A Framework for Learning Morphology using Suffix Association Matrix

Mrs. Shilpa Desai
Dr. Jyoti Pawar
Prof. Pushpak Bhattacharyya

The 5th Workshop on South and Southeast Asian Natural Language Processing
The 25th International Conference on Computational Linguistics
Dublin, Ireland
23rd August 2014
Outline of the presentation

• Morphology
  – Introduction
  – Types
  – For Indian Languages Hindi and Konkani
• Approaches to Morphology Learning
• Suffix Association Matrix (SAM)
• Experimental Results Using SAM
• Learning Morphology Using SAM
• Conclusion
Morphology
A study of word structure (1/2)...

• Words are made up of Morphemes
  – walking = walk + ing
  – unplugged = un + plug + ed
Morphology
A study of word structure (2/2)...

• Words are made up of Morphemes
  – walking = walk + ing
  – unplugged = un + plug + ed

• Morphemes
  – Stems
  – Affixes
    • Prefixes, suffixes, infixes and circumfixes
Types of Morphology

**Inflectional**
- Deal with the variations of forms of the same word
  - \textit{walk} \rightarrow \textit{walks, walking, ...}
- Give rise to \textit{inflectional affixes}

**Derivational**
- Deal with the production of new words
  - \textit{learn (Verb) + er} \rightarrow \textit{learner (Noun)}
- Give rise to \textit{derivational affixes}
Morphology For Indian Languages

Hindi
• Affixes that apply
  – Prefixes
  – Suffixes
    • Inflectional Suffixes
      – Noun (moderate)
      – Verb (high)
    • Derivational Suffixes (moderate)

Konkani
• Affixes that apply
  – Prefixes (very rare)
  – Suffixes (common)
    • Inflectional Suffixes
      – Noun (high > 100)
      – Verb (very high > 800)
    • Derivational Suffixes (moderate)
Approaches used to Learn Morphology

• **Rule Based / Finite State Based**
  – Used for word segmentation
  – Used by Stemmers and Morphological Analyzers

• **Unsupervised**
  – Used for word segmentation, affix identification, stemming
  – Can be used for automatic paradigm generation
Approaches used to Learn Morphology

• **Rule Based / Finite State Based**
  – Linguistic knowledge of language required to build
  – Time consuming, linguistic experts are required hence costly

• **Unsupervised**
  – Language independent
  – Data driven approach
Suffix Association Matrix (SAM)

- SAM measures how many times a suffix occurs with some other suffix in corpus.
- Sample instance of SAM

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>er</th>
<th>ing</th>
<th>ed</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>-</td>
<td>46</td>
<td>225</td>
<td>129</td>
</tr>
<tr>
<td>er</td>
<td>46</td>
<td>-</td>
<td>22</td>
<td>15</td>
</tr>
<tr>
<td>ing</td>
<td>225</td>
<td>22</td>
<td>-</td>
<td>21</td>
</tr>
<tr>
<td>ed</td>
<td>129</td>
<td>15</td>
<td>21</td>
<td>-</td>
</tr>
</tbody>
</table>
Learning Morphology using Suffix Association Matrix (SAM)

- Unsupervised approach.
- Identifies derivational suffixes using lexicon as input.
- Identifies inflectional and derivational suffixes using corpus as input.
- Works for concatenative morphology.

Contact author: sndesai@gmail.com
Learning Morphology using Suffix Association Matrix (SAM)

• Generates paradigms
  – Paradigm is defined as a set of suffixes which go with a stem.

• For Indian languages like Konkani where most inflectional forms have suffixes, SAM helps identify stem and suffixes
# Experimental Results

Paradigms generated using Lexicon as input

<table>
<thead>
<tr>
<th>Language</th>
<th>Suffix Set</th>
<th>Corresponding Word Stem</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>{ist, y}</td>
<td>anarch, entomolog, metallurg, misogyn, phthalmolog, optometr, ornitholog, ...</td>
</tr>
<tr>
<td>English</td>
<td>{NULL, ation, ed}</td>
<td>confirm, disorient, ferment, fix, infest, ...</td>
</tr>
</tbody>
</table>

Sample segmentation obtained: **anarchist = anarch + ist**
## Experimental Results

Paradigms generated using Lexicon as input

<table>
<thead>
<tr>
<th>Language</th>
<th>Suffix Set</th>
<th>Corresponding Word stems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hindi</td>
<td>{क, ण, ित}</td>
<td>आरक्ष araksh, नियंत्र nyantr, निर्धार nirdhar, पोष posh, प्रदूष pradush, शोष shosh, ...</td>
</tr>
<tr>
<td>Hindi</td>
<td>{NULL, ाना, ी}</td>
<td>गड़बड़ gadbad, गरम garam, झिलमिल zilmil, दोस्त dost, धमक dhamak, मालिक malik, मेहनत mehanat, ...</td>
</tr>
</tbody>
</table>

Sample segmentation obtained: नियंत्रण = नियंत्र + ण
nityantran = nitayantra + n
## Experimental Results

Paradigms generated using Lexicon as input

<table>
<thead>
<tr>
<th>Language</th>
<th>Suffix Set</th>
<th>Corresponding Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Konkani</td>
<td>{NULL, ाचें, ी}</td>
<td>अवतार avtar, आर्यसमाज aaryasamaj, उपेग upegh, एकमत ekmath, करप karap, गुलाब gulab, ...</td>
</tr>
<tr>
<td>Konkani</td>
<td>{NULL, ावप, ीत}</td>
<td>उजवाड uzvad, कुचकुच kuchkuch, खटखट katkat, खडखड khadkhad, ...</td>
</tr>
</tbody>
</table>

Sample segmentation obtained: उजवाडावप = उजवाड + ावप

ujvadavap = ujvad + avap
A Framework for Learning Morphology using SAM

Input: Lexicon /Corpus

- Suffix Identifier
- Candidate Stem-Suffix List
- Stem-Suffix Pruner
- Stem-Suffix List

Output: Morphology Paradigms

- Morphology Paradigm Generator
- Suffix Association Matrix
- Suffix Association Matrix Generator
- Initial Paradigms

Primary Paradigm Generator

Contact author: sndesai@gmail.com

COLING 2014
Learning Morphology using SAM – Step 1

**Suffix Identifier Module:**
Identifies candidate stem and candidate suffix

**Example:**
Input L = \{walk, walks, walking, talk, talks, tall, talking, take\}
Candidate Stem = \{walk, talk\}
Candidate Suffix = \{s, ing, NULL\}
Here every stem occurs with at least two suffixes and every suffix occurs with at least two stems.
To get possible **stem** from two words \{walk, walking\} look at maximum common beginning letters.
If a stem is found for a word the remaining part is considered **suffix** \{walker, walking\}
Learning Morphology using SAM – Step 2

Stem Suffix Pruner Module:
Fixes problem of over-stemming applying Heuristic H1

Example:
Input L = \{addict, addiction, addictive, affirmation, affirmative, apprehension, apprehensive, contradict, contradiction, contradictive\}

Before pruning
Candidate Stem = \{addict, affirmati, apprehensi, contradict\}
Candidate Suffix = \{NULL, ion, ive, on, ve\}

After pruning
Stem = \{addict, affirmat, apprehens, contradict\}
Suffix = \{NULL, ion, ive\}
Learning Morphology using SAM – Step 3

Primary paradigm Generator:
Generates paradigm for Stem – Suffix List
Example:
Input L = {addict, addiction, addictive, affirmation, affirmative, apprehension, apprehensive, contradict, contradiction, contradictive}
Stem = {addict, affirmat, apprehens, contradict}
Suffix = {NULL, ion, ive}
Paradigm
1. {NULL, ion, ive} → {addict, contradict}
2. {ion, ive} → {affirmat, apprehens}
Suffix Association Matrix (SAM) Generator: Generates the suffix association matrix.

1. \{NULL, ion, ive\} \rightarrow \{addict, contradict, extort, extract, insert, intercept\} 6 stems
2. \{ion, ive\} \rightarrow \{affirmat, apprehens\} 2 stems

**SAM**

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>ion</th>
<th>ive</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>ion</td>
<td>6</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>ive</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
Learning Morphology using SAM – Step 5

Morphology Paradigm Generator:
Refines initial paradigms generated using suffix association matrix to prune chance segmentations like cannot = canno + t cannon = canno + n

Figure 1: Unsupervised Morphology Learner (UML) Framework
Conclusion (1/3)...

• Significance of **Suffix Association Matrix (SAM)**
  – SAM can be used to segment words correctly.
  – Example 1:
    • Input word: **cannon**
    • Possible segmentation cannon = **canno + n** if the word **cannot** is in corpus
    • Check value for (n,t) in SAM, value will be low so reject segmentation cannon = **canno + n**
Conclusion (2/3)...

- Significance of **Suffix Association Matrix (SAM)**
  - Example 2:
    - Input word: `bother`
    - Possible segmentation `bother` = `both` + `er`
    - Value for `(er,NULL)` in SAM is high so check for some different high association suffixes of `er` such as `ing`
    - Check for existence of `bothing` in large corpus.
    - If many high association suffix words are found, accept the segmentation, otherwise reject
Conclusion (3/3) ...

• Related methods, normally place a restriction on stem lengths
• SAM helps remove stem length restriction and is an alternate method which works for short stem length words
Thank You

dेव बरें करु
Dev bore koru