# Statistically trained orthographic to sound models for Thai

*Ananlada Chotimongkol and Alan W Black*

Language Technologies Institute
Carnegie Mellon University
ananlada@.cs.cmu.edu, awb@cs.cmu.edu

## ABSTRACT

Many languages have a non-obvious, but not unrelated, relationship between orthography and pronunciation. Traditional methods for automatic conversion from letters to phones involve hand-crafted letter-to-sound rules, but these require care and expertise to develop. This paper presents a letter-to-sound rule system for Thai, that is trained automatically from lexicons. A statistical model, decision trees, is used to predict phones from letters. Letters mappping to multi-phones are used to solve the problem of implicit vowels and final consonants propagation and pre- and post-processing techniques are used to handle the inversion of initial consonants and vowels. For tone prediction, hand-crafted rules are used instead since there is no ambiguation if the phonological composition is known. Combining the n-gram of phone model with the decision trees, we can achieve 68.76% word accuracy which is better than 65.15% word accuracy in the rule-based approach.

## 1. INTRODUCTION

Finding the pronunciation of a word is important in both text-to-speech and automatic speech recognition systems. Although lexicons are the most reliable method for many languages, they will never be complete due to neologisms, proper names, morphological variants etc. Thus a fallback position is required that predicts the pronunciation from the orthographic form.

For Thai, like many other languages, this problem is not trivial due to a weak relationship between letters and phones. Some letters produce different sounds in different context. For example “h” can be pronounced as /h/ in “hat” or can become silent in “hour”. In Thai, “ท” can be pronounced as /th/ in “มณโฑ” or /d/ as in “มณฑป”. The pronunciation of the letter in Thai is based on its phonological composition in the syllable (initial consonant, vowel and final consonant). Traditional letter-to-sound modules usually consist of two stages. First, the input word is divided into a sequence of syllables by matching the string of letter against hand-crafted rules of all possible syllabic structures. After that, each phonological composition is mapped to phone, and the rules for derived tone are then applied. This technique is used in [5] and [6] for Thai speech synthesis, and also in [2] and [4] for a similar problem romanization and soundex system respectively.

The problem of the rule based approach is in the first step, syllabification. There are many cases where more than one rule can be applied to the input string, due to the implicit vowels and final consonant propagation. In order to disambiguate between these rules, extensive linguistic knowledge must be used. However, the complicated rules may not be practical in the real system and also consume a lot of time and prone to error. The conventional approaches usually use context independent rules and resolve the ambiguity by always choosing the longest rule that match or the most frequently used one. This might cause the system to make the mistake in the case that the context is needed. Another technique is to generate all possible syllable sequences and then use the probabilistic model to choose the most probable one [2]. However, the number of all possible syllable sequences is very large for some words. Therefore, a searching technique must be used to reduce the search space.

Statistical modeling has been successfully applied to the problem of letter-to-sound rules in many languages such as English, French and German [1]. This approach can eliminate the time consuming step of rule writing, so the training process can be done almost fully automatically. Based on the techniques in [1], we use decision trees to predict phones based on letters and their context. However, some augmentations need to be done when we apply it for Thai. In those other languages, letters map to epsilon, a phone or occasionally two phones. For example, in English letters map to epsilon in consonant clusters and map to two phones for some letter such as “x”, which often be pronounced as the phone combination of /k/ and /s/ as in “box”. In Thai we found that letters may map to many more phones for implicit vowels and final consonants propagation. Another problem in aligning letters to phones is inversion of initial consonants and vowels, which occurs in syllables with leading vowels. To handle tones, hand-written rules are used instead of the decision trees. If phonological composition and a tone marker are known, applying the rules for predicting tone is straightforward. The decision tree makes a prediction based on the letter context alone. So we take into account the phone context by using the n-gram of phone model on the possible phone groups generated by decision trees.

## 2. PROBLEM IN THAI LETTER-TO-SOUND RULES

Mapping from letters to sounds in Thai is not a trivial problem since the relationship between letters and sounds is not one-to-one. The summary of Thai alphabet and phones in table 1 is good evidence of this weak relationship. The numbers are based on Thai pronunciation dictionary “LEXITRON” and [5].

| Alphabet | Phones |
|---|---|
| 44 consonants (42 currently in use) | 21 initial consonants |
| | 17 consonant clusters |
| | 8 final consonants |
| 19 vowels (in term of ASCII character) | 24 unique vowels (excluding ones that have implicit final consonants) |
| 4 tones | 5 tones |
| 1 silent marker | - |

**Table 1:** Summary of number of alphabet and phones in Thai.

Like English, multi-letter to phone mapping is needed for consonant clusters and multi-letter vowels, this is achieved through mapping some letters to epsilon. However, there are some characteristics of Thai pronunciation system that poses more difficulty in letter-to-sound rules.

1) Some letters can be pronounced differently depending on its phonological composition in the syllable. For instance, "ร" is pronounced as /r/ when it functions as an initial consonant, but is pronounced as /n/ when it functions as a final consonant. It can also be pronounced as /a/ when it is combined with another "ร" and functions as a vowel.

2) Some vowels can be pronounced implicitly without having any written forms. For instance, "วรพล" /w-@:-0|r-a-3|ph-o-n-0/[1], a single letter "ว" is pronounced as one syllable /w-@:-0/ with an implicit vowel /@:/ while a single letter "ร" is pronounce as /r-a-3/ with an implicit vowel /a/. The third syllable "พล" also has an implicit vowel /o/.

3) Final consonants can be propagated to be initial consonants of the following syllables. For example, "จัตุรัส" /c-a-t-1|t-u-1|r-a-t-1/, "ต" functions as both a final consonant, /t/, of the first syllable and an initial consonant, /t/, of the second syllable.

4) The position of the initial consonant and vowel are inverted in a syllable with leading vowel. For instance, "เก" /k-e-0/, "เ" is the one that has /e/ sound and "ก" is the one that has /k/ sound. This is compounded when a leading vowel is followed with multiple initial consonants as phone order may be inverted across a syllable boundary. For example, the word "เกษม", /k-a-1|s-e-m-4|/, has two initial consonants "ก" and "ษ". 'เ' is pronounced as /e/ in the second syllable while "ก" is pronounced as /k-a-1/ in the first syllable.

# 3. STATISTICAL TRAINED MODEL

## 3.1 Training Decision Trees

In our statistical trained model, we use decision trees to predict phones from letters and their context. Decision trees are trained from a lexicon of pronunciations. Before training, we must align each letter to its corresponding zero or more phones. Then from these alignments, a decision tree is trained for each letter. The detail of the training process is given below.

1. **Define the set of allowable paring of letters to phones.**
   In this step, we define all possible phones and multi-phones for each letter. From the first problem in the previous section, /r/, /n/ and /a/ are considered as allowable phones for a letter 'ร'. Letters map to epsilon is needed for consonant clusters and multi-letter vowels. For example, 'ป' in 'ปร' is mapped to /pr/ while 'ร' is mapped to epsilon. All tonal makers are mapped to epsilon since we will use rules to predict tones after we predict all other phones. The detail of tone prediction is described in section 3.3.

   Mapping letter to multi-phones is used to solve implicit vowels and consonants propagation. A syllable boundary marker is also considered as a phone, as we use it to predict a tone in the next stage. For the case of implicit vowel, consider the word "วรพล" /w-@:-|-r-a-|-p-o-n-|/[2], "ว" is mapped to three phones /w-@:-|/ and "ร" is mapped to /r-a-|/. For the last syllable, "พ" is mapped to two phones /o-ph/. The reason for the reverse order of the phone will be described in the next step. For the case of final consonant propagation, considered the word "จัตุรัส" /c-a-t-|-t-u-|-r-a-t-|/, "ต" which functions as both final consonant and initial consonant of the following syllable is mapped to three phones /t-|-t /. We need one letter to map up to five phones when the final consonant that propagates also has implicit vowel. For example, "วิทยา" /w-i-t-|-t-a-|-j-a:-|/, "ท" is mapped to /t-|-t-a-|/, since the final consonant of the first syllable propagates to become the initial consonant of the second syllable which also has implicit /a/ vowel.

2. **Preprocess the lexicon.**
   Before aligning the letters to phones, some preprocessing is needed. From the inversion of the initial consonant and the leading vowel, we need to invert the order of the phones in corresponding to the letters that generate them. For instance, the pronunciation of "เก" is changed to /e-k-|/. The first letter is easily mapped to the first sound and so is the second letter. For the implicit /o/ vowel, since /o/ is also a leading vowel, it is placed before the initial consonant to be consistent. Therefore, "พ" with implicit /o/ vowel is mapped to /o-ph/.

3. **Align letters to zero or more phones.**
   We find all possible alignments given the set of allowable letter/phone group mappings and calculate the likelihood of each. Then we use that information to score all possible alignments and select the most likely as the actual alignment.

4. **Build decision trees**
   The CART technique is use to build decision trees. Up to three letters preceding and following the considered letter and word boundary markers are used as context for predicting phones. A decision tree is trained for each letter. Figure 1 shows a sample decision[3] tree for "ร" which has three allowable phones, /a/, /n/ and /r/.

[1] "-" and "|" indicate a phone boundary and a syllable boundary respectively. The numbers in the pronunciation represent the tones.

[2] All tones are dropped from the pronunciation since we will use rules to predict them instead of decision trees.

[3] The real decision tree is much more complex than the sample tree.

```
((p.name is า)
 ((n.name is #)
  (((a 0) (n 1) (r 0) n)))
 ((n.name is ◌ั)
  (((a 0) (n 0) (r 1) r)))
((p.name is ก)
 ((n.name is ร)
  ((n.n.name is ม)
   (((a 0.672) (n 0.251) (r 0.077) n))))))
```

**Figure 1:** Decision tree for "ร"

"p.name" means previous letter while "n.name" and "n.n.name" mean next letter and next next letter respectively. The first rule can be read as following: If the previous character is "า" and the next character is a word boundary marker, then "ร" should be pronounced as /n/ with probability equal to 1 and should be pronounced as /a/ and /r/ with probability equal to 0. So given this context "ร" should be pronounced as /n/.

## 3.2 Run-time Model

At run time we need to generate the string of phones from the model. From the training, we have one decision tree for each letter in the alphabet. Thus we take the unknown word, split it into a list of letter and apply the appropriate tree to each letter. The result is a string of phone groups (possibly including epsilon). This string is then processed to remove epsilons and split the multi-phone groups giving a string of phones plus syllable boundaries. After that, the corresponding phones of the initial consonant and the leading vowel are reverted back to their ordinary position in the pronunciation.

## 3.3 Predicting Tones

For predicting tones, hand-written rules are used as no ambiguity exists, if the following components of each syllable are known; Initial consonant group (high, middle and low), The length of the vowel, Final consonant and A tone marker. The detail of the rules can be found in [7] and also in most of the Thai grammar books. The rules can cover most of the cases, except for the following 2 cases.

1) A modified form of some words that still uses the same tone as the original word instead of the tone which correspond to the new form. For example, "กำราบ" /k-a-m-0|r-a:-p-**1**|/ is modified from "กราบ" /kr-a:-p-**1**|/. So "ราบ" in "กำราบ" has a low tone (represented by "1") as in "กราบ" instead of a falling tone (represented by "2") as when it is alone, "ราบ" / r-a:-p-**2**/.

2) For some loan words from Pali and Sanskrit, the tone in a syllable may be influenced by the previous syllable, which make it different from the tone predicted from the written form of that syllable.

# 4. EXPERIMENT

Our training set consists of 22,818 words from a Thai pronunciation dictionary, LEXITRON, with 2,535 words, which is every tenth word in the dictionary, held-out for testing.

| Model | Letter Accuracy | Word Accuracy | | |
|---|---|---|---|---|
| | | Exp1 | Exp2 | Exp3 |
| Rule-based | - | 68.23% | 83.39% | 65.15% |
| Decision Tree | 94.47% | 68.71% | 79.21% | 62.17% |
| Decision Tree (without syllable boundaries) | 95.25% | 72.47% | 83.50% | 66.26% |

**Table 2:** Accuracy of the models. Exp1 is the word accuracy ignoring tones. Exp2 is word accuracy ignoring tones and the length of the vowels. Exp3 is word accuracy including tones and the length of the vowels.

In our initial model, we achieved 94.47% letter accuracy and 62.17% complete word accuracy. Letter accuracy is defined as the number of letters that are correctly converted to epsilon, a phone, or multi-phone. Word accuracy is defined as the number of words that all phones (once resplit) match exactly all the phones in the entry in the test data.

As many of the errors were caused by misplaced syllable boundaries, and assuming we can predict syllable boundaries from a sequences of phones, we retrained without them. This improved our scores to 95.25% letter accuracy and 66.26% word accuracy. The second model out-performs a previously existing rule-based model, which achieves 65.15% word accuracy on the same data.

The accuracy drops more in our model than the rule-based model when including the tone. This is due to the inconsistent of the length of the compound vowels in the dictionary. Some compound vowels are coded as short vowels, but the are intended to be pronounced as long vowels as reflect in tones. Even we can get the same vowel length as in the dictionary, we cannot get the same tone since it does not follow the rules.

Another frequent mistake is the length of the vowels. However, sometimes it is negligible since in some words both pronunciations are actually acceptable. The length of the vowels is sometimes depended on the context and also varies from dialects to dialects. For example, "ไห้" in "ร้องไห้" /r-@:-ng-3| **h-a:-j-2**|/ is pronounced using a long vowel, /a:/, while in "เสาไห้" /s-a-w-4| **h-a-j-2**|/ is pronounced using a short vowel, /a/. Since we can ignore the variation in the length of the vowels as long as it does not change to meaning of the word, the word accuracy is higher about 10% on both models.

The above simple model uses decision trees to predict the most likely phone group based on the letter context alone. No account of previous (or following) phone predictions is taken into account. In an attempt to improve prediction, we built a second more complex model following [3] where the decision trees predict a probability distribution of possible phone groups and best path is selected using Viterbi search and n-grams. More formally we wish to find most probable string of phone groups given a string of letters

$$\text{argmax } P(p_1,...p_n \mid l_1, ... l_n) \quad (1)$$

We can approximate this as

$$\text{argmax} \prod_{i=1}^{N} P(p_i \mid p_{i-1}, p_{i-2}, ...) P(l_i) \quad (2)$$

The $P(l_i)$ term in (2) is approximated unigram, the results from the decision tree, which actually gives $P(p \mid l_{i-3},...l_{i+3})$ but can be inverted by dividing by the unigram probability of p. P(p) is approximated by an n-gram. We tried varies n-gram, with Good-Turing smoothing (sm1) and backoff (bk) and found the following results.

| Experiment | Letter Accuracy | Word Accuracy | | |
|---|---|---|---|---|
| | | Exp1 | Exp2 | Exp3 |
| No ngram | 94.47% | 68.71% | 79.21% | 62.17% |
| Unigram (raw) | 94.71% | 70.29% | 83.23% | 65.25% |
| Unigram (sm1) | 94.71% | 70.29% | 83.23% | 65.25% |
| Unigram (bk) | 94.71% | 70.29% | 83.23% | 65.25% |
| Bigram (raw) | 95.37% | 74.79% | 86.00% | 67.53% |
| Bigram (sm1) | 95.40% | 74.79% | 86.04% | 67.53% |
| Bigram (bk) | 95.33% | 74.04% | 85.76% | 67.22% |
| Trigram (raw) | 95.16% | 73.33% | 85.64% | 67.14% |
| Trigram (sm1) | 95.08% | 73.17% | 85.92% | 67.18% |
| Trigram (bk) | 95.60% | 75.31% | 86.94% | 68.76% |

**Table 3:** Letter accuracy and word accuracy of the n-gram models. Exp1, Exp2 and Exp3 are the same as in Table 2.

When we use equi-probable phones we get the same result as using the best predicted value from the decision tree. But we get an immediate gain with unigrams with appropriate probabilities, this gets substantially better for bigrams, though reduces for trigrams. As we increase the size of the context the n-gram representation gets larger such that even for trigrams we have a model which requires hundred of megabytes of memory to run. Even if we had more efficient representations of n-grams, it seems that bigrams offer the best compromise between accuracy and space.

Interestingly when we apply the tree plus n-gram model using the trees that do not predict syllable boundaries our overall word accuracy on bi-gram and tri-grams drops some 1%. This implies that the syllable boundary information is in fact useful in the prediction.

# 5. CONCLUSION

We have presented a statistically trained model, decision tree, for Thai letter to sound rules, with augmentation for solving letter to phone alignment problem in Thai. The results reveal that the decision trees with n-gram of phones model out-performs the traditional rule-based model. We also found that the bigrams model offers the best compromise between accuracy and space.

There is some more work that can be done to improve the model. The problem which is still left unsolved is the inversion of phones across syllable boundaries. One way to handle this problem is to revert the position of the letters to make them correspond to the pronunciation, in the preprocessing process. The word "เกษม" /k-a-1|s-e-m-4|/, will become "กเษม" which "ก" is now easily mapped to the first syllable and so as "เษม" to the second syllable. However, this will eliminate another possible pronunciation /k-e-0|s-o-m-4|/, which "เก" go to the first syllable and "ษม" go to second syllable, that needs the original written form. Giving both written forms to the model, the normalized probabilities from the decision trees can be used to select the most probable one.

Using the decision tree to predict syllable boundary instead of phone is also worth a try. If a syllable boundary is obtained, the pronunciation of most syllables can be predicted accurately by using a rule.

# 6. ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Virach Sornlertlamvanich, the head of Software and Language Engineering Laboratory, National Electronics and Computer Technology Center, Thailand, for the Thai pronunciation dictionary, 'LEXITRON' and the rule-based letter-to sound rule. We also would like to thank Monthika Boriboon for the linguistic discussion.

This work was supported in part by an NSF Combined Research and Curriculum Development grant, and by Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The opinions expressed in this work do not necessarily represent the opinions of these funding bodies.

# 7. REFERENCES

[1] Black, A., Lenzo, K. and Pagel, V. "Issues in Building General Letter to Sound Rules", *3rd ESCA Speech Synthesis Workshop*, pp. 77-80, Jenolan Caves, Australia, 1998.

[2] Charoenporn, T., Chotimongkol, A., and Sornlertlamvanich, V. "Automatic Romanization for Thai", in *Proceeding of the 2 nd International Workshop on East-Asian Language Resources and Evaluation"*, Taipei, Taiwan, 1999.

[3] Jiang, L., Hon, H. and Huang, X. "Improvements on a Trainable Letter-to-Sound Converter", *Eurospeech 97*, Rhodes, Greece, 1997.

[4] Karoonboonyanan, T., Sornlertlamvanich, V. and Meknavin, S. "A Thai Soundex System for Spelling Correction", *Proceeding of the National Language Processing Pacific Rim Symposium*, pp. 633-636, Phuket, Thailand, 1997.

[5] Mittrapiyanuruk, P., Hansakunbuntheung, C., Tesprasit, V. and Sornlertlamvanich, V. "Issues in Thai Text-to-Speech Synthesis: The NECTEC Approach", *Proceedings of NECTEC Annual Conference 2000*, Bangkok Thailand, 2000.

[6] Taisetwatkul, S., Kanawaree, W. "Thai Speech Synthesizer" (in Thai), *Senior Project for completion of B. Eng.,* Faculty of Engineering, Chulalongkorn University, Bangkok Thailand, 1996.

[7] Thonglo, K., *Thai Grammar* (in Thai), Bangkok, Thailand 1952.