Automatic Corpus-based Thai Word Extraction with the C4.5 Learning Algorithm

Virach Sornlertlamvanich, Tanapong Potipiti, and Thatsanee Charoenporn
National Electronics and Computer Technology Center (NECTEC), THAILAND
Problems of Thai Word Identification

- No word boundary -> Thais have difficulties in defining words.
  
  Example:
  
  …notwithstandingiampresenting...

  Notwithstanding, Not + with + standing,
  Not + withstanding

- Thai processing relies on human created dictionaries which have several limitations.
  - inconsistency
  - coverage
Introduction (2)

- Words cannot be defined clearly and consistently:
  - problems in
    - Machine Translation, Information Retrieval
    - Speech Synthesis
    - Speech Recognition
    - etc.
Our Approach (1)

- Corpus-Based Word Extraction
  - Unlabelled Corpus-Based
  - Automatic
  - Clear and Computable
Our Approach(2)

- Building a suffix array of 3-to-30-character substrings from the corpus
- Word/Non-word string disambiguation
- Applying the C4.5 machine learning
- The attributes applied to the disambiguation are:
Attributes(1) : Left and Right Mutual Information

\[ Lm(xyz) = \frac{p(xyz)}{p(x)p(yz)} \quad Rm(xyz) = \frac{p(xyz)}{p(xy)p(z)} \]

where

- \( x \) is the leftmost character of string \( xyz \)
- \( y \) is the middle substring of \( xyz \)
- \( z \) is the rightmost character of string \( xyz \)
- \( p(\cdot) \) is the probability function.

High mutual information implies that \( xyz \) co-occurs more than expected by chance. If \( xyz \) is a word, its \( Lm \) and \( Rm \) must be high.

...Efunction... and ...Efunction...
Attributes(2) : Left and Right Entropy

\[ Le(y) = - \sum_{x \in A} p(xy \mid y) \log_2 p(xy \mid y) \]
\[ Re(y) = - \sum_{z \in A} p(yz \mid y) \log_2 p(yz \mid y) \]

where
- \( x \) is the leftmost character of string \( xyz \)
- \( y \) is the middle substring of \( xyz \)
- \( z \) is the rightmost character of string \( xyz \)
- \( p(\ ) \) is the probability function.

Entropy shows the variety of characters before and after a word. If \( xyz \) is a word, its left and right entropy must be high.
Example: \( \ldots ?function\ldots \) , \( \ldots ?unction\ldots \)
Attributes(3): Frequency, Length, Functional Words

- **Frequency**
  Words tend to be used more often than non-word string sequences.

- **Length**
  Short strings are likely to happen by chance.
  The long and short strings should be treated differently.

- **Functional Words**
  Functional words are used mostly in phrases. They are useful to disambiguate words and phrases.
  
  \[ Func(s) = \begin{cases} 
  1 & \text{if } s \text{ contains functional words} \\
  0 & \text{if otherwise} 
  \end{cases} \]
Attributes(4): First Two and Last Two Characters

- **Frequency of the first-two characters of the considered string which appears in the first-two characters of words in the dictionary**
  
  high frequency -> the beginning of the considered string conforms to the Thai spelling system.

  Ex. 

  *Function*: how likely *fu* can be the beginning of word.

- **This idea can be also applied to the last-two characters.**
Applying C4.5 to Word Extraction

Extracting Strings from the Training Corpus
- Computing the Attributes Value
- Tagging the Strings
  - C4.5
  - The Decision Tree

Extracting Strings from the Test Corpus
- Computing the Attributes Value
  - $Re > 1.78$
    - $Y$
    - $N$
  - $Lm > 14233$
    - $Y$
      - $Func = 0$
        - $Y$
          - is not a word
        - $N$
          - is not a word
    - $N$
      - is not a word
  - $Rm = 132.6$
    - $Y$
      - is not a word
    - $N$
      - is not a word
  - is a word

Word Extraction

The Decision Tree
### Experimental Results (1)

#### The Precision of Word Extraction

<table>
<thead>
<tr>
<th></th>
<th>No. of strings extracted by the decision tree</th>
<th>No. of words extracted</th>
<th>No. of non-word strings extracted</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training Set</strong></td>
<td>1882 (100%)</td>
<td>1643 (87.3%)</td>
<td>239 (12.7%)</td>
</tr>
<tr>
<td><strong>Test Set</strong></td>
<td>1815 (100%)</td>
<td>1526 (84.1%)</td>
<td>289 (15.9%)</td>
</tr>
</tbody>
</table>

#### The Recall of Word Extraction

<table>
<thead>
<tr>
<th></th>
<th>No. of words that has more than 2 occurrences in corpus</th>
<th>No. of words extracted by the decision tree</th>
<th>No. of words in corpus that are found RID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training Set</strong></td>
<td>2933 (100%)</td>
<td>1643 (56.0%)</td>
<td>1833 (62.5%)</td>
</tr>
<tr>
<td><strong>Test Set</strong></td>
<td>2720 (100%)</td>
<td>1526 (56.1%)</td>
<td>1580 (58.1%)</td>
</tr>
</tbody>
</table>

Remark: These precision and recall are measured against 30,000 strings that occur more than 2 times in the corpus and conform to some simple Thai spelling rules.
## Experimental Results (2)

### Word Extraction VS. a Dictionary

<table>
<thead>
<tr>
<th></th>
<th>No. of words extracted by the decision tree</th>
<th>No. of words extracted by the decision tree which is in RID</th>
<th>No. of words extracted by the decision tree which is not in RID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training Set</strong></td>
<td>1643 (100.0%)</td>
<td>1082 (65.9%)</td>
<td>561 (34.1%)</td>
</tr>
<tr>
<td><strong>Test Set</strong></td>
<td>1526 (100.1%)</td>
<td>1046 (68.5%)</td>
<td>480 (31.5%)</td>
</tr>
</tbody>
</table>
The Relationship of Accuracy, Frequency and Length

- Both precision and recall are getting higher as the length and frequency of strings increase.
- The new created words have tendency to be long. Our extraction yields a high accuracy in extracting temporal words.
Conclusion

- C4.5 has been applied to word extraction, using attributes: mutual information, entropy, frequency, length, functional words, and the first two and last two characters.
- Our approach yields 85% in precision and 56% in recall measure.
- Our approach is promising for building a corpus-based dictionary for non-word boundary languages.
Thank You for Your Attention