A Novel Data Driven Algorithm for Tamil Morphological Generator

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

Rajendran S
Dept. of Linguistics
Tamil University
Thanjavur, India

ABSTRACT
Tamil is a morphologically rich language with agglutinative nature. Being agglutinative language most of the word features are postpositionally affixed to the root word. The morphological generator takes lemma, POS category and morpho-lexical description as input and gives a word-form as output. It is a reverse process of morphological analyzer. In any natural language generation system, morphological generator is an essential component in post processing stage. Morphological generator system implemented here is based on a new algorithm, which is simple, efficient and does not require any rules and morpheme dictionary. A paradigm classification is done for noun and verb based on Dr.S.Rajendran's paradigm classification. Tamil verbs are classified into 32 paradigms with 1884 inflected forms. Like verbs, nouns are classified into 25 paradigms with 325 word forms. This approach requires only minimum amount of data. So this approach can be easily implemented to less resourced and morphologically rich languages.

General Terms
Data driven approach, Natural Language Processing, Morphological Generator, Machine Translation

Keywords
Paradigm, Suffix table, word-forms, Morpho-lexical information, Tamil morphological generator

1. INTRODUCTION
The aim of Natural Language Processing (NLP) is studying the problems in the automatic generation and understanding of natural languages. Computational models of natural language have to be build for its various analysis and generation. Tamil is morphologically rich and agglutinative language [10]. Tamil words are postpositionally inflected with various grammatical features. Tamil verb specifies almost everything like gender, number, and person markings and also with auxiliaries which represents mood and aspect [9]. Tamil noun inflects for plural, case suffixes and post positions. Morphological generator generates a word-form from a lemma, a word class tag, and morpho-lexical description. In Tamil language the lemma undergoes morphological change when it get attach to certain morphemes. This system handles morpho-phonemic change without any hand coded rules.

Morphological generator can be an individual module or integrated with several NLP applications like machine translation, automatic sentence generation, etc. Any automated machine translation system requires, morphological analyzer of source language and morphological generator of the target language. In this paper we describe a fast and simple morphological generator for Tamil using an efficient algorithm. This novel approach can be applied to any morphologically rich language. As the user gives the information of the lemma, POS category and morpho-lexical inflection, the developed algorithm generates the intended word form. Three different modules are developed to build this system. The first module takes the lemma and POS category as input and gives the lemma's paradigm number and word's stem as output. The second module takes morpho-lexical information as the input and gives its index number as the output. In third module a suffix-table is used to generate the word with the information from the above two modules. The result obtained is encouraging.

2. RELATED WORKS
The most competent approach to morphological generator is using Finite State Transducers [3]. Letter transducers based morphological analyzer and generator was developed by Alicia Garrido. Perez Aguiar has used an intuitive pattern-matching approach for developing morphological generator to Spanish language. Guido Minnen and his team have developed a morphological generator based on Finite state techniques and it is implemented using the widely available Unix Flex utility [5].

For Indian languages many attempts have been made to build morphological generator. A Hindi morphological generator has been developed based on database driven approach [4]. *Tel-More* Morphological generator for Telugu is based on linguistic rules and Perl program [8]. Morphological generator has been designed for syntactic categories of Tamil using Paradigm based approach and sandhi rules [1]. Finite state machines are used for developing morphological generator for Tamil [2].

3. MORPHOLOGICAL GENERATOR FOR TAMIL
Generally, morphological generator tool is developed using rule based approach where it requires a set of morpho-phonemic (spelling) rules and morpheme dictionary. In this novel approach rules and dictionaries are not necessary. This algorithm only requires the Suffix table and the code for paradigm classification. Here, the morphological generator receives an input in the form of *lemma+word_class+ Morpho-lexical Information*, where lemma specifies the lemma of the word-form to be generated, *word_class* specifies the grammatical category (POS category) and Morpho-lexical Information specifies the type of inflection.

The Morpho-lexical Information has been extracted from our morphological analyzer tool for Tamil [7]. Example of the Tamil morphological generator system is given bellow.

Example for Tamil Morphological Generation

ஓடு + V + FT_3SM = ஓடுவான்
Odu + V + FT_3SM = OduvAn
(Run)

காடு + N + ACC = காட்டை
kAdu + N + ACC = kAddai
(Forest)

மரம் +LOC =மரத்தில்
maram + N + LOC = maraththil
(Tree)

In the above example "V" represents verb and "FT_3SM" represents future tense with third person singular masculine."N" be a symbol of noun and ACC means accusative case and LOC represents locative case marker.

3.1 Challenges in Tamil Morphological Generator

Tamil is morphologically rich and agglutinative language. Each verb can be inflecting with more than two thousand form including auxiliaries and clitics. The inflection also includes finite, infinite, adjectival, adverbial and conditional forms of verbs. In the generation of these verbal forms the inflections vary from one set of verbs to another. To solve this complexity, a classification of Tamil verbs based on tense markers and inflections is made. The verbs have been classified into thirty-two paradigms, based on their tense markers and morphophonemic change. Nouns are classified into twenty-five paradigms [9]. Verb paradigms are given bellow (see Table 1).

Table 1. Verb Paradigms

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	நட -wada
வா-vA	கொல்-kol	என்-en
போ-pO	நோகு-woku	

Normally paradigm based approach is used for developing morphological generator. In paradigm based approach, the paradigm number of the input root word is identified using the dictionary. The dictionary contains lemma with word class and its paradigm number. If user's input lemma is not present in the dictionary the system will fail to identify its paradigm number. At the same time, it is not possible to build a dictionary with all the lemmas. We cannot include all the proper nouns and compound word forms. Noun paradigms are given bellow (see Table 2).

Table 2. Noun 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		

This challenging task can be solved if the system can automatically identify the paradigm number of the lemma. When the user gives the lemma as an input our system automatically identifies its paradigm number based on the lemma's end characters. Another challenging task is to handle the morpho-phonemic change. Our system handles this very simply, by joining the stem of the generating word with the remaining inflections in the suffix table. So there is no need for any separate morpho-phonemic rule. Creation of this suffix table plays an important role in this challenging job. The creation of the suffix table is explained in the next sub section.

3.2 Creation of Suffix Table

The Suffix table is the most essential file in this algorithm. This is a simple two-dimensional (2D) table where row corresponds to the morpho-lexical form and column corresponds to the paradigm number. Each syntactic category has its own suffix table. Here we have only created for noun and verb. The noun suffix table contains 325 rows (word-forms) and 25 columns (paradigms) similarly verb suffix table contains 628 rows and 32 columns (paradigms).

Number of paradigms for each word class (noun/verb) is defined. In Tamil there are 32 paradigms for verb and 25 for noun [9]. Table -3 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 generator system.

Table 3. Paradigms and Inflections

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

For every paradigm a word is selected and this is termed as head word. For this head word, all morpho-lexical forms are created for noun and verb individually. In Tamil there are more than thousand word-forms are possible for each verb. Here we have selected 628 most frequently used Morpho-lexical forms for verb including 25 auxiliary verbs (clitics are handed separately) and for noun it is 325 including postpositions. The similar verb/noun morpho-lexical information pattern should be followed for all the paradigms. A morpho-lexical Information list is also created for the above morpho-lexical forms. Using all the word-forms a table is created, each column of the table corresponds to its paradigm. In that table, stem of the each paradigm is removed from its word-form. Now this table is represented as a Suffix table. Table.4 illustrates the sample suffix-table for Tamil verbs. In this table row (MLI-1, MLI-2…) specifies the morpho-lexical inflection and column (P-1, P-2…) indicates paradigm number.

Table 4. Suffix Table

	P-1	P-2	P-3	P-4	P-5	
MLI-1	ththAn	wAn	inAn	thAn	Ran	.....
MLI-2	ththAL	wAL	inAL	thAL	RAL	.....
MLI-3	ththAr	wAr	inAr	thAr	RAr	.....
MLI-4	kinRAn	kinRAn	kinRAn	kinRAn	kinRAn	.....
MLI-5						.....

3.3 Algorithm Developed for Morphological Generator

In this section we are going to describe about the new algorithm which is developed for morphological generator. The main advantage for this algorithm is simple and accurate. This algorithm is implemented using Perl program. The simple algorithm and its explanation is given bellow,

Input = (*Lemma +word class + morpho-lexical Information*)

1. *lemma,wc,morph* =SPLIT(*Input*)
2. *roman_lemma*=ROMAN(*lemma*)
3. *parnum*=PARNUM(*roman_lemma,wc*)
4. ***col-index***=*parnum*
5. ***row-index***=INDEX(*morph,wc*)
6. *suff*=SUFFIX-TABLE[*row-index*][*col-index*]
7. *stem*=STEM(*roman_lemma,wc,parnum*)
8. *word*=JOIN(*stem,suff*)
9. *output*=UNICODE*(word)*

Where, in the first step, *lemma* represents the lemma, *wc* represents the word class and *morph* represents the morpho-lexical information. The input from the user is divided into lemma, word class and Morpho-lexical information this is done by using the SPLIT function. The lemma or the root word in Unicode format is romanized using the function ROMAN.

roman_lemma represents the romanized lemma. parnum represents paradigm number of lemma.PARNUM identifies the paradigm number this is done using the Perl program. Romanized lemma and paradigm number are given as input to STEM function along with the word class. This function, stems the lemma. The morpho-lexical information given by user is matched with the morpho-lexical information list, and the corresponding index number is retrieved, this index number is referred as row-index. Paradigm number of the input lemma is named as col-index. Using the row and column index the suffix part is retrieved from the Suffix-table. The stem and the retrieved suffix are attached to generate the word form. This word form is then converted to Unicode format which is the final output.

3.4 Implementation

The morphological generator system needs to handle three major things, first one is the lemma part, then the word class and finally the morpho-lexical information. By the way the generator is implemented makes it distinct from other morphological generator. The input which is in Unicode format is first Romanized and then the paradigm number is identified by end characters. For sake of easy computation we are using romanized form. A Perl program has been written for identifying paradigm number, which is referred as column index. The morpho-lexical information of the required word class is given by the user as input. From the morpho-lexicon information list the index number of the corresponding input is identified, this is referred as row index. A verb and noun suffix tables are used in this system. Using the word class specified by the user the system uses the corresponding suffix table. In this two-dimensional suffix table rows are morpho-lexical information index and columns are paradigm numbers.

For each paradigm we have created a complete set of morphological inflections corresponding to the morpho-lexical information list. Finally using the column index and row index morphological suffix is retrieved from the suffix table. This suffix form is affixed with the stem to generate the word form. In this work a morphological generator is designed for each of the syntactic categories and then combined to generate a complete sentence. Bellow steps explain the simple procedure of the system.

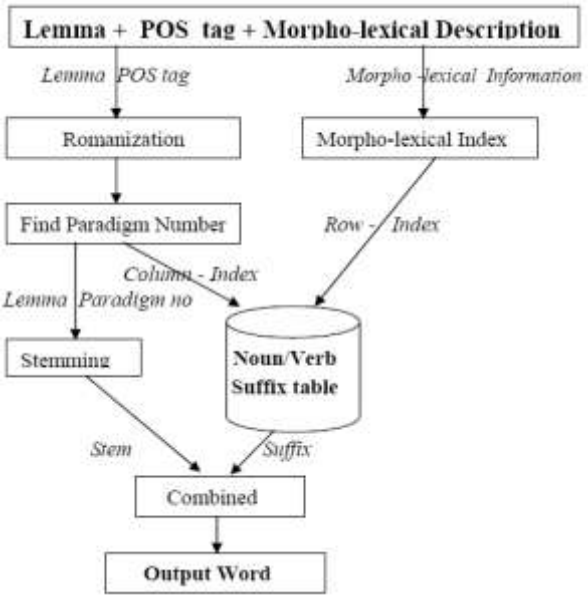
Step1: *Identify Paradigm of the Root word*

Step2: *Stem the root word based on Paradigm*

Step3: *Find Morpho-lexical Index*

Step4: *Retrieve Inflection from Suffix Table*

Step5: *Append Inflection with the Stem.*



"Figure 1. Morphological Generator System"

3.5 Advantages of this system

Morphological Generator is needed for various applications in Natural Language Processing. It acts as post-processing component in NLP applications like Machine translation. In this system we have also handled some difficult tasks like morpho-phonemic change and automatic paradigm identification. Some of the important advantages of this novel approach are mentioned below.

- Automatic paradigm identification.
- Uses Very less data.
- Simple, efficient and High Speed.
- Handles compound words and Proper nouns.
- Handles Transitive and Intransitive forms
- No morpho-phonemic Rules.
- No verb/noun dictionary for paradigm identification.
- No morpheme dictionary.
- Easily updatable.
- Applicable for any morphologically rich language.

4. GUI

The Graphical User Interface (GUI) (see in figure 2) of morphological generator tool has been developed using Net beans IDE and Perl programme. In this user friendly environment the user have to enter the lemma in Tamil and select the corresponding word class (noun or verb). Based on the word class our tool visualizes the possible morpho-lexical information in the bellow field. From this morpho-lexical information user have to select the inflection and click the generate button to generate the intended word. If the user wants to exercise the tool, they require little linguistic knowledge about the inflections of word-form. We cannot anticipate this from all the users so we have an option in our tool to generate all the possible word-forms for a user's input lemma.

"Figure 2. GUI for Morphological Generator"

5. CONCLUSION AND FUTURE WORK

The Morphological generator which is explained here is a novel approach. It is developed using a very simple and efficient method. This is not a language specific method, so this can be applicable for any morphologically rich language. Using this approach now we are developing morphological generator for Malayalam and Telugu languages. This system provides a vast application in NLP field mainly in Machine Translation.

It is used in noun declension, verb conjugation and automatic sentence generation. This system is unique that handles auxiliaries and clitics for verbs. It does not require any spelling rules and dictionary. This work can be further used for implementing morphology based translation system, from any language to Tamil. Using this morphological generator we have also developed a verb conjugator and noun declension. Currently we are developing SMT (Statistical Machine Translation) system for English to Tamil language where this Morphological generator is an important component in the post processing stage.

6. ACKNOWLEDGMENTS

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 also like to thank MHRD) for the successful completion of this work.

7. REFERENCES

[1] Anandan, P., Geetha, T.V., and Paratasarathy, R. 2001." Morphological Generator for Tamil ", In Proceedings of the Tamil Inayam Conference, Malaysia, 46-54.

[2] A. G. Menon, S. Saravanan, R. Loganathan, Dr. K. P. Soman," Amrita Morph Analyzer and Generator for Tamil:

A Rule-Based Approach ” Proceedings of Tamil Internet Conference 2009 , Cologne, Germany,October 2009.

[3] Garrido, Alicia, Amaia Iturraspe, Sandra Montserrat, Herm´ınia Pastor, and Mikel L. Forcada. 1999. “A compiler for morphological analysers and generators based on finite-state transducers ”. Procesamiento del Lenguaje Natural, 25:93–98.

[4] Goyal, V, Singh Lehal, G. “Hindi Morphological Analyzer and Generator ” Emerging Trends in Engineering and Technology, 2008. ICETET '08.

[5] Guido Minnen, John Carroll, and Darren Pearce. 2000. “Robust applied morphological generation.” Proceedings of the First International Natural Language Generation Conference, pages 201.208, 12.16 June.

[6] Irimia, E. ROG - A Paradigmatic Morphological Generator for Romanian.,2007, In Proceedings of the 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. Poznań, Poland.

[7] M Anand kumar, V Dhanalakshmi. , K P Soman, S Rajendran ,“A Novel Apporach For Tamil Morphological Analyzer” Proceedings of Tamil Internet Conference 2009 , Cologne, Germany, Page no: 23-35, October 2009.

[8] Madhavi Ganapathiraju and Lori Levin, 2006, - “TelMore: Morphological Generator for Telugu Nouns and Verbs ”. Proc. Second International Conference on Universal Digital Library, Vol Alexandria, Egypt, Nov 17-19, 2006

[9] S.Rajendran, Arulmozi, S., Ramesh Kumar, Viswanathan, S. 2001. “Computational morphology of verbal complex “. Language in india Volume 3 : 4 April 2003

[10] Thomas Lehmann, 1992 second edition. “A Grammar of Modern Tamil ”. Pondicherry Institute of Linguistics and Culture, Pondicherry.