การกำหนดคีย์เวิร์ดให้กับเว็บเพจภาษาไทย

KEYWORD DETERMINATION FOR THAI WEB PAGES

โดย

นายชิดพงษ์ นาคะเกศ

นายฐิติ ชุมภูปัน

อาจารย์ที่ปรึกษา

ดร. ชุติเมษฎ์ ศรีนิลทา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การกำหนดคีย์เวิร์ดให้กับเว็บเพจภาษาไทย

KEYWORD DETERMINATION FOR THAI WEB PAGES

ผู้จัดทำ

1. นาย ชิดพงษ์ นาคะเกส รหัสประจำตัว 39014132

2. นาย จิติ ชุมภูปัน รหัสประจำตัว 39014146

ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษา

(ดร. ชุติเมษฎ์ ศรีนิลทา)

การกำหนดคีย์เวิร์ดให้กับเว็บเพจภาษาไทย

นาย ชิดพงษ์ นาคะเกศ 39014132
นาย ฐิติ ชุมภูปัน 39014146
ดร. ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

ปัจจุบันเทคโนโลยีทางด้านอินเตอร์เน็ต มีการพัฒนาค่อนข้างรวดเร็ว ทำให้การนำไปใช้งานค่อนข้างง่าย กลุ่มของผู้ใช้งานก็ขยายตัวกว้างขึ้นเรื่อย ๆ เนื่องจากเป็นแหล่งที่สามารถให้ข้อมูล ความรู้ ความบันเทิง และความสะดวกสบายต่าง ๆ บริการที่เป็นที่นิยมอย่างหนึ่งคือเวิร์ลไวด์เว็บ ซึ่งมีผู้นิยมใช้เป็นจำนวนมาก โดยข้อมูลที่มีอยู่ในเครือข่ายเวิร์ลไวด์เว็บนั้นจะอยู่ในรูปแบบเอกสารที่เรียกว่า "เว็บเพจ" ซึ่งมีจำนวนมาก และมีเนื้อหาเกี่ยวกับเรื่องราวต่าง ๆ ที่หลากหลาย ดังนั้นในการหาข้อมูลที่ตรงกับที่เราต้องการจึงเป็นสิ่งที่ทำได้ยาก จึงทำให้เกิดเครื่องมือค้นหา หรือเสิร์ชเอ็นจิ้นขึ้นมาเพื่อช่วยให้การค้นหาเป็นไปได้ง่ายขึ้น และขั้นตอนหนึ่งที่มีผลต่อประสิทธิภาพการทำงานของเสิร์ชเอ็นจิ้น คือ การกำหนดคีย์เวิร์ดที่เหมาะสมให้กับแต่ละเว็บเพจ เพื่อให้ได้ข้อมูลที่มีเนื้อหาที่ใกล้เคียงกับที่เราต้องการมากที่สุด

วิทยานิพนธ์ฉบับนี้เป็นแนวทางหนึ่งในการหาคีย์เวิร์ดโดยได้นำเอาทฤษฎีการกำหนดคีย์เวิร์ดในภาษาอังกฤษ และตำแหน่งของข้อความหลังแท็กภาษาแฮชทีเอ็มแอล มาใช้ในการกำหนดคีย์เวิร์ดให้กับเว็บเพจภาษาไทย อย่างไรก็ตามงานวิจัยนี้ยังสามารถที่จะนำไปประยุกต์ใช้ในการทำเสิร์ชเอ็นจิ้นต่อไปได้อีกด้วย

Keyword Determination For Thai Web Pages

Chidpong Nakhakes

Thiti Choomphupan

Dr. Chutimet Srinilta Advisor

ABSTRACT

Today 's technology in the field of internet has developed rapidly and the growth of user has continued because internet is resource of data, knowledge, entertainment and the World Wide Web is a popular service. Information in World Wide Web network has been called "Web Page". There are many web pages and content about variety of topics. To search information that match with your need is difficult. So, you can use the tools that has been known as search engine. This tools will help you to search information and one of stage that affect to the performance of search engine is determine proper keyword for each web page. The good performance will help you to get the information that has the topics match your need.

This research is a way of searching keywords by use the indexing theory and the ranking of web page for search engine. However, this research has to apply with the search engine in the future.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

ชิดพงษ์ นาคะเกศ

จิติ ชุมภูปัน

สารบัญ

หน้าที่

สารบัญตาราง

หน้าที่

สารบัญภาพ

หน้าที่

บทที่ 1

บทนำ

1.1 ความสำคัญ และที่มา

เทคโนโลยีทางด้านอินเตอร์เน็ตในปัจจุบัน มีความก้าวหน้าค่อนข้างสูง ทั้งในด้านความเร็ว และประสิทธิภาพในการทำงานที่สูงขึ้น อีกทั้งกลุ่มผู้ใช้งานขยายตัวกว้างขึ้นเรื่อย ๆ เนื่องมาจากประโยชน์ในด้านแหล่งข้อมูล ความรู้ ความบันเทิง และการติดต่อสื่อสาร โดยวิธีการเชื่อมต่อ หรือติดต่อสื่อสารของเครือข่ายอินเตอร์เน็ตผ่านโพรโตคอล (Protocol) โดยโพรโตคอลที่เป็นพื้นฐาน สำหรับการเชื่อมโยงของเครือข่ายอินเตอร์เน็ต จะใช้ TCP/IP (Transmission control protocol / Internet protocol) ซึ่งโพรโตคอลนี้ถือเป็นโพรโตคอลมาตรฐานในการกำหนดรายละเอียดการทำงาน ทำให้สามารถเชื่อมโยงคอมพิวเตอร์ที่มีความแตกต่างกันได้

ระบบเครือข่ายเวิร์ลไวด์เว็บ เริ่มด้วยข้อมูลที่มีลักษณะ Interactive hypermedia หรือกล่าวอีกอย่างว่า เป็นรูปแบบหรือเอกสารที่ใช้งานที่เรียกว่า ไฮเปอร์เท็กซ์ (Hypertext) ซึ่งเป็นเอกสารที่สามารถ เชื่อมโยงกับเอกสารต่าง ๆ ที่มีความสัมพันธ์กัน โดยภาษาที่ใช้เป็นข้อกำหนดในการสร้างเอกสารรูปนี้คือภาษาแฮชทีเอ็มแอล (Hypertext markup language) ภาษาแฮชทีเอ็มแอล มีการกำหนดส่วนที่เรียกว่ามาร์คอัพ (Markup) หรือจุดที่จะเชื่อมโยงส่วนเอกสารต่าง ๆ ไปยังแหล่งข้อมูลอื่น ๆ วิธีการ ข้อกำหนดในการรับ-ส่ง และข้อมูลของระบบเว็บ จะอาศัยโพรโตคอลแฮชทีเอ็มแอล (Hypertext transfer protocol) รูปแบบข้อมูลจะเรียกว่า ไฮเปอร์มีเดีย (Hypermedia) ทั้งนี้เพราะข้อมูลมีความหลากหลายรูปแบบ การใช้งานของตัวข้อมูล ไม่ว่าจะเป็น เท็กซ์ (Text) กราฟฟิก(Graphics) รูปภาพ (Image) เสียง (Audio) วิดีโอ (Video) และอื่น ๆ ซึ่งในส่วนของการแสดงผลหากเป็นการใช้งานแบบเดิมหรือในยุคแรก ๆ ของอินเตอร์เน็ต จะแสดงผลเป็นเท็กซ์อย่างเดียว โดยใช้ Lynx ซึ่งเป็นคำสั่งใช้งานบนระบบปฏิบัติการยูนิกซ์ (UNIX) เป็นตัวค้นหาข้อมูลหรือเอกสาร (แบบเดียวกันกับบราวเซอร์) แต่ปัจจุบันสามารถใช้งานผ่านโปรแกรมจำพวกเว็บเบราวเซอร์ (Browser) ที่มีขีดความสามารถสูงทำให้สามารถใช้งาน ได้กับข้อมูลที่มีรูปแบบหลากหลายได้ โดยการจัดรูปแบบการนำเสนอยังต้องอาศัยภาษา แฮชทีเอ็มแอล ในการกำหนด และสร้างเอกสาร

เราทราบกันดีอยู่แล้วว่า ในอินเทอร์เน็ต มีเว็บไซต์ที่คอยให้บริการอยู่มากมาย หลายแห่งด้วยกัน แต่ผู้ใช้บริการของเว็บไซต์เหล่านั้น มักต้องพบกับปัญหา เพราะผลของการค้นข้อมูลที่ออกมานั้น มักจะได้เว็บไซต์เป็นจำนวนมาก ซึ่งส่วนใหญ่แล้วก็เป็นเว็บไซต์ที่ไม่ได้มีข้อมูลเกี่ยวข้องกับความต้องการของผู้ใช้บริการ ดังนั้นจึงเกิดปัญหาในการใช้เว็บไซต์ค้นหาข้อมูลอย่างไร เพื่อให้ได้ข้อมูล ที่ตรงกับความต้องการมากที่สุด

แม้เว็บไซต์ค้นข้อมูล จะมีชื่อเป็นที่รู้กันทั่วไปว่า เครื่องจักรค้นหา (Search engine) แต่ความจริงแล้ว เว็บไซต์หลายแห่งไม่ได้มี เครื่องจักรสำหรับค้นหาข้อมูลตามชื่อ ยกตัวอย่างเช่น Yahoo! ที่มีการทำงานเบื้องหลัง เป็นการค้นหาข้อมูลตามหมวดคำ (Search directory) เป็นต้น ความแตกต่างของเว็บไซต์

ค้นข้อมูลทั้งหลาย จึงอยู่ที่เทคนิคที่ใช้ ในการค้นข้อมูลนั่นเอง ซึ่งแบ่งออกได้เป็น 2 ลักษณะดังที่กล่าวมา คือ การใช้เครื่องจักรค้นหา และการใช้หมวดคำค้นหา

ในทางกลับกันหากเรามีข้อมูลที่เป็นเอกสารแฮชทีเอ็มแอลอยู่จำนวนหนึ่ง และต้องการที่จะทราบว่าเอกสารดังกล่าวนี้มีใจความเกี่ยวกับเรื่องอะไร ซึ่งเราจะนำเอาหลักการของเครื่องจักรค้นหาหรือเสิร์ชเอ็นจิ้นนี้มาประยุกต์ใช้งานกับโครงงานวิจัยนี้ เพื่อจะหาคำที่เป็นคีย์เวิร์ดของเว็บเพจนั้น ๆ ได้

และเนื่องจากเสิร์ชเอ็นจิ้นที่มีอยู่ในปัจจุบัน ไม่สนับสนุนกับเว็บไซต์ที่เป็นภาษาไทย จึงทำให้งานวิจัยนี้มีแนวคิดที่จะหาวิธีการกำหนดคีย์เวิร์ดให้กับเว็บเพจที่เป็นภาษาไทย เพื่อนำไปประยุกต์ใช้กับการสร้างเสิร์ชเอ็นจิ้นที่สนับสนุนภาษาไทยในอนาคตต่อไป

โครงงานการกำหนดคีย์เวิร์ดให้กับเว็บเพจภาษาไทยนั้นเป็นการวิจัยโครงสร้างของเอกสารแฮชทีเอ็มแอล เพื่อหาส่วนประกอบพื้นฐานภายในเอกสารแฮชทีเอ็มแอล ที่จะสามารถนำมาใช้ประกอบการกำหนดคีย์เวิร์ด โดยจะมีการสร้างโปรแกรมการค้นหาคีย์เวิร์ดภายในเอกสารแฮชทีเอ็มแอล เป็นลักษณะซีจีไอโปรแกรม ซึ่งจะอยู่ในเครื่องเซิร์ฟเวอร์ โดยจะทำการรับข้อมูลที่เป็นยูอาร์แอลของเว็บเพจ ที่ผู้ใช้ต้องการทราบว่า เป็นเว็บเพจเกี่ยวกับเรื่องอะไร หรือรับข้อมูลที่เป็นไฟล์จากเครื่องของผู้ใช้ผ่านทางโฮมเพจที่ได้จัดทำขึ้น แล้วทำการแสดงผลลัพธ์เป็นคีย์เวิร์ดจากยูอาร์แอลของเว็บเพจหรือจากไฟล์ที่ผู้ใช้ส่งมาให้กับเซิร์ฟเวอร์

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษาความเป็นไปได้ในการประยุกต์วิธีการกำหนดคีย์เวิร์ดในเอกสารภาษาอังกฤษ มาใช้ในการกำหนดคีย์เวิร์ดให้กับเว็บเพจภาษาไทย

1.2.2 เพื่อวิเคราะห์ หาแท็ก และค่าน้ำหนักของคำหลังแท็ก ที่เหมาะที่จะนำมาใช้ในการกำหนดคีย์เวิร์ด

1.2.3 สร้างโปรแกรมในการกำหนดคีย์เวิร์ด ส่วนการติดต่อใช้งานโปรแกรมกับผู้ใช้ที่สามารถใช้งานผ่านอินเตอร์เน็ตได้

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้เป็นการเสนอวิธีที่จะใช้กำหนดคีย์เวิร์ดให้กับเว็บเพจภาษาไทยวิธีหนึ่ง โดยพิจารณาจากความถี่ที่ปรากฏของคำ และตำแหน่งของคำที่อยู่ในโค้ดแฮชทีเอ็มแอล โดยพิจารณาค่าความสำคัญของคีย์เวิร์ดจากปัจจัยเหล่านี้เป็นหลัก ซึ่งจะใช้ค่าน้ำหนักเป็นสิ่งที่กำหนดความสำคัญของคำพบ ซึ่งคำที่มีค่าความสำคัญมากก็จะมีความเป็นไปได้สูงที่จะเป็นคีย์เวิร์ด และในงานวิจัยนี้จะทำการสร้างเว็บไซต์ เพื่อให้ผู้ใช้ป้อนยูอาร์แอลที่ต้องการหาคีย์เวิร์ด หรือทำการอัพโหลดไฟล์ที่ต้องการหาคีย์เวิร์ดได้ และแสดงคีย์เวิร์ดที่หาได้ตามจำนวนที่ผู้ใช้ต้องการ

โดยงานวิจัยนี้จะคำนึงถึงเว็บเพจที่เขียนถูกต้องตามหลักไวยากรณ์ของภาษาแฮชทีเอ็มแอล จึงยังมีข้อจำกัดของตัวโปรแกรมในหลาย ๆ ด้าน

1.4 วิธีการดำเนินงาน

งานวิจัยในโครงงานนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลัก ๆ อยู่ 4 เรื่องด้วยกัน คือ ทฤษฎีการกำหนดคีย์เวิร์ด การเขียนโปรแกรมด้วยภาษาวิชวลเบสิก การเขียนโปรแกรมซีจีไอ และโครงสร้างภาษาแฮชทีเอ็มแอล ซึ่งมีรายละเอียดดังในบทที่ 2 โดยแยกอธิบายออกเป็นหัวข้อต่าง ๆ จากนั้นก็จะนำเอาความรู้ที่ได้ศึกษาทั้งหมดมาทำการวิเคราะห์ และสร้างโปรแกรมในบทที่ 3 โดยจะอธิบายการทำงานของโปรแกรมเป็นสองส่วนคือ ส่วนการทำงานของโปรแกรมซีจีไอ และส่วนโปรแกรมที่ใช้ในการกำหนดคีย์เวิร์ด อธิบายเหตุผลที่มาของแนวคิด ซึ่งอาจจะมีการนำเอาส่วนต่าง ๆ ที่อยู่ในทฤษฎีมาอ้างอิงด้วย

ต่อไปในบทที่ 4 จะเป็นการทดลอง และการวิเคราะห์โปรแกรม เพื่อวิเคราะห์หาค่าน้ำหนักที่เหมาะสมที่จะนำมาใช้ในการกำหนดคีย์เวิร์ด และบทที่ 5 จะเป็นการสรุปการทำงาน ผลที่ได้รับจากงานวิจัยนี้ และแนวทางในการนำไปประยุกต์ใช้

บทที่ 2

ทฤษฎี และหลักการ

2.1 ทฤษฎีการกำหนดดัชนี (Indexing theory)

การกำหนดดัชนีเป็นผลมาจากการทำการวิเคราะห์เอกสาร (Document analysis) ซึ่งการวิเคราะห์เอกสารจะช่วยให้เกิดประสิทธิภาพในการเข้าถึงแต่ละเอกสารหรือเอกสารย่อย ๆ ที่จัดเก็บเอาไว้ เมื่อมีการเขียนหนังสือขึ้นมาเล่มหนึ่งหรือเขียนเอกสารขึ้นมาเป็นจำนวนมาก โดยปกติแล้วผู้เขียนหนังสือจะทำดัชนีเอาไว้ โดยที่ดัชนีจะเก็บคำ (Term) ที่ได้เลือกเอาไว้ โดยจะเก็บชื่อ และตำแหน่งที่อยู่ภายในเอกสารต่าง ๆ ของคำ ๆ นั้นเอาไว้ และดัชนีพิเศษอาจถูกเพิ่มเข้ามาเพื่อใช้ระบุตำแหน่งของรูปภาพ ทฤษฎีบทในคณิตศาสตร์ ชื่อพืชในหนังสือการจัดสวน คำสดุดีถึงผู้เขียน และอื่น ๆ ด้วยเหตุนี้ในระบบการค้นหาข้อมูล (Information retrieval) จึงได้มีการนำดัชนีมาประยุกต์ใช้งานด้วย เมื่อเอกสารมีอยู่เป็นจำนวนมากก็จะทำให้คุณภาพของการทำดัชนีลดลงในด้านประสิทธิภาพ และประสิทธิผลการดึงข้อมูล

ขณะที่หนังสือทางวิชาการโดยทั่วไปจะมีดัชนีอยู่ แต่ส่วนพวกเอกสารที่มีเนื้อหาสั้น ๆ เช่น รายงานการวิจัยค้นคว้า และเอกสารรายงานการประชุม หรือหนังสือพวกวรรณคดี นิยายวิทยาศาสตร์ บทละคร และบทกวีนั้นจะไม่ค่อยมีการทำดัชนีเอาไว้ ด้วยเหตุนี้เอกสารอะไรก็ตามที่มีการเก็บไว้ในฐานข้อมูลที่เป็นเท็กซ์ (Text) มักจะต้องมีการสร้างดัชนีเอาไว้ด้วย

ดัชนีจะถูกสร้างโดยยึดหลักของภาษาการทำดัชนี (Indexing langauge) ที่จะประกอบด้วยเซตของ Index terms โดยที่คำเหล่านี้อาจจะเป็นคำเดียวโดด ๆ วลีหรือเป็นทั้งสองอย่างก็ได้ เพื่อให้เกิดความแน่นอนในฐานข้อมูลแล้วต้องทำการพิจารณาถึงลักษณะของภาษาการทำดัชนีที่จะมาทำเป็น Index terms

คุณสมบัติของภาษาที่ใช้ในการทำดัชนีจะต้องประกอบด้วย 2 ลักษณะต่อไปนี้ คือ

1. Exhaustivity คือ การมีใจความ (Topic) ที่ครอบคลุมตัวดัชนีได้สมบูรณ์
2. Specificity คือ ระดับของความถูกต้องของการทำดัชนี

ในการทำดัชนีนั้นจะต้องพิจารณาถึงจุดประสงค์หลัก 3 ข้อ คือ

1. ทำให้หาตำแหน่งของเอกสารได้ง่ายโดยดูจากใจความของเอกสารนั้น
2. กำหนดส่วนที่เกี่ยวกับใจความของเอกสาร และเกี่ยวเนื่องกับเอกสารอื่นมากที่สุด
3. ทำให้สามารถคาดเดาความเกี่ยวข้องกันของเอกสาร ที่มีอยู่กับข้อมูลที่ต้องการหา

การทำดัชนีสามารถแบ่งออกได้เป็น 2 ประเภทหลัก ๆ คือ

1. การทำดัชนีแบบแมนนวล (Manual indexing) จะใช้มนุษย์เป็นผู้ทำ ซึ่งจะกระทำโดยผู้กำหนดดัชนี (Indexer) ที่ได้รับการฝึกฝนมาเป็นอย่างดีแล้ว แต่ว่าการทำดัชนีด้วยวิธีการนี้ยังมีปัญหาในเรื่องของการขาดความแน่นอน (Lack of consistency) คือ ในการทำดัชนีให้กับเอกสารที่เป็นกลุ่มใหญ่หรือมีจำนวนเอกสารเป็นจำนวนมาก ๆ นั้น โดยทั่วไปแล้วจะต้องใช้ผู้ทำดัชนีเป็นจำนวนหลายคน ซึ่ง

จากการศึกษาในหลาย ๆ ด้านแล้วทำให้รู้ว่าผู้ทำดัชนีแต่ละคนแม้จะได้รับการฝึกฝนมาเป็นอย่างดีแล้วก็ตาม ก็ไม่สามารถกำหนดดัชนีได้เหมือนเดิมเสมอไปแม้จะเป็นเอกสารชุดเดียวกันก็ตาม

2. การทำดัชนีแบบอัตโนมัติ (Automatic indexing) เป็นการใช้โปรแกรมเข้ามาช่วยในการกำหนดดัชนี ซึ่งอัลกอริธึมในการทำดัชนีของเอกสารแบบอัตโนมัตินี้โดยส่วนใหญ่แล้วจะยึดหลักของการนับความถี่ (Frequency) ของคำเป็นหลัก

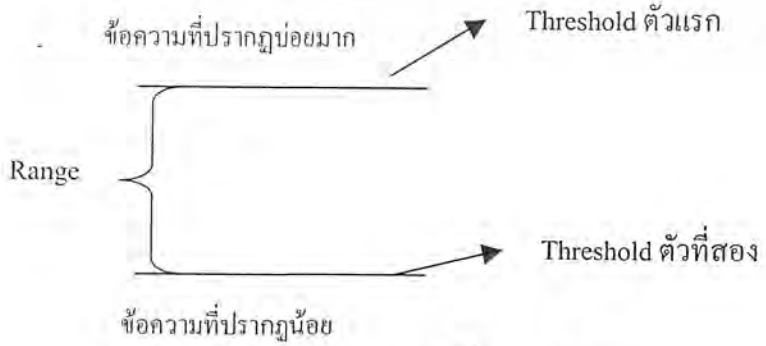
ซึ่งแนวโน้มในการทำการวิจัยค้นคว้าในเรื่องการทำดัชนีให้กับเอกสารนั้นก็ได้ยึดเอาทั้งสองแบบนี้เป็นหลัก

2.1.1 การกลั่นกรอง และการวิเคราะห์คำดัชนี

ลักษณะอย่างหนึ่งของภาษาโดยทั่วไป คือ ส่วนประกอบพื้นฐานของตัวภาษา เช่น ตัวอักษร, คำ, วลี หรือ ประโยคนั้น โดยปกติแล้วจะปรากฏหรือมีไม่เท่ากันภายในเนื้อหาที่กำลังพิจารณาอยู่ในขณะนั้น

โดยคำที่มักจะปรากฏบ่อยครั้งนั้นส่วนใหญ่จะเป็นคำจำพวก the, and, of, OH!, he, she, his, to และa ซึ่งจะเป็นคำจำพวกสรรพนาม บุพบท สันธาน และอุปทาน (คำเหล่านี้เราเรียกว่าเป็นคำที่มีอยู่ทั่วไปหรือคำที่เป็นสามัญ (Common words)) ซึ่งคำเหล่านี้มักจะปรากฏเป็นส่วนใหญ่ในข้อความหรือเอกสารต่าง ๆ และคำเหล่านี้ถือได้ว่าเป็นคำดัชนีที่ไม่ดีพอหรือแย่มาก (Poor) ด้วยเหตุผล 2 ประการ คือ ประการแรก คำเหล่านี้ปรากฏบ่อยเกินไป และปรากฏในเกือบทุก ๆ ข้อความหรือเอกสาร และประการที่สอง คำเหล่านี้ไม่มีส่วนเกี่ยวพันหรือมีน้อยกับใจความในเอกสาร และโดยตัวมันเองแล้วไม่ค่อยมีความหมาย

เมื่อพิจารณาถึงคำที่ปรากฏน้อยมาก คือ มีความถี่ต่ำ โดยปรากฏในเอกสารเพียงแค่ครั้งหรือสองครั้งก็มีแนวโน้มที่จะเป็นคำดัชนีที่ไม่ดีหรือแย่ ด้วยเหตุผลทั้ง 2 ข้อนี้ อาจกล่าวได้ว่าเราสามารถกำหนดค่าขีดเริ่ม (Threshold) สองค่าให้กับคำดัชนี โดยเทรสโฮลด์ตัวแรกสำหรับคำที่มีปรากฏบ่อย และเทรสโฮลด์ตัวที่สองสำหรับคำที่มีปรากฏน้อยมาก ซึ่งคำที่มีลักษณะดังสองประการที่กล่าวมาแล้วจะถือว่าเป็นคำดัชนีที่ไม่ดี ส่วนคำที่ปรากฏอยู่ระหว่างช่วงค่าขีดเริ่มทั้งสองค่านี้จะถูกใช้เป็นคำดัชนีต่อไป



รูปที่ 2-1การกำหนดค่าขีดเริ่ม(Theshold)

2.1.2 วลี และความใกล้ชิด (Phrase and proximity)

แม้ว่าหลักการนับความถี่ที่ปรากฏของคำแต่ละคำบนเอกสารนั้นจะเป็นพื้นฐานในการกำหนดดัชนี แต่หลักการนี้ไม่ได้คำนึงถึงความสัมพันธ์ที่เกิดขึ้นระหว่างคำหลาย ๆ คำ ซึ่งวลีเป็นตัวอย่างที่เห็นได้ชัด ยกตัวอย่างเช่น เมื่อมีคำว่า Information ปรากฏในเอกสาร โดยอยู่ในรูปของกลุ่มคำต่อไปนี้ เช่น Information structure, Information retrieval หรือ Information system ซึ่งวลีหรือกลุ่มคำเหล่านี้มีความหมาย ซึ่งคำแต่ละคำในวลีหรือกลุ่มจะมีหรือไม่มีความหมายในตัวมันเองก็ตาม ด้วยเหตุนี้มันจึงสำคัญหรือจำเป็นที่จะต้องทราบ และพิจารณาการนับวลีที่มีความสำคัญในเอกสารไว้ด้วย

การนับความถี่ที่ปรากฏของวลีสามารถกระทำได้เช่นเดียวกับการนับความถี่ของคำเดียวโดด ๆ ที่ปรากฏ และการกำหนดค่าน้ำหนักให้ก็สามารถนำมาใช้ได้เช่นเดียวกัน

2.1.3 หลักการของการทำดัชนีแบบอัตโนมัติ

จากหลักการการทำดัชนีแบบอัตโนมัตินั้นจะใช้โปรแกรมเข้ามาช่วยในการทำงาน โดยที่ตัวโปรแกรมจะมีอัลกอริธึมในทำงานดังนี้

- ทำการสแกนตัวเอกสารเพื่อตรวจดูโครงสร้างของเอกสาร เช่น หัวเรื่อง, จำนวนย่อหน้า, วันที่ และส่วนอื่น ๆ
- ทำการสแกนหาโทเคินของคำ (Word token) ซึ่งเป็นพวกตัวเลข, ตัวอักษรพิเศษ, พวกตัวอักษรตัวใหญ่เป็นต้น

ซึ่งภาษาต่าง ๆ ทั่วโลกที่มีใช้อยู่ เช่น ภาษาจีน ภาษาไทยนั้นเป็นภาษาที่มีลักษณะของหน่วยย่อยของคำ (Morphological unit) ติดกันยาว ซึ่งต่างจากคำในภาษาอังกฤษที่มีการเว้นวรรคคำแต่ละคำ ดังนั้นเพื่อจะทำโทเคนของคำในภาษาไทยหรือภาษาจีนนั้นต้องมีการทำการตัดคำเสียก่อน โดยส่วนนี้ก็จะเป็นหน้าที่ของอัลกอริธึมของการตัดคำ

- หลังจากนั้นทำการกำจัดคำพวกสต็อปเวิร์ด (Stopwords) ทิ้ง โดยสต็อปเวิร์ดจะเป็นคำจำพวกคำสามัญหรือคำที่มีอยู่ทั่วไปภายในเอกสาร ซึ่งไม่ค่อยมีความสำคัญหรือส่วนเกี่ยวข้องกับใจความของเอกสาร เมื่อคำเหล่านี้อยู่โดด ๆ จะไม่สามารถสื่อความหมายที่ชัดเจนให้เห็นได้ โดยคำเหล่านี้ได้แก่ the, and, or, a, these, across, here, can, by, actually เป็นต้น โดยจะเก็บเอาไว้ในรายการคำ (Short list)

เพื่อช่วยให้กระบวนการการกำหนดดัชนีเป็นไปอย่างมีประสิทธิภาพนั้น หลังจากที่มีการกำจัดคำที่มีอยู่ทั่วไป และคำจำพวกสต็อปเวิร์ดแล้ว คำที่เหลือก็จะถูกทำการลดรูปคำให้อยู่ในรูปของรากศัพท์ของคำนั้น ๆ ซึ่งจะเรียกว่า Stem word โดยเราจะทำการลดรูปคำศัพท์ต่างให้อยู่ในรูปของรากศัพท์ของมันได้โดยใช้ Stemming algorithm ซึ่งเป็นอัลกอริธึมที่ใช้ในการลดรูปของคำ โดยคำต่าง ๆ

เช่น คำกิริยา คุณศัพท์ คำวิเศษณ์ คำนามจะถูกลดรูปให้อยู่ในรูปของรากศัพท์ เช่น คำว่า "Computer", "Computation", "Computed", "Computes", "Computing", "Computable", "Computational", "Computationally" จะถูกลดรูปให้เป็น "Comput"

ปัญหาที่เกิดขึ้นกับ Stemming algorithm

- การลดรูปของรากศัพท์ผิด โดยส่วนใหญ่แล้ว "-ed" จะโดยตัดออกจากคำเพื่อลดรูป แต่ในคำว่า "Bed" ไม่ควรที่ตัดทิ้งเพราะจะทำให้ไม่มีความหมาย ซึ่งสามารถแก้ไขได้โดยมีรายการพิเศษของคำที่ต้องยกเว้น เช่น มีการกำหนดว่าตัว Stem word ต้องมีตัวอักษรไม่น้อยกว่า 3 ตัว ก็จะทำให้คำว่า "bed" ไม่ต้องลดรูปอีก ส่วนคำว่า "breed" ต้องทำการลดรูป เราจึงต้องเก็บคำว่า "Breed" ไว้ในรายการนี้ เป็นต้น

- เรื่องของรูปพหูพจน์ของคำ เช่น คำว่า "Knife" เป็น "Knives"

ทำการนับความถี่ของคำที่ปรากฏในเอกสาร โดยจะทำการกำจัดคำที่มีความถี่สูงมาก ๆ และเป็นพวกคำที่มีอยู่ทั่วไปหรือคำสามัญทิ้ง และคำที่มีความถี่ต่ำมาก ๆ คือแทบจะไม่ปรากฏในเอกสารเลย โดยจะใช้หลักการการกำหนดค่าขีดเริ่มต้นเป็นเกณฑ์ในการกำจัดคำทิ้ง และหลังจากนั้นจะทำการกำหนดค่าความถี่ให้กับคำทุกคำที่เหลือ ซึ่งค่าความถี่ของคำ (Term frequency :TF) นี้เองที่ใช้เป็นตัวกำหนดว่าจะตัดคำทิ้งหรือไม่ โดยถ้าค่าความถี่ของคำนี้มีอยู่ในช่วงของค่าขีดเริ่มที่กำหนดไว้ ก็จะเก็บคำเหล่านั้นเอาไว้ ส่วนที่เหลือก็ทำการตัดทิ้งไป

2.2 ทฤษฎีการตัดคำ

การตัดคำ (Word segmentation) หรือการแบ่งข้อความที่ต่อเนื่องกันออกเป็นหน่วยคำ ๆ (Morpheme) หรือลักษณะของการรู้จำ (Recognition) คำหนึ่ง ๆ ในข้อความที่ต่อเนื่องกันนั้นเริ่มจะมีความหมายมากขึ้นเป็นลำดับเมื่อมีการนำเอาคอมพิวเตอร์เข้ามาช่วยในการประมวลผลข้อมูลทางภาษามากยิ่งขึ้น ความยากง่ายหรือวิธีการที่จะนำมาใช้ในการตัดคำนั้นต้องขึ้นอยู่กับลักษณะเฉพาะของภาษานั้น ๆ เป็นอย่างมาก

การตัดคำ คือ การแบ่งสายอักขระ (String) เพื่อหาขอบเขตของแต่ละหน่วยคำ (Morpheme) เนื่องจากโดยปกติทั่วไปแล้ว ภาษาไทยมีการเขียนในลักษณะที่ติดต่อกันโดยไม่มีการใช้เครื่องหมายวรรคตอนใด ๆ ยกเว้นแต่มีการเว้นวรรคเป็นระยะ ๆ เพื่อให้ผู้อ่านได้หยุดพัก และทำความเข้าใจความหมายเป็นตอน ๆ ไปเท่านั้น แม้ว่าการเว้นวรรคในการเขียนบทความไม่ได้มีกฎเกณฑ์ที่ชัดเจนก็ตาม แต่ถ้ามีการใช้การเว้นวรรคด้วยความระมัดระวังแล้วก็จะสามารถช่วยลดความคลุมเครือของคำหรือประโยคได้

ไม่ว่าจะด้วยวิธีการใดก็ตามถ้าหากสามารถรู้เขตแบ่งของแต่ละคำได้แล้ว การจัดการกับข้อความนั้น ๆ ก็จะเป็นไปได้อย่างสะดวก และถูกต้อง ในระบบคอมพิวเตอร์จึงจำเป็นอย่างยิ่งที่จะต้องคำนวณหาขอบเขตของแต่ละคำ ให้ได้เพื่อที่จะสามารถลดภาระของผู้ใช้หรือเพื่อที่จะให้กระบวนการที่อยู่ในระดับที่ลึกกว่าสามารถทำงานต่อไปได้ ดังเช่น ฟังก์ชั่นการขอบขวา (Word wrapping) ใน Word processor การ

ตรวจคำผิด การค้นหาคำใน text หรือเป็นตัวช่วยในการกำหนดคำเพื่อทำการวิเคราะห์ต่อไปในระบบของเครื่องแปลภาษา

2.2.1 อัลกอริธึมที่ใช้ในการตัดคำ

อัลกอริธึมสำหรับการตัดคำโดยใช้พจนานุกรม ในการตัดคำหรือการหาขอบเขตของหน่วยคำในข้อความที่ต่อเนื่อง ถ้าหากเราเก็บคำทุกคำที่มีอยู่ในภาษานั้น ๆ ลงในพจนานุกรมทั้งหมดแล้วจากนั้นก็ค้นหา และเปรียบเทียบหาคำศัพท์นั้น ๆ ว่ามีอยู่ในพจนานุกรมหรือไม่ เพียงเท่านั้นก็จะสามารถหาขอบเขตของคำแต่ละคำได้ แต่ในความเป็นจริงแล้วไม่สามารถจะทำได้เนื่องจากว่าเป็นไปไม่ได้ที่จะบรรจุคำทุกคำที่มีอยู่ลงในพจนานุกรมได้ทั้งหมด โดยเฉพาะคำนามที่เป็นชื่อเฉพาะหรือคำที่เกิดขึ้นมาจากการใช้ใหม่ ๆ ดังนั้นการตัดคำถึงแม้ว่าจะอาศัยการเปรียบเทียบคำจากพจนานุกรมก็จำเป็นที่จะต้องยอมให้มีคำที่ไม่ได้บรรจุไว้ในพจนานุกรมเกิดขึ้นได้เช่นกัน

คำที่บรรจุอยู่ในพจนานุกรม ไม่จำเป็นที่จะต้องเป็นหน่วยคำที่ย่อยที่สุด ที่คงความหมายไว้เสมอไป คือ อาจจะเป็นคำประสม เช่น แม่น้ำ, ดูแล, ช่างทอง เป็นต้น หรือวลี เช่น แสงอาทิตย์ หนีเสือปะจระเข้ เป็นต้น ทั้งนี้ขึ้นอยู่กับว่าตำแหน่งของคำในวลี และโครงสร้างทางวากยสัมพันธ์ นั้น ๆ คงที่แน่นอนหรือไม่ และจะสามารถกำหนดความหมายที่ชัดเจนให้กับวลีนั้น ๆ ได้หรือไม่ โดย อัลกอริธึมที่ใช้ในการตัดคำในภาษาไทยโดยใช้พจนานุกรม มีดังต่อไปนี้

2.2.1.1 กฎทางอักขรวิธี

แม้ว่าการเขียนข้อความในภาษาไทยจะไม่มีการเว้นวรรคระหว่างคำ ไม่มีการใช้เครื่องหมายวรรคตอนที่ชัดเจน ไม่มีตัวชี้บ่งหน้าที่ทางไวยากรณ์ ไม่มีการแปรรูป (Inflection) ไม่มีการใช้ตัวอักษรใหญ่-เล็ก แต่ภาษาไทยก็มีกฎทางอักขรวิธีที่กำหนดลักษณะของการประสมอักษร การเว้นวรรคดังได้กล่าวไว้ในหัวข้อเรื่อง Longest matching และการขึ้นย่อหน้า ซึ่งทั้ง 3 ลักษณะนี้จะเป็นตัวช่วยในการชี้บ่งขอบเขตของการพิจารณาในการเปรียบเทียบกับคำในพจนานุกรม

1. การขึ้นย่อหน้าเป็นตัวชี้บ่งได้ดีถึงการสิ้นสุดของข้อความ
2. การเว้นวรรคเป็นตัวชี้บ่งถึงความเป็นไปได้ของการสิ้นสุดของคำหรือประโยค

กฎทางอักขรวิธีเป็นตัวชี้บ่งถึงความเป็นไปได้ของการพิจารณาในการที่จะแยกสายอักขระ (String) ออกเพื่อการพิจารณาคำหรือไม่ ดังนั้นจึงมีการแบ่งอักขระออกใหม่ดังนี้
อักขระกลุ่มที่ 1 กลุ่ม Non-spacing character คือ กลุ่มของรูปสระ วรรณยุกต์ และเครื่องหมายพิเศษประกอบการเขียนที่เมื่อประสมเข้ากับพยัญชนะใด ๆ แล้วไม่ทำให้มีการเคลื่อนขวาของตำแหน่งที่จะเขียนต่อไป อักขระในกลุ่มนี้จะไม่สามารถอยู่เดี่ยว ๆ ได้ อักขระกลุ่มที่ 2 กลุ่มอักขระที่จำเป็นที่จะต้องมีรูปพยัญชนะตามเสมอ อักขระกลุ่มที่ 3 กลุ่มอักขระที่จำเป็นที่จะต้องมีรูปพยัญชนะอยู่หน้าเสมอ อักขระกลุ่มที่ 4 กลุ่มของอักษรที่เป็นตัวการันต์ที่มีไม้ทัณฑฆาตบังคับข้างบน เช่น ย์ ใน ทิพย์, น์ ใน กาญจน์ เป็นต้น

เนื่องจากว่าตัวการันต์เป็นพยัญชนะสุดท้ายที่ไม่อ่านออกเสียง ดังนั้นจะไม่มีการพิจารณาให้เป็นตัวอักษรแรกของคำ อักขระกลุ่มที่ 5 กลุ่มของอักษรที่เหลือทั้งหมด

2.2.1.2 Longest matching

วิธีการนี้ถือได้ว่าเป็นวิธีทาง Heuristic อันหนึ่ง โดยจะทำการตรวจสอบคำในพจนานุกรมแล้วแยกเป็นคำ ๆ จากซ้ายไปขวา การตรวจสอบจะเริ่มตั้งแต่หน่วยของข้อความที่สั้นที่สุดที่จะสามารถคัดมาได้ เนื่องจากว่าภาษาไทยไม่มีเครื่องหมายบอกจุดสิ้นสุดของประโยค ดังนั้นจากเกณฑ์ทางอักขระดังในหัวข้อเรื่องกฎทางอักขรวิธีจะเป็นตัวบอกถึงขอบเขตของหน่วยข้อความได้ เมื่อตรวจสอบกับคำในพจนานุกรมแล้วถ้าไม่พบก็จะทำการลดความยาวของข้อความลงทีละตัวไปตามเกณฑ์ทางอักขรวิธี ยกตัวอย่างเช่น ในข้อความ "ความก้าวหน้าทางด้านวิทยาศาสตร์มีบทบาทสำคัญ" เมื่อไม่ปรากฏในพจนานุกรมแล้วข้อความนี้ก็จะถูกลดเหลือ "ความก้าวหน้าทางด้านวิทยาศาสตร์มีบทบาทสำคั" แล้วก็เป็น "ความก้าวหน้าทางด้านวิทยาศาสตร์มีบทบาทสำ" จนในที่สุดก็จะได้คำว่า "ความก้าวหน้า" เป็นคำแรก ผลของการตัดคำดังตารางที่ 2-1

ส่วนของคำที่ยาวที่สุด	ส่วนที่เหลือ
ความก้าวหน้า	ทางด้านวิทยาศาสตร์มีบทบาทสำคัญ
ทาง	ด้านวิทยาศาสตร์มีบทบาทสำคัญ
ด้าน	วิทยาศาสตร์มีบทบาทสำคัญ
วิทยาศาสตร์	มีบทบาทสำคัญ
มี	บทบาทสำคัญ
บทบาท	สำคัญ
สำคัญ	

ตารางที่ 2-1 ผลของการตัดคำโดยวิธีการ Longest matching ภายใต้เงื่อนไขของอักขรวิธี

วิธีการนี้จะให้ความถูกต้องในการตัดคำไปดังประมาณ 80% ข้อบกพร่องที่เห็นได้ชัดเจน คือ การเลือกขอบเขตของคำที่ยาวเกินไปตั้งแต่แรก จึงทำให้คำที่ตามมาผิดเพี้ยนไป เช่น ข้อความ "กีฬาเป็นการออกกำลังกายอย่างหนึ่ง" จะถูกแบ่งออกเป็น "กีฬา/เป็นการ/ออกกำลัง/กาย/อย่างหนึ่ง" เสมอ เพราะคำที่ยาวที่สุดที่จะพบได้ในพจนานุกรมสำหรับคำที่สองหลังจากที่ได้คำว่า "กีฬา" แล้วจะเป็นคำว่า "เป็นการ" เสมอ ดังนั้นจึงไม่สามารถที่จะแบ่งให้ได้คำว่า "การออกกำลังกาย" ที่มีความหมายที่ถูกต้องได้

2.2.1.3 วิธีการตัดคำให้ได้จำนวนคำ และคำที่ไม่มีในพจนานุกรมน้อยที่สุด

วิธีการตัดคำให้ได้จำนวนคำ และคำที่ไม่มีในพจนานุกรมน้อยที่สุดเป็นวิธีการทาง Heuristic อีกวิธีหนึ่ง และจะช่วยได้มากถ้าข้อความนั้นประกอบด้วยคำจำนวนมากหรือมีความยาวของข้อความมากพอสมควร วิธีนี้จะทำบนอัลกอริธึมของ Longest matching อีกทีหนึ่ง โดยจะทำการหาความเป็นไปได้ทั้งหมดในการตัดคำในข้อความนั้น ๆ

แบล็กแทรกกิ้ง (Backtracking) จะเริ่มกระทำหลังจากที่ได้คำตอบจากวิธี Longest matching แล้วจะทำไปทีละคำจากซ้ายไปขวา เช่น "กีฬา/เป็นการ/ออกกำลัง/กาย/อย่างหนึ่ง" ซึ่งเป็นผลที่ได้จากการตัดคำแบบ Longest matching แล้วทำการ Backtrack ที่คำว่า "กีฬา" แต่เนื่องจากว่าจะไม่เกิดประโยชน์อันใดในการที่จะยอมให้เกิดคำที่ไม่ม่ในพจนานุกรมอีก ดังนั้นการ Backtrack จึงสิ้นสุดตรงคำว่า "กีฬา" ต่อไปก็ทำ Backtracking ที่คำว่า "เป็นการ" เพื่อตรวจดูความเป็นไปได้ทั้งหมดในการแบ่งคำ ผลที่ได้คือจะสามารถแบ่งได้เป็น "เป็น/การ ..." และการทำ Backtracking ก็สิ้นสุดตรงคำว่า "เป็น"

เมื่อทำแบล็กแทรกกิ้งทุกคำแล้วก็ทำการคำนวณหา Cost ให้กับแต่ละ Path ที่เป็นไปได้ โดยบังคับให้มีการเกิดคำที่ไม่มีในพจนานุกรมให้น้อยที่สุดแล้วทำการเรียงผลของการตัดคำที่ได้ใหม่โดยให้คำตอบที่น่าจะเป็นไปได้มากที่สุดหรือถูกต้องมากที่สุดมาในอันดับแรก ดังแสดงในตารางที่ 2-2

ผลจากการ Backtracking	Cost
กีฬา/เป็น/การออกกำลังกาย/อย่างหนึ่ง	4
กีฬา/เป็นการ/ออกกำลัง/กาย/อย่างหนึ่ง	5
กีฬา/เป็นการ/ออก/กำลังกาย/อย่างหนึ่ง	5
กีฬา/เป็นการ/ออกกำลัง/กาย/อย่าง/หนึ่ง	6
กีฬา/เป็นการ/ออกกำลัง/กา/ยอ/ย่าง/หนึ่ง	7
กีฬา/เป็นการ/ออ/[ก]/กำลังกาย/อย่างหนึ่ง	11
กีฬา/เป็นการ/ออกกำลัง/กาย/อย่า/[ง]/หนึ่ง	12
กีฬา/เป็นการ/ออ/[ก]/กำลังกาย/อย่าง/หนึ่ง	12
กีฬา/เป็นการ/ออ/[ก]/กำลังกาย/อย่า/[ง]/หนึ่ง	18

ตารางที่ 2-2 ผลของการตัดคำโดยเรียงตาม Cost ที่คำนวณได้

2.3 วิชวลเบสิก 6.0 (Visual basic 6.0)

ในปัจจุบันระบบปฏิบัติการ (Operating system) ในลักษณะของวินโดวส์ได้เข้ามาแทนที่ระบบปฏิบัติการในลักษณะเดิม ซึ่งส่วนใหญ่ที่นิยมใช้กันอยู่ก็คือเอ็มเอสดอส (MS-DOS) เนื่องจากรูปแบบของจอภาพที่ใช้ติดต่อระหว่างคอมพิวเตอร์ และผู้ใช้อยู่ในรูปแบบของกราฟิคยูสเซอร์อินเตอร์เฟส (Graphic

user interface : GUI) ที่ใช้รูปภาพแทนคำสั่งต่าง ๆ แทน ซึ่งต่างจากเอ็มเอสดอสที่รูปแบบของคำสั่งจะอยู่ในรูปแบบของตัวอักษร และเป็นแบบป้อนทีละบรรทัด หรือที่เรียกว่าคอมมานด์ไลน์ (Command line) ซึ่งผู้ใช้จะต้องเรียนรู้ และจดจำรูปแบบของแต่ละคำสั่งให้ถูกต้อง และแม่นยำ จึงจะใช้งานโปรแกรมนั้น ๆ ได้เป็นอย่างดี และด้วยเหตุนี้ได้ส่งผลต่อการพัฒนาโปรแกรมเช่นเดียวกัน

วิชวลเบสิกเป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมนำมาใช้ในการพัฒนาโปรแกรมบนวินโดวส์ เนื่องจากเป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะแบบวิชวล (Visualize) ซึ่งเพียงแต่เลือกคอนโทรลที่เหมาะสมแล้วทำการวาดลงบนฟอร์ม ก็สามารถสร้างจอภาพที่ใช้สำหรับติดต่อผู้ใช้ รวมทั้งการใช้เทคนิคการเขียนโปรแกรมแบบอีเวนต์ดริฟเวนต์ (Event-driven) ซึ่งเป็นการเขียนโปรแกรมเพื่อกำหนดขั้นตอนการทำงานให้กับคอนโทรลต่าง ๆ ที่สร้างขึ้นตามเหตุการณ์ต่าง ๆ ที่เกิดขึ้น เช่น การเลื่อนเมาส์หรือการรับข้อมูลจากคีย์บอร์ด ฯลฯ เป็นต้น ประกอบกับภาษาที่ใช้เขียนโปรแกรมเป็นภาษาเบสิกซึ่งเป็นภาษาคอมพิวเตอร์ที่ผู้ใช้บนคอมพิวเตอร์ส่วนบุคคลส่วนใหญ่คุ้นเคย จึงส่งผลให้การพัฒนาโปรแกรมบนวินโดวส์ด้วยวิชวลเบสิกมีขั้นตอนน้อย กระทำได้ง่ายและสะดวกต่อการใช้งาน

2.3.1 ความสามารถของ VB6 กับการจัดการฐานข้อมูล

VB6 เป็นคอมไพล์เลอร์ที่มีความสามารถ และเหมาะสมเป็นอย่างมากในการพัฒนาระบบงานฐานข้อมูล ทั้งแบบที่ใช้งานคนเดียว แบบใช้หลายคนพร้อมกัน หรือการสร้างโปรแกรมเป็นฟร้อนท์เอน (Fronend) ของเซิร์ฟเวอร์ ซึ่งได้รับความนิยมอย่างกว้างขวางทั้งใน และต่างประเทศ ทำให้ปัจจุบันมีระบบงานฐานข้อมูลที่พัฒนาด้วย VB6 เป็นจำนวนมากเนื่องจาก

1. VB6 สามารถติดต่อ และจัดการฐานข้อมูลได้หลากหลายชนิด เช่น Microsoft access, dBase, Paradox, Foxpro และอื่น ๆ ซึ่ง VB6 มีส่วนโปรแกรมที่ติดต่อกับฐานข้อมูลได้โดยตรง (Database engine) ช่วยให้เราสามารถสร้างโปรแกรมติดต่อฐานข้อมูล และนำไปติดตั้งได้อย่างเบ็ดเสร็จ โดยเครื่องที่จะติดตั้งโปรแกรมนั้นไม่จำเป็นต้องมีระบบจัดการฐานข้อมูล (Database management system หรือ DBMS) อยู่ก่อนเลย

2. นอกจากความสามารถในการติดต่อกับฐานข้อมูลที่มีผู้ใช้งานคนเดียว หรือหลายคนพร้อมกันบนเครื่องพีซีแล้ว VB6 ยังสามารถติดต่อกับฐานข้อมูลขนาดใหญ่ หรือดาต้าเบสเซิร์ฟเวอร์ (Database server) ได้เป็นอย่างดีอีกด้วย

3. สามารถจัดการฐานข้อมูลได้อย่างง่ายดายเนื่องจาก VB6 มีเครื่องมือที่เรียกว่าดาต้าคอนโทรล (Data control) ทำให้ลดเวลาในการเขียนโปรแกรมเพื่อติดต่อ และจัดการกับข้อมูลอีกด้วย

4. VB6 มีเครื่องมือที่เรียกว่าแอปพลิเคชั่นวิซาด (Application wizard) ทำให้เราสามารถสร้างโปรแกรมได้โดยไม่ต้องมีประสบการณ์มาก่อน เพียงตอบคำถามบางอย่างกับวิซาดเท่านั้น เราก็สามารถจะสร้างระบบงานที่ใช้งานได้จริง และใช้เวลาในการเขียนโปรแกรมน้อยมาก

5. มีเครื่องมือในการสร้างรายงาน กราฟ และการแสดงรูปภาพจากฐานข้อมูลได้โดยตรง

6. สามารถสร้างระบบงานที่ใช้งานได้จริงเพราะ VB6 มีเครื่องมือในการตรวจสอบความผิดพลาดของข้อมูลนำเข้า (Input) ก่อนการบันทึกเข้าไปในฐานข้อมูล เช่น การใช้งาน Maskededit เป็นต้น การยกเลิกการบันทึกข้อมูลที่บันทึกไปแล้ว รวมถึงการป้องกันความผิดพลาดที่อาจเกิดขึ้น ด้วยการใช้คำสั่ง On error... ทำให้โปรแกรมที่เราพัฒนาขึ้นมามีความเชื่อถือได้สูง

7. เราสามารถสร้างระบบงานฐานข้อมูลเพื่อใช้งานบนอินเตอร์เน็ตได้โดยอาศัยเอ็กซ์ทีฟเอ็กซ์คอนโทรล (ActiveX control)

8. มีวิซาดเพื่อช่วยในการสร้างแผ่นติดตั้งโปรแกรม (Setup disk) ทำให้โปรแกรมเมอร์ไม่ต้องยุ่งยากในการเรียนรู้โปรแกรมเพื่อสร้างแผ่นติดตั้งโปรแกรมอื่น ๆ ที่ค่อนข้างยุ่งยากซับซ้อน เช่น โปรแกรม InstallShield เป็นต้น โดยเราสามารถสร้างแผ่นเพื่อติดตั้งโปรแกรมได้อย่างง่ายดาย

2.3.2 ไมโครซอฟต์แอกเซส (Microsoft access : MS-Access)

MS-Access เป็นระบบจัดการฐานข้อมูล (DBMS) ที่มีความสามารถสูงโปรแกรมหนึ่ง และเป็นส่วนหนึ่งของโปรแกรมไมโครซอฟต์ออฟฟิศ (Microsoft office) ซึ่งมีอยู่หลายเวอร์ชัน ตั้งแต่เวอร์ชัน 2.0, 95, 97 และ 2000 ตามลำดับ สำหรับ VB6 นั้นสามารถติดต่อกับทุกเวอร์ชันของ MS-Access ได้

2.3.3 การสร้าง และการจัดการฐานข้อมูล

หัวข้อนี้จะอธิบายวิธีสร้างฐานข้อมูลตามที่ได้ออกแบบไว้โดยใช้โปรแกรม Visual data manager

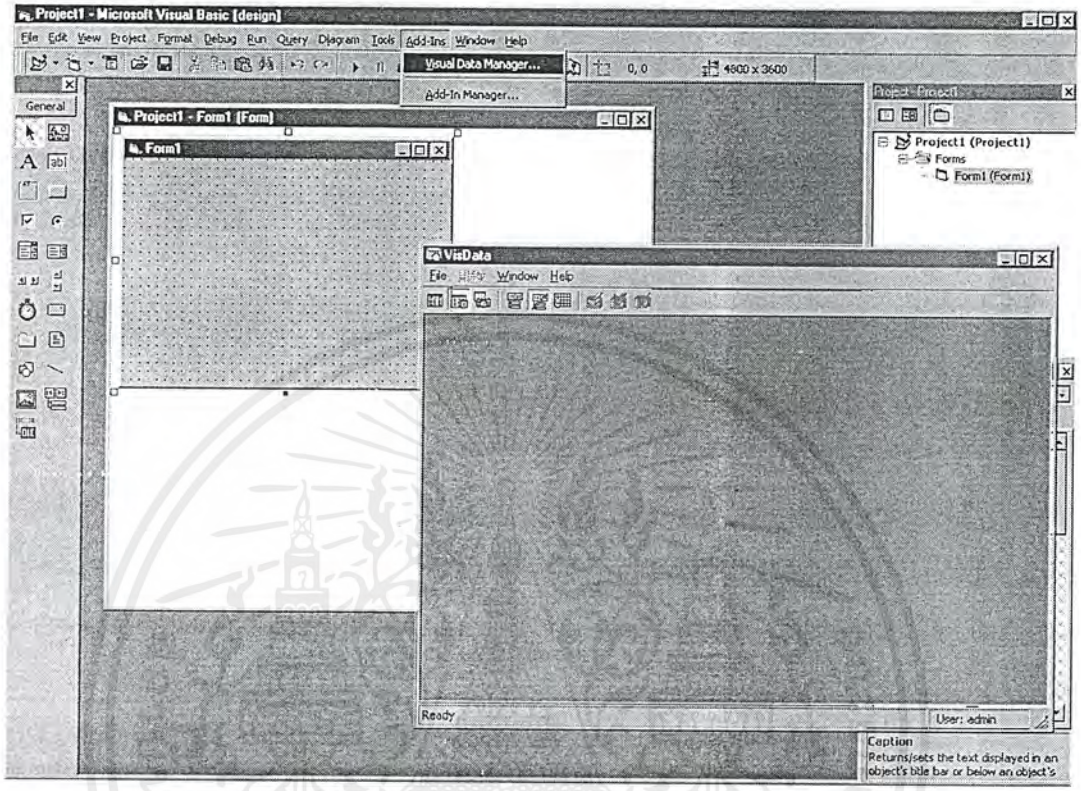
2.3.4 Visual data manager

Visual data manager เป็นโปรแกรมที่ให้มากับ VB6 โดยโปรแกรมนี้จะช่วยให้เราสร้าง และจัดการกับฐานข้อมูลของเราได้ เช่น การสร้างตาราง ลบตารางที่มีอยู่ การสร้างอินเด็กซ์ การเพิ่มเติม แก้ไขข้อมูล การใช้งาน SQL เป็นต้น สำหรับฐานข้อมูลที่ Visual data manager สนับสนุน คือ MS-Access, dBase, FoxPro, Paradox และฐานข้อมูลอื่น ๆ โดยผ่านทาง ODBC (Open database connectivity) รวมถึงการใช้เท็กซ์ไฟล์ธรรมดาอีกด้วย

ถึงแม้ว่า Visual data manager จะสามารถติดต่อกับฐานข้อมูลได้หลากหลายก็ตาม แต่ที่จะใช้ในวิทยานิพนธ์ฉบับนี้จะใช้ฐานข้อมูล MS-Access ซึ่งได้รับความนิยมและง่ายต่อการใช้งานมาก

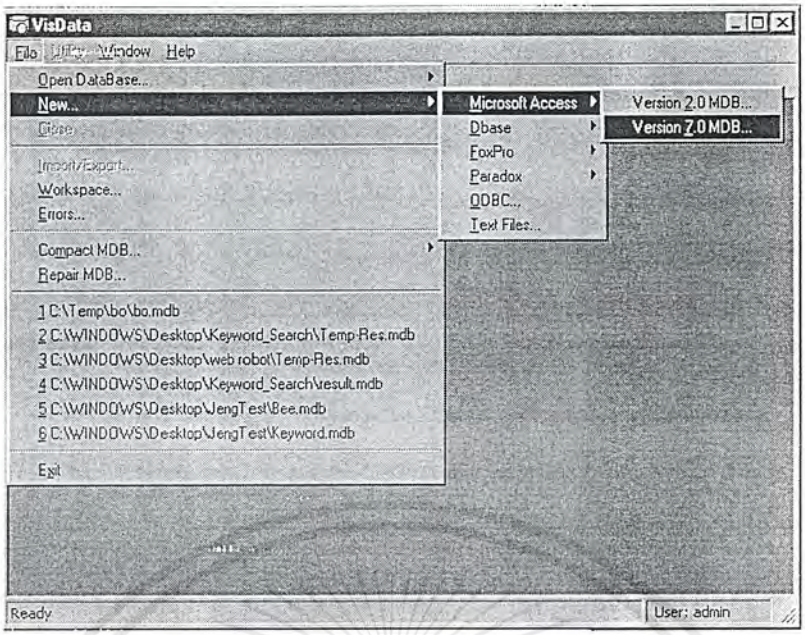
2.3.5 วิธีการใช้งานโปรแกรม Visual data manager

การเปิดโปรแกรม Visual data manager สามารถทำได้โดยเลือกเมนูคำสั่ง Add-Ins> Visual data manager ใน VB6 หลังจากนั้นจะปรากฏหน้าจอของโปรแกรมดังรูปที่ 2-2



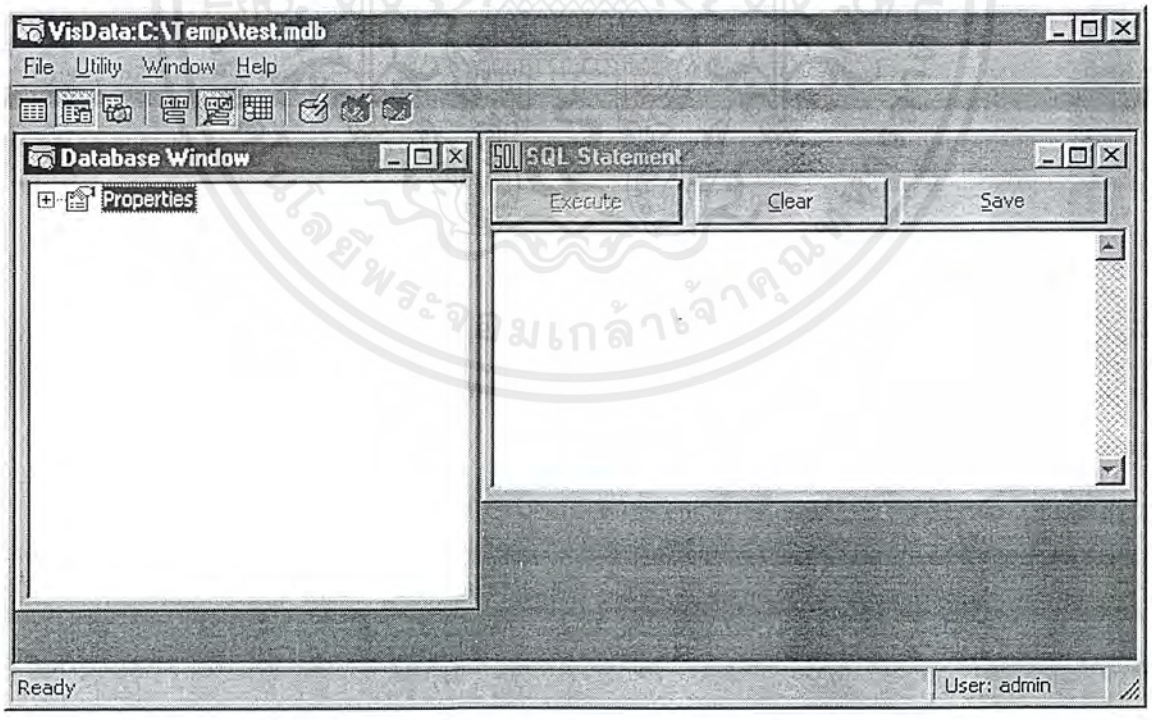
รูปที่ 2-2 หน้าจอโปรแกรม Visual data manager

หลังจากที่เปิดโปรแกรม Visual data manager แล้วให้เลือกคำสั่ง File>New...>Microsoft Access>Version 7.0 MDB.. เพื่อสร้างฐานข้อมูลแบบ MS-Access ดังรูปที่ 2-3



รูปที่ 2-3 การสร้างฐานข้อมูล MS-Access โดยใช้ Visual data manager

หลังจากนั้นทำการเลือกไดร์ฟ ไดเร็กทอรี่ที่ต้องการจะสร้างฐานข้อมูล และใส่ชื่อฐานข้อมูลที่ต้องการในช่อง File name ซึ่งเราก็จะได้ฐานข้อมูลใหม่ตามชื่อที่เราตั้งเอาไว้ รวมทั้งจะปรากฏหน้าจอ Visual data manager ขึ้นมา ดังรูปที่ 2-4



รูปที่ 2-4 ฐานข้อมูลที่ถูกสร้างโดยใช้โปรแกรม Visual data manager

2.4 Common gateway interface (CGI)

เมื่อเกิดระบบเครือข่ายเวิร์ลไวด์เว็บขึ้นมาใช้งานจนเป็นที่นิยมหลาย ๆ เว็บไซต์ (Web site) เริ่มต้องการนำเสนอข้อมูล ภายในองค์กรที่เคยใช้งานกับโปรแกรมประยุกต์ของตน ภายในองค์กรมาใช้งานผ่านเว็บเพจหรือโฮมเพจ (Homepage) ของตน จึงเกิดปัญหาว่าจะสามารถทำอย่างไร ทั้งนี้เพราะทั้งสองแอปพลิเคชั่นนี้อยู่คนละส่วนกัน และวิธีการทำงานก็แตกต่างกันอย่างสิ้นเชิง ทางออกก็คือการพัฒนาแอปพลิเคชั่นในลักษณะเหมือนโปรแกรมประยุกต์ที่องค์กรใช้งานอยู่ โดยอาศัยหลักการของซีจีไอในการพัฒนา แต่นี่ยังเป็นเพียงแค่จุดเริ่มต้นของความต้องการเท่านั้น เพราะปัจจุบันเราจะเห็นได้ว่ามีแอปพลิเคชั่น หลากหลายรูปแบบบนระบบเว็บ เช่น การให้บริการส่งเพจ การให้บริการค้นหางาน การให้บริการความช่วยเหลือแบบออนไลน์ การให้บริการการลงทะเบียน เป็นต้น ซึ่งแอปพลิเคชั่นเหล่านี้เกิดจากความต้องการที่หลากหลาย และต่างความคิด รวมไปถึงวิสัยทัศน์ของแต่ละคนในการที่จะคิดประยุกต์และร่วมกันสร้างกิจกรรมต่าง ๆ บนระบบเว็บ จนทำให้การใช้งานระบบเว็บนี้กลายเป็นส่วนสำคัญหลักของเครือข่ายอินเตอร์เน็ตในปัจจุบัน และด้วยความสามารถของหลักการซีจีไอนี้เองทำให้หลาย ๆ องค์กรต้องการนำมาประยุกต์ใช้ในองค์กรจนเกิดคำว่า "แอปพลิเคชั่นในอนาคต คือ แอปพลิเคชั่นที่ใช้งานผ่านเว็บเบราวเซอร์หรือใช้งานภายใต้พื้นฐานเว็บ (Web-based application)"

เราทราบว่าข้อมูลที่ให้บริการผ่านเวิร์ลไวด์เว็บนั้น จะต้องถูกจัดเก็บอยู่ในรูปแบบของเอกสารแฮชทีเอ็มแอล เมื่อเซิร์ฟเวอร์ได้รับการร้องขอไฟล์จากไคลเอ็นท์ เว็บเซิร์ฟเวอร์จะค้นหา และส่งไฟล์ที่ไคลเอ็นท์ต้องการกลับไป ซึ่งปัญหาของข้อมูลที่เก็บด้วยรูปแบบแฮชทีเอ็มแอล คือ เมื่อจะต้องมีการอัพเดตข้อมูลจะเป็นงานที่ยุ่งยาก และเสียเวลาเป็นอย่างมาก เพราะว่าไฟล์เอกสารแฮชทีเอ็มแอลมีลักษณะการจัดเก็บแบบตายตัว (Static) ยิ่งถ้ามีข้อมูลมาก ๆ ด้วยแล้ว การจัดเก็บข้อมูลแยกออกเป็นไฟล์ ๆ ยิ่งจะทำให้ดูแลแก้ไขได้ยากมากขึ้นเป็นเงาตามตัว โดยเป้าหมายของการใช้ซีจีไออย่างหนึ่งก็คือ ทำให้เอกสารแฮชทีเอ็มแอลที่ผู้ใช้ร้องขอเข้ามามีความยืดหยุ่นหรือที่เรียกว่าเป็นแบบเปลี่ยนแปลงหรือแบบไดนามิค (Dynamic)

วิธีการทำเอกสารแฮชทีเอ็มแอลให้มีความเป็นแบบไดนามิค คือ แทนที่จะเก็บข้อมูลแยกเป็นไฟล์แฮชทีเอ็มแอลหลาย ๆ ไฟล์ เราก็อาจจะเก็บข้อมูลทั้งหมดไว้ในไฟล์เดียว เมื่อผู้ใช้หรือไคลเอ็นท์ต้องการข้อมูลอะไรสักอย่างก็กำหนดให้ป้อนเงื่อนไขที่ต้องการให้แก่ซีจีไอ หลังจากนั้นซีจีไอจะไปค้นหรือดึงเอาเฉพาะข้อมูลที่ตรงตามที่ผู้ใช้หรือไคลเอ็นท์ต้องการ จากนั้นจึงนำข้อมูลนั้นมาสร้างเป็นเอกสารแฮชทีเอ็มแอลจัดส่งกลับไปแสดงยังผู้ใช้หรือไคลเอ็นท์ ดังนั้นเอกสารแฮชทีเอ็มแอลที่แต่ละคนได้รับอาจไม่เหมือนกันโดยขึ้นอยู่กับเงื่อนไขความต้องการของผู้ใช้ ในกรณีนี้ซีจีไอจะทำหน้าที่เป็นประตู หรือเกตเวย์ (Gateway) ระหว่างฐานข้อมูลในเซิร์ฟเวอร์กับไคลเอ็นท์นั่นเอง

ลักษณะการทำงานของซีจีไอต้องอาศัยการประมวลผลที่เซิร์ฟเวอร์ แล้วสร้างคำตอบออกมาเป็นเนื้อหาแบบแฮชทีเอ็มแอล จากนั้นจึงส่งเนื้อหากลับไปให้ไคลเอ็นท์ ดังนั้นเซิร์ฟเวอร์ใดที่ยอมให้มีการรันซีจีไอได้ จึงต้องทำงานหนักกว่าเซิร์ฟเวอร์ที่ให้บริการเอกสารแฮชทีเอ็มแอลเพียงอย่างเดียว แนวความ

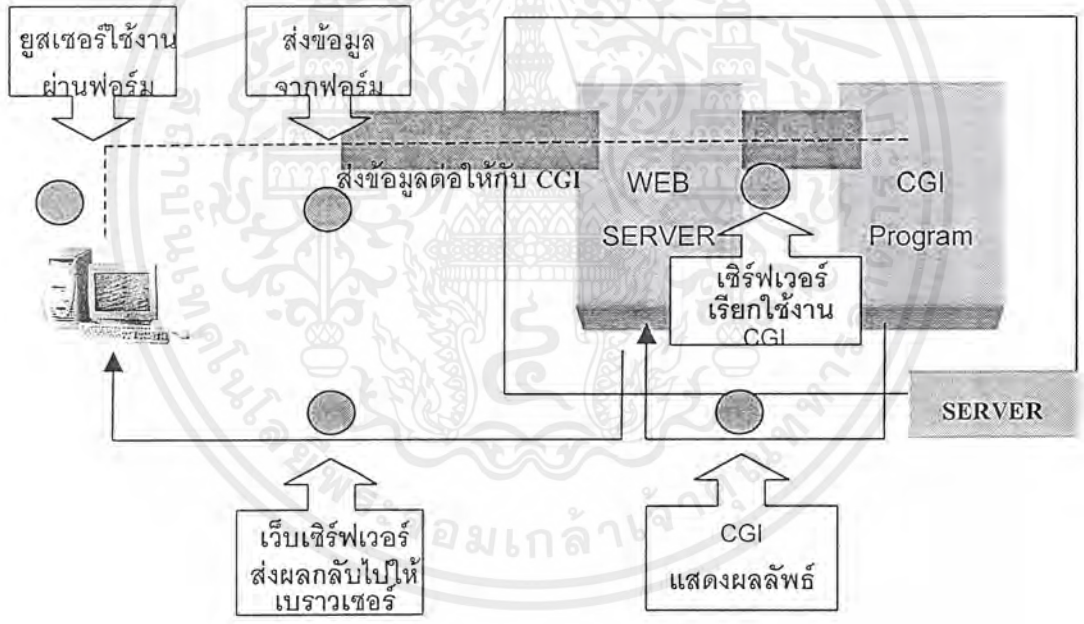
คิดกาทำงานของซีจีไอจะเป็นแบบที่เรียกว่ารวมศูนย์ (Centralize) คือ งานทุกอย่างต้องวิ่งเข้ามารันที่เซิร์ฟเวอร์หมด

ซีจีไอ (Common gateway interface) เป็นเทคโนโลยีที่ใช้เชื่อมต่อกับเซิร์ฟเวอร์ในอินเตอร์เน็ต โดยมีการเรียกจากไคลเอ็นท์ (Request) พร้อมกับส่งข้อมูลไปให้ เมื่อเซิร์ฟเวอร์ได้รับการร้องขอแล้วก็จะมีการรันโปรแกรมเพื่อทำตามที่ไคลเอ็นท์ได้ร้องขอมา (มักใช้ข้อมูลที่ได้รับมาจากไคลเอ็นท์เป็นอินพุทของโปรแกรม)

เมื่อโปรแกรมทำงานเสร็จ เซิร์ฟเวอร์ก็จะส่งผลลัพธ์กลับไปให้ไคลเอ็นท์ที่ได้ร้องขอ (Response) ซึ่งมักจะอยู่ในรูปของไฟล์แฮชทีเอ็มแอล

2.4.1 องค์ประกอบของซีจีไอ

จากแนวความคิดการทำงานของซีจีไอนั้นจะเห็นว่าซีจีไอจะต้องประกอบไปด้วยส่วนต่าง ๆ ซึ่งมีรายละเอียดดังนี้



รูปที่ 2-5 แนวความคิดการทำงานของซีจีไอ

2.4.1.1 ไคลเอ็นท์ และเว็บเซิร์ฟเวอร์

เพราะซีจีไอเป็นเทคโนโลยีที่อยู่ในฝั่งของเซิร์ฟเวอร์ ดังนั้นเว็บเซิร์ฟเวอร์ที่ติดตั้งต้องรองรับการใช้งานซีจีไอโดยมีเว็บเซิร์ฟเวอร์หลาย ๆ ตัวที่รองรับการใช้งานซีจีไออยู่แล้ว เช่น OmniHTTPd, Web site, Web server เป็นต้น ส่วนเว็บเซิร์ฟเวอร์จากค่ายไมโครซอฟต์ก็รองรับการทำงานของซีจีไอเช่นเดียวกันทั้ง PWS (Personal web server) และ IIS (Internet information server)

2.4.1.2 Standard input/Stanadrd output

เพราะซีจีไอมีแหล่งกำเนิดมาจากยูนิกซ์ ซึ่งมีการสื่อสารกันโดยใช้ไฟล์เป็นหลัก แต่ละไฟล์ก็จะมีหมายเลข (File descriptor) เอาไว้อ้างอิง และหมายเลขไฟล์สำคัญที่ควรจะรู้ก็คือ STDin และ STDout

สำหรับการใช้งานซีจีไอนั้น Stdin และ Stdout จะเป็นการรับเอาข้อมูลจากเว็บเซิร์ฟเวอร์ซึ่งได้รับมาจากไคลเอ็นท์อีกที และการส่งผลลัพธ์กลับไปยังเว็บเซิร์ฟเวอร์เพื่อส่งต่อไปยังไคลเอ็นท์

2.4.1.3 ตัวแปรสภาพแวดล้อม (Environment variable)

การรันซีจีไอโปรแกรมโดยใช้ข้อมูลที่มาจากไคลเอ็นท์แต่เพียงอย่างเดียวนั้นยังไม่เพียงพอ ซีจีไอ โปรแกรมอาจจะต้องการข้อมูลอื่น ๆ นำมาประกอบการใช้งาน เช่น อาจจะต้องการทราบว่าไคลเอ็นท์ที่ร้องขอเข้ามี IP address เป็นอะไร หรือ ต้องการทราบว่าเว็บเซิร์ฟเวอร์ชื่อว่าอะไร ฯลฯ ซึ่งข้อมูลเหล่านี้จะเก็บไว้ในตัวแปรสภาพแวดล้อม สำหรับตัวแปรสภาพแวดล้อมที่ใช้กันบ่อย ๆ แสดงได้ดังตารางที่ 2-3

ตัวแปร	คำอธิบาย
REQUEST_PROTOCOL	ชื่อของโพรโตคอล และเวอร์ชันที่รองรับการร้องขอที่เข้ามา เช่น HTTP/1.0
REQUEST_METHOD	วิธีการส่งข้อมูลไปให้กับเซิร์ฟเวอร์
EXECUTABLE_PATH	เป็นพาธของซีจีไอโปรแกรม (เป็น logical path เซิร์ฟเวอร์ต้องนำมาแปลงเป็น Physical path)
QUERY_STRING	เป็นข้อมูลขนาดเล็กที่ส่งพ่วงไปกับ URL เพื่อใช้เป็นอินพุทให้ซีจีไอโปรแกรม
SERVER_NAME	ชื่อของเซิร์ฟเวอร์หรือ IP address
SERVER_PORT	หมายเลขพอร์ตที่ใช้กับ HTTP request
SERVER_ADMIN	อีเมลล์แอดเดรสของผู้ที่ดูแลเซิร์ฟเวอร์
REMOTE_HOST	เป็นชื่อคอมพิวเตอร์หรือโดเมนเนมของไคลเอ็นท์
REMOTE_ADDR	เป็น IP address ของไคลเอ็นท์
CONTENT_TYPE	ประเภทข้อมูลที่ส่งไปยังเซิร์ฟเวอร์ด้วยเมธอด POST
CONTENT_LENGTH	ขนาดของข้อมูลมีหน่วยเป็นไบต์ที่ถูกส่งไปยังเซิร์ฟเวอร์ด้วยเมธอด POST

ตารางที่ 2-3 ตัวแปรสภาพแวดล้อมที่นิยมใช้กัน

2.4.1.4 ซีจีไอ โปรแกรม

ซีจีไอ โปรแกรมเป็นโปรแกรมที่จะถูกเรียกใช้งานจากเว็บเซิร์ฟเวอร์ ซึ่งมักจะมีขนาดเล็กทั้งนี้เพื่อจะสามารถทำงาน และตอบสนองไคลเอ็นท์ได้อย่างรวดเร็ว โดยส่วนใหญ่แล้วเว็บเซิร์ฟเวอร์จะเก็บโปรแกรมซีจีไอเอาไว้ในไดเร็กทอรี่ CGI-BIN หรือ Scripts

2.4.1.5 ภาษาที่ใช้ในการพัฒนาซีจีไอโปรแกรม

ภาษาซี/ซีพลัส ๆ (C/C++ langauges) สามารถพัฒนาบนระบบปฏิบัติการยูนิกซ์ (UNIX) วินโดวส์ และแมคอินทอช (Macintosh) ได้ จึงเป็นภาษาที่มีความนิยมมาก แต่ว่าภาษาซี/ซีพลัส ๆ นั้นมีลักษณะโครงสร้างของการเขียนโปรแกรมที่ค่อนข้างยาก เนื่องจากในตัวของภาษาเองมีความเข้มงวดมาก และยังขาดความสามารถในเรื่องของแพตเทิร์นแมตช์ชิ่ง (Pattern-matching) แม้ว่าจะสามารถเขียนในรูปแบบโมดูล และฟังก์ชั่น ซึ่งช่วยทำงานได้สะดวกขึ้น อย่างไรก็ตามซีจีไอโปรแกรมที่เขียนด้วยภาษาซีก็สามารถเรียกใช้ค่าตัวแปรสภาพแวดล้อมได้ และก็มีการใช้งานได้ง่าย

ซีเชลล์ (C shell) มักไม่ค่อยได้รับความนิยมในการพัฒนาโปรแกรมซีจีไอ เนื่องจากว่าสามารถใช้ได้เพียงบนระบบปฏิบัติการยูนิกซ์ (UNIX) อย่างเดียว และมีข้อจำกัดเข้มงวด

แอปเปิลสคริปต์ (Apple scripts) สามารพัฒนาบนระบบปฏิบัติการแมคอินทอช (Macintosh) ได้เพียงอย่างเดียวเท่านั้น แต่มีข้อดีในเรื่องของการจัดการกับข้อมูลได้หลากหลาย เนื่องจากว่าตัวแอปเปิลสคริปต์สามารถทำอินเตอร์เฟสได้ดีกับแอปพลิเคชั่นของแมคอินทอชหรืองานที่อยู่บนเครื่องแมค

เพิร์ล (Perl) สามารถพัฒนาบนระบบปฏิบัติการยูนิกซ์ (UNIX) วินโดวส์ และรวมไปถึงแมคอินทอช ด้วยเหตุนี้ภาษาเพิร์ลจึงเป็นภาษาที่ใช้ในการเขียนซีจีไอ โปรแกรมอย่างแพร่หลายที่สุด เนื่องจากว่าเพิร์ลมีข้อดีมากมายที่ช่วยสนับสนุนในการเขียนโปรแกรม

2.4.2 วิธีการส่งข้อมูลให้กับเซิร์ฟเวอร์

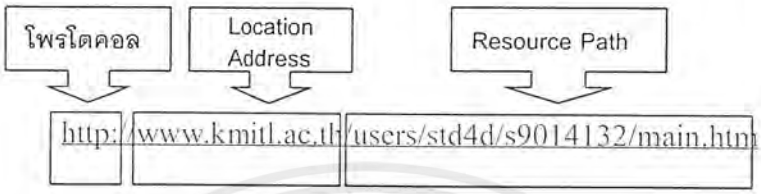
ในการส่งข้อมูลจากไคลเอ็นท์ที่มักจะทำงานบนเว็บเบราวเซอร์ไปยังเซิร์ฟเวอร์ เพื่อให้เริ่มทำงานนั้นมีอยู่ 2 วิธีที่นิยมกัน ได้แก่

เมธอด GET เป็นการส่งข้อมูลไปยังเซิร์ฟเวอร์โดยส่งไปในตัวแปรสภาพแวดล้อม QUERY_STRING ซึ่งจะพ่วงท้าย URL โดยมีเครื่องหมาย "?" คั่น แต่จะมีข้อจำกัดว่าข้อมูลนั้นจะมีความยาวไม่เกิน 256 ตัวอักษร

เมธอด POST เป็นการส่งข้อมูลที่มีขนาดใหญ่ โดยจะเตรียมเนื้อที่หน่วยความจำไว้ก่อน โดยตรวจสอบจากตัวแปรสภาพแวดล้อม CONTENT_LENGTH

2.4.3 ยูอาร์แอล

URL ย่อมาจาก Uniform resource locators ซึ่งเรามักจะได้ยินว่าการใช้งานอินเตอร์เน็ตต่าง ๆ จะสามารถใช้งานได้ภายใต้พื้นฐานของระบบเว็บหรือที่นิยมเรียกว่า "Web-base" ทั้งนี้เพราะโปรแกรมเว็บเบราเซอร์อาศัยหลักการของยูอาร์แอลในการสร้างความหลากหลายของการใช้งานหรือขอบริการในรูปแบบต่าง ๆ ซึ่งยูอาร์แอลก็คือการระบุถึงแหล่งข้อมูลที่ต้องการขอบริการ โดยมีโครงสร้างคร่าว ๆ ดังรูปที่ 2-6



รูปที่ 2-6 โครงสร้างของยูอาร์แอล

2.5 โครงสร้างเอกสารแฮชทีเอ็มแอล

ในเว็บเพจสามารถแบ่งโครงสร้างหลัก ๆ ออกเป็น 2 ส่วน คือ ส่วนแรกเราเรียกว่าเฮดเดอร์ (header) เป็นส่วนสำหรับกำหนดค่าต่าง ๆ สำหรับเว็บเพจนั้น ๆ ส่วนที่สองเป็นส่วนเนื้อหา (body) หรือตัวข้อมูลของเว็บเพจ

การแบ่งส่วนนั้นเราใช้แท็กเป็นตัวกำหนด โดยเริ่มจากเขียนแท็ก <html> </html> ครอบคำสั่งและข้อมูลทั้งหมดในเว็บเพจ แล้วก็ใช้แท็ก <head> </head> ครอบส่วนเฮดเดอร์ และแท็ก <body> </body> ครอบส่วนที่เป็นข้อมูลของเว็บเพจ

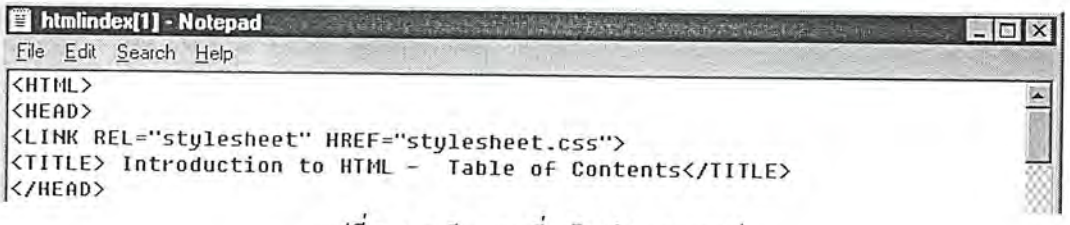
```
<html>
<head> </head>
<body>
ส่วนนี้เป็นเนื้อหาที่ต้องการใส่
</body>
</html>
```

รูปที่ 2-7 โครงสร้างเอกสารแฮชทีเอ็มแอล

2.5.1 ส่วนเฮดเดอร์

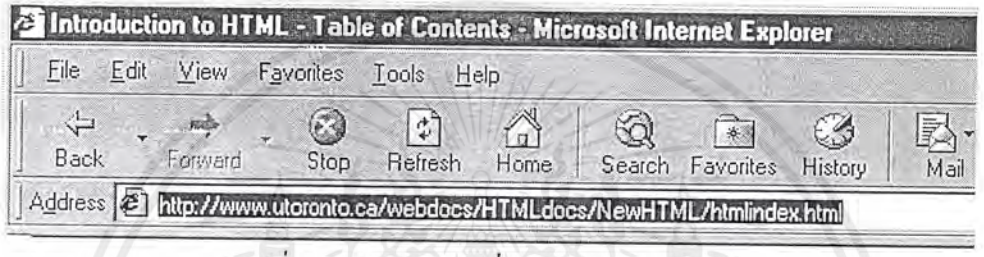
ภายใน <head> </head> ซึ่งเป็นส่วนเฮดเดอร์ โดยเราจะสามารถกำหนดค่าต่าง ๆ ให้กับเว็บเพจได้ ข้อมูลที่ใส่ในส่วนนี้จะไม่แสดงผลออกมาในเว็บเพจโดยตรง โดยข้อมูลที่มักจะกำหนดไว้ในส่วนนี้เสมอคือการกำหนดชื่อเรื่อง (title) ที่แสดงอยู่ที่แถบแสดงชื่อเรื่อง (title bar) หรือแถบบนสุดของ

เว็บเบราเซอร์ ข้อความ title นี้ส่วนใหญ่แล้วจะเป็นชื่อของเว็บเพจนั้น ๆ การกำหนด title ทำได้โดยใส่แท็กชื่อ title ลงระหว่าง <head> กับ </head> ดังรูปที่ 2-8



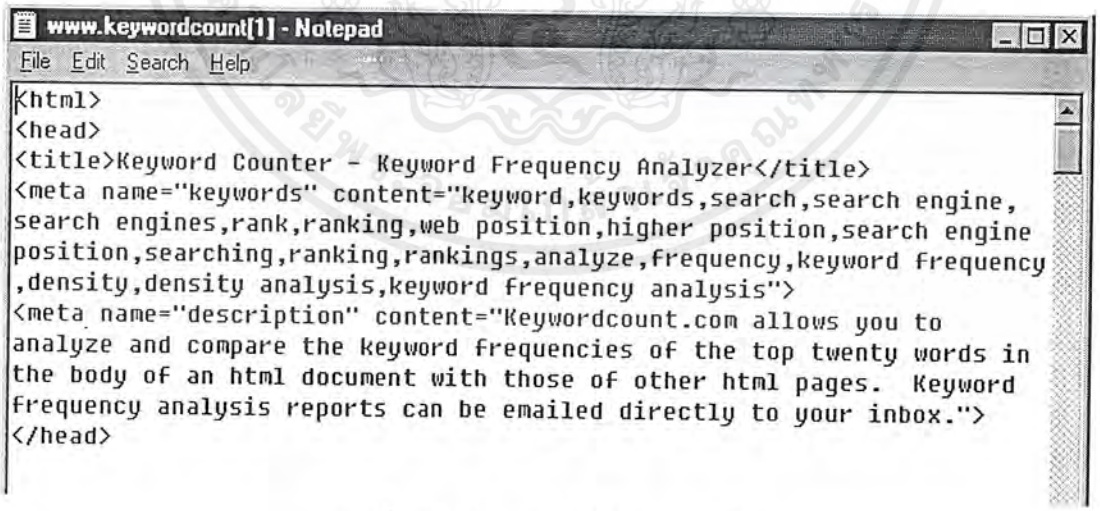
รูปที่ 2-8 แท็ก title ที่อยู่ในส่วนเฮดเดอร์

โดยผลลัพธ์ที่ได้แสดงออกมาในเว็บเบราเซอร์ดังรูปที่ 2-9



รูปที่ 2-9 เว็บเบราเซอร์ที่แสดง title ของเว็บเพจ

นอกจากการกำหนด title bar แล้วยังมีอีกหลายอย่างที่มักจะใส่ลงในส่วนเฮดเดอร์ เพื่อบอกรายละเอียดของเว็บเพจนั้น ๆ อย่างเช่น ชื่อผู้เขียนเว็บเพจนั้น เนื้อหาย่อ ๆ ของเว็บเพจ ส่วนโค้ดเริ่มต้นของจาวาสคริปต์ สไตล์ชีต และอื่น ๆ



รูปที่ 2-10 แท็ก meta ในส่วนเฮดเดอร์

จากรูปที่ 2-10 แสดงตัวอย่างการใช้แท็ก meta ซึ่งผู้เขียนจะใช้แท็ก meta เป็นตัวบอกถึงรายระเอียดต่าง ๆ ที่เกี่ยวกับเว็บเพจ โดยเราใช้แท็ก meta name = “keywords” เป็นตัวบอกถึงคำที่เป็นคีย์เวิร์ดในเว็บเพจนั้น ๆ และแท็ก meta name = “description” เอาไว้บอกถึงคำบรรยายเกี่ยวกับเว็บเพจนั้น ๆ

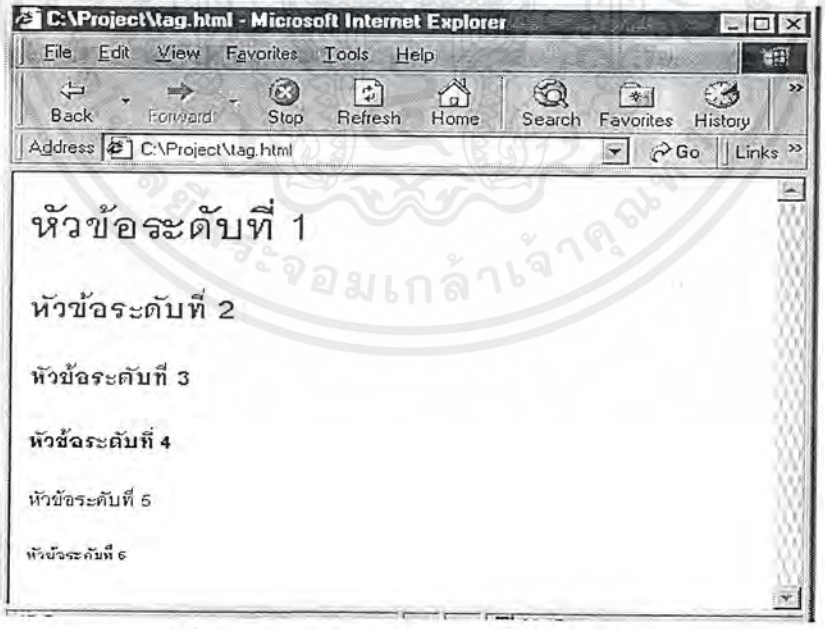
2.5.2 ส่วนเนื้อหา

ส่วนเนื้อหาหรือ body เป็นส่วนสำหรับใช้เขียนข้อมูลกับคำสั่งภาษาแฮชทีเอ็มแอลที่จะแสดงออกมาในเว็บเพจโดยตรง ไม่ว่าจะเป็นข้อความ ภาพ เสียง ลิงก์จะถูกกำหนดไว้ในส่วนเนื้อหานี้ การกำหนดส่วนเนื้อหาทำได้โดยใช้แท็ก <body> </body> ดังตัอย่างที่ได้กล่าวไปแล้ว

ในการกำหนดหัวเรื่องด้วยภาษาแฮชทีเอ็มแอลจะใช้แท็ก <h1>, <h2>, <h3>, <h4>, <h5> และ <h6> โดยแต่ละแท็กจะต้องมีแท็กปิดด้วย เช่น แท็ก <h1> ก็มีแท็กปิด </h1> เป็นต้น ข้อความที่กำหนดเป็นหัวเรื่องจะเป็นตัวหนา ส่วนตัวเลข 1-6 ที่อยู่ในชื่อแท็กแสดงถึงระดับความสำคัญของหัวข้อและขนาดของตัวอักษรของหัวเรื่องนั้น เริ่มจากแท็ก <h1> ซึ่งเป็นหัวข้อหลัก และจะมีตัวอักษรขนาดใหญ่ที่สุด ไปจนถึงแท็ก <h6> เป็นหัวข้อย่อยที่สุด และมีตัวอักษรขนาดเล็กที่สุด

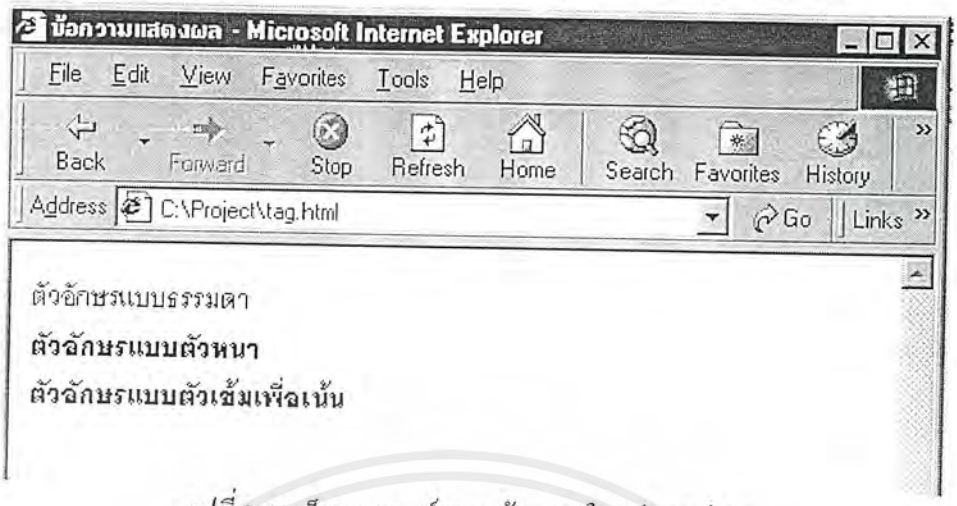
```
<h1>หัวข้อระดับที่ 1</h1>
<h2>หัวข้อระดับที่ 2</h2>
<h3>หัวข้อระดับที่ 3</h3>
<h4>หัวข้อระดับที่ 4</h4>
<h5>หัวข้อระดับที่ 5</h5>
<h6>หัวข้อระดับที่ 6</h6>
```

รูปที่ 2-11 แท็กการกำหนดหัวเรื่อง



รูปที่ 2-12 เว็บเบราเซอร์แสดงหัวเรื่องขนาดต่าง ๆ

ตัวหนา และตัวเข้ม เราจะใช้แท็ก <b>, <big> และ ตามลำดับ



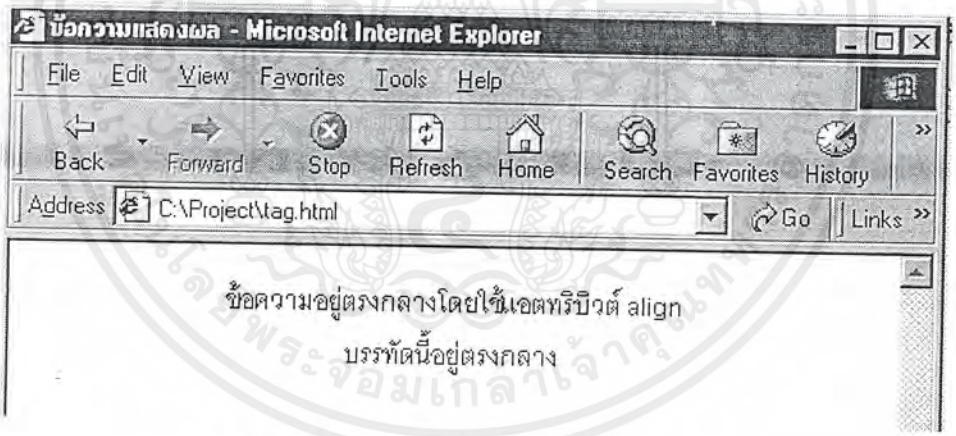
รูปที่ 2-13 เว็บเบราเซอร์แสดงข้อความในรูปแบบต่าง ๆ

ถ้าเราต้องการจัดให้ข้อความอยู่ตรงกลางจะทำด้วยแอตทริบิวต์ align ของแท็ก <p> หรือใช้แท็ก

<p align = center> ข้อความอยู่ตรงกลางโดยใช้แอตทริบิวต์ align

 บรรทัดนี้อยู่ตรงกลาง

รูปที่2-14 แท็กที่ใช้กำหนดให้ข้อความอยู่ตรงกลาง

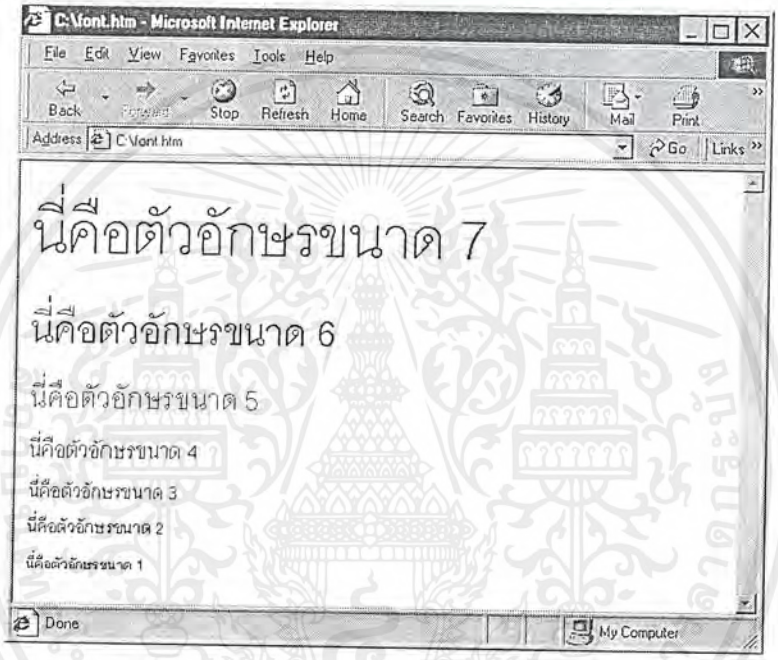


รูปที่2-15 เว็บเบราเซอร์แสดงข้อความให้อยู่ตรงกลาง

แท็กฟอนต์ (FONT) ใช้สำหรับกำหนดขนาดของตัวอักษรที่จะใช้แสดงเอง โดยจะมีแอตทริบิวต์ SIZE เป็นตัวบอกขนาดซึ่งมีตั้งแต่ 1-7 โดยมีรูปแบบเป็น <FONT SIZE="n"> และหากไม่กำหนดแล้วจะมีขนาดเป็น 3 เสมอ

```
<FONT SIZE=7>นี่คือตัวอักษรขนาด 7</FONT>
<FONT SIZE=6>นี่คือตัวอักษรขนาด 6</FONT>
<FONT SIZE=5>นี่คือตัวอักษรขนาด 5</FONT>
<FONT SIZE=4>นี่คือตัวอักษรขนาด 4</FONT>
<FONT SIZE=3>นี่คือตัวอักษรขนาด 3</FONT>
<FONT SIZE=2>นี่คือตัวอักษรขนาด 2</FONT>
<FONT SIZE=1>นี่คือตัวอักษรขนาด 1</FONT>
```

รูปที่ 2-16 แท็กการกำหนดขนาดของตัวอักษร

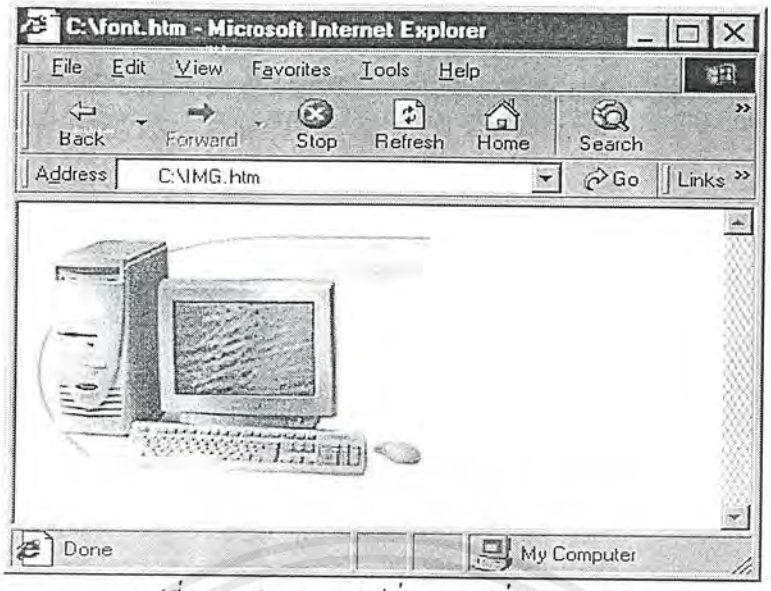


รูปที่2-17 เว็บเบราเซอร์ที่แสดงตัวอักษรขนาดต่าง ๆ

ในการแสดงรูปภาพเราจะใช้แท็ก IMG ในการแสดง โดยมีรูปแบบคำสั่งดังรูปที่ 2-18

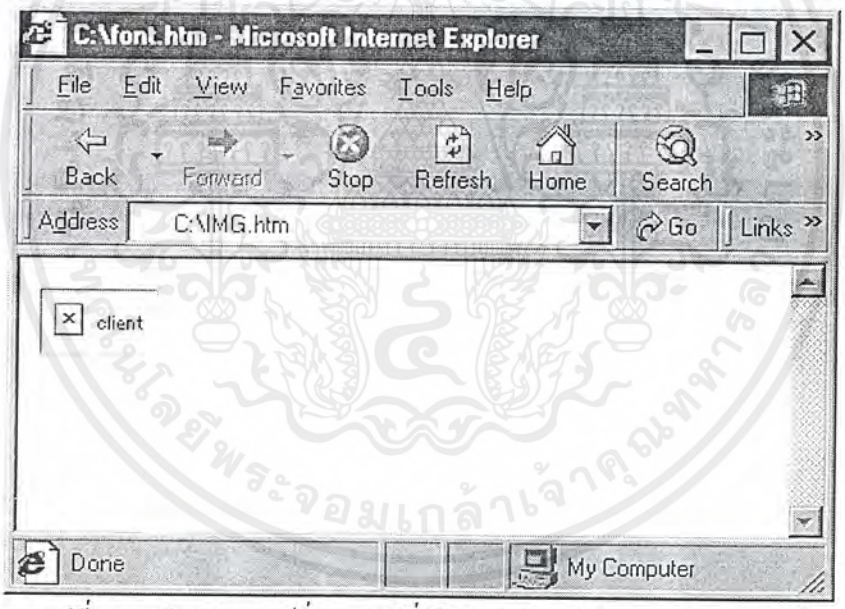
```
<IMG src="c:\client.gif" alt="client">
```

รูปที่ 2-18 แท็กที่ใช้ในการแสดงรูปภาพในเว็บเพจ



รูปที่2-19 เว็บเบราเซอร์ที่แสดงรูปที่ใช้แท็กIMG

และเมื่อไม่สามารถโหลดรูปภาพได้เว็บเบราเซอร์ก็จะแสดงข้อความที่อยู่หลังแอตทริบิวต์ ALT ออกมาดังรูปที่ 2-20

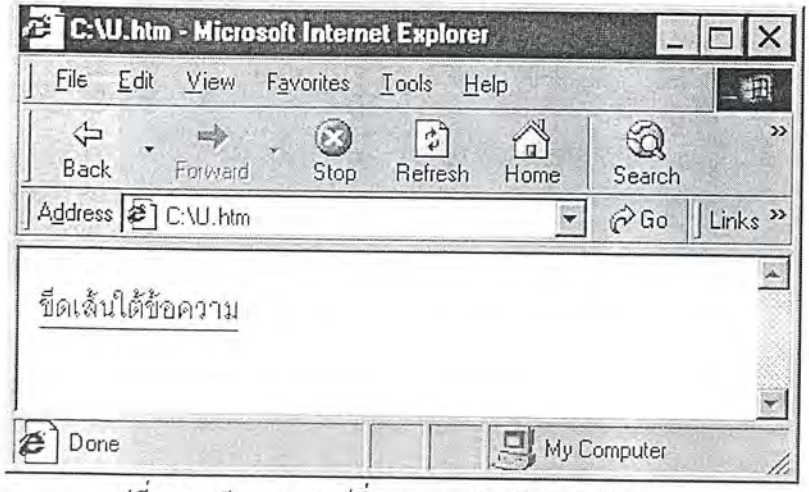


รูปที่2-20 เว็บเบราเซอร์ที่แสดงรูปที่ใช้แท็กIMGแต่ไม่สามารถโหลดรูปได้

การขีดเส้นใต้ให้กับข้อความเพื่อเป็นการเน้นข้อความนั้น ซึ่งจะเป็นการขีดเส้นใต้ข้อความตั้งแต่แท็กเปิด <U> ไปจนถึงแท็กปิด </U> ดังแสดงในรูปที่ 2-21

<U>ขีดเส้นใต้ข้อความ</U>

รูปที่2-21 แท็กที่ใช้ในการขีดเส้นใต้ข้อความ



รูปที่2-22 เว็บเบราเซอร์ที่แสดงการขีดเส้นใต้โดยใช้แท็ก U

แท็ก BR ใช้ในการขึ้นบรรทัดใหม่มีรูปแบบการใช้ดังนี้

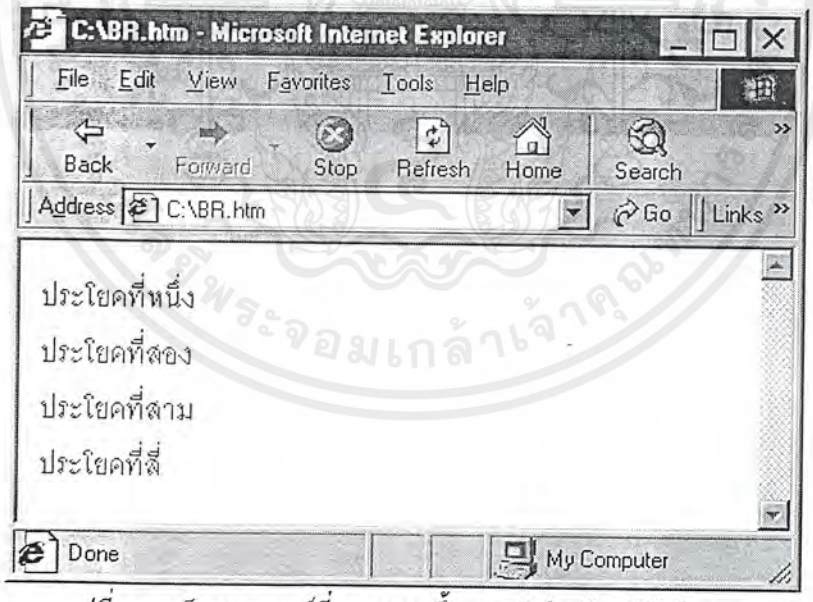
ประโยคที่หนึ่ง

ประโยคที่สอง

ประโยคที่สาม

ประโยคที่สี่

รูปที่ 2-23การใช้แท็ก BR เพื่อขึ้นบรรทัดใหม่

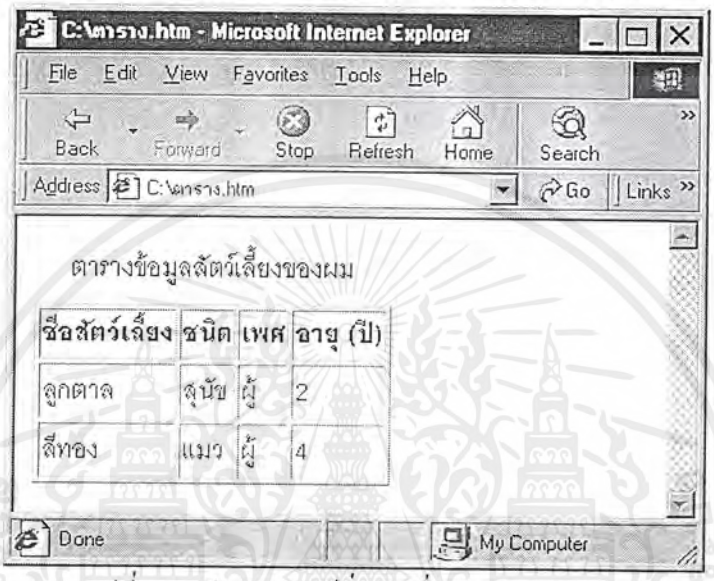


รูปที่2-24 เว็บเบราเซอร์ที่แสดงการขึ้นบรรทัดใหม่โดยใช้แท็ก BR

ในการสร้างตารางจะใช้แท็กดังต่อไปนี้ โดยแท็ก caption จะเป็นการกำหนดชื่อของ ตาราง ตัวอย่างดังรูปที่ 2-25

```
<table border>
<caption>ตารางข้อมูลสัตว์เลี้ยงของผม</caption>
<tr> <th>ชื่อสัตว์เลี้ยง</th>
<th>ชนิด</th> <th>เพศ</th> <th>อายุ (ปี)</th> </tr> <tr> <td>ลูกตาล</td>
<td>สุนัข</td> <td>ผู้</td> <td>2</td> </tr>
<tr> <td>สีทอง</td> <td>แมว</td> <td>ผู้</td> <td>4</td> </tr>
```

รูปที่ 2-25 การใช้แท็ก CAPTION ที่ใช้การสร้างตาราง

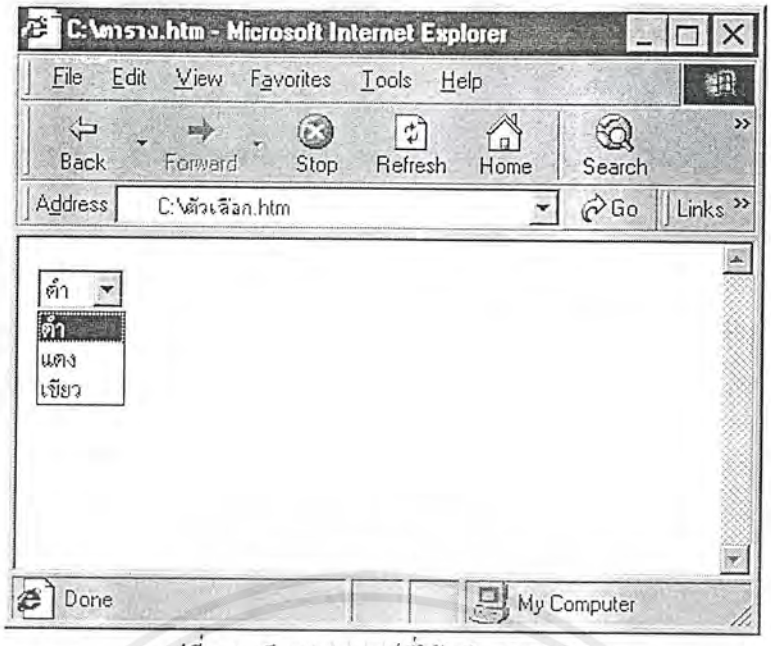


รูปที่2-26 เว็บเบราเซอร์ที่แสดงชื่อตาราง และตาราง

แท็ก option เป็นการใช้ในการแสดงค่าที่ให้เลือกจากหลาย ๆ ตัวเลือกมีรูปแบบการใช้ดังรูปที่ 2-27

```
<select >
<option>ดำ</option>
<option>แดง </option>
<option>เขียว </option>
</select>
</form>
```

รูปที่ 2-27 การใช้แท็ก OPTION ที่ใช้การสร้างตัวเลือก



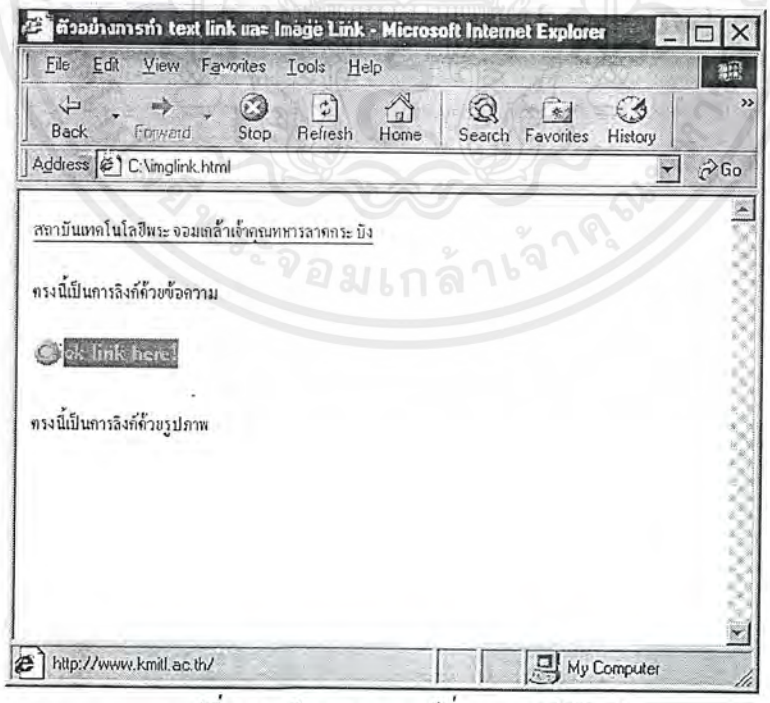
รูปที่2-28 เว็บเบราเซอร์ที่ใช้แท็ก OPTION

แท็ก A เป็นแท็กที่ใช้ทำหน้าที่ในการเชื่อมโยงไปยังเพจอื่น ๆ หรือเว็บไซต์อื่น ซึ่งอาจเป็นการลิงก์ด้วยข้อความหรือรูปภาพก็ได้ แอททริบิวต์ที่ใช้ก็คือ href แล้วตามด้วยยูอาร์แอลของไฟล์หรือเว็บไซต์ที่ต้องการลิงก์ไป โดยรูปแบบการใช้ลิงก์ดังรูปที่ 2-29

ข้อความ หรือรูปภาพ

<img src="ชื่อภาพ" Attribute1 Attribute2>

รูปที่ 2-29 ตัวอย่าง แท็กที่ใช้ลิงก์ด้วยข้อความและรูปภาพ



รูปที่ 2-30 เว็บเบราเซอร์ที่แสดงการใช้ลิงก์

2.6 หลักการจัดอันดับเว็บเพจของเสิร์ชเอ็นจิ้น

โดยปกติแล้วการป้อนคีย์เวิร์ดเพื่อค้นหาในเสิร์ชเอ็นจิ้นที่สำคัญส่วนใหญ่นั้นจะได้ผลลัพธ์เป็นยูอาร์แอลของเว็บเพจนับล้านเพจออกมา ซึ่งตัวเสิร์ชเอ็นจิ้นมีจุดมุ่งหมายที่จะแสดงเว็บเพจผลลัพธ์ที่สอดคล้องกับคีย์เวิร์ดที่ใช้ค้นหาให้มากที่สุดเป็นจำนวน 10 หรือ 20 เว็บเพจในลักษณะเรียงลำดับจากเว็บเพจที่มีความสอดคล้องมากที่สุดและลดลงมาเรื่อย ๆ

ดังนั้นอย่างน้อยที่สุดเราจะพบไซต์ที่ตรงกับความต้องการในผลลัพธ์ 20 อันดับแรกในเสิร์ชเอ็นจิ้นโดยส่วนใหญ่ ดังนั้นถ้าเว็บไซต์ของคุณอยู่ในลำดับที่ต่ำกว่าลำดับ 20 ลงไปแล้ว ก็จะทำให้ไม่ค่อยมีผู้เข้ามาเยี่ยมชม ยิ่งไปกว่านั้นเว็บไซต์ที่ปรากฏในลำดับที่ต่ำมาก ๆ ก็จะยิ่งทำให้แทบจะไม่มีผู้เข้าเยี่ยมชมเลยก็ได้

ไซต์ต่าง ๆ มากมายถูกจัดลำดับในลำดับล่าง ๆ จากผลการค้นหาของเสิร์ชเอ็นจิ้นก็เพราะว่าไซต์ต่าง ๆ เหล่านั้นไม่ได้พิจารณาถึงการทำงานของเสิร์ชเอ็นจิ้นว่าทำงานอย่างไรหรือ การออกแบบเว็บเพจอย่างไรให้เหมาะสมกับเสิร์ชเอ็นจิ้น ซึ่งการออกแบบที่ดีย่อมทำให้เว็บไซต์เข้าถึงได้ง่ายและถูกจัดลำดับดีขึ้นเพื่อเพิ่มโอกาสที่เสิร์ชเอ็นจิ้นจะพบและทำการจัดอยู่ในลำดับที่ดีขึ้น

2.6.1 หลักการของเสิร์ชเอ็นจิ้นที่แตกต่างกันคืออะไร

อันดับแรก เสิร์ชเอ็นจิ้นบางตัวมีขนาดใหญ่มากเพราะว่ามันทำการกำหนดัชนีให้กับเว็บเพจได้มากกว่าเสิร์ชเอ็นจิ้นตัวอื่น และบางตัวก็ทำการกำหนดัชนีให้กับเว็บเพจได้บ่อยครั้งกว่า ดังนั้นไม่มีเสิร์ชเอ็นจิ้นตัวใดที่จะมีชุดของเว็บเพจเหมือนกันจากการค้นหา

เสิร์ชเอ็นจิ้นต่าง ๆ มีการให้ความสำคัญเพิ่มขึ้นกับบางเว็บเพจในการจัดลำดับด้วยเหตุผลต่าง ๆ กันไป ยกตัวอย่างเช่น Infoseek, Lycos, Excite และ Web Crawler ใช้ลิงก์เป็นส่วนหนึ่งของวิธีการจัดลำดับของพวกมัน ซึ่งทำโดยการหาว่ามีกี่เว็บเพจที่ทำการลิงก์ไปถึง เสิร์ชเอ็นจิ้นบางตัวให้ความเกี่ยวข้องเพิ่มขึ้นกับเว็บเพจโดยใช้เจ้าหน้าที่ของพวกเขาทำการพิจารณาเว็บเพจเหล่านั้น

เสิร์ชเอ็นจิ้นมีการเปลี่ยนแปลงและพัฒนาแนวทางในการให้คะแนนแก่เว็บเพจอยู่เสมอเพื่อให้แน่ใจได้ว่าผู้ใช้บริการสามารถที่จะได้รับผลการค้นหาที่รวดเร็ว และผลลัพธ์ที่ได้มีความเกี่ยวข้องสูง รายละเอียดของการให้คะแนนเพื่อให้ลำดับของเพจอยู่ในลำดับต้น ๆ ของเสิร์ชเอ็นจิ้น ในแต่ละตัวนั้นจะแตกต่างกัน ซึ่งในหัวข้อนี้จะกล่าวถึงปัจจัยที่เสิร์ชเอ็นจิ้นใช้ในการจัดลำดับเว็บเพจ

2.6.2 คีย์เวิร์ด

ขั้นแรกของวิธีการ คือ ต้องจัดการกับคีย์เวิร์ดที่มีความเกี่ยวข้องกับไซต์ของคุณ เมื่อผู้ค้นหากำลังมองหาผลิตภัณฑ์หรือบริการของคุณ ให้ลองคิดดูสัก 2-3 นาที ซึ่งมันอาจจะไม่เป็นคำที่เห็นได้ชัดเสมอไป ซึ่งจะเป็นคำที่ผู้ค้นหามักจะใช้ในการหา โดยปกติแล้วต้องเป็นคำที่ผู้ซึ่งไม่คุ้นเคยกับผลิตภัณฑ์ หรือบริการของคุณมักจะทำการป้อนในการค้นหา ซึ่งคีย์เวิร์ดจะคือคำใด ๆ ที่พวกเขาใช้ในการค้น

หาเว็บไซต์ที่มีเนื้อหา ผลิตภัณฑ์ หรือบริการคุณ วิธีการที่ดีที่จะใช้ในการตัดสินใจกำหนดคีย์เวิร์ด คือ เลือกมาอย่างน้อย 40 คำ แล้วทำการจำกัดลงมาให้ตรงกับเนื้อหาของเว็บไซต์ของคุณให้มากที่สุด

ต่อไปให้คิดว่าคีย์เวิร์ดใดมีส่วนเกี่ยวข้องกับเพจใด และกลุ่มของคีย์เวิร์ดกลุ่มใดที่ใช้ร่วมกันได้ภายในไซต์ของคุณ ผู้ท่องเว็บส่วนใหญ่จะเริ่มต้นใช้คำที่ค้นหาอย่างน้อย 2 คำ เพราะการใช้เพียงหนึ่งคำจะมีความหมายกว้างมากไปที่เสิร์ชเอ็นจิ้นจะจำกัดขอบเขตของการค้นหาที่พวกเขาต้องการได้ ตัวอย่างเช่น ถ้าต้องการหาภัตคารตามท้องถิ่นแห่งหนึ่ง คีย์เวิร์ดตัวหนึ่งที่น่าจะใส่ก็ควรจะเป็น city หรือ suburb เป็นต้น

2.6.3 การออกแบบเว็บเพจและการกำหนดตำแหน่งของคีย์เวิร์ด

ต้องแน่ใจว่าคีย์เวิร์ดของคุณอยู่ในตำแหน่งที่สำคัญมากที่สุดบนเว็บเพจของคุณ ซึ่งชื่อเรื่องของเพจ (Title) ควรจะมีคีย์เวิร์ดอยู่มากที่สุด และในยูอาร์แอลควรจะมีคีย์เวิร์ดอยู่ด้วยเช่นกันเพราะเสิร์ชเอ็นจิ้นบางตัวให้ความสำคัญกับมัน

ในเสิร์ชเอ็นจิ้นบางตัวจะดูว่ามีคีย์เวิร์ดปรากฏอยู่ในเพจนั้นมากน้อยแค่ไหน ซึ่งควรจะใช้คำที่เป็นคีย์เวิร์ดในหัวเรื่องของเว็บเพจ และถ้าเป็นไปได้ก็ควรใส่ใน 2-3 ย่อหน้าแรกของเพจด้วย และไม่ควรใส่คีย์เวิร์ดลงในตาราง เพราะเสิร์ชเอ็นจิ้นจะอ่านตารางเป็นคอลัมน์ ดังนั้นซึ่งจะทำให้เสิร์ชเอ็นจิ้นอ่านข้อความที่อยู่ในตารางต่างกับลำดับที่คุณได้จัดเรียงไว้ ทำให้คีย์เวิร์ดมีความสำคัญน้อยลง ถ้าคุณออกแบบให้ตารางอยู่ทางซ้ายมือของเพจ แล้วเสิร์ชเอ็นจิ้นบางตัวจะมองว่าคำที่อยู่ในส่วนนี้เป็นคีย์เวิร์ดของเพจได้

ชื่อเรื่องของเพจที่ถูกใช้ในส่วนหัวของเพจควรจะถูกระบุให้เป็นหัวข้อแทนที่จะใช้เป็นข้อความธรรมดา ซึ่งตัวสไปเดอร์ของเสิร์ชเอ็นจิ้นจะถือว่าเป็นคีย์เวิร์ดด้วย เช่น <h1>title</h1> เป็นต้น

2.6.4 แท็ก META

แท็ก META เป็นส่วนประกอบที่จำเป็นเพราะว่าเสิร์ชเอ็นจิ้นบางตัวใช้เพียงแค่แท็ก META ในการทำดัชนีให้กับเพจคุณเท่านั้น ดังนั้นคีย์เวิร์ดและคำอธิบายของแท็ก META จะช่วยให้คุณควบคุมคำอธิบายไซต์ของคุณ และแท็ก META ยังสามารถใช้ในการบ่งบอกถึงผู้สร้างเพจ ภาษาแฮชทีเอ็มแอลที่ใช้ และพารามิเตอร์สำหรับรีเฟรส (ซึ่งใช้ทำให้เว็บเพจทำการรีเฟรสตัวเอง หรือโหลดเพจอื่นขึ้นมา) ซึ่งแท็ก META ทุกตัวควรจะอยู่ระหว่าง <head> และ </head> แท็ก META ที่พิจารณามากสุดคือ แท็ก META Description และ META Keywords

META DESCRIPTION : ให้ใส่คำบรรยายเกี่ยวกับเว็บไซต์ของคุณ โดยใส่คีย์เวิร์ดและวลีลงในส่วน description นี้ และไม่ควรใส่เกิน 200 ตัวอักษรซึ่งจำกัดโดยตัวเสิร์ชเอ็นจิ้นส่วนใหญ่ที่มีอยู่

<meta name = "description" content = "place your site description here">

META KEYWORDS : คีย์เวิร์ดที่ใช้ควรเป็นรูปพหูพจน์ เพราะเสิร์ชเอ็นจิ้นโดยส่วนใหญ่จะทำงานกับทั้งรูปเอกพจน์ และพหูพจน์ และควรหลีกเลี่ยงการใช้คีย์เวิร์ดที่ซ้ำกัน เพราะจะทำให้เสิร์ชเอ็นจิ้นมองว่าใช้

คำฟุ่มเฟือย การใส่คีย์เวิร์ดในส่วน META Keywords นี้ให้ทำการสร้างรายการคีย์เวิร์ดที่แยกคำออกจากกันด้วยเครื่องหมาย "," และไม่ควรเกิน 10,000 ตัวอักษร <meta name = "keywords" content = "place, your, site, keywords, here">

คุณอาจจะใส่คีย์เวิร์ดบางตัวลงในแท็กคอมเมนต์ด้วย สำหรับเสิร์ชเอ็นจิ้นที่ไม่ได้ตรวจดูแท็ก META โดยให้ใช้แท็กคอมเมนต์ในลักษณะตามตัวอย่างนี้ ในส่วนของ Body ของเพจ <!--// This page is about the Australian Banner Exchange's resources for Site development. //--!>

แท็ก META บางตัวสามารถช่วยคุณแก้ปัญหาเรื่องตาราง เฟรมและปัญหาในพื้นที่ส่วนอื่นได้ ซึ่งคุณควรจะใช้แท็ก META โดยเฉพาะอย่างยิ่งกับคีย์เวิร์ดและคำบรรยายที่จำเป็น แต่นั่นไม่ได้หมายความว่าจะทำให้ไซต์ของคุณอยู่ในลำดับที่สูงได้

คีย์เวิร์ดต้องเป็นอะไรที่สะท้อนให้เห็นถึงเนื้อหาของเพจอย่างถูกต้อง เสิร์ชเอ็นจิ้นบางตัวจะทำการกำหนดดัชนีกับข้อความใน ALT (ซึ่งจะซ่อนอยู่ใต้รูปภาพ) และแท็ก META และข้อความคอมเมนต์ แต่เพื่อให้แน่ใจว่าตัวเสิร์ชเอ็นจิ้นได้อ่านคีย์เวิร์ดที่เกี่ยวข้อง แล้วการใช้ข้อความในโค้ดแฮชทีเอ็มแอลที่มองไม่เห็นต้องมีความรอบคอบเป็นอย่างมาก

ตัวสไปเดอร์ของเสิร์ชเอ็นจิ้นส่วนใหญ่นั้นแตกต่างกันและทุกตัวก็ทำการค้นหาบางอย่างบนเว็บเพจของคุณที่แตกต่างกันออกไป ดังนั้นการออกแบบเว็บเพจของคุณต้องทำให้ง่ายต่อการที่สไปเดอร์ของเสิร์ชเอ็นจิ้นจะเข้ามาค้นหา ยกตัวอย่างเช่น เสิร์ชเอ็นจิ้นที่สำคัญบางตัวไม่สามารถไล่ไปตามการเชื่อมโยงแบบเฟรม (frame links)ได้ หรือว่ามันอาจจะไม่อ่านการทำการแมปรูปภาพ ดังนั้นคุณจำเป็นที่ต้องแน่ใจได้ว่ามีวิธีการอื่น ๆ ให้กับตัวสไปเดอร์เหล่านั้นเข้ามาและทำการสร้างดัชนีไซต์ของคุณได้ โดยอาจใช้แท็ก META หรือการออกแบบที่ดี เพื่อช่วยในการจัดลำดับอีกด้วย

บทที่ 3

การคำนวณ การสร้าง และการออกแบบ

ในส่วนของการออกแบบนั้น ตัวโปรแกรมจะประกอบด้วย 2 ส่วนหลัก ๆ คือ ส่วนแรกเป็นส่วนของโปรแกรมซีจีไอที่เขียนด้วยภาษาวิชวลเบสิก เวอร์ชัน 6.0 ซึ่งโปรแกรมซีจีไอจะทำหน้าที่ในการรับข้อมูลต่าง ๆ จากผู้ใช้โดยผ่านทางฟอร์ม และส่วนที่สอง คือ ส่วนโปรแกรมที่ใช้ในการกำหนดคีย์เวิร์ดซึ่งจะนำข้อมูลที่รับมาโดยส่วนโปรแกรมซีจีไอ นำมาทำการประมวลผลเพื่อหาคีย์เวิร์ดออกมา ซึ่งขั้นตอนการทำงานของโปรแกรมจะมีลักษณะการทำงานดังนี้

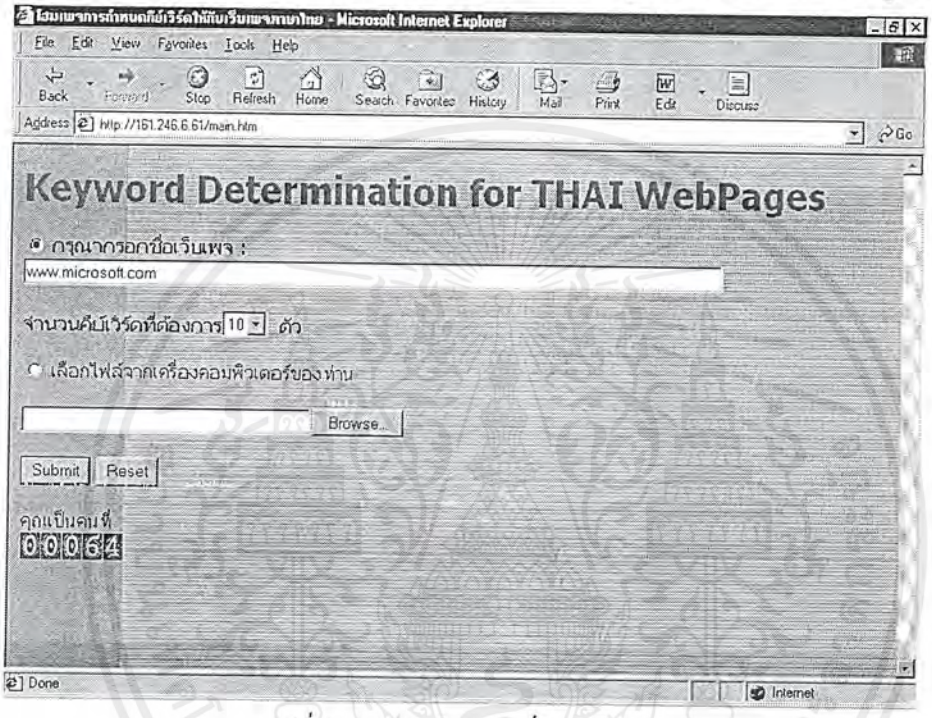
1. รับไฟล์แฮชทีเอ็มแอล โดยผู้ใช้จะป้อนยูอาร์แอลของไฟล์ แฮชทีเอ็มแอล ที่ต้องการกำหนดคีย์เวิร์ด หรือทำการอัพโหลดไฟล์แฮชทีเอ็มแอลจากเครื่องของผู้ใช้เอง
2. นำไฟล์แฮชทีเอ็มแอลที่ได้มาทำการตัดคำ โดยจะเรียกใช้โปรแกรม CTTEX ซึ่งเป็นโปรแกรมแบบ executable file (ไฟล์นามสกุล .EXE) ซึ่งเป็นโปรแกรมการตัดคำของเนคเทคมาใช้งาน
3. ทำการเก็บเฉพาะแท็กที่พิจารณาพร้อมกับเนื้อความที่อยู่ภายในแท็กนั้นเท่านั้น โดยแท็กที่พิจารณา ได้แก่ <TITLE>, <FONT>, <B>, <STRONG>,<CENTER> ฯลฯ เป็นต้น
4. ทำการกำจัดคำจำพวกสต็อปเวิร์ดทิ้ง โดยเปรียบเทียบจากสต็อปเวิร์ดลิสต์ที่เก็บไว้
5. ทำการนับความถี่ของคำแต่ละคำที่ปรากฏอยู่ในไฟล์แฮชทีเอ็มแอล แล้วนำไปเก็บไว้ในฐานข้อมูลที่ได้เตรียมไว้
6. ทำการกำหนดค่าน้ำหนักให้กับคำแต่ละคำที่ปรากฏในไฟล์แฮชทีเอ็มแอล โดยค่าน้ำหนักของคำแต่ละคำจะเป็นไปตามแท็กที่คำ ๆ นั้นปรากฏอยู่ ซึ่งค่าน้ำหนักของแต่ละแท็กนี้จะได้มาจากการเลือกสุ่มไฟล์แฮชทีเอ็มแอลมาจำนวนหนึ่ง แล้วทำการทดสอบกับค่าน้ำหนักชุดหนึ่งและทำการปรับเปลี่ยนค่าน้ำหนักจนกว่าจะได้ค่าน้ำหนักชุดที่เหมาะสมที่สุดที่ทำให้ได้คีย์เวิร์ดที่ดีที่สุดออกมา

ซึ่งจากขั้นตอนเหล่านี้สามารถนำมาเขียนเป็นแนวความคิดของการทำงานของโปรแกรมได้ดังรูปที่ 3-1

รูปที่ 3-1 แนวความคิดในการทำการกำหนดคีย์เวิร์ด

3.1 ส่วนการทำงานของโปรแกรม ซีจีไอ

ในการออกแบบโปรแกรมซีจีไอนั้น จะเริ่มทำการออกแบบโฮมเพจซึ่งเป็นอินเตอร์เฟสที่จะใช้รับข้อมูลจากผู้ใช้ก่อน เพื่อกำหนดว่าจะมีตัวแปรอะไรบ้างที่โปรแกรมซีจีไอจะรับจากฟอร์มไปใช้งาน โดยหน้าจอส่วนติดต่อนี้จะแบ่งออกเป็น 2 ส่วน คือ ส่วนที่รับยูอาร์แอลของเว็บเพจที่ผู้ใช้ต้องการให้กำหนดคีย์เวิร์ด กับส่วนที่รับไฟล์จากเครื่องของผู้ใช้เพื่อนำมาใช้กำหนดคีย์เวิร์ด โดยไฟล์ที่ทำการอัพโหลดมานั้นควรจะอยู่ในรูปของ TXT file กับ HTML file และผู้ใช้ยังสามารถที่จะกำหนดจำนวนคีย์เวิร์ดที่ต้องการให้แสดงออกหน้าจอได้อีกด้วย โดยที่หน้าจอสำหรับผู้ใช้งานก็จะมีลักษณะดังรูปที่ 3-2



รูปที่ 3-2 หน้าจอโฮมเพจที่ใช้รับข้อมูล

และในส่วนที่ 2 ซึ่งเป็นส่วนของโปรแกรมซีจีไอจะใช้ภาษาวิชวลเบสิก เวอร์ชัน 6.0 ในการเขียนโปรแกรม พร้อมกับมีโมดูลดีบั๊กเพื่อใช้ทำการดีบั๊กการทำงานของโปรแกรมได้ โดยตัวโปรแกรมซีจีไอที่ออกแบบมานั้นจะมีการทำงาน 3 ส่วนหลัก ๆ ดังต่อไปนี้

ส่วนแรกจะเป็นส่วนที่ทำการโหลดตัวแปรสภาพแวดล้อมที่อยู่ในเซิร์ฟเวอร์มาใช้และทำการกำหนดค่าเริ่มต้นของตัวแปรต่าง ๆ ที่ใช้ในโปรแกรม

ส่วนที่สองจะเป็นส่วนที่ทำการอ่านข้อมูลที่ส่งมาจากเซิร์ฟเวอร์ โดยจะมีการตรวจสอบตัวแปรสภาพแวดล้อม REQUEST_METHOD ว่ามีการร้องขอเป็น POST GET HEAD หรือ PUT ถ้าการร้องขอที่เข้ามาเป็น POST จะทำการอ่านค่าข้อมูลจากแฮนเดิลอินพุต แต่ถ้าเป็นการร้องขอแบบ GET HEAD และ PUT จะทำการอ่านค่าข้อมูลจากตัวแปรสภาพแวดล้อมที่ชื่อ QUERY_STRING แทน

และในส่วนที่สามซึ่งเป็นส่วนหลักของโปรแกรมจะทำการประมวลผลข้อมูล และทำการส่งข้อมูลกลับไปยังเซิร์ฟเวอร์ โดยที่จะทำการตรวจสอบก่อนว่าผู้ใช้ได้ทำการกรอกยูอาร์แอลของเว็บไซต์มาหรือทำการอัพโหลดไฟล์จากเครื่องของผู้ใช้เอง หลังจากนั้นจึงทำการเซฟเป็นไฟล์เพื่อทำการตัดคำโดยใช้

โปรแกรมการตัดคำที่ได้มาจากเนคเทค เมื่อทำการตัดคำเรียบร้อยแล้วก็จะเข้าสู่ขั้นตอนของการหาคีย์เวิร์ด ด้วยส่วนของโปรแกรมการกำหนดคีย์เวิร์ด เมื่อได้คีย์เวิร์ดแล้วก็จะทำการส่งผลลัพธ์กลับไปยังเซิร์ฟเวอร์ โดยผ่านข้อมูลไปทางแฮนเดิลเอาท์พุต

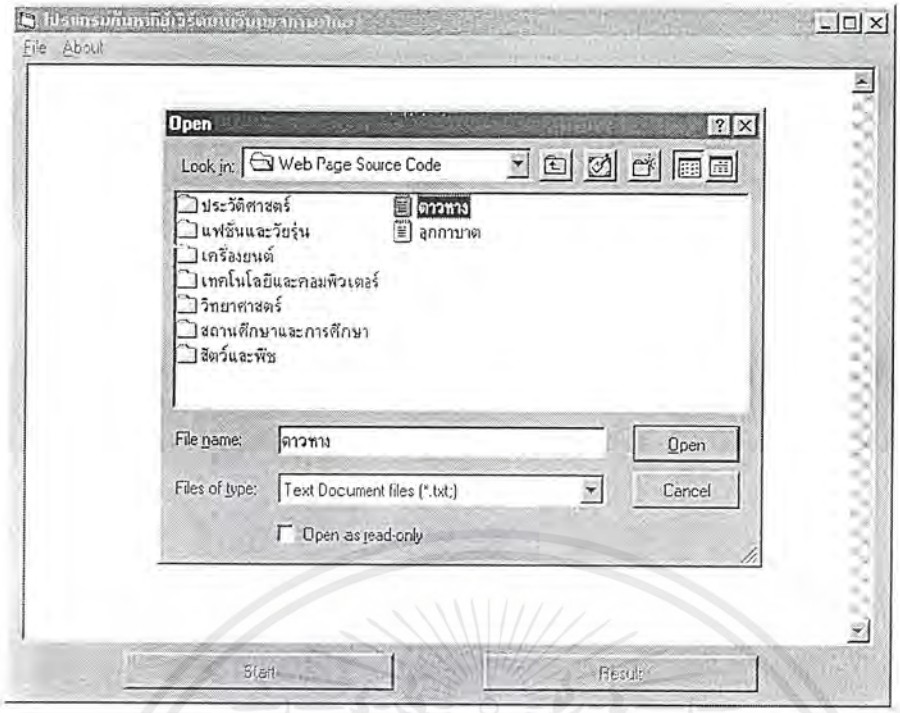
3.2 ส่วนโปรแกรมที่ใช้ในการกำหนดคีย์เวิร์ด

ในส่วนนี้จะทำการเขียนโปรแกรมต้นแบบก่อน โดยสร้างเป็นแอปพลิเคชั่นบนวินโดวส์เพื่อใช้ในการทดสอบการทำงานของการกำหนดคีย์เวิร์ด หลังจากนั้นเมื่อโปรแกรมมีการทำงานที่ถูกต้อง และมีประสิทธิภาพแล้ว ก็จะนำไปรวมกับส่วนของโปรแกรมซีจีไอ เพื่อให้สามารถทำงานผ่านอินเตอร์เน็ตได้

ตัวโปรแกรมต้นแบบที่เป็นแอปพลิเคชั่นบนวินโดวส์จะมีหน้าจอที่ใช้ติดต่อกับผู้ใช้ โดยมีปุ่ม Start ใช้ในการรันโปรแกรม และปุ่ม Result ใช้ในการดูผลลัพธ์ที่ได้จากการหาคีย์เวิร์ด ดังรูปที่ 3-3

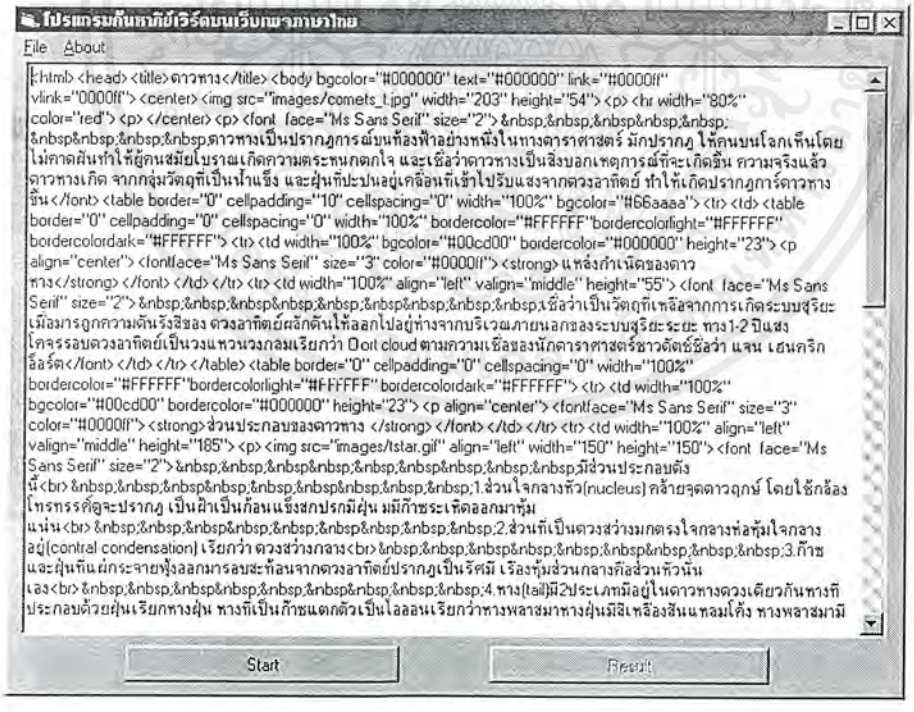
รูปที่ 3-3 หน้าจอของโปรแกรมต้นแบบ

ตัวโปรแกรมจะรับเท็กซ์ไฟล์เป็นอินพุตของตัวโปรแกรม ซึ่งไฟล์อินพุตจะต้องผ่านการตัดคำก่อน โดยใช้โปรแกรม CTTEX.EXE ในการตัดคำ เราสามารถเลือกไฟล์ที่ต้องการได้จากคำสั่ง File --> Open ดังรูปที่ 3-4



รูปที่ 3-4 หน้าจอในการเปิดไฟล์โค้ดแฮชทีเอ็มแอลของโปรแกรม

เมื่อเลือกไฟล์ที่ต้องการแล้ว ซอร์สโค้ดของไฟล์ก็จะแสดงบนหน้าจอ โดยไฟล์ที่สามารถเลือกได้นั้นจะเป็น Text Document files เท่านั้น ซึ่งก็คือเท็กซ์ไฟล์ มีนามสกุลเป็น .txt ดังรูปที่ 3-5



รูปที่ 3-5 หน้าจอภายหลังจากการเปิดไฟล์โค้ดแฮชทีเอ็มแอล

หลังจากนั้นให้คลิ๊กปุ่ม Start เพื่อทำการกำหนดคีย์เวิร์ด เมื่อทำการคลิ๊กปุ่ม Start แล้วโปรแกรมจะขึ้นซอร์สโค้ดบนหน้าจอพร้อมกับให้ใส่ชื่อฐานข้อมูลที่ต้องการใช้เก็บผลลัพธ์ ซึ่งสามารถดูผลลัพธ์ได้จากการคลิ๊กปุ่ม Result ตัวโปรแกรมจะทำการเชื่อมต่อกับฐานข้อมูลไมโครซอฟต์แอคแซสที่ได้สร้างไว้เพื่อแสดงผลลัพธ์ที่เก็บไว้ในฐานข้อมูล โดยผลลัพธ์จากการกำหนดคีย์เวิร์ดนั้นจะบอกลำดับที่ คีย์เวิร์ดที่ได้ ค่าความถี่ของคีย์เวิร์ดที่ปรากฏ และค่าน้ำหนักของคีย์เวิร์ดที่ได้จากการประมวลผลโดยโปรแกรม ดังรูปที่ 3-6


ผลลัพธ์จากการค้นหา

ผลลัพธ์จากการค้นหา

	ID	คีย์เวิร์ด	ความถี่	ค่าน้ำหนัก
▶	1	พาง	28	106.7
	2	ดาว	24	95.50002
	3	วง	16	50.6
	4	โคจร	11	36.60001
	5	ดวง	11	30.80001
	6	อาทิตย์	11	30.80001
	7	ส่วน	8	28.20001
	8	โลก	6	22.6
	9	ก๊าซ	8	22.40001
	10	ตาม	7	19.6
	11	ปรากฏ	7	19.6
	12	ฝุ่น	7	19.6
	13	ออก	6	16.8
	14	กลาง	6	16.8
	15	กัน	6	16.8
	16	เศราะห์	6	16.8
	17	เกิด	6	16.8
	18	ประกอบ	3	14.2
	19	ต่อ	3	14.2
	20	ห้อม	5	14

รูปที่ 3-6 หน้าจอผลลัพธ์จากการค้นหาคีย์เวิร์ด

3.2.1 ส่วนในการพิจารณาแท็กที่นำมาใช้

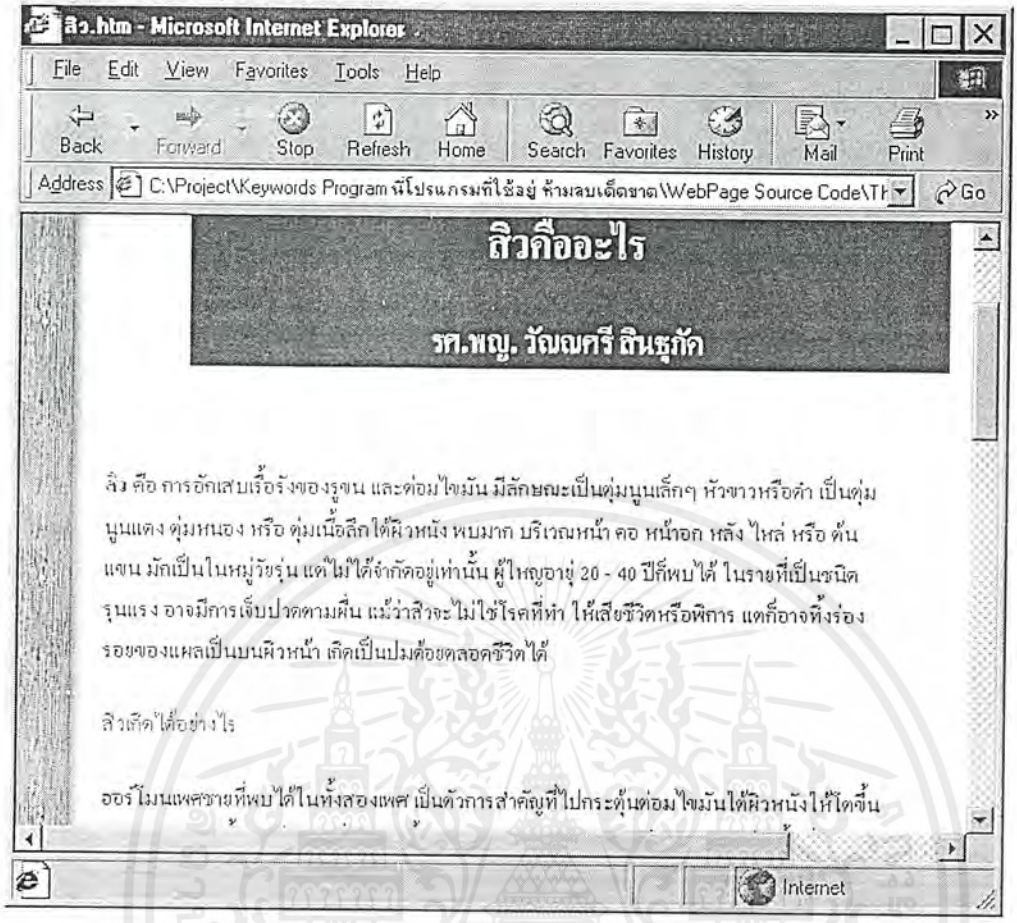
หลักในการทำงานของโปรแกรมจะมีการพิจารณาปัจจัยต่าง ๆ ที่มีความเกี่ยวข้องกับการกำหนดคำที่จะเป็นคีย์เวิร์ดของเว็บเพจ

โดยปัจจัยที่นำมาพิจารณาในการกำหนดคีย์เวิร์ดมีดังต่อไปนี้

1) การจัดวางตำแหน่งของคีย์เวิร์ด (Keyword Placement) พิจารณาถึงตำแหน่งของคำที่ ปรากฏในเว็บเพจ

2) ความถี่ของคำ (Keyword Frequency) พิจารณาว่าคำนั้นปรากฏมากน้อยเพียงใดในเว็บเพจหนึ่งๆ

ในการพิจารณาว่าตัวโปรแกรมจะมีการใช้แท็กใดบ้าง เราจะเริ่มทำการพิจารณาจากการดูเว็บเพจที่มีการแสดงผลลัพธ์บนเว็บเบราเซอร์ด้วยตา เพื่อพิจารณาว่าคำที่เป็นคีย์เวิร์ดของแต่ละเว็บเพจมักจะปรากฏในตำแหน่งใดบ้างของเว็บเพจ



รูปที่ 3-7 เว็บเพจที่มีอยู่โดยทั่วไป

จากรูปที่ 3-7 ใจความของเว็บเพจกล่าวถึงเรื่อง "สิว" โดยมีหัวข้อขึ้นต้นด้วยข้อความว่า "สิวคืออะไร" ถัดมาในย่อหน้าแรกขึ้นต้นด้วยคำว่า "สิว" โดยเขียนด้วยตัวหนาก่อนขึ้นประโยค แล้วตามด้วยคำอธิบาย ต่อมาในหัวข้อแรกเขียนว่า "สิวเกิดได้อย่างไร" ซึ่งเป็นการบ่งบอกว่าเว็บเพจนี้มีใจความเกี่ยวกับเรื่องของสิว ซึ่งจะสังเกตเห็นได้ว่าคำที่เป็นคีย์เวิร์ดของเว็บเพจ มักจะปรากฏในส่วนที่เป็นหัวข้อส่วนต้นของย่อหน้า และเป็นคำที่มักจะมีการเน้นด้วยสีสันหรือทำให้แตกต่างจากคำอื่น ๆ ที่มีอยู่ภายในเว็บเพจ

หลังจากพิจารณาจากการดูด้วยตาแล้ว ก็ทำการพิจารณาซอร์สโค้ดเพื่อดูว่าพบคีย์เวิร์ดอยู่ภายในแท็กใดบ้าง แล้วนำมาใช้พิจารณาการกำหนดคีย์เวิร์ดให้กับเว็บเพจ

โดยแท็กที่นำมาใช้ในการพิจารณามีดังต่อไปนี้

1. **TITLE** เนื่องจากแท็ก TITLE นี้เป็นแท็กที่ใช้กำหนดชื่อเรื่อง (title) ที่แสดงอยู่ที่แถบแสดงชื่อเรื่อง (title bar) หรือแถบบนสุดของเว็บเบราเซอร์ ดังนั้นคำที่ปรากฏภายในแท็ก TITLE จึงเป็นคำที่ใช้ระบุถึงหัวข้อหลักที่สื่อถึงข้อมูลที่อยู่ในเพจนั้น

2. **Hn** เป็นแท็กแสดงถึงระดับความสำคัญของหัวข้อ และขนาดของตัวอักษรของหัวเรื่องนั้น

โดย n จะมีค่าเท่ากับ 1 ถึง 6 ซึ่งเริ่มจากแท็ก <H1> ซึ่งเป็นหัวข้อหลัก และจะมีตัวอักษรขนาดใหญ่ที่สุด ไปจนถึงแท็ก <H6> เป็นหัวข้อย่อยที่สุด และมีตัวอักษรขนาดเล็กที่สุด ซึ่งคำที่ปรากฏในหัวข้อต่าง ๆ เหล่านี้มักจะมีส่วนเกี่ยวข้องกับเนื้อความของเว็บเพจ

3. **CENTER** แท็ก center นี้ บ่อยครั้งที่คำที่มีความสำคัญมักจะเป็นส่วนที่อยู่กลางหน้ากระดาษ ซึ่งแท็ก CENTER นี้จะเป็นแท็กที่ใช้กำหนดข้อความให้อยู่ตรงกลางหน้าเว็บเพจ

4. **B, BIG, STRONG** แท็กต่าง ๆ เหล่านี้ จะเป็นแท็กที่ใช้กำหนดข้อความตัวหนาซึ่งข้อความที่เป็นตัวหนาก็มักจะมีความสำคัญที่จะเป็นคีย์เวิร์ดเช่นกัน

5. U คำที่ขีดเส้นใต้ก็มีโอกาสที่จะมีความสำคัญในการเพิ่มค่าน้ำหนักได้

6. **CAPTION** เป็นแท็กที่ใช้แสดงชื่อตาราง ซึ่งชื่อตารางก็มีความสำคัญเช่นกันในการกำหนดคีย์เวิร์ด เพราะตารางที่ใช้ก็ควรจะเกี่ยวข้องกับเนื้อหาของเว็บเพจด้วย

7. **OPTION** แท็กนี้จะใช้ในการกำหนดตัวเลือกซึ่งคำที่เป็นตัวเลือกเองในบางครั้งก็อาจจะเป็นคำที่เป็นคีย์เวิร์ดได้เช่นกัน

8. **FONT** ในบางครั้งข้อความที่เขียนตัวใหญ่กว่าตัวอื่น ๆ ก็เป็นการเน้นความสำคัญของผู้เขียน และแท็กนี้เองก็เป็นการกำหนด ขนาดของข้อความต่าง ๆ ซึ่งต้องนำมาพิจารณา

9. **IMG** รูปภาพที่ใช้แสดงในเว็บเพจนั้นจะเป็นรูปภาพที่เกี่ยวกับเนื้อหาของเว็บเพจเอง โดยใช้เนื้อหาบรรยายรูปภาพ ซึ่งภายในแท็ก IMG นี้จะมีแอททริบิวต์ชื่อ ALT ซึ่งใช้ในการแสดงข้อความที่ใช้สำหรับอธิบายรูปภาพ เมื่อเว็บเบราเซอร์ไม่สามารถแสดงรูปภาพได้ ก็จะแสดงข้อความที่อยู่ภายในแอททริบิวต์ ALT นี้แทน ดังนั้นมีความสำคัญในการใช้กำหนดคีย์เวิร์ดเช่นกัน

10. A เป็นแท็กที่ใช้สำหรับลิงก์ไปยังเว็บเพจอื่นที่เกี่ยวข้อง ซึ่งข้อความที่อยู่ในส่วนของแท็กนี้จะมีความน่าจะเป็นคีย์เวิร์ดได้สูง

11. **DFN** ในคำศัพท์ต่าง ๆ มักจะมีคำอธิบายซึ่งคำอธิบายเหล่านี้เองอาจจะสามารถนำมาใช้ในการเพิ่มค่าน้ำหนักได้ แท็กนี้เป็นการกำหนดข้อความว่าเป็นคำอธิบายของคำศัพท์

12. **DD, BR , P** เป็นการนำเอาคำหลังแท็กเหล่านี้มาพิจารณา เพื่อเป็นการเพิ่มความถี่ของคำที่อาจจะพบได้หลังแท็กเหล่านี้

ซึ่งเมื่อข้อมูลที่เป็นซอร์สโค้ดแฮชทีเอ็มเข้ามาในโปรแกรมจะมีลักษณะดังแสดงในรูปที่ 3-8 และหลังจากทำการตัดคำ พร้อมกับเก็บเฉพาะแท็ก และข้อมูลที่อยู่หลังแท็กที่พิจารณาแล้วเท่านั้น ก็จะได้ซอร์สโค้ดแฮชทีเอ็มแอลมีลักษณะดังรูปที่ 3-9

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>E-LIB ทำอย่างไร ถึงจะใช้ข้อมูลจากเว็บให้คุ้ม</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-874">
<META content="E LIB" name=Author LIBRARY? ?ELECTRONIC>
<META content="MSHTML 5.50.3825.1300" name=GENERATOR>
<TABLE bgColor=#800000>
 <TBODY>
 <TR><TD><FONT color=#ffff00 size=4><B>การ ค้น หา ข้อ มูล ข่าว สาร
    ทางอิน เทอร์เน็ต โดย ใช้ สิ่ง ที่ เรียก ว่า "โฮม เพจ"
    อาจ จะ ไม่ ใช่ เรื่อง ที่ น่า จะ สะดวก สบาย นัก ที่ เรา จะ ต้อง ค้น หา ข้อ มูล
    จากเว็บ เพจ ที่ มี ประมาณ 300 ล้าน เพจ แต่ มี วิธี ที่ จะ ทำ ให้ ได้ ที่ อยู่ ของ ไซต์
    ที่ คุณ ต้อง การ ได้ ใน เวลา อัน รวด เร็ว โดย สิ่ง ที่ เรียก ว่า "Web Search Tool"
    และ คุณ จะ รู้ ถึง การ ทำ งาน ของ มัน ใน บท ความ นี้ ด้วย
</B></FONT></TD></TR>
</TBODY>
</TABLE>
```

รูปที่ 3-8 โค้ดแฮชทีเอ็มแอลก่อนทำการเก็บแท็ก

```
<TITLE>E-LIB ทำอย่างไร ถึงจะใช้ข้อมูลจากเว็บให้คุ้ม</TITLE>
<FONT color=#ffff00 size=4><B>การ ค้น หา ข้อ มูล ข่าว สาร
    ทางอิน เทอร์เน็ต โดย ใช้ สิ่ง ที่ เรียก ว่า "โฮม เพจ"
    อาจ จะ ไม่ ใช่ เรื่อง ที่ น่า จะ สะดวก สบาย นัก ที่ เรา จะ ต้อง ค้น หา ข้อ มูล
    จากเว็บ เพจ ที่ มี ประมาณ 300 ล้าน เพจ แต่ มี วิธี ที่ จะ ทำ ให้ ได้ ที่ อยู่ ของ ไซต์
    ที่ คุณ ต้อง การ ได้ ใน เวลา อัน รวด เร็ว โดย สิ่ง ที่ เรียก ว่า "Web Search Tool"
    และ คุณ จะ รู้ ถึง การ ทำ งาน ของ มัน ใน บท ความ นี้ ด้วย
</B></FONT>
```

รูปที่ 3-9 โค้ดแฮชทีเอ็มแอลที่เก็บแท็กมาได้

ส่วนในเรื่องของความหมายของแท็กแต่ละแท็กนั้นสามารถดูเพิ่มเติมได้จากส่วนทฤษฎี บท ซึ่งอยู่ในหัวข้อ 2.5 หลังจากเก็บแท็ก และข้อความหลังแท็ก เฉพาะส่วนที่มีผลต่อการหาคีย์เวิร์ดแล้ว ก็จะนำผลที่ได้มาพิจารณาในการตัดคำที่ไม่มีความหมายพอที่จะเป็นคีย์เวิร์ดออก

โดยหลังจากทำการทดลอง และวิเคราะห์จากเว็บเพจที่ได้ทำการสุ่มจำนวน 500 เว็บเพจ ซึ่งเป็นเว็บเพจที่มีเนื้อหาเฉพาะเรื่อง เช่น เว็บเรื่อง "ดาวหาง" "น้ำหอม" "การใช้งานอินเตอร์เน็ต" เป็นต้น โดยไม่ใช่เว็บเพจที่เป็นลักษณะเว็บเพจแบบ Sanook, Yahoo หรือ AltaVista ซึ่งเป็นเว็บเพจที่รวมเอาเว็บไซต์ต่าง ๆ มากมายเอาไว้หลายประเภท ซึ่งจากทั้ง 500 เว็บเพจที่ทำการสุ่มเลือกมานี้ เราจะ

สามารถตัดแท็กที่มีความสำคัญต่อคำที่จะเป็นคีย์เวิร์ดน้อยมาก ๆ ออก โดยแท็กที่พิจารณาจะเหลือแท็กดังต่อไปนี้

1. แท็ก TITLE	5. แท็ก U
2. แท็ก Hn	6. แท็ก FONT
3. แท็ก CENTER	7. แท็ก IMG
4. แท็ก B, BIG, STRONG	8. แท็ก A

3.2.2 ส่วนที่ใช้ในการกำจัดคำสต็อปเวิร์ดทิ้ง

จากทฤษฎีการกำหนดดัชนี กล่าวไว้ว่า "คำที่มักจะปรากฏบ่อยครั้งนั้นส่วนใหญ่เป็นคำจำพวก the, and, of และ a (คำเหล่านี้เราเรียกว่าเป็นคำที่มีอยู่ทั่วไปหรือคำที่เป็นสามัญ (common words) ซึ่งคำเหล่านี้มักจะปรากฏเป็นส่วนใหญ่ในข้อความหรือเอกสารต่าง ๆ และคำเหล่านี้ถือได้ว่าเป็นคำดัชนีที่ไม่ดีพอหรือแย่มากด้วยเหตุผล 2 ประการ คือ หนึ่ง คำเหล่านี้ปรากฏบ่อยเกินไป และปรากฏในเกือบทุก ๆ ข้อความ หรือเอกสาร สอง คำเหล่านี้ไม่มีส่วนเกี่ยวพันหรือมีน้อยกับใจความในเอกสาร และโดยตัวมันเองแล้วไม่ค่อยมีความหมาย"

โดยทั่วไปแล้วคำในภาษาต่าง ๆ จะแบ่งออกเป็น คำนาม คำสรรพนาม คำบุพบท เป็นต้น ซึ่งคำเหล่านี้ก็สื่อความหมายในลักษณะที่ต่างกัน จะเห็นว่าคำบุพบท เช่น คือ เป็น กับ ฯลฯ เมื่ออยู่โดด ๆ จะไม่สามารถแสดงความหมาย ในตัวมันเองได้ คำที่ไม่สามารถสื่อความหมายหรือ มีความหมายเมื่ออยู่โดด ๆ ได้นี้ เรียกว่า **คำสต็อปเวิร์ด** ซึ่งจะมีในทุก ๆ ภาษา และคำที่เป็นคำสต็อปเวิร์ด นั้นสามารถดูได้จากภาคผนวก ก

การทำงานของโปรแกรมในส่วนนี้ คำสต็อปเวิร์ดจะเก็บไว้ในไฟล์เท็กซ์ที่ชื่อ Stopword ซึ่งอยู่ภายนอกโปรแกรม โปรแกรมจะทำการเปิดไฟล์นี้ขึ้นมาเพื่อจะนำเอาคำสต็อปเวิร์ดเหล่านั้นแต่ละคำมาทำการหาในโค้ดแฮชทีเอ็มแอล ถ้าหากพบว่าคำนั้นเป็นคำสต็อปเวิร์ดแล้วก็จะทำการตัดคำนั้นทิ้ง ตัวอย่างการกำจัดคำสต็อปเวิร์ดดังรูปที่ 3-10 และรูปที่ 3-11

```
<HTML>
<HEAD>
<TITLE>การ พัฒนา ของ super computer</TITLE>
<BR>   วิวัฒนาการ ของ คอมพิวเตอร์ ได้ ก้าว มา ไกล มาก จาก
       เครื่อง คอมพิวเตอร์ ที่ มี ขนาด ใหญ่ เท่า ห้อง
       ที่ ทำ งาน ด้วย หลอด สุญญากาศ
       ทุก วัน นี้
       แทบ ทุก โต๊ะ ใน สำนัก งาน แทบ ทุก แห่ง
       จะ มี เครื่อง คอมพิวเตอร์ ตั้ง อยู่
</HEAD>
</HTML>
```

รูปที่ 3-10 โค้ดแฮชทีเอ็มแอลก่อนทำการกำจัดสต็อปเวิร์ด

```
<HTML>
<HEAD>
<TITLE>  พัฒนา  super computer</TITLE>
<BR>   วิวัฒนาการ คอมพิวเตอร์  ก้าว  ไกล
       เครื่อง คอมพิวเตอร์   ขนาด ใหญ่  ห้อง
       ทำ งาน ด้วย หลอด สุญญากาศ
       วัน
       แทบ โต๊ะ  สำนัก งาน แทบ
       เครื่อง คอมพิวเตอร์ ตั้ง
</HEAD>
</HTML>
```

รูปที่ 3-11 โค้ดแฮชทีเอ็มแอลหลังทำการกำจัดสต็อปเวิร์ด

จากรูปที่ 3-11 จะเห็นได้ว่าไม่มีคำสต็อปเวิร์ดปรากฏอยู่ในโค้ดแล้ว ซึ่งจะช่วยทำให้เวลาในการประมวลผลลดน้อยลง

3.2.3 ส่วนที่ใช้ในการกำจัดคำที่ใช้แทนสัญลักษณ์ในภาษาแฮชทีเอ็มแอล

จากการศึกษาเว็บเพจที่ได้รับความนิยมโดยทั่วไปนั้น บางครั้งจะมีการใช้สัญลักษณ์พิเศษเพื่อกำหนดให้กับเนื้อหาของผู้เขียนที่ต้องการแสดงให้ผู้อ่านเห็นเด่นชัด หรือดูสวยงาม เช่น ©, θ, ϕ, ® เป็นต้น ซึ่งไม่สามารถพิมพ์ได้จากแป้นพิมพ์ตามปกติ ดังนั้นจึงต้องมีคำสั่งพิเศษที่ใช้เขียนเพื่อแสดงสัญลักษณ์พิเศษเหล่านี้ได้

ซึ่งโค้ดที่ใช้แทนสัญญณ์เหล่านี้ไม่มีหรือมีความเกี่ยวข้องกับตัวเนื้อหาของเว็บเพจน้อยมาก ดังนั้นจึงทำการพิจารณาตัดคำเหล่านี้ออกไปด้วย โดยจะถือว่าเป็นคำจำพวกเดียวกับคำสต็อปเวิร์ด การทำงานในส่วนนี้มีลักษณะคล้ายกับส่วนที่ใช้ในการกำจัดคำสต็อปเวิร์ดทิ้ง แต่ต่างกันที่การกำจัดคำที่ใช้แทนสัญลักษณ์นี้จะกำจัดสัญลักษณ์เหล่านี้ทั้งที่อยู่ติดกับคำ หรืออยู่แยกเฉพาะ ตัวอย่างการกำจัดดังรูปที่ 3-12 และรูปที่ 3-13

```
<HTML>
<HEAD>
<TITLE>&copy;การ พัฒนา ของ super computer&copy;</TITLE>
<BR>  วิวัฒนาการ ของ คอมพิวเตอร์ ได้ ก้าว มา ไกล มาก จาก
      เครื่อง คอมพิวเตอร์ ที่ มี ขนาด ใหญ่ เท่า ห้อง
      ที่ ทำ งาน ด้วย หลอด สุญญากาศ
      "ทุก วัน นี้ ####
      แทบ ทุก โต๊ะ ใน สำนัก งาน แทบ ทุก แห่ง
      จะ มี เครื่อง คอมพิวเตอร์ ตั้ง อยู่"
</HEAD>
</HTML>
```

รูปที่ 3-12 แสดงโค้ดแฮชทีเอ็มแอลก่อนทำการกำจัดสัญลักษณ์

```
<HTML>
<HEAD>
<TITLE> พัฒนา super computer</TITLE>
<BR>  วิวัฒนาการ คอมพิวเตอร์ ก้าว ไกล
      เครื่อง คอมพิวเตอร์ ขนาด ใหญ่ ห้อง
      ทำ งาน ด้วย หลอด สุญญากาศ
      วัน
      แทบ โต๊ะ สำนัก งาน แทบ
      เครื่อง คอมพิวเตอร์ ตั้ง
</HEAD>
</HTML>
```

รูปที่ 3-13 แสดงโค้ดแฮชทีเอ็มแอลหลังทำการกำจัดสัญลักษณ์ และกำจัดสต็อปเวิร์ด

ซึ่งโค้ดแฮชทีเอ็มแอลที่เหลือ จะมีแต่แท็กที่ใช้พิจารณาในการกำหนดคีย์เวิร์ดตามที่ได้กำหนดเอาไว้ และกำจัดสต็อปเวิร์ดเรียบร้อยแล้ว ก็จะเริ่มทำการกำหนดคีย์เวิร์ดโดยพิจารณาจากความถี่และค่าน้ำหนักของคำ

3.2.4 ส่วนที่ใช้ในการกำหนดคีย์เวิร์ด

หลังจากทำส่วนต่าง ๆ ในหัวข้อข้างบนแล้ว ส่วนต่อไปก็จะเป็นส่วนหลักของโปรแกรม โดยแนวความคิดในส่วนนี้เริ่มจากทำการแทนช่องว่างระหว่างคำด้วยแท็กแฮชทีเอ็มแอล <WBR> เพื่อเป็นการสะดวกในการแยกพิจารณาคำแต่ละคำ

```
<HTML>
<HEAD>
<TITLE>สิว</TITLE>
<BR>สิว เป็น จุด หรือ ตุ่ม เล็ก สิว หัว แดง
        เมื่อ สุก เต็ม หัว หนอง สี เหลือง
</HEAD>
</HTML>
```

รูปที่ 3-14 โค้ดแฮชทีเอ็มแอลก่อนการแทนด้วยแท็ก <WBR>

```
<HTML>
<HEAD>
<TITLE> สิว</TITLE>
<BR>สิว<WBR>เป็น<WBR>จุด<WBR>หรือ<WBR>ตุ่ม<WBR>
เล็ก<WBR>สิว<WBR>หัว<WBR>แดง<WBR>เมื่อ<WBR>สุก<WBR>เต็ม
<WBR>หัว<WBR>หนอง<WBR>สี<WBR>เหลือง
</HEAD>
</HTML>
```

รูปที่ 3-15 โค้ดแฮชทีเอ็มแอลหลังการแทนด้วยแท็ก <WBR>

จะเห็นว่าหลังการแทนแท็ก <WBR> ลงไปแล้ว ทำให้คำแต่ละคำจะอยู่ระหว่างแท็กเสมอ ซึ่งจะทำให้การทำงานของโปรแกรมจะทำการไล่เก็บแท็ก และคำตั้งแต่บนสุดของโค้ดแฮชทีเอ็มแอลไปจนจบทำได้ง่าย และสะดวกขึ้น การไล่โค้ดแฮชทีเอ็มแอลจะมีผลดังนี้

เมื่อโปรแกรมทำการสแกนจนพบแท็ก ก็จะนำเอาแท็กที่พบมาตรวจสอบกับแท็กที่เก็บไว้ในไฟล์เท็กซ์ที่ชื่อ WeightTag โดยไฟล์เท็กซ์นี้จะเก็บแท็ก และค่าน้ำหนักของแท็กเอาไว้ดังรูปที่ 3-16 หากเป็นแท็กที่อยู่ในรูปที่ 3-16 แล้วก็จะทำการเก็บแท็กนั้นลงในสแต็ก พร้อมกับให้ค่าน้ำหนักตามค่าที่เก็บไว้ ซึ่งแท็กที่ถูกเก็บไว้ในสแต็กเหล่านี้จะถูกเอาออก ก็ต่อเมื่อแท็กที่พบเป็นแท็กปิด เพราะฉะนั้นหากแท็กใดไม่มีแท็กปิด ก็จะไม่ทำการเก็บแท็กนั้นลงในสแต็ก

เนื่องจากแท็กบางตัวได้แก่แท็ก FONT และIMG จะพิจารณาค่าแอททริบิวต์ของแท็กบางค่าเท่านั้น ซึ่งจำเป็นกับการพิจารณาความสำคัญของคำที่เป็นคีย์เวิร์ดได้ ซึ่งจะต้องมีการเก็บลงในสแต็กดังต่อไปนี้

1. แท็ก **FONT** ในแท็กนี้แอททริบิวต์ที่ใช้ในการให้ค่าน้ำหนักก็คือ SIZE ซึ่งเป็นแอททริบิวต์ที่ใช้กำหนดขนาดของตัวอักษรที่ต้องการแสดงในเว็บเพจ เนื่องจากขนาดของคำที่ต่างกันนั้นย่อมมีความสำคัญที่จะเป็นคีย์เวิร์ดได้ต่างกัน คือ คำที่มีขนาดตัวอักษรที่ใหญ่กว่าย่อมมีความสำคัญมากกว่าคำที่มีขนาดตัวอักษรที่เล็กกว่า เพราะเป็นการบ่งบอกถึงความสำคัญ หรือบ่งบอกว่าผู้เขียนต้องการเน้นคำๆนั้นมากเป็นพิเศษ เพราะฉะนั้นเมื่อพบแท็ก FONT จะมีกรณีต่าง ๆ ดังนี้

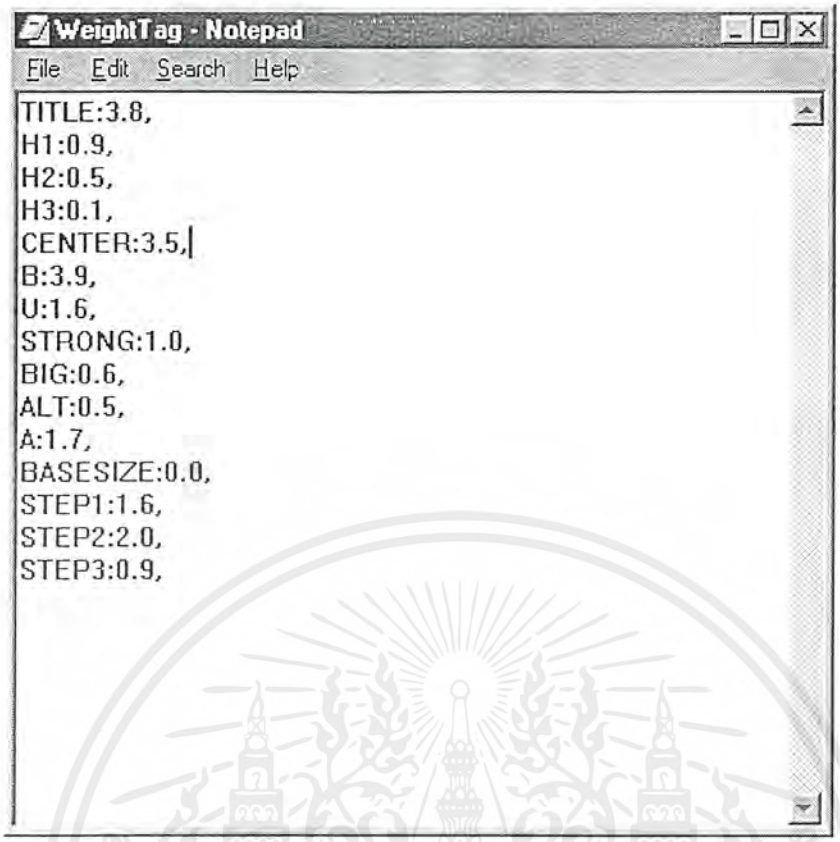
ก) ถ้าหากภายในแท็กไม่มีแอททริบิวต์ SIZE ก็จะให้ค่าขนาดตัวอักษรของคำเป็นขนาดฐาน (BaseSize) ซึ่งมีค่าเป็น 3 โดยเป็นค่า Default ที่ทางเว็บเบราวเซอร์ทำการกำหนดให้กับตัวอักษรภายในเว็บเพจนั้นหากผู้เขียนไม่ได้ทำการกำหนดขนาดของตัวอักษรที่เป็นค่าฐานให้กับเว็บเพจ และให้ทำการจัดเก็บแท็กในสแต็กเป็น BASESIZE

ข) ถ้าหากพบแอททริบิวต์ SIZE ก็ให้ทำการจัดเก็บค่าขนาดตัวอักษรของคำตามที่ได้กำหนด ในแอททริบิวต์ SIZE เช่น <FONT SIZE="2"> จะเก็บค่าขนาดของตัวอักษรของคำที่อยู่ในแท็กนี้ลงในสแต็กเป็น SIZE2 และ <FONT FACE="ARIAL,HELVETICA,SANS-SERIF" SIZE="1"> จะเก็บแท็กลงใน สแต็กเป็น SIZE1 เป็นต้น

เมื่อได้ขนาดของตตัวอักษรที่ SIZE ต่าง ๆ แล้ว โดยทำการตรวจสอบดูว่าค่าขนาดของตัวอักษรที่กำหนดมีค่ามากกว่า น้อยกว่า หรือเท่ากับขนาดของตัวอักษรฐานที่เป็น BASESIZE

- หากค่าขนาดของตัวอักษรมีค่ามากกว่า ให้นำมาลบกับค่าขนาดตัวอักษรที่เป็นฐานแล้วกำหนดเก็บลงในสแต็กเป็น STEP(ผลต่างของขนาดของตัวอักษรกับขนาดตัวอักษรที่เป็นฐาน) เช่น ถ้าหากของตัวอักษรฐานเป็น SIZE3 ซึ่งเป็น BASESIZE และแท็ก FONT เป็น <FONT FACE="CordiaUPC" SIZE="7"> ก็จะได้ว่าเป็น STEP4 ถ้าเป็น <FONT FACE="ARIAL,HELVETICA,SANS-SERIF" SIZE="5"> ก็จะได้ว่าเป็น STEP2 เป็นต้น
- หากค่าขนาดของตัวอักษรมีค่าน้อยว่า ให้นำมาลบกับค่าขนาดตัวอักษรที่เป็นฐานแล้วกำหนดเก็บลงในสแต็กเป็น SIZE แล้วตามด้วยขนาดของตัวอักษรที่ระบุในแอททริบิวต์ SIZE เช่น ถ้าหากของตัวอักษรฐานเป็น SIZE3 ซึ่งเป็น BASESIZE และแท็ก FONT เป็น <FONT SIZE="2"> ก็จะเก็บลงสแต็กเป็น SIZE2 เป็นต้น
- หากค่าขนาดของตัวอักษรมีค่ามากกว่า ให้นำมาลบกับค่าขนาดตัวอักษรที่เป็นฐานแล้วกำหนดเก็บลงในสแต็กเป็น BASESIZE

ค) ถ้าหากพบเครื่องหมาย +,- หลังแอททริบิวต์ SIZE หมายความว่า มีการเพิ่มหรือลดขนาดของตัวอักษรจากค่าขนาดของตัวอักษรที่เป็นฐาน ก็ให้นำค่าที่บวก หรือลบนี้เพิ่มต่อท้ายค่า STEP ที่จะเก็บลงสแต็ก เช่น <FONT SIZE=+1> เมื่อเบสไซต์มีค่าเท่ากับ 3 ก็จะทำการเก็บแท็กลงสแต็กเป็น STEP1 เป็นต้น



รูปที่ 3-16 แสดงแท็ก และค่าน้ำหนักของแต่ละแท็กที่เก็บไว้ในไฟล์เท็กซ์ชื่อ WeightTag

2. แท็ก **IMG** แอททริบิวต์ที่ใช้ในการให้ค่าน้ำหนักคือ ALT โดย ALT จะใช้เก็บข้อความที่อธิบายถึงรูปภาพ เมื่อเว็บเบราวเซอร์ไม่สามารถโหลดรูปภาพจากเซิร์ฟเวอร์ได้ ก็จะทำการแสดงข้อความที่อยู่ภายใน ALT แทน ตัวอย่างของแท็ก IMG ที่มีการใช้แอททริบิวต์ ALT เช่น

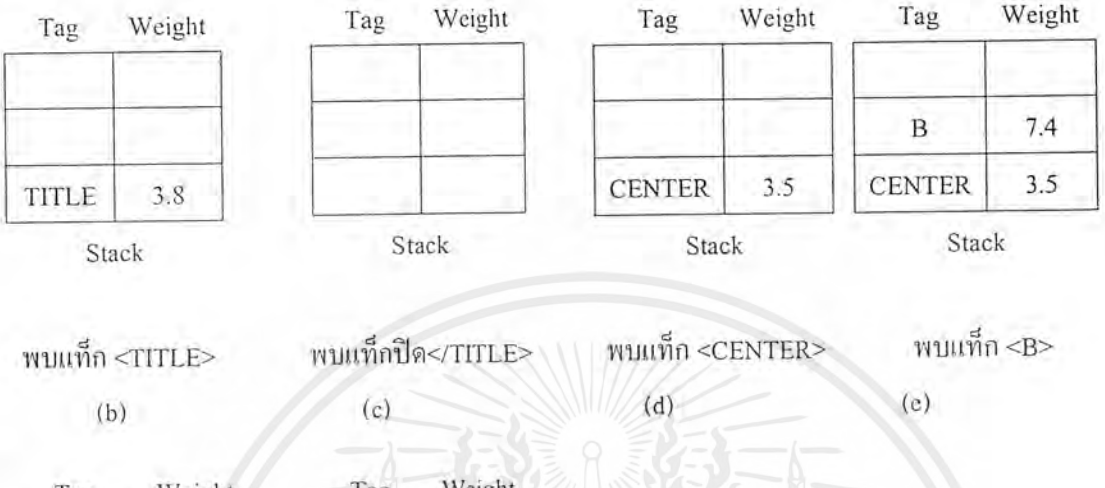
<IMG SRC="/IMAGES/003/LARRYKING.GIF" ALIGN=RIGHT WIDTH="55" HEIGHT="45" ALT="LARRY" BORDER="0">

คำที่อยู่หลังแอททริบิวต์ ALT ก็จะถูกนำมาใช้เป็นคีย์เวิร์ดได้ เพราะฉะนั้นถ้าหากแท็ก **IMG** ไม่มีแอททริบิวต์ ALT ก็จะไม่นำมาใช้ในการพิจารณาคีย์เวิร์ด และถ้าหากพบแอททริบิวต์ดังกล่าวก็จะทำการบวกค่าน้ำหนักให้แก่คำที่อยู่หลังแอททริบิวต์ ALT หลังจากให้ค่าน้ำหนักคำนั้นไปแล้วก็จะทำการลดค่าน้ำหนักนั้นลง เนื่องจากแท็ก **IMG** ไม่มีแท็กปิดจึงไม่ต้องทำการเก็บแท็กนี้ไว้ในสแต็ก ตัวอย่างการเก็บแท็กของโปรแกรมดังรูปที่ 3-17

<TITLE> สิว</TITLE>

<CENTER>สิว อักเสบ<B>ร้าย แรง </B>อันตราย</CENTER>

(a)



รูปที่ 3-17 ตัวอย่างการเก็บแท็ก และค่าน้ำหนักลงในสแต็ก

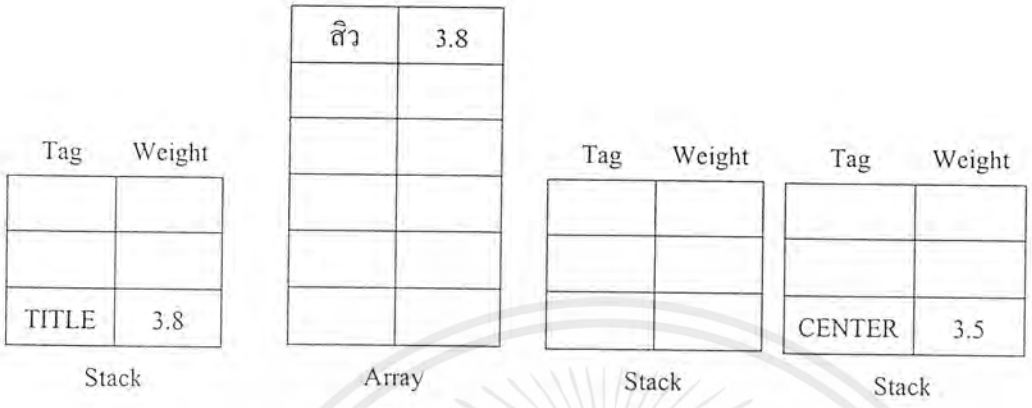
จากรูปที่ 3-17 (a) เป็นซอร์สโค้ด ซึ่งการไล่จะเริ่มตามลำดับตั้งแต่ (b) ถึง (g) รูปที่ 3-17 (b) เริ่มตั้งแต่ลำดับแรกพบแท็ก TITLE ก็จะทำการเก็บแท็กลงในสแต็กแล้วเก็บคำที่พบภายในแท็กนี้พร้อมกับให้ค่าน้ำหนักของคำตามค่าน้ำหนักของแต่ละแท็ก จากนั้นเมื่อสแกนจนเจอแท็กปิดของ TITLE ก็จะทำการป๊อบค่าออกจากสแต็ก รูปที่ 3-17 (d) และ 3-17(e) สแกนพบแท็ก <CENTER> และแท็ก <B> ตามลำดับทำการเก็บลงสแต็กพร้อมกับให้ค่าน้ำหนักของคำที่อยู่หลังแท็กนี้ รูปที่ 3-17 (f) และ 3-17(g) สแกนพบแท็กปิดของแท็ก CENTER และ B ก็ทำการป๊อบค่าออกจากสแต็ก

ต่อจากนั้นหากตรวจพบคำแล้ว เมื่อพบคำก็จะนำคำนั้นไปเก็บไว้ในอาเรย์ ซึ่งอาเรย์นี้จะทำการเก็บคำที่จะเป็นคีย์เวิร์ด และค่าน้ำหนักของคีย์เวิร์ดนั้นด้วย โดยค่าน้ำหนักที่ได้จะได้มาจากค่าน้ำหนักบนสุดของสแต็ก ก่อนที่จะทำการเก็บลงในอาเรย์ได้ต้องมีการตรวจดูว่าคำที่ได้นั้นเคยถูกเก็บไว้ในอาเรย์แล้วหรือไม่ ถ้าหากเคยถูกเก็บแล้วก็จะทำการบวกค่าน้ำหนักเดิมเข้ากับค่าน้ำหนักที่ได้มาใหม่ ตัวอย่างการเก็บคำ และค่าน้ำหนักดังรูปที่ 3-18

<TITLE> สิว</TITLE>

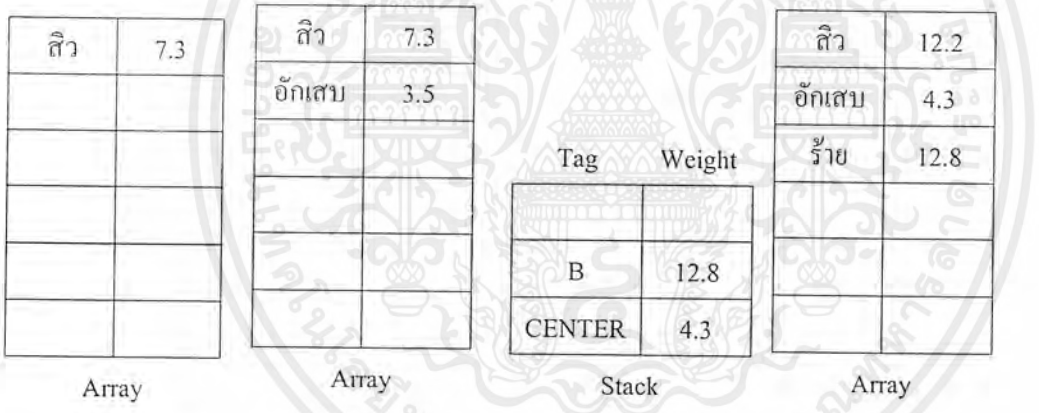
< CENTER >สิว อักเสบ<B>ร้าย แรง </B>อันตราย</ CENTER >

(a)



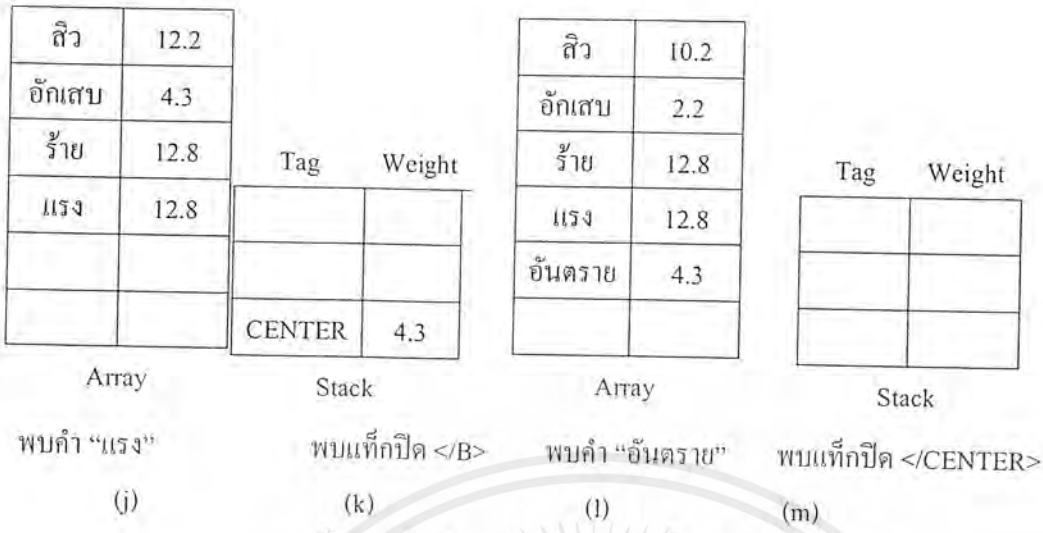
พบแท็ก<TITLE> พบคำ "สิว" คำแรก พบแท็กปิด</TITLE> พบแท็ก<CENTER>

(b) (c) (d) (e)



พบคำ "สิว" คำที่ 2 พบคำ "อักเสบ" พบแท็กปิด<B> พบคำว่า "ร้าย"

(f) (g) (h) (i)



รูปที่ 3-18 ตัวอย่างการเก็บคำ และค่าน้ำหนักลงในสแต็ก

ตัวอย่างในรูปที่ 3-18 นี้ให้ดูค่าน้ำหนักของแท็กในรูปที่ 3-16 ประกอบด้วย โดยค่าน้ำหนักของแท็ก TITLE เป็น 3.8 แท็ก CENTER เป็น 3.5 และแท็ก B เป็น 3.9 จะสังเกตเห็นว่าเมื่อพบคำว่า "สิว" หลังแท็ก TITLE ซึ่งมีค่าน้ำหนักเป็น 3.8 และต่อมาพบคำว่า "สิว" อีกครั้งหลังแท็ก CENTER ก็จะทำการบวกค่าน้ำหนักจาก CENTER คือ 3.5 กับของเดิมคือ 3.8 ซึ่งจะได้ค่าน้ำหนักใหม่เป็น 7.3 นั่นเอง

ซึ่งเมื่อเจอคำที่ซ้ำกันมาก ๆ แสดงว่าคำนั้นมีค่าความถี่มาก และเป็นการเพิ่มค่าน้ำหนักให้มากขึ้นด้วย คำ ๆ นั้นจึงมีโอกาสที่จะเป็นคีย์เวิร์ดเพิ่มมากขึ้น

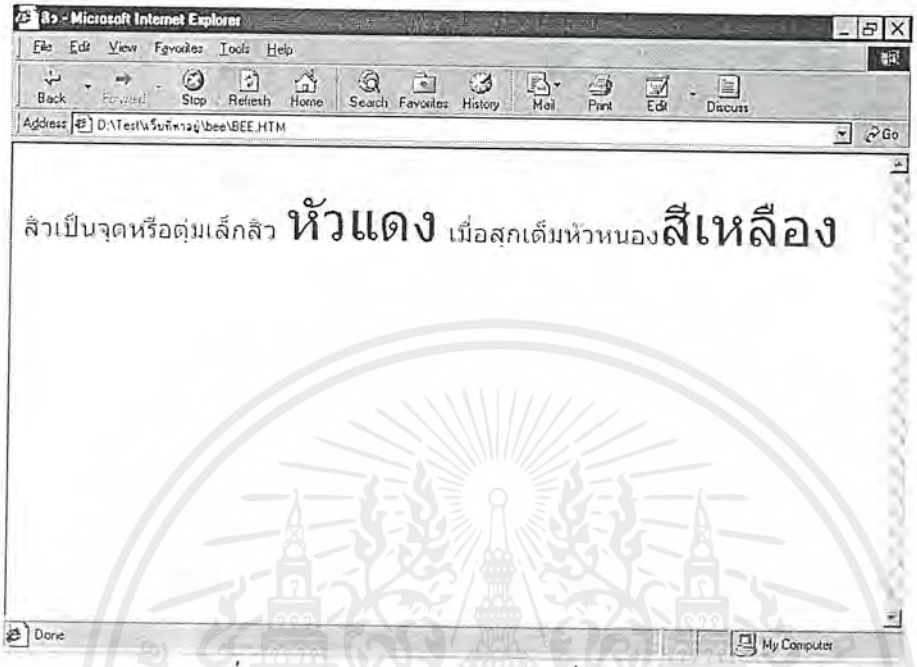
ในกรณีที่เว็บเพจนั้นมีการใช้แท็กเปิด แต่ปรากฏว่าไม่มีแท็กปิดตามหลัง ซึ่งผิดไวยากรณ์ของการเขียนภาษาแฮชทีเอ็มแอล แต่เนื่องจากว่าเว็บเบราวเซอร์ในปัจจุบันมีความสามารถสูง ดังนั้น ถึงแม้จะไม่มีการปิดแท็กตามไวยากรณ์ของภาษาแฮชทีเอ็มแอลก็ตาม เว็บเบราวเซอร์ก็ไม่แสดงข้อความผิดพลาดออกมาแต่อย่างใด อีกทั้งยังสามารถแสดงผลได้อีกด้วย

ถ้าแท็กเปิดตัวใดไม่มีแท็กปิด แล้วเว็บเบราวเซอร์จะถือว่าข้อความตั้งแต่หลังแท็กเปิดนั้นมีคุณสมบัติตามแท็กเปิดนั้นไปจนหมดโค้ดแฮชทีเอ็มแอล ยกเว้นข้อความซึ่งอยู่ระหว่างแท็กเปิดและแท็กปิด จะมีคุณสมบัติตามแท็กที่มันอยู่ ดังรูปที่ 3-19

```
<HTML>
<HEAD>
<TITLE>สิว</TITLE>
        <BR><FONT SIZE="5">สิวเป็นจุดหรือตุ่มเล็กสิว
        <FONT SIZE="7">หัวแดง</FONT>
        เมื่อสุกเต็มหัวหนอง<FONT SIZE="7">สีเหลือง</FONT>
</HEAD>
</HTML>
```

รูปที่ 3-19 โค้ดแฮชทีเอ็มแอลในกรณีที่ไม่มีแท็กปิด

จากรูปที่ 3-19 จะเห็นว่าข้อความใดที่อยู่หลังแท็ก <FONT SIZE="5"> จะมีขนาดของตัวอักษรเป็น 5 ไปจนหมดโค้ด แต่ข้อความ "หัวแดง" และ "สีเหลือง" จะมีขนาดของตัวอักษรเป็น 7 เนื่องจากอยู่ภายในแท็ก <FONT SIZE="7"> ซึ่งผลลัพธ์เมื่อแสดงออกหน้าจอเว็บเบราวเซอร์จะเป็นดังรูปที่ 3-19

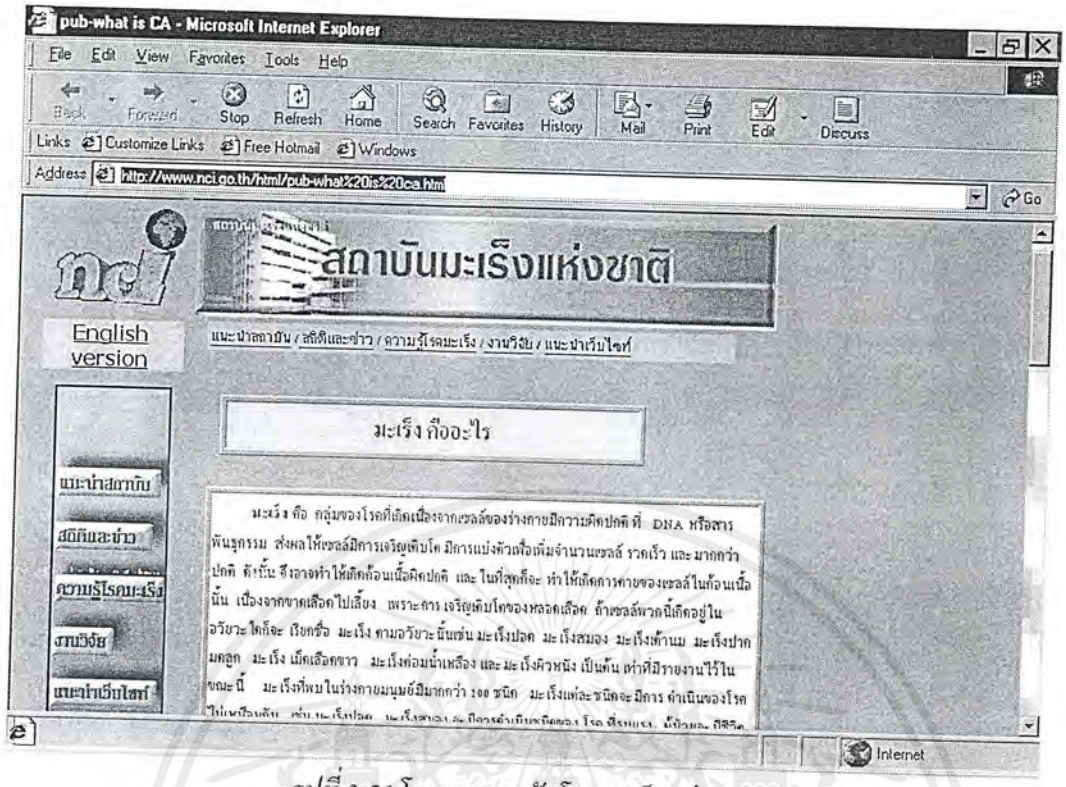


รูปที่ 3-20 เว็บเบราวเซอร์แสดงผลเมื่อไม่มีแท็กปิด

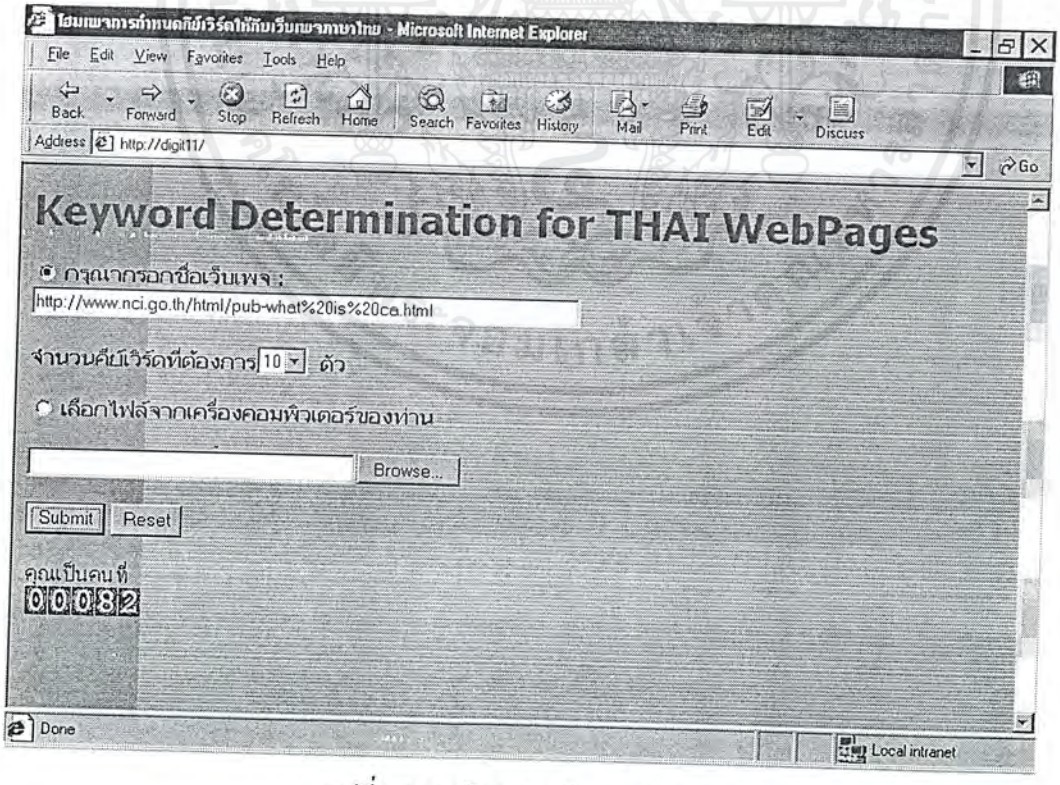
ซึ่งการทำงานในส่วนนี้เป็นการเก็บแท็กพร้อมกับค่าน้ำหนักของแท็ก ดังนั้นถ้าเป็นการทำงานกับโค้ดในรูปที่ 3-19 เมื่อตัวโปรแกรมสแกนเจอแท็ก FONT ตัวแรกจะเก็บค่าน้ำหนักตามขนาดของตัวอักษร และข้อความที่อยู่หลังแท็กนี้ก็จะมีค่าน้ำหนัก แต่เมื่อสแกนจนเจอแท็ก <FONT> ตัวที่สอง ข้อความที่อยู่ภายในแท็กนี้ก็จะมีค่าน้ำหนักเป็นค่าน้ำหนักของแท็ก <FONT> ตัวแรกรวมกับค่าของแท็ก <FONT> ตัวที่สอง แต่เมื่อเจอแท็กปิด ข้อความที่อยู่หลังแท็กปิดก็จะมีค่าน้ำหนักเท่ากับค่าน้ำหนักของแท็ก FONT ตัวแรกตัวเดียวเท่านั้น และเมื่อเจอ FONT ตัวที่สาม ก็จะเหมือนกับ <FONT> ตัวที่สอง

3.3 ตัวอย่างการใช้โปรแกรมผ่านอินเตอร์เน็ต

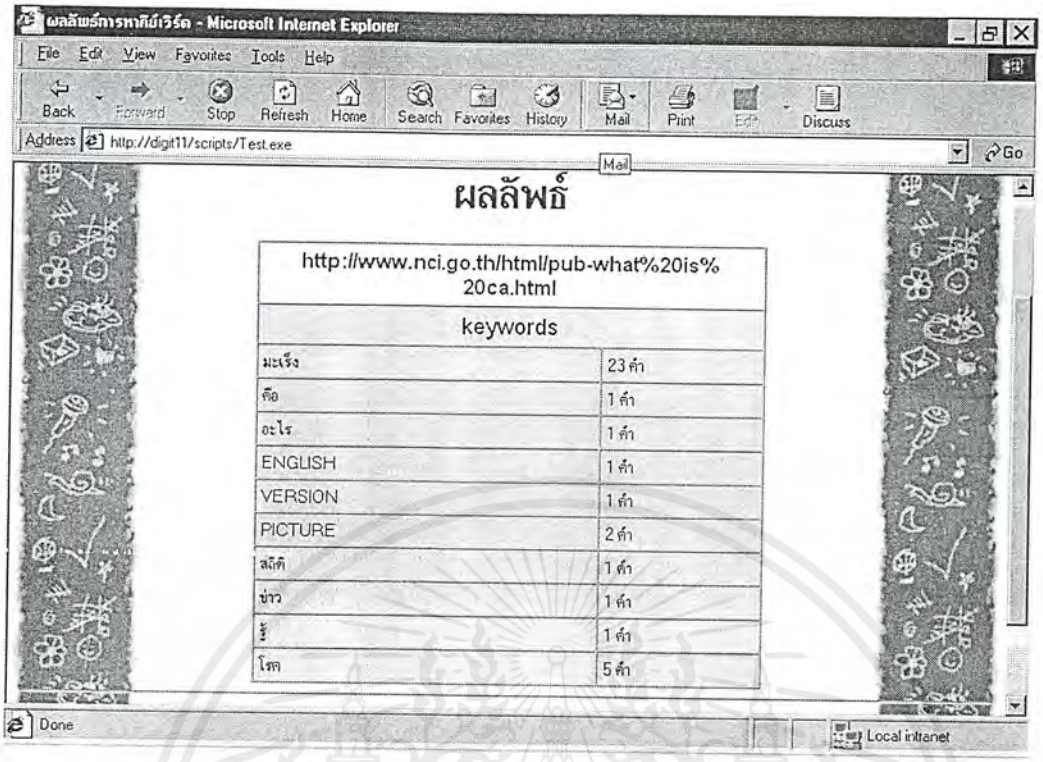
ตัวอย่างการใช้งานโปรแกรมการกำหนดคีย์เวิร์ด โดยให้ผู้ใช้สามารถป้อนยูอาร์แอลที่ต้องการทำการกำหนดคีย์เวิร์ดลงบนฟอร์ม โดยทำการกรอกยูอาร์แอลเป็น http://www.nci.go.th/html/pub-what%20is%20ca.html ซึ่งเป็นเว็บไซต์ของสถาบันมะเร็งแห่งชาติ ซึ่งเป็นบทความเกี่ยวกับเรื่องโรคมะเร็ง ดังรูปที่ 3-21 และทำการกำหนดให้คีย์เวิร์ดที่ต้องการแสดงในหน้าจอผลลัพธ์เป็นจำนวนทั้งหมด 10 ตัว ดังรูปที่ 3-22 แล้วทำการคลิ๊กที่ปุ่ม Submit เพื่อแสดงผล ก็จะได้หน้าจอแสดงผลลัพธ์ดังรูปที่ 3-23



รูปที่ 3-21 โฮมเพจสถาบันโรคมะเร็งแห่งชาติ



รูปที่ 3-22 หน้าจอสำหรับกรอกข้อมูล



รูปที่ 3-23 หน้าจอผลลัพธ์จากการกำหนดคีย์เวิร์ด

จากรูปที่ 3-23 จะเห็นว่าจากการหาคีย์เวิร์ดบนโฮมเพจของสถาบันมะเร็งแห่งชาติ ซึ่งจะได้ผลลัพธ์เป็นคีย์เวิร์ดคำว่า "โรค" "มะเร็ง" "คือ" และ "อะไร" ปรากฏอยู่ในลำดับที่ 10, 1, 2 และ 3 ตามลำดับ

บทที่ 4

การวิเคราะห์ และการทดลอง

ในบทที่ 4 นี้เป็นการทดลองเพื่อหาค่าน้ำหนักที่เหมาะสม และทดสอบค่าน้ำหนักดังกล่าว ทั้งในแง่ความถูกต้อง และประสิทธิภาพ การทดลองจะแบ่งออกเป็น 2 ส่วน

การทดลองที่ 1

วัตถุประสงค์การทดลอง

เพื่อวิเคราะห์หาค่าน้ำหนักที่เหมาะสมของแท็กแฮชทีเอ็มแอลต่าง ๆ ในการกำหนดคีย์เวิร์ด

ขั้นตอนการทดลอง

1. สุ่มเว็บเพจตัวอย่างจำนวน 500 เว็บเพจ ซึ่งเป็นบทความทั่วไปที่มีหลาย ๆ เรื่องคละกัน โดยใน 500 เว็บเพจนี้ จะประกอบไปด้วย

เว็บเพจประเภทศิลปะ และวรรณกรรม	จำนวน	30	เว็บเพจ
เว็บเพจประเภทธุรกิจ และเศรษฐกิจ	จำนวน	50	เว็บเพจ
เว็บเพจประเภทคอมพิวเตอร์ และอินเตอร์เน็ต	จำนวน	70	เว็บเพจ
เว็บเพจประเภทการศึกษา	จำนวน	50	เว็บเพจ
เว็บเพจประเภทบันเทิง	จำนวน	50	เว็บเพจ
เว็บเพจประเภทสุขภาพ	จำนวน	100	เว็บเพจ
เว็บเพจประเภทวิทยาศาสตร์	จำนวน	50	เว็บเพจ
เว็บเพจประเภทสังคม และวัฒนธรรม	จำนวน	50	เว็บเพจ
เว็บเพจประเภทกีฬา และนันทนาการ	จำนวน	50	เว็บเพจ

2. นำเว็บเพจที่หาได้ในข้อ 1 มาหาคีย์เวิร์ดโดยพิจารณาด้วยตา ซึ่งปกติแต่ละเว็บเพจจะมีจำนวนคีย์เวิร์ดมากกว่า 1 คำ

3. นำเว็บเพจแต่ละเว็บเพจมาพิจารณาโค้ดแฮชทีเอ็มแอล โดยพิจารณาว่า คำที่ถูกจัดว่าเป็นคีย์เวิร์ด อยู่ในขอบเขตของแท็กแฮชทีเอ็มแอลใดบ้าง ทั้งนี้จะเก็บสะสมจำนวนของคีย์เวิร์ดหลังแท็กต่าง ๆ เอาไว้ เพื่อทำการวิเคราะห์ต่อไป

แท็ก TITLE เป็นแท็กที่ใช้บอกชื่อเรื่องของเว็บเพจนั้น

แท็ก Hn เป็นแท็กที่ใช้กำหนดหัวข้อภายในเว็บเพจ

แท็ก CENTER เป็นแท็กที่ใช้สำหรับจัดให้ตัวอักษรอยู่ตรงกลางเว็บเพจ

แท็ก B เป็นแท็กที่ใช้ทำให้ตัวอักษรหนาขึ้น

แท็ก U เป็นแท็กที่ใช้สำหรับขีดเส้นใต้ให้กับตัวอักษร

แท็ก STRONG เป็นแท็กที่ใช้สำหรับเน้นตัวอักษร

แท็ก BIG เป็นแท็กที่ใช้สำหรับกำหนดให้ตัวอักษรเป็นตัวใหญ่

แท็ก IMG เป็นแท็กที่ใช้ในการแสดงรูปภาพ โดยพิจารณาแอททริบิวต์ ALT

แท็ก A เป็นแท็กที่ใช้เชื่อมโยงไปยังเพจอื่น หรือไซต์อื่น

แท็ก FONT เป็นแท็กที่ใช้กำหนดคุณลักษณะของตัวอักษร โดยพิจารณาแอททริบิวต์ SIZE

ในการพิจารณาแท็ก FONT จะพิจารณาหาแอททริบิวต์ BASEFONT SIZE ว่ามีค่าเท่าได้จากนั้นหาแท็ก FONT ที่มีแอททริบิวต์ SIZE มากกว่าค่า BASEFONT เป็น 1, 2 และมากกว่า 2 ขึ้นไป

เช่น BASEFONT มีค่าเป็น 2 ก็จะทำการพิจารณาหลังแท็ก FONT ที่มีแอททริบิวต์ SIZE ตั้งแต่ 3, 4 และ 5 ขึ้นไป การพิจารณาแท็ก FONT ในลักษณะนี้ เนื่องจากการเขียนเว็บเพจโดยทั่วไปของเว็บไซต์ส่วนใหญ่ จะเขียนด้วยขนาดของข้อความที่เท่ากันทั้งหน้า แต่หากมีข้อความสำคัญ หรือข้อความที่ใช้ขึ้นหัวข้อก็จะมีการเพิ่มขนาดของตัวอักษร เพื่อเน้นให้เห็นได้ชัดเจนยิ่งขึ้น หลังจากทำขั้นตอนทั้งหมดจากข้อ 1 ถึง 3 แล้ว จะนำมาสรุปหาค่าน้ำหนักของแท็กต่อไป

ผลการทดลอง

การทดลองที่ 1 แสดงดังตารางที่ 4-1

แท็ก	จำนวนเว็บเพจ	พบแท็ก (เว็บเพจ)	จำนวนคีย์เวิร์ดที่พิจารณาด้วยตาทั้งหมด(คำ)	จำนวนคีย์เวิร์ดที่พบหลังแท็กทั้งหมด(คำ)	คิดเป็นร้อยละ [1]
TITLE	500	473	2605	1011	38.81
H1	500	112	2605	237	9.10
H2	500	62	2605	137	5.26
H3	500	36	2605	49	1.88
CENTER	500	327	2605	925	35.51
B	500	348	2605	1014	38.93
U	500	175	2605	426	16.35
STRONG	500	98	2605	252	9.67
BIG	500	73	2605	164	6.30
ALT	500	337	2605	124	4.76
A	500	423	2605	435	16.70
FONT (+1)	500	186	2605	413	15.85
FONT (+2)	500	150	2605	538	20.65
FONT (>+2)	500	99	2605	225	8.64

[1] ได้มาจาก (จำนวนคีย์เวิร์ดที่พบหลังแท็กทั้งหมด / จำนวนคีย์เวิร์ดที่พิจารณาด้วยตาทั้งหมด) X 100

ตารางที่ 4-1 การหาคีย์เวิร์ดหลังแท็กต่าง ๆ

วิเคราะห์การทดลอง

แนวทางในการพิจารณาหาค่าน้ำหนักดังนี้ จากตารางที่ 4-1 พบคีย์เวิร์ดหลังแท็ก TITLE 1011 คำ เพราะฉะนั้น คำที่พบหลังแท็ก TITLE มีโอกาสจะเป็นคีย์เวิร์ดเท่ากับ 38.81 % และเช่นเดียวกัน คำที่พบหลังแท็ก H1 มีโอกาสจะเป็นคีย์เวิร์ดเท่ากับ 9.10 % ดังนั้นคำที่พบหลังแท็ก TITLE มีโอกาสที่จะเป็นคีย์เวิร์ด มากกว่าคำที่พบหลังแท็ก H1 ค่าน้ำหนักที่จะให้แท็ก TITLE ควรมีค่าน้ำหนัก มากกว่าแท็ก H1 ค่าที่จะนำมาใช้เป็นค่าน้ำหนักของแท็กต่างๆ จึงพิจารณาจากค่าร้อยละในตารางที่ 4-1 เพื่อลดค่าน้ำหนักให้เหลือตัวเลขที่เป็นช่วงที่แคบลง จึงใช้อัตราส่วนเป็นค่าน้ำหนักโดยหารด้วย 10 ได้ค่าน้ำหนักดังต่อไปนี้

แท็ก TITLE มีค่าน้ำหนักเป็น 3.8
แท็ก H1 มีค่าน้ำหนักเป็น 0.9
แท็ก H2 มีค่าน้ำหนักเป็น 0.5
แท็ก H3 มีค่าน้ำหนักเป็น 0.1
แท็ก CENTER มีค่าน้ำหนักเป็น 3.5
แท็ก B มีค่าน้ำหนักเป็น 3.9
แท็ก U มีค่าน้ำหนักเป็น 1.6
แท็ก STRONG มีค่าน้ำหนักเป็น 1.0

แท็ก BIG มีค่าน้ำหนักเป็น 0.6
แท็ก ALT มีค่าน้ำหนักเป็น 0.5
แท็ก A มีค่าน้ำหนักเป็น 1.7
แท็ก FONT +1 มีค่าน้ำหนักเป็น 1.6
แท็ก FONT +2 มีค่าน้ำหนักเป็น 2.0
แท็ก FONT >+2 มีค่าน้ำหนักเป็น 0.9

สรุปผลการทดลอง

จากตารางที่ 4-1 พบจำนวนคีย์เวิร์ดหลังแท็ก B มีจำนวนมากที่สุด เนื่องมาจากเว็บเพจส่วนใหญ่มีการเน้นข้อความ หรือคำด้วยแท็กนี้กันมาก การให้ค่าน้ำหนักกับแท็กนี้ควรมีค่ามากที่สุดเช่นกัน เพราะค่าน้ำหนักที่มากจะแสดงถึง โอกาสที่จะพบคีย์เวิร์ดหลังแท็กนั้น ๆ มีความเป็นไปได้สูง

การทดลองที่ 2

วัตถุประสงค์ในการทดลอง

เพื่อทดสอบค่าน้ำหนักที่หามาได้ กับกลุ่มตัวอย่างเว็บเพจที่ทำการสุ่มมาในแง่ของประสิทธิภาพและความถูกต้อง

ขั้นตอนการทดลอง

1. สุ่มเว็บเพจทั้งเว็บเพจภาษาไทย และเว็บเพจภาษาอังกฤษ
2. นำเว็บเพจที่หาได้ในข้อ 1 มาหาคีย์เวิร์ดโดยพิจารณาด้วยตา ซึ่งแต่ละเว็บเพจสามารถหาคีย์เวิร์ดออกมาได้มากกว่า 1 คำ
3. นำเว็บเพจที่ได้ในข้อ 1 มาหาคีย์เวิร์ด โดยใช้โปรแกรมพิจารณาหาตั้งแต่ 1 ถึง 10 คำเพื่อเปรียบเทียบ ระหว่างการหาคีย์เวิร์ดที่พิจารณาด้วยตา และการหาคีย์เวิร์ดที่พิจารณาด้วยโปรแกรม

ตัวอย่างการทดลองหาค่าประสิทธิภาพ และความถูกต้อง

เปิดเว็บเพจที่ได้ด้วยเว็บบราวเซอร์ เพื่อหาคีย์เวิร์ดโดยพิจารณาด้วยตา การพิจารณาจะอ่านเว็บเพจที่ได้ จากนั้นก็สรุปใจความสำคัญของเว็บเพจมาเป็นคำ ๆ จะได้กลุ่มคีย์เวิร์ดของเว็บเพจนั้น เช่น โรค หวัด ยา สุขภาพ และไวรัส

จากนั้นให้โปรแกรมหาคีย์เวิร์ดจำนวน 1 คำ เช่นได้คำว่า โรค

ประสิทธิภาพจะหาได้จาก (1/1) X 100 มีค่าเท่ากับ 100 %

ความถูกต้องจะหาได้จาก (1/จำนวนคีย์เวิร์ดที่หาได้โดยพิจารณาด้วยตา) X 100 มีค่า (1/5) X 100 เท่ากับ 20 %

เมื่อให้โปรแกรมหาคีย์เวิร์ดจำนวน 2 คำ เช่น ได้คำว่า โรค ระบาด

ประสิทธิภาพหาได้จาก (1/2) X 100 มีค่าเท่ากับ 50 %

ความถูกต้องหาได้จาก (1/5) X 100 มีค่าเท่ากับ 20 %

เมื่อให้โปรแกรมหาคีย์เวิร์ดจำนวน 3 คำ เช่น ได้คำว่า โรค ระบาด ไวรัส

ประสิทธิภาพหาได้จาก (2/3) X 100 มีค่าเท่ากับ 66.67 %

ความถูกต้องหาได้จาก (2/5) X 100 มีค่าเท่ากับ 40 %

ให้โปรแกรมหาจนครบทั้งสิ้น 10 คำ โดยพิจารณาหาค่าประสิทธิภาพ และความถูกต้องในลักษณะดังกล่าว

ผลการทดลอง

ทดลองกับเว็บเพจทั้งหมด 345 เว็บเพจ ค่าเฉลี่ยประสิทธิภาพของแต่ละเว็บเพจ หาได้จากผลรวมของค่าประสิทธิภาพแต่ละเว็บเพจ หารด้วยจำนวนเว็บเพจทั้งหมด และค่าเฉลี่ยความถูกต้องของแต่ละเว็บเพจ หาได้จากผลรวมค่าความถูกต้องแต่ละเว็บเพจหารด้วยจำนวนเว็บเพจทั้งหมด

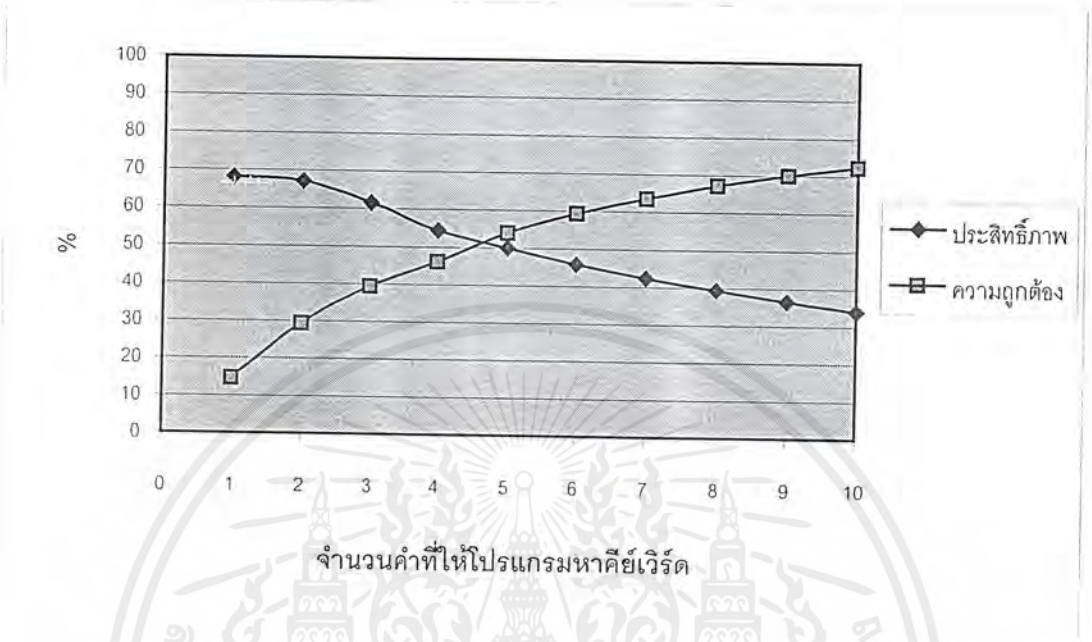
จำนวนคำ (คำ)	จำนวนคีย์เวิร์ดที่ได้ตรงกับคีย์เวิร์ดที่พิจารณาด้วยตา (คำ)	จำนวนคีย์เวิร์ดที่โปรแกรมหามาได้ทั้งหมด (คำ)	ค่าเฉลี่ยประสิทธิภาพของแต่ละเว็บเพจ (%)	ค่าเฉลี่ยความถูกต้องของแต่ละเว็บเพจ (%)
1	235	345	68.12	14.64
2	463	690	67.10	29.35
3	637	1035	61.55	39.35
4	750	1380	54.35	46.08
5	862	1725	49.97	53.94
6	949	2070	45.85	59.33
7	1024	2415	42.40	63.48
8	1086	2760	39.35	67.14
9	1136	3105	36.59	70.06
10	1174	3450	34.03	72.38

ตารางที่ 4-2 ประสิทธิภาพและความถูกต้องเมื่อให้โปรแกรมหา เปรียบเทียบกับการหาคีย์เวิร์ดโดยพิจารณาด้วยตา

วิเคราะห์ผลการทดลอง

เมื่อนำค่าเฉลี่ยประสิทธิภาพของแต่ละเว็บเพจ และค่าเฉลี่ยความถูกต้องของแต่ละเว็บเพจ มาเขียนกราฟ จะได้ดังรูปที่ 4-1

พิจารณาค่าเฉลี่ยประสิทธิภาพของแต่ละเว็บเพจ ทำให้เราทราบว่า เมื่อให้โปรแกรมหาคีย์เวิร์ดจำนวน 5 คำแล้วจะพบคีย์เวิร์ด 2.5 คำ ให้โปรแกรมหาคีย์เวิร์ด 10 คำ จะพบคีย์เวิร์ด 3.4 คำ เป็นต้น ซึ่งการให้โปรแกรมหาคีย์เวิร์ดจำนวนน้อย ๆ ย่อมมีประสิทธิภาพที่ดีกว่า



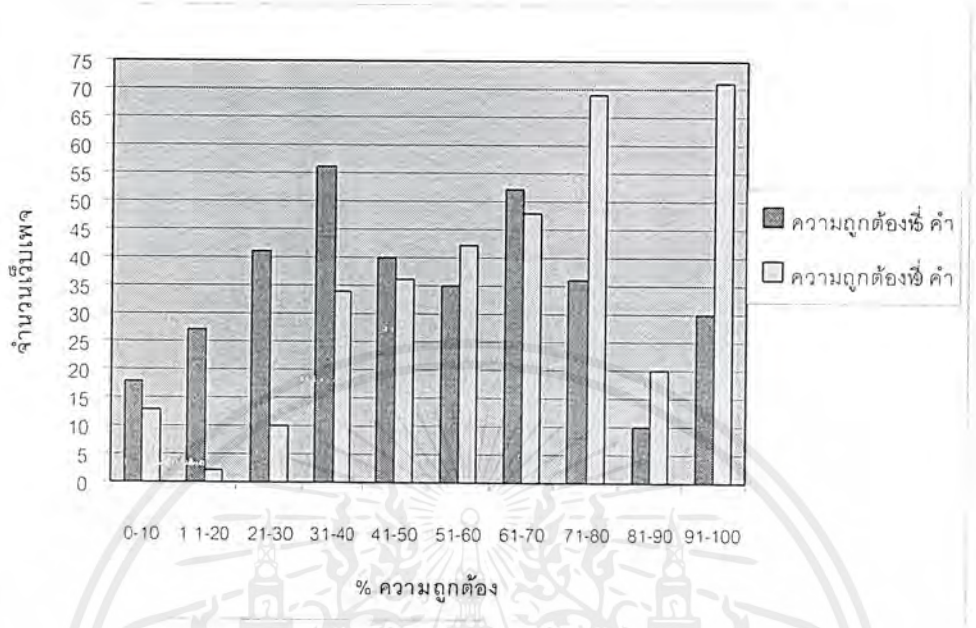
รูปที่ 4-1 ค่าเฉลี่ยประสิทธิภาพของแต่ละเว็บเพจ และค่าเฉลี่ยความถูกต้องของแต่ละเว็บเพจ เมื่อใช้โปรแกรมหาคีย์เวิร์ด

ลองพิจารณาค่าเฉลี่ยความถูกต้องของแต่ละเว็บเพจ ทำให้เราทราบอีกเช่นกันว่า เมื่อหาคีย์เวิร์ดโดยพิจารณาด้วยตาได้ 5 คำ จากนั้นให้โปรแกรมหาคีย์เวิร์ด 10 คำ จะพบคีย์เวิร์ด (72.38/100) X 5 มีค่าเท่ากับ 3.6 คำ จากคีย์เวิร์ดที่พิจารณาได้ด้วยตา 5 คำ ซึ่งถ้าให้โปรแกรมหาคีย์เวิร์ดเป็นจำนวนมากขึ้นโอกาสที่โปรแกรมจะพบคีย์เวิร์ดได้ครบทั้ง 5 ตัวสูงขึ้น

จากรูปที่ 4-1 การเลือกจำนวนคำที่ให้โปรแกรมหาคีย์เวิร์ด จะขึ้นอยู่กับผู้ใช้โดย ถ้าผู้ใช้ต้องประสิทธิภาพสูง ๆ ก็สามารถเลือกจำนวนคำที่ให้โปรแกรมหาคีย์เวิร์ด จำนวนน้อย ๆ และถ้าหากผู้ใช้ต้องการความถูกต้องในการหาสูง ๆ ก็สามารถเลือกจำนวนคำที่ให้โปรแกรมหาคีย์เวิร์ด จำนวนมาก ๆ และถ้าหากผู้ใช้ต้องการประสิทธิภาพ เท่ากับความถูกต้อง ก็สามารถเลือกได้ที่จุดตัดของกราฟในรูปที่ 4-1 นั่นคือ 4.67 คำหรือประมาณ 5 คำ

และแนวทางหนึ่ง ในการเลือกจำนวนคีย์เวิร์ดที่จะให้โปรแกรมพิจารณา คือพิจารณาที่ผลรวมของค่าเฉลี่ยประสิทธิภาพของแต่ละเว็บเพจ กับค่าเฉลี่ยความถูกต้องของแต่ละเว็บเพจ มีค่ามากที่สุด ในกราฟรูปที่ 4-1 คือ 9 คำ ค่าเฉลี่ยประสิทธิภาพได้ 36.59 % และความถูกต้อง 70.06 % ซึ่งผลรวมของค่าเฉลี่ยประสิทธิภาพของแต่ละเว็บเพจ กับค่าเฉลี่ยความถูกต้องของแต่ละเว็บเพจ มีค่ามากที่สุด โดยพบคีย์เวิร์ด 3.7 คำ และความถูกต้องสูงถึง 70.06 %

เมื่อนำจำนวนคีย์เวิร์ดที่ให้โปรแกรมหาที่ 5 และ 9 คำ มาพิจารณาการกระจายความถูกต้องจะได้กราฟแท่งในรูปที่ 4-2



รูปที่ 4-2 การกระจายความถูกต้องของแต่ละเว็บเพจ เมื่อใช้โปรแกรมหาคีย์เวิร์ด 5 และ 9 คำ

จากกราฟแท่งดังแสดงในรูปที่ 3-2 จากจำนวนเว็บเพจทั้งหมด 345 เว็บเพจ ค่าเปอร์เซ็นต์ความถูกต้องในการหาคีย์เวิร์ดของโปรแกรม เมื่อเทียบกับการหาคีย์เวิร์ดด้วยมนุษย์จะอยู่ในช่วง 0-10% ซึ่งเมื่อพิจารณาเว็บเพจที่อยู่ในช่วงนี้จะพบว่า

1) บางเว็บเพจ ไม่พบแท็กที่โปรแกรมใช้พิจารณาจำนวนหลายแท็ก
2) เว็บเพจมีการใช้รูปภาพจำนวนมาก โดยไม่กำหนดคำอธิบายรูปภาพภายในแท็ก IMG แอท-ทริบิวต์ ALT
3) ไม่พบคีย์เวิร์ดที่มนุษย์หาออกมา เช่น เรื่องเกี่ยวกับเทสนิส ซึ่งคีย์เวิร์ดควรเป็นคำว่า "กีฬา" แต่เว็บเพจนั้นไม่พบคำว่า "กีฬา" ตามที่มนุษย์หาออกมาได้

และเมื่อเปรียบเทียบความถูกต้องในการหาคีย์เวิร์ดของโปรแกรม เมื่อเทียบกับการหาคีย์เวิร์ดด้วยมนุษย์ จำนวนคีย์เวิร์ดที่ให้โปรแกรมหาเป็นจำนวน 5 และ 9 คำแล้ว จะเห็นว่า เมื่อให้โปรแกรมหาคีย์เวิร์ดจำนวน 9 คำ โปรแกรมสามารถหาความถูกต้องในการหาคีย์เวิร์ดของโปรแกรม เมื่อเทียบกับการหาคีย์เวิร์ดด้วยมนุษย์ ได้สูง 91-100% จำนวนมาก

สรุปผลการทดลอง

จากการทดลองรูป 4-1 ประสิทธิภาพในการหาคีย์เวิร์ดของโปรแกรม เมื่อเทียบกับการหาคีย์เวิร์ดด้วยมนุษย์ จะมีค่าลดลงเมื่อให้โปรแกรมหาจำนวนคีย์เวิร์ดมากขึ้น และในทางตรงกันข้าม ความถูกต้อง

ในการหาคีย์เวิร์ดของโปรแกรม เมื่อเทียบกับการหาคีย์เวิร์ดด้วยมนุษย์ จะมีค่าสูงขึ้นเมื่อให้โปรแกรมหาจำนวนคีย์เวิร์ดมากขึ้น

บทที่ 5

บทวิจารณ์ และสรุป

เสิร์ชเอ็นจิ้นเป็นบริการค้นหาข้อมูลบนอินเตอร์เน็ตที่เป็นที่นิยมอย่างแพร่หลายนั้น ยังไม่สามารถสนับสนุนการค้นหาเว็บเพจที่เป็นภาษาไทยได้ โดยบริการการค้นหาที่มีอยู่จะเป็นลักษณะการเก็บเว็บไซต์ภาษาไทยที่แยกออกเป็นหมวดหมู่ (Directory) ไว้ในฐานข้อมูล ซึ่งจะเก็บยูอาร์แอลของเว็บไซต์เหล่านั้นเอาไว้ วิทยานิพนธ์ฉบับนี้จึงมีแนวความคิดที่จะทำการหาคีย์เวิร์ดให้กับเว็บเพจภาษาไทย โดยนำหลักการและทฤษฎีที่ใช้กับภาษาอังกฤษมาประยุกต์ใช้งาน และทดสอบหาค่าน้ำหนักที่เหมาะสมในการกำหนดคีย์เวิร์ด เพื่อนำไปประยุกต์ใช้ในการสร้างเสิร์ชเอ็นจิ้นที่สนับสนุนการใช้ภาษาไทยต่อไป

เนื่องจากเนื้อหาในวิทยานิพนธ์ฉบับนี้ เป็นแนวคิดหนึ่งในการหาคีย์เวิร์ดของเว็บเพจ ซึ่งจะเป็นประโยชน์ต่อการสร้างเสิร์ชเอ็นจิ้น โดยหลักการของวิธีการกำหนดคีย์เวิร์ดนั้นจะอาศัยตำแหน่ง และความถี่ของคำที่ปรากฏในเว็บเพจ เป็นตัวกำหนดค่าน้ำหนักของคำแต่ละคำ คำที่มีค่าน้ำหนักสูงที่สุดก็จะมีโอกาสเป็นคีย์เวิร์ดได้มากที่สุด อย่างไรก็ตามแนวคิดนี้มีข้อดีเมื่อเทียบกับการหาคีย์เวิร์ดโดยใช้วิธีการนับความถี่ของคำที่พบในเว็บเพจเพียงอย่างเดียว คือประสิทธิภาพในการหาคีย์เวิร์ดที่ได้มีความถูกต้องมากขึ้น เพราะตำแหน่งของคำที่ปรากฏบนเว็บเพจนั้นมีความสำคัญต่อการเป็นคีย์เวิร์ดของเว็บเพจ โดยค่าน้ำหนักที่ใช้พิจารณาหาได้จากการทดลองสุ่มเว็บเพจตัวอย่างมาชุดหนึ่ง แล้วหาค่าน้ำหนักออกมา จากนั้นนำค่าน้ำหนักที่ได้นี้มาทดสอบกับกลุ่มเว็บเพจตัวอย่างนี้ เพื่อดูประสิทธิภาพของการกำหนดคีย์เวิร์ดที่ได้

ในการพิจารณาเว็บเพจบางเว็บเพจนั้น จะปรากฏแท็ก META ซึ่งภายในแท็ก META keyword และ description จะมีการเก็บคำที่เป็นคีย์เวิร์ดและคำที่เป็นคำอธิบายของเว็บเพจนั้นไว้ด้วย ดังนั้นคำเหล่านี้จึงเป็นคำที่น่าจะเป็นคีย์เวิร์ดของเว็บเพจมากที่สุด แต่เนื่องจากว่าในงานวิจัยนี้จะไม่พิจารณาในส่วนของแท็ก META เพราะเว็บเพจที่เป็นบทความภาษาไทยโดยส่วนใหญ่แล้วมักจะไม่มีการใส่ META ลงไปด้วย และผู้เขียนก็คิดว่าแท็ก META Keyword และ META Description นั้นจะมีการใส่ข้อความหรือคำที่เป็นคีย์เวิร์ดของเว็บเพจนั้นอยู่แล้ว จึงต้องการใช้วิธีการกำหนดคีย์เวิร์ดที่ออกแบบมาโดยดูจากโครงสร้างโดยรวมของเว็บเพจในส่วนอื่นแทน ซึ่งเราจะทำการหาค่าน้ำหนักมาใช้เป็นตัวกำหนดความสำคัญของคีย์เวิร์ดแต่ละตัว โดยค่าน้ำหนักได้มาจากการทดลอง และทดสอบกับเว็บเพจจำนวนหนึ่ง ซึ่งจากผลการทดลองในบทที่ 4 สามารถสรุปได้ว่า ประสิทธิภาพในการหาคีย์เวิร์ดของโปรแกรม เมื่อเทียบกับการหาคีย์เวิร์ดด้วยมนุษย์ จะมีค่าลดลงเมื่อให้โปรแกรมหาจำนวนคีย์เวิร์ดมากขึ้น และในทางตรงกันข้าม ความถูกต้องในการหาคีย์เวิร์ดของโปรแกรม เมื่อเทียบกับการหาคีย์เวิร์ดด้วยมนุษย์ จะมีค่าสูงขึ้นเมื่อให้โปรแกรมหาจำนวนคีย์เวิร์ดมากขึ้น

ข้อจำกัดในด้านการใช้งาน คือในส่วนของการตัดคำภาษาไทยนั้นไม่ได้ทำการเขียนโปรแกรมขึ้นมาเอง แต่เป็นการนำโปรแกรมการตัดคำจากเนคเทคมาใช้งาน ซึ่งโปรแกรมดังกล่าวยังไม่สามารถตัดคำในภาษาไทยที่เป็นคำสแลง คำประสมแล้วให้คำที่ตัดออกมาได้มีความหมายตามเนื้อความเดิม

ข้อจำกัดของการใช้งานอีกประการก็คือ เนื่องจากเว็บเพจในปัจจุบันมีลักษณะเป็นแบบไดนามิค และมีการใช้กราฟฟิค ทำให้ไม่สามารถใช้โปรแกรมที่มีอยู่ทำการกำหนดคีย์เวิร์ดให้กับเว็บเพจได้ เพราะตัวโปรแกรมจะทำงานในลักษณะของการสแกนข้อความที่เป็นเท็กซ์ที่อยู่ในซอร์สโค้ดไฟล์เอชทีเอ็มแอลเท่านั้นเช่น ในบางเว็บเพจมีการขึ้นหัวข้อซึ่งเป็นชื่อเรื่องของเว็บเพจด้วยข้อความที่เป็นกราฟฟิค จึงไม่สามารถหาคีย์เวิร์ดได้ในกรณีที่เว็บเพจนั้นมีการใช้ในลักษณะนี้

การทำงานของโปรแกรมยังไม่สามารถพิจารณาในทุก ๆ เฟรมของเว็บเพจ อีกทั้งโปรแกรมยังไม่สามารถพิจารณาเว็บเพจที่มีการลิงค์ไปยังเว็บเพจอื่น ๆได้ ข้อจำกัดข้อสุดท้ายก็คือ ถ้าหากเว็บเพจที่ในโปรแกรมหาคีย์เวิร์ดนั้นเป็นเว็บเพจที่มีเท็กไม่สมบูรณ์ หรือมีแท็กที่จะนำมาพิจารณาน้อยมาก ก็จะทำให้คีย์เวิร์ดที่หาออกมาได้ความถูกต้อง และประสิทธิภาพที่น้อยลง

แนวทางในการพัฒนาต่อไปในอนาคต

1. ทำการศึกษาในส่วนของการตัดคำ เพื่อหาแนวทางในการพัฒนาอัลกอริธึมในการตัดคำที่ดีกว่าเดิม
2. นำเรื่องการกำหนดคีย์เวิร์ดในเอกสารภาษาอังกฤษมาประยุกต์ใช้ร่วมด้วย เพื่อให้สามารถทำการกำหนดคีย์เวิร์ดที่เป็นทั้งภาษาไทย และภาษาอังกฤษได้
3. ปรับปรุงการทดลอง โดยการทดสอบกับเว็บเพจเป็นจำนวนมากขึ้นกว่าเดิมที่เคยทดสอบ เพื่อหาวิเคราะห์หาค่าน้ำหนักที่เหมาะสมที่จะมีประสิทธิภาพในการหาคีย์เวิร์ดที่ดีที่สุด

ภาคผนวก

สต็อปเวิร์ดในภาษาไทย

นะ อีก ต่าง ถ้า ใน ว่า และ จะ มี ได้ ของ ให้ เป็น นี้ ไม่ ความ การ เท่า ทุก แห่ง จาก ไป มา ทาง กล่าว โดย ซึ่ง ต้อง จำ ก็ แต่ ยัง ขึ้น อย่าง ทั้ง เพื่อ เข้า แล้ว ด้วย อยู่ นั้น หรือ เมื่อ ขณะ เปิด แห่ง ร่วม เพราะ ไร กว่า มาก ด้าน นอก ใหม่ ก่อน จึง หาก แก่ เช่น ทุก ไว้ บาง เพียง พร้อม ใต้ ดู อาจ หลาย ฯ ณ การ ครับ ค่ะ ฯลฯ คุณ เขา ฉัน เธอ นี่ นี้ ฉ. ปล. น. จ๊ะ จ้า เฮอ กับ เนี่ย ไหม อื่น ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙

สต็อปเวิร์ดในภาษาอังกฤษ

A

a, about, above, according, across, after, afterwards, again, against, albeit, all, almost, alone, along, already, also, although, always, among, amongst, am, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anywhere, apart, are, around, as, at, av

B

be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, below, beside, besides, between, beyond, both, but, by

C

can, cannot, canst, certain, cf, choose, contrariwise, cos, could, cu

D

day, do, does, doesn't, doing, dost, double, down, dual, during

E

each, either, else, elsewhere, enough, et, etc, even, ever, every, everybody, everyone, everything, everywhere, except, excepted,excepting, exception, exclude, excluding, exclusive

F

far, farther, farthest, few, ff, first, for, former, formerly, forth, forward, from, front, further, furthermore, furthest

G

get, go

H

had, halves, haedly, has, hast, hath, have, he, hence, henceforth, her, here, hereabouts, hereafter, hereby, herein, hereto, hereupon, hers, herself, him, himself, hindmost, his, hither, how, however, howsoever

I

i, ie, if, in, inasmuch, inc, include, included, including, indeed, indoors, inside, insomuch, instead, into, inward, is, it, its, itself

J

just

K

kind, kg, km

L

last, latter, latterly, less, lest, let, like, little, ltd

M

many, may, maybe, me, meantime, meanwhile, might, moreover, most, mostly, more, mr, mrs, ms, much, must, my, myself

N

namely, need, neither, never, nevertheless, next, no, nobody, none, nonetheless, noone, nope, nor, not, nothing, notwithstanding, now, nowadays, nowhere

O

of, off, often, ok, on, once one, only, onto, or, other, others, otherwise, ought, our, ours, ourselves, out, outside, over, own

P

per, perhaps, plenty, provide

Q

quite

R

rather, really, round

S

said, same, sang, save, saw, see, seeing, seem, seemed, seeming, seems, seen, seldom, selves, sent, several, shalt, she, should, shown, sideways, since, slept, slew, slung, slunk, smote, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, spake, spat, spoke, spoken, sprang, sprung, staves, still, such, supposing

T

than, that, the, thee, their, them, themselves, then, thence, thenceforth, there, thereabout, thereabouts, thereafter, thereby, therefore, therein, thereof, thereon, thereto, thereupon, these, they, this, those, thou, though, thrice, through, throughout, thru, thus, thy, thyself, till, to, together, too, toward, towards

U

ugh, unable, under, underneath, unless, unlike, until, up, upon, upward, us, use, used, using

V

very, via, vs

W

want, was, we, week, well, were, what, whatever, whatsoever, when, whence, whenever, whensoever, where, whereabouts, whereafter, whereas, whereat, whereby, wherefore, wherefrom, wherein, whereinto, whereof, whereon, wheresoever, whereto, whereunto, whereupon, wherever, wherewith, whether, whew, which, whichever, whichsoever, while, whilst, whither, who, whoever, whole, whom, whomever, whomsoever, whose, whosoever, why, will, wilt, with, within, without, worse, worst, would, wow

Y

ye, yet, year, yipee, you, your, yours, yourself, yourselves

บรรณานุกรม

[1] Gerald Salton, "*Automatic Text Processing : the transformation, analysis, and retrieval of information by computer*", Addison-Wesley, c1989.

[2] Peter D. Smith, "*An Introduction to Text Processing*", MIT Press, c1990.

[3] Timothy C. Craven, "*String Indexing*", Academic press, c1986.

[4] กิตติ ภักดีวัฒนะกุล, จำลอง ครูอุตสาหะ, "*Visual Basic 6.0 ฉบับโปรแกรมเมอร์*", บ. เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด

[5] สัจจะ จรัสรุ่งรวีวร ,"*คู่มือการสร้างแอปพลิเคชั่นด้วย Visual Basic 6.0 ฉบับสมบูรณ์*", สำนักพิมพ์ อินโฟเพรส, 1999

[6] ฉลองชัย จงประเสริฐพร, วรรวิภา ท่าพระนา, "*CGI/WEB Programming การพัฒนาโปรแกรมใช้งานบนเครือข่ายอินเตอร์เน็ต*", บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

[7] สัจจะ จรัสรุ่งรวีวร , "*Internet & Network Programming กับ VB 6.0 และ ASP*", สำนักพิมพ์ อินโฟ-เพรส

[8] http://www.abe.com.au/improved-search-engines-ranking.html, "Improved Search Engines Ranking".