Studies on
Thai Language Processing
with Emphasis on Input/Output Techniques

Mamoru SHIBAYAMA

December 1990

Studies on
Thai Language Processing
with Emphasis on Input/Output Techniques

Mamoru SHIBAYAMA

December 1990

Abstract

Written Thai differs from western languages, which is characterized by being unsegmental, that is, written without spaces between words and sentences, and phonetic.

Mechanical processing of Thai with emphasis on the linguistic approach to natural language processing therefore presents problems relating firstly to input/output methods, and secondly to successive levels of language processing such as morphological, lexical, syntactic, and semantic. However, few studies on natural language processing of Thai have currently been carried out.

This thesis introduces the implementation of a Thai input/output system consisting of an intelligent Thai computer terminal, a Thai syllable recognizer, and a Thai printing system, which represent the input, morphological, and output levels of language processing respectively. The system is based on several phonemic rules and definitions derived from linguistic analysis of the phonemes embedded in the syllabic structure and the orthography of Thai, which are here called the syllable formation rules.

At the input level, an intelligent Thai computer terminal with the function of automatic and consecutive conversion from Roman to Thai script has been designed. It employs a Transliteration Method (TM) or Simplified Transliteration Method (STM) using a transliteration table which can be applied for any given Roman spelling.

By using the proposed transliteration table, it is possible to reduce the number of keys on the keyboard by 46.7% compared with the method used for the ordinary IBM electronic Thai typewriter , called the Direct Mapping Method (DMM).

Estimated from the frequency of occurrence of each Thai letter in the text of the Three Seals Law (KTSD; Kotmai Tra Sam Duang), the number of key strokes required by the Transliteration Method was found to be greater than that required by the Direct Mapping Method, but 10.1% smaller than that required by the method proposed by J.F.Hartmann and G.M.Henry (1983).

An evaluation of learning effects from the number of key strokes and

the measurement of learning curves for the practical work of entering main entries of a Thai-Thai dictionary indicated that although the Transliteration Method required 42.9% more key strokes than the Direct Mapping Method, it achieved a 9.8% higher input rate in terms of characters per minute. Also, the average distance of movement of fingers was 0.630 for the Transliteration Method as compared to 0.822 for the Direct Mapping Method.

At the morphological level, segmentation was analyzed by the ordinary longest-match method for the input of the Three Seals Law. A revised method of segmentation, called the Syllable Longest-Match method (SLM), which incorporated a mechanism of back-tracking for each phoneme based on the syllable formation rules when the segmentation failed, was then revised to reduce the number of unsuccessful cases. This method indicated that the ratio of segmentation is a 98.0% in terms of sentences, which is 2.8% greater than value of 95.2% for the ordinary method.

In addition to the Syllable Longest-Match method used for the segmentation of a sentence into words of one or more syllables depending on the main entries of a dictionary, a finite automaton model which employs automatic segmentation from a sentence into monosyllables without reference to a dictionary, called the Thai syllable recognizer, was also proposed. In an experiment in segmentation for the input of the Three Seals Law, the recognizer gave a ratio of segmentation according to only the syllable formation rules of at most 49.6%.

A revised Thai syllable recognizer was also devised, in which knowledge rules based on the heuristics derived from the analysis of unsuccessful cases were adapted to the existing syllable formation rules. This gave a ratio of segmentation of 93.9% in terms of sentences for the input of the same text of the Three Seals Law.

The Thai printing system devised employs the function of high quality printing by choosing and combining the appropriate partial patterns of graphemes in the Thai letters based on the contextual constraints, and a table-driven context-sensitive optimization table is proposed for the system.

A printing strategy is also presented in which the syllable formation rules are adapted to allow justification of the right margin and synthesis of partial patterns of graphemes on a monitor screen.

Studies on

Thai Language Processing

with Emphasis on
Input/Output Techniques

Contents

Chapter 1 Introduction

1.1 Natural Language Processing and Thai Language

The availability of recent computer systems provides the ability of judging and deducing information in addition to the capabilities of memorizing and computing. Such increased capabilities of computer technology, of course, play an important role in the mechanical processing of natural language.

Language processing in computer-based information systems may summarily be divided into various portions as input, memorizing, processing and transformation of language information, including language understanding with decision and inference similar to that of human behavior, and the output of language information.

Those features and techniques with respect to each portion in the language processing are applicable to such systems as information retrieval, automatic question-answering, sophisticated processing of a document and full text, automatic abstracting, and machine translation systems [Nagao 84] [Salton 83].

However, many of the previous applications are currently approached by creating a simplified situation, for example, by restricting the allowable discourse area to a narrow part, or by impos-

ing limitations on the variety of natural language forms handled by the system because of the difficulties of complete linguistic analyses .

Those concerned with the design of information systems should now be concentrating on functional requirements for user-oriented, natural language systems of the future, and the linguistic approaches and techniques may have much more to offer the system designer than they have in the past [Sager 81].

Language processing is habitually recognized as having several different levels. These may be characterized as the phonological, morphological, lexical, syntactic, and semantic levels in addition to the man-machine interface at the input level. The framework composed of the main levels is shown in Fig.1.1.

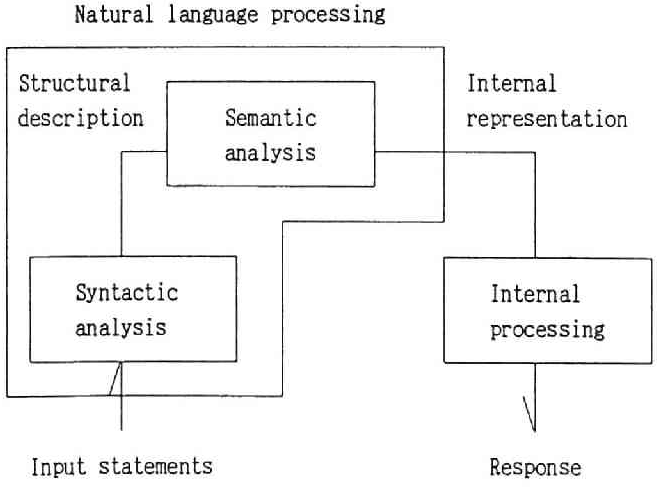


Fig.1.1 General framework of natural language processing [Tujii 88]

Chapter 1 Introduction

This thesis deals with the input/output and morphological levels in the above framework of natural language processing.

From the viewpoint of studies in the multi-lingual field, a project for developing a machine translation system which employs bidirectional translation via intermediate expression between English, Japanese and Asian languages has been pursued, and a development of an automatic telephone translation system also is one of the examples in the multi-lingual field [Yada 88].

Language processing with the writing system of non-Latin Script like Hebrew, Syrian, Arabic, Persian, Devanagari, and Burmese has the characteristics that letters differ from each other in size as well as shape.

Therefore, many more problems arise, especially in regard to input/output methods than occur with Latin script [Griffin 81].

Thai script treated in this thesis is the same as that of non-Latin script in the characteristics of input/output methods.

Furthermore, words embedded in a sentence in Thai has no spaces between them, and its script is more complicated as a writing system than others, such as Japanese, since punctuation is scarcely used [Allison 75].

Consequently, how to input/output Thai statements and the morphological analysis must be taken seriously in studies on natural language processing [Tanaka 89].

Chapter 1 Introduction

By retrieving bibliographic information concerned with the computer and information science, which are selected by the keyword "Thai" and other languages such as Arabic, Devanagari, Burmese, Laotian, Khmer, and Vietnamese from the titles of the database, INSPEC, that includes around 560,000 papers, the number of papers enumerated by each language and the field classification for Thai were obtained.

They are shown in Table 1.1 and 1.2 respectively. The number of papers for Japanese is 1,480.

Table 1.1
Enumeration of papers classified by languages in the database INSPEC

Language	Number of papers
Arabic	83
Hebrew	20
Devanagari, and Sanskrit	19
Tamil	4
Tibetan	4
Thai, Siamese	16
Vietnam	5
Burmese	2
Laos	2
Cambodia, Khmer	0
	155

Table 1.2
Field classification for Thai

Field of papers	
Character Recognition	7
Language Processing	3
Speech Synthesis	2
Documentation	2
Education	1
Library	1
	16

(Retrieved in April 1990)

Studies on the computer and/or information science for Thai as shown in Table 1.2 are fewer in number than those for Arabic, Hebrew or Japanese.

Chapter 1 Introduction

As for the studies on language processing for Thai, the contents of papers which are closely related to natural language processing are as follows; (1) Thai character handling on monitor screen including the internal code [Warotamasikkhadit 84], (2) an intelligent Thai computer terminal employing a transliteration method from Roman script to Thai letters [Shibayama & Hoshino 86], and (3) Thai syntax analysis using a simple Generalized Phrase Structure Grammars (GPSG) translator [Vorasucha 88].

It is important to refer to the studies on Thai romanization as the background for the studies on the input system for Thai because Thai letters are dealt with by using the ordinary keyboard and display screen configured to the computer.

This is the common method, not only for native speakers but also non-native speakers, who wish to handle Thai script on materials.

The first attempts to romanize Thai were made by Jesuit missionaries in 1620. A scheme that is very similar to the Library Congress (LC) romanization system as shown in Fig.A-1 in the Appendix was devised by Bishop Pallegoix in 854.

In the early decades of the 20th century, especially, King Rama IV in 1913 produced two tables, one for words of Indic origin and one for indigenous Thai, and King Vajiravudh who was King Rama IV's grandson proposed a new system that has romanization for Sanskrit and Pali loan-words, and for purely Siamese words [Diller 79].

Chapter 1 Introduction

In the studies of the Thai transliteration scheme using a computer, a transliteration system based on the above principle that vernacular Thai can be automatically generated by computer, alongside a romanized entry used for management and manipulation of bibliographic materials has been devised [Hartmann 83a], and a new transliteration table, here called Hartmann's Transliteration Method (HTM), has been proposed [Hartmann 83b].

As for syntax analysis and machine translation for Thai, Thai syntactic rules can be treated by the Generalized Phrase Structure Grammars (GPSG) in a very simple manner and a GPSG translator which translates GPSG's rules into the Definite Clause Grammar (DCG) format has been proposed [Vorasucha 87] [Vorasucha 88].

In addition, a machine translation system between Thai and English that has been used for the management of the rice mill correspondence exchanged by the rice exporters [Thajcayapong et al. 87].

In Thai text processing, a KWIC index of the Three Seal Law (KTSD; Kotmai Tra Sam Duang) [Ishii 69a] [Ishii 88b], which is virtually the single source for those who wish to study the 19th century legal texts of Thailand, has been made in the National Museum of Ethnology, and was published in 1981.

At the same time, an input/output method and sorting scheme have been presented [Sugita 80].

According to a plan for publishing a revised version of KTSD,

named "A Computer Concordance to KTSD", and building a database of the KTSD [Ishii 88b] [Shibayama 88], an automatic conversion system which employs the transliteration method from Roman script to Thai letters has been designed [Shibayama 85] [Shibayama & Hoshino 86], and its technique has been applied to an implementation of a Thai text editor [Shibayama 84].

Such contents as treated above are the subject of chapter 3 in this thesis. Besides, a comparative study of keyboard input and its learning characteristics for Thai input methods is introduced [Shibayama et al. 87], [Shibayama 87], and is also discussed in Chapter 4 of this thesis.

As for the control strategy of Thai characters on the display monitor, a display technique, that divides a character into 3 lines which are more controllable including the function of back spacing than a division by 4 lines, has been discussed [Waratamasikkhadit 84].

Printing methodology for Asian and African languages by dynamically dividing/synthesizing a character into/from 15 partial patterns has been introduced [Sakamoto 79]. And a synthesizing mechanism composed of three partial patterns and a context-sensitive formatter has been implemented in the case of the printing system for Thai and has been adopted to output the computer concordance to KTSD [Shibayama 87].

1.2 Outline of This Thesis

This thesis describes a Thai input/output system and the morphological analysis as the first phase or as the basis of a framework in natural language processing for Thai.

In the input methods for Thai, two input methods which employ a Roman-Thai conversion scheme based on the revised transliteration table have been proposed; (1)one is a Transliteration Method (TM) and another one is a Simplified Transliteration Method (STM).

This thesis shows that such methods provide effective manipulation of the keys for non-native speakers of Thai. To estimate the effectiveness between such input methods, accordingly, the quantitative measurement of the burden of memorization, input speed, and its learning characteristics using the text of the KTSD or Thai-Thai dictionary [Photchana nukrom Thai 82] is performed.

As a result, it is confirmed that the burden of memorization, the average distance of movement of fingers, and the learning curve of non-skilled operators in our input methods are more efficient than the ordinary one, for example, the IBM electronic Thai typewriter or the Hartmann's Transliteration Method (HTM).

In Thai text processing, the segmentation into word units segmented by the longest-match method, which is used for Japanese morphological analyses in general, has been discussed firstly, and it is followed by the automatic segmentation of a sentence without diction-

ary and its implementation for Thai. Then the result of the experiment of automatic segmentation obtained from the text of the KTSD is discussed compared with the longest-match method.

The next chapter gives preliminaries, which are divided into 5 sections. Section 2.1 introduces the characteristics of Thai text and dictionary, which are treated in the succeeding chapters 3, 4, and 5 to give the comparative study or to confirm the effect of the proposed algorithm quantitatively. Section 2.2 describes the orthography of Thai and its characteristics. Section 2.3 describes the general transliteration scheme for Thai and, to examine the key typing behavior, the concept and the estimation model for the learning characteristics are represented. Section 2.4 introduces the morphological analyses in the analytical levels for linguistic approach of natural language processing, and the longest-match and the character type division methods, that depend on studies on the morphological analyses of Japanese, are presented in order to adapt those methods to the segmentation of Thai sentences. Also, this section gives the fundamentals of a finite automaton model to formulate the automatic segmentation algorithm of Thai sentences in Chapter 5. Section 2.5 describes how to output non-Latin symbols including Thai onto output devices, and it focuses especially on the laser beam printer and its techniques.

Chapter 3 proposes the function of automatic and consecutive conversion, which employs a Transliteration Method (TM) and a trans-

literation table, from Roman script to Thai letters by any given Roman spelling. By estimating the number of key strokes from the frequency of occurrence of each Thai letter in the text of the KTSD, it is found that the number of key strokes required by the Transliteration Method is more effective than that required by the ordinary method such as the HTM. Section 3.1 is an introduction to Chapter 3. Section 3.2 gives the Thai alphabet table with the phonetic notation classified into 21 categories phonetically, and proposes a revised transliteration table, which is called the Transliteration Method. Section 3.3 embodies a transliteration algorithm based on the table in the previous section. Section 3.5 compares the Transliteration Method with other ordinary methods by estimating the number of key strokes for inputting the text of the KTSD, and shows that the Transliteration Method is effective for non-native speakers of Thai.

Chapter 4 describes an evaluation of the input methods for Thai based on the methodology and its characteristics by applying the results of measurements for such methods to a certain model of key typing behavior, which is divided into 5 sections. Section 4.1 is an introduction to Chapter 4. Section 4.2 introduces a Thai text editor and the two kind of input methods; Direct Mapping Method and Transliteration Method, and presents their characteristics. Section 4.3 describes the measurement of the number of key strokes for inputting the main entries of a Thai-Thai dictionary. Then, Section 4.4 applies the results of measurement to a certain model of learning characteristics in order to evaluate the learning effect. Section 4.5 introduces a Simplified Transliteration Method (STM) and the qualitative

comparison using the text of the KTSD compared with two other methods.

Chapter 5 describes the segmentation of Thai from a sentence to words using the well-known manner, the longest-match method, and proposes a revised longest-match method, which is called the Syllable Longest-Match method (SLM) which depends on syllable formation rules. Based on the results of heuristic analysis from an outcome obtained by executing the longest-match method for inputting the sentences of the KTSD, it is indicated that the SLM is the most effective method for the segmentation of Thai. Also, this chapter proposes a finite automaton model for automatic segmentation of a sentence without the dictionary, and shows a Thai syllable recognizer and the practical results of segmentation in the input of the same text of the KTSD. Chapter 5 is divided into 5 sections. Depending on the viewpoint of Thai language processing, Section 5.2 describes the rules which are used to formulate a word from the Thai symbols according to their orthography. Section 5.3 shows the results of two kinds of segmentation; one is segmented by the use of the longest-match method, and the other is segmented by the use of a newly revised method, SLM, which uses the back-tracking function based on the Thai syllable formation rules. Section 5.3 is divided into 4 sections; Section 5.3.1 and 5.3.2 deal with the environment and method for the experiment. Section 5.3.3 and 5.3.4 present the experiment of the ordinary longest-match and the Syllable Longest-Match methods respectively. Section 5.4 presents an automatic self-segmented model which is

called Thai Syllable Recognizer, based on the theory of finite state automata. According to the model, Section 5.5 shows the experiments of segmentation into monosyllables for inputting the text of the KTSD, and discusses the results of the experiment, an evaluation, and its effect for revising the model on the basis of knowledge rules derived from the heuristic analysis.

To output Thai letters to the printing devices or monitor screen with the vernacular Thai and high quality, Chapter 6 introduces an implementation of the Thai printing system which employs the controls of width, height, and adjustments of adjacent characters, and so on according to an algorithm of synthesizing partial patterns and the context-sensitive optimization table. Such techniques concerned with the monitor screen and the laser beam printer are treated in Section 6.2 and 6.3 respectively. In Section 6.4, a justification algorithm at the right margin based on the syllable formation rules is proposed.

The final chapter, Chapter 7, gives a summary and remaining subjects as concluding remarks.

Chapter 2 Preliminaries

2.1 Thai Language Processing

2.1.1 Thai Language Text and the Dictionary

Written Thai is characterized by being unsegmental, which is written without spaces between words in a sentence or between sentences. Thai language processing presents such problems as a man-machine interface that is concerned with the input/output method of text and the automatic segmentation from a sentence to words in the morphological level of natural language processing. The latter problem, especially, has a very similar property to that of Japanese characteristics which is unsegmental.

In language processing in which the language has the unsegmented characteristics, for instance, in the case of building the database, making a concordance or KWIC index, segmentation into the appropriate units as a word, a sentence, or a syllable for the original text may be required. The practical work of segmentation is too time-consuming and excessive, and reduces the accuracy and consistency of the result if the work depends on the effort and handiwork of an expert without the assistance of a computer or if automatic segmentation using the computer is impossible.

After three years efforts a computer concordance to the Three

Chapter 2 Preliminaries

Seals Law (KTSD: Kotmai Tra Sam Duang) was organized including the intensive work for half a year by an expert of the KTSD in the actual work for the segmentation from the original text into sentences and the re-segmentation into words, besides the work was based on the original text of machine readable form entered by Sugita (1979) at the National Museum of Ethnology.

Those who wish to progress language processing for making a concordance or KWIC index and for analyzing the text in the natural language processing would be comfortable if such segmentation could be performed by the computer.

In Thai language processing, there are no studies on the models, algorithms, and the evaluation associated with the segmentation for Thai text.

Consequently, this thesis discusses the automatic segmentation of Thai language in Chapter 5 in addition to an implementation of an intelligent Thai computer terminal in Chapter 3, the measurement and evaluation of Thai input methods in Chapter 4, and Thai printing system in Chapter 6 on the basis of the KTSD. The introduction to the KTSD is given in the following section.

The major reasons why the KTSD has been adopted for the estimation of learning effect for each input method and the measurement and evaluation of automatic segmentation as an experimental text of Thai language processing are as follows:

(1) The KTSD text is the largest one in existing text in machine-

readable form, and it is very useful for carrying out the studies on Thai history as mentioned below.

(2) The KTSD text covers all of the 44 consonants and a wide variation in the expression of statements in a period of 455 years, whereas the modern Thai writing system uses only 42 consonants in the sentences, and punctuation is sometimes used. Thus, it is considered that the KTSD text would provide more effective and extensive results for automatic segmentation.

(3) The KTSD text has already been segmented by the units of sentence and word by the handiwork of an expert. Therefore, that information of segmentation can be used to confirm the results of segmentation in the experiments.

The dictionary in the language processing with respect to the segmentation on the morphological level as well as the syntactic, and semantic analyses of natural language processing plays an important role in analyzing the text more efficiently.

As a result, a machine-readable Thai dictionary, with a total of about 31,200 words, composed of the main entries, has been newly made using our Thai text editor [Shibayama 84] as shown in Fig.A-2. This dictionary is based on the Thai-Thai dictionary published by Thai Royal Institute [Photchana nukorm Thai 82].

This dictionary is used for the experiment of Thai syllable recognizer in Chapter 5, and the learning characteristics of operators for the Thai input work when this dictionary input has been carried out is presented in detail in Chapter 4. Also, another dic-

tionary, with a total of 20,475 words, arranged by main entries in the computer concordance of the KTSD is used in the experiment of segmentation for the same text of the KTSD.

2.1.2 Kotmai Tra Sam Duang (KTSD)

The Kotmai Tra Sam Duang (KTSD), the Three Seals Law, is a popular appellation given to a corpus of traditional laws of Thailand. It was compiled in 1805 by the order of King Rama I of the Chakri Dynasty. The importance of the KTSD lies in the fact that it is virtually the single source for those who wish to study legal texts of Thailand from fourteenth until nineteenth century.

The name of KTSD is derived from the fact that on each volume of corpus are stamped all three official seals of the Ministers of the North, South, and Central, who were respectively in charge of the three provincial divisions constituting the traditional kingdom of Thailand.

The original corpus consists of four groups; the preface, the introduction, the body of laws, and the royal decrees. The body of laws is composed of 27 sections associated with the daily life of people, religious tradition, water irrigation, and so on.

The computerization project of the KTSD was planned and started for the first time in 1978 at the National Museum of Ethnology. A study group to make the KWIC index was organized at the laboratory of computer science with the wide participation of computer scientists

and Thai language specialists. The KWIC index to the KTSD, a total of 75 volumes, about 350,000 lines, was published in 1981.

A new project in which the author participated by making a computer concordance, "Datchani Kotmai Tra Sam Duang", and by building a database of the KTSD was started at The Center for Southeast Asian Studies, Kyoto University in 1983.

The computer concordance, "Datchani Kotmai Tra Sam Duang", a total of 5 volumes, about 5,000 B5 pages, was published in Thailand [Ishii et al. 90], and the service of information retrieval for the KTSD also was started in Data Processing Center, Kyoto University in 1990.

The source text of the KTSD, which is a copy of Thammasat University's version, is a total of 5 volumes, 1,700 pages, and about 34,000 lines, is as shown in Fig.2.1 [Khrusapha 61] [Ishii 69]. An example of the machine-readable form is shown in Fig.A-3 in the Appendix.

ประมวลกฎหมาย รัชกาลที่ ๑
จุลศักราช ๑๑๖๖
พิมพ์ตามฉะบับหลวง ตรา ๓ ดวง

คำนำ ของผู้ประศาสน์การมหาวิทยาลัย

เนื่องจากมหาวิทยาลัยวิชาธรรมศาสตร์และการเมืองได้กำหนดการศึกษาลักษณะวิชาประวัติศาสตร์กฎหมายไทยไว้ในหลักสูตรชั้นปริญญาโท จึงเป็นการจำเป็นที่จะต้องให้นักศึกษามีบทกฎหมายซึ่งได้ชำระสะสางในรัชกาลที่ ๑ นั้นไว้ค้นคว้าได้โดยสะดวก เพราะบทกฎหมายเหล่านี้ ถึงแม้ว่าจะได้พบเอกสารที่เก่าแก่กว่าก็ตาม ก็ยังคงใช้เป็นหลักฐานสำคัญเพื่อแสวงหาความรู้เรื่องกฎหมาย ในสมัยกรุงศรีอยุธยาได้ต่อไปอีกนาน

ตั้งแต่ปลายรัชกาลที่ ๓ เป็นต้นมา ประมวลกฎหมายในรัชกาลที่ ๑ นั้นได้มีผู้จัดพิมพ์ขึ้นหลายครั้งแล้ว ฉะบับที่แพร่หลายที่สุดคือฉะบับของหมอบรัดเลย์ ซึ่งเรียกกันว่ากฎหมายเก่า ๒ เล่ม และฉะบับของเสด็จในกรมหลวงราชบุรีฯ ทั้งสองฉะบับนี้ในปัจจุบันไม่มีจำหน่ายแล้ว และนับวันจะหายากขึ้นทุกที

Fig.2.1 A part of the KTSD (The Three Seals Law)

2.2 Thai Orthography

The Thai writing system is one of many varieties of the Devanagari writing system which have spread from India. The immediate source is the Cambodian variety. The particular adaptation of the Thai alphabet as a separate system of writing was devised by Ramkhamhaeng the Great of Sukhothai. The first written monument, an engraved block of stone known as the Inscription of King Ramkhamhaeng, is assigned to the year 1283 A.D. The modern Thai writing system is directly derived from a form of writing preserved in this earliest inscription, though certain changes have been introduced since that time [Haas 56].

2.2.1 Syllable and Word Formation

Written Thai differs from western languages in several points.

Thai letters are phonetic; they are monosyllables with 5 tones and are composed of "Constant + Vowel" or "Consonant + Vowel + Consonant". The constants, vowels, and tones must be placed in appropriate positions as shown in Fig.2.2.

The line of the central region is called the base line. Each consonant or vowel is defined as a phoneme. Hence, a syllable consists of several phonemes, and the smallest unit which forms a word in Thai is made up of one monosyllable or more syllables.

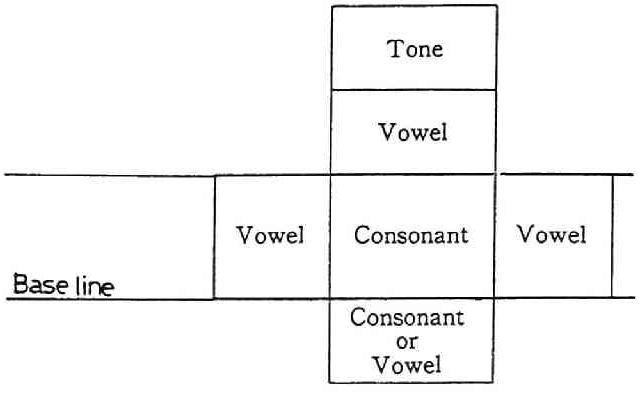


Fig.2.2 Positions for consonants, vowels, and tones

Consonants

Thai letters consist of 44 consonants as shown in Fig.2.3 (a). However, only slightly more than one-half of them are used extensively. The consonant sounds at the beginning/end of a syllable and the relation of Thai consonants to other languages are beyond the scope of this thesis, except the phonemes related with the transliteration.

Also, a syllable is defined that it consists of several graphemes based on the Thai orthography and these are the phonemes of spoken language. Eighty graphemes are needed to express all of the Thai letters. Every Thai word has at least one written consonant since vowels must always be used with the consonant. The rare vowels as shown in No.31 and No.32 of Fig.2-3(b)

NO.	Letter	Pronun-ciation	NO.	Letter	Pronun-ciation	NO.	Letter	Pronun-ciation	NO.	Letter	Pronun-ciation
1	ก	ko:	12	ฉ	cho:	23	ท	tho:	34	ย	jo:
2	ข	khǒ:	13	ช	jo:	24	ธ	tho:	35	ร	ro:
3	ฃ	khǒ:	14	ฌ	do:	25	น	no:	36	ล	lo:
4	ค	kho:	15	ญ	to:	26	บ	bo:	37	ว	wo:
5	ฅ	kho:	16	ฐ	thǒ:	27	ป	po:	38	ศ	sǒ:
6	ฆ	kho:	17	ฑ	tho:	28	ผ	phǒ:	39	ษ	sǒ:
7	ง	go:	18	ฒ	tho:	29	ฝ	fǒ:	40	ส	sǒ:
8	จ	co:	19	ณ	no:	30	พ	pho:	41	ห	hǒ:
9	ฉ	chǒ:	20	ด	do:	31	ฟ	fo:	42	ฬ	lo:
10	ช	cho:	21	ต	to:	32	ภ	pho:	43	อ	?o:
11	ซ	so:	22	ถ	thǒ:	33	ม	mo:	44	ฮ	ho:

(a) Consonants

NO.	Letter	Pronunciation	NO.	Letter	Pronunciation
1	-ะ, -ั-	(a?), (a)	17	ไ-	(aj)
2	ิ	(i?), (i)	18	เ-า	(aw)
3	ึ	(ɯ?) (ɯ)	19	-า	(a:)
4	ุ	(u?), (u)	20	ี	(i:)
5	เ-ะ, เ็	(e?), (e)	21	ื, ึ	(ɯ:)
6	แ-ะ, แ็	(ɛ?), (ɛ)	22	ู	(u:)
7	โ-ะ, - -	(o?), (o)	23	เ-	(e:)
8	เ-าะ, ็อ	(ɔ?), (ɔ)	24	แ-	(ɛ:)
9	เ-อะ	(ə?)	25	โ-	(o:)
10	เ็ียะ	(ia?)	26	-อ	(ɔ:)
11	เ็ือะ	(ɯa?)	27	เ-อ, เ็	(ə:)
12	็วะ	(ua?)	28	เ็ีย	(i:a)
13	ฤ	(rɯ), (ri)	29	เ็ือ	(ɯa)
14	ฦ	(lɯ)	30	็ว, -ว-	(ua)
15	ำ	(am)	31	ฤๅ	(rɯ:)
16	ใ-	(aj)	32	ฦๅ	(lɯ:)

(b) Vowels

Fig.2.3 Thai letters [Kawabe 80]

are also listed as consonants in this thesis as well as in the Thai dictionaries since they are the vowels which may be written alone, without any accompanying consonant.

Vowels

The Thai language uses a total of 32 basic vowels as shown in Fig.2-3(b), plus a few vowel substitutes. For example, a few consonants are used as vowels or in the vowel combinations (See No.8-12, 21, and 26-30 in Fig.2-3(b)).

Each vowel has a unchanging form and no inflection. A vowel sound is sometimes unwritten, and a vowel sound written alone or coming at the beginning of syllable must be written with the silent consonant like the pronunciation [a:] (See No.43 of (a) and No.19 of (b) in Fig.2-3).

A vowel may consist of more than one grapheme, for example, No.5-12, 18, 21, and 27-30 in Fig.2-3(b), and each vowel is always written next to the consonant(s) with which it is used: either before, after, above, under, or in a combination of positions as mentioned.

Tones

The Thai language uses 5 tones, normal, low, rising, falling, and high, discriminated by four written tonal marks to distinguish meanings. The tone rules apply only to syllables and, therefore, a word having more than one syllable may have more than one tone, and

the position in spelling is in the order in which it appears in the syllable.

System of writing

Thai writing is from left to right, the same as Western languages, but vowels, tonal marks, and other symbols are not restricted to positions on the writing base line. Normally, Thai words do not have spaces between them. Such a language is called as being unsegmental. Also, punctuation is scarcely used.

2.2.2 Thai Grammar

Thai grammar is easier to learn in many respects than that of western languages. Thai verbs are never conjugated and, therefore, variations in person or tense are shown by other words or by the contextual inference. The typical forms of the Thai sentence are as follows:

(1) s -> vp
(2) s -> vp objd
(3) s -> subj vp objd
(4) s -> objd subj vp
(5) s -> vp subj
(6) s -> subj vp
(7) s -> vp objd obji
(8) s -> subj vp obji
(9) s -> obji subj vp objd
(10)s -> objd subj vp obji

Here, the symbol 's' means sentence, and the symbols 'subj', 'vp', 'objd' ,and 'obji' represent the subject, verb phrase, direct object, and indirect object respectively [Vorasucha 88].

The characteristics of Thai are illustrated as follows:

(1)She is beautiful.

เขา สวย
khao suai
(She) (beautiful)

(2)I love you.

ผม รัก คุณ
phom rag khun
(I) (love) (you)

(3)It is raining heavily.

มี ฝน ตก มาก
mii fon tog maag
(there is) (rain) (fall) (a lot of)

(4)Yesterday, I didn't go anywhere.

เมื่อวานนี้ ผม ไม่ ไป ที่ไหน
mwawaannii phom mai pai thiinai
(yesterday) (I) (not) (go) (anywhere)

(5)I am taking him to see a movie.

ผม พา เขา ไป ดู หนัง
phom phaa khao pai duu nan
(I) (take) (him) (go) (see) (movie)

In these examples, spaces have been inserted between words for clarity, although this is not normal practice. The characteristics of these statements are as follows. (1) The word order is subject +

adjective verb (S + AV), and no "be" verb is used. (2) The form is subject + verb + object, as in the corresponding English (S + V + O). (3) This statement has the "be" verb mii, and corresponds to the form "There is ..." in English. (4) This is an example of the past tense. This sentence would be in the present without the word mwawaannii ("yesterday"). (5) This statement has a direct object and complement after the subject and verb (S + V + O + C) [Shibayama 89].

2.3 Transliteration Schemes and Learning Characteristics

Translation for an orthography from/to a symbolic form like Roman-spelling which uniquely represents, for example, the pronunciation is defined as 'Transliteration', whereas the word 'Transcription' is used when translation between an orthography and the phonetic realization using the romanization with the diacritical marks is executed.

An orthographic-to-phonemic translation method makes it possible to represent the pronunciation of statements made in any language in a way which allows the generation of a reasonable approximation of natural speech [Chomyzyn 86]. Such translation is considered very important when applied to Thai input/output methods, but it is beyond the scope of this thesis.

2.3.1 Transliteration Schemes

The transliteration schemes, in which a correlation between a

letter on the keyboard and a letter in the script to be displayed /printed may be established, requires the following points:

(1) It should be easy to master, so that with a little practice it becomes quite natural. Each letter in the transliteration should as nearly as possible represent the original pronunciation.

(2) Since transliteration is based on the pronunciation of its spoken language, a letter may be best represented by more than one Roman character. For example, the Thai letters 'ป' and 'พ' are commonly represented by the Latin 'P' and 'PH'. In case of an orthography, like Thai, Devanagari, or so on, has a much larger number of letters than the Roman alphabet, it is usually best to assign a multi-character sequence in the Roman alphabet for a single letter in the language, for example, 'KH1' and 'KH2'.

(3) The transliteration must be unambiguous [Griffin 81].

A lack of a precise romanization can lead to false starts and failure. J.F. Hartmann and G.M. Henry proposed a transliteration system [Hartmann & Henry 83a] to eliminate this problem by which a Thai letter can be automatically generated by a computer alongside a romanized entry used for management and manipulation of bibliographic materials.

This system which employs the function of parallel display for both a romanized entry and a generated Thai letter allows users to confirm whether any error of transliteration occurred or not, and is

implemented on an Apple II Plus computer. A part of the transliteration table which was proposed by J.M.Hartmann is shown in Fig.2.4.

Fig.2.4 Transliteration table by J.H.Hartmann

2.3.2 Modeling of Typing Speed and Learning Effect

In the mechanical processing of natural language, such problems associated with the input of text such as which method and machine will be used, especially from the view of man-machine interface may be offered.

In order to make progress in the investigation of input methods for Thai, study should be based on the consequences underlying Japanese language processing that have already been experienced, since characteristics of Thai, with no spaces embedded in the sentence, make it an unsegmental language similar to Japanese.

In order to give the comparison between the input methods for Thai and Japanese in the following chapters, the Japanese input scheme and its characteristics in accordance with emphasis on the

number of key strokes and the learning effect are summarized. Input methods for Japanese are mainly as follows:

(1) Multi-shift Type Input Method
(2) Total Character Assignment Method
(3) Code Input Method
(4) Kana-Kanji Translation Method
(5) Pattern Recognition Method

In the Multi-shift Type Input Method, the range of the number of shift keys is 4 to 30, and the number of Kanji characters capable of being stored is about 2,300 to 5,300. Average speed of input in a 12-shift machine (2,304 characters) is 53.6 characters/minute [Watanabe 82].

In the Pen-touch Input Method which is one of the total character assignment methods, Kanji characters can be input at 45.5 characters/minute [Shutoh & Itoh 82].

The two-stroke method, which is a typical code input method, employs the input for Japanese statements and is capable of about 99% by typing an arbitrary 2 strokes out of 49 keys, and the critical number of input characters are theoretically 200 characters/minute [Murayama 82].

In the Kana-Kanji translation method, many basic and applied investigations, for example, automatic segmentation for transformation of Kana into Kanji, Kana alphabet to Kanji conversion system, and so on have been performed and such experimental systems have been

made [Makino 82]. Nowadays, this scheme is very popular.

In the input method by pattern recognition, the Kanji OCR can recognize the printed text of Japanese modern and ancient literatures by 98% and 83% respectively more with the speed of 30 characters per second, and may also be popular in near future [Horiike & Hoshino 90].

The evaluation of the learning effect in regard to the input speed in which a system has an arbitrary input method and Keyboard assignment is shown in Fig.2.5 in general.

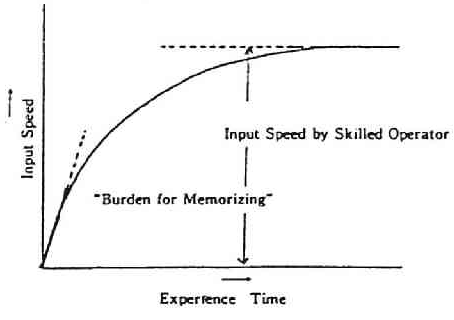


Fig.2.5 Learning curve of input speed [Morita 87]

Since the learning curve of a trainee or operator, which shows the distribution of input speed according to their experience time, varies person to person, it is difficult to indicate the practical characteristics for the learning curve.

Consequently, suppose the following two items characterize the

theoretical value of a learning curve using 2 items; one is the input speed by a skilled operator and another is the burden of memorization. The burden of memorization, which is determined by the number of keys and a certain manner of key assignment on the keyboard, is defined by the load of endeavor necessary for memorizing the key assignment.

According to the definition, the burden for memorizing, here called BM, with all of the keys on the keyboard is in proportion to the following expression;

$$BM \propto n \cdot \log_2 n \ ,$$

where n indicates the number of keys on the keyboard [Morita 87]. An estimation model for the typing speed is shown in Fig 2.6.

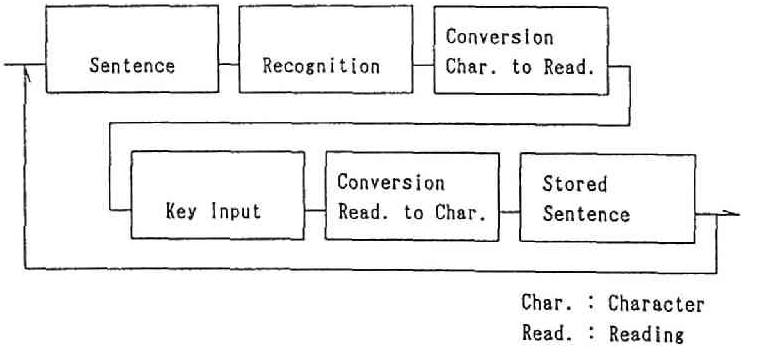


Fig.2.6 Estimation model for the key typing speed

Figure 2.6 is built depending on the analysis of Japanese text input, but it can be applied to the design of a model for Thai text input. The characteristic of key input speed, which is proposed by

C.L.Hill (1943), in other words, the number of key strokes per minute is given by

$$S(t) = M \left(1 - e^{-G(t+t_0)} \right),$$

where the function S(t) of time t means the number of key strokes per minute after the experience time t; M, G, and t_0 mean the saturated number of key strokes, the coefficient of starting, and the preliminary experience time in terms of hour respectively [Nakayama & Kurosu 84a].

The saturated number of key strokes M is related to the keyboard assignment, namely, M appears as

$$M = \frac{60}{T} \qquad \text{(strokes/min.)},$$

where T means the average time for a stroke.

For the overall key input in the arbitrary text, the average time for a key stroke is evolved as follows:

$$T = \sum_{i}^{n} \sum_{j}^{n} p_{ij} \cdot t_{ij},$$

where p_{ij}, t_{ij}, and n are the frequency of occurrence of adjacent two characters i and j, the average stroke time (second/stroke) for the arbitrary adjacent two characters, and the total number of characters respectively [Nakayama & Kurosu 84b].

The frequency of occurrence of characters depends on the language provides p_{ij}.

Let t_{ij} be defined as follows:

$$t_{ij} = \frac{1}{2}\left(\frac{R_{ki}}{Q_{hfi}} + \frac{c \cdot R_{kj}}{Q_{hfj}} \right),$$

where Q_{hf} is a coefficient of stroke speed according to the hand and finger, and is quoted from Table 2.1.

Table 2.1 Typing ability of fingers

Fingers		Hands	
f	Position	Left:h=1	Right:h=2
1	Thumb	65.0	70.0
2	Forefinger	97.1	100.0
3	Middle finger	88.5	94.2
4	Third finger	83.5	89.7
5	Little finger	75.9	78.7

Coefficient c is shown as follows:

1) When the previous stroke is by the same hand, but by a different finger, and in the inside direction, then c=0.9.

2) When the previous stroke is by the same hand, but by a different finger, and in the outside direction, then c=1.0.

3) When the previous stroke is by the same hand, by the same finger, but by a different key, then c=1.3.

4) When the previous stroke is by the same hand, by the same finger, and by the same key, then c=1.1.

5) When the previous stroke uses the other hand, then c=0.8.

6) When the j-th stroke is in accompany of the shift-key at the same time, then c=1.5*c.

R_k is defined as follows;

$$R_k = a \log_2 \left(\frac{d_k}{d_o} + 0.5 \right) ,$$

where R_k means the basic time necessary for finger movement, and d_k is the distance from the home key to a target key k(mm), d_o is the width of the key top, and a is the coefficient of time which is a constant 3.67.

2.4 Morphological Analyses and Finite State Automata

2.4.1 Morphological Analyses

It is customary to recognize several different levels of language processing as shown in Chapter 1. This thesis deals with the morphological analysis in the framework of natural language processing.

The morphological analysis which is to recognize the individual

morpheme embedded in a sentence composed of morphemes, is concerned with the processing of individual words form and of recognizable portions of words.

The major roles of the morphological analysis are as follows:

[1] To recognize the least linguistic unit in a sentence.

(1) Segmentation into the individual word form in the case of unsegmental language, here called automatic segmentation.

(2) The recognition and removal of word suffixes and prefixes and the generation of word stems.

(3) Separation of compound nouns or verbs into the morphemes.

In the case of Thai, the second item is not relevant because no conjugation is used in Thai.

[2] To investigate the adequacy of analysis after recognizing the morpheme.

The investigation of adequacy in the processing of recognizable portions of words is followed by syntactic, semantic, and contextual analyses. Furthermore, not only the extraction of all recognizable successions but also the syntactic or heuristic information makes morphological analysis possible.

These problems are concerned with the ambiguities underlying the result of the morphological analysis that may be increased if the characteristics of object language which is to be analyzed is unsegmental like Japanese.

The reasons for the ambiguity, for example, are mainly as follows:

(1) No space is written between words and all words are connected in a sentence.

(2) They have almost no rules for segmentation and punctuation.

(3) There are many prefixes and suffixes, and appropriate rules for generating a compound word are not well-established.

(4) By adjoining an arbitrary number of characters, for example, an arbitrary Kanji string, a word can be made relatively freely.

To advance the studies on the morphological analysis for Thai, it is important to clarify such problems since the characteristic of Thai is written without spaces as same as that of Japanese.

(a) アルプスの少女は美しい

Longest-match part	Remaining string
アルプス	の少女は美しい
の	少女は美しい
少女	は美しい
は	美しい
美しい	

(b) アルプスのやまは美しい

Longest-match part	Remaining string
アルプス	のやまは美しい
のやま	は美しい
は	美しい
美しい	

Fig.2.7 Right-directed Longest-Match method for Japanese sentences

Longest-Match method

As one of the methods used in the morphological analysis, the longest-match method according to the dictionary is used for the unsegmental statement. Figure 2.7 illustrates the process analyzed by the longest-match method for two Japanese sentences. This is an example with ambiguity in a sentence. The error ratio in the longest-match method, in general, is 20% obtained experimentally [Tanaka 89]. When any error of segmentation occurs, the back-tracking method such as the depth first method is used so that the analysis is brought back to the previous phrase location.

The problem with the longest-match method is that, if as long a word or phrase as possible at the left side of sentence is extracted, the remaining string will inevitably be shorter. To eliminate such a tendency, the Least BUNSETSU's Number Method, which evaluates the result of the analysis based on the number of phrases as the outcome of the analysis, has been proposed, and which error ratio of segmentation is about 4% [Yoshimura et al. 83]. This technique includes the function of back-tracking and may be applicable to the segmentation of Thai statements, but those concerned with it have not been reported it anyway.

Character Type Division Method

As another method used in the morphological analysis, the segmentation of sentences based on the character type information, which is segmented at a location where the character class in the string

changes, here called the character type division method, is also useful in Kana-Kanji mixed sentences of Japanese.

Such types composed of the character class are, for example, as follows;

(1) Punctuation
(2) Kanji, Kana, and Roman alphabet
(3) Numeral
(4) Others

In the character type division method, analysis with the aid of the dictionary may be much more efficient, and it is reported that the error ratio for analyzing 500 phrases is 3.5% [Nagao et al. 78]. Also, the automatic segmentation of Hiragana strings using the characteristics of a string has been reported [Tanaka & Koga 81].

2.4.2 Finite Automaton Model

Computer programs in which the analysis of the sequence of input symbols plays a central role in determining the program actions are syntax-directed programs, for example, editors, text formatters, command interpreters, and so on. To implement the syntax-directed program as a Thai syllable recognizer in Chapter 5, the finite automaton model is adopted.

In the Thai syllable recognizer, the strings, which are finite in length, is treated as an ordered sequence of symbols. A set of strings S is defined as follows:

$$S = s^*,$$

where s^* is all the strings formed by an arbitrary finite sequence of symbols including a string of zero length. An input sentence is a particular subset of s^*.

For programmed realization of such finite state automaton, the state, being a member of a finite set, is represented by the state variable, and it is denoted by K as follows:

$$K = \{ q_0, q_1, \ldots\ldots, q_n \}.$$

For this recognizer, when a input string denoted by s is given, the configuration of the automaton is defined as pair (q_i,s). Here, the variable q_i and s show the present state and the remaining input string respectively, and i=0,1, ... ,n.

Also, a set of transition functions is denoted by P, and each transition function, which defines a unique new state q_j for each (q_i,s), is expressed as follows:

$$d(q_i,s) = q_j.$$

From these definitions, therefore, such finite state automaton (FSA) is denoted by M as follows:

$$M = \langle K,S,P,q_0,F \rangle,$$

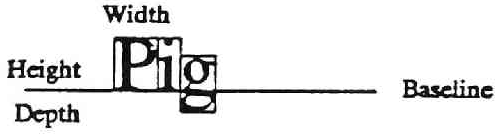
where F is a set of the final states, which is composed of the elements of the state q_i [Gough 88]. Furthermore, in the state diagrams of Chapter 5, the non-deterministic finite automaton (NFSA), which is that the next state function is not uniquely defined for all possible (state, symbol) pairs, is adapted.

2.5 Thai Printing Techniques

The printing goal for the development of the Thai printing system is to produce Thai script with the letters set adequately larger and well-formed to satisfy a user's calligraphic preference.

The printing systems, in general, are divided into two transition stages; formatter and device driver. The formatter uses the font library to obtain all the detailed size and positioning information it needs to decide exactly how to place each character. The device drivers may also require information from the font library [Brown 81]. A few widely available formatters which are device-independent have recently been delivered, for example, T_EX [Knuth 79].

Font information which is required by the formatter mainly for Roman script, for example T_EX, is needed as follows:
(1) Information on the height, width, and depth of the character. The formatter deals with rectangular boxes because of no knowledge of the shapes of the characters it handles as shown below:



(2) Ligature information

Substitutions of certain characters are performed, in some fonts, by composite characters known as ligatures. The most commonly

used ligatures are made up of 'f' followed by 'i', or 'f' and 'l'.

(3) Kerning information

Careful spacing is required because of the relative shape of two adjacent characters, for example, a capital 'W' for 'A'.

(4) Extension pieces

For specialized fonts, for example, especially a mathematical font, it may be necessary to identify similar characters in different sizes or to build large characters out of a number of separate pieces.

In addition to all this information about individual characters, general information such as the design size of a font, the minimum, normal, and maximum width of a space in the font, and the slant of the font may be needed as a whole.

Furthermore, in a non-Latin printing system like Thai, Khmer, and so on the following points must be considered.

(1) Differences of shape and size among the letters are much larger than that of characters in the Roman alphabet. Hence, a letter must be divided into several partials, for example, 15 partials for Khmer [Sakamoto 79].

(2) The synthesizing techniques which generate a new letter by substituting the several letters depend on the orthography must be required, for example, for the synthesizing of Arabic orthography [Hoskins & McMaster 81].

(3) The dead-key control mechanism required for carriage advance of a

specialized typewriter like Thai and Devanagari must be considered [Miller & Glover 81].

Chapter 3 Intelligent Thai Computer Terminal

3.1 Introduction

Written Thai differs from western languages in several points; (a) Thai letters are phonetic; they are monosyllables accompanying 5 tones and are composed of "Consonant + Vowel" or "Consonant + Vowel + Consonant". (b) The words in a sentence are not separated from each other. (c) Some vowels are placed before the consonant in the writing system. (d) Punctuation is scarcely used. Thai letters consist of 44 consonants and 32 vowels (excluding compound vowels) as shown in Fig.2.3 of the preceding chapter.

For the mechanical processing of Thai text having such characteristics, the input/output methods for Thai and Thai letters, especially in the natural language form, are of major importance and present sophisticated problems.

By viewing the input methods for Thai compared with the basis of the Japanese input methods described in Section 2.3.2 of Chapter 2, the following input methods are considered.

(1) Total character assignment method

(2) Translation method

(3) Pattern recognition method

Another two methods, multi-shift type input and code input

methods described in the preceding chapter, need not be considered in the case of Thai, because the number of Thai letters is smaller than in Japanese. The pattern recognition method also is not a direct object of implementation because that is image processing, in contrast to character or string processing by using a keyboard.

Consequently, two input methods are considered.

On the IBM electronic typewriter which is a type of total character assignment method that is currently popular, each key corresponds to one Thai letter. This key assignment was used by Sugita [Sugita 80]. In this mode, the Thai text is entered by keys corresponding uniquely to Thai letters, as shown in Fig.3.1.

The input mode using Roman spelling proposed by J.F.Hartmann and G.M.Henry [Hartmann & Henry 83] has also been adopted in the terminal. This method which corresponds to the translation method in Roman Kanji conversion is a transliteration approach for the generation of Thai letters from Roman spelling according to their pronunciations, here called the Transliteration Method. For example, "TH", "TH1", "TH2", "TH3", "TH4", and "TH5" are used respectively for ท, ฑ, ธ, ฐ, ฑ, and ฒ.

Another sophisticated problem is that several Thai letters differ from each other in the kind of letter, namely, the shape, whereas their phonemics completely coincide with each other. The consideration for improvement of man-machine interface which directly

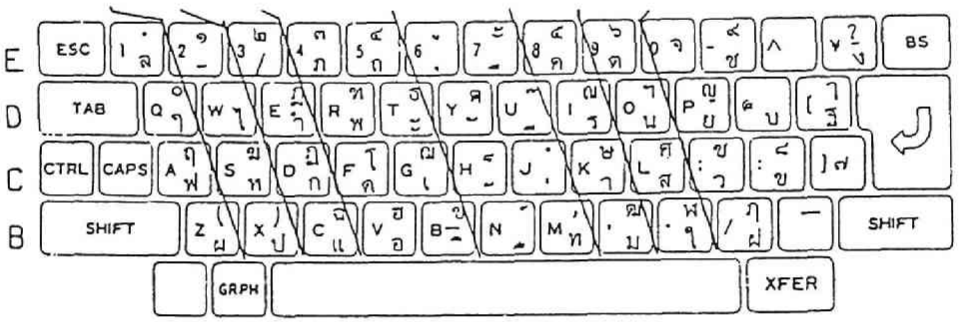


Fig.3.1 Keyboard assignment for DMM

(a) Consonants

GN	LCN 1	2	3	4	5	6	7	8
1	K	KH	KH1	KH2	KH3	KH4		
	ก	ค	ข	ฆ	ฃ	ฅ		
2	C	CH	CH1	CH2				
	จ	ช	ฌ	ฉ				
3	D	D1						
	ด	ฎ						
4	T	TH	TH1	TH2	TH3	TH4	TH5	T1
	ต	ท	ถ	ธ	ฐ	ฑ	ฒ	ฏ
5	N	N1	NG					
	น	ณ	ง					
6	P	PH	PH1	PH2				
	ป	พ	ผ	ภ				
7	F	F1						
	ฝ	ฟ						
8	L	L1	L2	LEU				
	ล	ฬ	ฦ *	ฦๅ*				

GN : Group Number
LCN : Local Classification Number

GN	LCN 1	2	3	4
9	R	R1	REU	
	ร	ฤ *	ฤๅ*	
10	Y	Y1		
	ย	ญ		
11	S	S1	S2	S3
	ส	ษ	ศ	ซ
12	H	H1		
	ห	ฮ		
13	B			
	บ			
14	M			
	ม			
15	W			
	ว			
16	?			
	อ			

*: Vowel

Fig.3.2 Transliteration table for consonants and vowels (continued)

(b) Vowels

GN	LCN 1	2	3	4	5	6	7	8	9	10	11
17	A	A-	A:	AI	AII	AE	AE-	AE:	AM	AW	A.
	◌ั -	- ะ	- า	ไ -	ใ -	แ◌ั	แ - ะ	แ -	- ำ	เ - า	-
18	I	I:	IA	IA-							
	◌ิ	◌ี	เ - ีย	เ - ียะ							
19	U	U:	UA	UA-	UAI						
	◌ุ	◌ู	◌ัว	◌ัวะ	- ว -						
20	E	E-	E:	EU	EU:	EUI:	EUA	EUA-			
	เ◌ั	เ - ะ	เ -	◌ึ	◌ื	◌ือ	เ◌ือ	เ◌ือะ			
21	O	O:	OU	OU-	OU:	OE	OE:	OE-	O.		
	โ - ะ	โ -	◌็อ	เ - าะ	- อ	เ◌	เ - อ	เ - อะ			

Fig.3.2 Transliteration table for consonants and vowels

GN / LCN	22	23	24	25	26	27	28	29	30	31
1	0	1	2	3	4	5	6	7	8	9
	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙
GN / LCN	32	33	34	35	36	37	38			
1	'	<	>	+	Q	Z	V			
	◌่	◌้	◌๊	◌๋	ๆ	ๆ	◌์			
GN / LCN	39	40	41	42	43	44	45	46	47	48
1	/	*	(	)	,	.	−	:	sp	@
	/	*	(	)	,	.	-	:	sp	@

GN:49, LCN:1 Carriage Return

Fig.3.3 Transliteration table for numerals, tones, special symbols, and control codes

affects the behavior of typing may arise from the correlation between the letters on the keyboard and the letters to be displayed.

This chapter presents an algorithm for transliteration and proposes the use of the devised transliteration table. Furthermore, the Direct Mapping Method, our transliteration method, and the scheme of J.F. Hartmann and G. M. Henry, here called HTM, are compared and evaluated by the number of key strokes estimated from the frequency of occurrence of every Thai letter in the text of the Three Seals Law (Kotmai Tra Sam Duang).

3.2 Classification of Thai Letters

The Thai consonants shown in Fig.2.3(a) can be classified phonetically into 21 categories. Accordingly, the Roman spellings of Thai letters were classified into 21 groups as shown in Fig.3.2. Each group comprises character strings headed by the same Roman character. The numerals, tones, special symbols, and control codes including the carriage return are classified into 28 groups each of one character, as shown in Fig.3.3. A group number, GN, is assigned to each of these 49 groups. Each character string in a group is discriminated by a local classification number, LCN. Thus we can define $R=\{r_i\}$, $r_i=\{r^*_{i,j}\}$, where i and j are the GN and LCN of the character string $r^*_{i,j}$. For example, $r^*_{4,3}$ refers to TH1 in Fig.3.2.

This transliteration table was used to implement the Thai intelligent terminal.

The characters are classified as follows:

(1) Subset CV: Consonants and Vowels CV=$\{r_1, r_2, \ldots, r_{21}\}$

(2) Subset NT: Numerals and Tones NT=$\{r_{22}, r_{23}, \ldots, r_{35}\}$

(3) Subset SS: Special Symbols SS=$\{r_{36}, r_{37}, \ldots, r_{47}\}$

(4) Subset CC: Control Codes for Shift Code and CR (Carriage Return) CC=$\{r_{48}, r_{49}\}$

The characters following the initial characters of the Roman spellings shown in Figs.3.2 and 3.3 are 16 in number, i.e.,"H", "1", "2", "3", "4", "5", "G", "-", ":", "A", "I", "U", "E", "M", "W", and ".", and are denoted by c_1, c_2, ... , c_{16}. For example, c_7 means "G". Thus we can define the set RC and SC_i for $1 \leq i \leq 16$ such that RC=$\{c_i\}$ and $SC_i=\{c_j\}$, where each element c_j is a character contained in the character string r_i.

The elements of SC are as follows:

(1) Subset SC_1:

$SC1=\{c_1, c_2, c_3, c_4, c_5\}$

(2) Subset SC_2 and SC_6:

SC_2, $SC_6=\{c_1, c_2, c_3\}$

(3) Subset SC_3, SC_7, SC_{10}, and SC_{12}:

SC_3, SC_7, SC_{10}, and $SC_{12}=\{c_2\}$

(4) Subset SC_4:

$SC_4=\{c_1, c_2, c_3, c_4, c_5, c_6\}$

(5) Subset SC_5:

$SC_5=\{c_2, c_7\}$

(6) Subset SC_8:

$SC_8=\{c_2,c_3,c_{12},c_{13}\}$

(7) Subset SC_9:

$SC_9=\{c_2,c_{12},c_{13}\}$

(8) Subset SC_{11}:

$SC_{11}=\{c_2,c_3,c_4\}$

(9) Subset SC_{13}, SC_{14}, SC_{15}, and SC_{16}:

SC_{13}, SC_{14}, SC_{15}, and $SC_{16}=\{\ \}$

(10) Subset SC_{17}:

$SC_{17}=\{c_2,c_8,c_9,c_{11},c_{13},c_{14},c_{15},c_{16}\}$

(11) Subset SC_{18}:

$SC_{18}=\{c_8,c_9,c_{10}\}$

(12) Subset SC_{19}:

$SC_{19}=\{c_2,c_8,c_9,c_{10}\}$

(13) Subset SC_{20}:

$SC_{20}=\{c_2,c_8,c_9,c_{10},c_{12}\}$

(14) Subset SC_{21}:

$SC_{21}=\{c_8,c_9,c_{12},c_{13},c_{16}\}$

3.3 Algorithm of Transliteration

The proposed transliteration is made up of decision, mapping, and inversion processes as shown in Fig.3.4. The following notations are used; the n-th character in $r^*_{i,j}$ is designated by $(r_{i,j})_n$. Also, $(r_{i,j})_t$ is defined as the character which satisfies the decision table for the input character $(r_{i,j})_n$, where t has a value for the next $(r_{i,j})_n$ decided in the decision table by executing the procedure d, described later.

```
d:procedure(t,k);
   f=[false];
   dcc=0;
   do while( dcc not = DCN_max or f=[false]);
     dcc=dcc+1;
     call dcc->p(w,x,y,z);
   end;

   p;procedure(w,x,y,z);
       if t=1 or (r   ) =w then
         if t=x then do,
            t=z; k=y; f=[true];
         end;
         else;
       else;
   end p;
 end d;
```

Fig.3.6 Algorithm of the decision process

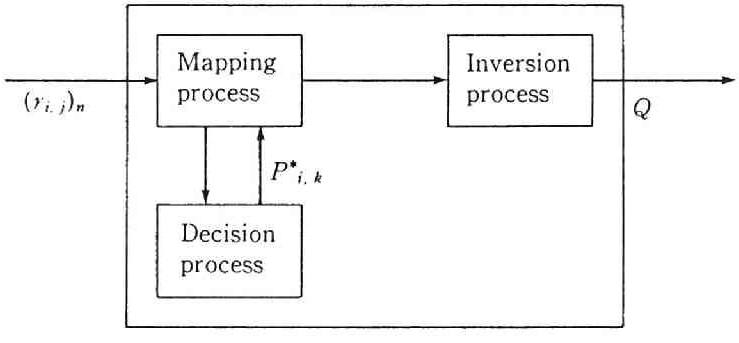


Fig.3.4 Transliteration process

When $(r_{i,j})_n$ keyed in is equal to $(r_{i,j})_t$, the mapping

$$(r_{i,j})_n \rightarrow P_{i,k}$$

is executed, where $P_{i,k}$ is the Roman spelling for one of the Thai letters in Figs.3.2 and 3.3, and k is an LCN.

The decision process is the procedure d for deciding $P_{i,k}$ where i is equal to the i of $r_{i,j}$ and k is set as the result of the decision making. For example, $P_{20,2}$ means a Thai letter "ใ". The decision table and the algorithm of the decision process are shown in Fig.3.5 and 3.6. The decisions are grouped in such a way that the decisions belonging to a given group require the same decision making, that is, the sequence of arguments shown in Fig.3.5 is common for decisions in that group. The decision group number, DGN, is attached to the group of decisions, and the decision column number, DCN, in the group is attached to identify each character in the character string discriminated by LCN like "E" in "LEU".

Decision making is processed as follows:

(1) The DGN is decided by $(r_{i,j})_n$ when t is equal to 1.

(2) For deciding the k of $P_{i,k}$, $(r_{i,j})_n$ is checked by invoking procedure p (see Fig.3.6) with the arguments in a group in ascending order of DCN (initial value is 1) up to DCN_{max}, where DCN_{max} designates the maximum value of DCN specified for DCN as shown in Fig.3.5.

Decision Group Number	Decisions	Decision Column Number	Arguments
1	D_1	1	(*, 1, 1, 2)
		2	(C_1, 2, 2, 3)
		3	(C_2, 3, 3, 1)
		4	(C_3, 3, 4, 1)
		5	(C_4, 3, 5, 1)
		6	(C_5, 3, 6, 1)
2	D_2, D_6	1	(*, 1, 1, 2)
		2	(C_1, 2, 2, 3)
		3	(C_2, 3, 3, 1)
		4	(C_3, 3, 4, 1)
3	D_3, D_7 D_{10}, D_{12}	1	(*, 1, 1, 2)
		2	(C_2, 2, 2, 1)
4	D_4	1	(*, 1, 1, 2)
		2	(C_1, 2, 2, 3)
		3	(C_2, 2, 8, 1)
		4	(C_2, 3, 3, 1)
		5	(C_3, 3, 4, 1)
		6	(C_4, 3, 5, 1)
		7	(C_5, 3, 6, 1)
		8	(C_6, 3, 7, 1)
5	D_5	1	(*, 1, 1, 2)
		2	(C_2, 2, 2, 1)
		3	(C_7, 2, 3, 1)
6	D_8	1	(*, 1, 1, 2)
		2	(C_2, 2, 2, 1)
		3	(C_3, 2, 3, 1)
		4	(C_{13}, 2, 0, 3)
		5	(C_{12}, 3, 4, 1)
7	D_9	1	(*, 1, 1, 2)
		2	(C_2, 2, 2, 1)
		3	(C_{13}, 2, 0, 3)
		4	(C_{12}, 3, 3, 1)
8	D_{11}	1	(*, 1, 1, 2)
		2	(C_2, 2, 2, 1)
		3	(C_3, 2, 3, 1)
		4	(C_4, 2, 4, 1)
9	D_{13}, D_{14}, D_{15}, D_{16}	1	(*, 1, 1, 1)

continued

Decision Group Number	Decisions	Decision Column Number	Arguments
10	D_{17}	1	(*, 1, 1, 2)
		2	(C_8, 2, 2, 1)
		3	(C_9, 2, 3, 1)
		4	(C_{11}, 2, 4, 3)
		5	(C_{13}, 2, 6, 3)
		6	(C_{14}, 2, 9, 1)
		7	(C_{15}, 2, 10, 1)
		8	(C_2, 3, 5, 1)
		9	(C_8, 3, 7, 1)
		10	(C_9, 3, 8, 1)
		11	(C_{16}, 2, 11, 1)
11	D_{18}	1	(*, 1, 1, 2)
		2	(C_9, 2, 2, 1)
		3	(C_{10}, 2, 3, 3)
		4	(C_8, 3, 4, 1)
12	D_{19}	1	(*, 1, 1, 2)
		2	(C_9, 2, 2, 1)
		3	(C_{10}, 2, 3, 3)
		4	(C_8, 3, 4, 1)
		5	(C_2, 3, 5, 1)
13	D_{20}	1	(*, 1, 1, 2)
		2	(C_8, 2, 2, 1)
		3	(C_9, 2, 3, 1)
		4	(C_{12}, 2, 4, 3)
		5	(C_2, 3, 0, 4)
		6	(C_9, 3, 5, 1)
		7	(C_{10}, 3, 7, 4)
		8	(C_9, 4, 6, 1)
		9	(C_8, 4, 8, 1)
14	D_{21}	1	(*, 1, 1, 2)
		2	(C_9, 2, 2, 1)
		3	(C_{12}, 2, 3, 3)
		4	(C_{13}, 2, 6, 4)
		5	(C_{16}, 2, 9, 1)
		6	(C_8, 3, 4, 1)
		7	(C_9, 3, 5, 1)
		8	(C_9, 4, 7, 1)
		9	(C_8, 4, 8, 1)

Fig.3.5 Decision table

In procedure d, t is equal to that of $(r_{i,j})_t$, and it is set to 1 at the beginning of the transliteration process (see Fig.3.7). The arguments w, x, y, and z are constants specified in the decision table: $w \in SC_i$ is a character; x is the value of t for w; y is the LCN of the Roman spelling for which f is assigned [true] in procedure p for arguments (w,x,y,z) invoked in procedure d; and z is the value of t for the next character $(r_{i,j})_n$. A positive value of k at exit from procedure d means that the Roman spelling $P_{i,k}$ shown in Fig.3.2 was found, where i and k are GN and LCN. The expression of $P_{i,k}$ in Thai is denoted by $P^*_{i,k}$. Note that $P^*_{i,k}$ is not always complete because the appropriate consonant must be inserted at the position marked '-' in $P^*_{i,k}$.

In the mapping process, when f is [false] at exit from procedure d, the last input character $(r_{i,j})_n$ is considered as the first character of another Roman spelling. Then t is reset to one and the last character $(r_{i,j})_n$ is passed to the decision process again. When $k > 0$ and f=[true] at exit from procedure d, $P^*_{i,k}$ is stored into a temporary storage area U, and the next character $(r_{i,j})_n$ is read from the keyboard. When $U \neq NULL$ and t is equal to 1 after reading the character, U is stacked in a storage area S. This process is shown in Fig.3.7.

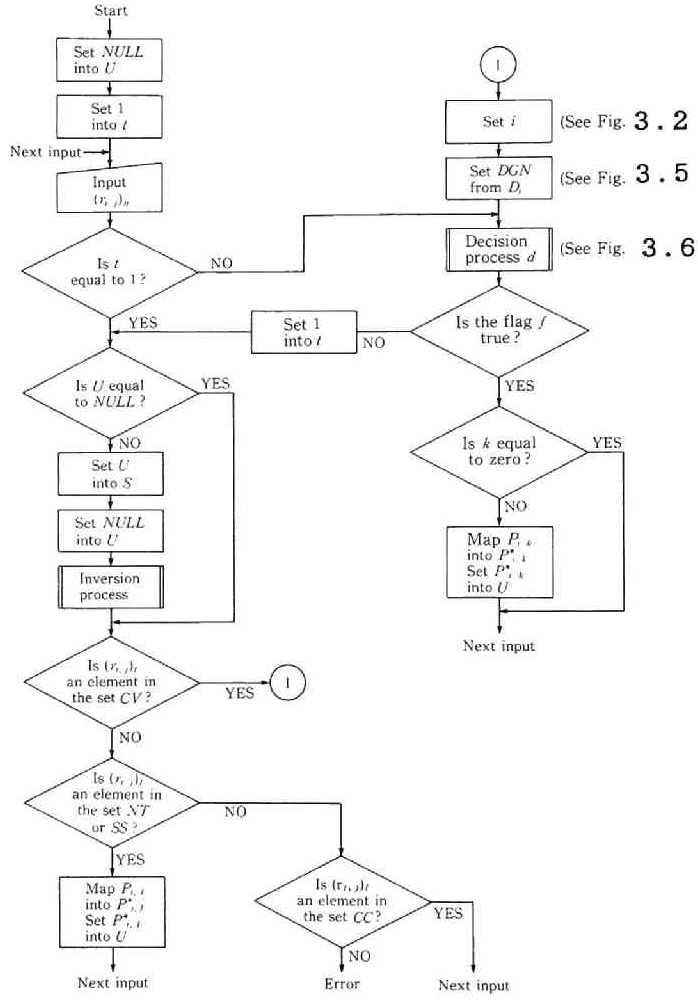


Fig.3.7 Flow chart of the transliteration process

3.4 Inversion Process

Let a Thai character string in complete form be denoted by Q. The conversion from $P^*_{i,k}$ to Q is expressed as follows:

When the "-" is included in $P^*_{i,k}$ the consonant is taken out of S, and inserted in the "-" position in Q.

When the consonant is double like กร, กล, ขร, คร, ขล, คล, ตร, ปร, ปล, พร, ผล, พล, ขว, คว, and กว the two previous consonants in S must be inserted.

In addition, a dead key operation, whereby a character pattern overlaps the preceding pattern without the carriage moving, is needed in order to display in 6 regions as shown in Fig.2.2 of chapter 2. The detail of scheme is presented in later.

3.5 Comparison

In the Transliteration Method, 49 keys are mapped on the keyboard. This is 53.3% of the number required in the Direct Mapping Method (92). The total number of key strokes required for all consonants and vowels in Fig.3.2 is 176 which is represented by Σr_i, where r_i is the number of characters used for the Roman spelling of Thai letters, and the suffix i ranges from 1 to 79. This is 12.5% less the number required by J. F. Hartmann's proposal.

Figure 3.8 shows the frequency of occurrence of each Thai letter

NO.	Letter	Freq.	NO.	Letter	Freq.	NO.	Letter	Freq.	NO.	Letter	Freq.
1	ก	41407	12	ฌ	28	23	ท	22768	34	ย	28653
2	ข	10018	13	ญ	4241	24	ธ	3169	35	ร	55392
3	ฃ	774	14	ฎ	1900	25	น	70137	36	ล	27657
4	ค	13916	15	ฏ	150	26	บ	16903	37	ว	29376
5	ฅ	104	16	ฐ	276	27	ป	15405	38	ศ	4233
6	ฆ	698	17	ฑ	52	28	ผ	8069	39	ษ	5661
7	ง	39532	18	ฒ	63	29	ฝ	1455	40	ส	19316
8	จ	17421	19	ณ	4485	30	พ	18866	41	ห	33185
9	ฉ	865	20	ด	27053	31	ฟ	1493	42	ฬ	125
10	ช	11772	21	ต	17658	32	ภ	2938	43	อ	37310
11	ซ	2652	22	ถ	7989	33	ม	38624	44	ฮ	5

NO.	Letter	Freq.	NO.	Letter	Freq.	NO.	Letter	Freq.	NO.	Letter	Freq.
45	ะ	18116	54	ๆ	37019	63	๐	4694	72	๙	594
46	ั	37497	55	แ	17549	64	๑	4094	73	็	62913
47	า	92754	56	โ	6303	65	๒	1827	74	่	41798
48	ิ	20112	57	ฤ	691	66	๓	1738	75	้	17
49	ี	21840	58	ฦ	18	67	๔	1456	76	๊	7
50	ึ	7810	59	ำ	8285	68	๕	1401	77	๋	1708
51	ื	10739	60	ใ	18049	69	๖	1105	78	ๆ	1186
52	ุ	10844	61	ไ	15403	70	๗	721	79	ฯ	369
53	ู	14867	62	ๅ	3006	71	๘	817			

Fig.3.8 The frequency of occurrence of Thai letters in the text of the KTSD

in the text of the Three Seals Law. The total number of letters is 1,111,141. For entering this text (about 1700 pages), the ratio of the numbers of key strokes, T.D., required by the TM and the DMM can be represented as follows:

$$\text{T.D.} = \frac{\Sigma f_i r_i}{\Sigma f_i} \qquad (3\text{-}1),$$

where f_i is the frequency of occurrence of the i-th Thai letter indicated in the NO. column in Fig.3.8, and r_i is the number of characters in its Roman spelling. Also, $\Sigma f_i r_i$ represents the total number of key strokes for the text. The number of key strokes is estimated to be 21.9% higher for our transliteration method than the Direct Mapping Method, whereas it is 32.0% higher for the method proposed by J. F. Hartmann, which employs the expressions "TH'", "TH''", "TH'''","TH''''", and "TH''''''" instead of our "TH1", "TH2", "TH3", "TH4", and "TH5".

By the Direct Mapping Method, the operator can input the text with accuracy from the shape of the Thai letters. In the Transliteration Method, however, because certain Thai phonemes are represented by several Thai letters, the operator must remember which Roman spelling corresponds to which letter: for example, he must recognize the difference in the shape of Thai letters between, say, "TH" and "TH1". This problem can be solved by including a function whereby all of the Roman spellings, for in-

stance "TH", "TH1", ... , "TH5" when "TH" is typed are simultaneously displayed on the CRT with the corresponding Thai letters, as happens with Roman-Kanji conversion in Japanese, thus enabling the operator to select the correct Thai letter.

Consequently, as for the burden for memorizing in the keyboard assignment for the case of transliteration scheme, it may be smaller than the case by the Direct Mapping Method, especially for non-native speakers for Thai, and non-skilled operator. The detail discussion is presented in the following chapter.

From the viewpoint of implementation, Transliteration Method does not require an intelligent terminal if the transliteration approach is used for displaying Thai letters and the main-frame computer performs the function of transliteration. Contrast with the transliteration scheme, the Direct Mapping Method requires an intelligent terminal to identify which Thai letter is input;in other words, the ordinary teletype terminal cannot be used for the input/output of Thai letters because it is impossible to display/print Thai letters on the terminal.

3.6 Using the Terminal

As one of the tasks, these two input methods, DMM and TM, in the terminal design were adapted. Its terminal can be used for the retrieval of the database KTSD under the information retrieval system (IRS), called FAIRS (FACOM Automatic Information Re-

trieval System), running on a main-frame computer, FACOM M-780. This system has been constructed as shown in Fig.3.9.

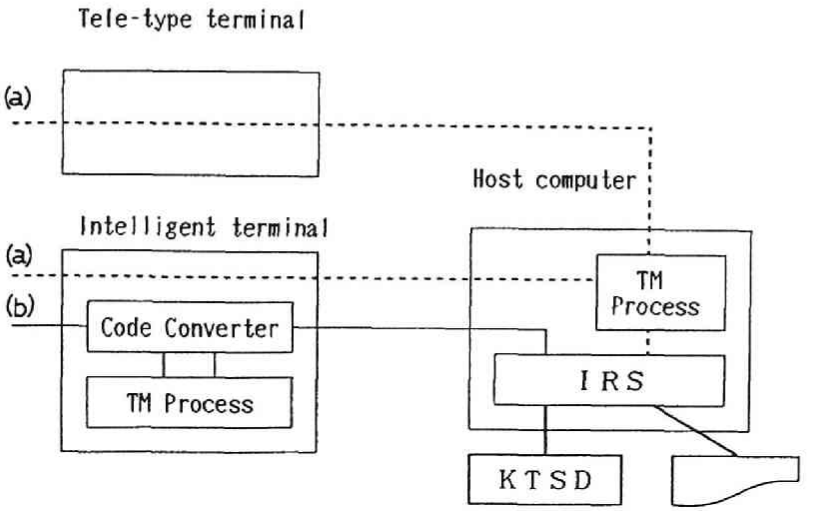


Fig.3.9 Structure of information retrieval system for the database of the KTSD

Two methods of operation are possible for typing Thai letters on the terminal. Firstly, the typing can be displayed in equivalent Roman letters on the screen as associated with the keyboard, namely, the transliteration process is put on the main-frame computer. This means that the operator must think in Thai but gets feedback in Roman letters ((a) in Fig.3.9). Secondly, to eliminate this burden of mental transliteration, the typing may be performed in the graphic or extra character font mode where each character typed is echoed as a Thai representa-

tion of the intended input letter ((b) in Fig.3.9).

The transliteration process located in the host computer in Fig.3.9 has been implemented as an attached program, namely, using a portion of exit routines of FAIRS which provides the function, for example, the substitution or replacement of query statements or the retrieval conditions for making possible the formulations of precise requests that reflect the user needs, depending on the characteristics of database built in the FAIRS.

The code converter and transliteration process located in the intelligent terminal in Fig.3.9 have been incorporated into the terminal program and run on the personal computer NEC PC-9800 series connected to the host computer. Its code converter employs the conversion between Thai representation on the monitor screen or on the keyboard and the Thai internal code. Also, the transliteration process is used for converting the Roman spelling input to Thai letters on the monitor screen according to the transliteration table if the TM is selected as the input mode in the terminal operation.

This transliteration scheme is applicable to the design of terminals for other Southeast Asian languages like Cambodian and Laotian.

The Data Processing Center, Kyoto University currently provides retrieval service for the Three Seal Law (KTSD) under the information retrieval system FAIRS using this terminal program.

Chapter 4 Thai Input Methods and its Characteristics

4.1 Introduction

For the processing Thai text, the input/output methods for Thai text and Thai letters are of major importance and present sophisticated problems.

Two input methods are considered. One is the Direct Mapping Method (DMM) by which each Thai letter corresponds uniquely to one of the keys on the keyboard[Sugita 80].

The other method is a transliteration approach using a table by which Thai letters are generated from Roman letters according to their pronunciation, like the Roman-Kanji conversion in Japanese, and is a method, here called Hartmann's Transliteration Method(HTM), proposed by J.F.Hartmann and G.M.Henry [Hartmann & Henry 83]. This approach requires a greater number of key strokes for input than the Direct Mapping Method, but it is readily operable by non-native speakers of Thai.

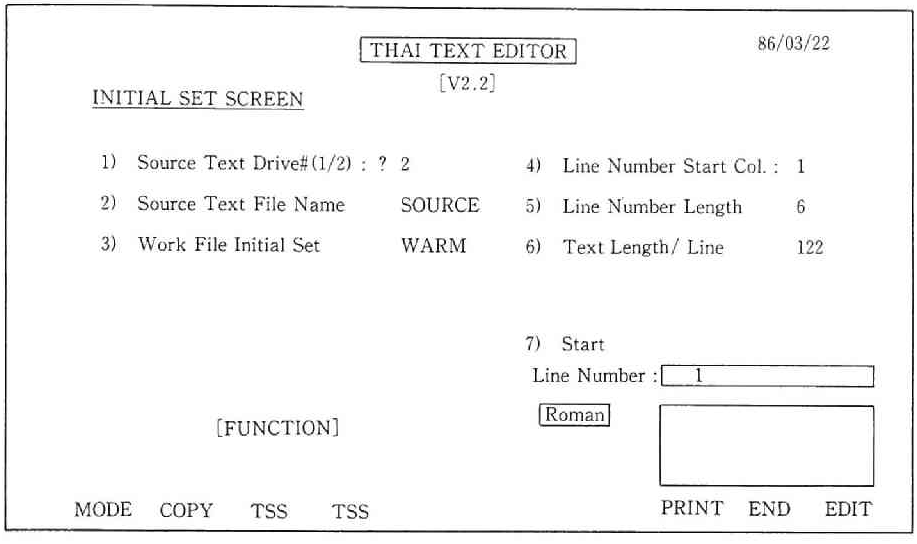
Thai text editor which employs the DMM as one of the input methods has been implemented first [Shibayama et al. 84]. New transliteration table which is modified from the HTM, here called the Transliteration Method (TM) has been proposed, which was incorporated into the text editor [Shibayama et al. 85]. And, a method Roman-Thai

conversion which is called the Simplified Transliteration Method (STM) has also been proposed [Shibayama et al. 87]

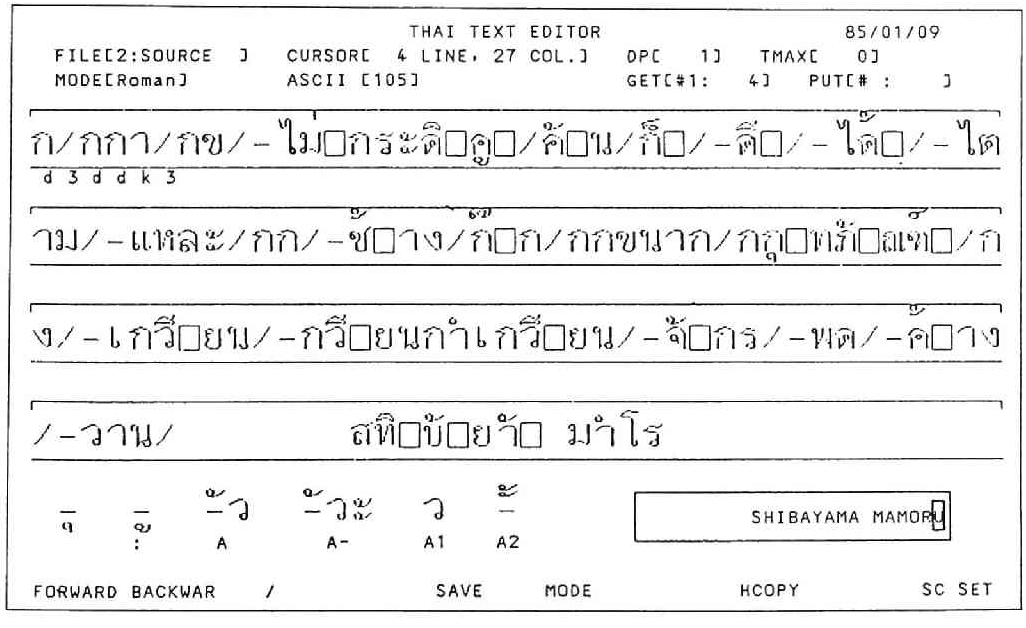
This chapter describes the characteristics of the DMM and the TM implemented on the Thai text editor, compares the DMM, TM, and HTM by the results of typing speed and learning curves measured for the practical work of inputting the main entries of a Thai-Thai dictionary by the DMM and TM methods. The comparison between on the burden of memorization and the learning effect of key typing based on the number of keys in the key assignment is performed. The basic idea of the STM and an estimation, which may be more readily operable for non-native speakers of Thai, is also presented.

4.2 Thai Text Editor and the Input Methods

Figure 4.1 shows opening and edit screens of the Thai text editor implemented on a personal computer. This editor employs the functions shown in Table 4.1. The first column "Screen" in Table 4.1 shows whether the screen is in (a) Opening screen or (b) Edit screen in Fig.4.1. The screen in Fig.4.1(b) is in the TM mode described in section 4.2 and 4.3, and corresponds to a record, namely, a line in the text which has a maximum of 160 characters in Thai, and which is divided into 4 lines in order to display Thai characters.



(a) Opening screen



(b) Edit screen

Fig.4.1 Screens of the Thai text editor

Table 4.1 Functions of Thai Text Editor

Screen	Key	Indication	Function
(a) (b)	f. 1 f. 6	MODE MODE	Input Method is Specified for Thai
(a)	f. 2	COPY	Back-up of Current File is Executed
(a)	f. 3, f. 4	TSS	TSS Emulator is Invoked
(a)	f. 8	PRINT	File Printing
(a)	f. 9	END	Quit the Editor
(b)	f. 1	FORWARD	Move to Next Record
(b)	f. 2	BACKWARD	Move to Previous Record
(b)	f. 3	"/"	"/" is Inserted
(b)	f. 5	SAVE	Editing File is Saved
(b)	f. 8	HCOPY	A Record is Printed
(b)	f. 10	SC SET	Screen (a) is Invoked
(a)	f. 10	EDIT	Editing is Restarted

Below each display line is the area of movement of cursor. The cursor can be moved to any directions using the arrow keys, and Thai characters are displayed using graphic instructions on the personal computer.

Direct Mapping Method (DMM)

In the Thai text editor implemented on the personal computer, we adopted the keyboard assignment, called the Direct Mapping Method (DMM), as shown in Fig.4.2.

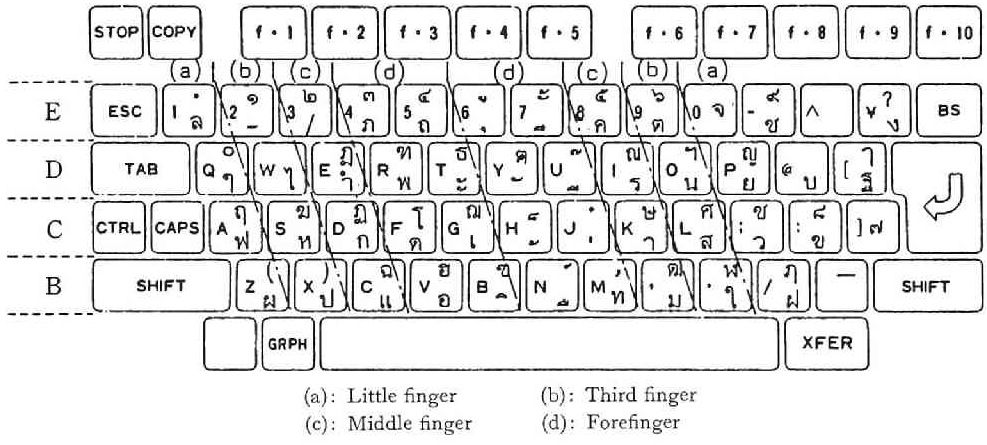


Fig.4.2 Keyboard Assignment of DMM

Since this keyboard assignment is almost equivalent to that of the IBM electronic Thai typewriter, and the editor employs the dead key control, whereby a character pattern overlaps the preceding patterns without the carriage moving, so that the consonants, vowels, and tones can be displayed in their appropriate positions on the CRT, the editor can be used like the IBM electronic Thai typewriter.

Transliteration Method (TM)

The input method of Thai letters by means of the Roman letters representing the Thai pronunciation, namely, the Transliteration method (TM), has the benefit of improving the operability of typing

for non-native speakers of Thai and for those who are accustomed to the normal keyboard assignment of Roman letters. At the same time, the transliteration table should be simple for typists and must be designed to decrease the number of key strokes and the range of finger movement. To this end, we have proposed the revised transliteration table from Roman to Thai letters shown in Fig. 3.2 and implemented a function capable of automatic and consecutive conversion according to this table.

The characteristics of the transliteration table are as follows:

(1)As shown in Section 3.2 of Chapter 3, the Roman spellings of the Thai consonants and vowels are classified into 21 groups according to their pronunciations, each group comprising character strings headed by same Roman character. The numerals, tones, special symbols, and control codes are classified into 28 groups.

(2) The transliteration approach by Hartmann and Henry distinguishes different Thai letters with the same pronunciation by use of apostrophes, for example, "TH", "TH'", "TH''", "TH'''", "TH''''", and "TH'''''". In our system, the distinction is represented by adding a number to the Roman spelling.

The Thai letters are arranged, moreover, in order of decreasing frequency of occurrence in the text of the Three Seals Law. In this way, the number of key strokes required by the operator is decreased. For example , "TH", "TH1", "TH2", "TH3", "TH4", and "TH5" are used for ท , ธ , ถ , ฐ , ฑ , and ฒ , instead of "TH", "TH'", "TH''",

"TH'''", "TH''''", and "TH'''''". An advantage of this scheme is that the ordinary teletype terminal can be used for input/output of Thai letters if the function of interconversion of Roman and Thai letters is implemented on the host computer.

Comparison of DMM, TM, and HTM

While the TM requires at most 49 keys to be used on the keyboard, the DMM, in which each Thai letter corresponds uniquely to one key, requires the use of 92 keys. Thus the TM allows the number of keys to be reduced by 46.7%. However, the number of key strokes required to input all Thai letters by the TM is 176, which is 91.3% higher than the number required by the DMM.

Compared with the DMM, the number of key strokes required to input a text with the same frequency of occurrence of Thai letters as the text of the Three Seals Law, it was estimated that the HTM, which is the transliteration method proposed by J.F.Hartmann and G.M.Henry, requires 32.0% more strokes and the TM 21.9% more strokes [Shibayama et al. 86].

4.3 Measurement of Learning Effect

Input of the main entries of the Thai-Thai dictionary published by the Thai Royal Institute [Photchana nukrom Thai 2525], a total of 31,202 words, was completed in about 6 weeks by 3 persons (about 9 man-weeks). The frequency of occurrence of each Thai letter in the

dictionary, the learning effect measured for the elapsed time of the input work, and its evaluation are as follows.

Frequency of occurrence of Thai letters

The main entries in the dictionary contained a total of 217,926 letters, which included all 72 character patterns. The percentages of consonants, vowels, tones, and others were 63.5%, 29.9%, 5.4%, and 1.2% respectively. Figure 4.3 shows the distribution of probability for the frequency of occurrence of each Thai grapheme, where the Y axis indicates the probability of occurrence of each Thai letter plotted on a log scale.

The figure shows that the probability of occurrence is greater than 0.005 for about one half of all letters. The learning effect for inputting main entries of dictionary is generally considered to be higher than for ordinary texts, because the dictionary entries are arranged in lexicographical order. therefore, for this input work, the same character sub-strings appearing in consecutive words were replaced by a special symbol in order to reduce the number of key strokes.

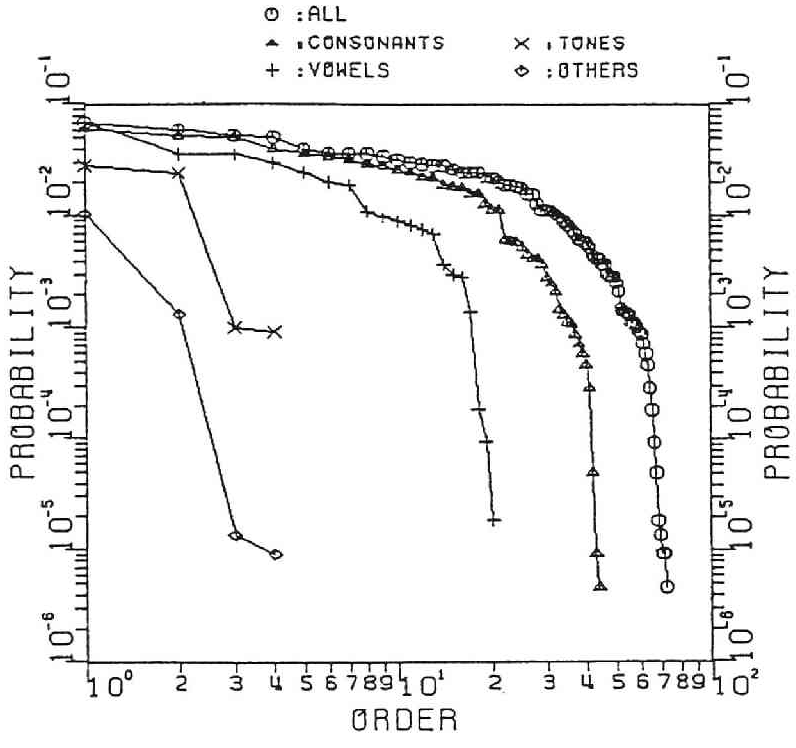


Fig.4.3 Frequency of occurrence of consonants, vowels, tones, and others

Table 4.2 shows the frequency of occurrence of Thai letters in the main entries of the dictionary. For inputting this text, the ratio of the number of key strokes, T.D., required by the TM and DMM can be represented as the equation (3-1) in Chapter 3.

It was found that the number of key strokes required by the TM was 42.9% higher than by the DMM.

Table 4.2 Frequency of occurrence of Thai Letters in the Thai Dictionary

(a) Frequency of Occurrence of Consonants

NO.	Letter	Freq.	NO.	Letter	Freq.	NO.	Letter	Freq.	NO.	Letter	Freq.
1	ก	11,015	12	ฌ	11	23	ท	3,956	34	ย	7,024
2	ข	2,477	13	ญ	821	24	ธ	1,263	35	ร	13,058
3	ฃ	1	14	ฎ	125	25	น	11,350	36	ล	6,389
4	ค	3,437	15	ฏ	232	26	บ	4,164	37	ว	6,158
5	ฅ	2	16	ฐ	311	27	ป	3,898	38	ศ	1,358
6	ฆ	190	17	ฑ	243	28	ผ	900	39	ษ	918
7	ง	7,487	18	ฒ	63	29	ฝ	278	40	ส	5,427
8	จ	2,841	19	ณ	1,295	30	พ	3,455	41	ห	4,748
9	ฉ	544	20	ด	4,841	31	ฟ	463	42	ฬ	99
10	ช	2,519	21	ต	5,640	32	ภ	1,168	43	อ	8,599
11	ซ	627	22	ถ	993	33	ม	7,946	44	ฮ	157

(b) Frequency of Occurrence of Vowels and Tones

NO.	Letter	Freq.	NO.	Letter	Freq.	NO.	Letter	Freq.
45	ะ	5,315	54	เ	7,854	63	◌๊	6,211
46	◌ั	7,893	55	แ	2,415	64	◌๋	5,308
47	า	14,874	56	โ	2,118	65	◌์	205
48	◌ิ	6,596	57	ฤ	301	66	◌ั	221
49	◌ี	4,421	58	ฦ	4	67	◌็	2,302
50	◌ึ	627	59	◌ำ	1,625	68	ๆ	292
51	◌ื	1,798	60	ใ	652	69	ฯ	2
52	◌ุ	4,092	61	ไ	1,495	70	ๅ	20
53	◌ู	1,988	62	◌ๅ	806			

Environment of measurement

The model of behavior in the input work by the typist in the making of the database for the Thai dictionary is shown in Fig.4.4

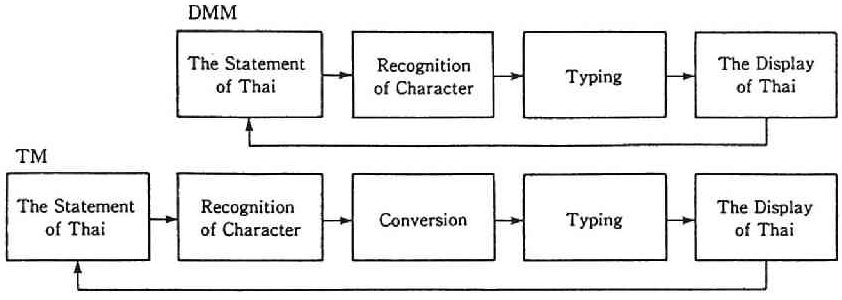


Fig.4.4 Model of Behavior for Typing of Thai

We have measured the learning effect for two persons in the actual input work by the DMM and TM in conjunction with the text editor implemented by both methods. On the editing screen for this input work, of which an example is shown in Fig.4.1(b), the slash(/) indicates the division between the words, and the hyphen(-) means that the previous character string with no hyphen is duplicated in this position. The Thai character string in the second row from the bottom in Fig 4.1(b) is a prompt for the next input in Roman spelling, displaying the group of Thai letters above and their corresponding Roman spellings below. Figure 4.1(b) shows the prompt when "U" was typed as the next input. The Thai character string in the center of the third row from the bottom represents the result of transliteration of the input of the Roman spelling inside the box in the second row from the bottom.

Table 4.3 shows the amount of text input by two operators, the

total number of words and characters input being 30,084, and 191,248 respectively. These two operators had no prior knowledge of Thai letters or Thai language, but were able to input about 200 letters per minute of Roman script.

Table 4.3 Enumeration of Main Entries in the Dictionary

Input Method	Operator(A)		Operator(B)	
	Number of Words	Number of Characters	Number of Words	Number of Characters
DMM	12,455	78,340	4,478	29,181
TM	8,490	54,527	4,661	29,200
Total	20,945	132,867	9,139	58,381

4.4 Evaluation

The burden for memorization, which is denoted by BM, in both DMM and TM methods are obtained by substituting the number of keys on the keyboard into the following expression;

$$BM \propto n \log_2 n$$

The result of burden for memorization is shown in Table 4.4. Burden for memorization in DMM is 139.1% higher than the method by TM.

Table 4.4 Comparison of Burden for Memorization

Method	Burden for Memorization	Ratio
English (Qwerty)	186	0.42
JIS	320	0.73
DMM	440	1.0
TM	184	0.42

It was assumed that the operators used their fingers and hands in accordance with the key assignment shown in Fig.4.2. The frequency of the use of fingers, hands, and each row of the keyboard by the typist is as shown in Fig.4.5 in the input work of the Thai dictionary text.

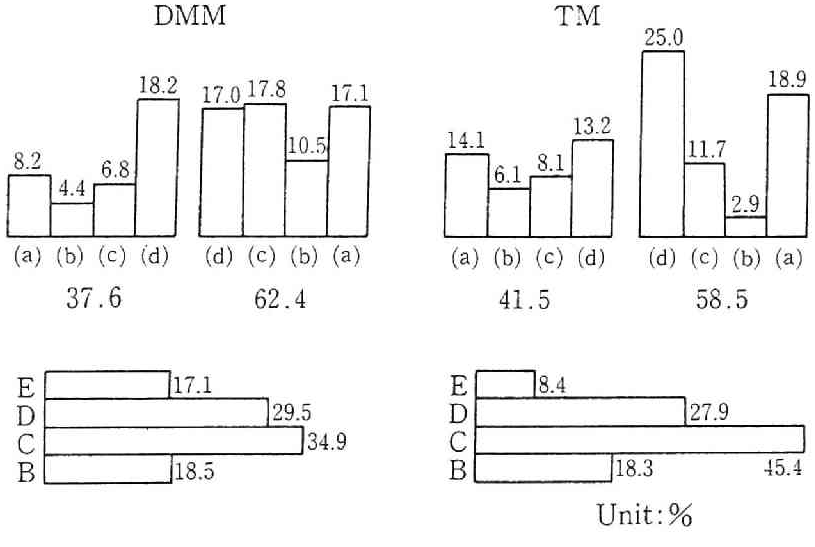


Fig.4.5 Utilization of the Fingers, Hands, and Rows of Keyboard

It is noticeable that the little fingers, which are considered least effective, are used frequently in both methods, and that the right hand works more than left hand by 24.8% and 17.0% respectively in the DMM and the TM. All of the rows without space bar from the top one in order are discriminated by E, D, C, and B respectively as shown in Fig.4.2. The home row C is used most frequently, which is considered to be effective, and the average distance of movement of fingers is as follows:

$$d_{DMM} = 0.185*1 + 0.349*0 + 0.295*1 + 0.171*2 = 0.822$$
$$d_{TM} = 0.630$$

where d_{DMM} and d_{TM} are the average distance of movement in the DMM and the TM estimated from the utilization in Fig.4.5. In comparison, the figures for the standard English keyboard d_E for Roman script input (Qwerty), RICOH d_2 (two-strokes method) for Japanese input, and JIS d_j (JIS keyboard) for Japanese input are 0.66, 0.60, and 0.91 respectively.

For the overall key input in the dictionary, the average time T for a key stroke, namely, for inputting a grapheme is defined as follows:

$$T = \sum_{i}^{n} \sum_{j}^{n} p_{ij} \cdot t_{ij},$$

where the p_{ij}, t_{ij}, and n are the frequency of occurrence of adjacent two-character i and j, the average stroke time(second/ stroke) for

the arbitrary adjacent two-characters, and the total number of characters respectively, as described in Section 2.3.2 of Chapter 2.

The calculated value of T on the basis of the frequency of occurrence of adjacent two-characters in the Thai dictionary approximately obtained that the TM and DMM are 0.621 and 0.491 respectively. As a result, it is found that T=0.621 in the TM is 26.47% higher than the DMM by 0.491, and the both T in the TM and DMM mean that the estimations of upper limit of the typing speed M correspond to M=96.6 and M=122.2 respectively by means of a calculation according to the following equation;

$$M = \frac{60}{T} \quad .$$

Unit:strokes/min.

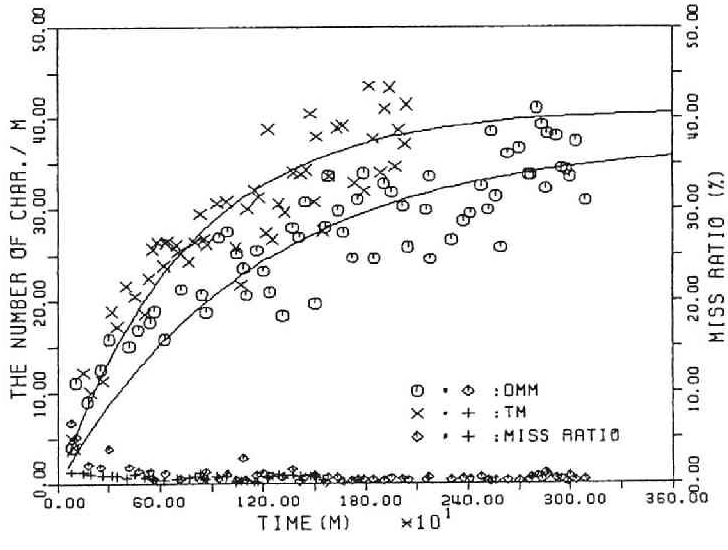


Fig.4.6 Typing Speed and the Learning Curve for Operator(A)

By comparing the result with the other methods of key assignment like Roman script T_E, Kana 50-tones T_{50}, JIS T_J, and JUNG-UM T_K for Korean alphabet, it is found that the average times for a key stroke in the both methods of Thai are to be higher than that by the other methods. The T_E, T_{50}, and T_J are 0.102, 0.310, and 0.150 respectively except T_K by 0.780 [Hee 86].

Figure 4.6 shows the result of the measurement of typing speed by both methods and the learning curves for operator(A). The X axis in Fig.4.6 indicates the elapsed time in the actual input work, and the Y axis indicates the number of characters input per minute. To represent the learning curves, the speed of typing S(t) is fitted to a function of the elapsed time t as follows:

$$S(t) = M (1 - e^{-Gt}),$$

where M is the upper limit of the typing speed, and G is the coefficient of training efficiency. Fitting of the learning curves to the measured values by use of this relation gives M=37.25, G=0.0527 for the DMM, and M=40.9, G=0.0797 for the TM.

Despite the time required to consult the transliteration table in the TM, and the 42.9% greater number of key strokes than the DMM, the typing speed is 9.8% higher by the TM than the DMM.

As a result, the both Ms for the DMM and TM in this measurement differ from the theoretical values, which were derived from the

calculated values of T on the basis of the frequency of occurrence of adjacent two characters in the Thai dictionary as mentioned above.

It is considered that such differences between the theoretical and experimental values are brought by the following reasons: (1) The typing speed and the learning characteristics vary according to the human behavior of individual person. (2) The result of experiment was based on the measurement by only one typist.

However, it is found that the TM is more readily operable by non-native speakers of Thai accustomed to inputting Roman script. It is also expected that the typing speed would increase if the elapsed time could be extended as previously described.

4.5 Simplified Transliteration Method

As shown in section 4.2 and from the experiment just described, to input any Thai letter by the TM, the typist has to memorize or consult the transliteration table to identify the Roman spelling, namely, the LCN in Fig.3.2.

It is difficult, however, to memorize the transliteration table in a short time, especially for non-native speakers of Thai, and the need to consult it reduces the speed of typing.

To eliminate the overhead time for memorizing and consulting the transliteration table in the TM, we have devised a simplified transliteration table composed of only GN's groups, without the distinc-

tion of LCN, namely, the Simplified Transliteration Method (STM, See Fig.4.7). For example, the Roman spelling "K" corresponds to "ก", "ค", "ข", "ฆ", "ฃ", and "ฅ".

After pressing "K", the typist then selects the appropriate Thai letter from the group by pressing the "XFER" key (See Fig.4.2) on the keyboard, which causes the Thai letters to appear one by one cyclically, and by pressing any key except the "XFER" key for the next input when the appropriate Thai letter appears.

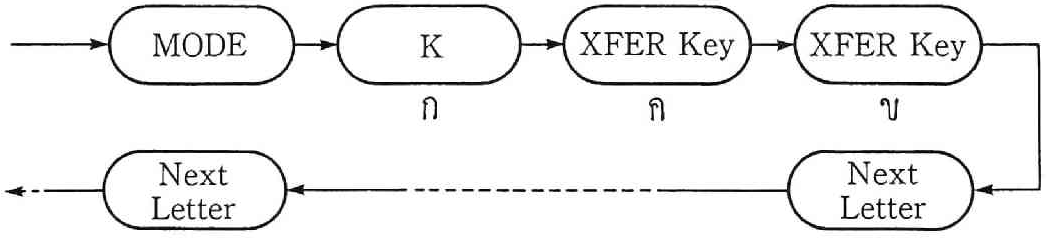


Fig.4.7 Simplified Transliteration Method

To estimate the number of key strokes of the STM using the frequency of occurrence of Thai letters as shown in Table 4.2, the ratio of the number of key strokes , S.D., required by the STM and

the DMM can be represented as follows:

$$S.D. = \frac{\Sigma f_i \cdot n_j}{\Sigma f_i}$$

where f_i is the frequency of occurrence of the i-th Thai letter indicated in the NO. column in Table 4.2, and the suffix i ranges from 1 to 70. The n_j is the number of key strokes for extracting the i-th Thai letter indicated in Table 4.2, namely, the value of LCN, in the j-th group belonging its i-th Thai letter and the suffix j corresponds to the value of GN shown in Fig.3.2. For example, the number of key strokes required for " ก " is 2 ("K" and "XFER" keys) which corresponds the value of LCN in GN=1. And $\Sigma f_i \cdot n_j$ represents the total number of key strokes for the text.

It is estimated that the number of key strokes required by the STM for inputting a text with same frequency of occurrence of Thai letters as the main entries of the dictionary is 61.6% (43.0% for the consonants and 93.9% for the vowels and tonal marks) higher than by the DMM. Compared with the TM, the STM requires 18.7% more strokes. However, the STM is more readily operable by non-native speakers of Thai than the TM, and the number of key strokes can be reduced if the sequence of appearance of Thai letters, especially the vowels, in a group is changed according to the text, like the learning function for the Roman-Kanji conversion in Japanese. This scheme can also be implemented on an intelligent terminal capable of displaying Thai letters, such as a personal computer.

Chapter 5 Thai Automatic Segmentation

5.1 Introduction

The Thai language, which is one of the target languages in a development of multi-country machine translation among the languages mainly in Asian countries that is supported by grants of the Ministry of International Trade and Industry of Japan, occupies an important position in the multi-lingual environment [Yada 88]. A database of the KTSD of Thailand that the keyword on the retrieval command line can be specified by using natural language expressions for retrieving the provisions or sentence required by a user also has been built and developed [Shibayama 90].

In the first phase of linguistic approaches for such natural languages, the segmentation which separates a sentence into words, which is concerned with the processing of recognizable portions of words, called the morphological analysis, is needed, particularly in the case of unsegmental language like Japanese.

Segmentation in the case of an unsegmental language generally uses the longest-match method on the basis of the fact that the longest-match is the simplest and most popular method. Furthermore, the capacity and access method for dictionary are one of the key factors and subjects for the system design, so the reduction in the

capacity of the dictionary making it as small as possible may be able to provide an effective response time when searching main entries. Also, the data structure and algorithm are very important factors.

In this chapter, the longest-match method for an analysis of morphological level in Thai text processing is adopted, and this represents the outcome of how well it is segmented first.

Secondly, a morphological analysis using the revised longest-match method, with a back-tracking function which allows the rejected sentences to be re-segmented based on the sequence of the appearance of phonemes, is incorporated in the program. Such longest-match method is called a Syllable Longest-Match method (SLM), and an algorithm on the basis of the experiment is proposed.

Thirdly, a finite automaton model is proposed for recognizing Thai syllables, using the morphological knowledge which is based on the syllable formation rules of symbol corresponding to the phoneme embedded in a syllable.

Fourthly, according to the finite automaton model, a Thai syllable recognizer is implemented. And the linguistic procedures and the result of the recognizer obtained from the experiment are presented.

Finally, a detailed analysis of the morphological level and an evaluation based on the outcome derived from unsuccessful results of segmentation are presented, and the consideration as to what is

attempted to reduce the error ratio by incorporating heuristic knowledge into the syllable formation rules based on the analysis is described.

5.2 Syllable Formation Rules

Characteristics of formalizing the syllable, which is defined as a word because the Thai language is a phonogram, are listed as follows:

(a) Thai words are phonetic; they are basically monosyllables with 5 tones and are composed of "Consonant + Vowel" or "Consonant + Vowel + Consonant". Then the monosyllable is called as the syllable in this chapter.
(b) Thai word also is formed by the combination of several monosyllables in many cases. Furthermore, the word is formed by the combination of several words, called the compound word.
(c) Thai letters consist of 44 consonants and 32 vowels. However some vowels as shown in Fig 2.3(b) are represented with a combination of several symbols.

The smallest recognizable unit on the Thai syllable recognizer or the least unit of character in machine readable form is defined as a symbol, and also the Thai alphabet expression corresponding to a symbol is defined as a letter; for example, each character in Fig 2.3(a), is a letter.

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
ก	ค	ข	ฆ	ฃ	ฅ	จ	ช	ฌ	ฉ
C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
ด	ฎ	ต	ท	ถ	ธ	ฐ	ฑ	ฒ	ฏ
C21	C22	C23	C24	C25	C26	C27	C28	C29	C30
น	ณ	ง	บ	พ	ฝ	ภ	ฝ	ฟ	ล
C31	C32	C33	C34	C35	C36	C37	C38	C39	C40
ฬ	ร	ย	ญ	ส	ษ	ศ	ช	ห	ฮ
C41	C42	C43	C44						
บ	ม	ว	อ						

(a) consonants

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
null	◌ั	า	◌ิ	◌ี	◌ึ	◌ื	◌ุ	◌ู	เ
V11	V12	V13	V14	V15	V16	V17	V18	V19	V20
แ	ว	◌็	◌์	อ	ไ	ใ	โ	ฤ	ฦ

(b) vowels

t1	t2	t3	t4
◌่	◌้	◌๊	◌๋

(c) tones

S1	S2	S3	S4
ๆ	ฯ	◌ๅ	sp

(d) special symbols

n1	n2	n3	n4	n5	n6	n7	n8	n9	n10
๑	๒	๓	๔	๕	๖	๗	๘	๙	๐

(e) numeral symbols

Fig.5.1 Notation of Thai graphemes

From the characteristics mentioned above (a), the consonants and vowels are denoted as follows:

[C + V] (5-1)

[C + V + C] (5-2)

where C and V correspond to a consonant and vowel respectively, "[ ]" means a word unit, and "+" means the concatenating operator among the symbols.

Hence, it is found that the sentence formation, namely, the individual words which are embedded in a sentence, is specialized from the expressions (5-1) and (5-2) as the following sentence patterns;

[C + V] + [C + V] + ... (5-3)

[C + V] + [C + V + C] + ... (5-4)

[C + V + C] + [C + V] + ... (5-5)

[C + V + C] + [C + V + C] + ... (5-6)

Next, the Thai symbols based on Thai letters are defined.

Thai symbols are classified as follows:

(1) Subset CS : Consonant symbols

$CS = \{ c_1, c_2, c_3, \ldots, c_{44} \}$ (5-7)

Each consonant letter in Fig 5.1(a) is discriminated by c_i, which is each element in set CS, and i varies from 1 to 44.

(2) Subset VS : Vowel symbols

$VS = \{ v_1, v_2, \ldots, v_{20} \}$ (5-8)

Each symbol corresponds to each vowel letter in Fig.5.1(b)

respectively.

(3) Subset TS : Tone symbols

$$TS = \{ t_1, t_2, t_3, t_4 \} \quad (5\text{-}9)$$

Each symbol corresponds to each tonal mark letter in Fig 5.1(c) respectively.

(4) Subset SS : Special Symbols

$$SS = \{ s_1, s_2, s_3, s_4 \} \quad (5\text{-}10)$$

(5) Subset NS : Numeral symbols

$$NS = \{ n_1, n_2, n_3, \ldots, n_{10} \} \quad (5\text{-}11)$$

where such vowels are equivalent to the consonant as follows;

(a) $v_9 = c_{33}$

(b) $v_{12} = c_{43}$

(c) $v_{15} = c_{44}$

Consequently, the set of symbols which forms the Thai language is defined as;

$$T = \{ CS, VS, TS, SS, NS \},$$

namely, T means the Thai alphabet, and all the strings formed by an arbitrary finite sequence of symbols from T, including the null string, are designated by T^*, and T^* also is finite.

By analyzing the order of appearance of grapheme, in other words, symbols according to their phonemic rules for Thai, the number of consonant symbols at the end of a syllable in the form [C + V + C

] are 41. And the vowels which coincide with phonetic rule in sequence as shown in Fig.5.2(a) introduced from Fig.2.3(b) in Section 2.2 of Chapter 2 are defined. Contrary to its sequence, the vowels that the sequence of appearance of grapheme embedded in a syllable differs from its phonetic rule also are defined as shown in Fig.5.2(b). In both Fig.5.2(a) and (b), a consonant must be inserted in "-" position.

Case	1st	2nd	3rd	4th
1	-	v_2	-	
2	-	v_2	v_{12}	
3	-	v_2	v_{12}	v_6
4	-	v_3		
5	-	v_4		
6	-	v_4	v_{15}	
7	-	v_5		
8	-	v_6		
9	-	v_7		
10	-	v_8		
11	-	v_{10}		
12	-	v_{11}		
13	-	v_{12}	-	
14	-	v_{13}		
15	-	v_{14}		
16	-	v_{14}	v_{15}	
17	-	v_{15}		

(a) Vowel symbols coinciding with phonemic rule

Case	1st	2nd	3rd	4th	5th
1	v_{16}	-			
2	v_{17}	-			
3	v_{18}	-			
4	v_{18}	-	v_6		
5	v_{19}	-			
6	v_{19}	-	v_4		
7	v_{19}	-	v_6		
8	v_{20}	-			
9	v_{20}	-	v_3		
10	v_{20}	-	v_3	v_6	
11	v_{20}	-	v_4		
12	v_{20}	-	v_6		
13	v_{20}	-	v_7		
14	v_{20}	-	v_8	v_9	
15	v_{20}	-	v_8	v_9	v_6
16	v_{20}	-	v_{14}	v_{15}	
17	v_{20}	-	v_{14}	v_{15}	v_6
18	v_{20}	-	v_{15}		
19	v_{20}	-	v_{15}	v_6	

(b) Vowel symbols uncoinciding with phonemic rule

Fig.5.2 Sequence of appearance of graphemes in the vowels

Consonant and vowel symbols are represented as follows:

(6) Subset CS_e : Consonant symbols at end of the syllable

$$CS_e = CS - \{ c_{39}, c_{40}, c_{44} \} \quad (5\text{-}12)$$

where an arbitrary element in set CS_e is c_e $\{c_e : c_e \in CS_e\}$.

(7) Subset VS_n : Vowel symbols coinciding with phonemic rule

$$VS_n = VS - \{ v_1, v_9, v_{16}, v_{17}, v_{18}, v_{19}, v_{20} \} \quad (5\text{-}13)$$

where an arbitrary element in set VS_n is v_n $\{ v_n : v_n \in VS_n \}$.

The syllable formation rules based on the Thai grammatical rules are introduced below.

The expressions (5-3), (5-4), (5-5), and (5-6) indicated previ-

ously deduce to one rule.

[Rule 1] No boundary of a word unit is at a point immediately before a vowel. In other word, the segmentation should not be made between symbols of C and V in the [C + V] or [C + V + C] sequence.

Look at the table of vowels as shown in Fig.2.3(b) in Chapter 2 again. In Thai writing, the sequence of appearance of symbols is reversed as in No.5-11, 17-18, 23-25, and 27-29 in Fig.2.3(b) contrary to the sequence of phonemes. These symbols concerned with such vowels are 5; ไ , ใ , โ , แ , and เ , and they are denoted by v_{16}, v_{17}, v_{18}, v_{19}, and v_{20} respectively.

This result leads to a following rule.

[Rule 2] Symbols v_{16}, v_{17}, v_{18}, v_{19}, and v_{20} are the first symbols embedded in a word; for example, ไป (pai) is denoted by v_{16} and c_{24}.

From the table of vowels, the following symbols consisting of a vowel are classified into 3 groups. The first group is when the letter C is followed by the letter V; which consists of the symbols v_3, v_5, v_6, v_9, v_{12}, and v_{15}. The second is when the letter of a vowel is placed on appropriate position above the consonant; which consists of the symbols v_2, v_4, v_7, v_8, v_{13}, and v_{14}. The third is when the letter of a vowel is placed on appropriate position below the consonant; which consists of the symbols v_{10} and v_{11} respectively.

Assume that the symbol C is followed by the symbol V for all of the above groups in the sequence of appearance; for example, ก็ is a string c_1v_4 in machine readable form, where "ก" is denoted by c_1.

[Rule 3] No boundary of word unit is at a point immediately before all symbols of V belonging to the above 3 groups; for example, a word จะ (ca) is denoted by c_7 and v_6.

When either v_9, v_{12}, or v_{15} is not a vowel, an attribute of that consonant or vowel corresponds to a symbol is decided on by the position where it is embedded in a word based on [Rule 1]; for example, a word อออก (oo:k) is a string $c_{44}v_{15}c_1$, where the first อ is denoted by c_{44}.

[Rule 4] Discrimination between a consonant and a vowel is determined based on the position of its symbol embedded in a word; for example, the first symbol of วา (wa:) must be recognized as the consonant c_{43} instead of the vowel v_{12}.

If a symbol equals one of the tonal marks, then such a symbol is a following one after a consonant or vowel. Hence, by only one symbol of any tonal mark, a word cannot be formed.

[Rule 5] No boundary of a word unit is at a point immediately before any tonal mark.

In the expression (5-1) or (5-2), it is known that consonant C consists of two letters, which are represented by the two symbols for the following combination:

กร.กล.ขร.คร.ขล. คล.ตร.ปร.ปล.พร.ผล.พล.ขว.คว, and กว ,

where that symbol combination is called a double consonant, and the combinations above are denoted by c_1c_{32}, c_1c_{30}, c_3c_{32}, c_2c_{32}, c_3c_{30}, c_2c_{30}, $c_{13}c_{32}$, $c_{24}c_{32}$, $c_{24}c_{30}$, $c_{25}c_{32}$, $c_{26}c_{30}$, $c_{25}c_{30}$, c_3c_{43}, c_2c_{43}, and c_1c_{43} respectively.

Consequently, if a set of the double consonant above is denoted by C_d, then

(8) $$C_d = \{ c_1c_{32}, c_1c_{30}, \ldots , c_1c_{43} \}. \qquad (5\text{-}14)$$

[Rule 6] If two adjacent symbols are a double consonant, those two symbols are recognized and formed as a consonant. Hence, C in expression (5-1) or (5-2) is equivalent to two symbols of C; for example, พระ(pra) is denoted by the double consonant $c_{25}c_{32}$ and a vowel v_6.

Consider another two adjacent symbols, here denoted by c_ic_j. If $c_ic_j \notin C_d$, then c_ic_j should be considered as a word in which a vowel is omitted in the expression (5-2) when the symbols are a recognizable word in the lexical unit.

[Rule 7] If the two adjacent symbols are not a double consonant and form a recognizable word as a morpheme, the two symbols are a word without the vowel symbol in the form [C + V + C]; for example, a word คน (khon) is a string c_2c_{21}, where no vowel is used.

In the Thai writing system, other miscellaneous marks including numeral must be explained.

Important miscellaneous marks for forming a word are as follows:

(1) To abbreviate a long name or title, a letter ' ฯ ' is used. However, it is ignored in our Thai syllable recognizer.
(2) To indicate that the preceding word or group of words should be repeated, a letter ' ๆ ' is used, However, it also may be ignored in this recognizer.
(3) To show one, sometimes two or more, silent letters, a letter ' ์ ' is written in a position above a preceding letter. The letter is denoted by the symbol s_3.
(4) Numerals, punctuation, and others are described in the following section.

The explanation (3) above leads to the next rule.

[Rule 8] No boundary of a word unit is in a point immediately before the symbol s_3.

In the following sections, all of the analyses in the morphological level of Thai are carried out and evaluated on the basis of the rules defined above.

5.3 Longest-Match Method based on Phonemic Rules

5.3.1 Experimental Environment

In the first phase of the experiment of segmentation for Thai sentences, the right-directed longest-match method which plays the

major role for segmenting the unsegmental language has been used. The number of Thai input sentences for an experiment is 20,631, which is the full text of the KTSD input from a copy of Khrusapha [Khrusapha 62]. The dictionary used for the experiment is made up arbitrarily from the previously segmented statements.

The statistics of the input text is shown in Table 5.1.

Table 5.1 Statistics of input text of the KTSD

Number of sentences	20,631
Average words per sentence	11.8
Average symbols per word	4.7
Number of main entries in the dictionary	20,475

For the frequency of occurrence of each Thai letter, refer to Fig.3.8 in Chapter 3. A part of the dictionary is shown in Fig 5.3.

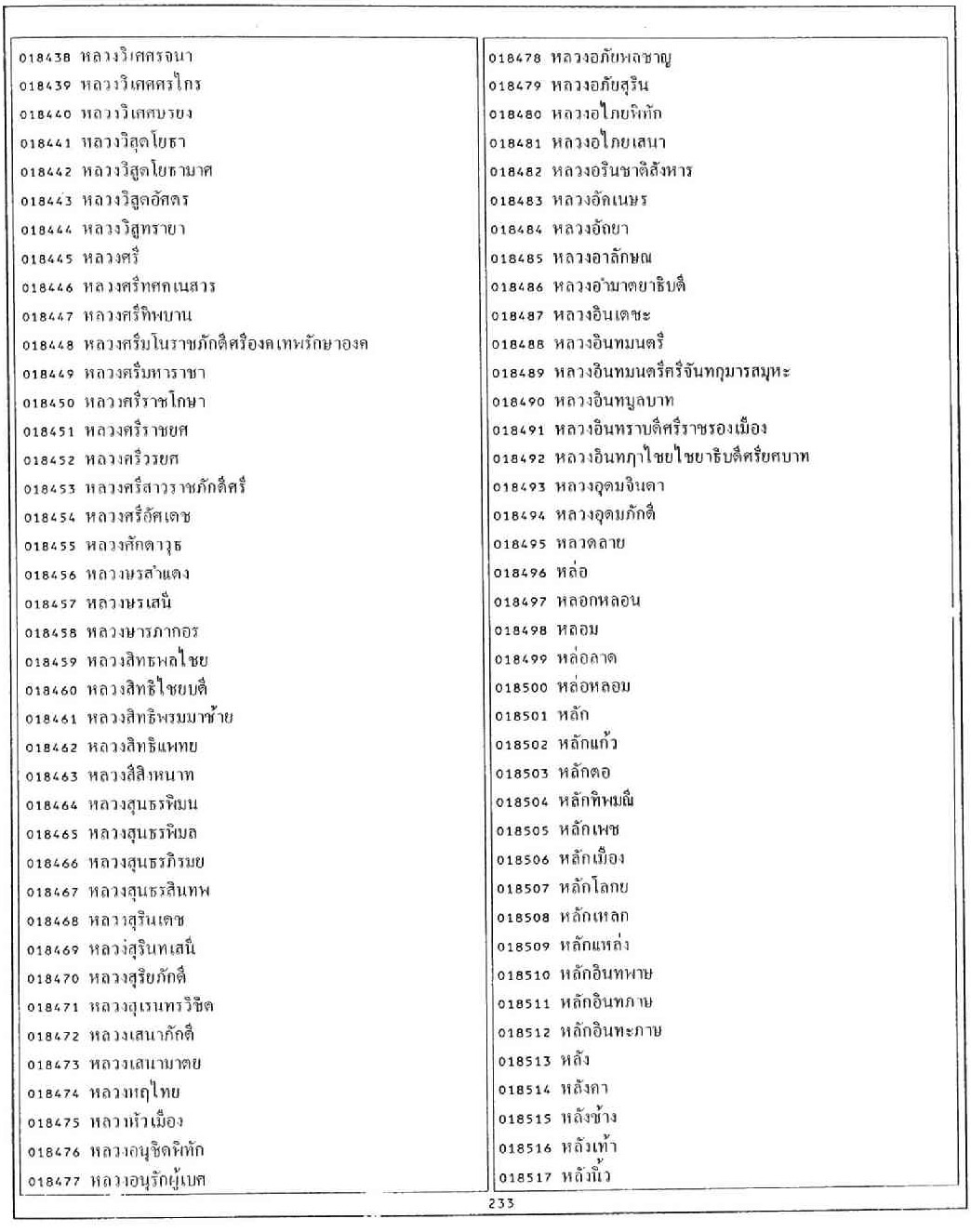


Fig.5.3 A part of dictionary of the KTSD

5.3.2 Experimental Method

Two experiments for segmenting Thai sentences of the KTSD are carried out as follows; (1) Using the ordinary longest-match method with reference to a dictionary first, that is not based on the syllable formation rules defined in Section 5.2, (2) Using the longest-match method which employs the back-tracking function based on the syllable formation rules with necessarily reference to a dictionary.

Each sentence is segmented from the extreme left side to the end of the right side using the right-directed longest-match method in turn. The input sentences of the KTSD have already been segmented correctly by the handiwork of an expert, and the delimiter "/" has been inserted between the words. But this delimiter is removed before segmenting the sentence in advance. Then it is used again for determining whether the segmentation has succeeded or not in every sentence.

To segment the sentence, the number of references to a dictionary is only one time whereby the segmentation is accomplished by selecting a longest entry out of the main entries corresponding to the string of the subject sentence, for example, as follows:

Object string	$x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ \cdots\ x_n$
Group of corresponding main entries	$x_1\ x_2$ $x_1\ x_2\ x_3\ x_4$ x_1

The next result shows that the longest entry $x_1\ x_2\ x_3\ x_4$ was selected by longest-match method.

Segmented result	$x_1\ x_2\ x_3\ x_4\ /\ x_5\ x_6\ \cdots\ x_n$
Next object string	$x_5\ x_6\ \cdots\ x_n$

In contrast to the above longest-match method, which is called the L-method, there is another right-directed longest-match method in which a symbol of the extreme right side is removed, and then the search is toward the left side in order until the object is matched with it in entries of the dictionary [Tanaka 89]. This is called the R-method.

In this method, to segment a word out of a string consisting of the symbols, the frequency of reference of the dictionary is equal to maximum n if the length of string is n, which is equal to the maximum number of symbols embedded in the string. In the previous L-method, it is one. Furthermore, the frequency of reference to the dictionary to accomplish segmentation for all of the words out of a string, which is denoted by R, is represented as follows:

$$R = \frac{n\ (\ n + 1\ )}{2},$$

where n is the number of symbols, and the n is greater than zero.

As a result of a comparison for the two longest-match methods,

it is found that the frequency of reference to the dictionary and its CPU time in the L-method are 8579 and 37.3 sec., respectively, for segmenting the first 1000 sentences of the KTSD. Compared to that, in the R-method, the frequency of reference to the dictionary is 26.75 times, and the CPU time is increased by 90.8 times.

The selection of the data structure and memorizing method for storing dictionary data, taking into account the capacity increase, are most important factors for reducing search time. The data structure of dictionary, especially, influences the efficiency for reducing the required time of transfer. Then, the strategy for the number of transfers between the main memory and auxiliary storage should be designed carefully when the dictionary database is placed on an auxiliary storage device. As the searching method for the dictionary, the hashing and the B-tree are well known [Nagao 78].

Of the dictionary through this experiment, a data structure only on main memory as shown in Fig.5.4 has been adopted. Figure 5.4 shows that an index of consonants in the dictionary consists of only 44 entries with each pointer pointing to a group of words corresponding to each consonant. For each region in the group, a word which the first consonant belongs to its group has been stored. The number of regions in each group is variable in size, and the length of each word also is variable. When the word of a reversed type based on Rule 2 is referred to the dictionary, the search must be performed using the first consonant of the second symbol.

5.3.3 Evaluation for Longest-Match Experiment

From the experiment of segmentation using the longest-match method, the results as shown in Table 5.2 were derived.

Table 5.2 Result of segmentation for KTSD

Input sentences	20,631
Frequency of reference to dictionary	243,918
Non segmented sentences	998
Ratio of segmentation	95.2%

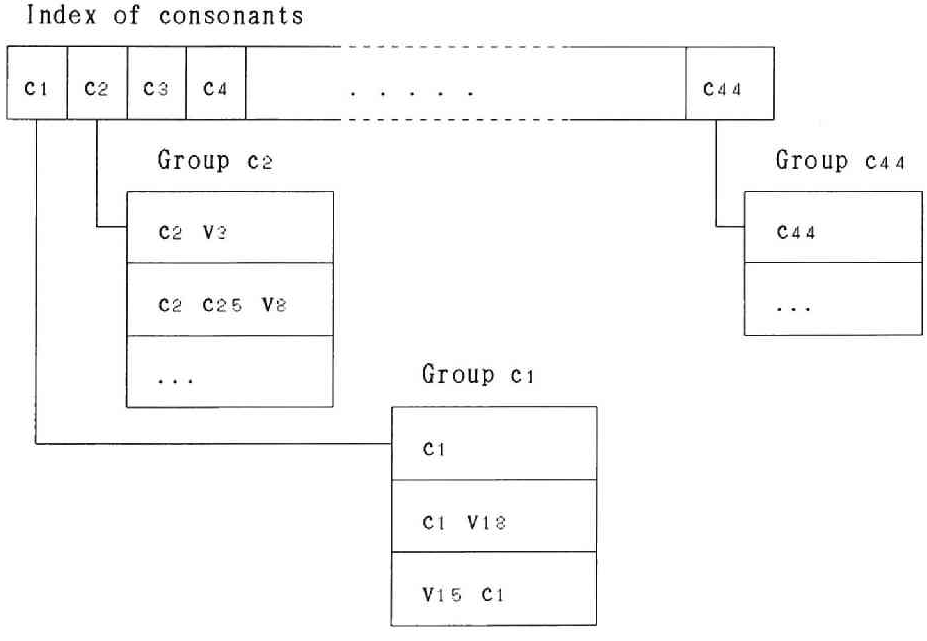


Fig.5.4 Data structure of dictionary

A detailed example of segmentation for unsuccessful cases is shown in Fig.5.5. As a result of segmentation, it is found that the ratio of automatic segmentation by using the ordinary right-directed longest-match method is 95.2% in terms of sentences, and its ratio is about 15% higher than the result of Japanese sentences by approximately 80% in general.

The characteristics of unsuccessful cases are mainly classified into two categories; one is segmented by the word form [C + V + C] for the sequences [C + V] plus [C + V] or [C + V] plus [C + V + C]; for example, (a), (b), and (c) in Fig.5.5. The second is when a syllable completely corresponds to the most left-side syllable embedded in a word in which the word is formed by several syllables or the compound word like (d), (e), (f), (g), and (h) in Fig.5.5.

(a) / v_{19} c_{30} / c_1 c_{32} v_6 ...
/ v_{19} c_{30} c_1 / c_{32} v_6 ...

(b) / v_{19} c_{32} / c_1 v_7 c_{41} c_{32} ...
/ v_{19} c_{32} c_1 / v_7 c_{41} c_{32}

/ c_{14} v_8 t_1 / c_{41} v_8 /
/ c_{14} v_8 t_1 c_{41} / v_8

(c) / v_{20} c_{35} v_8 v_9 / c_{11} t_2 v_{12} c_1 v_2 c_{21} /
/ v_{20} c_{35} v_8 v_9 c_{11} / t_2

(d) / c_{43} t_1 v_3 / c_{41} v_3 c_1 /
/ c_{43} t_1 v_3 c_{41} v_3 / c_1

(e) / v_{16} c_{41} t_1 / c_{35} c_{41} v_2 c_1 /
/ v_{16} c_{41} t_1 c_{35} c_{41} / v_2

(f) / c_{14} v_{10} c_1 / c_{14} v_3 t_1 /
/ c_{14} v_{10} c_1 c_{14} v_3 / t_1

/ c_{39} c_{21} v_{13} t_1 c_{23} / v_{19} c_{30} t_2 c_{43} /
/ c_{39} c_{21} v_{13} t_1 c_{23} v_{19} c_{30} / t_2

(g) / c_{15} t_2 v_3 / c_{39} v_3 c_1 v_2 c_{21} /
/ c_{15} t_2 v_3 c_{39} v_3 c_1 / v_2

(h) / c_{21} v_{15} c_1 / c_{32} v_2 t_2 v_{12} c_{14} v_{13} c_{41} /
/ c_{21} v_{15} c_1 c_{32} v_2 t_2 v_{12} / c_{14} v_{13} c_{41}

Upper line : Acceptable state
Lower line : Unsuccessful

Fig.5.5 An example of unsuccessful cases
by the ordinary longest-match method

5.3.4 Syllable Longest-Match Method

Let d=m when the result is segmented correctly except for m symbols; for example, d=1 in the case(a), (b), and (c) in Fig.5.5.

Table 5.3 summarizes the behavior of unsuccessful cases in the experiment using the ordinary longest-match method.

Table 5.3 Summary of unsuccessful cases

d : Difference in the number of symbols

	d=1	d=2	d=3	d=4
(A)Consonant	227	42	20	22
(B)Vowel	451	11	9	13
(C)Tone or others	135	31	0	2
(A)+(B)+(C)Total	813	84	29	37
(B)+(C)	586	42	9	15

d>4: 35, Total 998

By analyzing Table 5.3, it is found that the ratio of correct segmentation can be increased by the method based on the syllable formation rules in section 5.2.

Consider the longest-match method incorporated with the function of back-tracking when the segmentation cannot be carried out any more.

A Syllable Longest-Match method (SLM), which employs the back-

tracking function for bringing the analysis back to the preceding symbol based on syllable formation rules, especially, Rules 1, 3, 5, and 8 with respect to the rules of word boundary, has been proposed. This SLM also refers to a dictionary.

It is found that a ratio of segmentation by the Syllable Longest-Match method is 98.0%, which is 2.8% higher in ratio of the number of sentences than that by the previous method. Therefore, the Syllable Longest-Match method is found to be an effective scheme for Thai word segmentation.

5.4 Model of Thai Syllable Recognizer

To implement a machine for recognizing Thai syllables automatically, a syntax-directed program for Thai is devised, here called Thai syllable recognizer, which employs the function of automatic and consecutive segmentation for Thai sentences based on Thai syllable formation rules as shown in section 5.2 and without reference to a dictionary.

In the first place, the two types which form formulas (5-1) and (5-2) are classified into 6 models depending on the rules of appearance of symbols are proposed, and each model is built up by the nondeterministic finite automaton.

Each element in set CS of consonant symbols is defined as c_c $\{c_c : c_c \in CS\}$. Also, the double consonant for two adjacent symbols

$c_i c_j$ is defined as a set C_d (5-14). Therefore, if $c_i c_j$ is not an element of set C_d; $c_i c_j \notin C_d$, then c_i and c_j correspond to a vowel abbreviated model according to Rule 7. Because a symbol is fetched by the recognizer one by one, set C_d must be separated to the subsets composed of each symbol in the set.

The following subsets are introduced for two adjacent consonants.

(1) The sets are classified into 3 groups depending on the kind of combination of two symbols as follows:

In the double consonant,

(i) Subsets C_{i1} and C_{j1}: First group

If $C_{i1}=\{c_1,c_2,c_3\}$,

then $C_{j1}=\{c_{30},c_{32},c_{43}\}$

(ii) Subsets C_{i2} and C_{j2}: Second group

If $C_{i2}=\{c_{13},c_{24},c_{25}\}$,

then $C_{j2}=\{c_{30},c_{32}\}$

(iii) Subsets C_{i3} and C_{j3}: Third group

If $C_{i3}=\{c_{26}\}$,

then $C_{j3}=\{c_{30}\}$

(2) First consonant does not exist in that double consonant.

(i)Subset C_w

$C_w = CS - \{ C_{i1} \cup C_{i2} \cup C_{i3} \}$

(3) Second consonant does not exist in the set of second symbols

within double consonant.

(i)Subset C_{r1}

$$C_{r1} = CS_e - C_{j1}$$

(ii)Subset C_{r2}

$$C_{r2} = CS_e - C_{j2}$$

(iii)Subset C_{r3}

$$C_{r3} = CS_e - C_{j3}$$

Here, each element is discriminated as follows;

$a_1 : a_1 \in C_{i1}$, $b_1 : b_1 \in C_{j1}$, $a_2 : a_2 \in C_{i2}$, $b_2 : b_2 \in C_{j2}$,
$a_3 : a_3 \in C_{i3}$, $b_3 : b_3 \in C_{j3}$, $w : w \in C_w$,
$r_1 : r_1 \in C_{r1}$, $r_2 : r_2 \in C_{r2}$, and $r_3 : r_3 \in C_{r3}$ respectively.

Assume that each tonal mark is included in a preceding consonant or vowel, and represented by the denotation of the preceding symbol.

Consequently, a double consonant and abbreviated vowel model are represented as follows:

[A] A double consonant and abbreviated vowel model:[C + C]

This model provides the [C + C] case either two consonants are a double consonant or the vowel is abbreviated.

Finite automaton:

$M = \langle K,S,P,q_0,\{q_1\}\rangle$

Set of state:

$K = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

Input symbol:

$S=s^*$

Set of transition function:

$P = \{ d(q_0,a_1)=q_3,\ d(q_0,a_2)=q_4,\ d(q_0,a_3)=q_5,$
$d(q_0, w)=q_2,\ d(q_3,b_1)=q_6,\ d(q_4,b_2)=q_6,$
$d(q_5,b_3)=q_6,\ d(q_3,r_1)=q_1,\ d(q_4,r_2)=q_1,$
$d(q_5,r_3)=q_1,\ d(q_2,c_e)=q_1\ \}$

Figure 5.6(a) indicates a state diagram of the recognizer which represents a finite automaton model as above.

state	a_1	a_2	a_3	w	b_1	b_2	b_3	r_1	r_2	r_3	c_e	Accept state?
q_0	q_3	q_4	q_5	q_2								no
q_3					q_6			q_1				no
q_4						q_6			q_1			no
q_5							q_6			q_1		no
q_6												cont.
q_2											q_1	no
q_1												yes

Fig.5.6(b) Transition table of the double consonant and abbreviated vowel model

[B] General Model: [C + V + C] and [C + V]

This model provides [C + V] and [C + V + C] cases of which the symbol at the beginning of monosyllable is the consonant.

From the set VS_n, define VS_{n2} as:

$$VS_{n2} = VS_n - \{ v_2, v_4, v_{12}, v_{14} \}.$$

Each element of set VS_{n2} is v_{n2} $\{ v_{n2} : v_{n2} \in VS_{n2} \}$.

Finite automaton:

$$M = \langle K, S, P, q_0, \{q_1, q_7, q_{10}\} \rangle$$

Set of state:

$$K = \{q_0, q_6, q_7, q_8, q_9, q_{10}, q_1\}$$

Input symbol:

$$S = s^*$$

Set of transition function:

$$P = \{ d(q_0,c_c)=q_6, \quad d(q_6,v_2)=q_9, \quad d(q_6,v_{12})=q_8,$$
$$d(q_6,\{v_4,v_{14}\})=q_7, \quad d(q_6,v_{n2})=q_1,$$
$$d(q_9,v_{12})=q_{10}, \quad d(q_9,c_e)=q_1, \quad d(q_{10},v_6)=q_1,$$
$$d(q_8,c_e)=q_1, \quad d(q_7,v_{15})=q_1 \}$$

Figure 5.7(a) indicates a state diagram of the recognizer which represents a finite automaton model as above.

state	c_c	v_2	v_6	v_4	v_{12}	v_{14}	v_{15}	v_{n2}	c_e	Accept state?
q_0	q_6									no
q_6		q_9		q_7	q_8	q_7		q_1		no
q_7							q_1			yes
q_8									q_1	no
q_9					q_{10}				q_1	no
q_{10}			q_1							yes
q_1										yes

Fig.5.7(b) Transition table of the general model

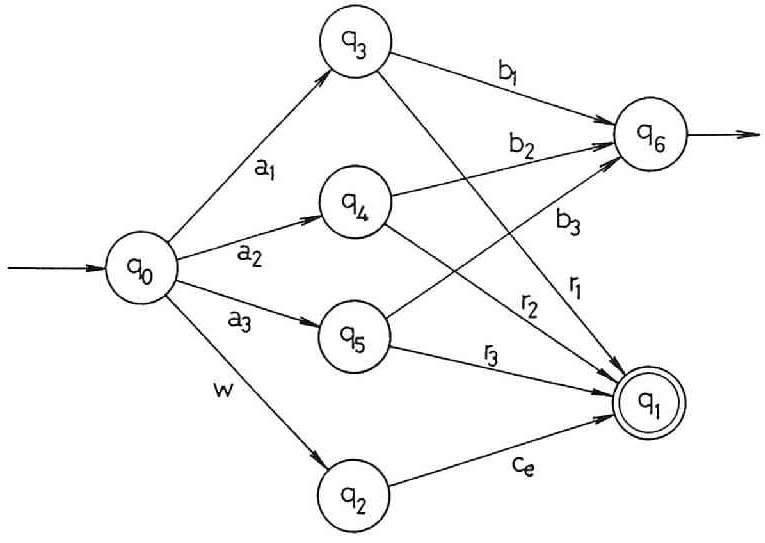


Fig.5.6(a) State diagram of the double consonant and abbreviated vowel model

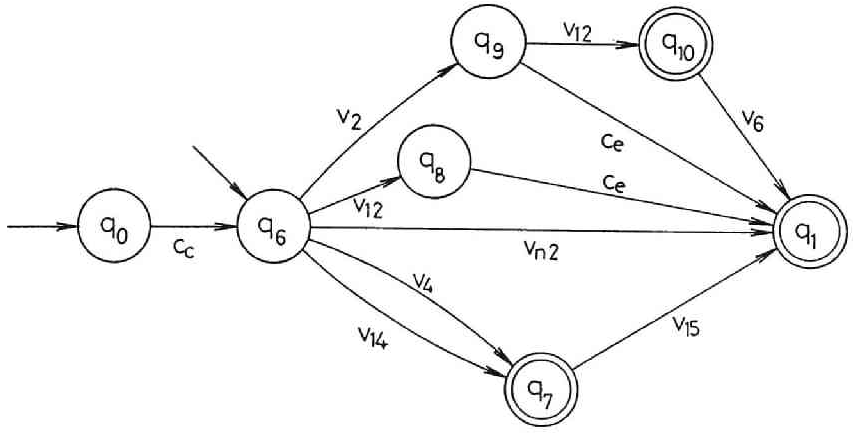


Fig.5.7(a) State diagram of the general model

[C] Deformed model (1) : [C + V] and [C + V + C]

This model provides the [C + V] and [C + V + C] cases of which the symbol at the beginning of monosyllable is v_{16} or v_{17}.

Finite automaton:

$$M = \langle K, S, P, q_0, \{q_1, q_{12}, q_{13}, q_{14}, q_{15}\} \rangle$$

Set of state:

$$K = \{ q_0, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_1 \}$$

Input symbol:

$$S = s^*$$

Set of transition function:

$$P = \{ d(q_0,\{v_{16},v_{17}\})=q_{11},\ d(q_{11},a_1)=q_{12},$$
$$d(q_{11},a_2)=q_{13},\ d(q_{11},a_3)=q_{14},\ d(q_{12},r_1)=q_1,$$
$$d(q_{12},b_1)=q_{15},\ d(q_{13},b_2)=q_{15},\ d(q_{13},r_2)=q_1,$$
$$d(q_{14},b_3)=q_{15},\ d(q_{11}, w)=q_{15},\ d(q_{14},r_3)=q_1,$$
$$d(q_{15},c_e)=q_1 \}$$

Figure 5.8(a) indicates a state diagram of the recognizer which represents a finite automaton model as above.

state	v_{16}	v_{17}	a_1	a_2	a_3	w		Accept state?
q_0	q_{11}	q_{11}						no
q_{11}			q_{12}	q_{13}	q_{14}	q_{15}		no

continued

state	b_1	b_2	b_3	r_1	r_2	r_3	c_e	Accept state?
q_{12}	q_{15}			q_1				yes
q_{13}		q_{15}			q_1			yes
q_{14}			q_{15}			q_1		yes
q_{15}							q_1	yes
q_1								yes

Fig.5.8(b) Transition table of the deformed model (1)

[D] Deformed model (2): [C + V] and [C + V + C]

This model provides the [C + V] and [C + V + C] cases of which the symbol at the beginning of monosyllable is v_{18}.

Finite automaton:

$$M = \langle K, S, P, q_0, \{q_1, q_{17}, q_{18}, q_{19}, q_{20}\} \rangle$$

Set of state:

$$K = \{ q_0, q_{16}, q_{17}, q_{18}, q_{19}, q_{20}, q_1 \}$$

Input symbol:

$$S = s^*$$

Set of transition function:

$$P = \{ d(q_0,v_{18})=q_{16},\ d(q_{16},a_1)=q_{17},\ d(q_{16},a_2)=q_{18},$$
$$d(q_{16},a_3)=q_{19},\ d(q_{16}, w)=q_{20},\ d(q_{17},\{r_1,v_6\})=q_1,$$
$$d(q_{18},\{r_2,v_6\})=q_1,\ d(q_{19},\{r_3,v_6\})=q_1,$$
$$d(q_{17},b_1)=q_{20},\ d(q_{18},b_2)=q_{20},\ d(q_{19},b_3)=q_{20},$$
$$d(q_{20},\{v_6,c_e\})=q_1 \}$$

Figure 5.9(a) indicates a state diagram of the recognizer which represents a finite automaton model as above.

state	v_{18}	a_1	a_2	a_3	w	Accept state?
q_0	q_{16}					no
q_{16}		q_{17}	q_{18}	q_{19}	q_{20}	no

continued

state	b_1	b_2	b_3	v_6	r_1	r_2	r_3	c_e	Accept state?
q_{17}	q_{20}			q_1	q_1				yes
q_{18}		q_{20}		q_1		q_1			yes
q_{19}			q_{20}	q_1			q_1		yes
q_{20}				q_1				q_1	yes
q_1									yes

Fig.5.9(b) Transition table of deformed model (2)

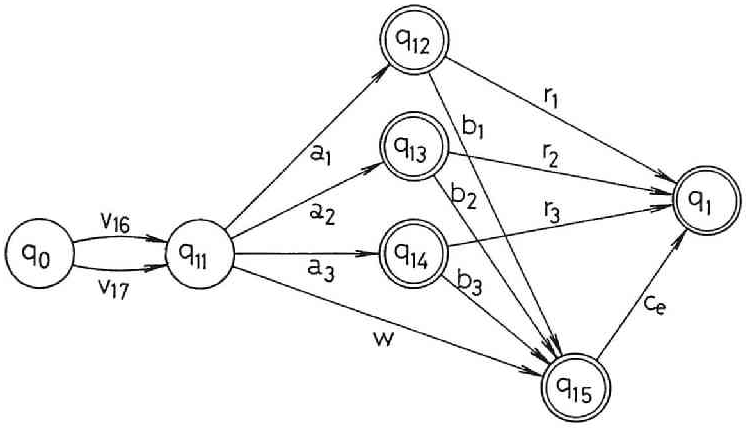


Fig.5.8(a) State diagram of deformed model (1)

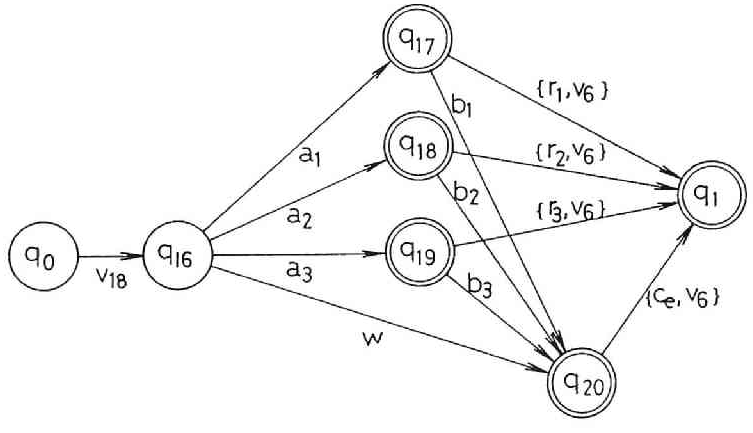


Fig.5.9(a) State diagram of deformed model (2)

[E] Deformed model (3): [C + V] and [C + V + C]

This model provides the [C + V] and [C + V + C] cases of which the symbol at the beginning of monosyllable is v_{19}.

Finite automaton:

$M = < K, S, P, q_0, \{q_{22}, q_{23}, q_{24}, q_{25}, q_1\} >$

Set of state:

$K = \{ q_0, q_{21}, q_{22}, q_{23}, q_{24}, q_{25}, q_1 \}$

Input symbol:

$S = s^*$

Set of transition function:

$P = \{ d(q_0,v_{19})=q_{21}, \; d(q_{21},a_1)=q_{22}, \; d(q_{21},a_2)=q_{23},$
$d(q_{21},a_3)=q_{24}, \; d(q_{21}, w)=q_{25}, \; d(q_{22},b_1)=q_{25},$
$d(q_{23},b_2)=q_{25}, \; d(q_{24},b_3)=q_{25},$
$d(q_{22},\{r_1,v_4,v_6\})=q_1,$
$d(q_{23},\{r_2,v_4,v_6\})=q_1, \; d(q_{24},\{r_3,v_4,v_6)=q_1,$
$d(q_{25},\{c_e,v_4,v_6\})=q_1 \; \}$

Figure 5.10(a) indicates a state diagram of the recognizer which represents a finite automaton model as above.

state	v_{19}	a_1	a_2	a_3	w	b_1	b_2	b_3	Accept state?
q_0	q_{21}								no
q_{21}		q_{22}	q_{23}	q_{24}	q_{25}				no
q_{22}						q_{25}			yes
q_{23}							q_{25}		yes
q_{24}								q_{25}	yes

continued

state	r_1	r_2	r_3	v_4	v_6	c_e	Accept state?
q_{22}	q_1			q_1	q_1		yes
q_{23}		q_1		q_1	q_1		yes
q_{24}			q_1	q_1	q_1		yes
q_{25}				q_1	q_1	q_1	yes
q_1							yes

Fig.5.10(b) Transition table of deformed model (3)

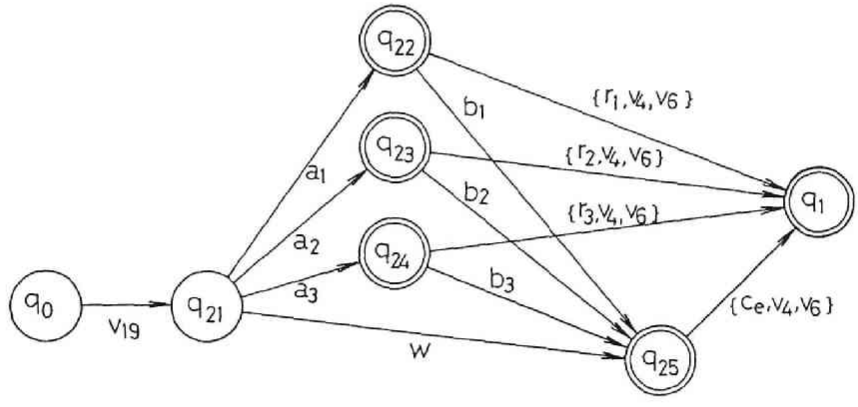


Fig.5.10(a) State diagram of deformed model (3)

[F] Deformed model (4): [C + V] and [C + V + C]

This model provides the [C + V] and [C + V + C] cases of which the symbol at the beginning of monosyllable is v_{20}.

Finite automaton:

$M = < K, S, P, q_0, \{q_{28}, q_{29}, q_{30}, q_{31}, q_{34}, q_1\} >$

Set of state:

$K = \{ q_0, q_{27}, q_{28}, q_{29}, q_{30}, q_{31}, q_{32}, q_{33}, q_{34}, q_1 \}$

Input symbol:

$S = s^*$

Set of transition function:

$P = \{ d(q_0,v_{20})=q_{27}, \; d(q_{27},a_1)=q_{28}, \; d(q_{27},a_2)=q_{29},$
$d(q_{27},a_3)=q_{30}, \; d(q_{27},c_c)=q_{31}, \; d(q_{28},b_1)=q_{31},$
$d(q_{29},b_2)=q_{31}, \; d(q_{30},b_3)=q_{31}, \; d(q_{28},r_1)=q_1,$
$d(q_{29},r_2)=q_1, \; d(q_{30},r_3)=q_1, \; d(q_{31},v_8)=q_{32},$
$d(q_{31},v_{14})=q_{33}, \; d(q_{31},\{v_3,v_{15}\})=q_{34},$
$d(q_{32},v_9)=q_{34}, \; d(q_{33},v_{15})=q_{34}, \; d(q_{34},v_6)=q_1,$
$d(q_{31},\{c_e,v_4,v_6,v_7\})=q_1 \}$

Figure 5.11(a) indicates a state diagram of the recognizer which represents a finite automaton model as above.

state	v_{20}	a_1	a_2	a_3	c_c	b_1	b_2	b_3	r_1	r_2	r_3	Accept state?
q_0	q_{27}											no
q_{27}		q_{28}	q_{29}	q_{30}	q_{31}							no
q_{28}						q_{31}			q_1			yes
q_{29}							q_{31}			q_1		yes
q_{30}								q_{31}			q_1	yes
q_{31}												cont.

continued

state	v_3	v_4	v_6	v_7	v_8	v_9	v_{14}	v_{15}	c_e	Accept state?
q_{31}	q_{34}	q_1	q_1	q_1	q_{32}		q_{33}	q_{34}	q_1	yes
q_{32}						q_{34}				no
q_{33}								q_{34}		no
q_{34}			q_1							yes
q_1										yes

Fig.5.11(b) Transition table of deformed model (4)

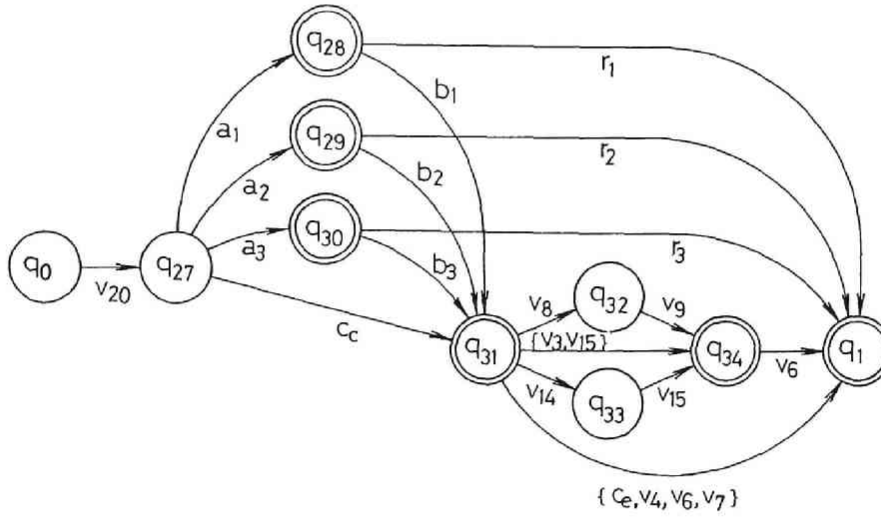


Fig.5.11(a) State diagram of deformed model (4)

5.5 Automatic Segmentation using Thai Syllable Recognizer

5.5.1 Implementation of Thai Syllable Recognizer

On the basis of the previous models, a syntax-directed recognizer, which enables automatic and consecutive segmentation for the recognizable portion of words in a Thai sentence has been designed. Figure 5.12 is a sketch of the structure of the Thai syllable recognizer, and an outline of the recognizer machine is shown in Fig.5.13.

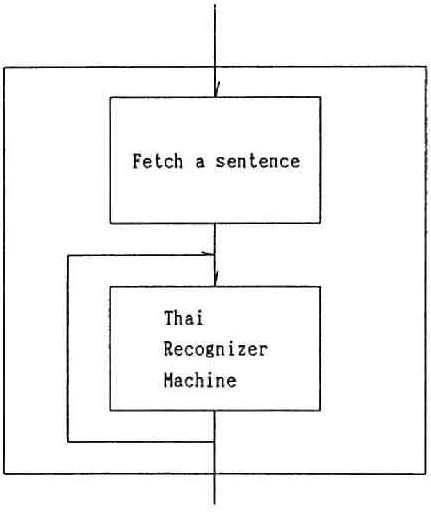


Fig.5.12 Structure of Thai syllable recognizer

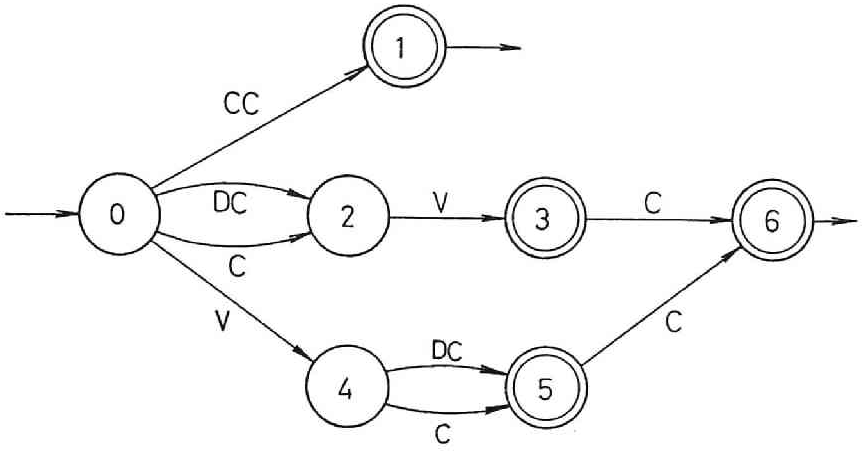


Fig.5.13 Model of Thai syllable recognizer

The model of the Thai syllable recognizer was introduced in the preceding section, where CC, DC, C, and V imply words with abbreviated vowel, double consonant, general consonant and vowel(s) respectively. Also, each numeral in the circles represents a state, and the accept states are distinguished by being drawn with a double circle in the recognizer.

One of the most important characteristics in this recognizer is that no dictionary is used, whereas the unsegmental characteristics generally need the use of a dictionary for segmentation. Segmentation without reference to the dictionary or by reducing the capacity of the dictionary reduce the time spent for dictionary reference. The capacity, data structure, or access method for the dictionary greatly influences the efficiency of segmentation.

5.5.2 Experiment using the recognizer

The use of a previous recognizer which segments a sentence for the same text, KTSD, in an experiment of the longest-match method has been attempted. The characteristics of the text are shown in Table 5.1. To analyze the result of segmentation in detail, especially about the ratio of segmentation and the sequence of the appearance of symbols in the unsuccessful cases, a Thai-Thai dictionary composed of 31,202 main entries (See Chapter 4) has been used. The frequency of occurrence of Thai letters in the main entries of the dictionary is shown in Table 4.2 in Chapter 4.

Segmented words obtained as the result of segmentation by the recognizer are specialized as monosyllables composed of [C + V] or [C + V + C], whereas the words before segmenting the sentences for the KTSD consist of monosyllables or compound words. Therefore, a criterion of a recognizable unit as a syllable is;

Original sentence

/ v_{19} c_{30} / v_{16} c_{42} t_2 c_1 c_{32} v_6 c_{14} v_{10} t_1 c_{42} /

Input sentence

v_{19} c_{30} v_{16} c_{42} t_2 c_1 c_{32} v_6 c_{14} v_{10} t_1 c_{42}

Segmented syllable by recognizer

/ v_{19} c_{30} / v_{16} c_{42} t_2 / c_1 c_{32} v_6 / c_{14} v_{10} t_1 c_{42} /.

Criteria of recognition are as follows:

(1) Delimiters at the position in the original sentence corresponded to the delimiters of the segmented sentence.

(2) Each symbol embedded in a string surrounded by both delimiters, "/" and "/", has the sequence [C + V] or [C + V + C] based on the rules in section 5.2.

If the above criteria are satisfied, a string between both delimiters "/" is to be recognized as a syllable, and this syllable is searched for in the main entries of the dictionary in order to confirm whether the syllable is registered or not.

5.5.3 Evaluation

The result of the experiment by inputting the sentences of the KTSD is shown in Table 5.3.

Table 5.3 Result of segmentation by recognizer

Input sentences	20,631	
Number of words	252,619	
Non segmented sentences	10,406	
Number of generated words	417,179	
Words not found	178,915	(42.9%)
Non segmented words	59,571	
Segmentation Ratio Sentence unit		49.6%

From the results, the segmentation ratio in terms of sentences becomes very much lower than by 95.2% of the longest-match method. Some of the biggest reasons for unsuccessful cases are as follows:
(1) Segmentation by string type [C + V] according to Fig.5.2(a)

happens in spite of the type [C + V + C]; for example, จาก and คุณ are case 4 and 11 respectively in Fig.5.2(a).
(2) Segmentation by string type [C + V] according to Fig.5.2(b) happens in spite of the type [C + V + C]; for example, เป็น and เขียง are case 11 and 14 respectively in Fig.5.2(b).

5.5.4 Heuristic Approach to the Segmentation

As for the key reasons for improperly segmented words, it is found that the rule for type [C + V + C] does not matched since all of the rules are only deduced from Thai grammatical rules and neither the characteristics in the case of (1) above, nor the case of (2) above has not been installed to the recognizer. For these two reasons, the heuristic approach that has rules based on experience is the most valuable method.

On the basis of heuristics, the following rules are deduced:
(1) When v_5 or v_6 appeared, those symbols imply a last symbol at end of the monosyllable. Then the following symbol, of course, should be a consonant in the top position in the monosyllable.
(2) Let subset VS_{n3} be Vowel Symbols given by

$$VS_{n3} = VS_n - \{ v_2, v_4, v_5, v_6, v_{12}, v_{14} \}$$
$$= \{ v_3, v_7, v_8, v_{10}, v_{11}, v_{13}, v_{15} \},$$

instead of the set VS_{n2} in [B]General model.

If input symbol s is equal to V_{n3} $\{ V_{n3} : V_{n3} \in VS_{n3} \}$, then, a consonant c_e in the set CS_e (5-12) sometimes is added.

(3) After the vowel v_4 and v_{14} in the general vowel: [C + V] or [C + V + C], a consonant c_e in the set CS_e sometimes is added.
(4) The following symbol after the vowel v_4 in the deformed model(3) sometimes is a consonant c_e in the set CS_e.
(5) In the deformed model (4), a consonant c_e in the set CS_e sometimes is added after the vowels v_4, v_7, v_9, and v_{15}.

Here, the first revised model, Revised model(1), which incorporates three features of the above rules (1), (2), and (3) for the general model as shown in Fig.5.14(a) is proposed. The second revised model, Revised model(2), which incorporates two features of the above rules (1) and (4) for the deformed model(3), and the third revised model, Revised model(3), which incorporates two features of the above rules (1) and (5) for the deformed model(4) are proposed as shown in Fig.5.15(a) and Fig.5.16(a) respectively.

A revised finite automaton model transformed according to the general model is as follows:

[A] Revised model(1) for the general model

Finite automaton:

$$M = \langle K, S, P, q_0, \{ q_1, q_7, q_8, q_{10} \} \rangle$$

Set of state:

$$K = \{ q_0, q_6, q_7, q_8, q_9, q_{10}, q_1 \}$$

Input symbol:

$$S = s^*$$

Set of transition function:

$P = \{ d(q_0, c_c)=q_6,\ d(q_6,v_2)=q_9,\ d(q_9,v_{12})=q_{10},$
$d(q_{10},\{v_6,c_e\})=q_1,\ d(q_9,c_e)=q_1,\ d(q_6,\{v_{n3},v_{12}\})=q_8,$
$d(q_8,c_e)=q_1,\ d(q_6,\{v_5,v_6\})=q_1,$
$d(q_6,\{v_4,v_{14}\})=q_7,\ d(q_7,\{v_{15},c_e\})=q_1 \}$

Figure 5.14(a) indicates a state diagram of the recognizer which represents a finite automaton model as above.

state	c_c	v_2	v_4	v_5	v_6	v_{12}	v_{14}	v_{15}	c_e	v_{n3}	Accept state?
q_0	q_6										no
q_6		q_9	q_7	q_1	q_1	q_8	q_7			q_8	no
q_7								q_1	q_1		yes
q_8									q_1		yes
q_9						q_{10}			q_1		no
q_{10}					q_1				q_1		yes
q_1											yes

Fig.5.14(b) Transition table of revised model(1)

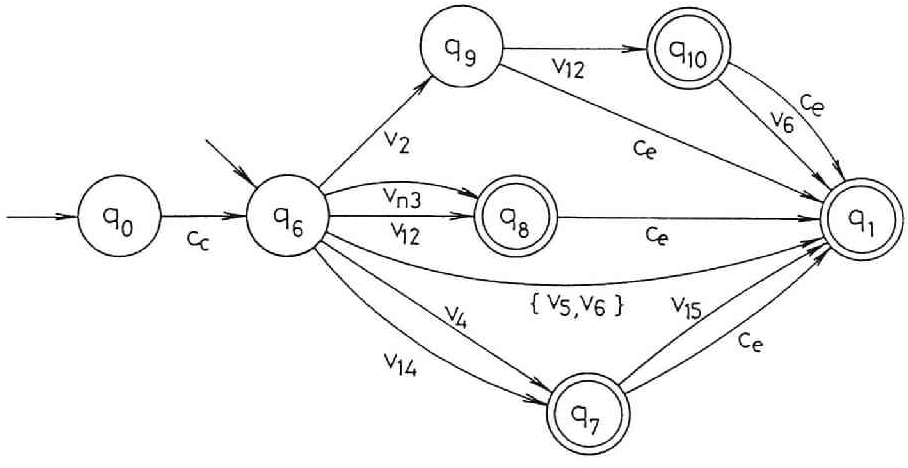


Fig.5.14(a) State diagram of revised model(1)

[B] Revised model(2) for the deformed model(3)

Finite automaton:

$M = \langle K, S, P, q_0, \{q_{22}, q_{23}, q_{24}, q_{25}, q_{26}, q_1\} \rangle$

Set of state:

$K = \{ q_0, q_{21}, q_{22}, q_{23}, q_{24}, q_{25}, q_{26}, q_1 \}$

Input symbol:

$S = s^*$

Set of transition function:

$P = \{ d(q_0,v_{19})=q_{21}, \quad d(q_{21},a_1)=q_{22}, \quad d(q_{21},a_2)=q_{23},$
$d(q_{21},a_3)=q_{24}, \quad d(q_{21}, w)=q_{25}, \quad d(q_{22},b_1)=q_{25},$
$d(q_{23},b_2)=q_{25}, \quad d(q_{24},b_3)=q_{25}, \quad d(q_{22},v_4)=q_{26},$
$d(q_{23},v_4)=q_{26}, \quad d(q_{24},v_4)=q_{26}, \quad d(q_{25},v_4)=q_{26},$
$d(q_{22},\{r_1,v_6\})=q_1, \quad d(q_{23},\{r_2,v_6\})=q_1,$
$d(q_{24},\{r_3,v_6\})=q_1, \quad d(q_{25},\{c_e,v_6\})=q_1,$
$d(q_{26},c_e)=q_1 \}$

Figure 5.15(a) indicates a state diagram of the recognizer which represents a finite automaton model as above.

state	v_{19}	a_1	a_2	a_3	w	Accept state?
q_0	q_{21}					no
q_{21}		q_{22}	q_{23}	q_{24}	q_{25}	no

continued

state	b_1	b_2	b_3	v_4	v_6	r_1	r_2	r_3	c_e	Accept state?
q_{22}	q_{25}			q_{26}	q_1	q_1				yes
q_{23}		q_{25}		q_{26}	q_1		q_1			yes
q_{24}			q_{25}	q_{26}	q_1			q_1		yes
q_{25}				q_{26}	q_1				q_1	yes
q_{26}									q_1	yes
q_1										yes

Fig.5.15(b) Transition table of revised model(2)

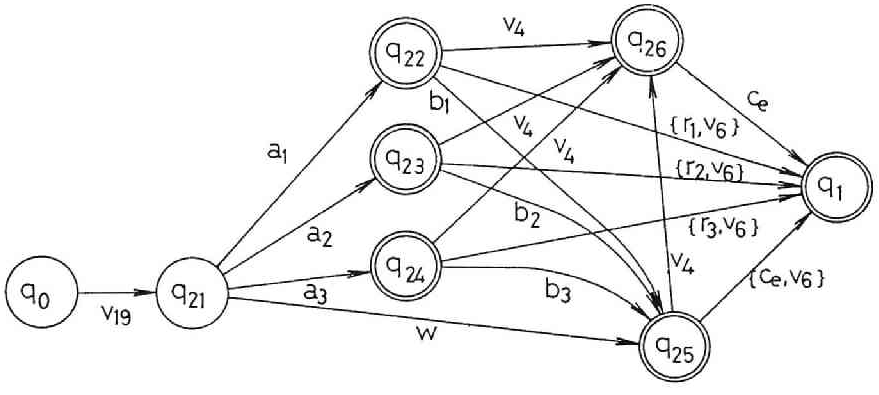


Fig.5.15(a) State diagram of revised model(2)

[C] Revised model(3) for the deformed model(4)

Finite automaton:

$M = < K, S, P, q0, \{q_{28}, q_{29}, q_{30}, q_{31}, q_{34}, q_{35}, q_{36}, q_1\} >$

Set of state:

$K = \{ q_0, q_{27}, q_{28}, q_{29}, q_{30}, q_{31}, q_{32}, q_{33}, q_{34}, q_{35}, q_{36}, q_1 \}$

Input symbol:

$S = s^*$

Set of transition function:

$P = \{ d(q_0,v_{20})=q_{27}, \; d(q_{27},a_1)=q_{28}, \; d(q_{27},a_2)=q_{29},$
$d(q_{27},a_3)=q_{30}, \; d(q_{27},c_c)=q_{31}, \; d(q_{28},b_1)=q_{31},$
$d(q_{29},b_2)=q_{31}, \; d(q_{30},b_3)=q_{31}, \; d(q_{31},v_8)=q_{32},$
$d(q_{31},v_{14})=q_{33}, \; d(q_{31},v_{15})=q_{34}, \; d(q_{32},v_9)=q_{34},$
$d(q_{31},\{v_4,v_7\})=q_{35}, \; d(q_{33},v_{15})=q_{34},$
$d(q_{31},v_3)=q_{36}, \; d(q_{28},r_1)=q_1, \; d(q_{29},r_2)=q_1,$
$d(q_{30},r_3)=q_1, \; d(q_{31},\{v_6,c_e\})=q_1,$
$d(q_{34},\{v_6,c_e\})=q_1, \; d(q_{35},c_e)=q_1, \; d(q_{36},v_6)=q_1 \}$

Figure 5.16(a) indicates a state diagram of the recognizer which represents a finite automaton model as above.

state	v_{20}	a_1	a_2	a_3	c_c	b_1	b_2	b_3	r_1	r_2	r_3	Accept state?
q_0	q_{27}											no
q_{27}		q_{28}	q_{29}	q_{30}	q_{31}							no
q_{28}						q_{31}			q_1			yes
q_{29}							q_{31}			q_1		yes
q_{30}								q_{31}			q_1	yes
q_{31}												cont.

continued

state	v_3	v_4	v_6	v_7	v_8	v_9	v_{14}	v_{15}	c_e	Accept state?
q_{31}	q_{36}	q_{35}	q_1	q_{35}	q_{32}		q_{33}	q_{34}	q_1	yes
q_{32}						q_{34}				no
q_{33}								q_{34}		no
q_{34}			q_1						q_1	yes
q_{35}									q_1	yes
q_{36}			q_1							yes
q_1										yes

Fig.5.16(b) Transition table of revised model(3)

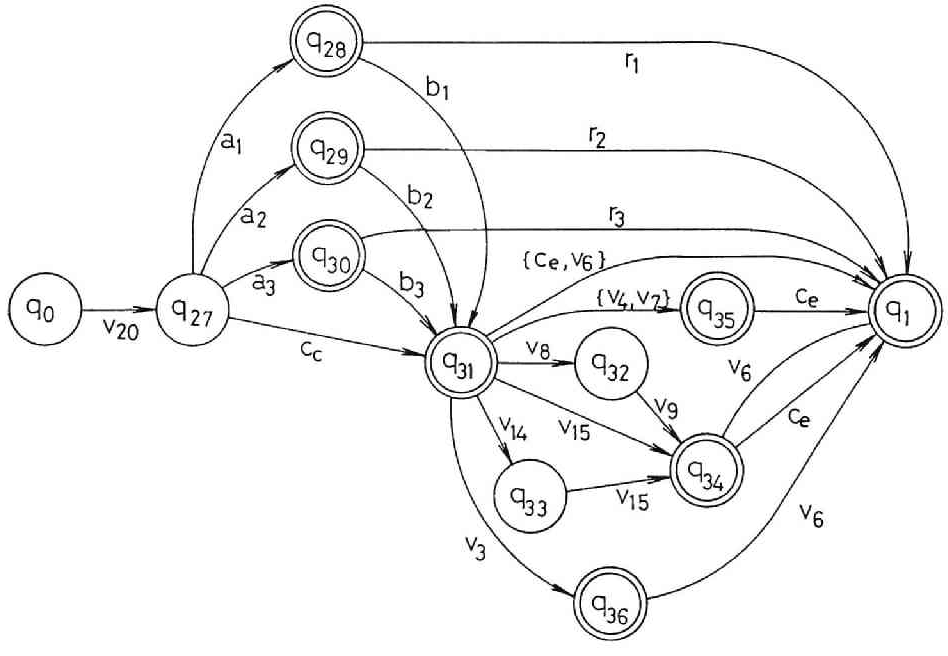


Fig.5.16(a) State diagram of revised model(3)

The result of the experiment using the revised model, is shown in Table 5.4.

Table 5.4 Result of experiment of revised model

Non segmented sentences	1,269	
Number of generated words	401,577	
Words not found		
Thai-Thai dictionary	98,021	(24.4%)
KTSD dictionary	112,095	(27.9%)
Non segmented words	1,371	
Segmentation Ratio sentence unit		93.9%

From the experiment in the input of the KTSD text, it is found that the segmentation ratio attained at most to 49.6% by the recognizer based on the syllable formation rules only.

By adapting the knowledge rules based on heuristics derived from the analysis of unsuccessful cases into exiting syllable formation rules, it is found that the ratio of segmentation has been improved by 93.9%, and it is 44.3% higher than that by the previous experiment of the recognizer, whereas 4.1% lower than the method by the Syllable Longest-Match method.

Class	Example	Frequency
(1)	/v_{20} c_1/ c_{30} t_1 v_{12}/ /v_{20} c_1 c_{30} t_1/ v_{12} ...	362
(2)	/c_7 c_{23} c_{32} v_2 c_1 c_{36}/ c_{44} v_2 c_{21}/ /c_7 c_{23}/c_{32} v_2 c_1/ c_{36} v_{15}/ v_2 ...	343
(3)	/c_{41} v_{10} c_{13} c_{32} c_{44} v_8/ /c_{41} v_{10}/c_{13} c_{32} v_{15}/ v_8 ...	37
(4)	/c_{35} c_{43} v_2 c_{11} v_8/ /c_{35} v_{12}/ v_2 ...	278
(5)	/c_{32} v_2 c_{25} c_{33}/ c_1 c_{43} v_3/ /c_{32} v_2 c_{25}/ c_{33} c_1 v_{12} ...	25
(6)	... c_{16} c_{32} c_{32} c_{42}/ c_{44} v_2 c_{21}/	74
(7)	/c_{41} v_{10} v_6/	61
(8)	/v_{20} c_7 v_8/ c_{33} v_2 c_{23}/ /v_{20} c_7 v_8 v_9/ v_2 ...	39
(9)	/c_{16} c_{32} c_{17} c_{44} v_7 c_{37} c_{43} c_{32}	4
(10)	/c_1 c_{32} c_{42} c_3 c_{43} v_3/	46
		1269

Unit of frequency: Sentences
Upper line: Input string
Below line: Recognized pattern

Fig.5.17 Enumeration of unsuccessful cases by the revised recognizer

Figure 5.17 shows the classification and its enumeration of improperly segmented sentences. The key reasons in the column 'class'

in Fig.5.17, which are classified into 10 classes, are as follows:

(1) The following two consonants after a vowel of [Rule 2], namely, by forming a type of '[V + C] + [C + V]', was recognized as the double consonant [C + C], whereas its word was formed by a combination of both monosyllable [C + V]s.

(2) The consonant c_{44} was recognized as a vowel v_{15} because the preceding syllable was [C + V + C + C] type, and the following one is [C + V].

(3) The following consonant c_{44} after the double consonant was recognized as a vowel v_{15} in the type [C + C + V + V].

(4) The second consonant c_{43} embedded in the monosyllable was recognized as the vowel v_{12}.

(5) The second consonant c_{43} in the syllable "Double consonant + Vowel" was recognized as a vowel v_{12} when the preceding syllable was [C + V + C + C].

(6) Successive two consonants of c_{32} were appeared. It should be replaced by a vowel v_2 based on the grammatical rule of Thai in advance.

(7) The following two vowels after a consonant were appeared. They were not in Fig.5.2 and syllable formation rules.

(8) The consonant c_{33} was recognized as a vowel v_9 according to $d(q_{32},v_9)=q_{34}$ in Fig.5.11(a), because the preceding symbol was a vowel v_8.

(9) They are the proper noun, which have no [C + V] or [C + V + C] type.

(10) They are the other cases like "Double consonant + Consonant +

Double consonant + Vowel".

A brief summary of key reasons mentioned above are characterized as follows:

(1) Several graphemes are used for both consonant and vowel, so the correct recognition cannot be performed. For example, c_{33}, c_{43}, and c_{44} are equivalent to v_9, v_{12}, and v_{15} respectively.

(2) The sequence of graphemes embedded in a word are [V + C] + [C + V] whereas that of phonemes are [C + V] + [C + V]. Namely, the difference between sequence of phonemes and graphemes in the syllabic structure causes the segmentation to be improperly segmented.

(3) The word is a proper noun, but it has no combination of [C + V] or [C + V + C].

(4) The sequence of graphemes in a word is irregular, which differs from the syllable formation rules. For example, successive two vowels which are being irregular are embedded in a word.

Chapter 6 Thai Printing System

6.1 Introduction

The craft of representing written language has been greatly influenced by the technology that has been developed in support of the writing and printing process [Yamada 81]. While these major developments in orthographic technology have been focused mainly on the Roman scripts, adaptations also have been made to accommodate a wide range of non-Roman scripts, for example, an academic typesetting facility which deals with non-Latin scripts such as Hebrew, Arabic, Devanagari, and so on at Oxford University [Griffin 81], synthesizing system of Devanagari orthography [Millar & Glover 81], and the mechanical processing of Asian and African languages [Sakamoto 79]. These scripts, including Thai, differ from the Roman script in a variety of ways. The most obvious differences are in the size and shape of the alphabet.

Thai orthography has a much larger number of symbols than the Roman. A Thai letter may be divided into three vertical and horizontal regions, as illustrated in Fig.2.2. Each region may hold a partial character which represents a vowel, consonant, or tonal mark. Each partial that needs to be added in either the upper or lower part of the character is implemented as a separate slug in

the IBM electronic typewriter [Shibayama 87].

The Thai printing system for a Japanese laser beam printer runs on a main-frame computer and a CRT display for a personal computer has been designed.

The characteristics of the system are that it generates Thai letters with a character set adequately large and well-formed to satisfy the user's calligraphic preference by choosing and combining appropriate graphemes based on the contextual constraints. An algorithm and a table-driven context sensitive optimization table which represents the contextual constraints are proposed. Furthermore, a justification algorithm at the right margin is introduced.

6.2 Thai Letter Synthesizing

It is found that a character pattern composed of 16(W)*32(H) mesh stored in RAM (Random Access Memory) can be displayed without appreciable delay by using the PUT@ statement in the graphic instructions of Basic language.

A new pattern in order to display the vowel and tonal marks on the GVRAM (Graphic Video RAM) by using the OR operation for all bits of the patterns has been synthesized as shown in Fig.6.1. This figure also shows that patterns other than tones should be shifted by 5 dots downward in the Y direction from the standard position in order to save the main memory area for storing all patterns.

This scheme has been implemented on the personal computer PC-9800 series, and equipped in the Thai text editor (See Chapter 4).

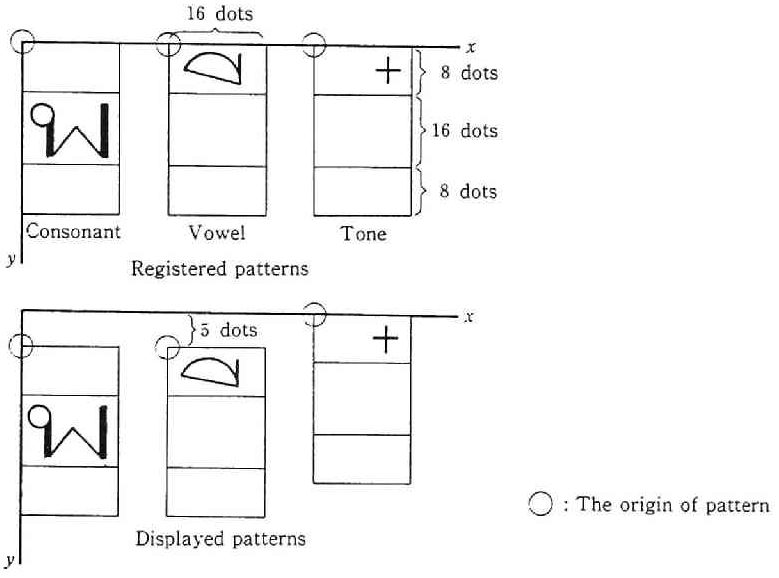


Fig.6.1 The Display on the CRT screen

6.3 Thai Printing Technology

Thai letters have the following characteristics:

(1) they differ from each other in size and as well as shape, for example, โ、 า、ญ、ำ、ฤ、and ด ,

(2) several letters are overlapped in printing, for example, ปั่ is composed of ป and ่ ,

(3) several letters are located above and between adjacent letters, like จัก ,

(4) the printing positions of the same letter are sometimes different, like ดั่ง and ว่า for a letter " ่ ".

Such problems can be solved for the printing of almost all

Asian and African letters by dividing letter patterns into sub patterns, namely partials, comprising the basic character with special information on the basic line and the added character with information on its basic point [Sakamoto 79]. This idea has been adopted in the design of Thai printing system.

A more controllable program that allows any line spacing with an integral number of dots by adding functions for overlap control of the character patterns, line position control for each line, and vertical position control of a character depending on the position of the previous character, namely, the contextual constraints has been developed.

The schemes of these controls are as follows:

(1) Discrimination of basic and added characters

Thai letters in combinations of consonant + vowel or consonant + vowel + consonant are composed of 6 regions centered on the first consonant of a word, as shown in Fig.6.2.

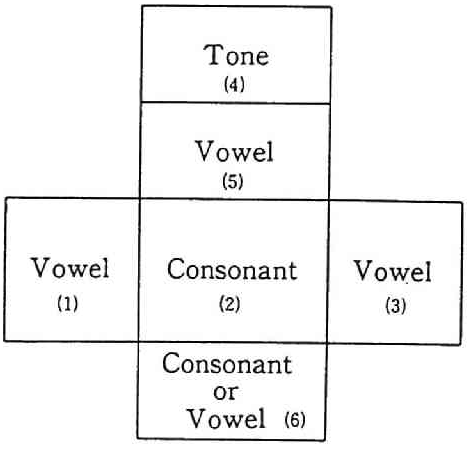


Fig.6.2 Positions of Consonants, vowels, and Tones

The characters located at (1), (2), and (3) in this Figure are categorized as basic characters, and those at (4), (5), and (6) as added characters. It is assumed that the sequence of the appearance of characters must be the basic character followed by added character.

(2) Control of basic character

Basic characters have the attribute basic-line segment, which shows the width of the character, and which determines the horizontal printing position of the letter relative to the preceding letter like the kerning information in the Latin alphabet. By employing this scheme, the width of letters can be controlled closely, and printing with proportional spacing is possible.

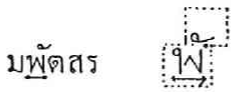
The under line with the arrows in the following figure shows the basic-line segment.

ผาลาเพียงไหล่

(3) Control of added character

An added character has a basic point together with information on its setting position, which serve to locate the character relative to the basic line of the preceding basic character. The printing position for a Thai letter with an added character is thus determined by the attribute setting position. This control is the same as the dead key control of a typewriter.

A character on the upside surrounded by the dotted line in the following figure is an added character. A big dotted mark on the dotted line shows the setting position of added character.



(4) Parent and child patterns and the line position control

Satisfactory printing quality of consonants can generally be achieved if a character is represented by 40*40 dots. However, similar consonants require 80*40 dots. Therefore, the string of Thai letters into three levels has been split, and divided the consonants and vowels represented by 80*40 dots into two partial patterns, here called parental and child patterns. The printing position for each pattern in a Thai letter is decided by the attribute level position, which shows the level of the parental and child patterns.

The central level in the following figure corresponds to the base line, namely, which is (1), (2), and (3) in Fig.6.2. Each pattern is parental, and the levels above and below the central one consist of the child patterns.

4-220-04 กฎ/ให้/ไว้/ณ/วันพุท/เดือน/สาม/แรม/สิบเบด/ค่ำ/

(5) Overlap Control

Every character pattern is synthesized by an OR operation for all dots. This scheme is necessary for such characters as ปี and ฟ้ like the ligature information in the Latin alphabet.

(6) Vertical control of added characters

To improve printing quality, four tonal marks in the vertical position need to be repositioned if the previous letter has a vowel like ◌ั, ◌ิ , or ◌ี . In this case, the setting position of the tonal mark is shifted vertically upward by an appropriate number of dots, and the tonal mark is printed overlapping the preceding vowel.

An algorithm and context-sensitive optimization table concerned with the above schemes are given in the following paragraphs.

Chapter 6 Thai Printing System

Printing Strategy

The printing system is made up of a font generator with font code table and formatter process as shown in Fig.6.3.

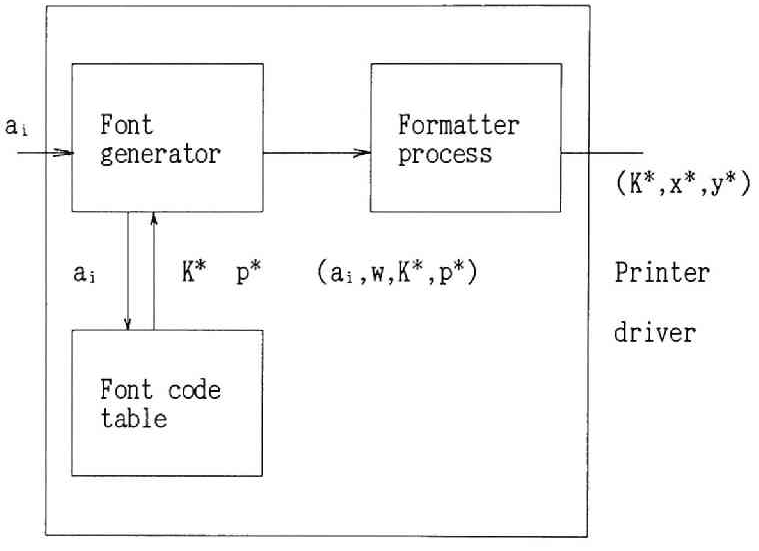


Fig.6.3 Structure of Thai printing system

Here, let's define a symbol to be printed as a_i. To print the symbol on the printer, a Thai character consisting of 3 partial patterns composed of parent and child patterns has been designed as shown in Fig.6.4. The procedure for making each font is described in Section 6.5. The level of each partial pattern discriminated by p, for example, central, upper, lower levels, have values 0,1, and 2 for p respectively.

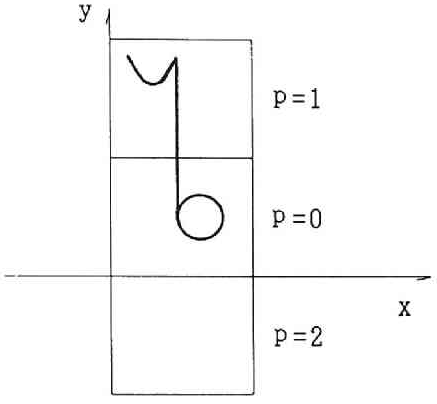


Fig.6.4 Structure of Thai Character

The partial patterns corresponding to the symbol are defined as the font code sequence K^*, and each pattern K_i indicates an element of the font code table. Hence, each K_i belongs to one of any of the levels denoted by p. The font code table is generally composed of items given as follows:

Input symbol	Width	Level:p 0	1	2
a_i	w	K_k	K_l	K_m

The i ranges from 1 to 80. The w indicates the width of the basic line for a character. Therefore, a K^* mapped by a_i is a basic character if $w > 0$ is satisfied, whereas K^* is the added character when w is equal to zero. Each K_k, K_l, and K_m which is obtained by looking up the font code table is each K_i corresponding to each p respectively. However, the conditions of k, l, and m are $k \neq l \neq m$, and the range of k, l, and m is 1 to 200.

The output of the font generator is defined as a pair (a_i, w, K^*, p^*), where a_i, w , K^*, and p^* are input symbol, width of a Thai character generated, a sequence of font codes, and a sequence of level identifier corresponding to the font codes respectively.

In the formatter process, a pair (a_i, w, K^*, p^*) as the input from the font generator is formatted and optimized as the printing image, and a sequence of pattern codes and the values of co-ordinates corresponding to one line are given to the printing driver.

The formatter process has a table, which is composed of a flag for vowel, the alternatives of origin of setting position, difference of setting position to X-axis direction and difference of setting position to Y-axis direction denoted by q,d,h, and v respectively corresponding to each entry of added characters, as follows:

	Output			
Added char.	Flag of vowel	Alternatives of origins	Difference X-direction	Y-direction
a_i	q	d	h	v

where q and d have two kinds of values; 0 and 1. For example, if a_i is equal to v_2 (vowel), then each q, d, h, and v is set to 1, 0, 5, and 0 according to the table, and value q would be checked to control the difference of Y-direction if the following symbol were a tonal mark. If a_i is equal to t_1 (tonal mark), then d is set to 1

in order to avoid an overlap among the two patterns like '◌ุ' and '◌ุ้' (tonal mark cannot be identified), namely, the origin of the pattern for t_1 is decided by which the most right side of the previous symbol is to be the basis for positioning the pattern.

Therefore, the origin can be obtained by subtracting the width of the pattern from the position of the most right side. Also, the pattern of t_1 should be shifted by appropriate dots towards the Y-direction if a previous q is equal to 1, which indicates that the previous symbol was a vowel like v_2, v_7, v_8, v_{13}, or v_{14}. Thus, the table in the formatter process is context-sensitive, and the printing control is optimized using this table as above. The printing adjustment for tonal marks and the special symbol s_1 is controlled by the algorithm as shown in Fig.6.5.

The output of procedure T in Fig.6.5 is the value in the context-sensitive optimization table as above. The he[max] and gc[max] imply the number of dots for a line and the size of the font generation table respectively. Also, vs(lc) indicates the real vertical position assigned by lc, and vd(p) gives a value out of 0, 30, or -30 dots for the difference in the Y-direction corresponding to p.

```
F:procedure ( a, K, x, y );
    array K[], x[], y[];
    array vs[], vd[];
    dcl ( hs init(0), he init(0), lc init(1),
          q  init(0), f init([false]) ) static;
    dcl ( sp init(1), ep init(0) ) static;

    if w not=  0 then do;
      hs=he;
      he=hs + w;
      if he > he[max] then do;
        output line;
        lc=lc+1;
        sp=ep+1;
        hs=0;
        he=w;
      end;
    end;

    gc=0;
    do while( gc not=  gc[max] and f=[false] );
      gc=gc+1;
      call gc -> T( ai, q, d, h, v );
    end;
    ep=ep+1;
    if d=0 then ep->x=hs+h;
           else ep->x=he-fsize+h;
    if pv=1 then ep->y=vs(lc)+vd(p)+v; pv=0;
            else ep->y=vs(lc)+vd(p);
    if q=1 then pv=1;

    T:procedure ( ai, q, d, h, v );
        if ai=g then do;
          d=dv; h=hv; v=vv; q=qv;
          f=[true];
        end;
        return;
    end T;
  end F;
```

Fig.6.5 Algorithm of Printing Adjustment based on a Context-Sensitive Table

Notation which is used in Fig.6.5 is as follows:

w : Width of a font pattern

hs : Starting position of current pattern which is to be printed on the horizontal line.

he : Ending position of current pattern which is deduced by adding the width of pattern w into the hs.

lc : Line counter which is used as the suffix of vs for deciding an absolute address in the vertical position.

fsize : Font size; it is 30 when 30x30 dots pattern is used.

gc : Control variable for table searching.

pv : Context-sensitive parameter which is decided by the value of previous q.

K : Font pattern code.

x : Physical number of x-distance in terms of dots for K.

y : Physical number of y-distance in terms of dots for K.

The notation above is common to the justification algorithm in the following section.

6.4 Justification Algorithm

The printing goal of this justification algorithm is to produce Thai printings with well-formed characters to satisfy the reader's calligraphic preferences. In the algorithm of the preceding section, the syllable across the right margin is sometimes cut at a position before the grapheme vowel, and the remaining letters that its top letter is a vowel are put into new line as shown in Fig.6.6.

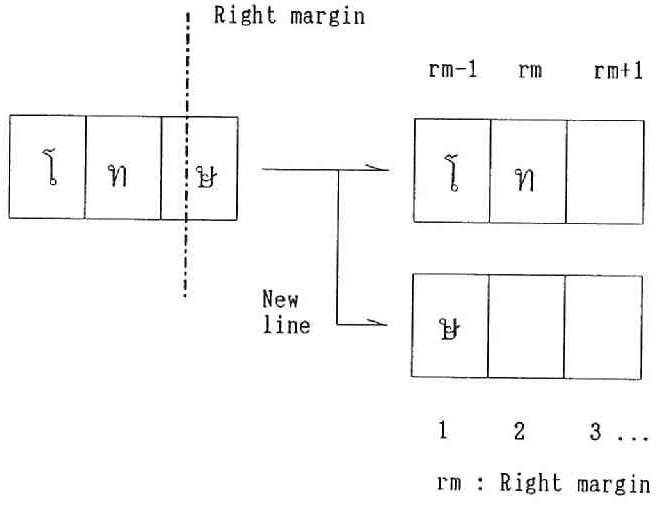


Fig.6.6 Control at right margin

Cases as shown in Fig.6.6. will happen when the following conditions occur;

[Case 1] When the following letter after the vowel v_2 encounters the right margin.

[Case 2] When a vowel such as v_3, v_5, or v_6 is across the right margin.

[Case 3] When the one preceding letter laid on the right margin is a vowel such as v_{16}, v_{17}, v_{18}, v_{19}, and v_{20}.
[Case 4] When the current letter encounters the right margin.

To justify the output at the right margin in general, the number of dots of difference that is the same as the portion from the left most side of the current letter until the right margin are distributed uniformly and the divided number is added into each value of the x co-ordinates from the first to the preceding letters on the same line if the current letter extends into the right margin.

The current symbol is defined as a_c, the preceding symbol as a_p, the starting position of the current symbol as hs, and the value of right margin as rm.

The justification algorithm is as follows:

(1) If $a_p \in$ Vp: Vp={v_{16}, v_{17}, v_{18}, v_{19}, v_{20}} and $a_c \in$ CS then

```
adjust=2;
ep=ep - 1;
wd=( rm + 1 ) - cp->x;
```

(2) If $a_c \in$ Va: Va={v_3,v_5,v_6} then

```
adjust=2;
ep=ep - 1;
wd=( rm + 1 ) - cp->x;
```

(3) If $a_p=v_2$ and $a_c \in$ CS then

```
adjust=2;
ep=ep-2;
wd=( rm + 1 ) - cp->x;
```

(4) Except conditions above

```
adjust=1;
wd=( rm + 1 ) - hs;
```

where adjust means the type of equation wd, namely, the cases (1)-(3) are the same types, wd means the number of dots to be adjusted, and ep indicates the last letter pointer of the current line to be printed, namely, to be passed into the driver. Also, the current pointer cp points to the preceding letter on the ring buffer.

The circumstances of a typical case are shown in Fig.6.7. Figure 6.8 shows the justification process for the difference wd.

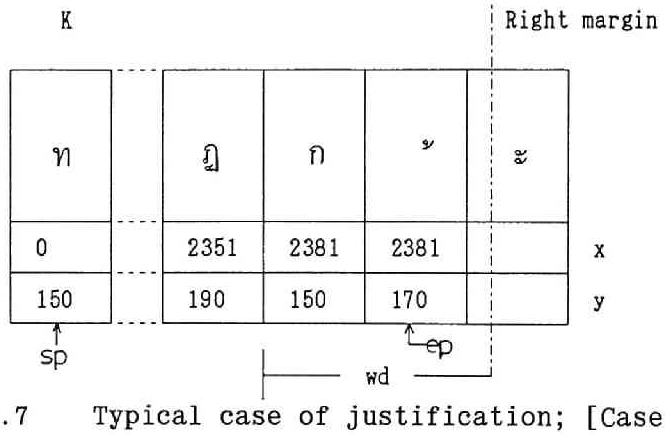


Fig.6.7 Typical case of justification; [Case 1]

```
ADJUST:procedure(sp,ep,adjust,x);
       array x[];
         ip=sp+2;
         if adjust=1 then wd=(rm+1) - hs;
                     else wd=(rm+1) - cp->x;
         do j=wd to 0;
           ip->x=ip->x+1;
           ip=ip+2;
           if ip > ep-1 then ip=sp+1;
         end;
end ADJUST;
```

Fig.6.8 Justification process

Also, sp indicates the starting pointer of the current line on the ring buffer, and ip is the working pointer in the process. The variable adjust, sp, and ep and array x are the parameters given by the calling process in Fig.6.5. For all of the basic characters directed from the value of the sp pointer until the ep pointer, the number of the difference wd is distributed by adding a value of one into x directed by pointer ip for every two characters. The reason for value two for the ip pointer is derived from the fact that almost all cases of word formation consist of [C + V]. The variable adjust is determined by the justification algorithm.

This justification control can be executed if this process is inserted at a position above "output line" statement in Fig.6.5.

6.5 Using the System

Figure 6.9 shows an output of the traced interactive process on the intelligent Thai computer terminal using the NEC Spin-writer ELF360 printer as an example of the output for retrieving the KTSD records under the information retrieval system FAIRS that runs on the host computer, FACOM M-780 [Shibayama 90].

The configuration of the developed system is shown in Fig.3.9 of Chapter 3. Incorporating the control schemes (1), (2), and (3) described in the previous Section 6.3 into the terminal design, the content of records with Thai internal code for the interactive process on the terminal can be output to an output device like the

ELF360 printer which can be printed by the impact of the font wheel called the thimble. Thus Thai letters in addition to the ordinary dot-matrix and page printers connected to the personal printer can be printed.

After a symbol "#", on 10 lines from the top in Fig.6.9, a following string IRS is a computer command for invoking the information retrieval system FAIRS on the host computer, and KTSD is the parameter for specifying the database KTSD. The following two messages show that the database KTSD is invoked, and after replying "N", which means that the Roman spelling, namely, the TM, is to be used for specifying the retrieval item in the search statement, the search command "SEA", used in order to retrieve the statement which contains "KT1", say, the word in Fig.6.9(b) from the database KTSD, is typed. The result of the output is the first one of fifteen statements.

For the goal of implementing a printing system to produce Thai documents with a character set adequately large and well-formed to satisfy readers calligraphic preferences in the practical work of making a computer concordance for the KTSD, the Thai printing system in which the portions of vowel and tonal marks can be controlled based on the contextual constraints has been newly devised and implemented, whereas the portions of all graphemes in the ordinary system [Sakamoto 79] cannot be changed.

Figure 6.10 shows one of the pages in Datchani Kotmai Tra Sam

Duang [Ishii et al. 90]. The word in brackets in the Figure shows the entry word which is followed by one or more sentences. Each sentence is prefixed by the identifier of location composed of the number of volume, page, and line in the text of the KTSD.

Thai character fonts which must be referred to at the location of the printer driver in Fig.6.4 and which are run on the host computer and its connected laser beam printer, called NLP (Nihongo Line Printer), have been provided as follows:

(1) Each one, which is denoted by K_i, in three partial patterns of Thai character has been registered to the extra font pattern library as an extra font of a Japanese character with Kanji (2 bytes) code defined by the user depending on the rules in JEF (Japanese processing Extended Feature) sub-system on the host computer.

(2) All of Thai character fonts (195 partial patterns) have been pre-registered using the ADJUST utility of the JEF sub-system as shown in Fig.6.11, and each pattern has been made in an arbitrary manner by the author.

(3) Dataset organization of the font pattern library is the VSAM (Virtual Storage Access Method) type, and a record key corresponds to a Kanji code, namely, a partial pattern of Thai.

(4) The Thai font pattern library in the database KTSD provides two pattern sizes; 30*30 and 40*40 dots for every partial pattern (See Fig.6.11).

```
+ KYOTO-UNIV  TSS SERVICE     --T2681134--
    WAITING JOBS( 0) , EXECUTE JOBS( 0) , SYSOUT JOBS( 4)
    *ACTIVE TSS USERS (141)  *TERMINAL-ID (E962 )
    *ESTIMATED PAYMENT          150,000 YEN
    *TOTAL APPROXIMATE CHARGE   117,385 YEN (SINCE 89.04.01)
KDS40613I THE USER'S LAST ACCESS DATE(1989.09.25),TIME(15:27:09).
KEQ56455I X52661 LOGON IN PROGRESS AT 15:36:17 ON SEPTEMBER 25, 1989
**   T2681134 52661 : (LOGON ACCEPTED) ** CN(01)
KEQ56951I NO BROADCAST MESSAGES

# IRS  KTSD

* THE LAW OF THREE SEALS (KOTMAI TRA SAM DUANG) *
* THAI IRS BEGINS *

*** SELECT TERMINAL TYPE ***
* THAI OR NORMAL ? (T/N) * : N

THAI-IRS: SEA ST EQ KT1
15 DOCUMENTS FOUND

THAI-IRS: OUT

*SELECT OUTPUTING DEVICE*
TSS TERMINAL OR NLP? (T/N): T

*DEVICE <T> IS SELECTED*

NO.00001 ----------------------------------------------------------------------
RNO  : KTSD00009022

VPL  : 311201311203

ST   : จี่/พระเจ้าอยู่หัว/ตรัส/ให้/พระสุภาวะดีศรีมลราดุลราช/ แล/นายท้าวราชบันท
ึค/นายสามขลา/เสมียน/กฎ/คำนับ/ไว้


VOL  : 3

PAGE : 112

LINE : 01

NO.00002 ----------------------------------------------------------------------
RNO  : KTSD00009629

VPL  : 316114316116

ST   : สมเดจบรมบพิตรพระเจ้าอยู่หัว/ มี/พระบันทูลสุรสิหนาท/คำหรัศตรัสแก่/ขุนศรี
ภูริปรีชา/ /ให้/กฎ/เปน/ตำรา/ไว้/ดั่นี้


VOL  : 3

PAGE : 161

LINE : 14
```

Fig.6.9 An example of retrieval of KTSD

กชิงหุ้มผ้าขาวเลว

000011 กชิงหุ้มผ้าขาวเลว
1-157-04 กชิงหุ้มผ้าแดงได้แต่หลานหลวงอยู่ในวัง อยู่นอกวัง/กชิงหุ้มผ้าขาวเลว/
000012 กชิงหุ้มผ้าแดง
1-157-03 /กชิงหุ้มผ้าแดง/ได้แต่หลานหลวงอยู่ในวัง อยู่นอกวังกชิงหุ้มผ้าขาวเลว
000013 กฎ
1-155-05 ส่งนายแวงหน้า ๒แวงหลัง ๒คำรวจในถือ /กฎ/สั่งเรือในพิเนศแลเรือขุนคาบแห่หน้าเรือชาววังตามหลัง
1-155-07 จึ่งนายแวงคำรวจในลงเรือหน้าแลเอา/กฎ/ไปแก่เจ้าเมืองแลกรมการ
1-156-11 เรือน๓ ห้อง ฝากระดานตรึงเหลก ขุดมุมใต้ท้องเรือนฦก๕ ศอกมีกระดานปกบนลันกุนแจตาม/กฎ/
2-023-01 ถ้าแลเจ้าหมู่สมุบาญชียมิได้ทำตาม/กฎ/ซึ่งทรงพระกรรุณาโปรดเกล้าโปรดกระหม่อมนี้ต่อกรมสัศดีจะเอาเลกนั้นจ่ายแลเจ้าหมู่จึ่งมาจำหน่ายเลกนั้น
2-023-06 ถ้าเลกหมู่นั้นกรมนั้นแลเจ้าหมู่สมุบาญชียได้ทำดุจหนึ่ง/กฎ/นี้แล้ว
2-044-17 มิได้มาตามนัด๓ นัดให้/กฎ/เอาเนื้อความเปนแพ้ ๒
2-045-01 อนึ่งถ้าขาดผัดส่งช้างม้าผู้คนถึง๓ ผัดจึ่งให้/กฎ/เอาคดีเปนแพ้ ๓
2-078-02 ท่านว่าให้เอาคำพญาณซึ่งระบุะอ้างค่อไปนั้น/กฎ/เอาไว้ แล้วให้บังคับบันชา
2-079-02 ๑๐ อนึ่งผู้มีอรรถคดีอ้างพญาณๆ ให้การแลแก้ค้านเข้าด้วยฝ่าย(โจท,จำเลย) ไช้ชื่อพญาณอาษาท่านบให้พึงฟังให้/กฎ/เอาคดีนั้นเปนแพ้
2-084-01 ท่านว่าอย่าให้มันพึงพิสูทด้วยพญานท่านให้/กฎ/เอาคดีมันเปนแพ้บพึงให้เผชิญสืบไป
2-090-02 ควรให้ตระลาการ/กฎ/เอาคู่ความผู้อ้างนั้นเปนแพ้คดี
2-090-08 มีผู้รู้เห็นเปนแม่นมั่นไช้ควรให้/กฎ/เอาคดีผู้นั้นเปนแพ้
2-090-12 มันไปสู่หาเจรจาด้วยพญาณแลมันซักซ้อมพญาณจับได้เปนสัจ ให้/กฎ/เอาคดีมันนั้นเปนแพ้
2-090-14 ถ้าจับหมีได้ข้างหนึ่งหมีค้านพญาณพญาณหากรับค้านอย่าให้บังคับบันชา/กฎ/เอาคดีมันเปนแพ้ก่อนท่านให้พิจารณาแต่ที่อันจริง
2-091-02 ให้/กฎ/เอาคดีมันเปนแพ้แก่สำนวน
2-091-15 ให้/กฎ/เอาคดีมันเปนแพ้
2-092-09 ผู้คู่ความก็ดีว่ามีผู้รู้เห็นผู้อ้างกล่าวข้อเนื้อความชี้แจงออกข้อความปรากฎให้พญาณฟังท่านว่าให้/กฎ/เอาคดีเปนแพ้
2-092-13 ท่านว่าอาวุธปากมันเองมันจนแก่พญาณ ให้/กฎ/เอาคดีมันเปนแพ้
2-093-02 ท่านให้/กฎ/เอาคดีมันเปนแพ้
2-093-09 เมื่อแลลักขณตระลาการเผชิญอยู่นั้น ให้/กฎ/เอาคดีเปนแพ้แก่ผู้หมีได้ดิดใจ
2-093-18 ให้/กฎ/เอาคดีมันเปนแพ้
2-097-16 ถ้าเปนสัจจริงไช้ให้/กฎ/เอาคดีผู้ทำกลธิบายนั้นเปนแพ้
2-098-08 พิจารณาเปนสัจดุจกล่าวมาไช้ให้/กฎ/เอาคดีมันเปนแพ้แก่สำนวน
2-101-05 ท่านให้/กฎ/เอาคดีมันเปนแพ้
2-102-18 ถ้าพญาณสาบาลให้การว่า ฝ่ายข้างหนึ่งไปซักซ้อมพญาณเปนสัจไช้ให้/กฎ/เอาคดีมันเปนแพ้
2-154-16 ท่านให้/กฎ/เอาความเดิมเปนแพ้แล้วให้เอาอ้ายผู้ร้ายนั้นขึ้นขาหย่าง แลปลงลงทวนด้วยลวดหนัง ๕๐ ที
2-196-02 ให้คืนฟ้องให้แก่มันแลส่งตัวมันไปยังกระลาการเก่าให้/กฎ/เอาเนื้อความเดิมเปนแพ้แก่กัน
2-197-16 แลผู้รับฟ้องนั้นให้ไหมตามบันดาศักดิแล้วให้ส่งตัวผู้ฟ้องไปแก่กระลาการเก่าให้/กฎ/เอาความนั้นเปนแพ้
2-198-12 ๑๓ อนึ่งมี/กฎ/ให้ไว้ว่า ราษฎรร้องฟ้องหาความแก่กันด้วยเนื้อความสิ่งใดๆ
2-199-17 ถ้าปรับมาเปนประการใดให้กระลาการทำตาม/กฎ/
2-199-18 /กฎ/ให้ไว้ณวันพุธเดือน ๘ ขึ้น ๑๐ ค่ำปีเถาะตรีนิศก
2-200-02 ๑๔/กฎ/ให้ไว้แก่กระลาการทังปวงว่า ราษฎรผู้มีคดีเหนว่าเนื้อความคนเพลียกพลำ
2-200-16 ๑๕/กฎ/ให้ไว้แก่กระลาการทังปวงว่า
2-201-07 ๑๖/กฎ/ให้ไว้แก่กระลาการทังปวงว่า
2-319-13 ให้นายยุกระบัตร/กฎ/เอาคำไว้แลคนนั้นรับว่าเปนทายเปนไพร่จริง
4-164-03 ๑/กฎ/ให้ไว้แก่ข้าทูลอองทุลีพระบาทผู้ใหญ่ผู้น้อยฝ่ายทหารพลเรือน
4-169-11 จะเอาตัวผู้มิได้กระทำตาม/กฎ/แลญาติโยมพระสงฆเถรเณรรูปนั้นเปนโทษตามโทษาณุโทษ
4-169-14 /กฎ/ให้ไว้ณะวันเสาเดือนสิบขึ้นสิบห้าค่ำ จุลศักราชพันร้อยสี่สิบสี่ ปีขาลจัตวาศก

Fig.6.10 An example of output of printing system

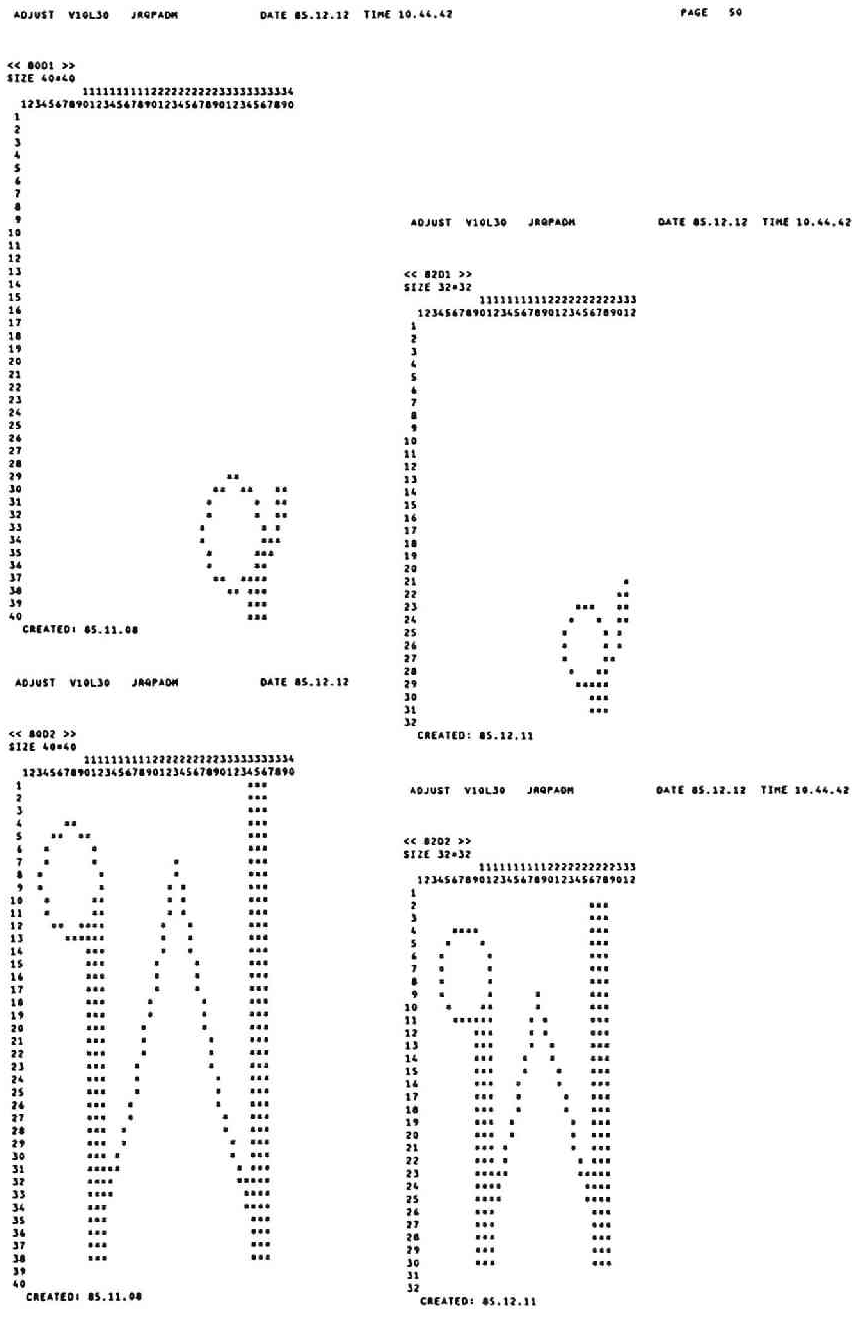


Fig.6.11 Thai character font in the pattern library

Chapter 7 Conclusions

7.1 Summary

Input/output methods for Thai present one of the complicated problems in the mechanical processing of the Thai language. The number of character patterns of Thai is 80, it is not so difficult to treat the input/output of Thai letters which are fewer in number than characters of Japanese, Korean, and Chinese, but they are greater in number than that by Roman alphabet and numerals.

However, such characteristics of Thai as a large difference in shape or size among letters, differences among the phonemes embedded in the syllabic structure and the orthographic form in the writing system, and printing complexity depending on each grapheme cause the input/output system to be much more sophisticated.

Simultaneously, Thai input/output cannot use the general devices specialized for Latin characters except by way of transcription. A transcription method from the viewpoint of natural language processing with emphasis on the man-machine interface would be not suitable in the system for treating full text processing, natural language understanding, and/or machine translation.

The next problem in mechanical processing of Thai is language

processing of recognized portions of words on the morphological level. The Thai language differs from western languages, which is characterized by being unsegmental, that is, written with no spaces between words in the closed nature as Japanese is unsegmental. The characteristic increases the complexity in mechanical processing of the Thai language. Studies on segmentation in the unsegmental languages, in general, have been carried out using grammatical structure rules and algorithms derived from heuristics depending on the individual language along with necessity of using a dictionary.

The data structure and access method of the dictionary, thus, are also the major factors to reduce the frequency of occurrence of reference to the dictionary and to improve the efficiency of analysis on the morphological level.

This thesis has discussed the diverse problems with the input/output methods mentioned above and morphological analysis for Thai, and modeled the syllabic structure and syllable formation rules based on heuristic knowledge derived from Thai language analysis. The algorithms of transliterating Roman to Thai script, recognizing syllables, and printing Thai letters which are implemented in an intelligent Thai computer terminal, a Thai syllable recognizer, and a Thai printing system respectively have been proposed. The characteristic in the evaluations of the experiments, also, is that all of the experiments have been performed with a large text and dictionaries in this thesis, and it ensures that the results of experiments are effective in practice.

Chapter 7 Conclusions

The major results are as follows:

(1) The intelligent terminal that has been designed has the facility for inputting Thai text by the Direct Mapping Method (DMM) and Transliteration Method (TM). The algorithm of the Transliteration Method which employs the function of automatic and consecutive conversion from Roman spelling to Thai letters has been presented in Chapter 3. This method has the advantage of requiring fewer keys on the keyboard than that of the Direct Mapping Method. This advantage is that the ordinary English (Qwerty) keyboard assignment can be used for the Transliteration Method if the transliteration from Roman spelling to Thai letters is incorporated as a function in the computer. This feature has been equipped and utilized under the information retrieval system of the KTSD [Shibayama 90] in Data Processing Center, Kyoto University.

(2) In addition to the Transliteration Method and Direct Mapping Method, the Simplified Transliteration Method has been designed. The method provides a function so that the Thai letters with respect to the arbitrary Roman spelling will appear one by one cyclically when a typist selects that Roman spelling. Also, the characteristics of these input methods have been compared by measuring the speed of typing and the learning curves in the actual input work of making a machine readable dictionary for Thai. Although the number of key strokes required by the text input is normally larger than by the Transliteration Method, the results of measurement and

evaluation of speed of typing indicated that this method is more readily operable by non-native speakers of Thai accustomed to input the Roman spelling. This Simplified Transliteration Method has been applied for a Thai text editor [Shibayama 87].

(3) The ordinary longest-match method of segmentation for inputting 20,631 sentences of the KTSD has been analyzed. Its analysis is based on the dictionary consisting of 20,475 main entries of the KTSD. Then the Syllable Longest-Match method (SLM) based on the analysis derived from the results of the preceding experiment has been devised, which employs the function of back-tracking for each phonemic unit based on the syllable formation rules. It is found that the back-tracking for one character, namely, for a grapheme is most effective by 98.0% in the ratio of segmentation as described in Section 5.3.4 of Chapter 5.

(4) Finite automaton model, called the Thai syllable recognizer, for segmenting a sentence into monosyllables only has been proposed in Chapter 5. The major characteristic of the recognizer is that the sentences can be segmented without reference to a dictionary. From an experiment for the input of the KTSD text, it is found that a ratio of segmentation by the recognizer gave at most 49.6%. It was based on syllable formation rules only as described in Section 5.5.

(5) By adapting the knowledge rules depending on heuristics derived from the analysis of unsuccessful cases into existing sylla-

ble formation rules, it was found that a ratio of segmentation of 93.9% was obtained. And it is indicated that the use of knowledge, which is derived from the heuristics depending on the properties of the individual language, plays a major role in the language analysis as described in Section 5.5.4. The characteristic of this manner in the recognizer is that no dictionary is required.

(6) Thai printing system with high quality letters and well-formed to satisfy a user's calligraphic preference has been designed and implemented. Its printing strategy depends on the algorithms for controlling the divided character patterns and for adapting the syllable formation rules to the mechanism of right justification. Also, a context-sensitive optimization table used for implementing the printing strategy has been proposed. The system described in the preceding chapter, in conjunction with the availability of multiple fonts in the arbitrary font printing system, enables the production of complex documents like Kotmai Tra Sam Duang by substituting the fonts and tables.

7.2 Remaining Subjects

This thesis has introduced the basic input/output system for Thai based on the input/output and morphological levels. The intention was to develop a practical Thai language processing system with emphasis on natural language processing, especially, on the basis of the linguistic approach. However, to allow the users to facilitate Thai language processing, an improvement of the hardware

environment in the input/output devices and studies on all levels of Thai language processing are required.

Remaining problems from this thesis are as follows:

(1) As for the measurement and the estimation of the typing speed and the learning effect in the both DMM and TM methods, the measurement by the some typists and the extension of experience time for the input work should be performed in order to clarify the characteristics of entering the Thai text in general.

(2) To discriminate between different letters with the same phonemic combination, and to overcome the differences among the syllabic structure and the orthographic forms, and modeling and implementation by means of a kind of Roman-syllable translation based on the dictionary database should be devised like Roman-Kanji translation in Japanese.

(3) In the segmentation based on the syllable formation rules and its syllabic symbols, it is necessary to increase the accuracy and consistency of segmentation by using a dictionary along with heuristic knowledge. It means that the elimination of monosyllables with no meaning and the formation of words by automatically synthesizing monosyllables can be accomplished.

(4) And, the features of dynamic self-learning should be equipped for the analytical process of segmentation to improve the ratio of segmentation.

(5) Studies on the syntactic and semantic analyses in Thai language processing such as an implementation of Thai GPSG translator [Vorasucha 87] and Thai Syntax Analysis based on the case frame [Shibayama 89] are most important studies in the field of natural language understanding and machine translation, and they are the subjects for future study.

Acknowledgements

The author would like to express his sincere appreciation to Professor Satoshi HOSHINO of the Data Processing Center, Kyoto University who always gave him the guidance and the encouragement since he was a staff member of the Data Processing Center, Kyoto University. Most of the work during the course of this study has been inspired and supported by him.

The author would like to express his great gratitude to Professor Yoneo ISHII, the former Director of The Center for Southeast Asian Studies, Kyoto University who led him to the field of Thai language processing first, and gave him continuous support and various criticism.

Thanks are due to Professor Shigeharu SUGITA of the National Museum of Ethnology, who made many valuable comments and provided the original text of the Three Seals Law, and Professor Aroonrut Wichienkeeo of Chiang Mai Teacher's College, who contributed to the proofreading of segmentation of the Three Seals Law and gave him instruction in the Thai language. Thanks are also due to Dr. Patamawadee Pochanukul of NIDA, Thailand who contributed to valuable discussions on Thai grammar.

Lastly, the author would like to thank Vice president Professor Shinichi ICHIMURA of Osaka International University for his

Acknowledgments

guidance and encouragement during this thesis. Although the author cannot list all names, he would like to thank all members and colleagues of the Faculty of Management and Information, Osaka International University, and the Data Processing Center, Kyoto University.

Bibliography

[Allison 73] Allison, G.H. :"Simplified Thai", Nibondh, Thailand, pp.1-76, 1973

[Allison 75] Allison, G.H.:"Modern Thai", Nibondh, Thailand, pp.219-222, 1975

[Chomyszyn 86] Chomyszyn, J.:" A phonemic transcription program for Polish", Int. J. Man-Mach. Stud., Vol.25, pp.271-293, 1986

[Diller 79] Diller, A. :"Problems in Thai cataloging in Western Libraries, Proc. of workshop on problems in Thai Cataloging 20th Biennial Conference, Library Association of Australia, pp.39-46, 1979

[Griffin 81] Griffin, C. :"Typesetting Exotic Language at Oxford university", TEXTPRO I, pp.133-144, 1981

[Haas 56] Haas, M. R. :"Thai system of writing", American Council of Learned Societies, Washington, D.C., pp.1-40, 1956

[Hartmann & Henry 83a] Hartmann, J.F and Henry, G.M.:"Thai Script Computer-converted from a Precise, Pronounceable Transliteration for Bibliographic Management and Manipulation", Bulletin of CORMOSEA, Vol.11, No.2 ,pp.5-10, 1983

[Hartmann & Henry 83b] Hartmann,J.F. and Henry, G.M.:"Transliteration Table for Computer-converted Thai Script", Bulletin of CORMOSEA, Vol.11, No.2, pp.11-16, 1983

Bibliography

[Hee 86] Hee Sung Chung: "JUNG-UM : A New Keyboard Arrangement for Korean Alphabet", IPSJ SIG Reports Japanese Document Processing 7-1, pp.1-8, 1986

[Horiike & Hoshino 90] Horiike, H. and Hoshino, S. :"On software tools for making and utilizing Japanese text database" (in Japanese), Informatics Symposium 1990 of JIPS, pp.153-160, 1990

[Ishii 69] Ishii, Y. : "Introductory Remarks on the Law of Three seals" (in Japanese), Southeast Asian Studies, Vol.6, No.4, pp.155-178, 1969

[Ishii 87] Ishii, Y.:" Computerization of the Thammasat Version of the Kotmai Tra Sam Duang", Studies on the Multi-Lingual text Processing for Assisting Southeast Asian Studies, The Center for Southeast Asian Studies of Kyoto University, Report by grants for scientific research from Japanese Ministry of Education, pp.46-50, 1988

[Ishii et al. 90] Ishii, Y., Shibayama, M., and Aroonrut, W.: "Datchani Kotmai Tra Sam Duang" (in Thai, Computer Concordance to The Three Seals Law), Amarin Publication, 5 volumes, Thailand, 4850 pages, 1990

[Kawabe 80] Kawabe,T.:"Thai Fundamentals" (in Japanese), Daigaku Syorin, pp.5-12, 1980

[Knuth 79] Donald E. Knuth: "T_EX and METAFONT", American Mathematical Society and Digital Press, pp.1-45, 1979

[Khrusapha 62] Khrusapha :"Kotmai Tra Sam Duang",(in Thai), Thammsat Univ., 5 vol.s, 1775 pages, 1962

Bibliography

[Murayama 82] Murayama, N. : "2-strokes Method for Japanese Text Input" (in Japanese), J. IPS Japan, Vol.23, No.6, pp.552-558, 1982

[Makino 82] Makino,H.:"A Japanese Input Method by Kana-Kanji Translation" (in Japanese),J. IPS Japan, Vol.23, No.6, pp.529-535, 1982

[Morita 87] Moriya,M.:"Quantitative Compositions on Performances of Various Japanese Text Input Systems" (in Japanese), JSEIC, Vol.J70-D, No.11, pp.2182-2190, 1987

[Nagao 84] Nagao,M.(ed):"Mechanical Processing of Natural Language" (in Japanese), [Gengo no Kikai Syori] Sansei-do, pp.4-8, 1984

[Nagao et al. 78] Nagao, M., Tsujii, J., Yamada, A. and Tatebe, S.: "Data Structure of a Large Japanese Dictionary and Morphological Analysis by using It" (in Japanese), Trans. IPS Japan, Vol.19, No.6, pp.514-521, 1978

[Nakanishi 75] Nakanishi, A. :"Written Characters in the World", Nakanishi Pub., pp.24-55, 1975

[Nakayama & Kurosu 84a] Nakayama,T., and Kurosu, M.:"Examination of Model for Estimating the Speed of Japanese Text Input" IPSJ SIG Reports, 13-1, pp.1-10, 1984

[Nakayama & Kurosu 84b] Nakayama,T., and Kurosu, M.:"Evaluation and Design for Kana Keyboard Assignment" (in Japanese), 17-4, pp.1-8, 1984

[Photchana nukorm Thai 82] Photchana nukorm Thai:"Chabap Ratcha

bandit sathan",(in Thai), Khrungtheep, Samnakphim Askon-chaoenthat, 1982(Thai 2525)

[Sager 81] Sager, N.:"Natural Language Information Processing, Addison-Wesley, pp.1-4, 1981

[Sakamoto 79] Sakamoto,Y.:"Computer Processing of Asian and African Language" (in Japanese), Information Management (Jyoho Kanri), Vol.22, No.7, pp.519-525, 1979

[Salton 83] Salton,G. and McGill,J.M:"Introduction to Modern Information Retrieval, McGraw-Hill, pp.257-302, 1983

[Shibayama et al. 84] Shibayama,M., Sugita,S., and Ishii,Y. :"Thai text processing using a personal computer" (in Japanese), Proc. 28th Annual Convention IPS Japan(S59), pp.1249-1250, 1984

[Shibayama et al. 85] Shibayama,M., Sugita,S., and Ishii,Y. :"Input scheme for Thai letters using Roman expression" (in Japanese), Proc. of 30th Annual Convention IPS Japan, pp.923-924, 1985

[Shibayama & Hoshino 86] Shibayama,M. and Hoshino,S.:"Implementation of an intelligent Thai computer terminal, J. Inf. Process., Vol.8, No.4, pp.300-306, 1986

[Shibayama et al. 87] Shibayama,M., Hoshino,S. and Ishii,Y. : "A Comparative Study of the Characteristics of Input Methods for Thai", Proc. of the Regional Symposium on Computer Science and its Applications, NRCT-JSPS, Thailand, pp.19.1-19.18, 1987

[Shibayama 87] Shibayama,M.:"Input/Output Methods for Thai",

Bibliography

Southeast Asian Studies, Vol.25, No.2, pp.279-296, 1987

[Shibayama 88] Shibayama, M. :"Computerization of the Thammasat Version of the Kotmai Tra Sam Duang", Studies on the Multi-lingual Text Processing for Assisting Southeast Asian Studies, CSEAS of Kyoto University, Report by grants for scientific research from Japanese Ministry of Education, pp.46-50, 1988

[Shibayama 89] Shibayama, M. :"Thai Syntax Analysis using the Case Frame",OIU Journal of International Studies, pp.107-117, 1989

[Shibayama 90] Shibayama, M.: "Database of Kotmai Tra Sam Duang of Thailand", (in Japanese) Reports of 28th research seminar in Data Processing Center, Kyoto University, pp.45-55, 1990

[Shutoh & Itoh 82] Shutoh,M. and Itoh,H.:"Pen-touch Input Method" (in Japanese), J. IPS Japan, Vol.23, No.6, pp.536-542, 1982

[Sugita 80] Sugita,S.:"TEXT PROCESSING OF THAI LANGUAGE == THE THREE SEALS LAW ==", Proc. of COLING80, 1980

[Takahashi 82] Takahashi, N.:"The Current State and Future Trend of Japanese Text Input Systems"(in Japanese), J. IPS Japan, Vol 23, No.6, pp.518-528, 1982

[Tanaka 89a] Tanaka, H.:"Fundamentals of Natural Language Analyses" (in Japanese) Sangyo-Tosyo, pp.133-137, 138-139, 1989

[Tanaka & Koga 81] Tanaka, Y. and Koga, K.: "Automatic Segmentation tion of Hiragana Strings Appearing in the Japanese Sentences", (in Japanese), J. IPS Japan, Vol.22, No.3,

pp.242-247, 1981

[Thajcayapong et al. 87] Thajcayapong,P., Tepmongkorn,P., and Chok chaipruk,s.:"Thai-English Machine Translation", Proc. of the Regional Symposium on Computer Science and its Application, NRCT-JSPS,Thailand, pp.17.1-17.8, 1987

[Tujii 88] Tanaka,H. and Tujii,J. :"Natural Language Understanding" (in Japanese), OHM, pp.191-214, 1988

[Vorasucha 87] Vorasucha,V. :"Thai syntax analysis based on GPSG", Proc. of the Regional Symposium on Computer Science and its Application, NRCT-JSPS, Thailand, pp.18.1-18.24, 1987

[Vorasucha 88] Vorasucha,V. and Tanaka,H. :"Thai syntax analysis based on GPSG", Jour. of JSAI, Vol.3, pp.78-85,1988

[Watanabe 82] Watanabe,S.:" Multi-shift Type Input Method for Japanese Test" (in Japanese), J. IPS Japan, Vol.23, No.6, pp.543-551, 1982

[Warotamasikkhabit 84] Warotamasikkhabit, V.:"Problems in using the the Thai alphabet in computing",Proc. of the 1984 South-east Asia Regional Computer Conference. SEARCC 84, SEACC, pp.18/1-8, 1984

[Yada 89] Yada,K.:"Introduction to the programming of machine translation" (in Japanese), Nikkan Kogyo, pp.11-34, 1988

[Yamada 80] Yamada,H.:"A Historical study of typewriters and typing methods : from the position of planning Japanese parallels, J. Inf. Process., Vol.2, No.4, pp.175-202, 1980

Bibliography

[Yoshimura et al. 83] Yoshimura,K., Hitaka,T., and Yoshida,S. : "Morphological Analysis of Non-marked-off Japanese Sentences by the Least Bunsetsu's Number method", (in Japan) J. IPS Japan, Vol.24, No.1, pp.40-46, 1983

List of Major Publications

Ishii, Y., Shibayama, M., and Aroonrut, W.: "Datchani Kotmai Tra Sam Duang" (in Thai, The Computer Concordance to the Law of the Three Seals), Amarin Publications, Thailand, 5 volumes, 3698 pages, 1990

Kanazawa, M., Shibayama, M., and Kitagawa, H. ; "Analysis and Improvement of Efficiency for Accessing the OS catalog"(in Japanese), Trans. IPS Japan, Vol.22, No.2, 1981

Shibayama,M. and Hoshino,S. : "Implementation of an intelligent Thai computer terminal, J. Inf. Process., Vol.8, No.4, pp.300-306, 1986

Shibayama,M., Hoshino,S. and Ishii,Y.:"A Comparative Study of the Characteristics of Input Methods for Thai", Proc. of the Regional Symposium on Computer Science and its Applications,NRCT-JSPS, Thailand, pp.19.1-19.18, 1987

Shibayama, M.:"Input/Output Methods for Thai", Southeast Asian Studies, Vol.25, No.2, pp.279-296, 1987

Shibayama, M. :"Thai Syntax Analysis using the Case Frame", OIU Journal of International Studies, Vol.1, No.1, pp.107-117, 1989

List of Major Publications

Shibayama, M. and Hoshino, S.;"Automatic Segmentation for Thai sentences", J. Inf. Process., (forthcoming), 1990

Shibayama, M.:"Thai Morphological Analysis", OIU Journal of International Studies, Vol.3, No.1, (forthcoming), 1990

Watanabe, T., Horiike, H., Ozawa, Y. and Shibayama, M.;"Analysis of Response Time in Information Retrieval System using Mass Storage System",(in Japanese), Trans. IPS Japan, Vol.25, No.5, 1983

List of Technical Reports and Convention Records

Shibayama,M., Sugita,S.,and Ishii,Y.:"Thai text processing using a personal computer" (in Japanese), Proc. 28th Annual Convention IPS Japan(S59), pp.1249-1250, 1984

Shibayama, M.:"Thai Language Processing", IPSJ Kansai Branch SIG Reports, 1984

Shibayama, M.:"Implementation of TSS Terminal Emulator with a File Transfer and Full Screen Editing" (in Japanese), Information, Vol.4, No.2, 1985

Shibayama,M., Sugita,S., and Ishii,Y. :"Input Scheme for Thai Letters using Roman Expression" (in Japanese), Proc. of 30th Annual Convention IPS Japan, pp.923-924, 1985

List of Technical Reports and Convention Records

Shibayama,M. and Hoshino, S.:"Methods and Characteristics of Thai Dictionary Inputs" (in Japanese), Proc. of 33th Annual Convention IPS Japan, pp.1727-1728, 1986

Shibayama,M.:"Input/Output Methods for Thai" (in Japanese), Reports of Research and Development Division of Data Processing Center, Kyoto University, No.1, pp.89-102, 1986

Shibayama, M.:"Thai Language Processing" (in Japanese), Studies on Data Base for Oriental Studies,Reports by grants for scientific research from Japanese Ministry of Education, DPC, Kyoto University, 1987

Hoshino, S. and Shibayama, M.:"On Heuristic Use of Computers to the Studies of History and Literature" (in Japanese), Informatics Symposium IPS Japan, 1987

Shibayama, M. :"Computerization of the Thammasat Version of the Kotmai Tra Sam Duang", Studies on the Multi-Lingual Text Processing for Assisting Southeast Asian Studies, CSEAS of Kyoto University, Reports by grants for scientific research from Japanese Ministry of Education, pp.46-50, 1988

Shibayama,M. and Hoshino, S.:"A Trial of Thai Syntax Analysis based on Case Frame" (in Japanese), Reports of Research and Development Division of Data Processing Center, Kyoto University, No.3, pp.45-56, 1988

Shibayama, M.(ed.):"Studies on the Multi-Lingual Text Processing

for Assisting Southeast Asian Studies", Reports by grants for scientific research from Japanese Ministry of Education, CSEAS, Kyoto University, 1988

Shibayama, M.:"Computerization of the Thammasat Version of the Kotmai Tra Sam Duang", Studies on the Multi-Lingual Text Processing for Assisting Southeast Asian Studies, Reports by grants for scientific research from Japanese Ministry of Education, CSEAS, Kyoto University, pp.51-57, 1988

Shibayama, M.:"Development of a computer concordance to the Three Seals Law" (in Japanese), Studies on Data Base for Oriental Studies,Reports by grants for scientific research from Japanese Ministry of Education, DPC, Kyoto University, pp.57-68, 1988

Shibayama, M.:"Database of Kotmai Tra Sam Duang of Thailand", (in Japanese) Reports of 28th research seminar in Data Processing Center, Kyoto University, pp.45-55, 1990

Shibayama,M. and Hoshino, S.:"Automatic Segmentation for Thai Ancient Text" (in Japanese), IPSJ SIG Reports, 90-CH-6, 1990

List of Abbreviations in Bibliography and Publications

KTSD	: Kotmai Tra Sam Duang (The Three Seals Law)
J. IPS Japan	: Journal of Information Processing Society of Japan
Trans. IPS Japan	: Transactions of Information Processing Society of Japan
J. Inf. Process.	: Journal of Information Processing
Int. J. Man-Mach. Stud.	: International Journal of Man-Machine Studies
TM	: Transliteration Method
HTM	: Hartmann's Transliteration Method
STM	: Simplified Transliteration Method
DMM	: Direct Mapping Method
CSEAS	: The Center for Southeast Asian Studies
SLM	: Syllable Longest-Match method
JEF	: Japanese processing Extended Feature
NIDA	: National Institute of Development Administration, Thailand

Appendix

Vowels				Consonants		
					Initial and medial	Final
อะ, อั	a	อัวะ	ua	ก	k	k
อา	ā	อัว, ว	ūa	ข, ฃ, ค, ฅ, ฆ	kh	k
อำ	am	ไอ,ไอ,อัย,ไอย	ai	ง	ng	ng
อิ	i	อาย	āi	จ	čh	t
อี	ī	เอา	ao	ฉ, ช, ฌ	ch	t
อึ	ึ	อาว	āo	ญ	y	n
อื	ื	อุย	ui	ด, ฎ, ท	d	t
อุ	u	โอย	ōi	ต, ฏ	t	t
อู	ū	ออย	ōi	ถ, ฐ, ท, ฑ, ธ	th	t
เอะ, เอ็	e	เอย	ēi	น, ณ	n	n
เอ	ē	เอือย	ūai	บ	b	p
แอะ	æ	อวย	ūai	ป	p	p
แอ	ǣ	อิว	iu	ผ, ฝ, ภ	ph	p
โอะ, อ	o	เอ็ว	eo	ฟ, ฟ	f	p
โอ	ō	เอว	ēo	ม	m	m
เอาะ	ǫ	แอว	ǣo	ย	y	–
ออ	ǭ	เอียว	ieo	ร	r	n
เออะ	œ	ฤ	rึ	ล, ฬ	l	n
เออ, เอิ	œ̄	ฤ	ri	ว	w	–
เอียะ	ia	ฦ	rึ̄	ศ, ษ, ส, ห, ฮ	s	t
เอีย	īa	ฦา	rื	อ	ʻ	–
เอือะ	ึa	ฦ	lึ	ห, ฮ	h	–
เอือ	ืa	ฦา	lื			

Fig.A-1 Thai romanization table

Appendix

000121 กระลาการ/กระลายกระลอก/กระลำ/กระลำพร/กระลำพัก/กระลมพาง/กระลุมพุก/กระลุมพู/กระลูน/กระลู่น์/กระเลียด/

000122 กระเลือก/

000123 กระโลง/กระวน/กระวนกระวาย/กระวัด/กระว่า/กระวาด/กระวาน/กระวี/กระวีกระวาด/กระวูดกระวาด/กระเวน/

000124 กระเวนกระวน/กระเวยกระวาย/

000125 กระแวน/กระโวยกระวาย/กระศก/กระศัย/กระษัตริย์/กระษัตรี/กระษิร/กระเษม/กระเษมสานต์/กระเษียร/

000126 กระสง/กระสน/

000127 กระสบ/กระสม/กระสรวล/กระสร้อย/กระสวน/กระสวย/กระสอบ/กระส่าย/กระสา/กระสานติ์/กระสาบ/

000128 กระสาย/กระส่าย/กระสินธุ/

000129 กระสือ/กระสือดูด/กระสุงกระสิง/กระสุน/กระสุนปืน/กระสุนวิถี/กระสันกระสาย/กระเส่า/กระเสาะกระแสะ/

000130 กระเสียน/กระเลียร/

000131 กระเสือกกระสน/กระแส/กระแสการเงิน/กระแสความ/กระแสจิต/กระราชดำรัส/กระแสรับสั่ง/กระแสลม/กระแสสลับ/กระแสเสียง/กระแสง/

000132 กระไส/กระหง่อง/

000133 ตระหง่อง/กระหน่อง/กระหน/กระหนก/กระหนกกินรี/กระหนกนารี/บ/กระหนาบคาบเกี่ยว/กระหน่ำ/กระหมวด/

000134 กระหมอบ/กระหม่อม/

Fig.A-2 A part of main entries in the Thai machine-readable dictionary

Appendix

532501 สาตราคม/เกรียดิยศ/ศักดานุภาพ/กรุงเทพพระมหานคร/ไป/
532502 ห้าม/อย่าให้มี/เพศบุรุษ/ลึง/อันลามก/อัปประมงคล/ไว้/ใน/สาร
532503 เทพารักษ/เปนอันขาด/ทีเดียว/ แล/ให้/ผู้รักษาเมือง/ผู้รั้ง
532504 กรมการ/กำนัน/พัน/นายบ้าน/ ไป/เกบเอา/มา/เผา/ไฟ/เสีย/จง
532505 สิ้น/ อย่าให้มี/อยู่/ณะ/โรงสาร/บ้านเมือง/นิคมเขต/ประเทษ/
532506 ใด ๆ /มี/ผู้รู้เหน/เอา/เนื้อความ/มา/ว่ากล่าว/ พิจารณา/เปนสัจ/
532507 จะ/เอา/ผู้รักษาเมือง/ผู้รั้งกรมการ/แล/พัน/นายบ้าน/แล/เชา
532508 บ้าน/ตำบล/ประเทษนั้น ๆ /เปนโทษ/ถึง/สิ้นชีวิตร/
532509 แล/ให้หมายบอก/จงทั่ว/ ให้กระทำตาม/พระราชกำ
532510 หนด/กฎหมาย/จงทุกประการ/ ถ้า/ผู้ใด/มิได้/กระทำตาม/พระ
532511 ราชกำหนด/กฎหมาย/นี้/ จะ/เอาตัว/ผู้นั้น/เปนโทษ/ตาม
532512 โททานุโทษ/
532513 กฎ/ให้ไว้/ณ/วันพุท/เดือนเก้า/ขึ้นสิบสามค่ำ/ จุลศัก-
532514 ราชพันร้อยสี่สิบสี่/ ปีขาน/จัตวาศก/
532515 ๓๖
532516 ๓๖ กฎ/ให้ไว้/แก่/พระสุรัสวดี (ซ้าย/.ขวา/) ใน/นอก/
532517 ให้/กฎหมาย/บอกแก่/ข้าทูลออง ฯ/ผู้ใหญ่/ผู้น้อย/ ฝ่าย
532518 หน้า/ฝ่ายใน/ฝ่ายทหาร/พลเรือน/ เจ้าต่างกรม/ผู้รักษาเมือง/
532601 ผู้รั้งกรมการเมือง(๑/.๒/.๓/.๔/) ตวันตก/ตวันออก/ปากใต้/ฝ่าย
532602 เหนือ/ทันเดา/ขอบเขด/ขันทเสมา/ขึ้นแก่/กรุงเทพมหานคร/
532603 จงทั่ว/
532604 ด้วย/พระบาทสมเดจบรมนาถบรมพิตรพระพุทธิ
532605 เจ้าอยู่หัว/ ผู้ทรงพระราชธรรมคุณ/อันประเสริฐ/ เสดจ/
532606 ได้/ปราบดาภิเศก/ใน/เอกไอยสุริยสมบัติ/ทุกวันนี้/ มี/พระ
532607 ราชประนิธาน/ปรารถนา/พระพุทธภูมโพธิญาณ/ เสดจออก/
532608 ณะ/พระที่นั่ง/บุษบกมาลามหาจักรพรรดิพิมาน/โดย/บูรรภา
532609 ภิมุกข/ พร้อมด้วย/อุภัยสมุหะบริรักษอัคมหาเสนาธิบดี/
532610 มนตรีกระวีชาติ/ราชปะโรหิตา/โหราจารย/แจ้ง/พระบาทบง-
532611 กชมาศ/จึ่ง/มี/พระราชโองการ/มา/ณ/พระทัยพุทธสุรสีหนาทดำ
532612 หรัส/ด้วย/วิจารณญาณ/หยั่ง/เหน/ไป/ใน/พระพุทธประเวณีย/
532613 สมเดจพระพุทธิเจ้า/ทั้งปวง/ เมื่อ/กำเนิด/พระบารมี/มา/ใน/
532614 วัตสงสาร/ ย่อม/มี/พระกรุณาญาณ/เมตา/สัตวโลกย/ทั้งปวง/
532615 จะ/ใคร่/ให้ได้/โลกิยศุข/แล/พระโลกุตรศุข/เปน/มูล/พระ-
532616 บาร/มี/มา/ แม้น/เสวย/พระชาติ/เปน/บรมกระษัตร/ หา/พระ

Fig.A-3 A part of the KTSD in machine-readable form

Appendix

Input symbols	Output			
	Flag of vowel	Alternatives of origin	Difference X-axis	Y-axis
a_i	q	d	h	v
v_2	1	0	5	0
v_4	0	0	5	0
v_7	1	0	5	0
v_8	1	0	5	0
v_{10}	0	1	5	0
v_{11}	0	0	5	0
v_{13}	1	0	5	0
v_{14}	1	0	5	0
t_1	0	1	0	15
t_2	0	1	0	15
t_3	0	1	0	15
t_4	0	1	0	15
s_3	0	1	8	0

Fig.A-4 Context-sensitive optimization table for the added characters