Intelligent Key Prediction
by N-grams and Error-correction Rules

Kanokwut Thanadkran, Virach Sornlertlamvanich and Tanapong Potipiti

Information Research and Development Division, National Electronics and Computer Technology Center (NECTEC), Thailand
Outline

- Introduction
- Related Works
- The Approach
- Experiments
- Conclusions
Solutions

The Thai-English Keyboard System: Problems and Solutions

Introduction
The Thai-English Keyboard System

Characters.
- Distinguish a set of languages and to use special keys to select
- 2 sets of characters.
- 2 languages.
- 4 characters/1 button.
- 39 in Symbol.
- 52 in English.
- 83 in Thai.
- 174 characters.
Two annoyances for Thais to type Thai-English bilingual texts.

To switch between languages by using a language switching key.

To distinguish a set of characters by using a combination of shift-key and a character-key.

Other multilingual users face the same problems.
Language Identification and Key Prediction
Automatic language identification.

Considering only surrounding text.

Problems.

- Difficult to switch to the proper language.
- Often detect incorrect language.
- No key prediction.

Language Identification

Microsoft Word 2000.

- Often detect incorrect language.
- No key prediction.
- Difficult to switch to the proper language.
Automatic word prediction based on lexical tree. Word class n-grams are employed. Base on lexical tree. Automatic word prediction. No language identification. Required large amount of memory space.
The Approach

Pattern Shortening

Overview, Language Identification, Key Prediction and
There are two main processes in our system.

Automatic language identification.

Automatic Thai key prediction.

Language? Thai

Thai Key Prediction

Eng

Language?

Thai

Key Input

The System Overview
(2) \( \left( \sum_{k}^{i} \right) \prod_{m} \mathcal{E}_{\text{prob}} = q \cdot \prod_{m} \mathcal{E}_{\text{prob}} \)

(1) \( \left( \sum_{k}^{i} \right) \prod_{m} \mathcal{L}_{\text{prob}} = q \cdot \prod_{m} \mathcal{L}_{\text{prob}} \)
(3)

\[
\left( i \mid i_k \right) \prod_{i=1}^{n} \max_{u} \arg \max_{u} \left( \sum_{c_1, c_2, \ldots, c_u} \right)
\]
Some character sequences often generate errors.

Error patterns are collected for error-correction process.

Error correction rules

Mutual information is used to optimize the patterns.

Errors from Tri-gram prediction

Training corpus

Tri-gram prediction

Tri-gram
<table>
<thead>
<tr>
<th>Correct Patterns</th>
<th>Error Key Pattern No.</th>
<th>Error Key Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\[
\begin{align*}
(5) & \quad \frac{(z) d(\lambda x) d}{(z\lambda x) d} = (z\lambda x) wM \\
(4) & \quad \frac{(z\lambda) d(\lambda x) d}{(z\lambda x) d} = (z\lambda x) wL
\end{align*}
\]
The Error-correction Process

Finite Automata pattern matching is applied. The longest pattern is selected.

Finite Automata

Terminal?

Yes

No

Key Input

Finite Automata

The Error-correction Process
Experiments

Language Identification and Key Prediction
The first 6 characters are enough.

Using artificial corpus for testing.

Bi-Gram is employed.

<table>
<thead>
<tr>
<th>Identification Accuracy (%)</th>
<th>Identification (number of first characters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.96</td>
<td>7</td>
</tr>
<tr>
<td>1.96</td>
<td>6</td>
</tr>
<tr>
<td>9.86</td>
<td>5</td>
</tr>
<tr>
<td>97.06</td>
<td>4</td>
</tr>
<tr>
<td>94.27</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Training Corpus (%)</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Unshifted Alphabets</td>
<td>88.63</td>
</tr>
<tr>
<td>Shifted Alphabets</td>
<td>11.37</td>
</tr>
</tbody>
</table>

Key Prediction Corpus Information

- 5 MB Test Corpus
- 25 MB Training Corpus
<table>
<thead>
<tr>
<th>Error Correction</th>
<th>Tri-gram Prediction</th>
<th>Tri-gram Prediction + Error Correction</th>
<th>Prediction Accuracy from Training Corpus (%)</th>
<th>Prediction Accuracy from Test Corpus (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.53</td>
<td>93.11</td>
<td>99.53</td>
<td>93.11</td>
<td>92.21</td>
</tr>
<tr>
<td>99.42</td>
<td>92.21</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

The Approach and Future Works
The Approach

We have applied the tri-gram model and error-correction rules for intelligent Thai key prediction and English-Thai language identification.

The experiment reports 99 percent in accuracy, which is very impressive.
Hopefully, this technique is applicable to other Asian languages and multilingual systems. Our future work is to apply the algorithm to mobile phones, handheld devices and multilingual input systems.
The End

Questions and Answers