การวิเคราะห์เพื่อหารูปแบบการแบ่งแยกคำภายในข้อความที่เหมาะสม

Thai Pattern Analysis For Word Segmentation

โดย

นางสาว กรรณิกา จินดาปทีป

รหัส 42067026

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2543
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การวิเคราะห์เพื่อหารูปแบบการแบ่งแยกคำภายในข้อความที่เหมาะสม
นักศึกษา	นางสาว กรรณิกา จินดาปทีป
อาจารย์ที่ปรึกษา	ดร. โชติพัชร์ ภรณวลัย
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

ภาษาไทยมีรูปแบบการเขียนที่มีลักษณะการจัดเรียงกันของหน่วยคำแบบต่อเนื่องกันไป ไม่มีการเว้นช่องว่างระหว่างหน่วยคำ การกำหนดขอบเขตที่แน่นอนของคำจึงเป็นเรื่องที่ทำได้ยาก ที่ผ่านมาได้มีการพัฒนาการตัดคำด้วยวิธีการต่างๆ หลายวิธี และแต่ละวิธีก็มุ่งหวังที่จะได้ผลการตัดคำที่ถูกต้อง แต่วิธีการต่างๆ เหล่านั้นยังคงสามารถแบ่งแยกคำได้ในระดับหนึ่งเท่านั้น ซึ่งยังไม่สามารถกำหนดขอบเขตของคำได้อย่างเหมาะสม

ในโครงการพัฒนานี้ ได้ศึกษาถึงการนำเอาคุณลักษณะเฉพาะของแต่ละคำ จากฐานข้อมูลคำ Orchid Corpus ที่ได้มีการวิเคราะห์เกี่ยวกับหน้าที่ของคำในประโยคไว้แล้ว มาช่วยในการกำหนดขอบเขตของคำ คุณลักษณะที่นำมาวิเคราะห์ในครั้งนี้ประกอบด้วย บริบทรอบข้างของคำ และลำดับการเกิดของคำในประโยค ซึ่งคุณลักษณะทั้งสอง จะช่วยให้ผลการตัดคำ มีความถูกต้องมากยิ่งขึ้น

การแก้ปัญหาความกำกวมในข้อความโดยการใช้คุณลักษณะเคยมีผู้เสนอแนวทางไว้ 2 วิธี ด้วยกันคือ วิธีแรกแก้ปัญหาโดยการใช้เซตของข้อความส่วนหน้า (prefix set) และ วิธีที่สองแก้ปัญหาโดยการใช้เซตของความสับสน (confusion set) ทั้ง 2 วิธีนี้มีข้อดีข้อเสียที่แตกต่างกัน ในโครงการพัฒนานี้จึงได้รวมเอาข้อดีของทั้ง 2 วิธีมาใช้ในการวิเคราะห์ความกำกวมร่วมกัน เพื่อเพิ่มประสิทธิภาพในการทำงานให้กับระบบการตัดคำ

Title	Thai Pattern Analysis For Word Segmentation
Student	Miss Kannika Jindapateep
Advisor	Dr. Chotipat Pornavalai
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

ABSTRACT

Thai language is written without interword delimiters. So finding boundary of each word is difficult. In the past, there were researches about word segmentation in many ways, which expect to segment Thai word correctly. But all of them can segment Thai words only in a definite level, which cannot segment all words suitably.

This project is about using feature-base of each word from Orchid Corpus, which has analyzed about role of word in the sentence, for word segmentation. The feature-base that was analyzed is word surrounding and word sequence. Both features will help us to segment Thai word more correctly.

There were 2 methods in solving ambiguous of word string by using feature-base. First, by using prefix set of word string. Second, by using confusion set. Both methods have different advantages and disadvantages. This project includes the advantages of both methods to analyze the ambiguous for increase the effective of word segmentation.

กิตติกรรมประกาศ

ขอขอบคุณ ดร. โชติพัชร์ ภรณวลัย อาจารย์ที่ปรึกษา ที่ช่วยให้คำแนะนำในการทำงาน และอำนวยความสะดวกเกี่ยวกับอุปกรณ์ที่ใช้ในการพัฒนาระบบมาโดยตลอด อันเป็นแรงผลักดันให้โครงการพัฒนาระบบการตัดคำนี้ สำเร็จลงได้ด้วยดี

ขอบคุณ Jica (เครื่อง Server ของระบบ) ที่ทำงานอย่างหนัก ด้วยความขยันขันแข็ง เคียงคู่กันมาโดยตลอด

ขอบคุณ พี่ๆ ห้องโปรเจค และเพื่อนๆ ทุกคนที่ให้คำแนะนำ และความช่วยเหลือต่างๆ ในการทำงาน

และขอบคุณ ทุกคนที่มีส่วนร่วมเป็นกำลังใจในการทำงานครั้งนี้ ด้วยดีเสมอมา

กรรณิกา จินดาปทีป

มีนาคม 2544

สารบัญ

หน้า

สารบัญตาราง

หน้า

ตารางที่

สารบัญภาพ

หน้า

ภาพที่

บทที่ 1

บทนำ

ภาษาไทย เป็นภาษาหนึ่ง ที่มีปัญหาในการแบ่งแยกคำภายในข้อความ เนื่องจากเป็นภาษาที่มีลักษณะการเขียนแบบเรียงต่อเนื่องกันไปของหน่วยคำ โดยไม่มีการเว้นช่องว่างระหว่างคำ จึงเป็นเรื่องยาก ที่จะนำข้อความภาษาไทยไปประมวลผลด้วยเครื่องคอมพิวเตอร์ในงานด้านต่างๆ ไม่ว่าจะเป็นงานเกี่ยวกับการแปลภาษาไทย-อังกฤษ (Thai-English Machine Translation) การสังเคราะห์เสียงภาษาไทย (Thai Speech Synthesis) หรือ การสืบค้นหาข้อมูลด้วยข้อความภาษาไทย (Thai Full Text Search) เพราะจะต้องทำให้คอมพิวเตอร์ ทราบว่าขอบเขตของคำที่แท้จริงสิ้นสุดที่ใด การตัดคำที่ถูกต้องจึงนับว่าเป็นสิ่งจำเป็นอย่างมากสำหรับงานในด้าน การประมวลผลภาษาธรรมชาติ เนื่องจากเป็นขั้นตอนแรกของการทำงาน เพราะหากตัดคำผิดแล้ว อาจมีผลทำให้ความหมายของข้อความผิดไปจากเดิม และอาจส่งผลกระทบต่อการทำงานในขั้นต่อไป ซึ่งจะเป็นเหตุให้ไม่สามารถได้รับผลการทำงานที่ถูกต้อง

1.1 ปัญหาของการตัดคำ

การตัดคำในภาษาไทย ได้มีการพัฒนามาหลายวิธีการด้วยกัน ได้แก่ การตัดคำโดยการใช้กฎการสะกดตามหลักภาษาไทย การใช้การเปรียบเทียบคำกับคำในพจนานุกรม การใช้สถิติการเกิดร่วมกันของคำ การใช้ไวยากรณ์ภาษา มาช่วยในการตัดคำ แต่การตัดคำด้วยวิธีดังกล่าว ให้ความถูกต้องได้ในระดับหนึ่งเท่านั้น ปัญหาของการตัดคำในภาษาไทย เกิดขึ้นจาก ข้อความที่พิจารณามีความกำกวม สามารถแบ่งแยกคำได้มากกว่า 1 รูปแบบ

เช่น ข้อความ มากรอง สามารถแบ่งแยกคำได้เป็น

มา กรอง และ มาก รอง

ทำให้ไม่สามารถเลือกได้ว่ารูปแบบใด เป็นรูปแบบที่เหมาะสม

ในโครงการนี้จึงได้นำเอาวิธีการพิจารณาคุณลักษณะต่างๆที่เกิดร่วมกันกับคำกำกวมมาช่วยในการพิจารณา เพื่อเลือกรูปแบบการตัดคำที่ถูกต้อง ในการแก้ปัญหาความกำกวมโดยการใช้คุณลักษณะนั้นมีผู้เสนอแนวทางไว้ 2 วิธีด้วยกันคือ การแก้ปัญหาโดยการใช้ เซตของข้อความส่วนหน้า (prefix set) และการใช้เซตของความสับสน (confusion set) ทั้ง 2 วิธีมีข้อดีข้อเสียที่แตกต่างกัน

โครงการพัฒนาระบบการตัดคำนี้จึงได้รวมเอาข้อดีของทั้ง 2 วิธีมาใช้ในการวิเคราะห์ความกำกวมร่วมกัน เพื่อเพิ่มประสิทธิภาพในการทำงานให้กับระบบการตัดคำ

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อให้การวิเคราะห์การตัดคำในภาษาไทย มีความถูกต้องมากยิ่งขึ้น ในด้านความหมาย โดยการใช้คุณลักษณะเฉพาะของแต่ละคำ มาช่วยในการพิจารณา

1.2.2 เลือกรูปแบบการตัดคำที่เหมาะสม ให้กับข้อความ สำหรับการประมวลผลภาษาธรรมชาติในขั้นตอนต่อไป

1.3 ขอบเขตของโครงการ

1.3.1 แก้ปัญหาการตัดคำ ที่เกิดกับข้อความกำกวม ของวิธีการตัดคำ โดยการเปรียบเทียบข้อความกับคำในพจนานุกรม ซึ่งสามารถตัดคำได้มากกว่า 1 รูปแบบ

1.3.2 เลือกรูปแบบ การตัดคำในข้อความที่เหมาะสม โดยการใช้คุณลักษณะเฉพาะของแต่ละคำ มาช่วยในการพิจารณา

1.3.3 นำเอาวิธีการเรียนรู้ของเครื่อง แบบวินโนว์ (Winnow) มาช่วยในการดึงคุณลักษณะเฉพาะของคำ แต่ละคำ จากคลังข้อความ Orchid Corpus ซึ่งแต่ละคำได้ถูกวิเคราะห์ หน้าที่ของคำไว้อย่างถูกต้องแล้ว

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาหลักการ วิธีการ และกระบวนการทำงานในขั้นตอนต่างๆ ของการประมวลผลภาษาธรรมชาติ ว่ามีขั้นตอนอย่างไรบ้าง เพื่อใช้เป็นความรู้พื้นฐานในการเรียนรู้เกี่ยวกับกระบวนการวิเคราะห์ ในแต่ละขั้นตอน ของการประมวลผล โดยเน้นไปที่ขั้นตอน การวิเคราะห์เพื่อแบ่งแยกคำในข้อความ ให้ได้รูปแบบการแบ่งแยกคำที่ถูกต้อง

1.4.2 ศึกษาวิธีการทำงานของระบบการแบ่งแยกคำในรูปแบบต่างๆ ว่ามีขั้นตอน และวิธีการทำงานอย่างไร ผลของการทำงานในแต่ละวิธีมีความถูกต้องมากน้อยแค่ไหน และมีปัญหาอะไรบ้างในการทำงานของแต่ละวิธี เพื่อนำมาเป็นแนวทางในการพัฒนาระบบการตัดคำ ที่ดีกว่าเดิม

1.4.3 ศึกษาวิธีการเรียนรู้ของเครื่อง แบบวินโนว์ ว่ามีวิธีการเรียนรู้เพื่อ แบ่งแยกความแตกต่างของ การตัดคำแต่ละรูปแบบอย่างไร

1.4.4 สร้างฐานข้อมูล เพื่อเก็บข้อมูลที่จำเป็นต่างๆ สำหรับการพัฒนาระบบการตัดคำ

1.4.5 พัฒนาโปรแกรมกำหนดหน้าที่ของคำ โดยโมเดลไตรแกรม แบบ Viterbi Algorithm

1.4.6 พัฒนาโปรแกรมและระบบการตัดคำ ด้วยความรู้ที่ได้จากการเรียนรู้ของเครื่องแบบวินโนว์

1.4.7 ทำการทดลองเพื่อวัดประสิทธิภาพในการทำงาน

1.4.8 จัดทำรายงานโครงการพัฒนา

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการพัฒนา

1.5.1 บทสรุปรวบรวม การตัดคำโดยวิธีการต่างๆ ที่ผ่านมา

1.5.2 รายละเอียดเกี่ยวกับการเรียนรู้ของเครื่อง แบบวินโนว์

1.5.3 โปรแกรมการกำหนดหน้าที่ของคำ

1.5.4 ระบบการตัดคำ ที่ให้ผลการตัดคำที่เหมาะสมกับข้อความมากกว่าเดิม

1.5.5 สรุปปัญหาในการตัดคำของโครงการ

บทที่ 2

งานวิจัย เกี่ยวกับทฤษฎีการตัดคำ

การตัดคำในภาษาไทย เพื่อใช้ในการประมวลผลภาษาธรรมชาตินั้น มีการพัฒนามาอย่างต่อเนื่องเป็นเวลานาน ปัจจุบันจึงมีการตัดคำอยู่หลายวิธี ที่ใช้งานกันอยู่อย่างแพร่หลาย ซึ่งแต่ละวิธีมีลักษณะการทำงานที่ต่างกันออกไป ในรายงานนี้ จะกล่าวถึงวิธีการต่างๆ โดยจำแนกตามรูปแบบการทำงาน ซึ่งสามารถแบ่งแยกได้ 4 ประเภท คือ

1. การตัดคำโดยใช้กฎการสะกดตามหลักภาษาไทย
2. การตัดคำโดยการใช้พจนานุกรม
3. การตัดคำโดยการใช้ค่าทางสถิติ
4. การตัดคำโดยการใช้คุณลักษณะของคำรอบข้าง

โดยงานวิจัย แต่ละประเภทจัดลำดับตามระยะเวลาที่เกิดการพัฒนาการขึ้น

2.1 การตัดคำโดยใช้กฎการสะกดตามหลักภาษาไทย

การตัดคำโดยการใช้กฎ เป็นตัวแบ่งแยกคำนั้น เป็นวิธีการที่พัฒนาขึ้นในช่วงแรกๆ ของการพัฒนาเกี่ยวกับการประมวลผลภาษาธรรมชาติ โดยในสมัยนั้นจะเน้นที่การสร้างกฎ ที่ใช้ในการตัดพยางค์ ตามหลักภาษาไทย เพราะพยางค์มีกฎเกณฑ์การสะกด ที่แน่นอนตายตัวกว่าการสะกดคำ และผลที่ได้จากการตัดคำ โดยวิธีการนี้สามารถแบ่งพยางค์ได้อย่างถูกต้องค่อนข้างมาก แต่ก็ยังคงมีปัญหากับการแบ่งพยางค์บางส่วนอยู่ เนื่องจากไม่สามารถสร้างกฎให้ครอบคลุมพยางค์เหล่านั้นได้ เพราะพยางค์ และคำในภาษาไทยเกิดมาจากหลายภาษาด้วยกัน ทั้ง บาลี สันสกฤต เขมร และคำที่เป็นคำไทยแท้ เป็นเหตุให้รูปแบบของการสะกดพยางค์ บางส่วนมีลักษณะที่แตกต่างกันมาก การสร้างกฎให้ครอบคลุมได้ทั้งหมดจึงเป็นเรื่องที่ยาก ซึ่งบางพยางค์ต้องจัดอยู่ในข้อยกเว้น ส่วนนี้เป็นเหตุให้ผลของการตัดคำมีความถูกต้องอยู่ในระดับหนึ่งเท่านั้น แต่ก็นับว่าเป็นก้าวแรกของการตัดคำไทย ที่เป็นตัวผลักดันให้เกิดวิธีการต่างๆ ที่มีประสิทธิภาพมากขึ้นตามมาในภายหลังอีกจำนวนมาก ผลงานการตัดคำโดยการใช้กฎ ที่น่าสนใจมีดังนี้

2.1.1. งานของ ยุพิน ไทยรัตนานนท์

Yupin Thairatananond (1981) เสนองานวิจัยเกี่ยวกับการตัดพยางค์ โดยหลักไวยากรณ์ภาษาไทย พัฒนาด้วยภาษาพีแอลวัน (PL/I) กฎที่ใช้ในการตัดพยางค์ พิจารณาจากลักษณะของอักขระที่ปรากฏในพยางค์หรือคำ โดยแบ่งหมวดหมู่ของอักขระออกเป็น 5 กลุ่มคือ

1. กลุ่มพยัญชนะ (Consonant)
- พยัญชนะที่อยู่หน้าพยางค์เสมอ
- พยัญชนะที่ส่วนใหญ่จะอยู่หน้าพยางค์
- พยัญชนะที่เป็นทำหน้าที่เป็นตัวสะกด
- พยัญชนะที่ทำหน้าที่เป็นสระ
- อื่นๆ
2. กลุ่มสระ (Vowel)
- สระที่ไม่ต้องมีตัวสะกด
- สระที่อยู่หน้าพยางค์เสมอ
- สระที่ต้องการตัวสะกด
- สระที่มีหรือไม่มีตัวสะกดร่วมก็ได้
3. กลุ่มวรรณยุกต์ (Tone Mark)
4. กลุ่มตัวเลข (Number)
5. กลุ่มอักขระพิเศษ (Special character)

การตัดพยางค์ของวิธีการนี้ ตัดพยางค์จากขวามาซ้าย ด้วยกฎต่างๆ ที่สร้างขึ้น ซึ่งถูกเก็บไว้ในรหัสต้นฉบับ (Source code) ทำให้การแก้ไขหรือเพิ่มเติมกฎต่างๆ ทำได้ไม่สะดวกนัก วิธีการนี้สามารถให้ผลการตัดพยางค์ที่มีความถูกต้องไม่น้อยกว่า 85%

2.1.2. งานของ สุรินทร์ จรรยาพรพงษ์

Surin Chamyapornpong (1993) เสนองานวิจัยเกี่ยวกับการตัดพยางค์ ด้วยหลักไวยากรณ์ภาษาไทย ลักษณะของกฎที่สร้างขึ้นในงานวิจัยนี้ แบ่งออกเป็น 2 ชนิด คือ กฎการหาขอบเขตหน้า (Front boundary recognition rule) และกฎการหาขอบเขตหลัง (Tail boundary recoguition rule) และในแต่ละชนิดแบ่งออกเป็นกลุ่มย่อย ได้แก่ กลุ่มเอ (Group A) กฎที่ใช้แบ่งแยกตามคุณลักษณะของตัวอักษร และ กลุ่มบี (Group B) กฎที่ใช้แบ่งแยกตามคุณสมบัติของรูปแบบการใช้สระแต่ละตัว

เนื่องจากลักษณะของตัวอักษรไทย สามารถที่จะเป็นตัวบอกขอบเขตของพยางค์ได้เป็นอย่างดี งานวิจัยนี้จึงนำเอาลักษณะของตัวอักษรมาสร้างเป็นกฎการตัดพยางค์

ตัวอย่างกฎกลุ่มเอ ในการหาขอบเขตหน้าของพยางค์

A-1F : สระต่างๆ เหล่านี้ อะ อา อิ อี อึ อื อุ อู อำ ไม้ไต่คู้ ไม้หันอากาศ และวรรณยุกต์ จะต้องมีพยัญชนะอยู่ข้างหน้าอย่างน้อย 1 ตัวเสมอ

A-2F : สระ เอ แอ ไอ โอ เป็นอักษรตัวแรกของพยางค์ ยกเว้นบางคำ เช่น ขโมย ชโลม ทแยง เป็นต้น

A-3F : สระ ใอ เป็นอักษรตัวแรกเสมอ ไม่มีข้อยกเว้น

ตัวอย่างกฎกลุ่มเอ ในการหาขอบเขตหลังของพยางค์

A-1T : พยัญชนะต่อไปนี้ ศ ณ ญ ษ ฐ ฎ ฏ ฒ ฬ ฌ จะเป็นตัวสะกดเสมอ ยกเว้นพยางค์ต่อไปนี้ ศก ศร ศง ศพ ษก ฐก ฬส ญวน ฌรงค์ ศตวรรษ ฯลฯ แต่พยางค์เหล่านี้ สามารถจัดการได้โดยใช้กฎ A-1F

A-2T : สระ อึ จะมีตัวสะกด 1 ตัวเสมอ ยกเว้น พึ รึ ฮึ

A-3T : ไม้หันอากาศ จะต้องมีตัวสะกดอย่างน้อย 1 ตัวเสมอ

นอกจากลักษณะของตัวอักษรแล้ว คุณสมบัติของรูปแบบการใช้สระแต่ละตัว ก็สามารถที่จะนำมาใช้แบ่งแยกพยางค์ได้

ตัวอย่างกฎกลุ่มบี ในการหาขอบเขตหน้าของพยางค์

B-1T : สระเหล่านี้ ไม้หันอากาศ อึ อื ถ้ามีวรรณยุกต์ ต้องมีตัวสะกด 1 ตัวเสมอ

B-4T : สระ อัว อัย อัวะ อือ เอาะ เอะ แอะ โอะ เอียะ เออะ ไม่ต้องการตัวสะกด

นอกจากกฎทั้ง 2 กลุ่มแล้ว ในงานวิจัยนี้ยังสร้างกฎที่จะจัดการกับพยางค์ที่ไม่ใช่ลักษณะของพยางค์ไทย ซึ่งพยางค์เหล่านั้นอาจมาจากภาษาต่างประเทศ หรือเป็นพยางค์ที่ประกอบด้วยอักขระพิเศษ และจากการวัดผลความถูกต้องในการตัดพยางค์ ซึ่งในงานวิจัยนี้ได้ทำการทดสอบกับเอกสารต่างๆ 10 ชนิด จำนวน 100 เล่ม ผลที่ได้จากการตัดพยางค์โดยวิธีการนี้ให้ความถูกต้องถึง 96%

2.2 การตัดคำโดยการใช้พจนานุกรม

นับว่าเป็นจุดเริ่มต้นของการตัดคำ หลังจากที่มีการสร้างกฎต่างๆ ขึ้นเพื่อตัดพยางค์ และสามารถให้ผลการตัดพยางค์ที่มีความถูกต้องค่อนข้างสูงแล้ว แต่การที่จะตัดคำได้อย่างถูกต้องนั้น กฎการตัดพยางค์เพียงอย่างเดียว ยังไม่สามารถหาขอบเขตของคำที่ถูกต้องได้ จึงได้มีการพัฒนาการ

ตัดคำโดยการใช้พจนานุกรมขึ้น เพื่อให้การตัดคำมีความถูกต้องมากขึ้นกว่าเดิม ซึ่งผลงานวิจัย การตัดคำโดยการใช้พจนานุกรม ได้มีผู้เสนอแนวคิดไว้ ในรูปแบบต่างๆ ดังนี้

2.2.1. งานของยืน ภู่วรวรรณ และ วิวรรธน์ อิ่มอารมณ์

ยืน ภู่วรวรรณ และ วิวรรธน์ อิ่มอารมณ์ ได้เสนองานวิจัย การใช้พจนานุกรมในการแบ่งพยางค์ ซึ่งเป็นงานวิจัยแรกที่ได้มีการนำพจนานุกรมมาใช้ในการตัดพยางค์ โดยการจัดเก็บพยางค์ต่างๆ ไว้ในพจนานุกรม และใช้ร่วมกับกฎไวยากรณ์อีก 18 กฎ เพื่อแก้ปัญหาพยางค์ที่ไม่พบในพจนานุกรม

หลักการทำงานของวิธีการนี้ คือการตรวจสอบสายอักขระที่เข้ามาจากซ้ายไปขวากับพยางค์ที่เก็บไว้ในพจนานุกรม ในกรณีที่ตรวจสอบแล้วพบว่ามีมากกว่า 1 พยางค์ในพจนานุกรม ก็จะเลือกพยางค์ที่ยาวที่สุด แล้วทำต่อไปจนจบสายอักขระ ถ้าในกรณีที่เลือกพยางค์ที่ยาวที่สุดแล้วทำให้ส่วนที่เหลือเกิดเป็นพยางค์ที่ไม่ปรากฏในพจนานุกรม ก็ยอมให้มีการย้อนรอย (Back Tracking) กลับไปเลือกพยางค์ที่มีความยาวรองลงมาแทน ซึ่งวิธีการนี้รู้จักกันในชื่อของ การตัดคำ (พยางค์) และเลือกคำ (พยางค์) ที่ยาวที่สุด (Longest Matching)

การตัดคำด้วยวิธีการนี้ให้ความถูกต้องถึง 99% แต่ต้องเสียเนื้อที่จำนวนหนึ่งในการจัดเก็บพจนานุกรมพยางค์

2.2.2. งานของ ดร. ดวงแก้ว สวามิภักดิ์

ดร. ดวงแก้ว สวามิภักดิ์ (2533) เสนองานวิจัย เกี่ยวกับซอฟต์แวร์วิเคราะห์ไวยากรณ์ภาษาไทยภายใต้ระบบยูนิกซ์ เป็นงานวิจัยที่ใช้กฎไวยากรณ์ร่วมกันกับพจนานุกรม เพื่อแก้ปัญหากรณีที่พบพยางค์ที่ไม่ปรากฏอยู่ในพจนานุกรม โดยกฎต่างๆที่สร้างขึ้นถูกทำให้อยู่ในรูปของนิพจน์ที่มีกฎเกณฑ์ ประกอบด้วย 43 กฎ ซึ่งถูกใช้ในการจัดการตัดคำโดยโปรแกรมเล็กซ์ (Lex) หลังจากที่พิจารณาตามกฎที่สร้างขึ้นแล้ว ก็จะทำการตรวจสอบจากพจนานุกรมอีกที

พจนานุกรมที่ใช้ในงานวิจัยนี้ เป็นฐานข้อมูลแบบรีเลชั่น (Relational DBMS) ซึ่งใช้คำเป็นดรรชนี (Index) และไฟล์ดรรชนีได้พัฒนาขึ้นโดยใช้โครงสร้างข้อมูลแบบบีทรี (B-Tree)

ผลจากการวัดความถูกต้องสำหรับการตัดคำ และตัดพยางค์ โดยการทดสอบกับเอกสาร 17 ชนิด วิธีการนี้สามารถตัดคำได้ถูกต้อง 98.11% และตัดพยางค์ได้ถูกต้อง 99.67%

2.2.3. งานของสัมพันธ์ ระรื่นรมย์

สัมพันธ์ ระรื่นรมย์ (2534) เสนองานวิจัย การตัดคำโดยใช้พจนานุกรมเพียงอย่างเดียว โดยเน้นที่การเพิ่มประสิทธิภาพด้านความรวดเร็วในการตัดคำ และลดขนาดของพจนานุกรม

ขั้นตอนการทำงานในการพิจารณาเลือกคำจากพจนานุกรม ทำด้วยวิธี Longest Matching ของ ยืน ภู่วรวรรณ แต่เปลี่ยนการจัดเก็บพจนานุกรมจากที่จัดเก็บพยางค์ มาจัดเก็บคำแทน ทำให้พจนานุกรมมีขนาดใหญ่ขึ้น เพื่อให้ขนาดในการจัดเก็บพจนานุกรมมีขนาดเล็กลง และสามารถสืบค้นคำได้รวดเร็วขึ้น ในงานวิจัยนี้จึงจัดเก็บพจนานุกรมด้วยโครงสร้างแบบทรัย (Trie) เนื่องจากเป็นโครงสร้างที่ใช้เนื้อที่ในการจัดเก็บน้อย และใช้พอยเตอร์เชื่อมโยงข้อมูลในการจัดเก็บ จึงสามารถสืบค้นข้อมูลได้อย่างรวดเร็ว (รายละเอียดของโครงสร้างแบบทรัย อธิบายไว้ในบทที่ 3)

ผลจากการทดสอบการตัดคำ โดยใช้พจนานุกรมที่มีโครงสร้างแบบทรัย ทำให้การทำงานรวดเร็วขึ้น สืบค้นคำได้ง่ายขึ้น

2.2.4. งานของ ดร. วิรัช ศรเลิศล้ำวานิช

ดร. วิรัช ศรเลิศล้ำวานิช (2536) เสนองานวิจัย การตัดคำที่เลือกแบบที่เหมือนที่สุด (Maximal Matching) วิธีการนี้สามารถแก้ไขความบกพร่องของการตัดคำแบบ Longest Matching ในกรณีที่เลือกคำที่ยาวเกินไปในช่วงแรก ทำให้ส่วนที่เหลือถูกแบ่งออกเป็นคำที่ไม่เหมาะสม เช่น

ข้อความ ไปหามเหสี

ตัดคำได้ 2 แบบ ไป หาม เห สี และ ไป หา มเหสี

Longest Matching ให้ผลเป็น ไป หาม เห สี

Maximal Matching ให้ผลที่ถูกต้อง คือ ไป หา มเหสี

หลักการของ Maximal Matching คือ ทำการตัดคำในข้อความให้ได้ทุกรูปแบบก่อน แล้วเลือกประโยคที่มีจำนวนคำน้อยที่สุด ดังตัวอย่างข้างต้น จะเห็นว่า Maximal Matching เลือกการตัดคำที่ให้ผลเป็นจำนวนคำที่น้อยกว่า แต่ถ้าสามารถตัดได้จำนวนคำเท่ากันในแต่ละรูปแบบก็จะเลือกรูปแบบที่ตัดคำได้ยาวกว่า เป็นคำตอบ โดยความยาวของคำนั้น เทียบกันระหว่างคำที่มีลำดับเดียวกัน

2.2.5. งานของ สิงห์ ตรงงาม

สิงห์ ตรงงาม (2540) เสนอวิธีการตัดคำโดยใช้พจนานุกรมในแบบที่พิจารณา คำจากขวามาซ้าย วิธีการนี้ถูกเรียกว่า การตัดคำแบบสมบูรณ์ (Complete Matching) มีการทำตารางเก็บความยาวสูงสุดของคำที่สามารถเป็นได้ โดยใช้อักขระ 2 ตัวที่ขึ้นต้นคำเป็นดรรชนีในการค้นหา หลักการทำงานจะค้นหาความยาวสูงสุดที่เป็นได้ของคำที่ขึ้นต้นด้วย อักขระ 2 ตัวที่ตรงกับข้อความก่อน แล้วตัดข้อความออกตามความยาวที่ได้มา จากนั้นนำข้อความที่ตัดมาไปทำการเปรียบเทียบกับคำในพจนานุกรม ถ้าไม่เจอก็ทำการลดความยาวของข้อความที่นำไปเปรียบเทียบลงทีละ ตัวอักษร

จนกระทั่งเป็นคำที่ตรงกับในพจนานุกรม สำหรับส่วนของข้อความที่เหลือจากการพิจารณาในครั้งแรก ก็จะถูกพิจารณาด้วยวิธีการเดียวกันจนกระทั่งจบข้อความ (ดูรายละเอียดได้จากภาคผนวก)

2.3 การตัดคำโดยการใช้ค่าทางสถิติ

หลังจากที่มีการพัฒนาการตัดคำโดยการใช้กฎ ก็มีการวิวัฒนาการต่อเนื่องมาเรื่อยๆ เพื่อให้การตัดคำมีประสิทธิภาพมากขึ้น การนำพจนานุกรมมาใช้ในการตัดคำทำให้การตัดคำมีความถูกต้องมากยิ่งขึ้น แต่ก็ยังมีข้อความบางส่วน ที่มีความกำกวมอยู่ภายในทำให้การตัดคำที่ใช้เพียงแค่พจนานุกรม หรือ กฎ เพียงอย่างเดียวไม่สามารถตัดคำได้ถูกต้อง จึงได้มีแนวคิดที่จะนำ ค่าทางสถิติการเกิดร่วมกันของคำต่างๆ จากคลังข้อความ Corpus มาช่วยในการแก้ปัญหาความกำกวมในข้อความ

2.3.1. งานของวิรัช ศรเลิศล้ำวานิช และ โฮซูมิ ทานากะ

Virach Sornlertlumvanich and Hozumi Tanaka (1995) เสนอแนวความคิด ในการนำเอาค่าทางสถิติ ของการเรียงกันของตัวอักษรที่เป็นคำอย่างถูกต้อง มาช่วยในการพิจารณา ด้วยเชื่อว่า การเรียงกันของตัวอักษรที่ถูกต้องมีความน่าจะเป็นในการที่จะปรากฏอยู่ในตำแหน่งต่างๆ ของเอกสารบ่อยครั้งกว่าการเรียงกันของตัวอักษรที่ไม่สามารถอ่านได้ โดยในวิธีนี้จะสร้าง n-gram ขนาดประมาณ 20-gram ขึ้น เพื่อสังเกตการเปลี่ยนแปลงที่เกิดขึ้นกับค่าทางสถิติเมื่อ n-gram ของตัวอักษรมีขนาดเปลี่ยนไป ซึ่งการขยายขนาดทางซ้าย และ ทางขวา ทำโดยเพิ่มขนาดขึ้นทีละ 1 cluster (cluster คือหน่วยที่เล็กที่สุดในการสะกดคำ เช่น ไม้หันอากาศจะต้องตามด้วยตัวสะกดเสมอ จะอยู่ลอยๆไม่ได้) แล้วทำการกำหนดขอบเขตของคำโดยใช้วิธีการ competitive selection และ unified selection

ขั้นตอนการเตรียม n-gram เป็นดังนี้

1. กำหนดสัญลักษณ์ที่ตำแหน่งช่องว่างระหว่างอักษร tab และตัวอักษรที่ขึ้นบรรทัดใหม่
2. สร้าง n-gram ตามกฎการสะกดคำไทย
3. ขยายขนาดของ n-gram ทั้งทางซ้าย และทางขวา
4. คำนวณผลต่างของจำนวนการเกิด n-gram เมื่อถูกขยายไปยัง cluster ถัดไป

ตารางที่ 2.1 ผลของการขยาย n-gram ทางขวา

String ที่ถูกขยายไปทางขวา	ความถี่ที่เกิดขึ้น	ความถี่ที่ลดลง
กระท	513	68
กระทร	445	0
กระทรว	445	0
กระทรวง	445	142
กระทรวงกา	303	0
กระทรวงการ	303	22
กระทรวงการค	281	0
กระทรวงการคลัง	281	274
กระทรวงการคลังกำ	7	0

ตารางที่ 2.2 ผลของการขยาย n-gram ทางซ้าย

String ที่ถูกขยายไปทางซ้าย	ความถี่ที่เกิดขึ้น	ความถี่ที่ลดลง
การกระทรวง	172	0
ว่าการกระทรวง	172	0
รีว่าการกระทรวง	172	42
ตรีว่าการกระทรวง	130	9
นตรีว่าการกระทรวง	121	0
มนตรีว่าการกระทรวง	121	7
รัฐมนตรีว่าการกระทรวง	114	107
งรัฐมนตรีว่าการกระทรวง	7	0

การค้นหาคำที่ถูกต้อง โดยใช้ Competitive selection พิจารณาเลือก n-gram ที่มีจำนวนการเกิดที่แตกต่างกันมากๆจาก n-gram ที่ขยาย cluster เพิ่มขึ้นเพียง 1 cluster ทั้งทางซ้ายและทางขวา แล้ว n-gram ที่เลือกนั้นต้องสามารถอ่านได้ด้วย จากนั้นใช้ unified selection ในการรวม n-gram ทั้ง 2 เข้าด้วยกันเพื่อกำหนดขอบเขตทั้งซ้าย และขวาของสายตัวอักษร

จากตัวอย่าง ตารางที่ 2.1 และ ตารางที่ 2.2 ใช้ Competitive sclection พิจารณาเลือก n-gram สำหรับการขยายขนาดทางขวาได้ "กระทรวงการคลัง" และ การขยายทางซ้ายได้ "รัฐมนตรีว่าการกระทรวง" จากนั้นใช้ unified selection ในการรวม n-gram ทั้ง 2 เข้าด้วยกันเพื่อกำหนดขอบเขตทั้งซ้าย และขวาได้ "รัฐมนตรีว่าการกระทรวงการคลัง"

2.3.2. งานของ อัศนีย์ ก่อตระกูล และคณะ

Asanee Kawtrakul et al. (1997) นำโมเดลไตรแกรมมาช่วยในการแก้ปัญหาการตัดคำ และการกำหนดหน้าที่คำ ในงานวิจัยเรื่อง "A Statistical Approach to Thai Word Filtering" การคำนวณหาค่าความน่าจะเป็นของประโยคสามารถทำได้ดังสมการ

$$P(W) = \prod_{i=1}^{n} P(w_{i,n}) = \prod_{i=1}^{n} P(w_i \mid w_{i-1}, w_{i-2}) \quad (2\text{-}1)$$

W คือประโยคที่ตัดคำแล้ว ซึ่งประกอบด้วยคำต่างๆ $W = w_1\ w_2 \ldots w_n$ โดยที่ w_i คือ คำศัพท์ และค่าความน่าจะเป็นของ w_i ขึ้นกับ w_{i-1} และ w_{i-2} เท่านั้น

แต่เนื่องจากการคำนวณค่าความน่าจะเป็นตามสมการ 2-1 ต้องใช้คลังข้อความขนาดใหญ่มาก ที่มีการเก็บข้อความไว้อย่างน้อยมากกว่า n^3 คำ โดยที่ n คือ จำนวนคำที่เป็นไปได้ทั้งหมด สาเหตุที่ต้องใช้คลังข้อความขนาดใหญ่ เนื่องจากต้องการค่าสถิติของการเกิดร่วมกันของคำ 3 คำแบบติดกันมาใช้ในการคำนวณ ดังนั้นเพื่อให้มีค่าสถิติของการเกิดคำ 3 คำที่ติดกันครบทุกแบบ อย่างน้อยที่สุดจะต้องใช้ n^3 คำ ซึ่งเป็นเรื่องยากที่จะมีคลังข้อมูลใหญ่ได้ขนาดนั้น จึงมีการประมาณสมการ 2-1 มาเป็น สมการ 2-2 ดังนี้

$$\prod_{i=1}^{n} P(w_i \mid w_{i-2}, w_{i-1}) = \prod_{i=1}^{n} (\lambda_1 P(w_n) + \lambda_2 P(w_n \mid w_{n-1}) + \lambda_3 P(w_n \mid w_{n-1}, w_{n-2})) \quad (2\text{-}2)$$

จากสมการที่ 2-2 สามารถแก้ปัญหาเรื่องจำนวนข้อมูลที่ต้องการใช้ให้ลดลงได้ โดยใช้ค่าความน่าจะเป็นของไบแกรม (Bi-gram) และ ยูนิแกรม (Uni-gram) มาช่วยในการคำนวณ และให้ค่า $\lambda_1, \lambda_2, \lambda_3$ ให้มีค่าเท่ากับ 0.1, 0.3, 0.6 ตามลำดับ ซึ่งได้มาจาก (Charniak, 1996)

ผลจากการวิจัยนี้สามารถลดรูปแบบการตัดคำที่ไม่เหมาะสมลงได้จำนวนมาก ส่งผลให้งานการวิเคราะห์หน่วยคำสามารถทำงานได้รวดเร็วขึ้น

2.3.3. งานของ สุรพันธ์ เมฆนาวิน และคณะ

Surapant Meknavin, Paisarn Charoenpornsawat and Boonserm Kijsirikul (1997) งานวิจัยเสนอการตัดคำโดยใช้หน้าที่คำแบบไตรแกรมโมเดล เป็นการนำเอาค่าความน่าจะเป็นของการเกิดหน้าที่คำอย่างต่อเนื่อง มาใช้ในการพิจารณาเลือกรูปแบบการตัดคำที่ดีที่สุด โดยเลือกจากประโยคที่มีค่าความน่าสูงที่สุดเป็นคำตอบ โดยค่าความน่าจะเป็นดังกล่าวหาได้จากสมการที่ 2-3

$$P(W_i) = \sum_T P(W_i, T_i) \qquad (2\text{-}3)$$
$$= \sum_T \prod_i P(t_i \mid t_{i-1}, t_{i-2}) x P(w_i, t_i)$$

จากสมการ W_i คือประโยคที่ตัดคำแล้ว ซึ่งนำมาจากประโยคที่มีคะแนนดีที่สุด N อันดับแรก โดยวิธีการตัดคำจะใช้การตัดคำแบบเหมือนที่สุด และ $W_i = w_1 w_2 ... w_n$ โดย w_i คือ คำที่ตัดได้ $T_i = t_1 t_2 ... t_n$ โดย t_i คือ หน้าที่คำ ของ w_i และ $P(w_i \mid t_i)$ กับ $P(t_i \mid t_{i-1}, t_{i-2})$ สามารถคำนวณได้จากคลังข้อความ ผลของสมการนี้ทำให้สามารถเลือกรูปแบบการตัดคำที่ดีที่สุด จากการพิจารณาผลรวมความน่าจะเป็นของหน้าที่คำทุกแบบที่เป็นไปได้ของแต่ละประโยค โดยหน้าที่คำของคำปัจจุบันขึ้นกับหน้าที่คำของ 2 คำก่อนหน้า ในวิธีการนี้ไม่ได้สนใจว่าหน้าที่คำที่ถูกต้องที่สุดสำหรับคำนั้นจะเป็นอะไร แต่สนใจว่าจะตัดคำอย่างไรจึงจะถูกต้องเท่านั้น ดังนั้นถ้าข้อความกำกวมที่เกิดขึ้นในข้อความ มีหน้าที่คำเหมือนกันก็จะไม่สามารถแก้ปัญหานี้ได้

2.3.4. การตัดคำในภาษาญี่ปุ่น (อักษรคันญิ)

Lillian Lee and Rie Ando (1999) เสนอวิธีการวิเคราะห์การหาขอบเขตของคำโดยการอาศัยหลักความจริงที่ว่าตัวอักษรที่ประกอบเป็นคำเดียวกันนั้นย่อมมีแรงยึดเหนี่ยวระหว่างกันมากกว่าตัวอักษรที่ไม่ใช่ส่วนประกอบของคำเดียวกัน ในวิธีการนี้ใช้การนับจำนวนของ n-gram ในการตัดสินใจตัดคำ n-gram ที่คร่อมระหว่างตัวอักษรของคำที่ต่างกัน จะมีความน่าจะเป็นในการเกิดขึ้นในเอกสารน้อยกว่า n-gram ที่คร่อมตัวอักษรของคำเดียวกันเอาไว้ เช่น ถ้าพิจารณาตัวอักษรที่เรียงต่อกัน 8 ตัว

A B C D W X Y Z

ที่ตำแหน่งที่ 4 สร้าง 3-gram ขึ้นที่บริเวณใกล้เคียงกับตำแหน่งนั้นได้ 3-gram ของ B C D และ W X Y ซึ่งเกิดขึ้นในเอกสารบ่อยครั้งกว่า C D W ที่แทบจะไม่เคยเกิดขึ้นเลย สิ่งนี้เป็นหลักฐานแสดงให้เห็นว่า ขอบเขตของคำน่าจะอยู่ระหว่าง D และ W เพราะดูเหมือนว่าตัวอักษรทั้ง 2 จะมีแรงยึดเหนี่ยวกันน้อยมาก ในการตัดคำโดยใช้วิธีการนี้กำหนดให้ N เป็นกลุ่มของ n-gram ที่จะใช้เป็นหลัก

ฐานในการตัดสินใจทำการตัดคำ ตัวอย่างเช่น กำหนดให้ N = {2} แล้ววาง #(s) เพื่อแสดงจำนวนครั้งในการเกิด n-gram s ขึ้นใน training corpus ที่ยังไม่มีการตัดคำ ซึ่งจะเสนอคำถามตามนี้คือ

Is #(C D) > #(D W) ?

Is #(W X) > #(D W) ?

โดยที่คำตอบ Yes จะเป็นหลักฐานที่แสดงว่าขอบเขตของคำอยู่ระหว่าง D กับ W ถ้ากำหนดให้ N = {2,4} เราก็จะมีคำถามเพิ่มขึ้นอีก 6 คำถามคือ

Is #(A B C D) > #(B C D W) ?

Is #(A B C D) > #(C D W X) ?

Is #(A B C D) > #(D W X Y) ?

Is #(W X Y Z) > #(B C D W) ?

Is #(W X Y Z) > #(C D W X) ?

Is #(W X Y Z) > #(D W X Y) ?

คำถามเหล่านี้จะเพิ่มขึ้นเมื่อขนาดของ n-gram เพิ่มขึ้น เราคำนวณหาค่าเฉลี่ยของคำตอบในแต่ละตำแหน่งที่ได้จากคำถามที่เสนอขึ้นของ n-gram แล้วนำค่าที่ได้มาหาค่าเฉลี่ยอีกครั้ง โดยเฉลี่ยจากค่าที่ได้ของตำแหน่งนั้นในแต่ละชุดของ n-gram ของ N แล้วสร้างขอบเขตของคำขึ้น เมื่อค่าเฉลี่ยของคำตอบที่ได้

- มากกว่าของทั้ง 2 ตำแหน่งที่อยู่ติดกันทางซ้าย และ ขวา หรือ
- มากกว่าค่า threshold ที่กำหนด (ถ้าค่า threshold มีค่ามากคำที่สร้างขึ้นจะมีขนาดยาว)

ถ้าให้ $g_n(x)$ แทน n-gram ที่สร้างขึ้นที่ตำแหน่งที่ x เช่น $g_4(2)$ = " B C D W" (จากตัวอย่างข้างต้น) และ $I_>(y,z)$ แทนคำถามที่ถูกเสนอ โดยจะมีค่าเท่ากับ 1 เมื่อ y > z และเป็น 0 ถ้า y !> z สำหรับแต่ละตำแหน่งที่ i ซึ่งสามารถคำนวณหา $v_n(i)$ ที่เป็นค่าเฉลี่ยของหลักฐานที่ได้จากการตอบคำถามเกี่ยวกับการเรียงกันของตัวอักษร n ตัว อย่างต่อเนื่อง ที่ตำแหน่งที่ i ได้ว่า

$$v_n(i) = \frac{1}{(2n-2)} \sum_{k=i-n+2}^{i} (I_>(\#(g_n(i-n+1)), \#(g_n(k))) + I_>(\#(g_n(i+1)), \#(g_n(k))) \quad (2\text{-}4)$$

จากนั้นนำมาหาค่าเฉลี่ย $v_N(i)$ ที่เกิดจากการตอบคำถามที่ตำแหน่งที่ i ของแต่ละ n-gram ที่อยู่ใน N ดังนี้

$$v_N(i) = \sum_{n \in N} \frac{v_n(i)}{|N|} \quad (2\text{-}5)$$

หลังจากคำนวณ $v_N(i)$ ที่แต่ละตำแหน่งที่ i ของตัวอักษรที่เรียงกันแล้ว ขอบเขตของคำจะถูกกำหนดที่ตำแหน่ง ℓ ซึ่ง

- $v_N(\ell) > v_N(\ell-1)$ และ $v_N(\ell) > v_N(\ell+1)$ หรือ
- $v_N(\ell) >$ t , (ค่า threshold)

ผลการวิเคราะห์ขอบเขตของคำโดยวิธีการนี้ ทำได้ดีกว่าวิธีเดิมๆที่ใช้การวิเคราะห์โครงร่างและส่วนประกอบของคำที่มีอยู่ในภาษาญี่ปุ่น แต่การประมวลผลยังคงต้องใช้เวลาพอสมควร เนื่องจากต้องการการประมวลค่อนข้างสูง

2.4 การตัดคำโดยการใช้คุณลักษณะ

การตัดคำโดยการใช้คุณลักษณะ เป็นวิธีที่เรียนแบบ การเรียนรู้ของมนุษย์ มีการเก็บรวบรวมความรู้ที่ได้จากการสังเกต คุณลักษณะของคำที่เกิดขึ้นรอบข้างข้อความกำกวม แล้วใช้ความรู้นี้ในการแบ่งแยกความแตกต่าง ของรูปแบบการตัดคำ แต่ละรูปแบบที่เป็นไปได้ของข้อความนั้น

2.4.1 งานของ ไพศาล เจริญพรสวัสดิ์

ไพศาล เจริญพรสวัสดิ์ (2541) เสนอวิธีการตัดคำโดยใช้คุณลักษณะ แบ่งแยกความแตกต่าง ของแต่ละรูปแบบการตัดคำ ที่เป็นไปได้ของข้อความกำกวม โดยการสร้างโครงข่ายวินโนว์ (Winnow) ขึ้น เพื่อเก็บรวบรวมความรู้ที่ได้จากการสังเกต คุณลักษณะของคำที่เกิดขึ้นรอบๆ ข้อความกำกวม เมื่อข้อความนั้นถูกแบ่งแยกไว้ ในลักษณะต่างๆ

หลังจากเก็บรวบรวมความรู้ไว้แล้ว จะนำความรู้นี้ไปใช้ในการแบ่งแยกความกำกวมที่เกิดขึ้น โดยพิจารณาว่ารอบข้างข้อความกำกวม มีคุณลักษณะใดบ้างที่ตรงกับความรู้ที่เก็บไว้ ของแต่ละรูปแบบการตัดคำที่สามารถเป็นไปได้ แล้วเลือกรูปแบบการตัดคำ ที่มีการเกิดร่วมกับคุณลักษณะที่นำมาพิจารณามากที่สุดเป็นคำตอบ

วิธีการที่ได้เสนอไว้เกี่ยวกับการแก้ปัญหาความกำกวมของข้อความ โดยการใช้คุณลักษณะนั้น มีอยู่ 2 วิธีด้วยกันคือ การพิจารณาโดยการใช้เซตของข้อความส่วนหน้า (prefix set) และเซตของความสับสน (confusion set) ทั้ง 2 วิธีมีข้อดีข้อเสียที่แตกต่างกัน ซึ่งจะอธิบายต่อไป

2.4.1.1. การพิจารณาความกำกวมที่เกิดขึ้นในข้อความ แบ่งออกเป็น 2 กลุ่ม คือ

2.4.1.1.1. ข้อความกำกวมที่ขึ้นกับบริบท

ข้อความประเภทนี้ต้องใช้บริบท (context word) และ การเกิดร่วมกันอย่างเป็นลำดับ (word collocation) ในการพิจารณาเลือกความหมาย เนื่องจากข้อความประเภทนี้มักตัดคำได้มากกว่า 1 รูปแบบ และ แต่ละรูปแบบต่างก็มีความหมายต่างกันออกไป

ตัวอย่าง ข้อความ "ตากลม" ข้อความกลุ่มนี้หากไม่มีคุณลักษณะมาช่วยในการพิจารณาแล้ว เชื่อว่าแม้นแต่มนุษย์ที่เป็นเจ้าของภาษาเองก็ยากจะบอกได้ว่า "ตากลม ที่ปรากฏขึ้นมาลอยๆ นั้น ควรจะแบ่งแยกคำเป็นอย่างไร จึงจะถูกต้อง ระหว่าง "ตา" กับ "กลม" หรือ "ตาก" กับ "ลม"

แต่เมื่อมีบริบทขึ้นมาประกอบรอบข้าง มนุษย์ก็จะสามารถบอกได้ทันทีว่ามันควรจะแบ่งแยกเป็นคำว่าอย่างไร เนื่องจากมนุษย์เรามีฐานความรู้ และประสบการณ์เกี่ยวกับคำนั้น เก็บไว้ในความทรงจำของแต่ละคน ซึ่งถ้าหากให้เด็กเล็กๆ มาอ่าน ก็อาจจะอ่านผิดได้ เนื่องจากเขาไม่มีประสบการณ์กับคำนั้นมากพอ

2.4.1.1.2. ข้อความกำกวมที่ไม่ขึ้นกับบริบท

ข้อความประเภทนี้ สามารถแบ่งแยกความกำกวม และเลือกรูปแบบการตัดคำที่เหมาะสมได้ โดยไม่ต้องอาศัยบริบท (context word) หรือ การเกิดร่วมกันอย่างเป็นลำดับของคำ (word collocation) มาช่วยในการตัดสินใจ เนื่องจากข้อความประเภทนี้ ถึงแม้นว่าจะสามารถตัดคำได้หลายรูปแบบ แต่ก็จะมีเพียงรูปแบบเดียวเท่านั้นที่มีความหมาย

ตัวอย่างเช่น ข้อความ "ไปหามเหลี" สามารถตัดคำได้มากกว่า 1 แบบ ได้แก่ "ไป" "หา" "มเหลี" และ "ไป" "หาม" "เห" "ลี" แต่มีรูปแบบเดียวเท่านั้นที่มีความหมาย ในที่นี้คือ "ไป" "หา" "มเหลี"

2.4.1.2. กลุ่มของความกำกวม

เมื่อการตัดคำสามารถตัดคำในข้อความสามารถทำได้มากกว่า 1 รูปแบบ รูปแบบต่างๆ เหล่านั้นจะถูกรวบรวมไว้ สร้างเป็นกลุ่มของความกำกวม เพื่อนำมาวิเคราะห์หาความแตกต่างของแต่ละรูปแบบของการตัดคำที่เป็นไปได้ที่ได้จากการตัดคำเบื้องต้น ไม่ว่าข้อความกำกวมนั้นจะเป็นความกำกวมแบบขึ้นกับบริบท หรือไม่ขึ้นกับบริบท

การพิจารณาลักษณะของความกำกวมที่เกิดขึ้นในข้อความ สามารถทำการพิจารณาได้ 2 วิธีด้วยกันคือ พิจารณาเป็นกลุ่มของข้อความส่วนหน้า (prefix set) และพิจารณาเป็นกลุ่มของความสับสน (confusion set) ซึ่งทั้ง 2 วิธีจะมีลักษณะในการพิจารณาที่ต่างกัน

2.4.1.2.1. กลุ่มของข้อความส่วนหน้าของข้อความกำกวม

ลักษณะของกลุ่มนี้ เป็นกลุ่มของความกำกวมที่ระบุคำส่วนหน้าที่มีขนาดเล็กกว่าขนาดของข้อความเอาไว้ เรียกว่า เซตของข้อความส่วนหน้า (prefix set) ซึ่งสมาชิกของเซต

ประกอบไปด้วยทุกคำที่มีขนาดเล็กกว่าข้อความกำกวม และเป็นคำแฝงอยู่ในส่วนหน้าของข้อความกำกวมนั้น เช่น เซตของข้อความ "มาก" จะประกอบไปด้วยสมาชิกภายในเซต คือ มา และ มาก

$$P_{\text{มาก}} = \{\text{มา, มาก}\}$$
$$P_{\text{มากมาย}} = \{\text{มา, มาก, มากมาย}\}$$
$$P_{\text{ตาก}} = \{\text{ตา, ตาก}\}$$

รูปที่ 2.1 ตัวอย่าง เซตของข้อความส่วนหน้า (prefix set)

ข้อดีของการพิจารณาความกำกวมด้วยวิธีนี้ คือ สามารถสร้างได้คลอบคลุมกับข้อความกำกวมทุกข้อความทั้งที่มี และไม่มีในพจนานุกรม

ข้อเสีย คือ สมาชิกแต่ละตัวที่อยู่ภายในเซตเดียวกัน ต้องเก็บคุณลักษณะจำนวนมาก ที่จะใช้ในการแบ่งแยกความแตกต่างกับสมาชิกตัวอื่น เนื่องจาก ในการพิจารณาข้อความกำกวมไม่ว่าจะเป็น "มากว่า" หรือ "มากรอง" หรือ "มากราบ" หรือ "มากลับ" หรือ ฯลฯ ต่างก็ต้องการใช้เซตของ $P_{\text{มาก}}$ ในการพิจารณาว่าคำแรกควรจะเป็น "มา" หรือ "มาก" ด้วยกันทั้งนั้น เพราะฉะนั้นสมาชิกที่อยู่ภายในเซตทั้ง "มา" และ "มาก" จึงต้องมีการเก็บรวบรวมคุณลักษณะเฉพาะไว้จำนวนมากเพื่อที่จะแบ่งแยกให้ได้ว่า คำแรกควรจะเป็น "มา" หรือ "มาก" ไม่ว่าข้อความที่พิจารณาจะเป็นความกำกวมข้อความใดก็ตาม ซึ่งเป็นเรื่องยากในการที่จะเก็บคุณลักษณะให้คลอบคลุมกับข้อความกำกวมทุกข้อความ

2.4.1.2.2. กลุ่มของรูปแบบที่ได้จากการตัดคำของข้อความกำกวม

ลักษณะของกลุ่มนี้ เป็นกลุ่มของข้อความกำกวมที่ประกอบไปด้วยรูปแบบที่สามารถตัดคำได้ทุกรูปแบบ เรียกว่า **เซตของความสับสน (confusion set)** เช่น เซตของข้อความ "มากว่า" จะประกอบไปด้วยสมาชิกภายในเซต คือ "มา" "กว่า" และ "มาก" "ว่า"

$C_{\text{มากว่า}}$ = {มา กว่า, มาก ว่า}

$C_{\text{มากรอง}}$ = {มา กรอง, มาก รอง}

$C_{\text{ตากลม}}$ = {ตา กลม, ตาก ลม}

รูปที่ 2.2 ตัวอย่าง เซตของข้อความสับสน (confusion set)

ข้อดีของวิธีนี้คือ สามารถให้การวิเคราะห์ที่เด่นชัดระหว่างแต่ละรูปแบบของความกำกวม ที่อยู่ภายในเซตเดียวกัน การวิเคราะห์ถูกทำเฉพาะกับข้อความกำกวมนั้นๆ การเก็บคุณลักษณะเฉพาะของสมาชิกแต่ละตัวจึงทำได้ง่าย และชัดเจน เนื่องจากเป็นการเก็บคุณลักษณะเป็นการเก็บเพื่อรูปแบบของการแบ่งแยกคำภายในข้อความกำกวมนั้นเพียงข้อความเดียว

ข้อเสีย คือ การสร้างเซตด้วยวิธีการนี้ ให้ครอบคลุมกับปัญหาของความกำกวมทั้งหมดที่จะเกิดขึ้นทำได้ยาก เนื่องจากข้อความกำกวมใหม่ๆ อาจเกิดขึ้นได้เสมอ

2.4.1.3. คุณลักษณะที่ใช้ในการพิจารณา

คุณลักษณะ (feature) ที่นำมาใช้ในการพิจารณาเพื่อเลือกรูปแบบการตัดคำที่เหมาะสม แบ่งออกเป็น 2 ประเภทใหญ่ๆ ได้แก่

2.4.1.3.1. บริบทรอบข้างของคำเป้าหมาย (context word)

แบ่งออก เป็น บริบทส่วนหน้า และบริบทส่วนหลัง

2.4.1.3.2. สิ่งที่เกิดร่วมกันอย่างมีลำดับของคำเป้าหมาย (collocation)

แบ่งออกเป็น ลำดับการเกิดร่วมกันของคำ และ ลำดับการเกิดร่วมกันของหน้าที่คำ

ซึ่งคุณลักษณะ (feature) ที่นำมาพิจารณามี 10 คุณลักษณะด้วยกัน คือ บริบทส่วนหน้า และ บริบทส่วนหลังของคำเป้าหมาย คำที่ 1 และ 2 ที่อยู่หน้า และหลังของคำเป้าหมาย หน้าที่ของคำที่ 1 และ 2 ที่อยู่หน้า และหลังของคำเป้าหมาย

บทที่ 3

โครงสร้างพจนานุกรมแบบทรัย

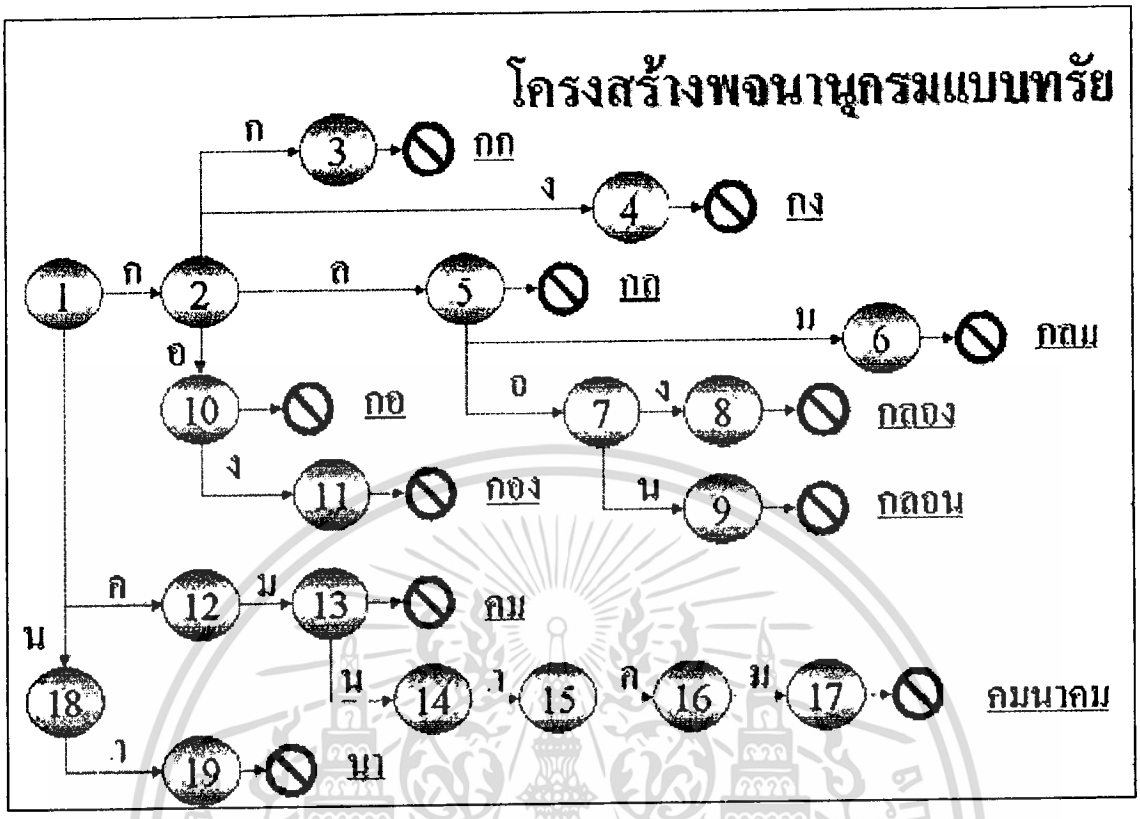
สำหรับการตัดคำโดยการใช้การเปรียบเทียบข้อความกับคำในพจนานุกรมนั้น ต้องการรูปแบบการจัดเก็บพจนานุกรมที่มีประสิทธิภาพ เพื่อให้สามารถคืบค้นคำได้อย่างรวดเร็ว และใช้เนื้อที่ในการจัดเก็บน้อย

โครงสร้างแบบทรัย มีลักษณะคล้ายกับโครงสร้างข้อมูลแบบต้นไม้ แต่ไม่ได้เก็บข้อมูลทั้งคำไว้แบบโครงสร้างต้นไม้ โครงสร้างแบบทรัยจะเก็บการเรียงกันของตัวอักษรในคำศัพท์ ทำให้ลดเนื้อที่ในการจัดเก็บคำที่มีตัวอักษรร่วมกันได้

การจัดเก็บของโครงสร้างแบบทรัย ประกอบด้วยโหนดต่างๆ ในแต่ละโหนดประกอบด้วยพอยเตอร์ชี้ไปยังโหนดของตัวอักษรถัดไป ซึ่งแต่ละโหนดมีจำนวนพอยเตอร์เท่ากับจำนวนตัวอักษรที่เป็นไปได้ ที่มีอยู่ในพจนานุกรม บวกกับตัวระบุจบคำศัพท์ (Terminator) อีก 1 ตัว

การสืบค้นคำจากโครงสร้างแบบทรัย เริ่มจากโหนดที่ 0 แล้วดูว่าตัวอักษรของคำศัพท์ที่ต้องการค้นหามีพอยเตอร์จากโหนด 0 ชี้ไปหรือไม่ ถ้ามีก็วิ่งตามพอยเตอร์ไปหาตัวอักษรในคำศัพท์ทีละตัวจนจบข้อความ แล้วตามด้วยตัวระบุจบคำศัพท์ ถ้าไม่พบตัวจบ ก็แสดงว่าไม่มีคำนั้นอยู่ในพจนานุกรมแบบทรัย

จากรูปที่ 3.1 แสดงตัวอย่าง การจัดเก็บข้อมูลคำศัพท์ ของคำว่า กก กง กล กลม กลอง กลอน กอ กอง คม คมนาคม และ นา ด้วยโครงสร้างแบบทรัย



รูปที่ 3.1 แสดงโครงสร้างข้อมูลแบบทรัย

บทที่ 4

การกำหนดหน้าที่คำ

หน้าที่คำ (Part of Speech: POS) คือ สิ่งที่ระบุว่าคำนั้นทำหน้าที่อะไรภายในประโยค ตามหลักไวยากรณ์ภาษาไทย เช่น เป็นคำนาม คำสรรพนาม กริยา เป็นต้น

การกำหนดหน้าที่คำ เพื่อให้การตัดคำมีประสิทธิภาพมากขึ้น แก้ปัญหาความกำกวมของข้อความที่สามารถแบ่งแยกคำได้มากกว่า 1 รูปแบบ หน้าที่คำจึงเป็นสิ่งจำเป็นอย่างหนึ่ง ในการที่จะบอกว่าข้อความกำกวมนั้นควรจะถูกแบ่งแยกคำ เป็นอย่างไร จึงจะเหมาะสม เนื่องจากคำแต่ละคำ มีความสามารถในการทำหน้าที่ เป็น คำนาม กริยา หรือ ส่วนขยาย ในประโยคได้ต่างกัน

วิธีที่ใช้ในการกำหนดหน้าที่คำ มักใช้ค่าทางสถิติ เนื่องจากวิธีการนี้สามารถ คำนวณหาค่าสถิติหน้าที่ของต่างๆ ได้โดยอัตโนมัติ ทำให้การทำงานสะดวกรวดเร็ว และมีความถูกต้องแม่นยำ ซึ่งค่าทางสถิติที่แสดงความน่าจะเป็นของหน้าที่คำต่างๆ ของแต่ละคำในประโยคหาได้จากสมการที่ 4-1

$$\tau = \max_{c_1,\ldots,c_t} \arg PROB(c_1,\ldots,c_t \mid w_1,\ldots,w_t) \qquad (4\text{-}1)$$

โดยที่ τ คือ $C_p,\ldots,C_t$ ที่ทำให้ค่าความน่าจะเป็นตามสมการที่ 4-1 มีค่ามากที่สุด C_i คือหน้าที่คำของ w_i ส่วน $w_p,\ldots,w_t$ คือลำดับของคำในประโยค และ $C_p,\ldots,C_t$ คือลำดับของหน้าที่คำในประโยค

ในการกำหนดหน้าที่คำ จะเลือกลำดับของหน้าที่คำ $C_p,\ldots,C_t$ ที่ทำให้สมการที่ 4-1 มีค่ามากที่สุด ให้เป็นหน้าที่คำที่เหมาะสมที่สุดของคำ $w_p,\ldots,w_t$ ในแต่ละประโยค แต่การใช้ค่าทางสถิติในการกำหนดหน้าที่คำตามสมการ 4-1 นั้น จำเป็นต้องมีคลังข้อมูลที่มีขนาดใหญ่มาก ซึ่งในความเป็นจริงแล้วเราไม่สามารถที่จะหาคลังข้อมูลที่มีขนาดใหญ่ได้ถึงขนาดนั้น จึงได้มีการนำกฎของเบย์ (Bayes' rule) ดังสมการที่ 4-2 เข้ามาเพื่อปรับปรุงสมการที่ 4-1 มาเป็นสมการที่ 4-3

$$PROB(A \mid B) = \frac{PROB(B \mid A) x PROB(A)}{PROB(B)} \qquad (4\text{-}2)$$

$$\tau = \max_{c_1,\ldots,c_t} \arg \frac{PROB(c_1,\ldots,c_t) x PROB(w_1,\ldots,w_t \mid c_1,\ldots,c_t)}{PROB(w_1,\ldots,w_t)} \quad (4\text{-}3)$$

จากสมการที่ 4-3 เราสามารถละค่าของ $PROB(w_1,\ldots,w_t)$ ได้เนื่องจากเป็นค่าคงที่ สมการที่ 4-3 จึงสามารถลดรูปมาเป็นสมการที่ 4-4

$$\tau = \max_{c_1,\ldots,c_t} \arg PROB(c_1,\ldots,c_t) x PROB(w_1,\ldots,w_t \mid c_1,\ldots,c_t) \quad (4\text{-}4)$$

จากสมการที่ 4-4 ซึ่งลดรูปลงมาแล้ว สามารถหาค่าโดยประมาณได้ จากโมเดลไตรแกรม (Tri-gram) และไบแกรม (Bi-gram) ทำให้การคำนวณทำได้ง่ายขึ้น และความต้องการในการใช้ข้อมูลจากคลังข้อมูลน้อยลง ซึ่งสามารถประมาณค่าส่วนต่างๆ ของสมการที่ 4-4 ได้ดังสมการที่ 4-5 และ 4-6

โดยกำหนดให้ค่าความน่าจะเป็นของหน้าที่คำหนึ่งๆ ขึ้นอยู่กับหน้าที่ของคำ 2 คำก่อนหน้าเท่านั้น

$$PROB(c_1,\ldots,c_t) \cong \prod_{i=1}^{t} PROB(c_i \mid c_{i-1}, c_{i-2}) \quad (4\text{-}5)$$

และค่าความน่าจะเป็นของคำที่ถูกกำหนดให้เป็นหน้าที่ต่างๆ ไม่ขึ้นกับคำรอบข้าง

$$PROB(w_1,\ldots,w_t \mid c_1,\ldots,c_t) \cong \prod_{i=1}^{t} PROB(w_i \mid c_i) \quad (4\text{-}6)$$

จากสมการที่ 4-4 จึงสามารถสรุปเป็นสมการที่ 4-7 ได้ดังนี้

$$\tau = \max_{c_1,\ldots,c_t} \arg \prod_{i=1}^{t} PROB(c_i \mid c_{i-1}, c_{i-2}) x PROB(w_i \mid c_i) \quad (4\text{-}7)$$

ผลจากการประมาณค่าของสมการ ทำให้ได้สมการที่ 4-7 ซึ่งใช้ข้อมูลในการคำนวณน้อยลง แต่การคำนวณหาค่าความน่าจะเป็นตามสมการนี้ ยังคงต้องใช้เวลาในการคำนวณอย่างมาก จึงต้องทำการเพิ่มประสิทธิภาพในการคำนวณ โดยการนำเอาเทคนิค ไดนามิคโปรแกรมมิ่ง (Dynamic Programing) เข้ามาช่วยเพื่อให้ เวลาในการคำนวณลดน้อยลง

ในการกำหนดหน้าที่คำที่ใช้ในโครงการพัฒนานี้ ใช้การกำหนดหน้าที่คำ จากค่าทางสถิติ ที่ใช้วิธีการคำนวณ แบบไดนามิคโปรแกรมมิ่ง ตามขั้นตอนของวิเทอร์บิ (Viterbi Alorithm) (Allen, 1995) ซึ่งมีการสร้าง Array ขนาด $N \times N \times T$ จำนวน 2 ชุด โดย N คือ จำนวนหน้าที่คำที่เป็นไปได้ทั้งหมด และ T คือ จำนวนคำในประโยค ที่จะนำมากำหนดหน้าที่คำ โดยที่ ***seqscore[i][j][t]*** เก็บค่าความน่าจะเป็นของการกำหนดหน้าที่คำที่ดีที่สุดของคำ $w_p \ldots, w_t$ ซึ่งหน้าที่ของคำที่ w_t และ w_{t-1} มีหน้าที่ของคำเป็น L_i และ L_j ตามลำดับ ส่วน ***backptr[i][j][t]*** เก็บหน้าที่คำของคำที่ **t-2** เมื่อคำที่ **t** และ **t-1** มีหน้าที่คำเป็น L_j และ L_i ตามลำดับ รายละเอียดการคำนวณแสดงไว้ตามรูปที่ 4-1

จากการคำนวณตามขั้นตอนของวิเทอร์บิ ทำให้เวลาในการคำนวณลดลงอย่างมาก จาก kN^T มาเป็น N^3T

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

Initialization Step

for i=1 to N do

for j=1 to N do

$$\text{seqscore}[i][j][2] = \text{Prob}(W_1 \mid L_i) \times \text{Prob}(L_i \mid \varnothing) \times \text{Prob}(W_2 \mid L_j) \times \text{Prob}(L_j \mid L_i, \varnothing)$$

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

$$\text{seqscore}[j][k][t] = \text{Max}_{i=1,N}(\text{seqscore}[i][j][t-1] \times \text{Prob}(L_k \mid L_j, L_i)) \times \text{Prob}(W_t \mid L_k)$$

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] มีค่ามากที่สุด

for i=T-2 to 1 do

C[i] = backptr[C[i+1]][C[i+2]][i+2]

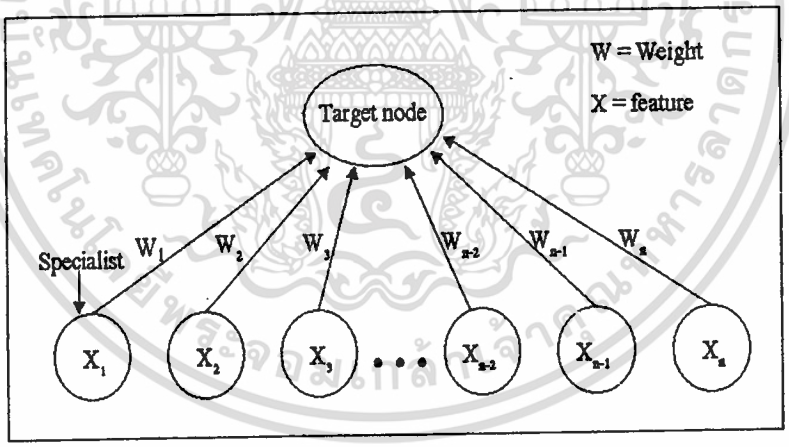
รูปที่ 4.1 ขั้นตอนการคำนวณของวิเทอร์บิ (Viterbi Algorithm)

บทที่ 5

การเรียนรู้ของเครื่องเพื่อแบ่งแยกความกำกวม

เพื่อให้เครื่องคอมพิวเตอร์ สามารถเรียนรู้แล้วเก็บรวบรวมความรู้ที่ได้เกี่ยวกับความแตกต่างของ แต่ละรูปแบบที่ได้จากการตัดคำที่เป็นไปได้ทุกกรณี ว่ามีลักษณะของคำรอบข้างและการเกิดร่วมกันกับคำที่อยู่ข้างเคียง ต่างกันอย่างไร เพื่อใช้ในการเลือกรูปแบบการตัดคำที่เหมาะสมให้กับแต่ละข้อความที่นำมาประมวลผล

วิธีการเรียนรู้ ที่ใช้ในการแบ่งแยกความแตกต่างของรูปแบบที่ได้จากการตัดคำ ด้วยโครงข่ายวินโนว์ (Winnow) (Charoenpornsawat, 1998; Blum, 1997; Golding and Roth, 1996) ซึ่งมีลักษณะคล้ายกับ โครงข่ายใยประสาท มาพิจารณาในการแบ่งแยกคำ เพื่อเลือกรูปแบบการตัดคำที่ถูกต้อง



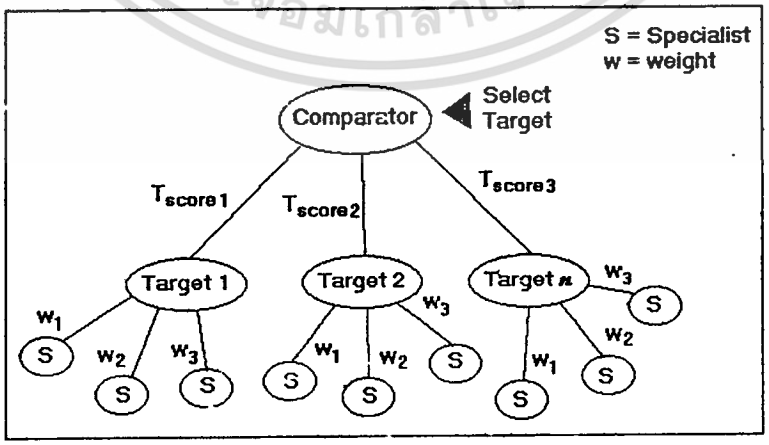
รูปที่ 5.1 โครงสร้างของรูปแบบเป้าหมาย

โดยจะใช้คำ หรือ รูปแบบการแบ่งแยกคำ จากเซตของข้อความกำกวมที่ได้สร้างไว้ กำหนดเป็นรูปแบบเป้าหมายของคำ (Target) และคุณลักษณะ (feature) ต่างๆของรูปแบบเป้าหมายของคำที่แสดงออกถึง การเกิดร่วมกันอย่างมีลำดับ (collocation) และ คำบริบทโดยรอบ (word context) ของรูปแบบเป้าหมายของคำนั้น จะถูกควบคุมโดย ผู้เชี่ยวชาญ (specialist) ซึ่งแต่ละตัวจะมีการควบคุมคุณลักษณะเพียง 1 ถึง 2 คุณลักษณะที่ต่างกันออกไป ทำให้ผู้เชี่ยวชาญ แต่ละตัวมีคุณสมบัติใน

การแบ่งแยกคำเป้าหมายที่ต่างกัน คือมีค่าที่ใช้ในการแบ่งแยก (weight) ที่ต่างกัน ซึ่งค่านี้ของทุกผู้เชี่ยวชาญที่ขึ้นกับรูปแบบเป้าหมายเดียวกัน จะถูกนำมารวมข้าด้วยกัน แล้วใช้เป็นค่าที่จะบอกว่ารูปแบบเป้าหมายนั้นเหมาะสมที่จะเป็นรูปแบบการตัดคำ ของข้อความกำกวม ในประโยคที่กำลังพิจารณามากน้อยแค่ไหน

ขั้นตอนการเรียนรู้ คัดเลือกประโยคตัวอย่าง ซึ่งมีคำที่เป็นสมาชิกในกลุ่มของ ข้อความกำกวมที่กำลังเรียนรู้อยู่ ซึ่งมีการตัดคำ และกำหนดหน้าที่ของคำไว้อย่างถูกต้องแล้ว ป้อนให้กับระบบเพื่อทำการเรียนรู้ ระบบจะสร้างผู้เชี่ยวชาญในการควบคุมคุณลักษณะต่างๆ ที่เกิดขึ้นรอบข้างรูปแบบเป้าหมายของข้อความ โดยให้ค่าที่ใช้ในการแบ่งแยกมีค่าเริ่มต้นที่ 1 และ ทำการปรับค่าที่ใช้ในการแบ่งแยก ของแต่ละผู้เชี่ยวชาญ เป็นระยะๆ เพื่อให้ได้ค่าที่ใช้ในการแบ่งแยก ที่เหมาะสม โดยที่จะทำการเพิ่มค่าที่ใช้ในการแบ่งแยก ด้วยการคูณด้วย 1.5 ให้กับผู้เชี่ยวชาญ ของรูปแบบหรือคำเป้าหมายที่ถูกต้อง เมื่อผู้เชี่ยวชาญนั้น ควบคุมคุณลักษณะ ที่ตรงกับคุณลักษณะที่เกิดขึ้นในประโยคตัวอย่างที่ถูกต้อง และลดค่าที่ใช้ในการแบ่งแยก ด้วยการคูณด้วย 0.5 ให้กับผู้เชี่ยวชาญของรูปแบบหรือคำเป้าหมายที่ผิด เมื่อผู้เชี่ยวชาญนั้น ควบคุมคุณลักษณะ ที่ตรงกับคุณลักษณะที่เกิดขึ้นในประโยคตัวอย่างที่ถูกต้อง ในขณะที่โครงข่ายวินโนว์ให้คำตอบที่ผิด เพื่อว่าคุณลักษณะที่ถูกต้องจะได้มีค่าที่ใช้ในการแบ่งแยกความแตกต่าง ที่สูงต่างจากคุณลักษณะอื่นๆ และใช้เป็นค่าชี้นำในการเลือกรูปแบบที่ถูกต้อง ในการเปรียบเทียบครั้งต่อไป

การตัดสินใจเลือกรูปแบบเป้าหมายของคำ ให้กับประโยคที่กำลังพิจารณาอยู่นั้น โครงข่ายวินโนว์จะเลือก รูปแบบเป้าหมายที่มีคะแนนสูงที่สุด เมื่อเปรียบเทียบกับรูปแบบอื่นๆ เป็นคำตอบให้กับประโยคนั้น นั่นคือ ประโยคที่นำมาพิจารณา มีคุณลักษณะที่เกิดขึ้นร่วมกันกับข้อความกำกวม ตรงกันกับความรู้เกี่ยวกับคุณลักษณะที่เก็บไว้ในระบบ ของรูปแบบเป้าหมายนั้นมากที่สุด



รูปที่ 5.2 โครงข่ายวินโนว์

บทที่ 6

โครงสร้างฐานข้อมูล

การเตรียมข้อมูลต่างๆ สำหรับใช้ในการสอนให้คอมพิวเตอร์เรียนรู้ และทดสอบความรู้ของคอมพิวเตอร์ที่ได้จากการเรียนรู้ เพื่อให้ง่ายต่อการดึงข้อมูลมาใช้งาน โครงการพัฒนานี้ จึงจัดเก็บข้อมูลต่างๆ ที่จำเป็นต้องใช้ในการเรียนรู้ และ ทดสอบ ไว้ในฐานข้อมูล Mysql

ข้อมูลที่จำเป็นสำหรับการพัฒนาการตัดคำ ประกอบด้วย

1. ข้อมูลประโยคต่างๆ ที่ได้วิเคราะห์หน้าที่คำ อย่างถูกต้องแล้วจาก Orchid Corpus
2 ข้อมูลกลุ่มของข้อความกำกวม ทั้งเซตข้อความส่วนหน้า และ เซตสับสน
3 ข้อมูลคุณลักษณะของแต่ละรูปแบบ หรือคำ ที่เป็นสมาชิกของกลุ่มข้อความกำกวม
4 ข้อมูลสถิติ ไบแกรม (Bi-gram) ไตรแกรม (Tri-gram) ของหน้าที่คำ
5 ข้อมูลสถิติของคำ เมื่อทำหน้าที่ต่างๆในประโยค เช่น เป็นนาม กริยา ส่วนขยาย เป็นต้น

6.1 โครงสร้างฐานข้อมูล

ฐานข้อมูลที่ใช้ในโครงการนี้ ประกอบด้วย

6.1.1 ตารางประโยคตัวอย่าง (examp_sent)

จัดเก็บข้อมูลประโยคตัวอย่าง ที่ได้จากฐานข้อมูล Orchid Corpus เพื่อนำมาใช้เป็นข้อมูลในการเรียนรู้ของเครื่อง ภายในตารางประกอบด้วย

- หมายเลขประโยค (sent_no)
- ประโยคตัวอย่าง (sent)

6.1.2 ตารางคำและหน้าที่คำในประโยค (word_of_sent)

เก็บข้อมูลคำ และหน้าที่ของแต่ละคำในประโยค ภายในตารางประกอบด้วย

- หมายเลขประโยค (sent_no)
- หมายเลขคำ (word_no)
- คำ (word)

- หน้าที่คำ (postag)

6.1.3 ตารางเซตข้อความส่วนหน้า (prefix_set)

เก็บข้อมูลของเซตข้อความส่วนหน้า ภายในตารางประกอบด้วย

- เจ้าของเซต (owner)
- คำส่วนหน้าที่เป็นสมาชิกในเซต (member)

6.1.4 ตารางเซตข้อความสับสน (confuse_set)

เก็บข้อมูลของเซตข้อความสับสน ภายในตารางประกอบด้วย

- เจ้าของเซต (owner)
- รูปแบบการแบ่งแยกคำ (pattern)

6.1.5 ตารางข้อมูลผู้เชี่ยวชาญของคำส่วนหน้า (prefix_spec)

เก็บข้อมูลคุณลักษณะต่างๆของคำส่วนหน้า ที่ถูกควบคุมโดยผู้เชี่ยวชาญ ภายในตารางประกอบด้วย

- คำส่วนหน้า (pref_word)
- หมายเลขผู้เชี่ยวชาญ (spec_no)
- คุณลักษณะที่ควบคุม (feature)
- ค่าที่ใช้ในการแบ่งแยก (weight)

6.1.6 ตารางคุณลักษณะของรูปแบบที่เกิดจากข้อความสับสน (conf_spec)

เก็บข้อมูลคุณลักษณะต่างๆของรูปแบบที่เกิดจากข้อความสับสน ที่ถูกควบคุมโดยผู้เชี่ยวชาญ ภายในตารางประกอบด้วย

- รูปแบบของข้อความสับสน (conf_pat)
- หมายเลขผู้เชี่ยวชาญ (spec_no)
- คุณลักษณะที่ควบคุม (feature)
- ค่าที่ใช้ในการแบ่งแยก (weight)

6.1.7 ตารางไบแกรมหน้าที่คำ (bi_postag)

เก็บข้อมูลสถิติการเกิดขึ้นของ ไบแกรมหน้าที่คำ ภายในตารางประกอบด้วย

- หน้าที่ของคำที่ 1 (tag1)
- หน้าที่ของคำที่ 2 (tag2)
- ความถี่ที่เกิดขึ้นในฐานข้อมูล Orchid Corpus (freq)
- ความน่าจะเป็นของการเกิดขึ้นของไบแกรม (prob)

6.1.8 ตารางไตรแกรมหน้าที่คำ (tri_postag)

เก็บข้อมูลสถิติการเกิดขึ้นของ ไตรแกรมหน้าที่คำ ภายในตารางประกอบด้วย

- หน้าที่ของคำที่ 1 (tag1)
- หน้าที่ของคำที่ 2 (tag2)
- หน้าที่ของคำที่ 3 (tag3)
- ความถี่ที่เกิดขึ้นในฐานข้อมูล Orchid Corpus (freq)
- ความน่าจะเป็นของการเกิดขึ้นของไตรแกรม (prob)

6.1.9 ตารางสถิติหน้าที่ของคำ (stat_word)

เก็บข้อมูลสถิติการเกิดขึ้นของ หน้าที่คำ ภายในตารางประกอบด้วย

- คำ (word)
- หน้าที่ของคำ (tag)
- ความถี่ที่เกิดขึ้นในฐานข้อมูล Orchid Corpus (freq)
- ความน่าจะเป็นที่คำ ทำหน้าที่ตามที่ระบุ (prob)

6.2 การคัดเลือกข้อมูล สำหรับการพัฒนาระบบ

จากฐานข้อมูล Orchid Corpus ข้อมูลที่ได้จาก Orchid Corpus ยากต่อการนำมาใช้ และมีบางส่วนเป็นข้อมูลที่ไม่จำเป็น สำหรับโครงการพัฒนานี้ เพื่อให้ได้ข้อมูลที่อยู่ในรูปแบบที่ต้องการและง่ายต่อการนำมาใช้งาน จึงได้ทำการกรองข้อมูลที่ไม่จำเป็นออก แล้วแปลงรูปข้อมูลก่อนทำการจัดเก็บในฐานข้อมูล Mysql (ดูตัวอย่าง ข้อมูล Orchid Corpus ได้จากภาคผนวก)

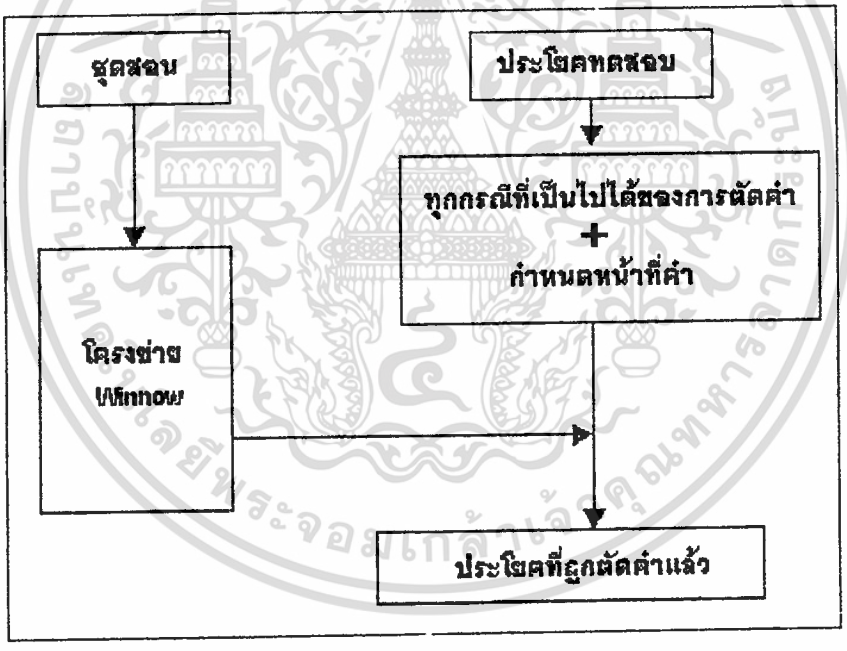
สำหรับการนำมาใช้งาน ในโครงการนี้แบ่งข้อมูลที่ได้ออกเป็น 2 ส่วน 80% สำหรับการเรียนรู้ของเครื่อง อีก 20% ใช้สำหรับการทดสอบความสามารถของเครื่องหลังจากการเรียนรู้

บทที่ 7

ระบบการตัดคำ

โครงการพัฒนานี้ มุ่งประเด็นไปที่การสร้างระบบการตัดคำ ที่สามารถแก้ปัญหาความกำกวมในข้อความ และเลือกรูปแบบการแบ่งแยกคำที่เหมาะสมให้กับข้อความกำกวมที่พิจารณา

การพัฒนาระบบแบ่งเป็น 4 ขั้นตอนหลักๆ คือ ขั้นตอนการเตรียมข้อมูลเบื้องต้น ขั้นตอนการเรียนรู้ และขั้นตอนการทดสอบ เพื่อนำไปใช้งาน และ การติดตั้งระบบการทำงานผ่าน web (Web Application)



รูปที่ 7.1 ระบบการตัดคำ

7.1. การเตรียมข้อมูลเบื้องต้น

7.1.1. สร้างพจนานุกรมที่มีโครงสร้างแบบทรัย

7.1.1.1. รวบรวมคำศัพท์ต่างๆ

7.1.1.2. จัดเรียงลำดับคำศัพท์ตามตัวอักษร

7.1.1.3. จัดเก็บเข้าในโครงสร้างแบบทรัย ที่อธิบายไว้ในบทที่ 3

7.1.2. สร้างเซตของความกำกวม

สร้างเซตของข้อความกำกวมขึ้น จากข้อมูลของคำในพจนานุกรม และ คำจากฐานข้อมูลคำ Orchid Corpus แล้วจัดเก็บไว้ในฐานข้อมูล Mysql ที่สร้างเตรียมไว้

การสร้างเซตของข้อความส่วนหน้า (prefix set) เริ่มจากการนำคำจากฐานข้อมูลคำ Orchid Corpus ที่ยังไม่เคยมีอยู่ในพจนานุกรม มาเพิ่มให้กับพจนานุกม จากนั้นใช้วิธีการเปรียบเทียบหาคำ ที่มีขนาดเล็กกว่า และ อยู่ในส่วนหน้า ของแต่ละคำในพจนานุกรม แล้วบันทึกไว้เป็นเซตของคำส่วนหน้า จัดเก็บในฐานข้อมูล Mysql

การสร้างเซตของความสับสน (Confusion set) จากขั้นตอนการสร้าง เซตของข้อความส่วนหน้า เราได้พจนานุกรมที่มีขนาดใหญ่ขึ้น โดยรวมคำที่ไม่เคยมีมาก่อน จาก Orchid Corpus เข้าด้วยกันกับคำในพจนานุกรมเดิม ในการสร้างเซตของความสับสน จะใช้คำจากพจนานุกรมนี้ มาพิจารณาว่าแต่ละคำสามารถแบ่งแยกคำ ในลักษณะอื่นได้อีกหรือไม่ โดยวิธีการตัดคำโดยการเปรียบเทียบกับคำในพจนานุกรม แล้วเก็บทุกรูปแบบที่สามารถแบ่งแยกได้ของแต่ละคำไว้เป็นเซตของความสับสนของคำนั้นๆ และเนื่องจากข้อความสับสนอาจเกิดขึ้นจากคำมากกว่า 1 คำได้ เช่น "มากรอง" อาจมาจากคำว่า มา กับ กรอง หรือ มาก กับ รอง ก็ได้ (คำว่า มากรอง ไม่มีอยู่ในพจนานุกรม เพราะมันไม่ใช่คำ แต่เป็นกลุ่มคำ หรือส่วนของข้อความ) จึงต้องสร้างเซตของความสับสนประเภทนี้ด้วย โดยนำแต่ละคำจากพจนานุกรม มาพิจารณาตัดคำส่วนหน้าออก แล้วนำส่วนที่เหลือไปรวมกับแต่ละคำในพจนานุกรม แล้วตรวจสอบดูว่าเป็นคำในพจนานุกรมหรือไม่ ถ้าใช่ก็นำมาตัดคำด้วยวิธีการเดิม แล้วเก็บทุกรูปแบบที่สามารถตัดคำได้ ไว้เป็นเซตของความสับสนของข้อความนั้นบันทึกไว้ในฐานข้อมูล Mysql

ตัวอย่างเช่น คำว่า มาก นำมาตัดคำในส่วนหน้าออก จาก มาก เป็น ก แล้วนำส่วนที่เหลือไปรวมกับคำอื่นๆ ในพจนานุกรม เช่นรวมกับคำว่า ว่า ได้เป็น กว่า ซึ่งเป็นคำที่มีอยู่ในพจนานุกรม แล้วนำทั้งหมดคือ มากว่า มาตัดคำด้วยการเปรียบเทียบกับพจนานุกรม ก็จะได้ มา กว่า กับ มาก ว่า แล้วเก็บทั้งสองรูปแบบไว้เป็นเซตของความสับสนของข้อความ มากว่า

7.1.3. เตรียมข้อมูลค่าทางสถิติต่างๆ สำหรับใช้ในการทำงาน

ค่าทางสถิติต่างๆ ได้มาจากการรวบรวมข้อมูล จากฐานข้อมูล Orchid Corpus ซึ่งเป็นฐานข้อมูลที่มีการแบ่งแยกประโยค คำในประโยค และระบุหน้าที่คำไว้อย่างถูกต้องแล้ว ข้อมูลจาก Orchid Corpus ถูกกรองเอาส่วนที่ไม่จำเป็นออกไป เก็บส่วนของประโยค คำ และหน้าที่คำเอาไว้ในฐานข้อมูล Mysql ของเรา แล้วรวบรวมสถิติต่างๆ จากข้อมูลนี้ บันทึกไว้ในส่วนต่างๆ ได้แก่ ข้อมูลสถิติไบแกรม ไตรแกรมหน้าที่คำ สถิติของคำที่ทำหน้าที่ต่างๆ เป็นต้น

7.2. ขั้นตอนการเรียนรู้

สร้างโครงข่ายวินโนว์ขึ้น โดยภายในโครงข่ายจะประกอบด้วย รูปแบบเป้าหมาย (target) ต่างๆ ที่มี ผู้เชี่ยวชาญ (specialist) คอยควบคุมคุณลักษณะ (feature) เฉพาะ 1 ถึง 2 คุณลักษณะ ที่จะใช้โนการแบ่งแยกความแตกต่างของแต่ละ รูปแบบเป้าหมาย ที่อยู่ภายในกลุ่ม ของความกำกวมกลุ่มเดียวกัน (prefix set /confusion set)

โครงข่ายนี้จะทำการเรียนรู้ จากตัวอย่างประโยคที่ถูกต้องที่ได้จาก Orchid Corpus แล้วทำการพิจารณาคุณลักษณะต่างๆที่ถูกควบคุมโดยผู้เชี่ยวชาญ แล้วปรับค่าที่จะใช้ในการแบ่งแยกคำ หรือ รูปแบบ ให้ได้ค่าที่เหมาะสม ที่จะใช้ในการแบ่งแยกแต่ละรูปแบบเป้าหมาย ที่อยู่ภายในเซตเดียวกัน

7.2.1. การกำหนดค่าเริ่มต้น

สำหรับผู้เชี่ยวชาญ ที่ถูกสร้างขึ้น เพื่อควบคุมคุณลักษณะต่างๆ ให้กับรูปแบบเป้าหมายที่ถูกต้อง จะกำหนดให้ ค่าเริ่มต้นของค่าที่ใช้การแบ่งแยก (weight) มีค่าเท่ากับ 1

การเพิ่มผู้เชี่ยวชาญ จะทำเมื่อประโยคที่ใช้ในการเรียนรู้เป็นประโยคที่ถูกต้อง

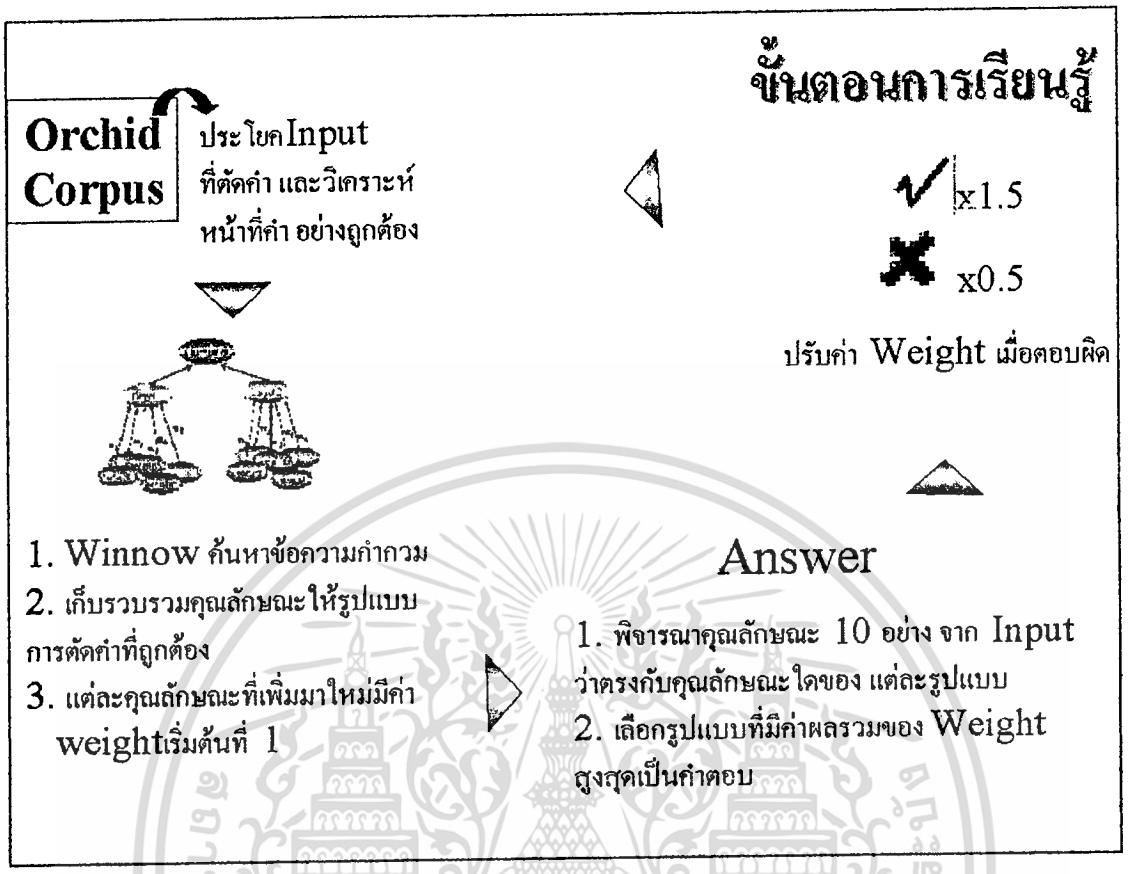
7.2.2. การปรับค่าของค่าที่ใช้การแบ่งแยก (weight)

การปรับค่าที่ใช้ในการแบ่งแยก จะทำเมื่อโครงข่ายวินโนว์ให้คำตอบผิด การปรับค่าที่ใช้ในการแบ่งแยกเสียใหม่ ให้ผู้เชี่ยวชาญที่ควบคุมคุณลักษณะที่ถูกต้องมีค่าที่ใช้แบ่งแยกสูงขึ้น เพื่อให้การเปรียบเทียบครั้งต่อไป วินโนว์จะเลือกรูปแบบที่ถูกต้อง

การปรับค่าที่ใช้ในการแบ่งแยก เมื่อประโยคที่นำมาเรียนรู้เป็นประโยคที่ถูกต้อง และ ระบบให้คำตอบที่ไม่ถูกต้อง

สำหรับรูปแบบเป้าหมายที่ควรเป็นคำตอบ (เป็นคำตอบที่ถูกต้องจริงๆ สำหรับประโยคนั้น) ผู้เชี่ยวชาญที่ควบคุมคุณลักษณะที่ตรงกับประโยคตัวอย่าง จะถูกปรับค่าที่ใช้ในการแบ่งแยก (weight) ให้เพิ่มขึ้นด้วยการคูณ 1.5

สำหรับรูปแบบเป้าหมายที่ไม่ควรเป็นคำตอบ (ไม่ใช่คำตอบที่ถูกต้องจริง สำหรับประโยคนั้น) ผู้เชี่ยวชาญที่ควบคุมคุณลักษณะที่ตรงกับประโยคตัวอย่าง จะถูกปรับค่าที่ใช้ในการแบ่งแยก (weight) ให้ลดลงขึ้นด้วยการคูณ 0.5



รูปที่ 7.2 ขั้นตอนการเรียนรู้ของระบบ

7.3. ขั้นตอนการทดสอบเพื่อนำไปใช้งาน

ขั้นตอนการทดสอบ ประกอบด้วย

1. รับประโยคทดสอบจาก Orchid Corpus (ส่วนที่เตรียมไว้สำหรับการทดสอบระบบ)
2. ตัดคำเบื้องต้น โดยใช้พจนานุกรม ตัดคำให้ได้ทุกรูปแบบที่เป็นไปได้
3. กำหนดหน้าที่คำให้กับแต่ละรูปแบบ ที่เป็นไปได้ ของประโยคที่นำมาพิจารณา
4. ส่งแต่ละรูปแบบให้กับ โครงข่ายวินโนว์ ที่สร้างขึ้นเพื่อพิจารณาเลือกรูปแบบ

7.3.1. การตัดคำเบื้องต้น

ในขั้นตอนนี้ ต้องการรูปแบบการตัดคำที่เป็นไปได้ทุกกรณี โครงการพัฒนาเลือกใช้วิธีการตัดคำโดย การเปรียบเทียบคำในพจนานุกรม ที่มีโครงสร้างแบบทรัย เพื่อให้สามารถประมวลผลการตัดคำได้อย่างรวดเร็ว จากนั้นเก็บผลการตัดคำที่ได้ไว้ในฐานข้อมูล เพื่อรอการกำหนดหน้าที่คำในขั้นตอนถัดไป

7.3.2. การกำหนดหน้าที่คำ

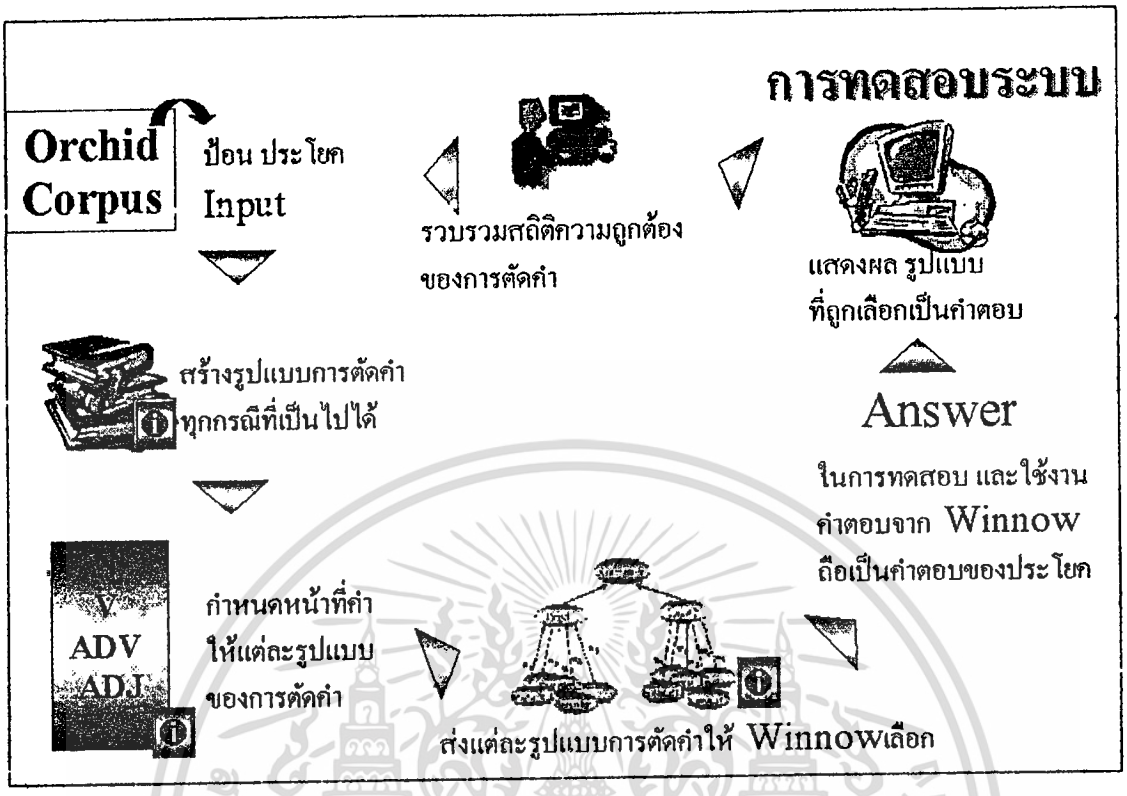
กำหนดหน้าที่คำ ด้วยโมเดลไตรแกรม ที่คำนวณค่าทางสถิติด้วยวิธีวิเทอร์บิ (Viterbi Algorithm) ค่าทางสถิติของชุดหน้าที่คำที่นำมาใช้ในการคำนวณ เป็นชุดเดียวกับของ Orchid Corpus ซึ่งประกอบด้วยหน้าที่คำ 47 ชนิด เนื่องจากข้อมูลค่าทางสถิติดังกล่าว ได้มาจาก Orchid Corpus นั่นเอง แต่ในการกำหนดหน้าที่คำ เราจะทำการแปลงรูปหน้าที่คำให้อยู่ในรูปของชุดหน้าที่คำของ Lexitron เพื่อให้ง่ายต่อการทำงาน (ดูรายละเอียดของชุดหน้าที่คำ ทั้ง 2 ชุดได้จากภาคผนวก)

หลังจากกำหนดหน้าที่คำแล้ว แต่ละรูปแบบที่สามารถตัดคำได้ ของประโยคที่นำมาพิจารณา จะถูกส่งให้โครงข่ายวินโนว์ที่ผ่านการเรียนรู้มาแล้ว ทำการตัดสินใจเลือกรูปแบบการแบ่งแยกคำที่เหมาะสม

7.3.3. การเปรียบเทียบโดยวินโนว์

โครงข่าย วินโนว์ ที่ได้ผ่านการเรียนรู้การแบ่งแยกคำมาแล้ว จะทำการวิเคราะห์หารูปแบบการแบ่งแยกคำที่ถูกต้องให้กับคำกำกวมแต่ละคำที่เกิดขึ้นในประโยค ขั้นตอนการวิเคราะห์เป็นดังนี้

- เลือกเซตของคำกำกวมที่จะใช้ในการวิเคราะห์ ในการทดสอบเราทำการทดสอบ 3 วิธีด้วยกัน คือ ทดสอบด้วยเซตของข้อความส่วนหน้า (prefix set), ทดสอบด้วยเซตของความสับสน (confusion set) และการทดสอบโดยการใช้ทั้งเซตของข้อความส่วนหน้า และเซตของความสับสน แล้วเปรียบเทียบผลการทำงานของแต่ละวิธี
- คำนวณผลรวมของค่าที่ใช้ในการแบ่งแยก (weight) จากผู้เชี่ยวชาญ (specialist) ที่ควบคุมคุณลักษณะที่ตรงกับคุณลักษณะในประโยคทดสอบ ของรูปแบบเป้าหมาย
- เลือกรูปแบบที่มีคะแนนสูงที่สุดเป็นคำตอบ
- ถ้าคำตอบที่ระบบเลือกตรงกันกับที่ปรากฏในรูปแบบประโยคที่กำลังพิจารณาก็จะทำการรวมเอาคะแนนของรูปแบบคำกำกวมที่เลือก ให้กับรูปแบบประโยคนั้น โดยการคูณคะแนนของรูปแบบที่เลือก เข้ากับคะแนนของประโยค เพื่อให้ระดับคะแนนของแต่ละรูปแบบประโยคมีค่าที่แตกต่างกันอย่างเด่นชัด
- จากนั้นก็พิจารณาคำกำกวมคำถัดไปจนกระทั่งจบข้อความในรูปแบบประโยคนั้น ทำในลักษณะเดียวกันทุกรูปแบบประโยค แล้วเลือกรูปแบบที่มีคะแนนสูงสุด เป็นรูปแบบการตัดคำของประโยคนั้น



รูปที่ 7.3 ขั้นตอนการทดสอบการทำงานของระบบ

ในการทดสอบความสามารถของระบบจะนำผลจากการวิเคราะห์เลือกรูปแบบการตัดคำของประโยคนั้น ไปเปรียบเทียบกับรูปแบบการตัดคำที่ถูกต้องของประโยคจาก Orchid Corpus ซึ่งเป็นผลการตัดคำที่ถูกวิเคราะห์โดยนักภาษาศาสตร์ ว่าคำตอบที่ได้จากระบบการตัดคำมีความถูกต้องมากน้อยเพียงใด เพื่อทำการปรับปรุงแก้ไขระบบให้สามารถทำงานได้ดีขึ้น และสุดท้ายเราจะทำการติดตั้งระบบการทำงานผ่าน web

7.4. การติดตั้งระบบการทำงานผ่าน web (Web Application)

ในระบบการทำงานผ่าน web มีขั้นตอนการทำงานในลักษณะเดียวกันกับการทดสอบระบบ ส่วนที่ต่างกันมีเพียงเล็กน้อยเท่านั้น คือ

7.4.1. เปลี่ยนจากการรับประโยคทดสอบจาก Orchid Corpus มาเป็นการรับประโยคที่ต้องการตัดคำจาก User

User จะระบุประโยคที่ต้องการตัดคำผ่านทาง Form แล้วกด submit จากนั้น Browser ก็จะทำการตีความ แล้วส่ง request ไปยัง server เพื่อร้องขอการทำงาน จากนั้น

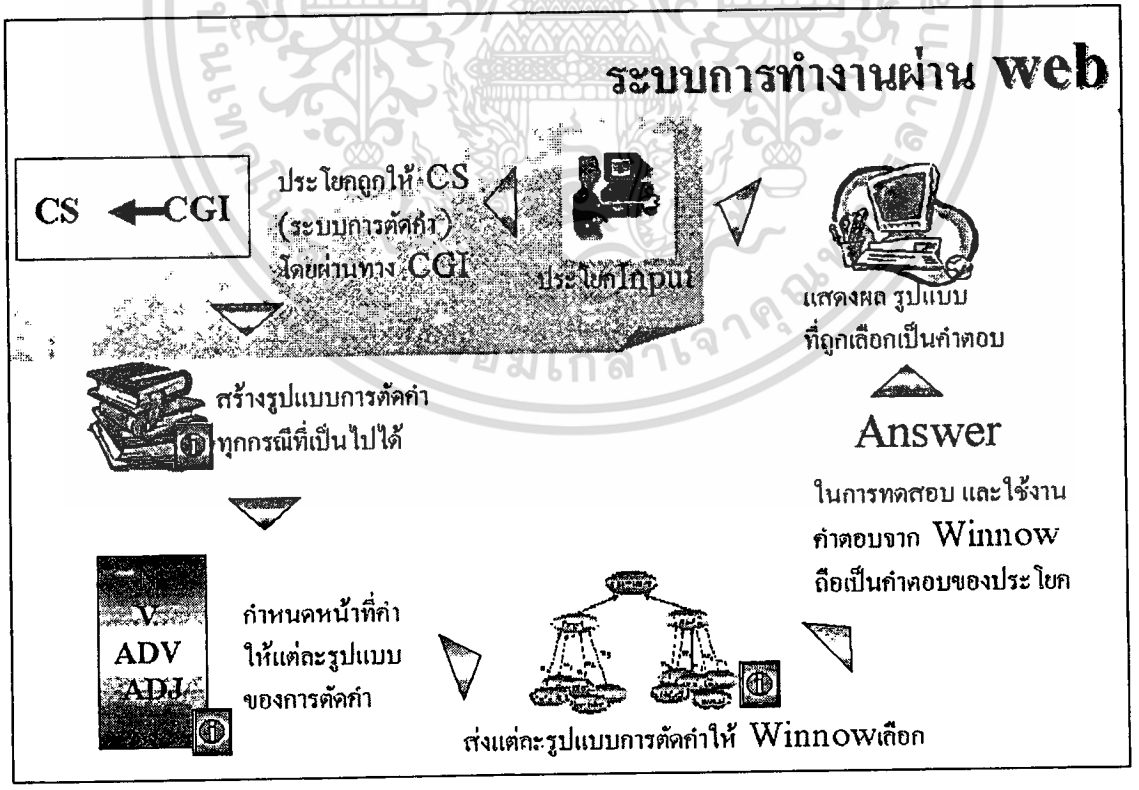
server ก็จะปลุก CGI ขึ้นมาทำงาน ให้ทำการส่งผ่านประโยคที่ต้องการตัดคำไปให้ระบบการตัดคำทำการประมวลผล

7.4.2. การเลือกเซตของข้อความกำกวมที่นำมาใช้ในการวิเคราะห์

เริ่มจากการหาขอบเขตของข้อความกำกวมที่เกิดขึ้นภายในประโยค แล้วเลือกเซตของข้อความสับสน (confusion set) ที่ตรงกันกับข้อความกำกวม หากเซตของข้อความสับสนที่ตรงกับข้อความที่พิจารณาไม่ได้ถูกสร้างไว้ ก็จะเลือกเซตของข้อความส่วนหน้า (prefix set) ขึ้นมาทำงานแทน เนื่องจากการใช้เซตของข้อความส่วนหน้าในการวิเคราะห์เฉพาะส่วนที่เซตของความสับสนไม่ได้ถูกสร้างไว้ ให้ผลการวิเคราะห์ที่ถูกต้องกว่าการเลือกรูปแบบการตัดคำที่ยาวที่สุดมาเป็นคำตอบ (จากสรุปผลการทดสอบระบบ ในบทที่ 8) ส่วนในการคำนวณเพื่อเลือกรูปแบบของคำกำกวมนั้นทำด้วยวิธีการเดียวกันกับที่ใช้ในการทดสอบระบบ

7.4.3. การแสดงผลของคำตอบที่ได้จากระบบ

สุดท้ายเมื่อทำการประมวลผลเสร็จก็จะส่งรูปแบบประโยคที่ตัดคำแล้วกลับไปแสดงผลโดย Browser ยังเครื่องของ User (ดู Application ได้ในภาคผนวก)



รูปที่ 7.4 ระบบการทำงานผ่าน web

บทที่ 8

สรุปผลการทำงานของระบบ

ระบบการตัดคำ ใช้การเรียนรู้ของเครื่องแบบวินโนว์ เพื่อดึงเอาคุณลักษณะที่สำคัญ ของแต่ละรูปแบบของข้อความกำกวม มาช่วยในการพิจารณาเลือกรูปแบบการแบ่งแยกคำให้กับข้อความในประโยค วิธีการนี้ต้องใช้เวลาในการให้เครื่องเรียนรู้มากพอสมควร เพื่อเก็บรวบรวมความรู้เกี่ยวกับรูปแบบของข้อความกำกวมไว้ให้มากพอ ที่จะนำมาใช้ในการตัดสินใจ

ในการพัฒนาระบบการตัดคำนี้ ได้ทำการทดสอบประสิทธิภาพการทำงานของระบบ กับข้อความกำกวมกลุ่มหนึ่ง ผลการทดสอบเป็นดังนี้

ตารางที่ 8.1 ผลการทดสอบการทำงานของระบบด้วยข้อมูลที่เคยใช้ในการเรียนรู้

ลำดับ	คำกำกวม	เปอร์เซ็นต์ความถูกต้องในการตัดคำกำกวม			เปอร์เซ็นต์ความถูกต้องในการเลือกรูปแบบประโยค		
		Prefix set	Confusion set	ใช้ทั้ง 2 แบบ	Prefix set	Confusion set	ใช้ทั้ง 2 แบบ
1	กำหนดการ	88.89%	100%	100%	66.67%	77.78%	77.78%
2	จัดการ	95.46%	78.41%	100%	56.82%	78.41%	87.50%
3	ทำการ	60%	90%	80%	35%	90%	75%
4	ที่อยู่	92.09%	98.46%	99.23%	50.39%	87.97%	88.74%
5	ที่ตั้ง	66.67%	83.34%	100%	60.42%	83.34%	100%
6	มีค่า	47.62%	90.48%	95.24%	28.57%	90.48%	95.24%
7	รายได้	100%	100%	100%	100%	100%	100%
8	วันที่	70%	100%	100%	50%	100%	100%
9	หลักการ	75%	100%	100%	50%	90%	90%
10	หรือไม่	68.18%	96.88%	100%	23.01%	96.88%	96.88%

ตารางที่ 8.2 ผลการทดสอบการทำงานของระบบด้วยข้อมูลที่ไม่เคยใช้ในการเรียนรู้

ลำดับ	คำกำกวม	เปอร์เซ็นต์ความถูกต้องในการตัดคำกำกวม			เปอร์เซ็นต์ความถูกต้องในการเลือกรูปแบบประโยค		
		Prefix set	Confusion set	ใช้ทั้ง 2 แบบ	Prefix set	Confusion set	ใช้ทั้ง 2 แบบ
1	กำหนดการ	100%	100%	100%	100%	100%	100%
2	จัดการ	63.33%	63.33%	83.34%	10%	43.33%	46.67%
3	ทำการ	50%	70%	70%	40%	70%	70%
4	ที่อยู่	100%	94.43%	100%	33.44%	52.48%	52.48%
5	ที่ตั้ง	87.50%	100%	100%	0%	70.84%	70.84%
6	มีค่า	50%	60%	70%	20%	40%	40%
7	รายได้	37.50%	25%	50%	37.50%	25%	50%
8	วันที่	50%	75%	100%	50%	50%	50%
9	หลักการ	70%	50%	90%	60%	50%	50%
10	หรือไม่	64.29%	83.93%	100%	16.07%	50%	53.57%

จากผลการทดสอบ เมื่อพิจารณาผลการวิเคราะห์เพื่อเลือกรูปแบบการตัดคำที่ถูกต้องให้กับข้อความกำกวมนั้น การใช้เซตของความสับสน (confusion set) ในการวิเคราะห์ให้ผลการวิเคราะห์ที่ถูกต้องกว่า การวิเคราะห์โดยการใช้เซตของข้อความส่วนหน้า (prefix set) และในกรณีที่ไม่มีเซตของความสับสน การเลือกเอาเซตของข้อความส่วนหน้ามาช่วยในการวิเคราะห์แทนให้ผลการวิเคราะห์ที่ถูกต้องกว่าการเลือกเอารูปแบบการตัดคำที่ยาวที่สุดมาเป็นคำตอบ

สรุปผลการทดสอบ ระบบสามารถเลือกรูปแบบของข้อความกำกวมที่สนใจได้ถูกต้องเหมาะสมกับประโยค ค่อนข้างสูง เนื่องจากระบบเคยได้รับการเรียนรู้เกี่ยวกับข้อความกำกวมนั้นมาแล้ว แต่ผลการเลือกรูปแบบประโยคที่ถูกต้องยังไม่ค่อยดีนัก เพราะภายในประโยคมีข้อความกำกวมมากกว่า 1 ข้อความ และระบบยังได้รับการเรียนรู้ไม่มากพอ ความรู้ที่ระบบเก็บไว้จึงยังไม่สามารถเลือกรูปแบบของความกำกวมได้ถูกต้องทั้งหมด แต่หากระบบได้รับการเรียนรู้มากขึ้น เชื่อว่าการทำงานของระบบก็จะมีความถูกต้องมากขึ้น

บทที่ 9

ข้อเสนอแนะ และแนวทางการพัฒนาต่อ

การพัฒนาระบบการตัดคำในลักษณะนี้ มีขั้นตอนในการทำงานค่อนข้างมาก แต่ละขั้นตอนมีความสลับซับซ้อน และต้องการข้อมูล ที่มีความถูกต้องจำนวนมากพอ ที่จะให้ระบบได้เรียนรู้แล้วเก็บรวบรวมความรู้เหล่านั้น ไว้ใช้ในการทำงานต่อไป

การทำงานของระบบนี้ ยังมีข้อจำกัดอยู่ ระบบยังไม่สามารถหาขอบเขตของประโยคได้ ผู้ใช้ยังต้องทำการกำหนดขอบเขตของประโยค ก่อนที่จะนำข้อมูลเข้าสู่ระบบการตัดคำ และการทำงานกับข้อมูลที่มีความซับซ้อนมากๆ ต้องใช้เวลาในการประมวลผลค่อนข้างสูง ประกอบกับระบบจะทำงานได้ดีต้องมีฐานข้อมูลความรู้ที่มีขนาดใหญ่พอ เมื่อฐานข้อมูลมีขนาดใหญ่ขึ้น การทำงานก็จะต้องใช้เวลามากขึ้นด้วย ดังนั้นในการพัฒนาระบบลักษณะนี้ ต้องการเครื่องที่มีประสิทธิภาพมากพอ จึงจะทำงานได้ดี ซึ่งเชื่อว่าในอนาคตอันใกล้นี้ ปัญหาเช่นนี้จะหมดไป เนื่องจากเครื่องคอมพิวเตอร์รุ่นใหม่ๆ มีประสิทธิภาพ และความสามารถในการทำงานมากขึ้น

อีกปัญหากับการทำงาน ที่มีผลต่อประสิทธิภาพการทำงานของระบบการตัดคำ ปัญหาเกิดขึ้นเนื่องจาก พจนานุกรมที่นำมาใช้ในการตัดคำเบื้องต้นนั้น ไม่ค่อยจะเหมาะสมนัก มีการจัดเก็บคำที่เป็นคำประกอบขนาดใหญ่ๆ ไว้จำนวนมาก เช่น "กระทรวงวิทยาศาสตร์ เทคโนโลยี และการพลังงาน" คำลักษณะนี้ทำให้การตัดคำเบื้องต้น ตัดคำได้หลายรูปแบบ เกิดรูปแบบจำนวนมากที่ต้องนำไปประมวลผล

ปัจจุบันผลการทำงานของระบบกับประโยคทั่วๆไป ที่รับมาจากผู้ใช้นั้น ยังให้คำตอบที่ถูกต้องไม่สูงมากนัก เนื่องจากระบบยังได้รับการเรียนรู้มาน้อย และประกอบกับข้อมูลที่ทำการเรียนรู้มีขอบเขตจำกัดในเรื่องใดเรื่องหนึ่ง เท่านั้น ดังนั้นเมื่อผู้ใช้ป้อนประโยคที่อยู่นอกขอบเขตของข้อมูลที่ระบบได้เรียนรู้มา ระบบก็จะไม่สามารถให้คำตอบที่ถูกต้องได้

การนำระบบนี้ไปพัฒนาต่อ ควรกรองข้อมูลคำศัพท์ที่คิดว่าเกินความจำเป็นออก เพื่อลดจำนวนของรูปแบบการแบ่งแยกคำในข้อความกำกวมลง แต่ต้องยังคงรูปแบบที่ถูกต้องไว้ กรองรูปแบบการแบ่งแยกคำกำกวม ในแต่ละเซตของข้อความกำกวม ที่ไม่เคยเป็นคำตอบที่ถูกต้องออก เพื่อลดขนาดของฐานข้อมูลความรู้ในส่วนที่ไม่จำเป็น และลดปริมาณความต้องการหน่วยความจำในการทำงานให้น้อยลง และควรทำการพัฒนาเพิ่มในส่วนของการกำหนดขอบเขตของประโยค และการกำหนดขอบเขตของคำที่ไม่มีอยู่ในพจนานุกรม เพื่อเพิ่มความสามารถในการทำงานของระบบ

ให้ รองรับกับความต้องการของผู้ใช้มากขึ้น และควรเพิ่มส่วนการทำงานที่เปิดให้ผู้ใช้สามารถเพิ่มข้อมูลความรู้ใหม่ๆ ที่ถูกต้องให้กับระบบได้ เพื่อเป็นการขยายฐานข้อมูลความรู้ ที่มีอยู่เดิมให้มีขนาดใหญ่ขึ้น และมีข้อมูลมากพอที่จะใช้ในการประมวลผลกับประโยคข้อความในด้านต่างๆ เพื่อเพิ่มขีดความสามรถในการประมวลผลของระบบให้มากขึ้น รองรับกับประโยคข้อความประเภทต่างๆ ได้มากขึ้น

บรรณานุกรม

ดวงแก้ว สวามิภักดิ์, 2523. การสร้างซอฟต์แวร์วิเคราะห์ไวยากรณ์ไทยภายใต้ระบบยูนิกซ์. กรุงเทพฯ : สำนักพิมพ์มหาวิทยาลัยธรรมศาสตร์.

วิรัช ศรเลิศล้ำวานิช, 2536. "การตัดคำภาษาไทยในระบบแปลภาษา." หน้า 50-55. ใน การแปลภาษาด้วยคอมพิวเตอร์ กรุงเทพ : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์ และคอมพิวเตอร์แห่งชาติ.

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

สิงห์ ตรงงาม, 2540. "ระบบการวิเคราะห์ประโยคภาษาไทยที่มีการละประธานที่ซ้ำกันของประโยค." วิทยานิพนธ์มหาบัณฑิต บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ไพศาล เจริญพรสวัสดิ์, 2541. "การตัดคำภาษาไทยโดยใช้คุณลักษณะ." วิทยานิพนธ์มหาบัณฑิต บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

Allen, J. 1995. **Natural Language Understanding.** The Benjamin/Cummings Publishing Company.

Blum, A. 1997. Empirical Support for Winnow and Weighted-Majority Algorithm: Result on a Calendar scheduling domain. [Online]. Available : http://mbone.it.kmitl.ac.th/~kannika/papers.

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

Charoenpornsawat, P. 1997. Feature-based Thai Word Segmentation. [Online]. Available : http://mbone.it.kmitl.ac.th/~kannika/papers.

Charoenpornsawat, P and Kijsirikul, K. 1998. Feature-base Thai Unknown Word Boundary Identification Using Winnow. [Online]. Available : http://mbone.it.kmitl.ac.th/~kannika/papers.

Golding, A. R., Roth, D. 1996. A Winnow-Based Approach to Context-Sensitive Spelling Correction. [Online]. Available : http://mbone.it.kmitl.ac.th/~kannika/papers.

Kawtrakul, A. et. al. 1997. Automatic Thai Unknown Word Recognition. [Online]. Available : http://mbone.it.kmitl.ac.th/~kannika/papers.

Lillian, L and Rie, A. 1999. Unsupervised Statistical Segmentation of Japanese Kanji Strings. [Online]. Available : http://www.cs.cornell.edu/home/llee/papers.

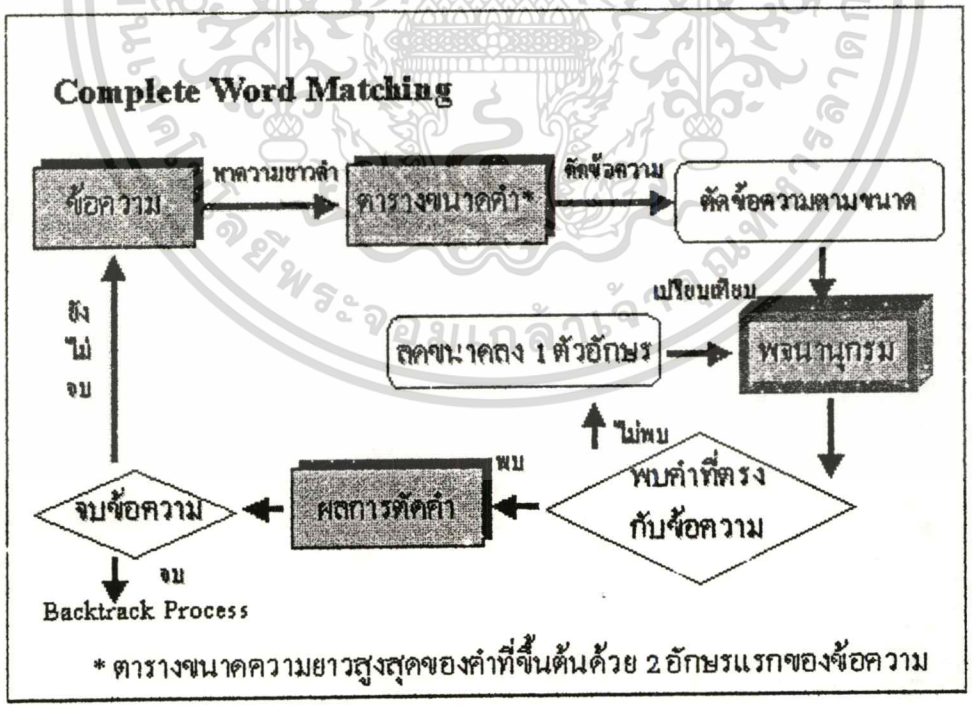
Sornlertlamvanich, V and Tanaka, H. 1995. The Automatic Extraction of Open Compounds from Text Corpora. [Online]. Available : http://www.links.nectec.or.th/virach/publication.html.

Thairattananond, Y. 1981. "Towords The Design of A Thai Text Syllable Analyzer." Master Thesis of Asian Institute of Technology.

ภาคผนวก

ตัวอย่าง ขั้นตอนการทำงานของวิธีการตัดคำ โดยใช้การเปรียบเทียบกับคำในพจนานุกรม แบบ Complete Word Matching

1. รับข้อความที่จะทำการตัดคำ
2. ตรวจสอบความยาวของคำศัพท์ที่ขึ้นต้นด้วย 2 ตัวอักษรแรกของข้อความที่จะทำการตัดคำในพจนานุกรมว่ามีขนาดเท่าไร
3. ตัดข้อความตามขนาดที่ได้จากข้อ 2 แล้วนำไปเปรียบเทียบกับคำในพจนานุกรม ถ้าไม่พบก็ทำการลดความยาวลงทีละ 1 ตัวอักษร จนกระทั่งพบคำที่ตรงกันก็จะใส่เครื่องหมายแยกคำ แล้วนำส่วนของข้อความที่เหลือมาพิจารณาต่อ จนจบข้อความ
4. ทำการย้อนกลับไปพิจารณาที่หน่วยคำสุดท้าย ที่ได้จากการตัดคำโดยลดขนาดของหน่วยคำสุดท้ายลงทีละ 1 ตัวอักษร เปรียบเทียบกับคำในพจนานุกรมเพื่อพิจารณาว่าสามารถแบ่งเป็นคำที่เล็กกว่าเดิมได้หรือไม่ ถ้าได้ก็จะได้รูปแบบการตัดคำเพิ่มขึ้น ถ้าไม่ได้ก็จะพิจารณาหน่วยคำก่อนหน้านั้นในวิธีการเดียวกัน จนกระทั่งถึงหน่วยคำแรกของข้อความ ซึ่งถ้าสามารถแบ่งเป็นคำที่เล็กกว่าเดิมได้ ก็จะต้องทำการพิจารณาในส่วนที่เหลือใหม่จนกระทั่งจบข้อความ



ขั้นตอนการทำงานของ การตัดคำโดยการเปรียบเทียบคำกับพจนานุกรม

ตัวอย่างเช่น

"ตากลม"

1. ข้อความที่รับเข้ามาคือ "ตากลม"
2. คำที่ยาวที่สุดในพจนานุกรมที่ขึ้นต้นด้วย "ตา" คือ 4
3. นำ "ตากล"ไปเปรียบเทียบกับคำในพจนานุกรม ถ้าไม่พบก็ทำการลดความยาวลงทีละ 1 ตัวอักษร จนกระทั่งพบคำที่ตรงกันก็จะใส่เครื่องหมายแยกคำ ในขั้นตอนนี้ได้คำว่า "ตาก" ทำเครื่องหมายแยกคำไว้ จากนั้นนำส่วนของข้อความที่เหลือของข้อความคือ "ลม" มาพิจารณาต่อ
4. คำที่ยาวที่สุดที่ขึ้นต้นด้วย "ลม" คือ 3
5. นำ "ลม" ไปเปรียบเทียบกับคำในพจนานุกรม ในขั้นตอนนี้ได้คำว่า "ลม" ซึ่งจบข้อความพอดี จบขั้นตอนนี้ได้ผลการตัดคำออกมาเป็น

ตาก / ลม

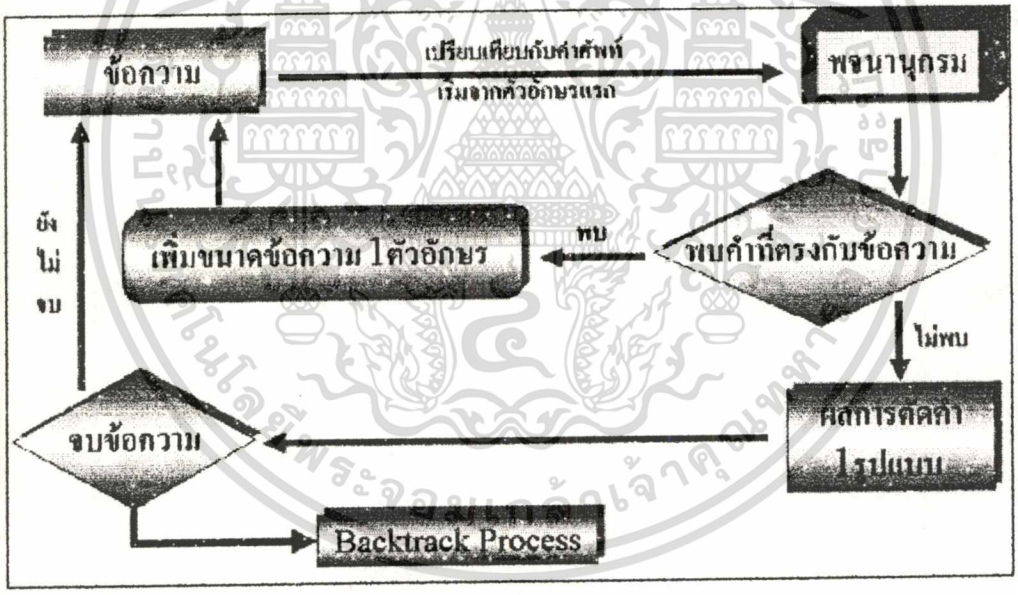
6. ทำการย้อนกลับไปพิจารณาที่หน่วยคำสุดท้าย ที่ได้จากการตัดคำคือ "ลม" โดยลดขนาดของหน่วยคำสุดท้ายลง 1 ตัวอักษรได้ "ล"
7. เปรียบเทียบกับคำในพจนานุกรม จากขั้นตอนนี้ไม่พบคำที่ตรงกัน แสดงว่าไม่สามารถแบ่งคำย่อยได้อีก
8. ย้อนกลับไปพิจารณาหน่วยคำก่อนสุดท้าย (ในที่นี้คือหน่วยคำแรก เพราะมีแค่ 2 คำ) คือ "ตาก" ลดขนาดของหน่วยคำลง 1 ตัวอักษรได้ "ตา"
9. เปรียบเทียบกับคำในพจนานุกรม จากขั้นตอนนี้ได้คำว่า "ตา" ทำเครื่องหมายแยกหน่วยคำไว้ แล้วนำส่วนที่เหลือคือ "กลม" มาพิจารณาต่อ
10. คำที่ยาวที่สุดที่ขึ้นต้นด้วย "กล" คือ 3
11. นำ "กลม"ไปเปรียบเทียบกับคำในพจนานุกรม ในขั้นตอนนี้ได้คำว่า "กลม" ซึ่งจบข้อความพอดี จบขั้นตอนนี้ได้ผลการตัดคำออกมาเป็น

ตา / กลม

12. ทำการย้อนกลับไปพิจารณาที่หน่วยคำสุดท้าย ที่ได้จากการตัดคำคือ "กลม" โดยลดขนาดของหน่วยคำสุดท้ายลง 1 ตัวอักษรได้ "กล"
13. เปรียบเทียบกับคำในพจนานุกรม จากขั้นตอนนี้ได้คำว่า "กล" ซึ่งทำให้ส่วนที่เหลือของข้อความคือ "ม" ที่เป็นเอักษรตัวเดียวไม่สามารถที่จะถูกพิจารณาเป็นคำได้ การวิเคราะห์จึงจบลงคำว่า "กลม" จึงไม่สามารถแบ่งคำย่อยได้อีก
14. สุดท้ายได้ผลของการตัดคำออกมา 2 แบบคือ ตาก / ลม และ ตา / กลม

ตัวอย่าง ขั้นตอนการทำงานของวิธีการตัดคำ โดยใช้การเปรียบเทียบกับคำในพจนานุกรม ที่ใช้ในโครงการพัฒนา

1. รับข้อความที่ต้องการตัดคำ
2. ทำการเปรียบเทียบกับคำในพจนานุกรมที่มีโครงสร้างแบบทรัยทีละ 1 ตัวอักษรจนกระทั่งเจอเครื่องหมายสิ้นสุดคำตัวสุดท้ายของเส้นทางนั้น ก็จะได้ 1 รูปแบบของคำนั้น ใส่เครื่องหมายแบ่งคำเอาไว้
3. นำข้อความส่วนที่เหลือมาพิจารณาต่อด้วยวิธีเดียวกันจนกระทั่งจบข้อความ
4. นำรูปแบบการตัดคำที่ได้ มาพิจารณาทีละคำจากคำสุดท้าย ไปถึงคำแรก (Backtrack process) เพื่อดูว่าแต่ละคำสามารถแบ่งแยกเป็นคำในรูปแบบอื่นได้อีกหรือไม่ เพื่อสร้างรูปแบบการตัดคำที่เป็นไปได้ทุกกรณีของข้อความนั้น



ขั้นตอนการตัดคำโดยการเปรียบเทียบกับคำในพจนานุกรมที่มีโครงสร้างแบบทรัย

<u>หมายเหตุ</u> ลักษณะการทำงานง่ายกว่าในแบบ Complete word matching ไม่ต้องตรวจความยาวของคำที่ขึ้นต้นด้วย 2 ตัวอักษร การเปรียบเทียบในวิธีการนี้เริ่มจากตัวอักษรแรกไปจนจบคำ การทำงานก็ทำได้เร็วกว่า เนื่องจากพจนานุกรมที่ใช้มีโครงสร้างการจัดเก็บโดยใช้ pointer เชื่อมโยงตัวอักษรระหว่างคำทำให้ง่ายต่อการค้นหา จุดสิ้นสุดคำก็สังเกตได้ง่ายจากเครื่องหมายจบคำ

ตัวอย่าง ข้อมูลจากฐานข้อมูล Orchid Corpus

#1 ← เลขที่ประโยค

ประเทศไทยได้มีการปรับเปลี่ยนโครงสร้างในการพัฒนาเศรษฐกิจของประเทศ \\
จากประเทศเกษตรกรรมไปสู่ความเป็นประเทศอุตสาหกรรมมากยิ่งขึ้น// ← ประโยค

ประเทศไทย/NPRP
ได้/XVAM
มี/VSTA
การ/FIXN
ปรับเปลี่ยน/VACT
โครงสร้าง/NCMN
ใน/RPRE
การ/FIXN
พัฒนา/VACT
เศรษฐกิจ/NCMN
ของ/RPRE
ประเทศ/NCMN
<space>/PUNC
จาก/RPRE
ประเทศ/NCMN
เกษตรกรรม/NCMN
ไปสู่/RPRE
ความ/FIXN
เป็น/VSTA
ประเทศอุตสาหกรรม/NCMN
มาก/ADVN
ยิ่งขึ้น/ADVN

← คำและหน้าที่

//

ตัวอย่าง ชุดหน้าที่คำที่นำมาใช้

ชุดที่ 1 ชุดหน้าที่คำจากฐานข้อมูล Orchid Corpus

ลำดับ	หน้าที่คำ	รายละเอียด	ตัวอย่าง
1	ADVI	Adverb with iterative form	เร็วๆ, เสมอๆ, ช้าๆ
2	ADVN	Adverb with normal form	เก่ง, เร็ว, ช้า, สม่ำเสมอ
3	ADVP	Adverb with prefixed form	โดยเร็ว
4	ADVS	Sentential Adverb	โดยปกติ, ธรรมดา
5	CFQC	Frequency classifier	ครั้ง, เที่ยว
6	CLTV	Collective classifier	คู่, กลุ่ม, ฝูง, เชิง, ทาง, ด้าน, แบบ
7	CMTR	Measurement classifier	กิโลกรัม, แก้ว, ชั่วโมง
8	CNIT	Unit classifier	ตัว, คน, เล่ม
9	CVBL	Verbal classifier	ม้วน, มัด
10	DCNM	Determiner, cardinal number expression	เสือ 2 ตัว
11	DDAC	Definite determiner, allowing classifier in between	นี้, นั้น, โน้น, นู้น
12	DDAN	Definite determiner, after noun without classifier in between	นี่, นั่น, โน่น, ทั้งหมด
13	DDAQ	Definite determiner, following quantitative expression	พอดี, ถ้วน
14	DDBQ	Definite determiner, between noun and classifier or preceding quantitative expression	ทั้ง, อีก, เพียง
15	DIAC	Indefinite determiner, following noun; allowing classifier in between	ไหน, อื่น, ต่างๆ
16	DIAQ	Indefinite determiner, following quantitative expression	กว่า, เศษ

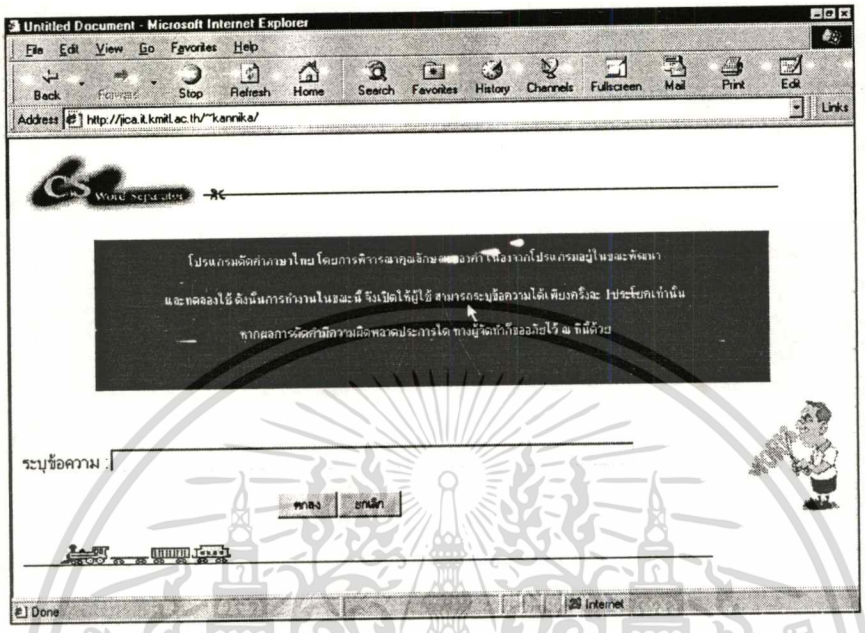
ลำดับ	หน้าที่คำ	รายละเอียด	ตัวอย่าง
17	DIBQ	Indefinite determiner, between noun and classifier or preceding quantitative expression	บาง, ประมาณ, เกือบ
18	DONM	Determiner, ordinal number expression	ที่หนึ่ง, ที่สอง, ที่สุดท้าย
19	EAFF	Ending for affirmative sentence	จ๊ะ, จ้ะ, ค่ะ, ครับ, นะ, น่า, เถอะ
20	EITT	Ending for interrogative sentence	หรือ, เหรอ, ไหม, มั้ย
21	FIXN	Nominal prefix	การทำงาน, ความสนุกสนาน
22	FIXV	Adverbial prefix	อย่างเร็ว
23	INT	Interjection	โอ้ย, โอ้, เออ, เอ๋, อ๋อ
24	JCMP	Comparative conjunction	กว่า, เหมือนกับ, เท่ากับ
25	JCRG	Coordinating conjunction	และ, หรือ, แต่
26	JSBR	Subordinating conjunction	เพราะว่า, เนื่องจาก, ที่, แม้ว่า, ถ้า
27	NCMN	Common noun	หนังสือ, อาหาร, อาคาร, คน
28	NCNM	Cardinal number	หนึ่ง, สอง, สาม, 1, 2, 3
29	NEG	Negator	ไม่, มิ, ไม่ได้, มิได้
30	NLBL	Label noun	1, 2, 3, a, b, ก, ข
31	NONM	Ordinal number	ที่หนึ่ง, ที่สอง, ที่1, ที่ 2
32	NPRP	Proper noun	วินโดวส์ 95, พระอาทิตย์
33	NTTL	Title noun	ดร., พลเอก
34	PDMN	Demonstrative pronoun	นี่, นั่น, ที่นี่, ที่นั่น
35	PNTR	Interrogative pronoun	ใคร, อะไร, อย่างไร
36	PPRS	Personal pronoun	คุณ, เขา, ฉัน
37	PREL	Relative pronoun	ที่, ซึ่ง, อัน, ผู้
38	PUNC	Punctuation	(,),", ,,;
39	RPRE	Preposition	จาก, ณ, ของ, ใต้, บน
40	VACT	Active verb	ทำงาน, ร้องเพลง, กิน
41	VATT	Attributive verb	อ้วน, ดี, สวย
42	VSTA	Stative verb	เห็น, รู้, คือ

ลำดับ	หน้าที่คำ	รายละเอียด	ตัวอย่าง
43	XVAE	Post-verb auxiliary	ไป, มา, ขึ้น
44	XVAM	Pre-verb auxiliary, after negator "ไม่"	ค่อย, น่า, ได้
45	XVBB	Pre-verb auxiliary, in imperative mood	กรุณา, จง, เชิญ, อย่า, ห้าม
46	XVBM	Pre-verb auxiliary, before negator "ไม่"	เกิด, เกือบ, กำลัง
47	XVMM	Pre-verb, before or after negator "ไม่"	ควร, เคย, ต้อง

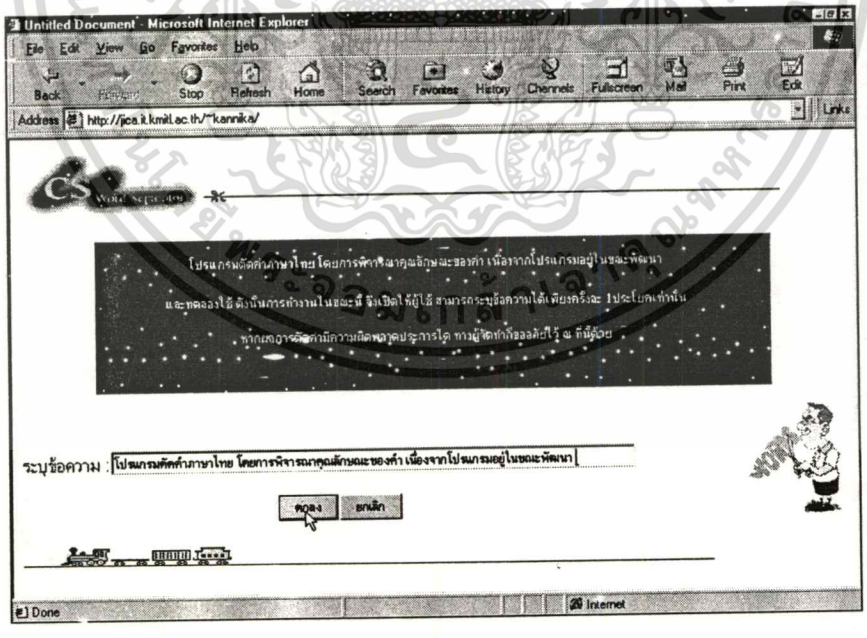
ชุดที่ 2 ชุดหน้าที่คำจาก Lexitron

ลำดับ	หน้าที่คำ	รายละเอียด	ตัวอย่าง
1	ADJ	Adjective	อ่วน, ดี, สวย
2	ADV	Adverb	เก่ง, เร็ว, โดยเร็ว, โดยปกติ
3	AUX	Auxiliary verb	กรุณา, ควร, เคย, ต้อง, น่า, เกือบ
4	CLAS	Classifier	ตัว, เล่ม, คน, ชั่วโมง
5	CLASS	Collective classifier	ฝูง, กลุ่ม, แบบ, รุ่น
6	CONJ	Conjunction	และ, หรือ, แต่, กว่า, เพราะว่า
7	DET	Determiner	นี่, นั่น, นี้, นั้น, โน่น, โน้น
8	END	Ending word	จ๊ะ, จ๋ะ, ค่ะ, ครับ, นะ, น่า, เถอะ
9	INT	Interjection	โอ้ย, โอ้, เออ, เอ๋, อ๋อ
10	N	Noun	หนังสือ, อาคาร, อาหาร
11	NEG	Negator	ไม่, มิ, ไม่ได้, มิได้
12	PIXP	Prefix word	การ, ความ, อย่าง
13	PREP	Preposition	จาก, ละ, ของ, ใต้, บน
14	PRON	Pronoun	คุณ, เขา, ฉัน, ที่นี่, ใคร
15	PUNC	Punctuation	(,),", ,,,;
16	V	Verb	ทำงาน, เดิน, กิน, เห็น, รู้

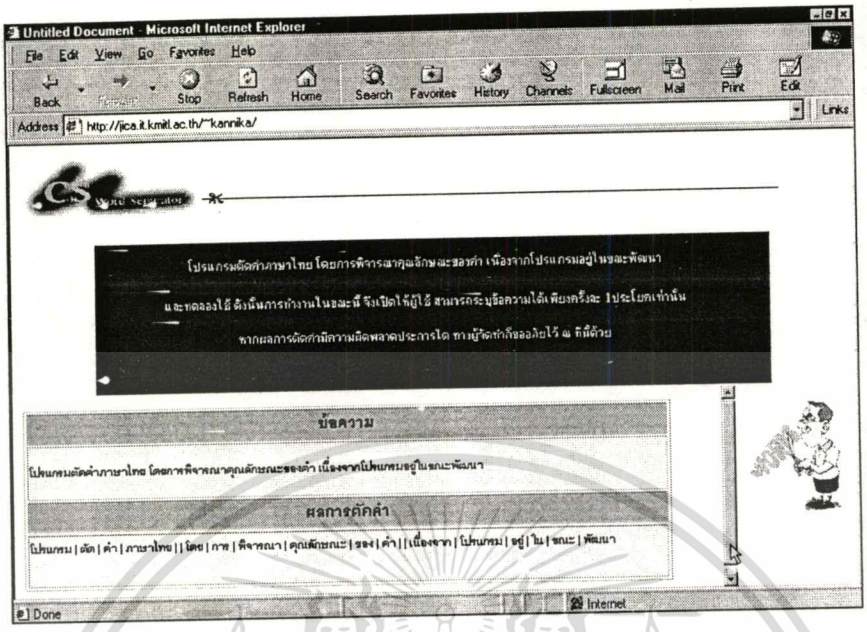
ระบบการตัดคำผ่าน Web (Web Application)



หน้าแรกของระบบการตัดคำ



การป้อนข้อมูลประโยคที่ต้องการวิเคราะห์



ผลการตัดคำที่ได้จากระบบ

ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาว กรรณิกา จินดาปทีป
เกิดวันที่	30 พฤษภาคม พ.ศ. 2519
สถานที่เกิด	กรุงเทพฯ
ประวัติการศึกษา	สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒวิทยาเขตประสานมิตร ในปี พ.ศ. 2541 และ เข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปี พ.ศ. 2542