

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

Kanokwut Thanadkran, Virach Sornlertlamvanich and Tanapong Potipiti

National Electronics and Computer Technology Center,  
National Science and Technology Development Agency,  
Ministry of Science and Technology Environment,  
22<sup>nd</sup> Floor Gypsum Metropolitan Tower 539/2  
Sriyudhya Rd. Rajthevi Bangkok 10400 Thailand

Email: kanokwutt@notes.nectec.or.th, tanapong@nectec.or.th and virach@nectec.or.th

## Abstract

In this paper, we propose an intelligent aid for text input method in order to provide an easier way for text inputting with the conventional keyboard. We use the character n-gram model and error correction rules to identify the language being typed and to predict the most probable character string without extra keystroke. The character n-gram model is also required only a small amount of memory spaces when using Bi-gram and Tri-gram. The paper also proposes rule-reduction algorithm applying mutual information to reduce the error-correction rules. Our algorithm archives more than 99% accuracy in both language identification and key prediction.

**Keywords:** key prediction, error correction rule and language identification.

## 1 Introduction

For Thai users, there are always two annoyances while typing Thai-English bilingual documents, which are usual for Thais. The first is when the users want to switch from typing Thai to English, they have to input a special key to tell the operating system to change the language mode. Further, if the language-switching key is ignored, they have to delete the token just typed and re-type that token after language switching. The second is that Thai has more than 100 alphabets, to input about half of all Thai characters, the user has to use combinations of two keys (shift key + another key) to input them. Some of the other Asian language users also have the same problem.

It will be wonderful, if there is an intelligent keyboard system that is able to perform these two tasks –switching language and shifting key– automatically. This paper proposes a practical solution for these disturbances by applying tri-gram character probabilistic model and error-correction rules. To optimize number of the generated error-correction rules, we propose a rule reduction approach using mutual information. More than 99 percent of key prediction accuracy results are reported.

## 2 Related Works

There are two related works on an intelligent aid for text input method.

Firstly, Microsoft Word 2000 provides simple automatic language detection by considering only the existing surrounding text. This feature cannot detect user's input intention during the typing. As a result, the input character sequence cannot switch back and forth to accept the proper language input.

Secondly, Zheng et al. [1] applied lexical trees and Chinese word n-grams to word prediction for inputting Chinese sentences by using digit keys. They reported 94.4% prediction accuracy. However, they did not deal with automatic language identification process. The lexicon trees they employed required a large amount of memory space.

## 3 The Approach

### 3.1 Overview

In the traditional Thai keyboard input system, a key button with the help of the language-switching key and the shift key can output 4 different characters.

For example, in the Thai keyboard the 'a'-key button can represent 4 different characters in different modes as shown in table 1.

	without Shift	with Shift
English Mode	'a'	'A'
Thai Mode	'พ'	'ถ'

*Table 1: A Key Button can Represent Different Characters in Different Modes.*

However, using NLP technique, the Thai-English keyboard system which can predict the key users intend to type without the combination use with the language-selection key and the shift key, should be efficiently implemented. We propose an intelligent keyboard system to solve this problem and have implemented with a successful result.

To solve this problem, there are basically two steps: language identification and Thai key prediction. Figure 1 shows how the system works.

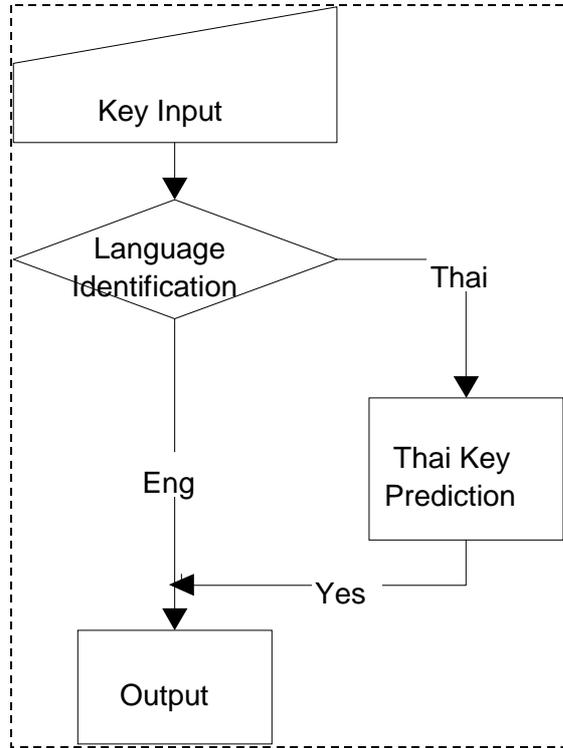


Figure 1: How the System Works

### 3.2 Language Identification

The following example illustrates the disturbance of language switching. In the Thai input mode, typing a word “language” will result “ลัฟฟล”. It is certain that the user has to delete sequence “ลัฟฟล” and then switches to the English mode before retyping the key sequence to get the correct result of “language”. Therefore an intelligent system to perform language switching automatically is helpful in eliminating the annoyance.

In general, different languages are not typed connectedly without spaces between them. It is a conventional reason to make the text legible. The language-identification process will start when a non-space character is typed after a space. Many works in language identification [3, 5] have claimed that the n-gram model gives a high accuracy on language identification. After trying both tri-grams and bi-grams, we found that bi-grams were superior. We then compare the following bi-grams’ probabilities for language identification.

$$Tprob = \prod_{i=1}^{m-1} p_T(K_i K_{i+1}) \quad (3.1)$$

and

$$Eprob = \prod_{i=1}^{m-1} p_E(K_i K_{i+1}) \quad (3.2)$$

where

$p_T()$  is the normalized probability of the bi-gram key buttons considered in Thai texts,

$K$  is the considering key button,

$p_E()$  is the normalized probability of the bi-gram key buttons considered in English texts,

$Tprob$  is the probability of the considering key-button sequence to be Thai,

$Eprob$  is the probability of the considering key-button sequence to be English,

$m$  is the number of the leftmost characters of the corresponding token. (See more details in the experiment.)

The current inputting language is identified by comparing the key sequence probabilities. The language identified as Thai if  $Tprob > Eprob$  and vice versa.

### 3.3 Key Prediction without Using Shift Key for Thai Input

#### 3.3.1 Tri-gram Key Prediction

The tri-gram model is selected to apply for the Thai key prediction. The problem of the Thai key prediction can be defined as:

$$\tau = \arg \max_{c_1, c_2, \dots, c_n} \prod_{i=1}^n p(c_i | c_{i-1}, c_{i-2}) \cdot p(K_i | c_i) \quad (3.3)$$

where

$\tau$  is the sequence of characters that maximizes the character string sequence probability,

$c$  is the possible input character that possible for the key button  $K$ ,

$K$  is the key button,

$n$  is the length of the considering token.

#### 3.3.2 Error Correction for Thai Key Prediction

In some cases of Thai character sequence, the tri-gram model fails to predict the correct key. To correct these errors, the error-correction rule method proposes by Brill [1, 2] is employed.

##### 3.3.2.1 Error-correction Rule Extraction

Only the prediction errors after applying tri-gram prediction to the training corpus are considered to prepare the error correction rule. The left and right three keys input

around each error character and the correct pattern corresponding with the error will be collected as an error-correction pattern. For example, if the input key sequence “gliklkl9in” is predicted as “เสริมฐุศาสตริ”, where the correct prediction is “เสริมฐุศาสตริ”. The string “iklkl9” is then collected as an error sequence and “มฐุศาสต” is collected as the correct pattern to amend the error.

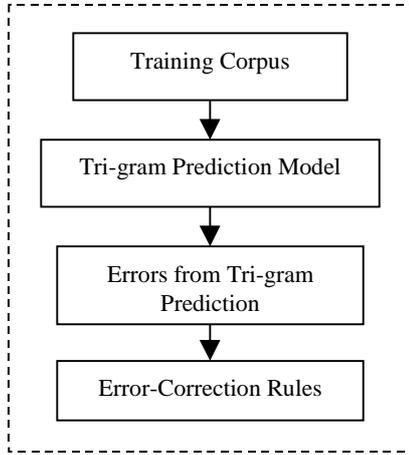


Figure 2: Error-Correction Rule extraction

### 3.3.2.2 Rules' Reduction

In the process of collecting the patterns, there are a lot of redundant patterns collected. For example, patterns no.1-3 in Table 2 should be reduced to pattern 4.

To reduce the number of rules, left mutual information and right mutual information [7] are employed. When all patterns are shortened, the duplicate patterns are then eliminated in the final step.

Pattern No.	Error Key Sequences	Correct Patterns
1.	k lkl9	มฐุศาสต
2.	mpk kl9	ทยาศาสต
3.	kkk kl9	ายาศาสต
4.	kl9	ศาสต

Table 2: Error-Correction Rule Reduction

Left mutual information ( $Lm$ ) and right mutual information ( $Rm$ ) are the statistics used to shorten the patterns.  $Lm$  and right  $Rm$  are defined as follows.

$$Lm(xyz) = \frac{p(xyz)}{p(x)p(yz)} \quad (3.4)$$

and

$$Rm(xyz) = \frac{p(xyz)}{p(xy)p(z)} \quad (3.5)$$

where

- $xyz$  is the pattern being considered,
- $x$  is the leftmost character of  $xyz$ ,
- $y$  is the middle sub-string of  $xyz$ ,
- $z$  is the rightmost character of  $xyz$ ,
- $p()$  is the probability function.

The pattern-shortening rules are as follows:

- 1) If the  $Rm(xyz)$  is less than 1.20 then pattern  $xyz$  is reduced to  $xy$ .
- 2) Similarly, If the  $Lm(xyz)$  is less than 1.20 then pattern  $xyz$  is reduced to  $yz$ .
- 3) Rules 1 and 2 are applied recursively until the considered pattern cannot be shortened anymore.

After all patterns are shortened, the following rules are applied to eliminate the redundant patterns:

- 1) All duplicate rules are unified.
- 2) The rules that contribute less 0.2 per cent of error corrections are eliminated.

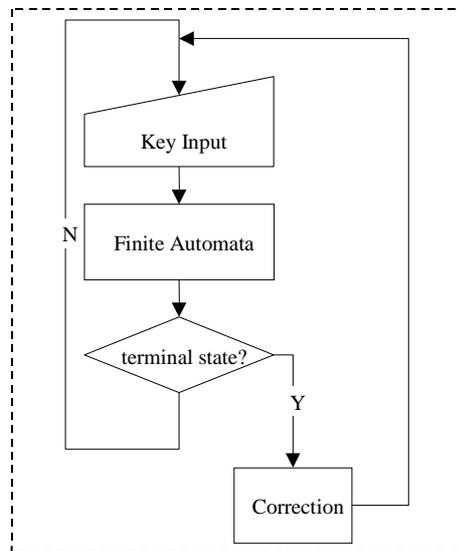


Figure 3: The Error-Correction Process

### 3.3.3 Applying Error-correction Rules

There are three steps in applying the error-correction rules:

- 1) Search the error patterns in the text being typed.

- 2) Replace the error patterns with the correct patterns.
- 3) If there are more than one pattern matched, the longest pattern will be selected.

In order to optimize the speed of error-correction processing and correct the error in the real time, the finite-automata pattern matching [4, 6] is applied to search error sequences.

We constructed an automaton for each pattern, then merge these automaton into one as illustrated in Figure 4.

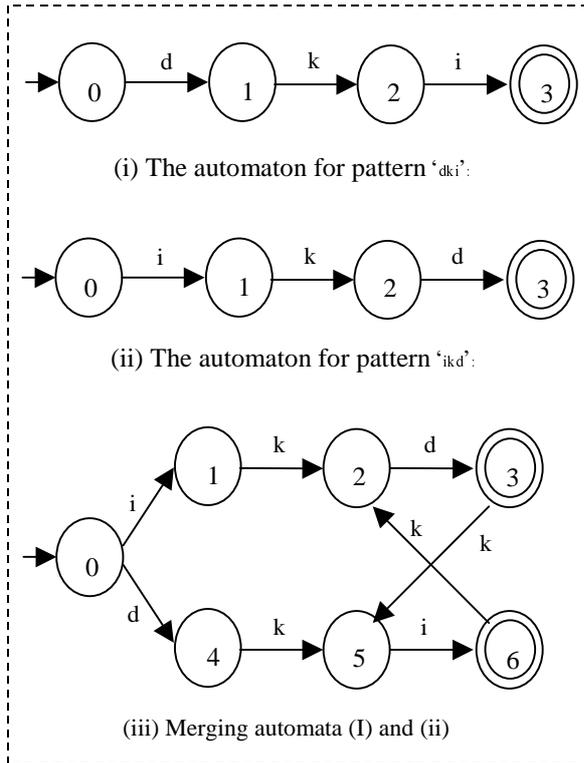


Figure 4: The Example of Constructing and Merging Automata

## 4. Experiments

### 4.1 Language Identification

For the language identification, we employ the approach of character bi-gram extracted from the 35,000 Thai words and 100,000 English words. To create an artificial corpus to test the automatic language switching, 10,000 random words from an English dictionary and 10,000 random words from a Thai dictionary are selected to build a corpus for language identification experiment. All characters in the test corpus are converted to their mapping characters of the same key button in normal mode (no shift key applied) without applying the language-switching key. For example, character '๗', '๗' and '๗' will be converted to '๗'. Table 3 shows the accuracy of our language identification algorithm with different values of  $m$  - the number of the

leftmost characters of the token used for language identification.

$m$ (the number of the first characters to be considered)	Identification Accuracy (%)
3	94.27
4	97.06
5	98.16
6	99.10
7	99.11

Table 3: The Accuracy of Thai-English Language Identification

As a conclusion the first 6 characters of the token are enough to yield a high accuracy on English-Thai language identification.

### 4.2 Thai Key Prediction

#### 4.2.1 Corpus Information

The sizes of training and test sets applied to our key prediction algorithm are 25 MB and 5 MB respectively. The table below shows the percentage of the shift and unshift alphabets used in the corpora.

	Training Corpus (%)	Test Corpus (%)
Unshift Alphabets	88.63	88.95
Shift Alphabets	11.37	11.05

Table 4: Information on Alphabets Used in Corpus

#### 4.2.2 Thai Key Prediction with Tri-gram

Because the Thai language has no word boundary, we trained the tri-gram model from a 25-MB Thai corpus instead of a word list from a dictionary as in the language identification. The tri-gram model was tested on another 5-MB corpus (the test set). Similarly, a typing situation without applying shift key was simulated for the test. The result is shown in Table 5.

Training Corpus (%)	Test Corpus (%)
93.11	92.21

Table 5: The Accuracy of Thai Key Prediction Using Tri-gram Model

#### 4.2.3 Error-correction Rules

From the errors of tri-gram key prediction when applied to the training corpus, about 12,000 error-correction rules are

extracted and then reduced to 1,500. These error-correction rules are applied to the result of key prediction. The results are shown in the table below.

	Prediction Accuracy from Training Corpus (%)	Prediction Accuracy from Test Corpus (%)
Tri-gram Prediction	93.11	92.21
Tri-gram Prediction + Error Correction	99.53	99.42

**Table 6: The According of Thai Key Prediction Using Tri-gram Model, and applying the Error-correction Rules to the Result of Tri-gram Model.**

## 5 Conclusion

In this paper, we have applied tri-gram model and error-correction rules for intelligent Thai key prediction and English-Thai language identification. The result of experiment shows a high accuracy of more than 99%. Through this system typing is much easier and enjoyable for Thais. This technique is expected to be able to apply to other Asian languages. Our future work is to apply the algorithm to mobile phones, handheld devices, inputting devices for disabled and multilingual input systems.

## Reference

- [1] Brill, E. (1997) Automatic Rule Acquisition for Spelling Correction. *ICML*.
- [2] Brill, E. (1993) A Corpus-Based Approach to Language Learning. *Ph.D. Dissertation*, University of Pennsylvania.
- [3] Cavnar, W. (1994) N-gram Based Text Categorization. *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, pp.161-169.
- [4] Cormen, T., Leiserson, C. and Rivest, R. (1990) *Introduction to Algorithms*, MIT Press
- [5] Kikui, G. (1998) Identifying the Coding System and Language of On-line Documents on the Internet. *Proceedings of the 16<sup>th</sup> International Conference on Computational Linguistics*, pp. 652-657.
- [6] Knuth, D., Morris J., and Pratt V. (1977) Fast pattern matching in strings. *SIAM Journal on Computing*. 6 (2), pp.323-350.
- [7] Sornlertlamvanich, V., Potipiti, T., and Charoenporn, T. (2000) Automatic Corpus-Based Thai Word Extraction with the C4.5 Machine Learning Algorithm. *The Proceedings of the 18<sup>th</sup> International Conference on Computational Linguistics*, pp. 802-807.

- [8] Zheng, F., Wu, J. and Wu, W. (2000) Input Chinese Sentences Using Digits. *The Proceedings of the 6<sup>th</sup> International Conference on Spoken Language Processing*, vol. 3, pp. 127-130