An Approach for Improving Thai Text Entry on Touch Screen Mobile Devices Based on Bivariate Normal Distribution and Probabilistic Language Model

Thitiya Phanchaipetch and Cholwich Nattee

School of Information, Computer, and Communication Technology,
Sirindhorn International Institute of Technology, Thammasat University
thitiya.phanchaipetch@student.siit.tu.ac.th, cholwich@siit.tu.ac.th

Abstract. This paper presents an approach to improve the correctness for Thai text inputting via virtual keyboard on touch screen mobile phones. The proposed approach is to generate candidate character based on statistical model from bivariate analysis of pre-collected coordinate data and apply the character trigram model to each candidate character sequence. From user's touch positions, a set of candidate characters with high position-based probability is generated. Then, the character trigram model is applied to each generated candidate characters sequence. For each character sequence, a probability is computed from the weighted combination of position-based and character trigram models. In the end, the character sequence with the highest probability is selected to be the most appropriate sequence. Experiments were conducted to compare the typing accuracy between an ordinary Thai virtual keyboard and our proposed algorithm using the same Thai keyboard layout. Results demonstrate that the proposed algorithm provides the improvement in the text entry accuracy in both character levels and word levels.

Keywords: virtual keyboard, thai text entry, touch screen mobile phone, touch screen mobile device, trigram model, thai keyboard, bivariate normal distribution.

1 Introduction

The majority of the worlds digital experiences now happen through mobile devices especially on touch screen mobile phones. One trend in consumer electronics is touch-sensitive screen that are controlled by human finger motions rather than a cursor or pointing device. Smartphone sales to end users were up 72.1% from 2009 and accounted for 19% of total mobile communications device sales in 2010.[1]

Virtual keyboard is a software component that allows a user to enter characters and can usually be operated with touch screen device where data entry is required a physical and keyboard is not available. Focusing on Thai language,

J.-S. Pan, S.-M. Chen, N.T. Nguyen (Eds.): ACIIDS 2012, Part III, LNAI 7198, pp. 187–196, 2012.

there are more alphabets in Thai than in English. The small area of touch screen mobile device causes the problem that each key on keyboard layout is located very close to each other and the size of each button becomes small. Even though every standard Thai keyboard is splitted into 2 sub-layouts which are shift sub-layout and non-shift sub-layout to cover all Thai consonants and vowels, it is still hard to accurately touch on the intended key button. Most human fingers are always larger than key buttons. Pressing on a small button with a large finger tends to generate error, especially when virtual keyboard has no tactile feedback. This problem is widely known and it decreases efficiency of Thai text entry on touch screen mobile phone.

This paper focuses on improving Thai key prediction for each press to reduce the typing error. Based on the assumption that each user has his/her own characteristic of typing. He/She will touch on particular position for the same key button. Each time, user's touch point is located closer to the set of points they have typed. Individual characteristic for each user can be formed. Bivariate Normal Distribution is assumed to be a statistical model for each particular key. From this statistical data, probability that user wants to type which key button given by the currently user touch point can be found. The right character should be the key which has the highest probability. We obtain the set of candidate keys which have high probability among all characters in one sub-layout. Another assumption in this paper is that most of words that user types are common words widely used in various articles. After obtaining the set of candidate keys, we concatenate all candidate keys to form candidate character sequence in every possible way. So, we grade the candidate sequence with probabilistic language model trained from a Thai corpus in order to find out the best candidate sequence with highest probability and display to user.

This paper consists of five parts. Background of this paper and other related works is explained in the next section. Bivariate Analysis-based Candidate Generation, Probabilistic Language Model for Assigning to candidate and Candidate Ranking are described in Section 3. Section 4 describes experiment setting and results and the last section is conclusion of the paper and our future direction.

2 Background and Related Works

There are many ways to improve performance of text entry on touch screen devices. Significantly changed in keyboard layout is one way to improve typing performance. Dvorak layout is well known as a QWERTY alternative, patented in 1936 by Dvorak and Dealey[7] The advantage of this layout is it uses less finger motion, increases typing rate and reduces errors compared to the standard QWERTY keyboard. MacKanzie, Zhang & Soukoreff (1999)[3] introduced a new virtual keyboard where the letters were laid in alphabetic ordering in two columns. The problem of new layout is it quite hard and takes time to make user get familiar with. This paper focuses on improving text entry base on standard keyboard layout, Thai Kedmanee which conforms to the Thai keyboard layout on Microsoft Windows to make user get more familiar with.

Research on improving input methods on standard keyboard layout has grown over recent year due to the limitation size of screen and lack of tactile input feedback. Potipiti et al.[5] introduced the approach by applied trigram model and error correction rules for intelligent Thai key prediction and English-Thai language identification. First technique is language identification which performs language switching automatically between Thai and English without pressing language-switch button. The second technique is Thai key prediction which applied character trigram probabilistic model and compare probability between Thai character sequence with shift key and without shift key. Trigram is well-known probabilistic language model that applied in this method. It proved that trigram is work for predicting the next character in such a text sequence. So, we try to apply this technique to our proposed method.

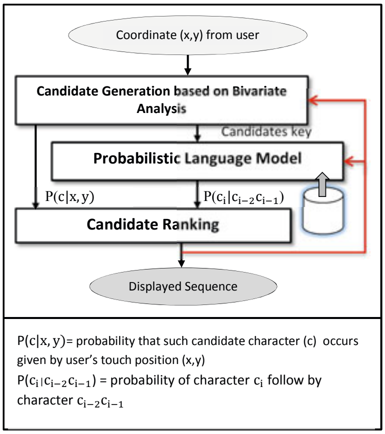
MacKanzie and Zhang[4] introduced Eye typing using word and letter predication in English language. User types by gazing at on-screen keyboard using eye tracker and software. Fixation algorithm is applied to determine which button on the keyboard receives eye-over highlighting. They considered these keys as candidate key. Word prediction was introduced by language model and a list of candidate word is produced as entry proceeds. The user selects the desired word. Even though these techniques can reduce the number of keystrokes per character of text entered and may increase text entry speed, it was sensitive to the correctness of the first several letters in a word and works only in letter prediction mode.

Janpinijrut et al.[2] introduced an approach for improving Thai text entry on standard Thai Keyboard based on distance and statistical language model. Touch area is defined by the area nearby the touch point. The characters which locate within touch area are considered as candidate characters. Each candidate characters are weighted based on their distances from the touch points. After that, the character trigram model is applied to select the combination of characters with the highest probability and suggest to the user.

After the experiments from this technique, we found that most of individual user generate the same typing error many times. This technique shows improvement in term of characters but it is not fit with variety of user typing characteristics because each touch point predicts character based on distance, not related to statistical analysis. The method that introduced in this paper is the alternative way to improve Thai text entry by consider individual typing characteristic and create adaptive user keyboard.

3 The Proposed Approach

The proposed approach consists of three main steps as shown in Figure 1, i.e., candidate generation based on bivariate analysis, probabilistic language model for assigning to candidate and candidate ranking.


Coordinate (x,y) from user
Candidate Generation based on Bivariate Analysis
Candidates key
Probabilistic Language Model
P(c|x,y)
P(c_i|c_{i−2}c_{i−1})
Candidate Ranking
Displayed Sequence

$P(c|x,y)$= probability that such candidate character (c) occurs given by user's touch position (x,y)

$P(c_i|c_{i-2}c_{i-1})$ = probability of character c_i follow by character $c_{i-2}c_{i-1}$

Fig. 1. Overall of proposed algorithm

3.1 Candidate Generation Based on Bivariate Analysis

From our assumption, suitable touch area should be the area that most people touch on. We assume each key has its own probability distribution as Bivariate Normal Distribution. In this case, we interested in the joint occurrence and distribution of values of the independent and dependent variable together. Bivariate Normal Distribution is joint distribution of two variables which are x and y. These values are considered to be the coordinates for a point geometry and can be obtained when user touches on screen. So, we find out which area those users mostly touch on each character. Coordinate, x and y are collected by shuffling all of Thai characters that exist on standard Thai keyboard. It composed of 43 characters per sub-layout. There are two sub-layouts which are shift and non-shift and each are mixed with alphabets, consonants and special characters. Both sub-layouts have the same layout. Therefore, it has 43 unique keys. 5 users are asked to type all 43 distinct characters 3 times in free style. After that, we obtained 15 coordinates of each character position. These coordinates are used through all step of proposed algorithm.

Figure 2 shows standard keyboard layout with distribution of 43 distinct character positions.

The formula (1) is known as Bivariate Normal Probability Density Function. Suppose we have two random variables, coordinate x and y. This formula can be integrated to obtain the probability that random coordinate x and y take a value in a given pre-collected coordinate data set for each character. Bivariate Normal Density Function is used to find the point of Normal Distribution curve:

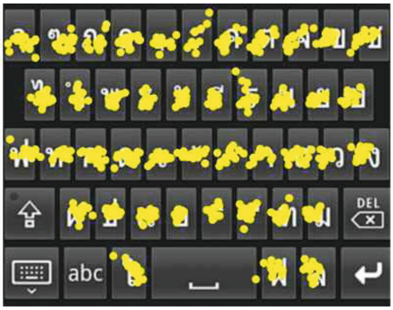


Fig. 2. Distribution of touch point of each character position for non-shift sub-layout

$$P_{pos}(c_i|p_i) = \frac{1}{2\pi\sqrt{\sigma_x\sigma_y(1-\rho_{xy}^2)}} \times exp\left\{-\frac{1}{2(1-\rho_{xy}^2)}\left[\left(\frac{p_{ix}-\mu_x}{\sqrt{\sigma_x}}\right)^2 + \left(\frac{p_{iy}-\mu_y}{\sqrt{\sigma_y}}\right)^2 \right.\right.$$
$$\left.\left. -2\rho_{xy}\left(\frac{p_{ix}-\mu_x}{\sqrt{\sigma_x}}\right)\left(\frac{p_{iy}-\mu_y}{\sqrt{\sigma_y}}\right)\right]\right\} \quad (1)$$

$P_{pos}(c_i|p_i)$ is probability that

candidate character c_i occurs given by user's touch position p_{ix} and p_{iy}.

It is involving with the individual parameter μ_x, μ_y, σ_x, σ_y and ρ_{xy} :

σ_x is covariance of a set of coordinate x belongs to that character,

σ_y is covariance of a set of coordinate x belongs to that character,

μ_x is mean of a set of coordinate x belongs to that character,

μ_y is mean of a set of coordinate y belongs to that character,

ρ_{xy} is ccorrelation coefficient between 2 values, coordinate x and y that belongs to that character.

Mean value can be calculated from below formula:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (2)$$

Covariance can be calculated from below formula:

$$Var(x) = \sum_{i=1}^{N}\frac{(x_i-\mu)^2}{N} \quad (3)$$

The correlation coefficient is computed as:

$$Corr(x,y) = \frac{\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y)}{\sqrt{\sum_{i=1}^{N}(x_i-\mu_x)^2\sum_{i=1}^{N}(y_i-\mu_y)^2}} \quad (4)$$

where

N is a number of data in one data set,
μ is mean of the data set.

When user touches on the screen to type a character, coordinate input x and y from user are calculated using the formula shown in Equation (1) with all 43 coordinates data set to find the probability of each character given by user coordinate input. After that, we can find a number of characters which have highest probability among 43 characters. In this paper, we conduct an experiment by select only top 3 characters to be candidate keys and process in the next step.

3.2 Probabilistic Language Model for Assigning to Candidate

An n-gram model is a type of probabilistic model for predicting the next item in such a sequence. This technique is applied to propose method to predict the next character given the previous character that user has typed before. In this paper, character trigram model is used in order to rank candidate key because mobile device has limited resource.

After generating candidate keys from previous step, each candidate concatenates all possible candidates to create all possible character sequences called candidate sequence. Then, we calculate probability for each candidate sequence by using character trigram model.

$$P_{lang}(c_1 c_2 \dots c_n) = \prod_{i=1}^{N} P(c_i | c_{i-2} c_{i-1}) \tag{5}$$

where $P_{lang}(c_1|c_2...c_n)$ is probability of each candidate sequence, $P(c_i|c_{i-2}c_{i-1})$ is probability of character c_i follow by character $c_{i-2}c_{i-1}$

From this point, we obtain a set of candidate sequences. Each is assigned by the probability from language model.

3.3 Candidate Ranking

The last step of proposed method is Candidate Ranking. From this point, each candidate sequence has its own probability given by position from 3.1 and given by language model from 3.2.

Candidate sequences are ranked by following scheme:

$$P(c_1 c_2 \dots c_n | p_1 p_2 \dots p_n) = \underset{c_1 c_2 \dots c_n}{\arg\max} \big(\alpha \cdot \sum_{i=1}^{n} P_{pos}(c_i|p_i) + \beta \cdot P_{lang}(c_1 c_2 \dots c_n)\big) \tag{6}$$

where

$P_{pos}(c_i|p_i)$ is probability that such candidate character c_i occurs given by user's touch position p_i which is calculated from section 3.1,

$P_{lang}(c_1c_2 \ldots c_n)$ is probability that such candidate sequence $c_1c_2 \ldots c_n$ occurs given by language model which is calculated from section 3.2 α and β are weight to counterbalance between character prediction by position and probabilistic language model. These two values should greater than 0 and sum of these two values is required to be exactly 1.0.

In section 5, we conduct experiment to find the most suitable value of α and β and the result shows that the best appropriate value of α is 0.6 and 0.4 for β.

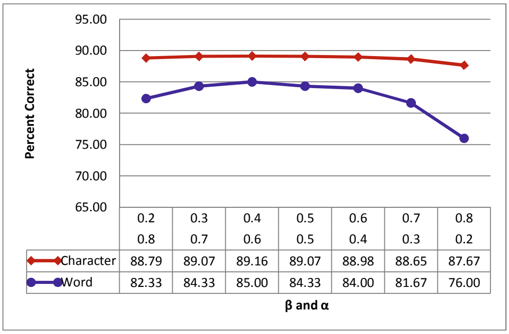
Each candidate sequence is processed by (6) scheme and rank. The candidate sequence which is in the first rank is suggested and displayed to user. This gives a result that every time user types a character, there is a chance that displayed sequence has changed to right sequence, even though they type wrong character. This is because ranking is processed every time user types a character. From assumption, the most of words users type are common words and widely used in various document. Even if users type position is not exactly on intended key button, the right character sequence should have highest probability in term of language consideration. Sometimes the right sequence is ranked to the top after processed and changes from wrong sequence to the right one.

4 Experiments and Results

In order to evaluate the performance of the proposed method, we conduct experiments based on collected coordinates set from user and compare the performance between proposed method and ordinary method.

There are many touch screen phones available in the market. Android is quickly becoming one of the most popular tools for mobile application development. The most prominent is Android's open-source nature. HTC Desire HD is capacitive touch screen Android phone that we use in our work. Its resolution is 480×800 pixels. Android 2.2.1 operating system is installed. CN Thai keyboard layout is chosen to be experiment keyboard. Experiment is fixing to portrait orientation because it has smaller keyboard area than landscape orientation. All experiments are implemented and tested on this phone.

For the experiment, we first find the most suitable value of α and β before we can evaluate the actual efficiency of proposed method. 5 million words from BEST 2010[6] Thai corpus created by NECTEC is used to train and create a character level 3-gram model. We randomly choose 30 words from the corpus to be test set. Each set consists of 10 words that occur with high, medium and low frequency in corpus as well as long, medium and short word length in the same average number. We let 10 users type all words in 3 test sets word by word in their style on mobile phone. Then we obtain 30 coordinates per user. Every coordinate is processed by the proposed algorithm on the mobile phone. We match these 30 output characters sequences against correct words to find the accuracy in character and word level. In character level, we match and count the number of correct characters in words. In word level, we match word by word.



	0.2 0.8	0.3 0.7	0.4 0.6	0.5 0.5	0.6 0.4	0.7 0.3	0.8 0.2
◆ Character	88.79	89.07	89.16	89.07	88.98	88.65	87.67
● Word	82.33	84.33	85.00	84.33	84.00	81.67	76.00

Fig. 3. Percent correctness in character and world level of various value of α and β

Figure 3 shows the accuracies at the different value of α and β in character and word level.

The graph shows that the correctness is increasing if α is decreased and β is increased. It obviously shows that probabilistic language model in section 3.2 takes a role in improvement. But when β is increased to 0.5, the correctness is continuing take a fall and dramatically dropped when β is 0.8. This is because of the lack of known word list. So, the unknown word can be the output if we set β to high value. We plan to apply dictionary to improve correctness in the future work. From this point, the most suitable values for and are 0.6 and 0.4 respectively which yield around 89.16% in character level and 85% in word level.

From the experimental results, the performance in word level is slightly dropped when we compare to character level because the method did not apply any statistics or information related to words. Character trigram model is used only in the proposed method.

After we obtained the most suitable values of α and β, 10 users are asked to type all 30 words 3 times. Five of them are the same users as we kept pre-collected coordinate data from them. The details had described in section 3.1.

Finally, we process coordinates from all users by using our proposed method and ordinary method to find the actual performance of proposed method against the ordinary typing method. The correctness of the ordinary method in character level is 96.65% and 80.67% in world level. While the correctness of our proposed method in character level is 98.91% and 93.78% in word level. This result shows that our method provides the improvement in the text entry accuracy on touch screen mobile phone for both character and word level. To ensure that the result of proposed algorithm is not better than ordinary algorithm *by chance*, we also conduct the paired t-test which assesses whether the means of two groups are statistically different from each other. The score associated with t-test is 1.69152×10^{-5} in word level and 2.69239×10^{-5} in character level. These values are less than 0.05. Therefore, there is a significant difference between of two groups.

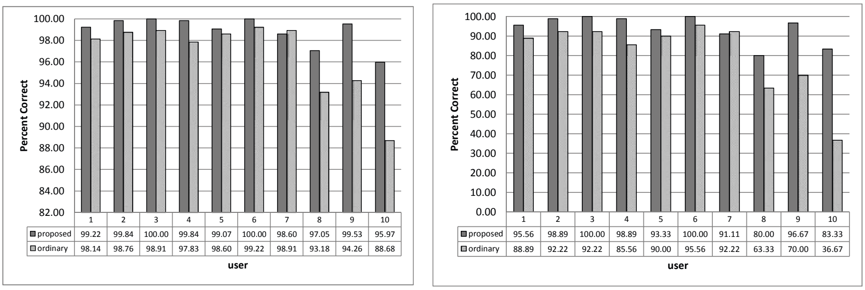


Fig. 4. Percent correctness in character and word level of 10 users by using two algorithms respectively

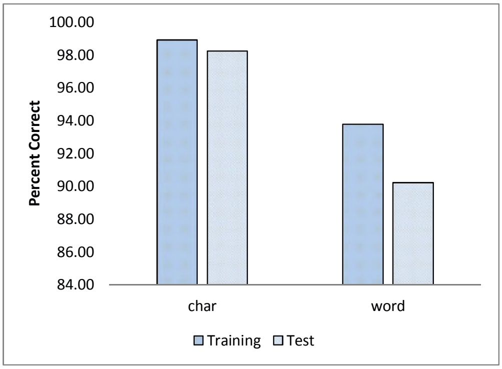


Fig. 5. Percent correctness in character and world level of various value of α and β

Moreover, based on the assumption that each user has their own characteristic of typing and can be formed to create distribution. If we focus on our algorithm by separate user into two groups, first group is five users that we kept pre-collected coordinate data from them (Training group) and second group; another five users (Test group), the performance for first group is better than second group in both character and word level. It ensures this assumption is correct because pre-collected coordinate that we obtained from first group is used to create bivariate normal distribution for each key. So, when user in training group types characters, the probability that such character occurs given by users touch position is probably higher than test group because the probability is based on distribution which generated from training group. Figure 4. shows percent of correctness of training group and test group in character level and world level.

4.1 Conclusion and Future Works

This paper has proposed a new approach to improve Thai text entry on touch screen mobile devices. Bivariate normal density on touch positions is used to generate candidate keys and then processes by using character trigram model. After that, all candidate sequences are ranked based on probability given by users coordinate and probability given by language model and suggested to user. We conduct experiment to find the best value of and to weight to counterbalance between character prediction by position and probabilistic language model. The most suitable value of α is 0.6 and β is 0.4. Our algorithm performs better than the ordinary method in both character level and word level. The correctness of the ordinary method in character level is 96.65% and 80.67% in world level. While the correctness of our proposed method in character level is 98.91% and 93.78% in word level.

For the future works, we plan to use list of frequently used words in order to improve performance of Probabilistic Language Model for Assigning to candidate sequence. Furthermore, we aim to do word completion and create user adaptive keyboard.

Acknowledgments. This research is financially supported by Thailand Advanced Institute of Science and Technology-Tokyo Institute of Technology (TAIST & Tokyo Tech), National Science and Technology Development Agency (NSTDA) Tokyo Institute of Technology (Tokyo Tech), Sirindhorn International Institute of Technology (SIIT), Thammasat University (TU) and Telecommunications Research and Industrial Development Institute (TRIDI).

References

1. Gartner, Inc.: Gartner Says Worldwide Mobile Device Sales to End Users Reached 1.6 Billion Units in 2010; Smartphone Sales Grew 72 Percent in 2010 (2010), http://www.gartner.com/it/page.jsp?id=1543014
2. Janpinijrut, S., Nattee, C., Kayasith, P.: An Approach for Improving Thai Text Entry on Touch Screen Mobile Phones Based on Distance and Statistical Language Model. In: Theeramunkong, T., Kunifuji, S., Sornlertlamvanich, V., Nattee, C. (eds.) KICSS 2010. LNCS, vol. 6746, pp. 44–55. Springer, Heidelberg (2011)
3. Mackenzie, I.S., Zhang, S.X., Soukoreff, R.W.: Text entry using soft keyboards. Behaviour and Information Technology 18, 235–244 (1999)
4. Mackenzie, I.S., Zhang, X.: Eye typing using word and letter prediction and a fixation algorithm
5. Potipiti, T., Sornlertlamvanich, V., Thanadkran, K.: Towards an intelligent multilingual keyboard system
6. Thailand National Electronics and Computer Technology Center (NECTEC): Benchmark for Enhancing the Standard of Thai language processing 2010, BEST 2010 (2010), http://www.hlt.nectec.or.th/best/?q=node/10
7. West, L.J.: The standard and dvorak keyboards revisited: Direct measures of speed. Tech. rep., Measures of Speed, Santa Fe Institute Working Papers Paper (1998)