Morphological Generator for Tamil

-Menaka S, Vijay Sundar Ram and Sobha Lalitha Devi, AU-KBC Research Centre
Overview

- Tamil Morphology Key ideas
- Morphosyntax and Morphophonemics
- Finite State Automata
- Morphological generator
- Evaluation
Morphological Generator

- What is it? Tool used in NLP
- What does it do? Root word -> Inflected form
- Who needs it? Inflecting languages
- Where is it used? MT, IR
Methods used

- Rule-based method (Ganapathiraju and Levin 2006)
- Corpus-based method (Lantin et al, Dasgupta and Ng, 2007)
- Finite-state method (Beesley and Karttunnen, 2003)
Tamil Morphology – key ideas

- **Agglutinative** – Suffixes attach in series to the root.

  \[ \text{arapi} + \text{katal} + \text{in} + \text{araci} \Rightarrow \text{arapikkatalinaraci} \]

  'Arabian' + 'sea' + GEN + 'queen' => 'Queen of the Arabian Sea'

- **Morphosyntax** – Order in which suffixes attach to the root.

- **Morphophonemetics** – Changes that take place during suffixation.
MorphoSyntax of Lexical Categories - Nouns

- Nouns (include pronouns)- Take Inflectional and Derivational Suffixes.
- Root + {number} + {case} + {DISJ/COOR/EMPH} + {PSP} + {EMP} + {INT/SUPP}
  
  paiyan-kaL-ai-a: => paiyankaLaiya:

  ‘boy’-PL-ACC-INT => ‘the boys(OBJ)?’

- Derivation of verbs, adjectives, adverbs from nouns is possible.

  azaku + a:na => azaka:na

  'beauty' + ADJ => 'beautiful'
MorphoSyntax of Lexical Categories – Verbs...(1)

- Finite Verbs
- Root + Tense + PNG + \{DISJ/EMPH/EMP/INT/SUPP\}
  
  \textit{pa:r-tt-a:n-a:m} => \textit{pa:rtta:na:m}
  
  ‘see’-PST-3SM-SUPP => ‘it seems (he) saw’

- Root + INF + NEGVERB + \{DISJ/COOR/EMPH/EMP/INT/SUPP \}

  \textit{pa:r-a-illai-a:m} => \textit{pa:rrkavillaiya:m}
  
  ‘see’-INF-NEGVERB-SUPP => ‘it seems (x) did not see’
MorphoSyntax of Lexical Categories – Verbs...(2)

- Relative participle
- Root + Tense/NEG + RP

\[ pa:r-tt-a \rightarrow pa:rtta \]

‘see’-PST-RP \(\Rightarrow\) ‘who saw’

- Pronominalisation

\[ pa:r-tt-a-avan \rightarrow pa:rttavan \]

‘see’-PST-RP-he \(\Rightarrow\) ‘he who saw’
MorphoSyntax of Lexical Categories – Verbs…(3)

- Non-Finite Verbs

  root + {NEG} +
  INF/VBP/COND/COND/COND/HORT/OPT +
  {DISJ/COOR/EMPH} + {EMP} + {INT/SUPP}

  \[ \text{pa:Tu-a} \rightarrow \text{pa:Ta} \]

  'sing'-INF => ‘to sing’

- Derivation of nouns, adjectives and adverbs.
Morphophonemics

- Changes that occur when a suffix attaches to a root word.
- Change depends on
  - the nature of the end letter of the root word
  - the nature of the start letter of the suffix

‘Mala’-INS => ‘by Mala’  ‘tooth’-INS => ‘using tooth’
Paradigm-based approach.

- Follows from the morphophonemic changes.
- Those root words which behave similar are grouped.
- Paradigmatic classification for Tamil
  - 36 noun paradigms and 34 verb paradigms

- *ya:ci* ‘beg’ takes *tt/kkiR/pp* as the three tense markers.
- *viya* 'wonder' takes *Ñt/kkiR/pp* as the three tense markers.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>‘beg’-PST-3SM</td>
<td>‘beg’-PRE-3SM</td>
<td>‘beg’-FUT-3SM</td>
</tr>
<tr>
<td>viya-Ñt-a:n</td>
<td>viya-kkiR-a:n</td>
<td>viya-pp-a:n</td>
</tr>
<tr>
<td>‘wonder’-PST-3SM</td>
<td>‘wonder’-PRE-3SM</td>
<td>‘wonder’-FUT-3SM</td>
</tr>
</tbody>
</table>
Finite State Automata...(1)

- A Finite-state automaton is a model of behavior consisting of a finite number of states, transitions from each state to another state and actions at each transition.

- Morphological generator moves from one state to another as each attribute is applied to the stem and the suffix is generated.

  \[ \text{paiyan-kaL-ai-a: } \Rightarrow \text{paiyankaLaiya:} \]

  ‘boy’-PL-ACC-INT => ‘the boys(OBJ)?’
Finite State Automata...(2)


<table>
<thead>
<tr>
<th>From State</th>
<th>To State</th>
<th>Attribute</th>
<th>Form Generated</th>
<th>Finalform</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>PL</td>
<td><em>paiyankaLa</em></td>
<td><em>paiyankaLa</em></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>ACC</td>
<td><em>paiyankaLaI</em></td>
<td><em>paiyankaLaI</em></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>INT</td>
<td><em>paiyankaLaIa:</em></td>
<td><em>paiyankaLaIa:</em></td>
</tr>
</tbody>
</table>

![Diagram of Finite State Automata](image)
Design of MorphGenerator for Tamil

- A finite state automaton
- Moves from one state to another while attaching suffixes.
- End state produces the desired output
- Resource files
  - Lexicon
  - Suffix table
  - State table
  - Morphophonemic rules
Evaluation – Experiment 1

2556 input words with noun roots spanning different paradigms and different attributes were tested.

<table>
<thead>
<tr>
<th>No. of True Positives (TP)</th>
<th>No. True Negatives (TN)</th>
<th>No. of False Positives (FP)</th>
<th>No. of False negatives (FN)</th>
<th>Precision TP/(TP + FP)</th>
<th>Recall TP/(TP + FN)</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>2413</td>
<td>115</td>
<td>5</td>
<td>23</td>
<td>0.997</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>
**Evaluation – Experiment 2**

19152 input words with verb roots spanning all the paradigms and various attributes were tested.

<table>
<thead>
<tr>
<th>No. of True Positives (TP)</th>
<th>No. True Negatives (TN)</th>
<th>No. of False Positives (FP)</th>
<th>No. of False negatives (FN)</th>
<th>Precision TP/(TP + FP)</th>
<th>Recall TP/(TP + FN)</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>17361</td>
<td>1451</td>
<td>38</td>
<td>302</td>
<td>0.997</td>
<td>0.98</td>
<td>0.99</td>
</tr>
</tbody>
</table>
Thank You!