An Improvised Morphological Analyzer cum Generator for Tamil:
A case of implementing the open source platform APERTIUM

Parameswari K,
Centre for Applied Linguistics and Translation Studies,
University of Hyderabad.
paramshekrishna@gmail.com
Overview

• Morphological Analyzer and Generator - two crucial tools in MT involving NLP.

• Deals with the improvized database implemented on Apertium for morphological analysis and generation.

• Discusses the evaluation of tools with large corpora to estimate the efficacy, coverage, and speed.
• A Morphological analyzer is a computational tool to analyze word forms into their roots, categories along with their constituent functional elements and the generator is an inverse of it.

• The 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 Finite State Transducers for one-pass analysis and generation, and the database is developed in the morphological model called Word and Paradigm.
Need for improvised Morphological tools

- Open Source - Easily Accessible
- Handling Derivation Morphology
- Speed
- User friendly
APERTIUM - ‘lttoolbox’

• Developed by the Tran'sducens research group at the Universitat d’Alacant in Spain.

• One of the open source machine translation systems has originated within the project “Open-Source Machine Translation for the languages of Spain”.

• A component called ‘lttoolbox’ for performing lexical processing tasks of language like Morphological Analyzer, Generator and POS Tagger.
The current versions of the Apertium toolbox as well as language data are available from the Sourceforge page sf.net/projects/apertium.
Data Organization In Apertium

Aperium 'lttoolbox' uses,

- **Word and Paradigm model** (Hockett, 1958) for linguistic database

- **Finite-State Transducers** (Jurafsky, 2003 and Mikel L. Forcada and et.al, 2008) as a computational algorithm for processing the data.

Lexical Resources required,

- **Paradigms**

- **Root word Dictionary**
Data Organization in Apertium

✓ Primarily the data is adopted from CALTS Morph.
✓ The data is improved.

1. Paradigm Improvization

2. Dictionary Improvization
Tamil Morphology

• Tamil, a Dravidian Language is known for its agglutinative morphology.

• Computing the analysis of Tamil Morphology demands a comprehensive but exhaustive analysis of its inflectional categories according to their functional properties.

• The present attempt classifies the morphological categories of Tamil based on their role in inflection. There are two classes,

    **Class A:** The forms which anchor with suffixes or morphosyntactic elements.

    **Class B:** The forms which are incapable of receiving such inflection.
Tamil Lexical Categories

Inflected

Open Class
- Nouns
- Verbs
- Adjectives

Closed Class
- Pronouns
- Numerals
- Locative Nouns

Uninflected

Indeclinables
- Postpositions
- Adverbs
- Conjunctions
- Interjections
- Particles
The present study considers pronouns as a distinct minor class because of its characteristic formation of oblique and idiosyncratic plural forms.

\textit{ufkalYE-} \textless nI (2p. pl)+ obl + Accusative\textgreater 'us+ Accusative'

The numerals have inflection with special particles such as \textbf{quantitative particles} (\textit{kal, arE, mukkAl etc}), \textbf{attributive particles} (\textit{per}) and \textbf{temporal particles} like time (\textit{maNi}) which involve peculiar inflection when compared to nouns.

\textit{walAvAyiram} \textless walA-attributive particle + Ayiram 'a thousand per head' \textgreater

The locative nouns are the indicators of time and space that are ascertained as a minor category because it exhibits an irregular inflection with morphosyntactic properties of noun. For instance,

\textit{arukil} \textit{aruku + Locative} 'near + Locative'
HANDLING INFLECTION

Paradigm definition section
Entry
String pair
left side
right side
Lexical Form of 'narampu'

The XML Format of inflectional paradigm for a noun.
HANDLING DERIVATION

<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 derivational paradigm for a deverbial pronoun.
A Dictionary Entry of the Lexeme 'maram'
# Database

## Paradigmatic Database

<table>
<thead>
<tr>
<th>Category</th>
<th>Number of Inflectional Classes</th>
<th>Number of Inflections per class</th>
<th>Category wise</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>20</td>
<td>743</td>
<td>57,322</td>
<td></td>
</tr>
<tr>
<td>Verb</td>
<td>29</td>
<td>934</td>
<td>10,114</td>
<td></td>
</tr>
<tr>
<td>Adjective</td>
<td>2</td>
<td>372</td>
<td>209</td>
<td></td>
</tr>
<tr>
<td>Pronoun</td>
<td>11</td>
<td>654</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Numeral</td>
<td>14</td>
<td>370</td>
<td>129</td>
<td></td>
</tr>
<tr>
<td>NST</td>
<td>7</td>
<td>67</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>Avy</td>
<td>-</td>
<td>-</td>
<td>206</td>
<td></td>
</tr>
</tbody>
</table>

The total number of inflectional classes is 68,060.
COMPILING AND PROCESSING

The data is compiled and processed by using the applications used in the lexical processing modules and tools (lttoolbox).

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 functioning the data.

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

START

Input (Inflected/Derived word form)

Compiled FST from XML encoded file

Process FST file
This is an Example FST for lexeme “pati” which can be a verb and noun.
Convert FST into Readable Format

Output

STOP
Morphological Analyzer:
Input: marafka\text{YE}
Output: marafka\text{YE/maram}<n><n><pl><3><o><E>$

Morphological Generator:
Input: ^vA<v><f><sg><3><o><nw>$
Output: vanwA\text{IY}
Evaluation

• The Morphological analyzer tool was tested using the corpus (CALTS corpus of 4.4 million words and EMILLI CIIL corpus of 4.8 million words) in order to find out its the coverage of the corpus.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Total words</th>
<th>Reg. words</th>
<th>coverage</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALTS Corpus</td>
<td>4,45,130</td>
<td>3,75,891</td>
<td>84.44%</td>
<td>0m0.289s</td>
</tr>
<tr>
<td>EMILLI CIIL Corpus</td>
<td>4,85,543</td>
<td>4,05,898</td>
<td>83.59%</td>
<td>0m0.297s</td>
</tr>
</tbody>
</table>
Speed

The morph analyzer is tested for its speed along with the other available Tamil Morphological analyzers which are developed in CALTS, University of Hyderabad and AU-KBC research Centre, Anna University. The speed of each modules for 1,00,000 words as follows.

<table>
<thead>
<tr>
<th>Morph</th>
<th>Processing Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALTS- MORPH</td>
<td>0m14.194s</td>
</tr>
<tr>
<td>AU-KBC MORPH</td>
<td>0m14.703s</td>
</tr>
<tr>
<td>CALTS-APERTIUM</td>
<td>0m0.198s</td>
</tr>
</tbody>
</table>

The above speed shows Calts-Apertium consumes less speed to analyze large number of data.
<table>
<thead>
<tr>
<th>Type</th>
<th>Word From</th>
<th>Frequency in the Corpus</th>
</tr>
</thead>
</table>
| 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)*                     | 765 occurrences  
                      | 6 occurrences       |
|                      | *vanweV 'You came'*  
                      | *(dialect)*                      |                   |
| Naturalized English words | *polIS 'police'* |                             | 20070 occurrences |
| Proper nouns          | *kaNNanY 'male name'*  
                      | *wamilYYnAtu 'Tamil Nadu'*       | 211 occurrences  
                      | 364 occurrences     |
Salient features of Apertium

• Easily Accessible.
• Speed.
• Two in One tool (Both Generator and Analyzer).
• GNU License – allow us to modify.
• Uses XML format which is easy to modify.
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 works more efficient and speedy parsing.

• The other advantage of the Apertium is that the current morphological database can be used to create a parallel morphological generator for Tamil.

• In future, further work can be carried out in this area to meet the other morphological lapses for attaining maximum coverage and precision.
Reference


THANK YOU