GENERATING PHONEMES FROM WRITTEN THAI USING LEXICAL ANALYSIS BASED ON REGULAR EXPRESSIONS

Leo van Moergestel[1] and John-Jules Meyer[2]
[1]*HU University of Applied Sciences Utrecht, Utrecht, The Netherlands*
[2]*Utrecht University, Utrecht, The Netherlands*

Keywords: Natural Language Processing, Thai Language, Romanization of Written Thai.

Abstract: This document describes the approach and techniques used in software that has been developed to generate phonemes from written Thai. This software has been used to generate the phonetic transcription of Thai words in a Thai-Dutch dictionary. The most important part of this software is a lexical analyzer based on regular expressions for matching patterns in the Thai writing system. Because most software tools that use regular expressions are still based on the 7-bit ASCII set, a mapping of Thai characters to ASCII-characters has been used.

1 INTRODUCTION

The written form of words, the so called orthography, differs in some languages from the pronunciation. Especially English uses the written form *ough* at least for seven different sounds as can be shown in the sentence: *"Though a rough cough and hiccoughs ploughed through him, he houghed the horse with thorough thoughtfulness."* For these languages it is difficult, if not impossible, to generate the pronunciation sounds in a written form using the international phonetic alphabet (IPA) (Samuel, 1983), except for the situation where a list of phonetic representations for every single word is available. For languages like Spanish and Italian the situation is a bit better, though it is still a hard problem to generate what is called the transcription. By transcription we mean the broad transcription that is used in dictionaries, not the narrow transcription that is used in studying dialects or differences in pronunciation of the same word among different people using that word. In the current paper, transcription of Thai is discussed, but the techniques and approaches could be useful for other non roman based writing systems.

2 FLEX

Flex (Levine, 2009) is a tool to build scanners and stands for fast lexical analyzer. Flex is based on an older tool called lex. A scanner is a program which recognizes lexical patterns in text. To use flex one has to write a flex source file where the most important part consists of rules. These rule are pairs containing a regular expression and an action for the scanner to perform when the regular expression is matched. By using flex, the flex source file can be converted to a C program file that can be compiled by a C compiler thus producing the scanner.

A regular expression is a pattern describing a set of text strings. A regular expression is said to match a text if the text is a member of the set described by the regular expression. A good treatment of regular expressions can be found in (Friedl, 2006). The theory of regular expressions is based on formal languages and finite state machines. (Hopcroft et al., 2006).

In the current work the most important part is the development of regular expressions that match with patterns of written Thai. By using these expressions in flex rules we can generate output from our scanner. This output explains the structure or blueprint of the Thai text. If the structure is known, it is easy to generate phonemes for the given text.

3 THE THAI WRITING SYSTEM

The Thai language has its own alphabet consisting of 44 consonants (Campbell and Shaweevongs, 1956). The vowels are a separate set of symbols that are sometimes combined to represent other vowels or

van Moergestel L. and Meyer J..
GENERATING PHONEMES FROM WRITTEN THAI USING LEXICAL ANALYSIS BASED ON REGULAR EXPRESSIONS.
DOI: 10.5220/0003735603060311
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 306-311
ISBN: 978-989-8425-95-9

vowel combinations (diphthongs). Finally there is a set of extra symbols that can be used to mark the tone of a syllable but also to suppress the sound of a syllable or character. Thai is a tonal language that has 5 tones, low, medium, high, rising and falling. For speakers of a language that does not use tones, it is rather difficult to get used to a tonal language. The Thai writing system is capable to express the tone of a certain syllable. So syllables with different tones are also written in a different way. A word consists of one or more syllables. Every syllable has its own tone. Here are some important rules of the Thai writing system:

- All consonant characters can be at the start of a syllable, but a few consonants are not used at the end of a syllable;
- A syllable can have an inherent short 'a' vowel sound;
- A syllable can have an inherent short 'o' vowel sound;
- A syllable end can be considered living or dead and this fact influences the tone;
- A vowel symbol (or group of symbols representing a vowel or diphthong) cannot be used on its own, it should always have an accompanying consonant character or a consonant cluster. This is the consonant representing the beginning of the syllable;
- Vowels can have a long or short duration
- Depending on the vowel symbol, the written position of the vowel can be before, after, above or under the consonant it is belonging to;

4 CLASSIFICATION OF THE SOUNDS IN SPOKEN THAI

In this section we will focus on the sounds of spoken Thai. For this section, it is important to understand what we are trying to accomplish. When we have a word written in Thai, we want a classification or blueprint for the syllables of this word. This means that after classification we have information for a word in a standardized form. The role of the symbols, both consonants and vowels, for each syllable is clear. Consider a word consisting of two syllables our aim is to produce a classification that looks like this string:

$$< SaaVbbEcc >< SxxVyyEzz >$$

Where *aa...zz* represent numbers for identification purposes. S stands for starting consonant, V for vowel sound, and E for end consonant for that syllable. The set of S types includes consonant clusters and special cases where we do not use a consonant at the beginning of a syllable. The same thing is true for the end consonants though that set of possibilities is significant smaller for the Thai language. So in our notation every syllable classification starts with $<$ and ends with $>$ and uses a fixed format.

4.1 Consonants

At the start of a syllable the following situations are possible:

- A single consonant sound consisting of one Thai character.
- A cluster of two consonants. All possible consonant clusters are summarized in table 1
- A single consonant sound consisting of two Thai characters. This is mostly the case where a consonant named *hoh-heep* is used to influence the tone of the syllable. In Thai this character in this situation is called *hoh-nahm*.
- No consonant sound at the beginning, though there is a Thai consonant character (*oh-ang*) to carry the vowel symbol that cannot stand on its own. Sometimes the aforementioned situation results in a glottal stop at the beginning of a syllable.

The end of a syllable is rather simple. There are only 13 possibilities of a single consonant sound. Three of them are only used in loanwords, mostly from English origin. The possibilities are summarized in table 3. Remark that the occlusives are unreleased finals in contrast to western European languages where these occlusives are mostly released. Table 2 shows the notation we use for the classification of the end consonants.

Table 1: Possible start clusters.

unaspirated			aspirated		
kr	kl	kw	khr	khl	khw
pr	pl		phr	phl	
tr			thr		

Table 2: End consonant types.

E00	no end consonant
E01	glottal stop
E02	end consonant
E03	end consonant and start of next syllable

The type of consonant at the beginning of a syllable as well as the ending of a syllable will influence the tone.

Table 3: All possible syllable ends.

k	m	w	f (loan words)
p	n	j	s (loan words)
t	ng	*no sound*	l (loan words)
		glottal stop	

4.2 Vowels

The Thai writing system uses a special symbol set to express vowel sounds. A vowel or vowel combination can use one or more symbols of this set. Even some consonant symbols are used as vowels or in combination with vowel characters. The patterns for these vowels play an important role is our scanner, because it will clearly indicate the beginning consonant or consonant cluster of a syllable.

4.3 Special Symbols

Special symbols are used to mute a character, repeat a word or denote the tone of a syllable.

4.4 Properties of the Thai Writing System

The Thai writing system is capable of expressing the tone as well as the sound of a vowel. By carefully analyzing written Thai, it is possible to generate most of the information for a transcription, there are however situation where this transcription is ambiguous. These situations must be taken care of before scanning. Also special constructions should be taken in account by the scanner.

5 DESIGN OF THE SCANNER

The scanner is based on lexical analysis of written Thai and is done in a sequence of passes. To use ASCII based tools, like lex and its GNU derivative flex we made a transcription to ASCII characters. The scanner focuses on a character string generated by the a pass called prelex. This pass maps the consonants and vowels in the syllables of a Thai word to ASCII characters. The following sequence of passes is used (in parenthesis is the name of the tool that is responsible for the action taken).

1. in the first pass the Thai string to parse is duplicated (*dupthai*). This is done because our analysis will result in an abstract classification of the syllables. To construct the transcription, we need the original Thai text for extra information like the tones of the syllables;
2. The duplicated string is converted to an ASCII based representation. Some Thai characters with identical properties are represented by the same ASCII character (*prelex*);
3. the next step is called ”dining vowels” (*dinvow*). In Thai, a vowel is always attached to one or more consonants. In this step we generate part of the classification of the sounds by letting these vowel symbols absorb the consonant(s) they belong to. That is the reason why this step is called ”dining vowels”. When this step is completed, all written vowels are classified together with a substantial amount of consonants.
4. There is now a partial transcription to consonant and vowel types, but there are still consonants left. Some of these dangling consonants have inherent vowel sounds depending on the situation of a closed or an open syllable. This pass tries to solve this situation by looking at the number of consonants and the position in the string. (*flexpass*). Another possibility is that these consonant are closing consonants for a syllable that is partially converted in step 2. Finally a scan is done to close all unclosed patterns. If there are syllables that were classified as possibly closed syllables in the classification a ! symbol is used. If this symbol is now followed by the start of a new syllable i.e. $<$, the ! symbol is replaced by an $>$ denoting the end of the syllable.

All steps are simple commands or tools. On a Unix based system, these commands are operating in a so called pipeline.

```
dupthai i | prelex | dinvow | flexpass o
```

After applying these commands, where i is the input file and o the output, the classification is done.

6 SCANNER DETAILS

Because the most complex parts are implemented using flex, this section will focus on the passes using flex (i.e. dinvow and flexpass). Of these two pases, dinvow is the most complex part and has 252 patterns with related actions pairs. To make the source better readable and also maintainable a set of macro definitions has been introduced.

```
S [YWHoGChKlsmTtPpwRLSNfc]
E [GChKlsmTtPwRLSY]
ot [1-4]?
Hn H[sYRLWN]
Kc [GCK][RLW]
Pc [Ppw][RL]
```

S is the set of possible start consonants, E the set of possible end consonants and ot an optional tone mark symbol. Hn shows all possible combinations of Hohnahm with other consonants. Kc is a shorthand for all possible K-clusters and Pc for the possible P-clusters. A possible pattern action pair is:

```
{S}M{ot}{E}     printf("<S01V03E02>");
```

If the scanner detects a single start consonant, represented by the macro {S}, followed by a vowel symbol, represented here by M, with one or zero tone marks, given by the macro {ot} and an end consonant that is in the set of possible end consonants {E} it will output <S01V03E02 >.

Another possible pattern action pair is:

```
e{Kc}I{ot}Y     printf("<S04V27!");
```

The pattern matches a K-cluster Kc in combination with a compound vowel symbol, represented here by e-IY (where the dash is the place of the start consonant or cluster), with one or zero tone marks, the action will be the generation of the output <S04V27E02!. This syllable could be complete, but there is also a possibility of an end consonant. In flexpass this situation will be solved. This is also a flex based scanner, but uses less patterns than dinvow. The actions of flexpass will be discussed in the example of the next section.

7 EXAMPLES

To explain in more detail the working of the scanner, in this section two examples will be given. Let us start with the famous ”Hello world” example. In written Thai *Hello world* looks like สวัสดีโลก and is pronounced sawatdee lohk, phonetic [sawatdi: lo:k]. These are the steps from the scanner to come to a blueprint where this phonetic result can be derived from. First the Thai string is duplicated with dupthai and then prelex makes a translation of Thai characters to standard ASCII characters. When the duplication of the original Thai text is ignored, this results in:

```
[SWMSmIOLG]
```

The [and]-signs are used as delimiters. Every character in this string represents a Thai symbol, a Thai vowel or a set (mostly with one member) of Thai consonants. Now we are ready to use dinvow to detect the vowels and associated consonants. The result is:

```
[S<S01V03E02><S01V10!<S01V18!G]
```

In this phrase, three vowel-symbols in written Thai are involved. The first one always needs an ending consonant, so this syllable is finished. For the second one there is the possibility of an end consonant and the same is true for the third one. This is the reason why these syllables are not closed by a > but by a ! symbol. At the beginning as well as at the end there is still a dangling consonant (symbol S and G). In the next step (flexpass) the dangling consonant at the beginning is treated as a single syllable with inherent a-sound and no end consonant. The dangling consonant at the end is combined with the possibly not closed syllable before it. The syllable denoted by S01V10 appears to have no possible end consonant so the combination !<will be translated to ><. The result is:

```
[<S01V02E00><S01V03E02><S01V10E00>
<S01V18E02>]
```

Now we have a complete blueprint of the phrase. In the section ’sound generation’ a tool will be introduced to make a phonetic representation. The difficult part however is now done by the scanner.

As a second example we consider the word เปรี้ยว meaning *sour* and pronounced priaw. This word consists of a consonant cluster pr ปร in combination with a compound vowel consisting of four symbols เ ี ยว and a tone symbol. To make things a bit more complicated, this compound vowel also uses consonant symbols. Using prelex the Thai characters are translated to:

```
[exxI2YW]
```

Using dinvow gives the result:

```
<S05V33E00>
```

The meaning of this blueprint is: this is a word with a p-consonant cluster (being either pl or pr according to table 1) and a compound vowel V33 (having the sound iaw) and no end consonant. In this case no operation is done by the next pass of the scanner, because the blueprint is already finished.

8 ERROR RECOVERY

There are situations where the scanner fails to produce the correct result. In this case a hint can be given to tell the scanner where the end of a syllable actually is. The scanner uses these hints as extra information and this could result in a correct transcription. Hinting is used in words where the syllables are not easy to determine. An example is the word: *birdcage* that is written in Thai using only five consonants กรงนก and is pronounced [krong nok]. In this situation the scanner has a choice of using an inherent ’o’ between two consonants or an inherent ’a’ after every single consonant or perhaps a combination of inherent ’o’ and ’a’ sounds. K-R-NG-N-K should however result in

krong nok, so only two inherent o-sounds. By hinting we tell the scanner that there is a syllable end after the NG. In that case the scanner separates this composite word in two parts that it will handle correctly. After using prelex, this results in:

```
[GRs<Z00>sG]
```

Where <Z00>represents the separator. Using this information, the next passes of the scanner are capable of generating the correct blueprint for two syllables.

```
[<S04V01E02><Z00><S01V01E02>]
```

If hinting does not give the expected result the Thai word is added to a file of exceptions where the correct set of sounds is entered by hand. In an extra pass at the beginning all these exceptions are searched for and the sounds are already added and marked as complete.

In the scanning system some decisions are beyond any doubt but some decisions are just good guesses. It is also possible to fall back on these guesses to produce another transcription in case of an erroneous result. This means that the aforementioned hints can also be generated by the system itself in a so called supervised learning situation, where an operator trains the system. This could be a useful add-on for this system.

9 SOUND GENERATION

By using the classification in combination with the original text, the sounds can be generated. This is done by a separate program that can use any table of transcription codes to generate the final transcription including a way to denote the tones of the syllables. This transcription can be based on the IPA, thus producing a *phonetic* transcription, but depending on the transcription code used, the transcription can also be adjusted to the phonemes of a certain target language. This results in a *phonemic* transcription. For example in Dutch the phonetic sound for [u:] is written as [oe], while in English it is written [oo], in German or Spanish [u] and in French [ou].

10 RESULTS

Figure 1 shows a dictionary entry (van Moergestel, 1995b) (van Moergestel, 1995a). The transcription is produced by the aforementioned scanner in combination with the sound generation program that translates according to a given set of sound representations including the tones. Because this is table-driven, one can produce different types of transcription as is explained in the previous section. A dictionary could use the IPA notation (Tingsabadh and Abramson, 1993). In the given example dashes, located at the baseline of the text, are used to indicate the low tone of both syllables.

ฝรั่ง [_fa_rang] ① buitenlander (blanke), westerling ② guave (vrucht)

Figure 1: Entry in the Thai-Dutch dictionary.

The small dictionary where this system is used for contained 6932 Thai words. The file of exceptions contains 373 words, meaning that 95 % can be transcribed by the parser. We used 630 words with hints to guide the scanner to the correct transcription. These results are summarized in table 4.

Table 4: Results.

Number of words	6932	100%
Exceptions	373	5%
Hints	630	9%

11 RELATED WORK

The Royal Institute of Thailand has published he Royal Thai General System of Transcription (RTGS). This system describes how the transcription of Thai words in the Latin alphabet should be done. Research in the field of Thai romanization has also been addressed, among others, by Aroonmanakun. In (Aroonmanakun et al., 2004) a system is described that can be used for transcribing English words, using the Thai writing system. By the same author a tool for transcription of Thai based on the system RTGS has been developed (Aroonmanakun, 2010). This is an interactive tool where the user enters Thai and the transcription is generated. Jucksriporn and Sornil developed a system for syllabification of Thai. They used a minimum cluster-based trigram statistical model (Jucksriporn and Sornil, 2011). This system could be used in combination with the system presented in the current paper and be useful for automatic hinting.

The system we present in the current paper can be seen as a generic approach for transcribing in general. This approach is based on a transposition of non ASCII characters to the ASCII-set and thus standard parsing tools that are used for compiler building can be used. By first generating a classification of the syllables, the final transcription can be generated in a flexible way. Most other transcription systems

use huge pattern bases and the transcription is generated without an intermediate classification. The system presented in this paper is compact, fast and it is possible to use the RTGS as well as symbols of the IPA or a transcription that maps the Thai sounds to a writing system that is familiar for readers of a certain language.

12 CONCLUSIONS

The Thai writing system is much better apt for automatic transcription than most European writing systems. Especially in English it is not possible to generate the sounds of pronunciation by using simple and consequent rules. The system presented here is useful for bulk transcription as needed in dictionaries but could be used as an intermediate system for generating sounds of the spoken language. The topic of future research will be the possibility of improvements of the scanner by studying patterns and classes of words in the exceptions and hinted groups.

REFERENCES

Aroonmanakun, W. (2010). ling.arts.chula.ac.th/tts. *Department of Linguistics, Chulalongkorn University, Bangkok*.

Aroonmanakun, W., Thapthong, N., Wattuya, P., Kasisopa, B., and Luksaneeyanawin, S. (2004). Automatic thai transcriptions of english words. *Southeast Asian Linguistics Society Conference 14 (SEALS 14), Bangkok, Thailand, May 19-21, 2004*.

Campbell, S. and Shaweevongs, C. (1956). *The Fundamentals of the Thai Language*. Paragon Book Gallery, New York.

Friedl, J. E. F. (2006). *Mastering Regular Expressions, third ed.* O'Reilly Media, Sebastopol, CA.

Hopcroft, J. E., Motwani, R., and Ullman, J. (2006). *Introduction to Automata Theory, Languages and Computation, third ed.* Addison Wesley.

Jucksriporn, C. and Sornil, O. (2011). A minimum cluster-based trigram statistical model for thai syllabification. *CICLing (2)2011 p493-505*.

Levine, J. R. (2009). *Flex and Bison*. O'Reilly Media, Sebastopol, CA.

Samuel, J. T. B. (1983). *Introduction to Practical Phonetics*. Summer Institute of Linguistics, England.

Tingsabadh, M. R. K. and Abramson, A. S. (1993). Thai. *Journal of the International Phonetic Association, 23, pp 24-28 doi:10.1017/S0025100300004746*.

van Moergestel, L. (1995a). *Woordenboek Nederlands-Thai (Dutch-Thai dictionary)*. Nangsue.

van Moergestel, L. (1995b). *Woordenboek Thai-Nederlands (Thai-Dutch dictionary)*. Nangsue.