Sirindhorn International Institute of Technology

Thammasat University

Thesis ICTES-MS-2010-01

TEXT ANALYSIS FOR THAI TEXT-TO-SPEECH SYNTHESIS SYSTEM ON EMBEDDED DEVICE

Arunee Ratikan

TEXT ANALYSIS FOR THAI TEXT-TO-SPEECH SYNTHESIS SYSTEM ON EMBEDDED DEVICE

A Thesis Presented

by

Arunee Ratikan

Master of Engineering

Information and Communication Technology for Embedded Systems

(ICTES) Program

Sirindhorn International Institute of Technology

Thammasat University

August 2010

TEXT ANALYSIS FOR THAI TEXT-TO-SPEECH SYNTHESIS SYSTEM ON EMBEDDED DEVICE

A Thesis Presented

By

Arunee Ratikan

Submitted to

Sirindhorn International Institute of Technology
Thammasat University
In partial fulfillment of the requirement for the degree of
MASTER OF ENGINEERING

Approved as to style and content by the Thesis Committee:

Advisor and
Chairperson of Thesis Committee _____ (Asst. Prof. Cholwich Nattee, Ph.D)

Committee Member and
Chairperson of Examination Committee _____ (Ananlada Chotimongkol, Ph.D)

Committee Member _____ (Assoc. Prof. Thanaruk Theeramunkong, Ph.D)

Committee Member _____ (Prof. Manabu Okumura, Ph.D)

External Examiner: Prof. Takao Kobayashi

August 2010

Acknowledgement

First of all, in my success I wish to express thankfulness to Asst. Prof. Dr. Cholwich Nattee who is my thesis advisor for his valuable advices and comments during the development of this study. He devotes time to review my papers and thesis, and gives good suggestions for me, although he is quite busy. I can remember that he told me "Cheer up" when I was excited for the first presentation in ICICTES 2010 conference. This word makes me feel good. Moreover, he is very kind and always smiles when I meet him. Without the help of my advisor, this thesis cannot be completed. I am glad to be a student of Asst. Prof. Dr. Cholwich Nattee.

I gratefully thank to Dr. Ananlada Chotimongkol, my thesis co-advisor from National Electronics and Computer Technology Center (NECTEC). Her wide knowledge and her logical way of thinking have been of great value for me. She gives useful suggestions and comments encouraged in my thinking and provided new insights and elucidations to theoretical during this study. Moreover, she supports a working seat for me at Human Language Technology (HLT) laboratory, thus I can advise her every time when confusing in theory. In this laboratory, I have an opportunity for learning many things from many researchers such as P'Aom, P'Tic, P'Noi, P'Pong, and etc. Dr. Ananlada is the first person who teaches me how to write the paper and helps revise the draft paper until late night.

I would also like to thank Assoc. Prof. Dr. Thanaruk Theeramunkong, my thesis co-advisor from SIIT, for his guidance and brilliant ideas that help me to have good progress on my thesis. He is always good-humoured, although he is very hard working. In the first year for my master course, some lessen is very difficult to understand, therefore he teaches my friend and me to clear in the content. Besides, he gives a great chance for my study tour in Japan Advanced Institute of Science Technology (JAIST) university, Japan. Because of this opportunity, I learn an education system and environment of JAIST.

I would like to express my sincere appreciation to Prof. Manabu Okumura, my thesis co-advisor from Tokyo Institute of Technology, for kindly providing useful suggestions and comments. There may be difficult to conduct meeting for my presentation, since we have to discuss via teleconference, but he dedicate the valuable time for me. I'm pleasured to be his student.

I am also indebted to Dr. Denduang Pradubsuwun, who was my advisor in bachelor degree. Since the first class until now, I appreciate all his contributions of time, ideas, valuable advices and suggestions to me. While I am sadness, he will solace me to relax and tell that "Do not be serious". In addition, he is the first person who suggests me to study in master degree, therefore I receive this scholarship. He introduced me to Asst. Prof. Dr. Cholwich Nattee. Without a good suggestion of Dr. Denduang, I would not be successful.

This research is financially supported by Thailand Advanced Institute of Science and Technology - Tokyo Institute of Technology (TAIST-Tokyo Tech), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology (Tokyo tech), Sirindhorn International Institute of Technology (SIIT), and Thammasat University (TU).

Furthermore, my graduation would not be achieved without the help of Mr. Konlakorn Wongpatikaseree, who gives the valuable ideas for me and devotes time to help collect data for my experiments. Moreover, he always suggests a good solution to solve the problems. Whether happiness or sadness, he will be with me and encourages me to better feeling.

Lastly, I wish to thank my family including my cousin and all friends for their warm understanding encouragement and huge support. Although they cannot help me in my work, they always cheer up me when I am disappointed. They try to do everything that makes me get better. Finally, I would to tell them that I apologize if I do mistake. Last but not least, I would like to thank everyone who has not mentioned for your experience, perspective and morale.

Abstract

Text Analysis for Thai Text-to-Speech Synthesis System on Embedded Device

August 2010

by

Arunee Ratikan

B. Sci, Science and Technology Faculty, Thammasat University 2007

A Text-To-Speech (TTS) system is a computer system which can synthesize speech from a given text. Nowadays, many Thai TTS systems have been developed on devices with powerful computing resources such as a personal computer. However, porting these systems to a resource-limited device such as a mobile phone is not an easy task. A TTS system generally composes of three main components: text analysis, prosodic analysis, and speech synthesis. In this thesis, we focus on only text analysis module for Thai TTS on embedded devices. Basically, the text analysis module consists of two routines: word segmentation and grapheme-to-phoneme (G2P) conversion. This module prepares information from a given text to the speech synthesizer module in order to generate speech output. For the thesis work, we conduct two main sections: a comparison of the existing Thai text analysis algorithms section and a proposed technique section. The first section aims at studying several techniques for Thai text analysis in order to choose the most appropriate technique for mobile devices. This experiment has three sub experiments: word segmentation experiment, G2P conversion experiment, and combined experiment. We found that the Longest Matching technique for word segmentation and the dictionary-based technique for G2P conversion are suitable for the embedded devices. The second section aims at improving the text analysis part especially the G2P conversion of a Thai TTS system on mobile devices. The output of the G2P conversion plays a significant role of the overall TTS system quality, since it connects to the speech synthesizer module when synthesizing the output speech. In this experiment, we propose a new G2P approach using the multiple-sized unit dictionary and set of conversion rules. It can generate an unknown word's pronunciation that the dictionary-based technique is unable to generate.

Table of Contents

Chapter	Title	Page
	Signature Page	i
	Acknowledgment	ii
	Abstract	iv
	Table of Contents	v
	List of Figures	vii
	List of Tables	viii
1.	Introduction	1
	1.1 Background Thai Text-to-Speech (TTS) system	1
	1.2 Motivation	2
	1.3 Objective	2
	1.4 Overview Thesis	3
	1.4.1 The comparison of the existing Thai text analysis algorithms section	3
	1.4.2 The proposed technique section	3
2.	Theoretical Backgrounds and the Related Works	5
	2.1 Characteristics of the Thai Language	5
	2.1.1 Consonant (C and S)	6
	2.1.2 Vowel (V)	8
	2.1.3 Tone (T)	9
	2.2 Word segmentation and G2P conversion problems	10
	2.2.1 Word segmentation problems	10
	2.2.2 G2P conversion problems	11
	2.3 Related work	12
	2.3.1 Word Segmentation	12
	2.3.2 Grapheme-to-Phoneme Conversion	16

3. Comparison of Text Analysis for Thai Text-to-Speech (TTS) Application on Mobile Devices 22
3.1 The existing text analysis techniques for mobile devices 22
3.2 Resources 24
3.3 Experiments and results 26
3.3.1 Word Segmentation Experiment 26
1. Comparing the Longest Matching and Maximal Matching techniques using the LEXiTRON dictionary 29
2. Comparing the Longest Matching and Maximal Matching techniques using the BEST_WordList dictionary 30
3. Investigating the affect of the size of a pronunciation dictionary 32
3.3.2 Grapheme-to-Phoneme conversion Experiment 33
3.3.3 Comparing the combination of word segmentation and G2P conversion techniques with the Syllable-based G2P conversion technique 38
3.4 Conclusion 39

4. Combining a Pronunciation Dictionary of Multiple-Sized Units and a Rules-Based Technique for Thai G2P Conversion 40
4.1 A problem in the dictionary-based technique 40
4.2 Architecture of a proposed technique 41
4.3 The techniques in new G2P approach 42
4.3.1 Pronunciation dictionary 42
4.3.2 Syllable Segmentation 42
4.3.3 G2P Generator 44
4.4 Experiments and Results 47
4.5 Conclusion 50

5. Conclusion and Future direction 51

Reference 53
List of Publications 55

List of Figures

Figure	Title	Page
1.1	Text-to-speech process diagram	1
2.1	Structure of Thai writing	5
2.2	Architecture of PGLR technique	17
3.1	Comparison architecture	23
3.2	Word Segmentation Experiment	26
3.3	A result of the Longest Matching technique	29
3.4	A result of the Maximal Matching technique	29
3.5	F-measures and OOV rates when varying the size of a dictionary	33
3.6	G2P conversion Experiment	33
3.7	Combined Experiment	38
4.1	The proposed technique	41
4.2	The steps of G2P Generator	44
4.3	Mapping Grapheme-to-Phoneme	46
4.4	The Longest Matching technique and proposed technique	49

List of Tables

Table	Title	Page
2.1	The phonetic symbol of Thai consonant [2]	6
2.2	The phonetic symbol of Thai cluster consonant [2]	8
2.3	The phonetic symbol of Thai vowels [2]	9
2.4	Comparison of the existing word segmentation algorithms	15
2.5	The example of rule patterns	18
2.6	Comparison of the existing G2P conversion techniques	21
3.1	Amount of data	25
3.2	Dictionary resources	26
3.3	The comparison of the performances of the Longest Matching and Maximal Matching techniques using the LEXiTRON dictionary	30
3.4	The comparison of the performances of the Longest Matching and Maximal Matching techniques using the BEST_WordList dictionary	31
3.5	The different word length of the LEXiTRON and BEST_WordList dictionary	31
3.6	The comparison of the OOV rates between the LEXiTRON dictionary and the BEST_WordList dictionary	31
3.7	The comparison of G2P conversion accuracies between the dictionary-based technique and the PGLR technique using TsynC dictionary	35

3.8 The performances of baseline G2P approaches tested on the 15,907 unique words 37
3.9 G2P accuracy when increasing the size of the word-level pronunciation dictionary 37
3.10 The comparison between the combination of word segmentation and G2P conversion, and the syllable-based G2P conversion 39
4.1 The example of word level and syllable level pronunciation dictionary 42
4.2 Tagging a unknown syllable to functional structure 45
4.3 Thai tonal modulation [19] 46

Chapter 1

Introduction

1.1 Background Thai Text-to-Speech (TTS) system

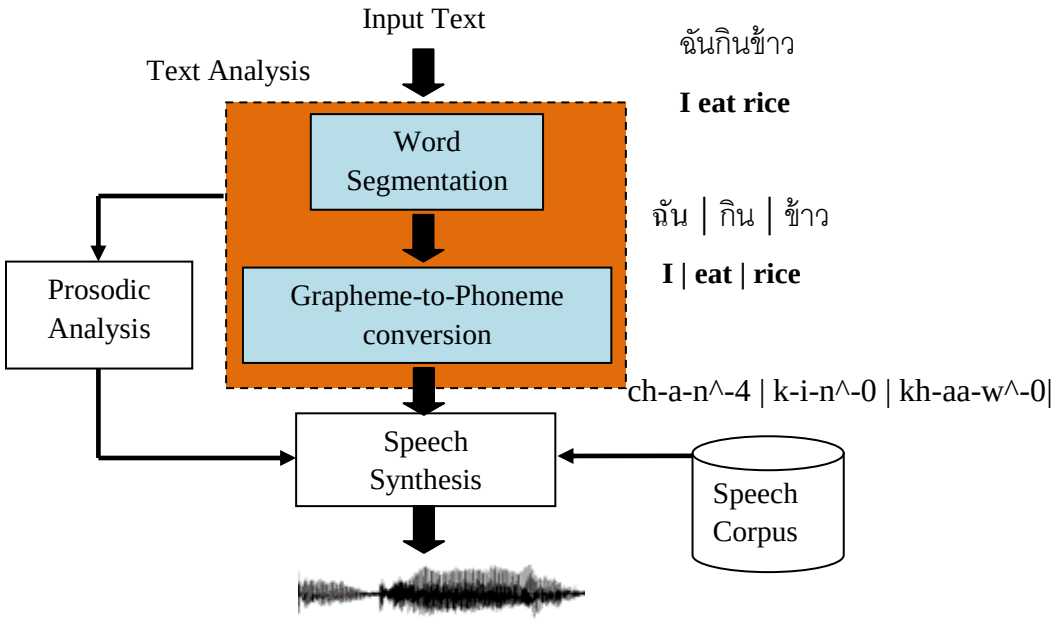

Figure 1.1 Text-to-speech process diagram

Figure 1.1 illustrates the overall process of a Thai TTS system. This system consists of three main parts: text analysis, prosodic analysis and speech synthesis. When an input text is entered into the TTS system as a string of characters, the text analysis module first analyzes the input sentence to find the corresponding pronunciation that is represented by a sequence of phonemes and tonal markers. The number 0-4 in the pronunciation represents five Thai tone levels that are mid, low, falling, high, and rising. Since Thai text is continuously written without using white space to separate words similar to Chinese, Korea, and Japanese, the input sequence needs to pass to the word segmentation module to generate a sequence of words before further processing. It is not an easy task to segment a string of texts into words according to the ambiguities. After a sequence of word is obtained, Grapheme-to-Phoneme

(G2P) conversion module is used in order to find the pronunciation of each segmented word. The speech synthesize module then generates an output wave file using the phoneme sequences from the text analysis module. To make the synthesized speech sounds more natural, the prosodic analysis module is used to predict the prosodic features (e.g. duration and an F0 contour) of the input sentence. The speech synthesis module utilizes these features when synthesizing the output speech.

1.2 Motivation

Recently, mobile phones become widely available and they make our daily life more convenient. We are able to access and share information easily. Voice communication is not the only main function of mobile phones. Many other applications are also available e.g. sending/receiving messages, route navigation, etc. A TTS technology also helps increase the usability of the mobile devices for people with visual disability as well as normal users. For blind people, it is not possible for them to read any text such as messages, electronic mails, or contents on web sites. The TTS technology therefore helps them communicate with other people and gets the information from text messages and the Internet. For normal people, the TTS technology can provide more convenient environment since a user can do something else while listening to read-aloud texts from a TTS engine. However, most Thai TTS systems are working only on personal computers (PCs) because of the adequate computational power and memory. Only a few Thai speech synthesizers are available on mobile devices and embedded systems. Two main problems can be considered for Thai TTS systems on these devices. Firstly, it may be difficult or impossible to port Thai TTS systems using the sophisticated techniques for generating speech output to the mobile devices, since those techniques require more computational time and a large memory space. Secondly, if it can be installed in embedded systems, it may not conduct the synthesis process within a suitable time period. Consequently, the techniques used in Thai TTS systems have to be analyzed by comparing advantages and disadvantages of each technique. We can then selecting the appropriate techniques to be developed to reduce the computing power consumption and memory.

1.3 Objective

Porting the TTS system to limited-resource systems such as mobile devices is not an easy task. The accuracy of the overall processing is reduced. We focus on a text analysis module of a Thai TTS system on mobile devices, thus the goals of this thesis are:

- To analyze and compare the existing algorithms for text analysis.
- To choose a suitable algorithm for Thai text analysis on mobile phone.
- To improve the accuracy of the G2P conversion of a Thai TTS system on a mobile device.

1.4 Overview Thesis

In this thesis, we conduct two main sections to investigate and solve this problem. The first section focuses on the comparison of the existing Thai text analysis algorithms. This section is to select a suitable technique for text analysis on the resource-limited devices by analyzing and comparing the existing text analysis techniques. The second section explains the proposed technique. We aim at improving the text analysis part especially the G2P conversion of a Thai TTS system on mobile devices. The G2P conversion is utilized by the speech synthesizer when synthesizing the output speech, consequently the accuracy of the input text's pronunciation is important and also affects the quality of the overall TTS system. More details of each section are shown below.

1.4.1 The comparison of the existing Thai text analysis algorithms section

We conduct three sub experiments to identify a suitable algorithm. The first sub experiment, word segmentation experiment, compares the performances of two widely-used algorithms for Thai word segmentation i.e. Longest Matching and Maximal Matching techniques. The algorithms are compared using the F-Measure [1], processing time and size of the program's footprint. The second sub-experiment, a Grapheme-to-Phoneme (G2P) conversion experiment, aims at evaluating existing techniques for G2P, which generates a pronunciation of an input word. We compare two techniques: dictionary-based and rule-based (Probabilistic Generalize LR) techniques. Their results are compared using the following criteria: accuracy at word, syllable, and phone level, processing time, and size of the program. Finally, the third sub experiment, the combined experiment, focuses on evaluating the overall performance of the text analysis module. We compare the best techniques from the first two experiments with the syllable-based G2P conversion which uses a rule-based technique to segment an input text into syllables and convert them into the phone sequence. For the TTS system, word segmentation may not be an essential component since the goal of the text analysis module is to accurately generate a sequence of phonemes from a given text. Therefore, syllable segmentation could be used instead. The evaluation metrics of this experiment are the same as the ones used in the second sub experiment.

1.4.2 The proposed technique section

In the previous section, the word segmentation and G2P conversion experiments are suitable technique for text analysis. However both techniques are the dictionary-based approach; therefore, the main problem is that the G2P conversion experiment cannot find the pronunciation's unknown word. This problem affects overall performance system, when the text analysis module combines to the speech synthesis module. To solve the problem, we propose a new G2P approach which utilizes both a pronunciation dictionary of multiple-sized units (i.e. word and syllable) and a set of conversion rules. To handle an unknown word, we segment it into syllables by a rule-based technique in order to increase the opportunity of

getting the pronunciation of a syllable in the multiple-sized unit dictionary. Moreover, an unknown syllable is addressed by a G2P generator which uses a set of rules to map the characters in the syllable into phones according to the Thai syllable structure and pronunciation rules. For the measure, we use accuracy at word, syllable, and phone level, processing time, as well as the size of program.

The rest of this thesis is organized as follows. Chapter 2 describes related backgrounds and related works. Chapter 3 describes the comparison of the existing text analysis techniques and result. Chapter 4 presents how to solve the pronunciation of unknown word problem by a new G2P approach. Chapter 5 concludes the thesis work and our future directions.

Chapter 2

Theoretical Backgrounds and the Related Works

This chapter describes the background and characteristic of Thai language, the problems of word segmentation, grapheme-to-phoneme (G2P) conversion, as well as the related works on TTS system. Firstly, we describe more details related to Thai writing structure and each of the components of language that is not similar to Western languages such as English, Spanish, German, etc. Secondly, we indicate the word segmentation and G2P conversion problems that are caused by Thai Characteristics of the Thai language. Finally, we review the related works that have been proposed to solve these problems.

2.1 Characteristics of the Thai Language

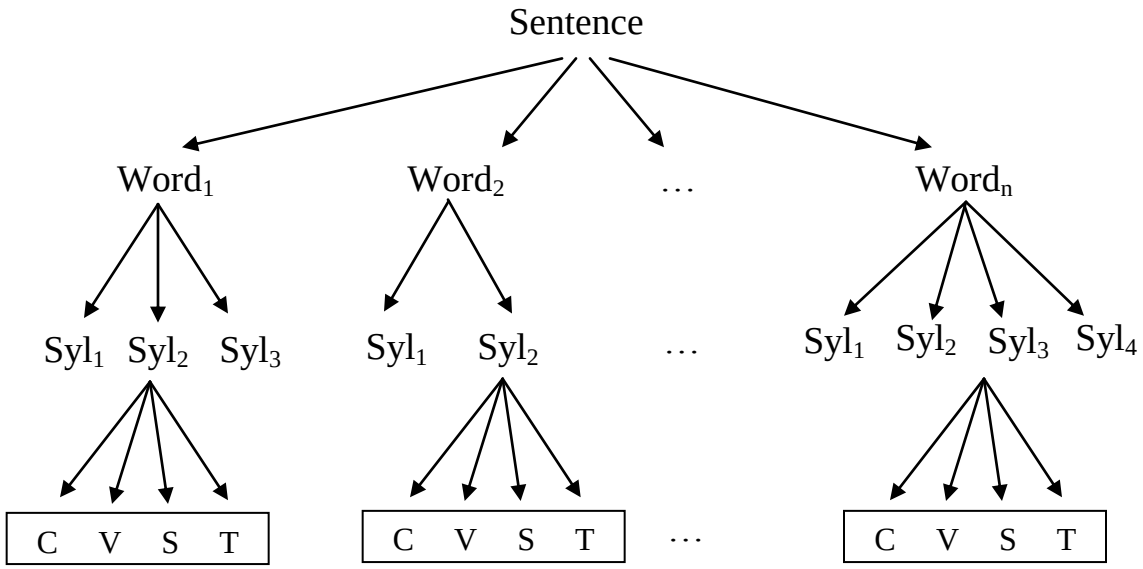

Figure 2.1 Structure of Thai writing

Figure 2.1 indicates a representation of Thai writing system that composes of four levels: sentence, word, syllable, and alphabet (character). Generally, in newspapers, articles, books, etc, Thai text is written in a form of sentence sequence as a paragraph. The space is sometimes used for separating each sentence in paragraph. One sentence is represented by a series of words. Each word is derived from a sequence of syllables and characters. Thai

characters correspond to their sound, called phoneme. Basically, the structure of a Thai syllable consists of at most four components: initial consonant (C), vowel (V), final consonant (S), and tone (T). A pronunciation of a Thai syllable is represented in the form of C V S T. More details of each component are explained below.

2.1.1 Consonant (C and S)

Thai language composes of 44 characters but it is represented by only 33 phonemes as shown in Table 2.1. Each character is used as initial and final consonants. The initial consonant contains 22 phonemes and is placed in front of vowel, whereas the final consonant consists of eight phonemes and has a position at behind the vowel. Most of all characters can be used as both initial consonant and final consonant except "ห", "อ", and "ฮ" characters. These can be only initial consonant. Thai sound can be classified by three criteria: manner of articulation, voice or voiceless, and place of articulation. Thus the phoneme of each character is not represented one character by one phoneme. For example, "ซ, ศ, ษ, ส" initial consonants are mapped in one phoneme as "|s|". We only use 21 phonemes to symbolize consonant (|k|, |kh|, |ng|, |c|, |ch|, |s|, |t|, |j|, |d|, |th|, |b|, |p|, |ph|, |f|, |m|, |n|, |r|, |l|, |h|, |w|, |z|) but nine consonants can be used as final consonants. Since the characters "ฃ" and "ฅ" do not occur any Thai syllable, they are not defined in the phonetic system.

Table 2.1 The phonetic symbol of Thai consonant [2]

Consonant	Phoneme		Consonant	Phoneme	
	Initial	Final		Initial	Final
ก	|k|	|k|	บ	|b|	|p|
ข, ค, ฆ	|kh|	|k|	ป	|p|	|p|
ง	|ng|	|ng|	ผ, พ, ภ	|ph|	|p|
จ	|c|	|t|	ฝ, ฟ	|f|	|p|
ฉ, ฌ, ช	|ch|	|t|	ม	|m|	|m|
ซ, ศ, ษ, ส	|s|	|t|	ร	|r|	|n|
ญ, ย	|j|	|j|	ล, ฬ	|l|	|n|
ฎ, ด	|d|	|t|	ว	|w|	|w|
ฏ, ต	|t|	|t|	ห, ฮ	|h|	-
ฐ, ฑ, ฒ, ถ, ท	|th|	|t|	อ	|z|	-
ณ, น	|n|	|n|			

Two characters can be combined into a cluster consonant such as “กร”, “ปร”, “ชร”, “อย”, etc. as shown in Table 2.2. The cluster consonants can be divided into four groups: [2]

1. True Cluster

   Two initial consonants are pronounced as one phoneme, for example “ขร” is pronounced as “|khr|”. The cluster consonant “khr” is combined the consonant “ข” (“|kh|”) with “ร” (“r”).

2. Pseudo Cluster

   Although, it is a cluster consonant, it is pronounced as single consonant. Pseudo cluster may be pronounced following the initial consonant, for example, “**จริ**ง” (believe) is pronounced as “|c-i-ng^-0|” which is the phoneme of the first consonant “จ”. A pseudo cluster may be pronounced as other single consonant such as combination of “ท” (“|th|”) and “ร” (“r”). Their sounds are changed to the other phonemes. For example, “แ**ทร**ก” (insert) is pronounced as “|s-xx-k^-2|”.

3. Parallel Consonant Character

   This cluster differs from the true and pseudo cluster since each character in cluster will contain vowel depended on each syllable. For instance, “**ปร**ปักษ์” (adversary) is pronounced as “|**p-@@**-z^-0|**r-a**-z^-3|p-a-k^-1|” or “สุ**จริ**ต” (honest) is pronounced as “|s-u-z^-1|**c-a**-z^-1|**r-i**-t^-1|”.

4. Leading Consonant Character

   There are only two cases. The case that “ห” (“|h|”) is initial consonant such as “หญิง” (female), “หมอ” (doctor), “หนู” (mice) etc. The case that “อ” (“|z|”) is initial consonant. There are only four syllables in this case i.e. “อย่า” (do not), “อยู่” (live), “อย่าง” (kind), “อยาก” (need). The pronunciation follows the second consonant, for instance “ห**ม**อ” is pronounced as “|**m**-@@-z^-4|” or “อ**ย**าก” is pronounced as “|**j**-aa-k^-1|”

Furthermore, Thai language initial consonants can be divided into three classes following original sound of each character without tone modulation. These classes provide the advantage to help predict the tone.

1. High class consists of 11 characters: ผ, ฝ, ถ, ฐ, ข, ฃ, ศ, ษ, ส, ห, ฉ
2. Middle class contains 9 characters: ก, จ, ด, ต, ฎ, ฏ, บ, ป, อ
3. Low class composes of 24 characters: ค, ฅ, ฆ, ง, ช, ซ, ฌ, ญ, ฑ, ฒ, ณ, ท, ธ, น, ย, ร, ว, ม, พ, ล, พ, ฟ, ภ, ฮ

Table 2.2 The phonetic symbol of Thai cluster consonant [2]

Cluster Consonant	Phoneme
True Cluster	ปร |pr|, ตร |tr|, กร |kr|, พร |pr|, ทร |thr|, คร |khr|, ขร |khr|, กล |kl|, พล |phl|, ผล |phl|, คล |khl|, ขล |khl|, กว |kw|, คว |khw|, ขว |khw|
Pseudo Cluster	ทร |s|, จร |c|, ซร |s|
Parallel Consonant Character	กล |k-a-z^-1|l-|, ปร |p-a-z^-1|l-|
Leading Consonant Character	อย |j|, หย |j|, หร |r|, หล |l|, หว |w|, หง |ng|, หญ |j|, หน |n|, หม |m|

2.1.2 Vowel (V)

Table 2.3 shows the representation of the 28 Thai vowels. They are classified into two main groups: short vowel and long vowel. One vowel only represents one phoneme while one consonant can represent more than one phoneme. The short vowel can be divided into three types: monophthong, diphthong, and vowel letter, while the long vowel consists of the first two types. The monophthong is a single uncompounded vowel sound. The diphthong is a vowel sound formed by the combination of two vowels in a single syllable. Besides the vowel letter is special short vowel sound composing of four vowels. They are transformed from "–ะ" and final consonant.

For instance, "–ำ" is combination of "–ะ" and "ม" as "|a-m|"

"ไ–" is combination of "–ะ" and "ย" as "|a-j|"

"ใ–" is combination of "–ะ" and "ย" as "|a-j|"

"เ–า" is combination of "–ะ" and "ว" as "|a-w|"

Table 2.3 The phonetic symbol of Thai vowels [2]

Type	Short vowel		Long vowel	
	Grapheme	Phoneme	Grapheme	Phoneme
Monophthong	–ะ	|a|	–า	|aa|
	–ิ	|i|	–ี	|ii|
	–ึ	|v|	–ื	|vv|
	–ุ	|u|	–ู	|uu|
	เ–ะ	|e|	เ–	|ee|
	แ–ะ	|x|	แ–	|xx|
	โ–ะ	|o|	โ–	|oo|
	เ–าะ	|@|	ออ	|@@|
	เ–อะ	|q|	เ–อ	|qq|
Diphthong	เ–ียะ	|ia|	เ–ีย	|iia|
	เ–ือะ	|va|	เ–ือ	|vva|
	–ัวะ	|ua|	–ัว	|uua|
Vowel letter	–ำ	|a-m|	-	-
	ไ–, ใ–	|a-j|	-	-
	เ–า	|a-w|	-	-

2.1.3 Tone (T)

Thai is a tonal language. There are five tone levels: mid, low, falling, high, and rising. These tones can be grouped into two levels: [3]

Level 1: Stable tone consists of three sounds: mid, low, and high.

Level 2: Tone composes of two sounds: falling and rising.

Sometimes, tone makes different meaning of word. This depends on the context. For example, “หา” has two meanings. It means “find something”, and it is used as an expression to show an alarm.

Based on Thai writing system, the tone marker and the pronounced tone are basically unmatched. This is quite ambiguous to predict a tone for each syllable. For example, the word “**เรียบ**” (smooth) has no tone marker but is pronounced with a falling tone which is normally

marked with a ‘ ้’ marker. To address this problem, we use the knowledge of three classes of initial consonant in Section 2.1.1, and live and dead syllables. The live syllable is an open syllable with a long vowel, or a closed syllable with live consonant ending, -m^,-n^, -ng^, -j^, and –w^, while the dead syllable is an open syllable with a short vowel, or a closed syllable with a dead consonant ending, -p^, -t^, -k^.

2.2 Word segmentation and G2P conversion problems

From the previous section, since Thai text is written continuously, there is no explicit marker between words in sentence such as white space, comma like English, Spanish, German, etc. It is not an easy task to segment a string of characters into words. Moreover, we have difficulty converting the string to phone directly, because one phoneme can be represented by more than one characters and some word has special pronunciation. The problems are explained below.

2.2.1 Word segmentation problems

There are two main problems of Thai word segmentation: ambiguous and unknown words.

Firstly, the problem of ambiguous words can be divided into two cases.

1. Type of this ambiguity makes a string to segment in several cases. Each case can be found in dictionary. Therefore, we cannot exactly decide which word is correct. For example, the word “ตากลม” is two possible segmentations “ตาก|ลม” (to expose wind), and “ตา|กลม” (round eyes). To solve this problem, we have to consider the context of the ambiguous words.

2. The word co-occurrence is a kind of ambiguous words. When the word is segmented, only one of the segmented result has meaning otherwise human will misunderstand. For instance, “นำมากลั่น” can divide into “นำ|มา|กลั่น|” (be refine) and “นำ|มาก|ลั่น|” (no meaning). In this case “นำ|มา|กลั่น|”, human can understand meaning. Based on Thai writing system, the case of “นำ|มาก|ลั่น|” cannot occur in real text since “ลั่น” cannot be located behind “มาก”. We need to include statistical information to correctly segment text.

Secondly, the problem is unknown word. This problem is often occurred when we use dictionary-based technique for segmenting the words. There are many causes, for instance, the word is rarely occurred in dictionary such as person name, abbreviation, technical term, and transliterated word. Since, presently, many new words are created that one word is combined

from another word or syllable. Another case is wrong writing. This case is error of user, for example user want to write a word “ประเทศไทย” (Thailand). Unfortunately, he/she forgets the vowel “ไ” that becomes“ประเทศทย”, accordingly if other modules have to use output of word segmentation such as the G2P conversion in TTS systems, the output of that module will be incorrect. Therefore, the statistical approach is applied to solve the unknown word problem that is quite successful.

2.2.2 G2P conversion problems

This module links between the speech synthesis module and the text analysis module in the TTS systems. It receives the output of word segmentation in order to find the corresponding pronunciation and also makes the pronunciation in form of C V S T. The speech synthesizer can directly utilize it in order to create the output speech. Nevertheless, mapping from Thai characters (or graphemes) to phones (or phonemes) is not straightforward since there are many ambiguous cases. Difficulties for Thai grapheme-to-phoneme can be described as seven main points: [2]

1. Ambiguous letter-to-sound: the same consonant can be pronounced differently. For example, ‘ฑ’ is pronounced as /d/ in “มณฑป” /m-o-n^-0|d-o-p^-1/, but is pronounced as /th/ in “มณฑา” /m-o-n^-0|th-aa-z^-0/.

2. Homograph: the same word can be pronounced differently. It depends on context around the word for example “เพลา” has two pronunciations: “/phl-a-w-o/” means “axle” and “/ph-e:-0/l-a:-0/” means “time”.

3. Length of Vowel: some words consist of short vowel according to grapheme such as “น้ำ” has sequence of phonemes “/n-a-m-3/” but it is regularly pronounced long vowel as “/n-a:-m-3/”.

4. Linking syllable: a final consonant can also be used as a linking syllable. For example, ‘ช’ in “รัชกาล” /r-a-t^-3|ch-a-z^-3|k-aa-n^-0/ is pronounced as both a final consonant /t/ and a linking syllable /ch-a-z^-3/.

5. Phonetic liaison: one character can be mapped into a combination of the final consonant and the initial consonant of its adjacent syllable. For instance, a cluster consonant ‘กร’ in “จักรี” /c-a-k^-1|kr-ii-z^-0/ is functioned as both a final consonant /k/ and an initial consonant /kr/ in the adjacent syllable.

6. Word boundary: some words in Thai can be differently segmentated since language has no explicit delimiter between words. For example, “ตากลม” may be segmented to “ตา| กลม” (round eye) or “ตาก| ลม (to expose wind)”.

7. Inversion order of a consonant and a vowel: some vowels may be used as the vowels of the next syllable, for example, ‘เส’ is a regular case that is pronounced as |s-ee-z^-4| but ‘เสน่ห์’ /s-a-z^-1|n-ee-z^-1/ is an irregular case that vowel ‘เ’ /ee/ becomes the vowel of the second syllable (starting from “น”) while ‘ส’ is the first syllable and is pronounced as /s-a-z^-1/.

2.3 Related work

There are many techniques that have been proposed to address the problems of Thai language. However, each proposed technique has different tradeoff. We discussed about them in this Section.

2.3.1 Word Segmentation

Word segmentation is an important problem since several applications require text in a form of word such as index retrieval, text-to-speech, machine translation, etc. Many algorithms have been proposed in order to solve this problem such as dictionary-based approach, rule-based approach, machine learning, and statistical-based. However, each technique has different advantage and disadvantage.

1. Dictionary-based approach

The concept of this algorithm is that it tries to segment the text into words by matching the word in a dictionary. It requires a large number of words in dictionary because if the size of dictionary is large, the opportunity of getting the word will increase.

Pooworawan [4] applied a dictionary to segment the word using Longest Matching technique. A series of string are scanned from left to right with dictionary. If there is more than one word, the algorithm selects the Longest Matching word. However, if the selected word causes the segmentation of the rest word to be stopped, the algorithm will backtrack to select the next longest word. However, this technique cannot solve the ambiguous and unknown words. For example, “ไป|หา|มเหสี” (go to see the queen) can be segmented as “ไป|หาม|เห|สี|” (“go to| carry| deviate| color|”). Obviously, this result is incorrect since this algorithm is greedy. In many cases, it can segment correctly as “โคลงเรือ” (a boat shakes). In their experiment, they compared the speed of word segmentation using dictionary with the technique using only rule-based technique. They found that the segmenting speed of dictionary-based approach is

faster than that of rule-based approach. Nonetheless, this requires more memory storing for dictionary.

Since Pooworawan's technique is a greedy-based technique, there are many chances that the result fails. Sornlertlamvanich [5] proposed Maximal Matching technique to solve the problem of Longest Matching technique. This technique is done like the Longest Matching technique but an input string is first segmented into all possible ways and the sentence containing the fewest words is chosen. For example, "ไป|หา|มเหสี" (go to see the queen) can be generated two possible cases based on the words in the dictionary i.e. "ไป|หา|มเหสี|" as the first case and "ไป|หาม|เห|สี|" as the second case. The first case is selected, since it contains three words while the other one contains four words. If more than one combination candidate has the same number of words, the algorithm basically uses the Longest Matching technique to choose the candidate. For example, "ตากลม" (to expose wind, or round eyes) can be segmented into two cases: (1) "ตาก|ลม" (to expose wind) and (2) "ตา|กลม" (round eyes). The algorithm chooses (1) as the segmentation result. This proposed technique can solve the ambiguous words more precisely than the Longest Matching technique. At the same time, it takes longer time for segmenting than Pooworawan's technique.

2. Rule-based approach

This approach constructs a collection of rules in term of regular expression. These rules are used as pattern of word. Therefore, they must correspond to Thai writing. Nevertheless, it is difficult to create the rules that cover various cases in the Thai writing system. It needs extensive linguistic knowledge.

Thairatananon [6] proposed the rule-based approach for syllable segmentation. She conducts a set of rule following Thai syllable characteristic and develop by PL/I language. A set of rules consists of five groups of Thai characters: consonant, vowel, tone marker, numeral, and special character. Since some syllables are not matched in any rule, it is kept in a file. The proposed rule is less complex but it is not flexible for changing or adding the rule. The research obtained more than 85% of syllable segmentation accuracy. Thairatananon's work is quite not popular, because it is not easy to edit the rules and the accuracy is not so high.

Sawamipak [7] presented a word segmentation technique under UNIX operating system. A proposed set of rules is used together with a dictionary to segment an input text into a sequence of words. The set of rules consisted of 43 rules in a form of a regular expression. A final consonant is not included in these rules. If the rules contain a final consonant, a segmentation result might be wrong since an initial consonant may be misinterpreted as a final consonant of the previous word. The input is first analyzed by rules. Then it was checked against a dictionary. Two criteria were used to measure the system performance i.e. syllable-

level accuracy and word-level accuracy. This approach achieved 98.11% word accuracy and 99.67% syllable accuracy as evaluated on 17 domains such as newspapers, magazines, text book, etc.

3. Machine learning and statistical-based approach

The idea of this approach is to extract characteristics or features of words from the collected text, and apply machine learning technique to generate models for select the most suitable segmentation. This approach requires a large memory and storage for the training set and an extensive computational power.

Charoenpornsawat [8] proposed a technique for word segmentation using feature-based method. The features help solving the ambiguous segmentation problem and the unknown word problem in Thai. This research compared two supervised learning techniques: RIPPER and WINNOW. The RIPPER can construct a set of rules automatically from examples. These rules are in terms of if-then-else form. The advantage of RIPPER has four points.

- It is easy to understand and has performance better than Neural Network and Decision tree.
- Time processing is faster than the other Rule learning algorithms in case of many examples for learning.
- The rules can be determined by user, thus the rule outputs have more efficiency.
- The RIPPER can use set-valued attribute, therefore rule output will concise.

The WINNOW is a weighted-majority learning algorithm and learns by examples which are fed into this algorithm. The WINNOW learns to form a network that consists of nodes and weights. The advantage of WINNOW algorithm has two main points: fast learning because it is an incremental algorithm and handling irrelevant features. In the experiment, ORCHID corpus is used that contains 25,000 sentences and the words in each sentence is segmented and tag with part of speech. From the corpus, all sentences are divided into 80% for training set and 20% for test set. In the case of ambiguous segmentation, the RIPPER technique obtained 85% segmentation accuracy while the WINNOW technique achieved 90% accuracy. For the case of unknown words, the segmentation accuracy is 70% for the RIPPER technique and 80% for the WINNOW technique.

Many techniques have been proposed for solving the word segmentation problems using this approach. Haruechaiyasak et al. [9] analyzed and compared various approaches to Thai word segmentation in order to evaluate the performance. They use ORCHID corpus to evaluate the performance. This corpus composes of 113,404 manually tagged words. They used statistical models estimated from training corpora using machine learning-based (MLB) approach. Four algorithms of the MLB approach are compared in terms of the segmentation

accuracy: Naive Bayes (NB), decision tree, Support Vector Machine (SVM), and Conditional Random Field (CRF). Thai word segmentation algorithms both of the Dictionary-based (DCB) approach and MLB approaches are compared. This research concluded that the accuracy of the Maximal Matching algorithm is higher than the Longest Matching algorithm, with 87.86% for the Maximal Matching algorithm and 87.55% for the Longest Matching algorithm respectively. This research suggested that a dictionary, used in the DCB approach, has to be large size in order to cover words. When the DCB approach compares to the MLB approach, F-Measure of both algorithms of the DCB approach is better than the NB algorithm and decision tree algorithm of the MLB approach, with 64.90% for the NB algorithm and 77.80% for decision tree respectively. However, the CRF algorithms are the highest accuracy by 95.38%. The disadvantage point of MLB approaches is their efficiency that depends on the characteristics of the document domain and the size of the training corpus. However, the MLB approach can handle unknown word and ambiguity problem.

Table 2.4 Comparison of the existing word segmentation algorithms

	Rule-based approach Duangkaew	Dictionary-based approach Choochart	Machine Learning approach Choochart
Accuracy	98.11%	87.55 - 87.86%	64.90 - 95.38%
Processing Time	Quite Short	Quite Short	Quite Long
Size	Quite Small	Depending on size of Dictionary	Quite Large

From literature review, we summarize the performance in term of accuracy, processing time, and size of program in Table 2.4. We compare three different techniques: rule-based, dictionary-based, and machine learning approach. The dictionary-based and machine learning approaches use the same as data set. Each method has different advantages and disadvantages; however no technique can correctly segment Thai text. Although, both the rule-based technique and the dictionary-based technique (the Longest Matching and Maximal Matching algorithms) are less complex, yield high accuracy, and do not require a large amount of memory, they cannot solve many ambiguous word combinations and the unknown word problem. Nevertheless the machine learning approach can handle word segmentation problems more effectively, the model is quite large and requires more computational time for segmentation.

2.3.2 Grapheme-to-Phoneme Conversion

The technique to generate the pronunciation of each word or syllable has been developed in many years. Since there are many cases in the difficulty of Thai grapheme-to-phoneme conversion, many techniques have been proposed in order to overcome those challenging problems for researchers. We can group the proposed technique into three main types:

1. Dictionary-based approach

Mostly, this technique straightforward looks up a given word in dictionary. Speed of this approach depends on data structure, which all words in dictionary are stored, and searching algorithm.

Luksaneeyanawin [3] used the dictionary-based technique to find the corresponding pronunciation of each word in a Thai TTS system. This research found that a large dictionary cannot solve the problem of the pronunciation of an unknown word.

2. Rule-based approach

The idea of this approach is to conduct the conversion based on the patterns from Thai writing structure. Sometimes, the rule may be in a form of context free grammar (CFG) or regular expression. These rules are mapped with a word input then covert graphemes in the word to their phonemes.

Mittrapiyanuruk et al. [10] discussed issues in Thai Text-to-Speech synthesis. For the G2P conversion, they used a set of letter-to-sound rules in order to handle the unknown words' pronunciation. To address this problem, there are two processes. The unknown word is divided into syllables using regular expressions and then each syllable is converted into a phoneme string using a simple G2P conversion mapping. This approach has two strong points: the flexible modification of the rules and the speed of the processing time. This research does not report accuracy of letter-to-sound rules. Nevertheless, we believe that the accuracy of G2P conversion may not be high, because only a set of rules is used to convert an unknown word to its phone sequence.

Tarsaku et al. [2] proposed a Probabilistic GLR (PGLR) parser technique for the G2P conversion. The PGLR technique is used to get the corresponding pronunciation of an input word. This technique is a rule-based technique and is able to handle any word. Moreover, the advantage of this technique is that it can capture both the global and local context in the probabilistic model. When an input string comes into the system, a GLR parser generates all possible parsed trees that are mapped grapheme to phoneme using the rules denoted in a context-free grammar (CFG). Then the probability of each tree is calculated from a GLR table.

The one with the highest probability is selected and then the graphemes are converted to phonemes by using table look-up as shown in Figure 2.2.

The CFG rules are designed following Thai syllable structure, as follow.

<Syl_Type> -> <ini-cons> V_x <fin-cons>

where <Syl_Type>, <ini-cons>, and <fin-cons> are non-terminal node. They represent a type of syllable, initial consonant, and final consonant respectively. V_x is a terminal node of vowel.

Not all Thai words are monosyllable, the CFG rules have to cover all possible Thai word characteristics. The equations (2.1)-(2.3) show examples of CFG rules.

<Word> → <Syllable> (2.1)

<Syllable> → <Syllable><Syl_Type> (2.2)

<Syllable> → <Syl_Type> (2.3)

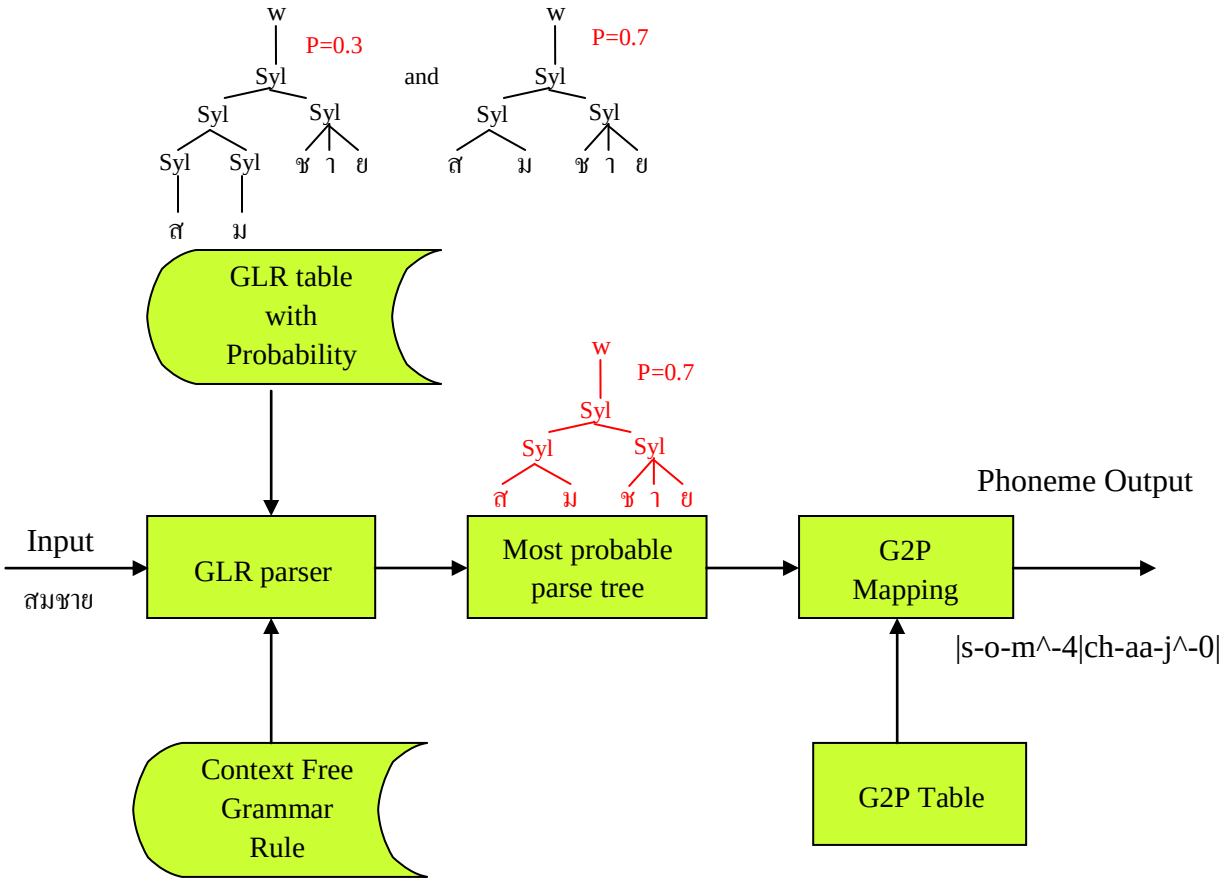

Figure 2.2 Architecture of PGLR technique

This work use the LEXiTRON dictionary, which is word-pronunciation dictionary and contains 23,000 entries. Around 18,400 entries, it is set as training set and the rest entries are test set. The PGLR approach obtained 72.87% word accuracy for an exact match and 90.44% word accuracy when the vowel length is ignored. These numbers are higher when compared to the results from rule-based and decision tree approaches. Because this system uses both probability and CFG, it helps increase the accuracy of G2P conversion.

Chinathimatmongkhon et al. [11] divided text analysis into two steps: syllabification and phoneme conversion. Syllable patterns that correspond to a set of Thai writing rules are used to segment an input sentence into syllables. When a text string is entered, each character in the string is substituted with its corresponding alphabets symbols. Then, these symbols are looked up in a pattern list using the Longest Matching technique. The result of matching is checked if it associates with any rules. After that the segmented syllable is transformed using a list of syllable-pronunciation pairs, generated from all possible Thai character combinations compatible with Thai pronunciation rules.

We present the examples of rules for converting syllable to its phone in Table 2.5. If any syllable matches with syllable pattern, its pronunciation will be converted by pronunciation pattern. For instance, the syllable "เนตร" corresponds to เXตร in syllable pattern, thus it is pronounced as "neet" following "Xeet" in pronunciation pattern. Where "X" is initial consonant of "น", "ee" is vowel of "เอ", and "t" is final consonant of "ตร".

Table 2.5 The example of rule patterns

Syllable pattern	Pronunciation pattern
Xุตร	X**u**t
แXตร	X**xx**t
โXติ	X**oo**t
เXตร	X**ee**t
เXTอะ	X**@**
เXTา	X**aw**

3. Statistical based approach

This approach is based on the data which come from collecting the statistics in corpus. This information is important knowledge for learning of machine. It requires a corpus that is tagged a correct boundary of syllables. Performance of this approach depends on the number of words and syllables in the training data.

Choltimongkol et al. [12] proposed an approach that uses a statistical model and decision trees to predict phones from an input string. Tones prediction is performed based on Thai writing rules. Decision trees are automatically trained from a pronunciation dictionary to handle the text-to-phone problem in Thai. The training set consists of 22,818 words from a Thai LEXiTRON pronunciation dictionary, with 2,535 words, which is every tenth word in the dictionary, held-out for testing. The result of the experiment showed that the decision trees together with an n-gram model can solve the problem better than a rule-based technique. Accuracies of the decision trees were about 94.47% on letter accuracy and 62.17% on complete word accuracy.

If a syllable boundary is obtained, the accuracy of most syllable pronunciation can be predicted effectively. To make a syllable boundary, it is not easy task, because it requires the specialist in linguistic knowledge.

Aroonmanakun [13] et al. proposed a way to produce sequence of phoneme by considering the caused problems that were ambiguity of syllable and word. This work thought those problems can be solved by using character level and syllable level. This work used trigram model of syllables to disambiguate syllable segmentations. This model composed of language model and pronunciation model. The language model, the highest probability of syllable segmentation is selected then it will be combined into words. The pronunciation model, the probability of pronunciation is judged from a word-pronunciation pair corpus that aligned between syllable and sequence of phone. The corrected pronunciation of each syllable was selected from the result of word segmentation and the statistical information of pronunciation. This research uses two test sets: general texts and geographical names. The first data set contains 18,388 words that come from a newspaper, because they need to evaluate the accuracy of romanization for general texts. The second data set contains 990 Thai geographical names. It is extracted from the Royal Institute's books. Their system achieved 94.44% accuracy of corrected general text and 97.07% accuracy of corrected names.

However, adding the person names in training corpus may increase the accuracy. Therefore, the improvement of accuracy depends on a number of words in corpus that affects the computer memory requirement.

Charoenpornsawat et al. [14] proposed an example-based G2P conversion for Thai (EBG2P). This system consists of five processes: syllable segmentation, G2P alignment, syllable selection, closest matching syllable, and syllable modification. Firstly, a language independent technique is applied in order to segment an input word into syllables. Secondly, for G2P alignment, they create a G2P mapping table that includes all possible alignments of graphemes and phonemes, and cross alignments of consonants and vowels. Thirdly, in the syllable selection process, the Longest Matching technique is used to select the syllable pronunciation with the highest frequency from a training corpus. Fourthly, closest matching syllable, if there are no matched syllables in the training corpus, the algorithm will find the closest matching syllable by looking at an editing distance in closest matching syllable step. Finally, when the closest matching syllable is found, a substituting technique is used in order to modify a phone of an unmatched letter. The EBG2P system achieved 80.99% on word accuracy and 94.19% on phone accuracy which significantly outperformed previous approaches for Thai.

The advantage of this proposed is the high accuracy since it uses the edit distance technique to find the closest matching syllable. Nonetheless, it is a cause of long processing time.

Table 2.6 summarizes the technique for G2P conversion from the literatures. Presently, G2P conversion mostly uses statistical based approaches, such as decision tree and example-based technique. Those techniques achieve high accuracy; however, they require more memory and long processing time. Therefore, they are difficult to be ported on to embedded devices such as mobile phones. On the other hand, the dictionary-based and the rule-based techniques need less storage and processing power when comparing to the machine learning and probabilistic approaches. The percentage of the PGLR and decision tree techniques, shown in the table, is tested with the LEXiTRON dictionary, but the number of words in test set is different.

Table 2.6 Comparison of the existing G2P conversion techniques

	PGLR Tarsaku P.	Dictionary-based Luksaneeyanawin S.	Decision Tree Choltimongkol A.
Word accuracy	72.87%	100% (not include unknown word)	62.17%
Syllable accuracy	-	100% (not include unknown syllable)	-
Phone accuracy	-	100% (not include unknown phone)	94.47%
Processing time	Quite Long	Short	Quite Long
Size	Quite Small	Depending on size of Dictionary	Quite Large

Chapter 3

Comparison of Text Analysis for Thai Text-to-Speech (TTS) Application on Mobile Devices

Basically, the text analysis process consists of two routines: word segmentation and grapheme-to-phoneme (G2P) conversion. Developing the Thai text analysis on mobile is a difficult task because of the limited processing power and memory on these devices. We aim to select the suitable text analysis technique for Text-to-Speech (TTS) system on mobile device, while the accuracies, processing time, and size are acceptable in real time system. In this chapter, we present the objective and architecture of this experiment, and then explain the algorithms used in experiments. Next, we describe issues related to the resources. After that we show the results and discussions for each experiment. Finally, we summarize all the experimental results.

3.1 The existing text analysis techniques for mobile devices

There are many techniques for text analysis, but not all the techniques work well on the limited devices. Each technique has different advantages and disadvantages. Therefore, we first compare and select the possible and appropriate techniques in order to run on mobile devices. This experiment aims at analyzing and comparing the existing Thai text analysis algorithms.

To find the appropriate text analysis algorithms for Thai TTS system on mobile devices, we follow the steps below.

1. Review Thai word segmentation and G2P conversion techniques.
2. Select the candidate techniques for detailed computing comparison.
3. Compare the techniques by accuracy, processing time and required memory size

We conduct three experiments following the diagram in Figure 3.1. The first experiment is word segmentation experiment. The second experiment is G2P conversion experiment, and the third is the experiment combining both previous steps. This follows the process for text analysis in an ordinary TTS system.

When a user enters an input sentence into the system, the first step is the word segmentation module that separates the input sentence into words, since Thai writing mostly is a continuous string without white space. In the first experiment, word segmentation experiment, we compare two word segmentation algorithms: Longest Matching and Maximal Matching techniques. Then, each word is converted into its corresponding pronunciation that is represented by a phone string in the G2P conversion module. In the second experiment, we compare the performance of the Probabilistic Generalize LR parser (PGLR) technique with the dictionary-based technique. From the first two experiments, we choose the best performed technique from each experiment to make a complete text analysis on mobile phone. After that, we compare the combined text analysis with the syllable-based G2P conversion technique which uses a rule-based technique to segment an input text into syllables then convert them into the phone sequence.

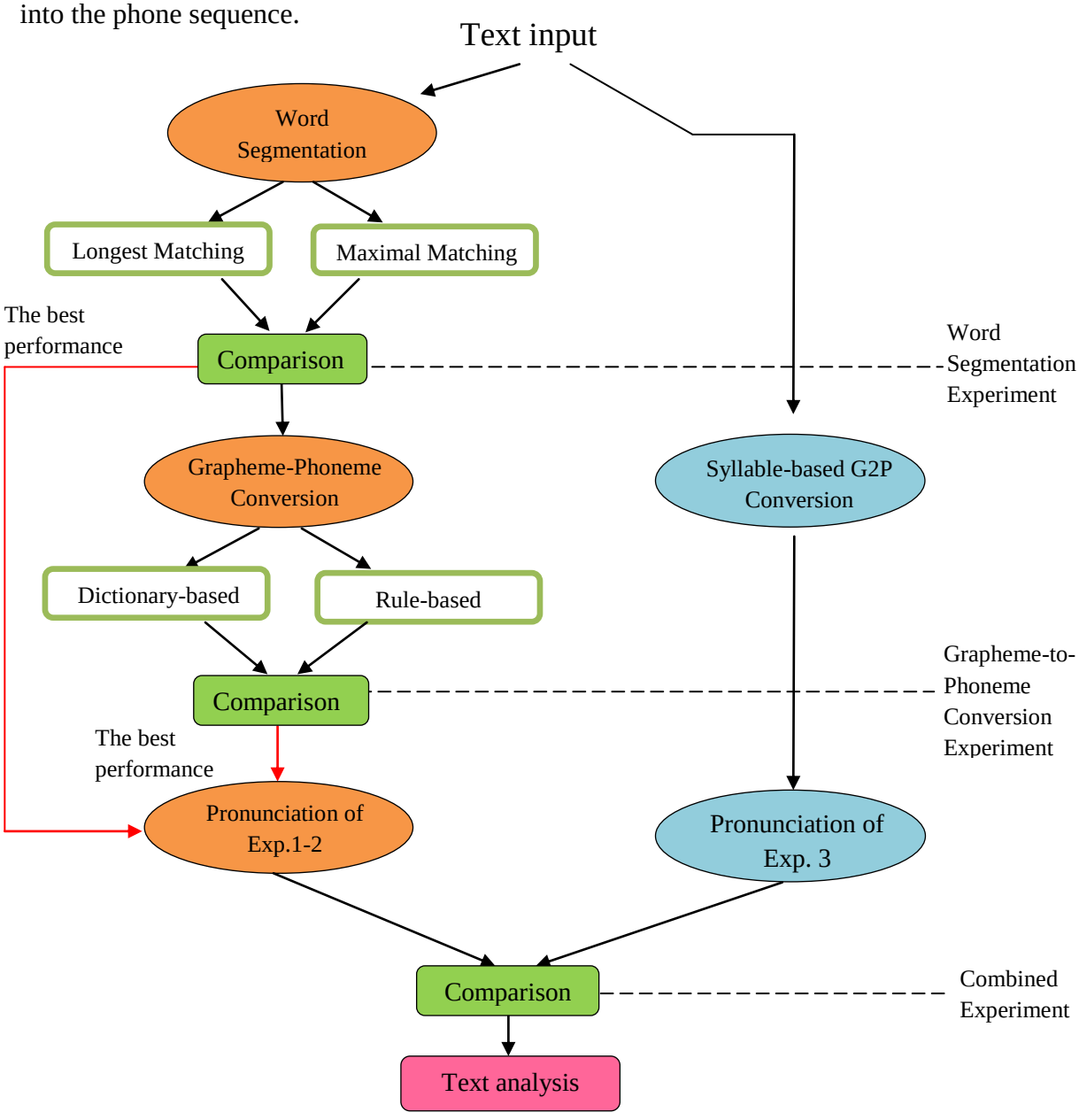

Figure 3.1 Comparison architecture

Based on the experimental results from the previous works, we select to conduct an experiment to compare only the performance of the Longest Matching and Maximal Matching techniques. A machine learning technique is not suitable for embedded devices because it requires quite large memory and needs longer processing time than the rule-based and dictionary-based techniques. A rule-based technique needs hand-crafted Thai writing rules. It requires expert to list a set of rules.

In the G2P conversion experiment, we compare two algorithms from the literatures, dictionary-based and PGLR based techniques. Even though a decision tree technique achieves the highest accuracy, it needs longer processing time and larger memory and storage than the rule-based and the dictionary-based technique. The PGLR and the dictionary-based techniques do not require a large amount of memory and the conversion process can be done within a short period of time.

In the combined experiment, we compare the results from the word segmentation and G2P conversion experiment with the combination of syllable-based G2P conversion technique. Since the objective of text analysis is to produce a sequence of phonemes, word segmentation may not be necessary in a Thai TTS system.

For testing environment, we implement a program with Visual Studio 2005 using C# programming language and then run it on the Pocket PC 2003 emulator. The detailed specification of the mobile phone is Window Mobile™ 2003 Second Edition using ARM 920 processor and 29.7 MB of memories.

3.2 Resources

The resources for the comparison of the existing Thai text analysis algorithms section, and the proposed technique section come from BEST2009 [1] and TsynC [15] as shown in Table 3.1, and the dictionaries as shown in Table 3.2. Language used in both resources is closed to written and spoken language of people life. The BEST2009 resource is a standard corpus for Thai word segmentation software contest, whereas the TsynC resource is a speech corpus, which is used for speech synthesizer. They are reliable resources.

1. BEST2009

The text resource contains Thai texts from various sources, and includes non-Thai characters such as English words, symbols, and digits. The texts are under four topics i.e. article, encyclopedia, news, and novel.

BEST_All, the overall text, consists of 509 files. It is divided into two sub sets:

- BEST_Training set composes of 489 files. It is then used to construct the BEST_WordList dictionary shown in Table 3.2

- BEST_Test set consists of 20 files. It is used as a test set in the word segmentation experiment.

2. TsynC

TsynC is a text transcription taken from the Thai speech corpus TsynC. This corpus consists of 5,200 sentences that are already segmented into words. These sentences were selected to cover all Thai and loan words.

Table 3.1 Amount of data

Resource	No. of words	No. of distinct words	No. of files
BEST_All	5 million	32,060	509
BEST_Training		31,472	489
BEST_Test		6,592	20
TsynC	429,588	15,907	1

3. Dictionaries

Since the experiments on word segmentation and G2P conversion in the comparison of the existing Thai text analysis algorithms section and syllable segmentation in the proposed technique section need word lists or dictionaries (word unit and syllable unit). We use five dictionaries as shown in Table 3.2.

1. LEXiTRON [16] and LEXiTRON Pro [17] dictionaries are pronunciation dictionaries developed by NECTEC. Both dictionaries contain words and their pronunciations.

    - LEXiTRON dictionary consists of 19,400 entries. Its size is about 949 KB.
    - LEXiTRON Pro (word unit) dictionary contains 104,869 words with the size of 5,620 KB.
    - LEXiTRON Pro (syllable unit) dictionary composes of 15,085 unique syllables that is another version of the LEXiTRON Pro (word unit) dictionary. The size of this dictionary is around 405 KB.

2. The BEST_WordList dictionary is generated from the distinct words found in the BEST_Training set. The size of this dictionary is about 760 KB (31,472 entries). Since, this dictionary is automatically generated from a text corpus, there is no pronunciation information attached.

3. TsynC dictionary is a list of words in the TsynC corpus. It contains both words and their pronunciation.

Table 3.2 Dictionary resources

Dictionary	No. of words/ syllables	Size(KB)*	Pronunciation
LEXiTRON	19,400 words	949	✓
LEXiTRON Pro (word unit)	104,869 words	5,620	✓
LEXiTRON Pro (syllable unit)	15,085 syllables	405	✓
BEST_WordList	31,472 words	760	×
TsynC	15,907 words	960	✓

*Size of each dictionary is measured from space on disk

3.3 Experiments and results

3.3.1 Word Segmentation Experiment

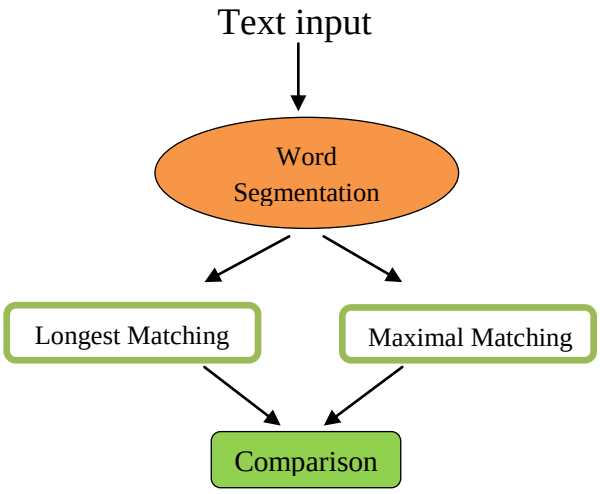

Figure 3.2 Word Segmentation Experiment

Figure 3.2 shows a process of the word segmentation, which is the first module for the text analysis. It separates a text input into sequence of words. For text input, we use the BEST_Test as test set. It contains 20 files in four topics: article, encyclopedia, news, and

novel. For the evaluation metrics, we measure the performance of each technique by F-Measure, processing time and size of the program. We are going to explain more details about the evaluation metrics in the next section. The word segmentation consists of three sub experiments. Firstly, we compare the performance of the Longest Matching and Maximal Matching techniques using the LEXiTRON dictionary. Secondly, we change the LEXiTRON dictionary to the BEST_WordList dictionary, because we believe that the dictionary affects the accuracies in the first sub experiment. Finally, we identify a suitable dictionary size from the LEXiTRON Pro dictionary.

3.3.1.1 Evaluation of word segmentation experiment

There are three criteria for measuring the performance of each technique: F-Measure, processing time, and size of program.

1. F-measure [1] is used to evaluate the accuracy of a word segmentation algorithm. F-Measure is calculated from recall and precision as shown in equation (3.1)-(3.3).

$$\text{F-Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.1}$$

$$\text{Precision} = \frac{\text{Corr}}{\text{OutputWord}} \tag{3.2}$$

$$\text{Recall} = \frac{\text{Corr}}{\text{RefWord}} \tag{3.3}$$

where, Corr is the number of words in the output that is correctly segmented,
OutputWord is the total number of words in the output, and
RefWord is the total number of words in the reference.

Examples of F-measure Calculation:

Input: ฉันกินข้าวที่สามย่านแล้ว

Answer: ฉัน|กิน|ข้าว|ที่|<NE>สามย่าน</NE>|แล้ว| RefWord = 6

* <NE> and </NE> are special tag for evaluating the boundary of the segmented word. In experiment, we do not consider the segmented words within the <NE>

Output1: ฉัน|กิน|ข้าว|ที่|สามย่าน|แล้ว|

OutputWord = 6,	Corr = 6,
Precision = 6/6,	Recall = 6/6

Therefore, we obtain the F-Measure = 1 or 100%.

Output2: ฉัน|กิน|ข้าว|ที่|สาม|ย่าน|แล้ว|

OutputWord = 6,	Corr = 6,
Precision = 6/6,	Recall = 6/6

Therefore, we obtain the F-Measure = 1 or 100%.

Output3: ฉัน|กิน|ข้าว|**ที่สาม**|ย่าน|แล้ว|

* **ที่สาม** is incorrect segmentation

OutputWord = 5,	
Corr = 4,	
Precision = 4/5,	Recall = 4/6

Therefore, we obtain the F-Measure = 8/11 ≈ 72.72%.

Output4: ฉัน|กิน|ข้าว|**ที่สาม|ย่านแล้ว**|

*both **ที่สาม|** and **ย่านแล้ว** are incorrect segmentation

OutputWord = 5,	Corr = 3,
Precision = 3/5,	Recall = 3/6

Therefore, we obtain the F-Measure = 6/11 ≈ 54.54%

2. Processing time is used to evaluate the runtime performance which is an important issue for developing a TTS system on mobile devices. We record the time since program starts until it finishes the segmentation process.

3. Size of the program is measured from the size of an executable file and a dictionary.

3.3.1.2 Comparing the existing word segmentation algorithms and results

1. Comparing the Longest Matching and Maximal Matching techniques using the LEXiTRON dictionary

The goal of this experiment is to choose an appropriate algorithm from the existing word segmentation algorithms for embedded devices. We compare the Longest Matching technique with the Maximal Matching technique. We use the LEXiTRON dictionary in the experiment.

Figure 3.3 and 3.4 demonstrate the result of the Longest Matching and Maximal Matching techniques respectively when running on the mobile emulator. The experimental results in terms of F-Measure, computational time, and size are shown in Table 3.3. We found that although F-measure of the Maximal Matching technique is about 0.03% higher than that of the Longest Matching technique, the Longest Matching technique needs only half of the processing time required by the Maximal Matching technique. The size of both techniques is not quite different.

Figure 3.3 A result of the Longest Matching technique

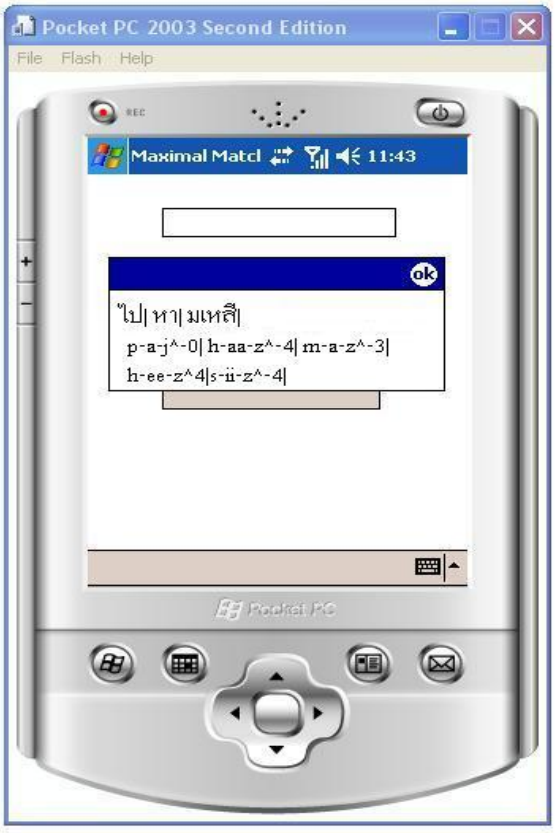

Figure 3.4 A result of the Maximal Matching technique

Table 3.3 The comparison of the performances of the Longest Matching and Maximal Matching techniques using the LEXiTRON dictionary

	Longest Matching technique	Maximal Matching technique
F-measure	78.09	78.12
Processing Time	4.45 min	8.17 min
Size(KB)	960	962.5

Nevertheless, we cannot select the Longest Matching technique for word segmentation since the F-measures of this algorithm is quite low when it is compared to the results presented by Haruechaiyasak et al.[9] where the F-measure of 87.5% is reported on the same techniques. Therefore, we set the next experiment in order to find the cause of the problem. We expect that a cause of problems may come from the dictionary used in the experiment.

2. Comparing the Longest Matching and Maximal Matching techniques using the BEST_WordList dictionary

The goal of this experiment is to test the effect of dictionaries on the F-Measure, therefore we use the BEST_WordList dictionary instead of the LEXiTRON dictionary.

Table 3.4 shows the experimental results. The difference in the F-measure between the Maximal Matching and Longest Macthing techniques is about 0.01%. This is smaller than what we obtained in the previous experiment. We notice that when we changed the dictionary from LEXiTRON to BEST_Wordlist, the F-measure is approximately 10% higher than the F-measure reported in the previous experiment. The number of words in the BEST_WordList dictionary is larger than that of the LEXiTRON dictionary. Moreover, the average word lengths in both dictionaries are different. The average word length in the LEXiTRON dictionary is longer than the average word length in the BEST_WordList dictionary as shown in Table 3.5. For example, the LEXiTRON dictionary contains one word "ถนนลาดยาว" (ladyaw road), while the BEST_WordList dictionary consists of two words: "ถนน" (road) and "ลาดยาว" (name of road). The unit size of words in the LEXiTRON dictionary is different from the unit size of words in the BEST2009 text.

From Table 3.3 and 3.4, we found that the Longest Matching technique is suitable for a resource-limited device since the processing speed is about one time faster than that of the Maximal Matching technique while the accuracies in terms of the F-measure and the sizes of the programs are not quite different.

Table 3.4 The comparison of the performances of the Longest Matching and Maximal Matching techniques using the BEST_WordList dictionary

	Longest Matching technique	Maximal Matching technique
F-measure	88.55	88.56
Size(KB)	771	773.5

Table 3.5 The different word length of the LEXiTRON and BEST_WordList dictionary

LEXiTRON dictionary	BEST_WordList dictionary
ถนนลาดยาว	ถนน
	ลาดยาว

Furthermore, the Out of Vocabulary (OOV) terms, which are the words found in the BEST_Test set but not in the BEST_WordList dictionary, affect the F-measure. Table 3.6 shows that the OOV rate of the LEXiTRON dictionary is higher than those of the BEST_WordList dictionary, by approximately 19% for the distinct words and 12% for the words respectively. Here, we can conclude that unit word length and OOV rates of dictionaries affect on F-Measure.

Table 3.6 The comparison of the OOV rates between the LEXiTRON dictionary and the BEST_WordList dictionary

	OOV rate of distinct words	OOV rate of words
LEXiTRON dictionary	25.94%	14.13%
BEST_WordList dictionary	6.88%	2.14%

Unfortunately, the BEST_WordList dictionary only contains words without their pronunciations. Therefore, we conduct one additional experiment to identify a suitable size of dictionary.

3. Investigating the affect of the size of a pronunciation dictionary

In this experiment, the objective is to find a suitable dictionary size for the embedded devices that does not affect F-Measure. We use the LEXiTRON Pro dictionary that contains both words and pronunciations. Since this dictionary is a pronunciation dictionary and contains 104,869 distinct entries, it may cover many words in test set more than the other dictionaries. This dictionary is not appropriate for embedded device because its size is too large.

We first identify the frequency of each entry in the LEXiTRON Pro dictionary. We found that from 104,869 entries, only 22,969 entries occur in the BEST_Training set. In this experiment, we selected the 5,000, 10,000, 15,000, 20,000, 22,969 (all the entries that occur at least once in the BEST_Training set), 25,000, and 30,000 most frequently-used words and created dictionaries of different sizes. We then evaluated the Longest Matching technique that uses these dictionaries in terms of F-measure, OOV rate, and program size.

Figure 3.5 shows the performances of the Longest Matching technique as measured by F-measure when we use the dictionaries with different sizes and different portions of OOV words. The F-measure increases gradually when the OOV rate decreases until the point where the dictionary contains 22,969 entries. At this point we obtained the F-measure of 85.68% with the OOV rate of 15.5% for distinct OOVs and 6.8% for all OOV tokens. After this point both the F-measure and the OOV rate do not change much as the additional entries in the dictionary rarely occur. Thus, increasing the number of entries in dictionary could not help improve the segmentation accuracy. In terms of the program's footprint, each entry in the dictionary approximately requires the same amount of memory; therefore, the size of the dictionary increases in linear proportion to the number of words.

Based on the experimental results, we can say that the performance of the dictionary-based word segmentation technique relies mainly on the portion of the OOV words. However, a larger dictionary does not always increase the F-measure because additional words in that dictionary may not be used. So, the LEXiTRON Pro dictionary that contains 22,969 words is suitable for a resource-limited device when we consider F-measure, OOV rate and size of the word segmentation program together.

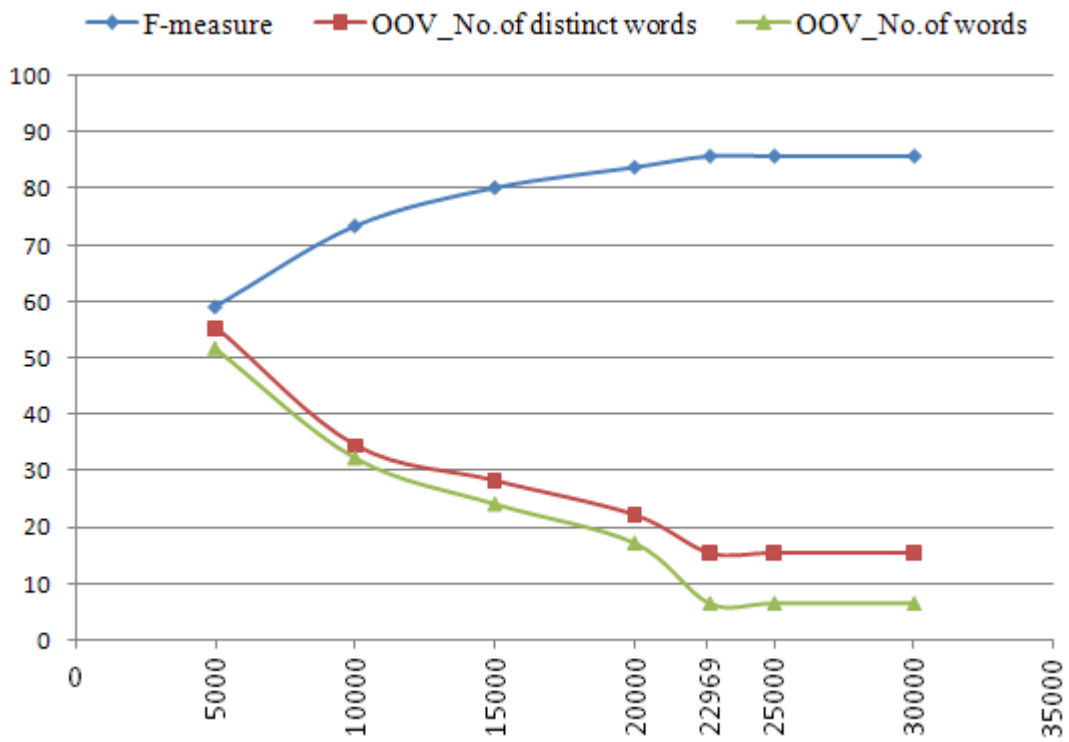

Figure 3.5 F-measures and OOV rates when varying the size of a dictionary

3.3.2 Grapheme-to-Phoneme conversion Experiment

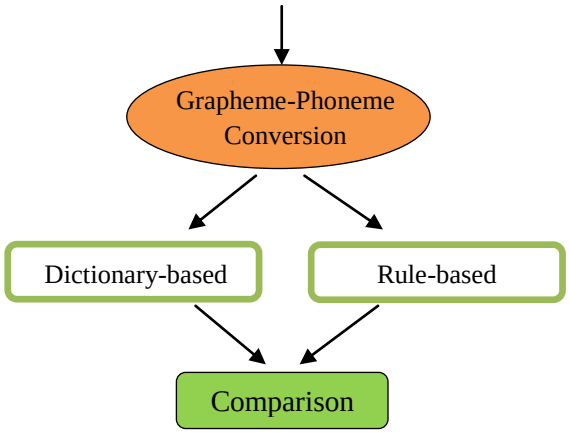

Figure 3.6 G2P conversion Experiment

The goal of this experiment is to select an appropriate algorithm for running on embedded devices as shown in Figure 3.6. We compare two algorithms: the dictionary-based technique and the PGLR technique. They are judged by accuracy at the levels of word, syllable, and phone, processing time and size of the program. We will describe the

experimental results in the next section. A test set is the TsynC set that contains approximately 430 K words with around 16 K distinct words. We chose this set as our test set since it is the set that has a manually-labeled pronunciation transcription of a continuous text.

Two techniques are implemented in different programming languages. The dictionary-based technique is written in C# using Visual Studio 2005 while the PGLR technique is written in C.

3.3.2.1 Evaluation of the G2P conversion experiment

We measure accuracy in terms of word, syllable, and phone levels, since one word may compose of one syllable or more than one syllable, in the same way one syllable consists of more than one phone. The phone is the smallest unit that is represented by Thai language. For measurement, it may not be effective if we evaluate only word accuracy. When accuracy in word level is incorrect, it may be caused by failing in some syllable. For example,

Input: กรรมการ (committee)

Answer: |k-a-m^-0|m-a-z^-3|k-**aa**-n^-0|

Output1: |k-a-m^-0|m-a-z^-3|k-**a**-n^-0|

The input "กรรมการ"contains three syllables. The output1 is wrong in the word level since "|k-**a**-n^-0|" is generated wrongly. When we evaluate this input in the syllable level, it obtains 66.67%. The idea of phone accuracy is similar to the syllable accuracy but the output is matched phone by phone, so it achieves 91.67%. Practically, we check the case of deletion and insertion.

Output 2: |k-a-m^-0|k-aa-n^-0|

Output 3: |k-a-m^-0|m-a-z^-3|**r-@@-z^-0**|k-aa-n^-0|**r-@@-z^-0**|

The output 2 shows a deletion case, while the output 3 shows an insertion case that two syllables were added. In the word level, both output 2 and 3 are incorrect. In the syllable and phone levels, the output 2 gets 66.67%, since the output2 has correct two syllables and one deletion. On the other hand the output 3 obtains 33.33% for syllable and phone accuracy, because it contains correct three syllables and two insertions.

3.3.2.2 Comparing the G2P conversion algorithms

For the dictionary-based technique, we use the LEXiTRON Pro dictionary that contains 22,969 words selected from word segmentation experiment. For the PGLR technique, we use the algorithm proposed by Tarsaku et al. [2]. This algorithm applies a context-free grammar representing a Thai syllable structure and probabilistic information to generate a sequence of phonemes from the input sequence of words. This technique is able to capture the global and local contexts by probabilistic model. The experiments were conducted on the computer that the detailed specification of the computer used is Window XP, Intel® CoreTM 2 Duo Processor (2.4 GHz.)

Table 3.7 demonstrates that all the accuracies of the PGLR technique are higher than those of the dictionary-based technique and the size of the program is small. On the other hand, the processing time of the PGLR technique measured on a personal computer is approximately 29 times longer than that of the dictionary-based technique, and it requires about 57 KB of memory which is twice as much as those required by the dictionary-based technique.

Table 3.7 The comparison of G2P conversion accuracies between the dictionary-based technique and the PGLR technique using TsynC dictionary

	Dictionary-based		PGLR	
Test set	No. of unique words	No. of words	No. of unique words	No. of words
Word accuracy	30.65	82.73	72.6	95.34
Syllable accuracy	43.78	85.11	84.11	95.66
Phone accuracy	49.97	89.79	90.27	97.33
Processing time on personal computer	1 sec.		29 sec.	
Processing time on mobile emulator	47 sec.		1363 sec.	
# word in 1 sec. (test on distinct word)	338		11	
Size(KB)	1,110		380	

The accuracies of the dictionary-based technique are low because this technique cannot generate the pronunciation of the unknown words. Moreover, the types of words in the LEXiTRON Pro dictionary differ from those in the TsynC test set. Most words in the TsynC set are long; many of them are technical words or organization and person names. In addition, we notice that when every token in the TsynC set is evaluated, the accuracy is much higher than the accuracy when only a set of distinct words is evaluated (76.25% vs. 25.82%) as frequently-occurred words are usually included in the dictionary and can be converted into phonemes with high accuracy.

When we consider the processing times of both algorithms on a mobile emulator, they are very different. The dictionary-based technique has a faster processing speed and can generate around 338 pronunciations per second while the PGLR technique needs much more processing time (30 times longer) which is not acceptable in a real-time system. Since a TTS system consists of both a text analysis module and a speech synthesis module, if the text analysis module requires a long processing time like the PGLR technique, the system will not respond within an appropriate time frame. Therefore, the dictionary-based technique is a promising technique for G2P on mobile devices/

Since the dictionary-based technique yields low accuracy, we try to conduct two experiments to investigate a way to increase the accuracies. Table 3.8 shows the performances of two G2P techniques discussed in Table 3.7, a dictionary-based and a PGLR technique, and the performance of the combination of both techniques which utilizes the PGLR technique only when encountering unknown words. The accuracies reported in Table 3.8 are calculated from the list of 15,907 unique words. We found that the dictionary-based technique is the lowest all accuracies, but its processing time is 28 times as short as the combination of both techniques. We expected that the combining method would achieve a better accuracy and also consume less processing time as a PGRL parser is utilized only for unknown words. However, we found that the combining method achieved slightly better accuracy than the PGLR but still consumed much more processing time than the dictionary-based technique. The combined method can process only 12 words per second which when integrating with other processes of a TTS system may not be done in real time.

Table 3.8 The performances of baseline G2P approaches tested on the 15,907 unique words

	Dictionary	PGLR	Dictionary + PGLR
Word accuracy	30.65	72.6	74.54
Syllable accuracy	43.78	84.11	84.18
Phone accuracy	49.97	90.27	90.22
# word processed in 1 sec.	338	11	12
Size(KB)	1,117	380	1,497

Table 3.9 G2P accuracy when increasing the size of the word-level pronunciation dictionary

	22,969 entries	45,938 entries	68,907 entries	104,869 entries
Word accuracy	30.65	33.66	38.1	40.72
#No. word processed in 1 sec.	338	284	230	193
Size(KB)	1,110	2490	3768	5620

A simple way to improve the accuracy of the dictionary-based technique is to add more entries to the dictionary. Therefore, we doubled the size of the dictionary by selecting more entries randomly from the words in the LEXiTRON Pro dictionary. We also experimented with the dictionary that is three times bigger and the one that contains all entries in the LEXiTRON Pro dictionary. The results are shown in Table 3.9. When the number of words in the dictionary increases, the accuracy of the dictionary-based G2P technique improves slightly. However, the size of the program and the processing time also increase. The best performance is obtained when the entire LEXiTRON Pro dictionary is used; nevertheless, the accuracy is still much lower than that of the PGLR technique shown in Table 3.8. Hence, increasing the number of words in the dictionary is not the best way to handle unknown words. Finally, from the first two experiments, we found that the Longest Matching technique using the 22,969 most frequently-used entries in the LEXiTRON Pro dictionary for word segmentation and the dictionary-based technique for G2P conversion are promising techniques for text analysis on mobile devices.

3.3.3 Comparing the combination of word segmentation and G2P conversion techniques with the Syllable-based G2P conversion technique

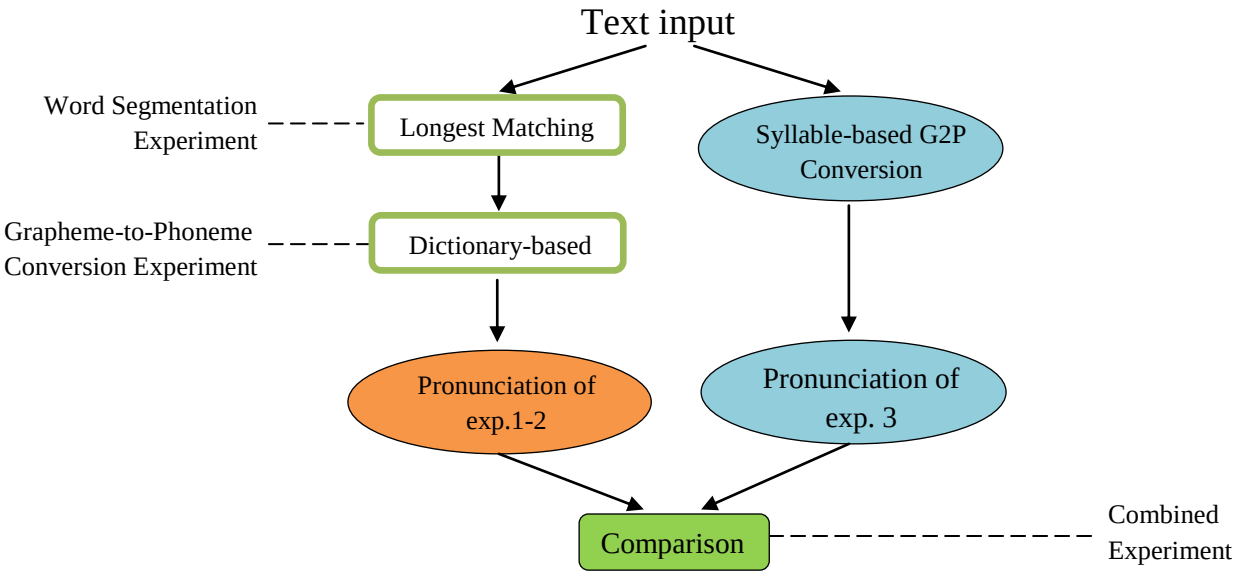

Figure 3.7 Combined Experiment

In this experiment, the goal is to evaluate the overall performance of the text analysis module. The best performance of the first two experiments is combined together. We select the Longest Matching technique that uses the LEXiTRON Pro dictionary with 22,969 entries from the first experiment and the dictionary-based technique from the second experiment as shown in Figure 3.7. These two techniques are applied sequentially to an input text. The resulted pronunciation is then compared to the result obtained from the syllable-based G2P conversion proposed by Chinathimatmongkhon et al [11]. In this experiment, we evaluate it by three criteria used in the G2P conversion experiment. We use TsynC with 429,588 entries as a test set. Both techniques are run on the Pocket PC 2003 emulator. The overall performances of this experiment are shown in Table 3.10. The accuracies of the combination of word segmentation and G2P conversion are higher than those of the syllable-based G2P conversion, since the syllable-based G2P conversion use only a set of rules that may not cover all Thai writing syllables. It also requires shorter processing time. Although the combination of word segmentation and G2P conversion needs more space for the executable file and dictionary (1.1 MB), this size is not too large for an embedded device.

However, we cannot ignore a higher performance of the PGLR technique; accordingly we changed the G2P conversion from the dictionary-based technique to the PGLR technique. We obtained 95.94% on word accuracy. Since the PGLR technique requires much longer processing time, this technique is not selected. We may need to reduce the processing time of

the PGLR technique or improve the accuracy of the dictionary-based technique in order to improve the performance of the G2P conversion.

Table 3.10 The comparison between the combination of word segmentation and G2P conversion, and the syllable-based G2P conversion

	Word segmentation and G2P conversion	Syllable-based G2P conversion
Word accuracy	84.22	56.89
Syllable accuracy	90.02	70.47
Phone accuracy	91.33	80.21
#word processed in 1 sec.	101	86
Size(KB)	1,101	12

3.4 Conclusion

We focus on developing a Thai text analysis for TTS system on mobile devices that responds in real time. We analyze and compare the existing text analysis algorithms in order to choose a suitable technique for a TTS system on mobile devices. We use F-measure, processing time, and size of the program to evaluate the performance of the word segmentation experiment. In this experiment, the Longest Matching technique which uses 22,969 most frequently-used entries in the LEXiTRON Pro dictionary is selected because its processing time is shorter than the Maximal Matching technique. The F-Measure does not affect choosing the Longest Matching, since F-Measure of both techniques is quite not different.

In the G2P conversion, accuracies in word, syllable, and phone level, processing time and size of the program's footprint are used as evaluation metrics. We found that the dictionary-based technique is more suitable since the PGLR technique needs longer processing time that is not acceptable in a real-time system. However, the PGLR technique gives all accuracies higher than the dictionary-based technique.

In the combined experiment, we use the same criteria as the G2P conversion experiment. A combination between the Longest Matching technique for word segmentation and the dictionary-based technique for the G2P conversion experiment give a better result than the syllable-based G2P conversion. Moreover they can produce in a suitable time. However the accuracies of the selected techniques are low when comparing the accuracies of the PGLR technique

Chapter 4

Combining a Pronunciation Dictionary of Multiple-Sized Units and a Rules-Based Technique for Thai G2P Conversion

This chapter presents a new G2P approach that improves the accuracies of the dictionary-based approach. Five sections are discussed in this chapter. Firstly, we describe a problem caused by the dictionary-based technique; secondly, we show the architecture of a proposed technique for G2P conversion. Thirdly, we explain all techniques used in the experiments. Fourthly, we discuss the experiments and results. Finally, we conclude the results of all the experiments.

4.1 A problem in the dictionary-based technique

From Chapter 3, we compared and analyzed the existing text analysis algorithms for a Thai Text-to-Speech (TTS) system on mobile devices. We conclude that the combination of word segmentation and grapheme-to-phoneme (G2P) conversion are more outstanding than the syllable-based G2P conversion technique especially on mobile emulator. A Longest Matching technique is a suitable technique for word segmentation while a dictionary-based technique is appropriate for G2P conversion. Although, those techniques are suitable for the TTS on a small device because of their short processing time, they do not obtain high accuracy when comparing to the sophisticated techniques. Since the combination of both techniques requires a dictionary, a pronunciation of unknown word cannot be generated. From literatures, Mittrapiyanuruk [10] uses only a rule-based technique to convert a grapheme to phoneme. We think that the accuracy of the orthographical-to-phonological mapping is not as high as the rules and it may not cover all cases in Thai pronunciation system. Hence, we propose a new G2P conversion approach. We apply the use of a syllable-level pronunciation dictionary in addition to a set of conversion rules to generate a phone sequence of each syllable. We believe that combining both techniques together help increase the accuracy than using only the rules. We will explain the new G2P conversion approach in next section.

4.2 Architecture of a proposed technique

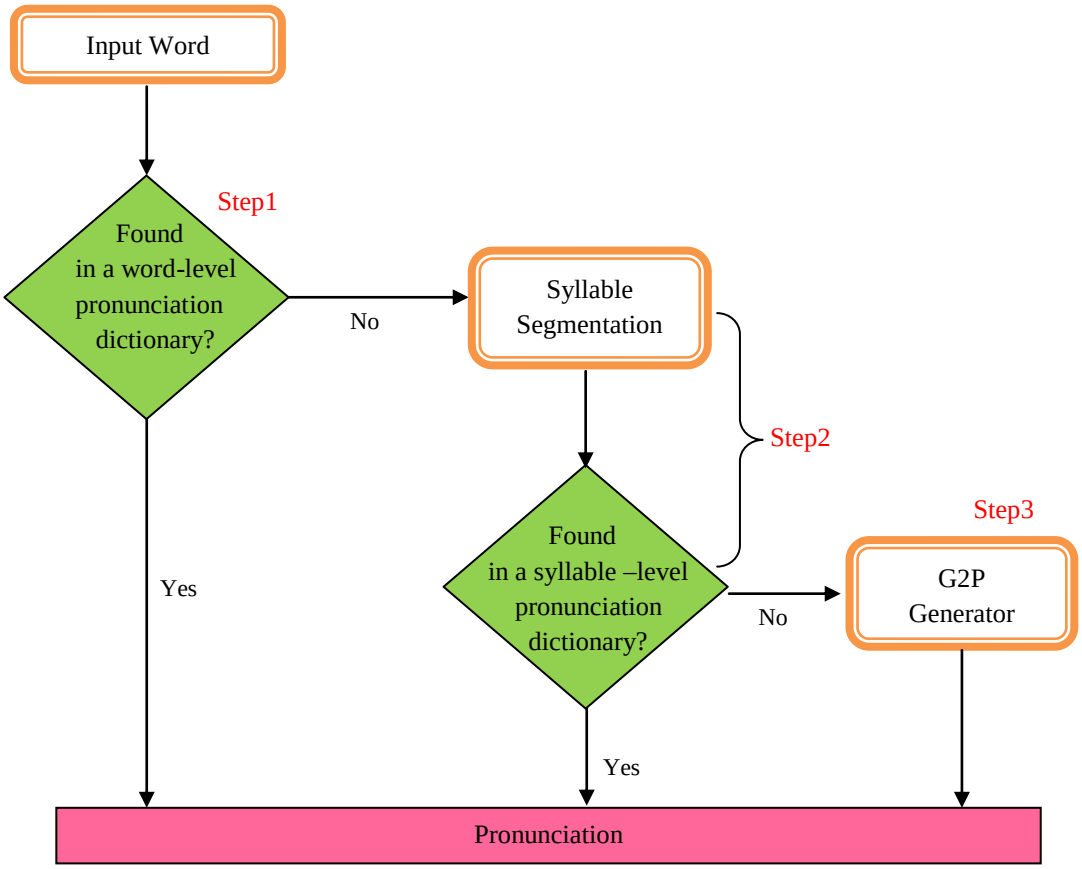

Figure 4.1 The proposed technique

The proposed technique aims at improving the text analysis part especially the G2P conversion of a Thai TTS system on mobile devices. It composes of three main steps as shown in Figure 4.1. When the user enters an input word "ประเทศไทย" (Thailand), the word is first looked up in a word-level pronunciation dictionary. If the word is found, otherwise the second step is applied to handle the unknown word. Step 2 separates the word into syllables using a set of 43 rules. In this case, we got three syllables "ประ|เทศ|ไทย|" separated by '|'. Each syllable is then looked up in a syllable-level pronunciation dictionary. For each syllable that is not in the dictionary, a G2P generator is used to produce the pronunciation of the unknown syllable with a set of rules. Finally, we get the pronunciation as "pr-a-z^-1|th-ee-t^-2|th-a-j^-0|".

4.3 The techniques in new G2P approach

We apply three techniques for the new G2P approach. Initially, the pronunciation dictionary is used for the dictionary-based technique. In addition, we handle an unknown word by conducting syllable segmentation. Lastly, a G2P generator is applied in order to cope with an unknown syllable.

4.3.1 Pronunciation dictionary

The proposed technique requires a dictionary that contains the pronunciation dictionary at both word and syllable levels. For the word-level pronunciation dictionary, we use the LEXiTRON Pro dictionary that contains 22,969 most frequently-used words. This dictionary comes from finding frequency of each word in a training set of BEST 2009 in word segmentation experiment in Chapter 3. For the syllable-level pronunciation dictionary, we also created from the LEXiTRON Pro dictionary. Each word in the LEXiTRON Pro dictionary and its pronunciation are manually segmented into syllables. There are 15,085 unique syllables. These syllables become entries in our syllable-level pronunciation dictionary as shown in Table 4.1. Each dictionary is stored in a binary search tree structure; therefore, an input can be searched directly from this structure.

Table 4.1 The example of word level and syllable level pronunciation dictionary

Pronunciation dictionary			
Word level		Syllable level	
Word	Pronunciation	Syllable	Pronunciation
เกื้อกูล	|k-vva-z^-2| k-uu-n^-0|	เกื้อ	|k-vva-z^-2|
กล้าหาญ	|kl-aa-z^-2|h-aa-n^-4|	กูล	|k-uu-n^-0|
		กล้า	|kl-aa-z^-2|
		หาญ	|h-aa-n^-4|

4.3.2 Syllable Segmentation

We use a set of 43 rules written in a form of a regular expression, which is proposed by Sawamipak [7] for syllable segmentation. The main advantage of these rules is that it does not require storage and memory for storing data and statistics like dictionary-based technique and computational time like machine learning or probabilistic approach. Whereas the disadvantage is that they are not easy to handcraft the rules because they are embedded in source code. The examples of the rules are shown below.

- เ [C] [T] ? [า] represents the syllables such as “เงา” (shadow) and “เก้า” (nine)
- [C] ◌ั [T]? ว represents the syllables such as “วัว”(cow), and “รั้ว” (fence)
- [เ แ โ] [C] [T] ?ะ represents the syllables such as “โต๊ะ” (table), and “แฉะ” (wet)
- เ [C] ◌ื [T] ? อ represents the syllables such as “เสือ” (tiger), and “เรือ” (boat)
- [ใ ไ] ห [ง ญ น ม ย ร ล ว] [T] ? represents the syllables such as “ใหญ่” (big), and “โหล” (dozen)

where [C] is an initial consonant; [CC] is a cluster consonant; [V] is a vowel; [V]? is an optional vowel; [S] is a final consonant; [S]? is an optional final consonant; [T] is a tone marker; [T]? is an optional tone marker.

Since the set of 43 rules does not cover all the syllables in Thai nor transliterated syllables, we designed seven additional rules as follows:

- [C]รร[S] represents the syllables such as “กรรม” (action), and “ธรรม” (doctrine)
- [C][V]?[S][ย ส ด ว ช ท น ศ พ ทร ทน ษณ มณ์] represents the syllables such as “จันทร์” (Monday, moon), and “วงศ์” (family)
- [V][C] ◌ิร์[S] represents the transliterated syllables such as “เวิร์ด” (word), “เบิร์ด” (bird)
- เ[C]อร์ represents the transliterated syllables such as “เตอร์” (compu<u>ter</u>), “เซอร์” (<u>sur</u>prise)
- [V][C CC][T]?[S]? [ย ส ด ว ช ท น พ] represents the transliterated syllables such as “เกรย์” (gray), and “เต็นท์” (tent)
- [C][ร ล][า อ][น ย][ด ต] represents the transliterated syllables such as “สง<u>กรานต์</u>” (the water festival celebrated in April), “บลอนด์” (<u>blond</u> color)
- [เ][C][T]?[า] [ย ส ด ว ช ท น พ] represents the transliterated syllables such as “เฮ้าส์” (house), and “เคาน์” (<u>coun</u>ter).

4.3.3 G2P Generator

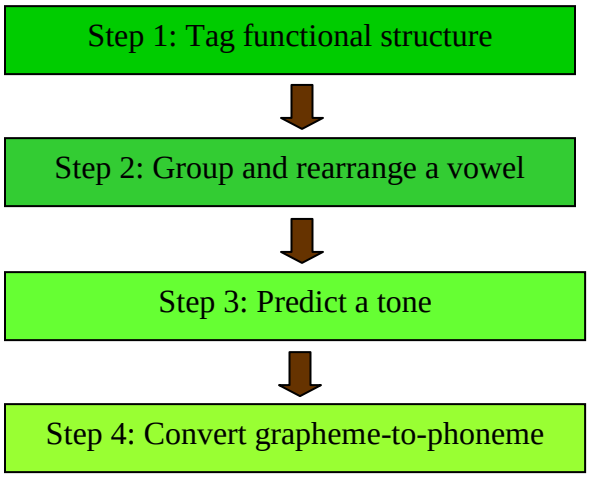

Figure 4.2 The steps of G2P Generator

This module consists of four steps to produce the pronunciation of an unknown syllable as shown in Figure 4.2. This process is used when the pronunciation cannot be found in the syllable-level pronunciation dictionary. In step1, step 2 and step 4, we follow by Sriman [18] but, in step 3, we predict a tone from Thai characteristic written by Kramonlawech [19]. The main advantage of Sriman's technique is that it does not take long time for working process and it covers Thai writing system.

Step 1, each character in the unknown syllable is tagged with its functional structure. We consider a Unicode code of each character The Unicode code range of Thai character "ก" until character "ฮ" starts with U+0E01 to U+0E2E. The vowel range can be divided into two ranges that are between U+0E2F and U+0E47, and U+0E4C until U+0 E4E. The tone marker is between U+0E48 and U+0E4B. Each character in the unknown syllable can be classified into one of the four functional structures: C for an initial consonant, V for a vowel, S for a final consonant, and T for a tone marker. For example, the syllable "เรียบ" (smooth) is tagged as "VCVVS" as shown in Table 4.2.

Table 4.2 Tagging a unknown syllable to functional structure

Character	Unicode code	Functional structure
เ	U+0E40	V
ร	U+0E23	C
◌	U+0E35	V
ย	U+0E22	V*
ป	U+0E1A	S

*If an unknown syllable has the character“ย,อ” and also has more than one vowels, the character “ย,อ” is set functional structure as vowel or V.

For classifying between initial consonant and final consonant, we consider a position of consonant in Thai written syllable structure as follow,

<Syllable> → <Initial_consonant><Vowel><Final_consonant> (4.1)

→ <Vowel>< Initial_consonant><Final_consonant> (4.2)

→ <Intial_consonant><Vowel> (4.3)

→ <Vowel><Intial_consonant> (4.4)

From equation (4.1)-(4.4), we notice that there are two positions of the initial consonant and two position of the final consonant. The initial consonant is placed in front of the vowel such as “ป**า**ก” (mouth), “ก่**อ**น” (before), etc. and immediately behind the vowel such as “โ**ล**ก” (world), “เ**ข**ต” (district), etc. The final consonant is put behind vowel such as “ฟ**ัน**” (teeth), “งา**น**” (work), etc. or located behind the initial consonant such as “เล่**น**” (play), “โล**ภ**” (greedy), etc.

Step 2, the characters that have the same functional structure are grouped together and rearranged into the CVST pattern. For grouping the vowels, we calculate the Unicode code of vowels that is prepared from the step 1. For example, the syllable “เรียบ” (smooth) has three vowels at the first, second, and fourth position that represents the grapheme vowels: “เ”, “◌ี”, and “ย”. The Unicode codes of the vowels “เ”, “◌ี”, and “ย” is U+0E40, U+0E35, and

U+0E22 respectively. When we add those codes together, the result of adding corresponds to a single vowel "เอีย". Then each functional structure is rearranged into the CVST pattern.

Step 3, we identify a tone in each syllable. The problem is that it is not straightforward to identify the tone from a tonal marker. Since the tone marker in some syllable does not match directly to its real sound when the syllable is pronounced. For example, the syllable "เรียบ" (smooth) has no tonal marker but has a falling tone when it is pronounced. Therefore, the G2P generator has to predict the tone of an unknown syllable using a set of hand-written rules. Table 4.3 shows the fundamental of Thai tonal modulation [19] which we used to construct the rules.

Finally, each grapheme or a group of graphemes in the CVST pattern is one by one mapped into its phoneme as shown in Figure 4.3. Hence, the pronunciation of the unknown syllable "เรียบ" is /r-iia-p^-2/.

Table 4.3 Thai tonal modulation [19]

Type	Consonant/ vowel class		Tone				
			Mid	Low	Falling	High	Rising
Live syllable	M		กา	ก่า	ก้า	ก๊า	ก๋า
	H			ข่า	ข้า		ขา
	L		คา		ค่า	ค้า	
Dead syllable	M			กะ	ก้ะ	ก๊ะ	ก๋ะ
	SV	H		ขะ	ข้ะ		
		L			ค่ะ	คะ	ค๋ะ
	LV	H		ขาก	ข้าก		
		L			คาก	ค้าก	ค๋าก

M is a medium class consonant; H is a high class consonant; and L is a low class consonant; SV is a short vowel; LV is a long vowel

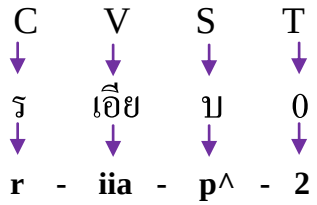

Figure 4.3 Mapping Grapheme-to-Phoneme

4.4 Experiments and Results

We used a text transcription taken from the Thai speech corpus TsynC as a test set for all the experiments. This test set contains 429,588 words with 15,907 unique words.

Table 4.4 shows the performance of the proposed G2P conversion approach. The second column, Word-dict, shows the performance when only the step 1 in our approach, looking up the word-level dictionary, is used. This is equivalent to the dictionary-based technique. The third column, Word-dict + syllable-dict, shows the performance when the first two steps, looking up the word-level and syllable-level dictionary, are applied. The last column, Word-dict + syllable-dict + rules, shows the performance when all the steps in the proposed technique are utilized. We calculate the accuracies from the list of 15,907 unique words.

Our technique achieved much higher G2P conversion accuracy when comparing to the performance of the dictionary-based approach in the second column. The word accuracy is twice as much. For the syllable accuracy and phone accuracy, the absolute improvement is more than 30%. Most of the improvement comes from the second step, the syllable-level dictionary. For many unknown words, the pronunciation of their syllables can be found in the syllable-level dictionary. However, our proposed approach needs more computational time. This technique can generate the pronunciation of around 91 words in one second that is about four times slower than the dictionary-based approach. In terms of size, the total sizes (program and dictionary) of both approaches are not quite different. When comparing the proposed approach to the PGLR approach in Table 3.8 in Chapter 3, our approach is about eight times faster, although the accuracy of the proposed approach is still not as good as that of the PGLR approach. These three steps are illustrated in Figure 4.4. We found that the accuracy of step 2, syllable segmentation and the syllable-level pronunciation dictionary, is 48.49% and the accuracy of step 3, G2P generator, is 64.60%. Most words that must be pronounced with linking syllable are generated wrong. For instance, the word "กรณี" (incident) has correct pronunciation as "k-@@-z^-0|**r-a-z^-3**|n-ii-z^-0|", but the proposed approach generates as "k-@@-z^-0|n-ii-z^-0|". The propose approach fails syllable "**r-a-z^-3**" that causes low accuracies. This problem can be improved in the future work. Nevertheless, the proposed approach is more suitable for embedded devices than the PGLR approach from its fast run time.

Table 4.4 The performance of each step in the proposed G2P conversion technique

	Word-dict	Word-dict + syllable-dict	Word-dict + syllable-dict + rules
Word accuracy	30.65	62.78	64.19
Syllable accuracy	43.78	71.17	74.16
Phone accuracy	49.97	79.22	83.62
# word processed in 1 sec.	338	115	91
Size(KB)	1,110	1,549	1,551

In Figure 4.4, we change a word input of the proposed technique to text sentence in order to evaluate the overall performance of text analysis. We try to add the Longest Matching technique for word segmentation experiment then each word is entered into the propose technique.

Table 4.5 shows the overall performance when word segmentation and G2P conversion are put together in the text analysis module. The test set for this experiment is all the sentences in the TSynC corpus that contains 429,588 words not the list of 15,907 unique words used in the previous experiments. The input sentences are first segmented into words using the Longest Matching approach. Then each word is converted into its corresponding phoneme transcription with three G2P approaches: dictionary-based, our technique (multiple-sized unit dictionary + rules), and PGLR.

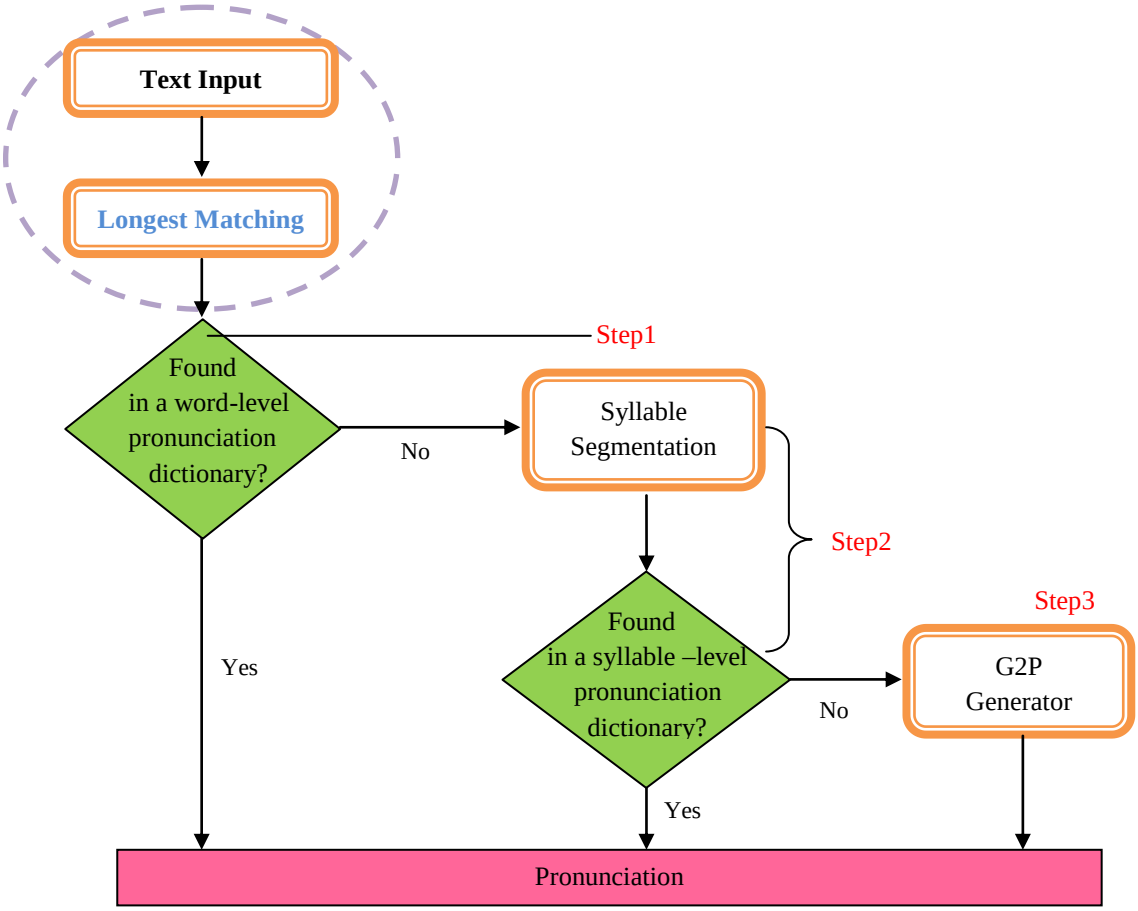

Figure 4.4 The Longest Matching technique and proposed technique

The result shows that the combination of the Longest Matching and PGLR techniques obtained the highest accuracy. Nonetheless, its processing time is not acceptable in a real-time system. If we combine this text analysis technique with the speech synthesis module, the system will not respond within an appropriate time frame. We notice that the combination of the Longest Matching and PGLR techniques is faster than the PGLR technique in Table 3.8 in Chapter 3 by three words in one second. Normally, the word segmentation requires time for processing, but when it is combined to the PGLR technique, it makes the PGLR technique work effectively by reducing the time in creating the parsed trees. Our approach obtains better accuracy than the dictionary-based technique, although it requires more space for two dictionaries and executable file of program (1.5 MB). This size is not too large for an embedded device. Therefore, the proposed technique is suitable for embedded devices.

We notice that in the last experiment, where sentences from the TsynC corpus was used as a test set instead of a list of words, all G2P approaches achieved higher accuracies. This indicates the real performance of the G2P approaches that the word occurrences are taken into the accuracy calculation

Table 4.5 The overall performance of the text analysis module

	Dictionary	Our approach	PGLR
Word accuracy	84.22	91.01	95.94
Syllable accuracy	90.02	93.73	97.92
Phone accuracy	91.33	96.09	98.87
# word processed in 1 sec.	101	97	14
Size(KB)	1,123	1,564	1,503

4.5 Conclusion

We proposed a technique in order to improve the accuracy of the G2P conversion of a Thai TTS system on a mobile device. Our proposed technique uses both a pronunciation dictionary of multiple-sized units (word and syllable) and a set of conversion rules to identify the pronunciation of each word. The syllable-level pronunciation dictionary and hand-written rules are used to handle unknown words. We evaluate the performance of the G2P conversion by the pronunciation accuracy at word, syllable, and phone levels, processing time, and the size of the program's footprint. We found that the proposed technique for Thai G2P conversion can identify pronunciations of unknown words with much better accuracy than a simple dictionary-based approach and also respond in real time. Although its accuracy is not the highest when we compare to the approach proposed for desktop computers, its short processing time makes it suitable for embedded devices.

Chapter 5

Conclusion and Future direction

In this thesis, we focus on the text analysis module for a Text-to-Speech (TTS) system on mobile devices. This module prepares information in terms of phone sequence to the speech synthesis module in order to generate output wave file. Since an embedded device such as mobile phone has poor resources both memory, and processing power, the technique, used in the text analysis, should require small size of memory and does not require more computational time. In the thesis, we contributed two main points.

Firstly, we compare and analyzed the existing text analysis algorithms. We found that a combination of the Longest Matching technique for word segmentation and the dictionary-based technique for G2P conversion are the most appropriate for the limited devices. Since those techniques do not need a long processing time. Moreover, in experiment, we constructed the word-level pronunciation dictionary from only the 22,969 most frequently-used entries. We use F-measure, processing time, and size of the program's footprint to evaluate the performance of the word segmentation experiment. The performance of the G2P conversion experiment is evaluated by accuracy at word, syllable, and phone, processing time, and size of program's footprint. We found that a main problem of the approach is a pronunciation of unknown word that is not generated. Therefore the accuracy of the overall performance text analysis is not so high.

Secondly, from the problems in the first part, we proposed a new G2P approach in order to improve the accuracy of G2P conversion by handling an unknown word's pronunciation. The proposed approach utilizes both a pronunciation dictionary of multiple-sized units (i.e. word and syllable) and a set of conversion rules. We measure the performance of pronunciation by accuracy in term of word, syllable, and phone level, the number of words processed in one second, and size of program. We notice that a syllable level pronunciation dictionary and set of conversion rules help increase more accuracies when comparing to only use of word level pronunciation dictionary. Moreover, it can respond in a real time, accordingly our proposed approach is suitable for an embedded device.

For the future works, the accuracy of word segmentation and the G2P conversion should be improved in a part of the quality of a dictionary by exploring an appropriate word unit and the types of words that should be included in dictionary. Moreover, the words and syllables that the pronunciation is often wrong should be analyzed in order to find cause of problem. After that, the selected algorithms could implement on an embedded device such as a mobile phone and then combine the text analysis module with the speech synthesis module to create a complete Thai TTS system.

Reference

[1] K. Kosawat, M. Boriboon, P. Chootrakool, A. Chotimongkol, S. Klaithin, S. Kongyoung, K. Kriengket, S. Phaholphinyo, S. Purodakananda, T. Thanakulwarapas, and C. Wutiwiwatchai, "BEST 2009 : Thai Word Segmentation Software Contest", in the Proceedings of SNLP 2009, Bangkok, Thailand, September 2009

[2] P. Tarsaku, V. Sornlertlamvanich and R. Tongprasert, "Thai Grapheme-to-Phoneme Using Probabilistic GLR parser" in the proceeding of Eurospeech Aalborg, Denmark 2001.

[3] S. Luksaneeyanawin, et al. "A Thai Text-to-Speech system" in the proceeding of 4th NECTEC conference, pp. 65-78, (in Thai)

[4] Y. Pooworawan, Dicitonary-based Thai Syllable Separation, Proceedings of the Ninth Electronics Engineering Conferences, 1986

[5] V. Sornlertlamvanich, "Word Segmentation for Thai in a Machine Translation system" (in Thai), papers on Natural Language processing, NECTEC, Thailand, 1995.

[6] Y. Thairarananond, "Towards the design of a Thai text syllable analyzer", Master thesis, Asian Institute of Technology, 1981.

[7] D. Sawamipak, "Developing Software for Analyzing Thai Syntax in Unix", Thammasat University Publishing, Bangkok, Thailand 1990 (in Thai)

[8] P. Charoenpornsawat "Feature-Based Thai word Segmentation" master thesis, department of Engineering, Chulalongkorn University.

[9] C. Haruechaiyasak, S. Kongyoung, and M. N. Dailey. "A Comparative Study on Thai Word Segmentation Approaches", in the proceedings of ECTI-CON. 2008, Krabi, Thailand, May 2008

[10] P. Mittrapiyanuruk, C. Hansakunbuntheung, V. Tesprasit and V. Sornlertlamvanich "Issues in Thai Text-to-Speech Synthesis: The NECTEC Approach," in the proceedings of NECTEC Annual Conference, Bangkok, Thailand, pp. 483-495, 2000.

[11] N. Chinathimatmongkhon, A. Suchato, and P. Punyabukkana, “Implementing Thai Text-to-Speech Synthesis for Hand-held Devices”, in the Proceedings of ECTI-CON 2008, Krabi, Thailand, May 2008

[12] A. Chotimongkol, and A. W Black, “Statistically trained orthographic to sound Models for Thai”, in the proceedings of ICSLP 2000, Beijing, China, 2000.

[13] W. Aroonmanakun, and W. Rivepiboon “A Unified Model of Thai Romanization and Word Segmentation”, in the proceeding of PACLIC 18, Tokyo, Japan 2004

[14] P. Charoenpornsawat and T. Schultz, “Example-Based Grapheme-to-Phoneme conversion for Thai”in the proceeding of Interspeech, Pittsburgh, PA, 2006.

[15] C. Hansakunbuntheung, V.Tesprasit, and V. Sornlertlamvanich, “Thai Tagged Speech Corpus for Speech Synthesis”, in the proceedings of the Oriental COCOSDA 2003, 2003, pp. 97&104

[16] LEXiTRON, a Thai-English electronic dictionary
Available: http://lexitron.nectec.or.th

[17] P. Chootrakool, C. Wuttiwiwatchai, and K. Kosawat, “A Large Pronunciation Dictionary for Thai Speech Processing” in the proceeding of ASIALEX 2009, Bangkok, Thailand, Aug 2009

[18] B. Sriman, “Thai Text to Phonetic Transcription using Linguistic Rule base and Longest Matching”.[in Thai] An independent study report submitted in partial fulfillment of the requirements for the degree of master of science in computer science, graduate school Khonkaen university, 2005

[19] P. Kramonlawech, All Tone in Thai Language. Horatanachai Publishers, Bangkok, 2005 (in Thai)

List of Publications

Thesis Title: Text Analysis for Thai Text-To-Speech Synthesis System on Embedded Device

Name of Student: Ms. Arunee Ratikan

School: School of Information and Communication Technology for Embedded Systems (ICTES)

Thesis Committee:

Advisor and Chairperson of Thesis Committee:	Asst. Prof. Cholwich, Ph.D.
Committee Member and Chairperson of Examination Committee:	Ananlada Chotimongkol, Ph.D.
Committee Member:	Assoc. Prof. Thanaruk Theeramunkong, Ph.D.
Committee Member:	Prof. Manabu Okumura, Ph.D.

Year of Graduation: 1st October / 2010

International Conference Proceedings

- Arunee Ratikan, Konlakorn Wongpatikaseree, Ananlada Chotimongkol, Ausdang Thangthai and Cholwich Nattee, "Comparison of Text analysis for Thai Text-to-Speech (TTS) application on mobile devices", *in the Proceeding of The first International Conference on Information and Communication Technology for Embedded Systems (ICICTES-2010),* Pathumthani, Thailand, 28-30 January 2010.

- Arunee Ratikan, Konlakorn Wongpatikaseree, Ananlada Chotimongkol, Cholwich Nattee, Thanaruk Theeramunkong, and Manabu Okumura, "Combining a Pronunciation Dictionary of Multiple-Sized Units and a Rules-Based Technique for Thai G2P Conversion", *in the Proceeding of The 7th International Joint Conference on Computer Science and Software Engineering (JCSSE 2010)*, Bangkok, Thailand, 12-14 May 2010, pp. 54-59.
- Konlakorn Wongpatikaseree, Arunee Ratikan, Ausdang Thangthai, Ananlada Chotimongkol and Cholwich Nattee, "A Real-time Thai Speech Synthesizer on a Mobile Device", *in the Proceeding of 8th Symposium on Natural Language Processing (SNLP-2009),* Bangkok Thailand, 20-21 October 2009, pp. 42 - 47.

- Konlakorn Wongpatikaseree, Arunee Ratikan, Ananlada Chotimongkol, Patcharika Chootrakool, Cholwich Nattee, Thanaruk Theeramunkong and Takao Kobayashi, "A Hybrid Diphone Speech Unit and a Speech Corpus Construction Technique for a Thai Text-to-Speech System on Mobile Devices", *in the Proceeding of ECTI-CON International Conference 2010*, Chiang Mai, Thailand, 19-21 May 2010.