# Grapheme-to-Phoneme for Thai

Pongthai Tarsaku and Virach Sornlertlamvanich

Information R&D Division, National Electronics and Computer Technology Center (NECTEC)

Gypsum Metropolitan Bldg., 22nd Fl., 539/2 Sri Ayudhaya Rd., Rajthevi, Bangkok 10400, Thailand.

ptarsaku@notes.nectec.or.th

## Abstract

Many difficulties in the Thai language such as the absence of boundary word, linking syllables in pronunciation, and homographs are challenging us in developing a Thai Grapheme-to-Phoneme (G2P) converter. Presently there are a couple Thai G2P systems which are proposed in ruled-based and decision-tree approaches. The rule-based approach has a drawback in the limitation of employing the context. The decision-tree approach is somehow able to capture the local context for making the decision. On the contrary, the Probabilistic Generalized LR (PGLR) approach is reported that both the global and local contexts are efficiently captured in the probabilistic model. In this paper, we implement a Thai G2P system based on the PGLR approach. The result of experiment shows 90.44% of word accuracy in case of ignoring vowels length and 72.87% of word accuracy in case of exact match evaluation. These results are superior to those of rule-based and decision-tree approaches.

## 1 Introduction

A Grapheme-to-Phoneme (G2P) module is a routine that converts an input word sequence into their corresponding phonetic transcription. It is one of the essential routines in developing a text-to-speech (TTS) module.

In order to implement a Thai G2P system, many language specific problems i.e. homograph, non-word boundary, tonal language, etc., need to be solved. Various approaches, i.e. dictionary-based, rule-based, and statistical-based approaches have already been proposed for Thai G2P system (Chotimongkol, 2000), (Khamya, 2000), (Luksaneeyanawin, 1990), (Mittapiyanurak, 2000), (Narupiyakul, 1999), and (Wiboon, 1999).

Luksaneeyanawin (Luksaneeyanawin, 1990) and Mittapiyanurak (Mittapiyanurak, 2000) proposed a dictionary-based Thai G2P system for using in the text-to-speech (TTS) system. This approach requires a large dictionary cannot deal with unregistered word problem. To overcome the problem, Narupiyakul (Narupiyakul, 1999) and Khamya (Khamya, 2000) proposed a rule-based approach for Thai G2P system. There is a set of rules of syllable construction which is implemented in a finite state machine for detecting syllable structures. Wiboon (Wiboon, 1999) proposed a rule-based approach using regular expression to deal with the unregistered word problem. However, the rule-based approach has a drawback in dealing with the ambiguity in syllabification in the Thai language. Chotimongkol (Chotimongkol, 2000) proposed a decision tree approach, but the problem of inversion of phonemes across syllable boundaries is left unsolved.

PGLR parsing is a technique invented for incorporating probabilistic value into each the parse tree of a sentence (Sornlertlamvanich, 1998). The PGLR is implemented by using the advantage of context-sensitivity in the GLR parsing framework (Tomita, 1991). As a result, the PGLR can efficiently capture the information from both the global and local contexts during parsing. This paper is organized in 5 sections. Section 2 describes Thai syllable structure and the difficulties in determining word pronunciation. Section 3 explains the architecture of G2P implemented on the PGLR framework. Section 4 shows the experimental results.

## 2 Thai Pronunciation System

### 2.1 Thai Syllabic Representation

A basic Thai-pronunciation unit is a syllable that can be represented in the form of $C_i$ V ($C_f$) (T), where $C_i$, V, $C_f$, and T denote an initial consonant, a vowel, a final consonant and a tonal marker, respectively. The components inside the parenthesis, i.e. $C_f$ and T, are optional. The details of each component are described in the followings:

1. Initial consonant ($C_i$): There are 44 consonants in Thai and all of them can be an initial consonant of a syllable (two of them, "ฃ" and "ฅ", are not often used in modern Thai). However, only 21 phonemes are needed to represent all pronunciation of these consonants as shown in Table 1.

   Table 1. Phonetic symbols of Thai consonants.

| Consonant | Phoneme | | Consonant | Phoneme | |
|---|---|---|---|---|---|
| | Initial | Final | | Initial | Final |
| ก | /k/ | /k/ | บ | /b/ | /p/ |
| ขฆ,ค, | /kh/ | /k/ | ป | /p/ | /p/ |
| ง | /ng/ | /ng/ | ผภ,พ, | /ph/ | /p/ |
| จ | /c/ | /t/ | ฝฟ, | /f/ | /p/ |
| ฉช,ฌ, | /ch/ | /t/ | ม | /m/ | /m/ |
| ซส,ษ,ศ, | /s/ | /t/ | ร | /r/ | /n/ |
| ญย, | /j/ | /j/ | ลฬ, | /l/ | /n/ |
| ฎด, | /d/ | /t/ | ว | /w/ | /w/ |
| ฏต, | /t/ | /t/ | หฮ, | /h/ | - |
| ฐธ,ท,ถ,ฒ,ฑ, | /th/ | /t/ | อ | /?/ | - |
| ณ,น | /n/ | /n/ | | | |

   Some initial consonants are written in pair and act as an initial consonant. This type of initial consonants is called 'double consonant'. There are four catagories of double consonant:

   a) True clusters: two consonants pronounced in 1 single phoneme. For example, ("ปร", /pr/), ("ตร", /tr/), ("กร", /kr/), ("กล", /kr/), ("กว", /kw/), ("พร", /phr/), ("ทร", /thr/), ("คร", /khr/), ("ขร", /khr/), ("พล", /phl/), ("ผล", /phl/),("คล", /khl/), ("ขล", /khl/), ("คว", /khw/), ("ขว", /khw/).
   b) Pseudo clusters: two consonants pronounced in one completely different phoneme or pronounced with only the leading consonant pronunciation. For example, ("ทร", /s/), ("จร", /c/), and ("ซร", /s/).
   c) Parallel consonant characters: For example ("กล", /k-a-1/l-/), ("ปร", /p-a-1/r-/), etc.
   d) Leading consonant characters.

   More information can be found in (Luksaneeyanawin, 1993).

2. Vowel (V): There are 28 vowels in Thai composing of 18 monophthongs, 6 diphthongs, and 4 vowel letters as shown in Table 2.

   Table 2. Phonetic symbols of Thai vowels.

| Type | Short vowel | | Long vowel | |
|---|---|---|---|---|
| | **grapheme** | **phoneme** | **grapheme** | **phoneme** |
| Monophthong | –ะ | /a/ | –า | /a:/ |
| | –ิ | /i/ | –ี | /i:/ |
| | –ึ | /v/ | –ื | /v:/ |
| | –ุ | /u/ | –ู | /u:/ |
| | เ-ะ | /e/ | เ- | /e:/ |
| | แ-ะ | /x/ | แ- | /x:/ |
| | โ-ะ | /o/ | โ- | /o:/ |
| | เ-าะ | /@/ | ออ | /@:/ |
| | เ-อะ | /#/ | เ-อ | /#:/ |
| Diphthong | เ–ียะ | /ia/ | เ–ีย | /i:a/ |
| | เ–ือะ | /va/ | เ–ือ | /v:a/ |
| | –ัวะ | /ua/ | –ัว | /u:a/ |
| Vowel Letter | -ำ | /am/ | - | - |
| | ไ-,ใ- | /aj/ | - | - |
| | เ-า | /aw/ | - | - |

3. Final consonant ($C_f$): Not all Thai consonants can be a final consonant. For example, "ห", and "ฮ" cannot be used as a final consonant. Only 9 phonemes are needed to represent the final consonants as show in Table 1. Like the initial consonant, the final consonant can be a double consonant. For example, /กร/ is mapped to /k/, /คร/ is mapped to /k/, /ตร/ is mapped to /t/, /ทร/ is mapped to /t/, and /ปร/ is mapped to /p/.
4. Tone (T): Like Chinese, Thai is a tonal language. There are five tones in Thai, i.e. mid, low, falling, high and rising. Four tonal markers and one non-mark are used to

indicate the tone. A tone is determined by the combination of syllable structure, initial consonant and the tonal marker.

### 2.2 Difficulties in Thai pronunciation system

Difficulties in transcribing a Thai letter can be classified as follows.

1. Ambiguity in grapheme-phoneme mapping: Some graphemes can be mapped to multiple phonemes depending on the context. For example, the grapheme "ฑ" in the word "มณฑา" is mapped to /th/, while it is mapped to /d/ in the word "มณฑป".
2. Homograph: Word that is pronounced differently according to the meaning. In order to pronounce a homograph word, the sentential context is taken into considerarion. For example, "เพลา" is pronounced /phl-a-w-0/ to mean "axle", but "เพลา" is pronounced /ph-e:-0/l-a:-0/ to mean "time".
3. Vowel length: The problem occurs when a vowel is pronounced a short (long) vowel according to its grapheme, but by any reasons it is pronounced as a long (short) vowel instead. For example, the word "น้ำ" which is literally pronounced as /n-**a**-m-3/ (short vowel), is conventionally pronounced as /n-**a:**-m-3/ (long vowel). The word "ท่าน" is literally pronounced as /t-**a:**-n-2/ (long vowel), but it is conventionally pronounced as /t-**a**-n-2/ (short vowel).
4. Linking syllable pronunciation: The problem occurs in a polysyllabic word where the final consonant of the forthcoming syllable is explicitly pronounced with /a/ vowel as an additional syllable. For example, the polysyllabic word "วิทยา" is pronounced as /w-i-t-3/th-a-2/j-a:-0/. In this case, the grapheme "ท", the final consonant of the first syllable, "วิท", is added with /a/ vowel to pronounce /th-a-2/ as the second syllable. As a result, the overall pronunciation is /w-i-t-3/th-a-2/j-a:-0/.
5. Ambiguity in consonantal functionality: In some Thai words, a consonant can be both the final consonant of forthcoming syllable and the initial consonant for the next syllable. For example, "ฐ" in "อัฐิ" (/?-a-t-1/th-i-1/) is both the final consonant /t/ of the first syllable "อัฐ" and the initial consonant /th/ of the following syllable "ฐิ".
6. Word boundary: Unlike some other languages such as English, Thai has no word boundary. Therefore, different segmentation can yield different words and pronounce in different syllable units. For example, the word "ตากลม" can be segmented into "ตา|กลม" (round eye) and "ตาก|ลม" (to expose wind) which are pronounced /t-a:-0/kl-o-m-0/ and /t-a:-k-1/l-o-m-0/ respectively.

## 3 Grapheme-to-Phoneme System

### 3.1 PGLR Approach

The entire Thai G2P system is implemented on the PGLR (Sornlertlamvanich, 1998) approach as shown in Figure 1. The input text, a sequence of Thai graphemes, are parsed by the GLR parser (Tomita, 1991) with the CFG rules for syllable construction. The outputs of the GLR parser are the parse trees with the corresponding probabilities. The most likely parse tree is

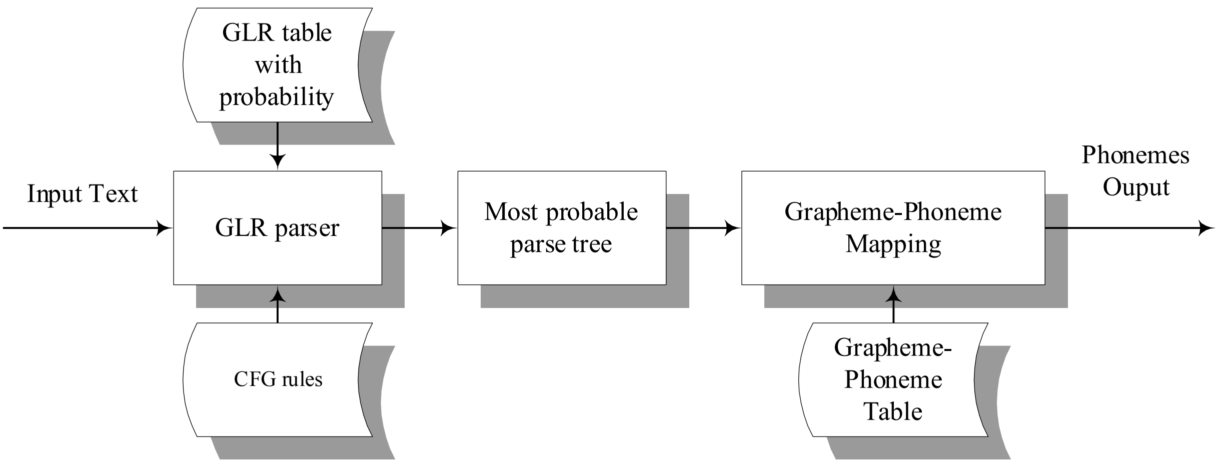


Figure 1. Thai G2P System: the PGLR approach

selected and then transcribed into a phoneme sequence by the Grapheme-Phoneme mapping module by looking up the Grapheme-Phoneme table.

### 3.2 CFG Rule's Construction

The G2P conversion rules are writen in CFG expressions in following strategy:

1. Each CFG rule represents the Thai syllable construction.
2. The CFG rules are grouped by the vowel symbols. An example of the CFG rules that are grouped by the vowels "า" and "เ" is as follows:
   <Grp_1> -> <Init> า
   <Grp_1> -> <Init> า <Final>
   <Grp_2> -> เ <Init>
   <Grp_2> -> เ <Init><Final>
   Consequently, the CFG rules for all possible syllable structures are prepared for each vowel symbol.
3. Due to some difficulties in Thai pronunciation as mentioned previously, some Thai syllables can not be analyzed by regular rules. To overcome this problem, specially written rules are constructed. In our current work, we limit ourselves to work on only two problems described below.
   i) An ambiguity in consonantal functionality problem: This problem occurs to some of the words containing more than one vowel. For example, the word "อัฐิ" which contains two vowels: "–ั"and "–ิ". In order to match this case, a special form of
   <Syl> -> <Init>Vow1<Final>Vow2 is created.
   ii) The linking syllable problem: This problem is simply solved with the construction of a linking syllable rule by modifying the existing regular rules. For example, the regular rule of the form
   <Syl> -> <Init>า<final>
   which can parse a syllable, for example, "กาย" (/k-a:j-0/), is modified into the linking syllable rule of the form
   <Syl> -> <Init>า<Final_Link>
   to parse a linking syllable in "กายกรรม" (/k-a:j-0/j-a-3/k-a-m-0/). As a result, the syllable /j-a-3/ is added after the syllable /k-a-j:0/ when the above rule is implemented.
4. Since a Thai word may compose of one or many syllables, we added the following rules to deal with polysyllable words.
   <Word> -> <Syllable>
   <Syllable> -><Syllable><Syl_Grp>
   <Syllable> -> <Syl_Grp>
   <Syl_Grp> -> <Grp_1>
   <Syl_Grp> -> <Grp_2>
   .. .. ..
   .. .. ..
   <Syl_Grp> -> <Grp_n>
   where <Grp_*I*> for $I = 1, 2, \ldots, n$ is a rule for monosyllable structure. With these extended rules, the CFG rules can deal with both monosyllable and polysyllable words.

### 3.3 Generating of Corpus

Since the CFG rules for syllable construction do not allow structure ambiguity, the corpus can simply be obtained by parsing a set of words

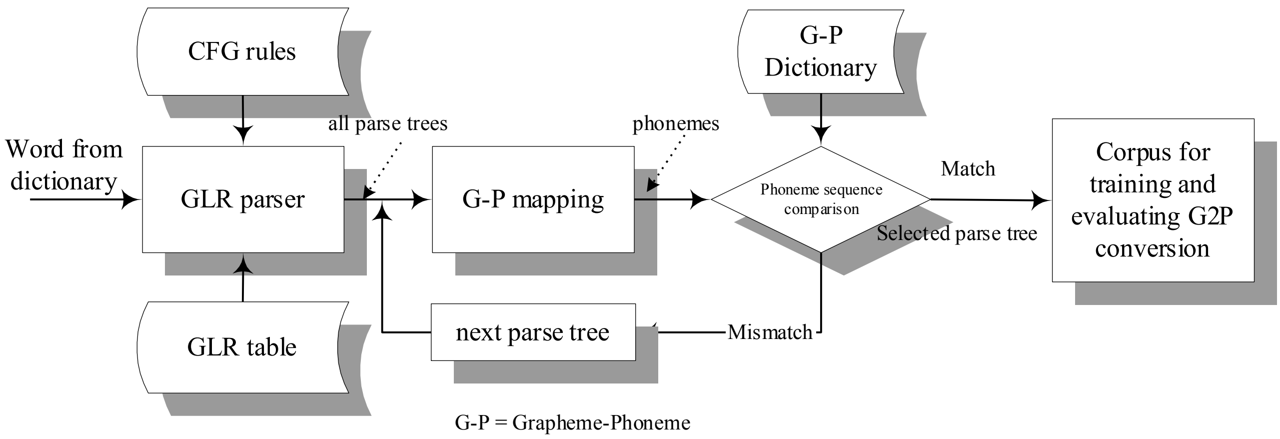


Figure 2. Generating of corpus

whose pronunciations are already known. The 23,000 words with their pronunciations are extracted from the LEXiTRON Dictionary (http://www.links.nectec.or.th/lexit/.). Each word is parsed for a parse tree that yields the correct pronunciation by consulting the dictionary. The process of generating the corpus is shown in Figure 2. As a result, each word with its correct parse tree is generated and will be used as the corpus for training the PGLR parsing table and evaluating the parsing accuracy.

### 3.4 Training of the PGLR parsing table

The probabilistic GLR parsing table is a LR table in which each action is assigned a probability gained from parsing a bracketed training set (Sornlertlamvanich, 1998). A four-fifth of the corpus (18,400 parse trees) is randomly selected for training. The PGLR parser parses the training set to obtain the probability of each action in the PGLR parsing table.

## 4 Experimental results

One-fifth of the corpus (4,600 parse trees) is used for testing the parse accuracy. Table 3 shows the results of PGLR approach comparing with rule-based (Wiboon, 1999) and decision-tree approaches (Chotimongkol, 2000). The evaluation result shows the grapheme-to-phoneme conversion accuracy in the case of completely correct pronunciation (exact match) and the case of vowel length ignoring pronunciation. The exact match accuracy indicates that the output of the model has exactly the same phoneme sequence as the one given in the corpus. The ignorance of vowel length match accuracy shows the model accuracy in yielding the phoneme sequence by ignoring the length of vowel which is less important than the phoneme sequence. It is expected that the vowel length can be determined by other means. In the implementation cases, the PGLR approach yields the best result comparing with those of the rule-based and decision-tree approaches. However, about half of the errors are the words that include linking syllables. This problem is not trival for native speakers either in case of the unseen words.

Table 3. Grapheme-to-phoneme conversion accuracy

| Model | Conversion (word) accuracy(%) | |
|---|---|---|
| | Exact match | Ignorance of Vow. Length |
| PGLR | 72.87 | 90.44 |
| Rule-based | 67.14 | 83.81 |
| Decision Tree | 68.76 | 86.94 |

## 5 Conclusion

Theoretically, PGLR can capture both local and global context which is reported the high performance in syntactic parsing for sentences. Our research has shown the best result in applying PGLR in G2P problem comparing with rule-based and decision tree based approaches. This concludes that the context for probabilistic decision in PGLR is good enough for a relatively small structure such as words (phoneme sequence) as well.

## Reference


Chotimongkol, A. "Statistically Trained Orthographic to Sound Models for Thai," Proceeding of the 6th International Conference on Spoken Language Processing," pp. 533–554, Oct. 2000.

Khamya, A. "SATTS: Syllable Analysis for Text-to-Speech System," Proceeding of the Fourth Symposium on Natural Language Processing 2000 (SNLP'2000), pp. 336–340, May 2000.

Luksaneeyanawin, S. "A Thai Text to Speech System," Proceeding of the Conference of the Regional Workshops on Computer Processing of Asian Languages. Asian Institute of Technology, pp. 305–315, 1990.

Luksaneeyanawin, S. "Speech Computing and Technology in Thailand," NLP in Thailand, pp. 276–321, 1993.

Mittapiyanurak, P., Hansakunbuntheung, C. and Tepprasit, V. and Sornlertlamvanich, V. "Issues in Thai Text-to-Speech Synthesis: The NECTEC Approach," Proceedings of NECTEC Annual Conference, Bangkok Thailand, pp. 483–495, 2000.

Narupiyakul, L. “The Phonetic Transcription of Thai Word,” Proceeding of the International Symposium on Intelligent Signal Processing and Communication Systems, pp. 789–792,1999.

Sornlertlamvanich, V. “ Probabilistic Language Modeling for Generalized LR Parsing,” Technical Report, Department of Computer Science, Tokyo Institute of Technology, September, 1998.

Tomita, M., “Generalized LR Parsing,” Kluwer Academic Publishers, 1991.

Wiboon, T. “The Engineering Training Report at NECTEC,” Engineering Faculty, Khasetsart University, 1999.

http://www.links.nectec.or.th/lexit/.