ระบบกลั่นกรองสารสนเทศ

Information Filtering System

โดย

นาย จุฑากร ปิงคลาศัย

นางสาว อำภา ทนุถนอมราษฎร์

อาจารย์ที่ปรึกษา

ผ.ศ.ดร. เอื้อน ปิ่นเงิน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบกลั่นกรองสารสนเทศ

Information Filtering System

ผู้จัดทำ

1. นาย จุฑากร ปิงคลาศัย รหัสประจำตัว 40010~~2190~~140
2. นางสาว อำภา ทนุถนอมราษฎร์ รหัสประจำตัว 40010~~932~~1008

อาจารย์ที่ปรึกษา

(ผ.ศ.ดร. เอื้อน ปิ่นเงิน)

ระบบกลั่นกรองสารสนเทศ

นาย จุฑากร ปิงคลาศัย รหัส 40010140
นางสาว อำภา ทนุถนอมราษฎร์ รหัส 40011008

ผศ.ดร. เอื้อน ปิ่นเงิน
ปีการศึกษา 2543

บทคัดย่อ

ในยุคของข้อมูลข่าวสาร สิ่งสำคัญที่สุดคือข้อมูลที่ถูกต้องและมีประสิทธิภาพ จึงจำเป็นต้องมีการกลั่นกรองข้อมูลที่มาจากแหล่งต่างๆเหล่านั้น เพื่อให้ได้ข้อมูลตรงตามความต้องการ ระบบกลั่นกรองสารสนเทศเป็นระบบที่สร้างขึ้นเพื่อคัดเลือกเว็บไซต์ที่ตรงตามความต้องการของผู้ค้นหาจากแหล่งข้อมูลขนาดใหญ่ที่สุดในโลกคืออินเทอร์เน็ต โดยจะนำเอาเทคโนโลยีของ Search Engine มาช่วยในการค้นหาและแบ่งประเภทเว็บไซต์ ในการที่จะคัดเลือกเว็บไซต์ให้ตรงตามความต้องการนั้น ระบบจะนำเอาข้อมูลของผู้ใช้มาเปรียบเทียบกับประเภทของเว็บไซต์ โดยใช้วิธีการทาง Artificial Intelligent กลั่นกรองเอาเฉพาะเว็บไซต์ที่ตรงกับความสนใจของผู้ค้นหา เพื่อให้ได้เว็บไซต์ที่ตรงกับความต้องการมากขึ้น นอกจากนี้ระบบผู้ค้นหายังสามารถสอนระบบให้ทราบถึงความต้องการของผู้ค้นหาเพิ่มเติมขณะใช้งานระบบได้อีกด้วย เพื่อให้ระบบพัฒนาการคำนวณค่าความน่าสนใจของเว็บไซต์ให้เหมาะกับผู้ค้นหายิ่งขึ้น

Information Filtering System

Mr. Chuthakorn Pingkhalasay **No.40010140**

Miss Ampa ThanuThanomrad **No.40011008**

Asst.Prof.Dr. Ouen Pinngern

2000

Abstract

In this information world, the most important thing is information, which must be accurate and efficient. In order to obtain relevant information, we have to filter data which come from many sources. Information Filtering System was built for filtering the web site, which meet searcher's need, from the largest resource in the world, internet, by using search engine technology to search and classify web site. In order to filter only needed web site, system will use user profile to compare with web site's category by using artificial intelligent techniques to choose only web site that match with searcher's interest. Furthermore, searcher can teach system about searcher's interest while using the system to improve the rating procedure.

กิตติกรรมประกาศ

ระบบกลั่นกรองสารสนเทศนี้ได้รับความช่วยเหลือจากหลาย ๆ ท่านด้วย ขอขอบพระคุณ อาจารย์ที่ปรึกษา ผ.ศ.ดร.เอื้อน ปิ่นเงิน ที่คอยดูแลเอาใจใส่และให้ข้อคิดเห็นต่าง ๆ มากมาย ขอขอบพระคุณ อาจารย์ ธนา หงสุวรรณ และ อาจารย์ อัครเดช ที่ได้แนะนำถึงข้อด้อยของระบบในการนำเสนอโครงงานในภาคเรียนที่ 1 ปีการศึกษา 2543

ทั้งนี้รวมไปถึงเพื่อน ๆ ที่คอยช่วยเหลือในด้านต่าง ๆ รุ่นพี่ที่คอยให้คำปรึกษาในทุก ๆ ด้าน น้อง ทุก ๆ คนที่ช่วยเป็นกำลังใจให้เสมอมา ผู้เขียนพร้อมสำนักพิมพ์ที่มีการจัดพิมพ์หนังสืออ้างอิง และ ผู้สร้าง Web Site หลาย ๆ Web Site ที่เป็นแหล่งข้อมูลให้ค้นคว้าข้อมูลมาประกอบการทำงาน และที่สำคัญ Search Engine ทั้ง 2 ตัวที่ใช้ประกอบระบบ คือ MetaCrawler และ Yahoo เป็นอย่างมาก หากขาดความช่วยเหลือจากทุก ๆ ด้าน ระบบนี้คงจะไม่สามารถสำเร็จลงได้ คณะผู้จัดทำระบบขอขอบคุณมา ณ ที่นี้

คณะผู้จัดทำ

สารบัญ

สารบัญรูปภาพ

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ที่มา

ข้อมูลข่าวสารเป็นสิ่งจำเป็นมากต่อสังคมในปัจจุบัน ทั้งต่อการพัฒนาด้านต่างๆ เพื่อความรวดเร็วทันต่อเหตุการณ์ของการดำเนินงานในองค์กรต่างๆ และเพื่อนตอบสนองความต้องการของผู้คน ข้อมูลที่ใช้จำเป็นต้องถูกต้องแม่นยำ รวดเร็ว และตรงกับงานที่ใช้ ยิ่งแหล่งข้อมูลอันมหาศาลอย่างอินเทอร์เน็ตซึ่งเป็นแหล่งข้อมูลที่มาจากแหล่งต่างๆทั่วทุกมุมโลก ปัจจุบันทุกคนยิ่งให้ความสำคัญกับตรงจุดนี้มากขึ้นเรื่อย ๆ นับวันยิ่งจะเพิ่มข้อมูลข่าวสารแก่แหล่งข้อมูลอันมหาศาลนี้ ทำให้ข้อมูลข่าวสารในอินเทอร์เน็ตเพิ่มขึ้นเป็นทวีคูณ การที่จะค้นหาข้อมูลให้ได้ตรงตามความต้องการจึงทำได้ยาก จำเป็นต้องมีผู้ช่วยเพื่อให้ค้นหาข้อมูลได้รวดเร็ว แม่นยำ และตรงตามความต้องการ

ในปัจจุบันเราใช้ Search Engine เป็นตัวช่วยในการค้นหาข้อมูลเว็บไซต์ในอินเทอร์เน็ต ซึ่งเป็นการกรองข้อมูลที่ต้องการในระดับหนึ่งจากข้อมูลจากแหล่งข้อมูลที่ใหญ่ที่สุดอย่างอินเทอร์เน็ตนี้ ทั้งนี้เทคโนโลยีของ Search Engine คือเป็นการรวบรวมลิงค์ และแบ่งแยกประเภทลิงค์โดยคำที่เกี่ยวข้องกับลิงค์นั้น โดยพิจารณาจากเนื้อหาของข้อมูลในลิงค์นั้น หรือพิจารณาจากตำแหน่งของคำในเนื้อหาของข้อมูลในลิงค์นั้น (รายละเอียดการทำงานของ Search Engine กล่าวไว้ในบทที่ 3) ซึ่งประเภทของลิงค์นั้นอาจไม่เป็นที่สนใจของผู้ค้นหาก็ได้ และ ลิงค์ผลลัพธ์ที่ได้จากการค้นหาก็มีจำนวนมากมายมหาศาล ขาดความน่าสนใจไป

ระบบกลั่นกรองสารสนเทศจึงถูกพัฒนาขึ้นมาเพื่อช่วยในการค้นหาข้อมูลเว็บไซต์จากแหล่งข้อมูลอันมหาศาลนี้ให้สามารถค้นหาข้อมูลเว็บไซต์ได้อย่างมีประสิทธิภาพตรงตามความต้องการของผู้ค้นหามากขึ้น โดยการทำงานคือ จะเข้าไปค้นหาข้อมูลเว็บไซต์ที่ต้องการแล้วทำการแบ่งประเภทเว็บไซต์ที่ได้นำมาเปรียบเทียบกับข้อมูลความสนใจของผู้ใช้แต่ละคน แสดงผลลัพธ์ของลิงก์โดยเรียงลำดับลิงก์ที่ผู้ใช้น่าจะสนใจมากที่สุดตามลำดับ เพื่อให้ได้ผลลัพธ์ที่ตรงตามความต้องการของผู้ใช้แต่ละคนมากขึ้น

1.2 วัตถุประสงค์ของโครงงาน

- เพื่อศึกษาการติดต่อสื่อสารทางอินเทอร์เน็ตและข้อมูลสารสนเทศบนอินเทอร์เน็ต
- ศึกษาการทำงานของ Search Engine เพื่อนำมาประยุกต์ใช้กับระบบ
- ศึกษาเกี่ยวกับข้อมูลส่วนบุคคล เพื่อนำมาใช้กรองข้อมูล
- สร้างระบบกลั่นกรองสารสนเทศที่สอดคล้องกับข้อมูลส่วนบุคคล

1.3 ขอบเขตของโครงงาน

โครงงานนี้วางขอบเขตไว้ที่การกลั่นกรองข้อมูลเว็บไซต์ทุกชนิดบนอินเทอร์เน็ต โดยยึดหลักการแบ่งปันข้อมูลนั่นคือเลือกใช้ฐานข้อมูลเว็บไซต์จาก Search Engine ที่มีอยู่แล้ว การพัฒนาจะเน้นใช้ภาษาจาวาที่สามารถทำงานได้กับทุกระบบปฏิบัติการ และมีโครงสร้างในลักษณะ OOP(Object-Oriented Program) เพื่อให้แบ่งการทำงานออกเป็นสัดส่วนอย่างชัดเจนและง่ายต่อการนำไปพัฒนาต่อ นอกจากนี้โครงงานยังเน้นถึงการเรียนรู้ความสนใจต่อเว็บไซต์ของผู้ใช้ระหว่างการใช้งานอีกด้วย

1.4 ผลที่คาดว่าจะได้รับ

โครงงานนี้สร้างขึ้นเพื่อให้การค้นหาเว็บไซต์บนอินเทอร์เน็ตเป็นไปอย่างมีประสิทธิภาพและตรงกับความต้องการของผู้ค้นหา ผลลัพธ์เว็บไซต์ที่ได้จะมีความเกี่ยวข้องกับคีย์เวิร์ดที่ผู้ค้นหาป้อนให้และตรงกับความสนใจของผู้ค้นหา โดยจะมีค่าความน่าสนใจของเว็บไซต์แสดงออกมาให้เห็นด้วย รวมไปถึงการแบ่งประเภทเว็บไซต์เพื่อให้ผู้ค้นหาทราบประเภทของเว็บไซต์นั้น เป็นการระบุลักษณะและเนื้อหาของเว็บไซต์ไปด้วย

นอกจากนี้จะมีการเก็บข้อมูลของผู้ค้นหาและเรียนรู้ความสนใจของผู้ค้นหาในระหว่างที่ใช้งานอยู่ด้วย เพื่อนำไปปรับปรุงค่าต่าง ๆ ให้ตรงกับผู้ค้นหาคนนั้น ๆ มากยิ่งขึ้น

1.5 รายละเอียดในวิทยานิพนธ์

ในบทที่ 2 จะกล่าวถึงตัวระบบ จุดประสงค์และเป้าหมาย การติดตั้ง และการใช้งานระบบกลั่นกรองข้อมูลสารสนเทศ เพื่อให้เข้าใจก่อนว่าระบบที่เราต้องการเป็นเช่นใด ต้องการปัจจัยใดบ้างเพื่อให้ได้ระบบที่ต้องการ

สำหรับทฤษฎีและหลักการที่นำมาใช้ในระบบนี้ ได้เรียบเรียงไว้ในบทที่ 3 ซึ่งเป็นทฤษฎีและหลักการทำงานของ Search Engine รายละเอียดการติดต่อกับ Search Engine เหล่านั้นของระบบ ด้วยโพรโตคอลเอชทีทีพี (HyperText Transfer Protocol; HTTP) ลักษณะและการทำงานของ Neural Network ซึ่งเป็นวิธีการหนึ่งที่ใช้ในการพิจารณาค่าความสนใจของผู้ใช้ โดยในระบบนี้จะนำ Backpropagation Neural Network มาประยุกต์ใช้

ในส่วนโครงสร้างและการทำงานของระบบ ซึ่งเป็นรายละเอียดการทำงาน ลักษณะโครงสร้างการทำงานภายในรวมถึงการติดต่อภายนอกเป็นอย่างไรนั้นจะกล่าวไว้ในบทที่ 4

สำหรับบทที่ 5 นั้นจะเป็นผลการทดลองและประเมินผลการทำงานของระบบ

และสุดท้ายในบทที่ 6 เป็นบทสรุปทั้งหมดของระบบและข้อเสนอแนะสำหรับนำไปประยุกต์หรือพัฒนาต่อไป

บทที่ 2
ทฤษฎีและหลักการ

2.1 User Profile

User profile เป็นข้อมูลต่างๆของผู้ใช้ อาจหมายถึงข้อมูลส่วนบุคคลที่แสดงความเป็นบุคคลนั้น ๆ หรือ ข้อมูลเฉพาะที่ผู้ใช้สนใจ โดยสามารถแบ่ง user profile ออกได้เป็น 2 ประเภท คือ

2.1.1 Simple Profiles

Simple Profiles มันประกอบด้วยเทอมที่มีค่าน้ำหนัก (weight) ต่างๆกันในบางระบบจะมีรูปแบบในการถามข้อมูลและเรียบเรียง profiles ที่น่าสนใจ และมีการปรับปรุง แก้ไขอยู่เป็นช่วงๆไป ในความหมายเดียวกัน profiles ที่ทำงานเป็น query มาตรฐานที่มีการเปลี่ยนแปลงแก้ไขอยู่เรื่อยๆนี้ เรียกว่า *routing query* ในทางกลับกัน *ad hoc query* คือ profiles ที่มีรูปแบบการถามเพียงครั้งเดียวในตอนแรก

Simple user profiles จะง่ายต่อการประสานกับข้อมูลในฐานข้อมูล อย่างไรก็ตามก็มีข้อจำกัดในการที่จะแสดงลักษณะเอกสารที่แต่ละคนต้องการใช้ เพราะถูกควบคุมด้วย key word, หรือ key phrases ที่มีอยู่ในเอกสาร

2.1.2 Extended Profiles

ข้อมูลที่อยู่ใน profile นี้ มีความสัมพันธ์ในตัวบุคคลมากกว่าข้อมูลเฉพาะที่ต้องการ ยิ่งไปกว่านั้นมันจะมุ่งประเด็นไปในข้อมูลที่ผู้ใช้ต้องการ โดยดูจากข้อมูลพื้นเพของผู้ใช้ เพื่อช่วยในการตัดสินใจเพิ่มขึ้นในการเข้าใช้ข้อมูล (retrieval) นั่นคือจะใช้เป็นข้อมูลชนิดหนึ่งในการดึงข้อมูล ซึ่ง profiles ชนิดนี้ทำได้ได้ข้อมูลเหล่านี้ด้วย

- Language Capability

  ทำให้สามารถเลือกเอกสารที่ใช้ภาษาเฉพาะภาษาที่ต้องการได้

- Reading habits

  เรียนรู้ลักษณะนิสัยของผู้ใช้

- Specific preferences

  Profile ชนิดนี้ไม่ได้นำมาใช้ค้นหาเอกสาร โดยตรงแต่นำมา retrieve กลุ่มของสิ่งที่อยู่ในความต้องการหรือเกี่ยวข้อง ที่ผู้ใช้สนใจและเป็นไปได้ที่จะกำจัดเอกสารที่ไม่ต้องการออกไป

จรรยาบรรณของการใช้ User Profile

ข้อมูลส่วนบุคคล หรือ User profile (แม้แต่ในรูปแบบที่ง่ายที่สุด) ก็ให้ model ของผู้ใช้ในรูปแบบที่ต่อขยายออกไป model ก็จะยิ่งมีรายละเอียดมาก การพัฒนาและการใช้ profile จะเพิ่มขึ้น 2 หัวข้อที่เกี่ยวกับจรรยาบรรณ

ข้อแรก : การใช้ profile เป็นการจำกัดผลลัพธ์ที่ให้แก่ผู้ค้นหา

สิ่งนี้เกิดจากการสันนิษฐานว่า การปรับเปลี่ยนตรงตามใจผู้ใช้ที่เขาต้องการความมีเหตุมีผลของสันนิษฐาน คือ สาระของเรื่องราวที่พิจารณาถูกต้องตามกฎหมาย อย่างไรก็ตาม จากตัวอย่าง information retrieval system ในปัจจุบันสร้างสันนิษฐานเรื่อง similarity ทั้งกลุ่มผู้ใช้ การกำหนดของเขตของผลลัพธ์ ไม่ว่าจะเป็นจำนวนของข้อมูล retrieval system ประกาศแน่นอนว่าข้อมูลที่ไม่แสดงออกไปนั้น ไม่เป็นที่ต้องการของผู้ใช้ การเรียงลำดับข้อมูลด้วยตัวความสำคัญ similarity ใน query การใช้ profile มาเป็นข้อจำกัดของขบวนการทำงาน จะทำให้มีการทำงานที่ซับซ้อนขึ้น แต่จะไม่เปลี่ยนแปลงลักษณะพื้นฐาน มันยังคงเลือกข้อมูลและแสดงอันที่ดีที่สุดแก่ผู้ใช้

ข้อที่สอง : การใช้ profile เป็นการบุกรุกความเป็นส่วนตัวของผู้ใช้ ในส่วนของ extended profile

คนเรามักจะไม่แสดง หรืออธิบายความเป็นตัวเอง ก็หมายความว่า profile เป็นสิ่งที่แสดงถึงข้อมูลนิสัยและความชอบของผู้ใช้ มันจะสามารถเก็บข้อมูลของพฤติกรรมของผู้ใช้ที่อาจจะลืมไปแล้วก็ได้

2.2 Search Engine

2.2.1 ความหมายของ Search Engine

Search Engine เป็นเครื่องมือที่ใช้ในการค้นหาข้อมูลที่ต้องการใน World Wide Web เนื่องจากข้อมูลต่าง ๆ ในอินเทอร์เน็ตนั้นมีอยู่มากมายมหาศาล จึงจำเป็นต้องมีเครื่องมือที่ช่วยในการค้นหาข้อมูลเพื่อให้ได้ข้อมูลตรงกับความต้องการของผู้ค้นหา

ตัวค้นหาข้อมูล (Search Engine) สามารถแบ่งออกเป็น 2 ประเภทใหญ่ ๆ ได้แก่

1. Search Engine เป็นตัวค้นหาที่คอยรวบรวมเว็บไซต์ต่าง ๆ โดยการส่งโปรแกรมพิเศษออกไปรวบรวมข้อมูลของแต่ละเว็บไซต์มาเก็บไว้ในฐานข้อมูล จนกลายเป็นฐานข้อมูลขนาดใหญ่ เมื่อต้องการค้นหา จะรับ keyword จากผู้ค้นหาแล้วใช้กลไกค้นหาของแต่ละ Search Engine เพื่อค้นหาข้อมูลในฐานข้อมูลตาม keyword ต่อไป ตัวอย่างของ Search Engine ประเภทนี้ เช่น Inktomi , Altavista , Hotbot เป็นต้น
2. Directory จะมีคนเป็นผู้ค้นหาข้อมูลของเว็บไซต์และแบ่งเว็บไซต์ออกเป็นหมวดหมู่ ทำให้ฐานข้อมูลมีขนาดเล็ก แต่จะมีคุณภาพเนื่องจากมีการคัดเลือกและจัดหมวดหมู่อย่างมีระเบียบ การค้นหาเว็บไซต์จะทำได้ง่ายและแม่นยำตรงกับความต้องการของผู้ค้นหา ตัวอย่างของ Search Engine ประเภทนี้ เช่น Yahoo! , Netscape Open Directory Project , About.com เป็นต้น

บาง Search Engine นำข้อดีของทั้ง 2 แบบมาผสมผสานกัน เช่น Yahoo! เป็น Search Engine แบบ Directory แต่นำกลไก Search Engine แบบ Inktomi มาประยุกต์ใช้ หรือ Hotbot นำเอาเทคโนโลยี Open Directory Project ซึ่งเป็นฐานข้อมูลไดเรกทอรีเข้ามาผสม

2.2.2 องค์ประกอบของ Search Engine

2.2.2.1 Web robot

Web Robot คือโปรแกรมที่จะเข้าไปเก็บข้อมูลใน World Wide Web โดยการเดินทางผ่านลิงค์ต่อลิงค์ของแต่ละหน้าในเว็บไซต์และรวบรวมข้อมูลของเว็บไซต์หน้านั้น ๆ กลับมาเก็บไว้ที่ฐานข้อมูลผ่านโปรโตคอล HTTP

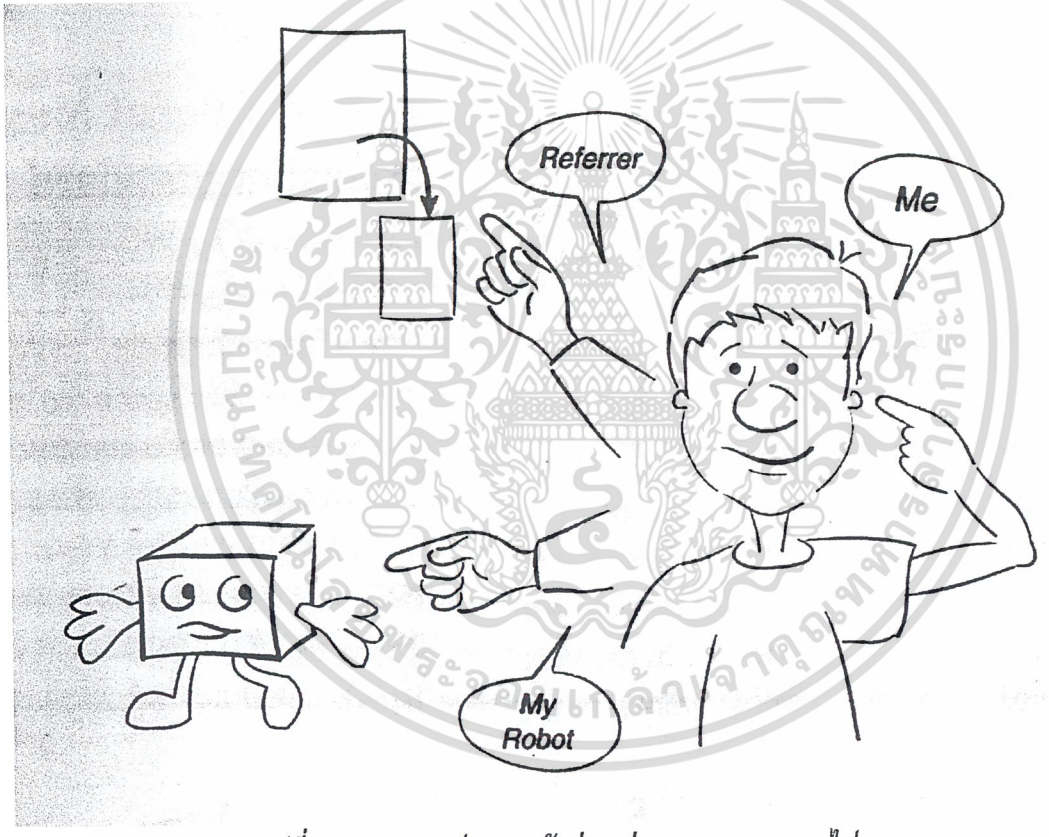
Web Robot มีจุดประสงค์หลัก ๆ อยู่ 3 ประการ คือ

1. ค้นหาข้อมูลในเว็บเพื่อนำมาเก็บและจัดเรียงในฐานข้อมูล
2. ตรวจลิงค์ที่มีไว้เชื่อมต่อกับเว็บไซต์อื่นเพื่อแก้ไขเมื่อเกิด dead link
3. สำรองข้อมูลเพื่อให้สามารถทำงานต่อได้ในกรณีที่เกิดระบบล่มและช่วยในการดาวน์โหลดไฟล์ เช่นการสำรอง FTP เว็บไซต์ ทำให้สามารถเลือกดาวน์โหลดได้จากหลาย ๆ แหล่ง

Robot Exclusion Standard เป็นมาตรฐานสำหรับเซิร์ฟเวอร์ที่จะระบุคุณสมบัติของ Web Robot ที่สามารถเข้ามาเก็บข้อมูลภายในเซิร์ฟเวอร์นั้นได้ โดยการสร้างไฟล์ Robot Exclusion เก็บไว้ใน root ชื่อ robots.txt ซึ่งไฟล์นี้จะต้องสามารถอ่านได้ด้วยโปรโตคอล HTTP

กฎ 4 ข้อสำหรับ Web Robot

1. ระบุตัวเองให้ผู้ดูแลเว็บของเซิร์ฟเวอร์ทราบ โดยสิ่งที่จะต้องระบุมี 3 อย่าง คือ ชื่อ Web Robot , ผู้ดูแล Web Robot และ ลิงค์ที่อ้างอิง ดังแสดงประกอบในรูปที่ 2.1
2. ทำตาม Robot Exclusion ก่อนที่จะเข้าไปเก็บข้อมูลในเซิร์ฟเวอร์ได้จะต้องอ่านไฟล์ Robot Exclusion เสียก่อน และปฏิบัติข้อกำหนดในไฟล์นั้น
3. ไม่ใช้ทรัพยากรของเซิร์ฟเวอร์มากเกินความจำเป็น โดยการกำหนดและเก็บข้อมูลเฉพาะสิ่งที่ต้องการจริง ๆ
4. เมื่อพบข้อผิดพลาดควรแจ้งให้ผู้ดูแลเว็บของเซิร์ฟเวอร์นั้นทราบ



รูปที่ 2.1 แสดงการประกาศตัวก่อนส่ง Web Robot ออกไป

ข้อควรปฏิบัติของผู้ดูแล Web Robot

1. ประกาศให้ปลายทางทราบก่อนที่จะส่ง Web Robot ออกไปเก็บข้อมูล
2. ทดสอบ Web Robot กับเซิร์ฟเวอร์ภายในเสียก่อน
3. ดูแลและตรวจสอบการทำงานของ Web Robot
4. ติดต่อกับผู้ดูแลเว็บอื่น ๆ
5. เคารพและปฏิบัติตามข้อกำหนดของผู้ดูแลเว็บ

6. แลกเปลี่ยนข้อมูลที่ได้กับผู้อื่น

2.2.2.2 Database

Database เป็นฐานข้อมูลที่ใช้เก็บข้อมูลที่ได้จากการรวบรวมของ Web Robot โดยจะจัดประเภทของเว็บไซต์แล้วจัดเก็บอย่างเป็นหมวดหมู่ เพื่อสะดวกต่อการค้นหาและการคัดเลือกเว็บไซต์ให้ตรงกับความต้องการของผู้ใช้

2.2.2.3 Search Mechanism

Search Mechanism เป็นกลไกหลักที่ใช้ในการค้นหาและกลั่นกรองข้อมูลที่เก็บไว้ในฐานข้อมูล สำหรับระบบกลั่นกรองสารสนเทศนี้จะใช้เทคโนโลยีของ Neural Network และ Heuristic Function มาใช้ผสมผสานกันเพื่อกลั่นกรองให้ได้ข้อมูลที่ถูกต้องแม่นยำและตรงกับความต้องการมากที่สุด สำหรับเทคโนโลยีเหล่านี้จะกล่าวถึงรายละเอียดในภายหลัง

2.2.3 ปัจจัยที่มีผลต่อการค้นหาเว็บไซต์

1. Search Engine Search Engine ที่ดีต้องมีการจัดระบบอย่างเป็นหมวดหมู่และมีข้อมูลเว็บไซต์อยู่เป็นจำนวนมาก อีกทั้งควรจะสามารถเรียงลำดับผลลัพธ์การค้นหาที่ได้ตามลำดับความสัมพันธ์ของเว็บไซต์นั้น ๆ กับ keyword ของผู้ค้นหา
2. ผู้สร้างเว็บไซต์ ควรมีการวาง Keyword ต่าง ๆ ให้ดี โดยเฉพาะในส่วนหัวเรื่องของเว็บไซต์ (Title) ซึ่งมักจะได้รับค่าความสัมพันธ์กับ Keyword ของผู้ค้นหาอย่างมาก และควรมี Data เพื่อความสะดวกในการเก็บรวบรวมข้อมูลของตัวค้นหาข้อมูล
3. ผู้ค้นหา ต้องมีทักษะในการค้นหา โดยทราบว่า Keyword ที่ตรงกับสิ่งที่ต้องการค้นหานั้นมีอะไรบ้าง แล้วอาจนำ Keyword หลาย ๆ คำมาผสมกันด้วย boolean เช่น and , or , not , near , () , “” เป็นต้น

ปัจจัยที่มีความสำคัญที่สุดคือตัวผู้ค้นหาเอง เพราะตั้งแต่เริ่มค้นหาจนสิ้นสุดการค้นหานั้นล้วนแล้วแต่ถูกกำหนดโดยผู้ค้นหาทั้งสิ้น นั่นคือการค้นหานั้นจะค้นหาจาก Keyword ที่ผู้ค้นหาใส่ ซึ่งจะค้นหาได้ตรงและแม่นยำแค่ไหนนั้นขึ้นอยู่กับ Keyword นี้เป็นหลัก และสุดท้ายเมื่อได้ผลลัพธ์ของการค้นหาออกมาแล้วผู้ที่จะตัดสินผลลัพธ์ที่ได้ยังคงเป็นผู้ค้นหาอีกนั่นเอง เพราะฉะนั้น Information Filtering System นี้จึงเน้นการพัฒนาด้านการช่วยเหลือผู้ค้นหาเป็นหลัก เพื่อให้ได้ผลลัพธ์ของการค้นหาตรงตามความต้องการ

2.2.4 องค์ประกอบเสริมสำหรับ Search Engine

องค์ประกอบเสริมเหล่านี้มีเพื่อช่วยให้ผู้ค้นหาสามารถเลือกเว็บไซต์ได้ตรงกับความต้องการมากยิ่งขึ้น เช่น

- Direct Hit จะมีการแสดงเว็บไซต์ที่ได้เรียงตามลำดับยอดผู้เข้าชมเว็บไซต์นั้น เพราะเว็บไซต์ที่มีผู้เข้าชมมากย่อมหมายความว่าเว็บไซต์นั้นมีคุณภาพและน่าสนใจ

- Realnames ใช้ในกรณีที่มีปัญหาในเรื่อง Domain Names คือ Domain Name ที่ต้องการใช้นั้นถูกใช้แล้วโดยผู้อื่น ทำให้ได้ชื่อที่ไม่สามารถสื่อถึงเว็บไซต์อย่างชัดเจน จึงมีการใช้ Internet Keyword โดยผ่านทาง Realnames.com เพื่อให้ได้ Keyword ซึ่งสื่อความหมายได้ตรงกับเว็บไซต์ ซึ่งในการค้นหาของบาง Search Engine จะมาค้นหา Keyword ใน realnames นี้ด้วย
- Alexa จะช่วยหาเว็บไซต์ที่มีลักษณะคล้ายคลึงกับเว็บไซต์ที่กำลังดูอยู่ในขณะนั้น

2.2.5 การเข้าดึงข้อมูลเว็บไซต์ของ Search Engine

สำหรับเว็บไซต์ที่ใช้ Web Bot เข้าไปดึงข้อมูลของเว็บไซต์นั้น ข้อมูลที่จะดึงย่อมต่างกันไปตามแต่ Web Bot นั้นๆ ตารางที่ 2.1 นี้แสดงถึงเทคนิคการเข้าดึงข้อมูลของ Search Engine

Crawling	Yes	No	Notes
Deep Crawl	All but...	Excite	
Instant Indexing	AltaVista (pages appear within days)	Others	
Frames Support	All but...	FAST	Reconfirming FAST as NO
Image Maps	AltaVista, NLight	Excite, FAST, Google, Inktomi	
robots.txt	All	n/a	
Meta Robots Tag	All but Excite	n/a	
Link Popularity Helps Deep Crawl	All	n/a	
Learns Frequency	AltaVista, Inktomi,	Excite, FAST, Google, NLight	
Paid Inclusion	Inktomi	Others	

ตารางที่ 2.1 แสดงเทคนิคการเข้าดึงข้อมูลของ Search Engine

2.2.6 การทำดัชนีเว็บไซต์ของ Search Engine

เมื่อ Web Bot ส่งข้อมูลเว็บไซต์มาที่เซิร์ฟเวอร์แล้วจะต้องมีการทำดัชนีสำหรับเว็บไซต์ เพื่อแยกประเภทและจัดเก็บลงฐานข้อมูล การทำดัชนีของแต่ละ Search Engine มีลักษณะดังตารางที่ 2.2 นี้

Indexing	Yes	No	Notes
Full Body Text	All	n/a	Some stop words may not be indexed
Stop Words	AltaVista, Excite, Inktomi, Google	FAST, NLight	
Meta Description	All but...	FAST, Google, NLight	
Meta Keywords	All but...	Excite, FAST, Google, NLight	

ALT text	AltaVista, Google	Excite, FAST, Inktomi, NLight	
Comments	Inktomi	Others	

ตารางที่ 2.2 แสดงการทำดัชนีเว็บไซต์ของแต่ละ Search Engine

2.2.7 การจัดลำดับเว็บไซต์ของ Search Engine

เมื่อเราใส่คำที่เราสนใจให้กับ Search Engine มันจะนำคำนั้นไปตรวจสอบกับเว็บไซต์หลายล้านเว็บไซต์ในฐานข้อมูลและให้คะแนนความน่าสนใจกับเว็บไซต์นั้น แล้วจึงนำเว็บไซต์ที่ตรงและสอดคล้องกับคำนั้นมากที่สุดมาแสดงผล แต่ผลลัพธ์ที่ได้จะมีประสิทธิภาพมากหรือน้อยนั้นขึ้นอยู่กับสูตรการให้คะแนนความน่าสนใจของแต่ละ Search Engine เอง คล้ายกับร้านอาหารที่มีสูตรอาหารเฉพาะตัว โดยทั่วไป Search Engine จะมีหลักการให้คะแนนความน่าสนใจอยู่ 2 หลักการใหญ่ ๆ ดังนี้

1. ตำแหน่งที่คำนั้นปรากฏในเว็บไซต์ แต่ละตำแหน่งมีค่าความสำคัญแตกต่างกันไป ตำแหน่งที่มีค่าความสำคัญมากที่สุดคือ Title ของเว็บไซต์ รองลงมาคือหัวข้อหรือย่อหน้าแรก ๆ ของเว็บไซต์นั้น หากมีคำที่ให้ปรากฏอยู่ในตำแหน่งเหล่านี้ก็จะได้ค่าความน่าสนใจมากขึ้น นอกจากนี้ยังมีอีกตำแหน่งหนึ่งที่ผู้สร้างเว็บมักจะละเลยไปคือ Meta Tag ซึ่งเป็นส่วนที่ Web Robot จะอ่านเพื่อนำไปแบ่งประเภทเว็บไซต์รวมถึงให้ค่าความน่าสนใจด้วย
2. จำนวนครั้งที่คำนั้นปรากฏในเว็บไซต์ ยิ่งมีคำนั้นปรากฏในเว็บไซต์มาก ก็มีความเป็นไปได้ที่เว็บไซต์นั้นจะเกี่ยวข้องกับคำนั้นมากยิ่งขึ้นด้วย

นอกจาก 2 หลักการนี้แล้วแต่ละ Search Engine ก็จะมีสูตรการให้ค่าความน่าสนใจเพิ่มเติมแตกต่างกันไปตัวอย่างเช่น

- Excite.com มีการคำนวณค่า "Link popularity" ซึ่งเป็นค่าที่แสดงว่ามีเว็บไซต์อื่นๆเชื่อมมายังเว็บไซต์นี้มากเท่าใด ยิ่งค่านี้มาก นั่นหมายถึงเว็บไซต์นี้ได้รับการยอมรับจากเว็บไซต์อื่น ๆ มากด้วย เว็บไซต์นี้จึงเป็นเว็บไซต์ที่น่าสนใจและจะได้รับค่าความน่าสนใจมากขึ้น
- Hotbot.com มีการตรวจสอบจำนวนผู้เข้าชมเว็บไซต์นั้น ถ้าหากเว็บไซต์นั้นมีผู้เข้าชมมากย่อมหมายถึงว่าเว็บไซต์นั้นมีความน่าสนใจมากนั่นเอง

และเนื่องจากมีการใช้ 2 หลักการนี้อย่างแพร่หลาย จึงมีผู้สร้างเว็บไซต์บางคนทำให้เว็บไซต์ของตนเองได้คะแนนความน่าสนใจจาก Search Engine ด้วยวิธีการต่าง ๆ เช่น มีคำ ๆ นั้นหลายร้อยคำเกินความจำเป็นในเว็บไซต์นั้น เป็นต้น วิธีการเหล่านี้เรียกว่า "Search Engine Spamming" ซึ่งทำให้ Search Engine เองต้องตรวจดูในจุดนี้ด้วย

สำหรับคะแนนความน่าสนใจที่ได้ บาง Search Engine อาจจะนำมาแสดงให้ผู้ใช้ดูเป็นตัวเลข แม้ว่าบาง Search Engine จะไม่แสดงค่าความน่าสนใจให้ดูแต่เว็บไซต์ที่แสดงอยู่ก็ได้เรียงลำดับตามค่านั้นแล้ว วิธีการคำนวณค่าความน่าสนใจแบบอื่น ๆ รวมไปถึงการเข้าไปดึงข้อมูลดังแสดงในตารางที่ 2.3

Ranking	Yes	No	Notes
Meta Tags Boost Ranking	Inktomi	AltaVista, Excite, FAST, Google, NLight	
Link Popularity Boosts Ranking	All	n/a	Very important at Google
Direct Hit Boost Ranking	HotBot	Others	
Spam	**Yes**	**No**	**Notes**
Meta Refresh	AltaVista	Excite, FAST, Google, Inktomi, NLight	
Invisible Text	Others	Excite, FAST, Google	
Tiny Text	AltaVista, Inktomi	Excite, FAST, Google, NLight	

ตารางที่ 2.3 แสดงการให้ค่าความน่าสนใจแก่เว็บไซต์ของแต่ละ Search Engine

2.2.8 Search Engine ในปัจจุบัน

Search Engine มีการพัฒนาเรื่อยมานับแต่ Search Engine ตัวแรก คือ Webcrawler จนในปัจจุบันมี Search Engine มากมายหลายร้อยตัวและใน Search Engine บางตัวสามารถเจาะเข้าไปค้นหาเฉพาะหัวข้อที่สนใจได้ เช่น Music, News เป็นต้น ทำให้เว็บไซต์ที่ได้ตรงกับความต้องการของผู้ค้นหามากยิ่งขึ้น นอกจากนี้ยังมี Search Engine อีกลักษณะหนึ่งเรียกว่า "Meta Searcher" ซึ่งจะเรียกใช้ฐานข้อมูลจาก Search Engine ตัวอื่น ๆ อีกหลายตัวแล้วนำข้อมูลที่ได้จากหลายแหล่งนั้นมาประมวลผลอีกครั้งหนึ่ง ดังจะได้กล่าวในหัวข้อ 2.7 ต่อไป

Search Engine ที่เป็นที่นิยมในปัจจุบันมีดังต่อไปนี้

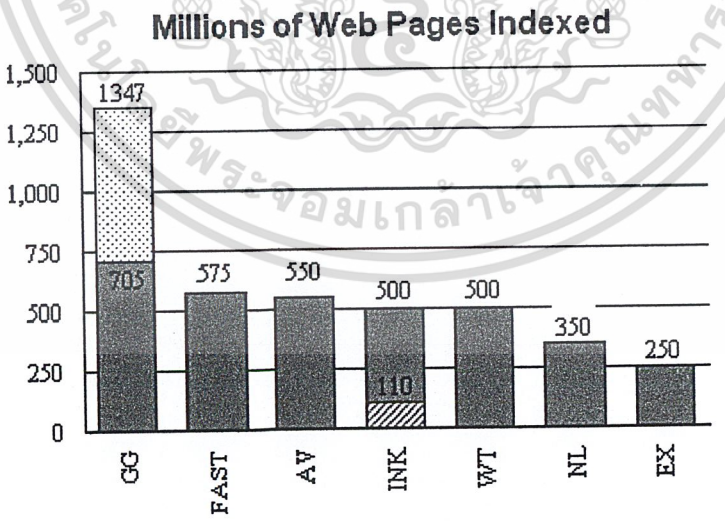
- Altavista เริ่มเปิดบริการครั้งแรกในเดือน ธันวาคม ปี 2538 โดย Digital ต่อมา Compaq ซื้อ Digital ไปในปี1998 หลังจากนั้นจึงแยกตัวออกมาเป็นบริษัทย่อยในเครือ CMGI AltaVista เป็นหนึ่งใน Search engine ที่มีฐานข้อมูลนับเป็นจำนวนหน้ามากที่สุดตัวหนึ่งและมีมาช้านานแล้ว มีฟังก์ชั่นการทำงานให้เลือกมากมาย รวมไปถึงการเลือกค้นหาเฉพาะด้าน เช่น ด้านข่าวสาร, ด้านการซื้อของ, ด้านสื่อบันเทิง เป็นต้น และยังมีการค้นหาแบบ Directory ผ่านทาง LookSmart ซึ่งเป็น Search Engine ยอดนิยมอีกตัวหนึ่ง
- Direct Hit มีจุดเด่นพิเศษกว่า Search Engine ตัวอื่น ๆ คือ สามารถวัดได้ว่ามีคนเข้าชมเว็บไซต์นั้นผ่านทางตัว Direct Hit เองและพันธมิตร เช่น HotBot มากน้อยเพียงใด หากเว็บไซต์นั้นมีคนเข้าชมมากย่อมหมายถึงเว็บไซต์นั้นน่าจะเป็นเว็บไซต์ที่ดีและน่าสนใจ และด้วยเหตุที่ Direct Hit เป็นพันธมิตรกับ Search Engine ตัวอื่น ๆ อีกมากมาย เช่น HotBot, MSN Search, Ask Jeeves เป็นต้น จึงทำให้ Direct Hit เป็นที่นิยมอย่างรวดเร็ว

- Excite มีฐานข้อมูลอยู่ในระดับกลาง และสามารถค้นหาแบบ Directory ผ่านทาง LookSmart เริ่มเปิดบริการในปลายปี 2538 และเติบโตอย่างรวดเร็วสามารถเข้าซื้ออีก 2 บริษัทคู่แข่งได้ คือ Magellan ในเดือน กรกฎาคม ปี 2539 และ WebCrawler ในเดือน พฤศจิกายน ปี 2539
- GoTo ต่างจาก Search Engine ตัวอื่น ๆ คือ GoTo จะขายฐานข้อมูลของตัวเองให้กับ Search Engine ตัวอื่น เช่น AltaVista, AOL, Search, Lycos, HotBot, Netscape Search เป็นต้น โดย GoTo จะเน้นการพัฒนาด้านความเกี่ยวข้องของข้อมูลกับคีย์เวิร์ด GoTo เริ่มเปิดบริการเมื่อปี 2540 ต่อมาร่วมกับ World Wide Web Worm ของ University of Colorado จนกระทั่งเดือน กุมภาพันธ์ ปี 2541 จึงเริ่มให้บริการแก่ Search Engine อื่น ๆ หลังจากนั้นไม่นานจึงเปลี่ยนไปร่วมมือกับ Inktomi สำหรับการให้ค้นหาแบบฟรี
- Google ใช้หลักการของ "Link Popularity" เป็นหลักในการจัดลำดับเว็บไซต์(ดู 2.2.7 การจัดลำดับเว็บไซต์ของ Search Engine ประกอบ) ซึ่งเหมาะกับการหาเว็บไซต์โดยใช้คำทั่ว ๆ ไป เช่น Car, Travel, Computer เป็นต้น เพราะเปรียบเสมือนว่าผู้สร้างเว็บไซต์หลาย ๆ คนออกความเห็นให้ว่าเว็บไซต์นี้เป็นเว็บไซต์ที่ดี Google ยังเป็น Search Engine ที่มีฐานข้อมูลมากที่สุดอีกด้วย และแบ่งปันฐานข้อมูลกับ Yahoo และ NetScape Search ด้วยเหตุนี้จึงทำให้ Google เป็นที่นิยมได้ในเวลาไม่กี่ปี
- HotBot เป็นที่นิยมของนักวิจัยและผู้ค้นหาข้อมูล เพราะมีฟังก์ชั่นการค้นหาหลากหลาย และตัวฐานข้อมูลก็มีการร่วมมือกับหลาย Search Engine ในหน้าแรกของผลลัพธ์ที่ได้จะมาจาก Direct Hit ในหน้าที่สองจะมาจาก Inktomi สำหรับการค้นหาแบบ Directory นั้นได้มาจาก Open Directory HotBot เริ่มเปิดบริการเมื่อเดือน พฤษภาคม ปี 2539 โดยบริษัท Wired Digital ที่พึ่งเริ่มเข้ามาในธุรกิจนี้ ต่อมา Lycos ซื้อไปในเดือน ตุลาคม ปี 2541
- Inktomi เริ่มโดยการพัฒนาและวิจัยใน UC Berkeley ต่อมาผู้สร้างได้ก่อตั้งบริษัทตั้งชื่อตามตัว Search Engine และจัดทำดัชนีเว็บไซต์ใหม่ ซึ่งถูกใช้ครั้งแรกโดย HotBot ต่อมาเริ่มเป็นที่นิยมมากขึ้นจึงเปิดบริการในด้านอื่น ๆ มากขึ้นเรื่อย ๆ โดยจะเน้นการให้บริการดัชนีแบบทั่วไปแก่ Search Engine อื่น ๆ
- LookSmart มีลักษณะคล้าย Yahoo คือ เป็น Search Engine แบบ Directory ที่จัดประเภทโดยมนุษย์ โดยให้บริการแก่ Search Engine ตัวอื่น ๆ เช่น MSN Search, Excite เป็นต้น เมื่อไม่พบเว็บไซต์นั้นในไดเรกทอรี่ จะส่งไปหาต่อใน Inktomi LookSmart เริ่มเปิดบริการขึ้นเมื่อเดือน ตุลาคม ปี 2539 โดยมีผู้สนับสนุนคือ Reader's Digest ประมาณ 1 ปี คณะผู้บริหารจึงซื้อกลับมาเพื่อบริหารอย่างเต็มรูปแบบ
- Lycos เริ่มต้นก่อตั้งขึ้นโดยเป็น Search Engine ต่อมาในเดือน เมษายน ปี 2542 ได้เปลี่ยนเป็นแบบ Directory คล้ายกับ Yahoo โดยดัชนีเว็บไซต์หลัก ๆ นั้นมาจาก Open Directory และ FAST Search ต่อมาได้ซื้อ HotBot โดยยังคงให้บริการเป็นอิสระจาก Lycos
- Northern Light เป็นอีกหนึ่ง Search Engine ที่เป็นที่นิยมของนักวิจัยและผู้ค้นหาข้อมูล เนื่องจากมีฐานข้อมูลที่ใหญ่และสามารถแยกประเภทเว็บไซต์ที่ได้เป็นหัวข้อ นอกจากนี้ยังมีดัชนี

ถึงกลุ่มของเว็บไซต์ประเภทที่เสียเงินเพื่อเข้าชมด้วย เช่น Newswires, Magazines, Databases เป็นต้น จะเสียค่าเข้าชมประมาณ $4 Northern Light เริ่มเปิดบริการเมื่อเดือน สิงหาคม ปี 2540

- Open Directory เริ่มเปิดบริการครั้งแรกในเดือน กรกฎาคม ปี 2541 โดยมีจุดมุ่งหมายเพื่อให้ทุกคนสามารถเข้ามาใช้ได้ เป็นบริการสาธารณะ มีชื่ออย่างเป็นทางการว่า "NewHoo" ต่อมาในเดือน พฤศจิกายน ปี 2541 ถูก Netscape ซื้อไป แต่ยังคงเปิดให้ใช้บริการทั่วไปอยู่ภายใต้กฎหมายลิขสิทธิ์ มี Search Engine หลายตัวที่ใช้บริการจาก Open Directory เช่น Netscape Search, Lycos, AOL Search เป็นต้น
- RealNames เป็นบริการสำหรับการตั้งชื่อเว็บไซต์เพื่อให้ง่ายต่อการเรียกใช้ ตัวอย่างเช่น ต้องการเข้าไปชมเว็บไซต์ของ "Nike" ก็สามารถพิมพ์คำว่า "Nike" ผ่านทาง Realnames เพื่อเข้าชมเว็บไซต์ได้ทันที
- Yahoo เป็น Search Engine ตัวแรกที่ให้บริการแบบ Directory เริ่มให้บริการครั้งแรกในปลายปี 2537 และยังคงได้รับความนิยมอย่างสูงเนื่องจากมีบริการเสริมอื่น ๆ อีกมากมาย และเป็นฐานข้อมูลที่ใช้มนุษย์จัดทำขึ้น ซึ่งใช้คนถึง 150 คนในการจัดประเภทเว็บไซต์ ในขณะนี้มีมากกว่า 1 ล้านเว็บไซต์ในฐานข้อมูลแล้ว หากไม่สามารถค้นหาเว็บไซต์นั้นได้ในฐานข้อมูล Yahoo จะไปเรียกใช้บริการของ Google ซึ่งมีฐานข้อมูลมากที่สุดใน Search Engine

แสดงการเปรียบเทียบจำนวนดัชนีเว็บไซต์ด้วยกราฟในรูปที่ 2.2 และ แสดงการประเมิน Search Engine แต่ละตัวจากหลาย ๆ แหล่งประเมินในตารางที่ 2.4 สำหรับตารางที่ 2.5 จะแสดงจำนวนไดเรกทอรีสำหรับ Search Engine แบบไดเรกทอรีที่นิยมใช้กันอยู่



อักษรย่อ: GG=Google, FAST=FAST, AV=AltaVista, INK=Inktomi, WT=WebTop.com, NL=Northern Light, EX=Excite.

รูปที่ 2.2 แสดงจำนวนของดัชนีเว็บไซต์ในฐานข้อมูลของแต่ละ Search Engine

Search Engine Reviews Chart

Service	Cnet Relevancy 6/00	Cnet Overall 6/00	Cnet 4/99	PC Mag 9/99	PC Mag 12/98	PCC MVP 1998	PC Mag 9/98	Cnet 1/98
Ask Jeeves	n/a	n/a			EC			
AltaVista	4.14	8	2/2				2	3.3
AOL Search	n/a	n/a					3	
Excite	n/a	n/a	4/2			2	1	3.1
Go2Net	n/a	n/a			EC			
Google	4.88	9			HM			
HotBot	n/a	n/a	5/3		HM	1	2	3.6
Go (Infoseek)	n/a	n/a	3/3				2	3.5
LookSmart	n/a	n/a					2	
Lycos	4.29	5	1/4				2	3.3
MSN Search	4.81	8					3	
Netscape	3.59	4					3	
Northern Light	n/a	n/a			EC			1.6
Snap	n/a	n/a					3	
WebCrawler	n/a	n/a						
Yahoo	3.71	9			EC	2	1	

อันดับของ Search Engine จากแต่ละผู้จัดอันดับ อันดับที่1 อันดับที่2 อันดับที่3

ตารางที่ 2.4 แสดงการจัดอันดับ Search Engine

Service	Type	Editors	Cats	Links...	As Of
Open Directory	D	36,000	361,000	2.6 million	4/01
LookSmart	D	200	200,000	2 million	8/00
Yahoo	D	100+	n/a	1.5 to 1.8 million	8/00
NBCi (Snap)	D	30	80,000	1.5 million	12/00
AskJeeves	AS	150	n/a	128 million	3/01
AltaVista	SE	See LookSmart			

Excite	SE	See LookSmart
HotBot	SE	See Open Directory
Lycos	D	See Open Directory
MSN Search	SE	See LookSmart
Netscape	SE	See Open Directory

ตารางที่ 2.5 แสดงรายละเอียดของ Search Engine แบบ Directory

2.2.9 Meta Searcher

Meta Searcher คือ Search Engine ที่เรียกใช้บริการจาก Search Engine ตัวอื่น ๆ อีกหลายตัวในการค้นหาข้อมูล กำลังเป็นที่นิยมมากขึ้นเรื่อย ๆ เพราะมีข้อได้เปรียบกว่า Search Engine ธรรมดา ดังนี้

1. ลดขนาดของฐานข้อมูลของเซิร์ฟเวอร์ของตัว Search Engine นั้นเองและสอดคล้องกับหลักการแบ่งปันข้อมูลที่มีของ Web robot ซึ่งทำให้มี Web Robot ออกมาค้นหาข้อมูลในอินเตอร์เน็ตน้อยลง เพื่อจะได้ไม่มี Web Robot มาใช้ Resource ของเซิร์ฟเวอร์ต่าง ๆ
2. ได้ฐานข้อมูลที่มีขนาดใหญ่มากขึ้น
3. สามารถคำนวนความน่าสนใจของเว็บไซต์ได้แม่นยำขึ้นโดยรวบรวมค่าความน่าสนใจของเว็บไซต์นั้นจาก Search Engine หลาย ๆ ตัวนำมาประมวลผลรวมกัน

ตัวอย่าง Meta Searcher ที่เป็นที่นิยมในปัจจุบัน มีดังนี้

- Dogpile สามารถกำหนด Search Engine ที่ต้องการให้ส่งคำไปค้นหาได้และยังสามารถกำหนดค่าพิเศษอื่น ๆเพิ่มเติมได้อีกด้วย แล้วจึงแสดงผลโดยแยกตามผลลัพธ์ที่ได้จากแต่ละ Search Engine
- Ixquick จะแสดงเว็บไซต์ที่ได้เพียง 10 อันดับแรกที่ได้รับมาจาก Search Engine ที่ส่งผลลัพธ์กลับมา
- MetaCrawler เป็น Meta Searcher ตัวแรก ๆ ที่มี เริ่มให้บริการครั้งแรกในเดือนกรกฎาคม ปี 2538 โดย University of Washington ต่อมาถูกซื้อไปโดย Go2Net ในเดือนกุมภาพันธ์ ปี 2540 ซึ่งทำให้มีเงินทุนในการพัฒนาให้บริการได้ดียิ่งขึ้น แสดงผลลัพธ์โดยมีค่าความน่าสนใจของแต่ละเว็บไซต์เต็ม 1000 คะแนน
- Search.com ควบคุมและพัฒนาโดย Cnet ซึ่งสามารถให้บริการค้นหาได้ทั้งแบบค้นหาทั่วไปหรือค้นหาเฉพาะอย่าง(เช่น เพลง , ข่าว) ใช้เทคโนโลยีของ SawySearch ซึ่งเป็นอีกหนึ่ง Meta Searcher ที่เก่าแก่มาก แต่ปัจจุบันได้ปิดให้บริการแล้ว
- Vivisimo เป็น Meta Searcher ในยุคใหม่ที่มีการพัฒนาให้จัดแบ่งประเภทของเว็บไซต์ที่ได้ให้ด้วย รวมถึงมีการแสดงค่าความน่าสนใจของเว็บไซต์นั้น ทำให้ผู้ใช้สามารถเลือกเข้าไปดูในประเภทของเว็บไซต์ที่ได้ที่ตรงกับที่ผู้ใช้ต้องการได้

- ProFusion เป็นอีกหนึ่ง Meta Searcher ที่เป็นที่นิยมมาก สามารถปรับแต่งการค้นหาได้ มีการตรวจสอบว่า URL ที่ได้สามารถเรียกใช้ได้หรือไม่ เริ่มพัฒนาโดย University of Kansas ต่อมาถูกซื้อโดย Intelliseek ในเดือนเมษายน ปี 2543

2.2.10 Search Engine ที่ใช้ในระบบกลั่นกรองสารสนเทศ

ระบบกลั่นกรองสารสนเทศเลือกใช้ Search Engine 2 ตัว คือ MetaCrawler และ Yahoo โดยมีลักษณะการใช้งานดังนี้

- MetaCrawler ใช้ค้นหา URL ตามคำที่ผู้ใช้ป้อนให้ พร้อมทั้งรับเอาคะแนนความน่าสนใจ (เต็ม 1000 คะแนน)มาด้วย เพื่อใช้ประกอบการให้คะแนนของระบบ สาเหตุที่เลือกใช้ MetaCrawler เพราะมีการให้คะแนนความน่าสนใจที่ค่อนข้างน่าเชื่อถือได้ เนื่องจาก MetaCrawler จะให้คะแนนความน่าสนใจโดยนำเอาคะแนนความน่าสนใจจาก Search Engine หลาย ๆ ตัวมาเปรียบเทียบกัน นั่นคือได้รับการกลั่นกรองมาแล้วจากหลาย Search Engine ด้วยกัน และตรวจดูความเกี่ยวข้องของคีย์เวิร์ดกับ URL นั้นเองด้วยสูตรการคำนวณค่าความน่าสนใจของ MetaCrawler เอง นอกจากนี้แล้วยังมีการเข้าใช้ข้อมูลที่ง่ายและมีความน่าเชื่อถือและความมีเสถียรภาพของผู้ควบคุมและดำเนินการของ MetaCrawler (Go2Net)
- Yahoo ใช้แบ่งประเภทของเว็บไซต์ที่ได้ เพื่อให้คะแนนความน่าสนใจตามประเภทของเว็บไซต์นั้น นอกจากนี้ยังใช้ในการค้นหา URL ที่ต้องการในประเภทของเว็บไซต์ที่ผู้ใช้สนใจด้วย สาเหตุที่เลือกใช้ Yahoo เพราะเป็น Search Engine แบบ Directories ที่มีมายาวนานมากและมี URL อยู่มากมาย การเข้าใช้ข้อมูลง่ายและมีเซิร์ฟเวอร์ที่มีเสถียรภาพสูงมากเนื่องจากมีเซิร์ฟเวอร์ให้สามารถเรียกใช้ได้ 6-7 ตัวทั่วโลก ถึงกับมีคนเคยพูดว่า "Yahoo never downs"

2.3 โพรโตคอลเอชทีทีพี (HTTP : HyperText Transfer Protocol)

โพรโตคอลนี้สร้างขึ้นสำหรับบริการที่เรียกว่า WWW(World Wide Web) ในเครือข่ายอินเตอร์เน็ตโดยเฉพาะ โพรโตคอลนี้จะเป็นตัวกำหนดวิธีการส่งข้อมูลหรือไฟล์ระหว่างเครื่องคอมพิวเตอร์ลูกข่าย (Client) และ เครื่องแม่ข่าย (Server) รวมถึงกำหนดกฎระเบียบในการติดต่อด้วย โพรโตคอลนี้ถูกออกแบบมาให้มีความกระทัดรัด สามารถทำงานได้รวดเร็ว มีการะบวนการทำงานที่ไม่ซับซ้อน และมีคำสั่งที่ใช้งานไม่มากนัก แต่สามารถรองรับข้อมูลได้ทุกแบบ ไม่ว่าจะเป็นข้อมูลทั่วไปที่เข้ารหัสแบบ MIME หรือข้อมูลที่เป็นกราฟิก เช่น ไฟล์ที่เป็น GIF หรือ JPEG เป็นต้น

2.3.1 วิธีการติดต่อของโพรโตคอล HTTP

โพรโตคอล HTTP อยู่บนพื้นฐานของระบบเครือข่ายไคลเอนต์ / เซิร์ฟเวอร์ (Client / Server) ที่ต้องมีการร้องขอรับบริการจากไคลเอนต์ (request) และการตอบสนองหรือการให้บริการของเซิร์ฟเวอร์ (response) จึงสามารถแบ่งการทำงานออกเป็น 2 ด้านคือ ด้านเว็บเซิร์ฟเวอร์ และด้านไคลเอนต์ โดยไคลเอนต์จะติดต่อเข้ามายังเซิร์ฟเวอร์และอ้างถึงแอดเดรสของเซิร์ฟเวอร์โดยใช้รูปแบบของ URL ส่วนด้านเซิร์ฟเวอร์จะส่งข้อมูลกลับมาในรูปแบบที่เป็นภาษา HTML(HyperText Markup Language) โดยที่โพรโตคอล HTTP ใช้วิธีการเข้ารหัสในแบบ MIME เป็นมาตรฐานของการทำงาน โพรโตคอลนี้ถูกออกแบบมาให้สามารถรับส่งข้อมูลผ่าน Proxy หรือ Firewall ต่าง ๆ ได้ โดยอาศัยการเชื่อมต่อผ่านทางโพรโตคอล TCP/IP อีกทีหนึ่ง โดยใช้พอร์ตหมายเลข 80 เป็นช่องทางมาตรฐานในการติดต่อ ในทางปฏิบัติจะใช้พอร์ตหมายเลขอื่นก็ได้ ในปัจจุบันเว็บบราวเซอร์ทั่วไปจะกำหนดค่ามาตรฐานไว้ที่พอร์ต 80 ดังนั้นหากมีการกำหนดไว้ที่พอร์ตอื่น จะทำให้เกิดความลำบากต่อผู้ใช้ที่ต้องระบุหมายเลขพอร์ตลงใน URL ด้วย

ด้วยเหตุที่การทำงานของโพรโตคอล HTTP เป็นแบบไคลเอนต์และเซิร์ฟเวอร์ ดังนั้นการติดต่อสื่อสารใดๆผ่านโพรโตคอลนี้จำเป็นต้องมีเครื่องตัวแม่กับตัวลูก การสื่อสารจะสมบูรณ์ได้ การติดต่อกันระหว่างไคลเอนต์ไปยังเซิร์ฟเวอร์ผ่านโพรโตคอล HTTP มีขั้นตอนดังนี้

ขั้นแรก : Open Socket

ไคลเอนต์ จะสร้างการเชื่อมต่อ (Connection) กับเซิร์ฟเวอร์ผ่านซ็อกเก็ต (Socket)

ขั้นที่สอง : Request

ไคลเอนต์ส่งคำร้องขอข้อมูลไปยังเซิร์ฟเวอร์

ขั้นที่สาม : Information Transfer

เซิร์ฟเวอร์จะไปหาข้อมูลที่ไคลเอนต์ต้องการ

ขั้นที่สี่ : Response

เซิร์ฟเวอร์ส่งข้อมูลตอบสนอง (Response) กลับมายังไคลเอนต์เสมอ

ขั้นสุดท้าย : Close Socket

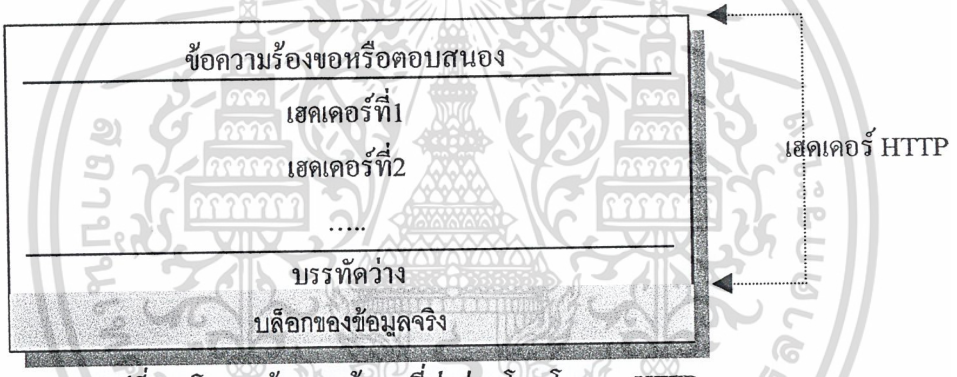
ปลดการเชื่อมต่อของซ็อกเก็ตของทั้งสองฝั่งออก

ด้วยการทำงานของโพรโตคอล HTTP ที่มีการเชื่อมต่อในระยะเวลาเพียงสั้น หรือที่เรียกว่าเป็นโพรโตคอลแบบ Connectionless ในลักษณะดังกล่าว ทำให้ในช่วงเวลาหนึ่ง ๆ เซิร์ฟเวอร์ที่ให้บริการ WWW สามารถรองรับไคลเอนต์ได้จำนวนมากพร้อม ๆ กัน เพราะไม่มีใครได้ทำการเชื่อมต่ออย่างถาวร

ในการร้องขอรับบริการจากไคลเอนต์ และการตอบสนองหรือการให้บริการจากเซิร์ฟเวอร์นั้น ย่อมต้องมีการรับส่งข้อมูลระหว่างกัน แต่ข้อมูลที่รับส่งกันในแต่ละครั้งไม่ได้มีเฉพาะข้อมูลเพียงอย่างเดียว แต่ละฝ่ายจะมีส่วนเฮดเดอร์ HTTP (HTTP header) เข้าไปในส่วนต้นของข้อมูลที่รับ-ส่งกันด้วย ซึ่งเฮดเดอร์ HTTP จะเป็นตัวบอกว่าข้อมูลที่ส่งมานี้เป็นข้อมูลอะไร เป็นข้อมูลการร้องขอจากไคลเอนต์ หรือเป็นข้อมูลตอบสนองจากเซิร์ฟเวอร์

2.3.2 โครงสร้างของโพรโตคอล HTTP

โครงสร้างของข้อมูล HTTP จะแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ ส่วนเฮดเดอร์ หรือเรียกว่า metadata จะเป็นส่วนเก็บข้อมูลที่จำเป็นต้องใช้ภายในโพรโตคอล ส่วนที่สองเป็นส่วนของข้อมูลจริงที่ต้องการรับส่ง ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 โครงสร้างของข้อมูลที่ส่งผ่านโพรโตคอล HTTP

2.3.2.1 เฮดเดอร์เอชทีทีพี (HTTP Header)

ดังรูปที่ 2.3 เฮดเดอร์เอชทีทีพี (HTTP Header) ประกอบด้วย 2 ส่วน คือ ส่วนข้อมูลร้องขอหรือตอบสนอง และส่วนเฮดเดอร์ย่อย

2.3.2.1.1 ส่วนข้อมูลร้องขอหรือตอบสนอง

ส่วนนี้เป็นส่วนในการแยกว่าเป็นข้อความตอบสนองจากเซิร์ฟเวอร์ หรือข้อความร้องขอจากไคลเอนต์ และรายละเอียด ดังต่อไปนี้

2.3.2.1.1.1. ข้อความร้องขอ (request)

จากรูปที่ 2.3 ในส่วนข้อความการร้องขอ(บรรทัดแรก) จะประกอบด้วย 3 ส่วน คือ

1. วิธีการร้องขอ หรือที่เรียกว่า "เมธอด"
2. ไดเรกทอรีและชื่อไฟล์ที่ต้องการจากเซิร์ฟเวอร์

3. เวอร์ชันของ HTTP ที่ไคลเอนต์ใช้อยู่

โดยที่แต่ละส่วนจะถูกแบ่งด้วยช่องว่าง (space) เขียนรูปแบบในการเขียนข้อความบรรทัดแรกของเฮดเดอร์ HTTP ได้ดังนี้

<Method> /path/file HTTP/x.x

สังเกตว่าการขอข้อมูลจากเซิร์ฟเวอร์ จะระบุเฉพาะไดเรกทอรี และชื่อไฟล์ที่ต้องการ ไม่ต้องบอกชื่อโฮสต์และโดเมนเนมเพราะได้มีการสร้างการเชื่อมต่อกับโฮสต์ในโพรเซสก่อนหน้านี้ไปแล้ว การร้องขอข้อมูลจึงไม่ต้องระบุชื่อโฮสต์ซ้ำอีกครั้ง

นอกจากคำร้องขอที่อยู่ในบรรทัดแรกแล้ว ยังมีข้อมูลอื่น ๆ ที่ส่งไปให้กับเซิร์ฟเวอร์ด้วย คือข้อมูลในส่วนที่สอง ซึ่งเรียกว่า “เฮดเดอร์” (header) ข้อมูลในเฮดเดอร์แต่ละเว็บบราวเซอร์แต่ละเวอร์ชันอาจจะไม่เหมือนกัน ซึ่งข้อมูลเฮดเดอร์นี้จะบอกรายละเอียดของผู้ส่งว่ามีอะไรบาง เฮดเดอร์นี้จะบอกให้เซิร์ฟเวอร์ทราบได้ว่าข้อความร้องขอถูกส่งมาจากใคร และสามารถนำไปใช้เพื่อประโยชน์ในเรื่องอื่น ๆ ต่อไปได้

จากโครงสร้างข้อมูลที่รับส่งระหว่างไคลเอนต์กับเซิร์ฟเวอร์ในรูปที่ 2.3 หลังจากเฮดเดอร์รายการสุดท้ายแล้ว จะมีบรรทัดว่าง หลังจากนั้นจะเป็นส่วนของบล็อกข้อมูล ทั้งนี้หากเป็นการร้องขอจากไคลเอนต์ ก็จะขึ้นอยู่ว่าใช้เมธอดใดในการร้องขอ หากเป็นเมธอดGET ก็ไม่จำเป็นต้องมีข้อมูลอะไรในส่วนนี้ เนื่องจากเซิร์ฟเวอร์จะไม่สนใจข้อมูลในส่วนนี้ ทั้งนี้เนื่องจากรูปแบบของโพรโตคอลนั่นเอง จะกล่าวละเอียดต่อไป

เมื่อไคลเอนต์เชื่อมต่อกับเซิร์ฟเวอร์เรียบร้อยแล้ว ไคลเอนต์จะเป็นฝ่ายส่งข้อมูลการร้องขอไปยังเซิร์ฟเวอร์ ซึ่งการร้องขอไปยังเซิร์ฟเวอร์นี้สามารถทำได้หลายวิธี ทั้งนี้ทั้งนั้นก็ขึ้นอยู่กับเวอร์ชันของโพรโตคอล HTTP ที่ใช้ หากเป็นเวอร์ชัน 1.0 จะมีวิธีการร้องขอมาตรฐาน 3 วิธี คือ GET, HEAD และ POST แต่หากเป็นโพรโตคอล HTTP เวอร์ชัน 1.1 จะมีวิธีการร้องขอเพิ่งจากเวอร์ชัน 1.0 อีกหลายวิธี เช่น OPTIONS, PUT, DELETE หรือ TRACE เป็นต้น ดังนั้นการที่เราจะเลือกว่าจะใช้โพรโตคอล HTTP เวอร์ชันไหน ต้องขึ้นอยู่กับเซิร์ฟเวอร์ และ ไคลเอนต์ที่จะทำงานด้วย คือ หากว่าเซิร์ฟเวอร์สนับสนุนการทำงานของ HTTP 1.1 แล้ว วิธีการร้องขอก็สามารถใช้ของเวอร์ชัน 1.1 ได้

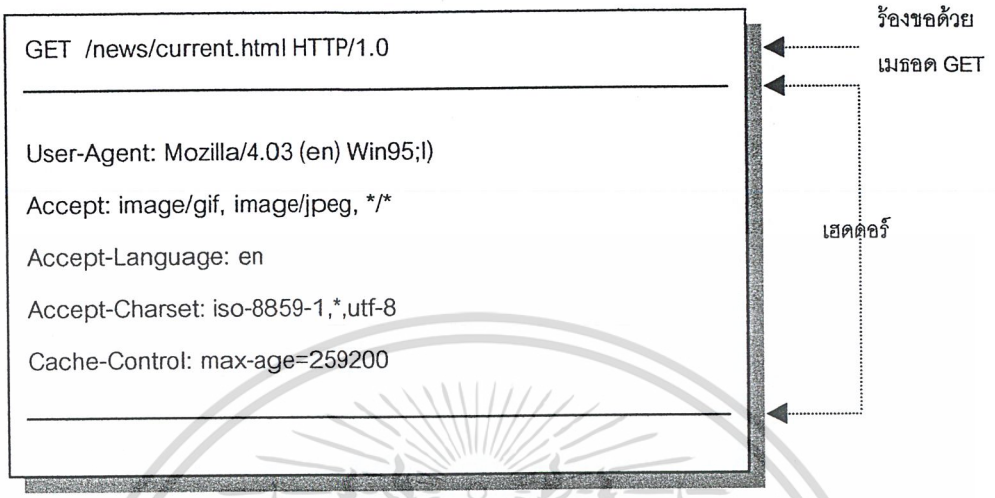
เพื่อไม่ให้เกิดปัญหาในการทำงาน จึงควรใช้โพรโตคอล HTTP เวอร์ชัน 1.0 ซึ่งเว็บเซิร์ฟเวอร์ส่วนใหญ่ในอินเตอร์เน็ตจะสามารถรองรับการร้องขอจากเวอร์ชันนี้ โดยโพรโตคอล HTTP เวอร์ชัน 1.0 มีวิธีการร้องขออยู่ 3 วิธีด้วยกัน โดยมีรายละเอียดดังนี้

1. การร้องขอด้วยเมธอด GET

มีรูปแบบดังนี้

GET /path/file HTTP/x.x

เป็นการร้องขอให้เซิร์ฟเวอร์ส่งไฟล์มาให้ หรือ เป็นการร้องขอโดยมีการส่งข้อมูลจากทางไคลเอนต์ไปให้ด้วยก็ได้ ดูตัวอย่างประกอบในรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างข้อความร้องขอด้วยเมธอด GET

นอกจากข้อความร้องขอด้วยเมธอด GET ใช้สำหรับการร้องขอข้อมูลจากเซิร์ฟเวอร์แล้ว ข้อความร้องขอด้วยเมธอด GET นี้ยังสามารถใช้สำหรับส่งข้อมูลไปยังเครื่องเซิร์ฟเวอร์ได้อีกด้วย โดยข้อมูลที่ต้องการส่งให้กับเซิร์ฟเวอร์นั้นจะต่อท้าย URL โดยมีเครื่องหมาย ? คั่นระหว่าง URL กับข้อมูลนั้น ลักษณะข้อมูลจะประกอบด้วยตัวแปร และค่าของตัวแปรนั้น โดยเขียนต่อในลักษณะ <Variable Name>=<Variable Value>&<Variable Name>=<Variable Value>&..... แต่มีข้อจำกัดด้านขนาดของข้อมูลที่ส่งไปให้เซิร์ฟเวอร์ด้วยเมธอดนี้ เนื่องจากให้ส่งได้ครั้งละไม่เกิน 256 ตัวอักษร(นับจากที่เข้ารหัสแล้ว) ความยาวทั้งหมดเริ่มนับจากชื่อไดเรกทอรีเป็นต้นไป แต่ความยาวนี้ขึ้นอยู่กับระบบปฏิบัติการอีกทีหนึ่ง

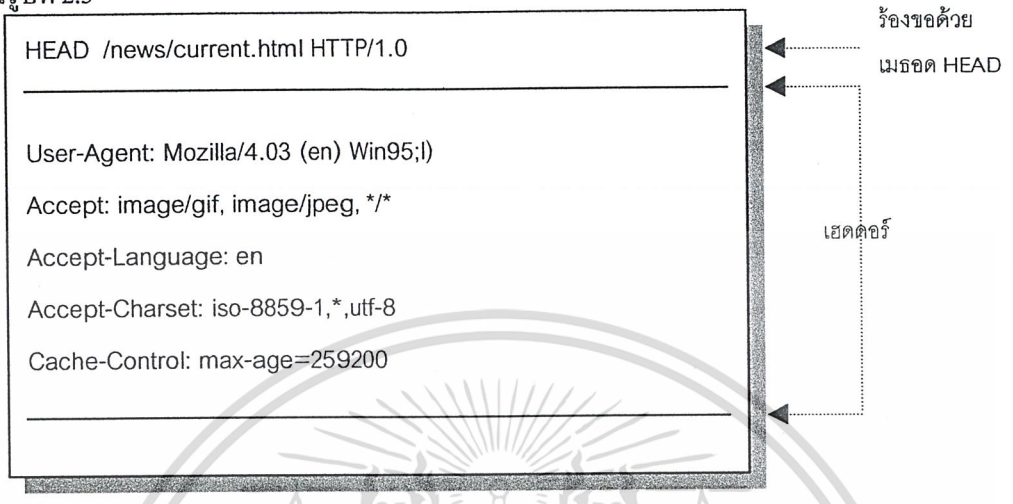
2. การร้องขอด้วยเมธอด HEAD
   มีรูปแบบดังนี้

```
HEAD /path/file HTTP/x.x
```

เป็นการร้องขอเพื่อถามเซิร์ฟเวอร์ว่ามีไฟล์ที่ต้องการอยู่ในเซิร์ฟเวอร์หรือไม่ (ถามเฉย ๆ ไม่ต้องการให้เซิร์ฟเวอร์ส่งไฟล์จริงมาให้ ซึ่งมีประโยชน์สำหรับการตรวจสอบว่ามีไฟล์ที่ต้องการอยู่ในเซิร์ฟเวอร์หรือไม่ หรือใช้ตรวจสอบความสมบูรณ์ของลิงค์ก็ได้ รหัสตอบสนองในบรรทัดสถานะจึงอาจเป็น 200 (มีไฟล์ที่ต้องการ) หรือ 404 (ไม่มีไฟล์ที่ต้องการ) และข้อมูลอื่นส่งเพิ่มเติมมาในเฮดเดอร์ด้วย เช่น วันที่ปรับปรุงแก้ไขไฟล์ครั้งสุดท้าย (Last Modified) เป็นต้น

การจะตรวจสอบว่ามีไฟล์นั้นอยู่ที่เซิร์ฟเวอร์หรือไม่ อาจใช้วิธีการร้องขอด้วยเมธอด GET แล้วใช้การเช็คผลการตอบสนอง แต่การตอบสนองจากการร้องขอด้วยเมธอด HEAD จะไม่

มีการส่งเนื้อหาของไฟล์มาให้ (ไม่มีบล็อกข้อมูล) ดังนั้นใช้เมธอด HEAD จึงทำงานได้คำตอบเร็วกว่า มักจะใช้เมธอดนี้ในการตรวจสอบกับทางเซิร์ฟเวอร์ หากมีการปรับปรุงไฟล์นั้น จึงจะมีการ download มาทับไฟล์เดิมในเครื่อง (ร้องขอไฟล์นั้นอีกครั้งด้วยเมธอด GET) ดูตัวอย่างประกอบในรูปที่ 2.5



รูปที่ 2.5 ตัวอย่างข้อความร้องขอด้วยเมธอด HEAD

3. การร้องขอด้วยเมธอด POST
   มีรูปแบบดังนี้

```
POST /path/file HTTP/x.x
```

เป็นการร้องขอให้เซิร์ฟเวอร์รับข้อมูลจากไคลเอนต์เพื่อนำไปประมวลผล หรือนำไปเก็บในฐานข้อมูลต่อไป โดยมีเงื่อนไขดังนี้

- ข้อมูลที่จะส่งไปให้เซิร์ฟเวอร์จะอยู่ภายในบล็อกข้อมูล ดังนั้นจึงต้องมีเฮดเดอร์เพื่อบอกรายละเอียดของข้อมูลในบล็อกข้อมูลแนบไปด้วย
- /path/file คือ ชื่อโปรแกรม CGI ในเซิร์ฟเวอร์ที่จะทำหน้าที่รับข้อมูลไปประมวลผล
- ข้อความตอบสนองที่เซิร์ฟเวอร์จะส่งกลับให้ไคลเอนต์ จะได้มาจากกการทำงานของโปรแกรม CGI ในเซิร์ฟเวอร์ ดังนั้น CGI ที่รับข้อมูลไปจึงต้องทำหน้าที่ส่งข้อความตอบกลับให้ไคลเอนต์

โดยส่วนใหญ่การส่งข้อมูลจากฟอร์มในเว็บเพจไปประมวลผลด้วย CGI ในเซิร์ฟเวอร์จะใช้เมธอดนี้มากที่สุด ความจริงแล้วการส่งข้อมูลไปยังเซิร์ฟเวอร์สามารถใช้เมธอด GET ก็ได้ แต่มีข้อจำกัดเรื่องความยาวของข้อมูลที่จะส่งไปให้เซิร์ฟเวอร์ ถ้าใช้เมธอด POST เพื่อร้องขอส่งข้อมูลไปยังเซิร์ฟเวอร์ เฮดเดอร์ที่ชื่อ Content-Type จะถูกกำหนดให้เป็น application/x-www-form-urlencoded เพื่อบอกแก่เซิร์ฟเวอร์ว่าข้อมูลที่ส่งไปให้มีการเข้ารหัส และเฮดเดอร์ Content-Length จะใช้สำหรับบอกความยาวของข้อมูลที่เข้ารหัสแล้ว เพราะข้อมูลที่กรอกผ่านอินพุต

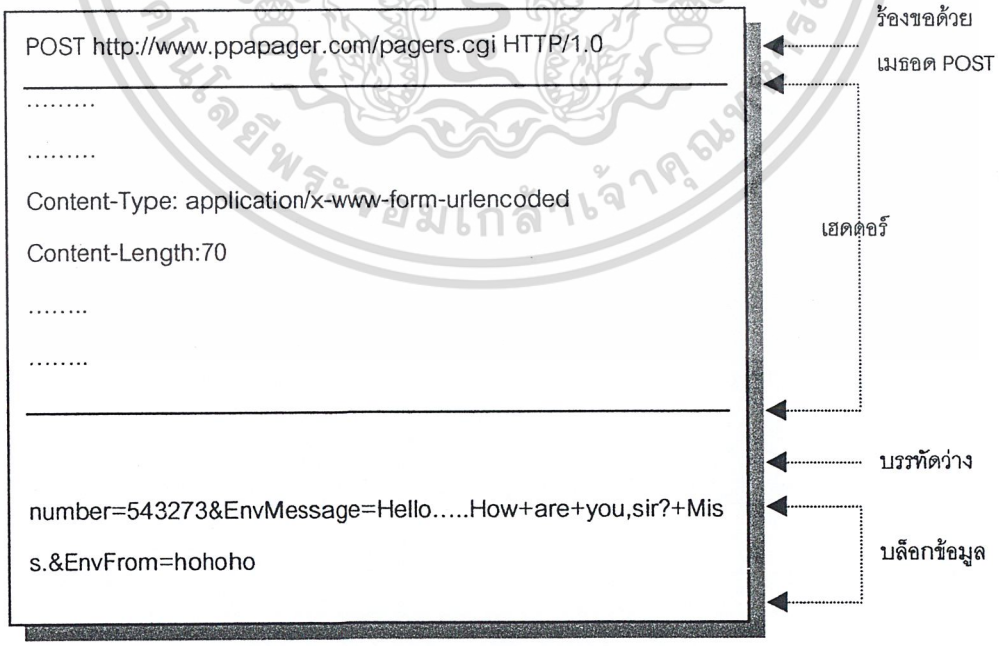
ในฟอร์มจะถูกเว็บบราวเซอร์เข้ารหัสก่อนส่งเสมอ การเข้ารหัสนี้เรียกว่า URL-encoded (ดูการเข้ารหัสในรูปที่ 2.6) ซึ่งมีรายละเอียดดังนี้

- แปลงตัวอักษรหรือเครื่องหมายบางตัวให้อยู่ในรูป %xx โดยที่ xx จะเป็นค่ารหัสแอสกีของตัวอักษรนั้น ตัวอักษรที่ต้องมีการแปลง เช่น =, &, % และ + เพราะเป็นเครื่องหมายที่ใช้เป็นตัวแบ่งแยกข้อมูลที่จะส่งไปให้เซิร์ฟเวอร์ ดังรูปที่ 2.6
- เปลี่ยนช่องว่าง (Space) ทุกตัวเป็นเครื่องหมายบวก (+)
- รวมชื่อตัวแปรและค่าตัวแปรเข้าด้วยกัน โดยคั่นกลางด้วยเครื่องหมาย = และคั่นระหว่างตัวแปรด้วยเครื่องหมาย &

อักขระ	รหัส (%xx)
%	%25
&	%26
‘	%27
+	%2B
=	%3D
?	%3F

รูปที่ 2.6 URL-encoded

ตัวอย่างข้อความร้องขอด้วยเมธอด POST ที่เว็บบราวเซอร์สร้างขึ้น เพื่อส่งข้อมูลจากฟอร์มในเว็บเพจไปให้แก่ CGI ที่ชื่อ pagerkara.cgi ในไดเรกทอรี / ของเซิร์ฟเวอร์ http://www.ppapager.com รับไปทำงาน ดังรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างข้อความร้องขอด้วยเมธอด POST

2.3.1.1.2 ข้อความตอบสนอง (respond) และสถานะการทำงาน

โพรโตคอล HTTP ได้กำหนดรหัสแสดงสถานะการทำงานของโพรโตคอลไว้ โดยแบ่งกลุ่มของรหัสสถานะออกไว้เป็น 5 กลุ่ม ดังแสดงในรูปที่ 2.8 ดังนี้

รหัสสถานะ	ประเภท	รายละเอียด
100-199	Informational	เป็นรหัสสถานะที่เปิดให้โปรแกรมประยุกต์ต่าง ๆ กำหนดใช้งานได้เอง
200-299	Successful	กลุ่มรหัสที่แสดงว่าการทำงานสำเร็จ
300-399	Redirection	กลุ่มรหัสนี้จะใช้ภายในโพรโตคอลHTTP เอง โดยเป็นการทำงานที่ต่อเนื่องมาจากโพรเซสก่อนหน้านี้ ซึ่งไคลเอนต์เป็นผู้สั่งงาน
400-499	Client Error	ใช้แสดงปัญหาที่เกิดขึ้นทางฝั่งไคลเอนต์
500-599	Server Error	ใช้แสดงปัญหาที่เกิดขึ้นทางฝั่งเซิร์ฟเวอร์

รูปที่ 2.8 กลุ่มของรหัสสถานะการทำงานของโพรโตคอล HTTP

รหัสแสดงสถานะในแต่ละตัว จะนำหน้าด้วยตัวเลข 3 หลัก และตามด้วยตัวอักษร ซึ่งรหัสในกลุ่ม 100-199 จะเปิดกว้างให้ผู้ที่พัฒนาโปรแกรมประยุกต์สามารถกำหนดค่าขึ้นมาใช้งานได้เอง ส่วนรายละเอียดของรหัสในกลุ่มอื่น ๆ จะแสดงในรูปที่ 2.9 ดังต่อไปนี้

รหัสสถานะ	รายละเอียด
100 Continue (1.1)	ใช้ในกรณีที่บราวเซอร์อยู่ในระหว่างส่ง Request แต่ยังส่งไม่หมด แต่เซิร์ฟเวอร์ต้องการให้ทราบว่าได้รับ Request แล้วให้ส่งส่วนที่เหลือต่อไป
101 Switching Protocol (1.1)	ใช้ร่วมกับเฮดเดอร์ Upgrade กรณีที่ต้องการเปลี่ยนไปใช้โพรโตคอลอื่นที่ความสามารถสูงกว่า เช่น HTTP/2.0 ซึ่งอาจจะมีในอนาคต
200 OK	การทำงานสำเร็จเรียบร้อย
201 Created	คำสั่ง POST ทำงานเสร็จสมบูรณ์
202 Accepted	ได้รับคำสั่งให้ทำงานเรียบร้อย แต่ไม่ต้องมีการตอบกลับ
203 Non-Authoritative(1.1)	การร้องขอประสบผลสำเร็จ
204 No Content	ทำงานตามคำสั่งเรียบร้อย แต่ไม่ต้องการแสดงข้อความใด ๆ บนหน้าจอ
205 Reset Content (1.1)	เซิร์ฟเวอร์ได้รับข้อมูลเรียบร้อยแล้ว และบอกให้บราวเซอร์ลบข้อความที่กรอกในแบบฟอร์มเดิมออก เพื่อนสะดวกในการกรอกข้อมูลถัดไป
206 Partial Content (1.1)	เซิร์ฟเวอร์ได้รับข้อมูลบางส่วนเรียบร้อยแล้ว
300 Multiple Choice	ถ้าค้นหาและพบแหล่งข้อมูลที่ต้องการหลายแห่ง เซิร์ฟเวอร์จะตอบกลับไปทั้งหมดเพื่อให้ไคลเอนต์สามารถเลือกแหล่งข้อมูลที่ต้องการเองได้
301 Moved Permanently	URL ที่ร้องขอได้ถูกย้ายไปที่อื่นแล้ว ดังนั้นการร้องขอใช้งาน กับ URL จะต้องเปลี่ยนเป็นแอดเดรสใหม่
302 Moved Temporarily	URL ที่ร้องขอมาได้ถูกย้ายไปที่อื่นชั่วคราว
303 See Other (1.1)	ใช้กรณีที่ต้องการบอกให้ทราบว่าสิ่งที่ต้องการอยู่ใน URI อื่น ซึ่งบราวเซอร์สามารถใช้ GET เพื่อเรียกดูเอกสารนั้น ๆ ได้
304 Not Modify	ใช้แสดงสถานะเมื่อใช้คำสั่ง GET ที่กำหนดเงื่อนไขเฉพาะเว็บไซต์ที่มีการเปลี่ยนแปลง ส่วนเว็บไซต์ที่ไม่มีการเปลี่ยนแปลงจะแสดงด้วยสถานะนี้
305 Use Proxy (1.1)	บอกให้บราวเซอร์ทราบว่าเอกสารที่ต้องการมีอยู่ใน Proxy ซึ่ง URL ของ Proxy จะกำหนดใน Location
400 Bad Request	คำสั่งจากไคลเอนต์ไม่ถูกต้อง
401 Unauthorized	ปฏิเสธการทำงานจากไคลเอนต์ที่ไม่ได้รับอนุญาต
403 Forbidden	เซิร์ฟเวอร์ไม่อนุญาตให้ใช้งาน หรือไคลเอนต์มีสิทธิ์ในการใช้งานเพียงพอ
404 Not Found	ไม่พบเว็บเซิร์ฟเวอร์ตาม URL ที่กำหนด
405 Method Not Allowed (1.1)	Method ที่ใช้ ไม่ได้รับอนุญาต กรณีนี้เซิร์ฟเวอร์จะระบุ Allow เพื่อบอกให้ทราบว่าอนุญาตให้ใช้ Method ใดบ้าง
406 Not Acceptable (1.1)	ข้อมูลที่ต้องการเป็นข้อมูลที่บราวเซอร์ไม่สามารถเข้าใจได้ เนื่องจากไม่อยู่ในรายการ Accept ที่ระบุใน Request Header

รูปที่ 2.9 รหัสสถานะ

รหัสสถานะ	รายละเอียด
407 Proxy Authentication (Unauthorized)Request (1.1)	เหมือนกับ 401 แต่ต้องได้รับอนุญาตจาก Proxy จะระบุเฮดเดอร์ Proxy Authenticate ให้ทราบซึ่งบราวเซอร์สามารถส่ง Requestใหม่โดยระบุเฮดเดอร์ Proxy-Authorization ด้วย
408 Request Timeout (1.1)	บราวเซอร์ไม่ส่ง Request ตามเวลาที่เซิร์ฟเวอร์รอได้
409 Conflict (1.1)	บราวเซอร์ส่งข้อมูลที่มีความหมายขัดแย้งกันเอง
410 Gone (1.1)	เอกสารที่ต้องการไม่ได้อยู่บนเซิร์ฟเวอร์แล้ว
411 Length Required (1.1)	เซิร์ฟเวอร์ต้องการให้ระบุ Content-Length ด้วย
412 Precondition Failed (1.1)	เงื่อนไขบางอย่างที่กำหนดใน Request Header ตกไป
413 Request Entity Too Large (1.1)	ข้อมูลที่ส่งมามีขนาดใหญ่เกินกว่าที่เซิร์ฟเวอร์จะรองรับได้
414 Request URI Too Long (1.1)	ค่า URL ที่ระบุ ยาวเกินไป
415 Unsupport Media Type (1.1)	ไม่รองรับการทำงานของ Media Type
500 Internal Server Error	เซิร์ฟเวอร์มีปัญหา
501 Not Implemented	เซิร์ฟเวอร์ไม่รองรับคำสั่งที่สั่งไป
502 Bad Gateway	Proxy Server รับคำสั่งที่ไม่ถูกต้องจากเว็บเซิร์ฟเวอร์
503 Service Unavailable	เซิร์ฟเวอร์กำลังทำงานอื่นอยู่ ไม่สามารถให้บริการได้ในขณะนี้
504 Gateway Timeout (1.1)	ในขณะทำหน้าที่เป็น Gateway หรือ Proxy ไม่ได้รับข้อมูลตอบจากเซิร์ฟเวอร์ปลายทางในเวลาที่กำหนด
505 HTTP Version not Supported (1.1)	เซิร์ฟเวอร์ไม่รองรับการทำงานของ HTTP เวอร์ชั่นนั้น ๆ

รูปที่ 2.9 รหัสสถานะ (ต่อ)

2.3.2.1.2 ส่วนเฮดเดอร์ย่อย

เฮดเดอร์ย่อยเป็นส่วนที่ใช้บอกรายละเอียดต่าง ๆ ของข้อมูล ทั้งการร้องขอและตอบสนอง โดยมีลักษณะเป็นข้อความธรรมดา ซึ่งมีรูปแบบการเขียนดังนี้

```
Header-name: Value
```

รายละเอียดปลีกย่อยของเฮดเดอร์ ได้แก่

- เฮดเดอร์อาจมีหลายประการ แต่ท้ายเฮดเอร์แต่ละรายการต้องปิดด้วยรหัสลงบรรทัดใหม่
- Header-name หรือชื่อของเฮดเดอร์จะพิมพ์ตัวเล็กหรือใหญ่ก็ได้ ไม่มีผลต่อการตีความหมาย
- หลังเครื่องหมาย : ของเฮดเดอร์แต่ละรายการอาจเป็นช่องว่าง (Space) หรือ แท็บ (Tab) ก็ได้

- เฮดเดอร์รายการใดที่ขึ้นต้นด้วยช่องว่างหรือแท็บ จะเสมือนว่าเป็นส่วนหนึ่งของเฮดเดอร์รายการก่อนหน้านี้ 1 บรรทัด

ใน HTTP เวอร์ชัน 1.0 กำหนดให้มีเฮดเดอร์ได้ถึง 16 รายการ แต่อาจจะไม่มีแม้แต่รายการเดียวเลยก็ได้ ส่วน HTTP เวอร์ชัน 1.1 กำหนดได้ 46 รายการ แต่ต้องมีเฮดเดอร์อย่างน้อย 1 รายการ คือ Host: เพื่อบอกชื่อโฮสต์และโดเมนเนม ในที่นี้จะขอกล่าวเพียงเฮดเดอร์ของ HTTP เวอร์ชัน 1.0 ซึ่งมีด้วยกันอยู่ 16 รายการ ดังแสดงในรูปที่ 2.10

เฮดเดอร์ย่อย	รายละเอียด
1. Allow	กำหนดเมธอดที่สนับสนุน จุดประสงค์ของข้อมูลนี้เพื่อเจาะจงเมธอดการร้องขอจากไคลเอนต์ โดยจะไม่สนับสนุนเมธอด POST
2. Authorization	สำหรับผู้ที่ต้องการการรับรอง (authentication) ด้วยตัวเองจากเซิร์ฟเวอร์ (ซึ่งอาจไม่จำเป็น) หลังจากได้รับการตอบสนองด้วยรหัสสถานะ 401 (Unauthorized) ควรจะมีการเพิ่มเฮดเดอร์นี้เข้าไปด้วย
3. Content-Encoding	ระบุการเข้ารหัสของข้อมูล
4. Content-Length	ใช้ระบุขนาดของข้อมูลในบล็อกข้อมูลมีหน่วยเป็นไบต์ (byte) เพื่อที่ผู้รับจะได้ทราบว่ามีข้อมูลส่งมาให้กี่ไบต์
5. Content-Type	ใช้ระบุชนิดของข้อมูลในบล็อกข้อมูลว่าเป็นข้อมูลประเภทไหน เช่น หากเป็นเอกสารแบบ HTML จะต้องระบุเป็น text/html ถ้าเป็นไฟล์รูปกราฟิกแบบ gif ก็ต้องระบุเป็น image/gif เป็นต้น แต่สำหรับเว็บบราวเซอร์รุ่นใหม่ ๆ แล้ว หากข้อมูลที่ได้รับไม่มีการระบุว่าเป็นประเภทไหนแล้ว เว็บบราวเซอร์จะถือว่าเป็นประเภท text/html เสมอ
6. Date	ข้อมูลส่วนนี้ใช้ระบุวันเวลาที่ข้อมูลการร้องขอถูกส่งมา
7. Expires	เป็นเฮดเดอร์ในส่วนการตอบสนองของเซิร์ฟเวอร์ใช้กำหนดวันที่หมดอายุของไฟล์ที่สงไปให้ไคลเอนต์ รายการนี้สามารถใช้ในทางเทคนิคเพื่อป้อนการเก็บไฟล์ไว้ในแคช (Cache) จากเว็บบราวเซอร์อย่าง Navigator ได้ โดยระบุวันที่ใน Expires: ให้ย้อนจากวันเวลาปัจจุบันนาน ๆ เมื่อเว็บบราวเซอร์รับไฟล์ไปก็จะเข้าใจว่าไฟล์นี้หมดอายุแล้ว ถึงแม้จะนำเนื้อหาไปแสดงในวินโดว์เว็บบราวเซอร์แต่จะไม่เก็บไว้ในแคช ทำให้ทางเซิร์ฟเวอร์มั่นใจได้ว่า ทุกครั้งที่ไคลเอนต์ร้องขอไฟล์จะต้องวิ่งมาขอจากเซิร์ฟเวอร์ใหม่ทุกครั้ง ถึงแม้จะเป็นการร้องขอไฟล์เดิม ๆ และทางเซิร์ฟเวอร์ไม่มีการอัพเดตก็ตาม
8. From	เป็นเฮดเดอร์ในส่วนการร้องขอของไคลเอนต์ จะประกอบด้วย e-mail address ของผู้ใช้ที่ควบคุมการส่งข้อมูลการร้องขอ

รูปที่ 2.10 รายละเอียดของเฮดเดอร์ย่อยของโพรโตคอล HTTP

เฮดเดอร์ย่อย	รายละเอียด
9. If-Modified-Since	เป็นเฮดเดอร์ในส่วนการร้องขอของไคลเอนต์ ถูกใช้ร่วมกับเมธอด GET เพื่อใช้ในการกำหนดเงื่อนไขบอกแก่เซิร์ฟเวอร์ว่า ถ้าไฟล์ที่ร้องขอไปมีการแก้ไขหลังจากวันที่ได้ระบุในเฮดเดอร์นี้ เซิร์ฟเวอร์จึงค่อยส่งไฟล์นั้นมาให้ แต่ถ้ายังไม่ได้มีการแก้ไขหลังช่วงวันที่ระบุ เซิร์ฟเวอร์ไม่ต้องส่งไฟล์นั้นมาให้ เพียงแต่เซิร์ฟเวอร์รหัสสถานะตอบสนองมาเป็น 304 (not Modified)แทน
10. Last-Modified	ข้อมูลนี้ใช้ระบุวันเวลาครั้งล่าสุดที่มีการ modify ข้อมูลนั้น ซึ่งจะต้องบอกวันที่และเวลาในรูปแบบของเวลามาตรฐาน GMT
11. Location	เป็นเฮดเดอร์ส่วนการตอบสนองของเซิร์ฟเวอร์ที่ใช้แจ้งแหล่งที่อยู่ของข้อมูลที่ไคลเอนต์ต้องการ สำหรับในรหัสสถานะการตอบสนองที่ 3xx Location จะชี้ URL ที่ใช้สำหรับรีไดเรก (redirect)
12. Pragma	ข้อมูลในส่วน Pragma นี่ถูกใช้รวมกับ implementation-specific derictives ที่สามารถนำมาใช้กับผู้รับตลอดสายการร้องขอ/ตอบสนอง โดยข้อมูล prama directoves ทั้งหมดจะกำหนดการกระทำจากในมุมมองของโพรโตคอล
13. Referer	เป็นเฮดเดอร์ในส่วนของการร้องขอของไคลเอนต์ เป็น URI ของแหล่งที่มาของการร้องขอ ข้อมูลส่วนนี้จะทำให้เซิร์ฟเวอร์สร้างรายการการย้อนหลัง (lists of back-links) การล็อกอินเข้าระบบ การจัดการ Cache และอื่น ๆ มันยังส่งผลให้ ข้อมูลในส่วน Referer นี้จะต้องไม่ถูกส่งมาถ้าไม่ได้มาจากแหล่งที่มี URI ของตัวเอง ตัวอย่างเช่น จาก คีย์บอร์ดของผู้ใช้เอง
14. Server	เป็นเฮดเดอร์ในส่วนการตอบสนองจากเซิร์ฟเวอร์ ประกอบด้วยข้อมูลเกี่ยวกับซอฟต์แวร์ซึ่งทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ โดยมีรูปแบบคือ Program-name/x.xx

รูปที่ 2.10 รายละเอียดของเฮดเดอร์ย่อยของโพรโตคอล HTTP (ต่อ)

2.3.2.2 ข้อมูลที่ต้องการรับ-ส่ง

จากรูปที่ 2.3 ในส่วนสุดท้าย ซึ่งต่อจากส่วนของเฮดเดอร์ย่อยของเฮดเดอร์ HTTP จะเป็นส่วนของบล็อกข้อมูล ซึ่งเป็นส่วนของข้อมูลที่เราต้องการส่งจริง อาจจะเป็น HTML file หรือ Text file หรือข้อมูลชนิดอื่นๆ

2.3.3 การเขียนโปรแกรมเพื่อการติดต่อสื่อสารผ่านซ็อกเก็ตในภาษาจาวา

ต่อไปนี้จะเป็นคลาสในภาษาจาวาที่ใช้ในการติดต่อสื่อสารผ่านซ็อกเก็ต

คลาส InetAddress

ใช้สำหรับหาค่า IP แอดเดรสของ Internet host จากชื่อแอดเดรสที่มี คลาสนี้ไม่มีคอนสตรัคเตอร์ อินสแตนท์ของคลาสนี้จะถูกสร้างมาจากเมธอดสแตติกเหล่านี้

- getLocalHost() throws UnknownHostException

  เมธอดนี้จะส่งอ๊อบเจคท์ของคลาส InetAddress ของเครื่องคอมพิวเตอร์ที่กำลังใช้งานอยู่กลับมา ในบางกรณี เช่นกรณีที่เครื่องอยู่ภายในการดูแลของไฟร์วอลล์ จะรีเทิร์น loopback Address ซึ่ง เท่ากับ 127.0.0.1

- getByName(String host) throws UnknownHostException

  เมธอดนี้จะส่งอ๊อบเจคท์ของคลาส InetAddress ของโฮสต์ที่ระบุ (host) กลับมาโดย host อาจระบุด้วยชื่อ (www.kmitl.ac.th) หรือด้วย IP Address (161.246.10.21)

- InetAddress[] getAllByName(String host) throws UnknownHostException

  เมธอดนี้จะส่งอาเรย์ของอ๊อบเจคท์ของคลาส InetAddress ของ IP Address ทั้งหมดของโฮสต์ที่ระบุ กลับมาโดยปกติในบางเว็บไซต์ที่มีอัตราการเข้าถึงข้อมูลสูงมักจะมีการขอ IP Address หลายๆอันสำหรับชื่อๆหนึ่งเพื่อไว้กระจายการทำงานให้กับหลายๆเครื่องได้

คลาส Socket

ซ็อกเก็ตจะถูกสร้างขึ้นสำหรับการติดต่อใน TCP network ไคลเอนต์ใช้คลาสนี้สำหรับสร้างช่องทางติดต่อระหว่างไคลเอนต์กับเครื่องปลายทางโดยอัตโนมัติ หากไม่สำเร็จ Exception จะถูกโยนออกมา ซ็อกเก็ตแต่ละตัวที่สร้างขึ้นจะต้องใช้พอร์ตในการติดต่อสื่อสาร ไม่ว่าจะเป็นซ็อกเก็ตในไคลเอนต์หรือในเซิร์ฟเวอร์ก็ตาม โดยนอกเหนือจากต้องระบุโฮสต์ปลายทางแล้ว จะต้องกำหนดพอร์ตให้เสมอ พอร์ตที่ว่านี้เป็นพอร์ตทางซอฟต์แวร์ ไม่ใช่พอร์ตที่เป็นฮาร์ดเวอร์ หลังตัวเครื่องคอมพิวเตอร์หมายเลขพอร์ตของเซิร์ฟเวอร์ที่เราจะติดต่อด้วยนั้นจำเป็นต้องใช้ตามที่เซิร์ฟเวอร์กำหนดไว้ โดยปกติโปรแกรมเว็บเซิร์ฟเวอร์จะถูกกำหนดอยู่ที่พอร์ตหมายเลข 80 แต่ในเซิร์ฟเวอร์หนึ่งตัวสามารถรันโปรแกรมเว็บเซิร์ฟเวอร์ได้หลายตัว โปรแกรมเว็บเซิร์ฟเวอร์ตัวที่สองอาจกำหนดให้รออยู่ที่พอร์ตหมายเลข 8080 หรือพอร์ตอื่นก็ได้ที่ยังไม่มีโปรแกรมหรือซ็อกเก็ตใดยึดไปใช้งาน ซึ่งกำหนดให้อยู่ในช่วง 1024-65535 สำหรับพอร์ตช่วงหมายเลข 1 ถึง 1023 นั้น ทางเซิร์ฟเวอร์จะสงวนสำหรับโปรแกรมและการทำงานเซิร์ฟเวอร์เอง

สำหรับภาษาจาวา เราสามารถสร้างซ็อกเก็ต โดยใช้คลาส Socket นี้ ซึ่งประกอบด้วย 6 คอนสตรัคเตอร์ ดังนี้

- protected Socket()

  คอนสตรัคเตอร์นี้ใช้สร้างออบเจ็คซ็อกเก็ตที่ยังไม่มีการเชื่อมต่อ ซึ่งสร้างขึ้นเพื่อรอการติดต่อมาจากไคลเอนต์ของเซิร์ฟเวอร์

- protected Socket(SocketImpl impl)

คอนสตรัคเตอร์นี้ใช้สำหรับ customer socket implementation เท่านั้น

- Socket(String host, int port) thows IOException

คอนสตรัคเตอร์นี้ใช้สร้างซ็อกเก็ตติดต่อไปยังโฮสต์ปลายทางที่ระบุเป็นสตริง(String host) และใช้พอร์ตที่ระบุมา (int port)

- Socket(InetAddress address, int port) throws IOException

คอนสตรัคเตอร์นี้ใช้สร้างซ็อกเก็ตติดต่อไปยังโฮสต์ปลายทางที่ระบุเป็นคลาส InetAddress (InetAddress address) และใช้พอร์ตที่ระบุมา (int port)

- Socket(String host, int port, InetAddress localAddr, int localPort) throws IOException

คอนสตรัคเตอร์นี้ใช้สร้างซ็อกเก็ต โดยมีการส่งค่าแอดเดรสต้นทางไปด้วย ใช้ได้เฉพาะเครื่องที่มีเน็ตเวิร์กอินเตอร์เฟสหลายๆอัน หากไม่ได้ระบุก็จะกำหนดเป็น default local address และหาก localPort เป็น 0 ก็จะสุ่มหาหมายเลขพอร์ตที่ใช้งานได้มา

- Socket (InetAddress address, int port, InetAddress localAddr, int localPort) throws IOException

คล้ายกับคอนสตรัคเตอร์ที่ 5 หากแต่ชนิดของการระบุโฮสต์เป็นคลาส InetAddress ไม่ใช่ String

สำหรับเมธอดในคลาส Socket นี้ มีดังนี้

1. InputStream getInputStream() throws IOException

จะส่งข้อมูลออกมาเป็น InputStream ใช้รับข้อมูลบน Stream-based communications โดยข้อมูลที่ได้นั้นถูกสร้างขึ้นมาจากโฮสต์ปลายทาง

2. OutputStream getOutputStream() throws IOException

จะส่งข้อมูลออกมาเป็น OutputStream มักใช้ OutputStreamWriter ห่อหุ้มอีกที เนื่องจากการอ่านเขียนผ่านสายข้อมูล (Stream) ความจะมีบัฟเฟอร์ หรืออาจจะใช้เป็น PrintStream จะมีความสะดวกในการใช้งานมากกว่า

3. Void close() throws IOException

ใช้ปิดซ็อกเก็ต ควรจะมีการปิดซ็อกเก็ตทุกครั้งเมื่อมีการเปิดซ็อกเก็ตขึ้นมา เพื่อเป็นการหยุดการเชื่อมต่อ แต่โดยทั่วไปทางเซิรฟเวอร์จะทำการปิดซ็อกเก็ตอยู่แล้ว เมื่อสิ้นสุดการส่งข้อมูล

4. InetAddress getInetAddress()

จะส่งค่า IP Address ของโฮสต์ปลายทาง กลับมาให้

5. int getPort()

จะส่งหมายเลขพอร์ตของโฮสต์ปลายทาง

6. InetAddress getLocalAddress()

จะส่งค่าแอดเดรสของเครื่องผู้ส่ง ณ ซ็อกเก็ตนี้ กลับมา

7. int getLocalPort()

   จะส่งค่าพอร์ตของเครื่องผู้ส่ง ณ ซ็อกเก็ตนี้ กลับมา

8. void setSoTimeout(int timeout) throws SocketException

   ใช้กำหนดเวลาในการติดต่อ (time out) มีหน่วยเป็นมิลลิวินาที

9. int getSoTimeout() throws SocketException

   จะส่งค่าเวลา time out กลับมา

10. void setTcpNoDelay(boolean on) throws SocketException

    ใช้กำหนดอัลกอริทึมการติดต่อสื่อสาร หากกำหนดให้เป็นจริง จะเป็นการกำหนดให้การติดต่อสื่อสารใช้ Nagle's algorithm ซึ่งเป็นวิธีในการเพิ่มประสิทธิภาพให้กับการส่งข้อมูล โดยการค่อยๆเก็บข้อมูลไว้ จนกระทั่งมีข้อมูลที่ถูกบัฟเฟอร์เพียงพอแล้วจึงส่งออกไป

11. boolean getTcpNoDelay() throws SocketException

    จะส่งค่าการกำหนดอัลกอริทึมการติดต่อสื่อสารว่าใช้ Nagle's algorithm หรือไม่

12. void setSoLinger(boolean on, int val) throws SocketException

    ใช้กำหนดค่าเวลาที่มากที่สุดที่จะรอ linger

13. int getSoLinkger() throws SocketException

    จะส่งค่าเวลาที่มากที่สุดที่จะรอ linger กลับมา

14. void setSendBufferSize(int size) throws SocketException

    ใช้กำหนดขนาดของบัฟเฟอร์ในการส่งข้อมูล

15. int getSendBufferSize() throws SocketException

    จะส่งค่าขนาดของบัฟเฟอร์ในการส่งข้อมูล กลับมา

16. void setReceiveBufferSize(int size) throws SocketException

    ใช้กำหนดขนาดของบัฟเฟอร์ในการรับข้อมูล

17. int getReceiveBufferSize() throws SocketException

    จะส่งค่าขนาดของบัฟเฟอร์ในการส่งข้อมูล กลับมา

18. static void setSocketImplFactory(SocketImplFactory factory) throws IOException

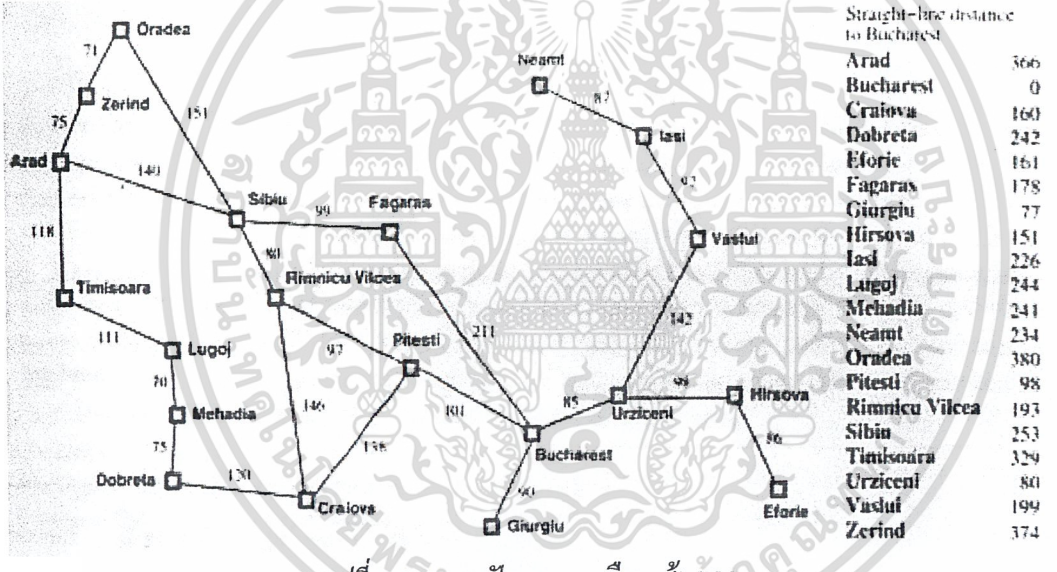
    เมธอดนี้เป็นเมธอดสแตติกใช้ในการตั้งค่าในการใช้งานซ็อกเก็ตของ JVM ซึ่งเมธอดนี้จะถูกเรียกครั้งแรกเพียงครั้งเดียวและถูกรักษาด้วย Security Manager

ในการติดต่อสื่อสารบนโพรโตคอล TCP เริ่มด้วยการสร้างซ็อกเก็ต แล้วใช้เมธอด getInputStream() และเมธอด getOutputStream() ในการรับ-ส่งข้อมูลระหว่างกัน ดังนั้นทางไคลเอนต์และเซิร์ฟเวอร์ก็ต้องมีตัวแปรชนิด InputStream และ OutputStream มารองรับด้วย

2.4 Heuristic Function

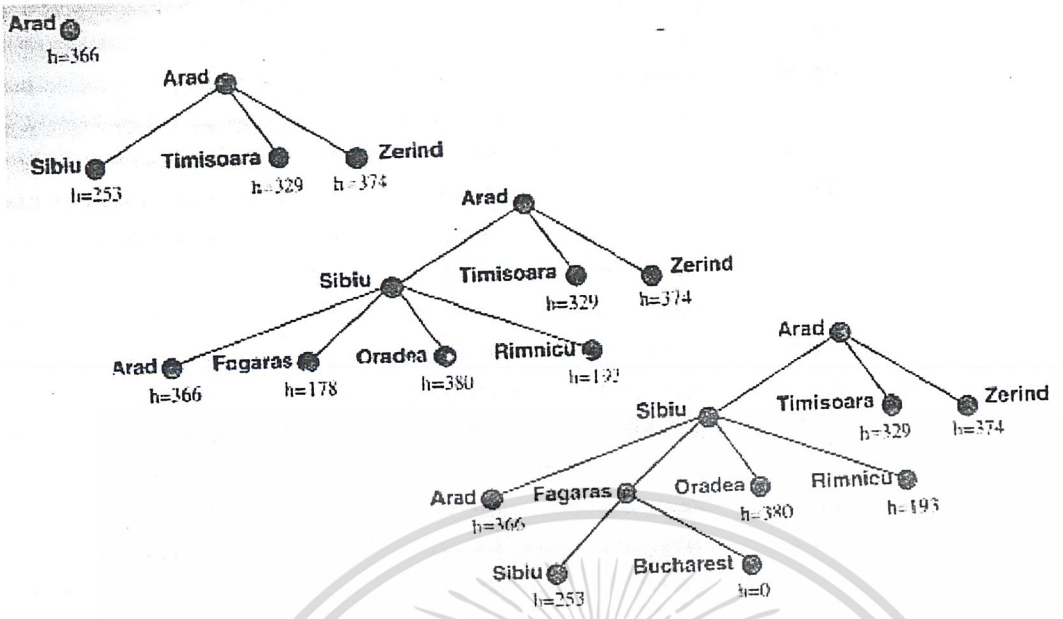
Heuristic Function เป็นหนึ่งในกระบวนการค้นหาที่ใช้ความรู้ที่เกี่ยวกับข้อมูลที่มีอยู่เข้ามาช่วยประมวลผลเพื่อเลือกข้อมูลที่ดีและเหมาสมที่สุด โดยการนำเอาความรู้เกี่ยวกับข้อมูลนั้นในหลาย ๆ แง่มุมมารวมกันและเปรียบเทียบ เพื่อตัดสินความเหมาะสมของข้อมูลนั้น

Heuristic มาจากกริยาในภาษากรีกว่า Heuriskein แปลว่า ค้นพบ เช่นเดียวกับที่อาร์คีมีดีสนักคณิตศาสตร์ในยุคก่อนที่ค้นพบแรงลอยตัวในขณะที่อาบน้ำอยู่และได้กล่าวว่า Heureka ซึ่งแปลว่า ได้ค้นพบแล้ว Heuristic Function เป็นการค้นหาข้อมูลที่พัฒนามาจาก Greedy Search ซึ่งจะค้นหาหนทางที่จะบรรลุเป้าหมายโดยการสูญเสียที่น้อยที่สุด เช่น ปัญหาการเลือกเส้นทาง Greedy Search จะรวบรวมเส้นทางจากจุดเริ่มต้นไปยังเมืองต่อไปแล้วเลือกเมืองที่ห่างจากเมืองเป้าหมายน้อยที่สุดไปเรื่อย ๆ จนกระทั่งถึงเมืองเป้าหมาย แต่ในความเป็นจริงแล้วเส้นทางนั้นอาจไม่ใช่เส้นทางที่สั้นที่สุดเสมอไป ดังจะแสดงตัวอย่างในรูปที่ 2.11 เป็นแผนที่และระยะทางระหว่างเมืองแต่ละเมือง



รูปที่ 2.11 แสดงปัญหาการเลือกเส้นทาง

หากต้องการเดินทางจากเมือง Arad ไปยังเมือง Bucharest โดยใช้วิธี Greedy Search ดังแสดงในรูปที่ 2.12 จะได้เส้นทางเป็น Arad -> Sibiu -> Faragas -> Bucharest รวมได้ระยะทาง 450 กิโลเมตร ซึ่งมากกว่าอีกทางหนึ่งที่ใช้วิธี Heuristic Search ดังแสดงในรูปที่ 2.13 ซึ่งจะได้เส้นทางเป็น Arad -> Sibiu -> Rimnicu Vilcea -> Pitesti -> Bucharest ซึ่งใช้ระยะทางเพียง 418 กิโลเมตรเท่านั้น



รูปที่ 2.12 แสดงการทำงานของ Greedy Search

เราใช้ Heuristic Function พัฒนากระบวนการหาด้วยการเพิ่มอีกเงื่อนไขหนึ่งที่จะมาประมวลผลเพื่อเลือกเส้นทางด้วยคือ ระยะทางระหว่างเมือง เมื่อนำทั้ง 2 เงื่อนไขมารวมกันจะทำให้ได้ผลลัพธ์เป็นเส้นทางที่สั้นที่สุดอย่างแท้จริง เงื่อนไขที่นำมาประมวลผลนั้นจะต้องถูกต้องและมีประสิทธิภาพ เราเรียกเงื่อนไขนั้นว่า Admissible Heuristic สำหรับตัวอย่างปัญหาการเลือกเส้นทางนี้ เงื่อนไขที่เป็น Admissible Heuristic คือ ระยะห่างจากเมืองเป้าหมาย แต่เพียงเงื่อนไขเดียวนั้นยังไม่เพียงพอกับการเลือกเส้นทางที่ดีที่สุดจึงต้องเพิ่มอีกเงื่อนไขหนึ่ง จึงได้เป็นสูตรว่า

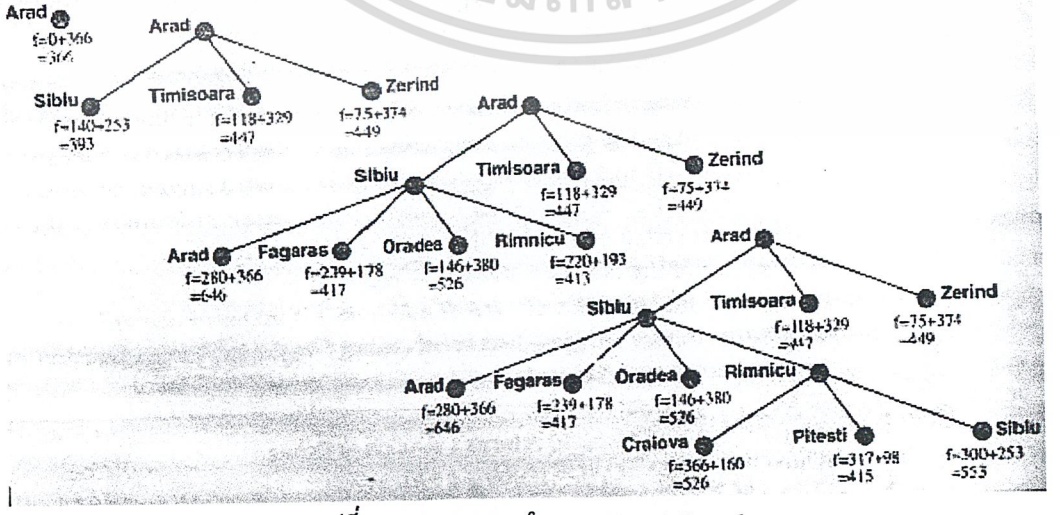
$$F(n) = g(n) + h(n)$$

F(n) คือ สูตรการคำนวณค่าน้อยที่สุดสำหรับการเลือกเส้นทาง

g(n) คือ ระยะทางระหว่างเมือง

h(n) คือ ระยะห่างจากเมืองเป้าหมาย

เราเรียกการค้นหาที่ใช้ F(n) ซึ่งมี h(n) ที่ Admissible Heuristic ในการค้นหาว่า A* Search



รูปที่ 2.13 แสดงการทำงานของ A Search*

2.5 Neural Network

นิวรัลเน็ตเวิร์ก เป็นการจำลองระบบการเรียนรู้ของสมอง แนวความคิดนี้ถูกริเริ่มโดย Ramon Y Cajal (1911) เป็นผู้เสนอแนวความคิดที่ว่าเซลล์ประสาท และโครงสร้างส่วนประกอบของสมองนั้นมี 5-6 ลำดับขั้นการทำงาน ซึ่งมีความเร็วน้อยกว่าการทำงานทางตรรกศาสตร์ของอุปกรณ์ซิลิกอน คือ ชิปซิลิกอน มีความเร็วในการทำงานอยู่ที่ 10^{-9} วินาทีต่อหนึ่งหน่วยการทำงาน อย่างไรก็ตาม สมองนั้นมีการปรับปรุงการทำงานของตัวเองอย่างช้า ๆ ด้วยการดูผลจากการทำงานที่ผ่านมาระหว่างเซลล์ประสาทนั้น ดังนั้นจึงสามารถประมาณได้ว่า มันจะต้องมีลำดับขั้นการทำงานมหาศาลสำหรับ 10^{13} เซลล์ประสาท ซึ่งมีการเชื่อมต่อของแกนเซลล์ประสาทถึง $60*10^{18}$ การเชื่อมต่อ (Shepherd and Korch,1990) นั่นก็คือ สองมีโครงสร้างขนาดใหญ่และซับซ้อนมาก ยิ่งไปกว่านั้น ประสิทธิภาพการทำงานของสมองอยู่ที่ 10^{-16} จูลต่อหน่วยการทำงานต่อวินาที ในขณะที่คอมพิวเตอร์ที่ดีที่สุดในสมัยนั้นมีประสิทธิภาพการทำงานอยู่ที่ 10^{-6} จูลต่อหน่วยการทำงานต่อวินาที

สมองเปรียบได้กับคอมพิวเตอร์ที่มีการทำงานแบบขนาน non-linear และมีความซับซ้อนสูง (information-processing system) ในขณะที่เพิ่งเริ่มต้นมีชีวิต สมองจะมีโครงสร้างและความสามารถในการตั้งกฎเกณฑ์ ความสัมพันธ์ ผ่านสิ่งที่เราเรียกว่า "ประสบการณ์" การพัฒนาลักษณะเฉพาะตัวของสมองจะพัฒนาดีที่สุดในช่วง 1-2 ปีแรก และจะมีการพัฒนาอย่างต่อเนื่องต่อไป และมีอัตราการเติบโตของเซลล์ประสาท 1 ล้านเซลล์ประสาทต่อหนึ่งวินาที

อาจจะนิยามนิวรัลเน็ตเวิร์กในแง่ของอุปกรณ์เครื่องกลได้ว่า

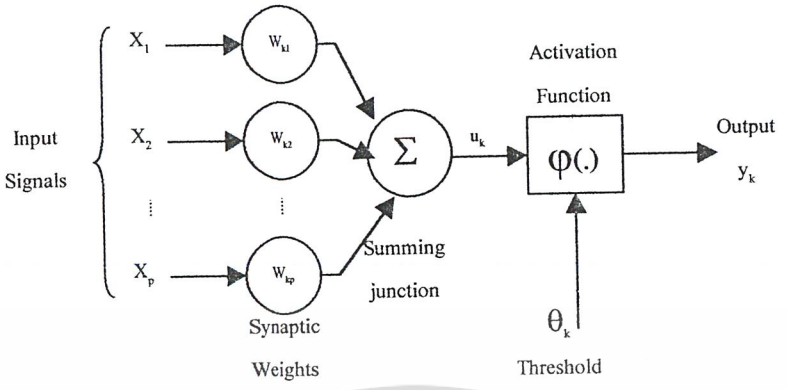
" นิวรัลเน็ตเวิร์กคือระบบประมวลผลแบบขนาน ซึ่งมีการทำงานแบบกระจายขนาดมหึมา ที่มีการเรียนรู้และจดจำสิ่งที่ได้เคยกระทำ เพื่อประโยชน์ในการทำงานครั้งต่อไป ซึ่งมีลักษณะคล้ายคลึงกับสมองอยู่ 2 สิ่ง คือ

1. ความรู้ (Knowledge) จะถูกเรียนรู้โดยเน็ตเวิร์กของระบบผ่านกระบวนการเรียนรู้ (Learning process)
2. ค่าความสัมพันธ์ระหว่างเซลล์ประสาท (Synaptic weights) ถูกนำมาใช้ในการจดจำความรู้

2.5.1 แบบจำลองของเซลล์ประสาท

มีองค์ประกอบ 3 ส่วน ดังแสดงในรูปที่ 2.14 ดังนี้

1. กลุ่มของการเชื่อมต่อระหว่างเซลล์ประสาท ซึ่งแต่ละอันจะมีค่าความสัมพันธ์ระหว่างเซลล์ตัวเองนั้นกับเซลล์อื่นๆ
2. ส่วนรวมค่าสัญญาณอินพุต ค่าที่ได้ขึ้นอยู่กับค่าพลังงานของเซลล์ประสาทนั้นๆ
3. ฟังก์ชันค่าพลังงานกระตุ้น (Activation Function) ใช้สำหรับการจำกัดค่าพลังงานของเซลล์ประสาท หรืออาจเรียกฟังก์ชันนี้ว่า Squashing Function โดยปกติขนาดของค่าพลังงานของเซลล์ประสาทจะกำหนดให้อยู่ในช่วง [0,1] หรือ [-1,1]



รูปที่ 2.14 แบบจำลองของเซลล์ประสาท

สามารถอธิบายเซลล์ประสาท k ในรูปแบบทางคณิตศาสตร์ ได้สมการดังนี้

$$u_k = \sum_{j=1}^{p} w_{kj} x_j \qquad (2.5.1)$$

และ

$$y_k = \varphi(u_k - \theta_k) \qquad (2.5.2)$$

$x_1, x_2, \ldots, x_p$ คือ ค่าสัญญาณอินพุต (Input signal)

$w_{k1}, w_{k2}, \ldots, w_{kp}$ คือ ค่าพลังงานของเซลล์ประสาทที่ k (Synaptic weights)

u_k คือ ผลรวมเชิงเส้น (Linear Combiner)

θ_k คือ ขีดจำกัดของค่าพลังงาน (Threshold)

$\varphi(.)$ คือ ฟังก์ชันค่าพลังงานกระตุ้น (Activation Function)

y_k คือ ค่าสัญญาณผลลัพธ์ (Output signal)

และจะได้ว่า

$$v_k = u_k - \theta_k \qquad (2.5.3)$$

ค่า v_k ที่ได้ ขึ้นอยู่กับว่า θ_k เป็นค่าบวกหรือลบ ความสัมพันธ์ระหว่างระดับการทำงานภายใน (Internal activity level) กับ ค่าความแตกต่างของพลังงานกระตุ้น v_k ของเซลล์ประสาท k และผลรวมเชิงเส้น u_k พิจารณาได้จากรูปที่ ???

θ_k คือตัวแปรภายนอกของเซลล์ประสาท k ดังนั้นจะได้สมการที่เกิดจากการรวมสมการ (1) , (2) และ (3) ได้เป็น

$$v_k = \sum_{j=0}^{p} w_{kj} x_j \qquad (2.5.4)$$

และ

$$y_k = \varphi(v_k) \tag{2.5.5}$$

และกำหนดให้

$$x_0 = -1 \tag{2.5.6}$$

และ

$$w_{k0} = \theta_k \tag{2.5.7}$$

ชนิดของฟังก์ชันค่าพลังงานกระตุ้น

1. Threshold Function

คือกำหนดให้ค่าพลังงานกระตุ้นมีค่าเป็น 1 เมื่อค่าความแตกต่างของพลังงานกระตุ้นมากกว่าหรือเท่ากับ 0 และจะมีค่าเป็น 0 เมื่อค่าความแตกต่างของพลังงานกระตุ้นมีค่าน้อยกว่า 0 ได้สมการ (8)

$$\varphi(v) = \begin{cases} 1 & \text{ถ้า } v >= 0 \\ 0 & \text{ถ้า } v < 0 \end{cases} \tag{2.5.8}$$

และในทำนองเดียวกันจะได้ว่า ค่าสัญญาณผลลัพธ์ ณ เซลล์ประสาทที่ k จะมีค่าเท่ากับ 1 เมื่อค่าความแตกต่างของพลังงานกระตุ้น ณ เซลล์ประสาทที่ k มีค่ามากกว่าหรือเท่ากับ 0 และค่าสัญญาณผลลัพธ์ ณ เซลล์ประสาทที่ k จะมีค่าเท่ากับ 0 เมื่อค่าความแตกต่างของพลังงานกระตุ้น ณ เซลล์ประสาทที่ k มีค่าน้อยกว่า 0 ดังสมการที่ (9)

$$y_k = \begin{cases} 1 & \text{ถ้า } v_k >= 0 \\ 0 & \text{ถ้า } v_k < 0 \end{cases} \tag{2.5.9}$$

โดยที่

$$v_k = \sum_{j=1}^{P} w_{kj} x_j - \theta_k \tag{2.5.10}$$

จากสมการ ค่าพลังงานของเซลล์ประสาท จะเท่ากับ 1 หากว่าค่าอินพุตของเซลล์ประสาทนั้นทั้งหมดมีค่ามากกว่า 0 เรียกรูปแบบนี้ว่า All-or-none property of McCulloch-Pitts model

2. Piecewise-Linear Function

คือกำหนดให้ค่าพลังงานกระตุ้นมีค่าเป็น 1 เมื่อค่าความแตกต่างของพลังงานกระตุ้นมากกว่าหรือเท่ากับ 0.5 , มีค่าเท่ากับค่าความแตกต่างของพลังงานกระตุ้น เมื่อค่าความแตกต่างของพลังงานกระตุ้นมีค่าอยู่ในช่วง –0.5 ถึง 0.5 และจะมีค่าเป็น 0 เมื่อค่าความแตกต่างของพลังงานกระตุ้นมีค่าน้อยกว่า –0.5 ได้สมการ

$$\varphi(v) = \begin{cases} 1 & \text{ถ้า } v >= 0.5 \\ v & \text{ถ้า } -0.5 > v > 0.5 \\ 0 & \text{ถ้า } v <= -0.5 \end{cases} \qquad (2.5.11)$$

เป็นการประมาณ (Approximation) เป็นลักษณะ non-linear มี 2 รูปแบบ คือ

1. ผลรวมเชิงเส้น (linear combiner) จะเพิ่มขึ้น ถ้าพื้นที่การทำงานเชิงเส้นไม่ถึงจุดอิ่มตัว
2. piecewise function จะลดลงเป็น threshold function ถ้าการเพิ่มจำนวนอินพุตขึ้นอย่างรวดเร็ว และไม่จำกัด

3. Sigmoid Function

เป็นฟังก์ชันการเพิ่มค่าที่มีลักษณะนิ่มนวลกว่าฟังก์ชันการทำงานที่ผ่านมา เป็นลักษณะเอ็กซ์โพเนนเชียล ตัวอย่างฟังก์ชันที่เป็น Sigmoid Function คือ Logistic function เขียนได้เป็น

$$\varphi(v) = \frac{1}{1+\exp(-av)} \qquad (2.5.12)$$

a คือ ค่าความชันของ Sigmoid Function ในกรณีที่ค่าความชันนี้เข้าใกล้อนันต์ แล้ว Sigmoid Function จะใกล้เคียง Threshold Function

จากสมการที่ (8), (11) และ (12) มีขอบเขตอยู่ระหว่าง 0 กับ 1 หากกำหนดใหม่เป็นให้อยู่ช่วงระหว่าง –1 ถึง 1 จาก (8) เปลี่ยนเป็น

$$\varphi(v) = \begin{cases} 1 & \text{ถ้า } v > 0 \\ 0 & \text{ถ้า } v = 0 \\ -1 & \text{ถ้า } v < 0 \end{cases} \qquad (2.5.13)$$

หากใช้ในลักษณะ hyperbolic tangent function จะได้

$$\varphi(v) = \tanh(\frac{v}{2}) = \frac{1-\exp(-v)}{1+\exp(-v)} \qquad (2.5.14)$$

2.5.2 Backpropagation Algorithm

นิวรัลเน็ตเวิร์กแบบแบ็กพรอเพเกชั่นเป็นนิวรัลเน็ตเวิร์กแบบหนึ่งที่ได้รับความนิยมในการใช้มากแบบหนึ่งในจำนวนนิวรัลเน็ตเวิร์กที่มีใช้กันอยู่ในปัจจุบัน ลักษณะคือจะมีการเชื่อมต่อระหว่างเซลล์ต่างเลเยอร์ แต่ในเลเยอร์เดียวกันจะไม่มีการเชื่อมต่อกันเลย แต่ค่าความผิดพลาดคำนวณได้ในกระบวนการ Training จะถูกส่งกลับมาคำนวณ (Backpropagate) เพื่อปรับปรุงค่าความสัมพันธ์ระหว่างเซลล์ สำหรับนิวรัลเน็ตเวิร์กวิธีแบ็กพรอพเพเกชั่นนี้ สามารถแบ่งการคำนวณออกได้เป็น 2 ส่วน คือ ส่วน Forward Pass อีกส่วนคือ Backward Pass

ในขั้นตอน Forward Pass ค่าความสัมพันธ์ระหว่างเซลล์ประสาทของระบบจะยังไม่มีการเปลี่ยนแปลง นั่นคือ จะมีการส่งค่าอินพุตผ่านโครงข่ายของระบบระหว่างเซลล์ประสาทจากชั้นแรก ไปยังชั้นต่อไป โดยมีฟังก์ชันการส่งผ่านค่าดังนี้

พิจารณาที่เซลล์ประสาทที่ j

$$y_j(n) = \varphi(v_j(n)) \qquad (2.5.15)$$

ซึ่ง $v_j(n)$ คือ ระดับการทำงานภายใน (Internal Activity Level) ของเซลล์ประสาทที่ j และ

$$v_j(n) = \sum_{j=0}^{p} w_{ji}(n) y_j(n) \qquad (2.5.16)$$

โดย p คือ จำนวนอินพุต ของเซลล์ประสาทที่ j

$w_{ji}(n)$ คือ ค่าความสัมพันธ์ระหว่างเซลล์ประสาทที่ i สู่ j

$y_i(n)$ คือ ค่าสัญญาณอินพุตของเซลล์ประสาทที่ j

ในทำนองเดียวกัน หากเซลล์ประสาทที่ j เป็น hidden layer แรกแล้ว i จะหมายถึงลำดับที่ของอินพุต ซึ่งสามารถเขียนได้เป็น

$$y_i(n) = x_i(n) \qquad (2.5.17)$$

โดยที่ $x_i(n)$ คือ อินพุตตัวที่ i ของ input vetor

และหากเซลล์ประสาทที่ j เป็น output layer ของระบบ และ j จะหมายถึงลำดับที่ของผลลัพธ์ของระบบ ซึ่งสามารถเขียนได้เป็น

$$y_j(n) = o_j(n) \qquad (2.5.18)$$

โดย $o_j(n)$ คือ ผลลัพธ์ตัวที่ j ของ output vector

เมื่อได้ผลลัพธ์แล้วนำไปเปรียบเทียบกับผลลัพธ์ที่มี $d_j(n)$ ก็จะได้ค่าความผิดพลาด $e_j(n)$ สำหรับเซลล์ output ที่ j ดังนั้นช่วง Forward Pass นี้ คือจะเริ่มต้นด้วยการส่งค่าสัญญาณอินพุต คำนวณตามขั้นตอนผ่านมาที่ hidden layer แล้วคำนวณตามขั้นตอนผ่านมาที่ output layer แล้วคำนวณค่าความผิดพลาดใน output layer นี้

สำหรับในช่วง Backward Pass ก็จะเริ่มจาก output layer โดยการส่งค่าความผิดพลาดผ่านระบบมายัง hidden layer กลับมาทีละชั้น ๆ ซึ่งช่วง Backward Pass นี่เอง จะส่งผลให้ค่าความสัมพันธ์ระหว่างเซลล์มีการเปลี่ยนแปลง

บทที่ 3
ระบบกลั่นกรองสารสนเทศ

ระบบกลั่นกรองสารสนเทศ (Information Filtering System) คือ ระบบที่กลั่นกรองข้อมูลข่าวสารจากแหล่งข้อมูลต่างๆในอินเทอร์เน็ตเพื่อให้ได้ข้อมูลที่ตรงตามความต้องการ และเหมาะสมกับผู้ค้นหามากที่สุด

3.1 เป้าหมายของระบบ

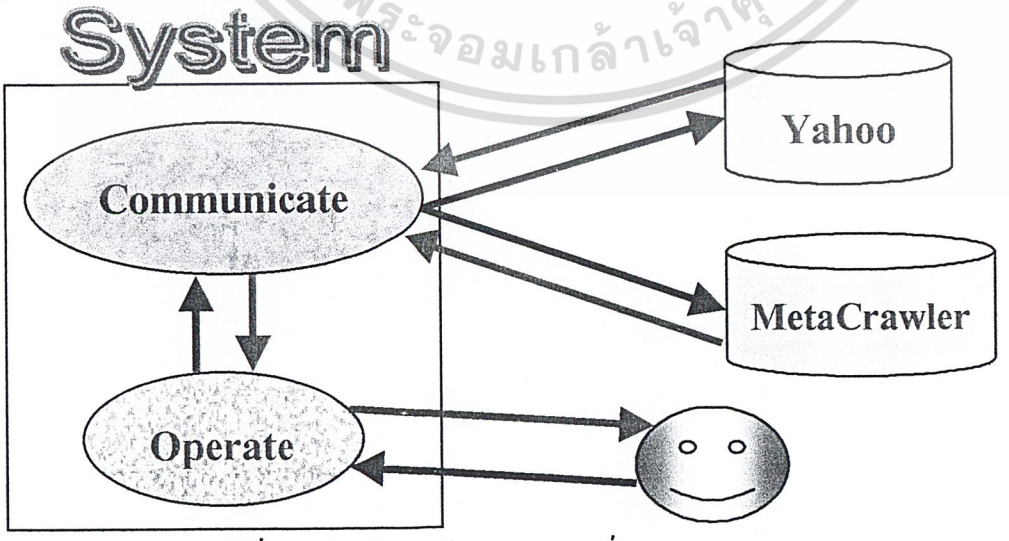
เพื่อให้ได้ระบบกลั่นกรองสารสนเทศที่กลั่นกรองข้อมูลอันมหาศาลในอินเตอร์เน็ตนั้นให้เหลือเพียงข้อมูลที่ตรงตามความต้องการและเหมาะสมกับผู้ค้นหามากที่สุดโดยระบบจะมีลักษณะพิเศษคือ มีการกลั่นกรองข้อมูลที่ดี, มีการกำหนดเวลาในการค้นหาข้อมูลใหม่ ๆ และสามารถเลือกนำเสนอข้อมูลใหม่สำหรับผู้ใช้ที่น่าสนใจ ทั้งนี้ผู้ใช้ควรจะต้องสอนระบบให้ทราบถึงความสนใจข้อมูลของผู้ใช้ด้วย เพื่อผลลัพธ์ที่แม่นยำยิ่งขึ้น

3.2 โครงสร้างของระบบ

ระบบกลั่นกรองสารสนเทศนี้แบ่งออกเป็น 2 ส่วนหลัก ๆ (ดังแสดงในรูปที่ 3.1)คือ

<u>ส่วนปฏิบัติการ</u> เป็นส่วนหลักของระบบ เพื่อดำเนินงานของระบบและจัดการฐานข้อมูลต่าง ๆ รวมไปถึงการติดต่อกับผู้ใช้ด้วย

<u>ส่วนติดต่อกับภายนอก</u> เป็นส่วนที่รับคำสั่งจากส่วนปฏิบัติการเพื่อไปดึงข้อมูลมาจาก Search Engine ที่ระบบใช้มี 2 ตัว คือ MetaCrawler และ Yahoo



รูปที่ 3.1 แสดงโครงสร้างของระบบกลั่นกรองสารสนเทศ

3.3 การใช้งานระบบ

แบ่งออกเป็น 3 ส่วนหลัก ๆ ดังนี้

3.3.1 <u>การค้นหา</u> เป็นส่วนหลักที่ใช้ทำงานของระบบ ผู้ใช้สามารถเลือกการทำงานได้ 3 แบบ คือ

- ค้นหาโดยใช้คีย์เวิร์ด
- ค้นหาตามความชอบของผู้ใช้
- ตั้งเวลาให้ค้นหาตามความชอบของผู้ใช้

3.3.2 <u>สมุดเก็บลิงค์</u> มีไว้เพื่อเก็บลิงค์เพื่อนำมาเรียกดูในภายหลังและอาจสอนระบบด้วยลิงค์ที่เก็บไว้ได้ มี 2 เล่ม คือ สมุดเก็บลิงค์ที่ผู้ใช้สนใจ และ สมุดเก็บลิงค์ที่ผู้ใช้ค้นหาตามความชอบของผู้ใช้

3.3.3 <u>จัดการกับข้อมูลผู้ใช้</u> เป็นส่วนที่ทำงานกับข้อมูลของผู้ใช้ มี 4 แบบ คือ

- *สร้างข้อมูลผู้ใช้คนใหม่* หากผู้ใช้ยังไม่เคยใช้งานระบบเลย จะต้องประกาศตัวเองให้ระบบทราบก่อน รวมไปถึงรายละเอียดข้อมูลของผู้ใช้เองด้วย มิฉะนั้นจะไม่สามารถใช้งานระบบได้
- *เลือกข้อมูลผู้ใช้* เมื่อเริ่มเข้าสู่ระบบ ผู้ใช้จะต้องประกาศตัวเองให้ระบบทราบทุกครั้ง หากผู้ใช้เคยแจ้งข้อมูลผู้ใช้ให้กับระบบมาก่อนแล้ว จะสามารถเลือกข้อมูลผู้ใช้ที่เคยแจ้งไว้ได้เลย โดยต้องใส่รหัสผ่านเพื่อยืนยัน
- *แก้ไขข้อมูลผู้ใช้* ผู้ใช้สามารถแก้ไขข้อมูลของผู้ใช้ที่เคยแจ้งไว้กับระบบได้
- *ลบข้อมูลผู้ใช้* เมื่อไม่ต้องการใช้ข้อมูลนี้แล้วสามารถลบออกจากระบบได้ โดยต้องยืนยันด้วยการใส่รหัสผ่านเช่นกัน

ทั้งนี้สามารถดูรายละเอียดเพิ่มเติมได้ในหัวข้อ 4.1 องค์ประกอบของโปรแกรม

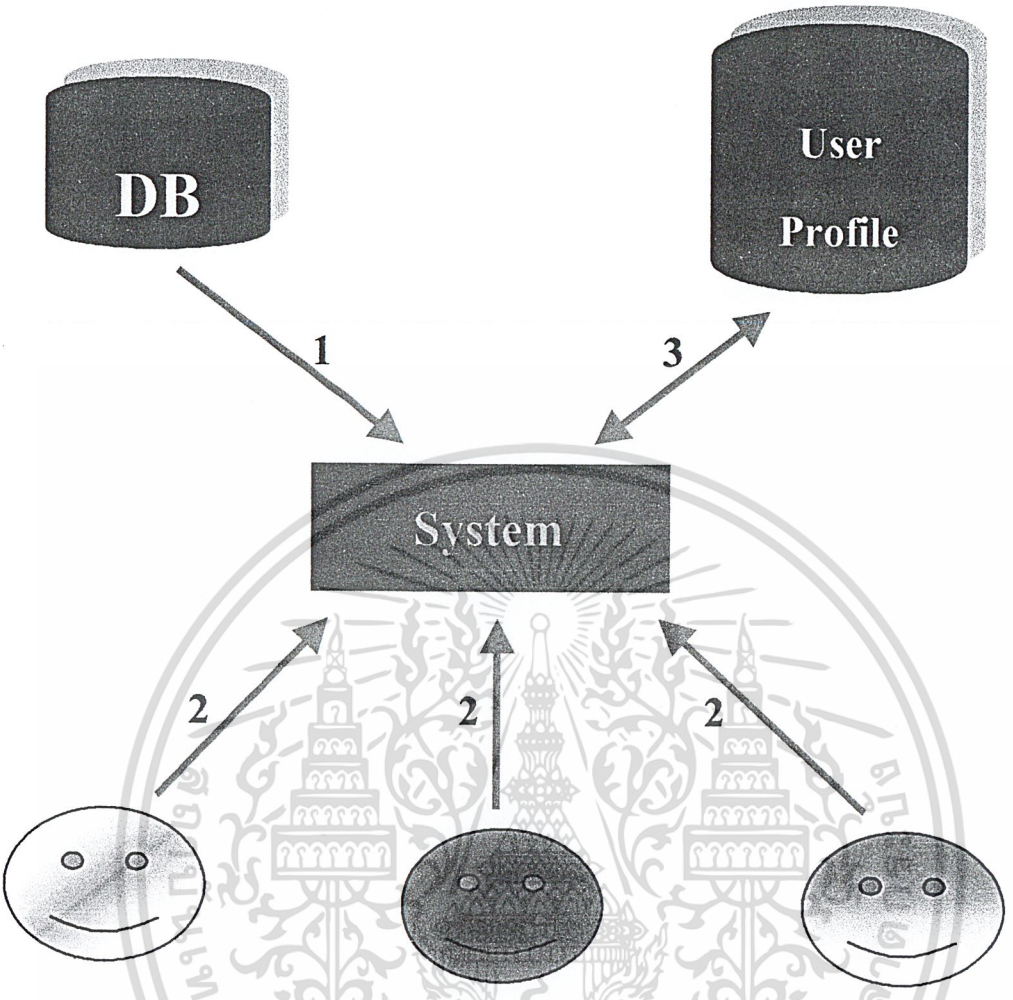
3.4 ขั้นตอนการทำงานของระบบ

จะขออธิบายถึงขั้นตอนการทำงานใน 2 ช่วงหลัก คือ ช่วงเริ่มต้นระบบ และ ช่วงการค้นหาข้อมูล

<u>ช่วงเริ่มต้นระบบ</u>

เป็นการเริ่มต้นก่อนที่จะพร้อมทำงานได้ดังแสดงในรูปที่ 3.2 มีขั้นตอนดังนี้

1. ระบบจะอ่านข้อมูลที่จำเป็นต้องใช้ เช่น รายชื่อของไดเรกทอรี เป็นต้น มาเก็บไว้ รวมทั้งกำหนดค่าตั้งต้นต่าง ๆ ของระบบ
2. ผู้ใช้ระบุตัวเองให้ระบบทราบ
3. ระบบดึงข้อมูลผู้ใช้จากฐานข้อมูล

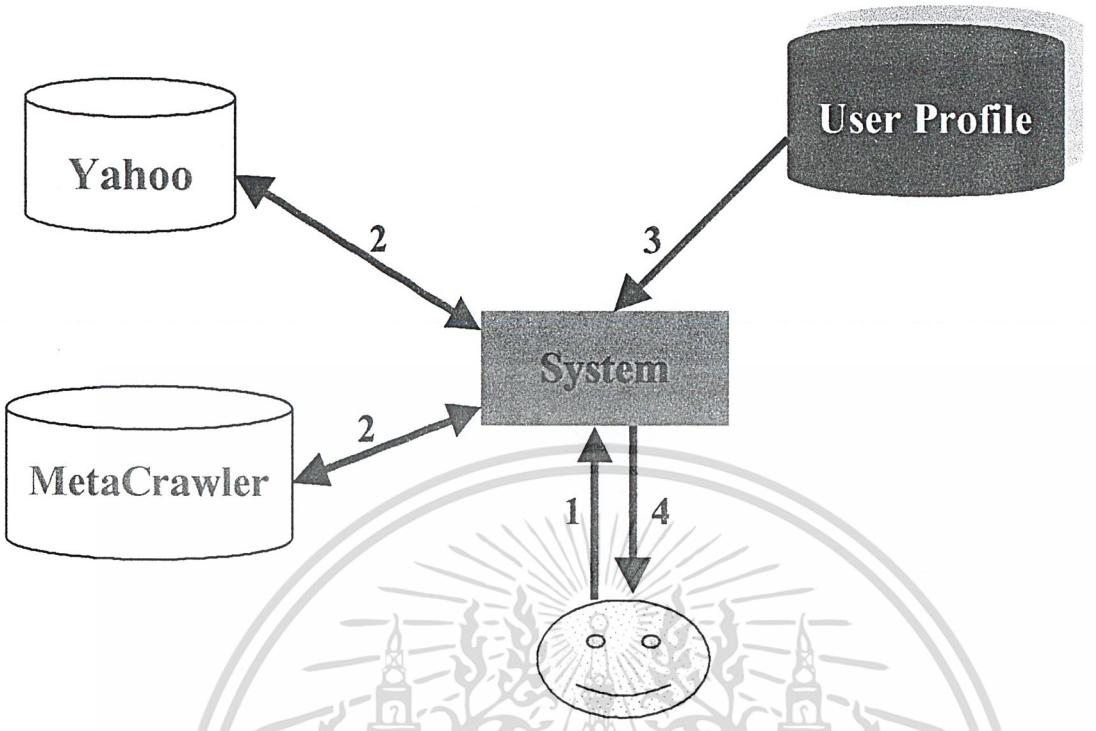


รูปที่ 3.2 แสดงขั้นตอนการเริ่มต้นระบบ

ช่วงการค้นหา

เป็นการทำงานของระบบดังแสดงในรูปที่ 3.3 มีขั้นตอนดังนี้

1. ผู้ใช้แจ้งความต้องการค้นหาแก่ระบบและใส่ข้อมูลที่ต้องการค้นหาและลักษณะการค้นหา
2. ระบบจะไปค้นหาข้อมูลในอินเทอร์เน็ตโดยผ่าน Search Engine 2 ตัว คือ MetaCrawler และ Yahoo จะได้ผลลัพธ์กลับมาเป็นข้อมูลของเว็บไซต์
3. ระบบจะประมวลผลโดยการคำนวณค่าความน่าสนใจของเว็บไซต์นั้นโดยเปรียบเทียบกับข้อมูลของผู้ใช้ ได้เป็นค่าความน่าสนใจของเว็บไซต์ มีค่าเต็ม 100
4. แสดงผลลัพธ์แก่ผู้ใช้



รูปที่ 3.3 แสดงขั้นตอนการค้นหาของระบบ

3.5 ความต้องการของระบบ

เครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการใดก็ได้ (ถ้าต้องการใช้อย่างเต็มประสิทธิภาพควรจะเป็นระบบปฏิบัติการ Windows) ที่เชื่อมต่อกับอินเทอร์เน็ตไม่ว่าจะด้วยวิธีการใดก็ตาม

3.6 การติดตั้งระบบ

ในชุดติดตั้งนี้จะมีไฟล์ทั้งหมดดังนี้

- dialog.java
- inform.java
- InterestYahooLinkParser.java
- MetacrawlerLinkParserThread.java
- NeuralNetwork.java
- NNFile.java
- YahooDirParserThread.java
- directory.txt

- newtrain.dat
- jdk1_3.exe
- directory -> user

การติดตั้งระบบมีขั้นตอนดังนี้

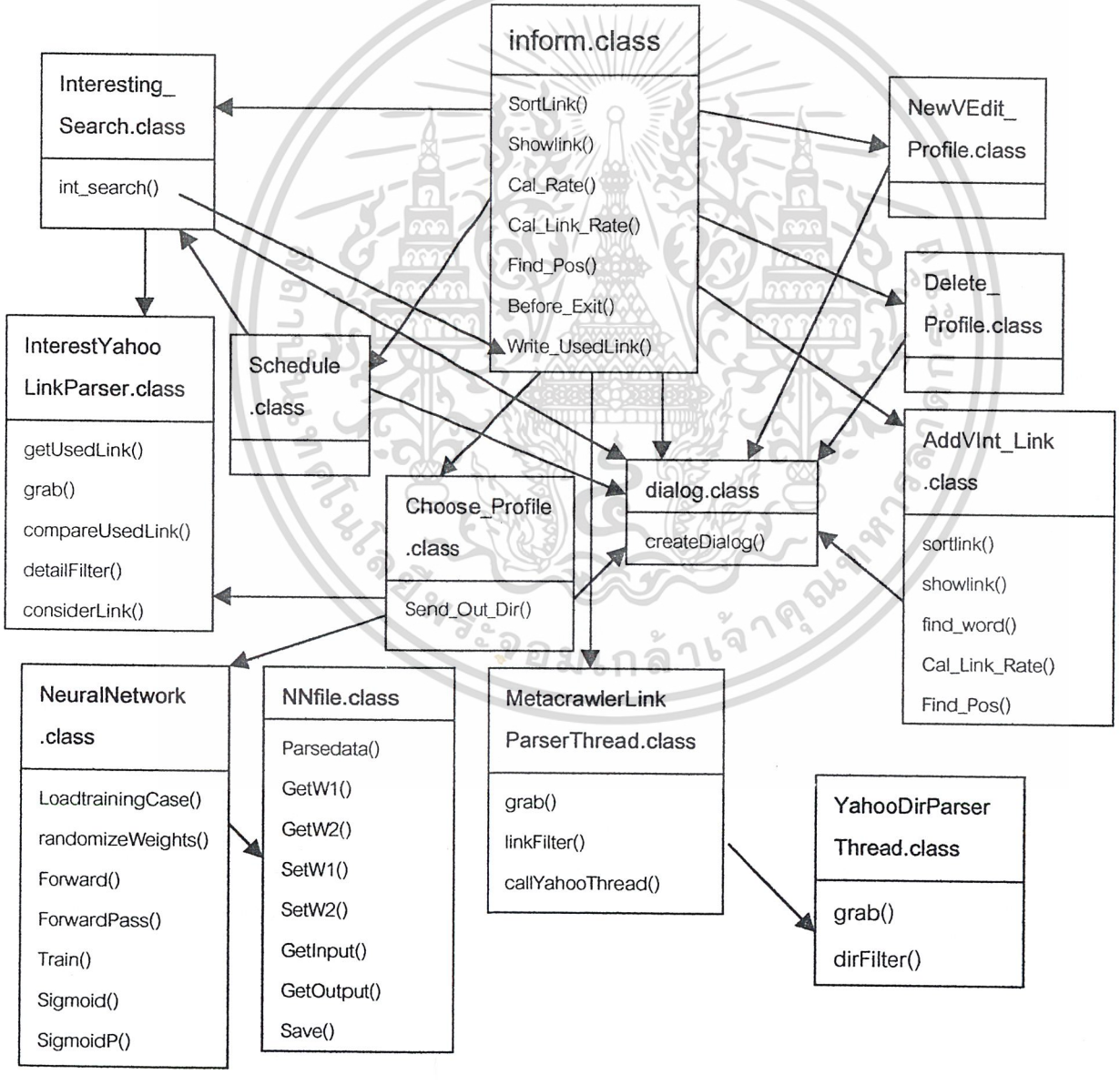
1. สำเนาไฟล์ทั้งหมดพร้อมทั้งไดเรกทอรีไปที่เครื่องที่ต้องการจะติดตั้ง
2. ติดตั้งโปรแกรม JDK เพื่อคอมไพล์ไฟล์จาวา โดยเรียกใช้ jdk1_3.exe สำหรับระบบปฏิบัติการ Windows สำหรับระบบปฏิบัติการอื่นให้หาไฟล์ที่จะคอมไพล์ไฟล์จาวาตามแต่ระบบปฏิบัติการนั้น
3. ติดตั้งค่าในไฟล์ autoexec.bat แล้วเรียกใช้ไฟล์ autoexec.bat นี้ ค่าที่ติดตั้งมีดังนี้
   - set path = c:\(ไดเรกทอรีที่เก็บโปรแกรม JDK)\bin
   - set classpath = .;c:\(ไดเรกทอรีที่สำเนาไฟล์ไว้ทั้งหมด)
4. คอมไพล์ไฟล์ที่มีนามสกุล .java ทุกไฟล์ โดยใช้คำสั่ง javac c:\ (ไดเรกทอรีที่สำเนาไฟล์ไว้ทั้งหมด)\(ชื่อไฟล์).java อาจใช้ผ่านปุ่ม Start แล้ว Run ใน Windows หรือใช้ผ่านทาง MS - Dos Mode ก็ได้

การติดตั้งจะสิ้นสุดเท่านี้ เมื่อต้องการจะเรียกใช้ระบบ หากเรียกใช้จาก MS - Dos Mode ให้ย้ายไดเรกทอรีไปที่ไดเรกทอรีที่สำเนาไฟล์ไว้แล้วเรียกคำสั่ง java inform แต่หากต้องการสร้าง shortcut ไว้เรียกใช้ ให้ใส่ Command Line ของ shortcut ว่า java inform เช่นกัน แต่ให้แก้ใน Properties ของ shortcut ที่แถบ shortcut ให้ Start in เป็น c:\(ไดเรกทอรีที่สำเนาไฟล์ไว้)

บทที่ 4
โครงสร้างและการทำงานของโปรแกรม

โปรแกรม Information Filtering System เขียนขึ้นด้วยภาษาจาวา สาเหตุที่เลือกใช้ภาษาจาวาเพราะมีจุดเด่นในเรื่องของการติดต่อสื่อสารทางอินเทอร์เน็ตและมีลักษณะแบบ Object-Oriented รวมไปถึงรูปแบบการเขียนที่ง่ายและสามารถนำไปพัฒนาต่อได้สะดวก และที่สำคัญที่สุดคือ "Write Once Run Anywhere" แต่ภาษาจาวายังคงมีข้อเสียในด้านกราฟฟิคและโปรแกรมเสริมที่จะช่วยในการเขียนโปรแกรมยังด้อยอยู่ และการติดต่อกับไฟล์หรือระบบปฏิบัติการยังทำได้ไม่ดีนัก

โปรแกรม Information Filtering System มีโครงสร้างแสดงในรูปที่ 4.1 ดังนี้



รูปที่ 4.1 แสดงโครงสร้างของโปรแกรม Information Filtering System

4.1 องค์ประกอบของโปรแกรม

โปรแกรมสามารถแบ่งออกเป็น 2 ส่วนหลัก ๆ คือ

1. **ส่วนปฏิบัติการ** มี inform.class เป็นคลาสหลักที่จะเรียกใช้คลาสอื่น เป็นโปรแกรมหลักที่จะตอบสนองผู้ใช้และแสดงผล เมื่อเริ่มเปิดโปรแกรมผู้ใช้จะต้องประกาศตัวของผู้ใช้ให้โปรแกรมทราบก่อนทุกครั้ง หากยังไม่เคยใช้โปรแกรมมาก่อน จะต้องกรอกข้อมูลของผู้ใช้พร้อมทั้งค่าความสนใจเว็บไซต์แต่ละประเภทให้กับโปรแกรมจึงจะสามารถใช้งานโปรแกรมได้ ผู้ใช้สามารถใช้งานโปรแกรมโดยแบ่งออกเป็น 3 ส่วนหลักได้แก่
   - ส่วนการค้นหา สามารถสั่งงานได้ 3 รูปแบบ คือ
     - *ค้นหาด้วยคีย์เวิร์ด* สำหรับการทำงานนี้ตัวโค้ดอยู่ใน inform.class อยู่แล้ว โดยจะนำคีย์เวิร์ดที่ได้เข้ารหัสแล้วส่งไปให้ MetacrawlerLinkParserThread เพื่อค้นหาลิงค์แล้วนำลิงค์ที่ได้มาคำนวนค่าความน่าสนใจและเรียงลำดับความน่าสนใจแล้วจึงแสดงผล
     - *ค้นหาโดยไม่ใส่คีย์เวิร์ด* เพื่อให้โปรแกรมค้นหาเว็บไซต์ที่น่าจะตรงกับความต้องการของผู้ใช้โดยไม่จำเป็นต้องใส่คีย์เวิร์ด โปรแกรมจะดูข้อมูลของผู้ใช้มาเพื่อนำไปค้นหาเว็บไซต์ตามความชอบของผู้ใช้ โดยจะเรียกใช้ Interesting_Search.class เพื่อค้นหา มี 2 รูปแบบ คือ
       - normal จะเข้าไปตรวจหาว่าไดเรกทอรีไหนที่ผู้ใช้สนใจมากที่สุดแล้วไปดึงเอาเว็บไซต์ในไดเรกทอรีนั้นจาก Yahoo
       - advance สามารถเลือกค้นหาเว็บไซต์ที่น่าสนใจตามประเทศหรือทวีปต่าง ๆ ได้ รวมทั้งยังแยกเป็นเว็บไซต์ประเภทท่องเที่ยวหรือเป็นเว็บไซต์ทั่วไปของประเทศหรือทวีปนั้น
     - *ตั้งเวลาให้ค้นหาเว็บไซต์* โดยจะเป็นการค้นหาแบบไม่ใส่คีย์เวิร์ดแบบ normal เท่านั้น โดยจะเรียกใช้ Schedule.class ซึ่งทำงานโดยให้ผู้ใช้ตั้งเวลาให้โปรแกรมค้นหาเว็บไซต์ที่น่าสนใจในภายหลังได้ ซึ่งเมื่อถึงเวลาที่กำหนดโปรแกรมจะไปเรียกใช้ method int_search() ของ Interesting_Search.class
   - ส่วนสมุดเก็บลิงค์ มีสมุดเก็บลิงค์อยู่ 2 เล่ม เล่มแรกจะเก็บลิงค์ที่ผู้ใช้ต้องการเก็บไว้เพื่อต้องการจะมาเปิดดูในภายหลังและอีกเล่มจะเก็บลิงค์ที่ค้นหาได้จากการค้นหาแบบไม่ใส่คีย์เวิร์ด โปรแกรมจะเรียกสมุดเก็บลิงค์โดยผ่านทาง AddVInt_Link.class ซึ่งผู้ใช้จะสามารถสอนโปรแกรมเพิ่มเติมได้ว่าชอบหรือไม่ชอบลิงค์ใด เพื่อให้โปรแกรมเรียนรู้ได้ แต่การสอนจะต้องเลือกเฉพาะลิงค์ที่มีไดเรกทอรีเท่านั้น
   - ส่วนข้อมูลของผู้ใช้ เป็นส่วนจัดการกับข้อมูลของผู้ใช้ มี 4 หัวข้อ คือ
     - *สร้างผู้ใช้คนใหม่* สำหรับผู้ใช้ที่ยังไม่เคยกรอกข้อมูลให้กับโปรแกรม ผู้ใช้จะต้องกรอกรายละเอียดต่าง ๆ ที่จำเป็นพร้อมทั้งความสนใจต่อเว็บไซต์ประเภทต่าง ๆ โปรแกรมจะเรียกใช้ NewVEdit_Profile.class เพื่อให้ผู้ใช้กรอกข้อมูลของผู้ใช้ เมื่อกรอกเสร็จเรียบร้อย โปรแกรม

จะสร้างไฟล์ข้อมูลของผู้ใช้และส่งค่าที่จำเป็นไปที่ inform.class แล้วจึงอนุญาตให้ผู้ใช้ใช้งานโปรแกรมได้

- *เลือกข้อมูลของผู้ใช้* สำหรับผู้ใช้ที่เคยกรอกข้อมูลแล้วหรือต้องการเปลี่ยนผู้ใช้ ต้องใส่รหัสผ่านก่อนเข้าใช้ โปรแกรมจะเรียกใช้ Choose_Profile.class เพื่อรับชื่อและรหัสผ่านของผู้ใช้ หากรหัสผ่านถูกต้อง ก็จะส่งข้อมูลของผู้ใช้ต่าง ๆ ที่จำเป็นไปที่ inform.class และอนุญาตให้ผู้ใช้ใช้งานโปรแกรมได้
- *แก้ไขข้อมูลของผู้ใช้* สำหรับให้ผู้ใช้เปลี่ยนแปลงแก้ไขข้อมูลของผู้ใช้ โปรแกรมจะเรียกใช้ NewVEdit_Profile.class เพื่อให้ผู้ใช้เปลี่ยนแปลงค่าและเก็บข้อมูลลงไฟล์พร้อมทั้งส่งค่าข้อมูลที่แก้ไขแล้วไปที่ inform.class ด้วย
- *ลบข้อมูลของผู้ใช้* เมื่อผู้ใช้ไม่ต้องการใช้ชื่อและข้อมูลหนึ่งข้อมูลใดอีกต่อไป สามารถลบทิ้งได้ โดยเลือกชื่อของผู้ใช้(แต่ต้องไม่ใช่ชื่อที่ใช้อยู่ในปัจจุบัน)และใส่รหัสผ่าน หากรหัสผ่านถูกต้องโปรแกรมจะลบข้อมูลของผู้ใช้ออก การทำงานเหล่านี้โปรแกรมเรียกใช้ผ่าน Delete_Profile.class

2. ส่วนติดต่อกับภายนอก

สำหรับส่วนที่ติดต่อกับภายนอกของโปรแกรมจะมีอยู่ 3 งาน คือ

1. งานค้นหาลิงค์ที่เกี่ยวข้องกับคีย์เวิร์ดที่ผู้ใช้ระบุ ใช้ MetacrawlerLinkParser.class โดยมีรายละเอียดดังนี้
   - เซิร์ฟเวอร์ที่ต้องการติดต่อด้วยคือ search.metacrawler.com/crawler
   - ใช้เมธอด GET ในการร้องขอ ซึ่งมีตัวแปรที่ต้องส่งค่าไปยังเซิร์ฟเวอร์ คือ
     1) general เป็นตัวแปรที่ใช้เก็บค่าของคีย์เวิร์ดที่ผู้ใช้ระบุ
     2) rpp เป็นตัวแปรที่ใช้เก็บค่าจำนวนลิงค์ผลลัพธ์ที่ได้จากการประมวลผล
   - เงื่อนไขที่ต้องการสำหรับงานนี้คือ ต้องการชื่อลิงค์, หัวข้อ (title), รายละเอียด (detail), และค่าความน่าสนใจของลิงค์นั้น
2. งานแยกประเภทของลิงค์ โดยอาศัยการแบ่งไดเรกทอรีของ Yahoo! โดยมีรายละเอียดดังนี้
   - เซิร์ฟเวอร์ที่ต้องการติดต่อด้วยคือ search.yahoo.com/bin/search
   - ใช้เมธอด GET ในการร้องขอ ซึ่งมีตัวแปรที่ต้องส่งค่าไปยังเซิร์ฟเวอร์ คือ
     1) q เป็นตัวแปรที่ใช้เก็บค่าของคีย์เวิร์ดที่ผู้ใช้ระบุ
   - เงื่อนไขที่ต้องการสำหรับงานนี้คือ ต้องการไดเรกทอรีของลิงค์ที่ระบุ
3. งานค้นหาลิงค์ที่น่าสนใจ โดยมีรายละเอียดดังนี้
   - เซิร์ฟเวอร์ที่ต้องการติดต่อด้วยคือ dir.yahoo.com/<subdirectory>

     โดยที่ <subdirectory> คือ ไดเรดทอรีย่อยที่ได้พิจารณาแล้วว่าเป็นไดเรกทอรีที่ผู้ใช้นิยมมากที่สุด
   - ใช้เมธอด GET ในการร้องขอ

- เงื่อนไขที่ต้องการสำหรับงานนี้คือ ต้องการลิงค์ในไดเรกทอรีที่ระบุจำนวนที่ไม่มากจนน่าเบื่อ หรือน้อยจนเกินไป

4.2 รายละเอียดในส่วนการทำงานต่าง ๆ ของโปรแกรม

4.2.1 การเก็บฐานข้อมูลของผู้ใช้

โปรแกรมเก็บข้อมูลของผู้ใช้ในไดเรกทอรีย่อย "user" ในไดเรกทอรีที่เก็บโปรแกรม โดยมีชื่อไฟล์เป็น Username ของผู้ใช้ และมีนามสกุลแตกต่างกันออกไป 5 ไฟล์ ดังนี้

1. .pro เป็นไฟล์สำหรับเก็บข้อมูลส่วนตัวของผู้ใช้และความสนใจของผู้ใช้ในแต่ละประเภทเว็บไซต์มีทั้งหมด 21 บรรทัด มีรายละเอียดดังนี้

   บรรทัดที่ 1 เก็บ Username
   บรรทัดที่ 2 เก็บ ชื่อและนามสกุล
   บรรทัดที่ 3 เก็บ อายุ
   บรรทัดที่ 4 เก็บ การศึกษา
   บรรทัดที่ 5 เก็บ เพศ
   บรรทัดที่ 6 เก็บ อาชีพ
   บรรทัดที่ 7 เก็บ รหัสผ่าน
   บรรทัดที่ 8-21 เก็บค่าความสนใจในแต่ละประเภทเว็บไซต์ 14 ประเภทหลัก

2. .add เก็บลิงค์ที่ผู้ใช้สนใจ ต้องการจะมีไว้ดูภายหลัง เป็นเสมือนสมุดจดสำหรับผู้ใช้ โดยผู้ใช้จะสามารถนำลิงค์มาเก็บไว้ได้จาก 2 แหล่ง คือ จากการค้นหาโดยคีย์เวิร์ด(แสดงผลในโปรแกรมหลัก)และจากการค้นหาโดยไม่ใช้คีย์เวิร์ด(แสดงผลในสมุดเก็บลิงค์ที่น่าสนใจ) รูปแบบการเก็บลิงค์จะเก็บข้อมูลของลิงค์ ลิงค์ละ 3 บรรทัด มีรายละเอียดดังนี้

   บรรทัดที่ 1 เก็บ URL ตามด้วย " , Directory:" ตามด้วยชื่อไดเรกทอรีอีก 3 ชั้น แต่ละชั้นคั่นด้วย ">"
   บรรทัดที่ 2 เก็บ หัวข้อ(title)ของ URL นั้น
   บรรทัดที่ 3 เก็บ รายละเอียดของ URL นั้น

3. .int เก็บลิงค์ที่ได้จากการค้นหาโดยไม่ใช้คีย์เวิร์ดและจากการตั้งเวลาค้นหา ลักษณะการเก็บลิงค์เหมือนกับลิงค์ที่เก็บในสมุดเก็บลิงค์ที่ผู้ใช้สนใจในไฟล์ .add

4. .tra เป็นไฟล์สำหรับเก็บค่าที่ผู้ใช้สอนให้โปรแกรมทราบความสนใจของผู้ใช้ต่อประเภทเว็บไซต์ ซึ่งจะเก็บเป็นประเภทเว็บไซต์ทั้งหมด 3 ชั้นของไดเรกทอรี มีรวมกันทั้งสิ้น 4,037 ไดเรกทอรี ลักษณะการเก็บมีดังต่อไปนี้

   บรรทัดที่ 1-14 เก็บ ค่าความสนใจของไดเรกทอรีชั้นที่ 1 มีทั้งหมด 14 ไดเรกทอรี (ไดเรกทอรีที่ 1-14)

บรรทัดที่ 15-36 เก็บ ค่าความสนใจของไดเรกทอรีชั้นที่ 2 ของไดเรกทอรีชั้นที่ 1 "Arts" (ไดเรกทอรีแรกของชั้นที่1) มีทั้งหมด 22 ไดเรกทอรี (ไดเรกทอรีที่ 1.1-1.22)

บรรทัดที่ 37-49 เก็บ ค่าความสนใจของไดเรกทอรีชั้นที่ 3 ของไดเรกทอรีชั้นที่ 2 "Art History" (ไดเรกทอรีแรกของชั้นที่2) มีทั้งหมด 13 ไดเรกทอรี (ไดเรกทอรีที่ 1.1.1-1.1.13)

บรรทัดที่ 50-55 เก็บ ค่าความสนใจของไดเรกทอรีชั้นที่ 3 ของไดเรกทอรีชั้นที่ 2 "Art Historians" (ไดเรกทอรีที่ 2 ของชั้นที่2) มีทั้งหมด 13 ไดเรกทอรี (ไดเรกทอรีที่ 1.2.1-1.2.6)

บรรทัดที่ 56-317 เก็บลักษณะเดียวกันไปเรื่อย ๆ จนครบทั้งหมดของไดเรกทอรีแรกของชั้นที่ 1 "Arts"

จากนั้นจึงเริ่มเก็บส่วนของไดเรกทอรีที่ 2 ของชั้นที่ 1 "Business and Economy" ต่อจนครบทั้งหมด 14 ไดเรกทอรี

5. .use เก็บ URL ที่ผู้ใช้ได้เคยเก็บมาแล้วไม่ว่าจะในสมุดเก็บลิงค์ที่ผู้ใช้สนใจหรือในสมุดเก็บลิงค์ที่ได้จากการค้นหาแบบไม่ใช้คีย์เวิร์ด URL เหล่านี้จะนำมาใช้ตรวจดูเมื่อผู้ใช้ค้นหาแบบไม่ใช้คีย์เวิร์ดเพื่อให้ได้เว็บไซต์ไม่ซ้ำกับที่เคยเรียกดูมาแล้ว การเก็บจะเก็บ URL ละ 1 บรรทัด

4.2.2 การอ้างถึงไดเรกทอรี

การอ้างถึงไดเรกทอรีที่มีทั้งหมดถึง 4,037 ไดเรกทอรีเป็นเรื่องยาก จำเป็นต้องกำหนดตำแหน่งของแต่ละไดเรกทอรีไว้ล่วงหน้า มีวิธีการอ้างถึงรายละเอียดต่าง ๆ ของไดเรกทอรีดังนี้

1. **ตำแหน่ง** โปรแกรมจะเก็บตำแหน่งของไดเรกทอรีทั้งหมดไว้เพื่อจะระบุได้เมื่อต้องการ ด้วยการเก็บเป็นอาเรย์ของจำนวนไดเรกทอรีเอาไว้ชื่อ toplus เป็นอาเรย์ 2 มิติ มีขนาด 14X45 การเก็บมีลักษณะดังนี้

   toplus[a][0] เก็บจำนวนไดเรกทอรีชั้นที่ 2 ของไดเรกทอรีที่ a+1 ของชั้นที่ 1

   toplus[a][1-22] แต่ละอันเก็บจำนวนไดเรกทอรีชั้นที่ 3 ของไดเรกทอรีชั้นที่ 2 แต่ละอัน

   ตัวอย่างเช่น ให้ a = 0 เป็นไดเรกทอรีแรกของชั้นที่ 1 ชื่อ "Arts" จะมีไดเรกทอรีชั้นที่ 2 ทั้งหมดเป็น 22 toplus[0][0] จะมีค่า 22 และ toplus[0][1] จะเก็บจำนวนไดเรกทอรีชั้นที่ 3 ที่มีอยู่ในไดเรกทอรีแรกของไดเรกทอรีชั้นที่ 2 ของ "Arts" คือ "Art History" ซึ่งมีไดเรกทอรีชั้นที่ 3 อยู่ทั้งหมด 13 ไดเรกทอรี เพราะฉะนั้น toplus[0][1] จะมีค่า 13

   การกำหนดเช่นนี้มีข้อเสียคือ ไดเรกทอรีชั้นที่ 1 ที่มีจำนวนไดเรกทอรีชั้นที่ 2 มากที่สุดคือมีถึง 44 ไดเรกทอรี ในขณะที่จำนวนที่น้อยที่สุดมีเพียง 4 ไดเรกทอรี ทำให้จองพื้นที่เกินไปถึง 40 ไดเรกทอรี

   การจะระบุไดเรกทอรีใดไดเรกทอรีหนึ่ง ๆ ทำได้ดังนี้

   <u>การระบุไดเรกทอรีชั้นที่ 1</u> เช่น ไดเรกทอรีที่ 5 "Entertainment" จะมีค่าตำแหน่งไดเรกทอรี เป็น 5

   <u>การระบุไดเรกทอรีชั้นที่ 2</u> เช่น ไดเรกทอรีที่ 5.6 "Consumer Electronics" จะต้องคำนวณค่าตำแหน่งไดเรกทอรี โดยเริ่มต้น ค่าตำแหน่ง pos เป็น 0 บวกจำนวนไดเรกทอรีชั้นที่ 1 คือ 14 ค่า pos จะ

เป็น 14 ต่อมาบวกจำนวนไดเรกทอรีทั้งหมดตั้งแต่ไดเรกทอรีที่ 1-4 ของชั้นที่ 1 ดังนี้เริ่มจากตรวจค่าใน toplus[0][0] ได้ 22 ให้บวกเข้าไปใน pos เป็น 36 จากนั้นบวกค่าตั้งแต่ toplus[0][1] ถึง toplus[0][22] จะได้จำนวนไดเรกทอรีชั้นที่ 3 ของ ไดเรกทอรีชั้นที่ 1 "Arts" ทั้งหมด มีค่าเท่ากับ 281 บวกเข้าไปใน pos ได้ 317 ทำเช่นเดียวกันกับไดเรกทอรีที่ 2, 3 และ 4 จะได้ pos มีค่า 1,205 จากนั้นบวกอีก 6 เพื่อระบุตำแหน่งไดเรกทอรีที่ 5.6 ได้ค่าตำแหน่งเป็น 1,211

การระบุไดเรกทอรีชั้นที่ 3 เช่น ไดเรกทอรีที่ 5.6.7 "Raffles" ให้นำค่า pos ของตำแหน่งไดเรกทอรีที่ 5.6 คือ 1,211 ลบออกด้วย 6 เป็น 1,205 แล้วจึงบวกค่า toplus[4][0] ซึ่งเป็นจำนวนไดเรกทอรีชั้นที่ 2 ในไดเรกทอรีที่ 5 ของชั้นที่ 1 (Entertainment) มีค่า 25 ค่า pos จึงเป็น 1,236 ต่อมาบวกด้วยจำนวนไดเรกทอรีชั้นที่ 3 ตั้งแต่ไดเรกทอรีที่ 1-6 ของไดเรกทอรีชั้นที่ 2 (toplus[4][1] ถึง toplus[4][6]) จะได้ค่า pos เป็น 1,286 แล้วบวกด้วยตำแหน่งของไดเรกทอรีชั้นที่ 3 คือ 7 ได้ค่าตำแหน่งเป็น 1,293

2. ชื่อ ชื่อไดเรกทอรีทั้งหมดจะถูกเก็บในไฟล์ directory.txt ซึ่งอยู่ที่เดียวกันกับไฟล์โปรแกรม มีตำแหน่งการเก็บตามการระบุตำแหน่ง ไดเรกทอรีที่อยู่ในบรรทัดเดียวกันจะอยู่ในไดเรกทอรีเดียวกัน แต่ละชื่อไดเรกทอรีจะคั่นด้วย "%" ชื่อเหล่านี้จะถูกนำมาใช้เพื่อระบุไดเรกทอรี ตัวอย่างการเก็บชื่อในไฟล์ directory.txt

บรรทัดที่ 1	เก็บ ชื่อไดเรกทอรีชั้นที่ 1 ทั้งหมด 14 ชื่อ
บรรทัดที่ 2	เก็บ ชื่อไดเรกทอรีชั้นที่ 2 ของไดเรกทอรีแรกของชั้นที่ 1 (Arts) ทั้งหมด 22 ชื่อ
บรรทัดที่ 3	เก็บ ชื่อไดเรกทอรีที่ 1-13 ซึ่งเป็นไดเรกทอรีชั้นที่ 3 ของไดเรกทอรีแรกของชั้นที่ 2 (Art History)
บรรทัดที่ 4	เก็บ ชื่อไดเรกทอรีที่ 1-6 ซึ่งเป็นไดเรกทอรีชั้นที่ 3 ของไดเรกทอรีที่ 2 ของชั้นที่ 2 (Artists) ทั้งหมด 6 ชื่อ
บรรทัดที่ 5-24	เก็บ ชื่อไดเรกทอรีชั้นที่ 3 ของไดเรกทอรีที่ 3-22 ของชั้นที่ 2
บรรทัดที่ 25	เก็บ ชื่อไดเรกทอรีชั้นที่ 2 ของไดเรกทอรีที่ 2 ของชั้นที่ 1 (Business and Economy) ทั้งหมด 26 ชื่อ

3. ค่าความสนใจ เป็นค่าที่ได้จากการสอนของผู้ใช้ให้โปรแกรมทราบความชอบของผู้ใช้ในแต่ละประเภทไดเรกทอรี การระบุตำแหน่งจะเป็นตำแหน่งเดียวกับตำแหน่งไดเรกทอรี ระหว่างการใช้งานโปรแกรมจะเก็บในตัวแปรชนิด Vector ชื่อ Train_Rate อยู่ในคลาส inform ดังนั้น Train_Rate จะมีสมาชิก 4,037 ตัวและจะเก็บลงไฟล์ของชื่อผู้ใช้โดยมีนามสกุลเป็น .tra โดยเก็บไดเรกทอรีละบรรทัด เพราะฉะนั้นไฟล์จะมีทั้งหมด 4,037 บรรทัด

4.2.3 การสอนโปรแกรม

ผู้ใช้สามารถสอนให้โปรแกรมทราบความชอบในแต่ละประเภทเว็บไซต์(ซึ่งก็คือไดเรกทอรี)ให้โปรแกรมทราบได้ โดยการกดปุ่ม Prefer หรือ Not Prefer ที่จะปรากฏอยู่ทุกครั้งในการแสดงลิงค์ การจะสอนนั้น ให้เช็คที่กล่องสี่เหลี่ยมหน้าลิงค์ที่ต้องการ โดยมีเงื่อนไขว่า ลิงค์ที่จะเช็คนั้นต้องเป็นลิงค์ที่มีไดเรกทอรีด้วย หากลิงค์นั้นไม่มีไดเรกทอรี โปรแกรมจะเตือนให้ทราบ ผู้ใช้สามารถดูไดเรกทอรีของลิงค์ได้ โดยจะแสดงอยู่ถัดจากชื่อลิงค์ เริ่มแรก

สุดถ้าผู้ใช้ยังไม่เคยสอนโปรแกรมค่าความสนใจของทุกไดเรกทอรีจะมีค่าเป็น 0 เมื่อผู้ใช้เลือกลิงค์แล้วกดปุ่ม Prefer หรือ Not Prefer โปรแกรมจะไปเพิ่มหรือลดค่าความสนใจในตัวแปรชนิด Vector ชื่อ Train_Rate ในคลาส inform ให้กับไดเรกทอรีของลิงค์นั้น โดยมีอัตราการเพิ่มหรือลดดังนี้

ไดเรกทอรีชั้นที่ 1	ปุ่ม Prefer เพิ่ม 1	ปุ่ม Not Prefer ลด 1
ไดเรกทอรีชั้นที่ 2	ปุ่ม Prefer เพิ่ม 2	ปุ่ม Not Prefer ลด 2
ไดเรกทอรีชั้นที่ 3	ปุ่ม Prefer เพิ่ม 3	ปุ่ม Not Prefer ลด 3

4.2.4 การให้ค่าความน่าสนใจแก่เว็บไซต์ของโปรแกรม

ถือเป็นหัวใจหลักสำคัญของโปรแกรมเพื่อกลั่นกรองเอาเว็บไซต์ที่มีค่าความน่าสนใจสูง ๆ มาแสดงให้ผู้ใช้ โดยใช้หลักการของ Heuristic Function ที่จะนำเงื่อนไขหลาย ๆ เงื่อนไขมาร่วมกันคิด สำหรับการให้ค่าความน่าสนใจนั้นใช้ 3 เงื่อนไขดังนี้

1. <u>ค่าความสนใจของผู้ใช้ต่อประเภทเว็บไซต์</u> เป็นเงื่อนไขที่สำคัญที่สุด โดยนำเอา 2 ค่ามาคำนวณ ค่าแรก คือ ความสนใจต่อไดเรกทอรีชั้นที่ 1 ของลิงค์นั้น ซึ่งผู้ใช้ใส่ให้โปรแกรมตั้งแต่ครั้งแรกที่ผู้ใช้ใส่รายละเอียดของผู้ใช้ ค่านี้จะเก็บอยู่ในตัวแปรอาเรย์ชนิด Integer ชื่อ Active_Rate ในคลาส inform มี 14 ค่าตามจำนวนไดเรกทอรีชั้นที่ 1 สำหรับค่าที่สอง คือ ความสนใจต่อไดเรกทอรีที่ 1-3 ของลิงค์นั้นซึ่งผู้ใช้ได้สอนให้กับโปรแกรม จะถูกเก็บอยู่ในตัวแปรชนิด Vector ชื่อ Train_Rate ในคลาส inform ค่าที่สองนี้เป็นเสมือนค่าที่บวกเพิ่มหรือลดให้กับค่าแรกเท่านั้น ค่าความสนใจมีสูตรการคำนวณค่าดังนี้

ค่าความสนใจของผู้ใช้ = (((AR*1)+TR1)+((AR*2)+TR2) +((AR*3)+TR3))/6

โดยที่ AR = ค่า Active_Rate ของไดเรกทอรีชั้นที่ 1 ของเว็บไซต์นั้น เต็ม 100
TR1 = ค่า Train_Rate ของไดเรกทอรีชั้นที่ 1 ของเว็บไซต์นั้น
TR2 = ค่า Train_Rate ของไดเรกทอรีชั้นที่ 2 ของเว็บไซต์นั้น
TR3 = ค่า Train_Rate ของไดเรกทอรีชั้นที่ 3 ของเว็บไซต์นั้น

$-100 < ((AR*1)+TR1) < 100$

$-200 < ((AR*2)+TR2) < 200$

$-300 < ((AR*3)+TR3) < 300$

ค่าความสนใจของผู้ใช้ที่ได้จะปัดเศษขึ้น อนึ่ง หากลิงค์นั้นไม่สามารถระบุไดเรกทอรีได้ จะให้ค่าความสนใจของผู้ใช้เป็น 50 (ค่ากลาง)

2. <u>ค่าความน่าสนใจของเว็บไซต์ที่ได้จาก MetaCrawler</u> เป็นค่าที่ให้ความสำคัญรองลงมา จะนำค่าที่ได้จาก Rating ที่ MetaCrawler ให้แก่ลิงค์นั้นซึ่งเต็ม 1000 มาหาร 10 แล้วปัดเศษขึ้นได้เป็นเต็ม 100
3. <u>ค่าความน่าสนใจที่ได้จาก Neural Network</u> เป็นค่าที่ให้ความสำคัญน้อยที่สุดเนื่องจากมีค่าความผิดพลาดมาก ได้จากการใช้ Neural Network คำนวณข้อมูลของผู้ใช้ออกมาเป็นค่า

ความสนใจในประเภทไดเรกทอรีชั้นที่ 1 ค่าที่ได้ออกมาจะอยู่ระหว่าง 0 ถึง 1 จึงนำมาคูณ 100 แล้วปัดเศษขึ้น มีค่าเต็ม 100

เมื่อคำนวณค่าตามเงื่อนไขทั้ง 3 ค่าแล้ว จะนำมาให้อัตราส่วนตามความสำคัญแล้วคิดรวมกันตาม Heuristic Function ได้เป็นค่าความน่าสนใจของลิงค์นั้น มีสูตรดังนี้

ค่าความน่าสนใจของลิงค์ = ((User*3)+(MetaCrawler*2)+(NeuralNetwork*1))/6

ผลลัพธ์ที่ได้จะปัดเศษขึ้น ได้ค่าความน่าสนใจมีค่าเต็ม 100 สำหรับในกรณีที่ไม่สามารถระบุไดเรคทอรีให้ลิงค์ได้นั้นจะไม่รวมเอาเงื่อนไข Neural Network เข้าไปด้วย สูตรจะกลายเป็น

ค่าความน่าสนใจของลิงค์ = ((User*3)+(MetaCrawler*2))/5

อีกกรณีหนึ่งคือในสมุดเก็บลิงค์จะไม่มีค่าที่ได้จาก MetaCrawler เนื่องจากไม่ได้ไปค้นหาลิงค์มา หากลิงค์นั้นไม่สามารถระบุไดเรกทอรีได้ ค่าความน่าสนใจของลิงค์จะเป็น 0 หากมีไดเรกทอรีจะมีสูตรการคำนวณเป็น

ค่าความน่าสนใจของลิงค์ = ((User*3)+(NeuralNetwork*1))/4

4.2.5 งานการติดต่อภายนอก

เป็นการติดต่อสื่อสารกับเซิร์ฟเวอร์ปลายทางที่เราขอรับบริการ โดยอาศัยโพรโตคอลเอชทีทีพีผ่านซ็อกเก็ต มีขั้นตอนการทำงานหลัก ๆ คือ

- *สร้างเส้นทางการติดต่อไปยังเซิร์ฟเวอร์ที่ตนร้องขอรับบริการ*

  โดยการระบุเซิร์ฟเวอร์ที่เราต้องการติดต่อ โดยการเปิดซ็อกเก็ต และระบุพอร์ต โดยที่เซิร์ฟเวอร์จะใช้พอร์ตมาตรฐานคือ ที่พอร์ต 80

- *ส่งข้อความร้องขอ*

  พิจารณาเมธอด จากฟอร์มในเว็บเพจที่เซิร์ฟเวอร์นั้นให้บริการ รวมถึงตัวแปรที่ต้องส่งค่าไปให้เซิร์ฟเวอร์ว่ามีอะไรบ้าง

- *ยกเลิกเส้นทางการติดต่อ*

- *Parse ข้อมูล เพื่อให้ได้ข้อมูลตามเงื่อนไขที่ต้องการ*

  ดูเงื่อนไขที่ต้องการ และหาอัลกอริทึมที่เหมาะสม และนำมาปฏิบัติ

งานที่ใช้ติดต่อภายนอก มีอยู่ 3 งาน มีรายละเอียดการทำงานดังนี้

1. **งานค้นหาลิงค์ที่เกี่ยวข้องกับคีย์เวิร์ดที่ผู้ใช้ระบุ**

   ทำงานโดยวัตถุในคลาส MetacrawlerLinkParserThread เงื่อนไขที่ต้องการสำหรับงานนี้คือ ต้องการชื่อลิงค์, หัวข้อ (title), รายละเอียด (detail), และค่าความน่าสนใจของลิงค์นั้น เมื่อได้รับข้อมูลผลลัพธ์ที่ทางเซิร์ฟเวอร์ตอบสนองมาแล้วก็ทำการ Parse ข้อมูลออกเอาเฉพาะในส่วนที่เราต้องการ ทั้งนี้ผลลัพธ์ที่ได้จะเป็น HTML File ให้พิจารณาว่าข้อมูลที่เราต้องการอยู่ส่วนไหน ในที่นี้ข้อมูลของแต่ละลิงค์จะขึ้นต้นแท็ก <dt> ดังนั้นโปรแกรมจะทำการวนรอบซ้ำค้นหาแท็ก <dt> และหาแท็ก <dt>

อันต่อไป จัดการนำสตริงระหว่างนั้นมาพิจารณาลิงค์ หัวข้อลิงค์ รายละเอียด และค่าความน่าสนใจที่ทาง Metacrawler ระบุของลิงค์นั้นเก็บในตัวแปรอาร์เรย์ llink[ ], titlelink[ ], detaillink[ ] และ WeightMeta[ ] ตามลำดับ

เมื่อได้ลิงค์มา 1 ลิงค์จะทำการเรียกวัตถุของคลาส YahooDirParserThread ในการแยกประเภทของลิงค์ โดยอาศัยการแบ่งไดเรกทอรีของ Yahoo! พร้อมทั้งส่งตัวแปร llink[ ] ไปด้วย ซึ่งการทำงานของคลาส YahooDirParserThread มีการทำงานเป็นเธรด ฉะนั้นวัตถุที่เกิดจากคลาส MetacrawlerLinkParserThread จะต้องมีการรอให้เธรด YahooDirParserThread ที่ตนเองเรียกไปทำงานเสร็จสิ้นเรียบร้อยทั้งหมดก่อน จากการทำงานเป็นเธรดนี้ทำให้การทำงานของโปรแกรมมีความรวดเร็วขึ้น

2. งานแยกประเภทของลิงค์ โดยอาศัยการแบ่งไดเรกทอรีของ Yahoo!

งานนี้จะถูกเรียกจากวัตถุในคลาส MetacrawlerLinkParserThread เงื่อนไขที่ต้องการสำหรับงานนี้คือ ต้องการไดเรกทอรีของลิงค์ที่ระบุ เมื่อได้รับข้อมูลผลลัพธ์ที่ทางเซิร์ฟเวอร์ตอบสนองมาแล้วก็ทำการ Parse ข้อมูลออกเอาเฉพาะในส่วนที่เราต้องการ ทั้งนี้ผลลัพธ์ที่ได้จะเป็น HTML File ให้พิจารณาว่าข้อมูลที่เราต้องการอยู่ส่วนไหน ลิงค์หนึ่งอาจจะมีไดเรกทอรีมากกว่าหนึ่งก็ได้ เราต้องทำการกรองข้อมูล เอาแต่ไดเรกทอรีของลิงค์นั้นๆออกมา ในที่นี้แต่ละไดเรกทอรีจะขึ้นต้นด้วย <dt> <font face=arial size=-1> และจะตามด้วยชื่อไดเรกทอรีทีละชั้น โดยกั้นด้วย > เพราะฉะนั้นเราจึงทำการพิจารณาทีละไดเรกทอรี โดยเก็บชื่อของไดเรกทอรีชั้นที่ m ของไดเรกทอรีที่ n ไว้ที่ตัวแปร dir [n][m]

เนื่องจากคลาส YahooDirParserThread เป็นคลาสเธรด เมื่อทำงานเสร็จแล้วต้องมีประกาศบอกให้วัตถุที่เรียกคลาสนี้ (วัตถุของคลาส MetacrawlerLinkParserThread) รู้ว่าได้ทำงานเสร็จเรียบร้อยแล้ว โดยการกำหนดค่าที่ตัวแปร setalready และ ตัวแปร getdiralready ให้เป็น true เพื่อให้วัตถุทำการรับทราบ และนำไปประมวลต่อ

3. งานค้นหาลิงค์ที่น่าสนใจ

เงื่อนไขที่ต้องการสำหรับงานนี้คือ ต้องการลิงค์จำนวนที่ไม่มากจนน่าเบื่อ หรือน้อยจนเกินไปในไดเรกทอรีที่ระบุ ในที่นี้กำหนดอยู่ที่ 10 ลิงค์ หากว่าลิงค์ในไดเรกทอรีนั้นมีจำนวนน้อยกว่าที่ต้องการให้ทำการพิจารณาลงในไดเรกทอรีย่อยของไดเรกทอรีนั้น การทำงานคือ เมื่อได้รับข้อมูลผลลัพธ์แล้ว จำทำการเก็บชื่อของไดเรกทอรีย่อยไว้ในตัวแปร subDirectory เป็นตัวแปรชนิด Vector หลังจากนั้นทำการพิจารณาว่ามีลิงค์ในไดเรกทอรีนี้หรือไม่ หากมีก็ทำการกรองข้อมูลแล้วนำมาข้อมูลที่ต้องการอันได้แก่ ชื่อลิงค์ หัวข้อของลิงค์นั้น (title), รายละเอียดของลิงค์นั้น (detail) และ ไดเรกทอรีของลิงค์นั้น เก็บไว้ที่ตัวแปร iinterest, titleinterest, detailinterest และ directory ตามลำดับ ตัวแปรทั้งหมดนั้นเป็นชนิด Vector ที่ต้องมีตัวแปร directory ด้วยเพื่อระบุว่าลิงค์นั้นอยู่ในไดเรกทอรีอะไร เพื่อให้โปรแกรมหลักนำไปประมวลผลต่อไป

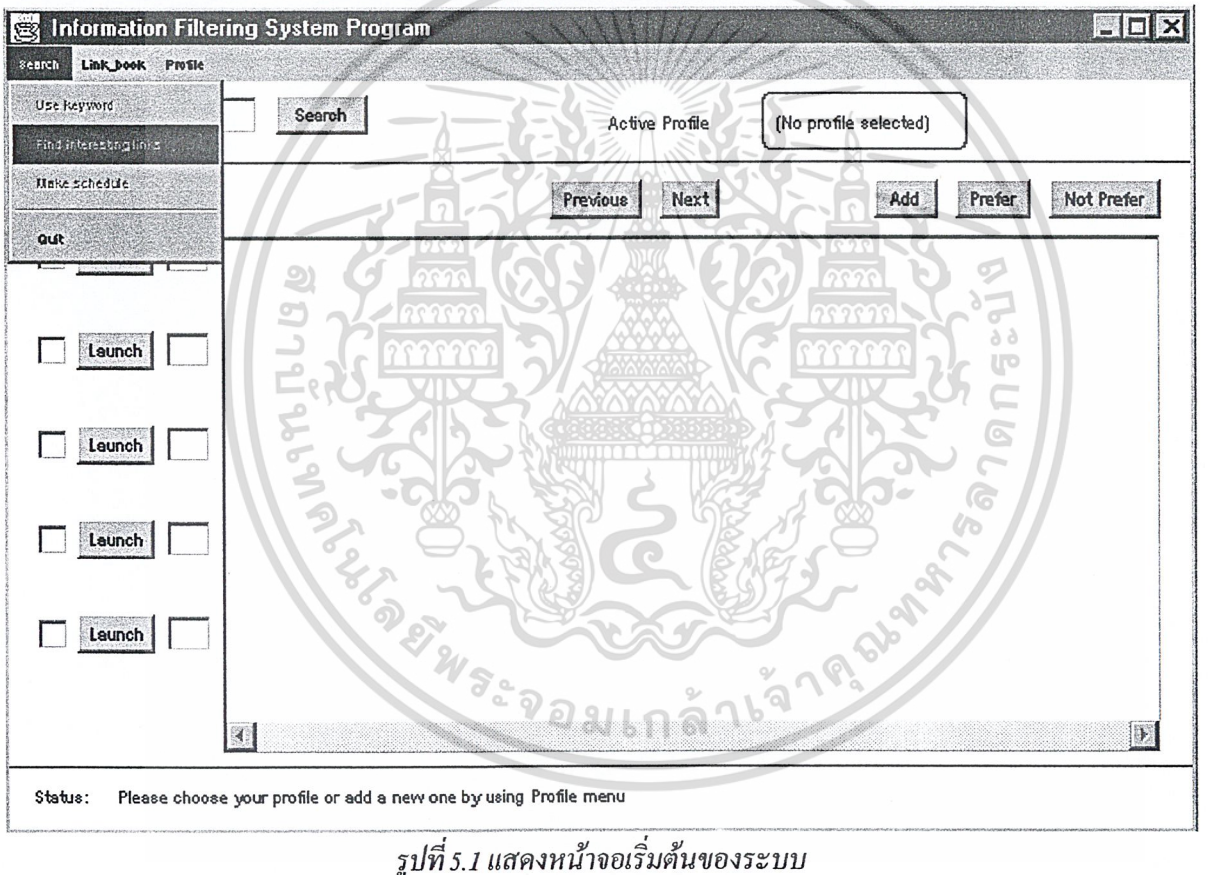
หากลิงค์ที่อยู่ในไดเรกทอรีปัจจุบันมีจำนวนน้อยกว่าจำนวนลิงค์ที่ต้องการ ก็จะสุ่มหาไดเรกทอรีย่อยจากไดเรกทอรีย่อยทั้งหมดออกมา แล้วทำการค้นหาลิงค์ในไดเรกทอรีย่อยนั้นต่อไปจนกว่าจะครบจำนวนลิงค์ตามที่ต้องการ

บทที่ 5
ผลการทดลองและการประเมินผล

5.1 ผลการทดลอง

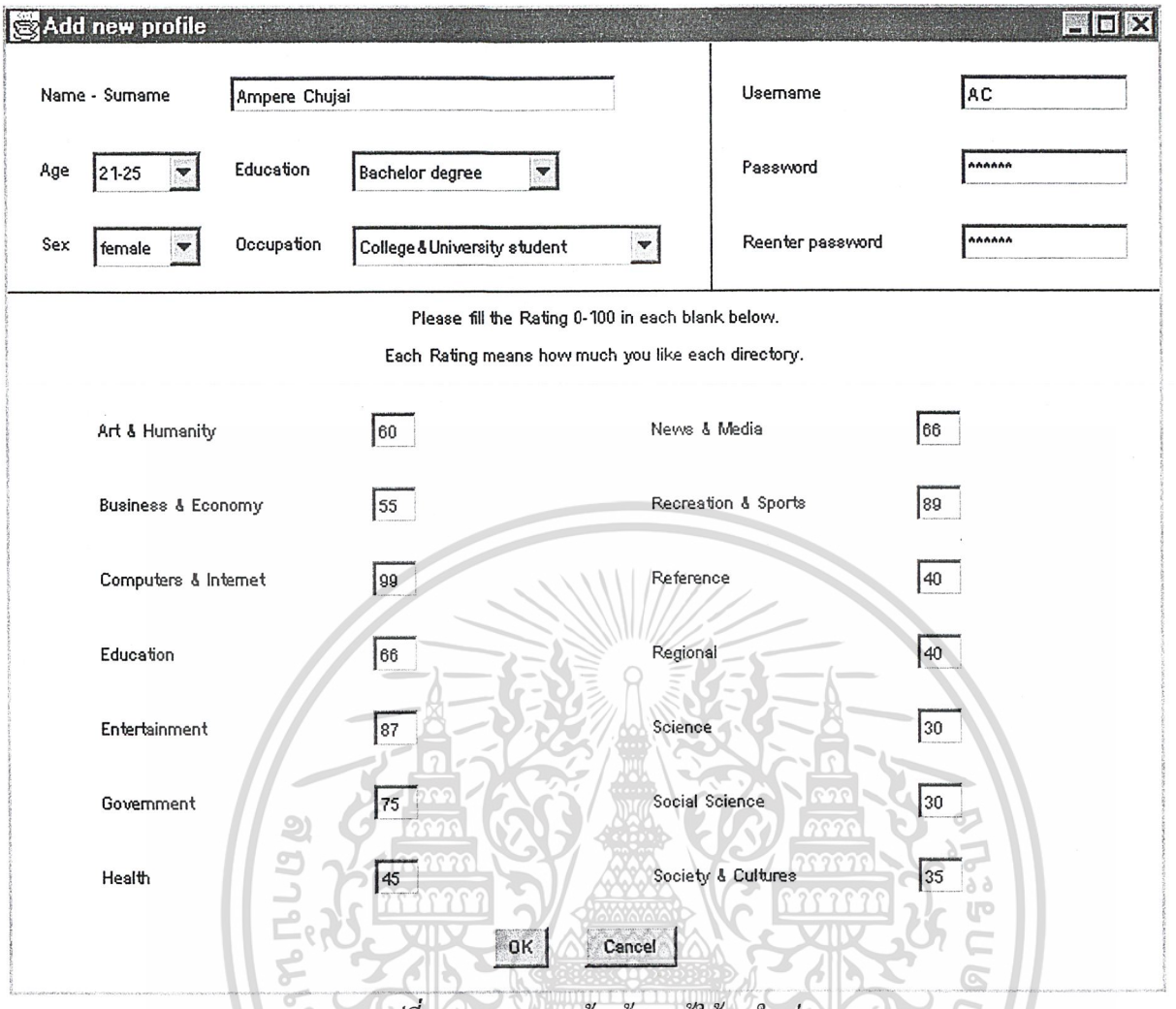
ผลการทดลองของระบบแสดงตามขั้นตอนการใช้งานดังนี้

1. หน้าจอเริ่มต้นของระบบเมื่อเริ่มเรียกใช้มีลักษณะดังรูปที่ 5.1 จากรูปสังเกตได้ว่าเราจะไม่สามารถเรียกใช้การทำงานอื่น ๆ ของระบบได้ เนื่องจากยังไม่ได้ระบุตัวผู้ใช้ให้ระบบทราบ



รูปที่ 5.1 แสดงหน้าจอเริ่มต้นของระบบ

2. เลือกเมนู Profile -> add new profile จะได้หน้าจอดังรูปที่ 5.2 เพื่อใส่ข้อมูลของผู้ใช้และความสนใจต่อเว็บไซต์ประเภทต่าง ๆ เมื่อใส่ครบแล้วจึงกดปุ่ม OK หากใส่ข้อมูลทุกอย่างถูกต้องจะกลับไปที่หน้าจอเริ่มต้น



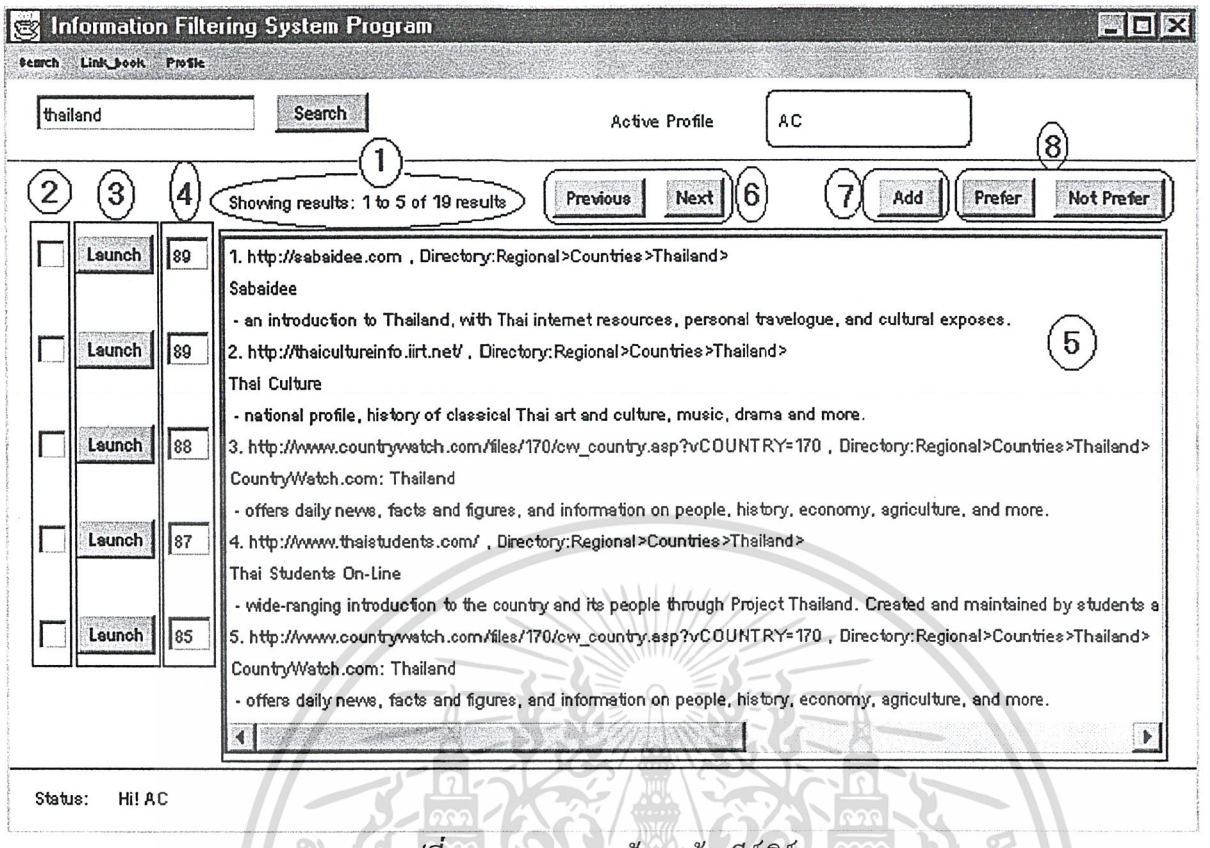
รูปที่ 5.2 แสดงการสร้างข้อมูลผู้ใช้คนใหม่

3. จะสามารถเริ่มใช้งานระบบได้ เริ่มจากการค้นหาด้วยคีย์เวิร์ด จากตัวอย่างในรูปที่ 5.3 ใช้คำว่า "thailand" ได้ผลลัพธ์ดังรูป รายละเอียดของส่วนต่าง ๆ ในหน้าจอนี้ตามที่แสดงในรูปมีดังนี้
   1. จำนวนลิงค์ที่ค้นหาได้และหมายเลขลิงค์ที่แสดงอยู่
   2. Checkbox สำหรับเลือกแต่ละลิงค์เพื่อใช้ร่วมกับปุ่ม Add, Prefer และ Not Prefer
   3. ปุ่ม Launch สำหรับเข้าไปดูลิงค์นั้น ๆ ผ่านทาง Internet Explorer
   4. Rating หรือค่าความน่าสนใจของแต่ละลิงค์
   5. รายละเอียดของลิงค์มีลิงค์ละ 3 บรรทัด ดังนี้

      <u>บรรทัดที่ 1</u> ประกอบด้วย 2 ส่วนคือ URL และ ไดเรกทอรีของลิงค์

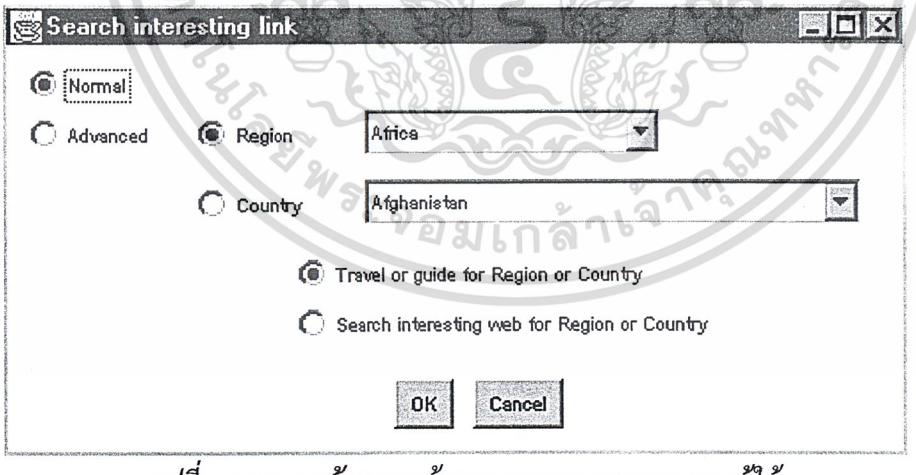
      <u>บรรทัดที่ 2</u> เป็นหัวข้อหรือ Title ของลิงค์

      <u>บรรทัดที่ 3</u> เป็นรายละเอียดของลิงค์
   6. ปุ่ม Prev และ Next ใช้เปลี่ยนหน้าเพื่อดูลิงค์ก่อนหน้าและลิงค์ถัดไป
   7. ปุ่ม Add สำหรับนำลิงค์ที่เลือกที่ Checkbox ไปเก็บไว้ในสมุดเก็บลิงค์
   8. ปุ่ม Prefer และ Not Prefer สำหรับให้ผู้ใช้สอนระบบว่าชอบหรือไม่ชอบลิงค์ที่เลือกไว้โดย Checkbox



รูปที่ 5.3 แสดงผลการค้นหาด้วยคีย์เวิร์ด

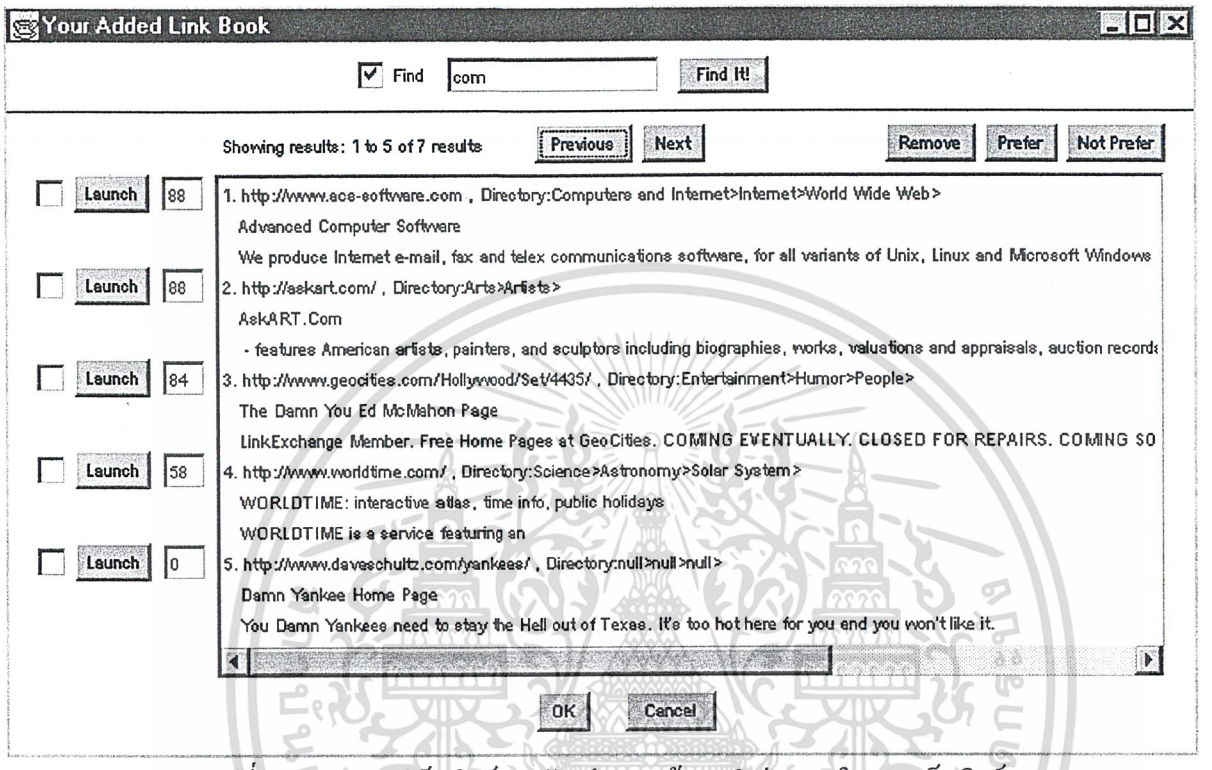
4. การค้นหาแบบตามความชอบของผู้ใช้สามารถทำได้โดยเลือกเมนู Search -> Find interesting links จะมีหน้าจอขึ้นมาดังรูปที่ 5.4 เพื่อให้เลือกว่าจะค้นหาแบบทั่วไปหรือจะระบุสัญชาติของเว็บไซต์ที่ต้องการค้นหาด้วย ผลลัพธ์ที่ค้นหาได้จะไปแสดงในสมุดเก็บลิงค์ที่น่าสนใจ



รูปที่ 5.4 แสดงหน้าจอการค้นหาแบบตามความชอบของผู้ใช้

5. เราสามารถตั้งเวลาให้ค้นหาแบบตามความชอบของผู้ใช้ได้โดยเลือกเมนู Search -> Make Schedule

6. เรียกดูสมุดเก็บลิงค์ได้ที่เมนู Link_book ซึ่งมีอยู่ 2 เล่ม คือ สมุดเก็บลิงค์ที่ต้องการเก็บไว้(Your added link book) และ สมุดเก็บลิงค์ที่น่าสนใจ(Your interesting link book) ซึ่งทั้ง 2 เล่มมีหน้าจอแสดงผลเหมือนกันดังแสดงในรูปที่ 5.5 หน้าจอจะคล้ายคลึงกับหน้าจอเริ่มแรก แต่จะมีเพิ่มมาที่การค้นหาคำใดๆ ในสมุดเก็บลิงค์นั้นได้ด้วย ดูตัวอย่างการค้นหาคำได้ในรูปที่ 5.5 เช่นกัน



รูปที่ 5.5 แสดงสมุดเก็บลิงค์และตัวอย่างการค้นหาคำว่า com ในสมุดเก็บลิงค์

7. การจัดการเกี่ยวกับข้อมูลผู้ใช้นั้นให้เลือกเมนู Profile สำหรับการสร้างข้อมูลผู้ใช้คนใหม่นั้นได้อธิบายไปในข้อที่ 2 แล้ว ส่วนการเลือกข้อมูลผู้ใช้(Choose active profile)และการลบข้อมูลผู้ใช้(Delete profile)นั้นการทำงานคล้ายคลึงกันคือเลือกชื่อระบุของผู้ใช้(username)แล้วจึงใส่รหัสผ่าน สำหรับการแก้ไขข้อมูลผู้ใช้นั้นมีหน้าจอคล้ายคลึงกับการสร้างข้อมูลผู้ใช้ใหม่มากต่างกันตรงที่การใส่รหัสผ่านนั้นเปลี่ยนเป็นการเปลี่ยนรหัสผ่าน

5.2 การประเมินผล

5.2.1 ส่วนปฏิบัติการ

ส่วนของ Graphic User Interface นั้นค่อนข้างราบเรียบและไม่ง่ายต่อการใช้งานนัก แต่ยังคงมีฟังก์ชั่นการทำงานหลัก ๆ ที่จำเป็นครบถ้วน มีการแบ่งสัดส่วนการเรียกใช้คลาสอื่น ๆ ตามฟังก์ชั่นการทำงานอย่างเหมาะสม เช่น คลาส NewVEdit_Profile ไว้สำหรับ เพิ่มข้อมูลผู้ใช้คนใหม่หรือแก้ไขข้อมูลผู้ใช้ ซึ่งในคลาสเดียวสามารถทำงาน 2 อย่างซึ่งมีลักษณะคล้ายคลึงกันได้

การใช้ภาษาจาวานั้นมีข้อดีในส่วนอื่น ๆ อีก เช่น การตั้งเวลา ซึ่งใน JDK 1.3 นี้มีเมธอด schedule ให้ใช้ ซึ่งจะทำงานโดยเรียก Thread ที่ป้อนให้มาทำงาน เมื่อถึงเวลาที่กำหนด ทำให้ง่ายต่อการเขียนโปรแกรมยิ่งขึ้น และมีฟังก์ชันการทำงานให้ใช้ได้หลากหลาย สะดวกต่อการพัฒนาโปรแกรม เนื่องด้วยภาษาจาวาถือได้ว่ามีลักษณะเป็นอ๊อบเจคท์โอเรียนเท็ด โดยสมบูรณ์ ดังนั้นจึง ควรทำการออกแบบระบบให้ดี จึงจะทำให้ได้ระบบที่มีประสิทธิภาพที่ดียิ่งขึ้น

5.2.2 ส่วนติดต่อภายนอก

ส่วนการติดต่อกับเซิร์ฟเวอร์ที่ต้องการขอรับบริการโดยการเปิดซ็อกเก็ตกระทำได้ง่าย เพราะมีฟังก์ชั่นการทำงานที่สามารถนำมาใช้ได้เลย หากแต่ต้องละเอียดรอบคอบและปฏิบัติตามข้อตกลงของการติดต่อสื่อสารผ่านเน็ตเวิร์กที่มี มิฉะนั้นทางเซิร์ฟเวอร์เองก็ไม่สามารถให้บริการแก่เราได้

เป็นที่ทราบกันดีอยู่แล้วว่าภาษาจาวาเหมาะกับการทำงานผ่านเน็ตเวิร์ก และมีฟังก์ชั่นการจัดการสตริงที่ดี ทำให้กระบวนการ Parse ข้อมูลไม่ยากดั่งที่เข้าใจไว้ในตอนแรก เพียงแต่การจัดการค่าตัวแปรต่าง ๆ เราต้องเข้าใจลักษณะอ๊อบเจ็คท์โอเรียนเท็ด เนื่องจากภาษาจาวาถือได้ว่ามีลักษณะเป็นอ๊อบเจคท์โอเรียนเท็ด โดยสมบูรณ์ ดังนั้นจึงควรทำการออกแบบระบบให้ดี จึงจะทำให้ได้ระบบที่มีประสิทธิภาพที่ดียิ่งขึ้น

ด้วยความที่เป็นอ๊อบเจคท์เต็มรูปแบบของภาษาจาวา เราจึงนำเธรดมาใช้ โดยระบบนี้นำเธรดมาใช้ ซึ่งทำให้โปรแกรมสามารถลดเวลาในการประมวลผลลงได้ ตัวอย่างในระบบคือ คลาส MetacrawlerLinkParserThread เมื่อกรองได้ลิงค์ที่นำไปแยกประเภทแล้ว จะทำการสร้างวัตถุของคลาส YahooDirParserThread ซึ่งเป็นคลาสเธรดให้ไปหาไดเรกทอรีของลิงค์นั้น ทีแรกเมื่อเริ่มสร้างระบบ ไม่ได้ใช้เธรด เวลาในการหาไดเรกทอรีสำหรับ 20 ลิงค์อยู่ที่ 45-50 วินาที แต่เมื่อมาใช้เธรดในการทำงานทำให้สามารถลดเวลาในการทำงานเหลือเพียง 15-20 วินาที

เนื่องจากการติดต่อขอใช้บริการจากเซิร์ฟเวอร์ภายนอกซึ่งเราไม่สามารถควบคุมได้ จึงอาจเกิดปัญหาในกรณีที่เซิร์ฟเวอร์มีการเปลี่ยนแปลงรูปแบบการให้บริการ

5.2.3 ภาพรวมของโปรแกรม

โปรแกรมเขียนด้วยภาษาจาวา ทำให้สามารถทำงานได้กับทุก ๆ ระบบปฏิบัติการ แต่สำหรับโปรแกรมนี้จะมีฟังก์ชั่นการทำงานบางอย่างที่จะใช้ได้เฉพาะบนระบบปฏิบัติการ Windows เท่านั้น เช่น ปุ่ม Launch ที่ใช้เพื่อเข้าไปดูลิงค์ที่ต้องการนั้น จะเรียกใช้ Internet Explorer ซึ่งหากไม่มี Internet Explorer จะไม่สามารถใช้งานปุ่มนี้ได้ หากต้องการให้ใช้งานได้จำเป็นต้องแก้ไขโค้ดของโปรแกรมเล็กน้อย เพื่อให้เรียกใช้ Browser ที่ใช้อยู่ในระบบปฏิบัติการนั้น

ในส่วนการคำนวณค่าความน่าสนใจของเว็บไซต์นั้นค่อนข้างแม่นยำและมีประสิทธิภาพ เนื่องจากประกอบด้วย 3 องค์ประกอบหลักที่สำคัญ ๆ (ดูรายละเอียดการคำนวณได้ที่หัวข้อ 4.2.4 การให้ค่าความน่าสนใจแก่เว็บไซต์ของโปรแกรม) แต่การคำนวณค่านี้ยังมีจุดบกพร่องบ้างในส่วนการคำนวณค่าความน่าสนใจของเว็บไซต์ที่ไม่มีสามารถระบุไดเรกทอรีและเว็บไซต์ที่เก็บไว้ในสมุดเก็บลิงค์ ที่จะต้องตัดบางองค์ประกอบออกไป

สำหรับการแบ่งประเภทเว็บไซต์นั้น ค่อนข้างครอบคลุมเว็บไซต์ทุกประเภทได้ เนื่องจากได้รูปแบบการแบ่งประเภทไดเรกทอรีนี้มาจาก Yahoo ซึ่งได้พัฒนารูปแบบไดเรกทอรีมากว่า 6 ปีแล้ว แต่อย่างไรก็ดี การแบ่ง

ประเภทของไดเรกทอรีของ Yahoo นี้เป็นแบบที่เหมาะสมกับการจัดการฐานข้อมูลของ Yahoo ซึ่งยังขัดอยู่บ้างกับการค้นหาของโปรแกรม เช่น ไดเรกทอรี Regional ซึ่งเก็บข้อมูลเกี่ยวกับประเทศหรือทวีปต่าง ๆ ไดเรกทอรีย่อยในประเทศหรือทวีปเหล่านั้นบางส่วนจะยังคงแบ่งตามแบบ 14 ไดเรกทอรีหลัก(ไดเรกทอรีชั้นที่ 1 ยกเว้นไดเรกทอรี Regional) แต่จะมีไดเรกทอรีพิเศษบางส่วนเสริมเข้ามาด้วย ซึ่งโปรแกรมจะไม่สามารถระบุไดเรกทอรีส่วนที่เสริมเข้ามานอกเหนือจากไดเรกทอรีหลักได้ เนื่องจากมีความลึกเกินชั้นที่ 3 ที่โปรแกรมเก็บอยู่และจะไม่มีชื่ออ้างอิงถึงไดเรกทอรีเหล่านี้

การค้นหาโดยผ่าน Search Engine ถึง 2 ตัวนั้นมีทั้งข้อดีและข้อเสียต่างกันไป ข้อดีคือ แบ่งงานกันเป็นสัดส่วนอย่างชัดเจนให้ MetaCrawler หาคำด้วยคีย์เวิร์ดและให้ Yahoo แบ่งประเภทเว็บไซต์ที่ได้ แต่ข้อเสียก็คือ ยิ่งใช้ Search Engine มากขึ้นจะยิ่งมีปัญหามากขึ้น รวมไปถึงการทำงานที่จะต้องมารอกัน จึงได้แก้การทำงานในส่วนที่ต้องรอกันนี้ให้เป็นเธรด คือ อ่านได้ลิงค์หนึ่งก็จะส่งไปแบ่งประเภททันที ทำให้รวดเร็วขึ้นมาก

สำหรับในส่วนการทำงานของ Neural Network ที่นำมาใช้ในการพิจารณาค่าความสนใจของผู้ใช้นั้น เนื่องจากลักษณะของ Neural Network คือเป็นการเรียนรู้ จดจำ เหมาะกับการทำงานที่มีผลลัพธ์ที่แน่นอนต่ออินพุตใดๆ ดังนั้นผลลัพธ์ที่ได้จากการคำนวณจึงไม่เป็นผลที่น่าพอใจเท่าใดนัก เนื่องจากอินพุตคือข้อมูลส่วนบุคคลของแต่ละคน ส่วนผลลัพธ์คือ ค่าความสนใจในหัวข้อต่าง ๆ แต่ละคนสามารถสนใจ หัวข้อเดียวกัน หรือต่างหัวข้อกันก็ได้ ไม่ใช่รูปแบบตายตัวแน่นอน ซึ่งไม่เหมาะที่จะนำ Neural Network มาใช้พยากรณ์ว่าบุคคลใด ชอบเรื่องใดได้อย่างถูกต้อง หากแต่เป็นเพียงทางเลือกที่ใช้ชี้ว่า บุคคลส่วนใหญ่ชอบน่าจะชอบและสนใจหัวข้อดังผลลัพธ์ที่ได้

บทที่ 6
บทสรุปและข้อเสนอแนะ

6.1 บทสรุป

ปัจจุบัน Search Engine ใหม่ ๆ ที่เกิดขึ้นในยุคหลัง ๆ นี้มีความแม่นยำในการค้นหามากยิ่งขึ้น มีการค้นหาและการแสดงผลที่หลากหลายยิ่งขึ้น แต่อย่างไรก็ดี ระบบกลั่นกรองสารสนเทศถือเป็นอีกทางเลือกใหม่สำหรับการค้นหาเว็บไซต์ซึ่งถือเป็นอีกมิติใหม่หนึ่งสำหรับการค้นหาเว็บไซต์ เพราะสามารถค้นหาได้ตามความชอบของผู้ใช้และยังสามารถเรียนรู้ความชอบของผู้ใช้เพิ่มเติมได้อีก รวมไปถึงการคาดเดาความชอบของผู้ใช้โดยการประเมินข้อมูลของผู้ใช้เทียบกับคนส่วนมากด้วย

ระบบกลั่นกรองสารสนเทศเป็นระบบการค้นหาเว็บไซต์ที่เน้นด้านความสนใจของผู้ใช้เป็นหลัก ซึ่งในความเป็นจริงแล้ว มนุษย์แต่ละคนมีความสนใจที่หลากหลายไม่แน่นอนและไม่จำกัด ทำให้ไม่สามารถกำหนดวิธีการกลั่นกรองอย่างตายตัวหรือเจาะจงได้ การแบ่งประเภทเว็บไซต์เป็นเพียงวิธีการหนึ่งที่ใช้ตีกรอบความสนใจของผู้ใช้ให้ระบบทราบ ในอนาคตน่าจะมีวิธีการที่ชัดเจนกว่านี้ ทั้งด้านการแบ่งประเภทเว็บไซต์และการตีกรอบความสนใจของผู้ใช้

ในการตรวจสอบความเกี่ยวข้องของคีย์เวิร์ดที่ใช้ค้นหากับเว็บไซต์ที่ได้นั้น ระบบส่งให้กับ MetaCrawler ช่วยทำหน้าที่นี้ ซึ่งผลลัพธ์ที่ได้นั้นค่อนข้างน่าเชื่อถืออยู่แล้ว (ดูการให้คำนี้ในหัวข้อ 3.2.10 Search Engine ที่ใช้ในระบบกลั่นกรองสารสนเทศ) แต่หากจะตรวจสอบในส่วนนี้เองจะต้องมั่นใจว่ามีกระบวนการตรวจสอบความเกี่ยวข้องที่ดีและเหมาะสมเพียงพอ

ระบบกลั่นกรองสารสนเทศยังเป็นระบบใหม่ที่ต้องการการพัฒนาเพิ่มเติมต่อไป เพื่อให้ได้ผลลัพธ์ที่ดียิ่งขึ้น ระบบจะมีประโยชน์อย่างมากสำหรับการค้นหาข้อมูลในอินเทอร์เน็ตซึ่งถือเป็นกิจกรรมหลักสำหรับผู้ใช้อินเทอร์เน็ต การค้นหาที่แม่นยำจะช่วยลดเวลาที่เสียไปและได้ข้อมูลที่ตรงตามความต้องการของผู้ใช้ นอกจากนี้ยังสามารถนำไปประยุกต์ใช้กับระบบค้นหาข้อมูลอื่น ๆ ได้อีก เช่น ระบบห้องสมุด , ระบบค้นหาเพลง เป็นต้น

6.2 ข้อเสนอแนะส่วนปฏิบัติการ

ส่วนของ Graphic User Interface นั้นสามารถพัฒนาให้สวยงามน่าใช้ยิ่งขึ้น โดยใช้ Java Swing ซึ่งจะมีรูปแบบของอินเทอร์เฟซให้เลือกใช้มากมาย และควรพัฒนาให้ง่ายต่อการใช้งานยิ่งขึ้น อาจมีคำแนะนำทุกครั้งในระหว่างการใช้งานเพื่อให้ผู้ใช้สามารถใช้งานโปรแกรมได้อย่างมีประสิทธิภาพ

นอกจากนี้ยังสามารถพัฒนาด้านการแสดงผลอื่น ๆ เพิ่มเติมได้อีกมากมาย เช่น เปลี่ยนรูปแบบการแสดงผลโดยนำลิงค์ที่ได้ทั้งหมดมาเข้ากลุ่มที่คล้ายคลึงกันอีกทีแล้วจึงแสดงผล เช่น การค้นหาคำว่า Computer ก็อาจ

แสดงผลได้หลายกลุ่ม เช่น Learning Computer, Computer Hardware, Computer Software, Computer Accessories เป็นต้น สามารถดูตัวอย่างและรายละเอียดได้ที่ www.vivisimo.com หรือ www.queryserver.com

6.3 ข้อเสนอแนะในภาพรวมของโปรแกรม

ตรวจสอบระบบปฏิบัติการที่ใช้และเขียนโค้ดเพื่อรองรับระบบปฏิบัติการนั้น ๆ สำหรับส่วนที่ยังไม่สามารถทำงานได้ทุกระบบปฏิบัติการจริง ๆ เพื่อให้เป็นโปรแกรมที่สามารถทำงานได้ทุกระบบปฏิบัติการอย่างแท้จริงและเต็มประสิทธิภาพ

ในส่วนการคำนวณค่าความน่าสนใจของเว็บไซต์นั้น สำหรับ Neural Network จะต้องแก้ไขข้อมูลที่ใช้สอน Neural Network โดยทำการสำรวจความคิดเห็นชุดใหม่ให้เจาะจงมากกว่านี้ เช่น อาจลดเหลือเพียง ชอบมาก ชอบ ธรรมดา ไม่ชอบ ไม่ชอบมาก เป็นต้น เพื่อให้ได้ข้อมูลที่เห็นความแตกต่างอย่างชัดเจน และผลลัพธ์ที่ได้จาก Neural Network จะมีประสิทธิภาพมากขึ้น รวมไปถึงการให้อัตราส่วนทั้งส่วนของไดเรกทอรีและส่วนของ 3 องค์ประกอบที่ใช้คำนวณ ควรหาอัตราส่วนใหม่ที่เหมาะสมกว่านี้ เพื่อให้ค่าความน่าสนใจที่ได้แม่นยำและตรงกับความต้องการมากยิ่งขึ้น

ส่วนของ Search Engine นั้นอาจพัฒนา Search Engine ขึ้นมาใช้เอง เพื่อจะได้มีฐานข้อมูลเป็นของตัวเอง และพัฒนาตัว Search Engine ของตัวเองให้ฉลาดยิ่งขึ้นได้อีก หรืออย่างน้อยที่สุดควรมีฐานข้อมูลเป็นของตัวเอง เพราะการใช้ฐานข้อมูลของผู้อื่นนั้นไม่สามารถใช้ได้ตลอดไป อาจเกิดเหตุการณ์ที่ฐานข้อมูลของเขาปิดให้บริการ หรือ มีการเปลี่ยนแปลงฐานข้อมูลของเขาได้ ดังนั้นแนวทางการพัฒนาคือ มีการพัฒนา Search Engine ขึ้นเอง และทำการร้องขอรับบริการผ่านฟอร์ม HTML หรือเป็นโปรแกรมเฉพาะอย่างในโปรแกรมนี้ก็ได้ หากแต่ระบบใหม่ที่กล่าวนี้ต้องการทรัพยากรที่เหมาะสมสำหรับ Search Engine ที่ดีตัวหนึ่งคือ มีระบบการจัดการฐานข้อมูลที่ดี รองรับการขอเข้าใช้บริการได้ทั่วถึง และรวดเร็วพอที่จะทำให้ผู้ค้นหาพึงพอใจได้ สำหรับ Web Robot ที่จะใช้ในการดึงข้อมูลนั้นอาจพัฒนาขึ้นเองหรืออาจเลือกใช้ Web Robot ที่มีอยู่แล้ว ดูรายชื่อ Web Robot ได้ในภาคผนวก *ก.รายชื่อพร้อมรายละเอียดต่าง ๆ ของ Web Robot*

ในส่วนการแบ่งประเภทเว็บไซต์นั้น อาจพัฒนาให้เหมาะสมมากยิ่งขึ้นกว่านี้ได้ ไดเรกทอรี 14 ไดเรกทอรีหลักในชั้นที่ 1 อาจเปลี่ยนเป็นหมวดหมู่อื่น ๆ ที่เป็นประเภทหลัก ๆ มากกว่านี้ เช่น อาจรวม Science กับ Social Science เข้าด้วยกันได้ เป็นต้น แต่การจะจัดแบ่งไดเรกทอรีตามต้องการได้นั้นเป็นงานที่หนัก ต้องใช้เวลาและจำนวนคนมากที่จะจัดแบ่งเว็บไซต์ตามไดเรกทอรีที่กำหนดขึ้นเอง เพื่อให้ได้ฐานข้อมูลที่มีไดเรกทอรีตามต้องการได้

บรรณานุกรม

Fah-Chang Cheong, "Internet Agents : Spiders, Wanderers, Brokers and Bots," New Reader Publishing. (Web Robot Construction), P. 105-120

Alex Benson, Stephen J. Smith, "Data warehousing, Data mining & OLAP," McGraw-Hill (Neural Network, Nearest Neighbor and Clustering) ISBN 0-07-006272-2, P.407-438

Stuart J. Russell and Peter Norving, "Artificial Intelligence," McGraw-Hill (Heuristic Functions) ISBN 0-13-360124-2, P. 92-115

Simon Haykin, "Neural Network : A comprehension foundation," Macmillan Publishing Company, ISBN 0-02-352761-7, P. 142-165

Mark Watson, "Intelligent Java Application for the Internet and Intranets," Morgan Kaufmann Publishers. (Neural Network) ISBN 1-55860-420-0, P. 55-86

Merlink Hughes, Michael Shoffner, Derek Hamner, "Java Network Programming," Manning Publication Co. (Client-side networking), ISBN 1-88477-49-X , P. 251-266

กิตติ ภักดีวัฒนะกุล, "JAVA ฉบับโปรแกรมเมอร์," หจก. ไทยเจริญการพิมพ์, ISBN 974-7042-98-3

ดร.วีระศักดิ์ ซึงถาวร, "JAVA Programming Volume I," บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน) ISBN 974-534-242-4

ดร.วีระศักดิ์ ซึงถาวร, "Fundamental of JAVA Programming Volume II," บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน) ISBN 974-534-117-7

ทรงเกียรติ ภาวดี, "แกะรอย CGI เพื่อเขียนสคริปต์เรียกเพจเจอร์/มือถือผ่านเว็บ," บริษัท วิตตี้ กรุ๊ป จำกัด ISBN 974-87003-0-5

www.searchenginewatch.com, "Search Engine Watch : Tips About Internet Search Engine & Search Engine Submission"

www.zdnet.com/searchiq/directory/multi.html, "ZDNet: Meta Search Engines Reviewed"

ภาคผนวก

ก. รายชื่อพร้อมรายละเอียดต่าง ๆ ของ Web Robot

The JumpStation Robot

Homepage: http://www.stir.ac.uk/jsbin/js

Run by: Jonathon Fletcher, J.Fletcher@stirling.ac.uk

From: pentland.stir.ac.uk

User-Agent: JumpStation

RBSE Spider

Homepage: http://rbse.jsc.nasa.gov/eichmann/rbse.html

Run by: Dr. David Eichmann, eichmann@rbse.jsc.nasa.gov

From: rbse.jsc.nasa.gov (192.88.42.10)

User-Agent: RBSE-Spider/0,1a

The WebCrawler

Homepage: http://webcrawler.com/

Run by: Brian Pinkerton, bp@biotech.washington.edu

From: surfski.webcrawler.com

User-Agent: WebCrawler/2.0 libwww/3.0

The NorthStar Robot

Homepage: http://comics.scs.unr.edu:7000/top.html

Run by: Fred Barrie, barrie@unr.edu

http://comics.scs.unr.edu/people/barrie.html

Billy Barron, billy@utdallad.edu

http://www.utdallas.edu/acc/billy.html

From: frognot.utdallas.edu possibly other sites in utdallas.edu and in cnidir.org

User-Agent: NorthStar

W4 (World Wide Web Wanderer)

Run by: Matthew Gray, mkgray@mit.edu

User-Agent: WWWWanderer v3.0 by Matthew Gray, mkgray@mit.edu

The Fish Search

Homepage: http://www.win.tue.nl/help/help-on-fish-all.html

Written by: Paul De Bra, debra@win.tue.nl

http://www.win.tue.nl/win/cs/is/debra/

Software: file://ftp.win.tue.nl/pub/infosystems/www/

Paper: http://www.win.tue.nl/win/cs/is/reinpost/www94/

From: Not set, but it's usually run from www.win.tue.nl

User-Agent: Fish-Search-Robot

The Python Robot

Homepage: http://info.cern.ch/hypertext/www/Tools/Python/Overview.html

Written by: Guido van Rossum, Guido.van.Rossum@cwi.nl

Software: ftp://ftp.cwi.nl/pub/python/python0.9.8.tar.z

ftp://ftp.cwi.nl/pub/python/demo/www/

HTML Analyzer

Run by: James E. Pitkow, pitkow@aries.colorado.edu

MOMspider

Homepage: http://www.ics.uci.edu/WebSoft/MOMspider

Written by: Roy Fielding, fielding@ics.uci.edu

Software: ftp://liege.ics.uci.edu/pub/arcadia/MOMspider/MOMspider-1.00.tar.Z

Paper: http://www.ics.uci.edu/WebSoft/MOMspider/WWW94/paper.html

From: Can run from anywhere

User-Agent: MOMspider/1.00 libwww-per/0.40

HTMLgobble

Run by: Andread Ley, ley@rz.uni-karlsruhe.de

From: tp70.rz.uni-karlsruhe.de

User-Agent: HTMLgobble v2.2

Software: ftp://ftp.rz.uni-karlsruhe.de/pub/net/www/tools/htmlgobble.tar.gz

WWWW-the WORLD WIDE WEB WORM

Homepage: http://www.cs.colorado.edu/home/mcbryan/WWWW.html

http://www.cs.colorado.edu/home/mcbryan/WWWWintro.html

Run by: Oliver McBryan, mcbryan@piper.cs.colorado.edu

http://www.cs.colorado.edu/home/mcbryan/Home.html

From: piper.cs.colorado.edu

WM32 Robot

Homepage: http://www-ihm.lri.fr/~tronche/

Written by: Christophe Tronche, Christophe.Tronche@lri.fr

http://www-ihm.lri.fr/~tronche/

From: E-mail address of the operator, usually tronche@lri.fr

User-Agent: W3M2/x.xx

Websnarf

Run by: Charlie Stross, charless@sco.com

From: ruddles.london.sco.com

The Webfoot Robot

Run by: Lee McLouchlin, L.McLoughlin@doc.ic.ac.uk

http://web.doc.ic.ac.uk/f?/lmjm

From: phoenix.doc.ic.ac.uk

Lycos

Homepage: http://lycos.cs.cmu.edu/

Run by: Dr. Michael L. Mauldin, fuzzy@cmu.edu

http://fuzine.mt.cs.cmu.edu/mlm/home.html

From: fuzine.mt.cs.cmu.edu

User-Agent: Lycos.x.x

NIKOS

Home Page: http://www.rns.com/cgi-bin/nomad

Written by: Rockwell Network Systems

Aspider (Associative Spider)

Written by: Fred Johansen, fred@nvg.unit.no

http://www.nvg.unit.no/~fred/

From: fred@nova.pcc.unit.no

User-Agent: Aspider/0.09

SG-Scout

Homepage: http://www-swiss.ai.mit.edu/~ptbb/SG-Scout/SG-Scout.html

Run by: Peter Beebee, ptbb@ai.mit.edubeebee@parc.xerox.com

From: set to operator, usually from beta.xerox.com

User-Agent: SG-Scout

EIT Link Verifier Robot

Homepage: http://wsk.eit.com/wsk/dist/doc/admin/webtest/verify_links.html

Written by: Jim McGuire, mcguire@eit.com

Software: ftp://ftp.eit.com/pub/wsk/doc/README.verify_links

From: can be run by anyone from anywhere

User-Agent: EIT-Link-Verifier-Robot/0.2

NHSE Web Forager

Run by: Bob Olseon, olson@mcs.anl.gov

From: Usually *.mcs.anl.gov

User-Agent: NHSEWlaker/3.0

WebLinker

Homepage: http://www.cern.ch/WebLinker/

Written by: James Casey, casey@ptsun00.cern.ch

http://www.maths.tcd.ie/hyplan/jcasey/jcasey.html

Paper: http://www.cern.ch/WebLinker/Paper/Welcome.html

User-Agent: WebLinker/0.0 libwww-perl/0.1

Emacs W3 Search Engine

Homepage: http://www.cs.indiana.edu/elisp/w3/w3toc.html#SEC39

http://www.cs.indiana.edu/elisp/w3/w3_1.html

Written by: William M. Perry, wmperry@spry.com

http://www.cs.indiana.edu/hyplan/wmperry.html

From: Various machines

User-Agent: Emacs-w3/v*.*

Arachnophilia

Run by: Vince taluski, taluskie@utpapa.ph.utexas.edu

http://www.ph.utexas.edu/people/vince.html

From: halsoft.com

User-Agent: Arachnophilia

ChURL

Homepage: http://www.engin.umich.edu/~yunke/scripts/#Churl

Run by: Justin Yunke, yunke@umich.edu

http://www.engin.umich.edu/~yunke/

Software: http://www.engin.umich.edu:80/~yunke/feedback/

Mac WWWWorm

Written by: Sebastien Lemieux, lemieuse@ERE.Umontreal.CA

http://alize.ere.umontreal.ca:8001/~lemieuse/sebast.html

Tarspider

Homepage: http://www.chemie.fu-berlin.de/user/chakl/Spider.html

Run by: Olaf Schreck, chakl@fu-berlin.de

http://www.inf.fu-berlin.de/~weissheh/chakl/ChaklHome.html

From: chakl@fu-berlin.de

User-Agent: tarspider version

The Peregrinator

Homepage: http://www.maths.usyd.edu.au:8000/jimr/pe/Peregrinator.html

Run by: Jim Richardon, jimr@maths.su.oz.au

http://www.maths.usyd.edu.au:8000/jimr.html

User-Agent: Peregrinator-Mathematics/0.7

Checkbot

Written by: Dimitri Tischenko, D.B.Tischenko@TWI.TUDelft.NL

http://www.twi.tudelft.nl/People/D.B.Tischenko.html

Run by: Hans de Graaff, j.j.degraaff@twi.tudelft.nl

From: Usually dutifp.twi.tudelft.nl

User-Agent: checkbot.pl/x.x libwww-perl/x.x

Webwalk

Written by: Rich Testardi, rpt@fc.hp.com

User-Agent: webwalk

Harvest

Homepage: http://harvest.cs.colorado.edu/

Run by: Darren Hardy, hardy@bruno.cs.colorado.edu

Software: http://harvest.cs.colorado.edu/harvest/gettingsoftware.html

Papers: http://harvest.cs.colorado.edu/harvest/papers.html

From: bruno.cs.colorado.edu

Katipo

Homepage: http://www.vuw.ac.nz/~newbery/Katipo.html

Run by: Michael Newvery, Michael.Newbery@vuw.ac.nz

http://www.vuw.ac.nz/~newbery

From: Michael.Newbery@vuw.ac.nz

User-Agent: Katipo/1.0

Infoseek Robot

Written by: Steve Kirsch, stk@infoseek.com

From: corp-gw.infoseek.com

User-Agent: Infoseek Robot 1.0

Open Text Corporation Robot

Run by: Tim Bray, tbray@opentext.com

User-Agent: OMW/0.1 libwww/217

GetURL

Written by: James Burton, burton@cs.latrobe.edu.au/~burton/

Software: http://www.cs.latrobe.edu.au/~burton/Public/

User-Agent: GetUrl.rexx v1.0

The TkWWW Robot

Homepage: http://fang.cs.synyit.edu/Robots/tkwww.html

Written by: Scott Spetka, scott@cs.sunyit.edu

http://fang.cs.sunyit.edu/Robots/spetka.html

Software: http://fang.cs.sinyit.edu/Spetka/tkwww

A Tcl W3 Robot

Homepage: http://hplyot.obspm.fr/~dl/robo.html

Written by: Laurent Demailly, dl@hplyot.obspm.fr

Software: http://hplyot.obspm.fr/~dl/geturl.tcl

http://hplyot.obspm.fr/~dl/w3cli.tcl

From: Usually hplyot.obspm.fr

User-Agent: dlw3robot/x.y

TITAN

Written by: Yoshihiko Hayashi, hayashi@nttnly.isl.ntt.jp

From: Usually nttnly.isl.ntt.jp

User-Agent: TITAN/0.1

CS-HKUST WWW Index Server

Homepage: http://dbx.cd.ust.hk:8000/

Written by: Budi Yuwono, yuwono-b@cs.ust.hk

From: Uually dbx.cs.ust.hk

User-Agent: CS-HKUST-IndexServer/1.0

Spry Wizard Robot

Homepage: http://www.compuserve.com/wizard/wizard.html

Written by: Spry info@spry.com

Weblayers

Homepage: http://www.univ-paris8.fr/~loic/weblayers/

Written by: Loic Dachary, loic@afp.com

Software: ftp://www.univ-paris8.fr/~loic/weblayers/weblayers-0.0.tar.gz

User-Agent: weblayers/0.0

WebCopy

Homepage: http://www.inf.utfsm.cl/~vparada/webcopy.html

Written by: Victor Parada, vparada@inf.utfsm.cl

Software: ftp://ftp.inf.utfsm.cl/pub/utfsm/perl/webcopy.tgz

From: not set

User-Agent: Webcopy/(version)

Aretha

Written by: Dave Weiner, davew@well.com

Scooter

Homepage: http://scooter.pa.x.dec.com/

Written by: Louis Monier, monier@pa.dec.com

From: Usually scooter.pa-x.dec.com

User-Agent: Scooter/1.0

WebWatch

Homepage: http://www.xebsei.com/users/janos/specter/

Written by: Joseph Janos, janos@scepter.com

From: not set

User-Agent: WebWatch