การตัดคำภาษาไทยโดยใช้คุณลักษณะ

นายไพศาล เจริญพรสวัสดิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2541
ISBN 974-332-382-1

FEATURE-BASED THAI WORD SEGMENTATION

Mr. Paisarn Charoenpornsawat

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering in Computer Engineering

Department of Computer Engineering

Graduate School

Chulalongkorn University

Academic Year 1998

ISBN 974-332-382-1

หัวข้อวิทยานิพนธ์	การตัดคำภาษาไทยโดยใช้คุณลักษณะ
โดย	นายไพศาล เจริญพรสวัสดิ์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร. บุญเสริม กิจศิริกุล
อาจารย์ที่ปรึกษาร่วม	อาจารย์ ดร. สุรพันธ์ เมฆนาวิน

บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

……คณบดีบัณฑิตวิทยาลัย
(ศาสตราจารย์ นายแพทย์ศุภวัฒน์ ชุติวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

…… ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. สมชาย ประสิทธิ์จูตระกูล)

……อาจารย์ที่ปรึกษา
(อาจารย์ ดร. บุญเสริม กิจศิริกุล)

……..อาจารย์ที่ปรึกษาร่วม
(อาจารย์ ดร. สุรพันธ์ เมฆนาวิน)

……กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

ไพศาล เจริญพรสวัสดิ์ : การตัดคำภาษาไทยโดยใช้คุณลักษณะ (Feature-based Thai Word Segmentation) อาจารย์ที่ปรึกษา : อ. ดร. บุญเสริม กิจศิริกุล, อ. ที่ปรึกษาร่วม : อ. ดร. สุรพันธ์ เมฆนาวิน ; 70 หน้า. ISBN 974-332-382-1

เนื่องจากลักษณะการเขียนของภาษาไทยนั้นไม่มีการใช้ตัวอักษรหรือสัญลักษณ์ที่นำมาใช้คั่นระหว่างคำ และงานต่างๆ ในด้านการประมวลผลภาษาธรรมชาตินั้นจำเป็นต้องทราบขอบเขตของคำก่อนถึงจะสามารถนำไปประมวลผลต่อไปได้ ดังเช่นการแปลภาษาไทย-อังกฤษ การสังเคราะห์เสียงภาษาไทย หรือการแก้ไขคำที่สะกดผิด เป็นต้น ทำให้การตัดคำนั้นถือได้ว่าเป็นปัญหาที่สำคัญปัญหาหนึ่งสำหรับงานด้านการประมวลผลภาษาธรรมชาติ

ในการตัดคำนั้นประกอบไปด้วยปัญหาหลัก 2 ปัญหาคือ 1. ปัญหาความกำกวม 2. ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม สำหรับแนวคิดในการตัดคำนั้นมีอยู่หลายแนวคิด เช่นการตัดคำแบบเลือกคำยาวที่สุด การตัดคำโดยเลือกแบบเหมือนมากที่สุด และการตัดคำโดยโมเดลไตรแกรม อย่างไรก็ตามแนวคิดต่างๆ เหล่านั้นไม่สามารถให้ความถูกต้องที่สูงในการแก้ปัญหาการตัดคำ เพราะว่ามีการใช้เพียงวิทยาการศึกษาสำนึก สำหรับการตัดคำโดยแบบเลือกคำยาวที่สุด และการตัดคำโดยเลือกแบบที่เหมือนมากที่สุด และสำหรับการตัดคำโดยใช้โมเดลไตรแกรมนั้นมีการพิจารณาแค่คำบริบทก่อนหน้าแค่เพียง 2 คำเท่านั้น ส่วนความถูกต้องในการแก้ปัญหาความกำกวมนั้นมีความถูกต้องประมาณ 53% และ 73% สำหรับการตัดคำโดยเลือกแบบเหมือนมากที่สุดและการตัดคำโดยใช้โมเดลไตรแกรมตามลำดับ

ในวิทยานิพนธ์นี้เสนอแนวคิดการนำคุณลักษณะโดยใช้การเรียนรู้ของเครื่อง 2 แบบคือริปเปอร์และวินโนว์ในการแก้ปัญหาการตัดคำภาษาไทย โดยคุณลักษณะคือข้อมูลที่อยู่รอบๆ ซึ่งสามารถนำมาประยุกต์ใช้ในการแก้ปัญหาได้ สำหรับคุณลักษณะที่นำมาใช้ในการแก้ปัญหาการตัดคำทั้ง 2 ปัญหา คือคำบริบท และสิ่งที่เกิดร่วมกันโดยมีลำดับ ในการทดลองมีการนำคลังข้อความที่มีการกำหนดหน้าที่คำจำนวน 80% เข้ามาใช้ในการเรียนรู้และส่วนที่เหลือนำมาใช้ในการทดสอบ สำหรับการวัดประสิทธิภาพนั้นได้มีการแบ่งออกเป็น 2 ส่วนคือ 1. วัดค่าความถูกต้องของการแก้ปัญหาความกำกวม 2. วัดค่าความถูกต้องของการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม สำหรับความถูกต้องโดยการใช้ริปเปอร์และวินโนว์ในการแก้ปัญหาความกำกวมนั้นให้ความถูกต้องมากกว่า 85% และ 90% ตามลำดับ ส่วนความถูกต้องในการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้นให้ความถูกต้องมากกว่า 70% และ 80% สำหรับริปเปอร์และวินโนว์ตามลำดับ

จากผลการทดลองแสดงให้เห็นว่าการตัดคำโดยใช้คุณลักษณะให้ประสิทธิภาพในการแก้ปัญหาได้ดีกว่าการตัดคำโดยใช้ไตรแกรมโมเดลและการตัดคำโดยเลือกแบบเหมือนมากที่สุด และยังแสดงให้เห็นว่าวินโนว์สามารถดึงคุณลักษณะต่างๆ จากคลังข้อความ เพื่อใช้ในการแก้ปัญหาการตัดคำได้ดีกว่าริปเปอร์

ภาควิชา ..... วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต .....

สาขาวิชา .... วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา...2541..... ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

4070366321 : MAJOR COMPUTER ENGINEERING

KEY WORD THAI / WORD / SEGMENTATION / FEATURE / CONTEXT / COLLOCATION / WINNOW / RIPPER

PAISARN CHAROENPORNSAWAT: FEATURE-BASED THAI WORD SEGMENTATION. THESIS ADVISOR: BOONSERM KIJSIRIKUL, Ph.D. THESIS COADVISOR: SURAPANT MEKNAVIN, Ph.D. 70 pp. ISBN 974-332-382-1.

In a Thai text, a delimiter for indicating the word boundary is not explicitly used. Many tasks of Natural Language Processing (NLP) such as Thai-English machine translation, Thai speech synthesis and spelling correction require boundaries of words. Therefore, word segmentation is one of the main problems in NLP.

There are two main problems in word segmentation. The first is the ambiguity problem and the second is the unknown word boundary problem. Many approaches such as longest matching, maximal matching and trigram model have been proposed. However, these approaches can not give high accuracy because longest matching and maximal matching use only heuristics and trigram model consider only two previous context words for solving the problems. The accuracy in solving ambiguity problem is about 53% and 73% for maximal matching and trigram model respectively.

This thesis proposes to use a feature-based approach with two learning algorithms namely RIPPER and Winnow in solving the problems in Thai word segmentation. A feature can be anything that tests for specific information in the context around the word in question, such as context words and collocations. In the experiment, we train the system by using RIPPER and Winnow algorithm separately, on an 80% of part-of-speech tagged corpus and the rest is used for testing. We divided the evaluation into two parts. One is the accuracy in solving the ambiguity problem and the other is the accuracy in solving the unknown word boundary problem. The accuracy using RIPPER and Winnow in solving the ambiguity problem is more than 85% and 90% respectively. On the other hand, the accuracy in solving the unknown word boundary problem is more than 70% and 80% for RIPPER and Winnow respectively.

The experiment results show the feature-based approach outperforms trigram model and maximal matching, and Winnow is superior to RIPPER for extracting the features from the corpus.

ภาควิชา ..... วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต .....

สาขาวิชา .... วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา...2541..... ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยคำแนะนำอย่างดียิ่งของ อ. ดร. บุญเสริม กิจศิริกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ และ อ. ดร. สุรพันธ์ เมฆนาวิน อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ผู้วิจัยขอขอบคุณ คุณเทพพิทักษ์ การุณบุญญานันท์ ผู้ช่วยนักวิจัย ห้องปฏิบัติการวิจัยและพัฒนาวิศวกรรมภาษาและซอฟต์แวร์ ศูนย์เทคโนโลยีอิเล็คทรอนิคส์และคอมพิวเตอร์แห่งชาติ ที่ใช้รหัสต้นฉบับ (Source code) โครงสร้างการจัดเก็บข้อมูลแบบทรัย ขอขอบคุณ คุณธนพงษ์ โพธิปิติ สำหรับการเขียนโปรแกรมในการทดลอง ขอขอบคุณ คุณวิรงรอง เทศประสิทธิ์ ที่ช่วยตรวจทานตัวสะกดในวิทยานิพนธ์ฉบับนี้ และขอขอบคุณ อ. ดร. วิรัช ศรเลิศล้ำ-วานิช อ. วันทนีย์ พันธชาติ และสมาชิกห้องปฏิบัติการวิจัยและพัฒนาวิศวกรรมภาษาและซอฟต์แวร์ทุกท่านที่คอยให้คำปรึกษา คำแนะนำ ความอนุเคราะห์ในการใช้คลังข้อความออร์คิด รายการคำศัพท์ภาษาไทย และอุปกรณ์ต่างๆ

ท้ายนี้ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดา-มารดา ซึ่งให้การสนับสนุนด้านการเงินและคอยให้กำลังใจแก่ผู้วิจัยเสมอมา

สารบัญ

หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมา

ปัจจุบันได้มีการนำคอมพิวเตอร์เข้ามาใช้ในงานด้านต่างๆ อย่างแพร่หลาย ไม่ว่าจะเป็นงานทางด้านการคำนวณ ด้านกราฟฟิก การจัดเก็บฐานข้อมูล รวมถึงการนำไปใช้งานด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing) การประมวลผลภาษาธรรมชาติคือกระบวนการที่จะทำให้คอมพิวเตอร์สามารถที่จะเข้าใจภาษามนุษย์ได้ ตัวอย่างเช่น การแปลภาษาไทย-อังกฤษ (Thai – English Machine Translation) การสังเคราะห์เสียงภาษาไทย (Thai Speech Synthesis) หรือ การสืบค้นหาข้อความทั้งเอกสาร (Full Text Search) เป็นต้น สำหรับภาษาที่ไม่มีการเว้นวรรคระหว่างคำ เช่น ภาษาจีน ภาษาญี่ปุ่น และภาษาอื่นๆ รวมทั้งภาษาไทยด้วย การหาขอบเขตของคำหรือการตัดคำจะเป็นสิ่งที่จำเป็นที่จะต้องทำเป็นอันดับแรกสำหรับงานด้านการประมวลผลภาษาธรรมชาติ และประสิทธิภาพของการตัดคำก็จะส่งผลถึงความถูกต้องของการประมวลผลภาษาธรรมชาติในระบบงานต่างๆ ด้วย ดังนั้นจะเห็นได้ว่าการตัดคำเป็นสิ่งที่สำคัญอย่างยิ่ง ในงานด้านการประมวลผลภาษาธรรมชาติ

งานหลายๆ งานในด้านการประมวลผลภาษาธรรมชาติ นอกจากที่จะต้องรู้ขอบเขตของคำแล้ว บางงานยังมีความจำเป็นต้องการทราบหน้าที่คำ (Part of Speech) หรือความหมาย (Semantic) ของคำด้วย เพื่อที่จะสามารถนำไปใช้ในการประมวลผลให้มีประสิทธิภาพมากยิ่งขึ้น ดังเช่นในการแปลภาษา การที่จะแปลให้ถูกต้องนั้น นอกจากจะต้องทราบขอบเขตของคำแล้ว การทราบหน้าที่คำจะช่วยเพิ่มความถูกต้องในการแปลด้วย เช่นคำว่า “เกาะ” อาจจะแปลเป็นภาษาอังกฤษได้เป็น “To attach” หรือ “Island” ซึ่งทั้ง 2 คำมีหน้าที่คำต่างกัน ดังนั้นถ้าทราบถึงหน้าที่ของคำ เช่นถ้าต้องการแปลคำว่า “เกาะ” ที่มีหน้าที่คำเป็นคำนาม เราก็จะแปลเป็น Island ดังนั้นการทราบหน้าที่คำจะส่งผลทำให้การแปลภาษามีความถูกต้องมากยิ่งขึ้น หรือในระบบแก้ไขคำผิด การทราบหน้าที่คำหรือความหมายของคำ ก็สามารถทำให้ระบบแก้ไขคำผิดเลือกคำที่ถูกต้องอย่างมีประสิทธิภาพมากยิ่งขึ้น หรือในโปรแกรมตรวจสอบความถูกต้องของไวยากรณ์ การทราบหน้าที่ของคำนั้นก็จะมีความจำเป็นอย่างมาก

ดังนั้นจะเห็นได้ว่าการตัดคำและการหารายละเอียดต่างๆ ของคำนั้นจะเป็นกระบวนการพื้นฐานสำหรับการประมวลผลภาษาธรรมชาติ และจะส่งผลถึงประสิทธิภาพของระบบต่างๆ ที่นำไปใช้ด้วย การ

ตัดคำนั้นได้มีการพัฒนาต่อเนื่องมานาน และได้มีการพัฒนาวิธีการต่างๆ เพื่อให้เหมาะสมกับงานแต่ละงาน โดยในระบบต่างๆ ที่นำการตัดคำไปใช้นั้นก็มีความต้องการประสิทธิภาพของการตัดคำไม่เท่ากัน เช่นในงานบางอย่างอาจจะต้องการความถูกต้องในการตัดคำอย่างมากมิฉะนั้นจะส่งผลให้ระบบไม่สามารถทำงานได้ถูกต้อง หรือบางงานอาจจะไม่ต้องการความถูกต้องมากนัก แต่ต้องการความรวดเร็วในการตัดคำมากกว่า ส่งผลทำให้มีการพัฒนาการตัดคำในแบบต่างๆ ส่วนในการหารายละเอียดต่างๆ ของคำ เพิ่งจะเริ่มมีการพัฒนาขึ้นมาไม่นานสำหรับภาษาไทย โดยรายละเอียดของคำที่เริ่มมีการพัฒนาก็คือ การหาหน้าที่ของคำ (Part-of-Speech Tagging) หรือ การหาความหมายของคำ (Semantic Tagging)

1.2 ปัญหาการตัดคำ

การตัดคำได้มีการพัฒนาอย่างต่อเนื่องมาเป็นเวลานานกว่า 10 ปี แต่ก็ยังไม่มีวิธีการใดที่สามารถจะตัดคำได้ถูกต้องทั้งหมด และงานทางด้านการประมวลผลภาษาธรรมชาติของภาษาไทยนั้นมีความจำเป็นที่จะต้องมีวิธีการตัดคำที่มีประสิทธิภาพมากที่สุด และจากงานวิจัยการตัดคำที่ผ่านมามีการนำเอากฎ พจนานุกรม ค่าสถิติ ไวยากรณ์ เข้ามาช่วยในการตัดคำ แต่ก็ยังไม่สามารถตัดคำได้ถูกต้องทั้งหมด โดยสาเหตุที่ทำให้การตัดคำโดยมีการใช้พจนานุกรมที่ผ่านมานั้น ไม่สามารถตัดคำได้ถูกต้อง เนื่องมาจากสาเหตุดังต่อไปนี้

1.2.1 ข้อความกำกวม ทำให้การตัดคำสามารถตัดคำได้หลายแบบ ทำให้เกิดความสับสนขึ้นว่าแบบไหนจะเป็นแบบที่ถูกต้องที่สุด

1.2.2 คำศัพท์ที่ไม่ปรากฏอยู่ในพจนานุกรม จะเป็นสาเหตุทำให้การตัดคำไม่สามารถทำได้ถูกต้อง ซึ่งนอกจากจะตัดคำที่ไม่มีในพจนานุกรมผิดแล้ว อาจจะส่งผลทำให้คำรอบข้างมีการตัดทำผิดด้วย

1.3 วัตถุประสงค์ของวิทยานิพนธ์

1.3.1 เพิ่มประสิทธิภาพของการตัดคำ โดยนำเอาคุณลักษณะ(Feature) เข้ามาแก้ปัญหาต่อไปนี้

1.3.1.1 แก้ไขปัญหาความกำกวม

1.3.1.2 แก้ไขปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมที่เป็นชื่อเฉพาะ (ชื่อคน, ชื่อองค์กร หรือ ชื่อสถานที่ เท่านั้น)

1.3.2 เปรียบเทียบประสิทธิภาพการเรียนรู้คุณลักษณะต่างๆ ที่จะนำมาแก้ปัญหาการตัดคำ โดยจะทำการเปรียบเทียบประสิทธิภาพของการเรียนรู้ของเครื่องระหว่าง ริปเปอร์ กับ วินโนว์

1.3.3 กำกับหน้าที่คำ

1.4 ขอบเขตของวิทยานิพนธ์

1.4.1 แก้ไขปัญหาที่ทำให้การตัดคำภาษาไทยโดยใช้พจนานุกรมไม่สามารถตัดคำได้ถูกต้อง ซึ่งวิทยานิพนธ์นี้จะทำการแก้ไขปัญหา โดยสามารถแบ่งปัญหาได้เป็น 2 กรณี คือ

1.4.1.1 ในกรณีที่ข้อความที่จะนำมาตัดคำไม่มีคำที่ไม่ปรากฏในพจนานุกรม โดยในกรณีนี้เมื่อทำการตัดคำแล้ว จะสามารถตัดได้หลายแบบ โดยที่ทุกๆ คำในแต่ละแบบจะปรากฏอยู่ในพจนานุกรมทั้งหมด ซึ่งอาจจะมีความหมาย หรือไม่มีความหมายก็ได้ และเป็นสาเหตุทำให้การตัดคำไม่ถูกต้อง ตัวอย่าง เช่นข้อความ “ตากลม” สามารถตัดได้เป็น “ตาก ลม” หรือ “ตา กลม” ซึ่งทั้ง 2 แบบจะมีความหมายทั้งคู่ หรืออีกตัวอย่างหนึ่งเช่น “ขนบนอก” สามารถตัดได้เป็น “ขน บน อก” และ “ขนบ นอก” โดยที่แบบแรกเท่านั้นที่มีความหมาย

1.4.1.2 ในกรณีที่ข้อความที่จะนำมาตัดคำมีคำที่ไม่ปรากฏในพจนานุกรม ในกรณีนี้คำที่ไม่ปรากฏในพจนานุกรมนั้นจะเป็นสาเหตุทำให้ตัดคำผิด ตัวอย่างเช่น “ไมโครซอฟต์” จะตัดคำได้เป็น “ไม โค ร ซอ ฟต์” สำหรับการตัดคำที่ใช้พจนานุกรมเพียงอย่างเดียว ซึ่งถ้าต้องการจะแก้ไขปัญหานี้ควรจะต้องมีการนำข้อมูลอื่นๆ เข้ามาประกอบด้วย โดยในงานวิทยานิพนธ์นี้จะทำการแก้ไขปัญหาประเภทนี้ โดยจะจำกัดเฉพาะคำที่เป็นชื่อคน ชื่อสถานที่ และชื่อองค์กรเท่านั้น

1.4.2 นำเอาการเรียนรู้ของเครื่องเข้ามาช่วยในการดึงคุณลักษณะต่างๆ จากคลังข้อความ เพื่อที่จะนำเอาคุณลักษณะต่างๆ ที่ได้จากการเรียนรู้ของเครื่องเข้ามาใช้ในการตัดคำ และทำการเปรียบเทียบการเรียนรู้ของเครื่องในแบบต่างๆ โดยการเรียนรู้ของเครื่องที่จะนำมาใช้นั้น มีอยู่ 2 วิธีคือ

1.4.2.1 ริปเปอร์ (RIPPER)

1.4.2.2 วินโนว์ (Winnow)

1.5 ขั้นตอนการวิจัย

1.5.1 ศึกษาการตัดคำวิธีการต่างๆ ที่ผ่านมา

1.5.2 ศึกษาการเรียนรู้ของเครื่องเพื่อที่จะนำมาประยุกต์ใช้ในการตัดคำ

1.5.2.1 ศึกษาการเรียนรู้ของเครื่องที่มีชื่อว่า ริปเปอร์

1.5.2.2 ศึกษาการเรียนรู้ของเครื่องที่มีชื่อว่า วินโนว์

1.5.3 พัฒนาโปรแกรมตัดคำและกำหนดหน้าที่ของคำโดยใช้โมเดลไตรแกรม

1.5.4 ออกแบบและพัฒนาระบบการตัดคำแบบใหม่ที่นำเอาการเรียนรู้ของเครื่องเข้ามาใช้

1.5.5 ทำการทดลองเพื่อวัดประสิทธิภาพและเปรียบเทียบผลของการตัดคำในแบบต่างๆ

1.5.6 สรุปผลการวิจัย และ จัดทำรายงานวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1.6.1 รวบรวมงานวิจัยทางด้านการตัดคำและงานที่เกี่ยวข้องที่ผ่านมา

1.6.2 เปรียบเทียบประสิทธิภาพการเรียนรู้ของเครื่องระหว่าง ริปเปอร์ กับ วินโนว์ ในการนำคุณลักษณะต่างๆ เข้ามาแก้ปัญหาการตัดคำ

1.6.3 นำกระบวนการเรียนรู้ของเครื่องมาประยุกต์ใช้ในการแก้ปัญหาการตัดคำ

1.6.4 การตัดคำแบบใหม่ที่มีประสิทธิภาพมากยิ่งขึ้น

1.6.5 สรุปปัญหาและอุปสรรคในการตัดคำ

1.6.6 แนวทางการเพิ่มประสิทธิภาพของการตัดคำ

1.7 สิ่งตีพิมพ์ที่ได้จากงานวิทยานิพนธ์

จากงานวิทยานิพนธ์นี้ได้มีบทความที่ได้รับการตีพิมพ์ทั้งหมดจำนวน 3 บทความคือ

1.7.1 บทความเรื่อง “Feature-based Thai Word Segmentation” ในงานประชุมวิชาการ Natural Language Processing Pacific Rim Symposium 1997 (NLPRS'97) (Incorporating SNLP'97)” โดย Surapant Meknavin, Paisarn Charoenpornsawat and Boonserm Kijsirikul . สถานที่จัด จ. ภูเก็ต วันที่ 2-4 ธันวาคม พ.ศ. 2540

1.7.2. บทความเรื่อง “Feature-Based Proper Name Identification in Thai” ในงานประชุมวิชาการ ”The National Computer Science and Engineering Conference'98 (NCSEC’98)” โดย Paisarn Charoenpornsawat, Boonserm Kijsirikul and Surapant Meknavin. สถานที่จัด มหาวิทยาลัยเกษตรศาสตร์ วันที่ 19-21 ตุลาคม พ.ศ. 2541

1.7.3 บทความเรื่อง “Feature-based Thai Unknown Word Boundary Identification Using Winnow” ในงานประชุมวิชาการ “1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS’98)” โดย Paisarn Charoenpornsawat, Boonserm Kijsirikul and Surapant Meknavin. สถานที่จัด จ. เชียงใหม่ วันที่ 24-27 พฤศจิกายน พ.ศ. 2541

บทที่ 2

งานวิจัยและทฤษฎีที่เกี่ยวข้อง

เนื่องจากการตัดคำได้มีการพัฒนาติดต่อกันมาเป็นเวลายาวนาน ทำให้มีงานวิจัยด้านการตัดคำเกิดขึ้นมากมายหลายวิธี ซึ่งในช่วงแรกนั้นได้มีการพัฒนาการตัดพยางค์ขึ้นมาก่อน หลังจากนั้นค่อยมีการพัฒนาการตัดคำตามมา ซึ่งในบทนี้จะกล่าวถึงวิธีงานวิจัยด้านการตัดพยางค์ การตัดคำและงานที่เกี่ยวข้องที่ผ่านมาในตั้งแต่อดีตจนถึงปัจจุบัน ในวิทยานิพนธ์นี้จะแบ่งวิวัฒนาการของการตัดคำหรือตัดพยางค์ที่ผ่านมาโดยแบ่งตามลักษณะฐานข้อมูลที่จะนำมาใช้ในการตัดคำ ซึ่งสามารถแบ่งได้เป็น 3 ยุคคือ 1. ยุคการใช้กฎ 2. ยุคการใช้พจนานุกรม 3. ยุคการใช้คลังข้อความ

2.1 ยุคการใช้กฎ

ในยุคนี้คอมพิวเตอร์ยังไม่มีความสามารถในประมวลผลสูงมากนัก ประกอบกับหน่วยความจำในเครื่องคอมพิวเตอร์มีขนาดเล็ก ทำให้ในยุคนี้มีการพัฒนาการตัดพยางค์ขึ้นมาก่อน เนื่องจากพยางค์นั้นมีกฎเกณฑ์ที่แน่นอนมากกว่าคำ ทำให้ในยุคนี้มีการนำกฎเข้ามาใช้ในการตัดพยางค์ ซึ่งจากการนำกฎเข้ามาใช้ในการตัดพยางค์แล้วผลปรากฏว่าจะสามารถแบ่งพยางค์ได้ถูกต้องจำนวนมาก โดยวิธีการแบ่งพยางค์นั้นมีการพัฒนาขึ้นมากมาย โดยงานการแบ่งพยางค์ที่ผ่านมามีดังต่อไปนี้

2.1.1 งานของ ยุพิน ไทยรัตนานนท์

งานของยุพิน ไทยรัตนานนท์ (Yupin Thairatananond, 1981) เป็นงานวิจัยการตัดพยางค์ โดยการใช้กฎในการตัดพยางค์ ซึ่งกฎต่างๆ ที่สร้างขึ้นมานั้นโดยอาศัยหลักไวยากรณ์ภาษาไทย แต่ก็จะมีปัญหาในการสร้างกฎเพราะมีบางพยางค์ไม่เป็นไปตามกฎที่ตั้งไว้ ทำให้มีการจัดเก็บพยางค์ต่างๆ ที่เป็นข้อยกเว้นไว้ในแฟ้มข้อมูล ซึ่งงานวิจัยนี้ได้พัฒนาโดยใช้ภาษาพีแอลวัน (PL/I)

ลักษณะของกฎที่นำมาใช้ในการตัดพยางค์ภายในงานวิจัยนี้ ได้สร้างมาจากลักษณะไวยากรณ์ทางภาษาไทย โดยมีการพิจารณาจากลักษณะของอักขระที่ปรากฏในพยางค์หรือคำ ซึ่งทำให้มีการจัดหมวดหมู่ตัวอักขระภาษาไทย โดยการแบ่งหมวดตามการนำไปใช้ ซึ่งสามารถแบ่งได้เป็น 5 กลุ่มใหญ่ๆ ดังต่อไปนี้ คือ

1. กลุ่มพยัญชนะ (Consonant)
    - พยัญชนะที่อยู่หน้าพยางค์เสมอ
    - พยัญชนะที่ส่วนใหญ่จะอยู่หน้าพยางค์
    - พยัญชนะที่เป็นตัวสะกด
    - พยัญชนะที่เป็นสระ
    - อื่นๆ
2. กลุ่มสระ (Vowel)
    - สระที่ไม่ต้องมีตัวสะกด
    - สระที่จะอยู่หน้าพยางค์เสมอ
    - สระที่ส่วนใหญ่จะมีตัวสะกดร่วมด้วย
    - สระที่มีหรือไม่มีตัวสะกดร่วมด้วย
3. กลุ่มวรรณยุกต์ (Tone mark)
4. กลุ่มตัวเลข (Numeral)
5. กลุ่มอักขระพิเศษ (Special character)

ขั้นตอนการทำงานของวิธีการนี้จะตัดพยางค์จากขวามาซ้าย โดยใช้กฎต่างๆ ที่สร้างขึ้นมาจากลักษณะของตัวอักขระดังที่ได้กล่าวไปแล้ว และกฎต่างๆ ที่สร้างขึ้นมานั้นจะจัดเก็บไว้ภายในรหัสต้นฉบับ (Source code) ซึ่งทำให้การเพิ่มหรือแก้ไขกฎไม่สามารถทำได้สะดวก และจากการทดสอบปรากฏว่าผลลัพธ์ที่ได้จากการตัดพยางค์ด้วยวิธีการนี้ จะได้ผลความถูกต้องไม่น้อยกว่า 85%

2.1.2 งานของ สุรินทร์ จรรยาพรพงษ์

สุรินทร์ จรรยาพรพงษ์ (Surin Charnyapornpong, 1983) ได้ทำการวิจัยเกี่ยวกับการตัดคำภาษาไทยโดยใช้พยางค์ โดยกฎที่นำมาใช้นั้นได้นำมาจากหลักไวยากรณ์ภาษาไทย และได้ทำการวิเคราะห์ลักษณะต่างๆ ของพยางค์ภาษาไทย โดยลักษณะของกฎที่ได้นี้สามารถแบ่งได้เป็น 2 ชนิดคือ กฎการหาขอบเขตหน้า (Front boundary recognition rule) และ กฎการหาขอบเขตหลัง (Tail boundary recognition rule) และในแต่ละกฎยังแบ่งออกเป็น 2 กลุ่มย่อยๆ คือแบ่งตามคุณสมบัติของตัวอักษรโดยกฎที่ได้ออกมานี้จะจัดให้อยู่ในกลุ่มเอ (Group A) และแบ่งตามคุณสมบัติของรูปแบบการใช้สระแต่ละตัวซึ่งกฎที่ได้ออกมานี้จะแบ่งให้อยู่ในกลุ่มบี (Group B)

เนื่องจากลักษณะของตัวอักษรภาษาไทยนั้นสามารถจะเป็นจุดที่บอกขอบเขตของพยางค์ได้อย่างดี ทำให้ในงานวิจัยนี้มีการนำลักษณะของตัวอักษรมาสร้างกฎการตัดพยางค์ซึ่งเรียกกฎเหล่านี้ว่า กฎที่ได้จากคุณสมบัติของอักษรหรือกฎกลุ่มเอ

ตัวอย่างกฎที่ได้จากคุณสมบัติของอักษรในการหาขอบเขตหน้าของพยางค์ เช่น

กฎ A-1F : สระต่างๆ เหล่านี้ ะ า ิ ี ึ ื ุ ู ็ ำ ั และ วรรณยุกต์ ่ ้ ๊ ๋ จะต้องมีพยัญชนะอยู่ข้างหน้าอย่างน้อย 1 ตัวอักษรเสมอ

กฎ A-2F : สระ เ แ ไ โ ส่วนใหญ่จะเป็นตัวอักษรแรกในพยางค์เสมอ ยกเว้นคำบางคำ ตัวอย่างเช่น ขโมย , ชโลม , ทแยง , จเร , สไบ เป็นต้น

กฎ A-3F : สระ ใ จะเป็นตัวอักษรแรกของพยางค์เสมอ โดยไม่มีข้อยกเว้น

ตัวอย่างกฎที่ได้จากคุณสมบัติของอักษรในการหาขอบเขตหลังของพยางค์ เช่น

กฎ A-1T : พยัญชนะต่อไปนี้ ศ ณ ญ ษ ฐ ฎ ฏ ฒ ฬ ฌ จะเป็นตัวสะกดเสมอ ยกเว้นพยางค์ดังต่อไปนี้ ศก ศร ศง ศพ ษก ฐก ฬส ญวน ณรงค์ ศตวรรษ และยังมีพยางค์อื่นๆ อีก แต่พยางค์เหล่านั้นสามารถจัดการได้โดยใช้ กฎ A-1F

กฎ A-2T : สระ ึ จะตัวมีตัวสะกดหนึ่งตัวเสมอ ยกเว้น พยางค์เหล่านี้ “ ึ ึ ึ”

กฎ A-3T : ไม้หันอากาศ (ั) จะต้องตามด้วยตัวสะกดอย่างน้อย 1 ตัวอักษรเสมอ

จากตัวอย่างที่แสดงข้างต้นจะเห็นว่าการหาขอบเขตของพยางค์สามารถจะนำคุณสมบัติของตัวอักษรต่างๆ มาสร้างเป็นกฎเพื่อใช้ในการแบ่งพยางค์ได้ นอกจากนี้ในงานวิจัยนี้ยังมีการสร้างกฎที่ใช้ในการแบ่งพยางค์ โดยกฎที่ได้จะพิจารณาจากคุณสมบัติของรูปแบบการใช้สระแต่ละตัวหรือกฎกลุ่มบี ซึ่งจะพิจารณาจากลักษณะของพยางค์ในภาษาไทย โดยจากที่สังเกตเห็นคือลักษณะของพยางค์ในภาษาไทยนั้นจะมีรูปแบบที่แน่นอนและตายตัว ทำให้ในงานวิจัยนี้มีการสร้างกฎต่างๆ โดยอ้างอิงจากรูปแบบของพยางค์ต่างๆ ที่ปรากฏในภาษาไทย

ตัวอย่างกฎที่ได้จากคุณสมบัติของรูปแบบการใช้สระ ในการหาขอบเขตหลังของพยางค์ เช่น

กฎ B-1T : สระเหล่านี้ “ ั ิ ึ ื ” ถ้ามีวรรณยุกต์ แล้วจะต้องมีตัวสะกด 1 ตัวเสมอ

กฎ B-4T : สระ ในรูปแบบดังต่อไปนี้ -ัว -ัย -ัวะ -ือ เ-า เ-อ เ-าะ เ-ะ แ-ะ โ-ะ เ-ียะ เ-อะ ไม่ต้องการตัวสะกด ยกเว้นบางพยางค์ในรูปแบบดังนี้ เ–า – และ เ – อ – โดยเครื่องหมาย – แทนตัวพยัญชนะ 1 ตัวอักษร

นอกจากการสร้างกฎจากคุณสมบัติของตัวอักษรและคุณสมบัติของรูปแบบการใช้สระแล้ว ในงานวิจัยนี้มีการสร้างกฎสำหรับจัดการกับพยางค์ที่ไม่ใช่ลักษณะของพยางค์ภาษาไทย ซึ่งพยางค์เหล่านั้นอาจจะประกอบไปด้วย อักขระพิเศษ ตัวเลข คำย่อ หรือ อักขระจากภาษาต่างชาติ

จากการทดลองวัดความถูกต้องในการตัดพยางค์ งานวิจัยนี้ได้ทดลองกับเอกสารต่างๆ จำนวน 100 เล่ม โดยเอกสารนั้นได้นำมาจากเอกสารชนิดต่างๆ จำนวน 10 ชนิด ดังเช่น หนังสือพิมพ์

ปรัชญา วิทยา-ศาสตร์ ศาสนา ภาษาศาสตร์ ฯลฯ และจากการทดสอบปรากฏว่าสามารถตัดพยางค์ได้ถูกต้องถึง 96%

2.2 ยุคการใช้พจนานุกรม

ในยุคนี้ถือได้ว่าเป็นยุคเริ่มแรกในการตัดคำ เนื่องจากในยุคนี้เครื่องคอมพิวเตอร์ได้มีการพัฒนาขึ้นและมีหน่วยความจำมากขึ้น จากยุคที่แล้วมีการนำเอากฎเข้ามาช่วยในการแบ่งพยางค์ แต่สำหรับการแบ่งคำแล้วการใช้กฎอย่างเดียวไม่สามารถที่จะหาขอบเขตของคำได้ ทำให้ในยุคนี้ได้มีการคิดค้นหาวิธีการแบ่งคำโดยมีการนำเอาพจนานุกรมเข้ามาใช้ร่วมกับกฎในการตัดคำด้วย โดยแนวคิดการตัดคำโดยใช้พจนานุกรมแบบต่างๆ มีดังนี้

2.2.1 ยืน ภู่วรวรรณ และวิวรรธน์ อิ่มอารมณ์

ในงานวิจัยนี้จะเป็นงานวิจัยการแบ่งพยางค์ด้วยพจนานุกรม (ยืน ภู่วรวรรณและ วิวรรธน์ อิ่มอารมณ์, 2529) ซึ่งถือได้ว่าเป็นงานวิจัยงานแรกของการตัดพยางค์ที่มีการนำพจนานุกรมเข้ามาใช้ โดยจะจัดเก็บพยางค์ต่างๆ ไว้ในพจนานุกรม และมีการนำกฎไวยากรณ์ต่างๆ จำนวน 18 กฎเข้ามาช่วยในกรณีที่ไม่พบพยางค์ในพจนานุกรม

หลักการทำงานของกระบวนวิธีการตัดพยางค์ด้วยพจนานุกรมนี้ก็คือ จะทำการตรวจสอบสายอักขระ (String) ที่เข้ามาจากซ้ายไปขวากับพยางค์ที่ได้เก็บไว้ในพจนานุกรม ในกรณีที่ทำการตรวจสอบแล้วปรากฏว่าพบพยางค์มากกว่า 1 พยางค์ในพจนานุกรม ก็ให้ทำการเลือกแบ่งพยางค์โดยเลือกพยางค์ที่ยาวที่สุด แล้วก็ทำต่อไปเรื่อยๆ จนจบสายอักขระ แต่ถ้าในกรณีที่เลือกพยางค์ที่ยาวที่สุดไปแล้ว ทำให้เกิดพยางค์ที่ไม่ปรากฏในพจนานุกรมก็ยอมให้มีการย้อนรอย (Back Tracking) กับไปเลือกพยางค์ที่ยาวรองลงมาแทน ซึ่งวิธีการนี้จะเป็นที่รู้จักกันในชื่อ การตัดคำ(พยางค์)แบบเลือกคำ(พยางค์)ยาวที่สุด (Longest Matching)

จากงานวิจัยนี้ได้มีการเปรียบเทียบความรวดเร็วในการแบ่งพยางค์ ซึ่งสรุปผลได้ว่าเมื่อนำพจนานุกรมเข้ามาใช้ในการแบ่งพยางค์จะสามารถตัดพยางค์ได้รวดเร็วกว่าการใช้กฎ โดยที่ความถูกต้องของการตัดพยางค์นั้นสามารถตัดได้ถูกต้องมากกว่า 99 % แต่สำหรับวิธีการนี้ก็ยังมีข้อเสียคือ ต้องเสียเนื้อที่ในการจัดเก็บพจนานุกรมในหน่วยความจำหลักเป็นจำนวน 50 กิโลไบต์แต่ก็สามารถเก็บข้อมูลพจนานุกรมไว้ในเครื่องคอมพิวเตอร์ในสมัยนั้นได้

2.2.2 งานของ ดวงแก้ว สวามิภักดิ์

งานวิจัยชิ้นนี้คือ การสร้างซอฟต์แวร์วิเคราะห์ไวยากรณ์ไทยภายใต้ระบบยูนิกซ์ (ดวงแก้ว สวามิภักดิ์, 2533) เป็นงานวิจัยด้านการตัดคำภาษาไทย โดยใช้กฎทางไวยากรณ์ที่สร้างขึ้นมาเอง และมีการนำพจนานุกรมเข้ามาใช้ประกอบร่วมด้วย โดยสาเหตุที่นำทั้งกฎไวยากรณ์และพจนานุกรมเข้ามาช่วยในการตัดคำนั้นก็เพื่อที่จะแก้ไขปัญหาการตัดโดยใช้พจนานุกรมเพียงอย่างเดียว (ยืน ภู่วรวรรณและวิวรรธน์ อิ่ม-อารมณ์, 2529) ซึ่งไม่สามารถตัดคำได้ถูกต้องในกรณีที่คำนั้นไม่มีอยู่ในพจนานุกรม

งานวิจัยการตัดคำนี้ ได้ทำภายใต้ระบบปฏิบัติการยูนิกซ์ และได้มีการนำโปรแกรมเล็กซ์ (Lex) เข้ามาช่วยจัดการการตัดคำ โดยมีการสร้างกฎต่างๆ ให้อยู่ในรูปแบบนิพจน์ที่มีกฎเกณฑ์ (Regular Expression) โดยกฎที่สร้างขึ้นมานี้ประกอบไปด้วย 43 กฎ (รายละเอียดของกฎต่างๆ สามารถดูได้ในภาคผนวก ก) ซึ่งกฎที่ได้มานี้จะไม่มีการรวมตัวสะกดเข้าไปในกฎด้วยยกเว้นบางกรณี เนื่องจากลักษณะของโปรแกรมเล็กซ์ จะพยายามสร้างกลุ่มตัวอักษร (Token) ที่มีขนาดที่ยาวที่สุดก่อน ดังนั้นถ้ามีการนำกฎที่มีตัวสะกดเข้ามาใช้ จะเป็นสาเหตุให้มีการรวมเอาอักษรตัวหน้าของคำถัดไปมาเป็นตัวสะกดได้ ซึ่งเมื่อได้ผ่านการวิเคราะห์ด้วยกฎแล้ว ขั้นตอนต่อไปก็จะมีการรวมกลุ่มตัวอักษรเข้าด้วยกัน โดยทำการตรวจสอบจากพจนานุกรม ส่วนโครงสร้างของพจนานุกรมที่นำมาใช้ในที่นี้คือฐานข้อมูลแบบรีเลชั่น (Relational DBMS) ซึ่งใช้คำเป็นดรรชนี (Index) และไฟล์ดรรชนีได้พัฒนาขึ้นโดยใช้โครงสร้างข้อมูลแบบบีทรี (B-Tree)

งานวิจัยนี้ได้แบ่งการวัดประสิทธิภาพของการตัดคำเป็น 2 ชนิดคือ 1. ความถูกต้องในเชิงของคำ และ 2. ความถูกต้องในเชิงของพยางค์ และได้ทดลองกับเอกสารจำนวน 17 ชนิด ซึ่งผลปรากฏว่าได้ความถูกต้องถึง 98.11% ในเชิงคำ และ 99.67% เชิงของพยางค์

2.2.3 สัมพันธ์ ระรื่นรมย์

ในงานวิจัยนี้เป็นงานวิจัยการแบ่งคำไทยด้วยพจนานุกรม (สัมพันธ์ ระรื่นรมย์, 2534) โดยเป้าหมายของงานวิจัยนี้จะเน้นที่การเพิ่มประสิทธิภาพในด้านความรวดเร็วของขั้นตอนวิธีในการตัดคำและการลดขนาดของพจนานุกรม เนื่องจากเมื่อนำพจนานุกรมเข้ามาใช้ในการตัดคำแล้วจะทำให้ความถูกต้องในการตัดคำเพิ่มขึ้นมากกว่าการตัดคำใช้กฎอย่างเดียว ดั้งนั้นในงานวิจัยนี้จึงไม่ได้เน้นการเพิ่มประสิทธิภาพในด้านความถูกต้องมากนักเพราะถือว่าการตัดคำโดยใช้พจนานุกรมจะให้ค่าความถูกต้องที่สูงอยู่แล้ว

โดยรายละเอียดของขั้นตอนวิธีการตัดคำนั้นจะมีวิธีการคล้ายกับงานวิจัยการแบ่งพยางค์โดยใช้ดิกชันนารี (ยืน ภู่วรวรรณและวิวรรธน์ อิ่มอารมณ์, 2529) ซึ่งในงานวิจัยนี้จะทำการจัดเก็บคำลงในพจนานุกรมแทนพยางค์ ส่วนขั้นตอนวิธีจะทำงานเหมือนเดิม คือใช้ขั้นตอนวิธีแบบเลือกคำที่ยาวที่สุดดังที่ได้กล่าวไปแล้ว ตัวอย่างการตัดคำโดยเลือกคำที่ยาวที่สุดแสดงดังตารางที่ 2-1

ตารางที่ 2-1 ตารางแสดงการตัดคำแบบเลือกคำที่ยาวที่สุด

ประโยค	คำที่ได้	คำที่ถูกเลือก
โคนมนอนบนกองหญ้า	โค, โคน	โคน
มนอนบนกองหญ้า	-	(ย้อนรอย)
โคนมนอนบนกองหญ้า	โค, โคน	โค (เลือกคำรองลงมา)
นมนอนบนกองหญ้า	นม	นม
นอนบนกองหญ้า	นอ, นอน	นอน
บนกองหญ้า	บน	บน
กองหญ้า	กอ, กอง	กอง
หญ้า	หญ้า	หญ้า

จากตารางที่ 2-1 จะแสดงการตัดคำแบบเลือกคำที่ยาวที่สุด โดยประโยคที่นำมาตัดคำคือ “โคนมนอนบนหญ้า” สามารถตัดคำได้เป็น “โค นม นอน บน หญ้า”

ส่วนโครงสร้างของพจนานุกรมที่ได้นำมาใช้ในงานวิจัยนี้คือ โครงสร้างข้อมูลแบบทรัย (Trie) ซึ่งจากการนำโครงสร้างทรัยเข้ามาใช้สามารถช่วยลดขนาดของพจนานุกรมได้ และนอกจากนี้โครงสร้างแบบทรัยนี้ยังสามารถสืบค้นหาคำศัพท์ได้อย่างรวดเร็วและสามารถจะเพิ่มเติมคำศัพท์ได้อย่างสะดวกและรวดเร็วด้วย โดยในรายละเอียดต่างๆ เกี่ยวกับโครงสร้างแบบทรัยจะอธิบายเพิ่มเติมในบทที่ 4 เรื่องโครงสร้างของพจนานุกรม

สรุปจากงานนี้ได้มีการนำโครงสร้างทรัยมาประยุกต์ใช้เพื่อลดขนาดของพจนานุกรม ซึ่งจากการเปรียบเทียบประสิทธิภาพในด้านความรวดเร็วและขนาดของพจนานุกรม ปรากฏว่าผลการเปรียบเทียบขนาดของพจนานุกรม จำนวน 5400 คำสามารถใช้เนื้อที่ 27975 ไบต์ ซึ่งมีขนาดน้อยกว่างานวิจัยการแบ่งพยางค์ด้วยพจนานุกรม (ยืน ภู่วรวรรณและวิวรรธน์ อึ่มอารมณ์, 2529) ซึ่งใช้เนื้อที่ประมาณ 32,482 ไบต์ ส่วนความซับซ้อนของขั้นตอนวิธีในการสืบค้นก็ลดลงด้วยเนื่องมาจากลักษณะทางโครงสร้างของทรัย

2.2.4 วิรัช ศรเลิศล้ำวานิช

สำหรับในงานวิจัยการตัดคำภาษาไทยชิ้นนี้ ได้มีการพัฒนาการตัดคำโดยเรียกว่า “การตัดคำโดยเลือกแบบเหมือนมากที่สุด (Maximal Matching)” (วิรัช ศรเลิศล้ำวานิช, 2536) ซึ่งขั้นตอนวิธีนี้ จะสามารถแก้ไขความบกพร่องของการตัดคำแบบเลือกคำยาวที่สุดได้ โดยจุดบกพร่องที่กล่าวนี้คือขั้นตอนวิธีการตัดคำแบบเลือกคำยาวที่สุดจะเลือกคำที่ยาวเกินไปตั้งแต่ครั้งแรก ทำให้ข้อความที่ตามมาเกิดข้อผิด

พลาดได้ ตัวอย่างเช่น ประโยค “ไปหามเหสี” จะตัดคำได้เป็น “ไป หาม เห สี” โดยที่ถูกต้องควรจะตัดเป็น “ไป หา มเหสี”

หลักการของการตัดคำโดยเลือกแบบเหมือนมากที่สุดคือ ขั้นตอนแรกคือจะทำการตัดคำที่เป็นไปได้ทุกๆ แบบก่อน แล้วหลังจากนั้นก่อนให้ประโยคที่มีจำนวนคำน้อยที่สุด ตัวอย่างเช่น “ไปหามเหสี” สามารถตัดได้เป็น “ไป หาม เห สี” กับ “ไป หา มเหสี” ซึ่งเมื่อพิจารณาจะจำนวนคำแล้ว วิธีการนี้ก็จะเลือกประโยค “ไป หา เมหสี” ซึ่งเป็นประโยคที่ถูกต้อง สำหรับในกรณีที่ตัดคำแล้วเกิดได้จำนวนคำที่เท่ากันก็ให้นำการตัดคำแบบเลือกคำยาวที่สุดเข้ามาช่วยพิจารณา ตัวอย่างเช่นประโยค “ฉันนั่งตากลม” สามารถตัดคำได้ทั้งหมด 2 แบบคือ “ฉัน นั่ง ตาก ลม” และ “ฉัน นั่ง ตา กลม” ซึ่งจะมีจำนวนคำเท่ากันทั้ง 2 ประโยค แต่เมื่อใช้การตัดคำแบบเลือกคำยาวที่สุดเข้ามาพิจารณา ประโยคที่ได้คือ “ฉัน นั่ง ตาก ลม”

สรุปวิธีการนี้จะสามารถช่วยแก้ไขข้อบกพร่องของการตัดคำแบบเลือกคำที่ยาวที่สุดได้ เพราะว่าการเลือกคำที่ยาวที่สุดเมื่อเจอข้อความที่กำกวมก่อน โดยไม่มีการพิจารณาถึงข้อความถัดไป ซึ่งมีลักษณะเหมือนการใช้ขั้นตอนวิธีแบบโลภ (Greedy Algorithm) ที่พิจารณาเฉพาะบริเวณใกล้ๆเท่านั้น แต่วิธีการตัดคำโดยเลือกแบบเหมือนมากสุดจะเป็นการใช้ขั้นตอนวิธีแบบโลภโดยพิจารณาข้อความทั้งหมดแทน แต่อย่างไรก็ตามเนื่องจากวิธีการนี้ใช้เฉพาะพจนานุกรมในการตัดคำเท่านั้น ดังนั้นการตัดคำนี้ยังไม่สามารถที่จะตัดคำได้ถูกต้องทั้งหมด แต่ถ้าจะให้ถูกต้องทั้งหมดนั้น จำเป็นจะต้องมีการนำโครงสร้างทางไวยากรณ์หรือความสัมพันธ์ทางความหมายเข้ามาใช้ประกอบในการพิจารณาด้วย

2.3 ยุคการใช้คลังข้อความ

จากการพัฒนาการตัดคำในยุคที่ผ่านมาเราใช้เพียงกฎ หรือพจนานุกรมในการแบ่งคำเท่านั้น ทำให้การตัดคำในยุคก่อนไม่สามารถที่จะตัดคำได้ถูกต้องทั้งหมด และในยุคนี้ (ปัจจุบัน) เครื่องคอมพิวเตอร์มีประสิทธิภาพมากยิ่งขึ้น มีหน่วยความจำมากขึ้นเป็นจำนวนมาก และได้มีการพัฒนาคลังข้อความ (Corpus) ขึ้นจำนวนมาก ทำให้ในยุคนี้ได้มีการพัฒนาการตัดคำขึ้นมาใหม่ โดยนอกเหนือการใช้กฎ พจนานุกรมแล้วยังมีการนำความรู้ต่างๆ จากคลังข้อความเข้ามาประยุกต์ใช้ด้วย ตัวอย่างความรู้ที่ได้จากคลังข้อความเช่น ค่าสถิติการใช้คำภายในคลังข้อความและลักษณะไวยากรณ์ที่ใช้ในคลังข้อความ เป็นต้น การตัดคำโดยใช้คลังข้อความในยุคนี้มีการพัฒนาในแบบต่างๆ ดังต่อไปนี้คือ

2.3.1 อัศนีย์ ก่อตระกูลและคณะ

ในงานนี้ทำการวิจัยเรื่อง “A Statistical Approach to Thai Word Filtering” (Asanee Kawtrakul et al. ,1997) เนื่องจากปัญหาของการวิเคราะห์หน่วยคำ (Morphological Analysis) สำหรับภาษาไทยนั้นจะมีปัญหาต่างๆ ดังนี้คือ ปัญหาความกำกวม ปัญหาความกำกวมของการกำหนดหน้าที่คำ

(Part-of-Speech Tagging Ambiguity) และปัญหาการสะกดคำผิด โดยปัญหาต่างๆ เหล่านี้จะทำให้เกิดผลลัพธ์ที่ตัดคำและกำหนดหน้าที่ของคำแล้วออกมาหลายๆ รูปแบบ ซึ่งในงานวิจัยนี้จะทำการลดผลลัพธ์ที่ไม่เหมาะสมออกไป เพื่อที่สามารถจะทำให้พาสเซอร์(Parser) ทำงานได้รวดเร็วขึ้น

ในงานวิจัยนี้จะนำเรื่องสถิติเข้ามาใช้แก้ปัญหาการตัดคำและการกำหนดหน้าที่ของคำ โดยมีการนำเรื่องโมเดลไตรแกรม เข้ามาช่วยในการแก้ปัญหาการตัดคำ การคำนวณค่าความน่าจะเป็นของประโยคโดยใช้โมเดลไตรแกรมสามารถคำนวณได้ดังที่แสดงในสมการที่ 2-1

$$\begin{aligned} P(W) &= \prod_{i=1}^{n} P(w_{i,n}) \\ &= \prod_{i=1}^{n} P(w_i \mid w_{i-1}, w_{i-2}) \end{aligned} \tag{2-1}$$

จากสมการที่ 2-1 คือการคำนวณหาค่าความน่าจะเป็นของแต่ละประโยค โดย W คือประโยคที่ตัดคำแล้ว และประโยค W จะประกอบไปด้วยคำต่างๆ ซึ่ง $W = w_1 w_2 \ldots w_n$ โดยที่ w_i คือคำศัพท์ และการคำนวณค่าความน่าจะเป็นของแต่ละประโยคจะมีข้อกำหนดว่า ความน่าจะเป็นของ w_i จะขึ้นอยู่กับ w_{i-1} และ w_{i-2} เท่านั้น

แต่เนื่องจากการคำนวณค่าความน่าจะเป็นตามสมการ 2-1 นั้นจะต้องใช้คลังข้อความขนาดใหญ่มาก โดยคลังข้อความควรจะมากกว่า n^3 คำ โดยที่ n คือจำนวนคำที่เป็นไปได้ทั้งหมด สาเหตุที่วิธีการนี้ต้องใช้คลังข้อความที่มีขนาดมากกว่า n^3 คำ เนื่องจากวิธีนี้จะต้องมีการนำค่าสถิติการเกิดของคำ 3 คำที่ติดกันมาใช้ในการคำนวณ ดังนั้นเพื่อให้มีค่าสถิติของการเกิดคำ 3 คำที่ติดกันทุกๆแบบ อย่างน้อยที่สุดจะต้องใช้ n^3 คำ ซึ่งในความจริงเราไม่สามารถหาคลังข้อความขนาดดังกล่าวได้ ทำให้มีการประมาณสมการที่ 2-1 เป็นสมการที่ 2-2 แทน

$$\begin{aligned} \prod_{i=1}^{n} P(w_i \mid w_{i-2,i-1}) = \prod_{i=1}^{n} (&\lambda_1 P(w_n) + \lambda_2 P(w_n \mid w_{n-1}) \\ &+ \lambda_3 P(w_n \mid w_{n-1}, w_{n-2})) \end{aligned} \tag{2-2}$$

จากสมการที่ 2-2 นี้จะเป็นการแก้ปัญหาเรื่องจำนวนข้อมูลที่นำมาใช้นั้นไม่เพียงพอ โดยจะมีการนำค่าความน่าจะเป็น ของไบแกรม (Bigram) และยูนิแกรม (Unigram) เข้ามาช่วยในการคำนวณด้วย และค่า λ_1 , λ_2 ,λ_3 ให้มีค่าเท่ากับ 0.1, 0.3, 0.6 ตามลำดับ ซึ่งได้นำมาจาก (Charniak, 1996)

สรุปผลจากงานวิจัยนี้สามารถลดรูปแบบของการตัดคำที่ไม่เหมาะสมลงไปได้จำนวนมาก ส่งผลให้งานการวิเคราะห์หน่วยคำนั้นสามารถจะทำงานได้รวดเร็วขึ้น

2.3.2 สุรพันธ์ เมฆนาวินและคณะ

จากการตัดคำที่ผ่านมาจะมีการนำเอาพจนานุกรมเข้ามาใช้ในการตัดคำเพียงอย่างเดียวเท่านั้น และการนำวิทยาการศึกษาสำนึก (Heuristics) ต่างๆ เข้ามาช่วยแก้ปัญหาความกำกวมที่เกิดขึ้นนั้นไม่สามารถที่จะแก้ปัญหาความกำกวมได้ทั้งหมด ดังนั้นจึงได้มีการพัฒนาการตัดคำขึ้น โดยมีการนำวิธีการทางสถิติเข้ามาช่วยในการแก้ไขปัญหาคำกำกวม ซึ่งวิธีการทางสถิติที่นำมาใช้คือการใช้ค่าสถิติที่เกิดจากลำดับของหน้าที่คำ หรืออาจกล่าวได้ว่าเป็นการนำเอาส่วนหนึ่งของไวยากรณ์ มาใช้ในการแก้ไขปัญหาความกำกวม

การตัดคำโดยใช้หน้าที่คำแบบไตรแกรมโมเดล (Surapant Meknavin, Paisarn Charoenpornsawat and Boonserm Kijsirikul, 1997) คือการตัดคำโดยมีการนำเอาค่าสถิติ ซึ่งพิจารณาจากความต่อเนื่องของหน้าที่คำ ส่วนวิธีการเลือกแบบการตัดคำที่ดีที่สุดนั้นทำได้โดยหาประโยคที่มีความน่าจะเป็นมากที่สุด โดยการหาความน่าจะเป็นของแต่ละประโยคสามารถคำนวณตามสมการที่ 2-3

$$\begin{aligned} P(W_i) &= \sum_{T} P(W_i, T_i) \\ &= \sum_{T} \prod_{i} P(t_i \mid t_{i-1}, t_{i-2}) \times P(w_i \mid t_i) \end{aligned} \qquad (2\text{-}3)$$

จากสมการที่ 2-1 W_i คือประโยคที่ตัดคำแล้ว ซึ่งนำมาจากประโยคที่มีคะแนนที่ดีที่สุด N อันดับแรก โดยวิธีการตัดคำคือการตัดคำโดยเลือกแบบเหมือนมากที่สุด และ $W_i = w_1 w_2 .. w_n$ โดย w_i คือคำที่ตัดได้ ส่วน $T_i = t_1 t_2 .. t_n$ โดย t_i คือหน้าที่คำของ w_i และ $P(w_i \mid t_i)$ กับ $P(t_i \mid t_{i-1}, t_{i-2})$ สามารถคำนวณได้จากคลังข้อความ สรุปความหมายจากสมการนี้คือการหาแบบการตัดคำที่ดีที่สุด โดยพิจารณาจากผลรวมความน่าจะเป็นของหน้าที่คำทุกแบบที่เป็นไปได้ของแต่ละประโยค และมีข้อกำหนดว่าความน่าจะเป็นของการเกิดหน้าที่คำที่ตำแหน่งปัจจุบันจะขึ้นอยู่กับหน้าที่คำของ 2 คำก่อนหน้าเท่านั้น กล่าวอีกนัยหนึ่งคือวิธีการนี้จะไม่สนใจว่าหน้าที่ของคำที่ถูกต้องที่สุดจะเป็นอะไร แต่จะสนใจว่าการตัดคำแบบไหนจะดีที่สุด ทำให้วิธีการนี้เหมาะสมสำหรับงานที่ต้องการทราบขอบเขตคำเพียงอย่างเดียวเท่านั้น

สรุปวิธีการนี้จะสามารถแก้ไขปัญหาความกำกวมได้ดีกว่าวิธีการก่อนๆ ที่ได้กล่าวมาทั้งหมด เนื่องจากมีการพิจารณาถึงหน้าที่ของคำเข้ามาประกอบด้วย แต่อย่างไรก็ตามในกรณีที่ข้อความกำกวมมีหน้าที่คำเหมือนกัน วิธีการนี้ก็ไม่สามารถที่จะแก้ไขปัญหาได้ดีนัก และข้อจำกัดอีกอย่างหนึ่งก็คือเราจะ

ต้องทำการเก็บค่าสถิติจากคลังข้อความ (Corpus) โดยที่คลังข้อความที่ดีควรจะนำมาจากเอกสารหลายประเภท และจะต้องมีขนาดใหญ่พอสมควร ดังนั้นประสิทธิภาพของวิธีการตัดคำแบบนี้จะขึ้นอยู่กับคลังข้อความด้วย

2.3.3 อัศนีย์ ก่อตระกูลและคณะ

ในยุคนี้เนื่องจากคอมพิวเตอร์มีความรวดเร็วในการประมวลผลมากขึ้น และมีหน่วยความจำขนาดใหญ่ทำให้ ปัญหาที่สำคัญของยุคนี้ไม่ใช่เรื่องความรวดเร็ว หรือการประหยัดเนื้อที่หน่วยความจำ แต่ปัญหาที่สำคัญในยุคนี้คือความถูกต้องของการตัดคำ เพราะในระบบต่างๆ มีความต้องการที่จะได้การตัดคำที่มีประสิทธิภาพสูงที่สุด ซึ่งในยุคนี้ได้มีการพัฒนาตัดคำวิธีการต่างๆ ดังที่ได้กล่าวไปแล้ว แต่วิธีการดังกล่าวก็ยังไม่สามารถที่จัดการกับคำศัพท์ที่ไม่ปรากฏในพจนานุกรมได้ ดังนั้นในยุคนี้จึงได้มีคิดค้นวิธีการที่จะนำมาใช้ในการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมขึ้น โดย อัศนีย์ ก่อตระกูลและคณะ (Asanee Kawtrakul et. al.,1997)

ในงานวิจัยได้ทำแก้ปัญหาเรื่องคำศัพท์ที่ไม่ปรากฏในพจนานุกรม โดยไม่ได้แค่หาขอบเขตของคำเท่านั้น แต่ยังสามารถที่จะบอกถึงหน้าที่คำและแสดงถึงลักษณะทางความหมาย (Semantic Attribute) และยังสามารถที่แก้ไขคำในกรณีที่เกิดการสะกดผิดด้วย ซึ่งในงานวิจัยนี้มีการนำวิธีการทางสถิติ (Statistical Model) และมีการนำกฎต่างๆ เข้ามาช่วยในการพิจารณาด้วย

ขั้นตอนวิธีในการแก้ปัญหาเรื่องคำที่ไม่ปรากฏในพจนานุกรม ประกอบด้วย 3 ขั้นตอนซึ่งแสดงดังต่อไปนี้

2.3.3.1 ทำการตัดคำอย่างง่ายๆ โดยใช้โมเดลไตรแกรม (Asanee Kawtrakul et al., 1995) ซึ่งเมื่อทำการตัดคำแล้วผลลัพธ์ที่ได้สามารถแบ่งออกได้เป็น 2 กรณีคือ

2.3.3.1.1 กรณีที่เกิดข้อความที่ไม่ปรากฏในพจนานุกรม

2.3.3.1.2 กรณีที่ไม่เกิดข้อความที่ไม่ปรากฏในพจนานุกรม

2.3.3.2 ถ้าผลลัพธ์จากข้อ 2.3.3.1 ที่ได้เป็นกรณีที่ 2.3.3.1.1 ให้ไปทำขั้นตอนที่ 2.3.3.3 แต่ถ้าเป็นในกรณีที่ 2.3.3.1.2 ให้ใช้ โมเดลการแบ่งโดยใช้ความหมาย (Semantic Segmenting Model) ซึ่งสามารถคำนวณได้ดังสมการที่ 2-4

$$\arg\max_{t_{1,n}} P(w_{1,n}, t_{1,n}) = \prod_{i=1}^{n} P(t_i \mid t_{i-1}, t_{i-2}) * P(w_i \mid t_i) \qquad (2\text{-}4)$$

โดย $w_{1,n}$ จะหมายถึงประโยคที่แบ่งคำแล้วได้ออกมาเป็น w_1 ถึง w_n และ $t_{1,n}$ คือ ลำดับแท็กความหมาย (Semantic tag) โดย t_i คือแท็กความหมายของ w_i ซึ่งในสมการนี้จะทำการหาแท็กความหมายของแต่ละคำ ที่จะทำให้ค่าความน่าจะเป็นของ $P(w_{1,n} , t_{1,n})$ มีค่ามากที่สุด แล้วนำค่ามาเปรียบเทียบกับค่าขีดเริ่มเปลี่ยน (Threshold) ตามเงื่อนไขดังต่อไปนี้

2.1 $P(w_{1,n} , t_{1,n}) \geq$ ค่าขีดเริ่มเปลี่ยน จะหมายความว่าไม่มีการเกิดคำศัพท์ที่ไม่ปรากฏในพจนานุกรมขึ้น

2.2 $P(w_{1,n} , t_{1,n}) <$ ค่าขีดเริ่มเปลี่ยน แล้วให้เลือกคำที่ทำให้ $P(w_{i,i+3}, t_{i,i+3})$ มีค่าน้อยที่สุด และให้ไปทำขั้นตอนที่ 3 ต่อไป

1. ขั้นตอนนี้จะเป็นการหาขอบเขตของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมและบอกถึงหน้าที่คำและความหมายคำ ซึ่งภายในขั้นตอนนี้จะประกอบด้วยขั้นตอนย่อย 4 ขั้นตอนคือ

1.1 การทายขอบเขตโดยใช้วิทยาการศึกษาสำนึก

1.2 สร้างเซตของตัวเลือกของคำศัพท์ที่ไม่ปรากฏในพจนานุกรม โดยใช้กฎที่มีการพิจารณาจากบริบท (Context Sensitive Rules) และมีการพิจารณาลักษณะของตัวอักษร

1.3 ลองแทนที่ส่วนที่น่าสงสัยว่าจะเป็นคำที่ไม่มีในพจนานุกรม ด้วยตัวเลือกต่างๆ (Unknown Word Candidate) สำหรับวิธีการสร้างตัวเลือกของคำที่ไม่ปรากฏในพจนานุกรมนั้น จะอธิบายในส่วนถัดไป

1.4 คำนวณค่าความน่าจะเป็น โดยใช้สมการที่ 2-4

ถ้า $P(w_{1,n}, t_{1,n}) >=$ ค่าขีดเริ่มเปลี่ยน แสดงว่าคำที่เลือกเป็นคำที่ถูกต้อง แต่ถ้า $P(w_{1,n},t_{1,n}) > P(w_{1,n},t_{1,n})$ ก่อนหน้า ให้กลับไปทำขั้นตอนที่ 3.3 สำหรับในกรณีอื่นๆ แสดงว่ามีข้อผิดพลาดเกิดขึ้น

วิธีการสร้างตัวเลือกของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมสามารถจะสร้างได้ดังต่อไปนี้

1. เมื่อมีการตัดคำแล้วเกิดข้อความที่ไม่ปรากฏในพจนานุกรมจำนวน 2 ชุดที่อยู่ใกล้กันโดยห่างกันไม่เกิน 2 ตัวอักษร ก็ให้สร้างคำใหม่ โดยรวมข้อความที่ไม่ปรากฏในพจนานุกรมและคำที่อยู่ระหว่างข้อความทั้ง 2 เข้าด้วยกัน

2. เมื่อทำการตัดคำแล้วพบข้อความที่ไม่ปรากฏในพจนานุกรม ก็ให้สร้างคำใหม่ ซึ่งสามารถจะสร้างได้ทั้งหมด 4 แบบคือ

2.1 ให้ข้อความนั้นเป็นคำเลย

2.2 สร้างคำใหม่โดยนำข้อความนั้นมารวมกับคำข้างหน้า

2.3 สร้างคำใหม่โดยนำข้อความนั้นมารวมกับคำถัดไป

2.4 สร้างคำใหม่โดยนำข้อความนั้นมารวมกับคำข้างหน้าและคำถัดไป

บทที่ 3

การกำกับหน้าที่คำ

การกำกับหน้าที่คำคือการระบุหน้าที่คำของคำที่กำหนดมา ส่วนหน้าที่คำ (Part of Speech: POS) คือสิ่งที่ระบุว่าคำนั้นทำหน้าที่ทางไวยากรณ์เป็นอะไรภายในประโยคหนึ่งๆ โดยคำหนึ่งคำอาจจะมีหลายหน้าที่ได้ขึ้นอยู่กับตำแหน่งภายในประโยคนั้นๆ เช่นคำว่า “ฉัน” สามารถจะมีหน้าที่ได้ 2 อย่างคือ 1. เป็นคำกริยา 2. เป็นคำสรรพนาม ตัวอย่างเช่น “พระฉันเพลก่อนเที่ยง” คำว่า ”ฉัน” ในที่นี้ก็จะทำหน้าที่เป็นคำกริยา แต่ถ้าในประโยค “ฉันกับน้องชอบไปดูหนังด้วยกัน” คำว่า ”ฉัน” ในที่นี้จะทำหน้าที่เป็นคำสรรพนาม เป็นต้น

สำหรับการวิเคราะห์ทางด้านภาษา ชุดหน้าที่คำ (POS Tag Set) ที่นำมาใช้จะมีผลต่อการวิเคราะห์เป็นอย่างมาก และในความจริงการระบุหน้าที่คำเพียงบอกว่าเป็น คำนาม คำกริยา คำสรรพนาม คำคุณศัพท์ คำวิเศษณ์ ฯลฯ นั้นไม่เพียงพอที่จะนำมาใช้ในการวิเคราะห์ทางภาษาศาสตร์ ดังนั้นนักภาษาศาสตร์จึงได้มีการสร้างชุดหน้าที่คำที่จะมีประสิทธิภาพเพียงพอต่อการนำไปใช้ในการวิเคราะห์ทางภาษาศาสตร์ โดยลักษณะชุดหน้าที่คำของแต่ละภาษานั้นจะมีลักษณะแตกต่างกันไปตามภาษานั้นๆ และในภาษาหนึ่งๆ อาจจะมีชุดหน้าที่คำได้หลายชุดโดยขึ้นอยู่กับแนวคิดของการนำชุดหน้าที่คำไปใช้ ตัวอย่างเช่นในภาษาอังกฤษได้มีการสร้างชุดหน้าที่คำออกมาหลายชุดเช่น ชุดหน้าที่คำเพนทรีแบงค์ (Penn Treebank tagset) ซึ่งในเพนทรีแบงค์นั้นได้แบ่งหมวดหมู่หน้าที่คำออกเป็น 36 ชนิด (Allen, 1995) และ ชุดหน้าที่คำบราวน์ (Brown tagset) ได้แบ่งหมวดหมู่คำออกเป็น 80 ชนิด สำหรับภาษาไทยได้มีสร้างชุดหน้าที่คำออกมาหลายชุดเช่นกัน ตัวอย่างเช่น ชุดหน้าที่คำออร์คิด (Orchid tagset) ซึ่งแบ่งหมวดหมู่คำเป็น 47 ชนิด (Virach Sornlertlamvanich, Thatsanee Charoenporn and Isahara, 1997) และชุดหน้าที่คำของมหาวิทยาลัยเกษตรศาสตร์ เป็นต้น

จากที่ได้กล่าวในบทนำว่า งานด้านการประมวลผลภาษาธรรมชาติสำหรับภาษาไทยนั้น การตัดคำจะเป็นงานขั้นตอนแรกที่จะต้องมีการทำก่อนที่จะนำไปประมวลผลอื่นๆ ต่อไป แต่ในบางงานนอกจากจะต้องตัดคำแล้ว ยังต้องการข้อมูลเพิ่มเติมของคำนั้นเช่น หน้าที่คำ หรือ ความหมาย เป็นต้น และสำหรับขั้นตอนวิธีของการตัดคำที่ใช้ในวิทยานิพนธ์นี้จะต้องมีการนำหน้าที่คำเข้ามาช่วยในการประมวลผลด้วย ดังนั้นในบทนี้จะอธิบายถึงวิธีการกำกับหน้าที่ของคำ (Part-of-Speech Tagging)

การกำกับหน้าที่ของคำนั้นมีอยู่หลายแนวคิด ได้แก่ แนวคิดการใช้กฎ (Rule-based Approaches) แนวคิดการใช้สถิติ (Statistic-based Approaches) ซึ่งสำหรับงานด้านการประมวลผลภาษาธรรมชาตินั้น นิยมแนวคิดทางด้านสถิติมากกว่า เนื่องจากสามารถรองรับข้อมูลในหลายๆ รูปแบบได้โดยไม่ทำให้เกิดข้อผิดพลาดขึ้น และยังสามารถที่จะคำนวณค่าสถิติต่างๆ ที่นำมาใช้ได้โดยอัตโนมัติ ส่วนแนวคิดการใช้กฎจะต้องมีการใช้มนุษย์วิเคราะห์เพื่อสร้างกฎให้ครอบคลุมภาษาที่ใช้ทั้งหมด ซึ่งจะเป็นเรื่องที่ยุ่งยากมาก และยิ่งมีจำนวนกฎเกณฑ์มากขึ้นก็จะยิ่งทำให้เกิดความกำกวมมากขึ้นตามไปด้วย แต่ในปัจจุบันได้มีการพัฒนาแนวคิดการใช้กฎให้สามารถมีการสร้างกฎขึ้นมาได้เอง โดยสามารถจะสรุปกฎจากคลังข้อความที่มีอยู่ได้ แต่อย่างไรก็ตามวิธีการที่พัฒนาขึ้นมานี้ยังทำงานได้ช้า ทำให้งานต่อมาได้มีการปรับปรุงเพื่อที่จะเพิ่มความเร็วในการทำงาน

เพื่อที่จะหาวิธีการที่จะนำมาใช้ในการแก้ไขปัญหานั้น ในบทนี้จะอธิบายถึงลักษณะปัญหาการกำกับหน้าที่คำซึ่งจะได้กล่าวในส่วนถัดไป

3.1 ลักษณะปัญหาของการกำกับหน้าที่คำ

จากนิยามของการกำหนดหน้าที่คำที่ได้กล่าวไปแล้วในตอนต้น สามารถกำหนดให้เป็นสมการได้ดังสมการ 3-1

$$\tau = \max_{c_1,\ldots,c_t} \arg PROB(C_1,\ldots,C_t \mid w_1,\ldots,w_t) \qquad (3\text{-}1)$$

โดยที่ τ คือ $C_1,\ldots C_t$ ที่ทำให้ค่าความน่าจะเป็นตามสมการที่ 3-1 มีค่ามากที่สุด C_i คือหน้าที่คำของคำ w_i ส่วน $w_1,w_2,\ldots,w_t$ คือลำดับของคำในประโยคหนึ่งๆ และ $C_1,C_2,\ldots,C_t$ คือลำดับของหน้าที่คำในประโยคนั้น ส่วนความหมายจากสมการที่ 3-1 จะหมายถึงว่าภายในประโยคหนึ่งๆ ประกอบไปด้วยลำดับของคำ $w_1,w_2,..,w_t$ และให้เลือกลำดับของหน้าที่คำ $C_1,C_2,\ldots C_t$ ที่ทำให้ค่าความน่าจะเป็นตามสมการที่ 3-1 มีค่ามากที่สุด

3.2 วิธีการแก้ปัญหา

สำหรับการกำหนดหน้าที่ของคำที่จะนำมาใช้ในวิธีการนี้ คือนำแนวคิดการใช้สถิติเข้ามาช่วย โดยนำการกำหนดหน้าที่คำแบบไตรแกรมเข้ามาใช้

เมื่อพิจารณาจากสมการที่ 3-1 จะเห็นว่าการที่จะหาค่าความน่าจะเป็นของลำดับหน้าที่คำในสมการนี้ จำเป็นจะต้องมีคลังข้อความขนาดใหญ่มาก ซึ่งในความเป็นจริงการหาคลังข้อความขนาดดัง

กล่าวจะไม่สามารถทำได้อย่างแน่นอน ดังนั้นจึงมีการปรับปรุงสมการที่ 3-1 โดยมีการนำกฎของเบย์ (Bayes' rule) เข้ามาใช้ ซึ่งแสดงในสมการที่ 3-2

$$PROB(A \mid B) = \frac{PROB(B \mid A) \times PROB(A)}{PROB(B)} \tag{3-2}$$

ดังนั้นเมื่อนำกฎของเบย์เข้ามาปรับปรุงสมการที่ 3-1 จะได้สมการใหม่ แสดงตามสมการที่ 3-3

$$\tau = \max_{c_1,\ldots,c_t} \arg \frac{PROB(C_1,...,C_t) \times PROB(w_1,....,w_t \mid C_1,...,C_t)}{PROB(w_1,....,w_t)} \tag{3-3}$$

จากสมการที่ 3-3 จะเห็นว่าต้องมีการคำนวณ $PROB\ (w_1,\ldots,w_t)$ ซึ่งเป็นค่าคงที่ ดังนั้นเราจึงสามารถละค่านี้ได้ โดยไม่กระทบกับผลลัพธ์ ทำให้สมการที่ 3-3 สามารถลดรูปได้ ซึ่งแสดงในสมการที่ 3-4

$$\tau = \max_{c_1,\ldots,c_t} \arg PROB(C_1,...,C_t) \times PROB(w_1,....,w_t \mid C_1,...,C_t) \tag{3-4}$$

เมื่อทำการลดรูปจากสมการที่ 3-1 มาเป็นสมการที่ 3-4 แล้ว ยังจำเป็นต้องการคลังข้อความจำนวนมากเช่นกัน แต่อย่างไรก็ตามสมการนี้สามารถที่จะทำการคำนวณโดยประมาณได้ ซึ่งจะทำให้การคำนวณสามารถทำให้ง่ายขึ้น และจำนวนคลังข้อความที่จะนำมาใช้นั้นมีขนาดลดลงอย่างมาก โดยสร้างสมมุติฐานว่าหน้าที่ของคำหนึ่งๆ จะขึ้นอยู่กับหน้าที่คำของคำก่อนหน้า 1 คำ หรือ 2 คำ ซึ่งสามารถเรียกได้ว่าเป็นแบบไบแกรม (Bigram) หรือ ไตรแกรม (Trigram) ตามลำดับ

สำหรับการกำกับหน้าที่คำที่จะนำมาใช้ในวิทยานิพนธ์นี้ จะนำโมเดลไตรแกรมเข้ามาใช้ ดังนั้นการคำนวณค่า $PROB(C_1,..,C_t)$ จะสามารถคำนวณได้ดังสมการที่ 3-5

$$PROB(C_1,...,C_t) \cong \prod_{i=1}^{t} PROB(C_i \mid C_{i-1}, C_{i-2}) \tag{3-5}$$

ส่วนการคำนวณค่า $PROB(w_1,\ldots,w_t \mid C_1,\ldots,C_t)$ ในสมการที่ 3-4 สามารถจะประมาณ โดยสมมุติว่าหน้าที่ของคำหนึ่งคำจะไม่ขึ้นอยู่กับคำก่อนหน้า หรือคำที่ตามหลัง ดังนั้นการคำนวณค่า $PROB(w_1,\ldots,w_t \mid C_1,\ldots,C_t)$ สามารถจะประมาณได้ดังสมการ 3-6

$$PROB(w_1,....,w_t \mid C_1,...,C_t) \cong \prod_{i=1}^{t} PROB(w_i \mid C_i) \tag{3-6}$$

ดังนั้นจากสมการที่ 3-4 สามารถจะประมาณตามสมการที่ 3-5 และ 3-6 ได้ดังสมการที่ 3-7

$$\tau = \max_{c_1,\ldots,c_t} \arg \prod_{i=1}^{t} PROB(C_i \mid C_{i-1}, C_{i-2}) \times PROB(w_i \mid C_i) \qquad (3\text{-}7)$$

เมื่อได้สมการในการหาลำดับของหน้าที่คำ ดังสมการ 3-7 แล้ว จะเห็นว่าคลังข้อความที่จะนำมาใช้ในการเก็บค่าสถิตินั้นจะมีขนาดน้อยลง ทำให้ในความเป็นจริงการหาคลังข้อความที่มีขนาดเพียงพอที่จะสามารถนำมาใช้ตามสมการ 3-7 นั้นเป็นไปได้จริง ดังนั้นในวิทยานิพนธ์นี้ก็จะนำสมการนี้มาใช้ในการกำกับหน้าที่คำ แต่ในความเป็นจริงถ้าเขียนโปรแกรมจากสมการนี้ตามตรงจะมีการคำนวณจำนวนมาก ทำให้โปรแกรมทำงานได้ช้า ดังนั้นจึงต้องมีการปรับปรุงโดยนำเทคนิคเรื่องไดนามิคโปรแกรมมิ่ง (Dynamic Programming) เข้ามาช่วย ส่วนในรายละเอียดนั้น จะทำการอธิบายในส่วนถัดไป

3.3 การเพิ่มประสิทธิภาพ

เนื่องจากเมื่อมีการเขียนโปรแกรมกำกับหน้าที่คำ โดยใช้สมการที่ 3-7 ขึ้นโดยตรงนั้นจะทำให้โปรแกรมได้ช้ามาก เนื่องจากถ้านำประโยคที่ตัดคำแล้วมากำกับหน้าที่คำ ซึ่งประกอบไปด้วยจำนวนคำ T คำ และจำนวนหน้าที่คำสามารถแบ่งออกได้เป็น N หมวดหมู่ ในกรณีที่แย่ที่สุดคือคำหนึ่งคำสามารถมีหน้าที่คำได้ทั้งหมด N หมวดหมู่ ดังนั้นในการคำนวณตามสมการที่ 3-7 จะต้องใช้การคำนวณประมาณ $k \times N^T$ โดยค่า k คือค่าคงที่ ซึ่งจะเห็นว่าวิธีการนี้จะใช้เวลาค่อนข้างมากโดยจะขึ้นอยู่กับจำนวนคำในประโยคที่จะนำมากำกับหน้าที่คำ โดยเวลาที่ใช้นั้นจะเป็นสัดส่วนแบบเอกซ์โปเนนเชียล (Exponential) ซึ่งจะยิ่งช้ามากถ้าจำนวนคำในประโยคมาก ดังนั้นจึงมีการพัฒนาโดยนำเทคนิคเรื่องไดนามิคโปรแกรมมิ่งเข้ามาช่วย และขั้นตอนวิธีที่นำมาใช้ในการปรับปรุงความเร็วนี้มีชื่อเรียกว่า ขั้นตอนวิธีวิเทอร์บิ (Viterbi Algorithm) เมื่อนำขั้นตอนวิเทอร์บิเข้ามาประยุกต์ใช้กับการกำกับหน้าที่คำแบบไตรแกรม ซึ่งแสดงได้ตามรูปที่ 3-1

ขั้นตอนวิธีวิเทอร์บิที่แสดงในรูปที่ 3-1 นั้น จะมีการสร้างแถวลำดับ (Array) ขนาด $N \times N \times T$ จำนวน 2 ชุดโดย N คือจำนวนหน้าที่คำที่เป็นไปได้ทั้งหมด และ T คือจำนวนคำในประโยคที่จะนำมากำกับหน้าที่คำ โดยที่แถวลำดับชุดแรกคือแถวลำดับ $seqscore[\ i\ ][\ j\ ][\ t\ ]$ จะทำการเก็บค่าความน่าจะเป็นที่ดีที่สุดของการกำกับหน้าที่คำของ $w_1, \ldots, w_t$ ซึ่งคำที่หน้าที่ของคำ w_t กับ w_{t-1} จะมีหน้าที่คำเป็น L_i และ L_j ตามลำดับ ส่วนแถวลำดับชุดที่สองคือ $backptr[\ i\ ][\ j\ ][\ t\ ]$ จะเก็บหน้าที่คำของคำ $t-2$ เมื่อคำที่ t และ t-1 มีหน้าที่คำเป็น L_j และ L_i ตามลำดับ

กำหนดให้ $w_1,..,w_t$ เป็นลำดับคำในประโยค $L_1,\ldots, L_n$ เป็นหน้าที่คำที่เป็นไปได้ $Prob(w_t \mid L_i)$คือค่าความน่าจะเป็นของคำศัพท์ w_t เมื่อกำหนดให้มีหน้าที่คำเป็น L_i และค่าความน่าจะเป็นของไตรแกรมคือ $Prob\ (\ L_k \mid L_i,L_j\ )$ ดังนั้นให้หาลำดับของหน้าที่คำ $C_1,..,C_T$ ที่เป็นของลำดับคำในประโยคที่มีความน่าจะเป็นมากที่สุด

Initialization Step

for i =1 to N do

for j=1 to N do

seqscore[i][j][1] = Prob($W_1|L_i$) × Prob($L_i \mid \phi$) × Prob($W_2 \mid L_j$) × Prob($Lj \mid Li, \phi$)

backptr[i] [j] [2] = 0

Iteration Step

for t=3 to T do

for j=1 to N do

for k=1 to N do

seqscore[j][k][t] = max $_{i=1,N}$ (seqscore[i][j][t-1] × Prob($L_k|L_j ,L_i$)) × Prob($W_t|L_k$)

backptr[j][k][t] = ค่า i ที่ทำให้ค่าสมการที่ผ่านมาเป็นค่าที่มากที่สุด

Sequence Identification Step

C[T] = k and C[T-1] = j โดยที่ j และ k นั้นทำให้ seqscore[j][k][T] มีค่ามาที่สุด

for i=T-2 to 1 do

C[i] = backptr [C[i+1]] [C[i+1]] [i+1]

รูปที่ 3-1 **ขั้นตอนวิธีวิเทอร์บิ** (Viterbi Algorithm)

จากการคำนวณหาหน้าที่คำโดยใช้ขั้นตอนวิธีวิเทอรบินั้นจะสามารถลดเวลาการคำนวนได้ โดยจากของเดิมที่ต้องใช้เวลาเป็นสัดส่วนกับ kN^T ส่วนวิธีการนี้ก็จะใช้เวลาเป็นสัดส่วน N^3T ดังนั้นจะเห็นว่าเมื่อนำขั้นตอนวิธีวิเทอร์บิเข้ามาใช้นั้นในการกำกับหน้าที่คำจะสามารถลดเวลาได้เป็นจำนวนมาก

บทที่ 4

โครงสร้างของพจนานุกรม

ในบทนี้จะขอกล่าวถึงวิธีการจัดเก็บคำศัพท์ที่จะนำมาใช้ในการตัดคำ เนื่องจากการตัดคำด้วยพจนานุกรมจะต้องมีการสืบค้นหาคำศัพท์ในพจนานุกรมเป็นจำนวนมาก ทำให้ต้องมีการพิจารณานำโครงสร้างข้อมูลแบบต่างๆ ที่เหมาะสมเข้ามาใช้ในการจัดเก็บคำศัพท์ในพจนานุกรมเพื่อที่จะสามารถจัดเก็บคำศัพท์ได้อย่างมีประสิทธิภาพในแง่ของความรวดเร็วในการสืบค้น และใช้เนื้อที่หน่วยความจำน้อย

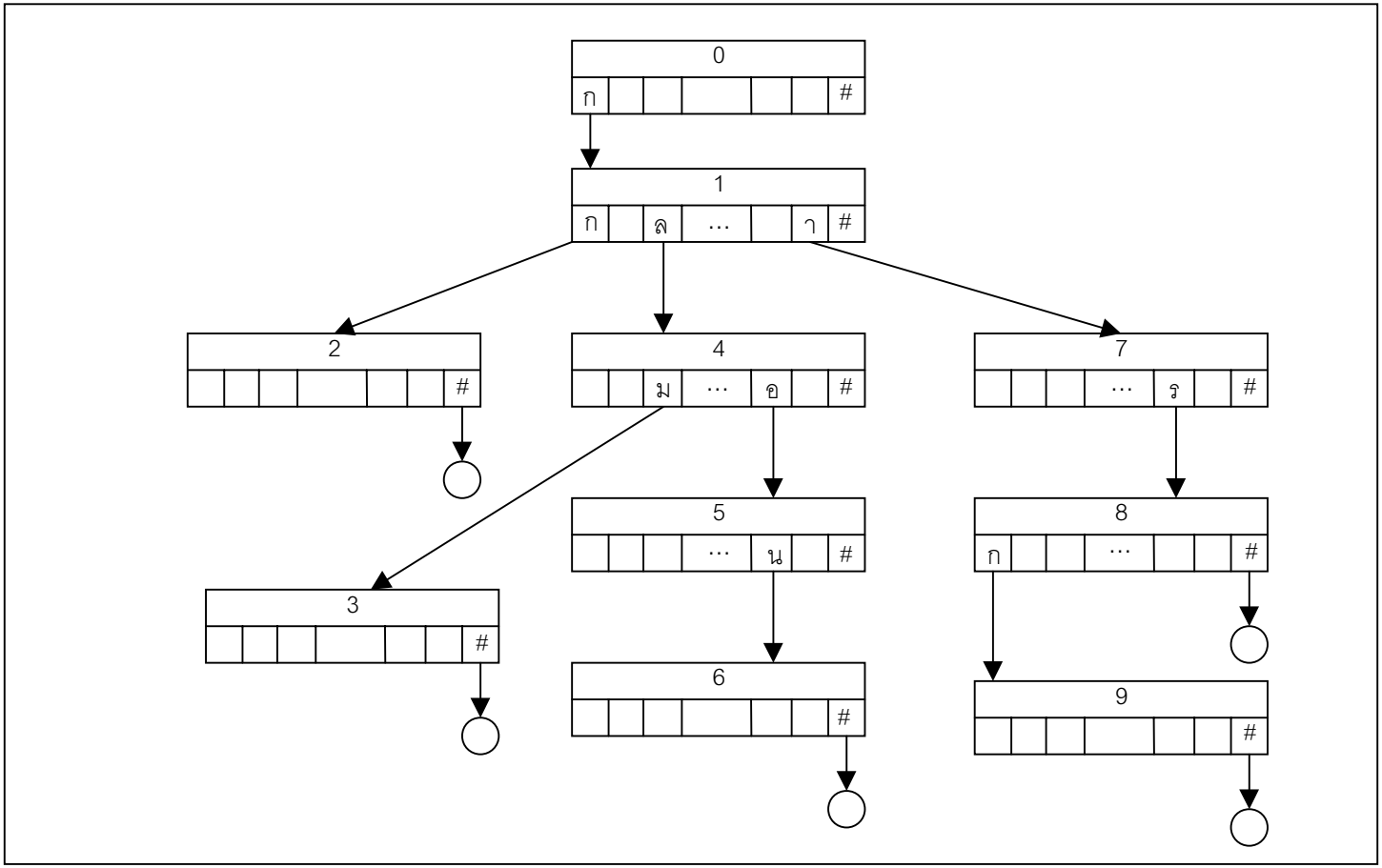
เนื่องจากการตัดคำส่วนมากจะนิยมทำการตัดคำจากซ้ายมาขวา ยกเว้นในกรณีที่นำการตัดคำมาใช้ในการขึ้นบรรทัดใหม่ (Word Wrap) เพียงอย่างเดียวเท่านั้นซึ่งอาจจะทำการตัดคำจากขวามาซ้ายก็ได้ สำหรับการตัดคำแบบเลือกคำที่ยาวที่สุด หรือการตัดคำโดยเลือกแบบที่เหมือนมากที่สุด หรือการตัดคำแบบอื่นๆ ที่ต้องมีการตัดคำที่เป็นไปได้ทุกๆ แบบก่อนแล้วค่อยนำมาประมวลผลต่อไป จะนิยมทำการตัดคำจากซ้ายมาขวามากกว่า ดังนั้นในวิทยานิพนธ์นี้จะกล่าวถึงโครงสร้างของพจนานุกรมที่เหมาะสมที่จะนำมาใช้กับการตัดคำจากซ้ายไปขวาเท่านั้น

สำหรับโครงสร้างของพจนานุกรมที่นำมาใช้ในวิทยานิพนธ์นี้คือโครงสร้างข้อมูลแบบทรัย (Trie) ซึ่งในส่วนถัดไปจะอธิบายถึงโครงสร้างข้อมูลแบบทรัย และสำหรับเหตุผลที่นำโครงสร้างข้อมูลแบบทรัยมาจัดเก็บพจนานุกรมนั้นจะอธิบายในส่วนถัดไป

4.1 โครงสร้างข้อมูลแบบทรัย

โครงสร้างข้อมูลแบบทรัย (Corman, Leiserson and Rivest, 1990: Frakes and Baeza-Yates, 1992) จะมีลักษณะคล้ายกับโครงสร้างข้อมูลแบบต้นไม้ แต่วิธีการจัดเก็บข้อมูลจะแตกต่างกัน โดยที่โครงสร้างข้อมูลแบบทรัยนี้จะจัดเก็บตัวอักษรของคำศัพท์ ซึ่งโครงสร้างข้อมูลแบบต้นไม้นั้นจะจัดเก็บข้อมูลทั้งคำ สำหรับโครงสร้างข้อมูลแบบทรัย แสดงในรูปที่ 4-1

จากรูป 4-1 เป็นตัวอย่างโครงสร้างข้อมูลแบบทรัย ที่ใช้ในการจัดเก็บคำศัพท์ กก กลม กลอน การ และ การก



รูปที่ 4-1 โครงสร้างข้อมูลแบบทรัย

จากรูป 4-1 โครงสร้างของทรัยจะประกอบไปด้วยโหนดต่างๆ โดยที่ข้อมูลภายใน 1 โหนดจะประกอบไปด้วย พอยเตอร์ที่ชี้ไปยังโหนดของตัวอักษรถัดไป ซึ่งมีจำนวนพอยเตอร์เท่ากับจำนวนตัวอักขระที่จะอนุญาตให้มีได้ในพจนานุกรมบวกกับอักขระที่ใช้ระบุเป็นตัวจบคำศัพท์ (Terminator) อีก 1 ตัวอักษร ซึ่งสัญลักษณ์ที่ใช้ในที่นี้คือเครื่องหมาย #

สำหรับการสืบค้นในโครงสร้างข้อมูลแบบทรัยนี้จะทำโดย เริ่มต้นที่โหนด 0 ถ้าต้องการค้นหาคำศัพท์ก็ให้นำอักษรทีละตัวจากคำศัพท์ที่ต้องการ มาดูว่าภายในโหนด 0 นั้นมีพอยเตอร์ของตัวอักษรที่ต้องการชี้ไปโหนดอื่นหรือไม่ ถ้าไม่มีแสดงว่าคำนั้นไม่มีอยู่ในพจนานุกรม แต่ถ้ามีพอยเตอร์ที่ชี้ไปที่โหนดถัดไปก็ให้เดินที่โหนดที่พอยเตอร์นั้นชี้ไป แล้วนำตัวอักษรตัวถัดไปมาทำตามขั้นตอนแบบเดิมจนหมด เมื่อนำตัวอักษรทั้งหมดจากคีย์มาเดินในทรัยแล้ว ให้เดินด้วยอักษร “#” แล้วดูว่าค่าพอยเตอร์มีค่าเท่ากับค่าว่าง (null) หรือไม่ ถ้าเท่าแสดงว่าไม่มีคำศัพท์นั้นในพจนานุกรม แต่ถ้าไม่เท่าก็แสดงว่ามีคำศัพท์นั้นอยู่ในพจนานุกรม โดยพอยเตอร์นี้ส่วนใหญ่จะชี้ไปที่ตำแหน่งของข้อมูลของคำนั้น

ตัวอย่างการสืบค้นคำศัพท์จากโครงสร้างข้อมูลแบบทรัย จากรูปที่ 4-1 ถ้าต้องการสืบค้นคำว่า “กลม” มีขั้นตอนดังนี้คือ โหนด 0 จะเป็นโหนดเริ่มต้น ดังนั้นนำตัวอักษร “ก” เข้ามาเดินภายในทรัยก็จะไปที่โหนด 1 หลังจากนั้นก็นำตัวอักษร “ล” เข้ามาเดินต่อจะไปที่โหนด 4 แล้วก็นำตัวอักษรตัวถัดไปคือ “ม” เข้ามาเดินจะไปที่โหนด 3 สุดท้ายเมื่อทำการค้นหามาถึงตัวอักษรสุดท้ายของคีย์แล้ว ให้เดินด้วย “#” ซึ่งค่าที่ได้ไม่เท่ากับค่าว่างแสดงว่าคำว่า “กลม” มีอยู่ในพจนานุกรม

4.2 ประสิทธิภาพด้านความเร็ว

โครงสร้างของพจนานุกรมที่เหมาะสมที่จะนำมาใช้ในการตัดคำต้องมีความรวดเร็วในการทำงาน ซึ่งเวลาที่ใช้ในการค้นหาคำศัพท์ในพจนานุกรมนี้จะขึ้นอยู่กับจำนวนครั้งทั้งหมดที่ใช้ในการค้นหาคำ และเวลาที่ใช้ในการค้นหาแต่ละครั้ง ดังนั้นโครงสร้างของพจนานุกรมที่ดีจะต้องมีจำนวนครั้งในการค้นหาคำที่มีอยู่ในประโยคที่นำมาตัดคำนั้นเป็นจำนวนน้อย และเวลาที่ใช้ในการค้นหาแต่ละครั้งจะต้องไม่มาก

พจนานุกรมถูกนำมาใช้ในการตัดคำเมื่อต้องการหาบริเวณของกลุ่มตัวอักษรในประโยคที่เป็นคำศัพท์ในพจนานุกรม ดังนั้นถ้าประโยคที่จะนำเข้ามาตัดคำดังแสดงในสมการที่ 4-1

$$S=c_1c_2\ c_3\ c_4\ c_5...c_n \tag{4-1}$$

จากสมการที่ 4-1 ให้ S คือประโยคที่นำมาตัดคำซึ่งจะประกอบไปด้วย $c_1c_2\ c_3\ c_4\ c_5...c_n$ โดยที่ c_i คือตัวอักษรภาษาไทย

สำหรับโครงสร้างของพจนานุกรมที่จะต้องมีการนำคีย์ (Key) ทั้งคีย์มาใช้ในการสืบค้นเช่น โครงสร้างข้อมูลแบบตารางแฮช (Hash Table) โครงสร้างข้อมูลแบบไบนารีทรี (Binary Tree) โครงสร้างข้อมูลแบบบีพลัส-ทรี (B^+ -Tree) หรือ โครงสร้างข้อมูลแบบอินเด็กซ์ซีเควนเชียล (Index Sequential) นั้นจำนวนครั้งที่ต้องใช้ในการหาคำศัพท์จากสมการที่ 4-1 จะใช้ทั้งหมดคือ $O(n^2)$ ครั้ง

แต่สำหรับโครงสร้างข้อมูลแบบทรัยนั้นไม่จำเป็นต้องใช้คีย์ทั้งคีย์มาใช้ในการสืบค้น สามารถทำการสืบค้นโดยนำเข้ามาทีละตัวอักษรได้ ดังนั้นทำให้จำนวนครั้งที่ต้องการในการหาคำศัพท์จากสมการที่ 4-1 ได้เพียง $O(n)$ ครั้ง ซึ่งจะใช้จำนวนครั้งในการสืบค้นน้อยกว่าโครงสร้างของพจนานุกรมที่จะต้องมีการเปรียบเทียบทั้งคีย์ ดังนั้นโครงสร้างข้อมูลแบบทรัยนี้จึงถือว่ามีประสิทธิภาพดีกว่า ในแง่จำนวนครั้งที่ใช้ในการสืบค้น

ส่วนความเร็วในการค้นหาภายในโครงสร้างข้อมูลแบบทรัยนั้นจะไม่ขึ้นอยู่กับจำนวนคำที่มีในพจนานุกรม แต่จะขึ้นอยู่กับความยาวของคีย์ที่ใช้ในการสืบค้น ซึ่งถือได้ว่าเป็นข้อดีสำหรับลักษณะโครงสร้างข้อมูลประเภทนี้ แต่สำหรับโครงสร้างข้อมูลแบบตารางแฮช โครงสร้างข้อมูลแบบไบนารีทรี โครงสร้างข้อมูลแบบบีพลัส-ทรี หรือ โครงสร้างข้อมูลแบบอินเด็กซ์ซีเควนเชียล นั้น เวลาที่ใช้ในการสืบค้นจะขึ้นอยู่กับจำนวนคำที่เก็บในพจนานุกรมด้วย

4.3 ประสิทธิภาพในการใช้หน่วยความจำ

จากลักษณะโครงสร้างการจัดเก็บข้อมูลแบบทรัยนี้ แสดงให้เห็นดังในรูปที่ 4-1 จะเห็นว่าทรัยนั้นต้องมีการจองเนื้อที่เป็นจำนวนมาก ซึ่งเนื้อที่ส่วนใหญ่จะไม่ได้ถูกใช้งาน ดังนั้นจึงได้มีผู้พัฒนาทรัยแบบใหม่ขึ้นมา ซึ่งจะช่วยลดเนื้อที่ในหน่วยความจำลง โดยโครงสร้างแบบใหม่มีชื่อว่า โครงสร้างทรัยแบบสามแถวลำดับ (Triple-Array Trie) ซึ่งพัฒนาโดยจอห์นสัน (Johnson ,1975) โดยที่วิธีการนี้จะมีการใช้แถวลำดับจำนวน 3 ชุดในการเก็บค่า ต่อมาได้มีการพัฒนาให้ใช้หน่วยความจำน้อยลงไปอีก วิธีการที่พัฒนาขึ้นมาใหม่นี้ได้พัฒนาโดยอะเอโอะ (Aoe, 1989) และโครงสร้างทรัยแบบใหม่ที่พัฒนาขึ้นมีชื่อว่า โครงสร้างทรัยแบบแถวลำดับคู่ ซึ่งในวิธีการนี้จะใช้แถวลำดับแค่เพียง 2 ชุดเท่านั้น ทำให้วิธีการนี้สามารถจะลดการใช้เนื้อได้มากขึ้น และจากลักษณะโครงสร้างข้อมูลแบบทรัยนั้น จะมีจุดเด่นคือมีการเก็บข้อความส่วนหน้าเข้าด้วยกัน ซึ่งเป็นสาเหตุให้โครงสร้างของทรัยนั้นมีขนาดเล็กกว่าโครงสร้างข้อมูลแบบอื่น

ส่วนในรายละเอียดของขั้นตอนวิธีการเพิ่มคำ ลดคำและ แก้ไขคำเข้าไปในโครงสร้างทรัยแบบแถวคู่นั้นสามารถศึกษาได้จาก (Aoe, 1989; สมปรารถนา รัทยานนท์, 2535)

บทที่ 5

ปัญหาความกำกวมและคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

ในบทนี้จะกล่าวถึงสาเหตุที่ทำให้การตัดคำด้วยพจนานุกรมไม่สามารถตัดคำได้ถูกต้อง โดยจะแบ่งปัญหาออกเป็นส่วนๆ เพื่อที่จะหาแนวทางในการแก้ไขปัญหาของแต่ละส่วนต่อไป สาเหตุที่ทำให้การตัดคำโดยใช้พจนานุกรมผิดพลาดมีอยู่ 2 สาเหตุคือ 1. ความกำกวม 2. คำศัพท์ที่ไม่ปรากฏในพจนานุกรม

5.1 ความกำกวม

ความกำกวมจะเกิดขึ้นเมื่อมีข้อความที่สามารถจะแบ่งได้หลายแบบ โดยทุกๆ คำที่เกิดขึ้นในแต่ละแบบจะเป็นคำศัพท์ที่พบในพจนานุกรมทั้งหมด ในกรณีนี้จะไม่พิจารณาถึงคำที่ไม่ปรากฏในพจนานุกรมจากความกำกวมที่เกิดขึ้น ในงานวิทยานิพนธ์นี้จะแบ่งประเภทของข้อความที่กำกวมออกเป็น 2 แบบตามลักษณะของข้อความที่กำกวมคือ

5.1.1 ข้อความกำกวมที่ขึ้นกับบริบท (Context Dependent Words)

คือข้อความกำกวมที่จำเป็นจะต้องพิจารณาข้อความรอบข้าง เพื่อเลือกแบบการตัดคำที่ดีที่สุด หรือกล่าวอีกนัยหนึ่งก็คือข้อความกำกวมประเภทนี้สามารถที่จะตัดคำได้หลายแบบ และแต่ละแบบก็มีความหมาย ทำให้การที่จะเลือกแบบตัดคำที่ถูกต้องนั้นจำเป็นต้องพิจารณาบริบทรอบๆ ด้วย เช่น

ตากลม สามารถตัดได้เป็น ตาก ลม หรือ ตา กลม
โคลง สามารถตัดได้เป็น โคลง หรือ โค ลง
ที่อยู่ สามารถตัดได้เป็น ที่อยู่ หรือ ที่ อยู่
มากว่า สามารถตัดได้เป็น มาก ว่า หรือ มา กว่า
สาวกลับ สามารถตัดได้เป็น สาวก ลับ หรือ สาว กลับ

5.1.2 ข้อความกำกวมที่ไม่ขึ้นกับบริบท (Context Independent Words)

คือข้อความกำกวมที่ไม่มีความจำเป็นต้องพิจารณาข้อความรอบข้าง และสามารถจะเลือกได้ทันทีว่าควรจะตัดคำแบบไหน หรือกล่าวอีกนัยหนึ่งก็คือข้อความที่สามารถตัดคำได้หลายๆ แบบแต่จะมีเพียงแบบเดียวเท่านั้นที่มีความหมาย ตัวอย่างเช่น

ขนบนอก	สามารถตัดได้เป็น	ขน บน อก หรือ ขนบ นอก
โคนกลีบดอก	สามารถตัดได้เป็น	โคน กลีบ ดอก หรือ โค นก ลีบ ดอก
นำมากลั่น	สามารถตัดได้เป็น	นำ มา กลั่น หรือ นำ มาก ลั่น
ไปหามเหสี	สามารถตัดได้เป็น	ไป หา มเหสี หรือ ไป หาม เห สี
คอกว้าง	สามารถตัดได้เป็น	คอ กว้าง หรือ คอก ว้าง

หมายเหตุ ข้อความที่ขีดเส้นใต้คือข้อความที่ถูกต้อง

จากลักษณะของข้อความกำกวมที่กล่าวมา จะเห็นว่าความกำกวมที่เกิดขึ้นนั้นมีอยู่ทั้งหมด 2 แบบ สำหรับการเพิ่มประสิทธิภาพของการตัดคำให้ดียิ่งขึ้นนั้นมีความจำเป็นที่จะต้องหาวิธีการต่างๆ มาแก้ปัญหาความกำกวมทั้ง 2 แบบ โดยการแก้ปัญหาความกำกวมที่ขึ้นกับบริบทนั้นจะทำได้ยากกว่า และจะต้องใช้วิธีการที่ซับซ้อนกว่าการแก้ปัญหาความกำกวมแบบที่ไม่ขึ้นกับบริบท นอกเหนือจากการแก้ไขปัญหาความกำกวมแล้ว สิ่งที่จะต้องพิจารณาต่อไปคือเรื่องของคำศัพท์ที่ไม่ปรากฏในพจนานุกรม ซึ่งจะกล่าวต่อไป

5.2 คำศัพท์ที่ไม่ปรากฏในพจนานุกรม

สำหรับปัญหาเรื่องคำศัพท์ที่ไม่ปรากฏในพจนานุกรม เป็นสาเหตุสำคัญที่ทำให้การตัดคำโดยใช้พจนานุกรมไม่สามารถจะตัดคำเหล่านั้นได้ถูกต้อง เนื่องจากคำในภาษาไทยสามารถที่จะเกิดขึ้นมาได้ใหม่โดยการประสมระหว่างคำหรือพยางค์ได้ ทำให้การหาขอบเขตของคำที่ไม่ปรากฏในพจนานุกรมทำได้ยาก แต่อย่างไรก็ตามปัจจุบันได้มีผู้คิดและพัฒนาแก้ไขปัญหาดังกล่าว ซึ่งสามารถแก้ไขปัญหาเหล่านี้ได้ดีพอสมควร โดยมีการนำเรื่องสถิติ ไวยากรณ์และความหมายเข้ามาช่วยพิจารณาในการแก้ปัญหา แต่ก็ยังไม่สามารถที่จะแก้ไขปัญหาได้ทั้งหมด ทำให้ยังคงต้องมีการพัฒนาและค้นหาวิธีการที่จะแก้ไขอีกต่อไป

ประเภทต่าง ๆ ของคำที่ไม่ปรากฏในพจนานุกรม

คำศัพท์ที่ไม่ปรากฏอยู่ในพจนานุกรมสามารถแบ่งออกได้เป็น 6 ประเภทคือ

1. ชื่อเฉพาะ
2. คำจากภาษาต่างประเทศ
3. คำทับศัพท์
4. คำย่อ
5. คำราชาศัพท์
6. คำที่สะกดผิด

จากการรวบรวมค่าสถิติของคำศัพท์ที่ไม่มีในพจนานุกรมนั้น จะแสดงดังตารางที่ 1 โดยการรวบรวมของ (Asanee Kawtrakul et al., 1997)

ตารางที่ 5-1 ตารางค่าสถิติของคำที่ไม่มีในพจนานุกรมประเภทต่างๆ ในเอกสารต่างๆ

ชนิดของคำที่ไม่มีในพจนานุกรม	ชนิดของเอกสาร (%)		
	วิทยาศาสตร์	ข่าว	สารคดี
ชื่อเฉพาะ	3.06	73.4	51.15
คำทับศัพท์	43.6	6.88	14.39
คำย่อ	3.90	9.63	4.63
คำจากภาษาต่างประเทศ	47.49	1.15	21.34
คำราชาศัพท์	-	-	5.91

จากตารางที่ 5-1 จะเห็นว่าคำศัพท์ที่ไม่มีอยู่ในพจนานุกรมในเอกสารประเภทข่าว และสารคดี ส่วนใหญ่จะเป็นชื่อเฉพาะ ส่วนในเอกสารประเภทวิทยาศาสตร์ จะมีการใช้คำทับศัพท์และคำจากภาษาต่างประเทศเป็นจำนวนมาก เนื่องจากการแก้ปัญหาการตัดคำของคำศัพท์ที่ไม่มีอยู่ในพจนานุกรมนั้นจะขึ้นอยู่กับประเภทของคำด้วย ดังนั้นคำศัพท์ประเภทแรกที่วิทยานิพนธ์นี้จะทำการแก้ปัญหาคือคำศัพท์ประเภทชื่อเฉพาะ เพราะว่าคำประเภทนี้มีการใช้เป็นจำนวนมาก และการแก้ปัญหาจะสามารถนำไปประยุกต์ใช้ในงานด้านการสืบค้นสารสนเทศ (Information Retrieval) ได้ด้วย

5.2.1 ลักษณะของคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

เนื่องจากคำในภาษาไทยนั้นสามารถจะเกิดขึ้นใหม่โดยอาจจะเกิดจากการประสมระหว่างคำ หรือระหว่างพยางค์เป็นต้น ดังนั้นจึงเป็นสาเหตุให้การหาขอบเขตของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้นทำได้ยาก โดยจะต้องมีการนำคำหรือข้อความรอบๆ บริเวณคำศัพท์ที่ไม่ปรากฏในพจนานุกรมมาช่วยในการหาขอบเขต ดังนั้นก่อนที่จะทำการอธิบายขั้นตอนในการหาขอบเขตของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้น ในส่วนนี้จะอธิบายถึงลักษณะของคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

ลักษณะของคำศัพท์ที่ไม่ปรากฏในพจนานุกรม จะเห็นว่าคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอาจจะประกอบไปด้วยข้อความที่มีในพจนานุกรม (Known String) กับข้อความที่ไม่ปรากฏในพจนานุกรม (Unknown String) และเมื่อพิจารณาจากลักษณะของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมสามารถที่จะแบ่งคำที่ไม่ปรากฏในพจนานุกรมได้เป็น 2 ประเภทใหญ่ๆ คือ

➢ คำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างชัดเจน (Explicit Unknown Word) คำศัพท์ประเภทนี้คือคำศัพท์ที่ไม่ปรากฏในพจนานุกรม โดยภายในคำนั้นๆ จะไม่มีข้อความส่วนใดๆ ภายในคำนั้นที่เป็นคำที่พบอยู่ในพจนานุกรม ตัวอย่างเช่นคำว่า “โลตัส”, “แฮรี่”, “สุนีย์” ฯลฯ

➢ คำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้น (Hidden Unknown Word) คำศัพท์ประเภทนี้คือคำศัพท์ที่ไม่ปรากฏในพจนานุกรม โดยภายในคำนั้นๆ จะมีข้อความส่วนหนึ่งส่วนใดภายในคำนั้นที่เป็นคำที่พบอยู่ในพจนานุกรม ตัวอย่างเช่นคำว่า “สุ**มา**นี”, “คธา**พงษ์**”, “ **สม ชาย**”, “**สม ศักดิ์**” เป็นต้น และคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นนี้สามารถจะแบ่งเป็นประเภทย่อยๆ ได้อีก 2 ประเภทคือ

1. คำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นบางส่วน (Partially Hidden Unknown Word) คือคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นที่เกิดจากการประกอบ ระหว่างคำที่ปรากฏในพจนานุกรมกับข้อความที่ไม่ปรากฏในพจนานุกรม ตัวอย่างเช่น “สุมานี” และ “คธาพงษ์”

2. คำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วน (Fully Hidden Unknown Word) คือคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นที่เกิดจากการประกอบไปด้วยคำที่ปรากฏในพจนานุกรมทั้งหมด หรืออาจกล่าวได้ว่าเป็นคำที่สร้างขึ้นใหม่โดยมีการนำคำศัพท์ต่างๆ มาประกอบกัน ตัวอย่างเช่น “สมชาย” เกิดจากการประสมระหว่าง “สม” กับ “ชาย” และคำว่า “สมหญิง” เกิดจากการประสมคำระหว่าง “สม” กับ “หญิง” เป็นต้น

จากลักษณะของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมที่ได้อธิบายมาในข้างต้นแล้ว จะเห็นว่าคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้นสามารถจะเกิดได้หลายๆ รูปแบบ ดังนั้นการแก้ไขปัญหานี้จะทำได้ยากเพราะจะไม่สามารถรู้ขอบเขตที่แน่นอนของคำได้ ทำให้การแก้ปัญหาจะต้องมีการนำคำบริบทเข้ามาช่วย ซึ่งในวิทยานิพนธ์นี้จะอธิบายวิธีแก้ปัญหาเรื่องคำศัพท์ที่ไม่ปรากฏในพจนานุกรมในบทที่ 7

บทที่ 6

การเรียนรู้ของเครื่อง

คอมพิวเตอร์ถือได้ว่าเป็นเครื่องจักรกลที่สามารถคำนวณหรือประมวลผลต่างๆ ได้รวดเร็วกว่ามนุษย์ อย่างไรก็ตามคอมพิวเตอร์ก็ยังคงต้องรับคำสั่งหรือความรู้ต่างๆ จากมนุษย์ ดังนั้นนักคอมพิวเตอร์จึงได้มีการพยายามที่จะให้คอมพิวเตอร์มีความสามารถในการเรียนรู้ โดยปัจจุบันได้มีการพัฒนาการเรียนรู้ของเครื่องในแบบต่างๆ ขึ้นมากมาย และได้นำไปประยุกต์ใช้กับงานต่างๆ เป็นจำนวนมาก เช่น มีการนำไปใช้ในการรู้จำตัวอักษร (Boonserm Kijsirikul, Sukree Sinthupinyo and Apinya Supanwansa, 1998) โดยมีการนำรูปภาพของแต่ละตัวอักษรเข้ามาสอนให้คอมพิวเตอร์รู้จักว่าเป็นตัวอักษรใด หรือนำไปใช้ในการสร้างกฎเพื่อใช้ในการแก้ไขคำผิด (Golding and Roth,1996 ; Surapant Meknavin et.al.,1998) เป็นต้น

จากงานการตัดคำที่ผ่านมามีการนำพจนานุกรม ค่าสถิติต่างๆ ของคำเข้ามาช่วยในการประมวลผล ซึ่งสามารถเพิ่มความถูกต้องได้มากกว่าเดิม แต่อย่างไรก็ตามยังมีอีกหลายกรณีที่ไม่สามารถแก้ปัญหาด้วยวิธีการดังกล่าวได้ ดังนั้นในวิทยานิพนธ์นี้จะนำเอาคุณลักษณะ (Feature) ต่างๆ เข้ามาช่วยในการแก้ไขปัญหาความกำกวม และชื่อเฉพาะที่ไม่พบในพจนานุกรม โดยคุณลักษณะต่างๆ ที่ได้มานั้นจะนำมาจากการเรียนรู้ของเครื่อง โดยวิธีการเรียนรู้ของเครื่องที่จะนำมาศึกษาและประยุกต์ใช้คือ ริปเปอร์ และ วินโนว์

ในบทนี้จะอธิบายถึงวิธีการเรียนรู้ของเครื่องที่ชื่อว่า ริปเปอร์ และ วินโนว์ ส่วนรายละเอียดและวิธีการที่จะนำการเรียนรู้ของเครื่องทั้ง 2 แบบมาประยุกต์ใช้ในการตัดคำจะนำมากล่าวในบทถัดไป

6.1 **ริปเปอร์** (RIPPER : Repeated Incremental Pruning to Produce Error Reduction)

ริปเปอร์ (Cohen, 1995) เป็นกระบวนการเรียนรู้ของเครื่องแบบหนึ่ง โดยริปเปอร์สามารถทำการเรียนรู้จากตัวอย่างที่ให้มา และจะสรุปออกมาเป็นกฎให้โดยอัตโนมัติ โดยรูปแบบของกฎที่ได้มาจะมีรูปแบบดังนี้

$$\text{if } T_1 \text{ and } T_2 \text{ and } \ldots T_n \text{ then } C$$

โดย T_i คือเงื่อนไขต่างๆ ส่วน C คือคำตอบ จากกฎที่ได้มานี้จะมีความหมายคือ ถ้าเงื่อนไข T_1, T_2, ... และ T_n เป็นจริงแล้ว ดังนั้นคำตอบก็คือ C ส่วนเงื่อนไขที่ริปเปอร์ยอมให้มีได้มีอยู่ 4 แบบคือ เท่ากับ (=) , มากกว่าเท่ากับ (≥), น้อยกว่าเท่ากับ (≤) และ การเป็นสมาชิกในเซต (∈)

ตัวอย่างกฎที่ได้จากการเรียนรู้ของริปเปอร์ ในการแก้ปัญหาความกำกวมของข้อความ “ที่อยู่” ซึ่งสามารถตัดคำได้ 2 แบบคือ “ที่อยู่” และ “ที่ อยู่” เช่น

- ถ้า pw1=”ของ” แล้วให้ตัดเป็น “ที่อยู่“
- ถ้า มี ∈ pw310 และ mt2=RPRE แล้วให้ตัดเป็น ที่อยู่

โดย pw1 คือ คำถัดไปทางด้านขวามือของข้อความ “ที่อยู่” pw310 คือคำบริบทที่อยู่ด้านขวามือของข้อความ “ที่อยู่” และ mt2 คือหน้าที่คำของคำที่อยู่ถัดไปทางด้านซ้าย 2 คำของข้อความ “ที่อยู่” ส่วน RPRE คืออักษรย่อที่ใช้แสดงหน้าที่คำ ซึ่งหมายถึงคำบุพบท

6.1.1 ขั้นตอนการเรียนรู้ของริปเปอร์

การเรียนรู้ของเครื่องริปเปอร์ได้มีการปรับปรุงมากจาก การเรียนรู้ของเครื่องไอเร็ป (IREP :Incremental Reduced Error Pruning) ซึ่งพัฒนาโดย (Furnkranz and Widmer, 1994) สำหรับรายละเอียดที่ริปเปอร์ได้มีการปรับปรุงจากไอเร็ป คือ 1. ในการลดกฎ (Pruning rules) โดยริปเปอร์จะยอมให้มีการลบกฎที่มีจำนวนเงื่อนไขมากกว่า 1 เงื่อนไขได้ ในขณะที่ไอเร็ปจะอนุญาตให้ลบกฎที่มีเงื่อนไข 1 เงื่อนไขเท่านั้น และ 2. ในการหยุดเพิ่มกฎ สำหรับริปเปอร์จะหยุดเพิ่มกฎเมื่อกฎที่ได้นั้นมีข้อผิดพลาดเกินกว่า 50% และ 3. ริปเปอร์อนุญาตให้มีคุณสมบัติเพิ่มเติมจากไอเร็ปคือ อนุญาตให้มีการใส่คุณสมบัติไม่ครบ อนุญาตให้มีตัวแปรแบบตัวเลข และอนุญาตให้คำตอบที่ได้หลายคลาส (Multiple class) ซึ่งในไอเร็ปยอมให้คำตอบเป็นไปได้แค่เพียง 2 คำตอบคือจริงกับเท็จเท่านั้น

ขั้นตอนการสร้างกฎของริปเปอร์ คือเมื่อมีการนำตัวอย่างต่างๆ เข้ามาป้อนให้ริปเปอร์ ริปเปอร์จะพยายามสร้างกฎต่างๆ ที่สามารถครอบคลุมตัวอย่างที่มีอยู่ทั้งหมดให้มากที่สุด เมื่อริปเปอร์สร้างกฎทั้งหมดแล้ว ริปเปอร์ก็จะพยายามลดกฎที่เป็นกฎที่เฉพาะเจาะจงสำหรับตัวอย่างบางตัวอย่างเท่านั้น ซึ่งวิธีการนี้จะเป็นข้อดีในกรณีที่เราไม่มั่นใจว่าตัวอย่างที่เราจะนำมาให้ริปเปอร์นั้นมีความถูกต้องถึง 100% หรือกล่าวอีกนัยหนึ่งคือยอมให้มีการป้อนตัวอย่างที่ผิดได้บ้างเล็กน้อย

6.1.2 ข้อดีของริปเปอร์

1. กฎที่ได้จากการเรียนรู้ของริปเปอร์นั้นจะอยู่ในรูปแบบ ถ้า...แล้ว (if-then rule) ทำให้ง่ายต่อความเข้าใจ โดยเฉพาะอย่างยิ่งถ้าต้องการพิจารณาเข้าไปถึงลักษณะของกฎที่ได้มากับตัวอย่างที่นำไปใช้ใน

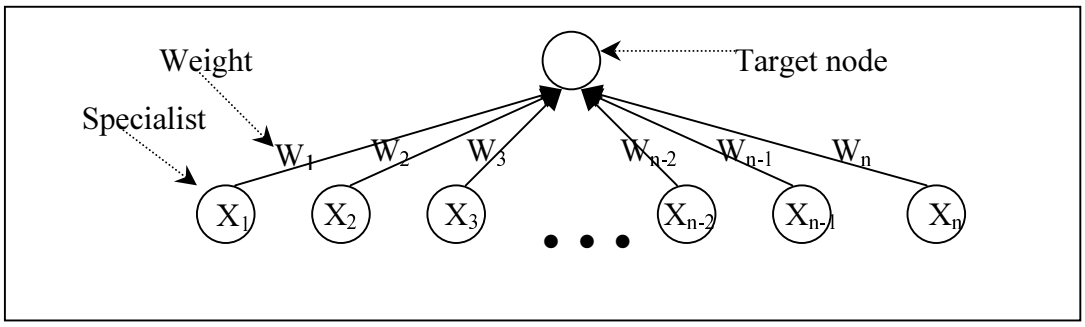
การเรียนรู้ ในกรณีนี้การใช้ริปเปอร์จะสามารถดีกว่าการนำกระบวนการเรียนรู้ของเครื่องแบบโครงข่ายประสาทเทียม (Neural network) หรือ การใช้ต้นไม้ตัดสินใจ (Decision tree)

2. การทำงานของริปเปอร์สามารถจะทำงานได้รวดเร็วกว่าขั้นตอนวิธีการเรียนรู้แบบกฎ (Rule learning algorithm) แบบอื่นๆ ในกรณีที่ตัวอย่างที่นำมาใช้ในการเรียนรู้มีจำนวนมากๆ

3. ริปเปอร์สามารถให้ผู้ใช้กำหนดเงื่อนไขสำหรับรูปแบบของกฎที่ได้ออกมา ทำให้กฎที่ได้ออกมานั้นจะมีความถูกต้องมายิ่งขึ้น

4. ริปเปอร์อนุญาตให้มีการใช้คุณสมบัติที่เป็นเซต (Set-valued attribute)ได้ ซึ่งในกรณีนี้ช่วยให้ลักษณะกฎที่ได้ออกมานั้นมีรูปแบบกะทัดรัด

6.2 **วินโนว์** (Winnow)



รูปที่ 6-1 โครงข่ายวินโนว์ (Winnow Network)

วินโนว์ (Littlestone, 1988; Golding and Roth, 1996; Blum, 1997) เป็นวิธีการเรียนรู้ของเครื่องวิธีหนึ่ง โดยวินโนว์จะเป็นการเรียนรู้โดยพิจารณาจากตัวอย่าง (Learning by examples) ที่ป้อนเข้ามา ซึ่งลักษณะของวินโนว์จะคล้ายกับโครงข่ายประสาทเทียม (Neural Network) ซึ่งจะแสดงในรูปที่ 6-1 วินโนว์จะประกอบด้วยโหนดต่างๆ ที่มีการเชื่อมไปสู่โหนดเป้าหมาย (Target node) โดยโหนดต่างๆ ซึ่งเราจะเรียกว่าผู้เชี่ยวชาญ (Specialist) โดยแต่ละผู้เชี่ยวชาญ จะทำการตรวจสอบค่าของคุณสมบัติ (attribute = value) จำนวน 2 คู่เท่านั้น เมื่อแต่ละผู้เชี่ยวชาญทำการตรวจสอบแล้วก็จะทำการทำนายผลลัพธ์ โดยแต่ละผู้เชี่ยวชาญ จะมีน้ำหนัก (Weight) หรือความน่าเชื่อถือไม่เท่ากัน เมื่อผู้เชี่ยวชาญได้ทำนายผลลัพธ์หมดแล้ว วินโนว์จะทำการรวมคะแนนโดยพิจารณาจากค่าความน่าเชื่อถือของแต่ละผู้เชี่ยวชาญแล้วค่อยทำนายออกมาเป็นคำตอบ

ตัวอย่างโคร่งข่ายวินโนว์ที่นำมาใช้ในการแก้ปัญหาความกำกวมของข้อความ “ที่อยู่” เช่น

- ผู้เชี่ยวชาญ X_1 ในรูป 6.1 จะพิจารณาเฉพาะตัวอย่างที่นำเข้ามาที่มี pw1=ของ และ pt1=RPRE แล้วทำการทำนายว่า จะต้องตัดคำได้ว่า “ที่อยู่” ด้วยค่าความน่าเชื่อถือ 20.2

- ผู้เชี่ยวชาญ X_2 ในรูป 6.1 จะพิจารณาเฉพาะตัวอย่างที่นำเข้ามาที่มีคำว่า “มี”∈pw310 และ pt1=RPRE แล้วทำการทำนายว่า จะต้องตัดคำได้ว่า “ที่ อยู่” ด้วยค่าความน่าเชื่อถือ 25.25 เป็นต้น

ดังนั้นเมื่อตัวอย่างที่นำเข้ามาป้อนให้กับวินโนว์ ผู้เชี่ยวชาญแต่ละตัวจะทำการตรวจสอบว่าคุณสมบัติและค่านั้นตรงกับค่าที่ผู้เชี่ยวชาญตัวนั้นรับผิดชอบหรือไม่ ในกรณีที่ตรงก็ทำการทำนายผลลัพธ์ออกมา แต่ถ้าไม่ตรงจะไม่ทำการทำนาย เมื่อทำนายครบทุกตัวแล้ว วินโนว์จะทำการรวมคะแนนแล้วทำนายผลลัพธ์ออกมา

6.2.1 **หลักการทำงานของวินโนว์**

1. กำหนดค่าเริ่มต้นให้กับความน่าเชื่อถือของผู้เชี่ยวชาญแต่ละตัวให้เป็น 1
2. นำตัวอย่างที่ทราบคำตอบแล้วส่งให้กับวินโนว์
3. ผู้เชี่ยวชาญแต่ละตัว จะทำการตรวจสอบคุณสมบัติที่ตนเองรับผิดชอบและทำการทำนายโดยพิจารณาจากข้อมูลที่ผ่านมา
4. วินโนว์จะทำการรวมคะแนนของผู้เชี่ยวชาญแต่ละตัว โดยพิจารณาจากค่าความน่าเชื่อถือด้วย
5. ในกรณีที่วินโนว์มีการทำนายผิด วินโนว์จะทำการปรับค่าความน่าเชื่อถือของผู้เชี่ยวชาญ
    5.1 สำหรับผู้เชี่ยวชาญที่ตอบผิด ค่าความน่าเชื่อถือจะถูกลดลงครึ่งหนึ่ง
    5.2 สำหรับผู้เชี่ยวชาญที่ตอบถูก ค่าความน่าเชื่อถือจะถูกเพิ่มขึ้น โดยคูณด้วย 1.5

หลังจากวินโนว์ได้ทำการเรียนรู้จากตัวอย่างที่ป้อนให้ทั้งหมดแล้ว เราก็จะได้โครงข่ายวินโนว์ (Winnow Network) และเราจะนำโครงข่ายนี้มาใช้ในการแก้ปัญหา

6.2.2 **ข้อดีของวินโนว์**

1. สามารถทำการเรียนรู้ได้เร็วเนื่องจากลักษณะของวินโนว์เป็นการเรียนรู้แบบค่อยๆ เรียนรู้ (Incremental algorithm)
2. สามารถจัดการกับคุณลักษณะต่างๆ ที่ไม่เกี่ยวข้องได้ดี (Littlestone, 1998)

บทที่ 7

การตัดคำภาษาไทยโดยใช้คุณลักษณะ

จากบทที่แล้วได้กล่าวถึงขบวนการเรียนรู้ของเครื่องแบบต่างๆ ไปแล้ว ในบทนี้จะกล่าวถึงวิธีการที่จะนำเอาการเรียนรู้ของเครื่องเข้ามาประยุกต์ใช้ในการเลือกคุณลักษณะต่างๆ จากคลังข้อความที่สามารถจะนำมาใช้ในการแก้ไขปัญหาคำกำกวมและปัญหาชื่อเฉพาะที่ไม่ปรากฏในพจนานุกรมได้ นิยามของคุณลักษณะในที่นี้หมายถึงข้อมูลใดๆ ที่สามารถจะนำมาใช้ในการแก้ไขปัญหาการตัดคำได้

7.1 **คุณลักษณะ**

คุณลักษณะที่จะนำมาใช้ในการแก้ไขปัญหาการตัดคำทั้งปัญหาความกำกวมและปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้น มีอยู่ 2 ชนิดคือ คำบริบท (Context Word) และ สิ่งที่เกิดร่วมกันโดยมีลำดับ (Collocation)

7.1.1 **คำบริบท** (Context Word)

คำบริบทคือ คำที่อยู่รอบๆ ข้อความหรือคำที่จะนำมาพิจารณา (Target String/ Target Word) สำหรับในงานวิทยานิพนธ์นี้จะนำบริบทที่อยู่ห่างจากข้อความหรือคำที่จะนำมาพิจารณาภายใน 10 คำก่อนหน้าหรือหลังข้อความที่จะนำมาพิจารณา ส่วนข้อความที่จะนำมาพิจารณาในที่นี้จะขึ้นอยู่กับลักษณะของปัญหาที่จะนำมาประยุกต์ใช้ ซึ่งในที่นี้จะนำไปประยุกต์ใช้กับ 2 ปัญหาคือ ปัญหาความกำกวม และ ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

ในการแก้ปัญหาความกำกวม ข้อความหรือคำที่จะนำมาพิจารณาคือข้อความที่กำกวมและคำบริบทสำหรับปัญหานี้ก็คือคำรอบๆ ข้อความที่กำกวมภายใน +/- 10 คำ สำหรับปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม ข้อความหรือคำที่จะนำมาพิจารณาคือ ตัวเลือกของคำที่ไม่ปรากฏในพจนานุกรม (Unknown Word Candidate) ส่วนบริบทสำหรับปัญหานี้ก็คือ คำรอบๆ ตัวเลือกของคำที่ไม่ปรากฏในพจนานุกรมภายใน +/- 10 คำ

ตัวอย่างของคำบริบทที่นำมาใช้ในการแก้ปัญหาความกำกวม เช่นถ้ามีข้อความกำกวม “ตากลม” และมีคำบริบทอยู่ด้านขวามือของข้อความนี้เป็นคำว่า “แป้ว” ก็จะตัดสินใจให้ตัดคำเป็น “ตา กลม”

7.1.2 **สิ่งที่เกิดร่วมกันโดยมีลำดับ** (Collocation)

สิ่งที่เกิดร่วมกันโดยมีลำดับคือ คำหรือหน้าที่คำที่ติดกับข้อความหรือคำที่จะนำมาพิจารณา สำหรับในงานวิทยานิพนธ์นี้จะนำคำหรือหน้าที่คำก่อนหน้าหรือหลังข้อความที่พิจารณาเพียง 2 คำ ส่วนข้อความหรือคำที่นำมาพิจารณาในที่นี้ จะขึ้นอยู่กับลักษณะของปัญหา สำหรับในที่นี้ปัญหาที่จะนำมาแก้ไข คือ ปัญหาความกำกวม และ ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม โดยวิธีการเลือกข้อความหรือคำที่จะนำมาพิจารณานั้น จะมีลักษณะเหมือนกับที่พิจารณาในคำบริบทดังที่ได้กล่าวไปแล้ว

ตัวอย่างของการนำสิ่งที่เกิดร่วมกันโดยมีลำดับมาใช้ในการแก้ปัญหาความกำกวม ตัวอย่างการแก้ความกำกวมของข้อความ “มากว่า” ซึ่งสามารถตัดคำได้เป็น “มา กว่า” หรือ “มาก ว่า” เช่น

ถ้า มากว่า *ตัวเลข* *CMTR* แล้วให้ตัดเป็น “มา กว่า”

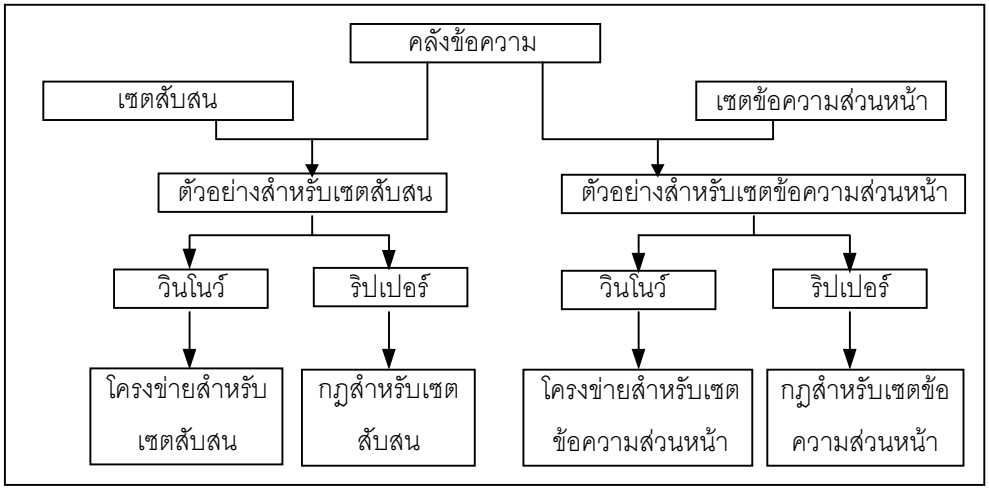
จากตัวอย่างข้างบนสิ่งที่เกิดร่วมกันโดยมีลำดับคือ ตัวเลข และ CMTR จากตัวอย่างข้างต้นหมายความว่า ถ้าพบข้อความ “มากว่า” แล้วตามด้วยตัวเลข และคำถัดไปมีหน้าที่คำเป็น CMTR แล้วให้ตัดคำเป็น “มา กว่า” โดยที่ CMTR หมายถึงหน่วยในการวัด เช่น ปี กิโลกรัม ชั่วโมง เป็นต้น

7.2 **การแก้ไขปัญหาความกำกวม**

เนื่องจากปัญหาความกำกวมนั้นเป็นปัญหาที่สำคัญในการตัดคำ และการแก้ปัญหาความกำกวมนั้นมีแนวทางในการแก้ปัญหาหลายๆ รูปแบบดังที่ได้กล่าวไปแล้วในบทที่ 2 สำหรับในงานวิทยานิพนธ์นี้ได้เสนอวิธีการใหม่ในการแก้ปัญหาความกำกวม โดยจะนำเอาคุณลักษณะแบบต่างๆ มาประยุกต์ใช้ โดยคุณลักษณะที่จะนำมาใช้คือคำบริบทและสิ่งที่เกิดร่วมกันโดยมีลำดับ ตามที่ได้กล่าวไปแล้วในตอนต้น

การแก้ปัญหาความกำกวมโดยใช้คุณลักษณะ จะแบ่งวิธีการแก้ไขปัญหานี้ออกเป็น 2 แบบโดยที่วิธีการทั้ง 2 แบบนี้สามารถจะนำไปใช้ในการแก้ปัญหาได้ ซึ่งในแต่ละแบบนั้นจะมีข้อดีและข้อเสียแตกต่างกัน โดยจะได้ทำการอธิบายในลำดับถัดไป สำหรับการแก้ปัญหาความกำกวมนั้นสามารถแบ่งวิธีการแก้ปัญหาได้ดังต่อไปนี้คือ เซตสับสน (Confusion Set) และเซตข้อความส่วนหน้า (Prefix Set)

สำหรับขั้นตอนการสร้างการเรียนรู้ให้กับวินโนว์และริปเปอร์ในการแก้ปัญหาความกำกวม โดยแบบที่ใช้เซตสับสน หรือเซตข้อความส่วนหน้า ดังแสดงในรูปที่ 7-1 ส่วนรายละเอียดการสร้างเซตสับสน และเซตข้อความส่วนหน้านั้นจะอธิบายในหัวข้อ 7.2.1 และ 7.2.2 ตามลำดับ



รูปที่ 7-1 ขั้นตอนการเรียนรู้คุณลักษณะเพื่อนำมาใช้ในการแก้ปัญหาความกำกวม

7.2.1. **เซตสับสน** (Confusion Set)

ลักษณะของปัญหาความกำกวมในการตัดคำคือ การที่สามารถตัดคำได้หลายๆ แบบสำหรับข้อความหนึ่งๆ ซึ่งจะเป็นสาเหตุทำให้เกิดความสับสนขึ้นว่าแบบไหนจะเป็นแบบที่ถูกต้องที่สุด ดังนั้นในวิทยานิพนธ์นี้จะแก้ปัญหาความกำกวมโดยมีการนำเซตสับสนเข้ามาประยุกต์ใช้ สำหรับเซตสับสนนั้นจะต้องสร้างขึ้นสำหรับทุกๆ ข้อความที่กำกวม ตัวอย่างเช่น ถ้ามีข้อความ “ตากลม” “มากว่า” และ “ขนมอบ” ซึ่งเป็นข้อความที่กำกวม วิธีการนี้จะต้องสร้างเซตสับสนของข้อความเหล่านี้ทั้งหมด

นิยามของเซตสับสนคือ เซตของข้อความที่กำกวม โดยสมาชิกภายในเซตนั้นประกอบไปด้วยข้อความที่ตัดคำที่เป็นไปได้ทุกๆ แบบสำหรับข้อความกำกวมนั้นๆ ตัวอย่างเช่น ข้อความ “มากว่า” เป็นข้อความกำกวม ดังนั้นวิธีการนี้จะต้องสร้างเซตสับสนสำหรับข้อความนี้ โดยที่ข้อความนี้เมื่อทำการตัดคำแล้วสามารถจะตัดได้เป็น “มาก ว่า” กับ “มา กว่า”ดังนั้นเซตสับสนของข้อความ “มากว่า” จะได้ดังแสดงตัวอย่างที่ 7-1

$$C_{\text{มากว่า}} = \{\text{มาก ว่า, มา กว่า }\} \qquad (7\text{-}1)$$

จากตัวอย่าง 7-1 $C_{\text{มากกว่า}}$ คือเซตสับสนของข้อความ “มากกว่า” ซึ่งสมาชิกภายในเซตนี้จะประกอบไปด้วย “มาก ว่า” และ “มา กว่า” ซึ่งจะหมายความว่าข้อความ “มากกว่า” สามารถจะตัดคำได้ทั้งหมด 2 แบบคือ “มาก ว่า” หรือ “มา กว่า”

เมื่อทำการสร้างเซตสับสนสำหรับข้อความกำกวมที่ปรากฏอยู่ในคลังข้อความทั้งหมดเป็นที่เรียบร้อยแล้ว ขั้นตอนต่อไปคือส่งตัวอย่างต่างๆ เข้าไปให้การเรียนรู้ของเครื่อง เพื่อที่จะให้การเรียนรู้ของเครื่องนั้นทำการเลือกคุณลักษณะต่างๆ ที่สำคัญออกมา โดยคุณลักษณะต่างๆ ที่เลือกออกมาได้นั้นสามารถนำมาใช้ในการจำแนกระหว่างสมาชิกภายในเซตนั้นๆ ได้ หรืออีกนัยหนึ่งก็คือสามารถที่จะนำคุณลักษณะต่างๆ เข้ามาใช้ในการระบุว่าข้อความที่กำกวมนี้สามารถจะตัดคำได้เป็นอย่างไร

ตัวอย่างการนำการเรียนรู้ของเครื่องเข้ามาใช้ในการแก้ไขปัญหาความกำกวม จากตัวอย่างการสร้างเซตสับสนของข้อความ “มากกว่า” ซึ่งแสดงในตัวอย่างที่ 7-1 ข้อความนี้สามารถแบ่งคำได้เป็น 2 แบบคือ “มาก ว่า” กับ “มา กว่า” ดังนั้นจะต้องนำประโยคที่มีคำว่า “มาก ว่า” หรือ “มา กว่า” จากคลังข้อความที่ทำการตัดคำและกำกับหน้าที่คำเรียบร้อยแล้ว มาเป็นตัวอย่างของการเรียนรู้ของเครื่องริปเปอร์ หรือวินโนว์ โดยให้ข้อความ “มากว่า” เป็นข้อความที่พิจารณา ส่วนคำรอบๆ ข้อความนี้ให้พิจารณาเป็น คำบริบท หรือ สิ่งที่เกิดร่วมกันโดยมีลำดับ สำหรับแต่ละตัวอย่างที่ส่งเข้าไปต้องระบุด้วยว่าคุณลักษณะที่นำเข้าไปให้ริปเปอร์กับวินโนว์นั้นเป็นคุณลักษณะของ “มาก ว่า” หรือ “มา กว่า”

เมื่อริปเปอร์และวินโนว์ได้ทำการเรียนรู้คุณลักษณะต่างๆ ที่ใช้ในการแก้ปัญหาความกำกวมในรูปแบบเซตสับสนเรียบร้อยแล้ว ดังนั้นเมื่อต้องการตัดคำสำหรับข้อความที่กำกวมที่ได้มีการสร้างเซตสับสน และทำการเรียนรู้ไปแล้ว คำรอบๆ ข้อความที่กำกวมนั้นจะถูกพิจารณาเป็นคำบริบท และสิ่งที่เกิดร่วมกันโดยมีลำดับ และเมื่อทำการส่งให้กับวินโนว์หรือริปเปอร์ วินโนว์หรือริปเปอร์จะพิจารณาจากคำบริบทและสิ่งที่เกิดร่วมกันโดยมีลำดับ โดยเปรียบเทียบกับโครงข่ายในกรณีที่ใช้วินโนว์ หรือกฎในกรณีที่ใช้ริปเปอร์ที่ได้มาจากการเรียนรู้ แล้วจะทำการตัดสินใจเลือกแบบการตัดทำที่ถูกต้อง

7.2.2. **เซตข้อความส่วนหน้า** (Prefix Set)

จากวิธีแรกเป็นการสร้างเซตสับสน ซึ่งทำการสร้างเซตของการตัดคำที่เป็นไปได้ทุกๆ แบบของข้อความที่กำกวม และเมื่อได้เซตสับสนของข้อความที่กำกวมแล้วจะทำการเรียนรู้เพื่อหาคุณลักษณะต่างๆ ที่จะสามารถนำมาจำแนกสมาชิกต่างๆ ภายในเซตได้ ซึ่งวิธีการนี้จะมีข้อจำกัดคือจะต้องมีการสร้างเซตสับสนสำหรับข้อความที่กำกวมที่เป็นไปได้ทั้งหมดก่อน ทำให้วิธีการนี้ไม่สามารถแก้ปัญหาข้อความกำกวมที่เกิดขึ้นมาใหม่ได้ แต่วิธีที่จะนำเสนอต่อไปนี้จะเป็นอีกวิธีหนึ่งที่สามารถยอมให้เกิดความกำกวมขึ้นมาใหม่ได้ ซึ่งวิธีการนี้เป็นการสร้างเซตอีกแบบหนึ่งที่จะนำมาใช้แก้ปัญหาความกำกวมเหมือนกัน โดยที่วิธีนี้จะสร้างเซตจากคำศัพท์ที่ปรากฏในพจนานุกรมทั้งหมด และเรียกเซตชนิดนี้ว่าเซตข้อความส่วนหน้า

นิยามเซตข้อความส่วนหน้า คือเซตที่ประกอบด้วยคำต่างๆ เป็นสมาชิก โดยคำที่มีจำนวนตัวอักษรน้อยกว่า จะเป็นข้อความส่วนหน้า (Prefix) ของคำศัพท์ที่มีจำนวนตัวอักษรมากกว่าที่ปรากฏภายในเซตนั้นๆ เสมอ

การสร้างเซตข้อความส่วนหน้า วิธีการนี้จะสร้างเซตข้อความส่วนหน้าของทุกๆ คำที่มีอยู่ในพจนานุกรม ยกเว้นในกรณีที่มีการสร้างเซตข้อความส่วนหน้าแล้วมีสมาชิกแค่เพียงสมาชิกเดียว ตัวอย่างการสร้างเซตข้อความส่วนหน้า สมมุติว่าคำศัพท์ที่มีอยู่ในพจนานุกรมมีดังต่อไปนี้คือ มา, มาก, มากมาย, ตา, ตาก และ ตาม ดังนั้นวิธีการนี้จะสร้างเซตข้อความส่วนหน้าของทุกๆ คำในพจนานุกรม ดังนั้นจะได้เซตข้อความส่วนหน้าของคำว่า มาก, มากมาย, ตาก และ ตาม โดยจะแสดงตามตัวอย่างที่ 7-2 , 7-3, 7-4 และ 7-5 ตามลำดับ ส่วนเซตข้อความส่วนหน้าของคำว่า มา กับ มาก นั้นไม่ต้องสร้างขึ้นมาเพราะเนื่องจากมีสมาชิกภายในเซตเพียงสมาชิกเดียว

$$P_{\text{มาก}} = \{\text{มา, มาก}\} \tag{7-2}$$

$$P_{\text{มากมาย}} = \{\text{มา, มาก, มากมาย}\} \tag{7-3}$$

$$P_{\text{ตาก}} = \{\text{ตา, ตาก}\} \tag{7-4}$$

$$P_{\text{ตาม}} = \{\text{ตา, ตาม }\} \tag{7-5}$$

จากตัวอย่างที่ 7-2, 7-3, 7-4 และ 7-5 นั้นสัญลักษณ์ P_a นั้นคือเซตข้อความส่วนหน้าของคำ a และในแต่ละตัวอย่างจะแสดงถึงสมาชิกภายในของแต่ละเซตข้อความส่วนหน้า

เมื่อมีการสร้างเซตข้อความส่วนหน้าเรียบร้อยแล้ว ขั้นตอนต่อไปคือการนำตัวอย่างไปให้ริปเปอร์กับวินโนว์เรียนรู้เพื่อที่เลือกคุณลักษณะที่สำคัญออกมา ซึ่งสามารถจะนำมาใช้ในการจำแนกระหว่างสมาชิกภายในเซตนั้น สำหรับวิธีการสร้างตัวอย่างของแต่ละเซตนั้น จะมีขั้นตอนการสร้างดังต่อไปนี้คือ 1. เลือกเซตข้อความส่วนหน้า 2. เลือกตัวอย่างประโยคจากคลังข้อความที่มีการตัดคำและกำกับหน้าที่คำเรียบร้อยแล้ว โดยให้เลือกประโยคที่มีคำที่เป็นสมาชิกภายในเซตนั้นๆ ขึ้นมา 3. ให้ส่งคำบริบท และสิ่งที่เกิดร่วมกันโดยมีลำดับของคำที่เป็นสมาชิกภายในเซตนั้นๆ แล้วต้องระบุด้วยว่าเป็นคำบริบทและสิ่งที่เกิดร่วมกันโดยมีลำดับเป็นของคำใด เมื่อริปเปอร์หรือวินโนว์ทำการเรียนรู้ของแต่ละเซตแล้ว จะได้คุณลักษณะในรูปแบบกฎสำหรับริปเปอร์ หรือ โครงข่ายสำหรับวินโนว์ ที่สามารถจะใช้จำแนกระหว่างสมาชิกภายในเซตข้อความส่วนหน้าได้

เมื่อมีการสร้างเซตข้อความส่วนหน้าและมีการสร้างการเรียนรู้ให้กับริปเปอร์หรือวินโนว์สำหรับแต่ละเซตข้อความส่วนหน้าแล้ว ขั้นตอนต่อไปจะแสดงการนำเซตข้อความส่วนหน้าเข้ามาใช้ในการแก้ปัญหาความกำกวม ตัวอย่างเช่นในกรณีที่พบข้อความว่า “มากมาย” ซึ่งเป็นข้อความที่กำกวม สำหรับวิธีการแก้ปัญหานี้คือการนำเซตข้อความส่วนหน้าของคำว่า “มากมาย” มาใช้ แล้วทำการส่งคำบริบท และการเกิดร่วมกันโดยมีลำดับให้กับวินโนว์หรือริปเปอร์ วินโนว์หรือริปเปอร์นั้นจะทำการเปรียบเทียบระหว่างคุณลักษณะที่ส่งเข้ามากับคุณลักษณะต่างๆ ที่ได้เคยเรียนรู้ไว้ และจะทำการตัดสินใจออกมาว่าควรจะตัดคำเป็นอย่างไร

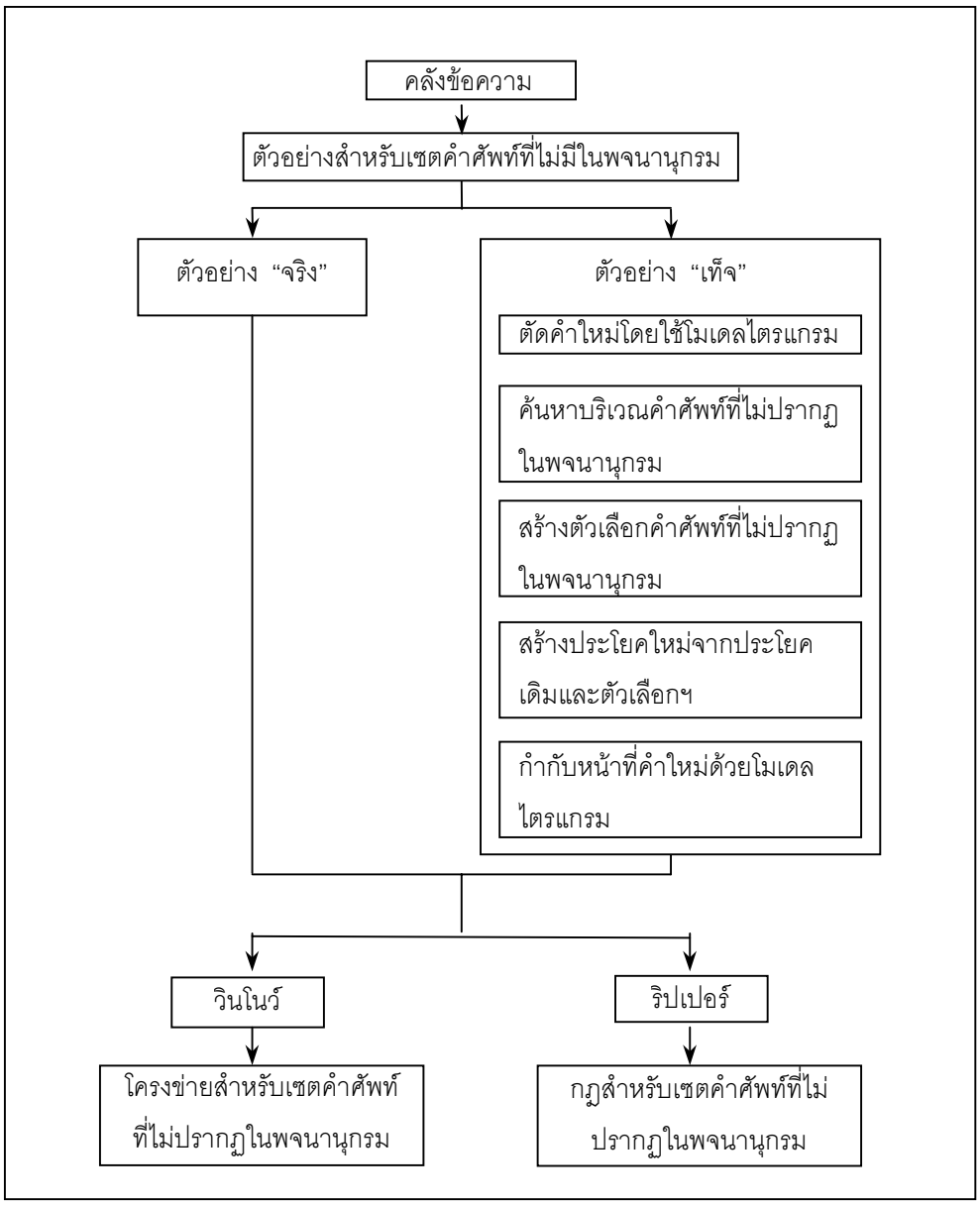
ข้อดีสำหรับวิธีการนี้คือ จะสามารถแก้ปัญหาของข้อความกำกวมที่ไม่เคยพบมาได้ เช่น สมมุติว่าพบข้อความ “มากรอง” เป็นข้อความกำกวมที่ไม่เคยพบมาก่อน ซึ่งวิธีนี้จะมีการสร้างเซตข้อความส่วนหน้าของคำว่า “มาก” (แสดงในตัวอย่างที่ 7-2) ดังนั้นวิธีการนี้จะนำเซตข้อความส่วนหน้าของคำว่า “มาก” เข้ามาใช้ในการแก้ปัญหานี้ ซึ่งเมื่อนำเซตข้อความส่วนหน้าของคำว่า “มาก” เข้ามาใช้ก็จะสามารถระบุได้ว่าข้อความนี้ควรจะตัดคำแรกให้เป็น “มา” หรือ “มาก” โดยที่ผลลัพธ์ที่ได้อาจจะเป็น “มากรอง” หรือ “มาก รอง” ซึ่งจะขึ้นอยู่กับคุณลักษณะต่างๆ ของข้อความนี้ แต่ถ้าใช้เซตสับสนแก้ปัญหาความกำกวมนี้ จะต้องมีการสร้างเซตสับสนของข้อความ “มากรอง” ก่อนถึงจะแก้ปัญหานี้ได้

7.3 การแก้ไขปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

จากบทที่ 5 ได้มีการอธิบายถึงลักษณะต่างๆ ของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมไปแล้ว ซึ่งจะเห็นได้ว่าคำศัพท์ประเภทนี้สามารถเกิดขึ้นได้หลายรูปแบบ และไม่มีกฎเกณฑ์ที่แน่นอน ทำให้การหาขอบเขตของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนี้เป็นไปได้ยาก ในวิทยานิพนธ์นี้จะนำเอาคุณลักษณะ เข้ามาใช้ในการหาขอบเขตของคำศัพท์ที่ไม่ปรากฏในพจนานุกรม ซึ่งคุณลักษณะต่างๆ เหล่านั้นได้มาจากการเรียนรู้ของเครื่อง ริปเปอร์ และ วินโนว์

จากวิธีการแก้ปัญหาความกำกวมนั้นจะทำการสร้างเซตสับสนหรือเซตข้อความส่วนหน้า หลังจากนั้นนำคำบริบทและสิ่งที่เกิดร่วมกันโดยมีลำดับของข้อความที่กำกวมนั้น มาส่งให้กับวินโนว์หรือริปเปอร์ทำการตรวจสอบและตัดสินว่าการตัดคำที่ถูกควรจะเป็นเช่นไร สำหรับการแก้ปัญหาเรื่องคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้นจะมีการนำตัวอย่างของคำศัพท์ที่ไม่ปรากฏในพจานุกรมที่เป็นจริง และตัวอย่างที่เป็นเท็จเข้ามาประยุกต์ใช้ในการแก้ปัญหานี้

สำหรับขั้นตอนการสร้างการเรียนรู้ให้กับวินโนว์และริปเปอร์ในการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมโดยใช้ตัวอย่างจริงกับตัวอย่างเท็จแสดงในรูปที่ 7-2 โดยที่ขั้นตอนต่างๆ จะประกอบด้วยขั้นตอนดังต่อไปนี้



รูปที่ 7-2 ขั้นตอนการเรียนรู้คุณลักษณะเพื่อนำมาใช้ในการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

- เลือกประโยคที่มีชื่อเฉพาะจากคลังข้อความมาเป็นตัวอย่าง
- สำหรับการเรียนรู้คุณลักษณะของตัวอย่างจริงนั้น ให้นำคำบริบทและสิ่งที่เกิดร่วมกันโดยมีลำดับรอบๆ ชื่อเฉพาะมาเป็นตัวอย่างให้กับการเรียนรู้เครื่องริปเปอร์และวินโนว์

- สำหรับการเรียนรู้คุณลักษณะของตัวอย่างเท็จ ขั้นแรกต้องมีการสร้างตัวเลือกของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมที่มีขอบเขตไม่ถูกต้อง แล้วจึงนำคำบริบทและสิ่งที่เกิดร่วมกันโดยมีลำดับรอบๆ ตัวเลือกที่สร้างขึ้นมาเป็นตัวอย่างให้กับการเรียนรู้ของเครื่องริปเปอร์และวินโนว์ โดยขั้นตอนการสร้างจะประกอบไปด้วยดังนี้
    - นำประโยคมาทำการตัดคำใหม่ โดยใช้ไตรแกรมโมเดล ตามที่ได้กล่าวไปแล้วในหัวข้อ 2.3.2
    - ทำการค้นหาบริเวณที่น่าจะเกิดคำที่ไม่ปรากฏในพจนานุกรม
    - สร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมที่มีขอบเขตคำ ไม่ถูกต้อง
    - สร้างประโยคใหม่จากประโยคเดิมที่ได้มีการเปลี่ยนชื่อเฉพาะให้เป็นตัวเลือกอื่นๆ ที่สร้างขึ้นมา
    - กำกับหน้าที่คำโดยใช้ไตรแกรมโมเดล สำหรับประโยคใหม่ที่ได้สร้างขึ้น ส่วนรายละเอียดนั้นได้กล่าวไว้ในบทที่ 3

สำหรับรายละเอียดของขั้นตอนต่างๆ ในการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้น จะกล่าวถึงในส่วนถัดไป

7.3.1 **การสร้างตัวอย่างจริงกับตัวอย่างเท็จ**

สำหรับการแก้ปัญหาเรื่องคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้น จะสร้างตัวอย่างจริงและตัวอย่างเท็จเพื่อนำไปให้การเรียนรู้ของเครื่องริปเปอร์หรือวินโนว์เรียนรู้คุณลักษณะต่างๆ ที่สามารถนำมาใช้ในการจำแนกระหว่างตัวอย่างจริงและตัวอย่างเท็จได้ ส่วนสาเหตุที่ต้องการให้ริปเปอร์หรือวินโนว์เรียนรู้คุณลักษณะในการจำแนกระหว่างตัวอย่างจริงกับตัวอย่างเท็จนั้น เนื่องจากคำศัพท์ที่ไม่ปรากฏในพจานานุกรมนั้นสามารถจะเกิดจากการประสมระหว่างคำได้หลายแบบ ดังนั้นในงานวิทยานิพนธ์นี้จะทำการค้นหาจุดที่น่าสงสัยที่มีโอกาสจะเกิดคำศัพท์ที่ไม่ปรากฏในพจานุกรมขึ้น แล้วจะทำการสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมขึ้นโดยการประสมระหว่างจุดที่น่าสงสัยกับคำบริเวณรอบๆ ในหลายรูปแบบ แล้วค่อยนำบริบทตัวเลือกนั้นมาป้อนให้กับริปเปอร์หรือวินโนว์ เพื่อค้นหาตัวเลือกที่มีโอกาสเป็นคำศัพท์ที่ไม่ปรากฏในพจนานุกรมมากที่สุด สำหรับการค้นหาจุดที่น่าสงสัยนั้นจะอธิบายในหัวข้อ 7.3.2

ในการสร้างตัวอย่างจริงให้กับการเรียนรู้เครื่องริปเปอร์หรือวินโนว์ คือนำบริบทของชื่อเฉพาะมาเป็นตัวอย่างจริง เนื่องจากวิทยานิพนธ์นี้จะทำการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมที่เป็นชื่อเฉพาะเท่านั้น ดังนั้นจึงพิจารณาให้ชื่อเฉพาะนั้น ส่วนตัวอย่างเท็จนั้นก็จะนำมาบริบทของตัวเลือกที่ไม่ปรากฏพจนานุกรมที่สร้างขึ้นมาจากชื่อเฉพาะเหล่านั้น ส่วนในรายละเอียดของการสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้นแสดงในหัวข้อ 7.3.3

7.3.2. การค้นหาบริเวณที่น่าสงสัย

เนื่องจากการเกิดคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้น มีโอกาสจะเกิดขึ้นค่อนข้างมาก โดยเฉพาะชื่อเฉพาะต่างๆ และคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้นสามารถมีได้หลายรูปแบบดังที่ได้กล่าวไปแล้วในบทที่ 5 โดยเฉพาะคำศัพท์ที่ไม่ปรากฏในพจนานุกรมแบบซ่อนเร้นทุกส่วนนั้นค่อนข้างจะตรวจสอบยากว่าบริเวณนั้นเป็นคำศัพท์ที่ไม่ปรากฏในพจนานุกรม ดังนั้นในวิทยานิพธ์นี้จะเสนอวิธีการตรวจสอบหาบริเวณที่มีโอกาสเกิดคำที่ไม่ปรากฏในพจนานุกรม ตามลักษณะประเภทของคำที่ไม่ปรากฏในพจนานุกรมดังต่อไปนี้

- บริเวณที่น่าสงสัยสำหรับคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างชัดเจนและอย่างซ่อนเร้นบางส่วน เนื่องจากลักษณะของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมประเภทนี้จะมีข้อความที่ไม่มีในพจนานุกรมเป็นส่วนประกอบ ดังนั้นการค้นหาบริเวณที่น่าจะเกิดคำศัพท์ที่ไม่ปรากฏในพจนานุกรมประเภทดังกล่าวทำได้โดยการค้นหาบริเวณที่ทำการตัดคำแล้วเกิดข้อความที่ไม่มีในพจนานุกรม

- บริเวณที่น่าสงสัยสำหรับคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วน เนื่องจากลักษณะของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมประเภทนี้ เกิดจากการประสมระหว่างคำศัพท์ที่ปรากฏในพจนานุกรม ทำให้การหาบริเวณที่น่าสงสัยสำหรับคำศัพท์ประเภทนี้ทำได้ยากกว่าแบบแรก ดังนั้นในวิทยานิพนธ์นี้จึงเสนอวิธีการค้นหาบริเวณที่น่าสงสัยสำหรับคำศัพท์ประเภทนี้ โดยมีวิธีการดังต่อไปนี้

1. หาค่าความน่าจะเป็นของ $P(w_i|t_i)$ ที่มีค่าน้อยกว่าค่าขีดเริ่มเปลี่ยน (Threshold)
2. หาค่าความน่าจะเป็นของ $P(t_i|t_{i-1},t_{i-2})$ ที่มีค่าน้อยกว่าค่าขีดเริ่มเปลี่ยน

เมื่อพบบริเวณที่น่าสงสัยว่าจะเกิดคำศัพท์ที่ไม่ปรากฏในพจนานุกรมแล้ว ขั้นตอนต่อไปก็คือการสร้างตัวเลือกของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมซึ่งจะอธิบายในส่วนถัดไป

7.3.3 การสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรม (Generating Unknown Word Candidate)

เมื่อพบจุดที่น่าส่งสัยในการเกิดคำที่ไม่ปรากฏในพจนานุกรมตามวิธีการที่ได้กล่าวไปแล้วนั้น ขั้นตอนต่อไปคือ การสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรม โดยสาเหตุที่ต้องมีการสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรม เนื่องจากงานวิทยานิพนธ์นี้จะต้องหาขอบเขตของคำที่ไม่ปรากฏในพจนานุกรม โดยที่คำศัพท์ประเภทนี้สามารถจะเกิดขึ้นได้หลายรูปแบบ และไม่มีกฎเกณฑ์แน่นอนในการหาขอบเขตของคำศัพท์ที่ไม่ปรากฏในพจนานุกรม ดังนั้นในขั้นตอนนี้จะทำการสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมที่เป็นไปได้ทั้งหมดก่อน หลังจากนั้นจะนำคุณลักษณะที่ได้จากการเรียนรู้ของริปเปอร์หรือวินโนว์ มาประยุกต์ใช้ในการเลือกตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมที่ดีที่สุด

สำหรับวิธีการสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้น สามารถแบ่งออกได้เป็น 2 วิธี ตามประเภทของคำศัพท์ที่ไม่ปรากฏในพจนานุกรม แสดงดังต่อไปนี้

➢ การสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมสำหรับคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างชัดเจนและอย่างซ่อนเร้นบางส่วน แสดงดังรูปที่ 7-3 (ค่า K ที่ใช้ในการทดลองเท่ากับ 4)

กำหนดให้ประโยคคือ $w_1 w_2 \ldots w_a U w_b \ldots w_n$ โดยที่ $w_i \in$พจนานุกรม $U \notin$ พจนานุกรม และ n คือจำนวนคำในประโยค

$UNK = \{\alpha U \beta \mid \alpha \in A, \beta \in B\}$ โดยที่ UNK คือเซตของตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

$A = \{w_{a-i,a-1}, i \in [0,\mathrm{K}]\} \cup \{\varepsilon\}$

$B = \{w_{b+1,b+i}, i \in [0,\mathrm{K}]\} \cup \{\varepsilon\}$

$w_{i,j} = w_{\mathrm{i}} .. w_{\mathrm{j}} : i<j$

ε คือข้อความที่ว่าง (Null string) และ K คือค่าคงที่

รูปที่ 7-3 สมการการสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างชัดเจนและอย่างซ่อนเร้นบางส่วน

➢ การสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมสำหรับคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วน ดังแสดงในรูปที่ 7-4 (ค่า K ที่ใช้ในการทดลองเท่ากับ 4)

7.3.4 การสร้างประโยคใหม่

หลังจากการค้นหาบริเวณที่น่าสงสัยว่าจะเกิดคำศัพท์ที่ไม่ปรากฏในพจนานุกรม และได้ทำการสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมแล้ว ขั้นตอนต่อไปก็คือการสร้างประโยคใหม่จากประโยคเดิมโดยให้นำตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมที่สร้างขึ้นมา มาทำการแทนที่บริเวณที่น่าจะเกิดคำศัพท์ที่ไม่ปรากฏในพจนานุกรม แล้วค่อยนำไปกำกับหน้าที่คำโดยใช้โมเดลไตรแกรม สุดท้ายทำการส่งไปให้การเรียนรู้ของเครื่องวินโนว์หรือริปเปอร์

ตัวอย่างเช่นประโยค นางเจนนีไปเดินเล่น เมื่อทำการตัดคำโดยใช้โมเดลไตรแกรมจะได้เป็น “นาง/NTTL เจน/VACT **นี**/NPRP ไป/XVAE เดิน/VACT เล่น/VACT” โดยหน้าที่คำ (NTTL, VACT ฯลฯ)สามารถดูรายละเอียดได้ในภาคผนวก ข. สำหรับผลลัพธ์จากการตัดคำจะเห็นว่า “นี” เป็นข้อความที่ไม่

กำหนดให้ประโยคคือ $w_1 w_2 \ldots w_a \ldots w_n$ โดยที่ $w_i \in$ พจนานุกรม w_a คือคำที่มีความน่าจะเป็นต่ำกว่าค่าขีดจำกัดของค่า $P(w_i|t_i)$ หรือ $P(t_i|t_{i-1},t_{i-2})$ ตามที่ได้อธิบายใน 7.3.2.2 และ n คือจำนวนคำในประโยค

$UNK = \{\alpha W \beta \mid \alpha \in A, \beta \in B\}$ โดยที่ UNK คือเซตของตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

$A = \{w_{a-i,a-1},\ i \in [0,\mathrm{K}]\} \cup \{\varepsilon\}$

$B = \{w_{a+1,a+i},\ i \in [0,\mathrm{K}]\} \cup \{\varepsilon\}$

$w_{i,j} = w_i..w_j : i<j$

$W = w_a$ ถ้า $P(w_i|t_i) <$ ค่าขีดจำกัด หรือ

$W \in \{w_a, w_{a-1}, w_{a-2}\}$ ถ้า $P(t_i|t_{i-1},t_{i-2}) <$ ค่าขีดจำกัด

ε คือข้อความที่ว่าง (Null string) และ K คือค่าคงที่

รูปที่ 7-4 สมการการสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วน

เกิดขึ้นในพจนานุกรม ดังนั้นบริเวณนี้จะต้องเกิดคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างแน่นอน ทำให้ต้องมีการสร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรม ดังนั้นเมื่อใช้สมการตามรูปที่ 7-3 โดยค่า K ที่ใช้มีค่าเท่า 2 ดังนั้นเซตตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรมจะได้ดังนี้ {นี, นีไป, นีไปเดิน, เจนนี, เจนนีไป, เจนนีไปเดิน, นางเจนนี, นางเจนนีไป} ดังนั้นประโยคใหม่ที่ได้จากการแทนที่ตัวเลือกลงไปในประโยคเดิม ก็จะได้ดังต่อไปนี้

1. นาง เจ นี ไป เดิน เล่น
2. นาง เจ นีไป เดิน เล่น
3. นาง เจ นีไปเดิน เล่น
4. นาง เจนนี ไป เดิน เล่น
5. นาง เจนนีไป เดิน เล่น
6. นาง เจนนีไปเดิน เล่น
7. นางเจนนี ไป เดิน เล่น
8. นางเจนนีไป เดิน เล่น

เมื่อได้ประโยคใหม่ดังที่ได้แสดงไปแล้ว หลังจากนั้นให้นำแต่ละแบบของการตัดคำส่งไปให้กำกับหน้าที่คำโดยใช้โมเดลไตรแกรม แล้วจึงนำส่งไปให้ริปเปอร์หรือวินโนว์ในการเลือกประโยคที่เหมาะสมที่สุด

บทที่ 8

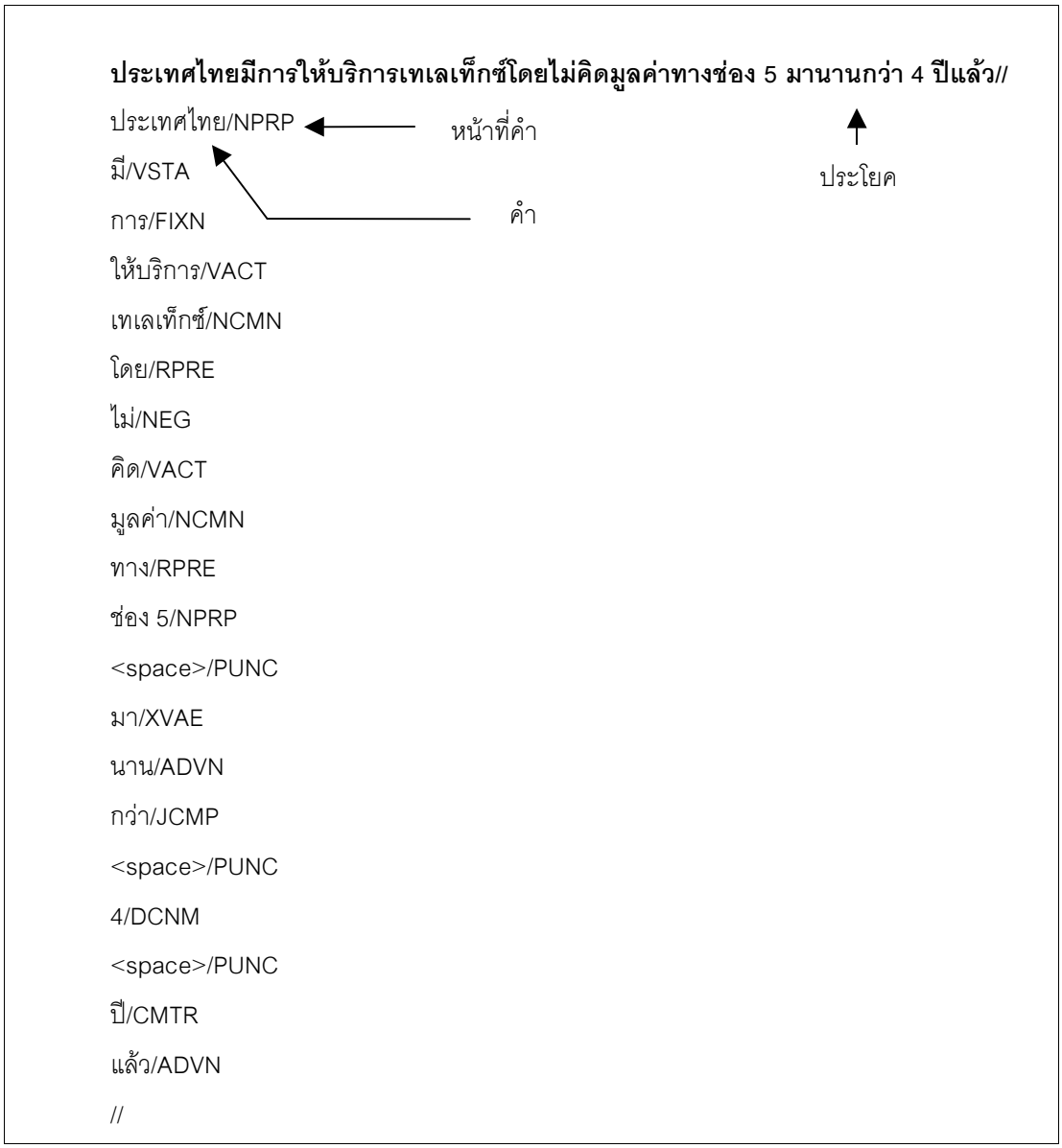
ประสิทธิภาพการตัดคำโดยใช้คุณลักษณะ

จากบทที่แล้วได้มีการอธิบายขั้นตอนการนำเอาการเรียนรู้ของเครื่องในแบบต่างๆ เข้ามาใช้ในการเรียนรู้คุณลักษณะต่างๆ ที่สามารถนำมาใช้แก้ปัญหาความกำกวม และคำศัพท์ที่ไม่ปรากฏในพจนานุกรม ในบทนี้จะแสดงถึงผลการทดลองเปรียบเทียบระหว่างการเรียนรู้ของเครื่องริปเปอร์, วินโนว์ และมีการเปรียบเทียบการแก้ปัญหาความกำกวมระหว่างการใช้คุณลักษณะกับวิธีการต่างๆ ที่ผ่านมา

คลังข้อความที่นำมาใช้ในการเรียนรู้ของเครื่องได้นำมาจาก คลังข้อความออร์คิด (Orchid Corpus) (Virach Sorlertlamvanich et al., 1997) โดยได้รับความอนุเคราะห์จาก ห้องปฏิบัติการวิจัยและพัฒนาวิศวกรรมภาษาและซอฟต์แวร์ (Software and Language Engineering Laboratory: SLL) ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (National Electronics and Computer Technology Center: NECTEC) โดยลักษณะของบทความที่นำมาใช้สร้างคลังข้อความนั้นได้นำมาจากรายงานการประชุมวิชาการของศูนย์ฯ เอง และจำนวนคลังข้อความนั้นมีอยู่ประมาณ 25,000 ประโยค โดยที่ภายในคลังข้อความนี้ได้ทำการแบ่งเป็นประโยค ส่วนภายในประโยคจะแบ่งเป็นคำต่างๆ และยังได้มีการกำหนดหน้าที่คำด้วย ซึ่งทั้งหมดทำโดยนักภาษาศาสตร์ ตัวอย่างประโยคภายในคลังข้อความออร์คิด แสดงในรูปที่ 8-1

จากรูปที่ 8-1 จะแสดงตัวอย่างประโยค "ประเทศไทยมีการให้บริการเทเลเท็กซ์โดยไม่คิดมูลค่าทางช่อง 5 มานานกว่า 4 ปีแล้ว" ซึ่งประโยคนี้ได้ทำการตัดคำและกำกับหน้าที่ของคำเรียบร้อยแล้ว โดยภายในหนึ่งบรรทัดจะประกอบไปด้วย 1 คำและจะบอกด้วยว่าคำนี้ทำหน้าที่อะไรอยู่ภายในประโยคนี้ ตัวอย่างเช่น "ประเทศไทย/NPRP" ในบรรทัดที่ 2 จะบอกว่าคำว่า "ประเทศไทย" มีหน้าที่คำเป็น NPRP ซึ่ง NPRP เป็นตัวอักษรย่อที่ใช้บอกหน้าที่คำที่เป็นคำนามประเภทชื่อเฉพาะ โดยสัญลักษณ์ตัวย่อต่างๆ เหล่านี้ สามารถดูได้ในภาคผนวก ข

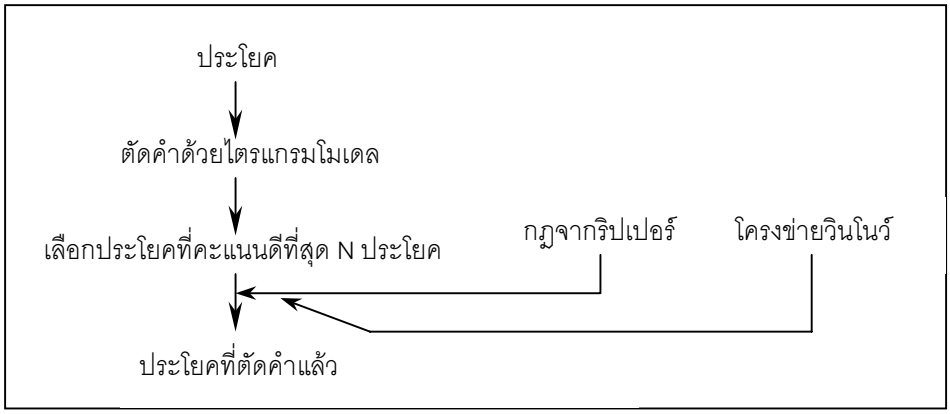
หน้าที่คำที่นำมาใช้ในคลังข้อความนี้ ได้ถูกแบ่งออกเป็น 47 หมวด (Virach Sorlertlamvanich et al., 1997) ซึ่งทำการแบ่งโดยนักภาษาศาสตร์ การที่ต้องแบ่งหน้าที่คำให้ละเอียดลงไปนั้น เนื่องจากถ้าแบ่งหน้าที่คำโดยแค่แบ่งเป็น คำนาม คำกริยา คุณศัพท์ ฯลฯ เท่านั้นจะไม่เพียงพอต่อการนำมาใช้ในการวิเคราะห์ทางภาษา จึงทำให้นักภาษาศาสตร์ได้มีการวิเคราะห์และแบ่งหน้าที่ของคำออกมาเป็นหมวดหมู่ต่างๆ



รูป 8-1 ตัวอย่างประโยคที่ทำการตัดคำและกำกับหน้าที่ของคำภายในคลังข้อความออร์คิด

สำหรับการทดลองวัดประสิทธิภาพของการเรียนรู้ของเครื่องริปเปอร์กับวินโนว์ เพื่อที่จะนำมาใช้ในการเรียนรู้คุณลักษณะต่างๆ ในการแก้ไขปัญหาความกำกวม ในการทดลองนี้ได้เลือกข้อความที่กำกวมที่เกิดขึ้นบ่อยจากคลังข้อความออร์คิด แล้วทำการสร้างเซตสับสนและเซตข้อความส่วนหน้าสำหรับข้อความกำกวมที่เลือกขึ้นมา จากนั้นนำประโยคต่างๆ จากคลังข้อความออร์คิดมาใช้ในการเรียนรู้และทดสอบ โดยแบ่งประโยคที่นำมาใช้นั้นออกเป็น 2 ส่วน ส่วนแรกจะเป็นชุดสอน (Training Set) จำนวน 80% เพื่อที่จะนำไปให้ ริปเปอร์กับวินโนว์ ใช้ในการเรียนรู้ ส่วนที่สองจะเป็นชุดทดสอบ (Test Set) จำนวน 20% เพื่อใช้ทดสอบดูประสิทธิภาพของการเรียนรู้ของ ริปเปอร์ และวินโนว์ เมื่อนำมาใช้กับข้อมูลที่ไม่ได้มีการนำไปใช้ในการสร้างกฎ

8.1 ขั้นตอนการนำคุณลักษณะเข้ามาใช้ในการแก้ปัญหาความกำกวม



รูปที่ 8-1 ขั้นตอนการแก้ไขปัญหาความกำกวมโดยใช้คุณลักษณะ

ขั้นตอนการตัดคำโดยใช้คุณลักษณะมาแก้ไขปัญหาความกำกวม ซึ่งขั้นตอนมีดังต่อไปนี้คือ (แสดงดังรูปที่ 8-1)

1. นำประโยคมาตัดคำโดยใช้ไตรแกรมโมเดล
2. เลือกประโยคที่คะแนนดีที่สุดจำนวน N ประโยค
3. นำกฎจากริปเปอร์ หรือโครงข่ายวินโนว์มาช่วยใช้ในการแก้ปัญหา

8.2 ผลการทดลองแก้ปัญหาความกำกวม

สำหรับตัวอย่างข้อความกำกวมต่างๆ ที่นำมาใช้ในการทดลองนั้นจะนำมาจากคลังข้อความออร์คิด โดยจะเลือกข้อความกำกวมที่เกิดขึ้นจำนวนมากในคลังข้อความ สำหรับรายละเอียดความถี่ของข้อความกำกวมต่างๆ สามารถแสดงดังในภาคผนวก ค.

การทดลองแก้ปัญหาความกำกวมนั้นได้แบ่งการทดลองออกเป็นดังนี้

1. การทดลองการแก้ไขปัญหาความกำกวม แบบที่ต้องใช้บริบท โดยใช้การแก้ปัญหาแบบเซตสับสน ซึ่งผลการทดลองแสดงในตารางที่ 8-1

2. การทดลองการแก้ไขปัญหาความกำกวม แบบที่ต้องใช้บริบท โดยใช้การแก้ปัญหาแบบเซตข้อความส่วนหน้า ซึ่งผลการทดลองแสดงในตารางที่ 8-2

3. การทดลองการแก้ไขปัญหาความกำกวม แบบที่ไม่ต้องใช้บริบท โดยใช้การแก้ปัญหาแบบเซตสับสน ซึ่งผลการทดลองแสดงในตารางที่ 8-3

4. การทดลองการแก้ไขปัญหาความกำกวมแบบที่ไม่ต้องใช้บริบท โดยใช้การแก้ปัญหาแบบเซตข้อความส่วนหน้า ซึ่งผลการทดลองแสดงในตารางที่ 8-4

8.3 สรุปผลการทดลองการแก้ปัญหาความกำกวม

จากการแก้ปัญหาความกำกวมตามตารางที่ 8-1, 8-2, 8-3 และ 8-4 แสดงให้เห็นว่าการนำคุณลักษณะเข้ามาใช้ในการแก้ปัญหาความกำกวมไม่ว่าจะเป็นความกำกวมแบบที่ต้องใช้บริบทหรือไม่ต้องใช้บริบทนั้นสามารถนำมาแก้ปัญหาได้ดีกว่าวิธีการเดิมคือวิธีการตัดคำโดยใช้โมเดลไตรแกรม และการตัดคำโดยเลือกแบบเหมือนมากที่สุด และคุณลักษณะที่ได้จากการเรียนรู้ของเครื่องวินโนว์จะให้ความถูกต้องมากกว่าริปเปอร์ เมื่อนำมาใช้ในการแก้ปัญหาความกำกวมทั้งสองประเภท ไม่ว่าจะเป็นการใช้เซตสับสนหรือเซตข้อความส่วนหน้า

การแก้ปัญหาสำหรับความกำกวมที่ต้องใช้บริบทนั้นจะมีความถูกต้องน้อยกว่าความกำกวมที่ไม่ต้องใช้บริบท โดยนำเซตสับสนมาประยุกต์ใช้ในการแก้ปัญหาความกำกวมทั้งสองแบบจะให้ผลความถูกต้องมากกว่าการนำเซตข้อความส่วนหน้ามาใช้

ดังนั้นจากผลการทดลองสามารถสรุปผลได้ว่าการนำคุณลักษณะเข้ามาใช้ในการแก้ปัญหาความกำกวมนั้นสามารถจะแก้ปัญหาได้ดีกว่าวิธีการตัดคำโดยใช้โมเดลไตแกรม และการตัดคำโดยเลือกแบบเหมือนมากที่สุด และวินโนว์มีประสิทธิภาพในการแก้ปัญหาความกำกวมได้ดีกว่าริปเปอร์

ตารางที่ 8-1 ตารางแสดงประสิทธิภาพการแก้ไขปัญหาความกำกวม แบบที่ต้องใช้บริบท (Context Dependent) โดยใช้การแก้ปัญหาแบบเซตสับสน

ข้อความที่กำกวม	เซตสับสน (Confusion Set)	วินโนว์		ริปเปอร์		โมเดลไตรแกรม	แบบเหมือนมากที่สุด
		ชุดสอน	ชุดทดสอบ	ชุดสอน	ชุดทดสอบ		
จัดการ	{ จัดการ , จัด การ }	100.00 %	98.03 %	99.61 %	92.06 %	84.55 %	91.82 %
หรือไม่	{ หรือไม่ , หรือ ไม่ }	100.00 %	92.07 %	96.20 %	79.26 %	83.54 %	68.35 %
ให้การ	{ ให้การ , ให้ การ }	100.00 %	94.37 %	95.54 %	89.29 %	64.29 %	10.71 %
ที่อยู่	{ ที่อยู่ , ที่ อยู่ }	100.00 %	90.09 %	95.12 %	84.21 %	74.04 %	18.27 %
พัฒนาการ	{ พัฒนาการ , พัฒนา การ }	100.00 %	93.10 %	95.45 %	69.75 %	67.47 %	18.07 %
ที่เกิด	{ ที่เกิด , ที่ เกิด }	100.00 %	99.71 %	94.29 %	93.75 %	46.59 %	6.82 %
ทางการ	{ ทางการ , ทาง การ }	100.00 %	84.06 %	98.15 %	69.23 %	43.48 %	30.43 %
คุ้มค่า	{ คุ้มค่า , คุ้ม ค่า }	100.00 %	95.36 %	93.75 %	100.00 %	95.00 %	95.00 %
ที่ตั้ง	{ ที่ตั้ง , ที่ ตั้ง }	100.00 %	96.38 %	100.00 %	80.00 %	53.85 %	42.31 %

ตารางที่ 8-2 ตารางแสดงประสิทธิภาพการแก้ไขปัญหาความกำกวม แบบที่ต้องใช้บริบท (Context Dependent) โดยใช้การแก้ปัญหาแบบเซตข้อความส่วนหน้า

ข้อความที่กำกวม	เซตข้อความส่วนหน้า (Prefix Set)	วินโนว์		ริปเปอร์		โมเดลไตรแกรม	แบบเหมือนมากที่สุด
		ชุดสอน	ชุดทดสอบ	ชุดสอน	ชุดทดสอบ		
จัดการ	{ จัด , จัดการ }	100.00 %	94.52 %	83.33 %	69.28 %	84.55 %	91.82 %
หรือไม่	{ หรือ , หรือไม่ }	100.00 %	93.27 %	99.58 %	72.62 %	83.54 %	68.35 %
ให้การ	{ ให้ , ให้การ }	100.00 %	91.71 %	0.00 %	87.45 %	64.29 %	10.71 %
ที่อยู่	{ ที่ , ที่อยู่ }	100.00 %	90.97 %	99.97 %	56.98 %	74.04 %	18.27 %
พัฒนาการ	{ พัฒนา , พัฒนาการ }	100.00 %	89.11 %	99.83 %	65.94 %	67.47 %	18.07 %
ที่เกิด	{ ที่ , ที่เกิด }	100.00 %	91.41 %	0.00 %	75.01 %	46.59 %	6.82 %
ทางการ	{ ทาง , ทางการ }	100.00 %	89.68 %	99.80 %	89.19 %	43.48 %	30.43 %
คุ้มค่า	{ คุ้ม , คุ้มค่า }	100.00 %	93.03 %	90.62 %	94.76 %	95.00 %	95.00 %
ที่ตั้ง	{ ที่ , ที่ตั้ง }	100.00 %	88.79 %	91.04 %	99.98 %	83.26 %	42.31 %

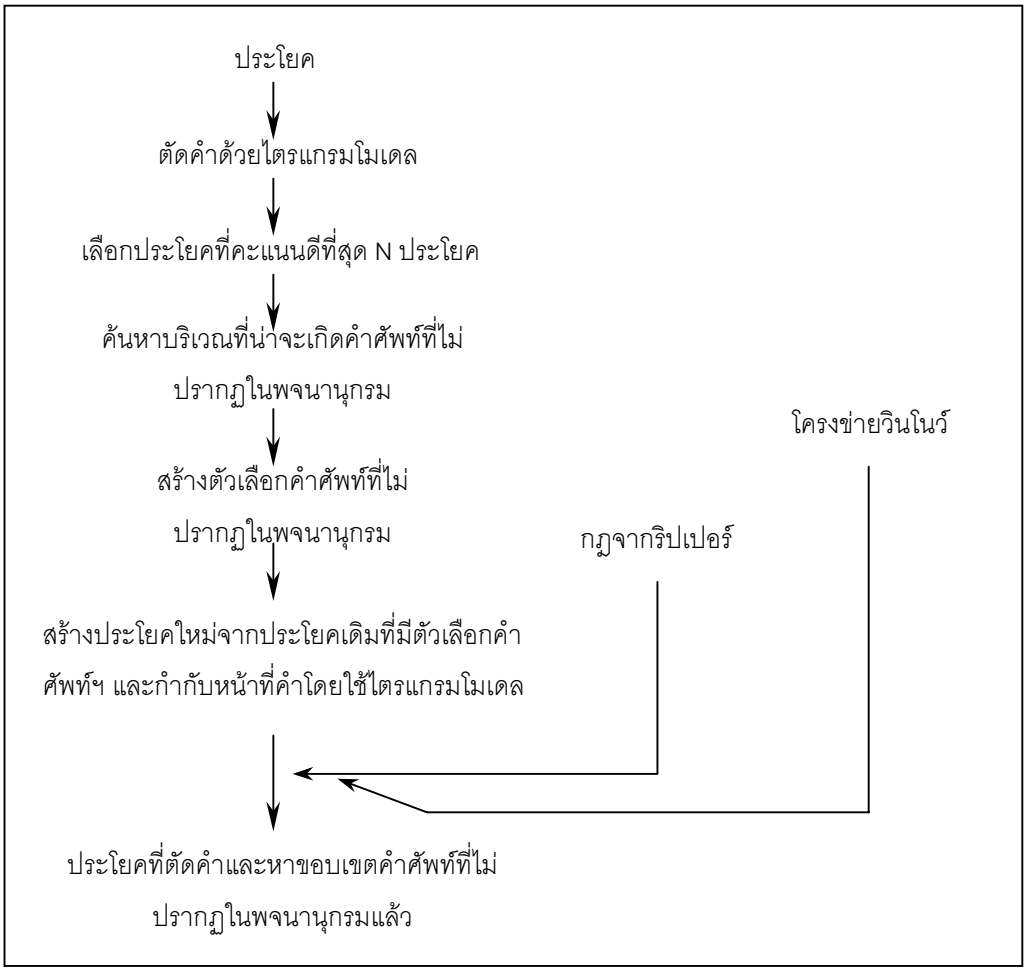
ตารางที่ 8-3 ตารางแสดงประสิทธิภาพการแก้ไขปัญหาความกำกวม แบบที่ไม่ต้องใช้บริบท (Context Independent)
โดยใช้การแก้ปัญหาแบบเซตสับสน

ข้อความที่กำกวม	เซตสับสน (Confusion Set)	วินโนว์		ริปเปอร์		โมเดลไตรแกรม	แบบเหมือนมากที่สุด
		ชุดสอน	ชุดทดสอบ	ชุดสอน	ชุดทดสอบ		
ข้อมูล	{ ข้อมูล , ข้อ มูล }	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
อบรม	{ อบรม , อบ รม }	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
หน้าที่	{ หน้าที่ , หน้า ที่ }	100.00 %	97.54 %	100.00 %	100.00 %	95.84 %	50.14 %
คุณภาพ	{ คุณภาพ , คุณ ภาพ }	100.00 %	100.00 %	100.00 %	100.00 %	98.67 %	74.27 %
กำลัง	{ กำลัง , กำ ลัง }	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
กำลังคน	{ กำลังคน , กำลัง คน , กำ ลังคน }	100.00 %	98.37 %	98.37 %	94.23 %	100.00 %	46.29 %
ลงทุน	{ ลงทุน , ลง ทุน }	100.00 %	100.00 %	100.00 %	100.00 %	92.07 %	60.97 %
รายได้	{ รายได้ , ราย ได้ }	100.00 %	96.78 %	100.00 %	100.00 %	94.54 %	74.50 %
ความรู้	{ ความรู้ , ความ รู้ }	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	42.31 %
เพื่อให้	{ เพื่อให้ , เพื่อ ให้ }	100.00 %	97.06 %	100.00 %	98.78 %	53.85 %	54.97 %

ตารางที่ 8-4 ตารางแสดงประสิทธิภาพการแก้ไขปัญหาความกำกวม แบบที่ไม่ต้องใช้บริบท (Context Independent)
โดยใช้การแก้ปัญหาแบบเซตข้อความส่วนหน้า

ข้อความที่กำกวม	เซตข้อความส่วนหน้า (Prefix Set)	วินโนว์		ริปเปอร์		โมเดลไตรแกรม	แบบเหมือนมากที่สุด
		ชุดสอน	ชุดทดสอบ	ชุดสอน	ชุดทดสอบ		
ข้อมูล	{ ข้อ , ข้อมูล }	100.00 %	100.00 %	98.58 %	91.23 %	100.00 %	100.00 %
อบรม	{ อบ , อบรม }	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
หน้าที่	{ หน้า , หน้าที่ }	97.24 %	94.38 %	97.68 %	92.00 %	95.84 %	50.14 %
คุณภาพ	{ คุณ , คุณภาพ }	100.00 %	100.00 %	98.33 %	91.01 %	98.67 %	74.27 %
กำลัง	{ กำ , กำลัง }	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
กำลังคน	{ กำ , กำลัง , กำลังคน }	99.54 %	90.74 %	98.37 %	90.46 %	100.00 %	46.29 %
ลงทุน	{ ลง , ลงทุน }	100.00 %	100.00 %	100.00 %	100.00 %	92.07 %	60.97 %
รายได้	{ ราย , รายได้ }	100.00 %	99.78 %	100.00 %	100.00 %	94.54 %	74.50 %
ความรู้	{ ความ , ความรู้ }	100.00 %	99.67 %	100.00 %	99.36 %	100.00 %	42.31 %
เพื่อให้	{ เพื่อ , เพื่อให้ }	100.00 %	97.46 %	100.00 %	94.17 %	53.85 %	54.97 %

8.4 ขั้นตอนการนำคุณลักษณะเข้ามาใช้ในการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม



รูปที่ 8-2 ขั้นตอนการแก้ไขปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมโดยใช้คุณลักษณะ

จากรูปที่ 8-2 แสดงขั้นตอนการทำงานของการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจานุกรมโดยใช้คุณลักษณะ ซึ่งจะมีขั้นตอนดังนี้คือ

1. นำประโยคมาทำการตัดคำโดยใช้ไตรแกรมโมเดล
2. เลือกประโยคที่ดีที่สุด N ประโยค
3. ทำการค้นหาบริเวณที่น่าจะเกิดคำศัพท์ที่ไม่ปรากฏในพจนานุกรม
4. สร้างตัวเลือกคำศัพท์ที่ไม่ปรากฏในพจนานุกรม
5. สร้างประโยคใหม่จากประโยคเดิมโดยนำตัวเลือกฯ ไปแทนที่
6. กำกับหน้าที่คำโดยไตรแกรมโมเดล
7. นำกฎจากริปเปอร์หรือโครงข่ายวินโนว์ เข้ามาใช้ในการเลือกตัวเลือกฯ ที่มีคะแนนมากที่สุด

8.5 ผลการทดลองการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

สำหรับการทดลองแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม จะแสดงดังตารางที่ 8-5 ซึ่งในตารางนี้จะวัดประสิทธิภาพในการเรียนรู้คุณลักษณะที่จะนำมาใช้ในการแก้ปัญหาระหว่างวินโนว์กับริปเปอร์ โดยจำนวนตัวอย่างที่ใช้ในการเรียนรู้ทั้งจะใช้ทั้งตัวอย่างที่ถูกและตัวอย่างที่ผิดจำนวน 1509 ตัวอย่างและ 9357 ตัวอย่างตามลำดับ สำหรับตัวอย่างที่ให้ทดสอบก็มีจำนวน 377 ตัวอย่างสำหรับตัวอย่างจริงและ 2337 ตัวอย่างสำหรับตัวอย่างเท็จ และสำหรับจำนวนคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างชัดเจนและอย่างซ่อนเร้นบางส่วนนั้นจะมีจำนวน 1235 ตัวอย่าง และจำนวนคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วนมีจำนวน 651 ตัวอย่าง โดยแบ่งออกเป็น 2 ส่วนคือ ส่วนแรก 80% สำหรับการสอน และอีก 20% สำหรับการทดสอบ

ตารางที่ 8-5 ตารางแสดงผลการทดลองการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

	คำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างชัดเจนและอย่างซ่อนเร้นบางส่วน		**คำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วน**	
	ชุดสอน	**ชุดทดสอบ**	**ชุดสอน**	**ชุดทดสอบ**
ค้นหาพบ	100.00 %	100.00 %	87.82 %	83.25 %
วินโนว์	95.26 %	92.75 %	72.87 %	68.21 %
ริปเปอร์	93.25 %	89.75 %	69.25%	65.03 %

จากผลการทดลองในตารางที่ 8-1 แสดงให้เห็นว่าการค้นหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างชัดเจนและอย่างซ่อนเร้นบางส่วนนั้น จะสามารถค้นหาได้ถูกต้องทั้งหมด ในขณะที่การค้นหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วนนั้นจะถูกค้นพบเพียงประมาณ 87% และสำหรับการเรียนรู้คุณลักษณะที่นำมาใช้ในการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้น วินโนว์จะให้ความถูกต้องสูงกว่าริปเปอร์สำหรับคำศัพท์ที่ไม่ปรากฏในพจนานุกรมทุกประเภท

8.6 สรุปผลการทดลองการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

วิธีการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม ภายในวิทยานิพนธ์นี้ จะสามารถแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมได้ทุกรูปแบบ โดยที่การค้นหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างชัดเจนและอย่างซ่อนเร้นบางส่วนจะสามารถค้นพบได้ทั้งหมด แต่สำหรับคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วนนั้นจะไม่สามารถค้นพบได้ทั้งหมด ส่วนการนำเอาวินโนว์กับริปเปอร์เข้ามาใช้ในการแก้ปัญหานั้นจะเห็นว่าวินโนว์สามารถแก้ปัญหาได้ดีกว่าริปเปอร์ทั้งในการแก้ปัญหาของคำศัพท์ที่ไม่ปรากฏใน

พจนานุกรมแบบอย่างชัดเจนและซ่อนเร้นบางส่วน กับคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วน

บทที่ 9

บทสรุปและแนวทางการพัฒนาต่อ

การตัดคำเป็นงานที่จำเป็นสำหรับงานด้านการประมวลผลภาษาและธรรมชาติสำหรับภาษาไทย ทำให้ได้มีการพัฒนาการตัดคำต่อเนื่องกันมาเป็นเวลานาน และจากงานวิทยานิพนธ์นี้ได้มีการนำคุณลักษณะต่างๆ เข้ามาใช้ในการตัดคำ ซึ่งจะสามารถแก้ปัญหาเรื่องความกำกวม และเรื่องคำศัพท์ที่ไม่ปรากฏในพจนานุกรม

9.1 ประสิทธิภาพการนำคุณลักษณะมาใช้ในการแก้ปัญหาการตัดคำ

สำหรับปัญหาเรื่องความกำกวม ภายในงานวิทยานิพนธ์นี้ได้แบ่งประเภทความกำกวมออกเป็น 2 ประเภท คือความกำกวมประเภทที่ต้องใช้บริบทในการแก้ปัญหา และความกำกวมที่ไม่จำเป็นต้องใช้บริบทในการแก้ปัญหา เมื่อนำการตัดคำโดยใช้คุณลักษณะเข้ามาช่วยนั้นสามารถจะแก้ปัญหาความกำกวมทั้ง 2 แบบได้ และให้ประสิทธิภาพดีกว่าการตัดคำที่ผ่านมา คือการตัดคำโดยใช้โมเดลไตรแกรม และการตัดคำโดยเลือกแบบเหมือนมากที่สุด

ส่วนปัญหาเรื่องคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้นในวิทยานิพนธ์นี้ได้มีการแบ่งประเภทของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมออกเป็น 2 ประเภทหลักๆ คือ คำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างชัดเจน และคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้น และสำหรับคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นนั้นยังแบ่งออกเป็นอีก 2 ประเภทย่อย ๆ คือ คำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นบางส่วน และ คำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วน ซึ่งในงานวิทยานิพนธ์สามารถแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมได้ทุกประเภท แต่สำหรับการแก้ปัญหาของคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วนนั้น จะแก้ได้ต่ำกว่าแบบอื่น เนื่องจากคำศัพท์ประเภทนี้จะทำการค้นหาได้ยากกว่าแบบอื่น

ดังนั้นการนำคุณลักษณะเข้ามาใช้ในการแก้ปัญหาความกำกวม และปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมนั้นจะมีประสิทธิภาพมากกว่า การตัดคำโดยใช้โมเดลไตรแกรม และการตัดคำโดยเลือกแบบที่เหมือนมากที่สุด

9.2 **ข้อเสนอแนะ**

วิทยานิพนธ์นี้ได้นำคุณลักษณะเข้ามาแก้ปัญหาทั้งความกำกวม และปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรม สำหรับการแก้ปัญหาความกำกวมแบบที่ไม่ต้องใช้บริบทนั้น ในทางปฏิบัติถ้ามีรายการข้อความที่กำกวมประเภทนี้ ก็สามารถจะนำไปใส่ไว้ในพจนานุกรมได้เลย และต้องมีข้อมูลบอกว่าควรจะตัดคำเป็นอย่างไร ซึ่งเมื่อบรรจุไว้ในพจนานุกรมแล้วจะช่วยให้การตัดคำสามารถทำงานได้เร็วขึ้น เนื่องจากไม่ต้องตัดคำที่เป็นไปได้ทุกแบบสำหรับข้อความกำกวมนั้น

สำหรับการแก้ปัญหาคำศัพท์ที่ไม่ปรากฏในพจนานุกรมภายในวิทยานิพนธ์นี้ จากการทดลองแสดงให้เห็นว่าคำศัพท์ที่ไม่ปรากฏในพจนานุกรมอย่างซ่อนเร้นทุกส่วนนั้น จะแก้ปัญหาได้ต่ำกว่าแบบอื่นๆ เนื่องจากการค้นหาคำศัพท์ประเภทนี้จะทำได้ยากกว่าแบบอื่นๆ ดังนั้นในการพัฒนาประสิทธิภาพในการแก้ปัญหาของคำศัพท์ประเภทนี้จึงควรจะต้องปรับปรุงวิธีการค้นหาบริเวณที่เกิดคำศัพท์ประเภทนี้

รายการอ้างอิง

ภาษาไทย

ดวงแก้ว สวามิภักดิ์. 2533. การสร้างซอฟต์แวร์วิเคราะห์ไวยากรณ์ไทยภายใต้ระบบยูนิกซ์. กรุงเทพ : สำนักพิมพ์มหาวิทยาลัยธรรมศาสตร์.

บุญเรือง ธนาสุนทรไพศาล. 2533. การออกแบบและพัฒนาส่วนเชื่องโยงสำหรับการตัดคำและการแทรกอักขระแบ่งคำภาษาไทย. วิทยานิพนธ์ ปริญญามหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

พิสิทธิ์ พรมจันทร์. 2540. การวิเคราะห์แนวทางการเปรียบเทียบสมรรถนะของโปรแกรมแยกคำภาษาไทย. วิทยานิพนธ์ ปริญญามหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

ยืน ภู่วรวรรณ และ วิวรรธน์ อิ่มอารมณ์. 2529. การแบ่งแยกพยางค์ไทยด้วยดิกชันนารี. รายงานการประชุมวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 9.

รัตติกร วรากูลศิริพันธ์, จงกล งามวิวิทย์, สมศักดิ์ จันวัน, สุธาทิพย์ จิวัธยากูล และ ศักดิ์ชัย ทิพย์จักษุรัตน์. 2538.การตัดคำจากประโยคภาษาไทยด้วยวิธีการเทียบคำที่ยาวที่สุด. Papers on Natural Language Processing, Compiled by Virach Sornlertlamvanich.

สมปรารถนา รัทยานนท์. 2535. โครงสร้างข้อมูลสำหรับพจนานุกรมอิเล็คทรอนิคส์ภาษาไทย. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

สัมพันธ์ ระรื่นรมย์. 2534. การแบ่งคำไทยด้วยพจนานุกรม. โครงงานวิศวกรรม ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.

วิรัช ศรเสิศล้ำวาณิช. 2536. การตัดคำภาษาไทยในระบบแปลภาษา. การแปลภาษาด้วยคอมพิวเตอร์. หน้า 50-55. กรุงเทพ : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

ภาษาอังกฤษ

Allen, J. 1995. Natural Language Understanding. 2nd ed. Redwood City, California : Benjamin/Cummings.

Aoe, J. 1989. An Efficient Digital Search Algorithm by Using a Double-Array Structure. IEEE Trans. Software Eng., Vol. 15, pp. 1066-1077.

Blum, A. 1997. Empirical Support for Winnow and Weighted-Majority Algorithm: Results on a Calendar Scheduling Domain, Machine Learning, 26: 5-23.

Charniak, E. 1996. Statistical Language Learning. Cambridge : MIT Press.

Charnyapornpong, S. 1983. A Thai Syllable Separation Algorithm. Master Thesis. Asian Institute of Technology.

Charoenpornsawat, P., Kijsirikul, B. and Meknavin, S. 1998. Feature-based Thai Unknown Word Boundary Identification Using Winnow. In Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS'98).

Charoenpornsawat, P., Kijsirikul, B. and Meknavin, S. 1998. Feature-Based Proper Name Identification in Thai. In Proceeding of the National Computer Science and Engineering Conference'98 (NCSEC'98).

Cohen, W., W. 1995. Fast Effective Rule Induction. In Proceedings of Twelfth International Conference on Machine Learning. Morgan Kaufmann.

Corman, T. ,H., Leiserson, C., E. and Rivest, R., L. 1990. Introduction to Algorithms. Mit Press.

Frakes, W., B. and Baeza-Yates, R. 1992. Introduction to Data Stuctures and Algorithms Related to Information Retrieval. New Jersey : Prentice Hall. pp.13-27.

Golding, A., R. and Roth, D. 1996. Applying Winnow to Context-Sensitive Spelling Correction. In Proceedings of the Thirteenth International Conference on Machine Learning.

Johnson, S. C. 1975. YACC-Yet another compiler-complier. NJ, Comput. Sci. Tech. Rep.32: 1-34.

Kanlayanawat, W., Prasitjutrakul, S. 1997. Automatic Indexing for Thai Text with Unknown Words using Trie Structure. In Proceedings of the Natural Language Processing Pacific Rim Symposium 1997(NLPRS'97).

Kawtrakul, A., Kumtanode, S., Jamjanya, T. and Jewriyavech C. 1995. A Lexicon Model for Writing Production Assistant System. In Proceedings of the Symposium on Natural Language Processing in Thailand'95.

Kawtrakul, A.,Thumkanon, C., Poovorawan, Y., Varasrai, P. and Suktarachan, M. 1997. Automatic Thai Unknown Word Recognition. In Proceedings of the Natural Language Processing Pacific Rim Symposium 1997(NLPRS'97).

Kijsirikul, B., Sinthupinyo, S. and Supanwansa, A. 1998. Thai Printed Character Recognition by Combining Inductive Logic Programming with Backpropagation Neural Network. In Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS'98).

Littlestone, N. 1998. Learning Quickly when Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. Machine Learning 2.

Meknavin, S.,Charoenpornsawat, P. and Kijsirikul, B. 1997. Feature-based Thai Word Segmentation. In Proceedings of the Natural Language Processing Pacific Rim Symposium 1997(NLPRS'97).

Meknavin, S., Kijsirikul, B., Chotimongkol, A. and Nuttee, C. 1998. Progress of Combining Trigram and Winnow in Thai OCR Error Correction. In Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS'98).

Thairatananond, Y. 1981. Towords the design of a Thai text syllable analyzer. Master Thesis. Asian Institute of Technology.

Sornlertlamvanich, V.,Charoenporn, T. and Isahara, H. 1997. ORCHID: Thai Part-Of-Speech Tagged Corpus. In Technical Report Orchid Corpus. Bangkok : NECTEC.

ภาคผนวก

ภาคผนวก ก

กฎที่ใช้ในการตัดพยางค์ ของดวงแก้ว สวามิภักดิ์

กฎที่ใช้ในการตัดพยางค์มีอยู่ทั้งหมด 43 กฎ โดยสัญลักษณ์ที่ใช้มีดังต่อไปนี้

c	::=พยัญชนะปรกติ (Consonant)
v	::= สระ (Vowel)
t	::= วรรณยุกต์ (Tonal Mark)
s	::= ตัวสะกด (speller)
[…]?	::= ทางเลือก กล่าวคือ อาจจะมีหรือไม่มีก็ได้
[a1 a2 … an]	::= เลือกตัวอักษรใดตัวอักษรหนึ่งระหว่าง a1 … an

กฎที่ได้ 43 กฎมีดังต่อไปนี้

1. [c][t]?[ะ ำ า]
2. [c][ิ ี ึ ื ุ ู]?[ะ ำ า]
3. [c][ิ ี ึ ื ุ ู][t]?
4. [c][t]?[s]
5. [c]ั[t]?[s]
6. [เ แ โ ใ ไ][c][t]?
7. [เ แ][c]็[s]
8. [เแโ][c][t]?ะ
9. [เแโ][ก ข ค ต ท บ ป พ ฟ จ ช ศ ส]ร[t]?ะ
10. [เแโ][ก ข ค บ ป ผ พ]ล[t]?ะ
11. เ[ก ข ค ต ท บ ป พ ฟ จ ช ศ ส]ร[า ิ ี ึ ื ุ ู][t]?
12. เ[ก ข ค บ ป พ ฟ]ล[า ิ ี ึ ื ุ ู][t]?
13. [เแโ][ก ข ค]ว[t]?ะ
14. เ[c]ี[t]?ย
15. เ[c][t]?าะ
16. เ[c][t]?[าะ]
17. เ[c][t]?
18. [โ ใ ไ]ห[ง ญ น ม ย ร ล ว][t]?
19. เ[c]ื[t]?อ
20. [c]ื[t]?อ
21. [c]์

22. [็ ์ ่ ้ ๊ ๋ ั ิ ี ึ ื ุ ู]
23. ฯ
24. เ[จ ส]ิ ญ
25. ห ร[t]?[ะ า ำ ุ ู]
26. ห ร[ิ ี ึ ื][t]?
27. เ[ก ข ค บ ป ผ พ ฟ]ลื[t]? อ
28. เ[ก ข ค ต ท บ ป พ ฟ]รื[t]? อ
29. เ[ก ข ค บ ป ผ พ ฟ]ลิ[t]?[s]
30. เ[ก ข ค ต ท บ ป พ ฟ]ริ[t]?[s]
31. แ[ก ข ค ต ท บ ป พ ฟ]ร[t]?[ก ง ว น บ ม ท ต ด]
32. แ[ก ข ค บ ป ผ พ ฟ]ล[t]?[ก ง ว น บ ม ท ต ด]
33. เห[ง ญ น ม ย ร ล ว][t]? า
34. [A-Z a-z เครื่องหมายพิเศษต่างๆ]*
35. ฯลฯ
36. ๆ
37. เ[ก ข ค]วี[t]? ย
38. เ[ก ข ค ต ท บ ป พ ฟ]รี[t]? ย
39. เ[ก ข ค ง บ ผ พ ฟ]ลี[t]? ย
40. “ “ (space)
41. [0-9]*
42. “\n”
43. . (ทุกตัวอักษรที่ไม่อยู่ในกฎ)

ภาคผนวก ข

ตารางแสดงหน้าที่ของคำในภาษาไทย จากคลังข้อความออร์คิด

ประเภทของคำ	รายละเอียด	ตัวอย่าง
NPRP	Proper noun	วินโดวส์ 95, โคโรน่า, โค้ก, พระอาทิตย์
NCNM	Cardinal number	หนึ่ง, สอง, สาม, 1, 2, 3
NONM	Ordinal number	ที่หนึ่ง, ที่สอง, ที่สาม, ที่1, ที่2, ที่ 3
NLBL	Label noun	1, 2, 3, 4, ก, ข, ค, ง
NCMN	Common noun	หนังสือ, อาหาร, อาคาร, คน
NTTL	Title noun	ดร., พลเอก
PPRS	Personal pronoun	คุณ, เขา, ฉัน
PDMN	Demonstrative pronoun	นี่, นั่น, ที่นั่น, ที่นี่
PNTR	Interrogative pronoun	ใคร, อะไร, อย่างไร
PREL	Relative pronoun	ที่, ซึ่ง, อัน, ผู้
VACT	Active verb	ทำงาน, ร้องเพลง, กิน
VSTA	Stative verb	เห็น, รู้, คือ
VATT	Attribute verb	อ้วน, ดี, สวย
XVBM	Pre-verb auxiliary, before negator “ไม่”	เกิด, เกือบ, กำลัง
XVAM	Pre-verb auxiliary, after negator “ไม่”	ค่อย, น่า, ได้
XVMM	Pre-verb, before or after negator “ไม่”	ควร, เคย, ต้อง
XVBB	Pre-verb auxiliary, in imperative mood	กรุณา, จง, เชิญ, อย่า, ห้าม
XVAE	Post-verb auxiliary	ไป, มา, ขึ้น
DDAN	Definite determiner, after noun without classifier in between	นี่, นั่น, โน่น, ทั้งหมด

ประเภทของคำ	รายละเอียด	ตัวอย่าง
DDAC	Definite determiner, allowing classifier in between between	นี้, นั้น, โน้น, นู้น
DDBQ	Definite determiner, between noun and classifier or preceding Quantitative expression	ทั้ง, อีก, เพียง
DDAQ	Definite determiner, following quantitative expression	พอดี, ถ้วน
DIAC	Indefinite determiner, following noun; allowing classifier in between	ไหน, อื่น, ต่างๆ
DIBQ	Indefinite determiner, Between noun and classifier or preceding quantitative expression	บาง, ประมาณ, เกือบ
DIAQ	Indefinite determiner, following quantitative expression	กว่า, เศษ
DCNM	Determiner, cardinal number expression	หนึ่งคน, เสือ 2 ตัว
DONM	Determiner, ordinal number expression	ที่หนึ่ง, ที่สอง, ที่สุดท้าย
ADVN	Adverb with noun form	เก่ง, เร็ว, ช้า, สม่ำเสมอ
ADVI	Adverb with iterative form	เร็วๆ, เสมอๆ, ช้าๆ
ADVP	Adverb with prefixed form	โดยเร็ว
ADVS	Sentential adverb	โดยปกติ, ธรรมดา
CNIT	Unit classifier	ตัว, คน, เล่ม
CLTV	Collective classifier	คู่, กลุ่ม, ฝูง, เชิง, ทาง, ด้าน, แบบ, รุ่น
CMTR	Measurement classifier	กิโลกรัม, แก้ว, ชั่วโมง

ประเภทของคำ	รายละเอียด	ตัวอย่าง
CFQC	Frequency classifier	ครั้ง, เที่ยว
CVBL	Verbal classifier	ม้วน, มัด
JCRG	Coordinating conjunction	และ, หรือ, แต่
JSBR	Comparative conjunction	กว่า, เหมือนกับ, เท่ากับ
RPRE	Preposition	จาก, ละ, ของ, ใต้, บน
INT	Interjunction	โอ้ย, โอ้, เออ, เอ๋, อ๋อ
FIXN	Nominal prefix	การทำงาน, ความสนุกสนาน
FIXV	Adverbial prefix	อย่างเร็ว
EAFF	Ending for affirmative sentence	จ๊ะ, จ้ะ, ค่ะ, ครับ, นะ, น่า, เถอะ
EITT	Ending for interrogative sentence	หรือ, เหรอ, ไหม, มั้ย
NEG	Negator	ไม่, มิได้, ไม่ได้, มิ
PUNC	Punctuation	(,), “, ,, ๆ

ภาคผนวก ค

ความถี่ของข้อความที่กำกวมจากคลังข้อความออร์คิด

ความถี่ของความกำกวมที่แบบไม่ต้องใช้บริบท 50 อันดับแรก

1 - 743 ที่จะ
407 ที่จะ|
336 ที่|จะ|

2 - 465 ทำให้
346 ทำให้|
119 ทำ|ให้|

3 - 384 ฝึกอบรม
226 ฝึกอบรม|
158 ฝึก|อบรม|

4 - 345 ไม่ได้
54 ไม่ได้|
291 ไม่|ได้|

5 - 330 ทำงาน
312 ทำงาน|
18 ทำ|งาน|

6 - 323 ความรู้
1 ความรู้|
322 ความ|รู้|

7 - 267 ได้รับ
253 ได้รับ|
14 ได้|รับ|

8 - 247 มากขึ้น
152 มากขึ้น|
95 มาก|ขึ้น|

9 - 242 ควรจะ
232 ควรจะ|
10 ควร|จะ|

10 - 240 แต่ละ
239 แต่ละ|
1 แต่|ละ|

11 - 232 ดำเนินการ
223 ดำเนินการ|
9 ดำเนิน|การ|

12 - 211 เพื่อให้
207 เพื่อให้|
4 เพื่อ|ให้|

13 - 194 ความต้องการ
6 ความต้องการ|
188 ความ|ต้องการ|

14 - 186 วิธีการ
164 วิธีการ|
22 วิธี|การ|

15 - 184 การที่
174 การที่|
10 การ|ที่|

16 - 180 อาจจะ
170 อาจจะ|
10 อาจ|จะ|

17 - 170 ความสามารถ
1 ความสามารถ|
169 ความ|สามารถ|

18 - 168 ในด้าน
86 ในด้าน|
82 ใน|ด้าน|

19 - 164 เหล่านี้
163 เหล่านี้|
1 เหล่า|นี้|

20 - 158 ปฏิบัติงาน
120 ปฏิบัติงาน|
38 ปฏิบัติ|งาน|

21 - 157 ต่อไป
156 ต่อไป|
1 ต่อ|ไป|

22 - 153 ประเทศไทย
142 ประเทศไทย|
11 ประเทศ|ไทย|

23 - 148 จัดการ
130 จัดการ|
18 จัด|การ|

24 - 143 ดังกล่าว
141 ดังกล่าว|
2 ดัง|กล่าว|

25 - 142 หรือไม่
92 หรือไม่|
50 หรือ|ไม่|

26 - 136 เข้ามา
129 เข้ามา|
7 เข้า|มา|

27 - 134 ทางด้าน
69 ทางด้าน|
65 ทาง|ด้าน|

28 - 134 มากกว่า
39 มากกว่า|
95 มาก|กว่า|

29 - 130 ไม่ใช่
50 ไม่ใช่|
80 ไม่|ใช่|

30 - 129 เท่านั้น
127 เท่านั้น|
2 เท่า|นั้น|

31 - 122 ดำเนินงาน
114 ดำเนินงาน|
8 ดำเนิน|งาน|

32 - 119 เป็นไป
111 เป็นไป|
8 เป็น|ไป|

33 - 116 เพิ่มขึ้น
82 เพิ่มขึ้น|
34 เพิ่ม|ขึ้น|

34 - 114 ทางเศรษฐกิจ
82 ทางเศรษฐกิจ|
32 ทาง|เศรษฐกิจ|
35 - 113 คงจะ
112 คงจะ|
1 คง|จะ|
36 - 112 เกิดขึ้น
58 เกิดขึ้น|
54 เกิด|ขึ้น|
37 - 112 มักจะ
110 มักจะ|
2 มัก|จะ|
38 - 111 ในประเทศ
33 ในประเทศ|
78 ใน|ประเทศ|
39 - 109 ความสัมพันธ์
2 ความสัมพันธ์|
107 ความ|สัมพันธ์|
40 - 108 ว่าการ
2 ว่าการ|
106 ว่า|การ|
41 - 105 ทำได้
26 ทำได้|
79 ทำ|ได้|
42 - 103 ในทาง
42 ในทาง|
61 ใน|ทาง|
43 - 102 วางแผน
101 วางแผน|
1 วาง|แผน|
44 - 100 แล้วก็
29 แล้วก็|
71 แล้ว|ก็|
45 - 99 การเงิน
68 การเงิน|
31 การ|เงิน|
46 - 98 การเมือง
86 การเมือง|
12 การ|เมือง|
47 - 95 ความจำเป็น
4 ความจำเป็น|
91 ความ|จำเป็น|
48 - 91 มีประสิทธิภาพ
22 มีประสิทธิภาพ|
69 มี|ประสิทธิภาพ|
49 - 91 ทุกคน
76 ทุกคน|
15 ทุก|คน|
50 - 90 ก็ได้
47 ก็ได้|
43 ก็|ได้|

ความถี่ของความกำกวมแบบที่ไม่ต้องใช้บริบท 50 อันดับแรก

1 - 812 สามารถ
812 สามารถ|
2 - 464 ข้าราชการ
464 ข้าราชการ|
3 - 388 ระดับ
388 ระดับ|
4 - 353 ต้องการ
353 ต้องการ|
5 - 310 ข้อมูล
310 ข้อมูล|
6 - 279 การศึกษา
279 การ|ศึกษา|
7 - 271 เกี่ยวกับ
271 เกี่ยวกับ|
8 - 262 ประชาชน
262 ประชาชน|
9 - 258 อบรม
258 อบรม|
10 - 253 จำเป็น
253 จำเป็น|
11 - 234 ตนเอง
234 ตนเอง|
12 - 229 รวมทั้ง
229 รวมทั้ง|
13 - 229 หน้าที่
229 หน้าที่|
14 - 228 คุณภาพ
228 คุณภาพ|
15 - 226 รายได้
226 รายได้|
16 - 224 สินค้า
224 สินค้า|
17 - 218 ระหว่าง
218 ระหว่าง|
18 - 216 เปลี่ยนแปลง
216 เปลี่ยนแปลง|
19 - 206 ลงทุน
206 ลงทุน|
20 - 204 การบริหาร
204 การ|บริหาร|
21 - 203 ปรับปรุง
203 ปรับปรุง|
22 - 203 หน่วยงาน
203 หน่วยงาน|
23 - 201 หัวหน้า
201 หัวหน้า|
24 - 200 ส่งเสริม
200 ส่งเสริม|
25 - 197 อย่างไร
197 อย่างไร|
26 - 192 เอกชน
192 เอกชน|
27 - 187 เหมาะสม
187 เหมาะสม|
28 - 169 กระทำ
169 กระทำ|
29 - 167 ราคา

167 ราคา|
30 - 167 ปกครอง
167 ปกครอง|
31 - 162 เข้าใจ
162 เข้าใจ|
32 - 160 พื้นฐาน
160 พื้นฐาน|
33 - 155 กำลัง
155 กำลัง|
34 - 153 แก้ไข
153 แก้ไข|
35 - 150 ควบคุม
150 ควบคุม|
36 - 148 โครงการ
148 โครงการ|
37 - 147 ชนบท
147 ชนบท|
38 - 145 โครงสร้าง
145 โครงสร้าง|
39 - 140 ป้องกัน
140 ป้องกัน|
40 - 139 ความสำคัญ
139 ความ|สำคัญ|
41 - 139 งบประมาณ
139 งบประมาณ|
42 - 139 ที่สุด
139 ที่สุด|
43 - 139 มหาวิทยาลัย
139 มหาวิทยาลัย|
44 - 136 โดยเฉพาะ
136 โดยเฉพาะ|
45 - 136 กำลังคน
136 กำลังคน|
46 - 133 เกี่ยวข้อง
133 เกี่ยวข้อง|
47 - 131 องค์การ
131 องค์การ|
48 - 130 บทบาท
130 บทบาท|
49 - 128 กฎหมาย
128 กฎหมาย|
50 - 125 กิจกรรม
125 กิจกรรม|

ประวัติผู้เขียน

นายไพศาล เจริญพรสวัสดิ์ เกิดวันที่ 16 กันยายน พ.ศ. 2517 กรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี 2539 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2540 ปัจจุบันทำงานตำแหน่งผู้ช่วยนักวิจัย ห้องปฏิบัติการวิจัยและพัฒนาภาษาและซอฟต์แวร์ ศูนย์เทคโนโลยีอิเล็คทรอนิกส์และคอมพิวเตอร์แห่งชาติ