

AUTOMATIC ROMANIZATION FOR THAI

Thatsanee Charoenporn, Ananlada Chotimongkol, and Virach Sornlertlamvanich

Software and Language Engineering Laboratory
National Electronics and Computer Technology Center
Gypsum Metropolitan Tower, 22nd Floor
539/2 Sriyudhya Rd., Rajthevi, Bangkok 10400, Thailand
{tcharoen, ananlada}@notes.nectec.or.th, virach@links.nectec.or.th

ABSTRACT

There is a common need in romanizing words in the languages other than English for the global communication. Especially the romanization of proper names are inevitable. Since there is no a mutual standard, writing a Thai word in English letters is not trivial, and it is quite a labor intensive task if it cannot be computerized. In this paper, we propose a new romanization system aiming at initiating the standardization process and implementing in a computer assisting module. The romanization is not a simple one-to-one matching. We need some linguistic rules to restrain the possible combination in terms of pronunciation availability and syllable construction. Before romanizing a Thai syllable, we need to break a word into a sequence of syllables. From the sequence of syllables, we then generate a roman script for it. In this paper, we prepare a syllable construction rule to drive the NFA to produce all possible sequences. The probabilistic n-gram is introduced to find the most probable one. Since there are a lot of ambiguities in breaking a word into a sequence of syllables, we rank the candidates according to their probabilities and apply a general beam search method to reduce the search space.

1. INTRODUCTION

At present, there is no rigorous rule to write a Thai word in roman script. A word, for example, “วิรัช” [wi-rat] which is a person name, is generally found in various spellings such as: Virach, Virat, Virax, Wirach, Wirat, etc. This can be found in writing a person name in a passport, street names, city names and so on. The problem is not trivial if we want to identify a word in more precise way. In 1908, the problem was unveiled when the first geographical map was created by a cooperation group between French and Thai governments.

To make a Thai word readable for other native speakers, we may write it in a phonetic form called the *transcription* method. This method is usually used in a dictionary by adopting an international standard such as the IPA (The International Phonetic Alphabet). For example, “คีน” is written as [khu:n]. Through the pronunciation of a word can be precisely written down, it is not widely used in general representation because many special symbols are introduced and the basic knowledge of phonetics is needed. To avoid such difficulty, we may transliterate a Thai word directly character by character which is called the *transliteration* method. Such that, the word “คีน” is written as [khuen]; the initial consonant,

the vowel and the final consonant are transliterated into [kh], [ue], and [n] respectively. However, the transliteration method may not work for a word including a vowel which is not placed in the normal reading position. In Thai, there are some vowels which are placed in front of the initial consonant, e.g. “เ” [e], or surrounding the consonant, e.g. “เ-า” [ao]. For example, “เตา” is transliterated into [eta], but the correct pronunciation is [tao]. In this paper, we propose to use the *romanization* method. This method uses only roman script to represent the pronunciation of a word. The romanizing table we use in this paper is a revised version of the table appeared in [3].

Before romanizing a Thai syllable, we need to break a word into a sequence of syllables. From the sequence of syllables, we then generate a roman script for it. In this paper, we prepare a syllable construction rule to drive the nondeterministic finite automata (NFA) to produce all possible sequences. The probabilistic n-gram is introduced to find the most probable one. Since there are a lot of ambiguities in breaking a word into a sequence of syllables, we rank the candidates according to their probabilities and apply a general beam search method to reduce the search space.

This paper is organized into: Section 2 briefly reviews the previous approaches in romanization for the Thai script.. The automatic romanization and the probabilistic n-gram model will be discussed in Section 3.

2. THOUGHTS ON THE ROMANIZATION OF THAI

The attempt to romanize a Thai word has been recorded since 17 th century, when many Europeans came into Thailand [6]. Unfortunately, the romanization was haphazardly defined by the proposers. They used the principle of their own languages to spell out the sounds they heard. Romanizing in the past, therefore, depended on the proposers’ native languages and the sharpness of their ears. A substituted romanization guide for the Thai script was significantly concerned in the early of 20 th century, when King VAJIRAVUDH (Rama VI) of the Thai Kingdom proposed a “graphic system” for romanizing the Thai script found in the article of “The Romanisation of Siamese words” in 1912 [2]. Rather than representing a word by the sound it is pronounced, the “graphic system” represents a Thai word according to the pronunciation of the original word. Many Thai words are loaned from Pali and Sanskrit. Therefore, keeping the original pronunciation was thought to be more useful to recognize the word, especially among the languages that have the loaned words from Pali and Sanskrit, e.g. Hindi,

Laotian, etc.. Comparing with the “phonetic system”, which transcribes the word pronunciations, the “graphic system” does not provide a sufficient pronunciation guide. However, the “phonetic system” does not provide a sufficient information about the spelling. For example, “พุทธศักราช” [6]

Graphic system BUDDHASAKARAJA
Phonetic system PHUTTASAKARAT

Table 1 shows some of the differences between the “phonetic” and “graphic” systems.

| Thai char. | Pronounce | Phonetic ¹ | Graphic ² |
|------------|-----------|-----------------------|----------------------|
| ข | Kh | Kh | gh |
| ด | Th | Th | d |
| ภ | Ph | Ph | bh |

Table 1 Some examples of the differences between the phonetic and graphic systems.

Later, the Royal Institute under the ministry of education proposed two romanizing systems in the “Romanization Guide for Thai script”. That is, the general system based on phonetic principle but ignored tone and vowel length, and the precise system based on both spelling and pronunciation. Table 2 shows some examples of the both systems.

| Thai char. | General System | | Precise System | |
|------------|----------------|-------|-----------------|---------------------|
| | Initial | Final | Initial | Final |
| ข | kh | k | kh ^c | k(kh ^c) |
| ด | th | t | th | t(th) |
| ภ | ph | p | ph ^c | p(ph ^c) |

Table 2 Some examples from the “Romanization Guide for Thai script” of the Royal Institute (1934)

The Royal Institute has been working on the system for romanization continuously. However, most of the people still use the system they are familiar to. Some use the “graphic system”, some prefer the “phonetic system”, and some use the combination of them. The system that is widely used presently, especially for geographic name and in government organization, is the one that was adapted from the Royal Institute’s general system by Royal Thai Survey Department in 1951. In 1998, the Royal Institute proposed a new guideline for the “transliteration of Thai characters into Latin characters” to the International Organization for Standardization (ISO). It is a kind of phonetic transcription system.

In conclusion, there are 2 major principles of romanization that are used in Thai, i.e. the “graphic system” and “phonetic system”. The phonetic system proposed by the Royal Institute seems to work well, but there are some defects. Namely, there is no distinction between short and long vowels, and the distinction between half-open and half-close back rounded vowels. We therefore propose a new phonetic system

which is improved from the Royal Institute proposal in 1998. However, the tones are omitted since the system will become too complicated. Table 3 and 4 are our new romanization tables for consonants and vowels respectively.

| Thai char. | Pronounce | Initial | Final |
|-------------|-----------|---------|-------|
| ก | [k] | k | k |
| ข, ฃ | [kh] | kh | k |
| ค, ฅ, ฆ | [kh] | kh | k |
| ง | [ng] | ng | ng |
| จ | [tc] | c | t |
| ฉ, ฌ | [tch] | ch | none |
| ช | [tch] | ch | t |
| ซ, ณ, ๓, ๔ | [s] | s | t |
| ญ | [j] | y | n |
| ฎ, ฏ | [d] | d | t |
| ฏ, ฏ | [t] | t | t |
| ฐ, ฑ, ฒ, ณ, | [th] | th | t |
| น, ๖ | | | |
| ณ, ๗ | [n] | n | n |
| บ | [b] | b | p |
| ป | [p] | p | p |
| ผ | [ph] | ph | none |
| ฝ | [f] | f | none |
| พ, ภ | [ph] | ph | p |
| ฟ | [f] | f | p |
| ม | [m] | m | m |
| ย | [j] | y | i |
| ร | [r] | r | n |
| ฤ, ฤๅ | [ru, ru:] | rue | none |
| ล, ฬ | [l] | l | n |
| ฦ, ฦๅ | [lu, lu:] | lue | none |
| ว | [w] | w | o |
| ห, ฮ | [h] | h | none |
| อ | [ʔ] | ʔ | o |

Table 3 Romanization table for consonants

¹ System of D.J.B. Pallegoix (1854) [8]

² System of King VAJIRAVUDH (1912) [4]

| Thai script | Pronunciation | Roman |
|----------------|---------------|-------|
| อะ, อั, อา | [a, a, a:] | a |
| อัวะ, อิว | [ua, u:a] | ua |
| อัม | [am] | am |
| อิ, อี | [i, i:] | i |
| อึ, อือ | [ui, u:i] | ue |
| อุ, อู | [u, u:] | u |
| เอย | [y:i] | oei |
| เอะ, เอ็, เอ | [e, e, e:] | e |
| เออะ, เออ, เอ็ | [y, y:, y:] | oe |
| เอา | [au] | ao |
| เอียะ, เอ็ย | [iə, i:ə] | ia |
| เอ็อะ, เอ็อ | [uə, u:a] | uea |
| แอะ, แอ็, แอ | [ɛ, ɛ, ɛ:] | ae |
| โอะ, โอ | [o, o:] | o |
| เอะ, เอ็, ออ | [ɔ, ɔ, ɔ:] | or |
| ไอ, ไอ, ไอย | [ai, ai, ai] | Ai |

Table 4 Romanization table for vowels (“อ” represents any initial consonants)

3. ALGORITHM FOR AUTOMATIC ROMANIZATION

In this paper, we aim to romanize a Thai word according to its pronunciation. It is not possible to transcribe the input string character by character. It is because some Thai characters can be pronounced differently according to the position in a syllable. For example, in the word “พั้น-ภพ” [phan-phop], the first “พ” which is the initial consonant of the syllable “พั้น”, is pronounced as [ph], while the second one which is the final consonant of the syllable “ภพ”, is pronounced as [p]. According to this distinction, we have to determine the boundary of each syllable in a word and then find out the function of each character in the syllable in order to produce the correct pronunciation. Moreover, there are some homographic syllables whose pronunciations are different while having the same forms. For example อ in อ-วรรณ is pronounced as ออ, and อ in อ-นังคิ is pronounced as อะ

It is not trivial in determining the boundaries and the pronunciations of the syllables since we have to deal with many ambiguities. We describe the problems and propose a probabilistic n-gram method for solving the ambiguities in the following subsections.

3.1. Syllable boundary determination

At most, a Thai syllable consists of 4 components, i.e. initial consonant, vowel, final consonant and tone. In this paper, we

formalize a syllable structure as a regular language. So that we can use a nondeterministic finite automata (NFA) as a machine to determine all possible syllable boundaries for an input string. The NFA converter machine used in our algorithm is similar to a nondeterministic Mealy transducer as described in [5]. This machine outputs the input string together with a syllable-breaking symbol (-) when it found a possible syllable ending according to Thai syllable construction rules.

However, placing a syllable-breaking symbol cannot be simply done in the following cases.

1. Propagation on final consonant: The final consonant of the previous syllable is propagated to become the initial consonant of the following syllable. This kind of consonant appears only once but we pronounce it in both syllables. To deal with this problem we use ~ to indicate the propagation consonant. For example “จักรา” (จัก-กรร) is written as จักกร - ~ ร.
2. Intervening syllable: An intervening syllable is handle in the same way as a propagation consonant. “ศุภวรรณ” (สฺย-พะ-วัน) is written as สฺยภ - ~ - วรรณ.
3. Leading consonants with leading vowel. For example “เกษม” (กะ-เสม). The pronunciation and the character position of “เ” and “ก” are reverted. Since we have to keep the original position order for training the syllable n-gram model, we group this kind of pronunciation as one special syllable.

The NFA converter machine can effectively generate all possible sequences of syllable boundaries according to the syllable construction rules. However, it raises a new problem when the number of possible syllable sequences grow very high. Many ambiguities occur because a single consonant can be pronounced as a syllable, e.g. “ว” and “ร” in “วรรณ” (ว-วรรณ). The other reason is that Thai vowels have various forms. Some of them are composed of many single vowels such as “เอ็ย” and some consonants can also function as vowels, e.g. “อ” and “ว” in “อออ” and “อัว”.

To solve the ambiguity in syllable segmentation problem we purpose a probabilistic model for selecting the most probable syllable segmentation. The probabilistic model was shown to be an effective technique in Thai word segmentation problem [7] and in many natural language processing problems. The syllable segmentation problem can be defined as in Equation (1).

$$\text{Argmax } P(S_i | C) = \text{argmax } P(S_i)P(C | S_i) / P(C) \quad (1)$$

where $C=c_1c_2\dots c_m$ is a character sting of the input word, and $S_i=s_1s_2\dots s_n$ is a possible syllable segmentation. Since $P(C|S_i)$ is equal to 1 and $P(C)$ is a fixed constant for every candidate, Equation (1) can be simplified to

$$\text{Argmax } P(S_i | C) = \text{argmax } P(S_i) \quad (2)$$

$P(S_i)$ can be formulated by using N-gram model of a syllable sequence. In this paper we use a bigram model to

avoid the sparse data problem. Therefore, $P(S_i)$ can be computed by using the following equation.

$$P(S_i) = \sum \prod P(S_i | s_{i-1}) \quad (3)$$

To reduce the size of search space in determining $P(S_i)$, we introduce a beam search algorithm in the NFA converter machine. $P(S_i | S_{i-1})$ can be used to prune off the syllable sequences which are not likely to occur such as “พ-ร-ร-ณ” for “พรวณ” since “ร” (a or an) is mostly function as a vowel, but not a separately syllable. However, there are still have some cases that the context of a bigram model is not sufficient for selecting the most probable syllable segmentation. For example, the occurrence of intervening syllables in Pali and Sanskrit compound words. We pass this kind of ambiguity to the process of romanization.

3.2 Romanization

Most of the pronunciation ambiguities of an input string are reduced in the syllable segmentation module because the syllable mostly has only one possible pronunciation. Therefore, after the process of syllable boundary determination, each syllable is romanized by a rule based on the functions of the characters in the syllable. However, the pronunciations of some syllables are still ambiguous, that is the homographic syllable. Linguistic rules are introduced for selecting the best syllable sequence, and converting it into the roman script. Following is an example of the romanization rules [1].

- 1) Syllable segmentation candidates:

ว-ร-พรวณ

ว-ร-พรวณ

Romanization rules:

r1 ว -> วอ | ร -> ระ / วรร

r2 รร -> รร / C_C

Results:

Syllable segmentation: ว-ร-พรวณ

Romanization: worraphan

- 2) Syllable segmentation candidates:

กัล-ยา

กัล-~-ยา

Romanization rules:

r3 ล -> น-ละ / _#CV

Results:

Syllable segmentation: กัล-~-ยา = กัณ-ละ-ยา

Romanization: kanlaya

- 3) Syllable segmentation candidates:

สัค-วา

สัค-~-วา

Romanization rules:

r4 ก -> ก-กะ / _#{ข,ร,ล,จ}V

Results:

Syllable segmentation: สัค-~-วา = สัค-กะ-วา

Romanization: sakkawa

For the ambiguities that cannot be eliminated, we will produce the romanization results ranked according to their syllable segmentation probabilities.

4 CONCLUSION

We define the problem of romanization of a Thai word as a problem of syllable segmentation and the roman script mapping. Most of the ambiguities occur in the syllable boundary determination process. Like the problem of word segmentation, we introduce a probabilistic n-gram model to rank the possible syllable segmentation candidates and beam search algorithm to reduce the size of search space. To the output of the syllable boundary determination process, we apply a linguistic romanization rule which converts a sequence of syllables into a string of roman script. In this process, some ambiguities are also eliminated and the romanization results are ranked according to the syllable segmentation probabilities.

ACKNOWLEDGEMENTS

We would like to express our gratitude to Wisarn Ucharoen of the Royal Institute for the helpful information. We also would like to thank Theppitak Karoonboonyanan and Dr.Surapant Meknavin of SLL, NECTEC for their valuable discussion.

REFERENCES

- [1] Dutoit, T. An Introduction to Text-to-Speech Synthesis. Kluwer Academic Publisher, 1997, The Netherlands..
- [2] Griswold, A.B., Thoughts on the Romanization of Siamese. 1969, Bangkok.
- [3] Kanchanawan, N., “How to write Thai with Roman?” Weekly Matichon Journal, 1999. 19(964-969)
- [4] Kanchanawan, N., “Thai-Roman Transliteration” The Royal Institute Journal, 1987, pp 28-44.
- [5] Karoonboonyanan, T., Sornlertlamvanich, V. and Meknavin, S., “A Thai Soundex System for Spelling Correction”, *Proceeding of the National Language Processing Pacific Rim Symposium 1997*, 1997, pp. 633-636.
- [6] Kliptri, P. Romanization of Thai in Rattanakosin Era. (Thesis), Thammasat University, 1995 (in Thai)
- [7] Meknavin, S., Charoenpornasawat, P., and Kijisirikul, B., “Feature-based Thai Word Segmentation”, *Proceeding of the National Language Processing Pacific Rim Symposium 1997*, 1997, pp. 41-46.
- [8] Ronnakiat, N. “Thoughts on the Transliteration Thai scripts into Roman alphabet.” *Thammasat Journal* (1986) 15 (1): pp. 7-33.