An Implementation of APERTIUM Morphological Analyzer and Generator for Tamil

Parameshwari K

CALTS, University of Hyderabad,
Hyderabad-500046.

parameshkrishnaa@gmail.com,

Abstract— A Morphological Analyzer and Generator are two crucial tools involving any Natural Language Processing of Dravidian Languages. The present paper discusses the improvization of the existing Morphological Analyzer and Generator for Tamil by defining and describing the relevant linguistic database required for the purpose of developing them. The implementation of an open source platform called Apertium to handle inflection as well as derivation for word level analysis and generation of Tamil is also discussed. The paper also presents the efficacy, coverage and speed of the module against the large corpora. The paper also draws inferences of the morphological categories in their inflection and problems in analysing them.

I. Introduction

A language like Tamil is regarded as morphologically rich wherein the words are formed of one or more stems/roots plus one or more suffixes. So the complexity of morphology requires a more sophisticated morphological analyzer and generator. A morphological analyzer is a computational tool to analyze word forms into their roots along with their constituent functional elements. The morphological generator is the reverse process of an analyzer i.e. from a given root and functional elements, it generates the well-formed word forms.

The present attempt involves a practical adoption of lttoolbox for the Modern Standard Written Tamil in order to develop an improvised open source morphological analyzer and generator. The tool uses the computational algorithm called Finite State Transducers for one-pass analysis and generation, and the database is based on the morphological model called Word and Paradigm.

II. Implementation of Apertium (lttoolbox[1])

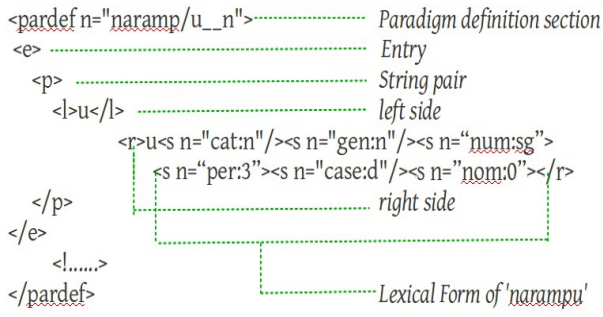
Apertium is an open source machine translation platform developed by the Transducens research group at the *Department de Llenguatges i Sistemes Inform`atics of the Universitat d'Alacant* in Spain. The lttoolbox is a toolbox for lexical processing such as morphological analysis and generation of words. The Document Type Definition (DTD) format is used in XML file for creating the lexical database in order to convert it into FST. The present attempt uses LINUX operating system with *fedora 10* platform for implementing the tool.

The analyzer as well as generator is obtained from a single morphological database, depending on the direction in which it is read by the system: read from left to right, we obtain the analyzer, and read from right to left, the generator.

The module requires the following database to build a Morphological Analyzer.

A. Paradigms and their definitions. A Paradigm here is referred to a complete set of related inflectional and productive derivational word forms of a given category. The database comprises of six distinct lexical categories viz., Noun, Verb, Adjective as open class and Pronoun, Number words and Locative Nouns as closed class. The Tamil Morphological Database available at the Centre for Applied Linguistics and Translation Studies (University of Hyderabad) Language Laboratory is extracted and improvised involving six distinct lexical inflectional categories for the purpose.

The Definition refers to the features and feature values of the root such as category, gender, number, person and case marking in the case of nouns and tense, aspect and modal category information in the case of verbs so on and so forth. The WX-notation[2] of transliteration is followed in this paper.



The XML format of inflection paradigm for Tamil 'narampu'

B. Linked Paradigms For Derivation. Derivational forms need the dynamic analysis rather than putting in the Dictionary. It is an alternative lexico-semantic modal which operates along

with inflection. There is a layer that introduces the lexemes into derivation and concurrently follows the inflection of the derived lexeme. For instance, *patikkirYavanY* 'one who(he) is reading' is a derived pronominal of the verb *pati* 'read'. It further takes all the inflections of pronoun 'avanY'. Here the derived pronoun is linked with the pronoun paradigm *avanY*.

```
<pardef n="pati__v">
 <e>
   <p>
     <l>kirYavanY</l>
       <r><s n="v"/><s n="m"/><s n="sg"/><s n="3"/><s n="0"/>
          <s n="kirY_a"/></r>
     </p><i>kirYavanY</i><par n="avanY__P"/>  ..... Linking paradigm for derivation
 </e>
    <!.....>
</pardef>
```

THE XML FORMAT OF PARADIGM TO HANDLE DERIVATION

C. LEXICON. A root word dictionary in Morphological Analyzer differs from a conventional dictionary. The dictionary for Morphological Analysis which is built for Word and Paradigm Model contains roots, categories and their corresponding paradigm. The present Morphological analyzer-generator lexicon contains the root/lemma, the part of the lemma which is common to all the inflected forms, that is, it contains the lemma cut at the point in which the paradigm regularity begins along with the appropriate paradigm and the paradigm name.

```
<e lm="maram">  ..... Element for Lemma
   <i>mara</i>  ..... The part of the Lemma
   <par n="mara/m__n"/>
 </e>           ..... Paradigm name
```

A DICTIONARY ENTRY OF THE LEXEME 'MARAM'

D. COMPILING AND PROCESSING. The data is compiled and processed by using the applications used in the lexical processing modules and tools (lttoolbox). The applications are responsible for compiling dictionaries into a compact and efficient representation (a class of finite-state transducers called augmented letter transducers) and processing the compiled data for the real time text.

The 'lt-comp' is the application responsible for compiling dictionaries used by Apertium into a compact and efficient representation.

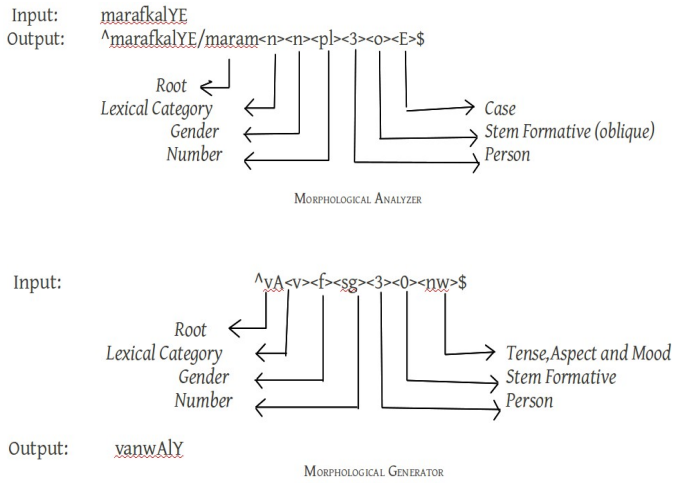
Synopsis **:** lt-comp [lr | rl] dictionary_file output_file

The dictionary which is compiled is processed by the application 'lt-proc' that is responsible for processing the data.

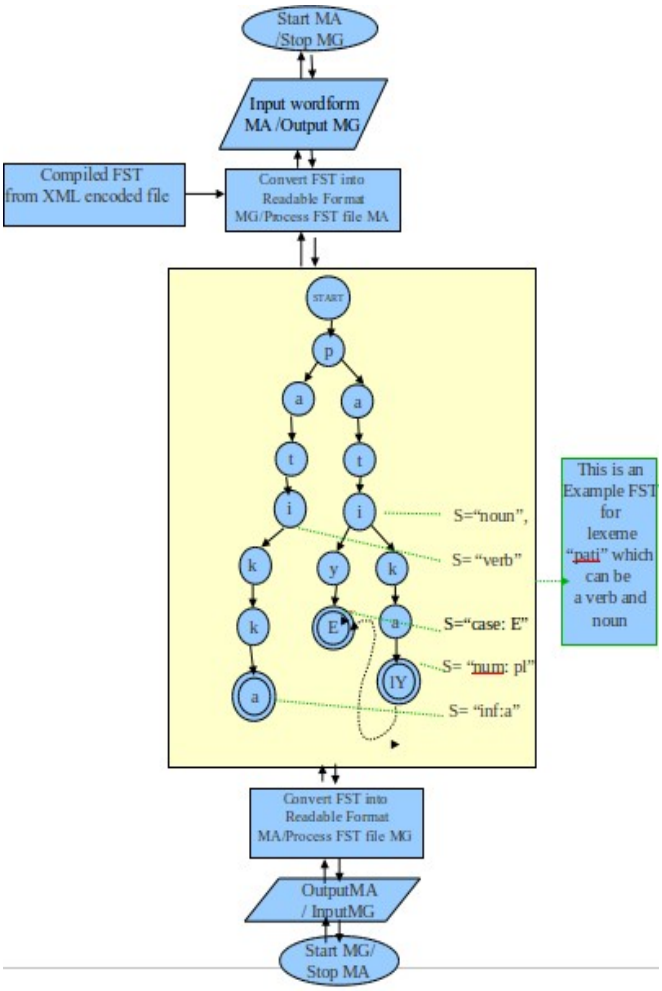
Synopsis **:** lt-proc [-c] [-a|-g] fst_file [input_file [output_file]]

The 'lt-proc' processes the stream with the letter transducers. Here 'fst_file' refers to the compilation file which is in FST format.

E. THE INPUT AND OUTPUT SPECIFICATION.



F. DATA FLOW IN MORPHOLOGICAL ANALYZER. The below figure is a flowchart that describes the data flow in the Morphological Analysis (MA) and Generation (MG).



G. Database. The following table shows the database of the Morphological module.

Paradigm			Dictionary Size (lemma) Number of Words	
Category	Number of Inflectional Classes	Number of Inflections per class	Category wise	Total
Noun	20	743	57,322	68,060
Verb	29	934	10,114	
Adjective	2	372	209	
Pronoun	11	654	18	
Numeral	14	370	129	
NST	7	67	62	
Avy	-	-	206	

Table 1 : Database

III. Testing and evaluation

The Morphological analyzer tool was tested with the corpus (CALTS corpus of 4.4 million words and EMILLI CIIL corpus of 4.8 million words) in order to find out it's coverage of the corpus. The coverage of the analyzer is calculated by dividing the analyzed word with the total number of words.

Corpus	Total words	Recognized words	Coverage	Speed
CALTS Corpus	4,45,130	3,75,891	84.44%	0m0.289s
EMILLI CIIL Corpus	4,85,543	4,05,898	83.59%	0m0.297s

The speed is an indication that CALTS-Apertium consumes less time to analyze a large number of data.

IV. Analysis

In the course of testing the tool, it has been found certain inconsistencies and lapses in recognizing certain words. The lapses are due to the lexical items with orthographic variation, inflectional variation, dialectal variation, naturalized loan words particularly from English into Tamil, proper nouns.

Type	Word From	Frequency in the Corpus
Orthographic Variation	*koyil* 'temple' *kovil* 'temple'	885 occurrences 204 occurrences
Inflectional Variation	*eVlYYuwwu-kkalY* 'letters' *eVlYYuwwu-kalY* 'letters'	57 occurrences 171 occurrences
Dialectal Variation	*vanwAy* 'You came'(standard) *vanweV* 'You came'(dialect)	765 occurrences 6 occurrences
Naturalized English loans	*polIS* 'police'	20070 occurrences
Proper nouns	*kaNNanY* 'male name' *wamilYYnAtu* 'Tamil Nadu'	211 occurrences 364 occurrences

The careful appraisal and study on the unrecognized words is conducted to identify and overcome the lapses by incorporating certain amount of data into the morphological database to enhance the coverage and the overall performance of the morphological tools. Other than these, the following problems are also well noted.

A. External sandhi. In Tamil, the obstruents (k,c,t,w,p) in the word initial position when preceded by a word form ending in a short vowel (a ,i, u, e, o), the diphthong (E), optionally glide y, ending in lYY and r appear as geminated and the first segment of which is always written as the final segment of the first word as shown below.

Examples for External Sandhi involves in Tamil.
*anwa**c** **c**attam* 'that law', *yAnYE**k** **k**utti* 'small elephant',
*curYrYulA**p** **p**ayaNi* 'tourists', *wAy**p** **p**Acam* 'motherly love',
*peVyar**p** **p**alakE* 'naming board', *wamilYY**w** **w**Ay* 'Mother of Tamil Nadu'.

However, the first words in each of these pairs is unrecognized because the additional word final consonant is the result of external sandhi. This requires the deletion of the consonants before they are passed on to the Morphological Analyzer.

B. Need for Sandhi Splitter. The words that are joined together require to be analyzed by Sandhi splitter beforehand. Or else, it will be a hectic task to add all the conjoined word forms in the database, since any subsequent words can be written together. The requirement of Sandhi Splitter is necessary to identify words which are combined together not due to inflectional rule. The sandhi splitter can separate these kinds of words which can be further forwarded to Morphological Analyzer. For instance,

nAteVfkum, nAtu + eVfkum 'nation+whole'
ifkulYlYa, ifku + ulYlYa 'here+being'
veNtumAnYAlum, veNtum+AnYAlum 'need+though'

C. Naturalized English Words. The words that are naturalized as Tamil especially from English need to be analyzed. The problem in identifying these words are a single word may have more than two orthographical and spelling variations. It differs according to the person how they pronounce. Therefore, it has to be studied through corpus that can reveal the different forms and their distributions.

For instance, for 'engineer'
inYginIr / eVnYginiyar / inYginYiyar

D. Colloquial Forms. In Tamil, the influence of colloquial forms can be normally seen in the written due to its nature of possessing two forms in Modern days as spoken and written. It is unavoidable to restrict the spoken, though it is informal. The problem may have been solved by providing the variant forms in the paradigmatic tables.

For instance,
porYAnY is used in spoken instead of *pokirYAnY* 'he is going'
paticcu for *patiwwu* 'having studied'

After implementing the above said suggestions, the analyzers may be expected to provide a more efficient and effective analysis.

V. Conclusion

The Apertium tool for Tamil is efficient in terms of time for processing a large number of words. The combination of Finite State Transducers (letter transducer) and the paradigm approach is more efficient and helps in faster parsing. The other advantage of the Apertium is that the current morphological database can be used to create a parallel morphological generator for Tamil.

[1] A finite state toolkit in Apertium to perform lexical processing

[2] Transliteration Scheme using wx-notation:
Tamil Orthography :
a A i I u U eV e E oV o O H
k f c F t N w n p m y r l v lYY lY rY nY j s h R

References

[1] Arden, A.H. 1976. *A progressive Grammar of the Tamil language*. Madras : The Christian Literature Society.

[2] Fercada, Mikel et.al. 2008. *Documentation of the Open-Source Shallow-Transfer Machine Translation platform Apertium.* Retrieved from http://www.gnu.org/copyleft/fd1.html.

[3] Parameswari K. 2009. *An improvised morphological Analyzer for Tamil: A case of implementing the opensource platform Apertium*. Unpublished M.Phil. Thesis. Hyderabad: University of Hyderabad.

[4] Ramaswamy, Vaishnavi. 2003. *A Morphological Analyzer for Tamil.* Unpublished Ph.D. Thesis. Hyderabad: University of Hyderabad.

[5] Uma Maheshwar Rao, G. Amba Kulkarni, P. and Christopher, M. 2007. *Morphological Analyzer and Its Functional Specifications for IL-ILMT System*. CALTS, Hyderabad: University of Hyderabad.

[6] Uma Maheshwar Rao, G. and Amba Kulkarni, P. 2006. *Computer Applications in Indian Languages*, Hyderabad: The centre for distance education, University of Hyderabad.

[7] Uma Maheshwar Rao, G. And Parameshwari, K. 2010. *On the Description of Morphological Data for Morphological Analysers and Generators: A case of Telugu, Tamil and Kannada.* Mona Parekh (ed.) in Morphological Analysers and Generators, pp73-81. Mysore:LDCIL,CIIL.www.ldcil.org/up/conferences/morph/presentation.html

[8] Uma Maheshwar Rao, G. and Christopher, M. 2010. *Word Synthesizer Engine.* Mona Parekh (ed.) in Morphological Analysers and Generators, pp73-81. Mysore: LDCIL,CIIL. www.ldcil.org/up/conferences/morph/presentation.html

[9] Uma Maheshwar Rao, G. 1999. *Morphological Analyzer for Telugu.* (electronic form). Hyderabad: University of Hyderabad.

[10] Uma Maheshwar Rao, G. 2002. *A Computational Grammar of Telugu*. (Momeograph) Hyderabad: University of Hyderabad.

[11] Vaidhya, Ashwini and Dipti Misra Sharma. 2009. *Using Paradigms for Certain Morphological phenomena in Marathi. 7th International Conference on NLP (ICON-2009).* New Delhi: Macmilan Publishers India Ltd.

[12] Viswanathan. S et.al. 2003. *A Tamil Morphological Analyser.* Recent Advances in NLP. 31-39. Mysore: Central Institute of Indian Languages.