A Sequence Labeling Approach to Morphological Analyzer for Tamil Language

Anand Kumar M, Dhanalakshmi V, Soman K.P
Computational Engineering and Networking (CEN),
AMRITA Vishwa Vidyapeetham,
Coimbatore, India.

Rajendran S
Head, Department of Linguistics,
Tamil University,
Thanjavur, India,

***Abstract*—Morphological analysis is the basic process for any Natural Language Processing task. Morphology is the study of internal structure of the word. Morphological analysis retrieves the grammatical features and properties of a morphologically inflected word. Capturing the agglutinative structure of Tamil words by an automatic system is a challenging job. Generally rule based approaches are used for building morphological analyzer. In this paper we propose a novel approach to solve the morphological analyzer problem using machine learning methodology. Here morphological analyzer problem is redefined as classification problem. This approach is based on sequence labeling and training by kernel methods that captures the non linear relationships of the morphological features from training data samples in a better and simpler way.**

Keywords- morphology; morphological analyzer; machine learning; sequence labeling.

I. 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 [5]. 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. Such morphologically rich language needs deep analysis at the word level to capture the meaning of the word from its morphemes and its categories. Each root is affixed with several morphemes to generate word. In general Tamil language is postpositionally inflected to the root word. Each root word can take a few thousand inflected word forms. Tamil language takes both lexical and inflectional morphology. Lexical morphology changes the word meaning and its class by adding the derivational and compounding morphemes to the root. Inflectional morphology changes the form of the word and adds additional information to the word by adding the inflectional morphemes to the root.

In general, dictionary contains only root words and derivational words. All the inflectional word forms are not available in dictionary. Morphological analyzer is also used to extract the root word which exists in dictionary. Generally rule based approaches are used for building morphological analyzer [13]. In this paper a novel method for the morphological analyzer of Tamil language using sequence labeling approach is implemented. The implementation process is done in three phases. In first phase the input word is converted into sequence of characters that is used for decoding. Second phase segment morphemes based on their boundaries. Finally segmented morphemes are tagged with their grammatical category. The morphological complexity among the Dravidian languages doesn't vary widely. So this methodology is implemented to all Dravidian languages. The morphological analysis engine has been developed for Tamil and the process is implemented for other Dravidian languages. This methodology can be also applied for any morphologically rich languages. The new state of art machine learning approach based on SVM outperforms MBT and CRF Taggers. This paper briefly describes about data creation for supervised learning technique and various stages in building the morphological analyzer using sequence labeling.

II. RELATED WORKS

Various methodologies have been adopted for developing morphological analyzer in various languages. A framework for Thai morphological analysis based on the theoretical background of conditional random fields formulates an un-segmented language as the sequential supervised learning problem [4]. Memory-based learning has been successfully applied to morphological analysis and part-of-speech tagging in Western and Eastern-European languages [6]. MBMA (Memory-Based Morphological Analysis), is a memory-based learning system .Memory-based learning is a class of inductive supervised machine learning algorithms that learn by storing examples of a task in memory. A corpus based morphological analyzer for unvocalized Modern Hebrew is developed by combining statistical methods with rule-based syntactic analysis [1].

Goldsmith shows how stems and affixes can be inferred from a large un-annotated corpus [11]. Data-driven method for automatically analyzing the morphology of ancient Greek used a nearest neighbor machine learning framework [12]. A language modeling technique to select the optimal segmentation rather than using heuristics is proposed for Thai morphological analyzer [3]. In Tamil language the first step towards the preparation of morphological analyzer for Tamil was initiated by Anusaraka group. Ganesan developed a morphological analyzer for Tamil to analyze CIIL corpus [14] .In this phonological and morphophonemic rules and takes into account morphotactic constraints of Tamil in building

morphological analyzer for Tamil. Resource Centre for Indian Language Technological Solutions-Tamil has prepared a morphological analyzer (Atcharam) for Tamil. Finite automata state-table has been adopted for developing this Tamil morphological analyzer [2].

III. CHALLENGES IN MORPHOLOGICAL ANALYZER FOR TAMIL

Tamil is a classical language which belongs to Dravidian language family. It is spoken by more than 66 million people all over the world [ref].Tamil literature has existed for over two-thousand years. 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. A single verb root can inflect for more than two-thousand word forms including auxiliaries. Noun root inflects with plural, oblique, case, postpositions and clitics. A single noun root can inflect for more than five hundred word forms including postpositions. The root and morphemes have to be identified and tagged for further language processing at word level. 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.

The simple finite verb forms table is given in "Table I". First column represents the PNG (Person-Number-Gender) markers and the further column represents past, present and future tenses respectively. For the word "படி" padi(study) various PNG markers are given in table.

TABLE I. SIMPLE FINITE VERBS

PNG	Root-Past-PNG	Root-Pres-PNG	Root-Fut-PNG
3SE	*padi-kinR-Ar*	*padi-thth-Ar*	*padi-pp-Ar*
3SM	*padi-kinR-An*	*padi-thth-An*	*padi-pp-An*
3SF	*padi-kinR-AL*	*padi-thth-AL*	*padi-pp-AL*
2S	*padi-kinR-Ay*	*padi-thth-Ay*	*padi-pp-Ay*
1PL	*padi-kinR-Om*	*padi-thth-Om*	*padi-pp-Om*
IS	*padi-kinR-En*	*padi-thth-En*	*padi-pp-En*
2SE	*padi-kinR-Ir*	*padi-thth-Ir*	*padi-pp-Ir*
3SN	*padi-kinR-athu*	*padi-thth-athu*	*padi-pp-athu*
2PE	*padi-kinR-IrkaL*	*padi-thth-IrkaL*	*padi-pp-IrkaL*
3PE	*padi-kinR-ArkaL*	*padi-thth-ArkaL*	*padi-pp-ArkaL*
3PN	*padi-kinR-ana*	*padi-thth-ana*	*padi-pp-ana*

Understanding of verbal complexity involves understanding the structure of simple finite verbs and compound verbs. By understanding the nature of the verbal complexity, it is possible to evolve a methodology to tackle the verbal complexity. In order to tackle the analysis of the verbal forms in which the inflection vary from one set of verbs to another, a classification of Tamil verbs based on tense markers is evolved. The inflection includes finite, infinite, adjectival, adverbial and conditional forms of verbs [13]. For the sake of our computational data modeling, Tamil verbs are classified into thirty two paradigms.

Compared to verb morphological analysis noun morphological analysis is less challenging. Noun can occur separately or with plural, oblique, case, postpositions and clitics suffixes. Nouns are classified into twenty five paradigms to resolve the challenges in noun morphological analysis. Based on the paradigm the root words are grouped into its paradigm. A corpus is developed with all morphological feature information. So the machine by itself captures all morphological rules, including *sandi* and morphotactic rule. Finally the morphological analysis is redefined as a classification task which is solved by using sequence labeling methodology. The various noun forms are given in the "Table II". The table represents the singular and plural form of the word "எலி" *eli(rat)* with the case markers.

TABLE II. NOUN CASE MARKERS

Case	Singular	Plural
Nominative	*eli*	*eli-kaL*
Accusative	*eli-ai*	*eli-kaL-ai*
Dative	*eli-uku*	*eli-kaL-uku*
Benfefactive	*eli-ukk-Aka*	*eli-kaL-ukk-Aka*
Instrumental	*eli-Al*	*eli-kaL-Al*
Sociative-Odu	*eli-Odu*	*eli-kaL-Odu*
Sociative-udan	*eli-udan*	*eli-kaL-udan*
Locative	*eli-il*	*eli-kaL-il*
Ablative	*eli-il-iruwthu*	*eli-kaL-il-iruwthu*
Genitive	*eli-in-athu*	*eli-kaL-in-athu*

IV. MORPHOLOGICAL ANALYZER USING MACHINE LEARNING

The morphological analysis identifies root and suffixes of a word. Generally rule based approaches are used for morphological analysis which are based on a set of rules and dictionary that contains root words and morphemes. In rule based approach, a particular word is given as an input to the morphological analyzer and if that corresponding morphemes or root word is missing in the dictionary then the rule based system fails [5]. Here each rule is depended on the previous rule. So if one rule fails, it affects the entire rule that follows.

Recently machine learning approaches are dominating the Natural Language Processing field. Machine learning is a branch of Artificial Intelligence concerned with the design of algorithms that learn from the examples. Machine learning algorithms can be supervised or unsupervised. Available input and required output examples are used in supervised learning. In unsupervised learning they use only input samples. The goal of machine learning approach is to use the examples and find the useful generalization and classification rules automatically from examples. Using this machine learning approach 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 [5]. It needs only corpora with linguistical information. These morphological or linguistical rules are automatically extracted from the annotated corpora.

A. Morphological Analyzer as Sequence Labeling

The sequence labeling is a significant generalization of the supervised classification problem. We assign a single label to each input element in a sequence. The elements that are trying to assign are typically things like parts of speech, syntactic chunk labels [9]. Many tasks are formalized as sequence labeling problems in various fields such as natural language processing and bioinformatics. There are two types in sequence labeling approaches.

- Raw labeling.
- Joint segmentation and labeling.

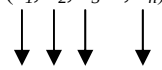
In raw labeling each element gets a single tag whereas in joint segmentation and labeling whole segments get a single label. In morphological analyzer sequence is usually a word and a character is an element or.

As mentioned earlier, in morphological analyzer input is a word and output is root and inflections. Input word is denoted as 'W', root word and inflections are denoted by 'R' and 'I' respectively.

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

In turn notation 'I' can be expressed as $i_1 + i_2 + \ldots + i_n$. Where 'n' are a number of inflections or morphemes. Further 'W' is converted into set of characters. Morphological analyzer accepts a sequence of character as input and generates a sequence of character as output. Let X be the finite set of input characters and Y be the finite set of output characters. Here the input string be 'x', it is segmented as $x_1x_2....x_n$ where each $x_n \epsilon$ X. Similarly y be an output string and it is segmented as $y_1y_2...y_n$ and $y_n \epsilon$ Y. where 'n' be the number of segments.

Inputs: x = (x_1, x_2, x_3…, x_n)



Labels: y = (y_1, y_2, y_3…, y_n)

The main objective of sequence labeling approach is predicting y from the given 'x'. In training data the input sequence 'x' is mapped with output sequence 'y'. Now the morphological analyzer problem is transformed into a sequence labeling problem. The information about the training data is explained in following sub sections.

B. Machine learning using Support Vector Machine

Support vector machine 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 [8]. These classifications can be done through supervised machine learning algorithms. In supervised learning set of input and output examples are used for training.

Support Vector Machine (SVM) is a machine learning algorithm for binary classification, which has been successfully applied to a number of practical problems, including NLP [10]. Let $\{(x_1, y_1), ..., (x_N, y_N)\}$ be the set of N training examples, where each instance x_i is a vector in R^N and $y_i \in \{-1, +1\}$ is the class label. SVM is a supervised pattern classification algorithm which has been successfully applied to a wide range of classification problems. SVM is attractive because it has an extremely well developed statistical learning theory. 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.

C. SVMTool

The 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 analysis for classification. 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 [10].

Different models are learned for the different strategies. Given a training set of annotated examples, it is responsible for the training of a set of SVM classifiers. So as to do that, it makes use of SVM–light an implementation of Vapnik's SVMs in C, developed by Thorsten Joachims. Given a text corpus (one token per line) and the path to a previously learned SVM model (including the automatically generated dictionary), it performs tagging of a sequence of characters.

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. Three standard machine learning approaches, SVM, CRF and memory-based learning, have been used to solve the classification problem. SVM is based on the idea of structural risk minimization. A principled technique is used for selecting a model which minimizes generalization error. Conditional Random Fields is another popular approach for sequence labeling which offer advantages over both generative models like HMM and classifiers applied at each sequence position. MBT is a memory-based tagger-generator and tagger. The tagger-generator part can generate a sequence tagger on the basis of a training set of tagged sequences. The tagger part can tag new sequences.

V. CREATING DATA FOR SUPERVISED LEARNING

Morphological analyzer separates the root and affixes from the given word. 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 will inflect with the same set of inflections. The second step is to collect the list of root words for all paradigms.

A. Paradigm Classification

Paradigm provides information about all possible word forms of a root word in a particular word class. Tamil noun and verb paradigm classification is done based on its case and tense markers respectively. Number of paradigms for each word class (noun/verb) is defined. In Tamil there are 32 paradigms for verb and 25 for noun [12]. "Table III", shows the number of paradigms and inflections of verb and noun which we handled. *WO-AUX* means count of the verb forms without auxiliaries and clitics and *WO-PP* means, count of the noun forms without postposition inflections. *Total* represents the total number of inflections that we have handled in this analyzer system. Verb and noun paradigm list is shown in "Fig. 1" and "Fig. 2".

TABLE III. NUMBER OF PARADIGMS AND INFLECTIONS

	No. of Paradigms	No. of Inflections		
		WO-AUX	*WO-PP*	*Total*
Verb	**32**	**95**	**--**	**1884**
Noun	**25**	**--**	**30**	**325**

படி-padi	ஏற்று-ERRu	சாகு-sAku
செய்-cey	புகழ்-pukaz	விடு-vidu
காண்-kAN	ஆள்-AL	பெறு-peRu
சொல்-col	உண்-uN	ஆகு-Aku
கல்-kal	பூண்-pUN	அகல்-akal
கேள்-kEL	உவ-uva	ஏறு-Eru
நில்-wil	அழு-azu	புகு-puku
ஓடு-Odu	தின்-thin	ஈன்-En
அறி-aRi	விழு-vizu	ந ட-wada
வா-vA	கொல்-kol	என்-en
போ-pO	நோகு-woku	

Figure 1. Verb Paradigms

புல்-pul	கல்-kal	மனிதன்-manithan
பொய்-poy	கால்-kAl	யானை-yAnai
ஈ -E	முள்-muL	தோள்-thOL
பூ-pO	ஆண்-AN	மரம்-maram
மான்-mAn	கண்-kaN	பொருள்-poruL
தேர்-thEr	நாய்-wAy	காடு-kAdu
பொன்-pOn	ஆறு-Aru	நரம்பு-warampu
பஸ்-paS	எலி-eli	வண்டு-vaNdu
கடா-kadA		

Figure 2. Noun Paradigms

B. Preprosessing

Preprocessing is an important step in data creation. It is involved in training stage as well as decoding stage. "Fig. 3" explains the preprocessing steps involved in the development of corpora. Morphological corpus which is used for machine learning is developed by following steps.

1) Romanization

These data are converted to Romanized forms using the Unicode to Roman mapping file. Romanization is done for easy computational processing. In Tamil, syllable exists as a single character, where we cannot separate vowel and consonant.

2) 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. To the segmented syllable postfix "*–C*" and "*–V*" to the consonant and vowel respectively. It is named as *C-V* representation i.e. Consonant – Vowel representation. In the output data morpheme boundaries are indicated by "*" symbol.

3) 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 "Table IV". .First column represents the input data and the second one represents output data."*" indicates the morpheme boundaries.

TABLE IV. SAMPLE TRAINING DATA FORMAT

I/P	O/P
p-C	p
a-V	a
d-C	d
i-V	i*
th	th
th-C	th*
A-V	A
n	n*

4) Mapping Mismatch segments

It is the key problem which occurred in mapping the input characters with output characters. Mismatching occurs in two cases i.e., either the input units are larger or smaller than that of the output units. The mismatching 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. Tamil verb *padikkayiyalum* is having 14 segments in input sequence but in output only 13 segments are present. The second occurrence of "*y*" in the input sequence becomes null due to the morpho syntactic 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. Using morpho syntactic rule 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.

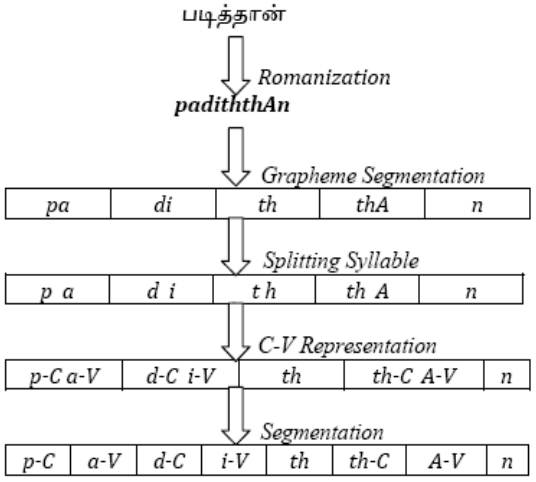


Figure 3. Preprocessing Steps

VI. IMPLEMENATATION OF MORPHOLOGICAL ANALYZER

Using the machine learning approach the morphological analyzer for Tamil is developed. We have developed separate engines for noun and verb. Noun morphological analyzer can handle nouns and proper nouns. The verb analyzer handles all the verb forms like finite, infinite and auxiliary forms. Morphological analyzer is redefined as a classification task. Classification problem is solved by using the Support Vector Machine. In this machine learning approach two training models are created for morphological analyzer. These two models are represented as *model-I* (segmentation model) and *model-II* (morpho-syntactic tagging model) [8].

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.

"Fig. 4" explains the three phases involved in the process of morphological analyzer.

- Pre-processing.
- Morpheme Segmentation.
- Morpho syntactic tagging.

A. Preprocessing

The word that has to be morphologically analyzed is given as the input to the pre-processing phase. The word primarily undergoes Romanization process. The romanized word is segmented based on Tamil graphemes. Tamil grapheme consists of vowel, consonant and syllable, are further processed for syllable identification. The identified syllable is broke up into vowel and consonant. To these consonant and vowel, *–C* and *–V* are suffixed.

B. *Segmentation of Morpheme*

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

C. *Identifying Morpheme*

The Segmented morphemes are given to the trained model-II. It predicts grammatical categories to the segmented morphemes. The system has been trained to give multiple outputs, to handle the compound words.

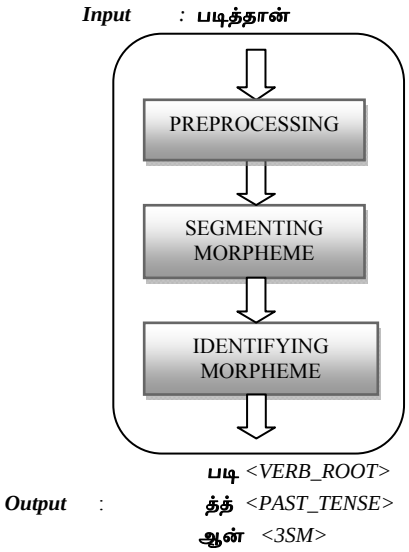


Figure 4. Implementaion Stages

VII. SYSTEM ESTIMATION

Efficiency of the system is compared in this section. Various machine learning tools are also compared using the same morphologically annotated data, of which SVM based tool holds good. The system accuracy is estimated at various levels, which are briefly discussed below.

A. *Training Data Vs Accuracy*

In "Fig. 5", X axis represents training data and Y axis represents accuracy. From the graph, it is found that Morphological Analyzer accuracy is increases with increase in proportion of training data size. Calculate the accuracy for training data from 30k to 130k. The accuracy increases rapidly.

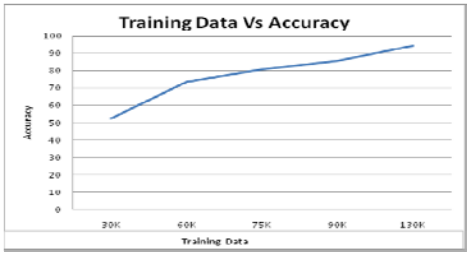


Figure 5. Training data Vs Accuracy

B. *Tagged and Untagged Accuracies*

In the sequence based morphological system, output is obtained in two different stages using the trained models. First stage takes a sequence of character as input and gives untagged morphemes as output using the trained *model-I*. It is also represent as morpheme identification. In second stage, these morphemes are tagged using trained *model-II*. Accuracies of the untagged and tagged morphemes for verbs and noun are shown in the "Table V".

TABLE V. TAGGED VS UNTAGGED ACCURACIES

Accuracy	Verb	Noun
Untagged(Model-I)	93.56%	94.34%
Tagged(Model-II)	91.73%	92.2 %

C. *Recall-Precision and F score*

Recall, precision and F-score are obtained with testing data set. The system was tested with two different test data sets which are data already available in training set and data not available in training set. These accuracies are represented recall and precision respectively. F-score was also calculated using recall and precision values. These accuracies are given in the "Table VI". The formula for calculating F-score is given bellow.

$$F\text{-}Score = 2PR/(P+R)$$

TABLE VI. RECALL-PRECISION F-SCORE

Category	Recall (R)	Precision (P)	F-Score
Verb	96.35%	90.43%	93.3%
Noun	97.26%	91.01%	94.03%

D. *Word level and Character level Accuracies*

Accuracies are compared with word level as well as character level. Two thousand three hundred verb data and one thousand seven hundred and fifty noun data are taken randomly from POS Tagged corpus for testing the system [7]."Table VII" shows the number of words as well as the characters in the whole testing data set.

TABLE VII. NUMBER OF WORDS AND CHARACTERS

Category	VERB		NOUN	
	Words	*Characters*	*Words*	*Characters*
Testing data	2300	20627	1750	10534
Predicted correctly	2071	19089	1639	9645

"Table VIII" shows the accuracies at word and character level. These accuracies are calculated using the trained *model-I* for the tested data size given in "Table VII"..

Word level accuracy =Number of words spitted correctly/Total number of words in Testing set
Character level accuracy =Number of characters tagged correctly/Total number of characters in Testing set

TABLE VIII. WORD AND CHARACTER LEVEL ACCURACIES

Accuracy	Verb	Noun
Word Level	90.0%	91.5%
Character level	92.5%	93.6%

E. Compare SVM with MBT and CRF

1) Memory Based Tagger (MBT)

MBT is implemented using the memory-based learning software package TiMBL. Memory-Based Tagging is an approach to POS Tagging based on Memory-Based Learning (MBL). As an adaptation and extension of the classical *k*-Nearest Neighbor (*k*-NN) approach to statistical pattern classification, MBL has proven to be successful in a large number of tasks in natural language processing [6]. Here MBT is used for morphological analyzer implementation.

2) CRF++

It is simple, customizable, and open source implementation of Conditional Random Fields (CRFs) for segmenting/labeling sequential data [15]. CRF++ is designed for generic purpose and will be applied to a variety of NLP tasks, such as Named Entity Recognition, Information Extraction and Text Chunking. Here morphological analyzer data is adapted and given to CRF++ for classification.

The number of possible input segments is 72 and possible output labels are 70.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 verbs and 30,000 nouns from an Amrita POS Tagged corpus [7]. The SVM based machine learning tool produced 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 Morphological Analyzer results are evaluated and compared in "Table IX". The output which was incorrect is noticed and its corresponding input and output labels are added in the training file and trained again. This increases the efficiency of the system. This is the main advantage of using machine learning approach to rule based approach.

TABLE IX. ACCURACIES OF DIFFERENT TOOLS

MODEL	CRF++	MBT	SVMTOOL
Accuracy (F-Score)	89.72%	90.38%	93.65%
Training time	High	Low	Medium
Testing time	Medium	Medium	High

VIII. GUI FOR MORPHOLOGICAL ANALYZER

Graphical user interface have been created for Morphological Analyzer using Net beans has compatibility with both Linux and Windows. Screenshot of our GUI is shown in "Fig.6". The GUI that we have created for Tamil morphological analyzer is simple and portable. As the user types the Tamil word (in Unicode format) into the input box and select weather the given word is noun or verb, system gives morphologically analyzed output in the output box. This can be done through already trained models. Perl language is used for preprocessing the input data. Complex and ambiguous words are handled by using multiple outputs.

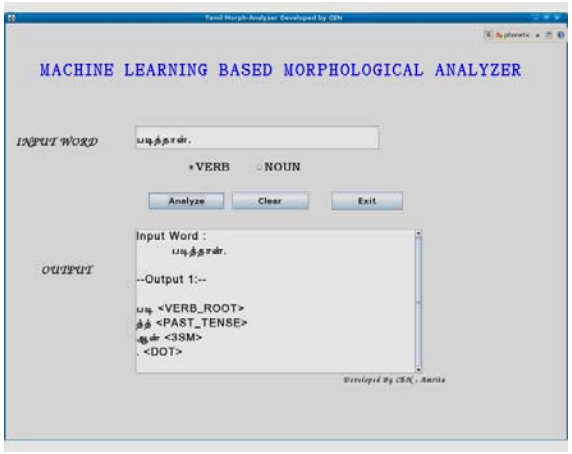


Figure 6. Graphical User Interface

IX. CONCLUSION AND FUTURE WORK

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 also implemented the same methodology for 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 language.

ACKNOWLEDGMENT

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.

REFERENCES

[1] Alon Itai, Erel Segal, 2003.A Corpus Based Morphological Analyzer for Unvocalized Modern Hebrew.Department of Computer Science Technion—Israel Institute of Technology, Haifa, Israel.

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

[3] Asanee Kawtrakul and Chalatip Thumkanon. 1997. A statistical approach to thai morphological analyzer. In Proc. of the 5th Workshop

on Very Large Corpora, M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science,

[4] Canasai Kruengkrai, Virach Sornlertlamvanich, Hitoshi Isahara , 2006 A Conditional Random Field Framework for Thai Morphological Analysis. In Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-06), May 24-26, 2006. Genoa, Italy.)

[5] 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

[6] Daelemans, W., J. Zavrel, A. Van den Bosch, and K. Van der Sloot. 2003. MBT: Memory based tagger, version 2.0, reference guide. Technical Report ILK 03-13, ILK Research Group, Tilburg University

[7] 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.

[8] 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

[9] Hal Daume , 2006. http://nlpers.blogsp-ot.com/2006/11/-getting-started-in-sequence-labeling.html.

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

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

[12] John Lee, 2008 “A Nearest-Neighbour Approach to the Automatic Analysis of Ancient Greek Morphology” CoNLL 2008: Proceedings of the 12th Conference on Computational Natural Language Learning, Manchester, August 2008: 127–134

[13] 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.

[14] Rajendran, S. ,2006 Parsing in Tamil –Present State of Art. Language in India, www.langu-ageinindia.com Vol 6 : 8 August, 2006.

[15] Taku kudo ,2005. CRF++:Yet Another CRFToolkit. , http://chasen.org/~taku/software

AUTHORS PROFILE

M.Anand kumar was born in Tamilnadu,India in 1984. He has completed his B.Tech in the department of Information Technology in 2006 and his M.Tech from AMRITA Vishwa Vidyapeetham university in the department of Computational Engineering and Networking (CEN) during 2006-2008. Now he is pursuing his P.h.D in Computational Lingustics under the Guidence of Professor K.P.Soman from Amrita Vishwa Vidyapeetham. Currently working as a Research Associate in Center for Computational Engineering and Networking (CEN) AMRITA Vishwa Vidyapeetham. His research interests includes Machine Learning approaches for NLP, Morphological Analyzer, Morphological Generator and Machine Translation.

V.Dhanalakshmi was born in Tamilnadu,India in 1975. She has completed MA M.Phill (Tamil) and MA (Mass communication). Now she is pursuing her P.h.D in Computational Lingustics under the guidence of Dr.S.Rajendran from Tanjore University . Currently working as a Research Associate in Center for Computational Engineering and Networking (CEN) AMRITA Vishwa Vidyapeetham. She has published several papers in the field of Computational Lingustics. Her research interests includes Machine Learning approaches for NLP, Computer assited learning, ,Grammar teaching tools, Morphological Analyzer, Shallow Parsing, Morphological Generator and Machine Translation.