Towards an Intelligent Multilingual Keyboard System

Tanapong Potipiti, Virach Sornlertlamvanich, Kanokwut Thanadkran

National Electronics and Computer Technology Center,
National Science and Technology Development Agency,
Ministry of Science and Technology Environment,
22nd Floor Gypsum Metropolitan Tower 539/2 Sriayudhya Rd. Rajthevi Bangkok 10400 Thailand
Email: tanapong@nectec.or.th, virach@nectec.or.th, kanokwutt@notes.nectec.or.th

ABSTRACT

This paper proposes a practical approach employing n-gram models and error-correction rules for Thai key prediction and Thai-English language identification. The paper also proposes rule-reduction algorithm applying mutual information to reduce the error-correction rules. Our algorithm reported more than 99% accuracy in both language identification and key prediction.

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 users also have the same problem.

It will be wonderful, if there is a 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 trigram 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 is only one related work on inputting Chinese words through 0-9 numpad keys. [8] 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 lexical trees they employed required a large amount of space. Their algorithm is required some improvement for a practical use.

3 THE APPROACH

3.1 Overview

In the traditional Thai keyboard input system, a key button with the help of 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.

Table 1: A key button can represent different characters in different modes.

English Mode without Shift	English Mode with Shift	Thai Mode without Shift	Thai Mode with Shift
‘a’	‘A’	‘ฟ’	‘ฤ’

However, using NLP technique, the Thai-English keyboard system which can predict the key users intend to type without 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 successful result.

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

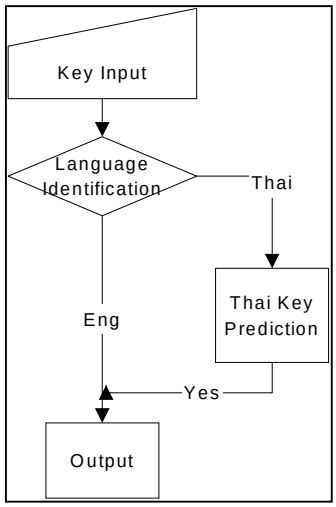

Figure 1: How the System Works

3.2 Language Identification

The following example illustrates the disturbance of language switching. In the Thai input mode, typing a word “language” will result “สฟิเฟเ”. It is certain that the user has to delete sequence “สฟิเฟเ” 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. The language-identification process starts when a non-space character is typed after a space. Many works in language identification, [3] and [5], have claimed that the n-gram model gives a high accuracy on language identification. After trying both trigrams and bigrams, we found that bigrams were superior. We then compare the following bigram probability of each language.

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

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

where

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

K is the key button considered.

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

Tprob is the probability of the considered key-button sequence to be Thai.

Eprob is the probability of the considered key-button sequence to be English.

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

The language being inputted is identified by comparing the key sequence probability. The language will be identified as Thai if *Tprob* > *Eprob* and vice versa.

3.3 Key Prediction without Using Shift Key for Thai Input

3.3.1 Trigram Key Prediction

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

$$\tau = \arg\max_{c_1, c_2 .., c_n} \prod_{i=1}^{n} p(c_i \mid c_{i-1}, c_{i-2}).p(K_i \mid c_i)$$

where

τ is the sequence of characters that maximizes the character string sequence probability,

c is the possible input character for the key button *K*,

K is the key button,

n is the length of the token considered.

3.3.2 Error Correction for Thai Key Prediction

In some cases of Thai character sequence, the trigram model fails to predict the correct key. To correct these errors, the error-correction rules proposed by [1] and [2] is employed.

3.3.2.1 Error-correction Rule Extraction

After applying trigram 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 "glik[lkl9in" is predicted as "เศรษฐฐาสตร์", where the correct prediction is "เศรษฐศาสตร์". The string "ik[lkl9" is then collected as an error sequence and "ษฐฐาสต" is collected as the correct pattern to amend the error.

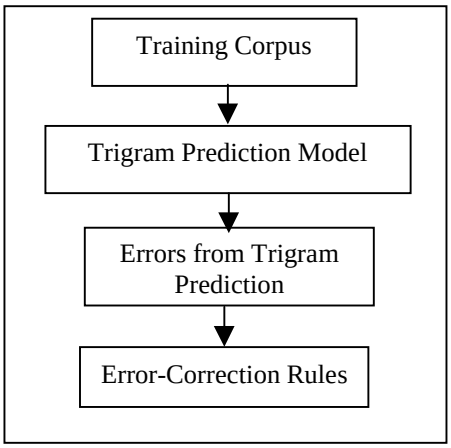

Figure 2: Error-Correction Rule Extraction

3.3.2.2 Rule 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.

Table 2: Error-Correction Rule Reduction

Pattern No.	Error Key Sequences	Correct Patterns
1.	k[lkl9	ษฐฐาสต
2.	mpklkl9	ทยาศาสต
3.	kkklkl9	าษาศาสต
4.	lkl9	ศาสต

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.

$$Lm(x\mathbf{y}z) = \frac{p(x\mathbf{y}z)}{p(x)\,p(\mathbf{y}z)},$$

$$Rm(x\mathbf{y}z) = \frac{p(x\mathbf{y}z)}{p(x\mathbf{y})\,p(z)},$$

where

*x***y***z* is the pattern being considered,

x is the leftmost character of *x***y***z*,

y is the middle substring of *x***y***z*,

z is the rightmost character of *x***y***z*,

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.

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] and [6]) is applied to search error sequences. We constructed an automaton for each pattern, then merge these automata into one as illustrated in Figure 3.

4. EXPERIMENTS

4.1 Language Identification

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 'ฟ', 'ๆ' and 'a' will be converted to 'a'. For the language identification, we employ the key-button bi-grams extracted As a conclusion the first 6 characters of the token are enough to yield a high accuracy on English-Thai language identification.

Table 3: The Accuracy of Thai-English 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

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 shift and unshift alphabets used in the corpora.

Table 4: Information on Alphabets Used in Corpus

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

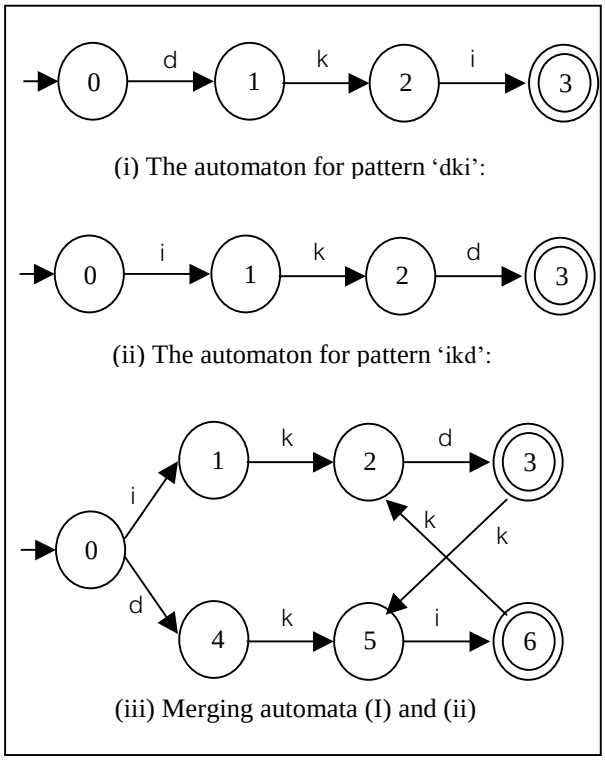

Figure 3: The Example of Constructing and Merging Automata

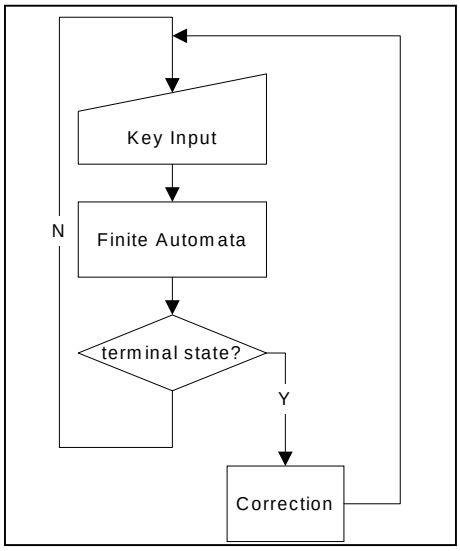

Figure 4: The Error-Correction Process

4.2.2 Thai Key Prediction with Trigram

Because the Thai language has no word boundary, we trained the trigram model from a 25-MB Thai corpus instead of a word list from a dictionary as in the language identification. The trigram 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 4.

Table 5: Thai Key Prediction Using Trigram Model

Training Corpus	Test Corpus
93.11	92.21

4.2.3 Error-correction Rules

From the errors of trigram 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.

Table 6: The Accuracy of Key Prediction Using Trigram Model and Applying Error-correction Rules

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

5 CONCLUSION

In this paper, we have applied trigram model and error-correction rules for intelligent Thai key prediction and English-Thai language identification. The result of the experiment shows the high accuracy of more than 99 percent accuracy, which is very impressive. Through this system typing is much more 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 and multilingual input systems.

REFERENCES

[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 16th 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 18th 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 6th International Conference on Spoken Language Processing*, vol. 3, pp. 127-130.