grammatically tagged corpus size of 3, 50,000 words. We have collected sentences from Dinamani news paper, yahoo Tamil news and Tamil short stories etc and tagged its POS categories [2]. Chunking the phrase form the next level of tagging. We have chunked the corpus size of 3, 50,000 words. For Dependency parser we have manually annotated corpus size of 25,000 words.

In an agglutinative language like Tamil the words are highly inflected with morphemes. For the word analysis we have segmented the morphemes and tagged its morphological categories [3]. Our corpora contain 7 lakhs morphologically tagged words. These grammatically annotated corpora are used for language teaching and learning.

3. NLP TOOLS FOR TAMIL GRAMMAR LEARNING AND TEACHING

Tamil being a Dravidian language has a very rich morphological structure which is agglutinative. Tamil words are made up of lexical roots followed by one or more affixes. Tamil grammar is broadly classified into *ezhuttu*, *col*, and *porul*. *ezhuttu* deals with the letters or the characters , *col* deals with the morphological features of words and *porul* deals with the meaning. We have developed the NLP Tools like Character analyzer to analyze each and every character or letter, Morphological Analyzer and Generator for the word level analysis and Parts of speech Tagger, Chunker and Dependency parser for the sentence level analysis.

This section discusses about the NLP tools which we have created for grammar teaching and learning. General components of Grammar Learning Tools are given in "Figure 1".



"Figure 1. NLP Tools for Grammar Learning"

- Tamil Character Analyzer (TCA)
- Tamil POS Tagger (TPosT) and Chunker (TC)
- Tamil Dependency Parser (TDP)
- Tamil Morphological Analyzer (TMA)
- Tamil Morphological generator (TMG) and Verb conjugator (TVC)

4. TAMIL CHARACTER ANALYZER (TCA)

Tamil Character analyzer analysis each and every character in a word. It predicts whether the character is a vowel or consonant or a syllable in the basic level. In the next level, it analysis if it is an alphabet whether it belongs to short or long vowel, if it is a consonant whether it belongs to *mellinam*, *vallinam* or *idaiyinam*. It also predicts *Mathirai* for the Tamil characters. *mathiri* represents how much time taken to pronounce a particular character. This analyzer will be helpful for learners to understand the vowels, consonants and their types. "Figure 2" shows the user interface of Tamil Character analyzer.

"Figure 2. GUI for Character Analyzer"

5. TAMIL PART-OF-SPEECH TAGGER (TPosT) AND CHUNKER (TC)

Part Of Speech tagging and chunking are the fundamental processing steps for any language processing task. Part of speech (POS) tagging is the process of labeling automatic annotation of syntactic categories for each word in a corpus [5]. Chunking is the task of identifying and segmenting the text into syntactically correlated word groups. These are done by the machine learning techniques, where the linguistical knowledge is automatically extracted from the annotated corpus. The capability for a tool to automatically POS tag and chunk a sentence is very essential for further analysis in many approaches to the field of NLP.

5.1 POS Tagging using SVM

The POS Tagging problem in this context becomes a multi-class classification problem. Since SVMs in general are binary classifiers, a binarization of the problem is performed. Here SVMTool [8] is trained for every POS tag in order to distinguish between examples of this class and all the rest. When tagging a word, the most possible tag according to the predictions of all binary SVMs is selected. We have considered a centered window of five tokens, from which basic and n-gram patterns are evaluated to form binary features [7]. Two previous tags are used as POS features. The suffix and prefix information are also considered. Hence the tag of a word in consideration w_0 depends on the following.

Features:

1. The word features namely, w_{-1}, w_{-2}, w_0, w_1, w_2

2. The POS features p_{-1}, p_{-2} ,p_0 ,.p_1 ,p_2

3. Prefix and suffix features of (Current word) w_0

These are extracted and encoded as binary features for training the model using support vector machine. ”Figure 3” shows the user interface of Tamil POS-Tagger and Chunker.

“Figure 3. GUI for POS Tagger and Chunker”

5.2 Chunking using CRF

A subsequent step after POS tagging focuses on the identification of basic structural relations between groups of words in a sentence. This recognition is usually referred to as chunking. It is essential for many NLP tasks such as structure identification, information extraction, parsing and phrase based machine translation system. Chunker divides a sentence into its major-non-overlapping phrases and attaches a label to each chunk. Chunking falls between tagging and parsing. Many of the machine learning techniques and algorithms are used in this task. We have used CRF for performing this task.

Conditional Random Fields is a machine learning technique [9].CRF provides a framework for building probabilistic models for segmenting and sequence labeling data. CRFs are used for sequence tagging tasks where a sequence of words must be annotated with a sequence of labels, one per word. CRF has the advantages of MEMMs and it also solves the label bias problem. And also overcome the disadvantages of hidden Markov models and stochastic grammars.

Consider X is a random variable over data sequences to be labeled, and Y is a random variable over corresponding label sequences. All components Yi of Y are assumed to range over a finite label alphabet y. For example, X might range over natural language sentences and Y range over part-of-speech tagging of those sentences, with Y the set of possible part-of-speech tags. The random variable X and Y are jointly distributed, but in a discriminative framework a conditional model is constructed as p (Y|X) from paired observation and label sequence [9].Lafferty et al, defines Conditional random fields as follows: "Let G = (V, E) be a graph such that so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X, the random variable obeys the Markov property with respect to the graph.

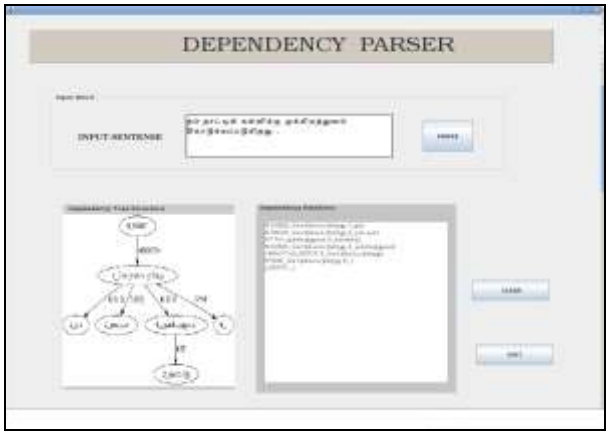
Example:

{*That Beautiful Girl*}
Sentence : அந்த அழகான பெண்
POStag : **DET ADJ NN**
Chunk Tag : **[Noun Phrase]**

6. TAMIL DEPENDENCY PARSING

Dependency grammar has until recently played a fairly marginal role both in theoretical linguistics and in natural language processing. Dependency grammar has largely developed as a form for syntactic representation used by traditional grammarians, the fundamental notion of dependency is based on the idea that the syntactic structure of a sentence consists of binary asymmetrical relations between the words of the sentence [3]. A dependency relation holds between a head and a dependent. Parsing in natural language context is a process of recognizing structure of an input sentence and assigning syntactic structure to it. The important thing of parser is that it should be able to solve ambiguities in various structures. "Fig.3" shows the interface for Dependency parser. Here input is a sentence and output is a dependency Tree and Dependency relations.

It is also used in Machine Translation, Grammar checking, question answering and information extraction systems. The dependency parsing approach here uses linguistic information to give relationship between words. The use of machine learning approach arises from the fact that rules are learned automatically from annotated data using learning and parsing algorithms to learn models and make predictions. The motive behind this work is to find relation between words and go to the next stage of disambiguation of word sense and also to find predicate argument structure. Dependency Parser for Tamil is developed using Machine Learning based MST parser and MALT parser. We have created an annotated corpora size of 25,000 words for Dependency Parser tool. This Tool is very useful for learner to understand the Structure of the sentence and dependency relationships between words. ”Figure 4” shows the user interface of Tamil Dependency parser.



“Figure 4. GUI for Dependency Parsing”

7. TAMIL MORPHOLOGICAL ANALYZER (TMA)

Morphological analysis is concerned with retrieving the structure, syntactic rules, morphological properties and the meaning of a morphologically complex word. It 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. It is also used in speech synthesizer, speech recognizer, lemmatization, noun decompounding, spell and grammar checker and machine translation.

7.1 Morphological analyzer using Machine Learning

Our Novel approach to morphological analyzer is based on sequence labeling and training by kernel methods [10]. It captures the non-linear relationships and various morphological features of Natural language in a better and simpler way [6]. 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.

Our morphological analyzer and the morphologically tagged words are used to increase vocabulary knowledge of the learners. From this tool the learners learn the morpheme, the smallest meaningful unit of grammar and its morphotactic construction. In addition the learners also understand the differences between a root word and suffixes. This is very important in academic word learning process [2]. The SVMTool is an open source generator of sequential taggers based on Support Vector Machine[8]. Generally SVMTool is developed for POS tagging but here this tool is used in morphological analysis for classification.

"Figure 5" shows the user interface of Tamil Morphological analyzer .

Example:{*studied*}

படித்தான் = படி <Verb-Root>
த்த் <Past-Tense>
ஆன் <3SM>

"Figure 5. GUI for Morphological Analyzer"

8. TAMIL MORPHOLOGICAL GENERATOR (TMG) AND CONJUCATER (TVC)

8.1 Morphological Generator

The Morphological Generator takes lemma and a Morpho-lexical description as input and gives a word-form as output. It is a reverse process of Morphological Analyzer. Morphological generator system implemented here is a new data driven approach which is simple, efficient and does not require any rules and morpheme dictionary [11]. There are two options available in this morphological Generator tool. As a first option User selects any Morpho-lexical information along with root word. This tool will produce intended word form corresponding to the user's input. The second option available will generate all the word forms of a particular word. If the user gives the root word or lemma as an input this tool it will automatically generate all the possible word forms (1800 for verbs and 313 for Noun). This tool is also used for Sentence Generator. So this can be very useful for teaching languages. Using this tool alone learners can understand the different word forms and morphological transformations of words in Tamil. This tool is very helpful for improving vocabulary.

8.2 Verb Conjugator

It is an application tool of Morphological generator. Conjugation is the traditional name of a group of verbs that share a similar conjugation pattern in a particular language. Verb conjugator conjugates the verb root and yields all the word forms for that particular root word. The main advantage of this tool over the morphological generator is that for a morphological generator the user should give the input information with linguistic knowledge, but for verb conjugator they can simply give a root word and get all the conjugate forms. It can conjugate for Tense, Participle, Modal, and Auxiliary forms. For Tense and Auxiliary forms it can generate along with the pronoun.

"Figure 6. GUI for Verb Conjugator"

9. APPLICATIONS

Grammar Teaching

It enhances the learners to:

- Improve vocabulary.
- Computer Assisted Learning/Teaching
- Improve writing and reading skills.
- Visualize grammatical structure of a sentence
- Develop interest in grammar
- Speed up the Learning of Second language

These NLP tools are also used in developing the following

Systems:

- Spell checker
- Information extraction and retrieval
- Simple Machine Translation system
- Grammar checker
- Content analysis
- Question answering system
- Automatic sentence Analyzer/generator
- Speech and Dialogue system
- Knowledge representation in learning
- Automatic assessment tool

10. CONCLUSION AND FUTURE WORK

In this paper we have described the development and creation of NLP Tools and annotated corpora which in turn aid in the Grammar Teaching and learning of Tamil language. Machine Learning techniques are playing an important role in the field of NLP. Due to the well advancement of these systems linguistical tools were developed with the grammatically annotated corpus. Both the NLP tools and the annotated corpus are interlinked to produce a healthy tool to teach and learn grammar of Tamil language at character, word and sentence level. Currently we are also developing these tools for Malayalam, Kannada and Telugu languages. In future, using these tools we are going to develop a simple Machine translation system between English to Dravidian languages and educational tools for learners to enhance the effortless language learning.

11. ACKNOWLEDGEMENT

This work was part of the "Creation of Machine Translation Tools and resources for English to Dravidian Languages" project funded by MHRD, Government of India. We would like to thank MHRD for the successful completion of this work.

12. REFERENCES

[1] David Collins, Alan Deck,Myra McCrickard "Computer Aided Instruction: A Study Of Student Evaluations And Academic Performance", Journal of College Teaching & Learning – November 2008 , Volume 5, Number 11

[2] Bellomo, T. (2009, April)." Morphological analysis and vocabulary development: Critical criteria." Reading Matrix, 9(1),44-55: http://www.readingmatrix.com/articles/bellomo/article.pdf

[3] Joakim Nivre, "Dependency Grammar and Dependency Parsing"

[4] http://stp.lingfil.uu.se/~nivre/docs/05133.pdf

[5] Dhanalakshmi V., Padmavathy P., Anand Kumar M., Soman K.P., Rajendran S., "Chunker for Tamil," artcom, pp.436-438, 2009 International Conference on Advances in Recent Technologies in Communication and Computing, 2009, IEEE Press,doi: 10.1109/ARTCom.2009.191

[6] Dhanalakshmi V., Anand Kumar M., Rekha R.U., Arun Kumar C., Soman K.P., Rajendran S.,"Morphological Analyzer for Agglutinative Languages Using Machine Learning Approaches," artcom, pp.433-435, 2009 International Conference on Advances in Recent Technologies in Communication and Computing, 2009,IEEE Press,doi: 10.1109/ARTCom.2009.184

[7] Dhanalakshmi V, Anandkumar M, Vijaya M.S, Loganathan R, Soman K.P, Rajendran S, "Tamil Part-of-Speech tagger based on SVMTool", In Proceedings of the COLIPS International Conference on natural language processing(IALP), Chiang Mai, Thailand. 2008.

[8] Jes´us Gim´enez and Llu´ıs M`arquez.(2004) "SVMTool: A general pos tagger generator based on support vector machines". In Proceedings of the 4th LREC Conference, 2004.

[9] Lafferty J, McCallum A, Pereira F. 2001. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". Proceedings of ICML: 282-289.

[10] Anand Kumar M, Dhanalakshmi V, Soman K.P and Rajendran S. "A Sequence Labeling Approach to Morphological Analyzer for Tamil Language", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 06, 2010, 2201-2208.

[11] Anand Kumar M, Dhanalakshmi V, Rekha R U, Soman K.P and Rajendran S. Article: "A Novel Data Driven Algorithm for Tamil Morphological Generator", International Journal of Computer Applications 6(12):52–56, September 2010.