Progress of Combining Trigram and Winnow in Thai OCR Error Correction

Surapant Meknavin
National Electronics and Computer Technology Center
73/1 Rama VI Road, Rajthevi, Bangkok, Thailand
surapan@nectec.or.th

Boonsert Kijsirikul, Ananlada Chotimongkol, and Cholwich Nuttee
Department of Computer Engineering
Chulalongkom University, Thailand
fengbks@chulkn.chula.ac.th

Abstract
From specific characteristics of Thai, Thai OCR errors frequently depend on nearby characters. To capture this characteristic of Thai OCR errors more appropriately, we propose the idea of using the varied n-gram of the character confusion probability for scoring approximately matched words. The value of n depends on characteristics of each character. For languages which have no explicit word boundary, word boundary ambiguity has to be resolved before correcting errors. In this paper, a maximal matching algorithm is used in stead of a more complicated word segmentation algorithm to reduce a time complexity problem. Finally, a hybrid method which combines a part-of-speech trigram model with Winnow algorithm is used to selected the most probable correction.

1. Introduction
Thai OCR error correction is very necessary because output from current Thai OCRs still have many errors, especially when the quality of input image is low. In the past several years, there were many research projects that involved a spelling correction problem [1]. However, the strategy for solving this problem is slightly different depending on a type of error and characteristics of each language.

The characteristics of Thai are quite different from those of other languages. In Thai, there is no word boundary delimiter between words which makes spelling correction much more complicated than languages which have explicit word boundary. Since information from words is needed for detecting and correcting errors, word boundary ambiguity also has to be resolved. Another characteristic is that there are three levels for placing Thai characters; the middle, the upper and the lower levels, and some characters can occupy more than one level. For example, in Figure 1 "θεριณ" consists of 8 characters {ι, θ, ρ, θ, θ, θ}. The character that occupies more than one level like ι usually connects to other characters in the upper or the lower level; in this case it connects to θ. In cases that the segmentation process of Thai OCR cannot separate the connected characters, it usually causes errors.

In this paper, we propose the idea for improving both quality and speed of Thai OCR error correction. Recently, a time complexity problem was solved by using a word segmentation algorithm to reduce a scope of correction to only dubious areas [2]. But this model is still complicated since error correction and word boundary disambiguation have to be done concurrently to assure that both tasks are correctly performed. For the task of error correction perfect segmentation is not actually needed. Therefore a simple word segmentation algorithm is sufficient to determine word boundaries in the input sentence before the process of error correction. The accuracy of correction can also be improved. From the problem of connected characters, the character confusion probability is frequently affected by nearby characters. So, to capture characteristics of Thai OCR errors more appropriately, the varied n-gram model of the character confusion probability is used in stead of a unigram model. For selecting the most appropriate corrected sentence, we use a hybrid method which combines a part-of-speech (POS) trigram model with a feature-based model called Winnow. Winnow is used in cases that POS trigram model cannot capture useful features for discriminating candidate words.

2. Characteristics of Thai
There are some characteristics of Thai that pose difficulty for correcting Thai OCR errors. First, Thai words are written consecutively without word boundary delimiters like Japanese and Chinese. For example, something like "I forgot to inform you" is written in Thai for "I forgot to inform you". This characteristic makes an error correction for languages that have no explicit word boundary much more complicated since a word segmentation problem also has to be concerned.

Another characteristic is that there are 3 levels for placing Thai characters; the middle, the upper and the lower levels, and some characters can occupy more than one level. From this characteristic, Thai characters occasionally connect to other characters which cause difficulty in the process of character segmentation of Thai OCR, and usually cause errors.
recognized as ә or ә, which is a completely different character, because the connected characters tend to have very different features from the original separated ones. Due to this problem, Thai OCR tends to merge connected characters into one which then causes characters in the upper or the lower level to be deleted.

upper level
topl ine
middle level
baseline
lower level

Figure 1: Three levels for placing Thai characters

3 Our Model

The problem of OCR error correction can be defined as: given the string of characters $S=c_1c_2\ldots c_n$ produced by OCR, find the word sequence $W=w_1w_2\ldots w_l$ that maximizes the probability $P(W|S)$.

3.1 Trigram Model

To find $W$ that maximizes $P(W|S)$, we use the following equation.

$$
\arg\max_{W} P(W|S) = \arg\max_{W} P(W)P(S|W) \tag{1}
$$

The probability $P(W)$ is given by the language model and can be estimated by the trigram model as:

$$
P(W) = P(W,T) = \prod_{i} P(t_i|t_{i-2},t_{i-1})P(w_i|t_i) \tag{2}
$$

$P(S|W)$ is the specific characteristics of each OCR, and can be estimated by using character confusion probability. The method used to model the character confusion probability is described in section 3.2.

The existing model for correcting Thai OCR errors is quite complicated since error correction and word boundary disambiguation have to be done at the same time [2]. To reduce a time complexity problem of this model, word boundary of each word in the input string is determined first. Then in the next step, only error correction of the segmented string will be focused. For the ordinary text which has no many word boundary ambiguities, the accuracy of a simple word segmentation algorithm is comparable to a more complicated one. Since its accuracy is sufficient for the task of error correction, we adopt a simple word segmentation algorithm called maximal-matching algorithm, which was described in [3], in our experiment. Our algorithm for correcting OCR errors is described below.

Algorithm for Correcting OCR Errors

1. Determine word boundaries in the input sentence: Determine word boundary of each word in the input sentence by using a maximal-matching algorithm. In this step, approximate boundaries of unknown strings can also be found.

2. Locate dubious areas:

   2.1 Unknown strings obtained from step 1 are dubious areas for non-word errors. However, these strings may not be complete non-word errors since some parts of the non-word errors can be words in a dictionary. For example, "inform at j on", $j$ is an unknown string, and inform at and on are words. In this case, known words around an unknown string have to be combined. So we will have "information" as a dubious area for a non-word error.

   2.2 Find all words $w_i$ whose local probabilities $P(w_{i-1},w_i,w_{i+1}|t_{i-2},t_{i-1},t_{i+1})$ are below a threshold from every real word produced by OCR. These words are hypothesized as real-word errors. Word boundaries of real-word errors may be incorrect because of errors in the input sentence. Therefore words around $w_i$ such as $w_{i-1}$ and $w_{i+1}$ are also combined for a dubious area.

3. Make hypotheses for non-word errors, real-word errors, and unknown words: Since dubious areas from step 2 have approximate boundaries, all of their substrings have to be considered.

   3.1 For correcting non-word and real-word errors, apply the candidate generation routine to generate approximately matched words within $k$ edit distances to every substring in dubious areas. The value of $k$ is varied proportionally to the length of the candidate word.

   3.2 Hypothesize all substrings in non-word error dubious areas as unknown words, except for ones that are not possible in Thai such as ones that are led by non-leading character, i.e. ә or ә.

4. Select the best corrected string: Select the best word sequences according to equation (1). For unknown words, $P(w_i|Unknown word)$ is computed by using the unknown word model described in [4].

3.2 Varied N-gram Character Confusion Probability

To reflect characteristics of Thai OCR errors more appropriately, the modification of the standard edit distance is used in the candidate generation routine to generate approximately matched words for correcting OCR errors. It is slightly different from the modified edit distance in [2]. In this paper, the modified edit distance allows a candidate word to insert arbitrary number of upper or lower level characters, and also allows a candidate word to substitute arbitrary number of upper or lower level characters with characters in the same level. In the middle level, it allows only $k$ substitutions.
and/or k insertions, but does not allow any deletions. This is because Thai OCR occasionally deletes middle level characters, but rarely inserts them. For example, applying the candidate generation routine with 1 edit distance to the string “ไห” gives the following set of candidates {คว, คุ, คิ, คิ, คื, คึ, คุ, คิ, คุ, คิ, คิ}.

The candidate generation routine may generate a large number of words with 1 edit distance for a short word since there are a lot of short words in Thai. So, the confusion probability, \( P(S|W) \) in equation (1), can be used to rank candidate words according to the similarity between them and the considered word. Candidates which have lower scores are eliminated from the list.

Errors from Thai OCR are frequently caused by connected characters consequently, the confusion probability of each character may be affected by the previous or the next character. From this characteristic, a trigram model can be used to calculate confusion probability, but it might be irrelevant in some cases. Therefore, a varied n-gram model is defined to use instead. Type and value of \( n \) depend on the characteristic of each character which quite relates to the level that it occupies and can be grouped into 6 groups, as shown in Table 1. This is quite similar to the types of single characters described in [5].

The character confusion probability can be estimated by collecting statistical information from the original text and the corresponding OCR output. Given the original word sequence \( W \) composed of characters \( v_1, v_2, \ldots, v_m \), OCR produces sequence of characters \( S = (c_1, c_2, \ldots, c_n) \) by repeatedly applying the following operation: substitute a character with another, insert a character, or delete a character. Let \( S_i \) be the i-prefix of \( S \) that is formed by the first character to the i\(^{th}\)-character of \( S = (c_1, c_2, \ldots, c_n) \), and similarly \( W_j \) is the j-prefix of \( W = (v_1, v_2, \ldots, v_j) \). Using dynamic programming technique, we can calculate \( P(S|W) = P(S_n|W_m) \) by the following equation:

\[
P(S_i|W_j) = \max\{P(S_{i-1}|W_{j-1}) \cdot P(\text{ins}(c|\text{context})), \)
\[
P(S_{i-1}|W_{j-1}) \cdot P(\text{del}(v|\text{context})),
\]
\[
P(S_{i-1}|W_{j-1}) \cdot P(c|v, \text{context})\}
\]

where \( P(\text{ins}(c|\text{context})), P(\text{del}(v|\text{context})) \), and \( P(c|v, \text{context}) \) are conditional probabilities that letter \( c \) is inserted given the context, letter \( v \) is deleted and letter \( v \) is substituted with \( c \) given that letter \( v \) and the context are concurrently occurred, respectively. These probabilities can be calculated by the following equations which similar to ones that are described in [6].

Context is a varied n-gram of characters defined in Table 1, and is also grouped into 6 groups to avoid a sparse data problem.

\[
P(\text{ins}(c|\text{context})) = \frac{\text{num}(\text{ins}(c) \cap \text{context})}{\text{num}(\text{context})}
\]
\[
P(\text{del}(v|\text{context})) = \frac{\text{num}(\text{del}(v) \cap \text{v} \cap \text{context})}{\text{num}(\text{v} \cap \text{context})}
\]
\[
P(c|v, \text{context}) = \frac{\text{num}(c \cap v \cap \text{context})}{\text{num}(v \cap \text{context})}
\]

3.3 Winnow Algorithm

After applying a trigram model to correct OCR errors, some real-word errors still remain. Information from a trigram model is not sufficient for detecting and correcting real-word errors since contexts are needed for dealing with this kind of error. Several previous works demonstrate that feature-based approaches are very effective for correcting real-word errors [2, 7, 8, 9]. For our task, we adopt a feature-based algorithm called Winnow that has been shown to be the best performer in English context-sensitive spelling correction [7], and also in Thai OCR error correction [2]. Winnow is a multiplicative weight updating and incremental algorithm which can combine several kinds of features in the context to select the most appropriate word from a

| Group | Level Occupied | Features Used | Example |
|-------|----------------|---------------|---------|
| 1     | Middle         |               | k → ต |
| 2     | Upper          | X X           | ต → ร, ร → ต |
| 3     | Lower          | X             | ร → ร |
| 4     | Upper Middle1  |               | ส → ร |
| 5     | Upper Middle2  | X             | ร → ร |
| 6     | Lower Middle   | X             | ร → ร |

(1) This sequence of character can be written according to Thai writing system, but does not actually occur.
A Winnow algorithm used in our experiment is the algorithm described in [10].

Following previous work [2,9], we have Winnow look at two types of features: context words and collocations to determine the most appropriate word from a k-modified edit distance confusion set. A k-modified edit distance confusion set $S = \{c, w_1, w_2, ..., w_n\}$ is composed of one centroid word $c$ which is a real word produced by OCR and words $w_1, w_2, ..., w_n$ generated by applying the candidate generation routine with maximum k modified edit distance to the centroid word. The value of $k$ is varied proportionally to the length of the centroid word.

To correct the remaining real-word errors, every real word produced by OCR except for ones that have been changed by a trigram model is evaluated by Winnow. This condition is used to avoid an excessive correction. After the most probable word in the confusion set is determined, the confidence level of that word is calculated. Confidence level of any word can be defined as the summation of all weights from specialists that vote for that word divided by the summation of all weights in the network. Finally, the most probable corrected sentence is selected based on the average of the confidence levels of all words in the sentence.

### 4. Experiments

Our experiment is based on a corpus which contains about 9,000 sentences (140,000 words, 1,300,000 characters). About 80% of the whole corpus is used as a training set and the rest is used as a test set. The results are shown in Table 2.

Table 2: The accuracy of correction after applying a unigram and a varied n-gram character confusion probability

| Types of Errors      | Unigram | Varied n-gram |
|----------------------|---------|---------------|
| Non-word Error       | 90.35%  | 91.15%        |
| Real-word Error      | 87.73%  | 88.06%        |
| Introduced Error     | 6.42%   | 4.54%         |

After applying a varied n-gram character confusion probability, the accuracy of correction is slightly improved comparing with a unigram model. The percentage of introduced errors is quite high since we selected sentences which contain many unknown words for testing our models. By analyzing the output of both algorithms, a varied n-gram model can deal with unknown words more effectively. A varied n-gram model can preserve many unknown words while a unigram model changes them to some closed known words. In case that both of them make a mistake, a varied n-gram model tends to produce output which is more similar to the correct one. This is because a varied n-gram model is more conservative; it looks at the context first before making any changes. For example, Thai OCR frequently deletes " when it connects to upper level character, such as "., or upper middle level character, such as ဇ. For a varied n-gram model the probability for inserting " to an unknown string is high when upper or upper middle level character is found in a nearby context. But for unigram model, this probability is high for every context which causes it to make mistakes in some cases.

### 5. Conclusions

We have applied a varied n-gram confusion probability, modified edit distance, POS trigram model and Winnow to the task of Thai OCR error correction. From our experiment, we found that both quality and speed of correction are improved. Using maximal matching algorithm to determine approximate word boundaries in the input sentence can reduce a number of hypotheses. A varied n-gram model is quite efficient for correcting errors which are caused by connected characters. It also deals with unknown words effectively.

### References

[1] K. Kakich, "Techniques for automatically correction words in text"; ACM Computing Surveys, vol. 24, no. 4, 1992.

[2] S. Meknavin, B. Kijsirikul, A. Chotimongkol, and C. Nuttee, "Combining Trigram and Winnow in Thai OCR Error Correction", In Proceedings of COLING-98, 1998.

[3] S. Meknavin, P. Charoenpornsawat, and B. Kijsirikul, "Feature-based Thai Word Segmentation", In Proceedings of NLRPS’97, 1997.

[4] M. Nagata, "Context-Based Spelling Correction for Japanese OCR", In Proceedings of COLING-96, 1996.

[5] W. Panich, S. Jitapunkul, and P. Choruengwiwat, "Segmentation of Connected Characters Using Distinctive Features of Thai Characters in Thai Character Recognition System, In Proceedings of NLRPS’97.

[6] X. Tong and D. A. Evans, "A Statistical Approach to Automatic OCR Error Correction in Context", In Proceedings of the Fourth Workshop on Very Large Corpora, 1996.

[7] A. R. Golding, and D. Roth, “Applying Winnow to context-sensitive spelling correction”, In Lorenzo Saitta, editor, Machine Learning:Proc. of the 13th International Conference, Bari, Italy, 1996.

[8] A. R. Golding, and Y. Schabes, "Combining Trigram-based and Feature-based Methods for Context-Sensitive Spelling Correction", Technical Report TR-93-03a, Mitsubishi Electric Research Laboratory, 1996.

[9] A. R. Golding, "A Bayesian hybrid Method for context-sensitive spelling correction", In Proceedings of the Third Workshop on Very Large Corpora, 1995.

[10] A. Blum, "Empirical Support for Winnow and Weighted-Majority Algorithm: Results on a Calendar Scheduling Domain", Machine Learning, vol. 26, 1997.