การศึกษาการตัดคำภาษาไทยโดยการเทียบคำศัพท์จากพจนานุกรม

A STUDY OF THAI WORD SEGMENTATION USING DICTIONARY

ณัฐฉัฐ ปานตระการ

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

A STUDY OF THAI WORD SEGMENTATION USING DICTIONARY

NATTACHAT PANTRAKARN

A SPECIAL PROJECT SUBMITED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2004

หัวข้อปัญหาพิเศษ การศึกษาการตัดคำภาษาไทยโดยการเทียบคำศัพท์จากพจนานุกรม
A STUDY OF THAI WORD SEGMENTATION USING DICTIONARY

ชื่อนักศึกษา นางสาวณัฐฉัฐ ปานตระการ 44050416

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา รศ.ไพรบูลย์ พันธรักษ์พงษ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2547

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	รศ.ธีรวัฒน์ ประกอบผล	
กรรมการ	อ.วิสันต์ ตั้งวงษ์เจริญ	
กรรมการและอาจารย์ที่ปรึกษา	รศ.ไพรบูลย์ พันธรักษ์พงษ์	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปัญหาพิเศษ การศึกษาการตัดคำภาษาไทยโดยการเทียบคำศัพท์จากพจนานุกรม
ชื่อนักศึกษา นางสาวณัฐฉัฐ ปานตระการ 44050416
ปริญญา วิทยาศาสตรบัณฑิต
ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2547
อาจารย์ที่ปรึกษา รศ.ไพรบูลย์ พันธรักษ์พงษ์

บทคัดย่อ

ในประโยคภาษาไทยนั้น จะมีโครงสร้างที่เขียนส่วนประกอบของประโยคต่อเนื่องกันไป โดยไม่มีเครื่องหมายหรือช่องว่างระหว่างคำอย่างชัดเจน ในการประยุกต์ใช้งานด้านต่างๆ เช่น การแปลภาษา จำเป็นต้องรู้หน่วยคำ การแยกแยะหน่วยคำออกจากกันมีการศึกษาวิจัยอยู่ 2 ลักษณะ คือ การตรวจสอบทางอักษรวิธี และการเทียบคำศัพท์จากพจนานุกรม

ในปัญหาพิเศษนี้ จะเป็นการศึกษาขั้นตอนวิธีการตัดคำไทย ณ ตำแหน่งท้ายบรรทัด โดยการเทียบคำศัพท์จากพจนานุกรม สำหรับข้อความที่มีความยาวเกินหนึ่งบรรทัด และจะต้องปัดเพื่อขึ้นบรรทัดใหม่ โดยจะค้นหาคำศัพท์ ซึ่งอยู่ใกล้กับตำแหน่งสิ้นสุดบรรทัดมากที่สุด ในการพิจารณาตำแหน่งเพื่อการตัดคำนั้น จะแบ่งเป็น 2 ช่วง คือ ช่วงที่ 1 เป็นการค้นหาคำศัพท์ที่อยู่ก่อนตำแหน่งสิ้นสุดบรรทัด และช่วงที่ 2 เป็นการค้นหาคำศัพท์ที่อยู่หลังหรือคร่อมตำแหน่งสิ้นสุดบรรทัด ข้อดีของขั้นตอนวิธีนี้ คือ ไม่ต้องตัดคำทุกๆคำที่มีอยู่ในข้อความนั้น จากผลการทดลองตัดคำกับข้อความลักษณะต่างๆ โดยใช้พจนานุกรมที่สร้างขึ้น ให้ความถูกต้องในการตัดคำโดยเฉลี่ยร้อยละ 97

Special Project Title	A STUDY OF THAI WORD SEGMENTATION USING DICTRIONARY
Student	Miss.Nattachat Pantrakarn 44050416
Degree	Bachelor of Science
Department	Mathematics and Computer Science, Faculty of Science
Programme	Computer Science
Academic Year	2004
Special Project Advisor	Assoc.Prof.Praiboon Pantaragphong

ABSTRACT

The sentences in the Thai language have a structure where each unit is continuous without having any explicit separator in-between. Many applications such as translation, are necessary to separate each unit of the sentences. Much researcher has focused on two areas of Thai word segmentation; one is rules of characters' sequence and the other is comparing vocabularies with a dictionary.

This special project focuses on Thai word segmentation at the end of each line, using vocabularies comparison with a dictionary. This applies to the strings with length longer than one line. The part exceeding one line will be moved to the next line by finding word in the position nearest to the end of line. The Thai word segmentation algorithm has two phases. The first phase is for finding word before the end of line, and the second one is for finding word after end of line. The advantage of this algorithm does not separate every word in the string. From the result of the experiment that using many kinds of strings with specific dictionary comparison, the average of the correctness is 97 percents.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษ หัวข้อเรื่อง การศึกษาการตัดคำภาษาไทยโดยการเทียบคำศัพท์จากพจนานุกรม ได้สำเร็จลุล่วงไปด้วยดีนั้น ก็ด้วยความกรุณาของ รศ. ไพรบูลย์ พันธรักษ์พงษ์ อาจารย์ที่ปรึกษาปัญหาพิเศษฉบับนี้ ที่ช่วยชี้แนะแนวทาง ให้คำแนะนำ ติชม และเป็นที่ปรึกษา ในขั้นตอนการทำงานต่างๆมาโดยตลอด ผู้จัดทำขอขอบพระคุณอาจารย์เป็นอย่างสูงมาในโอกาสนี้

ขอขอบพระคุณบิดา มารดา ของผู้จัดทำ ที่ช่วยสนับสนุนทั้งด้านทุนทรัพย์ และกำลังใจด้วยดีเสมอมา ทั้งยังช่วยแสดงความคิดเห็น เพื่อปรับปรุงให้ผลงานดียิ่งๆขึ้นไป

ขอขอบคุณเพื่อนๆ ทั้งในและนอกสถาบัน ที่ให้ความช่วยเหลือในด้านต่างๆ ไม่ว่าในทางตรงหรือทางอ้อม และขอบคุณสำหรับกำลังใจ ที่ช่วยผลักดันให้สามารถทำงานต่างๆจนสำเร็จไปด้วยดี

ผู้จัดทำ

มีนาคม 2548

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญภาพ

สารบัญภาพ (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

เนื่องจากรูปแบบโครงสร้างของภาษาไทยนั้น มีลักษณะเฉพาะซึ่งแตกต่างกับภาษาอื่นๆ กล่าวคือ ภาษาไทยมีโครงสร้างที่เขียนติดกันต่อเนื่องไปทั้งประโยค และไม่มีเครื่องหมายวรรคตอนในขณะที่ภาษาอื่นๆ เช่น ภาษาอังกฤษ จะมีการเว้นวรรคระหว่างคำทุกคำ หรือภาษาจีน ที่ตัวอักษรแต่ละตัวนั้นแยกกันโดยสิ้นเชิง ดังนั้น ภาษาไทยจึงอาจเกิดความกำกวมได้ง่าย ด้วยเหตุที่ไม่มีการเว้นวรรคระหว่างคำนั้น หากแบ่งแยกคำผิดไป ก็อาจทำให้ความหมายของคำหรือประโยคนั้นๆผิดไปจากเดิมได้

นอกจากโครงสร้างประโยคที่เขียนต่อเนื่องกันแล้ว หลายต่อหลายครั้งที่ภาษาไทยคำหนึ่งๆ อาจมีรูปแบบการตัดคำได้มากกว่าหนึ่งแบบ ซึ่งขึ้นอยู่กับว่า บริบทหรือเนื้อความโดยรวมของบทความนั้นเป็นไปในทางใด จึงจะสามารถเลือกได้ว่า คำๆนี้ควรจะตัดอย่างไร เพื่อให้ได้ความหมายที่ตรงตามความต้องการ

ด้วยเหตุนี้เอง การตัดคำในภาษาไทยจึงมีความจำเป็นเป็นอย่างยิ่ง สำหรับการประมวลผลใดๆที่ต้องใช้ภาษาไทยร่วมอยู่ด้วย เช่น การตรวจสอบคำสะกดในภาษาไทย การแปลงข้อความให้เป็นหน่วยเสียงภาษาไทย ปัญหาพิเศษนี้จึงมุ่งศึกษาวิธีการต่างๆในการตัดคำภาษาไทย ที่สามารถตัดคำได้ถูกต้องแม่นยำและมีประสิทธิภาพมากยิ่งขึ้น ทั้งคำไทยที่ใช้กันอยู่ทั่วไป และคำศัพท์ที่เป็นเฉพาะทาง เพื่อลดความกำกวมของภาษาให้เหลือน้อยที่สุด

และจากที่ได้พบปัญหาในการตัดคำไทยในโปรแกรมประยุกต์บางโปรแกรม เช่น Microsoft Word ซึ่งอาจทำให้การขึ้นบรรทัดใหม่ของข้อความ เมื่อสิ้นสุดขอบบรรทัดนั้น เกิดความผิดพลาดได้ ดังนั้น ปัญหาพิเศษนี้ จึงสนใจวิธีการที่จะทำให้สามารถตัดคำและขึ้นบรรทัดใหม่ โดยการพิจารณาการตัดคำจะเริ่มจากคำศัพท์ที่อยู่ตำแหน่งสิ้นสุดบรรทัดเป็นต้นมา

1.2 วัตถุประสงค์ของการทำปัญหาพิเศษ

เพื่อศึกษาและออกแบบขั้นตอนวิธีเพื่อการตัดคำในภาษาไทย ณ ตำแหน่งท้ายบรรทัด โดยอาศัยวิธีการเทียบคำศัพท์จากพจนานุกรม ซึ่งจะทำการค้นหาคำศัพท์ที่เป็นไปได้ที่อยู่ใกล้ตำแหน่งสิ้นสุดมากที่สุด เพื่อหาตำแหน่งที่สามารถตัดข้อความไปขึ้นบรรทัดใหม่ได้

1.3 ขอบเขตของปัญหา

1) ข้อมูลที่นำมาตัดคำนั้น จะต้องเป็นข้อมูลประเภทข้อความอย่างเดียวเท่านั้น ซึ่งสามารถตัดคำได้ทั้งข้อความที่เป็นประโยคและย่อหน้า โดยไม่สามารถตัดคำในข้อมูลที่เป็นภาพ หรือวัตถุอื่นๆ หรือข้อมูลภาพและข้อความรวมกัน
2) การตัดคำ จะใช้ข้อมูลรูปแบบตัวอักษร และขนาดตัวอักษร ตามค่ามาตรฐานในระบบปฏิบัติการวินโดวส์ ซึ่งอยู่ในลักษณะกราฟิก และมีหน่วยเป็นพิกเซล โดยข้อความที่จะนำมาตัดคำนั้น จะต้องมีรูปแบบตัวอักษรและขนาดเดียวกันหมดทั้งข้อความ
3) โดยปกติแล้ว วิธีการตัดคำในภาษาไทย จะแบ่งออกเป็น 2 ประเภทหลักๆ คือ การตัดคำโดยพิจารณาโครงสร้างภาษาไทย และการเทียบคำศัพท์จากพจนานุกรม โดยในปัญหาพิเศษนี้ จะพิจารณาเฉพาะการตัดคำโดยการเทียบคำศัพท์จากพจนานุกรมเท่านั้น
4) ฐานข้อมูลพจนานุกรมที่ใช้ในปัญหาพิเศษ จะเป็นพจนานุกรมที่สร้างขึ้นจากข้อความที่นำมาทำการตัดคำ
5) การสร้างขั้นตอนวิธีในการตัดคำภาษาไทยในปัญหาพิเศษนี้ จะพิจารณาเฉพาะความถูกต้องในการตัดคำ โดยยังไม่พิจารณาถึงความเร็วในการตัดคำ
6) การสร้างโปรแกรมเพื่อทดสอบขั้นตอนวิธีการตัดคำนั้น จะพัฒนาบนเครื่องคอมพิวเตอร์ส่วนบุคคลทั่วไป (PC-based) ที่มีระบบปฏิบัติการวินโดวส์ และทำงานเป็นเอกเทศ (Stand-alone)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) มีความรู้ความเข้าใจในขั้นตอนวิธีต่างๆในการนำมาประยุกต์ใช้ เพื่อค้นหาวิธีการตัดคำในภาษาไทย
2) ได้ขั้นตอนวิธีที่สามารถตัดคำในภาษาไทยได้อย่างถูกต้องและมีประสิทธิภาพ ทั้งสำหรับคำศัพท์ทั่วไปในชีวิตประจำวัน และศัพท์เฉพาะทาง
3) สามารถนำไปใช้ในการพัฒนา หรือเป็นพื้นฐานส่วนหนึ่ง ในการประยุกต์ใช้สำหรับงานด้านอื่นๆ เช่น การแปลภาษา การตรวจสอบคำสะกดในภาษาไทย หรือ การแปลงคำไทยให้เป็นหน่วยเสียง เป็นต้น
4) นำมาเป็นต้นแบบที่สามารถประยุกต์ใช้กับภาษาอื่นๆที่มีโครงสร้างใกล้เคียงกันได้

1.5 ขั้นตอนในการดำเนินงาน

1) รวบรวมข้อมูลและศึกษาโครงสร้างและรูปแบบคำในภาษาไทย
2) ศึกษางานวิจัยที่เกี่ยวข้องกับการตัดคำภาษาไทยด้วยวิธีต่างๆ

3) นำงานวิจัยที่เกี่ยวข้องมารวบรวม เพื่อออกแบบขั้นตอนวิธีการตัดคำภาษาไทย
4) สร้างพจนานุกรมภาษาไทย ซึ่งบรรจุทั้งศัพท์ทั่วไปและศัพท์เฉพาะทาง
5) พัฒนาโปรแกรมเพื่อทดสอบขั้นตอนวิธีการตัดคำภาษาไทยท้ายบรรทัด และรวมพจนานุกรมเข้าในโปรแกรม
6) ทดลองนำข้อความมาในลักษณะต่างๆ มาทำการตัดคำ เพื่อเปรียบเทียบผลลัพธ์ที่ได้
7) จัดทำเอกสารประกอบ

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1) เครื่องคอมพิวเตอร์ 1 ชุด
2) ระบบปฏิบัติการวินโดวส์
3) เครื่องมือที่ใช้ในการพัฒนาโปรแกรม โดยใช้ภาษา Visual C++
4) เครื่องมือที่ใช้สำรองข้อมูล เช่น แผ่นบันทึกข้อมูล CD-ROM Hard disk หรือ Flash-Drive

บทที่ 2
งานวิจัยที่เกี่ยวข้อง

2.1 ลักษณะของคำไทย

2.1.1 ประเภทของคำไทย

คำไทยนั้น สามารถแบ่งออกได้เป็น 2 ประเภท คือ

1) คำโดด คือ คำเดียวที่มีความหมายในตัวเอง เป็นคำที่ไม่สามารถตัดแบ่งพยางค์ออกไปได้อีกแล้ว เพราะเมื่อแบ่งพยางค์ออกไป ก็จะไม่ได้คำที่มีความหมายสมบูรณ์ในตัวเองอีก อาจเป็นคำที่มีพยางค์เดียวหรือหลายพยางค์ก็ได้ ตัวอย่างของคำโดด เช่น นอน อ่าน สะพาน นาฬิกา

2) คำประสม คือ การนำคำโดดตั้งแต่ 2 คำขึ้นไปมารวมกัน เพื่อให้เกิดคำที่มีความหมายใหม่ ซึ่งแตกต่างไปจากความหมายเดิม คำโดดแต่ละคำที่อยู่ในคำประสมนั้น อาจมีความหมายเกี่ยวข้องหรือไม่เกี่ยวข้องกันกับส่วนหนึ่งของความหมายของคำประสมก็ได้ ตัวอย่างคำประสม เช่น “แม่น้ำ” จะมาจาก แม่ + น้ำ และพอมารวมกันก็ไม่ได้มีความหมายว่า “แม่ของน้ำ” แต่มีความหมายเกี่ยวกับน้ำ ซึ่งเป็นส่วนหนึ่งของคำ บางครั้งความหมายของคำประสมก็อาจแตกต่างจากความหมายเดิมของคำโดดไปอย่างสิ้นเชิง เช่น ยินดี หายใจ หรือบางครั้ง การสร้างคำประสมก็อาจมาจากคำโดดที่มีความหมายใกล้เคียงกัน เช่น ดูแล สวยงาม รวมทั้งอาจเป็นคำซ้ำกันก็ได้ เช่น แดงๆ ดำๆ

2.1.2 องค์ประกอบของคำไทย

คำไทยทั่วๆไป จะมีองค์ประกอบในการสร้างคำอยู่ 6 องค์ประกอบด้วยกัน คือ

1) พยัญชนะต้น
2) สระ
3) วรรณยุกต์
4) ตัวควบกล้ำ
5) ตัวสะกด
6) ตัวการันต์

โดยคำทั่วไป จะต้องมีองค์ประกอบอย่างน้อยที่สุด คือ พยัญชนะต้น สระ และวรรณยุกต์ ในกรณีที่ไม่มีรูปวรรณยุกต์ ก็ถือว่ามีวรรณยุกต์สามัญ จึงจะเกิดเป็นคำได้ ส่วนตัวควบกล้ำ ตัวสะกด และตัวการันต์ อาจจะมีหรือไม่มีอยู่ในคำก็ได้

2.1.3 โครงสร้างของพยางค์ในภาษาไทย

โดยปกติแล้ว คำไทยส่วนใหญ่มักเป็นคำที่เป็นพยางค์เดียว (Monosyllable) แต่อาจมีบางคำที่มีหลายพยางค์ ซึ่งเราสามารถแบ่งโครงสร้างของพยางค์ในภาษาไทยได้เป็น 3 กลุ่ม

1) โครงสร้างของคำพยางค์เดียว แบ่งได้เป็น 2 ลักษณะ คือ

1.1)พยางค์ประกอบด้วยสระเสียงสั้น จะประกอบด้วย พยัญชนะ ตัวควบกล้ำ ซึ่งจะมีหรือไม่ก็ได้ สระเสียงยาว พยัญชนะที่เป็นตัวสะกด และวรรณยุกต์ เช่น จน เป็น กรุง

1.2)พยางค์ประกอบด้วยสระเสียงยาว จะประกอบด้วย พยัญชนะ ตัวควบกล้ำ ซึ่งจะมีหรือไม่ก็ได้ สระเสียงยาว ตัวควบกล้ำ ซึ่งจะมีหรือไม่ก็ได้ และวรรณยุกต์ เช่น อย่า ปลา กลัว

2) โครงสร้างของคำสองพยางค์ จะเป็นการประสมของคำพยางค์เดียว มี 2 รูปแบบ คือ

2.1)พยางค์แรกเป็นสระเสียงสั้น เช่น สนั่น สะกิด กะพริบ

2.2)พยางค์แรกเป็นสระเสียงยาวหรือมีตัวสะกด เช่น กากี มานพ กันดาร วันที่

3) โครงสร้างของคำสามพยางค์ขึ้นไป มักจะเป็นคำประสมหรือคำที่ยืมมาจากภาษาอื่น เช่น ไวโอลิน วิทยาลัย

2.1.4 ชนิดของหน่วยคำในภาษาไทย

หน่วยคำในภาษาไทย อาจแบ่งเป็น 2 ชนิด คือ

1) หน่วยคำอิสระ (Free Morpheme) คือ หน่วยคำที่ปรากฏได้ตามลำพังในประโยค หรือปรากฏร่วมกับหน่วยคำอื่นบางหน่วยที่ไม่อาจปรากฏได้ตามลำพัง

ตัวอย่างเช่น คำว่า “ตาย”

- แมวตายแล้ว
- เราไม่กลัวแม้แต่ความตาย

จะเห็นว่า “ตาย” ในประโยคที่ 1 จะสามารถอยู่ได้ตามลำพัง ส่วนประโยคที่ 2 จะต้องมีการประสมคำระหว่าง “ความ” กับ “ตาย”

2) หน่วยคำไม่อิสระ (Bound Morpheme) คือ หน่วยคำที่ปรากฏตามลำพังไม่ได้ในประโยค จะแบ่งเป็น 5 ประเภท ได้แก่

2.1) หน่วยหน้าศัพท์ (Prefix) คือ หน่วยคำไม่อิสระที่เกิดขึ้นหน้าหน่วยคำอื่นเสมอ เช่น

/การ-/ /ความ-/ /นัก-/ /ชาว-/ /อธิ-/ /อภิ-/ /อนุ-/

2.2) หน่วยหน้าศัพท์คู่ (Double Prefix) คือ หน่วยคำไม่อิสระที่มีรูปและเสียงเหมือนกัน 1 คู่ เช่น

/ กระ – กระ - /	+	จุ๋มจิ๋ม	=>	กระจุ๋มกระจิ๋ม
/ ขะ – ขะ - /	+	มอมแมม	=>	ขะมอมขะแมม
/ อี – อี - /	+	เก้กัง	=>	อีเก้อีกัง

2.3) หน่วยกลางศัพท์ (Infix) คือ หน่วยคำไม่อิสระที่ปรากฏกลางหน่วยคำอื่น เช่น

/ - อำ - /	เสร็จ	สำเร็จ
/ - อำ น - /	เกิด	กำเนิด

2.4) หน่วยท้ายศัพท์ (Suffix) คือ หน่วยคำไม่อิสระที่เกิดท้ายหน่วยคำอื่น เช่น

/ - การ /	กรรมการ	กิจการ
/ - ภาพ /	อิสรภาพ	มรณภาพ
/ - ศาสตร์ /	วิทยาศาสตร์	ภูมิศาสตร์

2.5) หน่วยเสริมสร้อย (Chaining Reduplication) คือ หน่วยคำไม่อิสระที่จะปรากฏหน้าหรือหลังหน่วยคำอื่น เช่น

/ - หนังหา /	หนังสือหนังหา

2.1.5 การวิเคราะห์ตัวอักษรในภาษาไทย

จากการวิเคราะห์คำต่างๆในภาษาไทย จะสามารถแบ่งตัวอักษรและอักขระต่างๆในภาษาไทย ได้ดังนี้

1) กลุ่มพยัญชนะ 44 ตัว ได้แก่ ก ข ฃ ค ... ฮ

2) กลุ่มสระ 17 ตัว ได้แก่ ะ ั า ำ ิ ี ึ ื ุ ู เ แ โ ไ ใ ฤ ฦ

3) กลุ่มวรรณยุกต์ 4 ตัว ได้แก่ ่ ้ ๊ ๋

4) กลุ่มตัวเลข 10 ตัว ได้แก่ 0 1 2 3 4 5 6 7 8 9

5) กลุ่มอักขระอื่นๆ ได้แก่ ็ ์ ๆ ฯ ๅ ( ) “ ! ? : ; , .

2.1.5.1 การวิเคราะห์ตัวอักษรของพยัญชนะในภาษาไทย แบ่งเป็น 5 กลุ่ม คือ

1) ฉ ผ ฝ ฮ	เป็นตัวอักษรตัวแรกของพยางค์
2) ห ภ ฆ ฟ ซ ฑ	ส่วนใหญ่เป็นอักษรตัวแรกของพยางค์
3) อ ว ร	สามารถเป็นได้ทั้งตัวพยัญชนะและสระในคำ ถ้าตัว ร เป็นรูป รร จะถือว่าเป็นสระของคำ
4) ศ ณ ญ ษ ฐ ฎ ฏ ฒ ฬ ฌ	ส่วนใหญ่จะเป็นตัวสะกด
5) ก ข ค ง จ ช ภ ถ ท น ย ป พ ม ย ล ส	สามารถเป็นได้ทั้งตัวสะกด และเป็นตัวอักษรนำ

2.1.5.2 การวิเคราะห์ตัวอักษรของสระในภาษาไทย แบ่งออกเป็น

1) เ แ ไ โ ใ จะเป็นสระนำหน้าพยัญชนะอื่นๆในพยางค์ หรือเรียกว่าสระหน้า

2) ะ ำ จะเป็นสระตัวสุดท้ายของคำ และไม่ต้องการตัวสะกด

3) ั ็ ิ ึ จะเป็นสระที่ต้องมีตัวสะกด

4) า ี ื ุ ู จะเป็นสระที่มีตัวสะกดหรือไม่ก็ได้

5) ฤ จะเป็นสระของพยางค์บางคำ

2.1.5.3 การวิเคราะห์ตัวอักษรของวรรณยุกต์ในภาษาไทย

1) เ แ โ ไ ใ จะอยู่ข้างหน้าคำโดยมีพยัญชนะตาม และมีวรรณยุกต์ด้วยก็ได้

2) ฤ ์ จะไม่มีวรรณยุกต์ตาม

3) ั ิ ็ ึ ื ุ ู จะมีวรรณยุกต์ตามหลัง

4) ะ า ำ อ ว จะมีวรรณยุกต์อยู่หน้าของกลุ่มคำนี้

2.1.6 รูปแบบของคำในภาษาไทย

จากการวิเคราะห์คำในภาษาไทยทั่วไป จะพบว่าจะมีรูปแบบของคำอยู่ 7 แบบ ดังนี้

1) $C(X)^{(T)}V(S(S))$ เช่น การ ปลาย ม้าม คล้าย

2) $C(X)^{V(T)}(S(S))$ เช่น กิน ขั้น ที่ มือ ปริ่ม

3) $C(X)_{V}^{(T)}(S(S))$ เช่น สุข ปู่ คลุก พรู

4) $VC(X)^{(T)}(S(S))$ เช่น เลย ไหน ใหม่ แกล้ง

5) $VC(X)^{(T)}V(S(S))$ เช่น เสา เกลา เศร้า แทะ

6) $VC(X)^{V(T)}(S(S))$ เช่น เกิน เพริด เจิ่ง

7) $VC(X)^{V(T)}V(S(S))$ เช่น เสื่อ เพื่อน เสียง เกรียง

โดย C คือ พยัญชนะ

X คือ ตัวควบกล้ำ ซึ่งเป็นพยัญชนะ ร ล ว โดยที่ $X \subseteq C$

V คือ สระกลาง(V) สระบน(^{V}) และสระล่าง($_{V}$)

T คือ วรรณยุกต์

S คือ ตัวสะกด โดยที่ $S \subseteq C$

( ) ตัวที่อยู่ภายในวงเล็บนั้น อาจจะมีหรือไม่ก็ได้

2.2 พจนานุกรมอิเล็กทรอนิกส์

พจนานุกรมอิเล็กทรอนิกส์ เป็นสิ่งจำเป็นอย่างมาก โดยเฉพาะกับการประมวลผลทางภาษาศาสตร์ เพราะจะเป็นแหล่งข้อมูลในการจัดเก็บหน่วยคำ ไวยากรณ์ วิธีการอ่าน หรือข้อมูลอ้างอิงอื่นๆที่เกี่ยวข้องกับภาษา ซึ่งจะต้องนำมาประมวลผลกับคอมพิวเตอร์ ดังนั้น พจนานุกรมที่ดีจะต้องสามารถค้นหาคำได้อย่างรวดเร็ว และมีข้อมูลเพียงพอที่จะสามารถพบคำที่ต้องการในพจนานุกรมได้เสมอ ลักษณะของพจนานุกรมนั้นอาจมีความแตกต่างกันไปตามลักษณะการใช้งาน เช่น พจนานุกรมที่ในการแปลภาษา ก็จะต้องมีความหมายของคำ พจนานุกรมที่ใช้ในการตรวจสอบตัวสะกด ก็อาจต้องจัดเตรียมพจนานุกรมคำคล้าย หรือ คำที่ใกล้เคียงกับคำที่สะกดผิดไว้ด้วย เพื่ออำนวยความสะดวกแก่ผู้ใช้

2.2.1 ประเภทของพจนานุกรมอิเล็กทรอนิกส์

1) แบ่งตามประเภทผู้ใช้งาน
   1.1) พจนานุกรมอเนกประสงค์ (General Dictionary) เช่น พจนานุกรมฉบับราชบัณฑิตยสถาน ปทานุกรมนักเรียน
   1.2) พจนานุกรมศัพท์เทคนิค (Technical Dictionary) จะใช้เก็บข้อมูลคำศัพท์เฉพาะสาขา หรือเฉพาะด้าน เช่น พจนานุกรมทางการแพทย์ พจนานุกรมศัพท์ทางวิศวกรรม พจนานุกรมศัพท์ทางคอมพิวเตอร์
   1.3) พจนานุกรมเล็กซิคอน (User Specific Lexicon) จะใช้เก็บข้อมูลศัพท์เฉพาะงานใดงานหนึ่งโดยเฉพาะ เช่น พจนานุกรมสำหรับการแปลภาษา
2) แบ่งตามคู่ภาษา (Classify by Language Pairs)
   2.1) พจนานุกรมที่มีคู่ภาษาเพียงภาษาเดียว (Mono – Lingual Dictionary) เช่น พจนานุกรมฉบับราชบัณฑิตสถาน (ไทย – ไทย) พจนานุกรมของลองแมน (อังกฤษ – อังกฤษ)
   2.2) พจนานุกรมที่มีคู่ภาษา 2 ภาษา (Bilingual Dictionary) เช่น พจนานุกรมไทย – อังกฤษ
   2.3) พจนานุกรมที่มีคู่ภาษามากกว่า 2 ภาษา (Multilingual Dictionary) เช่น พจนานุกรมไทย – จีน – ญี่ปุ่น
3) แบ่งตามเนื้อหาของข้อมูลที่บันทึก
   3.1) พจนานุกรม (Dictionary) จะเก็บข้อมูลเกี่ยวกับความหมายของคำศัพท์
   3.2) เล็กซิคอน (Lexicon) จะเก็บข้อมูลเกี่ยวกับคำศัพท์ เพื่อใช้ในงานเฉพาะด้านต่างๆ
   3.3) พจนานุกรมคำพ้อง (Thesaurus) จะเก็บรวบรวมคำศัพท์ที่มีความหมายใกล้เคียงกัน บรรจุไว้ด้วยกัน

3.4) สารานุกรม (Encyclopedia) จะเป็นข้อมูลอ้างอิงในเรื่องต่างๆที่เกี่ยวข้อง

4) แบ่งตามกลุ่มผู้ใช้

4.1) พจนานุกรมสำหรับมนุษย์ (Dictionary for Human)

4.2) พจนานุกรมสำหรับเครื่องคอมพิวเตอร์ (Dictionary for Computer)

2.2.2 ข้อมูลที่เก็บในพจนานุกรมอิเล็กทรอนิกส์

พจนานุกรมอิเล็กทรอนิกส์นั้น มีอยู่ด้วยกันหลากหลายประเภท ขึ้นอยู่กับวัตถุประสงค์ในการใช้งาน ดังนั้น รายละเอียดข้อมูลที่เก็บในพจนานุกรมก็จะแตกต่างกันไป พจนานุกรมที่สามารถเก็บข้อมูลทุกอย่างที่ครอบคลุมการใช้งานได้หลายๆด้าน ก็จะทำให้พจนานุกรมนั้นมีประสิทธิภาพสูงในการใช้งาน รายละเอียดข้อมูลที่มักเก็บในพจนานุกรมอิเล็กทรอนิกส์ ตัวอย่างเช่น

- คำหรือหน่วยคำ
- ข้อมูลระดับคำ และการแบ่งแยกคำและพยางค์
- ข้อมูลที่เกี่ยวกับการผันของคำ
- รายละเอียดเกี่ยวกับการผันของคำ
- การออกเสียง
- ชนิดของคำ
- ความถี่ในการใช้คำ และความสำคัญของคำ
- ข้อมูลเกี่ยวกับลักษณะของคำ
- โครงสร้างไวยากรณ์
- คำพ้องความหมาย
- การเชื่อมโยงในคำ และไวยากรณ์

นอกจากรายละเอียดข้อมูลดังที่กล่าวมาแล้ว ก็ยังอาจจะมีการเก็บรายละเอียดอื่นๆอีกมากขึ้นอยู่กับการใช้งานที่จำเป็นในการประมวลผลทางภาษาศาสตร์

2.2.3 โครงสร้างข้อมูลที่ใช้สำหรับพจนานุกรมอิเล็กทรอนิกส์

โครงสร้างข้อมูล หมายถึง การจัดรูปแบบของข้อมูล ซึ่งจะนำมาเก็บไว้ในหน่วยความจำ ดังนั้น การออกแบบโครงสร้างข้อมูลที่ดีจะเอื้อให้เกิดประสิทธิภาพที่ดีในการประมวลผลกับคอมพิวเตอร์ เช่น

- ทำให้การใช้หน่วยความจำได้อย่างมีประสิทธิภาพ
- สามารถนำข้อมูลที่ต้องการมาใช้ประมวลผลได้อย่างรวดเร็ว
- สามารถค้นหาคำได้รวดเร็ว

- สามารถปรับปรุงเพิ่มคำในพจนานุกรมได้ง่าย และพัฒนาพจนานุกรมได้อย่างมีประสิทธิภาพ

การเลือกวิธีการจัดเก็บข้อมูลสำหรับพจนานุกรม จะต้องคำนึงถึงจุดมุ่งหมายในการใช้งานเป็นหลัก เช่น ถ้าข้อมูลมีจำนวนมาก ก็ควรคำนึงถึงโครงสร้างที่สามารถรองรับข้อมูลได้ปริมาณมากๆ วิธีการในการค้นหาข้อมูลนั้นก็มีหลายวิธีด้วยกัน แต่ละวิธีก็จะมีความแตกต่างกัน ทั้งขนาดของข้อมูลและความรวดเร็วในการค้นหา รูปแบบของโครงสร้างข้อมูล มีดังนี้

1) โครงสร้างข้อมูลแบบลำดับ (Sequential List) เป็นโครงสร้างข้อมูลแบบง่ายที่สุด คือ จะเก็บคำต่อเนื่องกันไปเรื่อยๆ โดยไม่จำเป็นต้องมีการเรียงลำดับตามตัวอักษร การค้นหาคำจะใช้การอ่านคำจากพจนานุกรมทีละคำ แล้วนำไปเปรียบเทียบว่าคำนั้นตรงกับคำที่ต้องการหรือไม่ ไล่ตั้งแต่คำแรกไปจนคำสุดท้ายในพจนานุกรม หรือจนกว่าจะพบคำที่ต้องการ วิธีนี้จะพบคำที่ต้องการค่อนข้างช้า โดยเฉพาะถ้าหากคำที่ต้องการนั้น อยู่ในลำดับท้ายๆในพจนานุกรม เวลาที่ใช้ในการค้นหาอย่างมากที่สุดของอัลกอริทึมเป็น O(n) อัลกอริทึมของการค้นหาแบบลำดับ เป็นดังนี้

```
WHILE  i < n {
        IF WORD = LIST[i]   //พบคำที่ต้องการ
        { break; }
        }
        //ถ้า i >= n แสดงว่า ไม่พบคำที่ต้องการ
เมื่อ   WORD        เป็นคำที่ต้องการค้นหา
        n           เป็นจำนวนของคำทั้งหมดในพจนานุกรม
        LIST        เป็นแถวลำดับที่เก็บคำทั้งหมดในพจนานุกรม
```

เนื่องจากวิธีการนี้ จะค้นหาคำได้ช้า จึงไม่นิยมนำมาใช้ในการค้นหาคำในพจนานุกรม เป็นแต่เพียงวิธีการพื้นฐานในการค้นหาเพื่อนำไปสู่วิธีการอื่นๆต่อไป

2) โครงสร้างข้อมูลแบบเรียงลำดับ (Sorted List) จะเป็นโครงสร้างที่คล้ายคลึงกับโครงสร้างแบบลำดับ คือ ค้นหาตั้งแต่คำแรกไปจนถึงคำสุดท้าย แต่จะต้องมีการเรียงลำดับข้อมูลมาแล้ว อัลกอริทึมของการค้นหาแบบเรียงลำดับ เป็นดังนี้

```
WHILE  i < n  {
        IF WORD = LIST[i]   //พบคำที่ต้องการ
        { break; }
        IF WORD < LIST[i]   //ไม่พบคำที่ต้องการ
        { break; }
}
```

```
//ถ้า i >= n แสดงว่า ไม่พบคำที่ต้องการ
```

วิธีนี้ มักใช้กับการสร้างพจนานุกรมในยุคเริ่มแรก โดยการเก็บคำเรียงต่อกันไปตามลำดับในพจนานุกรม เช่น วิธี Table – Table – Linear Search [6]

3) โครงสร้างข้อมูลแบบดัชนี (Indexed List) เป็นวิธีการที่ใช้เพิ่มความเร็วในการค้นหาคำให้กับโครงสร้างข้อมูลแบบลำดับ โดยจะมีการออกแบบเป็น 2 ส่วน ส่วนแรกจะเก็บเฉพาะอักษรตัวแรกของแต่ละคำในพจนานุกรม ซึ่งจะมีดัชนี (Index) ชี้ตำแหน่งของคำที่อยู่ในพจนานุกรม ส่วนที่สองจะใช้เก็บคำทั้งหมดในพจนานุกรม การค้นหาคำจะพิจารณาตัวอักษรตัวแรกของคำที่ต้องการหาก่อน และเปรียบเทียบกับส่วนดัชนี เพื่อทราบตำแหน่งขอบเขตของคำ แล้วนำคำไปเปรียบเทียบกับพจนานุกรมในส่วนที่สอง วิธีนี้จะทำให้สามารถค้นหาคำได้รวดเร็วมากขึ้นกว่าแบบลำดับ

วิธีนี้มักใช้ร่วมกับวิธีเรียงลำดับ เช่น การใช้วิธีดัชนี 3 ชั้น [10] คือ มีดัชนีช่วยในการค้นหาคำเป็นช่วง และใช้วิธีค้นเรียงตามลำดับภายในช่วงคำศัพท์นั้น หรืออาจใช้ร่วมกับวิธีการค้นหาแบบทวิภาค ซึ่งจะได้กล่าวต่อไป

4) โครงสร้างข้อมูลแบบเรียงลำดับที่มีการค้นหาแบบทวิภาค (Binary Search) วิธีการค้นหาคำในลักษณะนี้ จะเป็นการแก้ข้อเสียของวิธีแบบลำดับ คือ ถ้าคำที่เราต้องการหานั้นอยู่ลำดับท้ายๆของพจนานุกรมจะทำให้เราต้องเสียเวลาในการค้นหาเป็นเวลานาน แต่การค้นหาวิธีนี้จะใช้ได้ก็ต่อเมื่อข้อมูลหรือคำในพจนานุกรมต้องมีการเรียงลำดับเอาไว้อยู่ก่อนแล้ว หลักการค้นหาแบบทวิภาค คือ จะแบ่งคำในพจนานุกรมออกเป็น 2 ส่วน แล้วนำคำที่อยู่กึ่งกลางมาเปรียบเทียบกับคำที่ต้องการค้น ถ้าคำในพจนานุกรมมีค่ามากกว่าคำที่ค้นหา แสดงว่า คำที่เราต้องการจะอยู่ครึ่งแรกของพจนานุกรม แต่ถ้าคำในพจนานุกรมมีค่าน้อยกว่า แสดงว่า คำที่เราต้องการอยู่ครึ่งหลัง จากนั้นจะทำการแบ่งครึ่งข้อมูลและเปรียบเทียบเช่นนี้ไปเรื่อยๆ จนกระทั่งพบคำศัพท์ที่เราต้องการ หรือไม่พบคำนั้นในพจนานุกรม เวลาที่ใช้ในการค้นหาเป็น O(log n) อัลกอริทึมของการค้นหาแบบทวิภาค คือ

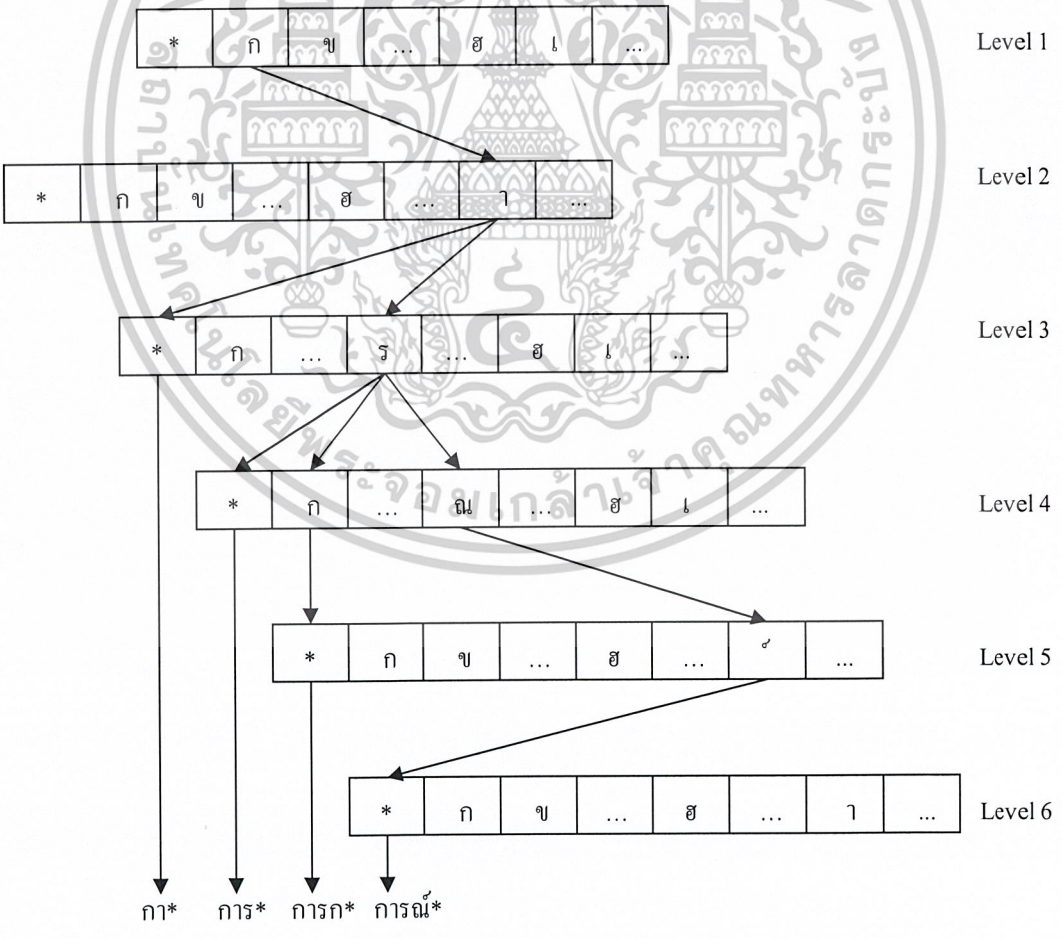
```
lower = 0;  upper = 0;
WHILE upper >= lower {
        i = (lower + upper) / 2;
        IF WORD = LIST[i]   //พบคำที่ต้องการ
        { break; }
        IF WORD < LIST[i]
        { upper = i – 1; }
        ELSE  { lower = i + 1; }
}
```

วิธีนี้ สามารถค้นหาคำได้รวดเร็ว และอาจใช้ร่วมกับดัชนี เพื่อกำหนดกลุ่มคำเป็นช่วงก่อน จากนั้นใช้วิธีการค้นหาแบบทวิภาค เพื่อค้นหาคำในกลุ่มนั้น เช่น วิธีการจัดเก็บข้อมูลแบบตารางดัชนี และไบนารีทรี [12] วิธี Table – Index Search [6]

5) โครงสร้างข้อมูลแบบต้นไม้ (Tree Structure) เป็นโครงสร้างที่มีลักษณะเป็นลำดับชั้น (Hierarchical Structure) ซึ่งแต่ละคำในพจนานุกรม จะจัดเก็บเป็น 1 ปม ในต้นไม้ โดยจะมีการเรียงลำดับค่าของคำเป็นระดับขั้น (Level) แต่ในการนำมาใช้เก็บคำในพจนานุกรมจริงๆ จะเกิดปัญหา ทำให้มีการออกแบบโครงสร้างข้อมูลแบบทรัย (Trie Structure) ซึ่งมีพื้นฐานมาจากโครงสร้างข้อมูลแบบต้นไม้ โดยจะเก็บคำแต่ละคำเป็นปมของตัวอักษรที่นำมาประกอบเป็นคำ ซึ่งจะเชื่อมโยงกับตัวอักษรถัดไปของคำ การเก็บตัวอักษรของแต่ละปม จะมีการเก็บแบบแถวลำดับหลายมิติ(Multiarray) ซึ่งจะทำให้สามารถค้นหาคำได้อย่างรวดเร็วกว่าการเก็บข้อมูลในแบบโครงสร้างต้นไม้

ตัวอย่างการเก็บข้อมูลในโครงสร้างแบบทรัย

คำว่า กา, การ, การก, การณ์, ...



รูปที่ 2.1 แสดงโครงสร้างข้อมูลแบบทรัย (Trie Structure)

6) โครงสร้างข้อมูลแบบแฮชชิ่ง (Hashing) เป็นวิธีการที่จะทำให้สามารถค้นหาคำได้อย่างรวดเร็ว โดยใช้เวลาในการค้นหาเป็น O(n) วิธีนี้จะหาตำแหน่งที่เก็บข้อมูลโดยการผ่านข้อมูลไปในแฮชชิ่งฟังก์ชั่น (Hashing Function หรือ Key – to – Address Transformation) เพื่อให้ได้ค่าคีย์ข้อมูล การกระจายข้อมูลจะขึ้นอยู่กับฟังก์ชั่นที่เหมาะสม แต่วิธีนี้อาจเกิดปัญหาการชนกันของข้อมูล คือ ข้อมูลต่างกันเมื่อผ่านแฮชชิ่งฟังก์ชั่นแล้วได้ค่าคีย์เป็นตำแหน่งเดียวกัน

ตัวอย่างการใช้พจนานุกรมที่มีโครงสร้างแบบแฮชชิ่ง เช่น การตัดคำไทยโดยใช้ดิกชันนารีที่มีโครงสร้างข้อมูลแบบแฮชชิ่ง [11]

2.2.4 ผลงานวิจัยในการสร้างพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

1) ผลงานวิจัยของ รองศาสตราจารย์ยืน ภู่วรวรรณ ใน “การสร้างพจนานุกรมสำหรับอัลกอริทึมไทย” [4] เป็นการสร้างพจนานุกรมจากคำต่างๆที่ใช้ในชีวิตประจำวัน เพื่อเป็นฐานของการสร้างวิธีการตัดแบ่งคำไทย และการตรวจสอบข้อผิดพลาดในเวิร์ดโปรเซสเซอร์ ได้มีการเสนอพจนานุกรมไว้หลายแบบ เช่น

- การแทนที่รหัสของตัวอักษรด้วยวิธีการของ Huffman (Huffman Code) เพื่อลดขนาดของรหัสตัวอักษร จาก 7 บิท เหลือ 5.27 บิท
- การตัดคำที่ใช้ ฉ ผ ฝ ฮ เป็นพยัญชนะหน้า โดยสร้างพจนานุกรมคำยกเว้น เช่น ดิฉัน ติฉิน ฯลฯ
- การตัดคำที่ใช้ เ แ ไ โ เป็นสระหน้า โดยสร้างพจนานุกรมคำยกเว้น เช่น ขโมย พเนจร ฯลฯ
- การตัดคำ โดยหลักที่ว่า ตัวการันต์จะอยู่ท้ายของคำเสมอ และสร้างพจนานุกรมคำยกเว้น เช่น คอร์ป กอล์ฟ ฯลฯ
- การตัดคำ โดยหลักที่ว่า สระอึ จะมีตัวสะกดเพียงตัวเดียว ยกเว้น รึ อึ
- พจนานุกรมคำที่ใช้ ฤ เช่น กฤตยาคม คฤหาสน์ ฯลฯ

2) ผลงานวิจัยของ รองศาสตราจารย์ยืน ภู่วรวรรณ และ ชัยยงศ์ วงศ์ชัยสุวัฒน์ ใน “การออกแบบและการลดขนาดข้อมูลคำไทยในพจนานุกรมสำหรับพิสูจน์ตัวอักษร” [6] ได้พยายามจะลดขนาดข้อมูลที่เก็บลงในพจนานุกรม โดยการจัดทำโครงสร้างแบบมีระดับ และทำให้การค้นหาเป็นไปอย่างรวดเร็ว โดยให้พจนานุกรมมีขนาดเล็กพอที่จะนำมาอยู่ในหน่วยความจำหลักทั้งหมด ไม่ต้องไปค้นหาในหน่วยความจำสำรอง ซึ่งจะทำงานได้ช้ากว่า งานวิจัยนี้ได้เสนอรูปแบบพจนานุกรมไว้ 3 ลักษณะ คือ

2.1) โครงสร้างข้อมูลแบบ Table – Table – Linear Search

โครงสร้างวิธีนี้จะเป็นการเก็บข้อมูลเรียงต่อกัน ดังนี้

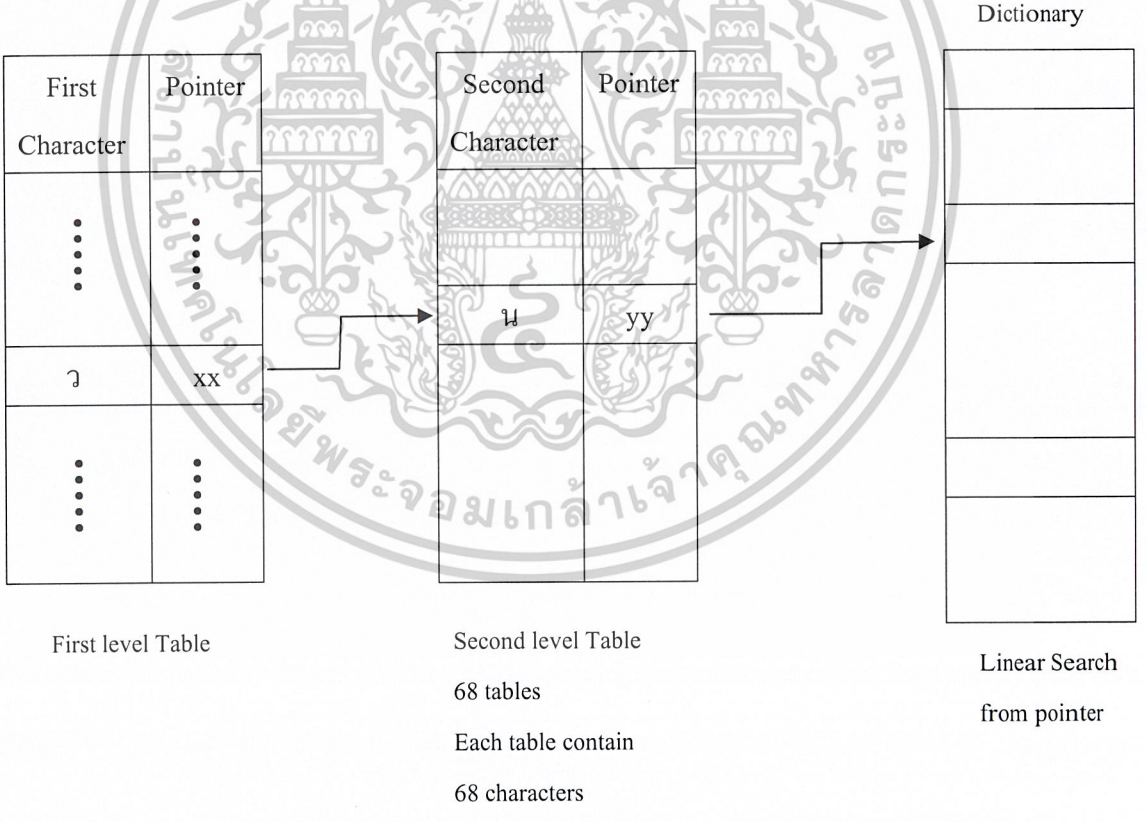
ก_____2กา 3กาก 3กาง 6กางเกง 3กาจ 5กาชาด_____

โดยตัวเลขที่นำหน้าคำจะเป็นจำนวนตัวอักษรที่ประกอบเป็นคำ ด้วยวิธีการเรียงพจนานุกรมใช้วิธี Table – Table จะใช้ตารางมานำหน้า 2 ตัวอักษร ดังนั้น โครงสร้างในพจนานุกรมจะลด 2 ตัวอักษรแรกมาอยู่ที่ตาราง ดังนี้

ก_____01ก 1ง 4งเกง 1จ 3ชาด_____

การจัดพจนานุกรมนี้ จะใช้ตัวเลข 1 ไบต์ ซึ่งเป็นเลขไบนารีมาคั่น ถ้าเป็นเลข 0 จะหมายถึง คำ 2 ตัวแรกเป็นคำในพจนานุกรม ตัวอื่นจะบอกความหมายของความยาวของคำในพจนานุกรมไปในตัว โดยลบออกจาก 2 ตัวแรก

การค้นหาข้อมูลใช้วิธี Table – Table – Linear Search เป็นดังนี้



รูปที่ 2.2 แสดงโครงสร้างข้อมูลของ Table – Table – Linear Search

ตารางระดับที่ 1 ประกอบด้วย ก – ฮ เ แ โ ไ ใ รวม 49 ตัวอักษร จะบอกตำแหน่งของตารางในระดับที่ 2 และตารางในระดับที่ 2 จะเป็นตารางที่เสมือนหาตำแหน่งคำใน

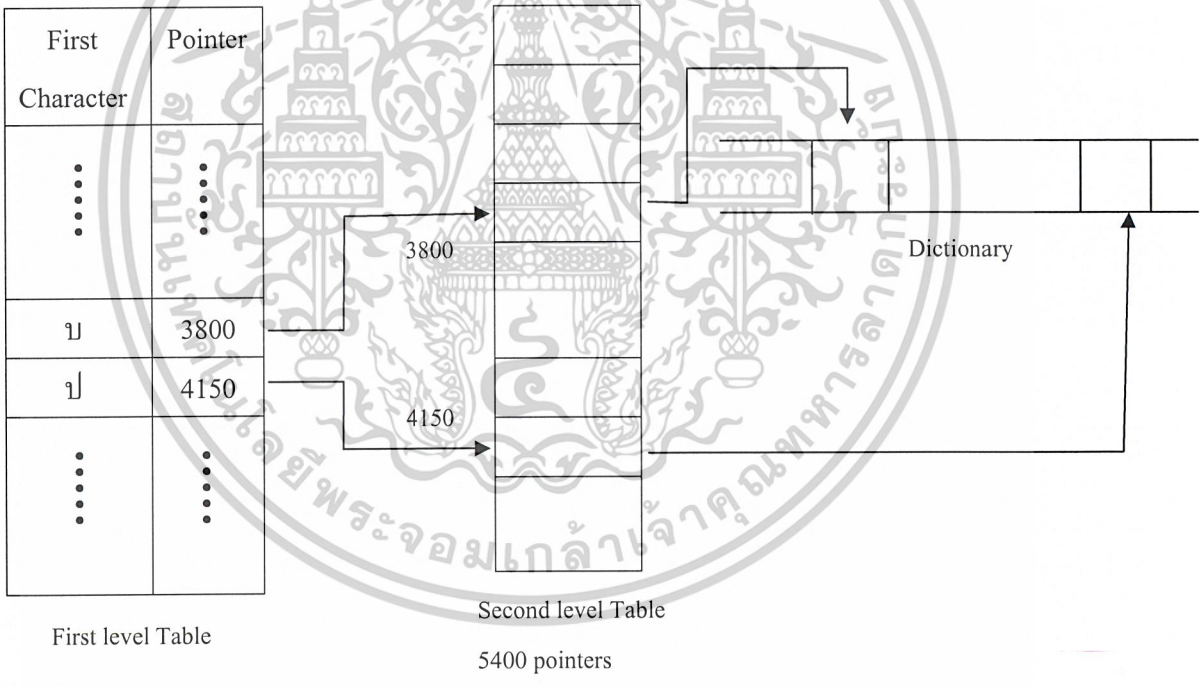
พจนานุกรม 2 ตัวแรก โดยตารางที่ 2 เป็นเสมือนตัวอักษรตัวที่สอง ดังนั้นจะมีทั้งหมด 49 ตาราง แต่ละตารางมีอักษรที่ใช้ 68 ตัว โดยตารางที่ 2 จะชี้ตำแหน่งของพจนานุกรมในตำแหน่งตัวอักษร 2 ตัวแรก

2.2) โครงสร้างข้อมูลแบบ Table - Index Search

โครงสร้างข้อมูลแบบนี้ ยึดหลักการให้ง่ายต่อการลดขนาด โดยเก็บข้อมูลพจนานุกรมมาเรียงติดต่อกันเหมือนเป็นสายอักขระหนึ่ง โดยมีตัวเลขเป็นตัวคั่นระหว่างคำ ดังนี้

2กา 3กาก 3กาง 6กางเกง 3กาจ 5กาชาด

การค้นหาจะใช้ตารางคล้ายวิธีที่ 1 โดยใช้ตัวอักษรตัวแรกปรากฏอยู่ในตารางขอบเขตกลุ่มตัวอักษร เก็บไว้ในตารางที่ 2 ซึ่งเป็นตัวชี้หาตำแหน่งคำในพจนานุกรมที่เก็บเรียงเป็นสายอักขระ ไดอะแกรมข้อมูลจะเป็นดังนี้

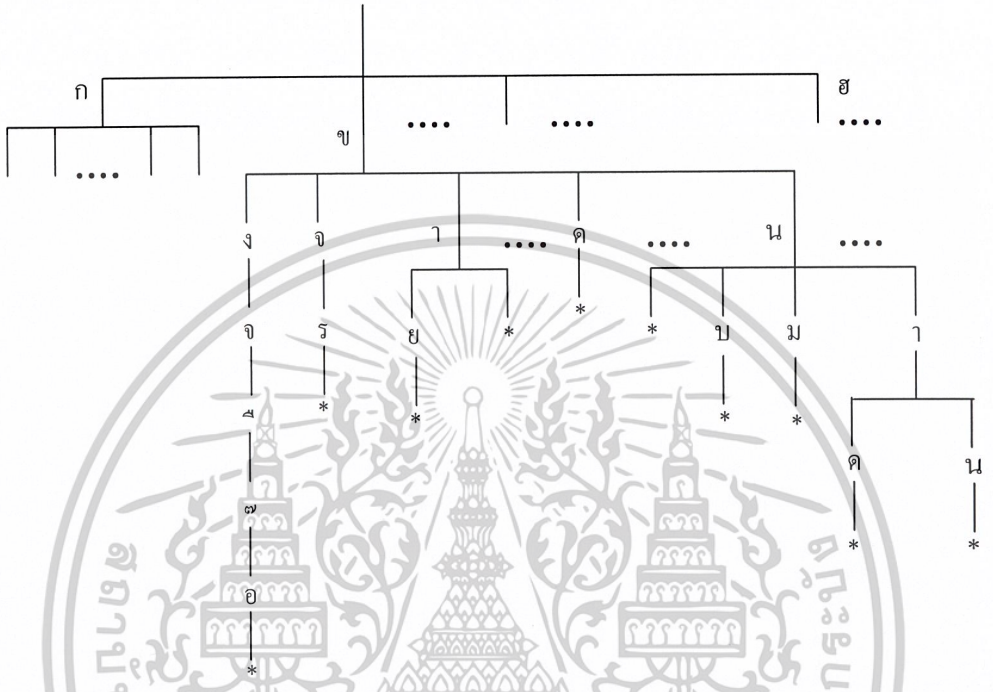


รูปที่ 2.3 แสดงโครงสร้างข้อมูลของ Table – Index Search

ในการค้นหา เช่น ต้องการหาคำว่า บ้าง ซึ่งขึ้นต้นด้วยตัว “บ” การมองตารางระดับ 1 จะบอกได้ว่ากลุ่มตัวอักษรที่ขึ้นต้นด้วยตัว บ. อยู่ที่ตำแหน่ง 3800 – 4150 ดังนั้น เราจะทำการค้นหาแบบทวิภาค ในตัวชี้ตารางระดับ 2 ตั้งแต่ 3800 – 4150 เพื่อหาคำในพจนานุกรม

2.3) โครงสร้างข้อมูลแบบต้นไม้

โครงสร้างข้อมูลแบบต้นไม้ (Tree Structure) เป็นโครงสร้างข้อมูลที่ทำการทดลองเพื่อเปรียบเทียบความเร็ว และขนาดของการเปรียบเทียบคำในพจนานุกรม ลักษณะของข้อมูลพจนานุกรมทั้งหมด เป็นดังนี้



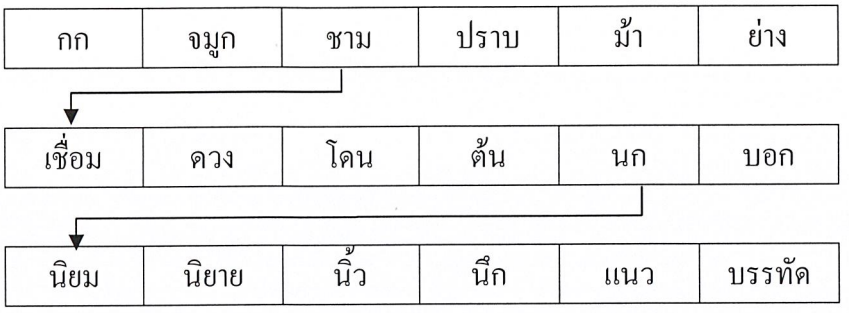
รูปที่ 2.4 แสดงโครงสร้างข้อมูลแบบต้นไม้

เมื่อเก็บอยู่ในรูปแบบแฟ้มข้อมูลเท็กซ์ จะเก็บแยกต้นไม้ออกเป็นต้นๆ ตามอักษรตัวแรก โดยมีการเข้าสู่ต้นไม้แต่ละต้นด้วยตาราง ส่วนการค้นหาคงจะใช้หลักการค้นหาตามกิ่งของต้นไม้ไปจนสุดกิ่ง ถ้าพบ * จะหมายถึงสิ้นสุดของคำที่เป็นไปได้

3) ผลงานวิจัยของ สิงห์ ตรงงาม ใน “ระบบการวิเคราะห์ประโยคภาษาไทยที่มีการละประธานที่ซ้ำกันในประโยค” [12] ได้ทำการรวบรวบงานวิจัยที่สร้างพจนานุกรมในแบบต่างๆ ดังนี้

3.1) โครงสร้างการจัดเก็บแบบลำดับดัชนี (Index Sequential)

ผลงานวิจัยของรองศาสตราจารย์ยืน ภู่วรวรรณ และ ชัยยงศ์ วงศ์ชัยสุวัฒน์ ใน “การประมวลผลภาษาธรรมชาติ” เป็นโครงสร้างในยุคต้นๆของพจนานุกรมคำศัพท์ เหมาะสำหรับข้อมูลที่มีจำนวนไม่มากนัก โครงสร้างนี้ง่ายต่อการจัดเก็บ เพิ่มเติมและแก้ไข หลักการ คือ จะแบ่งลำดับออกมาเป็นชิ้น แต่ละชิ้นจะมีการค้นหาแบบลำดับ โครงสร้างข้อมูลเขียนเป็นไดอะแกรมได้ดังรูป



รูปที่ 2.5 แสดงโครงสร้างแบบลำดับดัชนี

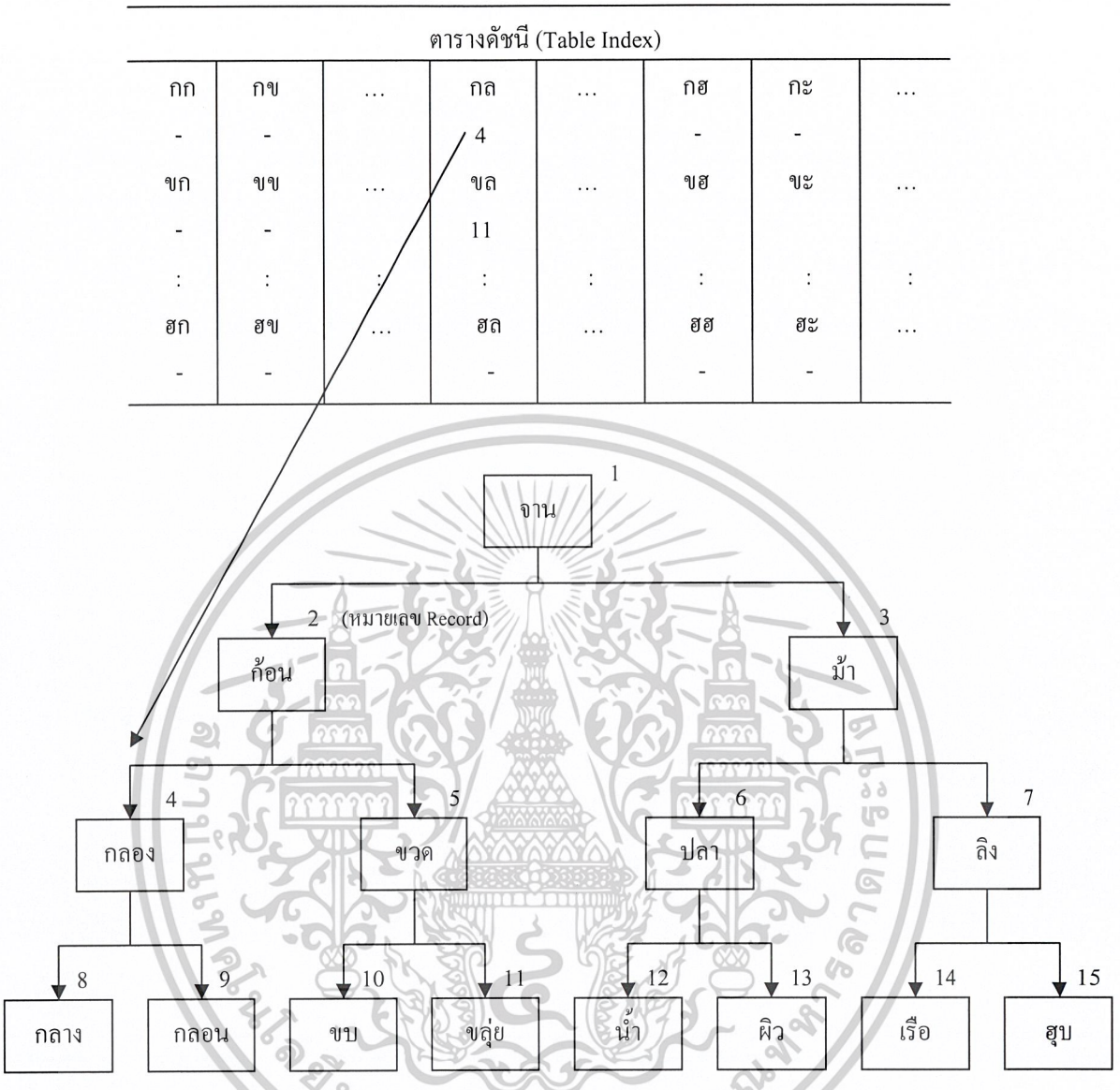
โครงสร้างแบบลำดับดัชนี จะมีการค้นหาแบบลำดับ ดังนั้น การค้นหาศัพท์ที่ต้องการ จะต้องทำการเปรียบเทียบกับคำศัพท์ในพจนานุกรมทุกตัว จึงจะทราบว่าคำศัพท์นั้นมีในพจนานุกรมหรือไม่ โครงสร้างแบบนี้ไม่เหมาะสำหรับการเก็บข้อมูลที่มีจำนวนมากและต้องการค้นหาข้อมูลอย่างรวดเร็ว

3.2) โครงสร้างการจัดเก็บข้อมูลแบบตารางดัชนี และไบนารีทรี (Table Index and Binary Tree)

ผลงานวิจัยของรัตติกร วรากูลศิริพันธุ์ และสิงห์ ตรงงาม ใน "ต้นแบบการเก็บบันทึกพจนานุกรมภาษาอังกฤษ – ไทยด้วยระบบคอมพิวเตอร์" เป็นโครงสร้างที่พัฒนามาจากโครงสร้างแบบไบนารีทรี โดยมีการเพิ่มตารางดัชนีขึ้นมาเพื่อการเข้าหาข้อมูลได้รวดเร็วขึ้น แบ่งออกเป็น 2 ส่วน คือ ส่วนของข้อมูล และส่วนของตาราง

ส่วนของข้อมูล : ในการเก็บข้อมูลจะคล้ายกับต้นไม้ โดยเปรียบเทียบข้อมูลใดมีค่ามากกว่ากัน ในที่นี้ใช้รหัสแอสกีของตัวอักษรมาใช้เปรียบเทียบ ถ้ามีค่าน้อยกว่าให้ไปทางซ้ายของกิ่ง และถ้ามีค่ามากกว่าก็จะไปทางขวาของกิ่ง แล้วเปรียบเทียบไปเรื่อยๆ จนกระทั่งถึงปลายสุด

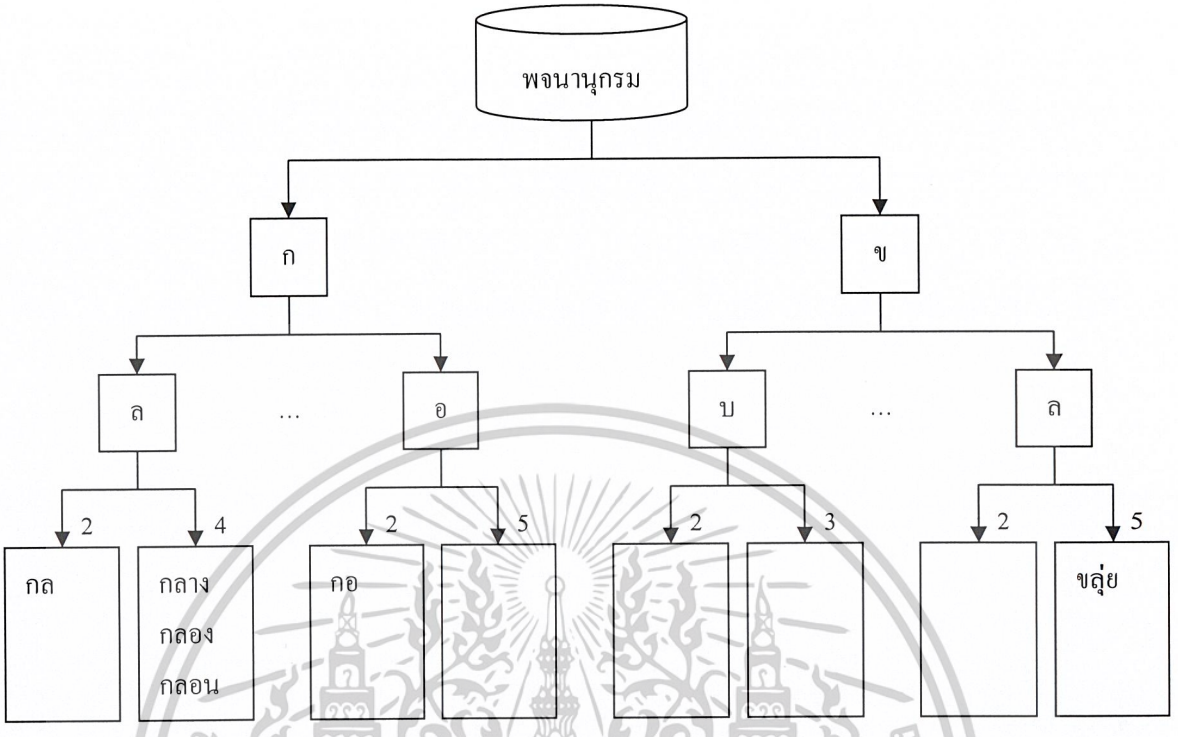
ส่วนของตาราง : เป็นส่วนที่ใช้เก็บตำแหน่งเริ่มต้นของข้อมูล โดยพิจารณาจากตัวอักษร 2 ตัวแรกของข้อมูลเป็นหลัก เพราะแทนที่จะเริ่มต้นที่ด้านบนสุดของต้นไม้ ก็สามารถข้ามไปยังตำแหน่งเริ่มต้นของข้อมูลที่ขึ้นต้นด้วยอักษรชุดนั้น โครงสร้างนี้ ถ้าจัดกิ่งได้สมดุล ก็จะเรียกค้นและเข้าถึงข้อมูลได้อย่างมีประสิทธิภาพ



รูปที่ 2.6 แสดงโครงสร้างแบบตารางดัชนีและไบนารีทรี

3.3) โครงสร้างการจัดเก็บแบบดัชนี 3 ชั้น และไล่ลำดับ (3 Index and Sequential)

ผลงานวิจัยของสมศักดิ์ จันวัน ใน “ระบบวิเคราะห์โครงสร้างภาษาไทย” [10] เป็นโครงสร้างพจนานุกรมที่พัฒนาขึ้นมาเพื่อการวิเคราะห์โครงสร้างประโยคภาษาไทย และการแบ่งคำของประโยคภาษาไทยโดยวิธี “Fast Word Matching” มีลักษณะคล้ายแบบลำดับดัชนี แต่จะมีการเพิ่มดัชนีเป็น ดัชนีชั้นที่ 1 ชั้นที่ 2 และชั้นที่ 3 เพื่อให้สามารถเข้าถึงข้อมูลได้อย่างรวดเร็ว หลักการคือ ใช้ตัวอักษร 2 ตัวแรกเป็นดัชนีตัวที่ 1 และ 2 ส่วนดัชนีตัวที่ 3 คือ จำนวนตัวอักษรของคำศัพท์นั้น ตัวอย่างของดัชนีคำศัพท์เป็นดังรูป



รูปที่ 2.7 แสดงโครงสร้างแบบ ดัชนี 3 ชั้น และไล่ลำดับ

โครงสร้างพจนานุกรมแบบนี้ สามารถค้นหาข้อมูลได้รวดเร็ว และได้มีผู้ที่นำพจนานุกรมลักษณะนี้ไปใช้กันมาก บางครั้งก็อาจสลับดัชนีตัวที่ 2 และ 3 คือ ดัชนีตัวที่ 2 เป็นจำนวนตัวอักษร และดัชนีตัวที่ 3 เป็นตัวอักษรตัวที่ 2 ของคำ ก็ตามแต่ความสะดวกในการใช้งานของผู้ใช้

4) ผลงานวิจัยของสรศักดิ์ ไทยแท้ ใน “การตัดคำไทยโดยใช้ดิกชันนารีที่มีโครงสร้างข้อมูลแบบแฮชชิ่ง” [11] ได้เสนอพจนานุกรมแบบใหม่ที่ใช้วิธีการเก็บข้อมูลแบบแฮชชิ่ง และค้นหาข้อมูลแบบแฮชชิ่ง ซึ่งทำให้สามารถสืบค้นคำได้อย่างรวดเร็ว โดยเลือกใช้ฟังก์ชั่นแฮชชิ่ง เพื่อกำหนดเลขที่อยู่แบบพหุนาม (Polynomial Addressing) มาทำการคำนวณโดยใช้ฟังก์ชั่นของ Horner ช่วย และโครงสร้างข้อมูลจะแบ่งเป็น 2 ส่วน คือ ส่วนของดัชนี และส่วนของคำที่เก็บในพจนานุกรม ส่วนการแก้ปัญหาการชนกันของข้อมูล จะใช้วิธีการค้นหาแบบลูกโซ่ (Chaining) รวมทั้งได้สร้างพจนานุกรมคำคล้าย เพื่อช่วยในการตรวจสอบตัวสะกดด้วย

2.3 การตัดคำในภาษาไทย

เนื่องจากภาษาไทยจะมีการเขียนต่อเนื่องติดกันไปทั้งประโยค แตกต่างกับภาษาอื่นๆ เช่น ภาษาอังกฤษที่มีการเว้นวรรคระหว่างคำทุกๆคำ จึงได้มีผู้ที่ทำการวิจัยเกี่ยวกับการแบ่งแยกคำในภาษาไทยออกจากกัน ซึ่งอาจแบ่งวิธีในการตัดคำภาษาไทย ออกเป็น 2 วิธีใหญ่ๆ คือ

1) การตัดคำโดยการตรวจสอบทางอักษรวิธี จะทำการตรวจสอบลักษณะการสะกดขององค์ประกอบต่างๆภายในคำ อันได้แก่ พยัญชนะ, สระ, วรรณยุกต์ และตัวการันต์ ว่าได้อยู่ในตำแหน่งที่ถูกต้องหรือไม่ รวมทั้งไวยากรณ์ในการเขียนลำดับของคำในประโยค การตัดคำด้วยวิธีนี้ได้มีผู้สร้างกฎของลักษณะคำไทยขึ้น เช่น

- พยัญชนะ ฃ และ ฅ ยกเลิกการใช้ไปแล้ว ไม่ควรปรากฏในคำอีก
- พยัญชนะ ฌ ญ ฏ ฒ ณ ฬ ไม่ปรากฏเป็นอักขระต้นของคำ นอกจากจะมีสระนำขึ้นก่อน ยกเว้น ณ
- สระหน้า เ แ ไ โ ใ จะต้องตามด้วยพยัญชนะเสมอ
- สระหลัง ะ า จะต้องตามด้วยพยัญชนะเสมอ
- สระหลังต้องมีพยัญชนะหรือวรรณยุกต์นำหน้าเท่านั้น
- วรรณยุกต์ต้องตามด้วยตัวสะกดหรือสระหลังเท่านั้น
- วรรณยุกต์ไม่สามารถเป็นอักขระแรกของคำได้
- การันต์ต้องอยู่ท้ายสุดของคำเท่านั้น ยกเว้นคำที่มาจากภาษาต่างประเทศ
- การันต์ต้องตามหลังพยัญชนะบางตัวหรือสระอิเท่านั้น

การตัดคำโดยใช้วิธีอักษรวิธี จะทำให้สามารถตัดคำได้อย่างรวดเร็ว เพราะอาศัยกฎการตัดคำเพียงไม่กี่ข้อก็สามารถทำการตัดคำได้แล้ว และไม่จำเป็นต้องมีการฝึกฝนมาก่อน ก็สามารถใช้งานได้ทันที โดยการทำเปรียบเทียบคำกับกฎไวยากรณ์ที่ตัดไว้ ซึ่งจะอาศัยการเปรียบเทียบน้อยครั้งกว่า จึงทำให้การตัดคำเป็นไปอย่างรวดเร็ว

แต่วิธีนี้ก็ยังมีข้อเสีย คือ อาจทำให้เกิดการตัดคำที่ผิดพลาดได้ง่าย เนื่องจากกฎการตรวจสอบที่ตั้งไว้นั้น ถึงอย่างไรก็ไม่สามารถครอบคลุมคำศัพท์ภาษาไทยทั้งหมดที่มีอยู่ได้ ทำให้อาจมีการตัดคำที่สะกดผิดหรือคำที่ไม่มีความหมายแต่ถูกต้องตามกฎการตรวจสอบ นอกจากนั้นวิธีนี้ยังไม่เหมาะต่อการตัดแบ่งคำเพื่อการแปลภาษา หรือเพื่อการตรวจสอบตัวสะกด ซึ่งต้องอาศัยความแม่นยำสูงในการแบ่งคำ และต้องการคำที่มีความหมายตามพจนานุกรม

2) การตัดคำโดยการตรวจสอบกับพจนานุกรม เป็นวิธีการนำคำไปเปรียบเทียบเพื่อค้นหาว่ามีคำที่ตรงกันกับคำในพจนานุกรมหรือไม่ ซึ่งจะทำให้คำที่ตัดได้มีความถูกต้องแม่นยำสูง เพราะเป็นคำที่มีความหมาย และอยู่ในพจนานุกรมจริงๆ สามารถทำได้ง่ายในทางปฏิบัติ แต่มีข้อเสีย คือ ต้องใช้เวลาในการตัดคำนานกว่าวิธีแรก เพราะต้องเสียเวลานำคำไปตรวจสอบและค้นหาใน

พจนานุกรม ซึ่งแน่นอนว่า คำที่มีในพจนานุกรมย่อมมีมากกว่ากฎทางอักษรวิธีที่ใช้ตรวจสอบ จึงต้องเปรียบเทียบมากครั้งกว่า นอกจากนั้น ถ้าพจนานุกรมบรรจุคำศัพท์ไว้น้อยเกินไป ก็อาจเกิดปัญหาในการตัดคำที่ไม่พบในพจนานุกรมได้

2.4 การตัดคำโดยการตรวจสอบทางอักษรวิธี

การตัดคำโดยวิธีทางอักษรวิธีนี้ จะแบ่งกลุ่มพยางค์ไทยออกเป็น 2 ลักษณะ คือ

1) กลุ่มตายตัว คือ รูปแบบขององค์ประกอบของคำจะมีลักษณะเดียวกัน สามารถตัดคำได้ทันทีโดยอาศัยกฎการตรวจสอบ เช่น สระ ื จะไม่พบรูปแบบ พยัญชนะต้นตามด้วย ื และ พยัญชนะต้นตามด้วย ื และตามด้วยตัวสะกด 2 ตัว
2) กลุ่มที่คลุมเครือ คือ รูปแบบขององค์ประกอบของคำอาจเป็นไปมากกว่า 1 แบบขึ้นไป ซึ่งทำให้ไม่สามารถทำการแบ่งแยกพยางค์ได้ในทันที เช่น สระ ิ อาจเป็นได้ทั้ง แบบ พยัญชนะต้นตามด้วย ิ , พยัญชนะต้น สระ ิ ตามด้วยตัวสะกด หรือ พยัญชนะต้น สระ ิ ตามด้วยตัวสะกด 2 ตัวก็ได้

2.4.1 หลักการในการกำหนดขอบเขตของคำ จะมีวิธีการ ดังนี้

1) พิจารณาขอบเขตของคำ ตามกฎการตรวจสอบทางอักษรวิธี ยกเว้นคำที่คลุมเครือสามารถเป็นได้มากกว่า 1 แบบขึ้นไป
2) พิจารณาโดยใช้ขอบเขตของอักขระตัวหน้าและตัวสุดท้ายของคำ
3) ทำเครื่องหมายกำหนดตำแหน่งของคำที่ไม่เข้าหลักเกณฑ์ เพื่อนำมาพิจารณาในภายหลัง

2.4.2 กฎการตัดคำโดยการตรวจสอบทางอักษรวิธี

1) กฎของการกำหนดขอบเขตของพยางค์ตัวหน้า

1.1) สระ า -ิ -ี -ุ -ู -ึ -ื -็ ะ ำ -ั และวรรณยุกต์ ่ ้ ๊ ๋ + จะต้องมีพยัญชนะนำอย่างน้อย 1 ตัว

1.2) สระ เ แ ไ โ จะต้องมีพยัญชนะนำตามหลัง ยกเว้นบางคำ เช่น ขโมย

1.3) สระ ใ จะมีพยัญชนะตามเสมอ

1.4) ฉ ผ ฝ ฮ จะเป็นอักษรนำหน้าเสมอ ยกเว้นมีสระ เ แ โ ไ ใ มาก่อน

1.5) ห ส่วนมากจะเป็นตัวนำของคำ ยกเว้น สห มหา คหบดี มหกรรม

1.6) ข้อ 1.1 อาจมีพยัญชนะมากกว่า 1 ตัว

2) กฎของการกำหนดขอบเขตของพยางค์ตัวหลัง

2.1) ศ ณ ญ ษ ฐ ฎ ฏ ฒ ฬ ฆ ส่วนมากเป็นตัวสะกด ยกเว้น ศก ศร ศพ ษก ญวน

2.2) ึ จะพบตัวสะกดเสมอ ยกเว้น ฤ ห ฮ

2.3) ไม้หันอากาศ จะมีตัวสะกด 1 ตัว

2.4) ำ และ ะ จะเป็นตัวสุดท้ายของพยางค์

2.5) ์ มักพบเป็นตัวสุดท้ายของคำ ยกเว้นคำที่มาจากภาษาอังกฤษ

2.6) วรรณยุกต์ที่อยู่ระหว่าง 2 พยัญชนะ ถ้าไม่มีสระ จะต้องมีตัวสะกดแน่นอน

2.7) วรรณยุกต์ที่ไม่มีสระในพยางค์ และตามด้วย อ จะต้องการตัวสะกด 1 ตัว

2.8) ิ ี ึ ื ที่มีวรรณยุกต์ จะต้องการตัวสะกด 1 ตัวเสมอ

2.9) ถ้า ื ปรากฏร่วมกับวรรณยุกต์อื่น นอกจาก ็ โดยมากจะไม่ต้องการตัวสะกด

2.10) เ–็- –ึ –็อ- แ–็- เ–ิ- จะต้องมี 1 ตัวสะกด

2.11) -ัว -ัย -ัวะ อือ เ–า เ–อ เ–าะ เ–อะ เ–ะ แ–ะ โ–ะ เ–ียะ เ–ือะ จะไม่มีตัวสะกด ยกเว้น เ–า- และ เ–อ–

2.12) สระ ใ จะมีอยู่ในภาษาไทยเพียง 20 คำเท่านั้น คือ

"ผู้ใหญ่หาผ้าใหม่ ให้สะใภ้ใช้คล้องคอ ใฝ่ใจเอาใส่ห่อ มิหลงใหล ใครขอดู
จะใคร่ลงเรือใบ ดูน้ำใส และปลาปู สิ่งใดอยู่ในตู้ มิใช่อยู่ใต้ตั่งเตียง
บ้าใบ้ถือใยบัว หูตามัวมาใกล้เคียง เล่าท่องอย่าละเลี่ยง ยี่สิบม้วนจำจงดี"

3) กฎการตัดแบ่งคำอื่นๆ

3.1) ในภาษาไทย จะใช้ . ในคำย่อ ซึ่งสามารถนำมาเป็นขอบเขตของคำได้

3.2) ฯ จะพบในคำ ฯลฯ และ ฯ พณ ฯ

3.3) ตัวอักษรย่อเป็นตัวเลข และคำที่เป็นภาษาต่างประเทศ เป็นการบอกขอบเขตของคำไทยได้

2.5 การตัดคำโดยการตรวจสอบกับพจนานุกรม

การตัดแบ่งคำไทยโดยอาศัยพจนานุกรมมาช่วยในการเปรียบเทียบคำ จะทำให้ได้คำที่ถูกต้องและมีความหมายแน่นอน ซึ่งจะมีผู้ทำการวิจัยในเรื่องนี้ไว้มากมาย และนำไปใช้ในวัตถุประสงค์ต่างๆกัน เช่น การตัดคำเพื่อการแปลภาษา การตัดคำเพื่อการแปลงเป็นหน่วยเสียง เป็นต้น

2.5.1 หลักการตัดแบ่งคำไทยโดยการตรวจสอบกับพจนานุกรม

วิธีการตัดแบ่งคำภาษาไทย อาจสรุปเป็นวิธีหลักๆ ได้ดังนี้

1) การตัดคำที่ยาวที่สุด (Longest Matching) เป็นวิธีที่ได้รับความนิยมมากในยุคต้นๆของการตัดคำ วิธีนี้จะสแกนประโยคจากซ้ายไปขวา และเลือกคำที่ยาวที่สุดที่พบในพจนานุกรม ในกรณีที่ไม่พบคำนั้นในพจนานุกรม ก็จะมีการถอยหลังไปยังตัวอักษรก่อนหน้า (Back tracking) เพื่อค้นหาคำที่ยาวที่สุดต่อไป วิธีนี้อาจจะทำให้ตัดคำออกมาผิดได้ เพราะ เลือกแต่คำที่ยาวที่สุด ซึ่งในความเป็นจริง คำที่ถูกต้องอาจไม่ใช่คำที่ยาวที่สุดก็ได้ เช่น “ไปหามเหสี” ถ้าใช้วิธีนี้ จะได้ว่า “ไป – หาม – เห – สี” ซึ่งการตัดคำที่ถูกต้อง คือ “ไป – หา – มเหสี”

2) การตัดคำที่ทำให้เกิดจำนวนคำที่น้อยที่สุด (Maximal Matching) วิธีนี้ถูกสร้างขึ้นเพื่อแก้ไขปัญหาของการตัดคำแบบวิธีแรก โดยมีหลักการ คือ จะตัดคำทุกแบบที่เป็นไปได้ออกมาก่อน ในกรณีที่ประโยคนั้นสามารถตัดคำได้หลายแบบ จากนั้นก็เลือกการตัดคำในแบบที่มีจำนวนคำในประโยคน้อยที่สุด แต่วิธีนี้ ยังมีข้อที่ต้องแก้ไขตรงที่ว่า ถ้ากรณีที่ตัดคำออกมาแล้ว ได้จำนวนคำในประโยคเท่ากัน โปรแกรมจะไม่สามารถตัดสินใจได้ว่า จะเลือกการตัดคำแบบใด ตัวอย่างเช่น “ตากลม” จะตัดได้เป็น “ตา – กลม” และ “ตาก – ลม” ซึ่งมี 2 คำเท่ากัน ถ้าเกิดกรณีเช่นนี้ ก็อาจต้องนำหลักการในวิธีแรกมาตัดสิน คือ เลือกคำว่า “ตาก – ลม” ก่อน เพราะ คำแรกที่พบนั้นยาวกว่า

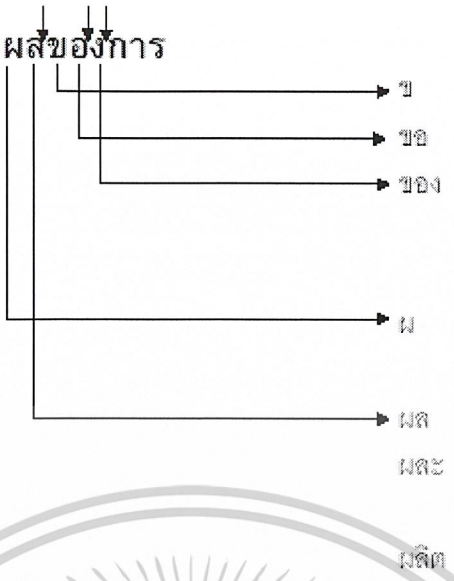
3) การตัดคำโดยอาศัยความถี่ของการใช้คำ (Statistical Matching) วิธีนี้จะต้องมีการหาค่าสถิติในการใช้คำเสียก่อน เวลาที่จะตัดคำ ก็จะเลือกคำที่ถูกใช้บ่อยที่สุดก่อน ข้อเสียของวิธีนี้นอกจากการเก็บข้อมูลทางสถิติ ซึ่งอาจไม่แม่นยำแล้ว ถ้าคำที่ถูกต้องเป็นคำที่ใช้ไม่บ่อย ก็อาจทำให้ตัดคำออกมาผิดได้

2.5.2 ผลงานวิจัยในการตัดคำโดยการตรวจสอบกับพจนานุกรม

1) ผลงานวิจัยของ รองศาสตราจารย์ยืน ภู่วรวรรณ ใน “การแบ่งพยางค์ไทยด้วยดิกชันนารี” [7] ได้เสนอวิธีการตัดคำไทยโดยการเปรียบเทียบกับดิกชันนารีที่มีอยู่ในหน่วยความจำ เช่น เมื่อป้อนข้อมูลตัวอย่างดังต่อไปนี้

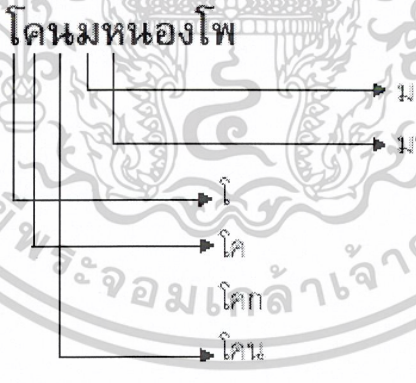
ผลของการวิจัยนี้ปรากฏว่า..

เมื่อคีย์ตัวแรก คือ “ผ” ตัวชี้จะชี้ในหมวด ผ ของดิกชันนารี เมื่อคีย์ตัวต่อไป คือ “ล” ตัวชี้จะชี้ ณ ตำแหน่งคำว่า “ผล” และเมื่อคีย์อักษรต่อไป คือ “ข” ก็จะพบว่า คำว่า “ผลข” ไม่มีในพจนานุกรม ซึ่งโปรแกรมจะทราบสุดเขตคำได้ทันที ตัวชี้จะเริ่มชี้ในหมวดตัว “ข” เป็นการเริ่มต้นพยางค์ใหม่



รูปที่ 2.8 แสดงการตัดคำโดยวิธีการตรวจสอบกับดิกชันนารี

ในการค้นหาจากดิกชันนารี อาจจะพบปัญหาในการตัดสินพยางค์สองพยางค์ที่กำกวม จึงใช้วิธีการแก้ปัญหา โดยการย้อนรอย (Back Tracking) เพื่อให้ข้อมูลนั้นตัดแบ่งได้ถูกต้องมากที่สุด พิจารณาจากตัวอย่าง



รูปที่ 2.9 แสดงการย้อนรอย เมื่อเกิดปัญหาในการตัดคำ

วิธีการทำงาน คือ เมื่อแสกนถึงคำว่า “โค” ซึ่งตรงกับคำในดิกชันนารี โปรแกรมจะทำเครื่องหมายว่ามีโอกาสตัดคำได้ แล้วสแกนต่อไปถึงคำว่า “โคน” ซึ่งก็ตรงกับดิกชันนารีอีก โปรแกรมก็จะทำเครื่องหมายว่ามีโอกาสที่ตัดคำได้ และเป็นคำที่ยาวกว่า เมื่อสแกนต่อไปเป็น “โคนม” จะไม่มีในดิกชันนารี โปรแกรมจะสมมติการตัดคำเป็น โคน - ม ซึ่งไม่ถูกต้อง เมื่อสแกนคำ

“มหน” จะไม่พบในดิกชันนารี โปรแกรมจะทำการย้อนถอยหลัง โดยถือว่าเป็น โค - นมหน แล้วตรวจสอบ นมหน ซึ่งจะได้ นม - หน

คำบางคำอาจไม่พบในพจนานุกรม เช่น ศัพท์เฉพาะ คำภาษาบาลีหรือสันสกฤต โปรแกรมจะละเลยไป และสแกนเดินหน้าไปจนกว่าจะหาคำต่อไปพบได้ในดิกชันนารี

2) ผลงานวิจัยของ ดร. รัตติกร วรากูลศิริพันธ์ และ สมศักดิ์ จันวัน ใน “การตัดคำจากประโยคในภาษาไทยด้วยวิธีการเทียบคำที่ยาวที่สุด” ได้เสนอวิธีการตัดคำที่เรียกว่า “Longest Word Mapping” เป็นการนำผลงานวิจัยของอาจารย์ยืน มาช่วยออกแบบ วิธีการตัดคำนี้ นำไปใช้ประกอบกับงานแปลภาษาด้วยเครื่องคอมพิวเตอร์ มีหลักการต่างกับวิธีแรกตรงที่จะเลือกคำที่ยาวที่สุดไปเปรียบเทียบกับพจนานุกรม ตัวอย่างเช่น

ฉันกินข้าว

ในขั้นแรก จะเอาคำที่ยาวที่สุด คือ คำว่า “ฉันกินข้าว” ไปเปรียบเทียบกับพจนานุกรมก่อน เมื่อไม่พบจะลดความยาวของคำลงทีละตัวอักษร เหลือ “ฉันกินข้า” แล้วเปรียบเทียบต่อไปจนกระทั่งพบคำ “ฉัน” จากนั้นจึงเลื่อนไปตำแหน่งถัดไป คือนำคำ “กินข้าว” ไปเปรียบเทียบในพจนานุกรมอีก เช่นนี้ไปเรื่อยๆ

3) ผลงานวิจัยของ สมศักดิ์ จันวัน ใน “ระบบวิเคราะห์โครงสร้างภาษาไทยด้วยคอมพิวเตอร์” [10] ได้เสนอวิธีการแยกแยะหน่วยคำด้วยวิธี “Fast Word Matching” ซึ่งได้พัฒนามาจากวิธี “Longest Word Mapping” คือตัดคำโดยนำคำที่ยาวที่สุดไปเปรียบเทียบก่อน แต่จะแตกต่างกันตรงที่จะตัดคำออกมาทุกแบบที่เป็นไปได้ ถ้าประโยคนั้นสามารถตัดคำได้หลายแบบ ทำให้สามารถแยกแยะหน่วยคำที่มีความกำกวมได้ ตัวอย่างเช่น

หลานมารอกราบปู่

เมื่อแยกแยะหน่วยคำนี้แล้ว จะได้ผลลัพธ์ออกมา 3 กรณี คือ

หลาน – มาร – อก – ราบ – ปู่

หลาน – มา – รอ – กราบ – ปู่

หลาน – มา – รอก – ราบ – ปู่

วิธีนี้จะไม่สนใจว่าเมื่อตัดคำออกมาแล้ว จะได้ประโยคที่ถูกต้องตามไวยากรณ์หรือไม่ ถ้าต้องการให้ได้ประโยคที่ถูกต้อง จะต้องไปพิจารณาไวยากรณ์ของประโยคอีกครั้งหนึ่ง เพื่อเลือกประโยคที่ถูกต้อง ซึ่งผลงานวิจัยนี้ ได้ใช้วิธีการวิเคราะห์โครงสร้างประโยคภาษาไทยด้วยวิธี M-ATN เพื่อเลือกประโยคที่ถูกต้อง คือ ประโยคที่ 2

4) ผลงานวิจัยของ ดร. รัตติกร วรากูลศิริพันธุ์ และคณะ ใน “การวิเคราะห์เลือกประโยคที่ถูกต้องจากความถี่ของการใช้คำ” [8] ได้เสนอการวิเคราะห์เพื่อเลือกประโยคที่ถูกต้อง เมื่อแยกแยะหน่วยคำด้วยวิธี “Longest Word Mapping” แล้ว โดยอาศัยความถี่ของการใช้คำต่างๆ ในชีวิตประจำวัน ทำการวิจัยโดยรวบรวมสิ่งพิมพ์ต่างๆมาเป็นคลังข้อมูลคำไทย เพื่อหาสถิติของการใช้คำ จากนั้น หาความน่าจะถูกใช้ในภาษาไทย (Probability of Usage หรือสัญลักษณ์ Pu) ถ้าค่า Pu สูง แสดงว่า มีสถิติหรือโอกาสในการถูกใช้ในภาษามาก ดังนั้น การเลือกประโยคที่ถูกต้อง หลังจากแยกแยะหน่วยคำแล้ว จะเลือกจากหน่วยคำของประโยคที่มีค่า Pu สูงกว่าหน่วยคำของประโยคอื่น จึงจะถือเป็นประโยคที่ถูกต้อง

5) ผลงานวิจัยของ สิงห์ ตรงงาม ใน “ระบบการวิเคราะห์ประโยคภาษาไทยที่มีการละประธานที่ซ้ำกันในประโยค” [12] ได้เสนอวิธีการตัดคำที่เรียกว่า “Complete Word” ซึ่งได้แก้ปัญหาของการแบ่งคำด้วยวิธี “Fast Word Matching” ซึ่งทำการแบ่งคำได้ช้า วิธีนี้จะมีหลักการเช่นเดียวกับวิธี “Fast Word Matching” แต่แทนที่จะเปรียบเทียบคำที่ยาวที่สุดเสมอ จะมีการเปิดตารางหาค่าความยาวสูงสุดที่พบในพจนานุกรมของข้อความที่ต้องการค้นหาก่อน ถ้ามีค่าน้อยกว่าข้อความ ก็ให้เริ่มแบ่งคำตั้งแต่ค่าที่น้อยที่สุด ตัวอย่างเช่น

ฉันมารอกราบ

ประโยคนี้มีความยาวของตัวอักษรเท่ากับ 11 และค่าที่ได้จากการเปิดตาราง “ฉั” พบว่ามีค่าสูงสุด คือ 5 ดังนั้น การแบ่งคำจะเริ่มตั้งแต่ตัวอักษรที่ 5 เป็นต้นไป คือ เริ่มหาคำ “ฉันมา” ก่อนเป็นอันดับแรก

6) ผลงานวิจัยของ สง่า คงสุพานิช ใน “การแปลงหน่วยคำภาษาไทยเป็นสัญลักษณ์แทนเสียง สำหรับงานสังเคราะห์เสียงจากประโยคภาษาไทย” [9] ได้เสนอการแบ่งคำด้วยวิธี “Suited Length Word Mapping” ซึ่งก็เป็นวิธีในลักษณะเดียวกับวิธี “Longest Word Mapping” นั่นเอง แตกต่างกันตรงที่ว่า วิธี “Suited Length Word Mapping” จะใช้พจนานุกรมที่มีดัชนีตัวแรกเป็นอักษรตัวแรก และดัชนีตัวที่ 2 เป็นความยาวของคำ และมีการหาความยาวของคำที่มากที่สุดในหมวดตัวอักษรนั้นๆ คล้ายกับ “Complete Word”

7) ผลงานวิจัยของ ประภาพรรณ คงวิทย์เศรณี ใน “การตรวจสอบและแก้ไขการสะกดคำในภาษาไทย” [2] ได้เสนอการแยกแยะหน่วยคำด้วยวิธี “Matching of Word Spelling” โดยจะสนใจในเรื่องการตรวจสอบคำผิดโดยเฉพาะ นั่นคือ หากไม่พบศัพท์ในพจนานุกรม จะตัดสินให้เป็นคำผิด แล้วหาจุดสิ้นสุดของคำที่สะกดผิดนี้ หลักการของวิธีนี้ คล้ายกับวิธี “Longest Word Mapping” แต่เมื่อไม่พบคำศัพท์ในพจนานุกรม จะนำคำไปเปรียบเทียบกับศัพท์ในกลุ่มอื่น

จนกระทั่งพบจึงจะแยกหน่วยคำออกได้ แต่ถ้าไม่พบ ก็เปรียบเทียบกับศัพท์กลุ่มอื่นต่อไป ถ้าเปรียบเทียบหมดแล้ว ยังไม่พบอีก จึงจะถือว่า คำนั้นสะกดผิด

8) ผลงานวิจัยของ สรศักดิ์ ไทยแท้ ใน “การตัดคำไทยโดยใช้ดิกชันนารีที่มีโครงสร้างข้อมูลแบบแฮชชิ่ง” [11] ได้เสนอวิธีการตัดคำโดยนำแนวทางของอาจารย์ยืน ภู่วรวรรณ ตามหลักการการย้อนรอย (Back Tracking) มาออกแบบ ซึ่งเสนอโครงสร้างข้อมูลแบบสแตก เข้ามาช่วยทำให้มีการทำงานแบบซ้ำ ซึ่งแตกต่างกับแบบรีเคอร์ชั่นเดิม หลักการ คือ จะนำตัวอักษรมาคำนวณผ่านแฮชชิ่งฟังก์ชั่น แล้วนำมาเปรียบเทียบกับคำในพจนานุกรมแบบแฮชชิ่ง ถ้าพบ ก็จะเก็บตำแหน่งนั้นไว้ในสแตกตัวที่ 1 แล้วค้นหาต่อไป จนกว่าจะได้คำที่ยาวที่สุดในการตัดแบ่งคำแต่ละรอบ แล้วเก็บลง สแตกตัวที่ 2 ทำเช่นนี้ไปเรื่อยๆ จนกว่าจะสิ้นสุดประโยค จะได้สแตก 2 ตัว โดยตัวแรกจะเก็บตำแหน่งคำที่ตัดได้ และตัวที่ 2 จะเก็บตำแหน่งที่ยาวที่สุด

2.6 ปัญหาที่มักเกิดขึ้นในการตัดคำภาษาไทย

เนื่องจากภาษาไทยจะมีลักษณะที่เขียนติดกันต่อเนื่องไปทั้งประโยค ดังนั้น อาจจะเป็นการยากที่จะตัดสินใจเลือกตัดคำที่ตำแหน่งใดบ้างในประโยค ปัญหาที่มักจะพบเสมอๆในการตัดคำ คือ

1) การตัดคำประสม

คำประสมในภาษาไทยนั้น เป็นการนำเอาคำโดดมารวมกันตั้งแต่ 2 คำขึ้น ซึ่งคำโดดแต่ละคำ ก็จะมีความหมายสมบูรณ์ในตัวเองอยู่แล้ว เมื่อนำมารวมกันให้เกิดความหมายใหม่เป็นคำประสม จึงยากที่จะวิเคราะห์ว่า ควรจะแบ่งคำเหล่านั้นเป็นคำโดดย่อยๆดี หรือจะตัดรวมเป็นคำประสมคำเดียวกันดี ตัวอย่างเช่น “หม้อหุงข้าว” ควรจะวิเคราะห์เป็นคำประสม 1 คำ หรือแยกเป็นคำโดด 3 คำ คือ “หม้อ” “หุง” และ “ข้าว” ถ้าหากเราวิเคราะห์ว่า “หม้อ – หุง – ข้าว” เป็นคำประสมคำเดียว แล้วการวิเคราะห์คำที่ใกล้เคียงกัน เช่น “หม้อ – หุง – ข้าว – ไฟ – ฟ้า” ก็ควรถือเป็นคำประสมคำเดียวด้วยเช่นกันใช่หรือไม่ หรือคำว่า “หนัง – สือ – รวม – บท – ความ – ทาง – วิ – ชา – การ – ใน – การ – ประชุม – สัม – มนา” คำนี้ควรวิเคราะห์เป็นคำเดียวกัน หรือแยกออกเป็น 9 คำ คือ “หนัง – สือ” “รวม” “บท – ความ” “ทาง” “วิ – ชา – การ” “ใน” “การ” “ประ – ชุม” “สัม – มนา”

จากปัญหาที่เกิดขึ้นจากการตัดคำประสมเหล่านี้ เป็นเหตุให้มีนักวิจัยหลายๆท่านได้ทำการวิจัยในเรื่องนี้ และได้มีนักวิจัยท่านหนึ่งได้เสนอวิธีการในการใช้การแทรกคำอื่นลงไประหว่างคำประสม เพื่อการพิจารณาในการตัดคำไทย โดยพยางค์ที่เรียงต่อกันจะถูกจัดเป็นคำ ก็ต่อเมื่อความหมายของคำเปลี่ยน ถ้ามีการแทรกพยางค์อื่นๆลงไประหว่างกลาง ตัวอย่างเช่น “หม้อ – หุง – ข้าว” จะถือเป็นคำเดียวกัน เพราะ ถ้าหากแทรกคำว่า “ที่ – ใช้ –

สำ – หรับ" ลงไประหว่างกลางของ "หม้อ" และ "หุง – ข้าว" กลายเป็น "หม้อ – ที่ – ใช้ – สำ – หรับ – หุง – ข้าว" ความหมายจะเปลี่ยนไปจากเดิม แต่คำว่า "เครื่อง – พิมพ์ – ดีด – ไฟ – ฟ้า" จะแยกออกเป็น 2 คำ คือ "เครื่อง – พิมพ์ – ดีด" และ "ไฟ – ฟ้า" เพราะ ถ้าเราแทรกคำว่า "ที่ – ใช้" ลงไประหว่างกลาง ความหมายของ "เครื่อง – พิมพ์ – ดีด – ที่ – ใช้ – ไฟ – ฟ้า" ก็ไม่แตกต่างไปจากความหมายเดิม

2) การตัดคำที่มีความกำกวม

คำบางคำในภาษาไทยสามารถตัดคำได้มากกว่า 1 แบบ ขึ้นอยู่กับบริบทโดยรอบของคำนั้น ว่ากล่าวถึงเรื่องอะไร ตัวอย่างของคำกำกวมเหล่านี้ เช่น

- ตากลม อาจตัดได้เป็น ตา – กลม / ตาก – ลม
- โคลงเรือ อาจตัดได้เป็น โค – ลง – เรือ / โคลง – เรือ
- ขนบนอก อาจตัดได้เป็น ขน – บน – อก / ขนบ – นอก
- มากว่า อาจตัดได้เป็น มา – กว่า / มาก – ว่า
- หลวงตามหาบัว อาจตัดได้เป็น หลวงตา – มหาบัว / หลวง – ตาม – หา – บัว

คำกำกวมเหล่านี้ทำให้เกิดปัญหาในการตัดคำในภาษาไทยเป็นอย่างมาก โดยเฉพาะการตัดคำที่อาศัยคอมพิวเตอร์ เพราะ ไม่มีกฎเกณฑ์แน่นอนในการตัดคำ ต้องอาศัยความหมายรอบข้างช่วยในการพิจารณาเพียงอย่างเดียว บางครั้งอาจต้องให้ผู้ใช้เป็นผู้เลือกเองว่าต้องการตัดคำในลักษณะใด

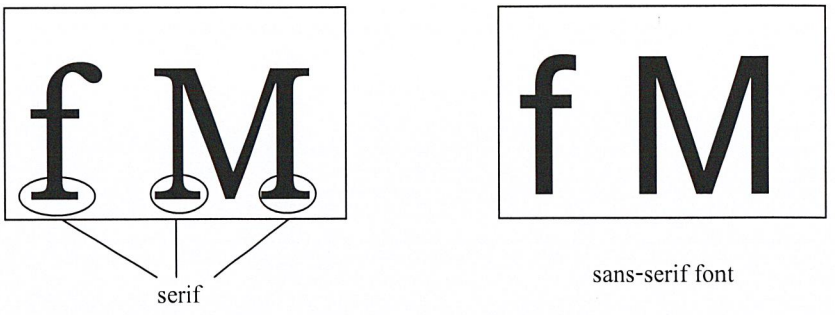
2.7 รูปแบบและความกว้างของตัวอักษร

2.7.1 รูปแบบตัวอักษร

รูปแบบตัวอักษร (Font) คือ อักขระและสัญลักษณ์ต่างๆที่มีการออกแบบในรูปแบบเดียวกัน ซึ่งจะประกอบไปด้วย 3 องค์ประกอบใหญ่ ดังนี้

1) แบบอักษร

แบบอักษร (Typeface) หมายถึง คุณสมบัติเฉพาะตัวของอักขระและสัญลักษณ์ต่างๆ ในแต่ละรูปแบบตัวอักษร ตัวอย่างเช่น ความกว้างของตัวอักษรในช่วงที่ลงเส้นหนาและลงเส้นบาง และการมีหรือไม่มีของ "Serif" โดย "Serif" คือเส้นขวางเล็กๆในตำแหน่งปลายเส้นตัวอักษรของอักษรโรมัน ซึ่งรูปแบบตัวอักษรที่ไม่มี "Serif" จะเรียกว่า "Sans-serif Font"



รูปที่ 2.10 แสดง Serif และ Sans-serif ของรูปแบบตัวอักษรบางชนิด

2) ลักษณะ

ลักษณะ (Style) หมายถึง น้ำหนักและความเอียงของรูปแบบตัวอักษร น้ำหนักของรูปแบบตัวอักษรจะเรียงตามลำดับจากบางไปยังเข้ม ดังต่อไปนี้

- Thin
- Extralight
- Light
- Normal
- Medium
- Semibold
- Bold
- Extrabold
- Heavy

ส่วนความเอียงของรูปแบบตัวอักษร จะมี 3 ลักษณะ คือ Roman, Oblique และ Italic

อักขระในรูปแบบอักษร “Roman” จะตั้งตรง ส่วนอักขระในรูปแบบ “Oblique” จะเอียงโดยการคำนวณ ซึ่งจะทำได้โดยนำอักขระในแบบ “Roman” มาแปลงให้เอียง (Shear Transformation) ส่วนอักขระในรูปแบบ “Italic” จะเป็นอักขระที่ถูกออกแบบมาให้อยู่ในลักษณะเอียงตั้งแต่แรก

3) ขนาด

ขนาด (Size) ของรูปแบบตัวอักษรจะเป็นค่าโดยประมาณไม่ชัดเจน โดยปกติแล้วขนาดจะสามารถคำนวณได้จากการวัดจากตำแหน่งล่างสุดของอักษร “g” ถึงตำแหน่งบนสุดของอักษร “M” อย่างที่แสดง ดังรูป

รูปที่ 2.11 แสดงขนาดของรูปแบบตัวอักษร

ขนาดของรูปแบบอักษรนั้นจะเรียกกันในหน่วยพอยต์ (Point) โดย 1 พอยต์ จะเท่ากับ 0.013837 นิ้ว หรือ 1 พอยต์ จะเท่ากับ 1/72 นิ้ว โดยประมาณ

2.7.2 ความกว้างและความสูงของตัวอักษร

ความกว้างของตัวอักษร (Character Widths) โดยปกติแล้วจะเป็นระยะจากขอบเขตด้านหน้าของตัวอักษรไปจนถึงขอบเขตด้านหลังของตัวอักษร ซึ่งความกว้างของอักขระแต่ละตัวนั้น ก็จะขึ้นอยู่กับรูปแบบตัวอักษร และขนาดของตัวอักษร รวมถึงน้ำหนักและความเอียงของตัวอักษรอีกด้วย นั่นคือ ถ้ามีรูปแบบตัวอักษรเหมือนกัน แต่ขนาดต่างกัน แม้จะเป็นตัวอักษรตัวเดียวกันก็ตาม แต่ความกว้างของตัวอักษรก็จะแตกต่างกัน ตัวอย่างเช่น

- อักษร “M” ในรูปแบบอักษร AngsanaUPC ขนาด 14 พอยต์ จะมีความกว้าง 11 พิกเซล
- อักษร “M” ในรูปแบบอักษร AngsanaUPC ขนาด 16 พอยต์ จะมีความกว้าง 12 พิกเซล

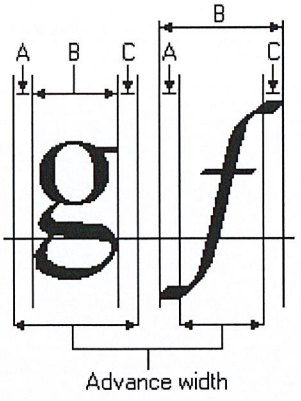
ค่าความกว้างของตัวอักษรที่สามารถนำมาประยุกต์ใช้ได้นั้น จะเรียกว่า Advance Width โดย Advance Width คือ ระยะห่างระหว่างเคอร์เซอร์ในจอแสดงผลหรือในหัวพิมพ์ของเครื่องพิมพ์ก่อนที่จะพิมพ์อักขระตัวถัดไปในข้อความนั้น ในรูปแบบตัวอักษรแบบดั้งเดิม จะระบุค่าความกว้างตัวอักษรโดยใช้ค่า Advance Width แต่สำหรับรูปแบบอักษรแบบใหม่ อย่างเช่น TrueType Font จะมีการระบุค่าความกว้างของตัวอักษรโดยแบ่งองค์ประกอบของความกว้างตัวอักษรเป็น 3 ส่วน คือ A, B และ C โดย

A คือ ช่องว่างที่เพิ่มเข้าไปในตำแหน่งปัจจุบันก่อนที่จะวางตัวอักษร

B คือ ความกว้างของตัวอักษรเอง

C คือ ช่องว่างจนถึงด้านขวาของตัวอักษร (White Space)

และความกว้างทั้งหมดของตัวอักษร หรือ Advance Width จะเท่ากับ A + B + C



รูปที่ 2.12 แสดงความกว้างของตัวอักษร (Advance Width)

ความสูงของตัวอักษร (Character Heights) จะเป็นการวัดจากตำแหน่งล่างสุดของตัวอักษร “g” จนถึงตำแหน่งบนสุดของตัวอักษร “M” ซึ่งจะเป็นค่าเฉพาะตัวของแต่ละรูปแบบตัวอักษรและขนาด หมายความว่า ถ้าเป็นรูปแบบอักษร และขนาดเดียวกันแล้ว ค่าความสูงของตัวอักษรนั้นจะมีขนาดเท่ากันหมดไม่ว่าจะเป็นตัวอักษรใดๆ เนื่องจากการนำความสูงของตัวอักษรไปใช้งานนั้น จะวัดระยะห่างของบรรทัดต่อบรรทัดมากกว่าที่จะใช้ความสูงตัวอักษรแต่ละตัว

ตัวอักษรทุกๆตัวจะมีตำแหน่งที่ถือเป็นเส้นอ้างอิงมาตรฐานในแนวนอน เรียกว่า เส้นหลักล่าง (Baseline) ซึ่งโดยปกติแล้ว อักษรทุกตัวจะเริ่มพิมพ์อักษรจากเส้นนี้เท่าๆกัน ตัวอักษรทั้งหลายจะมีส่วนที่อยู่เหนือเส้นหลักล่าง และส่วนที่อยู่ต่ำกว่าเส้นหลักล่าง เรียกว่า “Ascender” และ “Descender” ตามลำดับ องค์ประกอบต่างๆของความสูงจะแสดงได้ดังรูป

รูปที่ 2.13 แสดงความสูงของตัวอักษรและองค์ประกอบต่างๆของความสูง

ค่าความสูงต่างๆเหล่านี้ จะอยู่ในโครงสร้าง TEXTMETRIC สำหรับรูปแบบอักษรทั่วไป หรือ OUTLINETEXTMETRIC สำหรับรูปแบบอักษร TrueType Font และ OpenType Font ซึ่งใน

ปัญหาพิเศษนี้ ไม่ได้นำค่าความสูงของตัวอักษรมาใช้แต่อย่างใด จะพิจารณาเพียงค่าความกว้างของตัวอักษรเท่านั้น จึงไม่ขอกล่าวรายละเอียดเกี่ยวกับค่าความสูงของตัวอักษร

ความกว้างของตัวอักษรนั้นสามารถนำมาประยุกต์ใช้งานได้หลายประเภท โดยเฉพาะในการจัดข้อความให้อยู่ในความกว้างที่กำหนด ซึ่งการจะดึงค่าความกว้างตัวอักษรออกมาใช้ สามารถเรียกใช้ฟังก์ชั่นได้ 4 ลักษณะ ซึ่งในที่นี้ ฟังก์ชั่นในการคืนค่าความกว้างของตัวอักษรจะใช้ได้เฉพาะกับโปรแกรมที่พัฒนาบนระบบปฏิบัติการวินโดวส์เท่านั้น ฟังก์ชั่นที่ใช้ มีดังต่อไปนี้

- *GetCharWidth32* เป็นฟังก์ชั่นที่ใช้ในการดึงค่า Advance Width ของแต่ละอักขระหรือสัญลักษณ์ในข้อความ ฟังก์ชั่น GetCharWidth32 จะคืนค่า Advance Width ในแบบจำนวนเต็ม และฟังก์ชั่นนี้จะใช้ได้เฉพาะกับรูปแบบอักษรที่ไม่ใช่ TrueType Font เท่านั้น ถ้ากรณีที่เป็น TrueType Font ให้ใช้ฟังก์ชั่น GetCharABCWidths แทน
- *GetCharWidthFloat* เป็นฟังก์ชั่นที่ใช้ดึงค่า Advance Width ของอักขระเช่นเดียวกับ GetCharWidth32 แต่จะคืนค่าเป็นเลขทศนิยม
- *GetCharABCWidths* เป็นฟังก์ชั่นที่ใช้ดึงค่าองค์ประกอบของความกว้างตัวอักษร ซึ่งมี 3 ส่วนด้วยกัน คือ A, B และ C และความกว้างทั้งหมดของตัวอักษร หรือ Advance Width จะเท่ากับ A + B + C
- *GetCharABCWidthsFloat* เป็นฟังก์ชั่นที่ใช้ดึงค่าองค์ประกอบของความกว้างอักขระเช่นเดียวกับ GetCharABCWidths แต่จะคืนค่าเป็นเลขทศนิยม

ตัวอย่างค่าที่ได้จากการใช้ฟังก์ชั่น GetCharWidth32 และ GetCharABCWidths โดยในที่นี้ จะแสดงค่า A B และ C ที่ได้จากการใช้ฟังก์ชั่น GetCharABCWidths ของรูปแบบอักษร AngsanaUPC ที่มีขนาด 14 พอยต์ ดังนั้นความกว้างของตัวอักษร หรือ Advance Width จะมีค่าเท่ากับ A + B + C

ตารางที่ 2.1 แสดงค่าความกว้างของตัวอักษรที่ได้จากฟังก์ชั่น GetCharABCWidths

ตัวอักษร	A	B	C	Advance Width (A + B + C)
ก	0	6	1	7
ข	1	6	1	8
ฃ	0	7	1	8
ค	0	6	1	7
ฅ	0	6	1	7
ฆ	0	8	1	9
ง	1	4	1	6
จ	1	5	1	7
ฉ	0	8	0	8

ตารางที่ 2.1 แสดงค่าความกว้างของตัวอักษรที่ได้จากฟังก์ชั่น GetCharABCWidths (ต่อ)

ตัวอักษร	A	B	C	Advance Width (A + B + C)
ช	0	7	0	7
ซ	0	8	0	8
ฌ	0	9	1	10
ญ	0	9	1	10
ฎ	0	7	1	8
ฏ	0	7	1	8
ฐ	0	7	0	7
ฑ	0	9	1	10
ฒ	0	9	1	10
ณ	0	10	0	10
ด	0	6	1	7
ต	0	6	1	7
ถ	0	6	1	7
ท	0	7	1	8
ธ	0	6	1	7
น	0	8	1	9
บ	0	7	1	8
ป	0	7	1	8
ผ	0	6	1	7
ฝ	0	6	1	7
พ	0	8	1	9
ฟ	0	8	1	9
ภ	0	7	1	8
ม	0	7	1	8
ย	1	5	1	7
ร	1	5	0	6
ล	0	6	1	7
ว	0	5	1	6
ศ	0	7	0	7
ษ	0	7	1	8
ส	0	8	0	8
ห	0	8	1	9
ฬ	0	10	-1	9
อ	0	6	1	7
ฮ	1	6	0	7

2.7.3 ความกว้างและความสูงของข้อความ

ความกว้างของข้อความ (String Widths) จะเป็นการนำความกว้างของอักขระแต่ละตัวที่อยู่ในข้อความนั้นมารวมกัน ส่วนความสูงของข้อความ (String Heights) จะเป็นความสูงของตัวอักษรในรูปแบบอักษรและขนาดนั้น เช่นเดียวกับความสูงของตัวอักษรที่ได้กล่าวไว้ข้างต้น

นอกจากการดึงข้อมูลความกว้างของแต่ละตัวอักษรออกมาดังที่ได้กล่าวไว้แล้ว บางครั้งยังต้องมีการคำนวณหาความกว้างและความสูงของข้อความทั้งหมดอีกด้วย ฟังก์ชั่นที่ใช้ในการดึงค่าความกว้างและความสูงของข้อความ มี 2 ฟังก์ชั่น คือ

- GetTextExtentPoint32 เป็นฟังก์ชั่นที่ใช้ในการดึงค่าความกว้างและความสูงของสายอักขระ ที่ระบุ
- GetTabbedTextExtent เป็นฟังก์ชั่นที่ใช้ในการดึงค่าความกว้างและความสูงของสายอักขระ เช่นกัน แต่ใช้สำหรับสายอักขระ ที่มีอักขระแท็บรวมอยู่ด้วย

นอกจากนี้ สำหรับโปรแกรมที่ต้องการทำการตัดคำอัตโนมัติ (Word-wrapping) ก็อาจใช้ฟังก์ชั่น GetTextExtentExPoint ซึ่งเป็นฟังก์ชั่นที่จะคืนค่าจำนวนอักขระจากข้อความที่กำหนด ซึ่งสามารถบรรจุอยู่ในความกว้างที่กำหนดไว้ได้

2.7.3.1 ฟังก์ชั่น GetTextExtentPoint32

ฟังก์ชั่น GetTextExtentPoint32 เป็นฟังก์ชั่นที่ใช้ในการคำนวณความกว้างและความสูงของสายอักขระหรือข้อความที่ระบุ

พารามิเตอร์ต่างๆที่ใช้ในฟังก์ชั่น GetTextExtentExPoint มีดังนี้

```
BOOL  GetTextExtentPoint32(
      HDC       hdc,        //handle to DC
      LPCTSTR   lpString,   //text string
      int       cbString,   //characters in string
      LPSIZE    lpSize      //string size
      );
```

รายละเอียดของพารามิเตอร์แต่ละตัว เป็นดังนี้

- *hdc* [input] จะจัดการ Device Context
- *lpString* [input] เป็นตัวชี้ที่ชี้ไปยังสายอักขระที่ไม่จำเป็นต้องมีค่าว่าง (Null) ปิดท้ายสายอักขระ ก็ได้ เพราะจะมีพารามิเตอร์ cbString ที่ใช้ระบุจำนวนตัวอักษรในสายอักขระอยู่แล้ว
- *cbString* [input] จะระบุจำนวนตัวอักษรของ lpString

- *lpSize* [output] เป็นตัวชี้ที่ชี้ไปยังโครงสร้าง SIZE ซึ่งจะรับความกว้างและความสูงของสายอักขระ ในหน่วยพิกเซล

ตัวอย่างของการใช้ฟังก์ชั่น GetTextExtentPoint32

```
CString inputStr;
SIZE sz;
Cfont font;

CClientDC ClientDC(this);
ClientDC.SelectObject(font);      //set font
GetTextExtentPoint32(ClientDC, inputStr, inputStr.GetLength(), &sz);
printf("The width of input string = %d \n",sz.cx);
printf("The height of input string = %d \n",sz.cy);
```

ตารางที่ 2.2 แสดงตัวอย่างความกว้างและความสูงของข้อความ

ข้อความ	ความกว้าง	ความสูง
การทำงาน	51	26
ฉันกินข้าว	52	26
ทอย	22	26
ที่อยู่	22	26

<u>หมายเหตุ</u>

- ข้อความในตารางข้างต้นนั้น ใช้รูปแบบตัวอักษร AngsanaUPC มีขนาด 14 พอยต์ จะสังเกตได้ว่าทุกข้อความมีความสูงเท่ากันหมด เพราะใช้รูปแบบและขนาดตัวอักษรเดียวกัน ความสูงจะเท่ากัน ไม่ว่าจะเป็นข้อความใดก็ตาม
- ความกว้างและความสูงของข้อความที่ได้นั้น อยู่ในหน่วยพิกเซล
- โดยปกติแล้ว อักขระที่เป็นสระบนและสระล่างจะมีความกว้างหรือ Advance Width เท่ากับ 0

2.7.3.2 ฟังก์ชั่น GetTextExtentExPoint

ฟังก์ชั่น GetTextExtentExPoint เป็นฟังก์ชั่นที่ใช้ในการดึงจำนวนอักขระในสายอักขระที่กำหนด ซึ่งมีความกว้างไม่เกินที่ระบุไว้ และจะเก็บค่า "Text Extent" ของแต่ละตัวอักษรไว้ในแถวลำดับ โดย "Text Extent" หมายถึง ระยะห่างระหว่างตำแหน่งเริ่มต้นจนถึงตัวอักษรที่อยู่ในช่วงความกว้างนั้นๆ ข้อมูลเหล่านี้จะเป็นประโยชน์มากในการคำนวณการตัดคำโดยอัตโนมัติ

พารามิเตอร์ต่างๆที่ใช้ในฟังก์ชั่น GetTextExtentExPoint มีดังนี้

```
BOOL  GetTextExtentExPoint(
        HDC         hdc,          //handle to DC
        LPCTSTR     lpszStr,      //character string
        int         cchString,    //number of  characters
        int         nMaxExtent,   //maximum width of formatted string
        LPINT       lpnFit,       //maximum number of characters
        LPINT       alpDx,        //array of partial string widths
        LPSIZE      lpSize        //string dimension
        );
```

รายละเอียดของพารามิเตอร์แต่ละตัว เป็นดังนี้

- *hdc* [input] จะจัดการ Device Context
- *lpszStr* [input] เป็นตัวชี้ที่ชี้ไปยังสายอักขระ ซึ่งจะต้องมีค่าว่าง (Null) ปิดท้ายสายอักขระ
- *cchString* [input] จะระบุจำนวนอักขระที่มีอยู่ในสายอักขระ ที่ชี้โดยพารามิเตอร์ lpszStr
- *nMaxExtent* [input] จะระบุความกว้างมากที่สุดของข้อความที่ต้องการ ในหน่วยพิกเซล
- *lpnFit* [output] เป็นตัวชี้ที่ชี้ไปยังจำนวนเต็มที่จะรับจำนวนอักขระที่มากที่สุดซึ่งสามารถบรรจุลงในความกว้างที่กำหนดไว้ในพารามิเตอร์ nMaxExtent ได้
- *alpDx* [output] เป็นตัวชี้ที่ชี้ไปยังแถวลำดับของเลขจำนวนเต็ม ซึ่งจะรับความยาวส่วนหนึ่งของสายอักขระ แต่ละสมาชิกของแถวลำดับ จะเป็นระยะห่างระหว่างตำแหน่งเริ่มต้นของสายอักขระ จนถึงอักขระหนึ่งๆที่อยู่ในความกว้างที่ระบุในพารามิเตอร์ nMaxExtent ซึ่งแถวลำดับนี้จะต้องมีจำนวนสมาชิกอย่างน้อยที่สุดก็มากพอที่จะสามารถบรรจุอักขระทั้งหมดตามจำนวนในพารามิเตอร์ cchString ได้
- *lpSize* [output] เป็นตัวชี้ที่ชี้ไปยังโครงสร้าง SIZE ซึ่งจะรับความกว้างและความสูงของสายอักขระในหน่วยพิกเซล

ตัวอย่างของการใช้ฟังก์ชั่น GetTextExtentExPoint

```
CString inputStr;
int maxWidth;
int maxCharFit;
int arrChar[200];
SIZE sz;
Cfont font;

CClientDC ClientDC(this);
ClientDC.SelectObject(font);      //set font
GetTextExtentExPoint(ClientDC, inputStr, inputStr.GetLength(), maxWidth,
&maxCharFit, &arrChar, &sz);
printf("Number of Max character = %d \n", maxCharFit);
printf("The width of input string = %d \n",sz.cx);
```

ตารางที่ 2.3 แสดงตัวอย่างการตัดข้อความให้สามารถอยู่ในความกว้างที่กำหนด

ข้อความเดิม	ความกว้างเดิม	ความกว้างที่กำหนด	ข้อความที่ตัดแล้ว	ความกว้างใหม่	จำนวนอักขระที่อยู่ภายใน
การทำงาน	51	35	การทำ	31	5
ฉันกินข้าว	52	35	ฉันกิน	33	6
ทอย	22	15	ทอ	15	2
ที่อยู่	22	15	ที่อ	15	4

สำหรับค่าความกว้างหรือความสูงในหน่วยพิกเซล ที่ใช้ในฟังก์ชั่นข้างต้น จะเป็นค่ามาตรฐานที่ใช้ในระบบปฏิบัติการวินโดวส์ เมื่อนำมาแปลงเป็นค่าทางกายภาพ เช่น แปลงเป็นหน่วยมิลลิเมตร อาจจะแตกต่างกับค่าในโปรแกรมประมวลผลคำ เช่น Microsoft Word ซึ่งอาจจะมีการจัดรูปแบบที่แตกต่างกัน ทำให้ค่าพิกเซลต่อมิลลิเมตรแตกต่างกันไปด้วย

บทที่ 3
การออกแบบวิธีการตัดคำไทยท้ายบรรทัด

จากปัญหาที่มักพบบ่อยๆ เมื่อพิมพ์ข้อความลงในเวิร์ดโปรเซสเซอร์ หรือ โปรแกรมอื่นๆที่อยู่บนเครื่องคอมพิวเตอร์ เมื่อข้อความนั้นมีความยาวเกิน 1 บรรทัด เวิร์ดโปรเซสเซอร์ หรือโปรแกรมนั้น ก็จะปัดข้อความส่วนที่เกินบรรทัดนั้น ไปขึ้นบรรทัดใหม่ให้โดยอัตโนมัติ แต่การปัดข้อความขึ้นบรรทัดใหม่ของโปรแกรมเหล่านี้ บางครั้งอาจเกิดปัญหาการตัดคำที่ผิดพลาด คือ ตัดเอาคำที่ยาวต่อเนื่องออกจากกัน แล้วขึ้นบรรทัดใหม่ ซึ่งทำให้ความหมายของคำเสียไป ปัญหาพิเศษนี้จะสนใจการตัดคำภาษาไทย เมื่อข้อความนั้นยาวเกินกว่าที่จะจบในบรรทัดเดียวกันได้ และต้องมีการขึ้นบรรทัดใหม่ ซึ่งนอกจากจะต้องพิจารณาการตัดคำให้ถูกต้องแล้ว ยังต้องพิจารณาว่า มีเนื้อที่เหลือเพียงพอในการบรรจุคำใหม่ได้เพียงพอหรือไม่อีกด้วย

การออกแบบวิธีการตัดคำไทยท้ายบรรทัด มีปัจจัยที่ต้องพิจารณาอยู่ 4 ประการ คือ

1) พิจารณาปัจจัยที่ส่งผลต่อการตัดคำไทย
2) พิจารณาวิธีการกำหนดขอบเขตในการตัดคำ
3) พิจารณาวิธีการในการตัดคำไทยท้ายบรรทัด
4) พิจารณาวิธีการสร้างพจนานุกรม

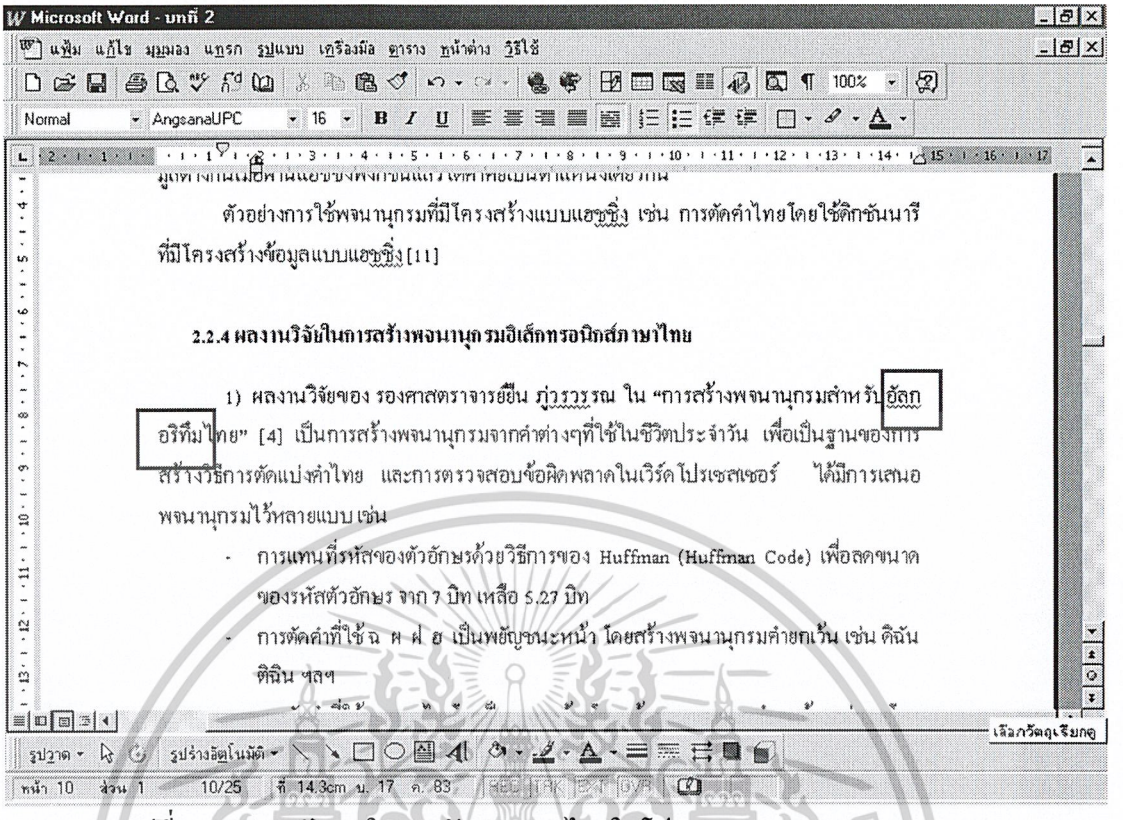
3.1 ปัญหาที่พบในการตัดคำภาษาไทยในโปรแกรมประยุกต์ที่มีอยู่ในปัจจุบัน

การตัดคำภาษาไทยในโปรแกรมประยุกต์ที่มีอยู่ในปัจจุบันนั้น บางครั้งก็ยังเกิดความผิดพลาด คือ ไม่สามารถตัดคำบางคำได้อย่างถูกต้อง เช่น คำทับศัพท์จากภาษาต่างประเทศ คำจากภาษาบาลีหรือสันสกฤต ที่เป็นคำสมาสหรือสนธิ คำต่างๆเหล่านี้ อาจพบข้อผิดพลาดได้ เช่นในโปรแกรมประยุกต์ ดังต่อไปนี้

3.1.1 ปัญหาการตัดคำภาษาไทยในโปรแกรม Microsft Word

โปรแกรม Microsoft Word ซึ่งเป็นเวิร์ดโปรเซสเซอร์หนึ่งที่นิยมใช้กันอย่างแพร่หลายในปัจจุบัน บางครั้งจะพบข้อสังเกตว่า อาจมีคำบางคำที่ตัดคำผิดพลาด สังเกตจากการขึ้นบรรทัดใหม่ของข้อความ ซึ่งทำให้คำนั้นผิดความหมายไป ตัวอย่างเช่น คำว่า “อัลกอริทึม” โปรแกรม Microsoft Word ไม่สามารถตัดคำได้ถูกต้อง

ในการสังเกตปัญหาที่ผิดพลาดของโปรแกรมในที่นี้ จะใช้โปรแกรม Microsoft Word รุ่น Microsoft Word 97



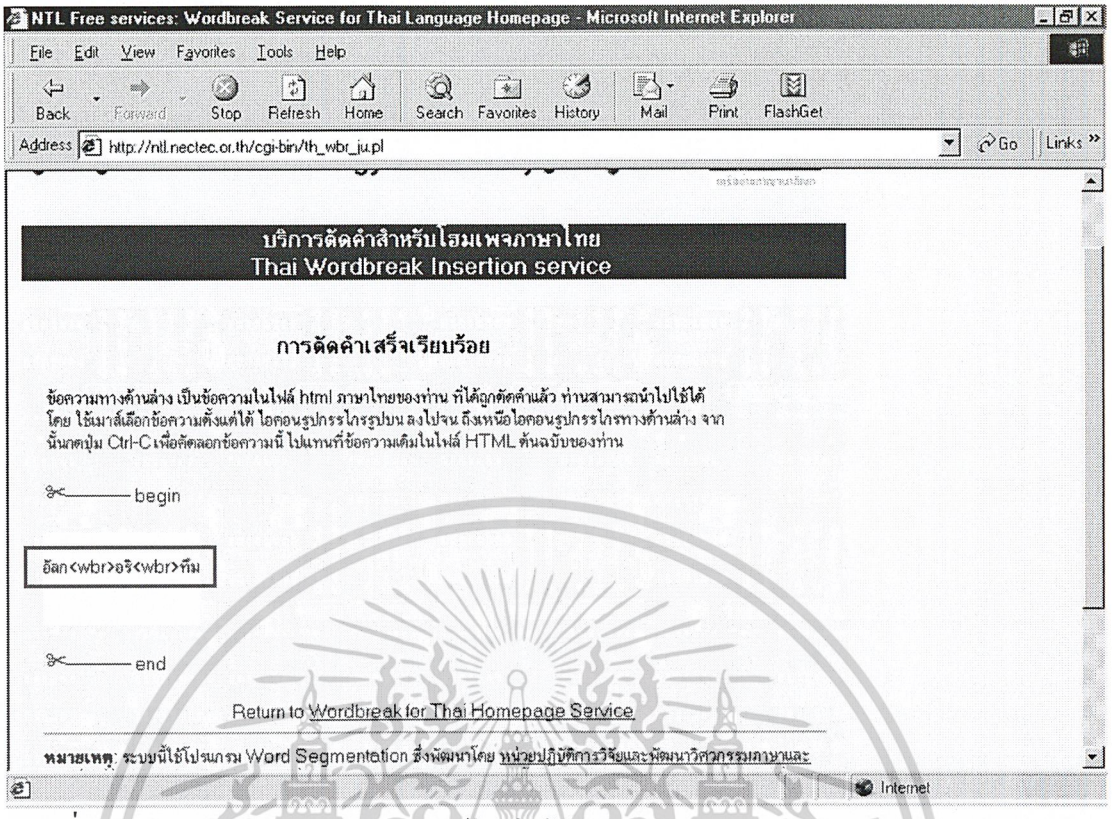
รูปที่ 3.1 แสดงปัญหาในการตัดคำภาษาไทยในโปรแกรม Microsoft Word 97

3.1.2 ปัญหาการตัดคำภาษาไทยในโปรแกรม Thai Word Break Insertion Service

โปรแกรม Thai Word Break Insertion Service เป็นบริการการตัดคำสำหรับโฮมเพจภาษาไทย ที่พัฒนาขึ้นโดยห้องปฏิบัติการคอมพิวเตอร์ของ NECTEC ซึ่งจะช่วยเพิ่มรหัสของการตัดคำเข้าไปในระหว่างข้อความภาษาไทยที่นำไปใช้บนโฮมเพจ โดยรหัสที่เพิ่มเข้าไป คือ <WBR> ซึ่งเป็นแท็กในภาษา HTML ซึ่งจะทำให้เวบบราวเซอร์สามารถรู้ได้ว่า ควรจะตัดคำที่ตำแหน่งใดบ้าง

บริการนี้เป็นบริการแบบออนไลน์อยู่บนเวบไซต์ ซึ่งสามารถจะใส่ข้อความที่เราต้องการจะตัดคำ หรือจะใส่ URL ของเวบเพจที่เราจะทำการตัดคำ เพื่อให้โปรแกรมประมวลผลและแทรกแท็ก <WBR> เข้าไประหว่างคำทุกคำ และจะแสดงผลลัพธ์ทางหน้าจอ หรือส่งข้อความกลับไปทางอีเมล์ สามารถใช้บริการนี้ได้ที่ http://ntl.nectec.or.th/services/www/thaiwordbreak.html

แต่โปรแกรมนี้ ก็ยังพบปัญหาในการตัดคำบางคำ ซึ่งไม่สามารถตัดให้ถูกต้องได้เช่นเดียวกับ โปรแกรม Microsoft Word ตัวอย่างเช่น คำว่า "อัลกอริทึม" จะตัดได้เป็น "อัลก – อริ – ทึม"



รูปที่ 3.2 แสดงปัญหาในการตัดคำภาษาไทยในโปรแกรม Thai Word Break Insertion Service

3.2 ปัจจัยที่ส่งผลต่อการตัดคำไทย

ในการตัดคำภาษาไทยจากข้อความที่ปรากฏบนเครื่องคอมพิวเตอร์นั้น ข้อความที่นำมาตัดคำนี้อาจจะมีขนาดต่างๆกัน ขึ้นอยู่กับลักษณะ รูปแบบ และขนาดของตัวอักษร ซึ่งจะส่งผลต่อการตัดคำ ว่าจะตัดคำได้ในตำแหน่งใด โดยตัวอักษรที่มีขนาดเล็ก ก็จะทำให้สามารถบรรจุข้อความลงในบรรทัดหนึ่งๆได้มากกว่าข้อความที่มีตัวอักษรขนาดใหญ่

นอกจากนั้น การจะระบุว่าข้อความนั้นๆจะบรรจุลงในแต่ละบรรทัดได้เท่าใด ก็จำเป็นต้องมีการกำหนดความกว้างของแต่ละบรรทัด ว่าต้องการให้บรรทัดหนึ่งๆมีความกว้างเท่าไหร่ ซึ่งโดยปกติแล้ว จะกำหนดความกว้างของบรรทัดในหน่วยมิลลิเมตร ตามความคุ้นเคยในการใช้งานทั่วไป และสามารถนำมาแปลงให้อยู่ในหน่วยพิกเซลได้

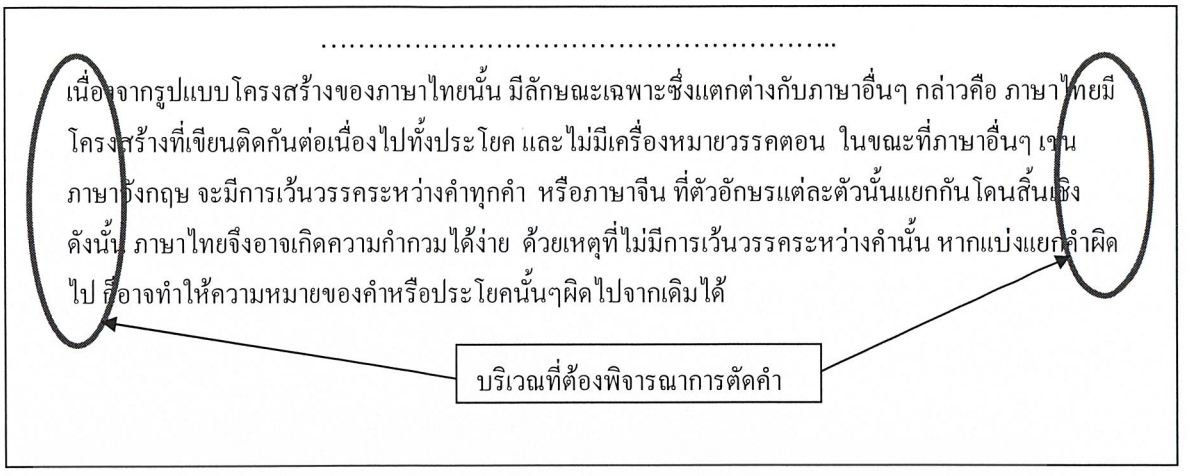
ปัจจัยที่จำเป็นต่อการตัดคำไทยท้ายบรรทัดนั้น มีดังต่อไปนี้

1) รูปแบบตัวอักษร ตัวอักษรแต่ละรูปแบบจะมีความกว้างของตัวอักษรต่างกัน ดังนั้น เราจึงต้องพิจารณาด้วยว่า ลักษณะรูปแบบของตัวอักษรที่ใช้นั้นเป็นอย่างไร ในที่นี้ จะใช้รูปแบบตัวอักษรโดยอ้างอิงจากรูปแบบตัวอักษรที่ใช้ในระบบปฏิบัติการวินโดวส์

2) ขนาดของตัวอักษร อักษรแต่ละรูปแบบนั้น ก็อาจมีขนาดต่างๆกันได้ โดยการกำหนดเป็นตัวเลขจำนวนเต็ม แสดงสัดส่วนขนาดของตัวอักษรนั้น ขนาดตัวอักษรนี้ ก็จะอ้างอิงจากขนาดของตัวอักษรในระบบปฏิบัติการวินโดวส์เช่นกัน
3) ความกว้างของหน้ากระดาษ หรือความกว้างของบรรทัด ซึ่งสามารถกำหนดได้ 2 วิธี คือ
   - กำหนดจากขนาดกระดาษ โดยเลือกขนาดกระดาษมาตรฐานทั่วไป เช่น A4 จะมีขนาด 210 x 297 มิลลิเมตร จากนั้นเลือกตั้งค่าหน้ากระดาษ โดยกำหนดขอบด้านซ้าย และขอบด้านขวา ว่าต้องการให้เริ่มพิมพ์ข้อความที่ตำแหน่งใด และสิ้นสุดบรรทัดที่ตำแหน่งใด มีหน่วยเป็นมิลลิเมตร
   - กำหนดความกว้างของกระดาษโดยตรง ว่าบรรทัดหนึ่งๆนั้น มีความกว้างเท่าใด มีหน่วยเป็นมิลลิเมตร
4) ข้อความนำเข้า ข้อความที่นำมาตัดคำนั้น จะต้องเป็นข้อความภาษาไทย ซึ่งไม่มีรูปภาพ หรือวัตถุอื่นๆ แทรกอยู่ การนำข้อความเข้ามาตัดคำนั้น ทำได้ 2 วิธี คือ
   - นำข้อความมาจากไฟล์ โดยการเลือกไฟล์ที่บรรจุข้อความที่ต้องการตัดคำ
   - นำข้อความจากหน้าจอที่รับข้อมูล โดยการพิมพ์ข้อความลงไปที่หน้าจอรับข้อมูลโดยตรง เพื่อนำข้อความนั้นมาตัดคำ
5) พจนานุกรม การที่จะตัดคำได้ถูกต้องหรือไม่นั้น ก็ขึ้นอยู่กับความถูกต้องของคำศัพท์ที่บรรจุในพจนานุกรมด้วย ซึ่งจะต้องมีคำศัพท์มากเพียงพอ เพื่อให้สามารถค้นหาคำได้พบ และเหมาะสมต่อการใช้งาน

3.3 วิธีการกำหนดขอบเขตในการตัดคำ

การตัดคำภาษาไทยในปัญหาพิเศษนี้ จะพิจารณาตัดคำเฉพาะบริเวณปลายบรรทัดและต้นบรรทัดใหม่ หรือตำแหน่งที่จะเป็นรอยต่อของแต่ละบรรทัด เพื่อพิจารณาเฉพาะการตัดคำเพื่อขึ้นบรรทัดใหม่เท่านั้น ไม่ได้ต้องการตัดคำไปเพื่อใช้ในการแปลภาษา หรือแปลงข้อความเป็นหน่วยเสียงแต่อย่างใด ดังนั้น จึงไม่จำเป็นต้องตัดคำ ณ ทุกๆตำแหน่งคำในข้อความนั้น จะพิจารณาตัดคำเฉพาะตำแหน่งที่ต้องการ คือ ตำแหน่งที่จะขึ้นบรรทัดใหม่ก็เพียงพอ



รูปที่ 3.3 แสดงบริเวณที่ต้องพิจารณาการตัดคำ

เนื่องจาก เราไม่จำเป็นต้องตัดคำทุกๆคำ ดังนั้น การตัดคำในปัญหาพิเศษนี้ จึงใช้เวลาในการพิจารณาตัดคำน้อยกว่า การจะตรวจสอบว่า จะต้องเริ่มตัดคำตั้งแต่ช่วงไหน หรือ การพิจารณาขอบเขตเริ่มต้นของการตัดคำ เป็นประเด็นหนึ่งที่ต้องสนใจ เพราะ ถ้าหากเรามีขอบเขตเริ่มต้นอยู่ใกล้เคียงกับตำแหน่งสิ้นสุดบรรทัดมากเท่าใด จำนวนครั้งที่ต้องเปรียบเทียบตัดคำก็จะลดน้อยลงเพียงนั้น

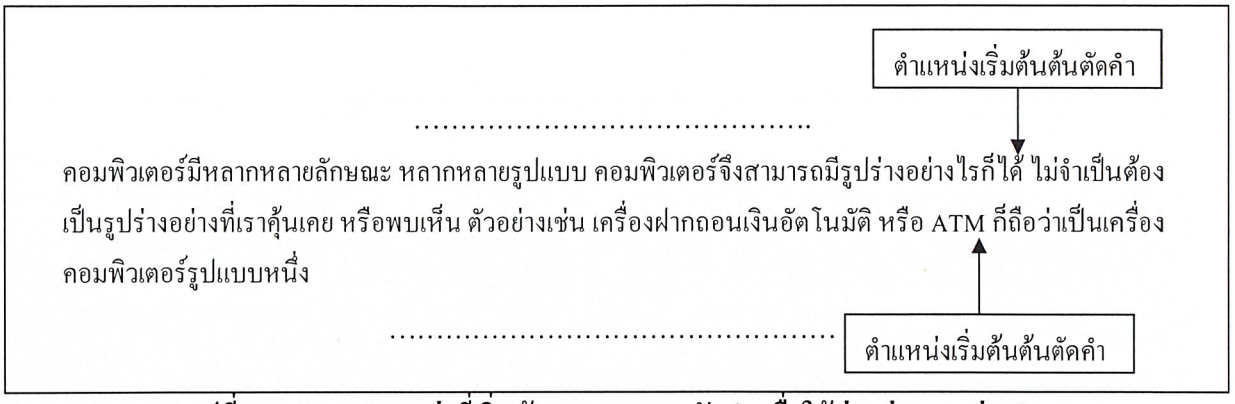
สำหรับการกำหนดขอบเขตสิ้นสุดของการตัดคำ หรือตำแหน่งท้ายบรรทัดนั้น จะอาศัยการนำข้อความทั้งหมดไปคำนวณหาความกว้างของสายอักขระด้วยฟังก์ชั่น GetTextExtentExPoint เพื่อตรวจสอบว่ามีอักขระกี่ตัวที่สามารถอยู่ในความกว้างบรรทัดที่กำหนดได้ โดยข้อมูลความกว้างของตัวอักษรนั้นจะมีขนาดเฉพาะแตกต่างกันไปตามแต่ละรูปแบบตัวอักษรและขนาดของตัวอักษร ซึ่งอักขระตัวสุดท้ายที่สามารถบรรจุในบรรทัดนั้นได้ จะถือว่าเป็นขอบเขตสิ้นสุดการตัดคำ

ส่วนการตัดสินว่าขอบเขตเริ่มต้นของการตัดคำควรอยู่ในตำแหน่งใด อาจแบ่งได้เป็น 3 วิธี ดังนี้

1) พิจารณาตำแหน่งจากการเว้นวรรค
2) พิจารณาตำแหน่งจากอักษรที่อยู่นำหน้าคำ หรืออยู่ท้ายคำเสมอ
3) พิจารณาตำแหน่งจากเครื่องหมายวรรคตอนต่างๆ

รายละเอียดในการพิจารณาขอบเขตเริ่มต้น เป็นดังนี้

1) พิจารณาตำแหน่งจากการเว้นวรรค ให้เลือกตำแหน่งที่อยู่หลังจากเว้นวรรคครั้งสุดท้ายของบรรทัด มาเป็นอักษรตัวแรกที่ต้องเริ่มพิจารณา วิธีนี้ เป็นหลักการแบบง่ายที่สุด คือ เลือกตัดคำเฉพาะประโยคสุดท้ายของบรรทัด ตัวอย่างเช่น

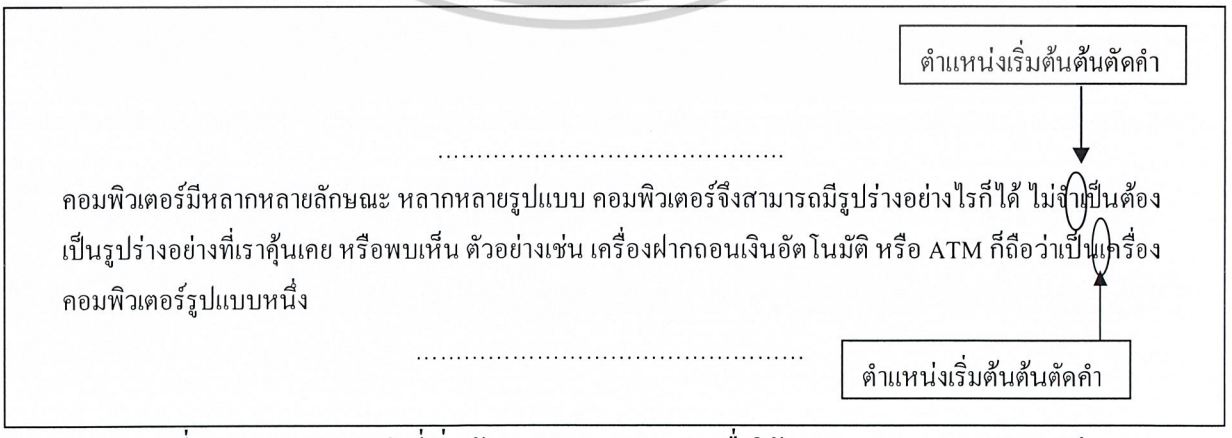


รูปที่ 3.4 แสดงตำแหน่งที่เริ่มต้นขอบเขตการตัดคำ เมื่อใช้ช่องว่างระหว่างคำ

วิธีนี้เป็นวิธีที่ง่าย เพราะพิจารณาเพียงช่องว่างระหว่างประโยคเพียงอย่างเดียว แต่มีข้อเสีย คือ ถ้าประโยคสุดท้ายของบรรทัดมีความยาวมาก ก็จะทำให้ต้องตัดคำเพิ่มมากขึ้น จนอาจจะต้องตัดคำเกือบทั้งบรรทัด ซึ่งบางครั้งอาจไม่มีการเว้นวรรคทั้งบรรทัดเลยก็ได้ นั่นหมายความว่า เราต้องพิจารณาตัดคำหมดทั้งบรรทัด ในกรณีที่ไม่มีเว้นวรรคเลย เป็นการเสียเวลาตัดคำมากเกินจำเป็น วิธีนี้จะใช้ได้ผลดี ก็ต่อเมื่อ ข้อความมีการเว้นวรรคบ่อยๆเท่านั้น

2) พิจารณาตำแหน่งจากอักษรที่อยู่นำหน้าคำ หรืออยู่ท้ายคำเสมอ ให้เลือกตำแหน่งเริ่มตัดคำโดยอาศัยตัวอักษรเฉพาะที่นำหน้าคำ วิธีนี้ได้นำแนวทางมาจากการตัดคำโดยอักษรวิธี ซึ่งพบว่ามีอักษรบางตัวที่จะอยู่นำหน้าของคำเสมอ เช่น สระหน้า เ แ ใ ไ โ จะเป็นอักษรตัวแรกของคำเสมอ แต่ยกเว้น คำบางคำ เช่น ขโมย จเร ชไมพร เป็นต้น นอกจากนั้น อักษรที่อยู่ท้ายของคำเสมอก็สามารถนำมาพิจารณาได้ด้วยเช่นกัน เช่น สระอำ จะอยู่ท้ายของคำเสมอ ดังนั้น ตำแหน่งเริ่มต้นของการตัดคำ คือ

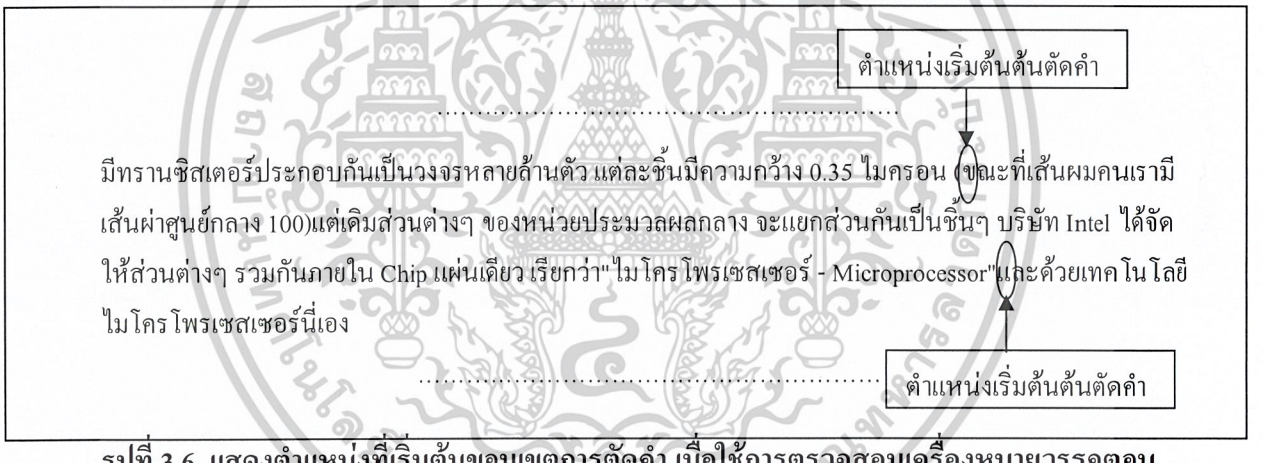
- อักษรที่อยู่หน้าคำเสมอเป็นตำแหน่งแรก
- ตำแหน่งที่อยู่ถัดจากอักษรที่อยู่ท้ายคำเสมอ



รูปที่ 3.5 แสดงตำแหน่งที่เริ่มต้นขอบเขตการตัดคำ เมื่อใช้การตรวจสอบทางอักษรวิธี

วิธีนี้จะช่วยทำสามารถเลือกตำแหน่งได้ใกล้เคียงกับตำแหน่งสิ้นสุดบรรทัดมากขึ้น แต่ยังคงมีข้อเสียอยู่บ้าง เพราะว่า ถ้าข้อความในบรรทัดนั้นไม่มีอักษรที่สามารถตัดได้ คือ ไม่มีทั้งอักษรที่เป็นอักษรตัวแรกของคำเสมอ และไม่มีทั้งอักษรที่เป็นตัวสุดท้ายของคำเสมอ ก็อาจทำให้ต้องตัดคำทุกๆคำในบรรทัดนั้น นอกจากนั้น ยังมีคำยกเว้นบางคำ มีอาจไม่ได้มีตัวอักษรเหล่านี้อยู่นำหน้า หรือ อยู่ข้างท้ายคำเสมอไป จึงอาจทำให้ไม่สามารถพิจารณาคำได้แน่นอน

3) พิจารณาตำแหน่งจากเครื่องหมายวรรคตอนต่างๆ วิธีนี้จะคล้ายคลึงกับการพิจารณาตำแหน่งจากการเว้นวรรค เพียงแต่เปลี่ยนจากเว้นวรรคมาเป็นเครื่องหมายวรรคตอนอื่นๆเท่านั้น เครื่องวรรคตอนเหล่านี้ อย่างเช่น . , ; ? ! “ ( ) ๆ ฯ เป็นต้น (ดูรายละเอียดเครื่องหมายวรรคตอนของไทย ได้ที่ ภาคผนวก ก) เมื่อพบเครื่องหมายวรรคตอนที่อยู่ใกล้สิ้นสุดบรรทัดมากที่สุด ก็จะเลือกตำแหน่งของตัวอักษรที่อยู่ถัดจากเครื่องหมายวรรคตอนนั้น เป็นตำแหน่งเริ่มต้นที่จะพิจารณาการตัดคำ



รูปที่ 3.6 แสดงตำแหน่งที่เริ่มต้นขอบเขตการตัดคำ เมื่อใช้การตรวจสอบเครื่องหมายวรรคตอน

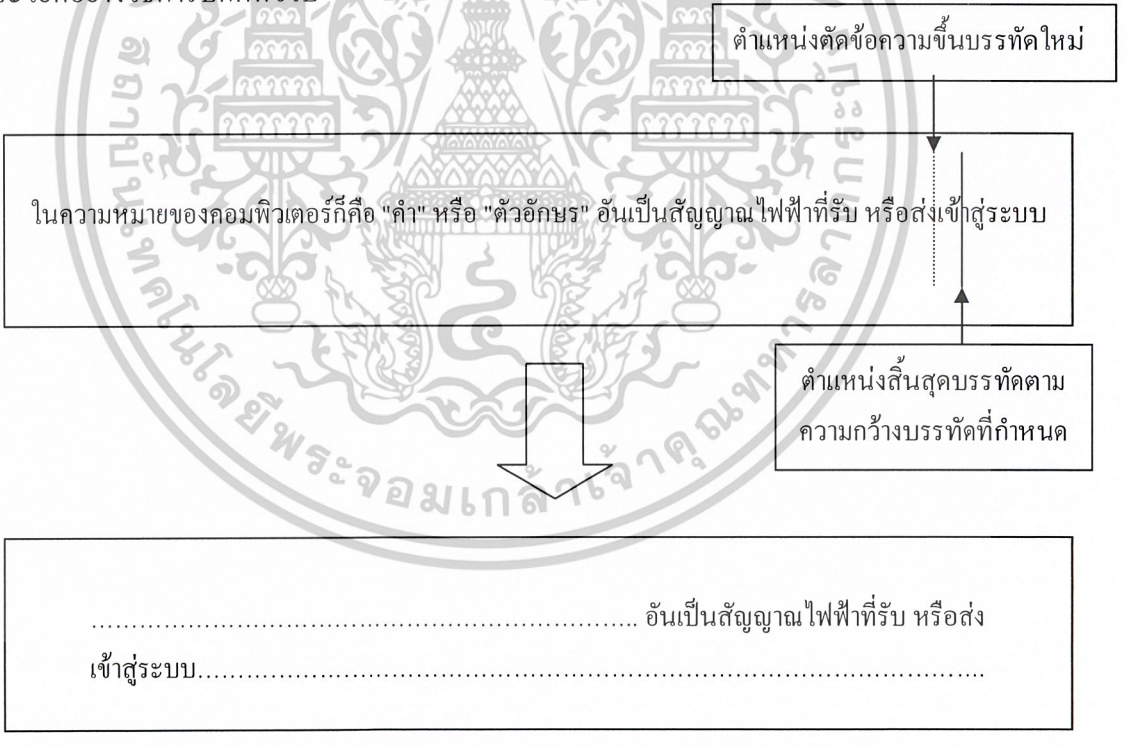
เนื่องจากทั้ง 3 วิธีก็มีข้อดีแตกต่างกัน ดังนั้น การตัดคำในปัญหาพิเศษนี้ จะเลือกพิจารณาขอบเขตเริ่มต้นทั้ง 3 กรณี โดยเลือกตำแหน่งที่ใกล้ตำแหน่งสิ้นสุดบรรทัดมากที่สุดมาเป็นตำแหน่งเริ่มต้น สำหรับวิธีที่ 2 นั้น เมื่อมีคำบางคำที่เป็นข้อยกเว้น สำหรับบางตัวอักษร ดังนั้น เราจะไม่ใช้ตัวอักษรที่มีคำยกเว้นมาพิจารณา จะใช้เฉพาะตัวอักษรที่ต้องอยู่นำหน้าคำ หรือ ตามหลังคำทุกครั้งเท่านั้น เช่น ใ (ไม้ม้วน) จะไม่มีคำใด ที่มีตัวอักษรอยู่หน้า ใ เลย จึงสามารถใช้อักษรตัวนี้ มาเป็นเกณฑ์พิจารณาได้

ในกรณีที่ไม่พบอักขระที่ช่วยในการตัดคำเลย ในช่วงท้ายบรรทัดนั้น เมื่อทำการค้นหาย้อนหลังจากขอบเขตสิ้นสุดบรรทัดมาจนกระทั่งถึง 15 ตัวอักษรแล้ว โดยไม่นับรวมสระบนและสระล่าง ให้ถือว่าขอบเขตเริ่มต้นเป็นอักขระที่นับจากขอบเขตสิ้นสุดมา 15 ตัวอักษรนั้น เนื่องจาก

การตัดคำนั้น จะเป็นการค้นหาคำศัพท์ในข้อความไปเปรียบเทียบกับพจนานุกรม ซึ่งคำที่ค้นหาก็จะอยู่ในช่วงขอบเขตเริ่มต้นถึงขอบเขตสิ้นสุด แต่จากงานวิจัยที่ว่า คำในภาษาไทย จะมีจำนวนตัวอักษรในคำมากที่สุด 15 ตัวอักษร โดยไม่นับรวมสระบนและสระล่าง [5] (ดูข้อมูลอ้างอิงได้ที่ภาคผนวก ค) จึงให้กำหนดขอบเขตเริ่มต้นเป็น 15 ตัวอักษร นับจากขอบเขตสิ้นสุดขึ้นมา

3.4 การออกแบบขั้นตอนวิธีการตัดคำไทยท้ายบรรทัด

การตัดคำไทยในปัญหาพิเศษนี้ จะเป็นการตัดคำเฉพาะตำแหน่งท้ายบรรทัด เพื่อปัดข้อความที่มีความยาวเกินกว่าหนึ่งบรรทัด ให้ไปขึ้นบรรทัดใหม่ แนวคิดของขั้นตอนวิธีในการตัดคำนี้ คือ จะทำการค้นหาคำศัพท์ที่อยู่ใกล้กับตำแหน่งสิ้นสุดบรรทัดมากที่สุด แล้วปัดข้อความที่ต่อท้ายคำศัพท์นั้นไปขึ้นบรรทัดใหม่ทั้งหมด แทนที่จะตัดคำโดยการเดินหน้า จากต้นประโยคด้านซ้ายมือไปท้ายประโยคทางขวามือ ก็จะใช้วิธีการตัดคำจากท้ายประโยคทางขวามือ ซึ่งใกล้กับตำแหน่งสิ้นสุดบรรทัดมากกว่า ย้อนกลับไปทางต้นประโยคด้านซ้ายมือแทน การตัดคำวิธีนี้ จะเป็นการค้นหาคำศัพท์เพียงคำเดียว ซึ่งย่อมจะทำการเปรียบเทียบน้อยกว่าการตัดคำทุกๆคำจากต้นประโยคอย่างวิธีการปกติทั่วไป



รูปที่ 3.7 แสดงการตัดคำไทย ณ ตำแหน่งท้ายบรรทัด และผลลัพธ์ที่ได้หลังจากการตัดคำ

ขั้นตอนวิธีในการตัดคำภาษาไทย ณ ตำแหน่งท้ายบรรทัด สามารถแสดงได้ดังไดอะแกรมดังต่อไปนี้

รูปที่ 3.8 แสดงไดอะแกรมขั้นตอนวิธีการตัดคำไทยท้ายบรรทัด

ขั้นตอนวิธีการตัดคำไทยท้ายบรรทัด จะมีขั้นตอนดังนี้

1) กำหนดข้อมูลนำเข้า โดย

string เป็นข้อความที่จะนำมาตัดคำ

pagewidth เป็นความกว้างของแต่ละบรรทัด

2) หาตำแหน่งสิ้นสุดบรรทัด คือ ตำแหน่งของอักขระตัวสุดท้ายที่สามารถบรรจุอยู่ในบรรทัดนั้นๆ ได้ ถือเป็นขอบเขตสิ้นสุดในการตัดคำ (endmark)

3) หาตำแหน่งเริ่มต้นในการตัดคำ โดยจะนับย้อนจากตำแหน่งสิ้นสุดบรรทัดขึ้นมา การพิจารณาตำแหน่งเริ่มต้นในการตัดคำ (begmark) จะมี 2 แบบ คือ

- หาตำแหน่งที่มีการเว้นวรรค อักขระที่สามารถใช้ตัดคำได้ หรือมีเครื่องหมายวรรคตอน ที่อยู่ใกล้ตำแหน่งสิ้นสุดบรรทัดมากที่สุด
- ถ้าไม่พบเครื่องหมาย หรืออักขระข้างต้น ให้กำหนดตำแหน่งที่นับย้อนจากตำแหน่งสิ้นสุดบรรทัด มา 15 ตัวอักษร

4) ทำการพิจารณาตำแหน่งเพื่อการตัดคำในช่วงที่ 1 (Phase 1) ซึ่งจะเป็นการนำคำศัพท์ในช่วงก่อนจะถึงตำแหน่งสิ้นสุดบรรทัด ไปเปรียบเทียบกับพจนานุกรม

5) หากการพิจารณาตำแหน่งในช่วงที่ 1 พบคำศัพท์ในพจนานุกรม ให้ตำแหน่งอักขระตัวสุดท้ายของบรรทัดนั้น เป็นตำแหน่งของอักขระตัวสุดท้ายของคำศัพท์ที่พบ

6) หากการพิจารณาตำแหน่งในช่วงที่ 1 นั้น ไม่พบคำศัพท์ในพจนานุกรม ให้ทำการพิจารณาตำแหน่งเพื่อการตัดคำในช่วงที่ 2 (Phase 2) ซึ่งจะเป็นการนำคำศัพท์ที่อยู่หลังหรือคร่อมตำแหน่งสิ้นสุดบรรทัด ไปเปรียบเทียบกับพจนานุกรม

7) หากการพิจารณาตำแหน่งในช่วงที่ 2 พบคำศัพท์ในพจนานุกรมแล้ว ให้ตำแหน่งอักขระตัวสุดท้ายของบรรทัดนั้น เป็นตำแหน่งอักขระที่อยู่ก่อนหน้าอักขระตัวแรกของคำศัพท์ที่พบ

8) หากการค้นหาคำศัพท์ในช่วงที่ 2 ก็ไม่พบอีกเช่นกัน ให้ตำแหน่งตัดข้อความขึ้นบรรทัดใหม่เป็นตำแหน่งเริ่มต้นการตัดคำในข้อ 3

สำหรับการพิจารณาตำแหน่งในการตัดคำในช่วงที่ 1 และ 2 จะขออธิบายอย่างละเอียดในหัวข้อต่อๆไป

Thai Word Separation Algorithm

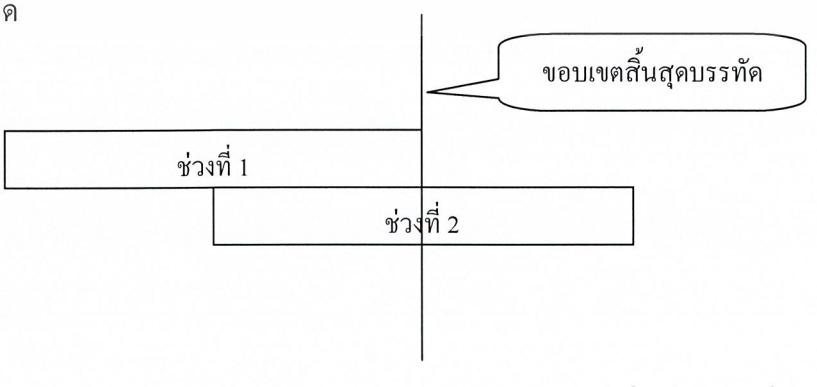
```
endmark = findEndMark(string,pagewidth,i)
if(end of string) {
        breakmark = length of string
        break
        }
```

```
begmark = findBegMark(string,endmark)
if(found english or number character){
        breakmark = begmark
        }
else{
        m1 = separateWord_Phase1(string,begmark,endmark)
        if(m1 == found){
                breakmark = m1
        }else{
                m2 = separateWord_Phase2(string,begmark,endmark)
                if(m2 == found){
                        breakmark = m2
                }else{
                        breakmark = begmark - 1
                        }
                }
        }
```

3.4.1 การตัดคำ ณ ตำแหน่งท้ายบรรทัดของแต่ละบรรทัด

การพิจารณาตำแหน่งการตัดคำในแต่ละบรรทัดนั้น จะใช้วิธีการค้นหาคำศัพท์ที่อยู่ในช่วงก่อนและหลังตำแหน่งสิ้นสุดบรรทัด เพื่อนำไปเปรียบเทียบกับพจนานุกรม ถ้าพบคำศัพท์ในพจนานุกรมแล้ว แสดงว่าพบตำแหน่งที่ใช้ในการตัดคำได้ ซึ่งจะแบ่งเป็น 2 ช่วง ได้แก่

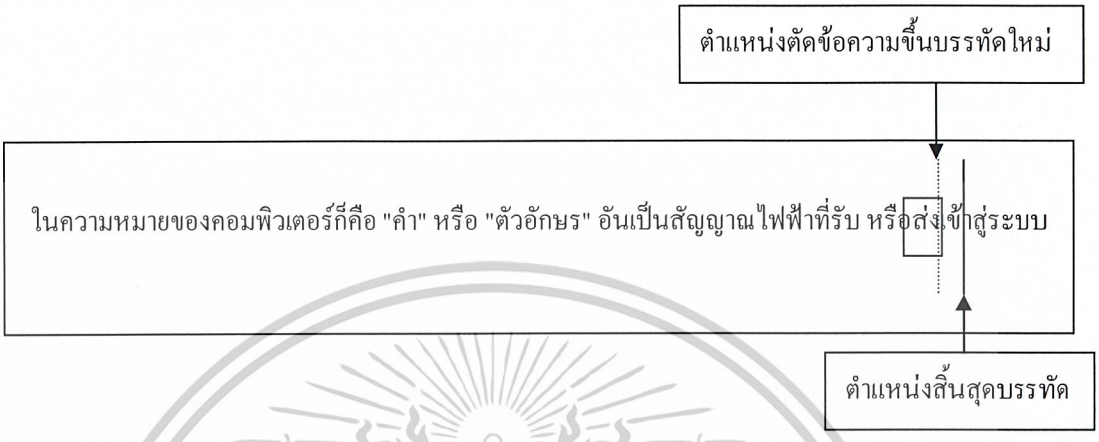
1) การพิจารณาตำแหน่งการตัดคำในช่วงที่ 1 จะพิจารณาคำศัพท์ที่อยู่ก่อนขอบเขตสิ้นสุดบรรทัด
2) การพิจารณาตำแหน่งการตัดคำในช่วงที่ 2 จะพิจารณาคำศัพท์ที่อยู่หลัง และคร่อมขอบเขตสิ้นสุดบรรทัด



รูปที่ 3.9 แสดงขอบเขตการพิจารณาตำแหน่งการตัดคำในช่วงที่ 1 และช่วงที่ 2

3.4.1.1 การพิจารณาตำแหน่งการตัดคำภาษาไทยช่วงที่ 1 (seperateWord_Phase1)

การตัดคำในช่วงแรกนี้ จะเป็นการค้นหาคำศัพท์ที่อยู่ก่อนจะถึงตำแหน่งสิ้นสุดบรรทัดที่ใกล้ท้ายบรรทัดมากที่สุด มาเป็นตำแหน่งที่ใช้ในการตัดข้อความที่เหลือไปขึ้นบรรทัดใหม่ ตัวอย่างดังรูป



รูปที่ 3.10 แสดงคำศัพท์ที่จะพบในการพิจารณาตำแหน่งการตัดคำในช่วงที่ 1

จากรูป จะเห็นว่าเมื่อทำการตัดคำในช่วงที่ 1 แล้ว จะได้ว่า คำว่า “ส่ง” เป็นคำสุดท้ายที่มีความหมายตามพจนานุกรม ก่อนที่จะถึงตำแหน่งสิ้นสุดบรรทัด ดังนั้น ตำแหน่งอักขระตัวสุดท้ายของบรรทัด จะเป็นอักขระตัวสุดท้ายของคำศัพท์ที่พบนั้น ในที่นี้ คือ ตัว “ง” แล้วจะยกข้อความที่เหลือ ที่อยู่ถัดจากคำศัพท์นั้นๆ ไปขึ้นบรรทัดใหม่ทั้งหมด คือจะได้ว่า

..... อันเป็นสัญญาณไฟฟ้าที่รับ หรือส่งเข้าสู่ระบบ.....

การพิจารณาตำแหน่งในช่วงที่ 1 นี้ จะเริ่มขึ้นหลังจากการกำหนดขอบเขตการตัดคำเรียบร้อยแล้ว คือ ทั้งขอบเขตเริ่มต้น และขอบเขตสิ้นสุด และเนื่องจาก ขอบเขตเริ่มต้นของการตัดคำนั้น จะเป็นไปได้อยู่ 2 กรณี ดังนั้น การพิจารณาตำแหน่งการตัดคำในช่วงที่ 1 นี้ ก็อาจแบ่งได้เป็น 2 กรณีเช่นกัน คือ

- กรณีที่ 1 เมื่อไม่พบอักขระที่ช่วยในการตัดคำในช่วงขนาดของคำที่มากที่สุด ซึ่งจะทำให้ขอบเขตของการตัดคำมีความกว้างมากที่สุด เท่ากับขนาดของคำในภาษาไทยที่จะเป็นไปได้ ในที่นี้ ขนาดของคำมากที่สุด จะเป็น 15 ตัวอักษร โดยไม่นับสระบนและสระล่าง
- กรณีที่ 2 เมื่อพบอักขระที่ช่วยในการตัดคำก่อนที่จะถึงช่วงขนาดของคำที่มีความกว้างมากที่สุด ซึ่งจะทำให้ขอบเขตของการตัดคำมีความกว้างต่ำกว่าความกว้างมากที่สุด

กรณีที่ 1 เมื่อไม่พบอักขระที่ช่วยในการตัดคำ และขอบเขตของการตัดคำ มีขนาด 15 ตัวอักษร โดยไม่นับรวมสระบนและสระล่าง

ในกรณีนี้ ถือได้เป็นกรณีที่ต้องมีการเปรียบเทียบคำมากที่สุด ซึ่งจะเกิดขึ้นเมื่อทำการค้นหาตำแหน่งขอบเขตเริ่มต้นแล้ว ไม่พบการเว้นวรรค อักขระที่ช่วยในการกำหนดขอบเขต หรือเครื่องหมายวรรคตอนใดๆ ในช่วงขนาดของคำที่มากที่สุด คือ ในช่วง 15 ตัวอักษรนี้เลย ดังนั้น จึงถือตามสมมติฐานว่า ขนาดของคำภาษาไทยนั้น จะมีความยาวไม่เกิน 15 ตัวอักษร โดยไม่รวมสระบนและสระล่าง วิธีการค้นหาคำศัพท์ จะใช้วิธีการเปรียบเทียบคำที่ยาวที่สุด (Longest Word Matching) ที่พบได้ในพจนานุกรม โดยยึดตำแหน่งสิ้นสุดบรรทัดเป็นหลัก จากนั้น ทำการเปรียบเทียบคำศัพท์ โดยนับอักขระจากตำแหน่งเริ่มต้น (begmark) ถึงตำแหน่งสิ้นสุด (endmark) ถือเป็นคำศัพท์หนึ่งคำ แล้วนำไปค้นหาในพจนานุกรม ถ้าหากไม่พบคำศัพท์คำนั้น ก็จะลดอักขระตัวหน้าสุดของคำลง 1 ตัวอักษร แล้วนำไปเปรียบเทียบในพจนานุกรมอีกครั้ง ทำเช่นนี้ไปเรื่อยๆ โดยลดอักขระตัวหน้าลงทีละ 1 อักขระจนกระทั่งคำศัพท์มีความยาวเพียง 1 ตัวอักษร หากไม่พบคำศัพท์นั้นอีก ก็ให้เลื่อนอักขระด้านท้ายขึ้นมา 1 อักขระ ในขณะที่ตำแหน่งเริ่มต้นจะต้องนับย้อนขึ้นไปอีก 15 ตัวอักษร เป็นขนาดของคำที่กว้างที่สุดเท่าเดิม แล้วค้นหาคำโดยลดอักขระตัวหน้าทีละ 1 อักขระอีก ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งตำแหน่งสิ้นสุดบรรทัดนั้น มีค่าเท่ากับตำแหน่งเริ่มต้นเดิมในครั้งแรก คือ ได้ทำการค้นหาคำศัพท์ทั้งหมดที่เป็นไปได้แล้วในช่วงขอบเขตนั้น หากไม่พบอีก ก็ให้ทำการค้นหาคำศัพท์เพื่อการตัดคำในช่วงที่ 2 ต่อไป

ตัวอย่างการค้นหาตำแหน่งในการตัดคำในช่วงที่ 1 ในกรณีที่ไม่พบอักขระที่ช่วยในการตัดคำ เป็นดังนี้

กำหนดให้ end คือ ตำแหน่งขอบเขตสิ้นสุด
beg คือ ตำแหน่งขอบเขตเริ่มต้น โดย beg = end – 15

รอบที่ 1

ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	beg	ถึง	end	มี 15 ตัวอักษร
ครั้งที่ 2	ช่วงคำศัพท์ ตั้งแต่	beg + 1	ถึง	end	มี 14 ตัวอักษร
ครั้งที่ 3	ช่วงคำศัพท์ ตั้งแต่	beg + 2	ถึง	end	มี 13 ตัวอักษร
		...			
ครั้งที่ 13	ช่วงคำศัพท์ ตั้งแต่	beg + 12	ถึง	end	มี 3 ตัวอักษร
ครั้งที่ 14	ช่วงคำศัพท์ ตั้งแต่	beg + 13	ถึง	end	มี 2 ตัวอักษร
ครั้งที่ 15	ช่วงคำศัพท์ ตั้งแต่	beg + 14	ถึง	end	มี 1 ตัวอักษร

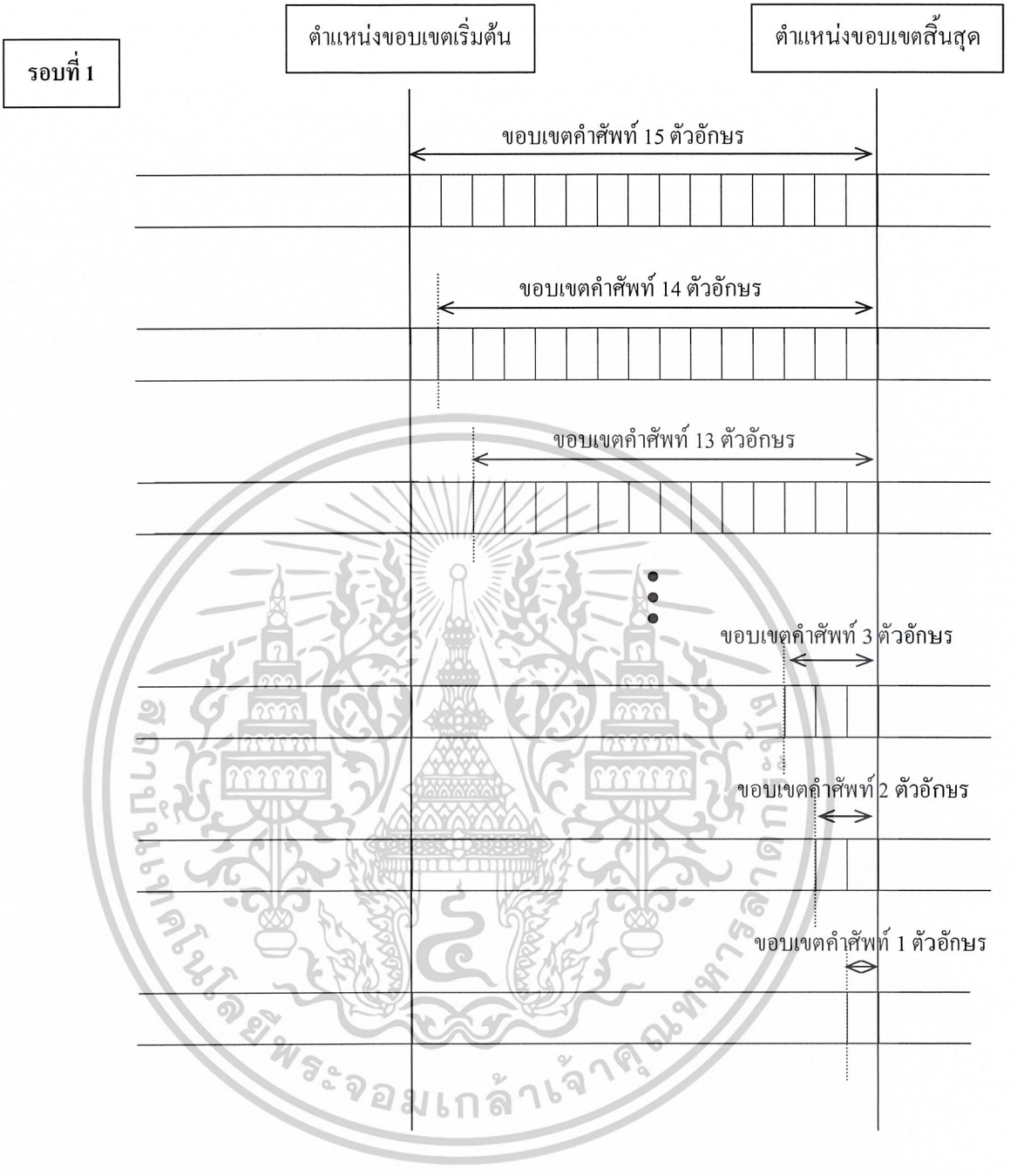
รอบที่ 2

ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	beg - 1	ถึง	end - 1	มี 15 ตัวอักษร
ครั้งที่ 2	ช่วงคำศัพท์ ตั้งแต่	beg	ถึง	end - 1	มี 14 ตัวอักษร
ครั้งที่ 3	ช่วงคำศัพท์ ตั้งแต่	beg + 1	ถึง	end - 1	มี 13 ตัวอักษร
		...			
ครั้งที่ 13	ช่วงคำศัพท์ ตั้งแต่	beg + 11	ถึง	end - 1	มี 3 ตัวอักษร
ครั้งที่ 14	ช่วงคำศัพท์ ตั้งแต่	beg + 12	ถึง	end - 1	มี 2 ตัวอักษร
ครั้งที่ 15	ช่วงคำศัพท์ ตั้งแต่	beg + 13	ถึง	end - 1	มี 1 ตัวอักษร

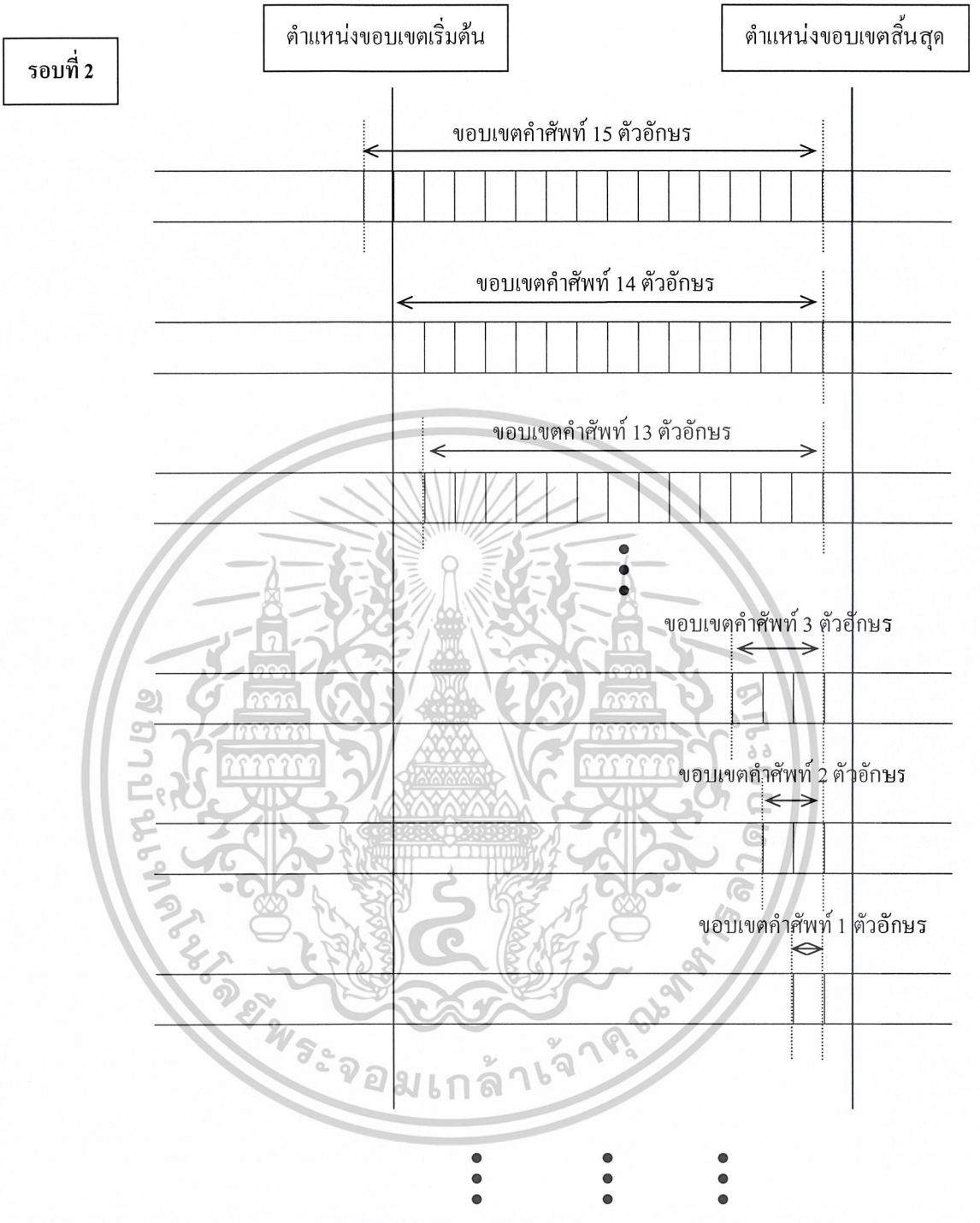
...

รอบที่ 15

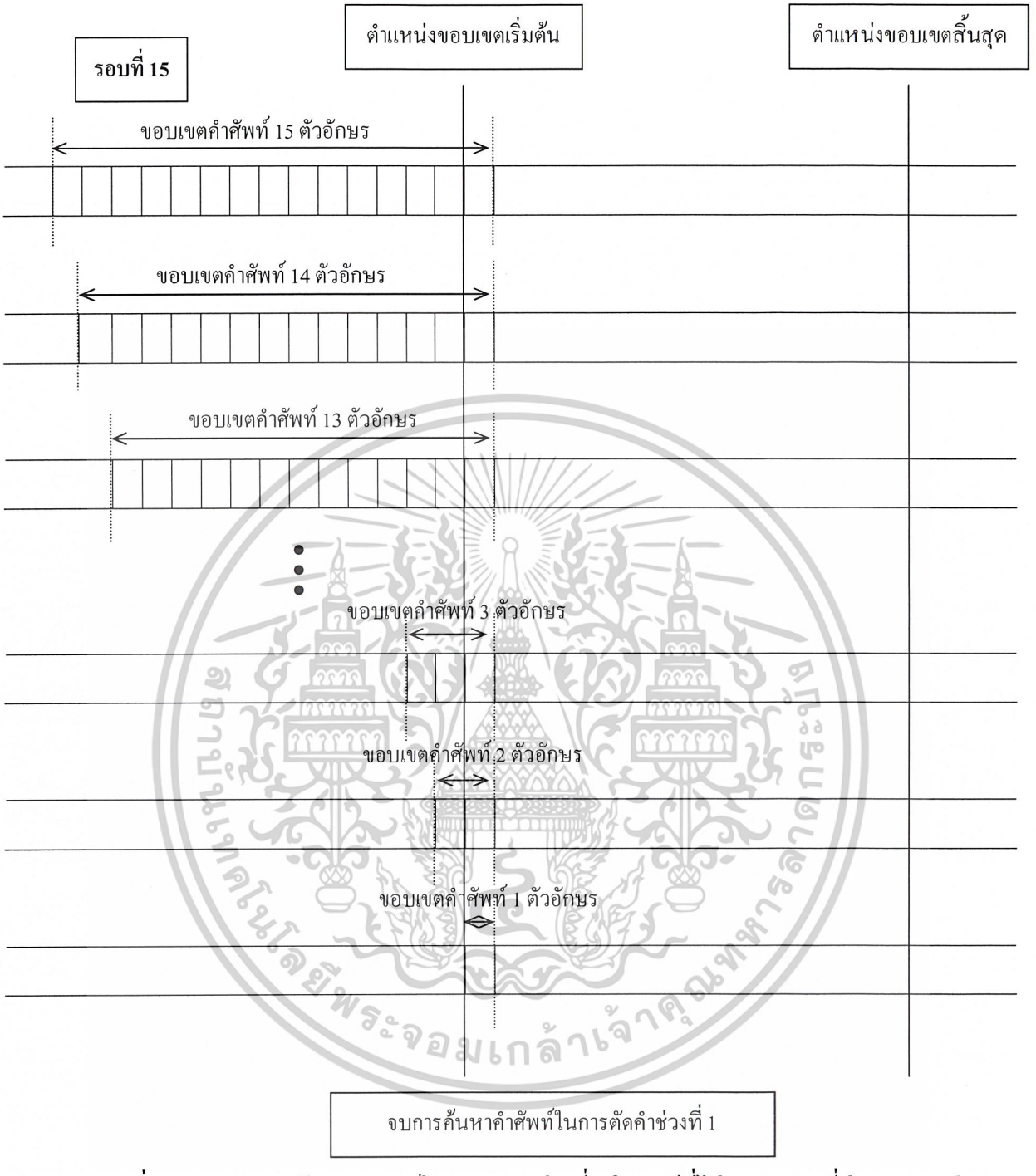
ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	beg - 14	ถึง	end - 14	มี 15 ตัวอักษร
ครั้งที่ 2	ช่วงคำศัพท์ ตั้งแต่	beg - 13	ถึง	end - 14	มี 14 ตัวอักษร
ครั้งที่ 3	ช่วงคำศัพท์ ตั้งแต่	beg - 12	ถึง	end - 14	มี 13 ตัวอักษร
		...			
ครั้งที่ 13	ช่วงคำศัพท์ ตั้งแต่	beg - 2	ถึง	end - 14	มี 3 ตัวอักษร
ครั้งที่ 14	ช่วงคำศัพท์ ตั้งแต่	beg - 1	ถึง	end - 14	มี 2 ตัวอักษร
ครั้งที่ 15	ช่วงคำศัพท์ ตั้งแต่	beg	ถึง	end - 14	มี 1 ตัวอักษร



รูปที่ 3.11 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 1 ในกรณีที่ไม่พบอักขระที่ช่วยตัดคำ



รูปที่ 3.11 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 1 ในกรณีที่ไม่พบอักขระที่ช่วยตัดคำ (ต่อ)



รูปที่ 3.11 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 1 ในกรณีที่ไม่พบอักขระที่ช่วยตัดคำ (ต่อ)

จะสังเกตได้ว่า การค้นหาคำศัพท์ในช่วงที่ 1 ในกรณีนี้จะทำการเปรียบเทียบคำศัพท์ที่ยาวที่สุดก่อน แล้วจึงค่อยๆลดขนาดคำศัพท์ให้เล็กลงเรื่อยๆ เมื่อลดขนาดคำศัพท์จนเหลือเพียง 1 ตัวอักษรแล้ว ก็จะเริ่มทำซ้ำในรอบใหม่ โดยเลื่อนช่วงคำศัพท์ขึ้นไปเรื่อย จนกว่าจะพบคำศัพท์ในพจนานุกรม หรือถ้าหากทำครบทั้ง 15 รอบแล้ว ก็ยังไม่พบคำศัพท์อีก ก็ให้ทำการค้นหาคำศัพท์ในการตัดคำช่วงที่ 2 ต่อไป

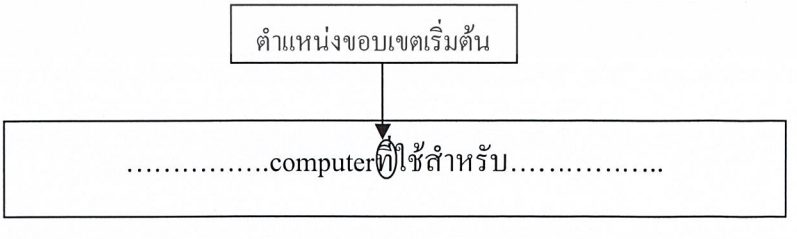
กรณีที่ 2 เมื่อพบอักขระที่ช่วยในการตัดคำ และขอบเขตของการตัดคำ มีขนาดต่ำกว่า 15 ตัวอักษร โดยไม่นับรวมสระบนและสระล่าง

กรณีนี้จะคล้ายคลึงกับกรณีที่ 1 เพียงแต่จะมีการพบเครื่องหมายเว้นวรรค อักขระที่ช่วยในการตัดคำ เช่น ไม้ม้วน ไม้ยมก ไปยาลน้อย หรือเครื่องหมายวรรคตอนอื่นๆ ทั้งนี้รวมไปถึงอักขระประเภทอื่นๆที่ไม่ใช่ภาษาไทยด้วย อักขระหรือเครื่องหมายเหล่านี้ ได้แทรกอยู่ในช่วงขอบเขต 15 ตัวอักษร ซึ่งสามารถนำอักขระและเครื่องหมายเหล่านี้มาเป็นหลัก เพื่อกำหนดขอบเขตเริ่มต้นได้

อักขระและเครื่องหมายต่างๆที่สามารถนำมาใช้ช่วยในการตัดคำ มีดังนี้

1) เครื่องหมายเว้นวรรค จะสามารถกำหนดขอบเขตเริ่มต้น ณ ตำแหน่งถัดจากเครื่องหมายเว้นวรรคได้
2) อักขระภาษาไทยบางตัว เช่น
   - ไม้ม้วน ตำแหน่งขอบเขตเริ่มต้น คือ ตำแหน่งของอักษรไม้ม้วน
   - ไม้ยมก ตำแหน่งขอบเขตเริ่มต้น คือ ตำแหน่งที่อยู่ถัดจากอักขระไม้ยมก
   - ไปยาลน้อย ตำแหน่งขอบเขตเริ่มต้น คือ ตำแหน่งที่อยู่ถัดจากอักขระไปยาลน้อย
3) เครื่องหมายวรรคตอน สามารถกำหนดขอบเขตเริ่มต้น ณ ตำแหน่งถัดจากเครื่องหมายวรรคตอนนั้น
4) ตัวเลข ขอบเขตเริ่มต้นจะอยู่ถัดจากตัวเลขตัวสุดท้ายที่พบ

5) อักษรในภาษาอื่นๆ เช่น ภาษาอังกฤษ ขอบเขตเริ่มต้นจะอยู่ถัดจากอักษรตัวสุดท้ายที่พบ



เมื่อได้ขอบเขตเริ่มต้นและขอบเขตสิ้นสุดแล้ว ก็จะเริ่มทำการค้นหาคำศัพท์เช่นเดียวกับในกรณีที่ 1 กล่าวคือ จะเริ่มจากคำศัพท์ที่เริ่มจากอักขระ ณ ตำแหน่งเริ่มต้นไปจนถึงตำแหน่งสุดท้าย ซึ่งความยาวของคำศัพท์นั้นจะมีความยาวต่ำกว่า 15 ตัวอักษร ถ้าไม่พบคำศัพท์คำนั้นในพจนานุกรม

ก็จะลดอักขระตัวแรกลงทีละตัว จนกระทั่งเหลือเพียง 1 ตัวอักษร หากไม่พบคำศัพท์อีก ก็จะเลื่อนขอบเขตสุดท้ายของคำศัพท์ขึ้นไป 1 ตัวอักษร แต่ตำแหน่งเริ่มต้นของคำศัพท์จะย้อนกลับไปเป็นตำแหน่งเดิมในครั้งแรก ซึ่งจะแตกต่างกับในกรณีแรก ที่คำศัพท์ที่เริ่มค้นหาในแต่ละรอบจะมีความยาวเป็น 15 ตัวอักษรเสมอ แต่ในกรณีนี้ ความยาวของคำศัพท์ที่เริ่มต้นค้นหาในแต่ละรอบ จะค่อยๆ ลดลงเรื่อยๆ นั่นคือ จะมีการค้นหาคำศัพท์น้อยครั้งกว่ากรณีแรก

ตัวอย่างการค้นหาตำแหน่งในการตัดคำในช่วงที่ 1 ในกรณีที่พบอักขระที่ช่วยในการตัดคำเป็นดังนี้

กำหนดให้ end คือ ตำแหน่งขอบเขตสิ้นสุด

beg คือ ตำแหน่งขอบเขตเริ่มต้น โดย beg = end – n เมื่อ $n \leq 15$

(ในที่นี้ สมมติให้ n = 10)

รอบที่ 1

ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	beg	ถึง	end	มี 10 ตัวอักษร
ครั้งที่ 2	ช่วงคำศัพท์ ตั้งแต่	beg + 1	ถึง	end	มี 9 ตัวอักษร
ครั้งที่ 3	ช่วงคำศัพท์ ตั้งแต่	beg + 2	ถึง	end	มี 8 ตัวอักษร
		...			
ครั้งที่ 8	ช่วงคำศัพท์ ตั้งแต่	beg + 7	ถึง	end	มี 3 ตัวอักษร
ครั้งที่ 9	ช่วงคำศัพท์ ตั้งแต่	beg + 8	ถึง	end	มี 2 ตัวอักษร
ครั้งที่ 10	ช่วงคำศัพท์ ตั้งแต่	beg + 9	ถึง	end	มี 1 ตัวอักษร

รอบที่ 2

ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	beg	ถึง	end - 1	มี 9 ตัวอักษร
ครั้งที่ 2	ช่วงคำศัพท์ ตั้งแต่	beg + 1	ถึง	end - 1	มี 8 ตัวอักษร
ครั้งที่ 3	ช่วงคำศัพท์ ตั้งแต่	beg + 2	ถึง	end - 1	มี 7 ตัวอักษร
		...			
ครั้งที่ 7	ช่วงคำศัพท์ ตั้งแต่	beg + 6	ถึง	end - 1	มี 3 ตัวอักษร
ครั้งที่ 8	ช่วงคำศัพท์ ตั้งแต่	beg + 7	ถึง	end - 1	มี 2 ตัวอักษร
ครั้งที่ 9	ช่วงคำศัพท์ ตั้งแต่	beg + 8	ถึง	end - 1	มี 1 ตัวอักษร

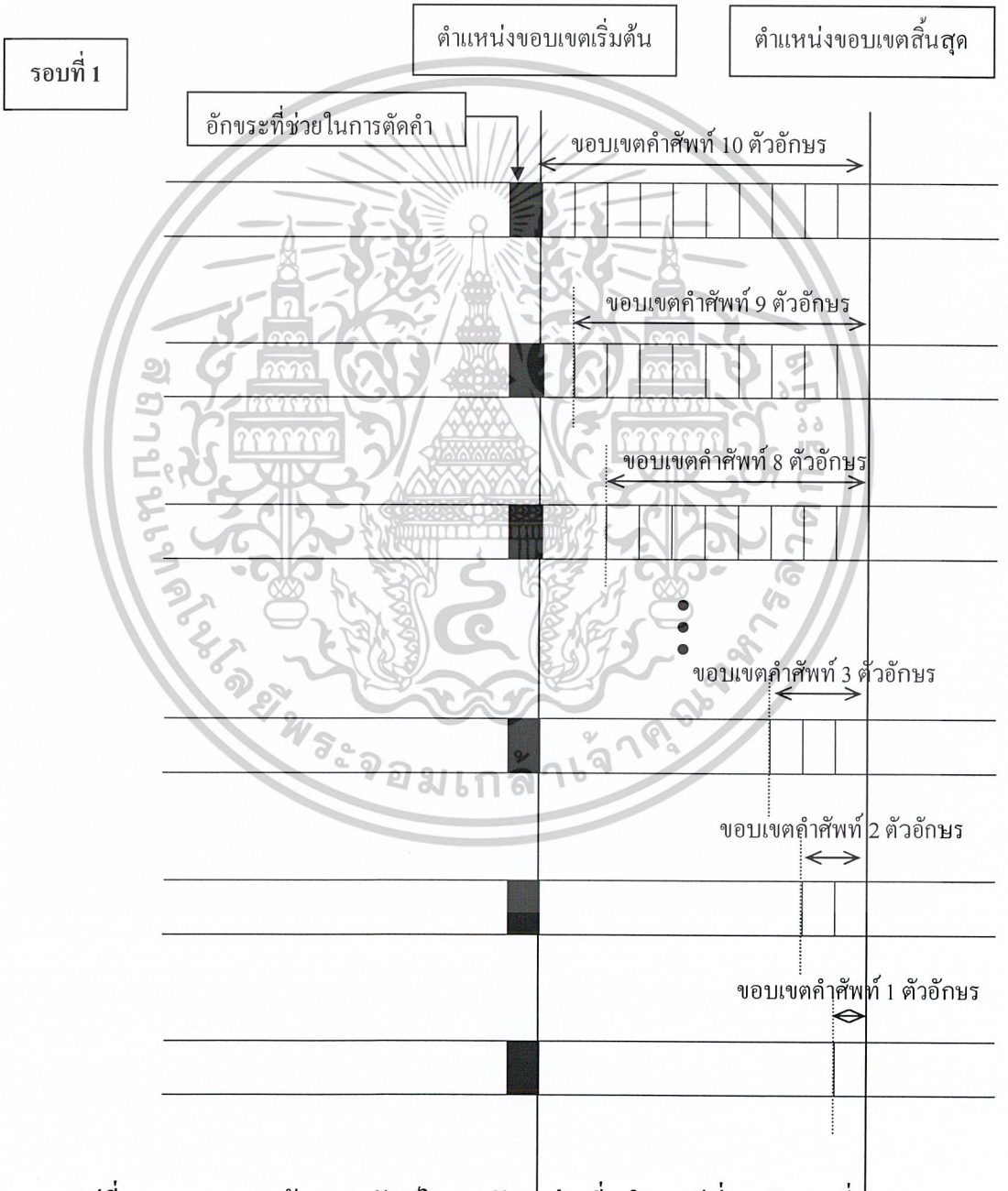
...

รอบที่ 9

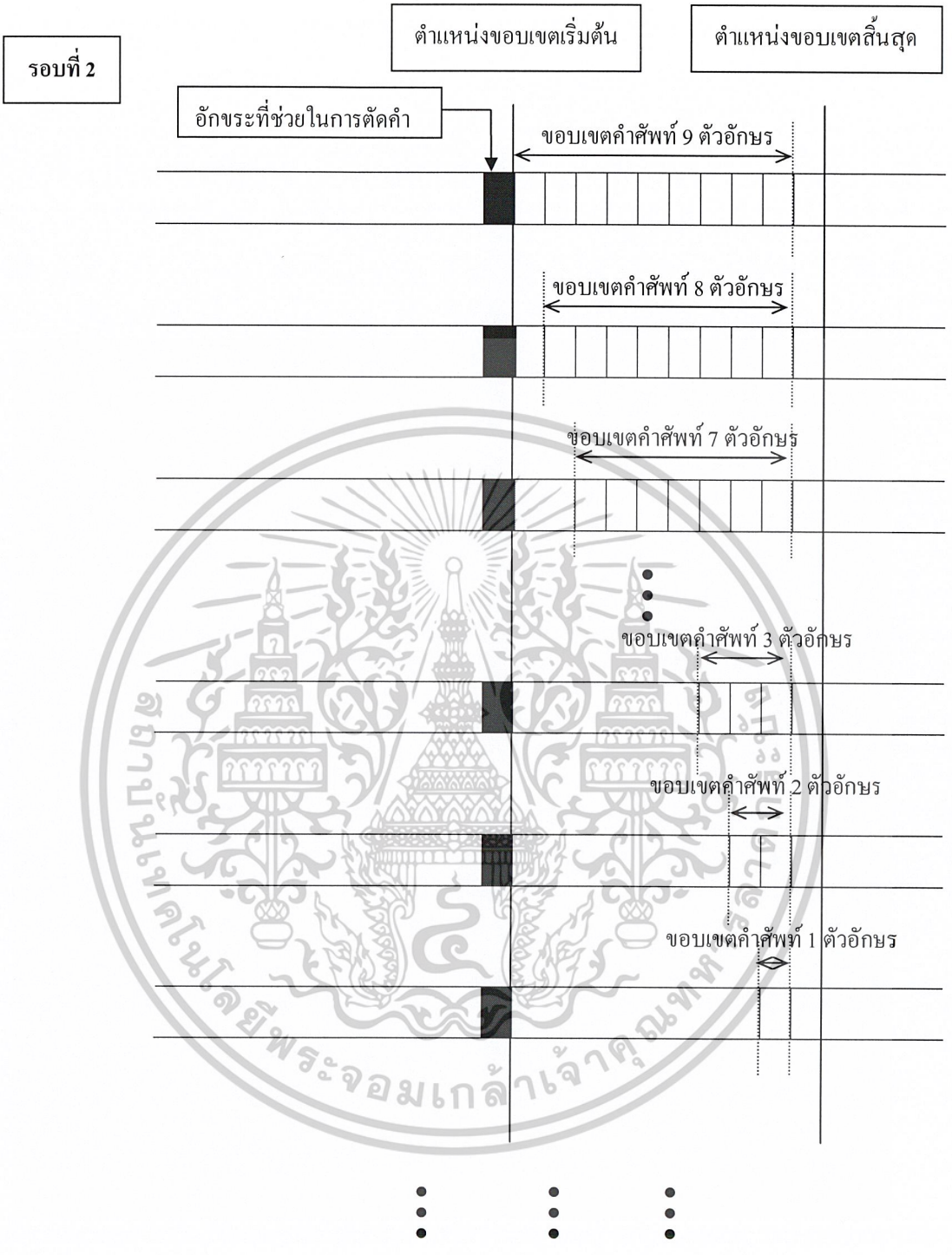
ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	beg	ถึง	end - 8	มี 2 ตัวอักษร
ครั้งที่ 2	ช่วงคำศัพท์ ตั้งแต่	beg + 1	ถึง	end - 8	มี 1 ตัวอักษร

รอบที่ 10

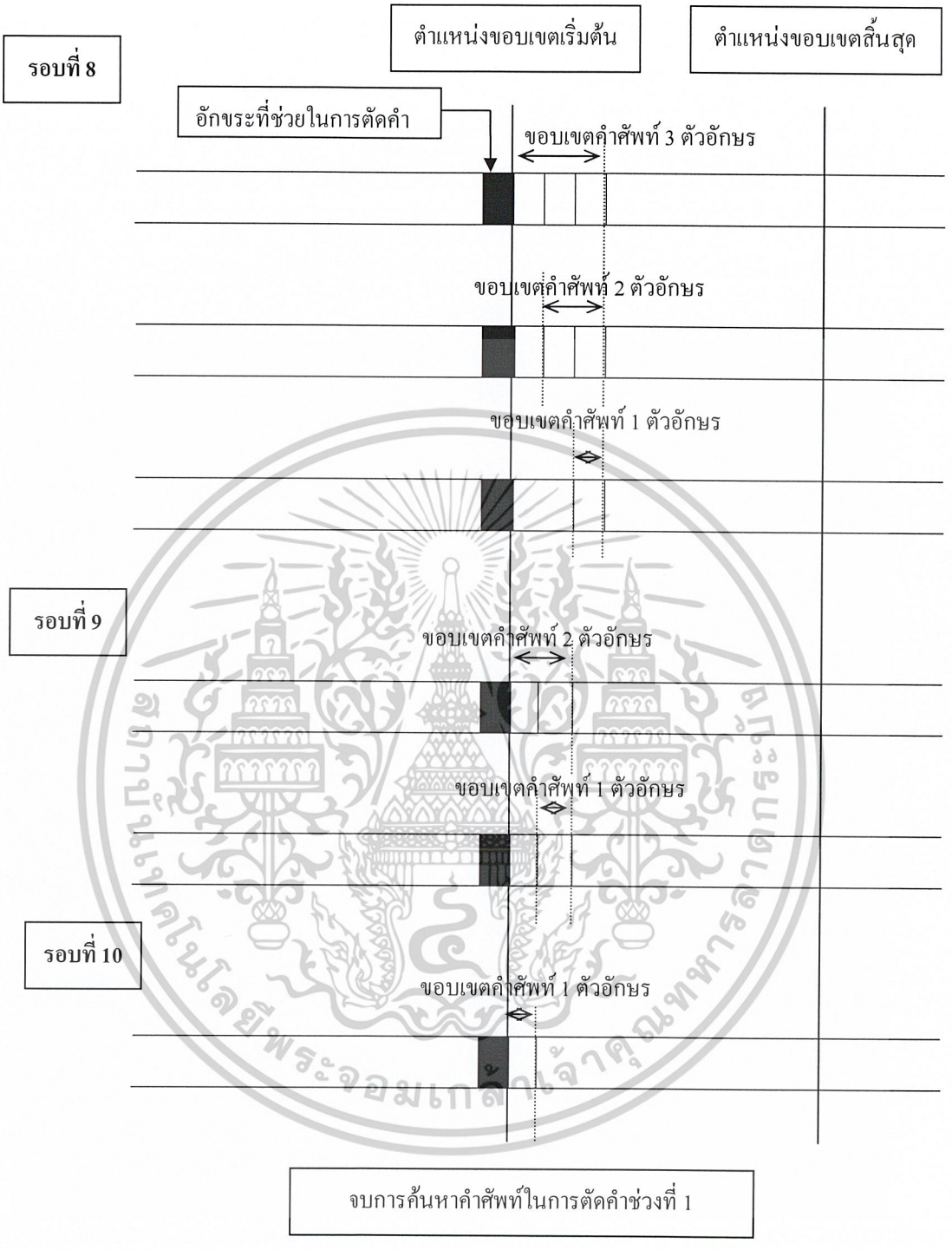
ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	beg	ถึง	end - 9	มี 1 ตัวอักษร



รูปที่ 3.12 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 1 ในกรณีที่พบอักขระที่ช่วยตัดคำ



รูปที่ 3.12 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 1 ในกรณีที่พบอักขระที่ช่วยตัดคำ (ต่อ)



รูปที่ 3.12 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 1 ในกรณีที่พบอักขระที่ช่วยตัดคำ (ต่อ)

การค้นหาคำศัพท์ในกรณีนี้ จะทำการเปรียบเทียบคำศัพท์น้อยครั้งกว่าในกรณีแรก เพราะขอบเขตเริ่มต้นด้านหน้านั้นจะมีตำแหน่งคงที่ และความยาวของคำศัพท์ที่ทำการค้นหา จะมีความยาวต่ำกว่า 15 ตัวอักษรเสมอ โดยไม่นับรวมสระบนและสระล่าง และเมื่อเปรียบเทียบคำศัพท์ทั้งหมดแล้ว ยังไม่พบคำนั้นในพจนานุกรมอีก ก็ให้ทำการค้นหาในการตัดคำช่วงที่ 2 ต่อไป

ตัวอย่างขั้นตอนวิธีในการค้นหาตำแหน่งในการตัดคำช่วงที่ 1 เป็นดังนี้

Separate Word Phase 1 Algorithm

```
success = false
begmark = oldbegmark
endmark = oldendmark
num = 0
while((success = false)  &&  (endmark != oldbegmark)){
        word = string.substr(begmark,endmark)
        while((wordlength > 0) && (not found word in Dict)){
                begmark++
                word = string.substr(begmark,endmark)
                }
        if(wordlength == 0){
                endmark--
                if(length of word is MAX){
                        num++
                        begmark = oldbegmark - num
                }else{
                        begmark = oldbegmark
                        }
        }else{
                success = true
                }
        }
if(success == true)         {mark = endmark}
else                        {mark = not found}
```

3.4.1.2 การพิจารณาตำแหน่งการตัดคำภาษาไทยช่วงที่ 2 (seperateWord_Phase2)

การตัดคำในช่วงที่ 2 จะเกิดขึ้นก็ต่อเมื่อการค้นหาคำศัพท์ในการตัดคำช่วงที่ 1 นั้นไม่ประสบผลสำเร็จ คือ ไม่สามารถค้นพบคำศัพท์ใดๆเลยในช่วงขอบเขตที่กำหนด ซึ่งอาจสันนิษฐานได้ว่า คำศัพท์ที่มีความหมายนั้น ได้คร่อมตำแหน่งขอบเขตสิ้นสุดบรรทัดอยู่ จึงได้มาทำการค้นหาคำศัพท์ที่อยู่คร่อมตำแหน่งสิ้นสุดบรรทัดที่ใกล้ท้ายบรรทัดมากที่สุดแทน โดยอักขระเริ่มต้นของคำศัพท์จะอยู่ในขอบเขต แต่อักขระสุดท้ายจะอยู่เลยขอบเขตสิ้นสุดไปแล้ว ซึ่งถ้าหาพบว่ามีคำศัพท์ในตำแหน่งเช่นนั้น จะถือว่าตำแหน่งที่เป็นอักขระเริ่มต้นของคำจะเป็นตำแหน่งเริ่มต้นของข้อความที่เหลือที่จะต้องไปขึ้นบรรทัดใหม่ ตัวอย่างดังรูป

รูปที่ 3.13 แสดงคำศัพท์ที่จะพบในการตัดคำในช่วงที่ 2

จากรูป จะเห็นว่า เมื่อทำการตัดคำในช่วงที่ 2 แล้ว จะได้ว่า คำว่า “เข้า” เป็นคำแรกที่มีความหมายตามพจนานุกรม ในตำแหน่งที่อยู่คร่อมตำแหน่งสิ้นสุดบรรทัด ดังนั้น ตำแหน่งอักขระตัวสุดท้ายของบรรทัด จะเป็นอักขระตัวที่อยู่ก่อนหน้าอักขระตัวแรกของคำศัพท์ที่พบนั้น ในที่นี้เมื่อคำศัพท์ที่พบ คือ “เข้า” และอักขระตัวแรกของคำศัพท์ คือ “เ” ดังนั้น อักขระที่อยู่ก่อนหน้าตัว “เ” คือ ตัว “ง” จากนั้นจะยกข้อความที่นับจากคำศัพท์ที่พบ ไปขึ้นบรรทัดใหม่ทั้งหมด คือ ประโยค “เข้า...” จะต้องไปขึ้นบรรทัดใหม่ ซึ่งจะได้ว่า

…….. อันเป็นสัญญาณไฟฟ้าที่รับ หรือส่งเข้าสู่ระบบ……..

การตัดคำในช่วงที่ 2 จะเริ่มขึ้นหลังจากตัดคำในช่วงที่ 1 เรียบร้อยแล้ว แต่ก็ยังค้นหาคำศัพท์ไม่พบ จึงต้องเข้ามาทำการค้นหาคำศัพท์ในช่วงที่ 2 ซึ่งในช่วงที่ 2 นี้จะทำการค้นหาคำที่อยู่คร่อมตำแหน่งสิ้นสุด โดยเริ่มจากการค้นหาคำศัพท์ ในอักขระตำแหน่งขอบเขตสิ้นสุดบรรทัด ไป

จนกระทั่งคำศัพท์นั้นมีความยาวมากที่สุดเท่าที่จะเป็นไปได้ ซึ่งก็คือ มีความยาว 15 ตัวอักษร โดยไม่รวมสระบนและสระล่าง หากไม่พบคำนั้นในพจนานุกรม ก็จะลดอักขระตัวสุดท้ายลงทีละ 1 ตัวอักษร จนกระทั่งตำแหน่งอักขระตัวสุดท้ายนั้นอยู่ ณ ตำแหน่งถัดจากขอบเขตสิ้นสุด หากไม่พบคำศัพท์ในพจนานุกรมเลย ก็จะเลื่อนขอบเขตเริ่มต้นของคำศัพท์ย้อนกลับไป 1 อักขระ และค้นหาคำศัพท์ที่มีความยาวมากที่สุด 15 ตัวอักษรอีกครั้ง แล้วลดอักขระตัวสุดท้ายทีละตัว ทำเช่นนี้ไปเรื่อยๆจนกระทั่งพบคำศัพท์ในพจนานุกรม หรือถ้าหากไม่พบคำศัพท์ใดๆเลย จนอักขระเริ่มต้นของคำศัพท์ที่ได้ทำการค้นหา ได้มาอยู่ตำแหน่งเดียวกับขอบเขตเริ่มต้นในการตัดคำช่วงที่ 1 แล้ว แสดงว่า ไม่มีคำศัพท์คำใดเลยที่สามารถใช้การตัดคำทั้ง 2 ช่วงพบ

ตัวอย่างการค้นหาตำแหน่งในการตัดคำในช่วงที่ 1 ในกรณีที่พบอักขระที่ช่วยในการตัดคำเป็นดังนี้

กำหนดให้ end คือ ตำแหน่งขอบเขตสิ้นสุด

beg คือ ตำแหน่งขอบเขตเริ่มต้น โดย $beg = end - n$ เมื่อ $n \leq 15$

รอบที่ 1

ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	end	ถึง	end + 15	มี 15 ตัวอักษร
ครั้งที่ 2	ช่วงคำศัพท์ ตั้งแต่	end	ถึง	end + 14	มี 14 ตัวอักษร
ครั้งที่ 3	ช่วงคำศัพท์ ตั้งแต่	end	ถึง	end + 13	มี 13 ตัวอักษร
		...			
ครั้งที่ 13	ช่วงคำศัพท์ ตั้งแต่	end	ถึง	end + 3	มี 3 ตัวอักษร
ครั้งที่ 14	ช่วงคำศัพท์ ตั้งแต่	end	ถึง	end + 2	มี 2 ตัวอักษร
ครั้งที่ 15	ช่วงคำศัพท์ ตั้งแต่	end	ถึง	end + 1	มี 1 ตัวอักษร

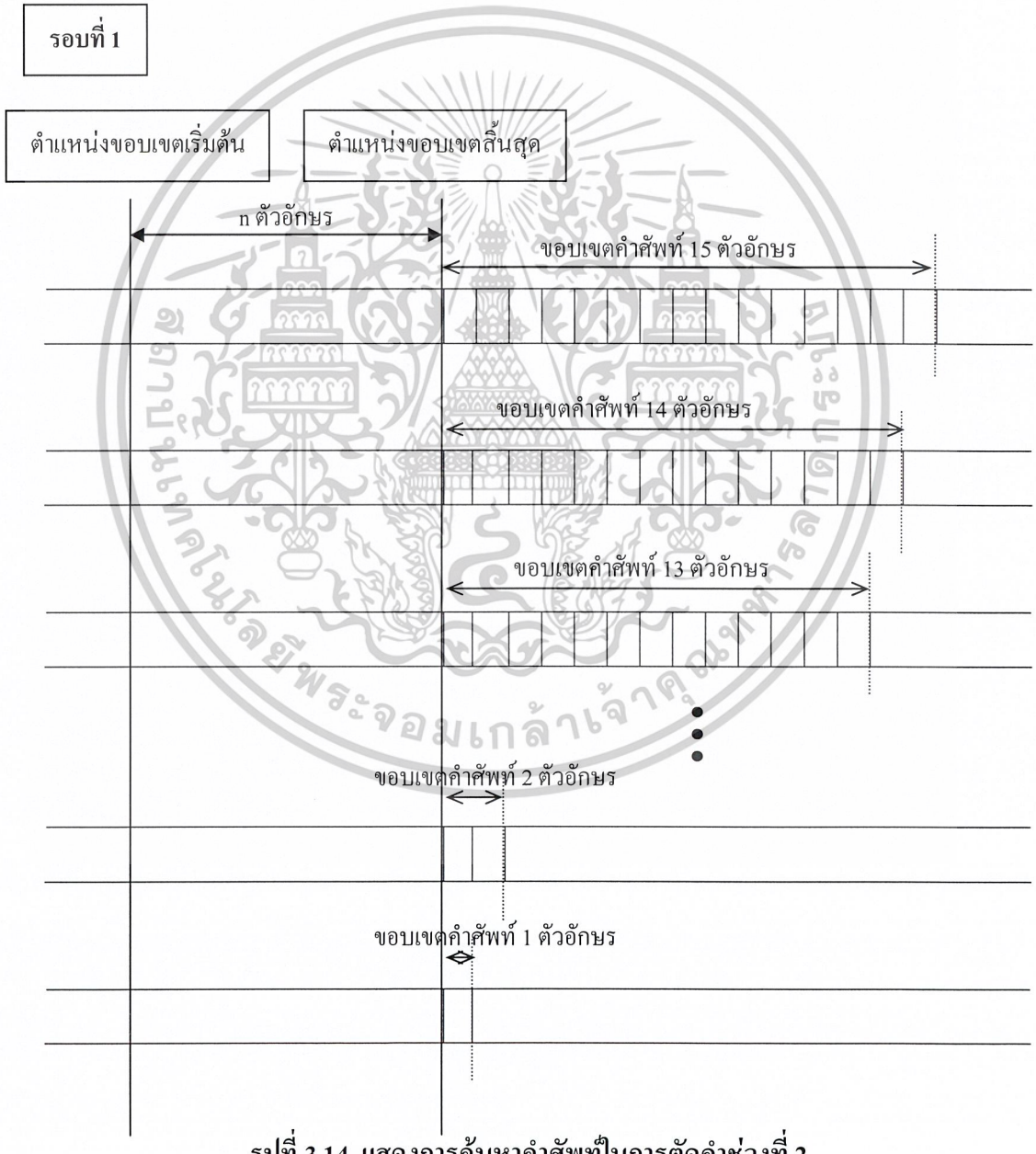
รอบที่ 2

ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	end - 1	ถึง	end + 14	มี 15 ตัวอักษร
ครั้งที่ 2	ช่วงคำศัพท์ ตั้งแต่	end - 1	ถึง	end + 13	มี 14 ตัวอักษร
ครั้งที่ 3	ช่วงคำศัพท์ ตั้งแต่	end - 1	ถึง	end + 12	มี 13 ตัวอักษร
		...			
ครั้งที่ 13	ช่วงคำศัพท์ ตั้งแต่	end - 1	ถึง	end + 2	มี 3 ตัวอักษร
ครั้งที่ 14	ช่วงคำศัพท์ ตั้งแต่	end - 1	ถึง	end + 1	มี 2 ตัวอักษร

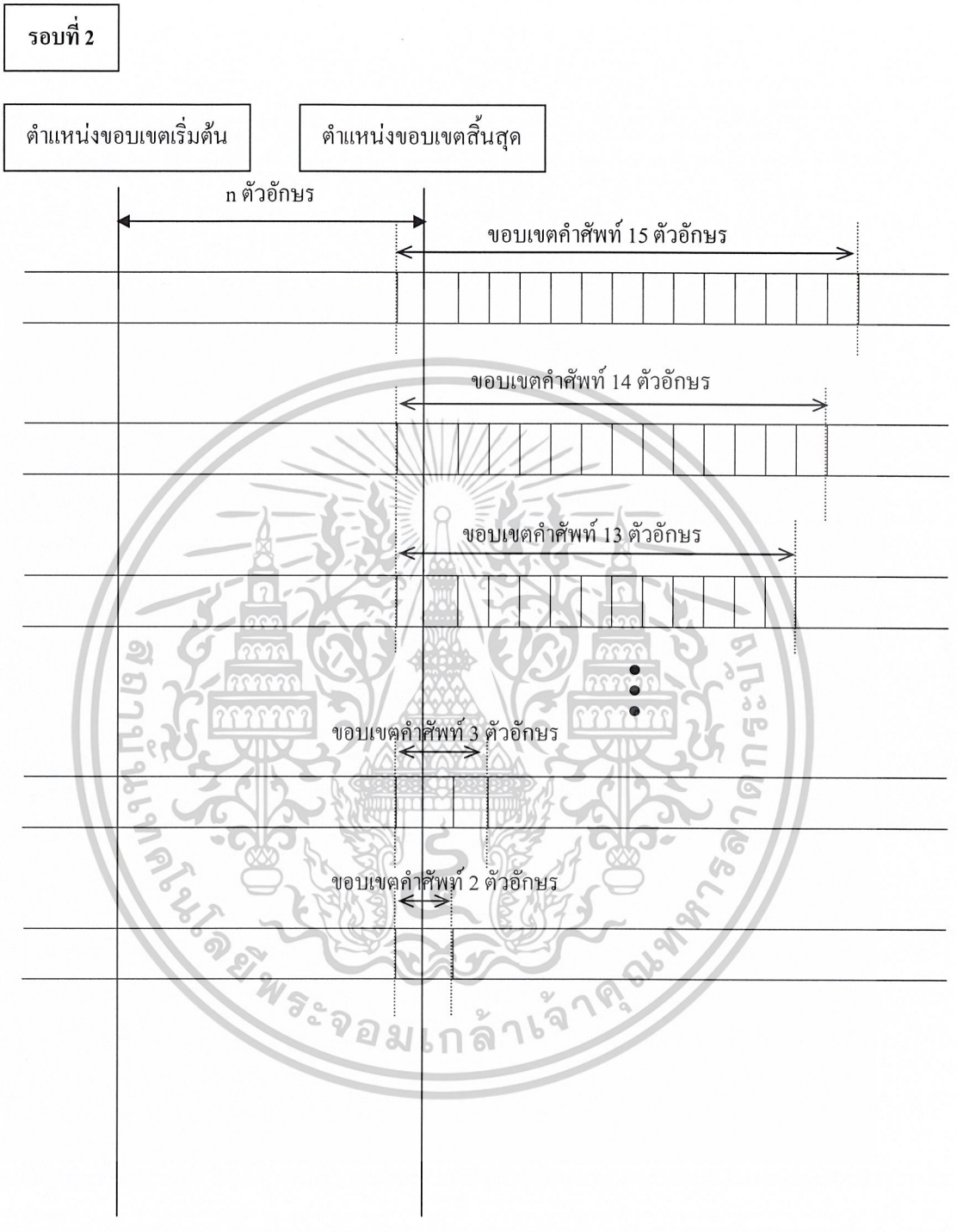
...

รอบที่ n

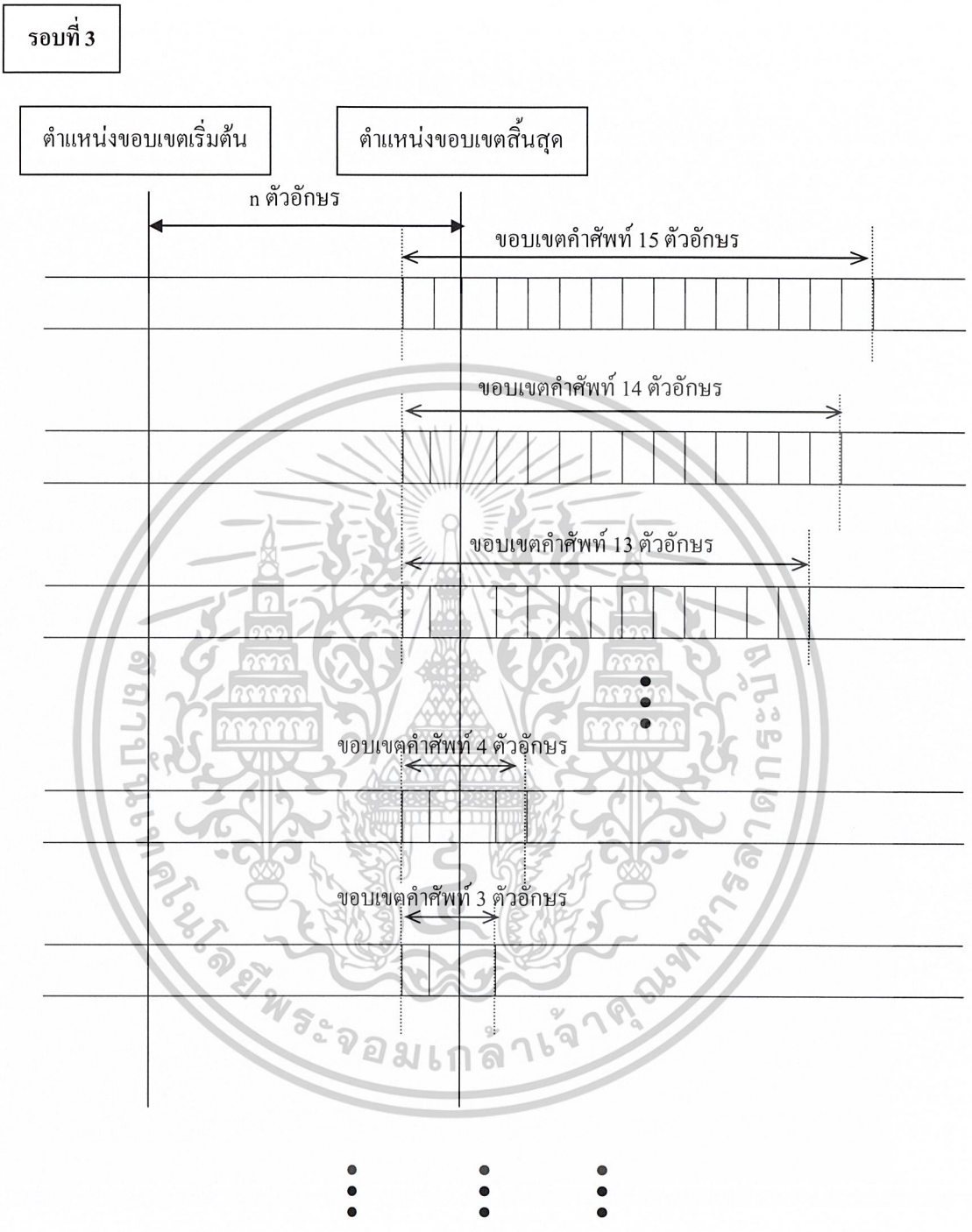
ครั้งที่ 1	ช่วงคำศัพท์ ตั้งแต่	end – (n + 1)	ถึง	end + (15 – n + 1)	มี 15 ตัวอักษร
ครั้งที่ 2	ช่วงคำศัพท์ ตั้งแต่	end – (n + 1)	ถึง	end + (14 – n + 1)	มี 14 ตัวอักษร
ครั้งที่ 3	ช่วงคำศัพท์ ตั้งแต่	end – (n + 1)	ถึง	end + (13 – n + 1)	มี 13 ตัวอักษร
		...			
ครั้งที่ 14 - n	ช่วงคำศัพท์ ตั้งแต่	end – (n + 1)	ถึง	end + 2	มี n + 1 ตัวอักษร
ครั้งที่ 15 - n	ช่วงคำศัพท์ ตั้งแต่	end – (n + 1)	ถึง	end + 1	มี n ตัวอักษร



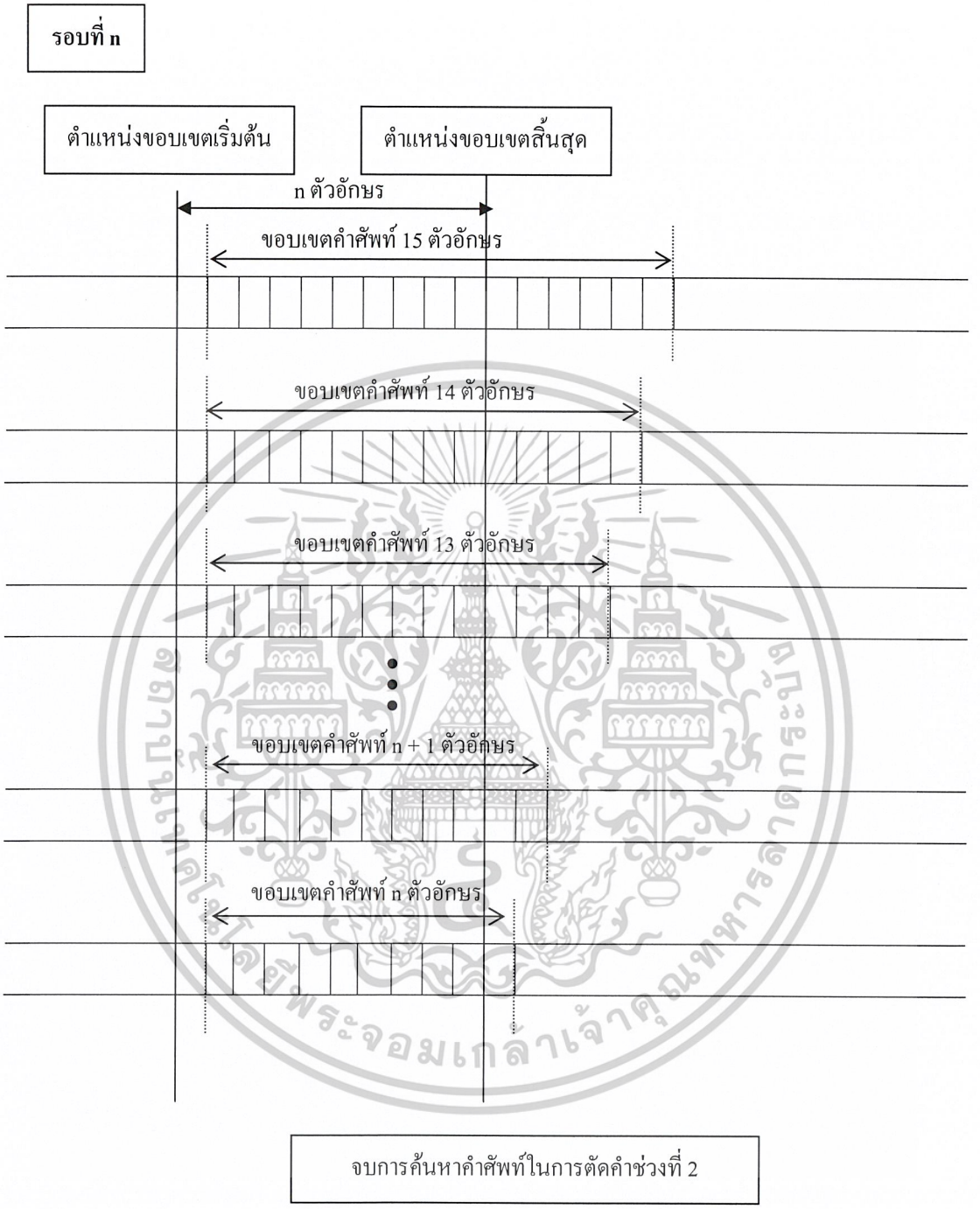
รูปที่ 3.14 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 2



รูปที่ 3.14 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 2 (ต่อ)



รูปที่ 3.14 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 2 (ต่อ)



รูปที่ 3.14 แสดงการค้นหาคำศัพท์ในการตัดคำช่วงที่ 2 (ต่อ)

สำหรับการค้นหาคำศัพท์ในช่วงที่ 2 นี้ ถ้าหากพบคำศัพท์ใดๆในพจนานุกรม แสดงว่าคำๆ นั้นเป็นคำแรกที่พบในบรรทัดถัดไป ดังนั้น ตำแหน่งอักขระตัวสุดท้ายของบรรทัดปัจจุบันนี้ ก็คือ ตำแหน่งอักขระที่อยู่ก่อนหน้าคำศัพท์คำนี้นั่นเอง

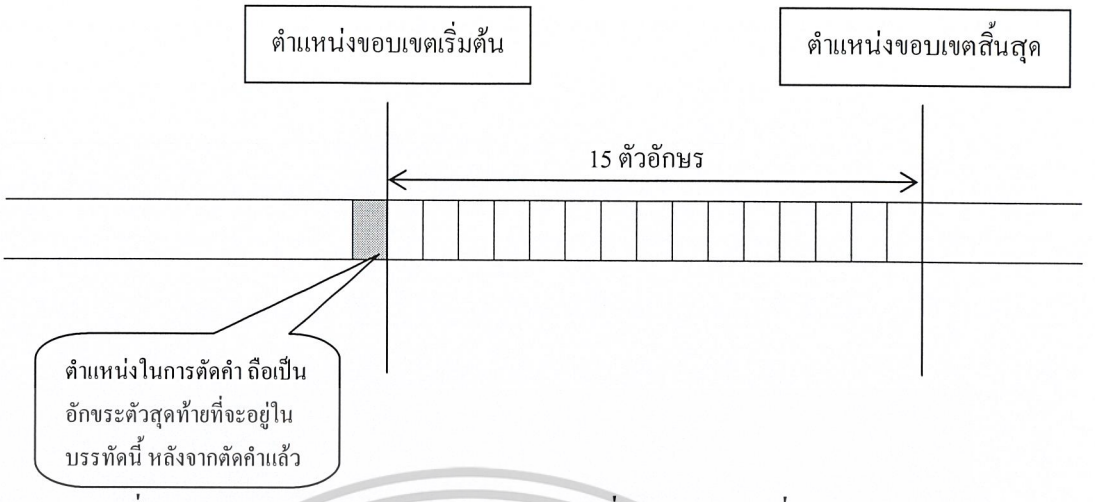
ตัวอย่างขั้นตอนวิธีในการค้นหาตำแหน่งในการตัดคำช่วงที่ 2 เป็นดังนี้

Separate Word Phase 2 Algorithm

```
success = false
num = 0
begmark = oldendmark
endmark = oldendmark + MaxWordLength //in middle line
oldend = endmark
while((success == false)  &&  (begmark != oldbegmark)){
        word = string.substr(begmark,endmark)
        while((not found word in Dict)  &&  (endmark > oldendmark)){
                endmark--
                word = string.substr(begmark,endmark)
                }
        if(endmark <= oldendmark){
                begmark--
                oldend--
                endmark = oldend
        }else{
                success = true
                }
        }
if(success == true)        {mark = begmark - 1}
else                       {mark = not found}
```

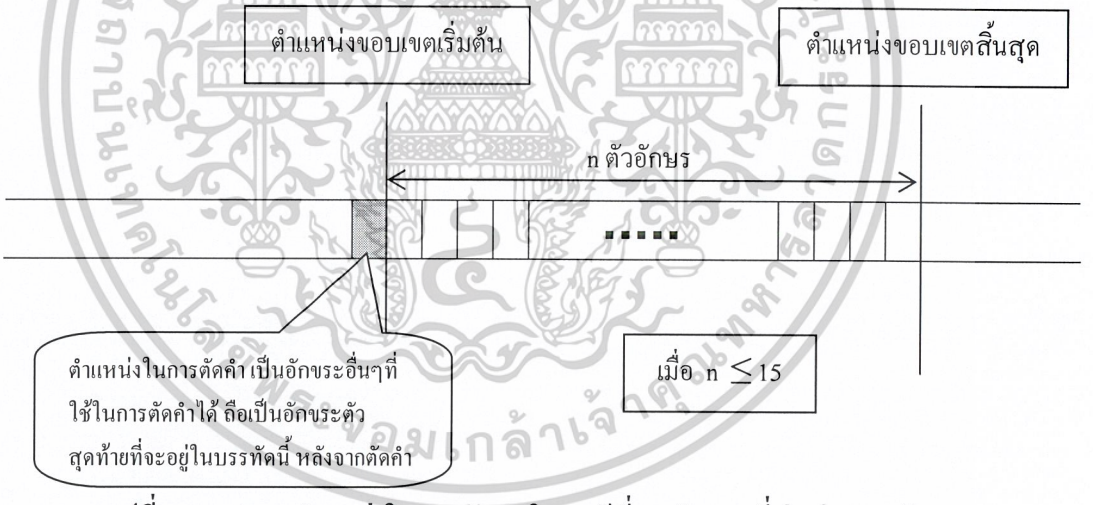
หากไม่พบคำศัพท์ใดๆเลย ไม่ว่าจะเป็นการค้นหาคำศัพท์ในการตัดคำช่วงที่ 1 หรือช่วงที่ 2 ก็ตาม ตำแหน่งตัดคำไปขึ้นบรรทัดใหม่ สามารถสรุปได้เป็น 2 กรณี คือ

- ถ้าหากระยะห่างระหว่างขอบเขตเริ่มต้น และขอบเขตสิ้นสุดนั้น มีค่าเท่ากับ 15 ตัวอักษร ซึ่งไม่รวมสระบนและสระล่าง หรืออีกนัยหนึ่ง คือ ไม่พบอักขระที่ช่วยในการตัดคำเลย เนื่องด้วยสมมติฐานที่ว่า ขนาดของคำนั้นมีความยาวไม่เกิน 15 ตัวอักษร ดังนั้น หากในช่วง 15 ตัวอักษรที่ทำการค้นหานั้น ไม่พบคำศัพท์ในพจนานุกรม ให้ถือว่า ตำแหน่งตัดคำ คือ ตำแหน่งที่อยู่ก่อนจะถึงขอบเขตเริ่มต้นนั่นเอง



รูปที่ 3.15 แสดงตำแหน่งในการตัดคำ ในกรณีที่ไม่พบอักขระที่ช่วยในการตัดคำ

- ถ้าหากระยะห่างระหว่างขอบเขตเริ่มต้น และขอบเขตสิ้นสุดนั้น มีต่ำกว่า 15 ตัวอักษร โดยไม่รวมสระบนและสระล่าง หรือได้พบอักขระที่ช่วยในการตัดคำ ให้ถือว่าอักขระที่ใช้ช่วยในการตัดคำนั้น เป็นตำแหน่งตัดคำ



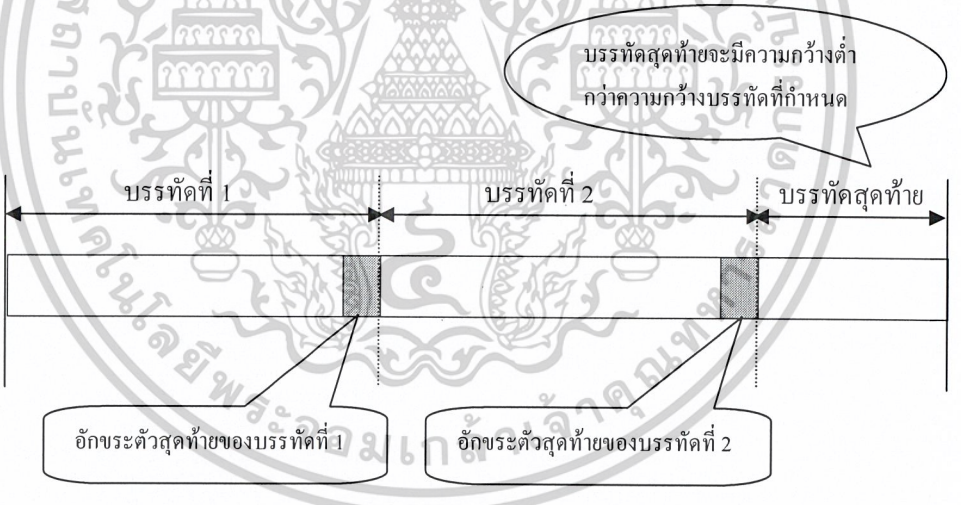
รูปที่ 3.16 แสดงตำแหน่งในการตัดคำ ในกรณีที่พบอักขระที่ช่วยในการตัดคำ

3.4.2 การตัดคำในข้อความทั้งย่อหน้า

จากหัวข้อที่ผ่านมา ได้อธิบายถึงขั้นตอนวิธีที่จะใช้ในการตัดคำช่วงที่ 1 และช่วงที่ 2 ไปแล้ว ซึ่งการตัดคำข้างต้นนั้น ได้แสดงวิธีการตัดคำในบรรทัดหนึ่งๆเท่านั้น แต่สำหรับข้อความจะจะนำมาตัดคำจริงๆนั้น โดยปกติแล้ว จะเป็นข้อความที่มีความยาวเกินกว่า 1 บรรทัดขึ้นไป ข้อความที่มีความยาวหลายๆบรรทัด หรือที่มีลักษณะเป็นย่อหน้านั้น ก็ย่อมจะต้องใช้ขั้นตอนวิธีที่ได้กล่าวมาแล้ว วนทำซ้ำไปเรื่อยๆ จนกระทั่งหมดทั้งข้อความ

หลักการในการตัดคำของข้อความทั้งย่อหน้า จะเป็นดังนี้

1) รับข้อความที่ต้องการตัดคำเข้ามา เป็นสายอักขระหนึ่งๆ รวมทั้งความกว้างของบรรทัดแต่ละบรรทัดที่ต้องการ
2) หาตำแหน่งขอบเขตสิ้นสุดของบรรทัดนั้น แล้วตัดสายอักขระนั้นออกมาพิจารณาเฉพาะส่วน
3) หาตำแหน่งขอบเขตเริ่มต้น ในสายอักขระย่อยที่ตัดออกมาจากข้อความหลัก
4) ค้นหาคำศัพท์ในการตัดคำช่วงที่ 1 และ 2 จนกว่าจะพบคำศัพท์ หรือจนกว่าจะจบขั้นตอนวิธีเพื่อหาตำแหน่งสุดท้ายของบรรทัด และตัดคำ
5) เมื่อพบตำแหน่งสุดท้ายของการตัดคำในบรรทัดนั้นแล้ว ให้เก็บค่าตำแหน่งไว้ ว่าบรรทัดลำดับที่เท่านี้ มีตำแหน่งตัดคำอยู่ตำแหน่งใด
6) จากนั้นเริ่มพิจารณาข้อความเดิม โดยเลื่อนตำแหน่งอักขระที่จะพิจารณาใหม่เป็นตำแหน่งที่อยู่ถัดจากตำแหน่งตัดคำของบรรทัดก่อนหน้า แล้วย้อนกลับไปหาตำแหน่งขอบเขตสิ้นสุดในข้อ 2 อีกครั้ง ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งข้อความที่เหลืออยู่ มีความกว้างของข้อความเท่ากับหรือต่ำกว่าความกว้างของบรรทัดที่กำหนดไว้ แสดงว่าข้อความได้ถูกตัดจนกระทั่งบรรทัดสุดท้ายแล้ว หรือ จบข้อความแล้วนั่นเอง



รูปที่ 3.17 แสดงการตัดคำในข้อความทั้งย่อหน้า

3.4.3 การตัดคำในเอกสารทั่วไป

สำหรับเอกสารทั่วๆไปนั้น ข้อความในเอกสารจะมีหลากหลาย ซึ่งก็อาจจะมีได้หลายๆย่อหน้า แต่ละย่อหน้าจะมีความยาวของข้อความในย่อหน้านั้นไม่เท่ากัน การจะทำการตัดคำในข้อความทั้งเอกสารได้นั้น จะต้องมีการแบ่งข้อความทั้งหมดออกทีละย่อหน้าเสียก่อน จากนั้นจึงจะดำเนินการตัดคำข้อความแต่ละย่อหน้า ดังที่ได้กล่าวไว้ในหัวข้อที่แล้ว และทำซ้ำไปเรื่อยๆ จนกระทั่งครบทุกข้อความในเอกสารนั้น

3.5 วิธีการสร้างพจนานุกรม

การสร้างพจนานุกรมเพื่อการตัดคำภาษาไทย จะไม่เหมือนพจนานุกรมสำหรับแปลภาษา เพราะไม่จำเป็นต้องพิจารณาความหมายว่า คำนั้นๆ มีความหมายว่าอะไร และไม่ต้องบรรจุไวยากรณ์ของคำลงไปด้วย ดังนั้น พจนานุกรมเพื่อการตัดคำจะมีขนาดเล็กกว่า เนื่องจาก มีเฉพาะตัวคำศัพท์เพียงอย่างเดียว และอาจจะมีจำนวนตัวอักษรของคำเพิ่มเข้ามาด้วย ทั้งนี้เพื่อช่วยในการสืบค้นหาคำได้รวดเร็วขึ้น

3.5.1 โครงสร้างของพจนานุกรม

การตัดคำภาษาไทยในปัญหาพิเศษนี้ จะอาศัยพจนานุกรมมาช่วยในการตรวจสอบคำศัพท์เป็นหลัก ว่าเป็นคำที่ถูกต้องและมีความหมายตรงตามพจนานุกรมหรือไม่ โดยไม่สนใจว่าความหมายของคำศัพท์นั้นๆคืออะไร ดังนั้น พจนานุกรมสำหรับการตัดคำไทยในปัญหาพิเศษนี้ จะบรรจุเฉพาะตัวคำศัพท์เพียงอย่างเดียวเท่านั้น

แต่เนื่องจากวิธีการในการตัดคำภาษาไทยท้ายบรรทัด จะเป็นการตรวจสอบคำศัพท์ที่อยู่ใกล้กับด้านท้ายบรรทัดมากที่สุด นั่นคือ เปรียบเทียบคำศัพท์ย้อนหลัง นับแต่ตำแหน่งสิ้นสุดบรรทัด ดังนั้น จะทำการเปรียบเทียบคำศัพท์เพื่อบอกว่าพบคำศัพท์นั้นในพจนานุกรมหรือไม่เท่านั้น จึงไม่อาจจะสร้างพจนานุกรมที่จะทำให้การสืบค้นมีความรวดเร็วมากขึ้นได้ เพราะคำศัพท์ที่เรียงลำดับตามพจนานุกรมนั้น จะเรียงตามอักษรตัวแรกไล่ไปจนกระทั่งอักษรสุดท้าย แต่ในการค้นหาคำศัพท์ในปัญหาพิเศษนี้ จะยึดตัวอักษรตัวสุดท้ายเป็นหลัก แล้วทำการค้นหาคำ หากไม่พบ ก็จะเปลี่ยนตำแหน่งตัวอักษรตัวแรกสุดในการค้นหาคำถัดไป ซึ่งจะทำให้คำศัพท์ที่ต้องทำการค้นหาในแต่ละครั้งนั้น ไม่ได้อยู่ตำแหน่งใกล้เคียงกันในพจนานุกรมเลย หากทำดัชนีเพื่อให้เข้าถึงคำศัพท์ได้เร็วขึ้นก็ตาม แต่เมื่อค้นหาคำต่อไป ก็ต้องเริ่มใช้ดัชนีในการเข้าถึงคำศัพท์ใหม่ตั้งแต่ต้น ซึ่งจะต้องเสียเวลาในการค้นหาที่ไม่แตกต่างกันกับการไม่ใช้ดัชนี

ด้วยเหตุผลดังที่กล่าวมานี้ การจัดโครงสร้างของพจนานุกรมในการตัดคำภาษาไทยที่ตำแหน่งท้ายบรรทัด จึงไม่มีความจำเป็นแต่อย่างใด สามารถจะพิจารณาใช้แถวลำดับของคำศัพท์ก็เพียงพอแล้ว นั่นคือ การตัดคำในปัญหาพิเศษนี้ ไม่จำเป็นต้องมีการออกแบบโครงสร้างการเก็บพจนานุกรมในหน่วยความจำเป็นพิเศษ

3.5.2 คำศัพท์ในพจนานุกรม

คำศัพท์ต่างๆที่นำมาบรรจุในพจนานุกรม ซึ่งนำมาใช้ทดสอบขั้นตอนวิธีในการตัดคำภาษาไทยท้ายบรรทัดนั้น จะเป็นได้ทั้งคำศัพท์ทั่วๆไปที่พบบ่อยในชีวิตประจำวัน และคำศัพท์เฉพาะทางที่ใช้กับบทความด้านใดด้านหนึ่ง ซึ่งในที่นี้ จะเป็นคำศัพท์เฉพาะทางด้านคอมพิวเตอร์

พจนานุกรมที่นำมาใช้นี้ ในขั้นต้น จะเป็นพจนานุกรมที่สร้างขึ้นเอง โดยบรรจุคำศัพท์ได้ในวงจำกัด อาจจะไม่สามารถครอบคลุมคำศัพท์ทั้งหมดที่มีอยู่ในปัจจุบันได้ เพราะไม่ใช่พจนานุกรมทั่วไป

คำศัพท์ที่บรรจุในพจนานุกรม อาจเป็นได้ทั้งคำโดด และคำประสม ซึ่งจะอ้างอิงตามพจนานุกรมฉบับเฉลิมพระเกียรติ พ.ศ.2530 คือเป็นคำศัพท์ที่พบในพจนานุกรมจริงๆ และสำหรับคำประสมนั้น อาจจะบรรจุทั้งคำโดดที่ประกอบเป็นคำประสม และคำประสมเองก็ได้ ตัวอย่างเช่น

เท่า	เท่ากับ	เท่านั้น
ตรวจ	ตรวจสอบ	
สาเหตุ	เหตุ	เหตุผล

สำหรับศัพท์เฉพาะทาง จะพิจารณาตามพจนานุกรมศัพท์เฉพาะทางนั้นๆ เช่น คำว่า “เครื่องพิมพ์” นอกจากนั้น ยังอาจมีทั้งชื่อเฉพาะ และคำทับศัพท์จากภาษาต่างประเทศอีกด้วย

3.5.3 วิธีการค้นหาคำศัพท์ในพจนานุกรม

สำหรับวิธีการค้นหาคำศัพท์ในพจนานุกรม เนื่องจากพจนานุกรมเป็นโครงสร้างแบบแถวลำดับ เรียงต่อกัน ดังนั้น ก็จะใช้วิธีการค้นหาแบบไบนารี โดยใช้พจนานุกรมที่มีการเรียงลำดับตามตัวอักษรเรียบร้อยแล้ว และนำคำศัพท์ที่ต้องการค้นหาไปเปรียบเทียบ ซึ่งถือว่าเป็นวิธีการค้นหาที่รวดเร็วที่สุดในปัจจุบัน

การเรียงลำดับของคำศัพท์ในพจนานุกรมเพื่อใช้ในการค้นหาแบบไบนารี จะแตกต่างกับการเรียงลำดับคำศัพท์ในพจนานุกรมทั่วไป กล่าวคือ การเรียงลำดับในพจนานุกรมที่จะค้นหาในคอมพิวเตอร์ จะต้องทำการเรียงลำดับตามรหัสแอสกี (ASCII) แทนที่จะเรียงลำดับตามตัวอักษรธรรมดา ตัวอย่างเช่น

ตารางที่ 3.1 แสดงการเปรียบเทียบการเรียงลำดับตามพจนานุกรมและตามรหัสแอสกี

เรียงลำดับตามพจนานุกรม	เรียงลำดับตามรหัสแอสกี
กอ	กอ
การ	การ
เก่ง	ของ
ของ	คน
โขน	เก่ง
คน	โขน

ดังนั้น คำศัพท์ในพจนานุกรมทดสอบ จะต้องนำมาเรียงลำดับตามรหัสแอสกีเสียก่อน จึงจะนำมาใช้ในการค้นหาแบบไบนารีต่อไป สำหรับลำดับของรหัสแอสกีนั้น สามารถดูได้ที่ภาคผนวก ข

การค้นหาแบบไบนารี ถือได้ว่าเป็นวิธีการค้นหาที่รวดเร็วมากที่สุดในปัจจุบัน ซึ่งจะมีงานที่ทำ ในกรณีแย่ที่สุดเป็น $\log_2(n) + 1$

3.6 วิเคราะห์งานที่ทำในขั้นตอนวิธี

จากขั้นตอนวิธีดังที่ได้กล่าวมาแล้วนั้น งานที่ต้องทำในขั้นตอนวิธี คือ จำนวนครั้งการเปรียบเทียบ เพื่อค้นหาคำศัพท์ในพจนานุกรม ซึ่งในกรณีที่แย่ที่สุด จะเกิดขึ้นเมื่อขอบเขตเริ่มต้นและขอบเขตสิ้นสุดการตัดคำ ห่างกันเป็นระยะ 15 ตัวอักษร โดยไม่นับรวมสระบน และสระล่าง เนื่องจากไม่พบอักขระที่ช่วยในการตัดคำ ซึ่งงานที่ต้องทำในขั้นตอนวิธีนี้ จะแบ่งเป็น 2 ช่วง คือ การพิจารณาตำแหน่งในการตัดคำช่วงที่ 1 และ ช่วงที่ 2

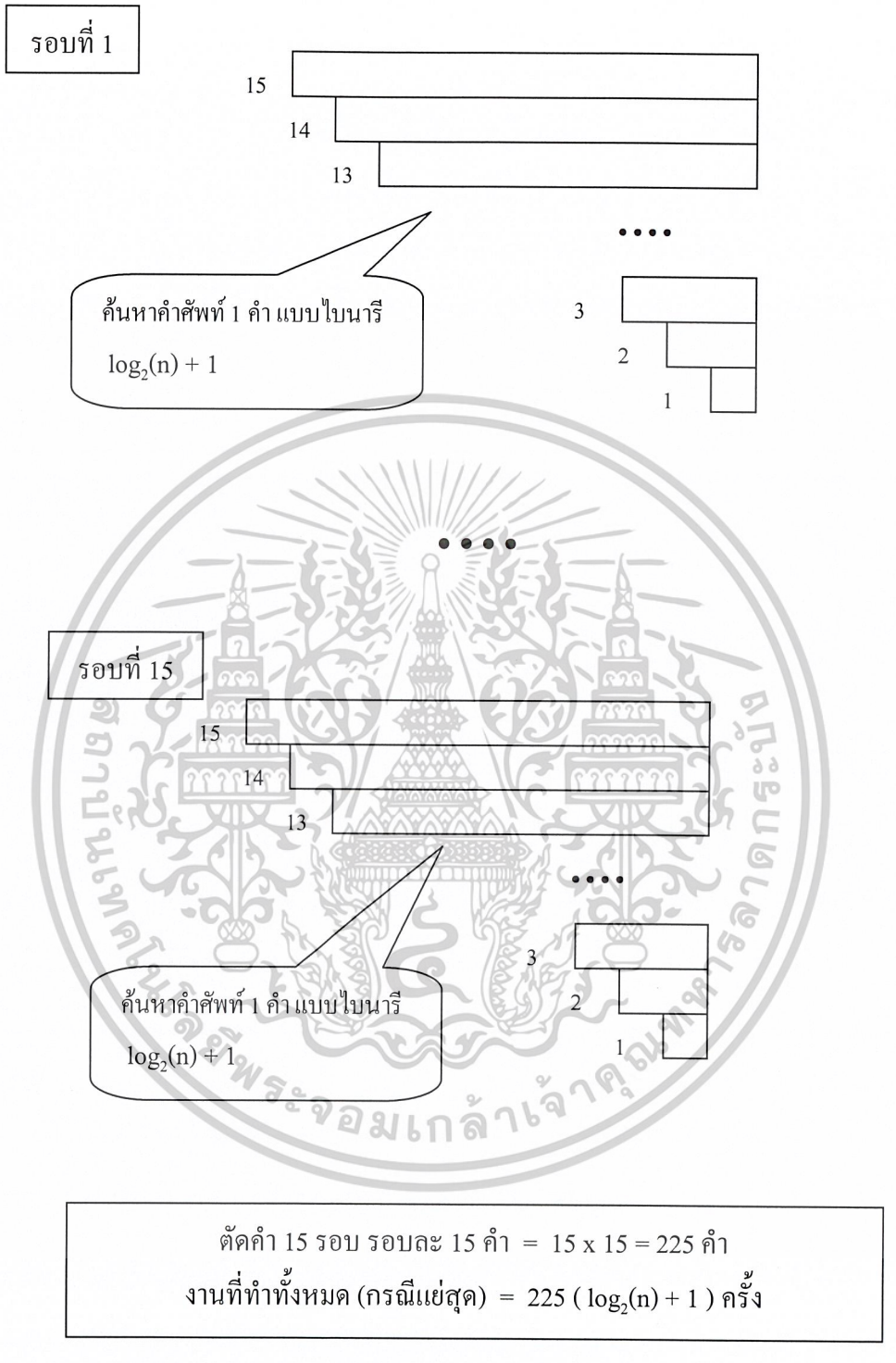
3.6.1 งานที่ทำในขั้นตอนการพิจารณาตำแหน่งการตัดคำช่วงที่ 1

ลักษณะการทำงานในช่วงที่ 1 ในกรณีที่แย่ที่สุด จะเกิดขึ้นเมื่อไม่พบอักขระที่ช่วยในการตัดคำ และระยะห่างระหว่างขอบเขตเริ่มต้น และขอบเขตสิ้นสุด มีความยาวมากที่สุด ในที่นี้คือ 15 ตัวอักษร (ดูรายละเอียดได้ใน บทที่ 3 หัวข้อ 3.4.1.1)

ในกรณีนี้ จะทำการพิจารณาตำแหน่ง โดยค้นหาคำศัพท์ ทั้งหมด 15 รอบ และในแต่ละรอบ จะเลื่อนอักขระตัวหน้าสุดเข้ามาเรื่อยๆ จากความยาว 15 ตัวอักษร จนเหลือเพียง 1 ตัวอักษร นั่นคือ ใน 1 รอบ จะค้นหาคำศัพท์ทั้งหมด 15 คำ รวมทั้งหมดเป็นการพิจารณา 15 x 15 = 255 คำ

ในแต่ละครั้ง จะต้องนำคำศัพท์ในข้อความ ไปเปรียบเทียบกับคำศัพท์ที่อยู่ในพจนานุกรม ซึ่งจะใช้วิธีการค้นหาแบบไบนารี (Binary Search) ถ้าคำศัพท์ในพจนานุกรมมีทั้งหมด n คำ แล้วงานที่ต้องทำในการค้นหาคำ 1 คำในพจนานุกรม ในกรณีแย่ที่สุด จะเป็น $\log_2(n) + 1$

ดังนั้น งานทั้งหมดที่ต้องทำ ในการเปรียบเทียบคำศัพท์ในพจนานุกรม เฉพาะการพิจารณาตำแหน่งในช่วงที่ 1 ในกรณีแย่ที่สุด จะเป็น $225 (\log_2(n) + 1)$ ครั้ง



รูปที่ 3.18 แสดงงานที่ทำในขั้นตอนการพิจารณาตำแหน่งในช่วงที่ 1

3.6.2 งานที่ทำในขั้นตอนการพิจารณาตำแหน่งการตัดคำช่วงที่ 2

ลักษณะการทำงานในช่วงที่ 2 ในกรณีที่แย่ที่สุด จะเกิดขึ้นเมื่อระยะห่างระหว่างขอบเขตเริ่มต้น และขอบเขตสิ้นสุด มีความยาวมากที่สุด ในที่นี้คือ 15 ตัวอักษร (ดูรายละเอียดได้ใน บทที่ 3 หัวข้อ 3.4.1.2)

ในกรณีนี้ จะทำการพิจารณาตำแหน่ง โดยค้นหาคำศัพท์ ทั้งหมด 15 รอบ และในแต่ละรอบ จะเลื่อนอักขระตัวท้ายสุดเข้ามาเรื่อยๆ จากความยาว 15 ตัวอักษร จนกระทั่งอักขระตัวสุดท้ายของคำนั้นอยู่ถัดจากตำแหน่งขอบเขตสิ้นสุด 1 อักขระ ตัวอย่างเช่น

รอบที่ 1 คำศัพท์ความยาว 15 ตัวอักษร จนถึง ความยาว 1 ตัวอักษร = 15 คำ

รอบที่ 2 คำศัพท์ความยาว 15 ตัวอักษร จนถึง ความยาว 2 ตัวอักษร = 14 คำ

รอบที่ 3 คำศัพท์ความยาว 15 ตัวอักษร จนถึง ความยาว 3 ตัวอักษร = 13 คำ

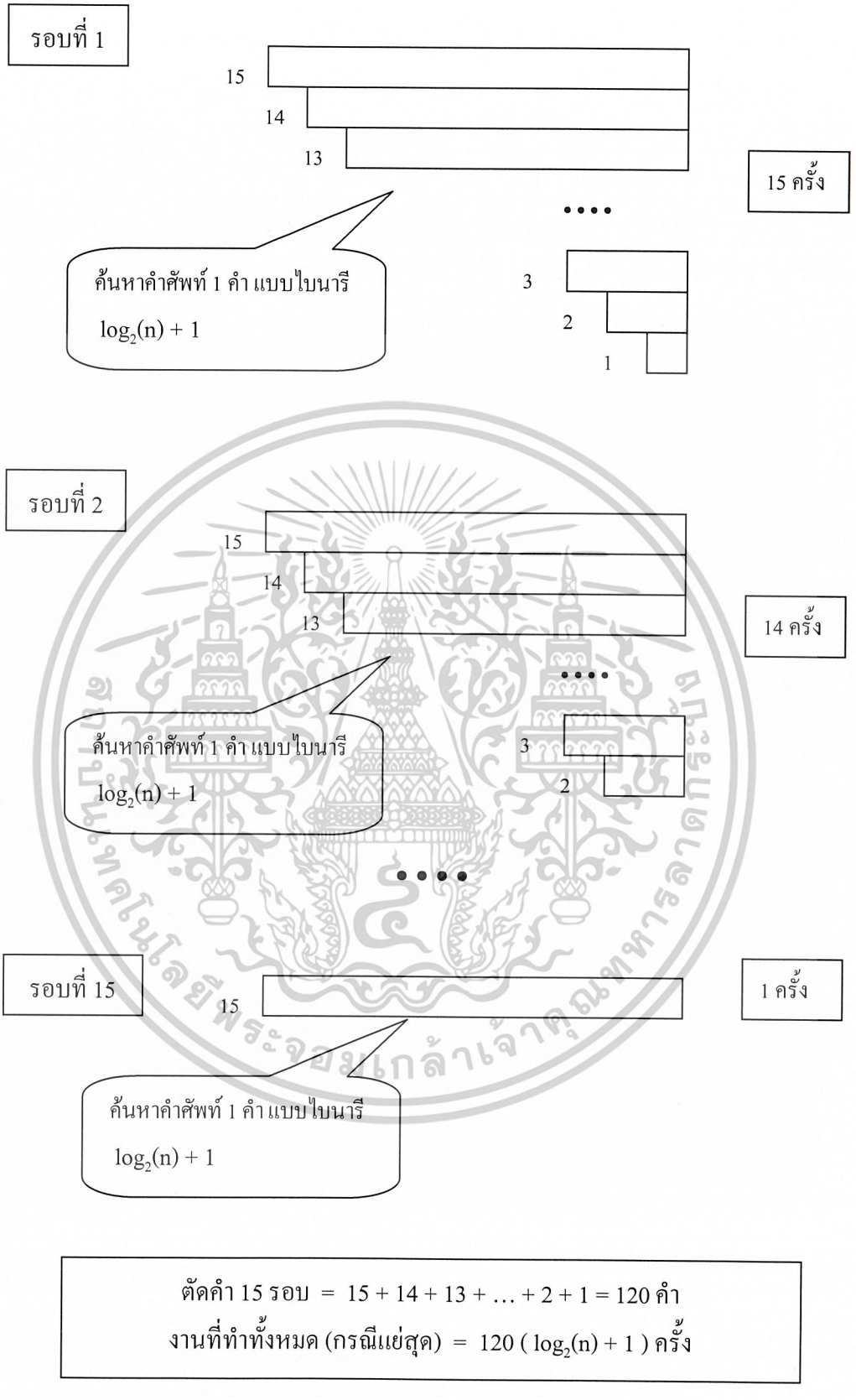
...

รอบที่ 15 คำศัพท์ความยาว 15 ตัวอักษร = 1 คำ

จากการพิจารณาตำแหน่งข้างต้น จะต้องพิจารณาทั้งหมด เป็น 15 + 14 + ... + 1 = 120 คำ

ในแต่ละครั้ง จะต้องนำคำศัพท์ในข้อความ ไปเปรียบเทียบกับคำศัพท์ที่อยู่ในพจนานุกรม ซึ่งจะใช้วิธีการค้นหาแบบไบนารี (Binary Search) นั่นคือ ถ้าคำศัพท์ในพจนานุกรมมีทั้งหมด n คำแล้ว งานที่ต้องทำในการค้นหาคำ 1 คำในพจนานุกรม ในกรณีแย่ที่สุด จะเป็น $\log_2(n) + 1$

ดังนั้น งานทั้งหมดที่ต้องทำ ในการเปรียบเทียบคำศัพท์ในพจนานุกรม เฉพาะการพิจารณาตำแหน่งในช่วงที่ 2 ในกรณีแย่ที่สุด จะเป็น $120 (\log_2(n) + 1)$ ครั้ง



รูปที่ 3.19 แสดงงานที่ทำในขั้นตอนการพิจารณาตำแหน่งในช่วงที่ 2

จากงานที่ทำทั้ง 2 ช่วงที่ได้กล่าวมา ทำให้สรุปได้ว่า งานทั้งหมดที่ต้องทำในขั้นตอนวิธีตัดคำไทยท้ายบรรทัดนี้ ในกรณีแย่ที่สุด คือ

งานที่ต้องทำทั้งหมด = งานที่ต้องทำในช่วงที่ 1 + งานที่ต้องทำในช่วงที่ 2

$= 225 (\log_2(n) + 1) + 120 (\log_2(n) + 1)$

$= 345 (\log_2(n) + 1)$ ครั้ง

ถ้าคำศัพท์ในพจนานุกรมที่จำนวนน้อย จะทำให้การค้นหาทำได้รวดเร็วมากขึ้น แต่ก็อาจเกิดปัญหาเมื่อไม่พบคำศัพท์ในพจนานุกรมได้ ทำให้การตัดคำผิดพลาด

บทที่ 4
การทดสอบขั้นตอนวิธี

การจะทดสอบขั้นตอนวิธีในการตัดคำภาษาไทยท้ายบรรทัดดังที่ได้ออกแบบไว้นั้น ทำได้โดยการสร้างโปรแกรมเพื่อใช้ในการตัดคำภาษาไทย ตามขั้นตอนวิธีที่ได้กล่าวมา จากนั้น จะนำข้อความต่างๆมาทดลองทำการตัดคำ เพื่อตรวจสอบประสิทธิภาพของขั้นตอนวิธีนี้ ว่าสามารถทำได้ดีเพียงใด

4.1 เครื่องมือที่ใช้ในการทดลอง

คอมพิวเตอร์ที่ใช้ในการทดลองในปัญหาพิเศษนี้ มีลักษณะความสามารถของเครื่อง เป็นดังนี้

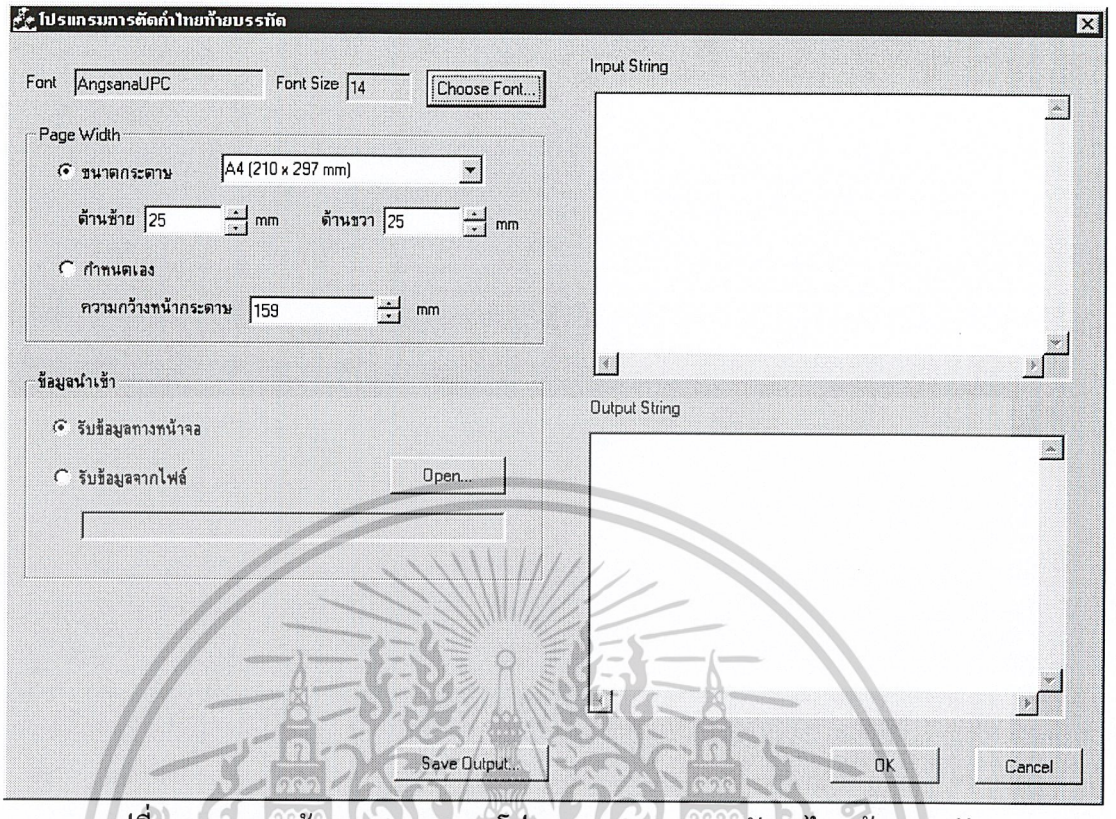
- หน่วยประมวลผลกลาง (CPU) Pentium III 450 MHz
- หน่วยความจำหลัก (RAM) ความจุ 256 MB
- หน่วยความจำสำรอง (Hard disk) ความจุ 40 GB
- ระบบปฏิบัติการ Windows 98 Second Edition

ในการพัฒนาโปรแกรมนั้น จะใช้ภาษา *Visual C++ Version 6.0* โดยจะมีการเรียกใช้ฟังก์ชั่นต่างๆที่อยู่แล้วในระบบปฏิบัติการวินโดวส์ ขึ้นมาประกอบเป็นส่วนหนึ่งของโปรแกรมด้วย ตัวอย่างของฟังก์ชั่นการทำงานภายในระบบปฏิบัติการวินโดวส์ ที่นำมาใช้ เช่น

- Font Dialog เป็นหน้าต่างที่ใช้ในการเลือกรูปแบบตัวอักษร ขนาดตัวอักษร
- Open / Save Dialog เป็นหน้าต่างที่ใช้ในการเปิดหรือบันทึกไฟล์ในเครื่อง
- ฟังก์ชั่น GetTextExtentPoint32 เป็นฟังก์ชั่นที่ใช้ในการคำนวณความกว้างและความสูงของข้อความในรูปแบบอักษรที่กำหนด ในหน่วยพิกเซล
- ฟังก์ชั่น GetTextExtentExPoint เป็นฟังก์ชั่นที่ใช้ในการคำนวณหาจำนวนอักขระสูงสุดที่สามารถบรรจุในความกว้างที่กำหนดไว้ได้

4.2 โปรแกรมทดสอบการตัดคำไทยท้ายบรรทัด

การสร้างโปรแกรมเพื่อการทดสอบขั้นตอนวิธีในการตัดคำไทย ณ ตำแหน่งท้ายบรรทัด จะอ้างอิงตามลักษณะการใช้ของผู้ใช้ทั่วไป ที่ใช้งานกันบนระบบปฏิบัติการวินโดวส์ เพื่อให้ผู้ใช้สามารถเข้าใจได้ง่ายและใช้งานได้สะดวกยิ่งขึ้น



รูปที่ 4.1 แสดงหน้าจอแสดงผลของโปรแกรมทดสอบการตัดคำไทยท้ายบรรทัด

โปรแกรมที่พัฒนาขึ้นจะมีลักษณะเป็นหน้าต่างไดอะล็อก ซึ่งจะคล้ายคลึงกับการใช้งานบนระบบปฏิบัติการวินโดวส์ทั่วไป โปรแกรมจะมีหน้าจอรับข้อมูลและแสดงผลรวมอยู่ในหน้าจอเดียวกัน

ส่วนรับข้อมูลของโปรแกรมทดสอบขั้นตอนวิธีการตัดคำภาษาไทย ผู้ใช้จะต้องระบุข้อมูลนำเข้าต่างๆ ดังนี้

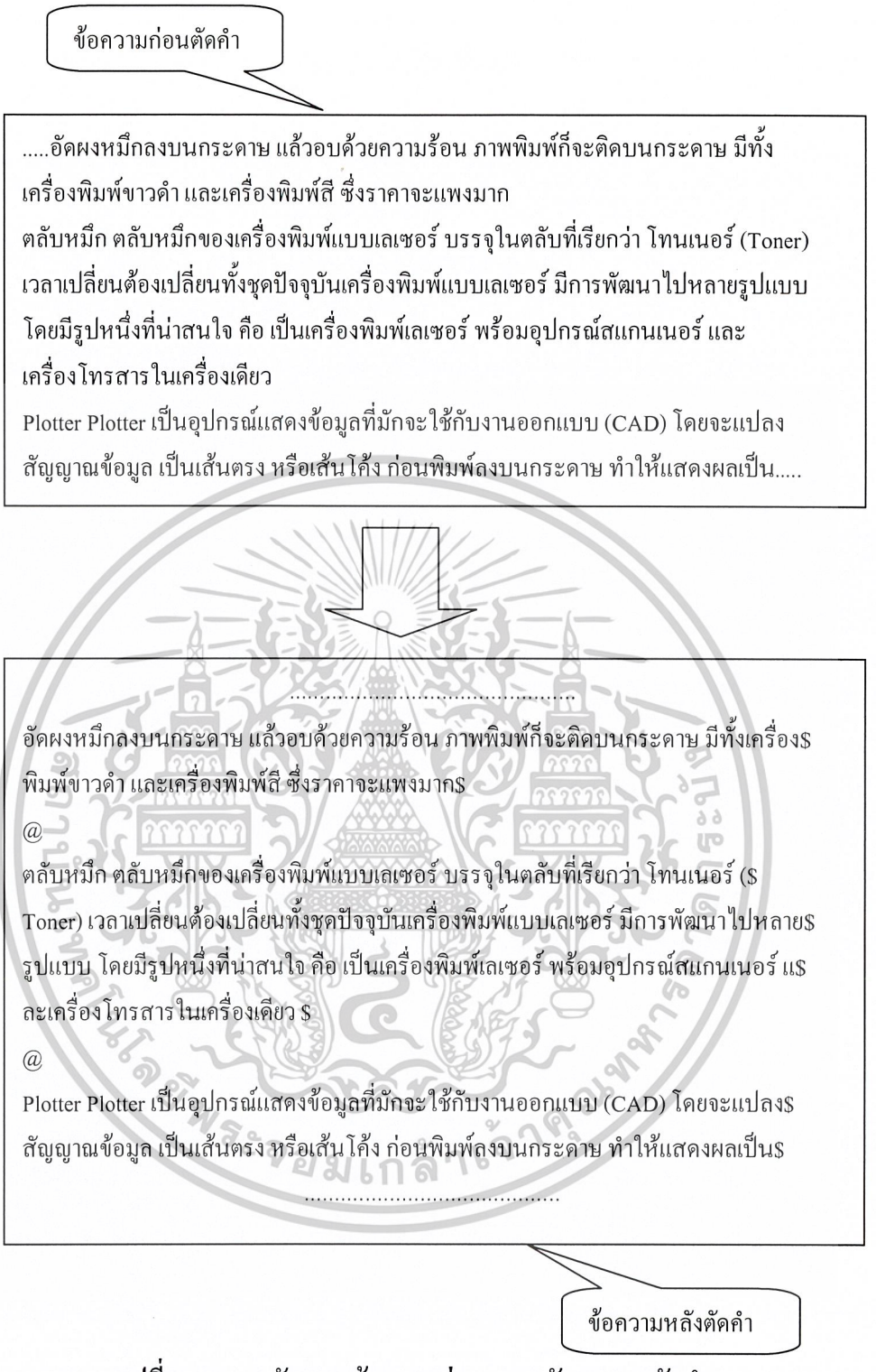
1) รูปแบบตัวอักษร และขนาดของตัวอักษร จะทำการระบุรูปแบบและขนาดโดยการเลือกจาก Font Dialog เมื่อคลิกปุ่ม Choose Font…
2) ความกว้างหน้ากระดาษ สามารถเลือกกำหนดได้ 2 วิธี คือ
   - กำหนดจากขนาดกระดาษ โดยเลือกขนาดกระดาษที่ต้องการจากคอมโบบ๊อกซ์ซึ่งในที่นี้ มี 3 ขนาดด้วยกัน คือ A4 (210 x 297 มิลลิเมตร), A5 (148 x 210 มิลลิเมตร) และ B5 (182 x 257 มิลลิเมตร) เมื่อเลือกขนาดกระดาษแล้ว จะทำการกำหนดกั้นหน้าด้านซ้าย และกั้นหลังด้านขวา ในหน่วยมิลลิเมตร

     ค่าเริ่มต้นจะใช้วิธีเลือกขนาดกระดาษเป็นขนาด A4 โดยมีกั้นหน้า 25 มิลลิเมตร และกั้นหลัง 25 มิลลิเมตร
   - กำหนดเอง โดยระบุความกว้างของแต่ละบรรทัดที่ต้องการในหน่วยมิลลิเมตร

3) ข้อความที่จะนำมาตัดคำ การกำหนดข้อความที่จะนำมาตัดคำ ทำได้ 2 แบบ คือ
   - รับข้อมูลจากหน้าจอแสดงผล โดยการพิมพ์ข้อความลงในเท็กซ์บ๊อกซ์ด้านบนขวา ค่าเริ่มต้นจะใช้วิธีรับข้อความจากหน้าจอ
   - รับข้อมูลจากไฟล์ที่มีอยู่แล้วภายในเครื่อง โดยการคลิกปุ่ม Open... เพื่อเลือกไฟล์ที่ต้องการจาก Open Dialog เมื่อเปิดไฟล์แล้ว จะแสดงข้อความที่อยู่ภายในไฟล์นั้นในเท็กซ์บ๊อกซ์ด้านบนขวา ซึ่งสามารถเปลี่ยนแปลงแก้ไขข้อความบนหน้าจอได้ทันที

ส่วนแสดงผลของโปรแกรม หลังจากคลิกปุ่ม OK แล้ว โปรแกรมจะทำการตัดข้อความในเท็กซ์บ๊อกซ์ด้านบนขวา แล้วนำมาแสดงผลในเท็กซ์บ๊อกซ์ด้านล่างขวา เป็นข้อความที่ตัดคำเรียบร้อยแล้ว ซึ่งจะมีการแทรกสัญลักษณ์เพื่อให้ทราบว่า เป็นการขึ้นบรรทัดใหม่จากการตัดข้อความในย่อหน้าเดียวกัน หรือเป็นการขึ้นย่อหน้าใหม่ สัญลักษณ์จะมี 2 แบบ คือ

1) สัญลักษณ์ "$" เป็นสัญลักษณ์ที่ใช้แทนการขึ้นบรรทัดใหม่ จากการตัดข้อความที่อยู่ภายในย่อหน้าเดียวกัน ซึ่งจะเกิดจากขั้นตอนวิธีตัดคำ
2) สัญลักษณ์ "@" เป็นสัญลักษณ์ที่ใช้แทนการขึ้นบรรทัดใหม่ จากการขึ้นย่อหน้าถัดไป



รูปที่ 4.2 แสดงลักษณะข้อความก่อนและหลังจากการตัดคำ

4.3 การออกแบบการทดลองการตัดคำ

เพื่อให้ทราบและสามารถประเมินผลการตัดคำของขั้นตอนวิธีนี้ ว่ามีความถูกต้องอย่างไร จะต้องทำการทดสอบโปรแกรมกับข้อความในรูปแบบต่างๆ และเปรียบเทียบผลที่เกิดขึ้น

4.3.1 ข้อความที่ใช้ในการทดลอง

การทดลองเพื่อจะตรวจสอบผลการตัดคำนั้น จะต้องเลือกข้อความในหลายๆลักษณะ เพื่อที่ว่า จะสามารถนำมาเปรียบเทียบซึ่งกันและกันได้ และเพื่อที่จะได้ผลที่น่าเชื่อถือได้ในระดับหนึ่ง ข้อความที่จะนำมาตัดคำ ควรมีความยาวพอสมควร อาจจะมีได้หลายย่อหน้า และแต่ละย่อหน้าควรจะมีความยาวตั้งแต่ 3 บรรทัดขึ้นไป เมื่อทำการตัดคำมาเรียบร้อยแล้ว และเมื่อรวมข้อความทั้งหมดแล้ว ควรจะได้จำนวนบรรทัดประมาณ 100 บรรทัด ซึ่งอาจจะมากกว่าหรือน้อยกว่า ขึ้นอยู่กับรูปแบบและขนาดตัวอักษร รวมทั้งความกว้างของบรรทัดที่กำหนดด้วย

ข้อความที่นำมาใช้ในการทดลอง จะมี 3 ลักษณะ ดังนี้

- **ลักษณะที่ 1** จะเป็นข้อความที่มีคำศัพท์อยู่ในพจนานุกรมทุกๆคำ ทั้งคำศัพท์ทั่วไป และคำศัพท์เฉพาะทาง ในที่นี้ จะเป็นศัพท์เฉพาะทางคอมพิวเตอร์ ซึ่งจะทำการทดลอง 2 ชุดข้อความ เรียกว่า ข้อความชุด A และ B

  สำหรับข้อความในลักษณะที่ 1 นั้น จะเป็นลักษณะที่ต้องให้ความสนใจเป็นพิเศษ เพราะในสภาพความเป็นจริง พจนานุกรมที่ใช้ควรจะมีคำศัพท์ครอบคลุมทุกคำ จึงต้องทำการทดลอง 2 ข้อความ เพื่อยืนยันถึงผลในกรณีที่มีคำศัพท์ที่ถูกต้องในพจนานุกรมเสมอ
- **ลักษณะที่ 2** จะเป็นข้อความที่มีคำศัพท์ทั่วไปที่พบในพจนานุกรมบางส่วน และคำศัพท์เฉพาะทางอีกบางส่วน ซึ่งจะทำการทดลอง 1 ชุดข้อความ เรียกว่า ข้อความชุด C
- **ลักษณะที่ 3** จะเป็นข้อความที่มีคำศัพท์ทั่วไปที่พบในพจนานุกรมบางส่วน แต่จะไม่พบคำศัพท์เฉพาะทางเกี่ยวกับคอมพิวเตอร์เลย เพราะเป็นข้อความที่มีเนื้อหาทางด้านอื่นๆ ในที่นี้จะเป็นเนื้อหาเกี่ยวกับด้านเศรษฐศาสตร์ ซึ่งจะทำการทดลอง 1 ชุดข้อความ เรียกว่า ข้อความชุด D

ข้อความที่นำมาใช้ในการทดลองครั้งนี้ จะมีทั้งหมด 4 ชุดข้อความ ซึ่งตัวอย่างของข้อความแต่ละชุด สามารถดูได้ในภาคผนวก ง

4.3.2 พจนานุกรมที่ใช้ในการทดลอง

พจนานุกรมคำศัพท์ที่นำมาใช้ทดสอบการตัดคำในที่นี้ จะใช้พจนานุกรมที่สร้างขึ้นเอง โดยบรรจุทั้งคำศัพท์ทั่วไป และคำศัพท์เฉพาะทางด้านคอมพิวเตอร์ ซึ่งข้อความลักษณะที่ 1 ทุกๆคำจะต้องพบอยู่ในพจนานุกรมอย่างแน่นอน ทั้งศัพท์ทั่วไป และศัพท์เฉพาะทาง

สำหรับลักษณะที่ 2 จะพบคำศัพท์ทั่วไปบางส่วน และคำศัพท์เฉพาะทางบางส่วน แต่ลักษณะที่ 3 อาจพบคำศัพท์ทั่วไปบ้างบางส่วน แต่จะไม่มีศัพท์เฉพาะทางเกี่ยวกับด้านคอมพิวเตอร์เลย

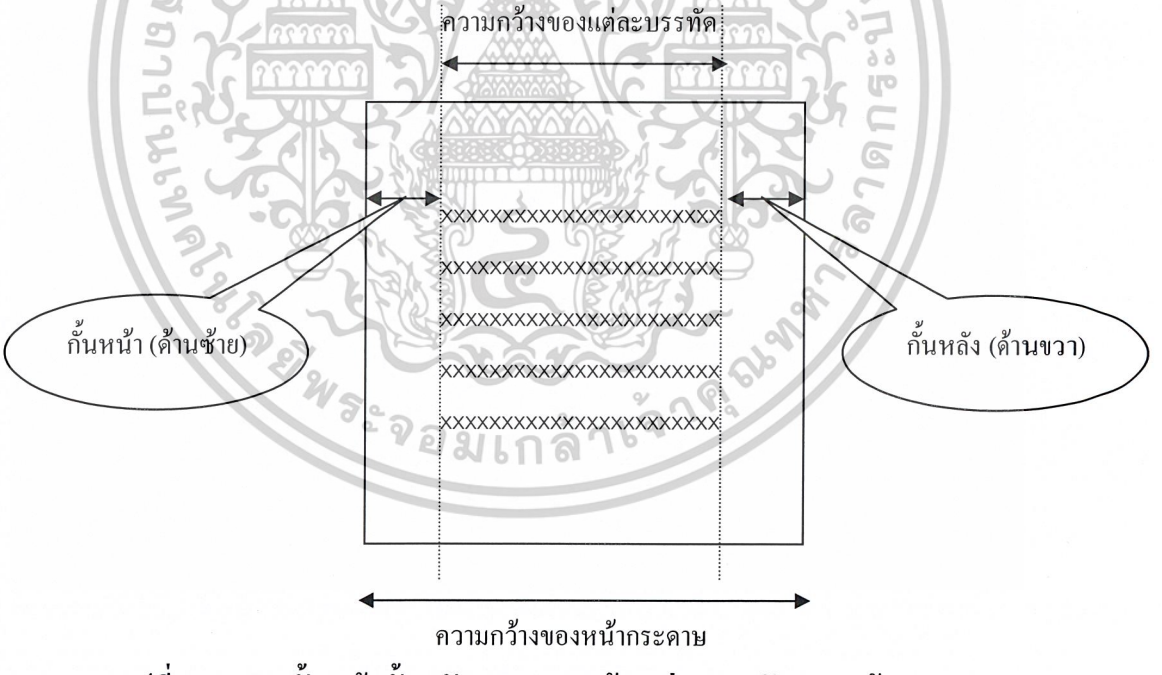
รายการคำศัพท์ทั้งหมดที่อยู่ในพจนานุกรมที่ได้นำมาใช้ในการทดลองครั้งนี้ สามารถดูได้ที่ภาคผนวก จ

4.3.3 ลักษณะการทดลอง

การทดสอบขั้นตอนวิธีการตัดคำไทยท้ายบรรทัด จะทำได้โดยการนำข้อความแต่ละลักษณะมาทดลองตัดคำ โดยจะกำหนดขนาดตัวอักษร และความกว้างของบรรทัดแตกต่างกัน ซึ่งจะส่งผลให้ตำแหน่งการตัดคำในแต่ละบรรทัดนั้น แตกต่างกันไปด้วย

ในปัญหาพิเศษนี้ จะกำหนดรูปแบบตัวอักษรที่ใช้ในการทดลองเป็น AngsanaUPC ซึ่งจะมีขนาดตัวอักษร 3 ขนาด คือ 14, 16 และ 18 พอยต์ ซึ่งในแต่ละขนาดตัวอักษรจะทดลองตัดคำ โดยกำหนดความกว้างหน้ากระดาษ จากขนาดกระดาษ A4 (210 x 297 มิลลิเมตร) และกำหนดกั้นหน้าด้านซ้าย และกั้นหลังด้านขวา แตกต่างกัน ดังนี้

1) กั้นหน้า 30 มิลลิเมตร และกั้นหลัง 30 มิลลิเมตร หรือมีความกว้างของแต่ละบรรทัดเท่ากับ 150 มิลลิเมตร หรือเท่ากับ 566 พิกเซล
2) กั้นหน้า 25 มิลลิเมตร และกั้นหลัง 25 มิลลิเมตร หรือมีความกว้างของแต่ละบรรทัดเท่ากับ 160 มิลลิเมตร หรือเท่ากับ 604 พิกเซล
3) กั้นหน้า 20 มิลลิเมตร และกั้นหลัง 20 มิลลิเมตร หรือมีความกว้างของแต่ละบรรทัดเท่ากับ 170 มิลลิเมตร หรือเท่ากับ 642 พิกเซล



รูปที่ 4.3 แสดงกั้นหน้า กั้นหลัง และความกว้างแต่ละบรรทัดของหน้ากระดาษ

จากขนาดตัวอักษรทั้ง 3 ขนาด และความกว้างของแต่ละบรรทัดทั้ง 3 แบบ จะทำให้ได้รูปแบบการทดลองตัดคำในข้อความหนึ่งๆได้ 9 กรณี คือ

กรณีที่ 1 ขนาดตัวอักษร 14 พอยต์ และความกว้างบรรทัด 150 มิลลิเมตร
กรณีที่ 2 ขนาดตัวอักษร 14 พอยต์ และความกว้างบรรทัด 160 มิลลิเมตร
กรณีที่ 3 ขนาดตัวอักษร 14 พอยต์ และความกว้างบรรทัด 170 มิลลิเมตร

กรณีที่ 4 ขนาดตัวอักษร 16 พอยต์ และความกว้างบรรทัด 150 มิลลิเมตร
กรณีที่ 5 ขนาดตัวอักษร 16 พอยต์ และความกว้างบรรทัด 160 มิลลิเมตร
กรณีที่ 6 ขนาดตัวอักษร 16 พอยต์ และความกว้างบรรทัด 170 มิลลิเมตร

กรณีที่ 7 ขนาดตัวอักษร 18 พอยต์ และความกว้างบรรทัด 150 มิลลิเมตร
กรณีที่ 8 ขนาดตัวอักษร 18 พอยต์ และความกว้างบรรทัด 160 มิลลิเมตร
กรณีที่ 9 ขนาดตัวอักษร 18 พอยต์ และความกว้างบรรทัด 170 มิลลิเมตร

การทดลองนั้น จะทำการตัดคำในข้อความหนึ่งๆ โดยกำหนดรูปแบบอักษรเป็น AngsanaUPC แล้วเปลี่ยนขนาดตัวอักษร และความกว้างบรรทัดไปเรื่อยๆ ให้ครบทั้ง 9 กรณี ซึ่งข้อความที่นำมาทำการตัดคำ จะมีอยู่ 4 ชุดดังที่ได้กล่าวมาแล้ว ดังนั้น จะทำการทดลองทั้งหมด 36 กรณี

การเปลี่ยนขนาดตัวอักษรและความกว้างบรรทัดให้แตกต่างกันในแต่ละกรณีนั้น จะทำให้ตำแหน่งที่ต้องพิจารณาการตัดคำในแต่ละบรรทัดแตกต่างกันไปด้วย ซึ่งจะทำให้ต้องพิจารณาคำศัพท์ที่แตกต่างกัน

4.4 ผลการทดลอง

หลังจากที่ได้ทำการทดลองตัดคำภาษาไทยท้ายบรรทัด โดยใช้ข้อความทั้ง 4 ชุดนั้น และพจนานุกรมที่สร้างขึ้น

โดยการจะพิจารณาว่าการตัดคำในแต่ละบรรทัดนั้น มีความถูกต้องเพียงใด จะมีหลักเกณฑ์ที่ใช้ในการพิจารณา ดังนี้

1) ถ้าคำสุดท้ายของบรรทัดนั้น และคำเริ่มต้นในบรรทัดถัดไป เป็นคำที่ถูกต้องและพบในพจนานุกรม แสดงว่า การตัดคำในบรรทัดนั้นถูกต้อง

2) ถ้าคำสุดท้ายของบรรทัดนั้น เป็นคำที่ถูกต้องและพบในพจนานุกรม แต่คำเริ่มต้นในบรรทัดถัดไป เป็นคำที่ไม่พบอยู่ในพจนานุกรมแล้ว ถ้าการแบ่งแยกคำนั้น ทำได้ถูกต้อง คือ แบ่งตามพยางค์ได้ถูกต้อง ให้ถือว่า การตัดคำในตำแหน่งนั้นถูกต้อง

ตัวอย่างคำศัพท์ที่สามารถแบ่งแยกตามพยางค์ได้ เช่น

- จำนวน สามารถแบ่งได้เป็น จำ – นวน
- คำนวณ สามารถแบ่งได้เป็น คำ – นวณ

3) ถ้าการตัดคำ เป็นการทำให้คำถูกแบ่งแยกออกไป โดยไม่มีความหมาย และไม่ได้เป็นการแบ่งแยกตามพยางค์ของคำ ให้ถือว่า การตัดคำในตำแหน่งนั้น เป็นการตัดคำที่ผิดพลาด

ตัวอย่างคำศัพท์ที่ถูกแบ่งแยกแล้วไม่มีความหมาย เช่น

- และ ถูกแบ่งเป็น แ – ละ โดยคำศัพท์ที่พบ คือ ละ
- เหมือน ถูกแบ่งเป็น เห – มือน โดยคำศัพท์ที่พบ คือ มือ
- สามารถ ถูกแบ่งเป็น สาม – ารถ โดยคำศัพท์ที่พบ คือ สาม
- เดียว ถูกแบ่งเป็น เดี – ยว โดยคำศัพท์ที่พบ คือ ดี

ซึ่งจากหลักเกณฑ์พิจารณาข้างต้น เมื่อนำมาประกอบกับการนำข้อความมาทดลองตัดคำ จะได้ผลจากการทดลอง ดังนี้

ตารางที่ 4.1 การตัดคำท้ายบรรทัดกับข้อความชุด A (มี 13 ย่อหน้า)

กรณี	ขนาดตัวอักษร (point)	ความกว้างบรรทัด (mm)	จำนวนบรรทัดที่ตัดได้	จำนวนความผิดพลาดในการตัดคำ	ความถูกต้อง (ร้อยละ)	คำที่ตัดคำผิดพลาด (ตัวเลขในวงเล็บเป็นจำนวนครั้งที่พบ)
1	14	150	96	2	97.92	สามารถ, เหมาะ
2	14	160	89	4	95.51	รูป, สามารถ(2), หาก
3	14	170	85	5	94.12	เก็บ, หมาย, และ(2), ระหว่าง
4	16	150	105	3	97.14	สามารถ, เฉพาะ, หาก
5	16	160	100	2	98.00	สามารถ, อดีต
6	16	170	93	4	95.70	สามารถ, ผลัก, มาตร, เดียว
7	18	150	124	5	95.97	สามารถ, เหมาะ, เกี่ยว, มาตร, เดียว
8	18	160	115	2	98.26	ด้วย, และ
9	18	170	110	4	96.36	และ, หาย, เสียง, เหมือน

หมายเหตุ ชุด A เป็นข้อความที่พบคำศัพท์ในพจนานุกรมทุกคำ ทั้งศัพท์ทั่วไปและศัพท์เฉพาะทาง

ตารางที่ 4.2 การตัดคำท้ายบรรทัดกับข้อความชุด B (มี 16 ย่อหน้า)

กรณี	ขนาดตัวอักษร (point)	ความกว้างบรรทัด (mm)	จำนวนบรรทัดที่ตัดได้	จำนวนความผิดพลาดในการตัดคำ	ความถูกต้อง (ร้อยละ)	คำที่ตัดคำผิดพลาด (ตัวเลขในวงเล็บเป็นจำนวนครั้งที่พบ)
1	14	150	99	2	97.98	เดียว, มาก
2	14	160	93	1	98.92	หนาม
3	14	170	88	1	98.86	เสียด
4	16	150	108	4	96.30	เสีย, ไอบีเอ็ม, หมาย, มาก
5	16	160	104	4	96.15	กราฟิก, การ, หนาม, ด้วย
6	16	170	97	2	97.94	มาตร, เดียว
7	18	150	129	5	96.12	หนาม(2), สามารถ(2), และ
8	18	160	120	5	95.83	เดียว, จุด, เชื่อม, การ, มาก
9	18	170	113	1	99.12	เหมือน

หมายเหตุ ชุด B เป็นข้อความที่พบคำศัพท์ในพจนานุกรมทุกคำ ทั้งศัพท์ทั่วไปและศัพท์เฉพาะทาง

ตารางที่ 4.3 การตัดคำท้ายบรรทัดกับข้อความชุด C (มี 14 ย่อหน้า)

กรณี	ขนาดตัวอักษร (point)	ความกว้างบรรทัด (mm)	จำนวนบรรทัดที่ตัดได้	จำนวนความผิดพลาดในการตัดคำ	ความถูกต้อง (ร้อยละ)	คำที่ตัดคำผิดพลาด (ตัวเลขในวงเล็บเป็นจำนวนครั้งที่พบ)
1	14	150	96	6	93.75	หมาย, มาก, บน, การ, ไปรษณีย์, นั้น
2	14	160	92	1	98.91	นั้น
3	14	170	87	4	95.40	เหมาะ, จำหน่าย, และ, สามารถ
4	16	150	109	4	96.33	และ, เก็บ(2), ไปรษณีย์
5	16	160	103	2	98.06	หมาย, รูป
6	16	170	95	6	93.68	หมาย, มาก, การ, รูป, สามารถ, นั้น
7	18	150	129	4	96.90	การ, และ, สามารถ(2)
8	18	160	117	2	98.29	มาย, จำหน่าย
9	18	170	111	5	95.50	รูป, มาก(2), เหมาะ, โปรแกรม

หมายเหตุ ชุด C เป็นข้อความที่พบศัพท์ทั่วไปบางส่วน และศัพท์เฉพาะทางบางส่วน

ตารางที่ 4.4 การตัดคำท้ายบรรทัดกับข้อความชุด D (มี 14 ย่อหน้า)

กรณี	ขนาดตัวอักษร (point)	ความกว้างบรรทัด (mm)	จำนวนบรรทัดที่ตัดได้	จำนวนความผิดพลาดในการตัดคำ	ความถูกต้อง (ร้อยละ)	คำที่ตัดคำผิดพลาด (ตัวเลขในวงเล็บเป็นจำนวนครั้งที่พบ)
1	14	150	101	4	96.04	บัญชี, เก็ง, นิยม, วิพากษ์
2	14	160	92	4	95.65	ผลิต, เสื่อม, มาตร, เทคนิค
3	14	170	88	1	98.86	สามารถ
4	16	150	111	3	97.30	ด้วย, เก็ง, บริหาร
5	16	160	106	3	97.17	บัญชี, ฐานะ, ภาวะ
6	16	170	99	2	97.98	ดึงดูด, สามารถ
7	18	150	132	3	97.73	ผลิต, ฐานะ, การณ์
8	18	160	121	1	99.17	สามารถ
9	18	170	115	2	98.26	ฐานะ, วิพากษ์

หมายเหตุ ชุด D เป็นข้อความที่พบศัพท์ทั่วไปบางส่วน แต่ไม่พบศัพท์เฉพาะทางเลย

4.4.1 ผลการตัดคำผิดพลาดเมื่อพบคำศัพท์ในพจนานุกรม

จากผลการทดลองที่ผ่านมา การตัดคำที่ผิดพลาดนั้น อาจจะเกิดจากการที่พบคำศัพท์ในพจนานุกรม แต่คำศัพท์นั้นอยู่ในตำแหน่งซ้อนในคำศัพท์อื่น หรือพบคำศัพท์ในช่วงรอยต่อระหว่างคำ ทำให้ตัดคำผิดพลาด ซึ่งคำศัพท์ที่ตัดคำผิดพลาดนั้น มักจะเกิดจากสาเหตุหลักๆ 2 สาเหตุ คือ

1) เป็นคำศัพท์ที่มีคำศัพท์ย่อยอื่นๆที่ซ้อนอยู่ภายใน ทำให้เวลาค้นหาคำศัพท์อาจทำให้พบคำศัพท์ที่อยู่ภายในก่อน จึงทำให้ตัดคำผิดพลาดได้

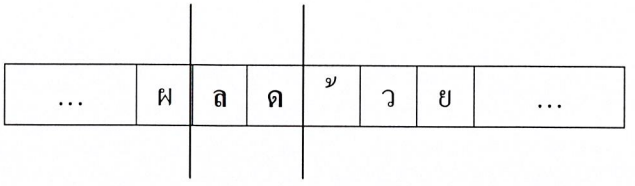
ตัวอย่าง คำศัพท์ที่มีคำศัพท์อื่นซ้อนอยู่ภายใน เช่น

สามารถ	มีคำศัพท์อื่นๆ คือ	สาม, มา, รถ
เก็ง	มีคำศัพท์อื่นๆ คือ	ก็
เสียง	มีคำศัพท์อื่นๆ คือ	เสีย, สี
มาก	มีคำศัพท์อื่นๆ คือ	มา
เหมาะ	มีคำศัพท์อื่นๆ คือ	เหมา, หมา, มา

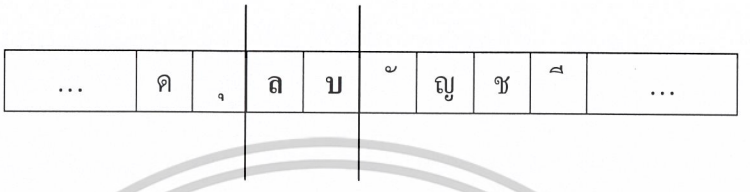
2) เมื่อคำศัพท์นั้นเขียนติดต่อกับคำศัพท์อื่น และในช่วงรอยต่อของคำศัพท์นั้น สามารถรวมกันและเกิดคำใหม่ได้ เวลาค้นหาคำศัพท์นั้น กลับไปพบคำศัพท์ที่เกิดขึ้นในช่วงรอยต่อ ทำให้การตัดคำผิดพลาด

ตัวอย่าง คำศัพท์ที่เกิดขึ้นในช่วงรอยต่อระหว่างคำศัพท์ เช่น

ผล + ด้วย ทำให้เกิดคำศัพท์ คือ ลด



ดุล + บัญชี ทำให้เกิดคำศัพท์ คือ ลบ



การตัดคำที่ผิดพลาดในกรณีที่พบคำศัพท์ในพจนานุกรม แต่ตัดคำได้ไม่ถูกต้องนั้น จะพบคำศัพท์ต่างๆในการทดลอง ดังตารางที่ 4.5 ซึ่งจะแสดงรายการคำศัพท์ที่ตัดคำผิดพลาด ลักษณะคำศัพท์ที่พบในพจนานุกรม และความถี่ที่พบในข้อความทั้ง 4 ชุดที่ทำการทดลอง

ความผิดพลาดส่วนใหญ่ มักเกิดจากคำศัพท์ที่อยู่ซ้อนภายในคำศัพท์อื่น ยกเว้น สำหรับคำศัพท์ที่ตัดคำผิดพลาดบางคำ เกิดจากการพบคำศัพท์ระหว่างรอยต่อของคำ ได้แก่

- การ พบคำศัพท์ คือ ยก (โดย<u>ยก</u>าร), พก (สภา<u>พก</u>าร)
- ด้วย พบคำศัพท์ คือ ลด (ผ<u>ลด้วย</u>), กด (อี<u>กด้วย</u>), กด (บว<u>กด้วย</u>)
- นั้น พบคำศัพท์ คือ จน (ธุรกิ<u>จนั้น</u>), ท่าน (เ<u>ท่านั้น</u>)
- บัญชี พบคำศัพท์ คือ ลบ (ดุ<u>ลบัญชี</u>)
- นิยม พบคำศัพท์ คือ ตน (จารี<u>ตนิยม</u>)
- บน พบคำศัพท์ คือ ลบ (ข้อมู<u>ลบน</u>)

นอกจากนี้ การตัดคำผิดพลาด อาจจะเกิดขึ้นได้ในกรณีที่ ไม่พบคำศัพท์ในช่วง 15 ตัวอักษร ทั้งก่อนและหลังขอบเขตสิ้นสุดบรรทัด ซึ่งในขั้นตอนวิธีนี้ ถือว่า คำศัพท์จะมีความยาวมากที่สุด 15 ตัวอักษร โดยไม่นับรวมสระบน และสระล่าง ทำให้ตัดคำในตำแหน่งนับจากขอบเขตสิ้นสุดบรรทัดขึ้นไป 15 ตัวอักษร ซึ่งอาจจะเกิดขึ้นได้ในกรณีที่ไม่พบคำศัพท์ใดๆในพจนานุกรม

ตารางที่ 4.5 รายการคำศัพท์ที่ตัดคำผิดพลาด

คำศัพท์ที่ตัดคำผิด	คำศัพท์ที่พบในพจนานุกรม	จำนวนครั้งที่พบ
สามารถ	สาม	16
และ	ละ	8
มาก	มา	7
การ	*	5
เดียว	ดี	5
หมาย	มา	5
มาตร	มา	4
รูป	รู	4
หนาม	หนา	4
เหมาะ	มา	4
เก็บ	ก็	3
ฐานะ	ฐาน	3
ด้วย	*	3
นั้น	*	3
เก็ง	ก็	2
จำหน่าย	น่า	2
บัญชี	*	2
ไปรษณีย์	ไป	2
ผลิต	ผล	2
วิพากษ์	พา	2
หาก	หา	2
เหมือน	มือ	2

คำศัพท์ที่ตัดคำผิด	คำศัพท์ที่พบในพจนานุกรม	จำนวนครั้งที่พบ
กราฟิก	กราฟ	1
การณ์	การ	1
เกี่ยว	กี่	1
จุด	จุ	1
เฉพาะ	พา	1
เชื่อม	ชื่อ	1
ดึงดูด	ดู	1
เทคนิค	คน	1
นิยม	*	1
บน	*	1
บริหาร	หา	1
โปรแกรม	กรม	1
ผลัก	ผล	1
ภาวะ	**	1
มาย	มา	1
ระหว่าง	ว่า	1
เสีย	สี	1
เสียง	สี	1
เสียด	สี	1
เสื่อม	สื่อ	1
หาย	หา	1
อดีต	ดี	1
ไอบีเอ็ม	อบ	1

<u>หมายเหตุ</u> * เป็นคำศัพท์ที่ตัดคำผิดพลาด ซึ่งเกิดจากการพบคำศัพท์ระหว่างรอยต่อของคำ

** เป็นคำศัพท์ที่ตัดคำผิดพลาด เนื่องจากไม่พบคำศัพท์ใดๆเลย จึงตัดคำ ณ ตำแหน่งนับจากขอบเขตสิ้นสุดขึ้นมา 15 ตัวอักษร

4.4.2 ผลการตัดคำผิดพลาดเมื่อไม่พบคำศัพท์ในพจนานุกรม

ในกรณีที่ไม่พบคำศัพท์ในพจนานุกรม แม้การตัดคำอาจจะถูกต้องก็ตาม กล่าวคือ สามารถตัดคำตามคำศัพท์ที่มีความหมายได้ถูกต้อง แต่ความผิดพลาดจะเกิดขึ้น เนื่องจากการตัดคำผิดตำแหน่ง ทั้งๆที่ยังมีเนื้อที่เหลือเพียงพอที่จะบรรจุคำต่อไปได้ แต่กลับตัดคำเพื่อปัดข้อความขึ้นบรรทัดใหม่ไปเสียก่อน ทำให้ไม่ได้ตำแหน่งตัดคำที่อยู่ใกล้ตำแหน่งสิ้นสุดบรรทัดมากที่สุดอย่างแท้จริง

การตัดคำที่ผิดตำแหน่งไป มักจะพบในข้อความที่มีคำศัพท์อยู่ในพจนานุกรมเพียงบางส่วน โดยเฉพาะข้อความในชุด D ซึ่งมีโอกาสที่จะไม่พบคำศัพท์ในพจนานุกรมมากกว่าข้อความในลักษณะอื่นๆ ตัวอย่างเช่น ส่วนหนึ่งของข้อความชุด D ที่มีขนาดอักษร 18 พอยต์ และความกว้างบรรทัด 150 มิลลิเมตร หรือเท่ากับ 566 พิกเซล เมื่อตัดคำเรียบร้อยแล้ว จะได้ผลดังนี้

โครงสร้างเศรษฐกิจการเมืองของไทยที่มีลักษณะรวมศูนย์อำนาจอยู่ในมือคนกลุ่มน้อยที่มี$
กรอบความคิดแบบจารีตนิยม และใช้นโยบายพัฒนาเศรษฐกิจแบบเป็นบริวารประเทศ$
ทุนนิยมพัฒนาอุตสาหกรรมทำให้การพัฒนาประเทศขาดความสมดุล การ$
กระจายทรัพย์สินรายได้และความรู้มีช่องว่างแตกต่างกันเพิ่มมากขึ้น ประชาชนไม่ได้รับ$
การพัฒนาให้มีความรู้ ความสามารถและมีวินัยในการสร้างชาติ$

จะสังเกตได้ว่า ในบรรทัดที่ 3 "...ขาดความสมดุล การ" และปัดข้อความ "กระจายทรัพย์สิน..." ไปขึ้นบรรทัดใหม่ในบรรทัดที่ 4 พื้นที่ที่อยู่หลังการคำว่า "... การ" ยังมีเหลืออยู่ ซึ่งเพียงพอที่จะให้คำถัดไป คือ คำว่า "กระจาย" ขึ้นมาต่อท้ายได้อีก แต่เนื่องจากคำว่า "กระจาย" นั้นไม่พบอยู่ในพจนานุกรม การค้นหาคำศัพท์จึงเลื่อนขึ้นมาจนกระทั่งพบคำว่า "การ" และตัดคำ ณ ตำแหน่งนั้น

จากข้อสังเกตที่ได้กล่าวมาแล้วนั้น ในความเป็นจริง ข้อความในบรรทัดที่ 3 นั้น จะมีความกว้างของข้อความทั้งหมด 463 พิกเซล ส่วนคำว่า "กระจาย" จะมีความกว้าง เท่ากับ 51 พิกเซล นั่นคือ ถ้าทำการตัดคำได้ถูกต้องตามตำแหน่งที่ควรจะเป็นจริงๆ ก็ควรรวมเอาคำว่า "กระจาย" ขึ้นไปอยู่ในบรรทัดที่ 3 ด้วย ซึ่งจะทำให้ได้ข้อความที่มีความกว้างทั้งหมด 514 พิกเซล และเป็นข้อความที่มีความกว้างไม่เกินความกว้างของบรรทัดที่กำหนดไว้ คือ 150 มิลลิเมตร หรือ 566 พิกเซล

ทุนนิยมพัฒนาอุตสาหกรรมทำให้การพัฒนาประเทศขาดความสมดุล การ	+	กระจาย		
463		51	=	514 pixels

4.4.3 งานที่ทำในการทดลอง

งานที่ทำในการทดลอง คือ จำนวนครั้งในการเปรียบเทียบคำศัพท์ในพจนานุกรม เมื่อต้องการค้นหาคำศัพท์ที่อยู่ใกล้ตำแหน่งท้ายบรรทัดมากที่สุด จากการทดลอง เพื่อทดสอบขั้นตอนวิธีดังกล่าว ในกรณีแย่ที่สุด เมื่อระยะห่างระหว่างขอบเขตเริ่มต้นและขอบเขตสิ้นสุดมากที่สุดเท่ากับ 15 ตัวอักษร และไม่พบอักขระที่ช่วยในการตัดคำ ซึ่งวิเคราะห์งานที่ทำในการพิจารณาตำแหน่งช่วงที่ 1 ได้เป็น $225 (\log_2(n) + 1)$ และงานที่ทำในช่วงที่ 2 เป็น $120 (\log_2(n) + 1)$ นั้น (ดูรายละเอียดเกี่ยวกับการวิเคราะห์งานที่ทำได้ใน บทที่ 3 หัวข้อ 3.6) เมื่อพจนานุกรมที่ใช้ในการทดลอง มีคำศัพท์ทั้งหมด 673 คำ หรือ $n = 673$ จะได้ว่า

ในกรณีแย่ที่สุด

งานที่ทำในการพิจารณาตำแหน่งช่วงที่ 1 $= 225 (\log_2(673) + 1) = 2338.75$ ครั้ง

งานที่ทำในการพิจารณาตำแหน่งช่วงที่ 2 $= 120 (\log_2(673) + 1) = 1247.34$ ครั้ง

ดังนั้น งานทั้งหมดที่ต้องทำในขั้นตอนวิธีนี้ ในกรณีแย่ที่สุด จะมีค่าเท่ากับ

งานที่ทำทั้งหมด = งานในช่วงที่ 1 + งานในช่วงที่ 2

$= 225 (\log_2(673) + 1) + 120 (\log_2(673) + 1)$

$= 345 (\log_2(673) + 1)$

= 3586.09 ครั้ง หรือประมาณ 3587 ครั้ง

แต่ในการทดลองนั้น จะพบว่า การพิจารณาค้นหาคำศัพท์ เพื่อให้ทราบตำแหน่งการตัดคำในแต่ละบรรทัดนั้น อาจเป็นไปได้ 4 กรณี ดังนี้

1) สามารถตัดคำได้ก่อนที่จะเริ่มพิจารณาในช่วงที่ 1 เนื่องจากพบอักขระอื่นๆ คือ อาจเป็นอักขระภาษาอื่น หรือตัวเลข ดังนั้น จะไม่มีการเข้าไปทำงานในการพิจารณาตำแหน่งในช่วงที่ 1 และ 2
2) พบคำศัพท์ในการพิจารณาตำแหน่งช่วงที่ 1 จะเข้าทำงานในช่วงที่ 1 และเมื่อพบคำศัพท์ก็จะทำการตัดคำทันที
3) พบคำศัพท์ในการพิจารณาตำแหน่งช่วงที่ 2 จะเข้าทำงานในช่วงที่ 1 แล้วไม่พบคำศัพท์ จึงต้องเข้าทำงานในช่วงที่ 2 ต่อไป และพบคำศัพท์ที่ต้องการค้นหาในช่วงนี้
4) ไม่พบคำศัพท์ใดๆทั้งในช่วงที่ 1 และ 2 หลังจากเข้าทำงานในช่วงที่ 1 และ 2 แล้ว แต่ก็ยังไม่พบคำศัพท์ที่ตรงกับคำศัพท์ในพจนานุกรม จึงทำการตัดคำทันที ซึ่งไม่จำเป็นว่า การที่ไม่พบคำศัพท์จะหมายความว่า จะทำให้การตัดคำผิดพลาดเสมอไป ในกรณีนี้ อาจพิจารณาว่าได้ทำงานเทียบเท่ากับการพบคำศัพท์ในการพิจารณาตำแหน่งช่วงที่ 2 ในกรณีแย่ที่สุด

ผลจากการทดลองตัดคำในข้อความลักษณะต่างๆ จะได้สถิติการทำงานในแต่ละกรณีข้างต้น ดังตารางที่ 4.6 – 4.9

ตารางที่ 4.6 แสดงสถิติการทำงานในแต่ละช่วง เมื่อทดลองกับข้อความ ชุด A

กรณี	ขนาดตัวอักษร	ความกว้างบรรทัด	อักขระอื่นๆ	ช่วงที่ 1	ช่วงที่ 2	ไม่พบคำศัพท์
1	14	150	0	70	1	12
2	14	160	0	63	0	13
3	14	170	0	57	4	11
4	16	150	0	86	0	6
5	16	160	0	74	0	13
6	16	170	0	70	0	10
7	18	150	0	95	1	15
8	18	160	0	86	1	15
9	18	170	0	81	3	13

ตารางที่ 4.7 แสดงสถิติการทำงานในแต่ละช่วง เมื่อทดลองกับข้อความ ชุด B

กรณี	ขนาดตัวอักษร	ความกว้างบรรทัด	อักขระอื่นๆ	ช่วงที่ 1	ช่วงที่ 2	ไม่พบคำศัพท์
1	14	150	2	73	0	8
2	14	160	0	65	2	10
3	14	170	1	60	0	11
4	16	150	1	82	1	8
5	16	160	0	81	0	7
6	16	170	2	66	1	12
7	18	150	3	97	2	11
8	18	160	2	92	0	10
9	18	170	1	86	1	9

ตารางที่ 4.8 แสดงสถิติการทำงานในแต่ละช่วง เมื่อทดลองกับข้อความ ชุด C

กรณี	ขนาดตัวอักษร	ความกว้างบรรทัด	อักขระอื่นๆ	ช่วงที่ 1	ช่วงที่ 2	ไม่พบคำศัพท์
1	14	150	1	69	0	12
2	14	160	0	66	0	12
3	14	170	1	58	2	12
4	16	150	1	83	1	10
5	16	160	0	75	0	14
6	16	170	0	72	0	9
7	18	150	1	99	2	13
8	18	160	0	87	2	14
9	18	170	0	83	1	13

ตารางที่ 4.9 แสดงสถิติการทำงานในแต่ละช่วง เมื่อทดลองกับข้อความ ชุด D

กรณี	ขนาดตัวอักษร	ความกว้างบรรทัด	อักขระอื่นๆ	ช่วงที่ 1	ช่วงที่ 2	ไม่พบคำศัพท์
1	14	150	3	79	1	4
2	14	160	1	71	0	6
3	14	170	3	66	0	5
4	16	150	3	83	0	11
5	16	160	0	79	1	12
6	16	170	2	79	2	2
7	18	150	8	98	0	12
8	18	160	5	84	2	16
9	18	170	5	89	1	6

จะสังเกตได้ว่า การทำงานส่วนใหญ่ มักจะทำงานในการพิจารณาตำแหน่งช่วงที่ 1 มากกว่าที่จะต้องเข้าไปทำงานในการพิจารณาตำแหน่งช่วงที่ 2 แสดงว่า ในการทดลองนั้น จะมีงานที่ทำทั้งหมดน้อยกว่า $345 (\log_2(673) + 1) = 3586.09$ ครั้ง

สำหรับในกรณีที่พบอักขระในภาษาอื่น หรือเป็นตัวเลข จะถือว่าทำงานน้อยกว่าในกรณีอื่นๆ เพราะจะทำการตัดคำทันทีที่ตำแหน่งสิ้นสุดอักขระที่เป็นภาษาอื่นหรือตัวเลขนั้น

นอกจากนี้ การที่ไม่พบคำศัพท์ทั้งในการพิจารณาตำแหน่งช่วงที่ 1 และ ช่วงที่ 2 ก็ไม่จำเป็นว่า จะเป็นการตัดคำที่ผิดพลาด ดังนั้น การพิจารณางานที่ทำ ว่าการตัดคำในแต่ละบรรทัดจะต้องเข้าทำงานในช่วงใดบ้าง ก็ไม่ได้มีความสัมพันธ์กับกรณีการตัดคำที่ผิดพลาดแต่อย่างใด ถือว่า เป็นคนละกรณีกัน

บทที่ 5
สรุปผลและข้อเสนอแนะ

จากการศึกษาการตัดคำภาษาไทยโดยมุ่งพิจารณาตัดคำ ณ ตำแหน่งท้ายบรรทัด เพื่อทำการตัดข้อความที่มีความยาวเกินกว่า 1 บรรทัดให้ปัดขึ้นบรรทัดใหม่ ขั้นตอนวิธีที่คิดค้นขึ้นจะพิจารณาตัดคำให้ถูกต้องตามคำศัพท์ภาษาไทยด้วยการพิจารณาส่วนของข้อความเฉพาะบริเวณท้ายบรรทัดเทียบกับคำศัพท์ในพจนานุกรมและให้แต่ละบรรทัดสามารถบรรจุข้อความได้มากที่สุดเท่าที่เป็นไปได้

5.1 หลักการของขั้นตอนวิธี

ขั้นตอนวิธีที่ออกแบบโดยสรุปมีดังนี

1) พิจารณาจากตำแหน่งท้ายบรรทัด ซึ่งเป็นตำแหน่งที่ยาวที่สุดที่บรรจุคำศัพท์ได้ใน 1 บรรทัด เริ่มจาก 15 ตัวอักขระและพิจารณาลดอักษรตัวหน้าลงทีละ 1 ตัวอักขระ โดยในแต่ละความยาวเป็นคำศัพท์ 1 คำ นำไปเทียบกับคำศัพท์ในพจนานุกรม ถ้าเป็นคำศัพท์ที่ถูกต้องจะได้ตำแหน่งสิ้นสุดคำศัพท์เป็นตำแหน่งตัดคำตัวสุดท้ายของบรรทัดนั้น และขั้นตอนวิธีหยุดทำงาน

2) พิจารณาจากตำแหน่งท้ายบรรทัด ค่อมไปทางท้ายบรรทัด เริ่มจาก 15 ตัวอักขระและพิจารณาลดอักษรตัวท้ายลงทีละ 1 ตัวอักขระ โดยในแต่ละความยาวเป็นคำศัพท์ 1 คำ นำไปเทียบกับคำศัพท์ในพจนานุกรม ถ้าเป็นคำศัพท์ที่ถูกต้องจะได้ตำแหน่งต้นคำศัพท์เป็นตำแหน่งตัดคำตัวขึ้นต้นบรรทัดใหม่ และขั้นตอนวิธีหยุดทำงาน

3) ถ้าขั้นตอนทั้งสองไม่พบคำศัพท์ที่ถูกต้องจะตัดคำ ณ ตำแหน่งท้ายบรรทัดตามที่คำนวณและยอมรับว่าอาจจะมีการตัดผิดตำแหน่ง หรือไม่มีคำศัพท์ในพจนานุกรม

5.2 สรุปผลการทดลอง

ข้อมูลที่ใช้ทดสอบความถูกต้องของขั้นตอนวิธีได้แบ่งเป็น 3 กรณีคือ

1) คำศัพท์ทั่วไปและคำศัพท์เฉพาะทาง ซึ่งคำศัพท์ทุกคำมีในพจนานุกรม

2) คำศัพท์ทั่วไปและคำศัพท์เฉพาะทาง ซึ่งคำศัพท์ทั่วไปมีในพจนานุกรม ส่วนคำศัพท์เฉพาะทางมีในพจนานุกรมประมาณร้อยละ 50

3) คำศัพท์ทั่วไปและคำศัพท์เฉพาะทางที่ไม่เกี่ยวข้องกัน ซึ่งคำศัพท์ทั่วไปจะมีในพจนานุกรม ส่วนคำศัพท์เฉพาะทางนั้นไม่มี

ผลการทดลอง

กรณีที่ 1 เป็นสภาพจำลองสภาวะที่จะมีการใช้งานจริงในสภาวะปกติคือมีคำศัพท์เฉพาะทางทุกคำในพจนานุกรม และเพื่อความมั่นใจจึงทดสอบกับข้อมูล 2 ชุดที่ไม่ซ้ำกัน

ชุดที่ 1 อัตราการตัดคำถูกตำแหน่งโดยเฉลี่ยร้อยละ 96.55

ชุดที่ 2 อัตราการตัดคำถูกตำแหน่งโดยเฉลี่ยร้อยละ 97.47

กรณีที่ 2 มีคำศัพท์เฉพาะทางบางส่วน อัตราการตัดคำถูกตำแหน่งโดยเฉลี่ยร้อยละ 96.31

กรณีที่ 3 ไม่มีคำศัพท์เฉพาะทางในพจนานุกรม อัตราการตัดคำถูกตำแหน่งโดยเฉลี่ยร้อยละ 97.57

จะพบว่าอัตราการตัดถูกตำแหน่งทั้ง 3 กลุ่มมีความใกล้เคียงกัน ไม่มีความแตกต่างอย่างมีนัยสำคัญ แสดงว่าการมีคำศัพท์ในพจนานุกรมหรือไม่ ไม่ได้ส่งผลอย่างชัดเจน การตัดผิดตำแหน่งจะอยู่ที่ขั้นตอนวิธีโดยตรง

ลักษณะการตัดคำผิดตำแหน่ง เกิดจาก

1) คำศัพท์ที่พบเป็นคำศัพท์ย่อยที่ซ้อนอยู่ในคำศัพท์ที่ควรจะเป็น ซึ่งจะสั้นกว่าคำศัพท์ที่ถูกต้อง เช่น มากมาย ตัดเป็น มา – กมาย หรือ สามารถ ตัดเป็น สาม – ารถ

2) คำศัพท์ที่พบเป็นคำที่พบในช่วงรอยต่อระหว่างคำ เช่น ลด เกิดจาก ผล + ด้วย

3) ไม่พบคำศัพท์ที่ถูกต้องในพจนานุกรม จึงทำให้ตัดคำก่อนจะถึงท้ายบรรทัด และมีพื้นที่เหลือเพียงพอจะบรรจุคำศัพท์ถัดไปลงในบรรทัดนั้นได้

5.3 ข้อเสนอแนะ

ควรปรับปรุงขั้นตอนวิธีให้ครอบคลุมยิ่งขึ้น โดย

1) ถ้าอักขระตัวถัดไปหลังตำแหน่งที่ตัดคำเป็นสระบน สระล่าง สระหลัง หรือวรรณยุกต์ ให้พิจารณารวมสระหรือวรรณยุกต์นั้นเข้าไปพร้อมกับพยัญชนะที่อยู่ก่อนหน้าด้วยเสมอ

2) เมื่อพบตำแหน่งที่จะตัดแล้ว ให้ค้นหาคำที่อยู่ก่อนหน้าและหลังคำที่พบ เพื่อจะได้คำที่ถูกต้องติดต่อกันมากขึ้นถึง 3 คำ และจะได้เชื่อมั่นว่าตัดคำถูกตำแหน่ง

3) ให้ทดสอบกับพจนานุกรมฉบับที่มีคำศัพท์สมบูรณ์ ซึ่งจะมีคำศัพท์ครอบคลุมมากขึ้น เช่น พจนานุกรมฉบับราชบัณฑิตยสถาน

เอกสารอ้างอิง

[1] ชัยยงศ์ วงศ์ชัยสุวัฒน์ และคณะ. "อิเล็กทรอนิกส์ดิกชันนารีสำหรับงานประมวลผลภาษาไทย." การประชุมวิชาการทางวิศวกรรมไฟฟ้า 9 สถาบัน ครั้งที่ 11. ธันวาคม 2531. หน้า 1-48-1 – 1-48-10.

[2] ประภาพรรณ คงวิทย์เศรณี. "การตรวจสอบและแก้ไขการสะกดคำในภาษาไทย." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541.

[3] มนัส รอดพ้น. "การใช้โทเคนพาสซิงอัลกอริทึมในการแก้ไขคำผิดใน OCR ภาษาไทย." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2546.

[4] ยืน ภู่วรวรรณ. "การสร้างพจนานุกรมสำหรับอัลกอริทึมไทย." การประชุมวิชาการทางวิศวกรรมไฟฟ้า 8 สถาบันอุดมศึกษา ครั้งที่ 7. ธันวาคม 2527. หน้า ค-280 – ค-289.

[5] ยืน ภู่วรวรรณ. "การวิเคราะห์ข้อมูลคำไทย." การประชุมวิชาการทางวิศวกรรมไฟฟ้า 8 สถาบันอุดมศึกษา ครั้งที่ 7. ธันวาคม 2527. หน้า ค-290 – ค-303.

[6] ยืน ภู่วรวรรณ และ ชัยยงค์ วงศ์ชัยสุวัฒน์. "การออกแบบและลดขนาดข้อมูลคำไทยในพจนานุกรมสำหรับงานพิสูจน์อักษร." วิทยาสารเกษตรศาสตร์ สาขาวิทยาศาสตร์, ปีที่ 23, ฉบับที่ 4, ตุลาคม 2532.

[7] ยืน ภู่วรวรรณ และ วิวรรธน์ อิ่มอารมณ์. "การแบ่งแยกพยางค์ไทยด้วยดิกชันนารี." การประชุมวิชาการทางวิศวกรรมไฟฟ้า สถาบันอุดมศึกษาแห่งประเทศไทย ครั้งที่ 9. ธันวาคม 2529.

[8] รัตติกร วรากูลศิริพันธุ์ และคณะ. "การวิเคราะห์เลือกประโยคที่ถูกต้องจากความถี่ของการใช้คำ." การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 12. พฤศจิกายน 2532. หน้า 521-530.

[9] สง่า คงสุพานิช. "การแปลงหน่วยคำภาษาไทยเป็นสัญลักษณ์แทนเสียงสำหรับงานสังเคราะห์เสียงจากประโยคภาษาไทย." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2540.

[10] สมศักดิ์ จันวัน. "ระบบวิเคราะห์โครงสร้างภาษาไทยด้วยคอมพิวเตอร์." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2534.

[11] สรศักดิ์ ไทยแท้. "การตัดคำไทยโดยใช้ดิกชันนารีที่มีโครงสร้างข้อมูลแบบแฮชชิ่ง."

วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541.

[12] สิงห์ ตรงงาม. "ระบบการวิเคราะห์ประโยคภาษาไทยที่มีการละประธานที่ซ้ำกันในประโยค." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2540.

[13] ศังกรศรัณย์ ล่องชูผล. "การวิเคราะห์ประโยคภาษาไทยจากย่อหน้าเพื่อการแปลภาษาด้วยคอมพิวเตอร์." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2538.

[14] อุปกิตศิลปสาร, พระยา. 2539. **หลักภาษาไทย : อักขรวิธี วจีวิภาค วากยสัมพันธ์ ฉันทลักษณ์.** กรุงเทพฯ : ไทยวัฒนาพานิช.

[15] Aroonmanakun W. "Collocation and Thai Word Segmentation." Proceeding of SNLP-Oriental COCOSDA, 2002. pp. 68-75

[16] Theeramunkong T., Usanavasin S. "Non-Dictionary-Based Thai Word Segmentation Using Decision Trees." The fourth Symposium on Natural Language Processing, 2000.

[17] Microsoft. "Fonts and Text." [Online]. Available : http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/fontext_8ctw.asp. 2004.

ภาคผนวก

ภาคผนวก ก

อักษรและเครื่องหมายในภาษาไทย

ที่มา : พระยาอุปกิตศิลปสาร. "หลักภาษาไทย : อักขรวิธี วจีวิภาค วากยสัมพันธ์ ฉันทลักษณ์."
สำนักพิมพ์ ไทยวัฒนาพานิช กรุงเทพฯ : 2539.

อักษรและเครื่องหมายในภาษาไทย

พยัญชนะ

พยัญชนะไทยมี 44 ตัว ดังนี้

ตารางที่ ก-1 แสดงพยัญชนะในภาษาไทย

ตัวที่	พยัญชนะ	คำอ่าน	ตัวที่	พยัญชนะ	ตัวอย่างคำที่ใช้
1	ก	กอ-ไก่	23	ท	ทอ-ทหาร
2	ข	ขอ-ไข่	24	ธ	ธอ-ธง
3	ฃ	ฃอ-ฃวด	25	น	นอ-หนู
4	ค	คอ-ควาย	26	บ	บอ-ใบไม้
5	ฅ	ฅอ-ฅน	27	ป	ปอ-ปลา
6	ฆ	ฆอ-ระฆัง	28	ผ	ผอ-ผึ้ง
7	ง	งอ-งู	29	ฝ	ฝอ-ฝา
8	จ	จอ-จาน	30	พ	พอ-พาน
9	ฉ	ฉอ-ฉิ่ง	31	ฟ	ฟอ-ฟัน
10	ช	ชอ-ช้าง	32	ภ	ภอ-สำเภา
11	ซ	ซอ-โซ่	33	ม	มอ-ม้า
12	ฌ	ฌอ-เฌอ	34	ย	ยอ-ยักษ์
13	ญ	ญอ-หญิง	35	ร	รอ-เรือ
14	ฎ	ฎอ-ชฎา	36	ล	ลอ-ลิง
15	ฏ	ฏอ-ปฏัก	37	ว	วอ-แหวน
13	ฐ	ฐอ-ฐาน	38	ศ	ศอ-ศาลา
17	ฑ	ฑอ-มณโฑ	39	ษ	ษอ-ฤาษี
18	ฒ	ฒอ-เฒ่า	40	ส	สอ-เสือ
19	ณ	ณอ-เณร	41	ห	หอ-หีบ
20	ด	ดอ-เด็ก	42	ฬ	ฬอ-จุฬา
21	ต	ตอ-เต่า	43	อ	ออ-อ่าง
22	ถ	ถอ-ถุง	44	ฮ	ฮอ-นกฮูก

หมายเหตุ ปัจจุบันนี้ พยัญชนะ “ฃ” และ “ฅ” ได้ยกเลิกไม่ใช้แล้ว

สระ

สระมี 21 รูป ดังนี้

ตารางที่ ก-2 แสดงสระในภาษาไทย

ตัวที่	สระ	คำเรียก	การใช้
1	ะ	วิสรรชนีย์	สำหรับประหลังเป็นสระ อะ และประสมกับสระอื่นเป็นสระ เอะ แอะ โอะ เอาะ เออะ เอียะ เอือะ อัวะ
2	๎	ไม้ผัด หรือ หันอากาศ	สำหรับเขียนข้างบนเป็นสระ อะ เมื่อมีตัวสะกด และประสมกับสระรูปอื่นเป็น สระ อัวะ อัว
3	๘	ไม้ไต่คู้	สำหรับเขียนข้างบนแทนวิสรรชนีย์บางตัวที่มีตัวสะกด เช่น เอ็น แอ็น อ็อน ฯลฯ และใช้ประสมกับตัว ก เป็นสระ เอาะ มีไม้โท คือ ก็
4	า	ลากข้าง	สำหรับเขียนข้างหลังเป็นสระ อา และประสมกับรูปอื่นเป็นสระ เอาะ อำ เอา
5	ิ	พินทุ์ อิ	สำหรับเขียนข้างบนเป็นสระ อิ และประสมกับรูปอื่นเป็นสระ อี อึ อื เอียะ เอีย เอือะ เอือ และใช้แทนตัว อ ของสระ เออ เมื่อมีตัวสะกดก็ได้ เช่น เกอน เป็น เกิน ฯลฯ
6	์	ฝนทอง	สำหรับเขียนข้างบนพินทุ์ อิ เป็นสระ อี และประสมกับรูปอื่นเป็นสระ เอียะ เอีย
7	๐	นฤคหิต หรือ หยาดน้ำค้าง	สำหรับเขียนข้างบนลากข้างเป็นสระ อำ และบนพินทุ์ อิ เป็นสระ อึ
8	็	ฟันหนู	สำหรับเขียนบนพินทุ์ อิ เป็นสระ เอือ และประสมกับสระอื่นเป็นสระ เอือะ เอือ
9	ุ	ตีนเหยียด	สำหรับเขียนข้างล่างเป็นสระ อุ
10	ู	ตีนคู้	สำหรับเขียนข้างล่างเป็นสระ อู
11	เ	ไม้หน้า	สำหรับเขียนข้างหน้า รูปเดียวเป็นสระ เอ สองรูปเป็นสระ แอ และประสมกับรูปอื่นเป็นสระ เอะ แอะ เอาะ เออะ เออ เอียะ เอีย เอือะ เอือ เอา
12	ใ	ไม้ม้วน	สำหรับเขียนข้างหน้าเป็นสระ ใอ
13	ไ	ไม้มลาย	สำหรับเขียนข้างหน้าเป็นสระ ไอ
14	โ	ไม้โอ	สำหรับเขียนข้างหน้าเป็นสระ โอ
15	อ	ออ	สำหรับเขียนข้างหลังเป็นสระ ออ และประสมกับรูปอื่นเป็นสระ อือ (เมื่อไม่มีตัวสะกด) เออะ เออ เอือะ เอือ
16	ย	ยอ	สำหรับประสมกับรูปอื่นเป็นสระ เอียะ เอีย
17	ว	วอ	สำหรับประสมกับรูปอื่นเป็นสระ อัวะ อัว
18	ฤ	รึ	สำหรับเขียนเป็นสระ ฤ
19	ฤๅ	รือ	สำหรับเขียนเป็นสระ ฤๅ
20	ฦ	ลึ	สำหรับเขียนเป็นสระ ฦ
21	ฦๅ	ลือ	สำหรับเขียนเป็นสระ ฦๅ

สระมี 32 เสียง ซึ่งประกอบมาจากรูปสระ 21 รูป โดยแบ่งเป็นสระเสียงสั้น (รัสสระ) และสระเสียงยาว (ทีฆสระ) จะจับกันเป็นคู่ๆ ดังนี้

ตารางที่ ก-3 แสดงสระเสียงสั้นและสระเสียงยาว

สระเสียงสั้น	สระเสียงยาว
อะ	อา
อิ	อี
อึ	อื
อุ	อู
เอะ	เอ
แอะ	แอ
โอะ	โอ
เอาะ	ออ
เออะ	เออ

สระเสียงสั้น	สระเสียงยาว
เอียะ	เอีย
เอือะ	เอือ
อัวะ	อัว
ฤ	ฤๅ
ฦ	ฦๅ
อำ	
ไอ	
ใอ	
เอา	

วรรณยุกต์

วรรณยุกต์ มีรูปต่างกัน 4 รูป ดังนี้

(1) ่ เรียกว่า ไม้เอก

(2) ้ เรียกว่า ไม้โท

(3) ๊ เรียกว่า ไม้ตรี

(4) ๋ เรียกว่า ไม้จัตวา

แต่วรรณยุกต์จะมีทั้งหมด 5 เสียง คือ เสียงสามัญ (ไม่มีรูป) เสียงเอก เสียงโท เสียงตรี และเสียงจัตวา ซึ่งโดยพื้นฐานแล้ว แต่ละเสียงจะมีรูปพื้นฐานของตนเอง กล่าวคือ เสียงเอก จะมีรูปพื้นฐานเป็น ไม้เอก เสียงโท จะใช้ ไม้โท เสียงตรี จะใช้ไม้ตรี และเสียงจัตวา จะใช้ไม้จัตวา ยกเว้นแต่เสียงสามัญ ที่ไม่มีรูป แต่บางครั้ง คำไทยอาจมีเสียงที่ไม่ตรงกับรูปก็ได้ เช่น มีรูปโท แต่ใช้เสียงตรี (เช่น คำว่า ค้า) หรืออาจไม่มีรูป แต่เป็นเสียงอื่นๆที่ไม่ใช่เสียงสามัญก็ได้ เช่น หมึก (เสียงเอก) ชัก (เสียงโท) ขา (เสียงจัตวา) เป็นต้น

ตัวการันต์

ตัวการันต์ คือ พยัญชนะสุดท้ายที่ไม่ต้องอ่านออกเสียงเป็นตัวสะกด โดยมากจะมีไม้ทัณฑฆาตบังคับข้างบน หรือ บางครั้งอาจไม่มีก็ได้ แต่ไม่ออกเสียง

เครื่องหมายวรรคตอน

เครื่องหมายวรรคตอนของไทย จะมีชื่อเรียกต่างๆกัน ดังนี้

ตารางที่ ก-4 แสดงเครื่องหมายวรรคตอนในภาษาไทย

ตัวที่	รูปเครื่องหมาย	คำเรียก	การใช้
1	,	จุลภาค	บอกเว้นวรรคตอนในประโยคเดียวกัน
2	;	อัฒภาค	ใช้ในประโยคซ้อน หรือใช้แยกประโยค
3	.	มหัพภาค	ใช้บอกจบประโยค หรือใช้กำกับอักษรย่อ
4	?	ปรัศนี	ใช้ในประโยคคำถาม
5	!	อัศเจรีย์	ใช้ใส่หลังคำอุทาน
6	(. . .)	นขลิขิต	เครื่องหมายวงเล็บ ใช้คร่อมข้อความพิเศษ
7	“ ...”	อัญประกาศ	เครื่องหมายคำพูด ใช้คร่อมข้อความสำคัญหรือคำพูด มี 2 แบบ คือ อัญประกาศคู่ “...” และ อัญประกาศเดี่ยว ‘...’
8	-	ยัติภังค์	ใช้ขีดเพื่อแยกคำให้ห่างกัน
9	ๆ	ยมก	ใช้แสดงว่าต้องอ่านซ้ำอีกครั้งหนึ่ง
10	ฯ	ไปยาลน้อย	ใช้ละคำที่รู้จักกันทั่วไปแล้ว เขียนไว้ท้ายคำ
11	ฯลฯ หรือ ...	ไปยาลใหญ่	ใช้ละตอนปลายไม่มีกำหนด เขียนไว้ท้ายคำ จะใช้รูป ฯลฯ และใช้ละตอนกลาง หรือ ตอนที่ไม่ต้องการอ่าน ใช้รูป ฯลฯ หรือ ...
12	=	เสมอภาค หรือ สมพล	ใช้แสดงว่าข้อความทั้งสองฝ่ายเสมอกัน
13	.....	สัญประกาศ	ใช้ขีดไว้ใต้ข้อความที่สำคัญ
14	„	บุพสัญญา	ใช้แทนคำที่อยู่ข้างบน
15	ย่อหน้าขึ้นบรรทัดใหม่	มหรรถสัญญา	เป็นเครื่องหมายขึ้นต้นข้อความใหญ่ โบราณใช้ฟันหนูฟองมัน (⊙) แต่ปัจจุบันใช้วิธีย่อหน้าขึ้นบรรทัดใหม่

ภาคผนวก ข

รหัสแอสกี (ASCII)

รหัสแอสกี (ASCII)

ตารางที่ ข-1 แสดงตัวอักษรไทยในรหัสแอสกี

View by: Decimal Name: nbspace Unicode: 00A0
Key:a0160 Dec: 160 Hex: A0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
		...	'	ั	็	+	์	'	ั	็	+	์			
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
	•	ั	็	ิ	ี	ึ	ื	'	ั	็	'	์			ฯ
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
	ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
ภ	ม	ย	ร	ฤ	ล	ฦ	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
ะ	ั	า	ำ	ิ	ี	ึ	ื	ุ	ู	ฺ	฿	เ	แ	โ	ใ
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
ไ	ๅ	ๆ	็	่	้	๊	๋	์	๎	๏	๐	๑	๒	๓	๔
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	——	—			๏	

ภาคผนวก ค

ความถี่ของการใช้ตัวอักษรไทยในแบบต่างๆ

ที่มา : ยืน ภู่วรวรรณ. "การวิเคราะห์ข้อมูลคำไทย." การประชุมวิชาการทางวิศวกรรมไฟฟ้า 8 สถาบันอุดมศึกษา ครั้งที่ 7. ธันวาคม 2527. หน้า ค-290 – ค-303.

ความถี่ของการใช้ตัวอักษรไทยในแบบต่างๆ

ตารางที่ ค-1 แสดงการแจกแจงความถี่การใช้ตัวอักษร

ตัวอักษร	จำนวนตัว	เปอร์เซ็นต์	ตัวอักษร	จำนวนตัว	เปอร์เซ็นต์
.	670	0.1446	ฤ	162	0.0850
ก	19963	4.3092	ล	10365	2.2374
ข	5599	1.2086	ว	12882	2.7807
ค	8332	1.7985	ศ	1615	0.3486
ฆ	111	0.0240	ษ	1139	0.2459
ง	19176	4.1393	ส	8511	1.8372
จ	7259	1.5669	ห	9802	2.1158
ฉ	519	0.1120	ฬ	63	0.0136
ช	4366	0.9424	อ	20346	4.3919
ซ	1049	0.2264	ฮ	120	0.0259
ฌ	1	0.0002	ะ	8765	1.8920
ญ	1239	0.2674	ั	32908	7.1035
ฎ	98	0.0212	า	3658	0.7896
ฏ	210	0.0453	เ	18084	3.9036
ฐ	465	0.1004	แ	6846	0.4773
ฑ	50	0.0108	โ	2321	0.5010
ฒ	167	0.0360	ใ	5346	1.1540
ณ	1663	0.3590	ไ	7068	1.5257
ด	10813	2.3341	ๅ	825	0.1781
ต	8407	1.8147	็	89	0.0192
ถ	2386	0.5150	่	4180	0.9023
ท	11004	2.3753	้	3993	0.8691
ธ	1361	0.2938	๊	8887	1.9183
น	28917	6.2420	๋	12388	2.6741
บ	7892	1.7036	์	2650	0.5720
ป	8266	1.7842	๎	5443	1.1749
ผ	2272	0.4904	๏	13129	2.8340
ฝ	117	0.0684	๐	4848	1.0465
พ	5548	1.1976	๑	22600	4.8784
ฟ	505	0.1090	๒	16684	8.6014
ภ	1101	0.2377	๓	165	0.0356
ม	16406	3.5414	๔	159	0.0343
ย	13854	2.9905	๕	2567	0.5541
ร	22720	4.9043	๖	1953	0.4216

ตารางที่ ค-2 แสดงการแจกแจงความถี่ตัวอักษรที่ใช้เป็นตัวแรกของคำ

ตัวอักษร	จำนวนตัวที่ใช้ทั้งหมด	จำนวนตัวแรก	เปอร์เซ็นต์
	670	0	0.0000
ก	19963	9648	48.3292
ข	5599	3955	70.6376
ค	8332	5804	69.6591
ฆ	111	16	14.4144
ง	19176	733	3.8225
จ	7259	4989	68.7285
ฉ	519	346	66.6667
ช	4366	2269	51.9698
ซ	1049	793	75.5958
ฌ	1	0	0.0000
ญ	1239	57	4.6005
ฎ	98	3	3.0612
ฏ	210	0	0.0000
ฐ	465	91	19.5699
ฑ	50	0	0.0000
ฒ	167	0	0.0000
ณ	1663	57	3.4275
ด	10813	2494	23.0648
ต	8407	4515	53.7052
ถ	2386	1512	63.3697
ท	11004	7918	71.9738
ธ	1361	378	27.7737
น	28937	5155	17.8269
บ	7892	2441	30.9301
ป	8266	3414	41.3017
ผ	2272	2007	88.3363
ฝ	317	306	96.5300
พ	5548	2984	53.7851
ฟ	505	260	51.4851
ภ	1101	649	58.9464
ม	16406	5038	30.7083
ย	13854	1697	12.2492
ร	22720	4359	19.1857
ฤ	162	56	34.5679
ล	10365	1924	18.5625
ว	12882	3127	24.2742
ศ	1615	657	40.6811
ษ	1139	0	0.0000
ส	8511	6069	71.3077
ห	9802	4846	49.4389
ฬ	63	0	0.0000
อ	20345	5347	26.2803
ฮ	120	50	41.6667
ะ	8765	0	1.0000
ั	32908	0	1.0000
า	3658	0	1.0000
ิ	18084	16915	93.5357
ี	6846	6690	97.7213
ึ	2321	1875	80.7841
ื	5346	5343	99.9439
ุ	7068	6766	95.7272
ู	825	0	0.0000
เ	89	15	16.8539
แ	4180	0	0.0000
โ	3993	0	0.0000
ใ	8987	0	0.0000
ไ	12388	0	0.0000
็	2650	0	0.0000
่	5443	0	0.0000
้	13129	0	0.0000
๊	4848	0	0.0000
๋	22600	0	0.0000
์	16684	0	0.0000
ๆ	165	0	0.0000
ฯ	159	0	0.0000
๏	2517	0	0.0000
๐	1953	0	0.0000

ตารางที่ ค-3 แสดงการแจกแจงความถี่ตัวอักษรที่ใช้เป็นตัวสุดท้ายของคำ

ตัวอักษร	จำนวนตัวที่ใช้ทั้งหมด	จำนวนตัวสุดท้าย	เปอร์เซ็นต์
ก	19963	7095	35.5408
ข	5599	189	3.3756
ค	8332	409	4.9088
ฆ	111	25	22.5225
ง	19176	17099	89.1684
จ	7259	1233	16.9856
ฉ	519	2	0.3854
ช	4366	325	7.4439
ซ	1049	15	1.4299
ฌ	1	0	0.0000
ญ	1239	389	31.3156
ฎ	98	54	55.1020
ฏ	210	60	28.5714
ฐ	465	264	56.7742
ฑ	50	0	0.0000
ฒ	167	2	1.1976
ณ	1663	750	45.0992
ด	10813	4242	39.2268
ต	8407	642	7.6365
ถ	2386	502	21.0394
ท	11004	567	5.1527
ธ	1361	139	10.2131
น	28917	18347	61.4915
บ	7892	4225	53.5352
ป	8266	1661	20.0944
ผ	2272	1	0.0440
ฝ	317	1	0.3155
พ	5548	503	9.0663
ฟ	505	146	28.9109
ภ	1101	8	0.7266
ม	16406	5883	35.8588
ย	13854	6348	45.8207
ร	22720	4857	21.3776
ฤ	162	0	0.0000
ล	10365	1110	10.7091
ว	12882	3371	26.1683
ศ	1615	506	31.3313
ษ	1139	202	17.7349
ส	8511	273	3.2076
ห	9802	8	0.0816
ฬ	63	0	0.0000
อ	20346	4745	23.3215
ฮ	120	0	0.0000
◌ั	8765	7386	84.2670
◌า	32908	10416	31.6519
◌ำ	3658	2217	60.6069
เ◌	18084	0	0.0000
แ◌	6846	0	0.0000
◌ิ	2321	0	0.0000
◌ี	5346	0	0.0000
◌ึ	7068	0	0.0000
◌ื	825	825	100.0000
◌ุ	89	74	83.1461
◌ู	4180	374	8.9474
◌่	3993	475	11.8958
◌้	8987	952	10.7123
◌๊	12388	3708	29.9322
◌๋	2650	17	0.6415
◌์	5443	0	0.0000
◌ั	13129	1	0.0076
◌็	4848	1552	32.0132
◌	22600	8491	37.5708
◌	16684	7713	46.2299
◌	165	6	3.6364
◌	159	8	5.0314
◌	2567	2397	93.3775
๒	1953	0	0.0000

ตารางที่ ค-4 แสดงการแจกแจงความถี่ของรูปแบบโครงสร้างของคำไทย

CCCG	159	CVSV	1166
CCCS	456	CVSVS	739
CCCSV	114	CVSVSS	229
CCCV	123	CVSVSSG	187
CCCVS	137	CVSVSSV	110
CCS	911	CVT	6934
CCTV	946	CVTS	4457
CCTVS	778	CVTSG	105
CCV	2727	CVTV	353
CCVCV	168	CVVC	227
CCVS	6241	CVVCS	153
CCVSS	233	VC	4739
CCVSV	101	VCC	4194
CCVSVS	512	VCCG	127
CCVSVSV	109	VCCS	593
CCVTS	813	VCCSG	121
CGCG	104	VCCSS	108
CS	4552	VCCT	519
CSTVS	791	VCCTS	111
CSVS	151	VCCTV	123
CSVT	742	VCCV	434
CTS	542	VCCVS	479
CTV	3544	VCCVSS	374
CTVS	4127	VCCVTSS	106
CTVSG	211	VCCVTVS	252
CV	12272	VCCVV	518
CVCS	150	VCS	380
CVCSVS	203	VCT	7859
CVCV	118	VCTS	2033
CVCVS	584	VCTV	1139
CVS	27490	VCV	2806
CVSG	397	VCVS	4805
CVSGS	515	VCVSS	1838
CVSSS	302	VCVTS	256
CVSSSG	246	VCVTSS	392
CVSSSV	110	VCVTV	364
CVSSV	213	VCVTVS	241
CVSSVS	241	VCVV	153
CVSSVSS	168	VTV	261

โดย C คือ พยัญชนะ

V คือ สระ

T คือ วรรณยุกต์

S คือ ตัวสะกด โดยที่ $S \subseteq C$

G คือ ตัวการันต์

ตารางที่ ค-5 แสดงการแจกแจงความถี่ของจำนวนตัวอักษรภายในคำ

ความยาวของคำ	จำนวนคำ	เปอร์เซนต์
1	55	0.0412
2	21796	16.3183
3	61391	45.9624
4	31012	23.2191
5	11532	8.6338
6	4191	3.1377
7	1857	1.3903
8	954	0.7142
9	287	0.2149
10	244	0.1827
11	152	0.1138
12	36	0.0270
13	30	0.0225
14	20	0.0150
15	0	0.0000

ภาคผนวก ง

ข้อความที่ใช้ในการทดลอง

ข้อความชุด A

1 ปัจจุบันจะพบว่าคอมพิวเตอร์มีหลากหลายลักษณะ หลากหลายรูปแบบ ทั้งคอมพิวเตอร์ขนาดพกพาคอมพิวเตอร์แบบตั้งโต๊ะ คอมพิวเตอร์แบบกระเป๋าหิ้ว คอมพิวเตอร์ขนาดใหญ่ เช่น คอมพิวเตอร์เมนเฟรม หรือซุปเปอร์คอมพิวเตอร์ แต่ไม่ว่าจะเป็นรูปแบบใดก็ตาม คอมพิวเตอร์ก็มีความหมายที่ชัดเจนในตัวของมันเอง คือ เครื่องคำนวณ ในรูปของอุปกรณ์อิเล็กทรอนิกส์ ที่สามารถรับข้อมูล และคำสั่ง ผ่านอุปกรณ์รับข้อมูล แล้วนำข้อมูลและคำสั่งนั้น ไปประมวลผลด้วยหน่วยประมวลผลเพื่อให้ได้ผลลัพธ์ที่ต้องการ และแสดงผลผ่านอุปกรณ์แสดงผล ตลอดจนสามารถบันทึกรายการต่างๆ ไว้เพื่อใช้งานได้ด้วยอุปกรณ์บันทึกข้อมูลสำรอง คอมพิวเตอร์จึงสามารถมีรูปร่างอย่างไรก็ได้ ไม่จำเป็นต้องเป็นรูปร่างอย่างที่เราคุ้นเคย หรือพบเห็น ตัวอย่างเช่น เครื่องฝากถอนเงินอัตโนมัติ ก็ถือว่าเป็นเครื่องคอมพิวเตอร์รูปแบบหนึ่ง

2 เหตุผลที่นำคอมพิวเตอร์มาใช้งาน สามารถบันทึกข้อมูลต่างๆ ได้รวดเร็ว เช่น การใช้เครื่องอ่านรหัสแท่ง อ่านเวลาเข้าออก ของพนักงาน และคิดราคาสินค้า ในห้างสรรพสินค้า สามารถเก็บข้อมูลจำนวนมาก ไว้ในฐานข้อมูล เพื่อใช้งานได้ทันที สามารถนำข้อมูลที่เก็บไว้มาคำนวณทางสถิติ แยกประเภท จัดกลุ่ม ทำรายงานลักษณะต่างๆ ได้ โดยระบบประมวลผลข้อมูล สามารถส่งข้อมูลจากที่หนึ่ง ไปยังอีกที่หนึ่งได้อย่างรวดเร็ว โดยอาศัยเทคโนโลยีสื่อสารข้อมูล สามารถจัดทำเอกสารต่างๆ ได้อย่างรวดเร็ว ด้วยระบบประมวลผลคำ ซึ่งเป็นส่วนหนึ่งของ ระบบสำนักงานอัตโนมัติ การนำมาใช้งานทั้งด้านการศึกษา การวิจัย การใช้งานธุรกิจ งานการเงิน ธนาคาร และงานของภาครัฐต่างๆ เช่น การนำคอมพิวเตอร์มาใช้กับงานบัญชี งานบริหารสำนักงาน งานเอกสาร งานการเงิน การจองตั๋วเครื่องบิน รถไฟ การควบคุมระบบอัตโนมัติต่างๆ เช่น ระบบจราจร, ระบบเปิดปิดน้ำของเขื่อน การใช้เพื่องานวิเคราะห์ต่างๆ เช่น การวิเคราะห์สภาวะดินฟ้าอากาศ สภาพของดิน น้ำ เพื่อการเกษตร การใช้คอมพิวเตอร์เพื่อจำลองรูปแบบ เช่น การจำลองในงานวิทยาศาสตร์ จำลองโมเลกุล จำลองรูปแบบการฝึกขับเครื่องบิน การใช้คอมพิวเตอร์นันทนาการ เช่นการเล่นเกม การดูหนัง ฟังเพลง การใช้คอมพิวเตอร์ร่วมกับเทคโนโลยีล้ำสมัยอื่นๆ เทคโนโลยีสื่อสารข้อมูล เกิดเครือข่ายอินเทอร์เน็ต เป็นต้น

3 ประเภทของเครื่องคอมพิวเตอร์ การจัดแบ่งประเภทของ เครื่องคอมพิวเตอร์ จะอาศัยคุณสมบัติต่างๆ เช่น ความเร็วของการประมวลผล และขนาดความจำ ของหน่วยบันทึกข้อมูล ซึ่งสามารถแบ่งได้ เป็น 4 ประเภท ได้แก่ ซูเปอร์คอมพิวเตอร์ เมนเฟรมคอมพิวเตอร์ มินิคอมพิวเตอร์ และไมโครคอมพิวเตอร์ ทั้งนี้คุณสมบัติที่นำมาแบ่งประเภทประกอบด้วย ขนาดของเวิร์ด ในความหมายของคอมพิวเตอร์ก็คือ คำ หรือ ตัวอักษร อันเป็นสัญญาณไฟฟ้า

ที่รับ หรือส่งเข้าสู่ระบบ โดยจะนับเป็นจำนวนครั้งละกี่บิต ดังนั้นคอมพิวเตอร์ที่มีประสิทธิภาพสูง จะสามารถรับส่งข้อมูลจำนวนบิตมากกว่าตามไปด้วย นอกจากนี้ ยังรวมถึงคุณสมบัติที่เกี่ยวกับความเร็วในการประมวลผล ความจุของหน่วยความจำ

4 ซูเปอร์คอมพิวเตอร์ เป็นคอมพิวเตอร์ที่มีสมรรถนะในการทำงานสูงกว่า คอมพิวเตอร์แบบอื่น ดังนั้นจึงมีผู้เรียกอีกชื่อหนึ่งว่า คอมพิวเตอร์สมรรถนะสูง คอมพิวเตอร์ประเภทนี้ สามารถคำนวณเลขที่มีจุดทศนิยม ด้วยความเร็วสูงมาก ขนาดหลายร้อยล้านจำนวนต่อวินาที งานที่ให้คอมพิวเตอร์ประเภทนี้ทำแค่ 1 วินาที ถ้าหากเอามาให้คนอย่างเราคิด อาจจะต้องใช้เวลานานกว่าร้อยปี ด้วยเหตุนี้ จึงเหมาะที่จะใช้คอมพิวเตอร์ประเภทนี้ เมื่อต้องมีการคำนวณมากๆ อย่างเช่น งานวิเคราะห์ภาพถ่าย จากดาวเทียมอุตุนิยมวิทยา หรือดาวเทียมสำรวจทรัพยากร งานวิเคราะห์พยากรณ์อากาศ งานทำแบบจำลองโมเลกุล ของสารเคมี งานวิเคราะห์โครงสร้างอาคาร ที่ซับซ้อน คอมพิวเตอร์ประเภทนี้ มีราคาค่อนข้างแพง ปัจจุบันประเทศไทย มีเครื่องซูเปอร์คอมพิวเตอร์ ใช้ในงานวิจัย อยู่ที่ห้องปฏิบัติการคอมพิวเตอร์สมรรถภาพสูง ศูนย์เทคโนโลยีอิเล็กทรอนิกส์ และคอมพิวเตอร์แห่งชาติ ผู้ใช้เป็นนักวิจัยด้านวิศวกรรม และวิทยาศาสตร์ทั่วประเทศ บริษัทผู้ผลิตที่เด่นๆ ได้แก่ บริษัทเครย์ รีเสิร์ซ, บริษัท เอ็นอีซี เป็นต้น

5 เมนเฟรมคอมพิวเตอร์ เป็นคอมพิวเตอร์ที่มีสมรรถนะสูงมาก แต่ยังต่ำกว่าซูเปอร์คอมพิวเตอร์ คือปกติสามารถทำงานได้รวดเร็ว หลายสิบล้านคำสั่งต่อวินาที สำหรับสาเหตุที่ได้ชื่อว่าเมนเฟรมคอมพิวเตอร์ ก็เพราะครั้งแรกที่สร้างคอมพิวเตอร์ลักษณะนี้ได้สร้างไว้บนฐานรองรับที่เรียกว่า คัสซี่ โดยมีชื่อเรียกฐานรองรับนี้ว่า เมนเฟรม นั่นเอง เหมาะกับการใช้งาน ทั้งในด้านวิศวกรรม วิทยาศาสตร์ และธุรกิจ โดยเฉพาะงานที่เกี่ยวข้องกับข้อมูลจำนวนมากๆ เช่น งานธนาคาร ซึ่งต้องตรวจสอบบัญชีลูกค้าหลายคน งานของสำนักงานทะเบียนราษฎร์ ที่เก็บรายชื่อประชาชนประมาณ 60 ล้านคน พร้อมรายละเอียดต่างๆ งานจัดการบันทึกการส่งเงิน ของผู้ประกันตนหลายล้านคน ของสำนักงานประกันสังคม กระทรวงแรงงาน ในปัจจุบัน ความนิยมใช้เครื่องเมนเฟรม ในหน่วยงานต่างๆ ได้ลดน้อยลงมาก เพราะราคาเครื่องค่อนข้างแพง การใช้งานค่อนข้างยาก และมีผู้รู้ด้านนี้ค่อนข้างน้อย สถานศึกษาที่มีเครื่องระดับนี้ไว้ใช้สอน ก็มีเพียงไม่กี่แห่ง เหตุผลสำคัญอีกประการหนึ่งคือ คอมพิวเตอร์ขนาดเล็กกว่า ได้รับการพัฒนาให้มีสมรรถนะมากขึ้น จนสามารถทำงานได้เท่ากับเครื่องเมนเฟรม แต่ราคาถูกกว่า อย่างไรก็ตามเครื่องเมนเฟรม ยังคงมีความจำเป็น ในงานที่ต้องใช้ข้อมูลมากๆ พร้อมๆ กันอยู่ต่อไปอีกทั้งนี้เพราะ เครื่องเมนเฟรมสามารถพ่วงต่อ และควบคุมอุปกรณ์รอบข้าง เช่น เครื่องพิมพ์ เครื่องขับเทปแม่เหล็ก เครื่องขับจานแม่เหล็ก ฯลฯ ได้เป็นจำนวนมากในเวลาเดียวกัน

6 มินิคอมพิวเตอร์ เป็นคอมพิวเตอร์ที่มีสมรรถนะน้อยกว่าเครื่องเมนเฟรม คือทำงานได้ช้ากว่า และควบคุมอุปกรณ์รอบข้างได้น้อยกว่า อย่างไรก็ตามจุดเด่นสำคัญ ของเครื่องมินิคอมพิวเตอร์ ก็คือ ราคาย่อมเยากว่าเมนเฟรม การใช้งานก็ไม่ต้องใช้ บุคลากรมากนัก

นอกจากนั้น ยังมีผู้ที่รู้วิธีใช้มากกว่าด้วย เพราะเครื่องประเภทนี้ มีใช้ตามสถานศึกษาระดับอุดมศึกษาหลายแห่ง มินิคอมพิวเตอร์ เหมาะกับงานหลากหลายประเภท คือใช้ได้ทั้งในงานวิศวกรรม วิทยาศาสตร์ อุตสาหกรรม เครื่องที่มีใช้ตามหน่วยงานราชการระดับกรมส่วนใหญ่ มักจะเป็นเครื่องประเภทนี้

7 ไมโครคอมพิวเตอร์ เป็นคอมพิวเตอร์ขนาดเล็ก และใช้ทำงานคนเดียว จึงนิยมเรียกอีกชื่อหนึ่งว่าคอมพิวเตอร์ส่วนบุคคล เป็นคอมพิวเตอร์ใช้งานที่พบได้อย่างแพร่หลาย จัดว่าเป็นเครื่องคอมพิวเตอร์ขนาดเล็ก ทั้งระบบใช้งานครั้งละคนเดียว หรือใช้งานในลักษณะเครือข่าย แบ่งได้หลายลักษณะตามขนาด เช่นเครื่องคอมพิวเตอร์ส่วนบุคคลแบบตั้งโต๊ะ หรือแบบพกพา หรือแบ่งตามผู้ผลิต ได้แก่ เครื่องกลุ่ม และแมคอินทอช เป็นต้น คอมพิวเตอร์ประเภทนี้ ที่เป็นตัวการผลักดันให้เกิด การเปลี่ยนแปลงขนาดใหญ่ในโลกคอมพิวเตอร์ คือ ทำให้เกิดความสนใจ ในเรื่องคอมพิวเตอร์แพร่หลายไปสู่คนทุกอาชีพ และทุกวัย อย่างเช่นในเมืองไทยนี้เอง ก็มีนายแพทย์จำนวนมาก สนใจซื้อคอมพิวเตอร์มาศึกษา จนถึงขั้นเขียนโปรแกรมขึ้นมาช่วยงานของโรงพยาบาลได้ อดีตปลัดกระทรวงสำคัญท่านหนึ่ง ก็ใช้คอมพิวเตอร์คล่อง จนถึงขั้นสามารถใช้เก็บข้อมูลสำคัญๆ ของกระทรวง ไว้ใช้ในการบริหารงานได้ ผู้บริหารงานราชการอีกหลายท่าน ก็มีความสามารถในด้านการใช้คอมพิวเตอร์ ในระดับที่ผู้เชี่ยวชาญก็ยังต้องอาย

8 องค์ประกอบของคอมพิวเตอร์ จากความหมายของ คอมพิวเตอร์ ก็คงจะมองออกว่าคอมพิวเตอร์จะทำงานได้ ต้องประกอบด้วยส่วนการทำงานอะไรบ้าง นั่นคือ คอมพิวเตอร์ต้องประกอบด้วยส่วนรับข้อมูลและคำสั่ง ส่วนประมวลผล ส่วนที่ใช้แสดงผลลัพธ์จากการประมวลผล และส่วนในการเก็บบันทึกข้อมูล ซึ่งเรียกรวมกันว่า องค์ประกอบของคอมพิวเตอร์ อันได้แก่ ส่วนที่ทำหน้าที่รับข้อมูล และคำสั่ง เรียกว่า หน่วยรับข้อมูล ส่วนที่นำเอาข้อมูลและคำสั่งไปประมวลผล เรียกว่า หน่วยประมวลผลกลาง ส่วนที่ทำหน้าที่แสดงผลลัพธ์เรียกว่า หน่วยแสดงผล ส่วนที่ทำหน้าที่บันทึกคำสั่งและข้อมูล อย่างถาวร เรียกว่า หน่วยความจำรอง นอกจากส่วนประกอบที่เกี่ยวข้องโดยตรงกับคอมพิวเตอร์ทั้ง 4 ส่วนยังมีส่วนประกอบอื่นๆ อีกดังนี้

9 ข้อมูล คือข้อมูลต่างๆ ที่เรานำมาให้คอมพิวเตอร์ทำการประมวลผลคำนวณ หรือกระทำการอย่างใดอย่างหนึ่งให้ได้มาเป็นผลลัพธ์ที่เราต้องการ ยกตัวอย่างเช่น ข้อมูลบุคลากรเกี่ยวกับรายละเอียดประวัติส่วนตัว ประวัติการศึกษาหรือ ประวัติการทำงาน ซึ่งอาจนำมาจำแนกเป็นรายงานต่างๆ เกี่ยวกับบุคลากรในหน่วยงานได้ หรือข้อมูลเกี่ยวกับตัวเลขมาตรไฟฟ้าของบ้านแต่ละหลัง ก็ใช้สำหรับคำนวณเป็นปริมาณไฟฟ้า ที่ใช้ในแต่ละเดือน แล้วคิดเป็นเงิน ที่จะต้องชำระให้กับการไฟฟ้าฯ ในปัจจุบันเราถือว่าข้อมูล มีความสำคัญอย่างยิ่ง ต่อการใช้งานคอมพิวเตอร์ ถ้าฮาร์ดแวร์หรือซอฟต์แวร์เสียหาย ไปยังหาซื้อใหม่ได้ แต่ถ้าหากข้อมูลเกิดการ

สูญหายแล้ว หน่วยงานอาจจะประสบปัญหาในการดำเนินงานได้ทันที

10 บุคลากร คือ เจ้าหน้าที่ปฏิบัติงานต่างๆ และผู้ใช้เครื่องคอมพิวเตอร์ในหน่วยงานนั้นๆ บุคลากรด้านคอมพิวเตอร์นั้น มีความสำคัญมาก เพราะการใช้เครื่องคอมพิวเตอร์ทำงานต่างๆ นั้นจะต้องมีการจัดเตรียมเปลี่ยนระบบ จัดเตรียมโปรแกรมดำเนินการต่างๆ หลายอย่าง ซึ่งไม่สามารถทำด้วยตัวเองได้ ถ้าหากไม่ใช่ผู้ที่รู้เรื่องคอมพิวเตอร์ มากนัก ดังนั้นเราจึงถือว่า บุคลากร เป็นส่วนประกอบที่สำคัญของ ระบบคอมพิวเตอร์ด้วย

11 ระเบียบ คู่มือ และ มาตรฐาน การนำคอมพิวเตอร์เข้ามาใช้ในหน่วยงานนั้น จะต้องไปสัมพันธ์กับเจ้าหน้าที่ และผู้ปฏิบัติงานจำนวนมาก บุคคลเหล่านี้บางคนก็เรียนรู้ได้เร็ว บางคนก็ช้า และนอกจากนั้นยังมีแนวคิด และทัศนคติที่แตกต่างกัน ดังนั้นเพื่อให้คนเหล่านี้ทำงานร่วมกันได้โดยไม่มีปัญหา จึงจำเป็นจะต้องมีระเบียบปฏิบัติ ให้เป็นแบบเดียวกัน มีการจัดทำคู่มือการใช้คอมพิวเตอร์ ให้ทุกคนเรียนรู้และใช้อ้างอิงได้ นอกจากนั้นเมื่อการใช้มาตรฐาน ด้านคอมพิวเตอร์และด้านอื่นๆ ที่เกี่ยวข้อง จะช่วยให้การประสานงาน ระหว่างหน่วยงานย่อยๆ ราบรื่นขึ้น การจัดซื้อจัดหา ตลอดจนการบำรุงรักษาเครื่องคอมพิวเตอร์ และซอฟต์แวร์ก็จะง่ายขึ้น เพราะทุกหน่วยงานใช้มาตรฐานเดียวกัน

12 หน่วยรับข้อมูล เป็นหน่วยที่ทำหน้าที่รับข้อมูลหรือคำสั่งเข้าสู่คอมพิวเตอร์เพื่อให้คอมพิวเตอร์ดำเนินการประมวลผล โดยอาศัยอุปกรณ์รับข้อมูลหลากรูปแบบ เช่น แป้นพิมพ์, เมาส์, บอลกลิ้ง, ก้านควบคุม ฯลฯ ข้อมูลที่นำเข้าคอมพิวเตอร์ เป็นได้ทั้งตัวอักษร ตัวเลข สัญลักษณ์ รูปทรง สี อุณหภูมิ เสียง ตลอดจนสิ่งอื่นๆ ที่สามารถส่งเข้าคอมพิวเตอร์เพื่อประมวลผล

13 หน่วยประมวลผลกลาง หน่วยประมวลผลกลาง เปรียบได้กับสมองของคอมพิวเตอร์ เป็นส่วนที่สำคัญที่สุด ทำหน้าที่เป็นศูนย์กลางการประมวลผลและควบคุมระบบต่างๆ ของคอมพิวเตอร์ให้ทุกหน่วยทำงานสอดคล้องสัมพันธ์กัน หลายท่านคงสงสัยว่า ไมโครโพรเซสเซอร์, ชิป, โพรเซสเซอร์ เหมือนหรือต่างจาก หน่วยประมวลผลกลาง หรือไม่ อย่างไร คำตอบก็คือ เหมือนกัน จะเรียกชื่ออะไรก็ได้ เนื่องจากส่วนประกอบภายในเป็นวงจรอิเล็กทรอนิกส์ที่ซับซ้อนจำนวนมาก มีทรานซิสเตอร์ประกอบกันเป็นวงจรหลายล้านตัว

* *ข้อความชุด A เป็นข้อความที่พบคำศัพท์ในพจนานุกรมทุกคำ ทั้งศัพท์ทั่วไปและศัพท์เฉพาะทาง (มีทั้งหมด 13 ย่อหน้า)*

ข้อความชุด B

1 เครื่องพิมพ์แบบกระทบ เครื่องพิมพ์แบบกระทบ มีหลายลักษณะ เป็นเครื่องพิมพ์ที่อาศัย การกดหัวพิมพ์กับแถบผ้าหมึก เพื่อให้เกิดตัวอักษร ได้แก่ เครื่องพิมพ์แบบเรียงจุด (Dot Matrix Printer) เป็นเครื่องพิมพ์ ที่ได้รับความนิยม โดยองค์ประกอบสำคัญได้แก่ หัวพิมพ์ (Print Head) ที่ประกอบไปด้วยเข็มพิมพ์ 9 เข็ม หรือ 24 เข็ม (ทำให้เรียกเครื่องพิมพ์ชนิดนี้ ได้อีกว่า เครื่องพิมพ์ 9 เข็ม และเครื่องพิมพ์ 24 เข็ม) ชุดของเข็มพิมพ์แบบ 9 เข็มจะเรียงตรงกันในแนวตั้งคอลัมน์เดียว ส่วนชุดของเข็มพิมพ์แบบ 24 เข็ม จะเรียงกันในแนวตั้งโดยแบ่งเป็น 3 คอลัมน์ ๆ ละ 8 เข็ม วางเหลื่อมกันระหว่างคอลัมน์ โดยหัวเข็มจะกระแทกผ่านผ้าหมึก ลงบนกระดาษ ทำให้เกิดอักษรขึ้นมา คุณภาพของเครื่องพิมพ์ประเภทนี้ พิจารณาจาก จำนวนหัวเข็ม โดยเครื่องพิมพ์ที่มีจำนวนหัวเข็มมากจะมีคุณภาพดีกว่าเครื่องพิมพ์ที่มีหัวเข็มน้อย ความเร็วในการพิมพ์ โดยปกติพิมพ์ได้ ตัวอักษรต่อวินาที ข้อดีของเครื่องพิมพ์ลักษณะนี้ คือ หมึกพิมพ์เป็นตลับ ราคาไม่สูงมากนัก สามารถพิมพ์กระดาษหลายก๊อปปี้ได้

2 การพิจารณาซื้อเครื่องพิมพ์แบบกระทบ จำนวนเข็มของหัวพิมพ์ เครื่องพิมพ์ที่ใช้ทั่วไป หัวพิมพ์มีเข็มเล็กๆ จำนวน 9 เข็ม แต่ถ้าต้องการให้งานพิมพ์มีรายละเอียดมาก หรือมีรูปแบบตัวหนังสือสวยขึ้น หัวพิมพ์ควรมีจำนวนเข็ม 24 เข็ม การพิมพ์ตัวหนังสือในภาวะความสวยงามนี้เรียกว่า ดังนั้นเครื่องพิมพ์ที่หัวพิมพ์มีเข็มจำนวน 24 เข็ม จะพิมพ์ได้สวยงามกว่าเครื่องพิมพ์ที่หัวพิมพ์มีเข็มจำนวน 9 เข็ม คุณภาพของหัวเข็มกับงานพิมพ์ หัวเข็มเป็นลวดที่มีกลไกขับเคลื่อน ใช้หลักการเหนี่ยวนำแม่เหล็กไฟฟ้า หัวเข็มที่มีคุณภาพดีต้องแข็ง สามารถพิมพ์สำเนากระดาษหนาได้สูงสุดถึง 5 สำเนา คุณสมบัติการพิมพ์สำเนานี้เครื่องพิมพ์แต่ละเครื่องจะพิมพ์ได้ไม่เท่ากันเพราะมีคุณภาพแรงกดไม่เท่ากัน ทำให้ความชัดเจนของกระดาษสำเนาสุดท้ายต่างกัน ความละเอียดของจุดในงานพิมพ์ ขึ้นอยู่กับขนาดของหัวเข็ม และกลไกการขับเคลื่อนของ เครื่องพิมพ์แต่ละรุ่น คุณภาพการพิมพ์ภาพกราฟิกขึ้นอยู่กับคุณลักษณะนี้

3 อุปกรณ์ตรวจสอบหัวพิมพ์ เครื่องพิมพ์แบบจุดบางรุ่นจะมีอุปกรณ์ตรวจสอบหัวพิมพ์ เช่น การตรวจสอบความร้อนของหัวพิมพ์ เพราะเมื่อใช้พิมพ์ไปนานๆ หัวพิมพ์จะเกิดความร้อนสูงมาก แม้มีอุปกรณ์ระบายความร้อนแล้ว ก็อาจไม่พอเพียง ถ้าความร้อนมาก อุปกรณ์ตรวจความร้อนจะส่งสัญญาณให้เครื่องพิมพ์ ลดความเร็วของการพิมพ์ลง ครั้งเมื่ออุณหภูมิลดลง ก็จะเพิ่มความเร็วของการพิมพ์ไปเต็มพิกัดอีก หรือ การตรวจสอบความหนาของกระดาษ เครื่องพิมพ์ส่วนใหญ่จะมีอุปกรณ์ตรวจสอบกระดาษ ถ้าป้อนกระดาษหนาไป จะทำให้หัวพิมพ์เสียหายได้ง่าย ตัวตรวจสอบความหนา จะหยุดการทำงานของเครื่องพิมพ์ เพื่อป้องกันความเสียหายของหัวพิมพ์ นอกจากนี้ยังสามารถสอบว่ากระดาษหมดหรือไม่อีกด้วย

4 ความเร็วของการพิมพ์ มีหน่วยวัดเป็นจำนวนตัวอักษรต่อวินาที การวัดความเร็วของ

เครื่องพิมพ์ ต้องมีคุณลักษณะการพิมพ์เป็นจุดอ้างอิง เช่น พิมพ์ได้ 300 ตัวอักษรต่อวินาที ในภาวะการพิมพ์แบบปกติ และที่ขนาดตัวอักษร 10 ตัวอักษรต่อนิ้วแต่หากพิมพ์แบบ NLQ โดยทั่วไปแล้วจะลดความเร็ว เหลือเพียงหนึ่งในสามเท่านั้น การทดสอบความเร็วในการพิมพ์นี้ อาจไม่ได้เท่ากับคุณลักษณะที่บอกไว้ ทั้งนี้เพราะขณะพิมพ์จริง เครื่องพิมพ์มีการเลื่อนหัวพิมพ์ขึ้นบรรทัดใหม่ ขึ้นหน้าใหม่ การเลื่อนหัวพิมพ์ไปมาจะทำให้เสียเวลาพอสมควร ความเร็วของเครื่องพิมพ์แบบจุดในปัจจุบันมีตั้งแต่ สองร้อยถึงห้าร้อย ตัวอักษรต่อวินาที

5 ขนาดแคร่พิมพ์ เครื่องพิมพ์ที่ใช้งานกันอยู่ในขณะนี้ มีขนาดแคร่ 2 ขนาด คือใช้กับกระดาษกว้าง 9 นิ้ว และ 15 นิ้ว หรือพิมพ์ได้ 80 ตัวอักษร และ 132 ตัวอักษรในภาวะ 10 ตัวอักษรต่อนิ้ว ที่พักข้อมูล คุณลักษณะในเรื่องที่พักข้อมูล (buffer) ก็เป็นเรื่องสำคัญ เพราะการพิมพ์งานนั้นเครื่องคอมพิวเตอร์ จะส่งข้อมูลลงไปเก็บในที่พักข้อมูล ถ้าที่พักข้อมูลมีขนาดใหญ่ ก็จะลดภาระการส่งงานของคอมพิวเตอร์ ไปยังเครื่องพิมพ์ได้มาก ขนาดของที่พักข้อมูลที่ใช้มีตั้งแต่ 8 กิโลไบต์ขึ้นไป อย่างไรก็ตาม เครื่องพิมพ์บางรุ่นสามารถเพิ่มเติมขนาดของที่พักข้อมูลได้ โดยการใส่หน่วยความจำลงไป ซึ่งต้องซื้อแยกต่างหาก ลักษณะการป้อนกระดาษ การป้อนกระดาษเป็นสิ่งอำนวยความสะดวกในการใช้งานเครื่องพิมพ์ คุณลักษณะที่กำหนดจะต้องชัดเจน การป้อนกระดาษมีตั้งแต่ การใช้หนามเตย ซึ่งจะใช้กับกระดาษต่อเนื่อง ที่มีรูด้านข้างทั้งสองด้าน เครื่องพิมพ์ส่วนใหญ่มีหนามเตยอยู่แล้วการใช้ลูกกลิ้งกระดาษโดยอาศัยแรงเสียดทานซึ่งเป็นคุณลักษณะของเครื่องพิมพ์ทั่วไป การป้อนกระดาษแบบอัตโนมัติ เพียงแต่ใส่กระดาษแล้วกดปุ่ม กระดาษจะป้อนเข้าไปในตำแหน่งที่พร้อมจะเริ่มพิมพ์ได้ทันที

6 การป้อนกระดาษเป็นแผ่น ส่วนใหญ่จะป้อนด้วยมือได้ แต่หากต้องการทำแบบอัตโนมัติจะต้องมีอุปกรณ์เพิ่มเพื่อทำหน้าที่ดังกล่าว อุปกรณ์นี้จะมีลักษณะเป็นถาดใส่กระดาษอยู่ภายนอกและป้อนกระดาษไปทีละใบเหมือนเครื่องถ่ายเอกสาร เครื่องพิมพ์บางเครื่องสามารถป้อนกระดาษเข้าเครื่องได้หลายทาง ทั้งจากด้านหน้า ด้านหลัก ด้านใต้ท้องเครื่อง หรือป้อนทีละแผ่น การป้อนกระดาษหลายทางทำให้สะดวกต่อการใช้งาน ภาวะเก็บเสียง เครื่องพิมพ์แบบจุดเป็นเครื่องพิมพ์ที่มีเสียงดัง ดังนั้นบางบริษัทได้พัฒนาภาวะการพิมพ์ที่เสียงเบาเป็นปกติ เพื่อลดภาวะทางเสียง จำนวนชุดแบบอักษร เครื่องพิมพ์ส่วนใหญ่จะมีจำนนชุดแบบอักษรภาษาอังกฤษ ที่ติดมากับเครื่องจำนวน 4 ถึง 9 ชุด ขึ้นกับเครื่องพิมพ์แต่ละรุ่นชุดแบบอักษรนี้สามารถเพิ่มได้โดยใช้ ตลับชุดแบบอักษรภาษาไทย ก็เป็นสิ่งสำคัญ เครื่องพิมพ์ส่วนใหญ่ที่ขายในเมืองไทยได้รับการดัดแปลงใส่ชุดแบบอักษรภาษาไทยไว้แล้ว

7 การเชื่อมต่อกับเครื่องคอมพิวเตอร์ การเชื่อมต่อกับเครื่องคอมพิวเตอร์ตามมาตรฐานสากลมีสองแบบ คือแบบอนุกรมและแบบขนาน เครื่องพิมพ์ส่วนใหญ่มักต่อกับคอมพิวเตอร์ โดยมีสายนำสัญญาณ คือมีขนาดจำนวน ยี่สิบห้าสาย การต่อกับเครื่องพิมพ์จะต้องมีสายเชื่อมโยงนี้ด้วย หากต้องการต่อแบบอนุกรม จะต้องกำหนดลงไปในเงื่อนไข เพราะเครื่องพิมพ์ส่วนใหญ่

จะมีตัวเชื่อมต่ออนุกรมเป็นเงื่อนไขพิเศษ

8 มาตรฐานคำสั่งการพิมพ์ เนื่องจากเครื่องพิมพ์ Epson ได้รับความนิยมมานาน ดังนั้น มาตรฐานคำสั่งการพิมพ์ของเครื่องพิมพ์ Epson จึงเป็นมาตรฐานที่เครื่องพิมพ์เกือบทุกยี่ห้อใช้ อย่างไรก็ตามเครื่องพิมพ์ไอบีเอ็มก็มีมาตรฐานของตนเองและเครื่องพิมพ์บางยี่ห้อก็ใช้ตาม หากจะต่อเครื่องพิมพ์เข้ากับเครื่องคอมพิวเตอร์แมคอินทอช เครื่องพิมพ์จะต้องมีคุณลักษณะในเรื่องภาวะการพิมพ์แตกต่างออกไป คือเป็นแบบโพสท์สคริปต์ การพิมพ์สี เครื่องพิมพ์บางรุ่น มีภาวะการพิมพ์แบบสีได้ การพิมพ์แบบสีจะทำให้งานพิมพ์ช้าลง และต้องใช้ริบบอนพิเศษ หรือริบบอนที่มีสี การสั่งงานที่แป้นสั่งงานบนเครื่อง ปัจจุบันเครื่องพิมพ์ส่วนใหญ่จะมีปุ่มควบคุมการสั่งงานอยู่บนเครื่องและมีจอภาพแอลซีดีขนาดเล็กเพื่อแสดงภาวะการทำงาน

9 เครื่องพิมพ์แบบพ่นหมึก เครื่องพิมพ์แบบพ่นหมึก โดยหัวพิมพ์ ซึ่งเป็นตลับหมึกของเครื่องพิมพ์ จะมีรูเล็กๆ ไว้พ่นหมึกลงบนกระดาษ ใช้หลักการพ่นหมึกลงในตำแหน่งที่ต้องการ โดยการควบคุมด้วย ไฟฟ้าสถิตย์จากคอมพิวเตอร์ ทำให้ไม่เกิดเสียงดัง ในขณะใช้งาน และยังสามารถพ่นหมึกเป็นสีต่างๆ เป็นเครื่องพิมพ์สีได้อีกด้วย เครื่องพิมพ์ประเภทนี้ มีชื่อเรียกหลายชื่อ ตามเทคโนโลยีของผู้ผลิต เป็นต้น เป็นเครื่องพิมพ์ที่ราคาไม่สูงมากนัก ปัจจุบันได้รับความนิยมอย่างสูง

10 หมึกพิมพ์ หมึกของเครื่องพิมพ์ จะเก็บไว้ในตลับ สามารถเปลี่ยนตลับใหม่ได้ ปัจจุบันมีวิธีฉีดสีเข้าไปในตลับ แทนที่จะเปลี่ยนตลับ ทำให้ประหยัดต่อผู้ใช้ โดยสีที่ใช้ประกอบด้วย แม่สีฟ้า (Cyan) แม่สีม่วง (Magenta) และแม่สีเหลือง (Yellow) โดยสีดำจะเกิดจากการผสมของแม่สีทั้งสามสี ซึ่งไม่ดำสนิท เหมือนตลับหมึกสีดำเฉพาะ (แต่ราคาก็ถูกกว่าด้วย)

11 การพิจารณาซื้อเครื่องพิมพ์แบบพ่นหมึก คุณภาพของงาน เครื่องพิมพ์แบบพ่นหมึกจะวัดคุณภาพ กันที่ความสามารถในการพิมพ์จุดต่อตารางนิ้ว (Dots Per Inch : DPI) โดยตัวเลขจะเป็นจำนวนจุดทางแนวนอน คูณจุดทางแนวตั้ง เช่น 300 X 300 Dpi เป็นต้น ซึ่งค่าจำนวนตัวเลขนี้ยิ่งมากก็ยิ่งดี เพราะจะสามารถพิมพ์ได้ละเอียดมากขึ้น ความเร็วในการพิมพ์งาน โดยปกติแล้วจะวัดเป็นจำนวนแผ่นต่อนาที โดยจะแบ่งเป็น 2 แบบคือ การพิมพ์แบบร่าง และการพิมพ์แบบมาตรฐาน ซึ่งถ้าได้จำนวนแผ่นต่อนาที มาก นั่นหมายความว่า สามารถพิมพ์งานได้รวดเร็ว

12 จำนวนหน้าที่สามารถพิมพ์ได้ ต่อการเปลี่ยนหมึกหนึ่งครั้ง เครื่องพิมพ์ แบบพ่นหมึกนี้โดยมากแล้วราคามักจะไม่แพงมาก อยู่ที่ประมาณ 3400 บาท - 6000 บาท แต่ราคาหมึกพิมพ์นั้นค่อนข้างแพงมาก เมื่อเทียบราคาต่อแผ่น กับเครื่องพิมพ์แบบอื่นๆ ดังนั้นจำนวนหน้าที่สามารถพิมพ์ได้ ต่อการเปลี่ยนหมึกหนึ่งครั้ง จึงถือว่าจำเป็นมากในการตัดสินใจเลือกใช้งาน ราคาของเครื่องพิมพ์ และราคาของหมึก เครื่องพิมพ์บางรุ่นราคาถูก แต่ หมึกพิมพ์มีราคาแพง เครื่องพิมพ์บางรุ่นมีราคาแพง แต่ราคาของหมึกพิมพ์ถูก ดังนั้นในการเลือกใช้งานต้องคำนึงถึง

ราคา ด้วย ถ้าซื้อเครื่องพิมพ์ที่มีราคาถูกมา แต่ต้องการพิมพ์งานที่มีจำนวนมาก ก็ต้องสิ้นเปลืองกับรายจ่ายที่ต้องเสียไปกับค่าหมึกเป็นจำนวนมาก

13 เครื่องพิมพ์เลเซอร์ เครื่องพิมพ์เลเซอร์ (Laser Printer) ใช้หลักการเปลี่ยนตัวอักษร และภาพให้เป็นสัญญาณภาพ ที่มีความละเอียดตั้งแต่ 200 จุดถึง 1200 จุดต่อนิ้ว จากนั้นใช้แสงเลเซอร์วาดภาพที่จะพิมพ์ลงบนกระบอกรับภาพ (เช่นเดียวกับ เครื่องถ่ายเอกสาร) โดยกระบอกรับภาพ จะมีประจุไฟฟ้า ตามรูปร่างของภาพ เมื่อกระบอกรับภาพ หมุนมาถึงตัวปล่อยผงหมึก ผงหมึกจะเกาะ เฉพาะบริเวณที่ไม่มีประจุไฟฟ้า แล้วกระบอกรับภาพ จะอัดผงหมึกลงบนกระดาษ แล้วอบด้วยความร้อน ภาพพิมพ์ก็จะติดบนกระดาษ มีทั้งเครื่องพิมพ์ขาวดำ และเครื่องพิมพ์สี ซึ่งราคาจะแพงมาก

14 ตลับหมึก ตลับหมึกของเครื่องพิมพ์แบบเลเซอร์ บรรจุในตลับที่เรียกว่า โทนเนอร์ (Toner) เวลาเปลี่ยนต้องเปลี่ยนทั้งชุดปัจจุบันเครื่องพิมพ์แบบเลเซอร์ มีการพัฒนาไปหลายรูปแบบ โดยมีรูปหนึ่งที่น่าสนใจ คือ เป็นเครื่องพิมพ์เลเซอร์ พร้อมอุปกรณ์สแกนเนอร์ และเครื่องโทรสารในเครื่องเดียว

15 Plotter Plotter เป็นอุปกรณ์แสดงข้อมูลที่มักจะใช้กับงานออกแบบ (CAD) โดยจะแปลงสัญญาณข้อมูล เป็นเส้นตรง หรือเส้นโค้ง ก่อนพิมพ์ลงบนกระดาษ ทำให้แสดงผลเป็นกราฟ แผนที่ แผนภาพต่าง ๆ ได้ โดยตัวพล็อตเตอร์ จะมีปากกามากกว่า 1 ด้าม เคลื่อนไปมา ด้วยการควบคุมของคอมพิวเตอร์ โดยปากกาแต่ละด้ามจะมีสี และขนาดเส้นที่ต่างกัน ทำให้ได้ภาพที่สวยงาม มีคุณภาพสูง และขนาดตามขนาด ของเครื่องพล็อตเตอร์

16 Diskettes หรือ Floppy Disk ดิสก์ หรือดิสก์เก็ต หรือฟลอบบี้ดิสก์ เป็นอุปกรณ์สำหรับเก็บข้อมูล ซึ่งใช้กันอย่างแพร่หลาย ซึ่งสามารถเก็บบันทึกข้อมูล หรือ ลบข้อมูล และบันทึกใหม่ได้ มีลักษณะกลมบาง ทำจากสารไมล่าร์ ที่ฉาบด้วยสารแม่เหล็ก บรรจุในซอง PVC หรือพลาสติกแข็ง เพื่อป้องกันฝุ่นละอองและการขูดขีด อาจจะเรียกว่า แผ่นดิสก์ หรือ Diskette มีอยู่ 3 ประเภท ได้แก่ แบบ 8 นิ้ว ซึ่งปัจจุบัน ไม่มีการใช้งานแล้ว แบบ 5 นิ้วซึ่งปัจจุบัน มีใช้อยู่น้อยมาก จะมีก็แต่ คอมพิวเตอร์รุ่นเก่า ประกอบด้วยสื่อบันทึกข้อมูลที่เป็นแผ่นพลาสติกบาง ทำจากสารไมล่าร์ ครอบผิวด้วยสารแม่เหล็ก ซึ่งเรียกว่า cookie กับซอง PVC หรือที่เรียกว่า Protective jacket ทำหน้าที่ป้องกันแผ่นบันทึกข้อมูล แบบ 3 นิ้วครึ่ง ปัจจุบันใช้กันมาก เป็นที่นิยม เพราะ มีขนาดเล็ก และ สะดวกในการพกพา

* *ข้อความชุด B เป็นข้อความที่พบคำศัพท์ในพจนานุกรมทุกคำ ทั้งศัพท์ทั่วไปและศัพท์เฉพาะทาง (มีทั้งหมด 16 ย่อหน้า)*

ข้อความชุด C

1 ซอฟต์แวร์ ซอฟต์แวร์ (software) หมายถึงชุดคำสั่งหรือโปรแกรม ที่ใช้สั่งงานให้คอมพิวเตอร์ทำงาน ซอฟต์แวร์จึงหมายถึง ลำดับขั้นตอนการทำงาน ที่เขียนขึ้นด้วยคำสั่งของคอมพิวเตอร์ คำสั่งเหล่านี้เรียงกันเป็น โปรแกรมคอมพิวเตอร์ จากที่ทราบมาแล้วว่า คอมพิวเตอร์ทำงานตามคำสั่ง การทำงานพื้นฐาน เป็นเพียงการกระทำกับข้อมูล ที่เป็นตัวเลขฐานสอง ซึ่งใช้แทนข้อมูลที่เป็นตัวเลข ตัวอักษร รูปภาพ หรือแม้แต่เป็นเสียงพูดก็ได้ โปรแกรมคอมพิวเตอร์ที่ใช้สั่งงานคอมพิวเตอร์ จึงเป็นซอฟต์แวร์ เพราะเป็นลำดับขั้นตอน การทำงานของคอมพิวเตอร์คอมพิวเตอร์เครื่องหนึ่ง ทำงานแตกต่างกันได้มากมาย ด้วยซอฟต์แวร์ที่แตกต่างกัน ซอฟต์แวร์จึงหมายรวมถึง โปรแกรมคอมพิวเตอร์ทุกประเภท ที่ทำให้คอมพิวเตอร์ทำงานได้

2 การที่เราเห็นคอมพิวเตอร์ทำงานให้กับเราได้มากมาย เพราะว่ามีผู้พัฒนาโปรแกรมคอมพิวเตอร์ มาให้เราสั่งงานคอมพิวเตอร์ ร้านค้าอาจใช้คอมพิวเตอร์ทำบัญชีที่ยุ่งยากซับซ้อน บริษัทขายตั๋วใช้คอมพิวเตอร์ช่วยในระบบการจองตั๋ว คอมพิวเตอร์ช่วยในเรื่องกิจการงานธนาคาร ที่มีข้อมูลต่าง ๆ มากมาย คอมพิวเตอร์ช่วยงานพิมพ์เอกสารให้สวยงาม เป็นต้น การที่คอมพิวเตอร์ดำเนินการให้ประโยชน์ได้มากมายมหาศาล จะอยู่ที่ซอฟต์แวร์ ซอฟต์แวร์จึงเป็นส่วนสำคัญของ ระบบคอมพิวเตอร์ หากขาดซอฟต์แวร์ คอมพิวเตอร์ก็ไม่สามารถทำงานได้ ซอฟต์แวร์จึงเป็นสิ่งที่จำเป็น และมีความสำคัญมาก และเป็นส่วนประกอบหนึ่งที่ทำให้ระบบสารสนเทศเป็นไปได้ตามที่ต้องการ

3 เมื่อมนุษย์ต้องการใช้คอมพิวเตอร์ช่วยในการทำงาน มนุษย์จะต้องบอกขั้นตอนวิธีการ ให้คอมพิวเตอร์ทราบ การที่บอกสิ่งที่มนุษย์เข้าใจ ให้คอมพิวเตอร์รับรู้ และทำงานได้อย่างถูกต้อง จำเป็นต้องมีสื่อกลาง ถ้าเปรียบเทียบกับชีวิตประจำวันแล้ว เรามีภาษาที่ใช้ในการติดต่อซึ่งกันและกัน เช่นเดียวกัน ถ้ามนุษย์ต้องการจะถ่ายทอด ความต้องการให้คอมพิวเตอร์รับรู้ และปฏิบัติตาม จะต้องมีสื่อกลางสำหรับการติดต่อ เพื่อให้คอมพิวเตอร์รับรู้ เราเรียกสื่อกลางนี้ว่า ภาษาคอมพิวเตอร์ เนื่องจากคอมพิวเตอร์ทำงานด้วย สัญญาณทางไฟฟ้า ใช้แทนด้วยตัวเลข 0 และ 1 ได้ ผู้ออกแบบคอมพิวเตอร์ใช้ตัวเลข 0 และ 1 นี้เป็นรหัสแทนคำสั่ง ในการสั่งงานคอมพิวเตอร์ รหัสแทนข้อมูลและคำสั่ง โดยใช้ระบบเลขฐานสองนี้ คอมพิวเตอร์สามารถเข้าใจได้ เราเรียกเลขฐานสองที่ประกอบกันเป็นชุดคำสั่ง และใช้สั่งงานคอมพิวเตอร์ว่า ภาษาเครื่อง

4 การใช้ภาษาเครื่องนี้ ถึงแม้คอมพิวเตอร์จะเข้าใจได้ทันที แต่มนุษย์ผู้ใช้จะมีข้อยุ่งยากมาก เพราะเข้าใจและจดจำได้ยาก จึงมีผู้สร้างภาษาคอมพิวเตอร์ในรูปแบบที่เป็นตัวอักษร เป็นประโยคข้อความ ภาษาในลักษณะดังกล่าวนี้เรียกว่า ภาษาคอมพิวเตอร์ระดับสูง ภาษาระดับสูงมีอยู่มากมาย บางภาษามีความเหมาะสมกับ การใช้สั่งงานการคำนวณ ทางคณิตศาสตร์และ

วิทยาศาสตร์ บางภาษามีความเหมาะสม ไว้ใช้สั่งงานทางด้านการจัดการข้อมูล ในการทำงานของคอมพิวเตอร์ คอมพิวเตอร์จะแปลภาษาระดับสูงให้เป็นภาษาเครื่อง ดังนั้นจึงมีผู้พัฒนาโปรแกรมคอมพิวเตอร์ สำหรับแปลภาษาคอมพิวเตอร์ระดับสูง ให้เป็นภาษาเครื่อง โปรแกรมที่ใช้แปลภาษาคอมพิวเตอร์ระดับสูงให้เป็นภาษาเครื่องเรียกว่า คอมไพเลอร์ (compiler) หรืออินเทอร์พรีเตอร์ (interpreter) คอมไพเลอร์จะทำการแปลโปรแกรมที่เขียนเป็นภาษาระดับสูง ทั้งโปรแกรมให้เป็นภาษาเครื่องก่อน แล้วจึงให้คอมพิวเตอร์ทำงานตามภาษาเครื่องนั้น

5 อินเทอร์พรีเตอร์จะทำการแปลทีละคำสั่ง แล้วให้คอมพิวเตอร์ทำตามคำสั่งนั้น เมื่อทำเสร็จแล้วจึงมาทำการแปลคำสั่งลำดับต่อไป ข้อแตกต่างระหว่างคอมไพเลอร์กับอินเทอร์พรีเตอร์ จึงอยู่ที่การแปลทั้งโปรแกรม หรือแปลทีละคำสั่ง ตัวแปลภาษาที่รู้จักกันดี เช่น ตัวแปลภาษาเบสิก ตัวแปลภาษาโคบอล ซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์ จึงเป็นส่วนสำคัญที่ควบคุม การทำงานของคอมพิวเตอร์ ให้ดำเนินการตามแนวความคิดที่ได้กำหนดไว้ล่วงหน้าแล้ว คอมพิวเตอร์ต้องทำงาน ตามโปรแกรมเท่านั้น ไม่สามารถทำงานที่นอกเหนือจากที่กำหนดไว้ในโปรแกรม

6 ประเภทของซอฟต์แวร์ ในบรรดาซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์ที่มีผู้พัฒนาขึ้นเพื่อใช้งานกับคอมพิวเตอร์มีมากมาย ซอฟต์แวร์เหล่านี้อาจได้รับการพัฒนาโดยผู้ใช้งานเอง หรือผู้พัฒนาระบบ หรือผู้ผลิตจำหน่าย หากแบ่งแยกชนิดของซอฟต์แวร์ตามสภาพการทำงาน พอแบ่งแยกซอฟต์แวร์ได้เป็นสองประเภท คือ ซอฟต์แวร์ระบบ (system software) และซอฟต์แวร์ประยุกต์ (application software)

7 ซอฟต์แวร์ระบบ คือซอฟต์แวร์ที่บริษัทผู้ผลิตสร้างขึ้นมาเพื่อใช้จัดการกับระบบ หน้าที่การทำงานของซอฟต์แวร์ระบบคือดำเนินงานพื้นฐานต่างๆ ของระบบคอมพิวเตอร์ เช่น รับข้อมูลจากแผงแป้นอักขระแล้วแปลความหมายให้คอมพิวเตอร์เข้าใจ นำข้อมูลไปแสดงผลบนจอภาพหรือนำออกไปยังเครื่องพิมพ์ จัดการข้อมูลในระบบแฟ้มข้อมูลบนหน่วยความจำรอง เมื่อเราเปิดเครื่องคอมพิวเตอร์ ทันทีที่มีการจ่ายกระแสไฟฟ้าให้กับคอมพิวเตอร์ คอมพิวเตอร์จะทำงานตามโปรแกรมทันที โปรแกรมแรกที่สั่งคอมพิวเตอร์ทำงานนี้เป็นซอฟต์แวร์ระบบ ซอฟต์แวร์ระบบอาจเก็บไว้ในรอม หรือในแผ่นจานแม่เหล็ก หากไม่มีซอฟต์แวร์ระบบคอมพิวเตอร์จะทำงานไม่ได้ ซอฟต์แวร์ระบบยังใช้เป็นเครื่องมือในการพัฒนาซอฟต์แวร์อื่น ๆ และยังรวมไปถึงซอฟต์แวร์ที่ใช้ในการแปลภาษาต่าง ๆ

8 ซอฟต์แวร์ประยุกต์ เป็นซอฟต์แวร์ที่ใช้กับงานด้านต่าง ๆ ตามความต้องการของผู้ใช้ ที่สามารถนำมาใช้ประโยชน์ได้โดยตรง ปัจจุบันมีผู้พัฒนาซอฟต์แวร์ใช้งานทางด้านต่าง ๆ ออกจำหน่ายมาก การประยุกต์งานคอมพิวเตอร์จึงกว้างขวางและแพร่หลาย เราอาจแบ่งซอฟต์แวร์ประยุกต์ออกเป็นสองกลุ่มคือ ซอฟต์แวร์สำเร็จ และซอฟต์แวร์ที่พัฒนาขึ้นใช้งาน

เฉพาะ ซอฟต์แวร์สำเร็จในปัจจุบันมีมากมาย เช่น ซอฟต์แวร์ประมวลคำ ซอฟต์แวร์ตารางทำงาน ฯลฯ

9 ซอฟต์แวร์ระบบ คอมพิวเตอร์ประกอบด้วย หน่วยรับเข้า หน่วยส่งออก หน่วยความจำ และหน่วยประมวลผล ในการทำงานของคอมพิวเตอร์จำเป็นต้องมีการดำเนินงานกับอุปกรณ์พื้นฐานที่จำเป็น ดังนั้นจึงต้องมีซอฟต์แวร์ระบบเพื่อใช้ในการจัดการระบบ หน้าที่หลักของซอฟต์แวร์ระบบประกอบด้วย ใช้ในการจัดการหน่วยรับเข้าและหน่วยส่งออก เช่น รับการกดแป้นต่าง ๆ บนแผงแป้นอักขระ ส่งรหัสตัวอักษรออกทางจอภาพหรือเครื่องพิมพ์ ติดต่อกับอุปกรณ์รับเข้า และส่งออกอื่น ๆ เช่น เมาส์ อุปกรณ์สังเคราะห์เสียง ใช้ในการจัดการหน่วยความจำ เพื่อนำข้อมูลจากแผ่นบันทึกมาบรรจุยังหน่วยความจำหลัก หรือในทำนองกลับกัน คือนำข้อมูลจากหน่วยความจำหลักมาเก็บไว้ในแผ่นบันทึก ใช้เป็นตัวเชื่อมต่อระหว่างผู้ใช้งานกับคอมพิวเตอร์ สามารถใช้งานได้ง่ายขึ้น เช่น การขอดูรายการสารบบในแผ่นบันทึก การทำสำเนาแฟ้มข้อมูล ซอฟต์แวร์ระบบพื้นฐานที่เห็นกันทั่วไป แบ่งออกเป็น ระบบปฏิบัติการ และตัวแปลภาษา ซอฟต์แวร์ทั้งสองประเภทนี้ทำให้เกิดพัฒนาการประยุกต์ใช้งานได้ง่ายขึ้น

10 ซอฟต์แวร์ประยุกต์ การที่เทคโนโลยีคอมพิวเตอร์ได้พัฒนาก้าวหน้าอย่างรวดเร็ว โดยเฉพาะการที่มีคอมพิวเตอร์ขนาดเล็ก ทำให้มีการใช้งานคล่องตัวขึ้น จนในปัจจุบันสามารถนำคอมพิวเตอร์ขนาดเล็ก ติดตัวไปใช้งานในที่ต่าง ๆ ได้สะดวก การใช้งานคอมพิวเตอร์ต้องมีซอฟต์แวร์ประยุกต์ ซึ่งอาจเป็นซอฟต์แวร์สำเร็จที่มีผู้พัฒนาเพื่อใช้งานทั่วไปทำให้ทำงานได้สะดวกขึ้น หรืออาจเป็นซอฟต์แวร์ใช้งานเฉพาะ ซึ่งผู้ใช้เป็นผู้พัฒนาขึ้นเองเพื่อให้เหมาะสมกับสภาพการทำงานของตน ในบรรดาซอฟต์แวร์ประยุกต์ที่มีใช้กันทั่วไป ซอฟต์แวร์สำเร็จเป็นซอฟต์แวร์ที่มีความนิยมใช้กันสูงมาก ซอฟต์แวร์สำเร็จเป็นซอฟต์แวร์ที่บริษัทพัฒนาขึ้นแล้วนำออกมาจำหน่าย เพื่อให้ผู้ใช้งานซื้อไปใช้ได้โดยตรง ไม่ต้องเสียเวลาในการพัฒนาซอฟต์แวร์อีก ซอฟต์แวร์สำเร็จที่มีจำหน่ายในท้องตลาดทั่วไป และเป็นที่นิยมของผู้ใช้มี 5 กลุ่มใหญ่ ได้แก่ ซอฟต์แวร์ประมวลคำ ซอฟต์แวร์ตารางทำงาน ซอฟต์แวร์จัดการฐานข้อมูล ซอฟต์แวร์นำเสนอ และซอฟต์แวร์สื่อสารข้อมูล

11 ซอฟต์แวร์ประมวลคำ เป็นซอฟต์แวร์ประยุกต์ใช้สำหรับการพิมพ์เอกสาร สามารถแก้ไข เพิ่ม แทรก ลบ และจัดรูปแบบเอกสารได้อย่างดี เอกสารที่พิมพ์ไว้จัดเป็นแฟ้มข้อมูล เรียกมาพิมพ์หรือแก้ไขใหม่ได้ การพิมพ์ออกทางเครื่องพิมพ์ก็มีรูปแบบตัวอักษรให้เลือกหลายรูปแบบ เอกสารจึงดูเรียบร้อยสวยงาม ปัจจุบันมีการเพิ่มขีดความสามารถของซอฟต์แวร์ประมวลคำอีกมากมาย ซอฟต์แวร์ประมวลคำที่นิยมอยู่ในปัจจุบัน ซอฟต์แวร์ตารางทำงาน เป็นซอฟต์แวร์ที่ช่วยในการคิดคำนวณ การทำงานของซอฟต์แวร์ตารางทำงาน ใช้หลักการเสมือนมีโต๊ะทำงานที่มีกระดาษขนาดใหญ่วางไว้ มีเครื่องมือคล้ายปากกา ยางลบ และเครื่องคำนวณเตรียมไว้ให้

เสร็จ บนกระดาษมีช่องให้ใส่ตัวเลข ข้อความหรือสูตร สามารถสั่งให้คำนวณตามสูตรหรือเงื่อนไขที่กำหนด ผู้ใช้ซอฟต์แวร์ตารางทำงานสามารถประยุกต์ใช้งานประมวลผลตัวเลขอื่น ๆ ได้กว้างขวาง ซอฟต์แวร์ตารางทำงานที่นิยมใช้

12 ซอฟต์แวร์จัดการฐานข้อมูล การใช้คอมพิวเตอร์อย่างหนึ่งคือการใช้เก็บข้อมูล และจัดการกับข้อมูลที่จัดเก็บในคอมพิวเตอร์ จึงจำเป็นต้องมีซอฟต์แวร์จัดการข้อมูล การรวบรวมข้อมูลหลาย ๆ เรื่องที่เกี่ยวข้องกันไว้ในคอมพิวเตอร์ เราก็เรียกว่าฐานข้อมูล ซอฟต์แวร์จัดการฐานข้อมูลจึงหมายถึงซอฟต์แวร์ที่ช่วยในการเก็บ การเรียกค้นมาใช้งาน การทำรายงาน การสรุปผลจากข้อมูล ซอฟต์แวร์จัดการฐานข้อมูลที่นิยมใช้ ซอฟต์แวร์นำเสนอ เป็นซอฟต์แวร์ที่ใช้สำหรับนำเสนอข้อมูล การแสดงผลต้องสามารถดึงดูดความสนใจ ซอฟต์แวร์เหล่านี้จึงเป็นซอฟต์แวร์ที่นอกจากสามารถแสดงข้อความในลักษณะที่จะสื่อความหมายได้ง่ายแล้วจะต้องสร้างแผนภูมิ กราฟ และรูปภาพได้ ตัวอย่างของซอฟต์แวร์นำเสนอ ซอฟต์แวร์สื่อสารข้อมูล ซอฟต์แวร์สื่อสารข้อมูลนี้หมายถึงซอฟต์แวร์ที่จะช่วยให้ไมโครคอมพิวเตอร์ติดต่อสื่อสารกับเครื่องคอมพิวเตอร์อื่นในที่ห่างไกล โดยผ่านทางสายโทรศัพท์ ซอฟต์แวร์สื่อสารใช้เชื่อมโยงต่อเข้ากับระบบเครือข่ายคอมพิวเตอร์ เช่น อินเทอร์เน็ต ทำให้สามารถใช้บริการอื่น ๆ เพิ่มเติมได้ สามารถใช้รับส่งไปรษณีย์อิเล็กทรอนิกส์ ใช้โอนย้ายแฟ้มข้อมูล ใช้แลกเปลี่ยนข้อมูล อ่านข่าวสาร นอกจากนี้ยังใช้ในการเชื่อมเข้าหามินิคอมพิวเตอร์หรือเมนเฟรม เพื่อเรียกใช้งานจากเครื่องเหล่านั้นได้ ซอฟต์แวร์สื่อสารข้อมูลที่นิยมมีมากมายหลายซอฟต์แวร์

13 ซอฟต์แวร์ใช้งานเฉพาะ การประยุกต์ใช้งานด้วยซอฟต์แวร์สำเร็จมักจะเน้นการใช้งานทั่วไป แต่อาจจะนำมาประยุกต์โดยตรงกับงานทางธุรกิจบางอย่างไม่ได้ เช่น ในกิจการธนาคาร มีการฝากถอนเงิน งานทางด้านบัญชี หรือในห้างสรรพสินค้าก็มีงานการขายสินค้า การออกใบเสร็จรับเงิน การควบคุมสินค้าคงคลัง ดังนั้นจึงต้องมีการพัฒนาซอฟต์แวร์ใช้งานเฉพาะสำหรับงานแต่ละประเภทให้ตรงกับความต้องการของผู้ใช้แต่ละราย

14 ซอฟต์แวร์ใช้งานเฉพาะมักเป็นซอฟต์แวร์ที่ผู้พัฒนาต้องเข้าไปศึกษารูปแบบการทำงานหรือความต้องการของธุรกิจนั้น ๆ แล้วจัดทำขึ้น โดยทั่วไปจะเป็นซอฟต์แวร์ที่มีหลายส่วนรวมกันเพื่อร่วมกันทำงาน ซอฟต์แวร์ใช้งานเฉพาะที่ใช้กันในทางธุรกิจ เช่น ระบบงานทางด้านบัญชี ระบบงานจัดจำหน่าย ระบบงานในโรงงานอุตสาหกรรม บริหารการเงิน และการเช่าซื้อ ความต้องการของการใช้คอมพิวเตอร์ในงานทางธุรกิจยังมีอีกมาก ดังนั้นจึงต้องมีความต้องการผู้พัฒนาซอฟต์แวร์เพื่อพัฒนาซอฟต์แวร์ใช้งานเฉพาะต่าง ๆ อีกมากมาย

* *ข้อความชุด C เป็นข้อความที่พบศัพท์ทั่วไปบางส่วน และศัพท์เฉพาะทางบางส่วน (มีทั้งหมด 14 ย่อหน้า)*

ข้อความชุด D

1 บทบาทของการมีส่วนร่วมในระบบเศรษฐกิจโลกของไทยเริ่มต้นเช่นเดียวกับหลายประเทศในเอเชีย ที่ได้พัฒนาและสร้างฐานะของประเทศโดยอาศัยความได้เปรียบจากต้นทุนแรงงานราคาถูก (Relatively Low Wage) ทำให้การผลิตมีต้นทุนต่ำ สินค้าจึงมีลักษณะเป็น Labor Intensive แต่ด้วยโครงสร้างการผลิตสินค้าของไทยที่ยังต้องพึ่งพาสินค้าทุน และวัตถุดิบนำเข้าจากต่างประเทศในสัดส่วนที่สูง กอปรกับประสิทธิภาพในการจัดสรรทรัพยากรในประเทศอยู่ในระดับต่ำ ส่งผลให้การผลิตสินค้าส่วนใหญ่ยังไม่ครบวงจร และเป็นเหตุให้มูลค่าการนำเข้ายังคงสูงกว่ามูลค่าการส่งออกอย่างต่อเนื่อง ขณะเดียวกัน การขยายตัวของการเปิดประเทศภายใต้การแข่งขันที่รุนแรงมากขึ้นในตลาดโลก ทั้งของคู่แข่งรายเดิมและรายใหม่ และการรวมกลุ่มการค้าของประเทศที่อยู่ในภูมิภาคเดียวกันเพื่อสร้างอำนาจต่อรองทางการค้าที่ได้แผ่ขยายความเชื่อมโยงกิจกรรมทางเศรษฐกิจการเงิน การค้าและการลงทุนไปทั่วโลก (Globalization) ตั้งแต่ช่วงทศวรรษ 1990 โดยที่โครงสร้างการผลิตของไทยปรับตัวอย่างเชื่องช้าและยังคงเน้นแรงงานราคาถูกเป็นปัจจัยดึงดูดเงินทุนต่างประเทศ ในที่สุดไทยก็ต้องสูญเสียความได้เปรียบในการผลิตสินค้าแบบ Labour Intensive ให้แก่ประเทศต้นทุนค่าแรงต่ำอาทิ ประเทศจีน อินเดีย กลุ่มอินโดจีน ซึ่งเด่นชัดมากขึ้นในครึ่งหลังของทศวรรษ 1990 เป็นเหตุให้การขยายตัวของการส่งออกของไทย ที่เคยเติบโตในอัตราสูงถึงร้อยละ 24.3 ต่อปี ในช่วงปี 2523-2532 ชะลอตัวลงเป็นเฉลี่ยร้อยละ 18.2 ต่อปี ในช่วงปี 2533-2538 จนกระทั่งลดลงร้อยละ 1.9 ในปี 2539 และส่งผลต่อเนื่องต่อการขยายตัวเพิ่มขึ้นของการขาดดุลบัญชีเดินสะพัด ที่เป็นปัญหาเรื้อรังต่อเศรษฐกิจไทย

2 จากประเด็นปัญหาต่างๆ ที่สะสมกันมาอย่างต่อเนื่องนับตั้งแต่การขาดดุลบัญชีเดินสะพัดเรื้อรัง การเพิ่มขึ้นของมูลค่าสินทรัพย์อย่างไร้เสถียรภาพในช่วงของภาวะเศรษฐกิจเฟื่องฟู การพึ่งพาเงินทุนจากต่างประเทศที่มีการกู้ยืมเป็นจำนวนมาก และส่งผลให้โครงสร้างหนี้ระยะสั้นของไทยอยู่ในภาวะวิกฤต ประกอบกับเงินบาทที่แข็งค่าเกินจริง และการส่งออกที่มีอัตราเติบโตชะลอลงถึงขั้นติดลบในปี 2539 เป็นเหตุให้ความเชื่อมั่นในเสถียรภาพเศรษฐกิจและการเงินของประเทศเสื่อมถอยลงอย่างต่อเนื่อง ส่งผลให้ Moody's Investors Service ปรับลดอันดับความน่าเชื่อถือทางการเงินของไทยเป็นครั้งแรกในเดือนกันยายน 2539 ในประเด็นตราสารทางการเงินระยะสั้นของไทย จากระดับ Prime - 1 มาสู่ระดับ Prime - 2 และติดตามมาด้วยการปรับลดความน่าเชื่อถือจากระดับ Prime - 2 มาสู่ระดับ Prime - 3 เมื่อวันที่ 2 ตุลาคม 2540 และปรับลดความน่าเชื่อถือครั้งล่าสุดจากระดับ Prime - 3 มาสู่ระดับ Not Prime เมื่อวันที่ 23 ตุลาคม 2540 สะท้อนให้เห็นภาวะวิกฤตของเศรษฐกิจการเงินไทยโดยแท้ อาทิ ภาระหนี้ต่างประเทศที่เพิ่มขึ้น การล้มสลายของธุรกิจการเงิน และธุรกิจภาคการผลิตบางประเภทที่เคย

เฟื่องฟูในอดีต ตลอดจนปัญหาสภาพคล่องตึงตัวของสถาบันการเงิน ท่ามกลางการชะลอตัวของเศรษฐกิจไทย ที่นำไปสู่ความปั่นป่วนของภาคเศรษฐกิจการเงินไทยในปี 2540

3 ช่วงปี 2540 ได้เกิดเหตุการณ์สำคัญหลายอย่างที่สะท้อนให้เห็นถึงความจำเป็นของทางการที่จำต้องเข้ามาแก้ไขอย่างเร่งด่วน นับตั้งแต่ปัญหาการปิดสถาบันการเงินที่กระทบต่อเนื่องไปยังสภาพคล่องของธุรกิจ ปัญหาค่าเงินบาทที่ขาดเสถียรภาพ และปัญหาภาวะตลาดทุนที่ซบเซาอย่างต่อเนื่อง เป็นต้น

4 ในเดือนกันยายน 2539 สถาบันจัดอันดับความน่าเชื่อถือ Moody's Investors Service ได้ปรับลดอันดับความน่าเชื่อถือเงินกู้ระยะสั้นของประเทศไทยจากระดับ Prime - 1 มาสู่ระดับ Prime - 2 จากการที่ประเทศไทยขาดดุลบัญชีเดินสะพัดในระดับสูงต่อเนื่องมาหลายปี และการพึ่งพาเงินทุนจากต่างประเทศมากเกินไป การปรับลดอันดับดังกล่าว ถือว่าเป็นสัญญาณเตือนภัยที่บ่งบอกถึงความจำเป็นที่จักต้องเร่งแก้ไขปัญหาดังกล่าว กอปรกับกระแสข่าวลือการลดค่าเงินบาทที่มีมาตั้งแต่ปลายปี 2539 และบริษัทเงินทุนแห่งหนึ่งประสบปัญหาทางฐานะการเงิน ทำให้นักลงทุนต่างประเทศเริ่มจับตาฐานะการเงินของประเทศไทยอย่างใกล้ชิด ปัญหาต่าง ๆ เหล่านี้ได้สะสมมานานและบานปลายในปี 2540 กลายเป็นวิกฤตการณ์ทางเศรษฐกิจการเงินที่ประวัติศาสตร์จะต้องจารึกไว้

5 ดังนั้นหากนำเหตุการณ์และความเคลื่อนไหวทางการเงินที่สำคัญในปี 2540 มาแจกแจงทำความเข้าใจในปัญหาอย่างแท้จริงแล้ว จะพบว่า ปัญหาหลักที่เกิดขึ้นในระบบสถาบันการเงินและเศรษฐกิจไทยในช่วงนั้น มีอยู่ 5 ประการ ซึ่งทางการได้ออกมาตรการแก้ไขปัญหาดังกล่าวอย่างต่อเนื่อง แต่ในท้ายที่สุด สภาวการณ์ต่าง ๆ ก็ยังไม่ดีขึ้น แต่กลับทรุดลงอย่างรวดเร็ว จนกระทั่งรัฐบาลไทยต้องขอรับความช่วยเหลือจากกองทุนการเงินระหว่างประเทศ (International Monetary Fund : IMF) เมื่อวันที่ 20 สิงหาคม 2540 อันเป็นทางออกสุดท้ายในการเรียกความเชื่อมั่นและแก้ไขปัญหาเศรษฐกิจของประเทศ อย่างไรก็ตาม มาตรการต่าง ๆ ที่นำมาใช้เพื่อแก้ไขปัญหา มีทั้งข้อดีและข้อด้อย ผลสำเร็จและล้มเหลวแตกต่างกันไป

6 สภาพวิกฤติเศรษฐกิจไทยเริ่มสะสมมาตั้งแต่ทศวรรษ 2530 จากการที่ภาคธุรกิจเอกชนด้วยการสนับสนุนของเจ้าหนี้ต่างประเทศ,รัฐ,และสถาบันการเงินในประเทศ กู้เงินทั้งจากในประเทศและต่างประเทศมาลงทุน ในธุรกิจอสังหาริมทรัพย์และธุรกิจอื่น ๆ เพิ่มขึ้นอย่างรวดเร็ว โดยไม่สามารถหาผลตอบแทนจากการลงทุนกลับมาใช้หนี้ได้ทัน ทำให้ธนาคารบางธนาคารและบริษัทเงินทุนหลายแห่งเริ่มอ่อนแอ มีปัญหาหนี้ด้อยคุณภาพเพิ่มสูง ขณะเดียวกันขีดความสามารถในการแข่งขันของไทยก็เริ่มลดลง ทำให้การส่งออกเริ่มชะลอตัว แต่การสั่งเข้ายังเพิ่มมาก ประเทศขาดดุลบัญชีเดินสะพัด (ดุลการค้าบวกด้วยดุลการบริการ) เพิ่มขึ้นอย่างต่อเนื่อง เงินทุนสำรองระหว่างประเทศลดลง ขณะที่หนี้ต่างประเทศเพิ่มขึ้นมาก โดยเฉพาะในช่วงปี 2538 - 2539 พอถึงต้นปี 2540 เจ้าหนี้ต่างประเทศเริ่มขาดความเชื่อถือความสามารถ

ในการชำระหนี้ของไทยและไม่อยากต่อสัญญาเงินกู้ ซึ่งครึ่งหนึ่งเป็นการกู้ระยะสั้น มีการเก็งกำไรโจมตีค่าเงินบาทซึ่งนักวิเคราะห์ทั่วไปเห็นว่ามีค่าสูงเกินจริง และธนาคารแห่งประเทศไทยพยายามใช้เงินทุนสำรองของประเทศไปทำสัญญาซื้อขายเงินตราต่างประเทศล่วงหน้า เพื่อปกป้องค่าเงินบาทให้คงที่ ซึ่งยิ่งทำให้เงินทุนสำรองของประเทศร่อยหรอ และฐานะทางการเงินของประเทศอ่อนแอลง

7 การเปลี่ยนนโยบายแลกเปลี่ยนเงินตราจากการผูกติดตะกร้าเงินที่มีดอลลาร์เป็นหลัก เป็นการให้เงินบาทลอยตัวในวันที่ 2 กรกฎาคม 2540 ทำให้ค่าเงินบาทลดลงถึง 40% น้ำมันขึ้นราคาและสินค้าอื่นๆ ก็ขึ้นราคาตามมา ภาคเอกชนและรัฐบาลที่เป็นหนี้ต่างประเทศอยู่ในเดือนสิงหาคม 2540 ราว 9 หมื่นล้านดอลลาร์สหรัฐ ต้องมีภาระหนี้เพิ่มขึ้นอีก 30 - 40 % รัฐบาลต้องขอกู้เงินฉุกเฉินอย่างมีเงื่อนไขจากกองทุน IMF ในเดือนสิงหาคม 2540 โดยต้องทำตามเงื่อนไข IMF ที่เน้นการเข้มงวดทางการเงินการคลัง การขึ้นภาษีมูลค่าเพิ่ม การตัดงบประมาณรายจ่ายของรัฐ การเข้มงวดกับบริษัทเงินทุนที่มีปัญหาการบริหารสภาพคล่อง และการกำหนดให้สถาบันการเงินคงอัตราดอกเบี้ยกู้สูงและปล่อยสินเชื่ออย่างระมัดระวัง

8 การทำตามเงื่อนไข IMF ผสมกับการสั่งพักการดำเนินงานบริษัทเงินทุน 58 แห่ง (ซึ่งมียอดรวมของธุรกิจที่เกี่ยวข้องไม่ต่ำกว่า 8 แสนล้านบาท) โดยไม่มีมาตรการแก้ไขที่รวดเร็วเพียงพอเป็นผลให้เกิดภาวะเศรษฐกิจซบเซาควบคู่ไปกับภาวะราคาเฟ้อ (Stagflation) อย่างรุนแรง ธุรกิจหลายประเภท เช่น อสังหาริมทรัพย์, ธุรกิจโฆษณา, สื่อสารมวลชน, บริษัทเงินทุน, อุตสาหกรรมรถยนต์ ฯลฯ ต้องทยอยปิดกิจการ, ลดขนาด, เลิกจ้างพนักงาน, ลดเงินเดือนและสวัสดิการ เพราะการขาดสภาพคล่อง ต้นทุนสูง ยอดขายตก สำนักงานคณะกรรมการพัฒนาเศรษฐกิจและสังคมแห่งชาติ คาดว่าในปี 2540 จะมีผู้ว่างงานรวมทั้งผู้ถูกเลิกจ้าง ประมาณ 1.3 ล้านคน หรือร้อยละ 4 ของแรงงานทั้งประเทศ วิกฤติที่เกิดขึ้นมีลักษณะเป็นวงจรชั่วร้าย ที่ซ้ำเติมให้เศรษฐกิจไทยทรุดหนักตามลำดับ

9 ผลเสียที่จะตามมา นอกจากภาวะเศรษฐกิจซบเซาและราคาเฟ้อแล้ว ก็คือ ทั้งสถาบันการเงิน รัฐวิสาหกิจ และธุรกิจสำคัญอื่น ๆ ของไทย กำลังถูกบีบคั้นให้บริษัทข้ามชาติเข้ามาซื้อกิจการได้ในราคาที่ต่ำลง และคนไทยจะถูกครอบนำโดยบริษัทข้ามชาติ และเป็นหนี้ต่างประเทศมากขึ้น ต้องเผชิญกับค่าครองชีพสูงขึ้น ปัญหาการว่างงานและรายได้ที่แท้จริงลดลง ทำนองเดียวกับที่ประเทศ เวเนซุเอลา เม็กซิโกซึ่งมีปัญหา และรับเงื่อนไข IMF คล้ายๆกับไทยได้ประสบมาแล้ว

10 วิกฤติครั้งนี้ส่งผลกระทบต่อประชาชนไทยในทางลบโดยทั่วหน้ากันทั้งในเรื่องการตกงาน, ถูกลดเงินเดือน สวัสดิการ, รายได้ที่แท้จริงลด เพราะ ต้นทุนในการผลิตและสินค้าอุปโภคบริโภคราคาสูงขึ้น ประชาชนกลุ่มที่ได้รับผลกระทบรุนแรงมากที่สุด คือ คนตกงาน ผู้ประกอบการที่ล้มละลาย รองลงมาคือ คนที่ถูกลดเงินเดือน สวัสดิการ ผู้ประกอบการที่ขาดทุน คนมีรายได้

ประจำที่ต้องซื้อของแพงขึ้น รวมทั้งเกษตรกรเอง ก็มีต้นทุนการผลิตที่สูงขึ้น กลุ่มที่ได้รับผลกระทบน้อยหรือได้ประโยชน์ คือ ผู้ส่งออก ซึ่งการที่ค่าเงินบาทต่ำลงทำให้สามารถส่งสินค้าไปแข่งขันในตลาดต่างประเทศได้มากขึ้น โดยเฉพาะผู้ส่งออกสินค้าประเภทใช้วัตถุดิบภายในประเทศสูง แต่สินค้าประเภทที่ต้องพึ่งการสั่งเข้าสูงก็มีต้นทุนสูงขึ้น ทำให้ไม่ได้ประโยชน์นัก รวมทั้งธุรกิจส่งออกก็ยังมีปัญหาการขาดเงินทุนหมุนเวียน รวมทั้งปัญหาต้นทุนจากปัจจัยการผลิตทั้งจากภายนอกและภายในประเทศที่จะสูงขึ้นอีกในปีหน้า

11 ปัญหาวิกฤติทางเศรษฐกิจคราวนี้รุนแรงมากกว่าครั้งใด เพราะเศรษฐกิจไทยในปัจจุบันมีขนาดการเปิดประเทศคือต้องพึ่งพาเงินทุนและการค้าระหว่างประเทศเป็นสัดส่วนสูงกว่ายุคใดๆ ในประวัติศาสตร์ ปัญหาวิกฤตินี้ไม่ใช่ความผิดพลาดทางเทคนิคของการบริหารจัดการทางเศรษฐกิจของประเทศในช่วง 1-2 ปี เท่านั้น แต่เป็นปัญหาที่มีสาเหตุสืบเนื่องมายาวนานจาก

12 โครงสร้างเศรษฐกิจการเมืองของไทยที่มีลักษณะรวมศูนย์อำนาจอยู่ในมือคนกลุ่มน้อยที่มีกรอบความคิดแบบจารีตนิยม และใช้นโยบายพัฒนาเศรษฐกิจแบบเป็นบริวารประเทศทุนนิยมพัฒนาอุตสาหกรรมทำให้การพัฒนาประเทศขาดความสมดุล การกระจายทรัพย์สินรายได้และความรู้มีช่องว่างแตกต่างกันเพิ่มมากขึ้น ประชาชนไม่ได้รับการพัฒนาให้มีความรู้ความสามารถและมีวินัยในการสร้างชาติ

13 การที่ผู้นำและผู้บริหารส่วนใหญ่ในภาครัฐและบางส่วนในภาคสถาบันการเงินเอกชน ขาดความซื่อสัตย์สุจริต ขาดความรู้ความสามารถในการนำ และขาดจินตภาพการมองการณ์ไกลในโลกยุคที่เปลี่ยนแปลงไปอย่างรวดเร็ว การดำเนินนโยบายการพัฒนาเศรษฐกิจที่ผิดพลาด หวังแต่กู้เงินต่างชาติมาลงทุน แสวงหากำไรระยะสั้น โดยไม่พัฒนาคุณภาพของประชาชนและเศรษฐกิจพื้นฐานภายในประเทศให้เข้มแข็ง ขณะที่ภาคธุรกิจเอกชน ก็มุ่งแสวงกำไรส่วนตัวระยะสั้น มากกว่าการมองการอยู่รอดระยะยาวและการเพิ่มประสิทธิภาพ การที่ระบบการศึกษาการสร้างและการถ่ายทอดความรู้ ยังเป็นแบบท่องจำจากตำราดั้งเดิมและจากต่างประเทศ ขาดการวิเคราะห์ค้นคว้าวิจัย ที่วิพากษ์วิจารณ์เป็นตัวของตัวเอง

14 จากความอ่อนด้อยทั้ง 4 ประการนี้ ชนชั้นนำของไทยจึงนำประเทศไปสู่วิกฤติ โดยไม่สนใจการเตือนภัยและไม่ได้สร้างระบบเตือนภัยเพื่อหาทางแก้ไขแต่ต้นมือ แม้เมื่อรู้ว่าวิกฤติกำลังเกิดขึ้นแล้ว ชนชั้นนำของไทยก็ยังไม่ตระหนักถึงความร้ายแรง และขาดความสามารถในการตัดสินใจแก้ไขอย่างสร้างสรรค์ รวดเร็วทันการณ์ ได้แต่รอให้ปัญหาเลวร้ายถึงที่แล้วถึงจะตามแก้ตามตำราที่ล้าสมัย ทำให้เศรษฐกิจของประเทศเสียหายรุนแรงและรวดเร็วมากกว่าที่ควรเป็น

* *ข้อความชุด D เป็นข้อความที่พบศัพท์ทั่วไปบางส่วน แต่ไม่พบศัพท์เฉพาะทางเลย (มีทั้งหมด 14 ย่อหน้า)*

ภาคผนวก จ

รายการคำศัพท์ในพจนานุกรมที่ใช้ในการทดลอง

รายการคำศัพท์ในพจนานุกรมที่ใช้ในการทดลอง

(มีทั้งหมด 673 คำ)

ก็	เก็บ	แข็ง	คู่มือ
กด	เกม	คง	เครย์
กรม	เกษตร	คน	เครือข่าย
กระดาษ	เก่า	ครอบ	เครื่อง
กระทบ	เกาะ	ครั้ง	เครื่องขับ
กระทรวง	เกิด	ครึ่ง	เครื่องบิน
กระทำ	เกี่ยว	คล่อง	เครื่องพิมพ์
กระแทก	เกี่ยวข้อง	ควบคุม	เคลื่อน
กระบอก	เกือบ	ควร	แค่
กระเป๋า	แก่	ความ	แคร่
กราฟ	ขณะ	ความเร็ว	โครงสร้าง
กราฟิก	ขนาด	ความหมาย	งาน
กลไก	ขนาน	ค่อนข้าง	ง่าย
กลม	ข้อ	คอมพิวเตอร์	เงิน
กลาง	ของ	คอลัมน์	เงื่อนไข
กล่าว	ข้อมูล	คัสซี่	จน
กลิ้ง	ขั้น	ค่า	จราจร
กลุ่ม	ขับ	คำ	จริง
กว่า	ข้าง	คำตอบ	จอง
กว้าง	ขาย	คำนวณ	จอภาพ
ก่อน	ขาว	คำนึง	จะ
ก๊อปปี้	ขีด	คำสั่ง	จัด
กัน	ขึ้น	คิด	จัดการ
กับ	ขูด	คือ	จาก
ก้าน	เข็ม	คุณภาพ	จากนั้น
การ	เข็มพิมพ์	คุณลักษณะ	จาน
กำหนด	เข้า	คุณสมบัติ	จำ
กิโล	เขียน	คุ้นเคย	จำนน
กี่	เชื่อน	คูณ	จำนวน

จำแนก
จำเป็น
จำลอง
จึง
จุ
จุด
จุดเด่น
จุดทศนิยม
เจ้าหน้าที่
ฉาบ
ฉีด
เฉพาะ
ชนิด
ช่วย
ชัดเจน
ช้า
ชาติ
ชำระ
ชิป
ชื่อ
ชุด
เช่น
เชี่ยวชาญ
เชื่อม
ใช่
ใช้
ซอง
ซอฟต์แวร์
ซับซ้อน
ซึ่ง
ซื้อ
ซุปเปอร์
ซูเปอร์
ฐาน
ฐานข้อมูล
ด้วย
ดัง
ดังนั้น
ดังนี้
ดัดแปลง
ดัน
ด้าน
ดาวเทียม
ดำ
ดำเนินการ
ดำเนินงาน
ดิน
ดิสก์
ดิสก์เก็ต
ดี๊
ดู
เด่นๆ
เดียว
เดียวกัน
เดือน
โดย
ใด
ได้
ได้แก่
ตน
ตนเอง
ตรง
ตรวจ
ตรวจสอบ
ตลอด
ตลับ
ตลับหมึก
ต่อ
ต้อง
ต้องการ
ต่อไป
ตั้ง
ตั้งแต่
ตัดสินใจ
ตัว
ตั๋ว
ตัวเลข
ตัวอย่าง
ตัวเอง
ต่าง
ต่างหาก
ต่างๆ
ตาม
ตาราง
ต่ำ
ตำแหน่ง
ติด
เต็ม
เตรียม
เติม
แต่
แตกต่าง
โต๊ะ
ใต้
ถอน
ถ้า
ถาด
ถ่าย
ถาวร
ถึง
ถือ
ถูก
แถบ
ทดสอบ
ทรัพยากร
ทรานซิสเตอร์
ท้อง
ทะเบียน
ทั้ง
ทั้งนี้
ทันที
ทั่ว
ทั่วไป
ทัศนคติ
ทาง
ท่าน
ทำ
ทำงาน
ทำให้
ที
ที่
ทุก
ทุกคน
เทคโนโลยี
เทป
เท่า
เท่ากับ
เท่านั้น

เทียบ
แท่ง
แทน
โทนเนอร์
โทรสาร
ไทย
ธนาคาร
ธุรกิจ
นอกจาก
นอกจากนั้น
นอกจากนี้
นอน
น้อย
นัก
นั่น
นั้น
นันทนาการ
นั้นๆ
นับ
น่า
นาที
นาน
นานๆ
นาย
นำ
น้ำ
นิยม
นิ้ว
นี้
เนื่องจาก
แนว
แนวคิด

ใน
บน
บรรจุ
บรรทัด
บริเวณ
บริษัท
บริหาร
บอก
บอล
บัญชี
บันทึก
บาง
บ้าง
บาท
บ้าน
บำรุง
บิต
บุคคล
บุคลากร
เบา
แบ่ง
แบบ
แบบจำลอง
ใบ
ไบต์
ปกติ
ปฏิบัติ
ปฏิบัติการ
ประกอบ
ประกัน
ประการ
ประจุ

ประชาชน
ประเทศ
ประเภท
ประมวลผล
ประมาณ
ประวัติ
ประสบ
ประสาน
ประสิทธิภาพ
ประหยัด
ปริมาณ
ปล่อย
ปลัด
ป้องกัน
ป้อน
ปัจจุบัน
ปัญหา
ปากกา
ปิด
ปี
ปุ่ม
เป็น
เป็นต้น
เปรียบ
เปลี่ยน
เปลี่ยนแปลง
เปลือง
เปิด
แป้น
แปลง
โปรแกรม
ไป

ผงหมึก
ผล
ผลลัพธ์
ผลัก
ผลิต
ผสม
ผ่าน
ผ้าหมึก
ผิว
ผู้
แผ่น
แผ่นดิสก์
แผนที่
แผนภาพ
ฝาก
ฝึก
ฝุ่นละออง
พก
พ่น
พนักงาน
พบ
พยากรณ์
พร้อม
พร้อมๆ
พล็อตเตอร์
พลาสติก
พ่วง
พอ
พอเพียง
พัก
พัฒนา
พา

พิกัด
พิจารณา
พิมพ์
พิเศษ
เพราะ
เพลง
เพิ่ม
เพียง
เพื่อ
แพง
แพทย์
แพร่หลาย
โพรเซสเซอร์
โพสท์สคริปต์
ฟลอบบี้ดิสก์
ฟัง
ฟ้า
ไฟฟ้า
ภาค
ภาพ
ภาพถ่าย
ภายนอก
ภายใน
ภาระ
ภาวะ
ภาษา
ม่วง
มอง
มัก
มัน
มา
มาก

มากๆ
มาตร
มาตรฐาน
มินิ
มี
มือ
เมนเฟรม
เมาส์
เมื่อ
เมือง
แม้
แมคอินทอช
แม่สี
แม่เหล็ก
โมเลกุล
ไม่
ไมโคร
ไมล่าร์
ยก
ย่อมเยา
ย่อยๆ
ยัง
ยาก
ยิ่ง
ยี่
ยี่ห้อ
แยก
โยง
รถไฟ
รวดเร็ว
รวม
ร่วม

รหัส
รอง
รองรับ
ร้อน
รอบข้าง
ร้อย
ระดับ
ระบบ
ระบาย
ระเบียบ
ระหว่าง
รักษา
รัฐ
รับ
ราคา
ร่าง
ราชการ
ราบรื่น
รายการ
รายงาน
รายจ่าย
รายชื่อ
รายละเอียด
ราษฎร์
ริบบอน
รีเสิร์ซ
รุ่น
รู
รู้
รูป
รูปทรง
รูปร่าง

เร็ว
เรา
เริ่ม
เรียก
เรียง
เรียนรู้
เรื่อง
แรก
แรง
แรงงาน
แรงเสียดทาน
โรงพยาบาล
ลง
ลด
ลบ
ลวด
ละ
ละเอียด
ลักษณะ
ล้าน
ล้ำ
ลูกกลิ้ง
ลูกค้า
เล็ก
เล็กๆ
เลข
เลเซอร์
เล่น
เลือก
เลื่อน
แล้ว
และ

โลก
วงจร
วัด
วัย
ว่า
วาง
วาด
วิเคราะห์
วิจัย
วิทยาศาสตร์
วิธี
วินาที
วิศวกรรม
เวลา
เวิร์ด
ไว้
ศึกษา
ศูนย์
สแกนเนอร์
ส่ง
สงสัย
สถานศึกษา
สถิตย์
สถิติ
สนใจ
สนิท
สภาพ
สภาวะ
สมควร
สมรรถนะ
สมรรถภาพ
สมอง
สมัย
สรรพ
สร้าง
ส่วน
ส่วนตัว
ส่วนประกอบ
สวย
สวยงาม
สอง
สอดคล้อง
สอน
สอบ
สะดวก
สั่ง
สังคม
สัญญาณ
สัญลักษณ์
สัมพันธ์
สากล
สาม
สามารถ
สาย
สาร
สารเคมี
สาเหตุ
สำคัญ
สำคัญๆ
สำนักงาน
สำเนา
สำรวจ
สำรอง
สำหรับ
สิ่ง
สิ้น
สินค้า
สิบ
สี
สื่อ
สื่อสาร
สุด
สุดท้าย
สู่
สูง
สูญหาย
เส้น
เส้นโค้ง
เส้นตรง
เสีย
เสียง
เสียหาย
แสง
แสดง
ใส่
หน่วย
หน่วยความจำ
หน่วยงาน
หนัง
หนังสือ
หนา
หน้า
หน้าที่
หนามเตย
หนึ่ง
หมด
หมายความ
หมึก
หมึกพิมพ์
หมุน
หยุด
หรือ
หลัก
หลักการ
หลัง
หลาก
หลากหลาย
หลาย
ห้อง
หัว
หัวเข็ม
หัวพิมพ์
หา
ห้า
หาก
ห้าง
หิ้ว
เหตุ
เหตุผล
เห็น
เหนี่ยวนำ
เหมาะ
เหมือน
เหล่า
เหลือ
เหลือง
เหลื่อม
แห่ง

ให้	ออก	อาจ	อุดมศึกษา
ใหญ่	ออกแบบ	อาชีพ	อุตสาหกรรม
ใหม่	อะไร	อ่าน	อุตุนิยมวิทยา
องค์ประกอบ	อักษร	อาย	อุปกรณ์
อดีต	อังกฤษ	อาศัย	เอกสาร
อนุกรม	อัด	อำนวย	เอง
อบ	อัตโนมัติ	อินเทอร์เน็ต	เอ็นอีซี
อย่าง	อัน	อิเล็กทรอนิกส์	เอา
อย่างไร	อากาศ	อีก	แอลซีดี
อย่างไรก็ตาม	อาคาร	อื่นๆ	ไอบีเอ็ม
อยู่	อ้างอิง	อุณหภูมิ	ฮาร์ดแวร์
			ฯลฯ