เอเจนท์กรองไปรษณีย์

Mail Filtering Agent

โดย

นางสาวธิดารัตน์ ตันทะสุวรรณ

รหัส 44067024

อาจารย์ที่ปรึกษา

ดร.ภัทรชัย ลลิตโรจน์วงศ์

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาการเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2546

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	เอเจนท์กรองไปรษณีย์
นักศึกษา	นางสาวธิดารัตน์ ตันทะสุวรรณ
อาจารย์ที่ปรึกษา	ดร. ภัทรชัย ลลิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

การปรากฏขึ้นของเวิลด์ไวด์เว็บ และการเพิ่มขึ้นเรื่อย ๆ ของข้อความที่เครื่องอ่านได้ ทำให้เกิดการจัดประเภทของข้อความ ซึ่งกลายมาเป็นส่วนสำคัญของกลไกการเรียนรู้ วิธีการประยุกต์ใช้อันหนึ่งที่มีความเป็นไปได้ว่าจะมีผลกับผู้ใช้เกือบจะทุกคนที่ใช้อินเทอร์เน็ต นั่นคือ การกรองไปรษณีย์อิเล็กทรอนิกส์ เพราะว่าจำนวนของข้อมูลบนอินเทอร์เน็ตเพิ่มขึ้น จึงจำเป็นต้องมีเครื่องมือที่ดีกว่าเดิมในการจัดการกับข้อมูลที่ทะลักเข้ามาสูงขึ้นตามไปด้วย ในโครงงานนี้อธิบายวิธีการพัฒนาเอเจนท์ซึ่งจะมีการปฏิบัติงานอยู่ระหว่างผู้ใช้ และเครื่องมือจัดการอีเมล โดยใช้เทคนิคกลไกการเรียนรู้ จากการสังเกตผลตอบกลับ และการสร้างกฎที่ผ่านมา เพื่อให้สามารถทำงานติดต่อกับเครื่องมือจัดการอีเมลได้โดยตรง โดยใช้รายละเอียดของข่าวสารในอีเมลในการกรอง มีการทำงานร่วมกัน 3 ส่วน ได้แก่ mail interface, rule generation module และ classification engine ซึ่งเป็นแนวทางหนึ่งเพื่อให้ค้นพบกฎในการกรองไปรษณีย์อิเล็กทรอนิกส์

Title	A Mail Filtering Agents
Student	Miss Thidarat Thantasuwan
Advisor	Dr. Pattarachai Lalitrojwong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2003

Abstract

The emergence of the World Wide Web and the ever-increasing amounts of machine-readable text have caused text classification to become an important aspect of machine learning. One application that has the potential to affect almost every user of the Internet is e-mail filtering. As the volume of data on the Internet increases, the need for better tools handling this flood of data is also growing. This project report describes the development of the agent sitting between a user and a mail program. This agent application employs machine-learning techniques from observation, feedbacks and past rule generations. The agent is then able to interact with the mail program directly. Three modules consisting of the mail interface, the rule generation module and the classification engines communicate through shared files to discover rules for filtering e-mails.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานฉบับนี้สำเร็จลุล่วงได้ด้วยดี ด้วยคำแนะนำ และความช่วยเหลือเป็นอย่างดียิ่งของ ดร.ภัทรชัย ลลิตโรจน์วงศ์ อาจารย์ที่ปรึกษาวิชาโครงการพัฒนาระบบงาน

ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ทุกคนที่ให้คำแนะนำ ให้ความช่วยเหลือ และเป็นกำลังใจที่ดีตลอดมา จนกระทั่งโครงงานนี้สำเร็จลุล่วงได้ด้วยดี

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ ที่ให้ความรัก และกำลังใจ รวมทั้งให้การสนับสนุนในทุก ๆ ด้าน เสมอมา และตลอดไป

ธิดารัตน์ ตันทะสุวรรณ

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญรูป

รูปที่ หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เมื่อไม่กี่ปีที่ผ่านมา การใช้เครือข่ายและระบบการค้นคืนสารสนเทศ (Information Retrieval) บนเครือข่ายเพิ่มขึ้นอย่างมาก ซึ่งไม่ได้เกิดขึ้นเพียงแต่ในวงการธุรกิจ และวงการศึกษาเท่านั้น เนื่องจากจำนวนที่เพิ่มขึ้นของผู้ให้บริการสื่อกลางการใช้อินเทอร์เน็ต เช่น Compuserve หรือผู้ให้บริการอินเทอร์เน็ตที่มิอิทธิพลมาก การเพิ่มขึ้นของผู้ให้บริการจำนวนมาก ออกมาในรูปแบบที่ทำให้การใช้งานมีความเป็นส่วนตัวมากขึ้น ในปัจจุบัน เครือข่ายอินเทอร์เน็ต มีมากกว่า 31,000 เครือข่าย มีคอมพิวเตอร์มากกว่า 2 ล้านเครื่องที่เชื่อมต่อ และมีผู้ใช้อีเมล (ไปรษณีย์อิเล็กทรอนิกส์) มากกว่า 20 ล้านคน ซึ่งโดยเฉลี่ยได้รับอีเมลอย่างน้อย 50 ฉบับต่อวัน โดยที่เป็นจดหมายขยะถึงร้อยละ 40 ของอีเมลทั้งหมด และมีแนวโน้มที่จะเพิ่มขึ้นเรื่อย ๆ การเพิ่มขึ้นนี้นำไปสู่การปะทุของการใช้งานทรัพยากรข้อมูลที่มีค่าบนเครือข่าย เมื่อข้อมูลมีค่า การสืบค้นหรือการค้นคืนสารสนเทศที่สนใจ หรือที่เกี่ยวข้องจึงมีความยากลำบากมากขึ้น อย่างไรก็ตาม ถึงแม้ว่าปริมาณข้อมูลจำนวนมากจะถูกเก็บเสมือนแฟ้มข้อมูลที่อยู่ต่างระบบการทำงานกัน แต่ส่วนใหญ่มันก็จะอยู่ในรูปของอีเมล, USENET news, World Wide Web servers และ Gopher servers เป็นต้น และเครื่องมือต่าง ๆ ก็ถูกพัฒนาขึ้นมาเพื่อให้สามารถเข้าถึงทรัพยากรเหล่านั้น ทำให้การเข้าถึงแหล่งข้อมูลกลายเป็นเรื่องง่าย ประหยัดเวลา ถูกต้อง ตรงตามความต้องการ และมีผู้ใช้งานมีปริมาณมากขึ้น ทำให้มีความต้องการที่จะใช้โปรแกรมที่สามารถใช้งานง่าย จึงมีการวิเคราะห์เรื่องนี้ขึ้นมา

1.2 วัตถุประสงค์

การสร้างเอเจนท์กรองไปรษณีย์นั้น มีวัตถุประสงค์เพื่อออกแบบ และพัฒนาเอเจนท์ที่สามารถกรองเอาข้อมูลที่ต้องการจริง ๆ ให้ได้มากที่สุดที่เป็นไปได้ ซึ่งเหมาะสมกับเหตุผลตามความต้องการของผู้ใช้ ขณะเดียวกันต้องคัดเอาข้อมูลซึ่งไม่อยู่ในความสนใจของผู้ใช้ ออกให้มากที่สุดเท่าที่มันทำได้

1.3 แนวคิดที่ใช้ในการพัฒนาระบบงาน

เนื่องจากความต้องการข้อมูลของผู้ใช้แต่ละคนมีการเปลี่ยนแปลงบ่อยมาก ระบบการกรองจะต้องให้ความเป็นส่วนตัวสูง เพื่อให้ผู้ใช้พอใจตามความต้องการของตนเอง แบบจำลองที่ใช้จะต้องมีการเรียนรู้จากการศึกษา (เช่น การกำหนดกฎหรือฐานความรู้) หรือจากตัวอย่าง (โดยการสังเกตผู้ใช้) หรือ จากการรับคำสั่งจากผู้ใช้โดยตรง เมื่อความต้องการของผู้ใช้เปลี่ยนแปลง แบบจำลองก็ควรมีการปรับปรุงให้สอดคล้องกัน โดยใช้แนวคิดของอินเทลลิเจนท์เอเจนท์ ซึ่งเป็นวิวัฒนาการจากปัญญาประดิษฐ์ในการออกแบบ และพัฒนาระบบ ซึ่งความสามารถของอินเทลลิเจนท์เอเจนท์ คือ ปฏิบัติงานได้โดยอิสระ มีฐานความรู้พื้นฐานเกี่ยวกับงานที่เราต้องการให้มันทำ และพร้อมที่จะปฏิบัติงานเมื่อถึงเวลา

เอเจนท์กรองไปรษณีย์ซึ่งจะมีการปฏิบัติงานอยู่ระหว่างผู้ใช้ และเครื่องมือจัดการอีเมล โดยใช้เทคนิคกลไกการเรียนรู้อีเมล จากการรับคำสั่งจากผู้ใช้โดยตรง จากการสังเกตผลตอบกลับ และการสร้างกฎที่ผ่านมา เพื่อให้สามารถทำงานติดต่อกับเครื่องมือจัดการอีเมลได้โดยตรง โดยใช้รายละเอียดของข่าวสารในอีเมล และเงื่อนไขที่ผู้ใช้กำหนดขึ้นเองในการกรอง

1.4 ขั้นตอนการดำเนินงาน

การออกแบบและพัฒนาเอเจนท์กรองไปรษณีย์มีขั้นตอนการดำเนินงาน ดังนี้

1. ศึกษาแนวคิดและทฤษฎีที่เกี่ยวข้องกับเอเจนท์กรองไปรษณีย์
2. ศึกษารูปแบบของอีเมล การรับและส่งอีเมล และโพรโทคอลที่ใช้ในปัจจุบัน
3. ศึกษาแนวคิดและทฤษฎีการกรองไปรษณีย์ และรวบรวมข้อมูลการกรองของเครื่องมือจัดการอีเมลที่มีอยู่ในปัจจุบัน เช่น Hotmail และ Outlook Express
4. รวบรวมอีเมลเพื่อใช้เป็นตัวอย่างทดสอบการทำงาน
5. ออกแบบและพัฒนาระบบงาน
6. สรุปผลการดำเนินงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการออกแบบและพัฒนาเอเจนท์กรองไปรษณีย์ ได้แก่

1. เอเจนท์สามารถกรองเอาอีเมลที่ต้องการได้มากที่สุดที่เป็นไปได้ ซึ่งสอดคล้องกับความต้องการของผู้ ใช้ ขณะเดียวกันก็คัดเอาข้อมูลซึ่งไม่อยู่ในความสนใจของผู้ใช้ออกให้มากที่สุดเท่าที่มันทำได้

2. เอเจนท์สามารถแยกประเภทของอีเมล ซึ่งสอดคล้องกับความต้องการของผู้ใช้ ใส่ในแฟ้มได้อย่างถูกต้อง
3. เข้าใจหลักการและขั้นตอนการทำงานของเอเจนท์กรองไปรษณีย์ วิธีการรับ-ส่งอีเมล
4. ผู้ใช้สามารถเข้าถึง และจัดการอีเมลของตนได้ ทำให้ประหยัดเวลาในการค้นหาและประมวลผลได้ง่าย

บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 ความหมายและคุณสมบัติของเอเจนท์

อินเทลลิเจนท์เอเจนท์ เป็นโปรแกรมคอมพิวเตอร์ที่กระทำการโต้ตอบตามความต้องการหรือความจำเป็นของผู้ใช้ ซึ่งจะเกี่ยวข้องกับระดับความเป็นอิสระและความเป็นอัตโนมัติ นั่นคือ เอเจนท์สามารถกระทำการโต้ตอบและริเริ่มทำงานได้ โดยไม่ต้องมีคน หรือเอเจนท์ตัวอื่นมาบอกมันว่าต้องทำอะไรบ้างทีละขั้นตอน ซึ่งคุณสมบัติหรือคุณลักษณะพิเศษของเอเจนท์ แสดงได้ดังนี้ (Murch and Johnson. 1999)

- ความสามารถปรับตัวได้ (Adaptability)

  เอเจนท์ต้องสามารถทำงานได้บนหลาย ๆ แพลตฟอร์ม เครือข่าย และหลาย ๆ ระบบปฏิบัติการ และในขณะเดียวกัน ต้องสามารถแก้ปัญหาเกี่ยวกับทางเทคนิคได้ด้วยตนเองโดยไม่ต้องรับข้อมูลจากผู้ใช้

- ความสามารถในการเคลื่อนที่ (Mobility)

  เอเจนท์ต้องสามารถเดินทางเข้าไปในเครือข่าย และในอินเทอร์เน็ต ได้อย่างอิสระ ซึ่งอยู่กับการตัดสินใจภายในตัวของมันเอง เกี่ยวกับที่ที่มันจะเข้าถึงข้อมูล เพื่อให้บรรลุผลตามที่ตั้งไว้

- ลักษณะโปร่งใสและสามารถอธิบายได้ (Transparency and Accountability)

  เอเจนท์จะต้องมีความตรงไปตรงมา เข้าใจง่ายและเปิดเผยอย่างเต็มที่แก่เจ้าของ และผู้ใช้ถ้าต้องการ ต้องมีลักษณะเฉพาะสำหรับบันทึกเหตุการณ์ว่าอยู่ที่ไหน ทำอะไร ใครที่มันกำลังติดต่อด้วย และเกิดขึ้นเมื่อไหร่ รวมทั้งมันต้องจัดหาข้อมูลตามความต้องการ

- ความเข้มงวด (Ruggedness)

  ถ้าเอเจนท์ต้องการเข้าไปสำรวจเครือข่าย ทั้งที่มีขนาดเล็ก และใหญ่ มันจะต้องมีการทำงานที่เข้มงวดเหมือนกัน สามารถจัดการกับข้อผิดพลาด ปัญหาทรัพยากรน้อย เครื่องของผู้ให้บริการที่มีกำลังความสามารถต่ำ และข้อมูลที่ไม่สมบูรณ์ รวมทั้ง สามารถอธิบายได้เมื่อได้รับข้อมูล รหัสคำสั่ง หรืออื่น ๆ ที่ต่างชนิดกัน มันต้องสามารถแก้ปัญหาได้หลาย ๆ แบบ ตามที่มันสามารถทำได้ โดยปราศจากการแทรกแซงจากมนุษย์

- เริ่มต้นและหยุดการทำงานได้ด้วยตนเอง (Self-starters)

  เอเจนท์ต้องสามารถเริ่มต้นและหยุดการทำงาน บนพื้นฐานตามมาตรฐานของมัน และตัดสินใจเก็บรวบรวมข้อมูล เพื่อใช้ในการทำสิทธิพิเศษสำหรับเจ้าของ ความบ่อยในการทำงานของเอเจนท์ อาจบ่อยมากเท่าที่เป็นไปได้ อาจเป็นชั่วโมง เป็นวัน เป็นสัปดาห์ หรือเป็นเดือน เอเจนท์จะต้องสามารถตัดสินใจที่จะเริ่มต้นและหยุด และเมื่อใดจะต้องส่งผล และเมื่อใดจะทำการติดต่อกับผู้ส่ง

- เน้นศูนย์กลางที่ผู้ใช้ (User centered)

  เอเจนท์จะต้องให้ความสนใจกับเจ้าของของมันมากที่สุด และให้สิทธิพิเศษในการกำหนดค่า มันต้องรับผิดชอบหน้าที่ตามที่กำหนดไว้ และไม่ออกนอกลู่นอกทาง แต่มันอาจจะมีความสามารถคิดหนทางใหม่ที่เป็นไปได้ว่าจะประสบความสำเร็จ รวมทั้ง มันอาจจะเสนอแนวทางใหม่ให้ทำงานบรรลุผล หรือ แนวทางที่ถูกต้องในการคิด

2.2 ประเภทของเอเจนท์

เราสามารถกำหนดลักษณะของเอเจนท์ ตามประเภทต่าง ๆ ดังนี้ (Murch and Johnson. 1999)

- Intelligent agent เป็นประเภทที่มีความหมายกว้างมาก ตามที่ได้กล่าวถึงข้างต้น และอาจมีความหมายรวมไปถึงลักษณะของเอเจนท์ประเภทอื่นด้วย เป็นเรื่องใหญ่ที่มีการวิจัยกันมากที่สุด และได้รับความสนใจจากนักพัฒนาโปรแกรมมากที่สุดในทางการค้า
- Learning agent เป็นโปรแกรมเอเจนท์ที่มีการเรียนรู้ขั้นพื้นฐานจากผู้ใช้หรือเจ้าของ การเรียนรู้ในที่นี้ คือ การปรับพฤติกรรมที่ได้จากประสบการณ์ หรือจากการตัดสินใจ หน้าที่อย่างหนึ่งจากการเรียนรู้ คือ เอเจนท์ ต้องสามารถออกคำสั่ง หรือแนะนำวิธีที่จะทำให้ตัวมันเองพัฒนาขึ้นได้ กระบวนการการเรียนรู้จะค่อย ๆ เกิดขึ้น และมีการโต้ตอบระหว่าง เครื่องคอมพิวเตอร์กับผู้ใช้งาน ซึ่งเป็นระบบที่มีการประมวลผลเมื่อได้รับคำสั่งหรือข้อมูล บางครั้งอาจเรียกว่า Adaptive agent เช่น บริษัท Firefly มีเว็บไซต์ให้บริการในการแนะนำเกี่ยวกับดนตรี และแจ้งผู้ใช้ที่มีรสนิยมคล้ายกันให้พบปะกัน มันเรียนรู้เกี่ยวกับดนตรี ที่ผู้ใช้ชอบจากความต้องการของผู้ใช้

  การเรียนรู้ของเอเจนท์มีได้ 4 วิธี ดังนี้

  1. ดูจากงานที่อยู่ในความรับผิดชอบของผู้ใช้ สังเกตว่าผู้ใช้ทำอะไร แล้วเลียนแบบการกระทำนั้น
  2. เอเจนท์สามารถเสนอคำแนะนำหรือการทำงานของผู้ใช้ แล้วเรียนรู้โดยการรับผลที่คืนกลับมา หรือการสนับสนุนจากผู้ใช้

3. เอเจนท์สามารถรับคำสั่งที่ชัดเจนจากผู้ใช้ได้ (เมื่อเกิดเหตุการณ์อย่างนี้ ต้องทำอย่างนี้)
4. สอบถามจากเอเจนท์ตัวอื่น เพื่อรับคำแนะนำ และเรียนรู้ได้จากประสบการณ์ของมันเอง

- Believable agent เป็นเอเจนท์ที่ดูคล้ายกับมีชีวิต เคลื่อนไหวได้ หรือบางครั้งบุคลิกลักษณะของมันก็ทำให้มันดูน่าเชื่อถือ ซึ่งเป็นส่วนหนึ่งของเทคโนโนยีเอเจนท์ที่อยู่ในช่วงแรกของการเริ่มต้นพัฒนา และคงต้องใช้เวลาอีกนานที่จะได้รับการยอมรับ ในอนาคตเอเจนท์ประเภทนี้จะมีความฉลาดมากขึ้น Believable agent ในปัจจุบันยังไม่ค่อยฉลาดนัก แต่มันมีประโยชน์หลายอย่าง สามารถปรับตัวได้ และใช้งานได้อย่างเป็นมิตร สะดวกสบายในการติดต่อกับผู้ใช้ เอเจนท์ประเภทนี้แบ่งย่อยได้อีก คือ โปรแกรมที่ใช้เป็นของเล่นสำหรับเด็ก ที่เรียกกันว่า "virtual pet" (สัตว์เลี้ยงจำลอง) เด็ก ๆ ต้องให้อาหาร เลี้ยงดู และทำความสะอาด หลังจากสัตว์เลี้ยงจำลองทำธุระส่วนตัวเสร็จ เช่นเดียวกับการเลี้ยงสัตว์จริง ๆ เช่น ทามากอตจิ (tamagotchi) ซึ่งเป็นของเล่นจากบริษัท Bandai ประเทศญี่ปุ่น
- Mobile agent เป็นเอเจนท์ประเภทที่ใช้แนะนำ ชักชวน มีความคล่องแคล่ว กระตือรือร้น เคลื่อนไหวได้ เปลี่ยนแปลงได้รวดเร็ว และสามารถเดินทางได้ มันอาจจะอยู่ฝั่งผู้ให้บริการหรือฝั่งผู้รับบริการคอมพิวเตอร์ก็ได้ และสามารถเข้าไปในคอมพิวเตอร์เครื่องอื่น เครือข่ายอื่นได้ หรือเข้าไปในอินเทอร์เน็ต เพื่อปฏิบัติงาน การเก็บข้อมูล หรือ แลกเปลี่ยนข้อมูลได้

2.3 ภาษาและเครื่องมือ

อนาคตของเทคโนโลยีเอเจนท์ ขึ้นอยู่กับงานวิจัยว่า เอเจนท์จะเคลื่อนย้ายเดินทางไปได้อย่างไร และชนิดของสภาพแวดล้อมแบบใดที่สนับสนุนมัน และช่วยให้มันทำงานได้สำเร็จ ซึ่งระบบที่สนับสนุนการขนส่งเคลื่อนย้ายของเอเจนท์ มีดังนี้ (Murch and Johnson. 1999)

- Active X ของ Microsoft : Active X เป็นการพัฒนาที่ได้รับความสำเร็จของ Microsoft ที่เริ่มต้นใช้ OLE และทีมพัฒนาใช้ Visual Basic หรือ C ในการพัฒนาไฟล์ โดยใช้การขยาย OCX แม้ว่าไฟล์นี้จะสามารถเดินทางได้ แต่มันก็ประสบปัญหาเกี่ยวกับความปลอดภัยต่ำ และการขยายตัวของโปรแกรมซึ่งทำให้การดาวน์โหลดใช้เวลานาน
- Telescript ของ General Magic : General Magic เพิ่งจะเข้ามาใช้พื้นฐานของภาษาจาวา ทำ mobile agent เรียกกันว่า Odyssey และบริการของมันเรียกว่า Portico ซึ่งอนุญาตให้เข้าถึงข้อมูลที่ถูกรวบรวมโดยเอเจนท์ของคุณเองได้ตลอด 24 ชั่วโมง/วัน ผ่านทางเสียง หรืออินเทอร์เน็ตเบราเซอร์ General Magic เพิ่งเริ่มพัฒนาเทคโนโลยี mobile agent เมื่อไม่กี่ปีมานี้ ทำให้สภาพแวดล้อมของ General Magic ไม่ทันสมัย ใช้งานยาก และผู้ใช้ต้องทำงานบนภาษาดั้งเดิม เช่น ภาษาซี แม้ว่าการโฆษณาที่ออกมาของ General Magic จะมีการพัฒนาออกมาน้อยมาก แต่ก็มี

การสั่งซื้อเข้ามามาก อีกทั้ง telescript ต้องการให้ผู้ใช้ ทุกคนมี GM server ซึ่งคิดว่าไม่น่าเป็นไปได้

- HotJava ของ Sun Microsystems : Sun เป็นกลุ่มของการนำ mobile agent ด้วยจาวา ซึ่งเป็นภาษาที่สนับสนุนความสามารถในการเคลื่อนที่ จาวาถูกใช้ในกลุ่มนักพัฒนานับพันคนทั่วโลกเรียบร้อยแล้วในการพัฒนาเอเจนท์ขึ้นมาใช้งานจริง ด้วยเหตุนี้ ทำให้เกิดภาษาจาวาสำหรับการเคลื่อนที่ และเป็นอีกทางเลือกหนึ่ง จาวาใช้ Remote Method Invocation (RMI) เป็นกุญแจสำคัญสำหรับเทคโนโลยีการเคลื่อนที่ได้ อย่างไรก็ตาม ในปัจจุบัน RMI ยังไม่สนับสนุนในเรื่องของประสิทธิภาพ เช่น สถานะและการขนส่งข้อมูล ซึ่งเป็นสิ่งที่ต้องการใน mobile agent อย่างแท้จริง
- Voyager ของ Objectspace : เป็น Java-based Object Request Broker (ORB) ซึ่งสนับสนุน mobile agent แต่โชคไม่ดีที่เครื่องผู้บริการทั้งหมดต้องเป็น Voyager เท่านั้น
- Aglets ของ IBM : เป็นระบบที่ใช้ภาษาจาวา ที่บางอย่างคลอบคลุมมาตรฐาน RMI และเป็นการเพิ่มสมรรถนะของ mobile agent รวมทั้งไม่ต้องการเครื่องผู้ให้บริการอื่นที่มีสภาพแวดล้อมอื่นนอกจากจาวาของมันเอง ความต้องการเฉพาะนี้ อาจจะช่วยให้ IBM ได้รับการใช้งานอย่างมากจาก Microsoft บนอินเทอร์เน็ตในอนาคต
- Obliq ของ Digital : ใช้พื้นฐานของ Modula 3 (เป็นปาสคาลเชิงวัตถุ) ซึ่งจัดเตรียมเทคโนโลยีเคลื่อนที่ได้ของอินเทอร์เน็ตในอนาคต
- Safe-Tcl : TCL/TK เป็นระบบโปรแกรมที่ถูกพัฒนาโดย John Outerhout จากมหาวิทยาลัยแคลิฟอเนีย เบิร์กเลย์ ซึ่งใช้งานง่ายและมีการทำส่วนกราฟิกติดต่อกับผู้ใช้ที่เป็นประโยชน์มากใช้งานสะดวก อย่างไรก็ตาม ดูเหมือนว่าระบบนี้ไม่น่าเป็นไปได้ นั่นคือ มีรูปแบบแต่ละส่วนของ mobile agent ในอนาคตออกมาหลากหลายมากเกินไป

Safe-Tcl และ HotJava ไม่ได้เป็นระบบเอเจนท์ ที่ขนส่งข้อมูลอย่างสมบูรณ์ แต่มันอนุญาตให้เคลื่อนย้าย โค้ดได้ Tcl script ฝังในข้อความจดหมายได้ และถูกประมวลผลที่ฝั่งผู้รับ (active mail) HotJava อนุญาตให้ Java script ถูกฝังในเอกสารเวิลด์ไวด์เว็บได้ และถูกประมวลผลที่ฝั่งผู้ดู

2.4 การใช้งานและรูปแบบการกรองจดหมายบนอินเทอร์เน็ตในปัจจุบัน

การสำรวจจากสถาบันวิจัย Ferris แสดงให้เห็นว่าผู้ใช้ทางธุรกิจโดยเฉลี่ยได้รับข้อความ 30 ข้อความต่อวัน บริษัทที่ทำวิจัยการตลาดทำนายไว้ว่า ในอีกประมาณหนึ่งถึงสองปีข้างหน้า ตัวเลขนี้จะเพิ่มขึ้นเป็นอย่างน้อย 50 ข้อความต่อวัน ยิ่งไปกว่านั้น ในอนาคตอันใกล้นี้ จดหมายขยะ (spam mail) จะมีสูงขึ้นถึง 40 เปอร์เซ็นต์จากการสื่อสารทางอิเล็กทรอนิกส์ทั้งหมด (Group Technology AG. 2001)

ในปัจจุบัน มีโปรแกรมประยุกต์เกี่ยวกับอีเมล ของผู้ใช้ฝั่งผู้รับบริการ อยู่เป็นจำนวนมาก ซึ่งการกรองอีเมลถูกใช้และมีให้เห็น โดยเฉพาะอย่างยิ่งในโปรแกรมของ POP3 (Post Office Protocol 3 standard) แบบจำลองการกรองในความเป็นจริงแล้วง่ายมาก โดยผู้ใช้ตั้งแบบทดสอบขึ้นด้วยถ้อยคำสนทนาธรรมดา แล้วใช้มันเป็นกฎเกณฑ์ในการค้นหาคำ หรือชุดของคำที่เข้ากับข้อความในจดหมายที่มีรูปแบบเป็นมาตรฐาน จากนั้นจึงทดสอบส่วนของข้อความ และประเมินผลการทดสอบซึ่งจะกระทำโดยตรงที่โปรแกรมประยุกต์ฝั่งผู้รับบริการ ในเวลาที่ข้อความถูกส่งมาจากผู้ให้บริการหรือถูกทดสอบโดยโปรแกรมอื่น ๆ การกระทำที่เป็นผลลัพธ์เป็นไปได้ที่จะใช้เป็นผลของการทดสอบมาตรฐานการจับคู่ message - action ซึ่งจะถูกกำหนดโดยประสิทธิภาพของโปรแกรมอีเมลฝั่งผู้รับบริการ และจำกัดขอบเขตว่าข้อมูล และการกระทำอะไรที่โปรแกรมอีเมลจะเข้าถึงได้ เช่น เป็นการขนส่งข้อความในจดหมายไปยังที่เก็บจดหมายส่วนตัวที่เฉพาะเจาะจง หรือการยอมให้ข้อความใหม่ไปยังระบบการขนย้าย

เอเจนท์ที่ขนส่งอีเมลบนมาตรฐาน SMTP (Simple Mail Transfer Protocol) ทั้งหมดจะมีความสามารถในการประเมินข้อความ และความสามารถปฏิเสธได้ถ้าที่เก็บจดหมายเฉพาะนั้นไม่สามารถรับข้อความนั้น หรือข้อความอยู่ในรูปแบบที่ผิด หรือเหตุผลทางเทคนิคอื่น ๆ (เช่น เกี่ยวกับรูปแบบโครงสร้าง หรือการสร้างประโยค) ฝั่งผู้ให้บริการอีเมล ยังจัดเตรียมชั้นที่มีความสำคัญเพิ่มขึ้นสำหรับการส่งจดหมาย บนพื้นฐานที่เกี่ยวข้องกับความแตกต่างของคำในข้อความไว้อีกด้วย ตัวอย่างเช่น จดหมายขยะ (junk mail หรือ spam) หรือปฏิเสธเนื้อหาที่ผิดปกติมากๆ ของข้อความ ซึ่งไม่เหมาะสมที่จะส่งในผู้รับ เช่น ผู้เยาว์หรือผู้ที่มีการศึกษาน้อย (Wall. 2001)

เวลาผ่านมาเพียงไม่นาน แต่การพัฒนาในตลาดเครื่องมือจัดการอีเมลเพิ่มความสำคัญขึ้นอย่างมาก นั่นคือ การแบ่งประเภทของผลิตภัณฑ์อย่างเฉพาะเจาะจงโดยมีเจตนาที่จะกรองจดหมายที่จะส่งออกไป โดยพิจารณาจากรายละเอียด บางองค์กรใช้วิธีการนี้เพื่อป้องกันเกี่ยวกับกรรมสิทธิ์หรือข้อมูลที่เป็นส่วนตัวจากการรั่วไหลในองค์กร ตามเหตุผลซึ่งยึดหลักปฏิบัติที่เห็นแก่ประโยชน์ของส่วนรวมเป็นที่ตั้ง ที่มีการใช้วิธีการนี้ก็เพื่อป้องกันการแพร่กระจายของไวรัสทางจดหมายตามมาตรฐานของ MIME

ยังมีเครื่องมืออื่น ๆ สำหรับผู้ใช้ในการกรองอีเมลของตน ที่ได้รับความนิยม เช่น โปรแกรม Procmail ซึ่งทำงานอยู่ระหว่างผู้ให้บริการ SMTP และปลายทางที่เก็บจดหมาย บนระบบการแบ่งกันใช้เวลา (time-sharing) มีเหตุผลหลายประการที่ใช้วิธีนี้ เช่น ผู้ให้บริการอีเมลบนมาตรฐาน POP อาจจะต้องการที่จะมีระดับการทำงานเดียวกับการกรองที่วิธีการของ POP จัดเตรียมให้

หรือผู้จัดการระบบอาจจะต้องการใช้มาตรฐานการกรองสำหรับกลุ่มผู้ใช้บางกลุ่ม มาตรฐานก็ควรจะเป็นแบบที่เฉพาะเจาะจง ดังนั้น จึงไม่เหมาะสมกับ การส่งและกรองประเภท SMTP หรือผู้ใช้ระบบ POP อาจจะต้องการจัดเรียงจดหมายไว้ก่อน เพื่อลดค่าใช้จ่ายในการดาวน์โหลดจดหมายสำหรับผู้ใช้ที่ไม่สนใจที่จะรับอีเมลนั้น อีกอย่างหนึ่ง ผู้ใช้ POP อาจจะต้องการวิธีการกรองอีเมลตามมาตรฐานของตนเอง โดยปราศจากการร้องขอความช่วยเหลือจากระบบ POP ในการตรวจสอบ เช่น กรณีกระบวนการ "vacation" เพื่อบอกกับผู้ที่ติดต่อกันทางจดหมายว่าผู้ใช้ไม่จำเป็นต้องอ่านจดหมายนั้น

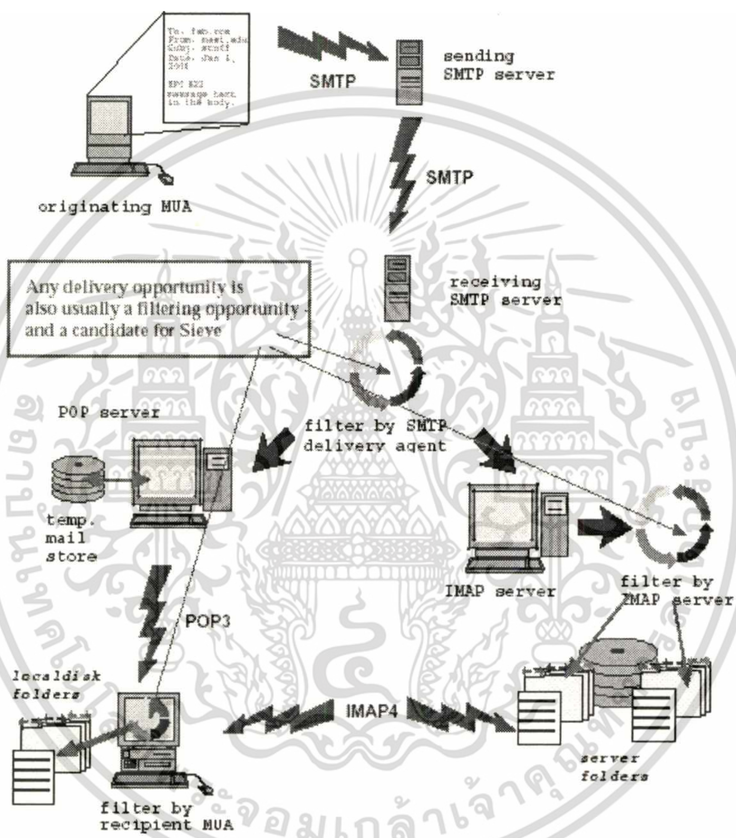
ยังมีอีกหลายกรณี ในรูปแบบการกรองที่ถูกใช้ภายในระบบอีเมล ตั้งแต่เริ่มต้นส่งข้อความไปยังกล่องอีเมลของผู้ใช้ เช่น ในกรณีที่ผู้ใช้หรือผู้จัดการระบบคัดแยกจดหมายที่เก่า แล้วออกจากระบบอีเมลบนมาตรฐาน IMAP (Interactive Mail Access Protocol) เพื่อให้เหลือที่ว่าง หรือทำการตรวจสอบเป็นช่วงเวลาตามมาตรฐานของ MIME เพื่อป้องกันไวรัส เครื่องมือในการจัดการระบบปฏิบัติการทดสอบและกระทำนี้ เป็นการกระทำในรูปแบบของการกรอง แต่ไม่ใช่การกรองในช่วงเวลาของการส่งอีเมล

ดังนั้น MUA (Mail User Agent) จึงมีการกระทำบางอย่างเพื่อให้ความสะดวกแก่ผู้ใช้ การกรองต้องสามารถปฏิบัติการได้ในหลายระดับในระบบอีเมล เพื่อจุดประสงค์ที่หลากหลาย (อาจจะต้องการการทำงานที่ขนานในเวลาการส่งที่แตกต่างกัน แม้ว่าจะไม่เป็นการจัดคู่หนึ่งต่อหนึ่งที่สมบูรณ์) โดยทั่วไป มีรูปแบบการกรองหลัก ๆ อยู่อย่างน้อย 3 รูปแบบ ซึ่งสามารถทำงานเป็นอิสระหรือทำงานร่วมกัน ได้แก่ (Wall. 2001)

- Client-side filtering เป็นเอเจนท์ที่ฝั่งผู้รับบริการตั้งกฎเกณฑ์การกรองขึ้น และกระทำการกรองเมื่อผู้รับบริการต้องการเข้าถึงส่วนของข้อความที่เฉพาะเจาะจง แนวทางหนึ่งที่จะอธิบายรูปแบบการกรองวิธีนี้คือ มันเป็นการทำงานแบบซิงโครนัส (เกิดขึ้นในเวลาเดียวกัน) กับจุดที่ผู้ใช้เข้าถึงจดหมาย
- Server-side filtering เป็นการกรองที่ถูกเก็บและปฏิบัติงานโดยระบบผู้ให้บริการไปรษณีย์ โดยทั่วไปจะปฏิบัติงานเมื่อเริ่มส่งจดหมายโดยผู้ให้บริการ SMTP ซึ่งเป็นการทำงานแบบอะซิงโครนัส
- Filtering by Proxy เป็นการกรองซึ่งปฏิบัติงานเพื่อประโยชน์ของผู้ใช้หรือเอเจนท์ในการจัดการจดหมายของผู้ใช้ (mail user agent) เป็นบุคคลภายนอกที่ทำงานอิสระ หรือไม่ถูกร้องขอในช่วงการเริ่มต้นการส่งและการเข้าถึง เราใช้วิธีนี้อธิบายการแบ่งประเภทโดยทั่ว ๆ ไปของการจัดการระบบ หรือโปรแกรมเอเจนท์ที่ใช้กรองอีเมล ซึ่งอาจไม่มีความจำเป็น

สำหรับผู้ใช้โดยตรง และอาจจะเกิดขึ้นระหว่างกระบวนการการขนย้าย เปรียบเป็นชั้นที่เพิ่มขึ้นมาหรือเป็นทางผ่าน

รูปที่ 2.1 จำลองการกรองที่เกิดขึ้นในการส่งอีเมลทั้งสามแบบตามที่กล่าวมา



รูปที่ 2.1 การกรองอีเมลบนอินเทอร์เน็ต

2.5 แนวคิดในการกรองข้อมูล

วิธีการในการกรองข้อมูล สามารถจำแนกได้เป็น 3 แนวทาง ดังนี้ (Payne. 1994)

- การกรองจากสิ่งที่เข้าใจหรือรับรู้ (Cognitive filtering) กรองจากลักษณะเฉพาะของข้อความ ได้แก่ เนื้อหา และความหมายของข้อความ จะมีส่วนที่ทำหน้าที่ค้นหาคำสำคัญ (keyword) หรือส่วนหนึ่งของข้อความ (วลี) ที่จะนำมาแบ่งแยกประเภทของข้อความได้อย่างชัดเจน

- Social filtering วิธีการนี้เป็นวิธีการที่สนับสนุนวิธี Cognitive filtering โดยรวมความสัมพันธ์ซึ่งกันและกัน ทั้งแบบส่วนตัว และจากองค์กรของผู้รับและผู้ส่ง ตัวอย่างเช่น การให้ความสนใจข้อความที่ได้รับจากผู้ที่อยู่ในตำแหน่งที่สูงกว่า เช่น เจ้านาย
- Economic filtering การกรองด้วยวิธีการนี้ขึ้นอยู่กับการประเมินผลกำไร ขาดทุนในการจัดการข้อความ เช่น ความยาวของข้อความ

ยังมีแนวทางอื่น ๆ อีกในการกรองข้อมูล เช่น ข้อความที่เข้ามาต้องตรงกับชนิดของเทมเพลต การทดสอบความเป็นไปได้ของฐานความรู้ของการแบ่งประเภท ซึ่งผู้ใช้จะใช้เพื่อตัดสินใจว่าจะเปิดอ่านข้อความ หรือไม่ และให้ผู้ใช้กำหนดกฎและชนิดของกฎที่จะใช้ แนวทางของปัญญาประดิษฐ์ เช่น ภาษาธรรมชาติ

2.6 ตัวอย่างโปรแกรมประยุกต์กรองอีเมล

แม้ว่าการกรองจะมีอยู่ในหลายโปรแกรมจัดการอีเมล แต่ปัญหาของจดหมายขยะก็ยังคงมีอยู่ ซึ่งยังคงเป็นคำถามในการพัฒนาว่า เครื่องมือในการกรองควรมีขอบเขตในการกรองได้ขนาดไหน และควรกรองได้ถึงจุดไหน ซึ่งจะพิจารณาการกรองในโปรแกรมต่อไปนี้

2.6.1 **Pegasus Mail for Win32** (Chew. 2002)

โปรแกรมจะนำเสนอเซตในการกรองอีเมล ซึ่งสามารถแยกอีเมลที่เข้ามาใส่ในโฟลเดอร์ ทำให้เกิดการกรองโดยอัติโนมัติ ตั้งแต่เริ่มติดตั้งโปรแกรม และกำหนดรูปแบบที่จะกรองใส่ในแต่ละโฟลเดอร์ โดยผู้ใช้เป็นผู้กำหนดรูปแบบในการกรอง และการกระทำที่ต้องการให้เกิดขึ้นกับอีเมล ที่ตรงตามกฎเมื่อเข้ามาในโปรแกรม อาจเป็นการให้นำไปใส่ในโฟลเดอร์ หรือลบทิ้ง ซึ่งรูปแบบการกรองมี ดังนี้

(1) Standard Header Match

โดยผู้ใช้ต้องกำหนดส่วนหัวของอีเมลที่ต้องการกรอง เช่น Sender, Subject, To, Cc, From และ Reply-To เพื่อให้โปรแกรมใช้ในการค้นหาอีเมล ที่มีส่วนหัวสอดคล้องกับที่ผู้ใช้ต้องการ แล้วกระทำตามที่ผู้ใช้กำหนดไว้ก่อนหน้านี้แล้ว

(2) Regular Expression Match

โดยผู้ใช้กำหนด Regular Expression เพื่อให้โปรแกรมใช้ในการตรวจสอบรายละเอียดในเนื้อหา ซึ่งการทำงานคล้ายกับวิธีข้างต้น นั่นคือ อีเมลที่สอดคล้องกับที่กำหนดก็จะถูกกรองออกมา

(3) Message Date Age

โดยผู้ใช้กำหนดวัน หรือช่วงเวลาที่ต้องการ ซึ่งอีเมล จะถูกกรองเอาไว้ ถ้าอยู่ในช่วงเวลาที่กำหนด

(4) Distribution List

โดยผู้ใช้กำหนดอีเมลแอดเดรสใน Distribution List ทำให้อีเมลที่สร้างจาก List นี้ ถูกกรองเก็บไว้

2.6.2 Microsoft Outlook 2000

รูปแบบการทำงานของ Microsoft Outlook 2000ไม่แตกต่างจาก Peagasus Mail มากนัก เพราะถึงแม้จะใช้ชื่อวิธีในการกรองต่างกัน แต่รูปแบบการทำงานภายในคล้ายกัน Microsoft Outlook 2000 ให้สิทธิ์ในการกำหนดเซตของการกรองอีเมล ที่เข้ามาใส่ในโฟลเดอร์ ทำให้เกิดการกรองโดยอัตโนมัติ ตั้งแต่เริ่มติดตั้งโปรแกรม และกำหนดรูปแบบที่จะกรองใส่ในแต่ละโฟลเดอร์ การกระทำที่เกิดขึ้นอาจเป็นการให้นำไปใส่ในโฟลเดอร์ หรือลบทิ้ง ซึ่งรูปแบบการกรองมี ดังนี้ (Chew. 2002)

(1) Time Filter

โดยผู้ใช้กำหนดเวลา หรือช่วงเวลาให้แก่เครื่องมือกรองอีเมล ทำให้อีเมลที่มีเวลา หรืออยู่ในช่วงเวลาที่กำหนดถูกกรองเก็บไว้

(2) Standard Header Match

การประยุกต์ใช้รูปแบบนี้ ผู้ใช้ต้องกำหนดส่วนหัวของอีเมล เช่น Sender, Subject, To, Cc, From และ Reply-To เป็นต้น ที่ต้องการกรองเก็บไว้ ทำให้อีเมลที่มีบางส่วนของส่วนหัวตรงกับที่กำหนดถูกกรองเก็บไว้

(3) Phrase Match Filter

โดยผู้ใช้กำหนดวลีธรรมดา ที่ต้องการใช้ เก็บไว้ในแฟ้ม ทำให้อีเมลแต่ละอันถูกนำไปตรวจสอบว่ามีส่วนของวลีที่กำหนดไว้หรือไม่ ถ้ามีก็จะถูกกรองเก็บไว้

(4) Unwanted E-mail Address

ผู้ใช้สามารถเพิ่ม หรือลดอีเมลแอดเดรสจากรายการของผู้ส่งอีเมลที่ไม่ต้องการรับ ทำให้อีเมลที่มีอีเมลแอดเดรสตรงกับที่กำหนด ก็จะถูกกรองทิ้งไป

2.6.3 Microsoft Hotmail

"*Keep unwanted e-mail from reaching your Inbox*" Hotmail ประกาศประโยคนี้ออกมาเมื่อผู้ลงทะเบียนเลือกไปที่ Junk Mail Filter อย่างไรก็ตาม ไม่สามารถยืนยันได้ว่า

สามารถจัดการ เมื่อจำนวนและความหลากหลายของจดหมายขยะเพิ่มมากขึ้นได้หรือไม่ เมื่อใช้งานไปเรื่อย ๆ โปรแกรมจะคอยเตือนให้ผู้ใช้พิจารณาโฟลเดอร์ของจดหมายขยะ เพื่อให้มั่นใจได้ว่าไม่มีจดหมายที่ต้องการเก็บไว้อยู่ในโฟลเดอร์นี้ การกระทำเช่นนี้ ทำให้ไม่ต้องกำหนดจุดประสงค์ในการกรองตั้งแต่แรก จากวิธีการนี้ ทำให้ผู้ใช้ไม่ต้องสอดส่องดูแลข้อมูลขนาดใหญ่ในส่วนของจดหมายขยะ และผู้ใช้สามารถมั่นใจได้ว่า เครื่องมือสามารถแยกประเภทได้ตามที่กำหนด เครื่องมือกรองอีเมลที่นำเสนอโดย Hotmail มีดังนี้ (Chew. 2002)

(1) Address Book Match

อีเมลที่มีแอสเดรสตรงกับ Address Book ของผู้ใช้ หรือ Safe List จะถูกนำไปไว้ในกล่องไปรษณีย์ของผู้ใช้ ส่วนอีเมลอื่น ๆ ก็จะถูกนำไปไว้ในโฟลเดอร์ของจดหมายขยะ วิธีการกรองนี้มีประโยชน์มาก ในกรณีที่อีเมลแอสเดรสถูกกำหนดไว้อย่างถาวร และเฉพาะเจาะจง อย่างไรก็ตาม ถ้ามีอีเมลที่สำคัญ ซึ่งส่งมาจากแอสเดรสที่ไม่รู้จัก อีเมลนี้ก็จะถูกกรองทิ้ง โดยที่ผู้ใช้ไม่ทราบ

(2) Block Sender

อีเมลที่ถูกส่งมาจากอีเมลแอสเดรส (หรือโดเมน) ที่อยู่ในรายการ block sender ของผู้ใช้ จะถูกลบทิ้ง ผู้ใช้จะไม่มีทางได้รับอีเมลเหล่านี้เลย และจะไม่ปรากฏใน Junk Mail folder และ Trash folder

(3) Standard Header Match Filters

Hotmail เสนอวิธีในการกรองโดยให้ผู้ใช้เป็นผู้กำหนดเองได้หลายวิธี เพื่อแยกประเภทเนื้อหาเมื่อได้รับอีเมล ซึ่งจะคล้ายกับวิธีการของ Header Match ที่ได้กล่าวถึงไปแล้วใน หัวข้อ 2.6.2 Microsoft Outlook 2000

2.7 โพรโทคอลที่ใช้กับการรับ-ส่งอีเมล

โพรโทคอลอีเมลที่เกี่ยวข้องกับการรับอีเมล ได้แก่ Post Office Protocol version 3 (POP3) และ Internet Message Access Protocol version 4 (IMAP4) ส่วนโพรโทคอลที่ใช้ในการส่งอีเมล ได้แก่ Simple Mail Transfer Protocol (SMTP)

2.7.1 **POP3** (กอบเกียรติ. 2545ข)

POP3 มีจุดประสงค์เพื่อให้เครื่องคอมพิวเตอร์ที่เชื่อมต่อกัน สามารถเข้าถึงและรับอีเมล จากเครื่องผู้ให้บริการอีเมล แต่ POP3 ไม่ได้จัดเตรียมหน้าที่ในการจัดการที่คลอบคลุมของอีเมล บน

เครื่องผู้ให้บริการ ทำให้ความสามารถของ POP3 มีจำกัด การใช้งานโดยปกติ คือ อีเมลจะถูกส่งออกและลบจากเครื่องผู้ใช้บริการ

หน้าที่ของ POP3 เริ่มต้นด้วย เครื่องผู้ให้บริการอีเมล ฟังการร้องขอใช้บริการผ่านทางพอร์ต TCP และเมื่อเครื่องผู้รับบริการที่ต้องการใช้บริการ ก็จะสร้างการเชื่อมต่อด้วย TCP ไปยังเครื่องผู้ให้บริการ เมื่อการเชื่อมต่อถูกสร้างขึ้นแล้ว เครื่องผู้ให้บริการก็จะส่งข้อความต้อนรับ ทำให้เครื่องผู้ให้และรับบริการ แลกเปลี่ยนคำสั่งและตอบสนองกัน จนกระทั่งการเชื่อมต่อถูกปิดลงหรือถูกยกเลิก

POP3 Server ได้แบ่งสภาวะของการเชื่อมต่อออกเป็น 3 สภาวะ ดังนี้

1. Authorization State เป็นสภาวะเริ่มต้นตั้งแต่เราเข้าไปติดต่อเซิร์ฟเวอร์ ส่งชื่อผู้ใช้และรหัสผ่าน คล้ายกับการแสดงตน และบัตรประจำตัวว่าเป็นเจ้าของตู้ไปรษณีย์
2. Transaction State หลังจากที่ตรวจรหัสผ่านถูกต้องแล้ว ก็เข้าสู่ช่วงการทำงานโดยเซิร์ฟเวอร์จะรับคำสั่งจากเรา เสร็จแล้วก็แสดงข้อมูลออกมา เช่น STAT ในการดูจำนวนอีเมล RETR ในการเปิดอ่านอีเมล หรือ DETE ลบอีเมล เป็นต้น
3. Update State หลังจากที่ส่งคำสั่ง QUIT แล้วจะเป็นช่วงอัพเดทกล่องไปรษณีย์ เช่น ก่อนหน้านี้เราได้สั่งลบอีเมล ซึ่งเซิร์ฟเวอร์ยังไม่ได้ลบทันที แต่จะทำเครื่องหมายไว้ก่อนว่าอีเมล ฉบับใดที่ต้องการลบ พอหลังจากที่ QUIT แล้ว เซิร์ฟเวอร์ก็จะลบอีเมลที่ทำเครื่องหมายไว้ เมื่อ QUIT ขณะอยู่ใน Transaction State ก็เป็นการเข้าสู่ Update State แต่หากส่งคำสั่ง QUIT ในช่วง Authorization State ก็ไม่ถือว่าเข้าสู่ Update State

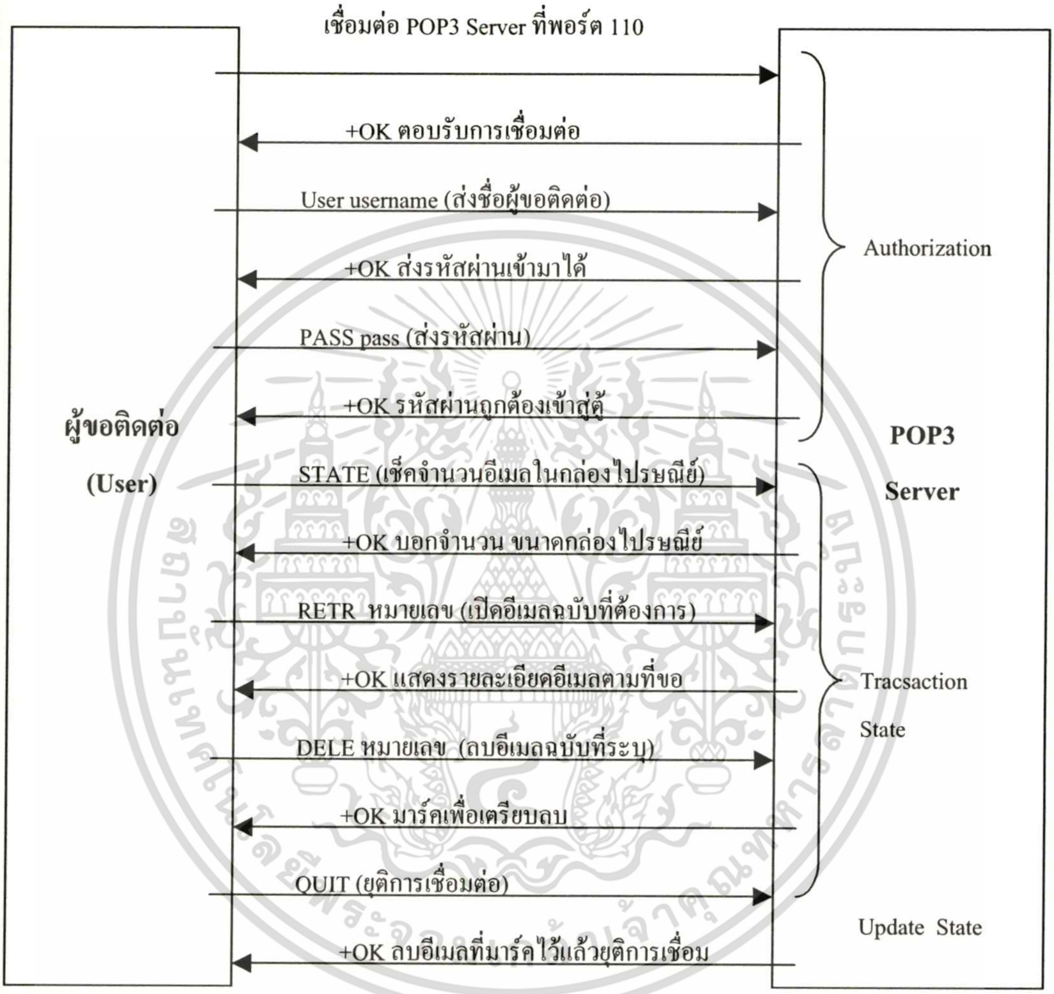
สำหรับการสื่อสารคุยกันระหว่างผู้มาขอติดต่อและเครื่องผู้ให้บริการ POP3 แสดงได้ดังรูปที่ 2.2

2.7.2 SMTP (กอบเกียรติ. 2545ก)

จุดประสงค์ของ SMTP คือ การส่งเมล์อย่างน่าเชื่อถือ และมีประสิทธิภาพ SMTP เป็นระบบที่ต้องการการจัดส่งเมล์อย่างอิสระ และช่องทางการจัดส่งที่เชื่อถือได้ ลักษณะที่สำคัญของ SMTP คือ มีสมรรถนะในการส่งต่อเมล์ในสภาพแวดล้อมชั้น transport service โดยที่ transport service จัดเตรียมสภาพแวดล้อมการสื่อสารระหว่างกระบวนการ (inter process communication environment-IPCE)ไว้ให้ ซึ่ง IPCE อาจจะคลอบคลุม ตั้งแต่หนึ่งหรือหลาย ๆ เครือข่าย ซึ่งเมล์สามารถสื่อสารระหว่างกระบวนการด้วย IPCE ที่แตกต่างกันได้ โดยการส่งต่อผ่านกระบวนการที่เชื่อมต่อกันด้วย IPCE ตั้งแต่ 2 ตัวขึ้นไป และเมล์ยังสามารถส่งต่อระหว่างเครื่องผู้ให้บริการที่มี transport system ที่แตกต่างกัน

SMTP จัดเตรียมกลไก สำหรับส่งอีเมล โดยตรงจากเครื่องผู้ให้บริการของผู้ส่งไปยังเครื่องผู้ให้บริการของผู้รับ เมื่อเครื่องผู้ให้บริการทั้งคู่เชื่อมต่ออยู่กับ transport service เดียวกัน หรือส่งผ่าน

SMTP service ตั้งแต่ 1 ตัวขึ้นไป ในกรณีที่เครื่องผู้ให้บริการต้นทางและปลายทาง ไม่ได้เชื่อมต่อด้วย transport service เดียวกัน



รูปที่ 2.2 ลำดับการสื่อสารคำสั่งและรหัสตอบรับ POP3

หน้าที่ของ SMTP อยู่บนพื้นฐานการออกแบบรูปแบบการสื่อสาร โดย SMTP ตัวส่งสร้างช่องทางการสื่อสารแบบสองทางกับ SMTP ตัวรับ SMTP ตัวรับ อาจเป็นได้ทั้งผู้รับคนสุดท้าย หรือผู้รับคนกลางก็ได้

การเริ่มต้น การสร้างช่องทางการติดต่อสื่อสาร SMTP ตัวส่งจะส่งคำสั่ง MAIL เพื่อบ่งบอกให้รู้ว่าเป็นคนส่งเมล์ ถ้า SMTP ตัวรับ สามารถยอมรับเมล์นั้นได้ มันก็จะส่ง คำตอบ OK กลับมา SMTP ตัวส่งก็จะส่งคำสั่ง RCPT เพื่อระบุผู้รับเมล์ปลายทาง ถ้า SMTP ตัวรับสามารถยอมรับเมล์

นั้นได้ มันก็จะส่ง OK กลับมา แต่ถ้ามันไม่สามารถรับได้ มันก็จะตอบปฏิเสธกลับมา SMTP ตัวส่งและตัวรับอาจจะจัดการกับผู้รับหลาย ๆ ราย เมื่อผู้รับได้ถูกตรวจสอบ และได้รับการพิสูจน์ว่าเป็นตัวจริง และ SMTP ตัวรับจะส่งเนื้อหาอีเมลจนจบ เมื่อ SMTP ตัวรับได้รับอีเมลที่สมบูรณ์ รับได้สำเร็จ มันจะตอบ OK กลับคืนไปให้

สำหรับการสื่อสารคุยกันระหว่างผู้มาขอติดต่อและเครื่องผู้ให้บริการ SMTP แสดงได้ดังรูปที่ 2.3

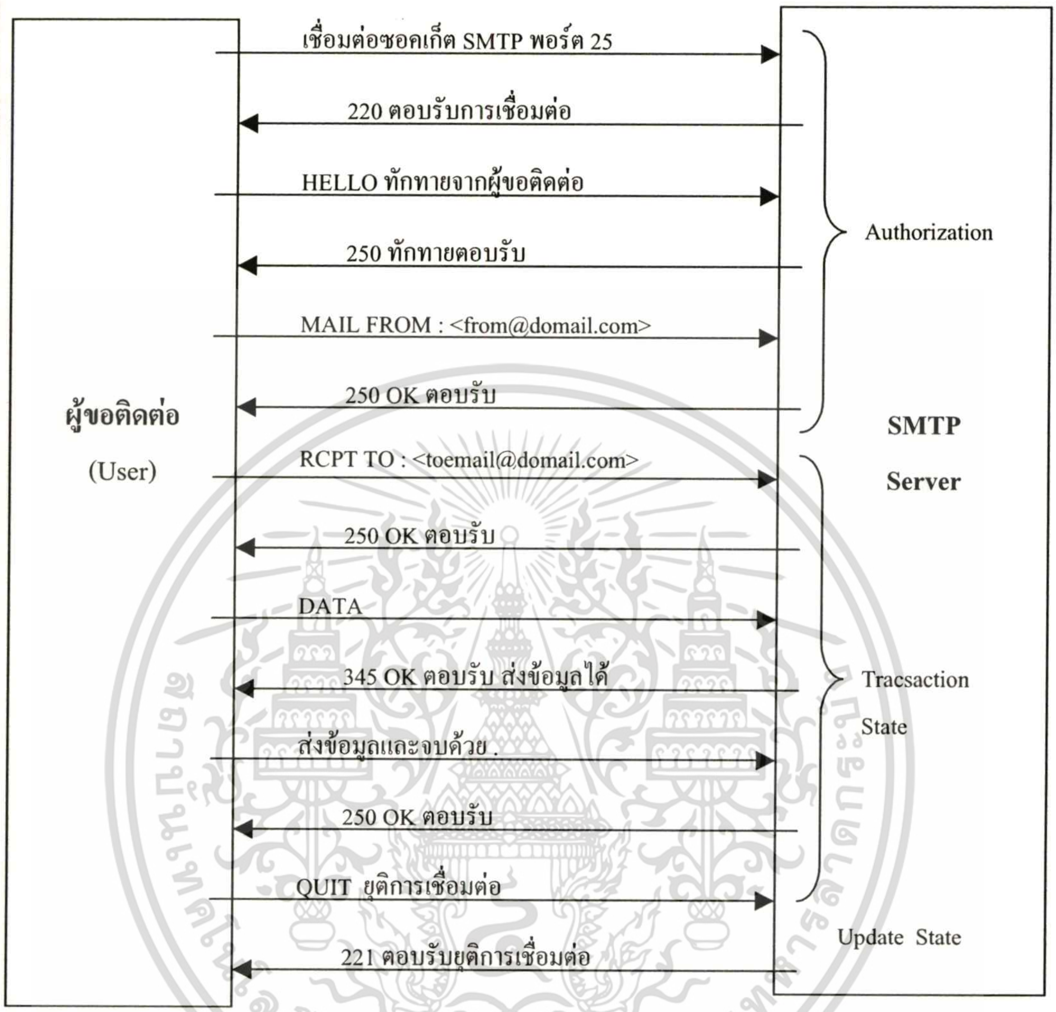
2.7.3 IMAP (กอบเกียรติ. 2545ข)

IMAP ให้สิทธิ์แก่เครื่องผู้ใช้บริการเข้าถึง และจัดการอีเมลบนเครื่องผู้ให้บริการได้ และอนุญาตให้จัดการกับกล่องไปรษณีย์ได้ ซึ่งมีรูปแบบการทำงานที่มีความสามารถเสมือนการทำงานที่เครื่องของตน อีกทั้ง IMAP4rev1 ยังจัดเตรียมสมรรถนะสำหรับให้เครื่องลูกสามารถจัดการกับอีเมลได้โดยไม่ต้องดาวน์โหลดเนื้อหามาที่เครื่องลูกทั้งหมด หรืออีกประการคือมีการแยกแยะได้ว่า เนื้อหาอีเมลฉบับใดเปิดอ่านแล้ว ฉบับใดยังไม่ได้เปิดอ่าน

IMAP4rev1 จัดเตรียมการทำงาน ดังนี้

1. สร้าง ลบ ตั้งชื่อใหม่ให้กับกล่องไปรษณีย์
2. ตรวจสอบว่ามีอีเมลใหม่หรือไม่
3. ลบอีเมลออกอย่างถาวร
4. ตั้งและล้างค่าสถานะต่าง ๆ
5. ค้นหาอีเมลที่ต้องการ
6. การเลือกคุณลักษณะของเนื้อหา หรือข้อความที่สนใจได้

แต่ IMAP4rev1 ไม่มีคำสั่งสำหรับส่งอีเมล เพราะหน้าที่นี้เป็นของ SMTP คอยจัดการ โพรโทคอล IMAP4rev1 จะตั้งสมมติฐานว่า TCP จัดเตรียมช่องทางการติดต่อสื่อสารที่เชื่อถือได้ไว้ให้แล้ว เครื่องผู้ให้บริการ IMAP4rev1 จะดักฟังการร้องขอใช้บริการที่พอร์ต 143 การเชื่อมต่อด้วย IMAP4rev1 ประกอบด้วย การสร้างการเชื่อมต่อเครือข่ายของผู้ให้และรับบริการ เริ่มต้นด้วยการตอบต้อนรับ จากเครื่องผู้ให้บริการ จากนั้นเครื่องผู้ให้และรับบริการก็จะสื่อสารกัน ซึ่งการสื่อสารของผู้รับและผู้ใช้บริการ ประกอบด้วยคำสั่งจากผู้รับบริการ ข้อมูล และการตอบสนองผลลัพธ์ที่สมบูรณ์จากผู้ให้บริการ



รูปที่2.3 ลำดับการสื่อสารคำสั่งและรหัสตอบรับ SMTP

2.8 ซ็อกเก็ต (ยุทธนา. 2544)

ซ็อกเก็ต (Socket) ถูกกำหนด หรือนิยามไว้ว่า เป็นคู่ของการสื่อสาร หรือคู่ของโพรเซส (หรือเธรด) โดยที่การสื่อสารบนเครือข่ายใช้คู่ของซ็อกเก็ตสำหรับแต่ละโพรเซส

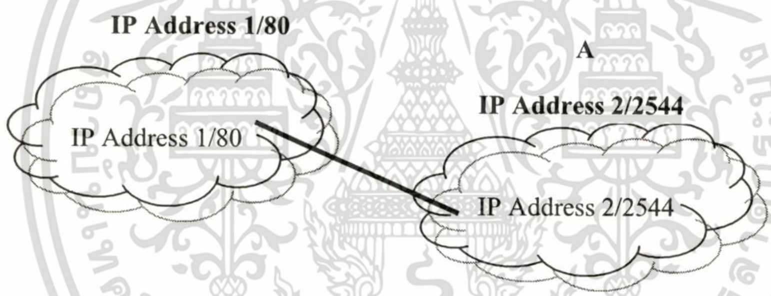
สำหรับซ็อกเก็ต ประกอบไปด้วย IP Address กับหมายเลขพอร์ต

โดยทั่ว ๆ ไป ซ็อกเก็ตใช้สถาปัตยกรรมไคลเอนท์เซิร์ฟเวอร์ เซิร์ฟเวอร์จะรอการเข้ามาตามการขอร้องของไคลเอนท์โดยการฟังที่พอร์ตเฉพาะ เมื่อการร้องขอได้รับ เซิร์ฟเวอร์ก็จะยอมรับการเชื่อมต่อจากซ็อกเก็ตของไคลเอนท์ เพื่อให้การเชื่อมต่อสมบูรณ์

เซิร์ฟเวอร์ที่สร้างการบริการเฉพาะ เช่น telnet, ftp, mail และ http จะฟังที่พอร์ตมีชื่อ เช่น เซิร์ฟเวอร์ telnet จะฟังที่พอร์ต 23 เซิร์ฟเวอร์ ftp จะฟังที่พอร์ต 21 หรือ เซิร์ฟเวอร์ http จะฟังที่พอร์ต 80 เป็นต้น

หมายเลขพอร์ตทั้งหมดที่ต่ำกว่า 1,024 จะถูกพิจารณาว่าเป็นพอร์ตที่มีชื่อเสียง เราสามารถใช้พอร์ตเหล่านี้เพื่อสร้างการบริการตามมาตรฐานได้

ตัวอย่างการสื่อสารด้วยซ็อกเก็ต เมื่อเธรดไคลเอนท์เริ่มต้นการขอร้องสำหรับการเชื่อมต่อ จะถูกกำหนดพอร์ตโดยโฮสท์คอมพิวเตอร์ พอร์ตนี้เป็นหมายเลขใด ๆ ก็ได้ที่มากกว่า 1,024 ตัวอย่างเช่น ถ้าไคลเอนท์บนโฮสท์ A มี IP Address 2 ต้องการที่จะสร้างการเชื่อมต่อกับเซิร์ฟเวอร์ http (ซึ่งฟังที่พอร์ต 80) ที่มี IP Address 1 โฮสท์ A จะถูกกำหนดพอร์ต 2544 และที่ฝั่งเซิร์ฟเวอร์จะเป็นพอร์ต 80 สถานการณ์นั้นสามารถแสดงได้ดังรูปที่ 2.4 แพ็คเก็ตไปมาระหว่างโฮสต์ทั้งสองจะถูกส่งไปยังเธรดที่เหมาะสม ซึ่งขึ้นอยู่กับหมายเลขพอร์ตปลายทางด้วย



รูปที่ 2.4 การสื่อสารโดยใช้ซ็อกเก็ต

การเชื่อมต่อทั้งหมดเป็นคุณสมบัติเฉพาะ ดังนั้นถ้าโพรเซสอื่น ๆ บนโฮสต์ A ต้องการสร้างการเชื่อมต่ออื่น ๆ กับเซิร์ฟเวอร์ http เดียวกัน เซิร์ฟเวอร์จะกำหนดหมายเลขพอร์ตที่มากกว่า 1024 และต้องไม่เท่ากับพอร์ต 2544 (เนื่องจากถูกใช้ไปแล้ว) การทำอย่างนี้ เพื่อให้แน่ใจว่าการเชื่อมต่อทั้งหมดประกอบด้วยคู่ที่ไม่ซ้ำกับใคร ของซ็อกเก็ต หรือเป็นสิ่งที่ไม่ซ้ำกับการเชื่อมต่ออื่น ๆ ของซ็อกเก็ต

โดยปกติแล้ว เซิร์ฟเวอร์จะมีหลาย ๆ การร้องขอเกิดขึ้นพร้อมกัน ดังนั้น จะต้องใช้ระยะเวลาหนึ่งที่ไคลเอนท์ต้องรอคอย เพื่อที่จะได้รับบริการโดยเซิร์ฟเวอร์เธรดเดียว ซึ่งการทำงานเช่นนี้ไม่สามารถรับได้ เพื่อแก้ไขสถานการณ์นี้เซิร์ฟเวอร์ต้องจัดการการร้องขอที่พร้อม ๆ กัน โดยการกำหนด เธรดแยกออกมาเพื่อบริการแต่ละการขอร้องที่เข้ามา ตัวอย่างเช่น เซิร์ฟเวอร์ http ที่ไม่ว่างจะกำหนดเธรดออกมาเพื่อบริการแต่ละการขอร้องสำหรับเว็บเพจ

ซ็อกเก็ต มีทั้งหมดสามชนิด คือ Connection-Oriented Socket , Connectionless Socket และ Raw Socket (กอบเกียรติ. 2545ก)

- Connection-Oriented Socket เป็นซ็อกเก็ตการเชื่อมต่อแบบต่อเนื่องที่อนุญาตให้โพรเซสเชื่อมต่อกับโพรเซสระยะไกล (Remote) โดยใช้โพรโทคอล TCP ดังนั้นด้วยวิธีการนี้ทำให้ข้อมูลมีความเชื่อถือได้ เมื่อการเชื่อมต่อได้เกิดขึ้น โพรเซสก็จะมีการส่งข้อมูลกลับไปจนกระทั่งฝั่งใดฝั่งหนึ่งหรือสิ่งอื่นปิดการเชื่อมต่อ ซ็อกเก็ตชนิดนี้ บางครั้งเรียกว่า สตรีมซ็อคเก็ต (Stream Socket) ทั้ง ftp และ http ใช้ซ็อกเก็ตแบบนี้ในการสื่อสาร
- Connectionless Socket หรือเรียกอีกอย่างว่า ดาต้าแกรมเป็นซ็อกเก็ตแบบไม่ต่อเนื่อง และนำมาใช้เป็นประโยชน์ในการส่งข้อความสั้น ๆ ซึ่งไม่สามารสนับสนุนส่วนหัว ดังนั้น จึงพิจารณาการเชื่อมต่อประเภทนี้เป็นแบบเชื่อถือไม่ได้ ซึ่งก็คือ การไม่รับประกันข้อมูลที่ถูกส่งออกไป ไม่เหมือนกับซ็อกเก็ตการเชื่อมต่อแบบต่อเนื่องที่ซ็อกเก็ตปลายทางถูกตรวจสอบเมื่อแพ็กเก็ตถูกส่งออกไป

  ซ็อกเก็ตแบบไม่ต่อเนื่อง เปรียบเสมือนกับการบริการของไปรษณีย์ที่ผู้ส่งจดหมายไปตามที่อยู่แล้วใส่ในกล่องรับจดหมาย ผู้ส่งจะไม่ทราบว่าผู้รับได้รับจดหมายหรือไม่ ซ็อกเก็ตแบบนี้นิยมใช้กันในเซิร์ฟเวอร์ DNS (Domain Name System) ที่ใช้ซ็อกเก็ตดาต้าแกรมในการตอบสนองต่อการขอร้องที่เข้ามามาก ๆ

  นอกจากนี้จะใช้ดาต้าแกรมซ็อกเก็ตในการกระจาย (Broadcast) ข้อความ หรือ Multicast เพื่อไปยังปลายทางหลาย ๆ แห่งพร้อมกัน ซึ่งเหมือนกับการกระจายเสียงวิทยุ หรือโทรทัศน์
- Raw Socket เป็นซ็อกเก็ต ที่อนุญาตให้การเข้าถึงโพรโทคอล Transport Raw Socket ยังสามารถนำมาใช้เพื่อจัดการข้อมูลส่วนหัว IP (IP Header) นอกจากนี้แล้วการใช้ซ็อกเก็ตชนิดนี้ ต้องการความรู้อย่างมากของโครงสร้างโพรโทคอลพื้นฐาน

ประโยชน์และข้อดีของการเชื่อมต่อระดับซ็อกเก็ต

1. สามารถตรวจเช็คอีเมลแอสเดรสว่ามีอยู่จริงหรือไม่
2. ตรวจเช็คเฉพาะโดเมนของอีเมลว่ามีอยู่จริงหรือไม่
3. อัพโหลดไฟล์เข้าเซิร์ฟเวอร์โดยไม่ต้องใช้ฟอร์มเลือกไฟล์
4. เช็คอีเมล และเปิดอ่านอีเมลแบบ POP3 จากเซิร์ฟเวอร์ใด ๆ ก็ได้

2.9 การค้นคืนสารสนเทศ

รูปแบบโดยทั่วไปของการค้นคืนสารสนเทศ พิจารณาจากเซตของคำสำคัญในเอกสารที่เรียกว่า "คำดรรชนี" (index term) เป็นลักษณะของคำง่าย ๆ ชัดเจน ตรงไปตรงมา ซึ่งช่วยในเรื่องความหมายของคำ เพื่อให้จดจำใจความหลักของเอกสารนั้นได้ ดังนั้น คำดรรชนีจึงถูกนำมาใช้เป็นตัวชี้ และตัวสรุปเนื้อหาของเอกสารนั้น

เซตของคำดรรชนีที่กำหนดให้กับเอกสาร เราจะสังเกตว่าทุก term ที่ถูกใช้ประโยชน์ในการอธิบายเนื้อหาของเอกสาร ไม่ได้มีค่าเท่ากันเสมอไป การพิจารณาปัญหาในเรื่องความสำคัญของ term เพื่อใช้สรุปเนื้อหาของเอกสารไม่ใช่เรื่องยากจนเกินไป เราสามารถแก้ปัญหาได้ โดยใช้ค่าน้ำหนัก กำหนดให้แก่คำดรรชนีแต่ละตัวในเอกสาร ดังนั้น ค่าน้ำหนักจึงเป็นสิ่งจำเป็นสำหรับคำดรรชนี เพื่อใช้อธิบายเนื้อหาของเอกสารนั้น

สำหรับการประเมินผล มีคำศัพท์อยู่สองคำที่เกี่ยวข้องในเรื่องนี้ คำแรก คือ Recall หมายถึง จากเอกสารทั้งหมดที่สัมพันธ์กันที่เก็บรวบรวมไว้ มีจำนวนที่สัมพันธ์กันถูกค้นคืนได้เท่าไร คำที่สอง คือ Precision หมายถึง เอกสารทั้งหมดที่ถูกรวบรวมในการค้นคืน มีเอกสารที่สัมพันธ์กันอยู่เท่าไร (Chew. 2002)

จากความหมายข้างต้น เป็นที่มาของการประมวลผลการค้นคืนสารสนเทศ 3 แบบ คือ Boolean Model, Probability Retrieval Model และ Vector Space Model (Chew. 2002)

2.9.1 Boolean Model

Boolean Model เป็นรูปแบบการค้นคืนแบบง่าย ๆ ใช้พื้นฐานของทฤษฎีเซต และพีชคณิตรูปแบบบูล (Boolean algebra) ซึ่งคำสั่งที่ใช้ปฏิบัติงานของ Boolean logic ได้แก่ logical sum (+) , logical product (x) และ logical difference (-)

Logical product หรือ AND logic ยินยอมให้ตัวค้นหากำหนดเจาะจงความสอดคล้องกันหรือการเกิดขึ้นพร้อมกันของเงื่อนไขตั้งแต่ 2 ข้อขึ้นไป logical sum หรือ OR logic ยินยอมให้ตัวค้นหามีทางเลือกจาก term ที่กำหนดให้ และ logical difference หรือ NOT logic จัดเตรียมเครื่องมือสำหรับแยกรายการออกไปจากเซต คำสั่งที่กล่าวมาดูเหมือนง่าย และไม่ค่อยมีความสำคัญ อย่างไรก็ตาม เมื่อมันถูกนำมาใช้ร่วมกัน คำสั่งที่ได้ก็จะมีความซับซ้อนมาก

Boolean Model มีอุปสรรคหลัก ๆ คือ กลไกการค้นคืนใช้หลักการตัดสินใจแบบไบนารีว่าสัมพันธ์หรือไม่สัมพันธ์กัน โดยปราศจากการวัดระดับความเชื่อมั่น ดังนั้น จึงไม่สามารถจัดการกับเซตที่คล้ายคลึงกัน ที่ขึ้นกับความถี่ได้ และผู้ใช้มีอิทธิพลมากในการสร้างสูตรในการค้นคืน โดยการรวมคำสั่ง AND , OR และ NOT ซึ่งในหลาย ๆ ครั้งที่ผู้ใช้มักสร้างสูตรที่แคบ หรือกว้างจนเกินไป

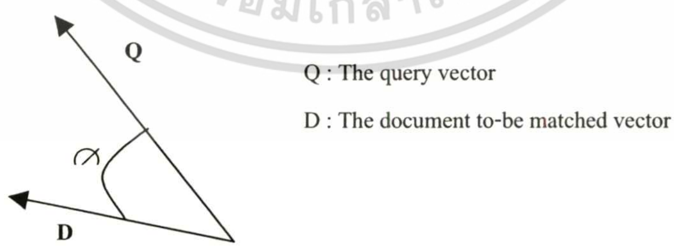
2.9.2 Probability Retrieval Model

Probability Retrieval Model พยายามที่จะแก้ปัญหาการค้นคืนสารสนเทศด้วยวิธีข้างต้น โดยใช้ เครือข่ายที่เกี่ยวข้องกับความน่าจะเป็น หลักการพื้นฐานของการเข้าคู่กันตามทฤษฎีความน่าจะเป็นของเอกสารที่กำหนดให้ กับที่ต้องการค้นหา มันต้องสามารถคำนวณความน่าจะเป็นที่เอกสารที่กำหนดมีความสัมพันธ์กับเอกสารที่ต้องการค้นหาออกมาได้

เมื่อโมเดลได้รับเอกสารที่ผู้ใช้ต้องการค้นหา นั่นคือ เซตของเอกสาร จะต้องมีเซตของเอกสารที่สัมพันธ์ หรือไม่มีเอกสารที่สัมพันธ์กัน เซตของผู้ใช้ในที่นี้ หมายถึง เซตในอุดมคติ Probability Retrieval Model ต้องพยายามหาผลลัพธ์ที่เข้าคู่กับเซตในอุดมคติมากที่สุด อย่างไรก็ตาม ในตอนเริ่มต้น Probability Retrieval Model อาจจะยังไม่มีแนวคิดว่า คุณสมบัติของเซตในอุดมคติของเอกสารควรเป็นอย่างไร ดังนั้น ในครั้งแรกต้องมีการติดต่อสื่อสารกับผู้ใช้ เพื่อยกระดับการอธิบายความน่าจะเป็นของเซตคำตอบในอุดมคติ หลังจากการติดต่อสื่อสาร Probability Retrieval Model จะได้รายการของเอกสารที่เรียงตามความน่าจะเป็นของความสัมพันธ์กับเอกสารที่ต้องการค้นหา

2.9.3 Vector Space Model

เมื่อมีการใช้ Vector Space Model การวัดความคล้ายคลึงกันก็จะเกี่ยวข้องกับระยะห่าง ตามทฤษฎีที่ว่า เอกสารที่ใกล้กันมากในพื้นที่ของเวคเตอร์ ก็จะมีความความคลึงกันสูง การจำลองแนวคิดของ Vector Space Model เอกสาร (D) และการสอบถามของผู้ใช้ (Q) สามารถนำเสนอด้วยทิศทางของเวคเตอร์ N ทิศทาง ดังรูปที่ 2.5 ซึ่งจะสังเกตเห็นว่า ค่าของ $\varnothing$ ที่น้อยที่สุด ก็จะเป็นเอกสารที่คล้ายคลึงกับที่ผู้ใช้ต้องการมากที่สุด



รูปที่ 2.5 โคไซน์ของ $\varnothing$ หาได้จากสูตร sim(d,q)

ในการใช้งานโดยทั่ว ๆ ไปของ Vector Space Model นั้น index term แต่ละคำของเอกสารต้องเข้าคู่กัน ซึ่งถูกนำมาเป็นแกนใน N-dimensional space ใช้เป็นทิศทาง N ทิศทางของ index term

ที่มีน้ำหนัก และแต่ละอันต้องทำมุมกัน ซึ่งมันอาจจะมีขนาดใหญ่มากสำหรับ index term ในเอกสารขนาดปานกลาง ทำให้ term space ใน vector space ใช้เวลานาน vector space สามารถสร้าง space vector หนึ่งตัวสำหรับเอกสารแต่ละอันที่เก็บรวบรวมไว้ โดยที่เอกสารถูกนำเสนอโดย index term ที่ถ่วงน้ำหนัก เอกสารสอบถามของผู้ใช้สามารถหาความสอดคล้อง โดยใช้วิธีทางเวคเตอร์ ในการค้นหา ดังนั้น รูปแบบระยะห่างระหว่างเอกสารสอบถามกับเอกสารที่รวบรวม สามารถใช้วิธีเกี่ยวกับเรขาคณิตโดยหาโคไซน์ (cosine) ของมุมระหว่าง vector ของเอกสารสอบถามกับเอกสารที่รวบรวม เพื่อหาความสอดคล้องกันของเอกสารสอบถามกับเอกสารที่รวบรวม ซึ่งความสอดคล้องกันของเอกสารสอบถาม (q) กับเอกสารที่เก็บรวบรวม (d) ใน N-space สามารถแสดงได้ ดังแสดงในรูปที่ 2.6

โดยที่ sim(d,q) มีค่าเริ่มต้นตั้งแต่ 0 ซึ่งหมายถึง มีความสอดคล้องน้อยที่สุด จนกระทั่งถึง 1 ซึ่งหมายถึง มีความสอดคล้องมากที่สุด ความสอดคล้องจะพิจารณาจากมุมที่วัดได้ จากจุดที่กำหนดตายตัว

$$sim(d,q) = \frac{\sum_{i=1}^{N} w_{id} \times w_{iq}}{\sqrt{\sum_{i=1}^{N} (w_{id})^2} \times \sqrt{\sum_{i=1}^{N} (w_{iq})^2}}$$

รูปที่ 2.6 สูตรคำนวณความสอดคล้องของเอกสาร

เรียกว่า จุดกำเนิด ถ้าจุดนี้ถูกเปลี่ยน เช่น กรณีที่ดรรชนีของเอกสารเปลี่ยนแปลง ก็จะมีผลทำให้มุมระหว่างเอกสารที่รวบรวมและเอกสารสอบถามเปลี่ยนแปลงตามไปด้วย โดยที่ในความเป็นจริง มันอาจเป็นเอกสารคนละอันกัน ทำให้จดหมายที่ประกอบด้วยคำ 50 คำ กับจดหมายที่ประกอบด้วยคำ 5,000 คำ อาจถูกพิจารณาว่าสอดคล้องกับเอกสารสอบถาม เพราะว่ามันถูกจัดอยู่ในเวคเตอร์เดียวกัน อีกทั้ง Vector Space Model สามารถกำหนดช่วงของผลลัพธ์ ดังนั้น เอกสารที่เข้าคู่กันสามารถกำหนดช่วงตามระดับความสอดคล้องกับเอกสารได้

วิธีการให้น้ำหนักแก่ term แบบอัติโนมัติ ที่ได้รับความนิยม และมีประสิทธิภาพได้แก่ วิธี tf * idf (Chew. 2002) ซึ่งตั้งอยู่บนพื้นฐาน 2 ข้อ คือ ข้อแรก Term Frequency (tf) ถูกนิยามเพื่อพิจารณา term ที่กำหนดให้ว่าอธิบายรายละเอียดของเอกสารได้ดีแค่ไหน ข้อที่สองคือ Inverse Document Frequency (idf) ถูกนิยามเพื่อพิจารณาว่า term ที่ปรากฏไม่บ่อยในเอกสารส่วนมากระบบสามารถแยกแยะได้ดีเพียงใด ดังนั้นการคูณกันของ tf และ idf จึงเกิดขึ้น การคำนวน tf และ idf แสดงได้ดังรูปที่ 2.7 และ 2.8 ตามลำดับ

$$tf = \frac{\log(t+1)}{\log(dl)}$$

t : Number of times term occurs in document
dl : Length of document (number of terms)

รูปที่ 2.7 สูตรคำนวณ Term Frequency (tf)

$$idf = \log\left(\frac{N}{n}\right)$$

n : Number of documents term occurs in
N : Number of documents in collection

รูปที่ 2.8 สูตรคำนวณ Inverse Document Frequency (idf)

2.10 การแยกประเภทสารสนเทศ

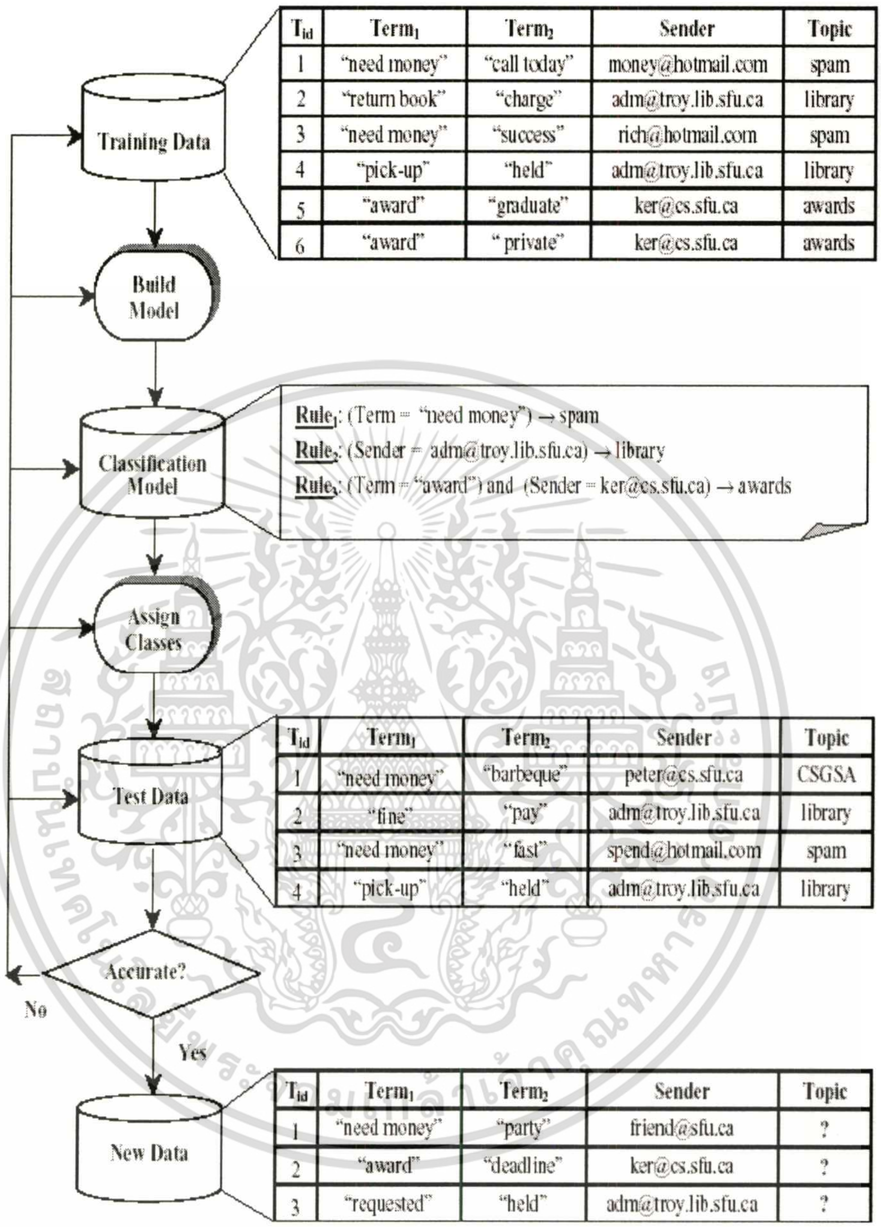
การแยกประเภทเป็นกระบวนการของการค้นหาเซตของรูปแบบ (หรือการทำงาน) ที่ใช้อธิบายและแบ่งประเภทข้อมูลและสาระสำคัญ จุดประสงค์ก็เพื่อให้รูปแบบที่สร้างขึ้นสามารถทำนายประเภทของสารสนเทศที่เรายังไม่รู้จัก หรือไม่เคยเห็นมาก่อนได้

รูปที่ 2.9 แสดงภาพการทำงานหลัก ๆ ของกระบวนการแยกประเภท จากตัวอย่าง เพื่อพิจารณาเนื้อหาของอีเมลว่าสอดคล้องกับหัวข้อใด ใช้เนื้อหาของอีเมลซึ่งประกอบด้วย term 2 terms และผู้ส่ง ในการทำนายหัวข้อที่เหมาะสมกับเนื้อหาของอีเมล

การแยกประเภทประกอบด้วย 2 ขั้นตอน คือ (Itskevitch. 2001)

1. สร้างรูปแบบการแยกประเภทโดยใช้ตัวอย่างการสอน ข้อมูลแต่ละตัวต้องถูกแยกประเภทไว้ก่อน เช่น กำหนดนิยามของการแบ่งประเภทไว้ก่อน จากรูปที่ 2.9 $Term_1$,$Term_2$ และ Sender ถูกสร้างขึ้นเพื่อเป็นรูปแบบตัวอย่างการแยกประเภทข้อความแต่ละข้อความ กลไกในการแยกประเภทอาจอยู่ในรูปแบบ classification rules, decision trees, mathematical formulae หรือ neural network จากตัวอย่างที่นำเสนอในรูปที่ 2.9 ใช้วิธี classification rules

2. รูปแบบที่ถูกสร้างในขั้นตอนที่ 1 จะถูกทดสอบโดยกำหนดข้อมูลเป็นตัวอย่างการทดสอบ ซึ่งต่างกับตัวอย่างการสอน ทุกส่วนของตัวอย่างการทดสอบถูกแบ่งประเภทในขั้นที่สูงขึ้น ความถูกต้องของการแบ่งประเภท ถูกตัดสินโดยการเปรียบเทียบการแบ่งประเภทที่ถูกต้องจริง ๆ ของตัวอย่างการทดสอบกับการแบ่งประเภทที่ได้จากระบบ ถ้าความถูกต้องของระบบได้รับความพอใจ มันก็จะถูกนำไปใช้ในการแบ่งประเภทต่อไป



T_{id}	$Term_1$	$Term_2$	Sender	Topic
1	"need money"	"call today"	money@hotmail.com	spam
2	"return book"	"charge"	adm@troy.lib.sfu.ca	library
3	"need money"	"success"	rich@hotmail.com	spam
4	"pick-up"	"held"	adm@troy.lib.sfu.ca	library
5	"award"	"graduate"	ker@cs.sfu.ca	awards
6	"award"	" private"	ker@cs.sfu.ca	awards

T_{id}	$Term_1$	$Term_2$	Sender	Topic
1	"need money"	"barbeque"	peter@cs.sfu.ca	CSGSA
2	"fine"	"pay"	adm@troy.lib.sfu.ca	library
3	"need money"	"fast"	spend@hotmail.com	spam
4	"pick-up"	"held"	adm@troy.lib.sfu.ca	library

T_{id}	$Term_1$	$Term_2$	Sender	Topic
1	"need money"	"party"	friend@sfu.ca	?
2	"award"	"deadline"	ker@cs.sfu.ca	?
3	"requested"	"held"	adm@troy.lib.sfu.ca	?

รูปที่ 2.9 กระบวนการแยกประเภท

2.11 ตัวอย่างแนวทางในการกรองข้อมูล

ธุรกิจผู้ค้าส่งเกี่ยวกับอุตสาหกรรมรถยนต์รายหนึ่งเปลี่ยนวิธีการติดต่อสื่อสารกับโรงงานรถยนต์จากโทรสาร มาเป็นการส่งอีเมล ซึ่งการใช้สื่อนี้ปรากฏว่าประสิทธิภาพการทำงานสูงขึ้นเพราะทำให้การทำงานเร็วขึ้น การแลกเปลี่ยนข้อมูลที่สำคัญมีความน่าเชื่อถือมากขึ้น เช่น รายละเอียดผลิตภัณฑ์ รายการราคา และแบบทางเทคนิค แต่โชคไม่ดี ที่การเปลี่ยนมาใช้อีเมล เป็นผล

ให้มีการติดต่อที่เป็นเรื่องส่วนตัว ซึ่งไม่ต้องการให้เพิ่มขึ้นจากการใช้อีเมล เนื่องจากบริษัทต้องการรักษาความปลอดภัยให้กับหุ้นส่วนทางธุรกิจที่ติดต่อสื่อสารกันอย่างถูกต้องตามกฎหมายกับตน จึงมองหาวิธีการที่เหมาะสมเพื่อจำกัดอีเมลที่ไม่ต้องการ ลดปริมาณการส่งอีเมลที่เป็นเรื่องส่วนตัวให้เหลือน้อยที่สุดเท่าที่จะทำได้ (Group Technologies AG. 2001)

บริษัทจึงมีความต้องการดังนี้

1. จำกัดการใช้อีเมลในเรื่องส่วนตัว ตรวจสอบเนื้อหาของอีเมล เพื่อมองหาคำ หรือประโยคที่ไม่ต้องการไปในข่าวสารนั้น ถ้าตรวจสอบพบก็จะขัดขวางการส่งนั้น
2. ป้องกันการส่งอีเมล จากพนักงานที่มีส่วนเกี่ยวข้องกับการส่งอีเมลน้อย เป็นแบบช่วงเวลา เข้มงวดกับการส่งอีเมลในเรื่องส่วนตัว ผู้ส่งเหล่านี้จะถูกกำหนดว่าเป็นกลุ่มที่จะถูกขัดขวางการส่ง
3. หุ้นส่วนทางธุรกิจบางรายเท่านั้นที่จะถูกกำหนดให้เป็นผู้รับอีเมลจากทางบริษัทได้ และสำหรับพนักงานบางคนที่สามารถส่งอีเมลได้

จะเห็นว่าความต้องการของบริษัทมีความหลากหลายมาก เฉพาะฉะนั้น กฎที่ใช้จึงควรมีความยืดหยุ่น ซึ่งสรุปได้ ดังนี้

1. สร้างรายการของคำที่ไม่ต้องการขึ้น และข่าวสารแต่ละอันที่จะรับหรือส่งออกไป จะต้องถูกตรวจสอบกับรายการของคำเหล่านี้
2. อีเมลที่ประกอบด้วยส่วนที่แสดงว่าห้ามส่งจะถูกกักไว้ในไดเรคทอรีผู้รับและผู้ส่งจะถูกกักไว้โดยดูจากข่าวสารในอีเมล
3. คนที่กระทำผิดจะถูกเก็บอีเมลแอสเดรสไว้ และข่าวสารที่ส่งจากกลุ่มคนเหล่านี้จะถูกกันไว้ ไม่ให้ถูกส่งออกไป
4. สำหรับการส่งของแต่ละคน จะมีการทำรายชื่อของผู้รับไว้ ซึ่งเป็นรายชื่อที่อนุญาตให้ส่งอีเมลไปหาได้

รูปที่ 2.10 แสดงตัวอย่างสูตรที่ใช้ตามที่กล่าวมาข้างต้น

```
CHECK Subject AND Body AGAINST WordList
  If Word "Cheesecake" = "4 Times in Body" THEN
Deny = 1


IF Deny = 1 THEN GOTO Action

Action:
 Mail = Copy to Quarantine
 Notify = Administrator, Recipient

--------------------------------------------------
-

IF Sender = in DenyList THEN GOTO Action

Action:
  Mail = Delete
  Notify = No

--------------------------------------------------
-

IF Recipient NOT in AllowList THEN Action

Action:
      Mail = Delete
      Notify = Sender
```

รูปที่ 2.10 ตัวอย่างสูตรการกรองอีเมล

บทที่ 3

งานที่เกี่ยวข้องกับการกรองไปรษณีย์

มีวิธีการมากมายที่จะบ่งชี้ว่าจดหมายที่ส่งมาเป็นจดหมายขยะ โดยสังเกตจาก วลีเฉพาะ เช่น Free money, Only $, be over 21 การใช้สัญลักษณ์ซ้ำ ๆ กันหลาย ๆ ครั้ง เช่น !!!!! หรือชื่อโดเมนของผู้ส่ง เช่น .com และจดหมายขยะส่วนใหญ่มักจะไม่ถูกส่งมาจาก .edu

การแยกอีเมล โดยวิเคราะห์จากอีเมลแอสเดรสที่รู้จักคุ้นเคย เช่น จะแทน Sdumais@microsoft.com ด้วย Susan Domais และการพิจารณาว่าส่งมาแบบ mailing list หรือไม่ อีกวิธี คือ การพิจารณาว่ามีเอกสารแนบหรือไม่ เนื่องจากจดหมายขยะมักจะไม่มีเอกสารแนบมาด้วย และพิจารณาจากเวลาในการส่ง เช่น จดหมายขยะมักถูกส่งในตอนกลางคืน

การกรองไปรษณีย์จะขึ้นอยู่กับความต้องการของแต่ละบุคคล และมักจะไม่เหมือนกัน ดังนั้น ระบบต้องมีความยืดหยุ่น และออกแบบให้มีความสามารถในการกรองที่หลากหลาย เพื่อรองรับความต้องการของแต่ละคนได้

3.1 การออกแบบเครื่องมือสำหรับกรองอีเมล

Chew (2002) ได้เสนอเครื่องมือสำหรับกรองอีเมล จุดประสงค์ของเครื่องมือนี้ คือ เพื่อแยกแยะระหว่างจดหมายที่ต้องการกับจดหมายขยะ เครื่องมือนี้ประกอบด้วยส่วนประกอบย่อยดังนี้ คือ

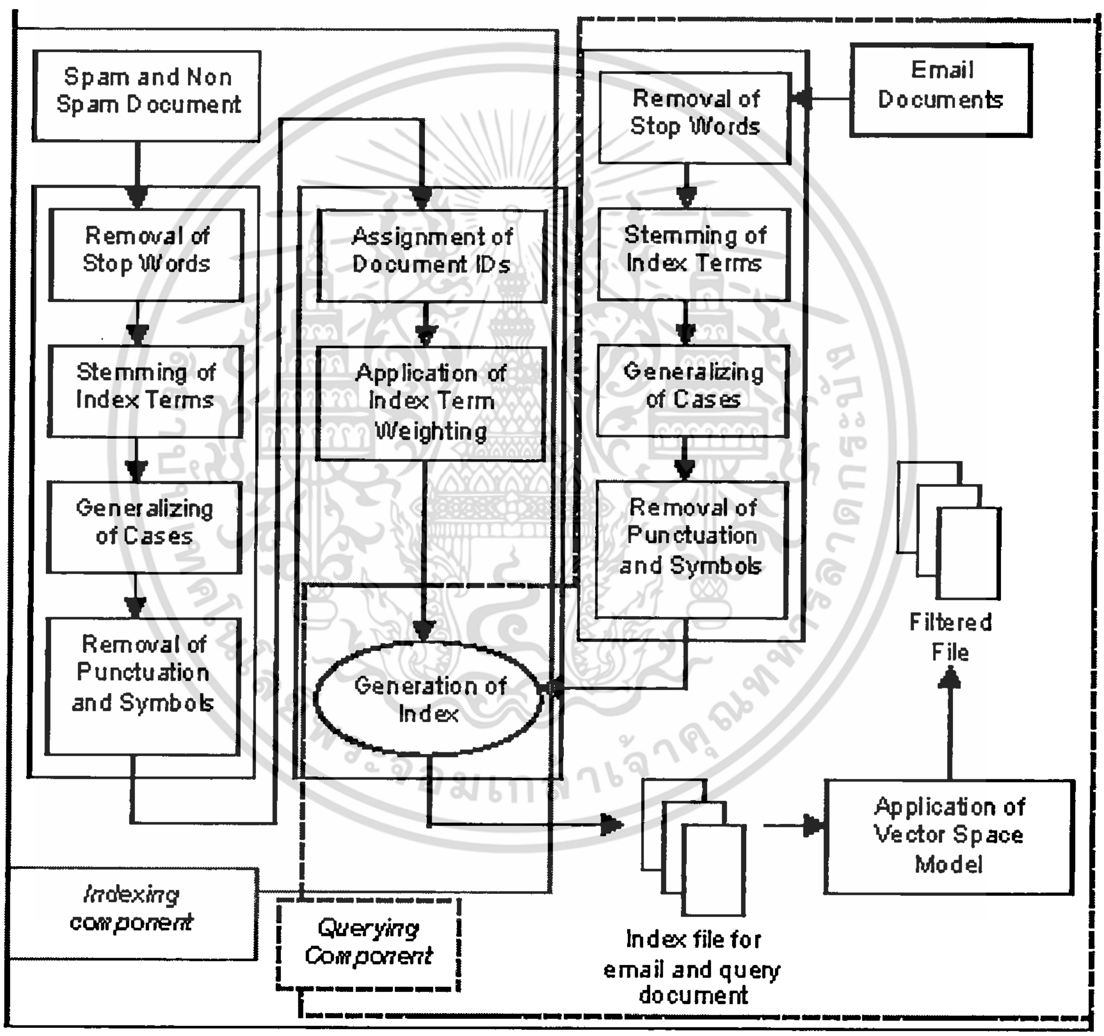
ส่วนที่หนึ่ง คือ indexing component ที่มีเซตของเอกสารที่เป็น spam และ non-spam โครงสร้างส่วนนี้จะสร้างดรรชนี โดยใช้ index term ส่วนที่สอง คือ querying component ที่มีเซตของเอกสารสอบถาม โปรแกรมจะวิเคราะห์เอกสารสอบถาม โดยเปรียบเทียบดรรชนี เพื่อสร้างผลลัพธ์สำหรับคำถาม ซึ่งก็คือ รายการของเอกสารที่อยู่ในช่วงที่สัมพันธ์กัน ภาพรวมการทำงานของทั้งสองส่วนแสดงได้ดังรูปที่ 3.1 ซึ่งใช้พื้นฐานของการประมวลผลคำในการประมวลผล

3.1.1 Indexing Component

ในส่วนของ Indexing Component จะรวบรวมเอกสารอีเมลเป็นข้อมูลเข้า และจัดการให้ข้อมูลเข้านี้อยู่ในรูปแบบมาตรฐาน คำนวณค่าถ่วงน้ำหนัก เพื่อประยุกต์ใช้กับคำดรรชนี และสร้างแฟ้มดรรชนีตามลำดับ การทำงานของส่วนประกอบนี้ มีรายละเอียดดังนี้

- แฟ้มข้อมูลเข้า

ส่วนนี้จะรวบรวมเอกสารที่เป็นข้อมูลเข้า ได้แก่ เอกสารที่เป็น spam และ non spam เพื่อใช้เป็นตัวอย่างการสอน ข้อมูลเข้าจะต้องอยู่ในรูปแบบมาตรฐาน เพื่อใช้ใน indexing component ทำให้สามารถจำแนกเอกสารจดหมายได้อย่างมีขอบเขต รูปที่ 3.2 แสดงรูปแบบของแฟ้มของข้อมูลเข้า



รูปที่ 3.1 โครงสร้างการออกแบบการกรองเฉพาะจดหมายที่ต้องการและจดหมายขยะ

```
.I <index>
.T
<title>
.W
<content>
```

รูปที่ 3.2 รูปแบบของแฟ้มข้อมูลเข้า

จากรูปที่ 3.2 จะมีตัวกำหนดอยู่ 3 ตัว สำหรับเอกสารแต่ละฉบับ อันได้แก่ I, T และ W และเป็นรูปแบบนี้เช่นเดียวกันกับทุกเอกสาร โดยที่ I เป็นตัวชี้เอกสาร เป็นเลขเฉพาะ และแสดงถึงเอกสารแต่ละฉบับที่เป็นอินพุต ตัวเลขจะเริ่มต้นจาก 1 และเพิ่มขึ้นทีละหนึ่ง T แสดงหัวข้อย่อยของเอกสาร เป็นการอธิบายเอกสารอย่างสั้น ๆ กะทัดรัด ว่าประกอบด้วยอะไรบ้าง และ W แสดงรายละเอียดของเอกสาร เป็นส่วนที่ indexing component ใช้ทำงาน และสร้างแฟ้มดรรชนี โดยใช้คำดรรชนีที่ถ่วงน้ำหนัก จากเอกสารแต่ละฉบับ

- การจัดรูปแบบมาตรฐาน

แฟ้มข้อมูลเข้าเกี่ยวกับรายละเอียดของเอกสาร จะนำมาทำให้อยู่ในรูปแบบมาตรฐานเฉพาะ โดยการเอา stop words ออก การ stemming ตัว index term การแปลงตัวอักษรให้เป็นตัวเล็ก และการตัดช่องว่างและสัญลักษณ์ออก เมื่อผ่านขั้นตอนนี้ จะได้รายละเอียดของเอกสารที่สั้น และกะทัดรัด ต่อมา โปรแกรมจะกำหนดตัวระบุ (ID) สำหรับเอกสารแต่ละฉบับ และคำนวณน้ำหนัก สำหรับ index term แต่ละคำในเอกสาร

- การถ่วงน้ำหนัก

การคำนวณน้ำหนัก จะใช้วิธีการตามที่แสดงในหัวข้อ 2.9.3 Vector Space Model โดยมีการคำนวณค่าน้ำหนัก 2 แบบ คือ tf และ tf * idf แฟ้มดรรชนีที่สร้างขึ้น ประกอบด้วย ข้อมูลของแต่ละ index term แต่ละคำและน้ำหนักที่สอดคล้องกับเอกสาร ดังแสดงในรูปที่ 3.3

```
.T <term>
.D <email_doc_no> <weight>
.D <email_doc_no> <weight>
...
.T <term>
.D <email_doc_no> <weight>
...
```

รูปที่ 3.3 รูปแบบของ index file

จากรูปจะเห็นว่า มีการใช้ตัวกำหนดเพิ่มขึ้น 2 ตัว ได้แก่ T คือ Index Term และ D คือ เลขที่เอกสารที่มีค่าน้ำหนักอยู่ด้วย การที่ D ถูกกำหนดอยู่ด้านล่างของ T นั่นหมายถึง มี T ปรากฏอยู่ในเอกสารนี้ querying component ใช้แฟ้มดรรชนีในการเปรียบเทียบในช่วงการวิเคราะห์เอกสารสอบถาม

3.1.2 Querying Component

Querying Component จะรวบรวมเอกสารสอบถามมาเป็นข้อมูลเข้า และทำข้อมูลเข้าให้มีรูปแบบมาตรฐาน ต่อมาก็จะคำนวณค่าน้ำหนักของ index term ของข้อมูลเข้า แฟ้มดรรชนีของ index component และค่าน้ำหนักของ index term ใน query component จะถูกนำมาใช้ในสูตรคำนวณความสอดคล้อง และจะได้แฟ้มผลลัพธ์ ซึ่งแสดงเอกสารที่เข้าคู่กัน การทำงานและการจัดเรียงตามความคล้ายกับเอกสารสอบถามของส่วนประกอบนี้ มีรายละเอียด ดังต่อไปนี้

- แฟ้มข้อมูลเข้า

เอกสารที่เป็นข้อมูลเข้าในกรณีนี้ คือ อีเมลที่เข้ามา หรือที่ได้รับจากการสุ่มจากอีเมล ที่เป็น spam และ non-spam ถ้าเป็นระบบการกรองที่ดีต้องสามารถแบ่งแยกได้ถูกต้อง ตามประเภทของเอกสารว่าควรอยู่ในส่วนไหน รูปแบบของแฟ้มข้อมูลเข้า มีลักษณะคล้ายกับแฟ้มข้อมูลเข้า ใน indexing component

- การจัดรูปแบบมาตรฐาน

เพื่อให้ง่ายในการทำงาน และมีความสอดคล้องกัน รูปแบบของเนื้อหาของเอกสารจึงกระทำอยู่บนพื้นฐานตามรูปแบบเช่นเดียวกับ indexing component

- การถ่วงน้ำหนัก

จาก index component ที่กล่าวถึงมาแล้ว ทำให้ขณะนี้มีค่าน้ำหนัก อยู่ 2 ค่า ที่นำมาใช้กับ index term ดังนั้น เมื่อใดก็ตามที่ค่าน้ำหนัก ที่นำมาประยุกต์ใช้กับ indexing component ต้องการเปรียบเทียบความใกล้เคียงกับน้ำหนัก ที่นำมาประยุกต์ใช้กับ querying component จะต้องมั่นใจว่า มันถูกคำนวณโดยใช้แบบแผนเดียวกัน เพื่อให้สามารถเปรียบเทียบค่าทั้งสองได้

- การวัดค่าความเหมือน

สูตรคำนวณความเหมือน ใช้น้ำหนักของ index term เพื่อพิจารณาการเข้าคู่กันของ querying component และ indexing component ค่าที่ได้จะถูกจัดเรียงจากมากไปน้อย และแสดงในแฟ้มผลลัพธ์ที่แสดงเอกสารที่เข้าคู่กัน ซึ่งจะมีการจัดลำดับตามระดับความเหมือนกับเอกสารสอบถาม ดังแสดงในรูปที่ 3.4

Query_ID	Doc_ID	Rank	*sim (d,q)* Score
1	40	1	0.351194230667077
1	26	2	0.300211991936467
...	...	...	...

รูปที่ 3.4 รูปแบบของแฟ้มผลลัพธ์

รูปที่ 3.5 แสดงการใช้ประโยชน์จากวิธี Vector Space Model โดยใช้สูตร sim(d,q) โดยลูกศร R แสดงถึง เอกสารที่นำมาเข้าคู่ และลูกศรที่มีตัวเลขกำกับ แสดงถึง เอกสารสอบถามที่แตกต่างกัน เมื่อใช้สูตร sim(d,q) จะสามารถคำนวณมุมระหว่างเอกสารสอบถาม และเอกสารที่นำมาเข้าคู่ออกได้ จะสังเกตได้ว่า ความยาวของเวคเตอร์จะไม่ถูกนำมาพิจารณา

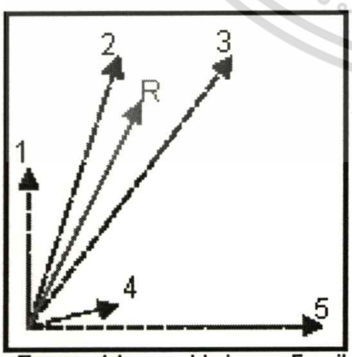


Figure 14 a : Various Email Document Vectors

Position No	Query No
1	2
2	3
3	1
4	4
5	5

Figure 14 b : Result Set After Being Applied *sim(d,q)*

รูปที่ 3.5 ตัวอย่างการใช้ sim(d,q) ในการคำนวณเวคเตอร์ของเอกสาร

3.1.3 การจัดรูปแบบเพื่อเพิ่มความถูกต้อง

เพื่อเพิ่มความถูกต้องให้กับระบบการกรอง และมั่นใจได้ว่าเอกสารอยู่ในรูปแบบมาตรฐานที่เฉพาะเจาะจง จึงมีการนำเทคนิคต่าง ๆ มาใช้กับเอกสารก่อนที่จะส่งต่อไปยังเครื่องมือกรองเอกสาร ดังนี้ (Chew. 2002)

- Memo Header Information

อีเมลประกอบด้วยโครงสร้างที่สำคัญ 2 ส่วน คือ ส่วนหัว และส่วนรายละเอียดเนื้อหาส่วนหัว ประกอบด้วย Sender, Subject, To, Cc, From และ Reply-To และเนื้อหาของรายละเอียด ประกอบด้วยข้อมูลของอีเมลนั้น โดยทั่วไป เมื่อผู้อ่าน อ่านข้อมูลส่วนหัวอย่างคร่าว ๆ ก็สามารถรู้ได้ว่าผู้ส่งคือใคร อย่างไรก็ตาม วิธีการที่มีอยู่ในปัจจุบัน ผู้ที่ต้องการส่งจดหมายขยะ สามารถเขียนโปรแกรมเพื่อเปลี่ยนแปลงข้อมูลให้ดูคล้ายว่าเป็นข้อมูลที่ต้องการจะเปิดขึ้นมาอ่านได้ง่าย ดังนั้น ส่วนหัวของเอกสารอย่างเดียวจึงไม่เพียงพอสำหรับการพิจารณาว่าเป็นจดหมายขยะหรือเป็นจดหมายที่ไม่ต้องการ ต้องพิจารณาข้อมูลส่วนรายละเอียดของเอกสารประกอบด้วย

- Stop Words

เพื่อให้การกรองเอกสารมีความถูกต้องยิ่งขึ้น จึงใช้แฟ้มของ stop words คือ คำที่พบบ่อย ๆ กว่าคำอื่น ๆ แต่ไม่มีผลกับการกรอง ซึ่งจะถูกตัดออกโดยตัวระบบ เช่น I, the และ although เพราะคำเหล่านี้ไม่ถูกนำไปใช้ในการคำนวณความสอดคล้องของเอกสาร

- การปรับสู่รากศัพท์ (Stemming)

Stemming คือ การพยายามลดการปรากฏของคำเดียวกัน แต่รูปแบบต่างกันในเอกสาร เช่น eating, eat หรือ ate ทั้งสามคำจะถูกแทนด้วย eat ซึ่งการทำเช่นนี้จะทำให้เสียไวยากรณ์ของคำไป แต่เรามุ่งเน้นไปที่การปรากฏของคำเพื่อการกรองที่ถูกต้องมากกว่าการสงวนไวยากรณ์ไว้

- การปรับตัวอักษรเล็กใหญ่ (Generalizing of the Cases)

โดยทั่วไป ในเนื้อหาของเอกสารประกอบด้วย ประโยคหลาย ๆ ประโยค และมีตัวอักษรที่เน้นความสำคัญมากมาย ดังนั้น เมื่อมีคำเหล่านี้ปรากฏ ก็มักจะขึ้นต้นด้วยอักษรตัวใหญ่ เพื่อให้อยู่ในรูปแบบเดียวกัน จึงต้องแปลงอักษรทุกตัวเป็นตัวเล็ก เช่น คำว่า Private จะถูกแปลงเป็น private อีกทั้ง ทำให้สนับสนุนการเกิดขึ้นของคำว่า private เพื่อนำไปใช้ในการคำนวณความสอดคล้องได้แม่นยำยิ่งขึ้น

- สัญลักษณ์ต่าง ๆ

เครื่องหมายวรรคตอนและสัญลักษณ์ในเอกสาร จะถูกพิจารณาว่า ไม่เกี่ยวข้องกับการนำไปตัดสินใจ จึงไม่นำไปเป็นข้อมูลสำหรับโปรแกรมการกรองจดหมายขยะ เพราะระบบไม่จำเป็นต้องใช้ประโยชน์จากขอบเขตประโยคในการวิเคราะห์เอกสาร ดังนั้น เครื่องหมายวรรคตอน และสัญลักษณ์ เช่น % , # และ @ จะถูกตัดออกจากระบบ

- การคำนวณน้ำหนัก

เพื่อใช้ประโยชน์จาก Vector Space Model จำเป็นต้องมีกลไกการคำนวณน้ำหนักด้วย ซึ่งมี 2 แบบ คือ tf และ tf * idf

3.2 อัลกอริทึมการเรียนรู้แบบอินดักทีฟ

Inductive Learning เป็นการเรียนรู้ด้วยวิธีพิสูจน์จากกรณีเฉพาะ ตัวอย่างของ Inductive Learning Algorithm ได้แก่ CN2 ซึ่งใช้ใน MAGI (Murch and Johnson. 1999)

การใช้อัลกอริทึมการเรียนรู้ ที่มีแนวคิดการลดทอนทำให้ไม่เกิดปัญหาคอขวด การทำงานของอัลกอริทึม นี้จะใช้วิธีการทำซ้ำจนกระทั่งไม่มีตัวอย่างใน training set ซึ่งแต่ละรอบของการทำซ้ำจะค้นหา complex ซึ่งครอบคลุมตัวอย่างที่มีอยู่มากมายของคลาส C และบางส่วนของคลาสอื่น ๆ complex คือ การเกิดขึ้นร่วมกันของการทดสอบแอตทริบิวต์ รูปแบบของ complex จะเป็นส่วนเงื่อนไขของกฎที่ถูกสร้างขึ้น เมื่อคลาส C เป็นผลลัพธ์ของกฎที่ถูกสร้าง และเมื่อค้นพบ complex ก็จะย้ายตัวอย่างที่เกี่ยวข้องออกจาก training set และนำกฎมาต่อท้าย decision list

if *complex* then *Class C*

complex เป็นตัวกำหนดว่าจะเพิ่ม term ใหม่ที่มีลักษณะร่วมกัน หรือเอาส่วนที่ไม่มีลักษณะร่วมกันออกไป star เป็น เซตที่ เก็บ complex ทั้งหมดที่กำลังพิจารณา เพื่อหา complex ที่ดีที่สุด การกำหนดขอบเขตที่เฉพาะเจาะจงใช้วิธีการทำซ้ำโดย intersect กับเซตของตัวเลือกที่เป็นไปได้ทั้งหมดกับ complex ณ ขณะนั้น ส่วนที่ไม่การเปลี่ยนแปลง และ null complex จะถูกย้ายออก

star จะถูกปรับแต่ง โดยการประเมินค่าของ complex ใหม่แต่ละอัน และตัด complex ที่อยู่ในช่วงที่ต่ำที่สุดทิ้ง โดยการใช้ ฟังก์ชันการประเมินสองแบบ คือ การใช้ตัววัด entropy measure และวิธีการทาง heuristic

แบบแรกการหาค่า entropy สามารถคำนวณได้จาก

$$Entropy = -\sum_{i} p_i \log_2(p_i)$$

โดยที่ค่า entropy ยิ่งต่ำ ค่า complex ก็จะยิ่งดี อันดับแรกถ้าพบ complex ที่ครอบคลุมเซต E' ของตัวอย่าง และการกระจายของความน่าจะเป็น $P = (p_1, \ldots, p_n)$ การวัดค่าด้านบนก็จะถูกนำมาประยุกต์ใช้กับ complex

สำหรับแบบที่ 2 เป็นวิธีการทาง heuristic ซึ่งใช้ในการค้นหา ในการปรับแต่ง star วิธีการประมาณความผิดพลาดโดยใช้วิธี Laplacian แสดงได้ดังนี้

$$Accuracy(n, n_c, k) = \frac{(n - n_c + k - 1)}{(n + k)}$$

เมื่อ

n = จำนวนรวมทั้งหมดของตัวอย่างที่เกี่ยวกับกฎข้อนั้น

n_c = จำนวนของตัวอย่างที่เป็น positive ที่เกี่ยวกับกฎข้อนั้น

k = จำนวนของ class ในปัญหา

การประเมินค่าแบบที่สองนี้ ใช้ตัดสินใจเมื่อ complex มีความสำคัญ ซึ่ง complex จะมีความสำคัญเมื่อมันประกอบด้วยกฎเกณฑ์ที่ไม่น่าเป็นไปได้ ที่ปรากฏโดยบังเอิญจากการทำงาน และสะท้อนให้เห็นภาพความสัมพันธ์ที่แท้จริงระหว่างค่าของแอตทริบิวต์ และคลาส

ความสำคัญคำนวณได้โดยใช้วิธี likelihood ratio statistic ดังนี้

$$2\sum_{i=1}^{n} f_i \log\left(\frac{f_i}{e_i}\right)$$

เมื่อการกระจาย $F = (f_1,\ldots,f_n)$ เป็นการกระจายความถี่ของการสังเกต และ $E = (e_1,\ldots,e_n)$ เป็นการกระจายความถี่ของค่าคาดหวัง วิธีการทางสถิตินี้จัดเตรียมวิธีวัดทางทฤษฎีของระยะห่างระหว่างการกระจายสองตัว ภายใต้การตั้งสมมุติฐานของสถานการณ์ วิธีการทางสถิตินี้มีค่าการกระจายโดยประมาณ λ^2 โดยมี degree of freedom n-1

3.3 ส่วนประกอบระบบการกรองไปรษณีย์ MAGI

3.3.1 ระบบที่ต้องการ

เอเจนท์ต้องพยายามเรียนรู้การกระทำต่อไปนี้ (Murch and Johnson. 1999)

- ข้อความที่จัดเก็บอยู่ในกล่องไปรษณีย์ที่แตกต่างกัน
- จดหมายขยะที่ผู้ใช้ไม่เคยสนใจที่จะเปิดอ่าน
- ข้อความที่ส่งต่อไปให้ผู้ใช้คนอื่น ๆ

จุดประสงค์ของการทำ MAGI เพื่อให้เอเจนท์ซ่อนรายละเอียด (transparent) เท่าที่จะเป็นไปได้ MAGI เป็นเอเจนท์ที่จะไม่ปรากฏให้ผู้ใช้เห็น และไม่ส่งผลกระทบต่อผู้ใช้ในขณะที่กำลังจัดการกับไปรษณีย์ แต่ผู้ใช้สามารถร้องขอความช่วยเหลือจากเอเจนท์ได้ ภาพรวมการทำงานของเอเจนท์สามารถดูได้จากรูปที่ 3.6 ซึ่งมีการแบ่งเป็นมอดูล โดยที่แต่ละมอดูล ติดต่อกันโดยการใช้แฟ้มร่วมกัน ซึ่งทั้ง 3 มอดูล สามารถสรุปได้ดังนี้

(1) Mail Interface มอดูลนี้รับผิดชอบในการสังเกตและติดต่อกับผู้ใช้ การกระทำของผู้ใช้ จะถูกบันทึกไว้ เพื่อนำไปใช้ในการสร้างกฎต่อไป ซึ่งการติดต่อกับผู้ใช้เป็นการปฏิบัติงานแบบอัตโนมัติ

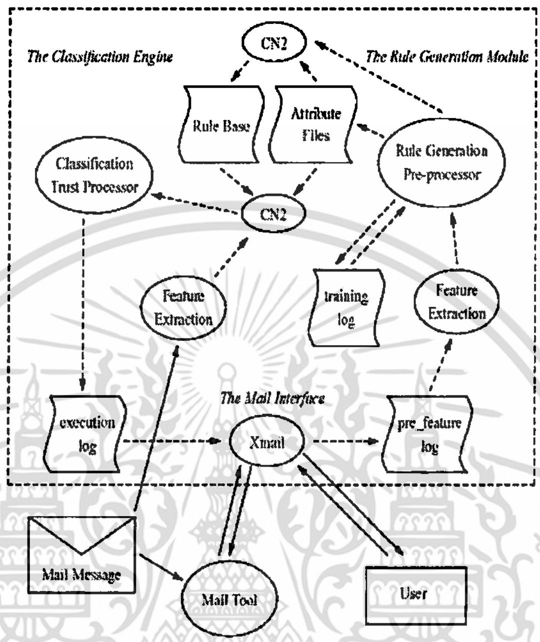
(2) Rule Generation Module บันทึกการกระทำของผู้ใช้ที่ทำกับข้อความในไปรษณีย์จะถูกเก็บไว้ โดย Mail Interface จะนำไปสร้างเป็นตัวอย่างฝึกสอน (training example) เพื่อนำไปสร้างเป็นกฎ การกระทำแต่ละอันจะได้รับ life-time ขณะที่มันมีส่วนช่วยในการสร้างกฎ นั่นคือ โครงสร้างของผู้ใช้สามารถเป็นตัวสะท้อนกลับแก่ผู้ใช้ว่าควรให้เวลาเท่าใด Rule Generation Module คือผู้รับผิดชอบในการจัดการส่วนนี้ และปรับแต่งชุดฝึกสอน (training set) เมื่อตัวอย่างฝึกสอนเริ่มที่จะเก่าไปแล้ว

(3) Classification Engine มอดูลนี้จะทดสอบข้อความในจดหมายใหม่ที่เข้ามาว่าตรงกับกฎที่มีอยู่หรือไม่ และวิเคราะห์ผลลัพธ์ที่ได้ นอกจากนี้มอดูลนี้ยังรับผิดชอบการประเมินค่าความมั่นใจของผลลัพธ์ที่ได้จากกฎนั้นด้วย

3.3.2 Mail Interface

Mail Interface จะอยู่ระหว่างผู้ใช้ และ mail tool ดังนั้น คำสั่งจากผู้ใช้จะถูกดักฟังโดย mail agent ก่อนที่จะถูกส่งไปยัง mail tool ทำให้เอเจนท์สังเกตการจัดการไปรษณีย์ของผู้ใช้ ในทำนองเดียวกัน การตอบสนองสามารถดักฟังโดย mail agent ได้ด้วย ทำให้ mail agent สามารถติดต่อได้โดยตรงทั้งผู้ใช้ และ mail tool จากแนวคิดนี้ผู้ใช้สามารถทดสอบ และเลือกการกระทำที่เอเจนท์เสนอ ขณะที่ใช้ mail tool ถ้าผู้ใช้พอใจการนำเสนอการกระทำของเอเจนท์แล้ว เอเจนท์ก็สามารถติดต่อกับ mail tool ได้โดยตรงโดยไม่ต้องรับผลตอบสนองกลับจากผู้ใช้

เนื่องจากเอเจนท์ซ่อนรายละเอียดการทำงานกับผู้ใช้ บางครั้งผู้ใช้อาจจะต้องการติดต่อกับเอเจนท์โดยตรง จึงต้องมีการจัดเตรียมการทำงาน เพื่ออนุญาตให้ผู้ใช้ติดต่อกับเอเจนท์



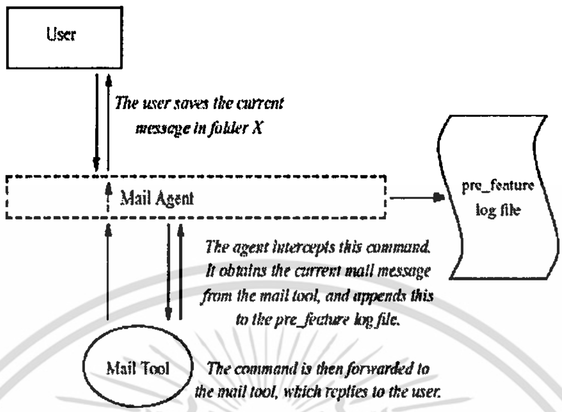
รูปที่ 3.6 ภาพรวมการทำงานของเอเจนท์

3.3.2.1 การตรวจจับการกระทำของผู้ใช้

เอเจนท์จะสนใจชุดคำสั่งเกี่ยวกับการจัดการไปรษณีย์เท่านั้น ซึ่งคำสั่งเหล่านั้นเป็นตัวรับผิดชอบหลักสำหรับแยกไปรษณีย์ใส่ใน mail box ที่แตกต่างกัน หรือลบไปรษณีย์ที่ไม่อ่าน คำสั่งจะถูกดังฟังโดยเอเจนท์ก่อนที่จะส่งต่อไปที่ mail tool ณ จุดนี้จะทราบคำสั่งทั้งหมดว่าข้อความที่ส่งมาต้องการแยกเก็บหรือลบทิ้ง ในการแยกเก็บจดหมายไว้ เอเจนท์จะร้องขอข้อความจาก mail tool เพื่อดึงลักษณะพิเศษ (feature) จากข้อความ เพื่อนำไปใช้ต่อไป ข้อความและคำสั่งจะถูกเก็บไว้ใน pre_feature log file เมื่อการสังเกตถูกสร้างขึ้น คำสั่งของผู้ใช้จะถูกกระทำ โดยเอเจนท์ส่งคำสั่งไปยัง mail tool โดยตรง และส่งผลลัพธ์กลับไปให้ผู้ใช้ ดังแสดงในรูปที่ 3.7

การลบจะเป็นสถานการณ์ที่แตกต่างออกไป เช่น มีการใช้ log file จัดการข้อความที่กำลังจะถูกลบ จำเป็นต้องมีวิธีการแก้ไขรูปแบบการทำงาน ถ้าข้อความนั้นกลายเป็นข้อความที่ไม่ต้องการลบ ในกรณีนี้เอเจนท์จะเก็บบันทึกข้อความทั้งหมดที่จะถูกลบ โดยไม่เก็บบันทึกนี้ลง log file ต่อมาภายหลัง ถ้าข้อความนี้กลายเป็นข้อความที่ถูกลบจริงๆ เมื่อการทำงานของ mail tool เสร็จสิ้นลง

เอเจนท์ถึงจะบันทึกข้อความทั้งหมดที่ถูกลบลง log file



รูปที่ 3.7 การปฏิบัติการสังเกตข่าวสารของ mail agent

3.3.2.2 การทำให้การกระทำของผู้ใช้เป็นอัตโนมัติ

กลไกการแบ่งประเภท ทดสอบ และตัดสินไปรษณีย์ที่เข้ามา ถ้าเอเจนท์เสนอ action (การกระทำ) มา action ที่เสนอมาจะถูกเก็บใน log file ที่เรียกว่า executing log ซึ่งประกอบด้วย action หรือ command (คำสั่ง) และความหมายบางส่วนของข้อความที่ action นั้นจะกระทำ เมื่อผู้ใช้ติดต่อกับเอเจนท์และตัดสินใจแล้วว่า action ใดจะถูกปฏิบัติ เอเจนท์จะค้นหาข้อความในกล่องไปรษณีย์ และปฏิบัติการ action นั้น แล้วแสดงบันทึกการปฏิบัติการ action นั้นแก่ผู้ใช้

ค่าความเชื่อมั่น (Confidence) ของ action ที่นำเสนอโดยเอเจนท์แสดงจำนวนของกฎที่ให้ผลจาก action เดียวกัน โดยนำ trust threshold มาใช้ในการตัดสินใจ ถึงแม้ว่า action ที่เอเจนท์นำเสนอจะมีค่าความเชื่อมั่นสูง โดยที่ค่า threshold นี้จะเท่ากันกับทุกการกระทำและเป็นค่าการตัดสินใจที่ได้จากการสังเกต

การนำเสนอ action และ การปฏิบัติงานโดยตัวเอเจนท์ ยังถูกนำไปพิจารณาสำหรับ trainning examples ในอนาคต ซึ่งเอเจนท์ที่ทำนาย action ได้ถูกต้อง นำไปสู่ action ที่ถูกต้อง ทำให้นำไปสู่การสร้างกฎที่ถูกต้องต่อไป

3.3.2.3 ผลตอบกลับของผู้ใช้

หน้าที่สำคัญในการติดต่อกับเอเจนท์ คือ สามารถเลือกคู่ของ message – action ที่นำเสนอโดยเอเจนท์มาปฏิบัติงาน ด้วยเหตุนี้ การทำเบราเซอร์ จึงถูกออกแบบขึ้นมา เพื่อแสดงรายการ message – action ทั้งหมด และแสดงคู่ที่กำลังปฏิบัติงาน โดยที่ข้อความที่ถูกอ้างอิงจาก message – action จะมีการกำหนดเลขที่ข้อความ (Message ID) ซึ่งผู้ใช้สามารถทดสอบบางข้อความจากราย

การนี้ โดย action ที่ไม่ต้องการให้ปฏิบัติงานก็สามารถที่จะไม่เลือก ถ้าผู้ใช้พอใจกับการทำนายของเอเจนท์ มันก็สามารถปฏิบัติงานได้ เพราะมีการจำแนกการคาดการณ์ที่ถูกต้อง และไม่ถูกต้องของเอเจนท์ จึงต้องมีการเพิ่มผลตอบกลับเข้าไป ซึ่งนำไปใช้กับคุณลักษณะที่ใช้ในการสร้างกฎ

อีกแนวทางหนึ่งในการเพิ่มความน่าเชื่อถือ โดยการเพิ่มค่าความเชื่อมั่นรวมไปกับ message – action แต่ละคู่ รวมทั้งกำหนดค่า trust thresholds ของทุก ๆ action ที่เป็นไปได้ สำหรับแต่ละคู่ที่เข้ามา ค่าความเชื่อมั่นที่ได้สามารถนำมาเปรียบเทียบกับค่า trust thresholds ที่กำหนดไว้ใน mail interface สำหรับแต่ละ action ถ้าค่าความเชื่อมั่นมากกว่าค่า trust threshold แล้ว คู่ที่เข้ามานี้ก็จะถูก mark สำหรับนำไปปฏิบัติ

3.3.3 Feature Extraction

กลไก feature extraction ถูกออกแบบเพื่อให้สามารถใช้ได้ทั้ง classification module และ rule generation module เพื่อให้แน่ใจว่าข้อความนั้นจะสร้างจากคุณลักษณะที่เหมือนกัน แต่ละข้อความ จะถูกแยกเป็น 2 ส่วน ส่วนแรก คือส่วนหัวของข้อความ เป็นข้อมูลเพื่อให้ไปถึงยังอีเมลเป้าหมาย อีกทั้งข้อมูลเกี่ยวกับผู้ส่ง เวลาและวันที่ ผู้รับและสถานะของข้อมูล เป็นต้น ส่วนที่สอง คือเนื้อหา

3.3.3.1 การกรองขั้นต้น

เนื้อหาของข้อความจะถูกแจงส่วนไปเป็นคำต่าง ๆ “คำ” คือ อักขระที่ต่อเนื่องกัน คำที่ได้จะถูกจัดเรียงตามความถี่ N คำแรกจะถูกใช้ในการอธิบายข้อความ ซึ่งตัวเลข N จะถูกกำหนดโดยตัวเอเจนท์ การใช้เครื่องหมายวรรคตอนและตัวเลขจะถูกคัดออก เพื่อลดคำที่แปลกปลอม

อัตราส่วนของคำที่มีความถี่ในการพบสูงโดยส่วนใหญ่จะเป็นคำทั่วไปที่พบในชีวิตประจำวัน เช่น and, is, the เป็นต้น จึงมีการทำแฟ้มที่ประกอบด้วยคำเหล่านี้ หรือ stoplist เพื่อกรองมันออกไป ซึ่งจะทำให้ประสิทธิภาพการทำงานดีขึ้น

มีปัญหาที่เป็นไปได้หลายอย่างเกิดขึ้นกับวิธีการนี้ ได้แก่ เนื้อหาของข้อความ ประกอบด้วยช่วงของคำที่กว้าง คำกริยาสามารถอยู่ในรูปแบบของกริยาที่แสดงเวลา เช่น is, was คำนามในกรณีที่แตกต่างกัน เช่น tree, trees คำพ้อง เช่น freedom, independence รวมถึงการสะกดที่หลากหลาย เช่น color, colour จึงมีการนำพจนานุกรมคำพ้องและคำที่มีความหมายตรงกันข้าม (Thesauri) และลำดับชั้นของคำมาใช้ร่วมด้วย

3.3.3.2 การกรองขั้นสูง

ปัญหาหนึ่งที่พบในวิธีการกรองในขั้นต้น คือ การจำกัดการรับผลตอบกลับ จากการประสบความสำเร็จจากการสร้างกฎ การนำเสนอของเอเจนท์ที่ประสบความสำเร็จจะสร้าง ตัวอย่าง

ฝึกสอน (trainning example) ออกมา อย่างไรก็ตาม บางคุณลักษณะอาจสนับสนุนมากกว่า 1 action และอาจเป็นสาเหตุให้การแบ่งประเภทผิดพลาด

แนวทางในการแก้ปัญหา คือ การใช้ฐานความรู้ที่เก็บคุณลักษณะที่แตกต่างกัน แต่ละคุณลักษณะ ประกอบด้วยค่า fitness ซึ่งเปลี่ยนแปลงได้ ขึ้นอยู่กับคุณลักษณะนั้นสนับสนุนการแบ่งประเภทได้ถูกต้องหรือไม่ เมื่อมีคุณลักษณะใหม่ที่ไม่เคยพบ เช่น ยังไม่อยู่ในฐานความรู้ จะมีการกำหนดค่า fitness มาตรฐานให้กับคุณลักษณะนั้น ค่า fitness จะเป็นตัวตัดสินว่าคุณลักษณะนั้นจะผ่านไปยังระบบการกรองหรือไม่ คุณลักษณะ ที่มีค่า fitness ต่ำจะถูกพิจารณาว่าไม่มีประโยชน์และจะถูกตัดออก ขณะเดียวกัน ค่า fitness สูงจะถูกนำไปใช้ในระบบการกรอง โดยค่า fitness จะเพิ่มขึ้นถ้าคุณลักษณะนำไปสู่การแบ่งประเภทที่ถูกต้อง และจะถูกลดค่าถ้าคุณลักษณะนำไปสู่การแบ่งประเภทที่ผิด

อย่างไรก็ตาม วิธีการนี้อาจจะมีข้อผิดพลาดได้ เช่น ถ้าค่า fitness ของคุณลักษณะในตอนเริ่มต้นต่ำกว่า filtering threshold นั่นหมายถึง ต้องถูกตัดออกทันที ซึ่งมันอาจมีประโยชน์ในเวลาต่อมา การแก้ปัญหานี้ก็คือการ เก็บคุณลักษณะในช่วงเวลาที่จำกัดช่วงหนึ่ง แล้วค่า fitness ก็จะค่อย ๆ ลดลงถ้าคุณลักษณะนั้นไม่ช่วยในการแบ่งประเภท จนกระทั่งค่า fitness ต่ำกว่าศูนย์ ก็จะถูกตัดออกจากฐานความรู้

3.3.4 การเรียนรู้กฎ

อัลกอริทึมการเรียนรู้ CN2 ใช้ trainning examples ในการสร้างกฎ ซึ่ง trainning examples ได้มาจากสองกลุ่ม กลุ่มแรกสร้างจากการสังเกตผู้ใช้ จนกระทั่งกฎถูกสร้างขึ้นโดยตัวเอเจนท์เอง ผลการสังเกตจะเก็บไว้ใน pre_feature log file ประกอบด้วยข้อความและ action ณ จุดนี้คุณลักษณะจะถูกดึงจากข้อความ กลุ่มที่สอง ได้จาก trainning examples ที่ใช้ในการสร้างกฎ ถ้า examples ถูกนำมาสร้างเป็นกฎ มันก็จะได้รับ shelf-life

shelf-life เป็นค่าจากการสังเกต เมื่อสร้างกฎขึ้นมาใหม่ ก็จะทิ้งกฎชุดเก่า ซึ่งบางทีกฎเก่าอาจยังคงใช้งานได้ แต่ไม่ได้ถูกสร้างโดยการสังเกตใหม่ วิธีแก้ปัญหา คือ การคง trainning examples ทั้งหมดนี้ไว้ในช่วงเวลาหนึ่ง จนกระทั่งเกินค่า shelf-life กฎข้อนั้นก็จะถูกตัดทิ้ง

ข้อความหนึ่งข้อความอาจจะสร้าง trainning examples ที่หลากหลาย จนถึง N คุณลักษณะ ซึ่งถูกดึงออกจากส่วนเนื้อหา เพื่อแสดงให้เห็นถึงลักษณะเฉพาะของข้อความนั้น ซึ่งอัลกอริทึมการเรียนรู้ ยอมรับเพียงค่าเดียว สำหรับแต่ละแอตทริบิวต์ แต่แอตทริบิวต์หนึ่ง อาจครอบคลุมการเกิดขึ้นพร้อมกันของหลาย ๆ คุณลักษณะ จำนวนของ trainning examples เพิ่มขึ้นจากการใช้แอตทริบิวต์ ซึ่งอาจจะประกอบด้วยมากกว่า 1 คุณลักษณะ จำนวนของ example ที่ถูกสร้าง แสดงได้ดังนี้

$$Num\ of\ Examples = (a_1 + a_2 + , \ldots , + a_m)$$

เมื่อ

a = จำนวนของคุณลักษณะที่เป็นไปได้สำหรับแอตทริบิวต์ที่ได้รับ

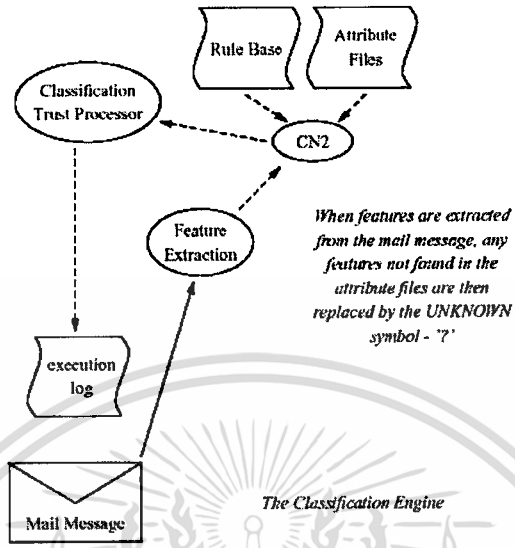
m = จำนวนของแอตทริบิวต์

rule generation module ยังสร้าง attribute files อีกด้วย เพื่อที่จะสร้างกฎ อัลกอริทึมการเรียนรู้ต้องได้รับรายการของแอตทริบิวต์และ action ที่ถูกต้อง ซึ่งจะถูกสร้างในตอนนี้ก่อนที่จะเรียนรู้กฎ อีกทั้ง attribute files ยังถูกใช้สำหรับการกรองในการแบ่งประเภทข้อความ

3.3.5 การแยกประเภทข้อความ

เพื่อที่จะแบ่งประเภทของข้อความในจดหมายที่เข้ามา จำเป็นต้องแบ่งแยกคุณลักษณะ ซึ่งจะนำไปใช้ในฐานความรู้ แต่ละชุดของคำที่ปรากฏสูงสุด จะถูกเลือกมาตรวจสอบว่า อยู่ใน attribute file ที่ถูกต้องหรือไม่ ถ้าไม่ปรากฏ ก็จะแทนที่ด้วยคำว่า UNKNOWN ดังแสดงในรูปที่ 3.8 เหตุผลที่เป็นเช่นนี้เพราะ classification engine ยอมรับสัญลักษณ์นี้ และไม่นำมาปฏิบัติงาน ถ้าไม่พบคุณลักษณะนั้น ใน attribute file แสดงว่าคุณลักษณะนั้นไม่อยู่ในฐานความรู้

ตัวอย่างทดสอบ (testing examples) ถูกสร้างขึ้น เพื่อทดสอบความสอดคล้องกับฐานความรู้ กฎทุกข้อจะถูกประมวลผลเพื่อนับจำนวนครั้งของกฎที่ถูกใช้ สำหรับแต่ละ action ที่แตกต่างกัน สำหรับแต่ละ action ถ้าจำนวนของกฎที่นำมาใช้มากกว่า trust threshold เอเจนท์จะเสนอ action สำหรับข้อความที่เหมาะสมกัน โดยเพิ่มมันลงไปต่อท้าย execution log



รูปที่ 3.8 classification engine

บทที่ 4

การออกแบบและพัฒนาโปรแกรมเอเจนท์กรองไปรษณีย์

4.1 ขอบเขตของการออกแบบเอเจนท์กรองไปรษณีย์

ออกแบบเอเจน์กรองไปรษณีย์ ที่สามารถกรองเอาอีเมลที่ผู้ใช้ต้องการจริง ๆ ให้ได้มากที่สุดที่เป็นไปได้ ซึ่งเหมาะสมกับเหตุผลตามความต้องการของผู้ใช้ นั่นคือ ต้องพยายามคัดเอาอีเมลซึ่งไม่อยู่ในความสนใจของผู้ใช้ออกให้ได้มากที่สุดเท่าที่ทำได้ รวมทั้งแยกประเภทของอีเมลที่รับเข้ามาเพื่อจัดเก็บในกล่องไปรษณีย์ที่แตกต่างกัน ซึ่งสอดคล้องกับความต้องการของผู้ใช้แต่ละคน เพื่อความสะดวก รวดเร็วในการเข้าถึง และใช้งานอีเมล

เนื่องจากความต้องการของผู้ใช้แต่ละคนไม่เหมือนกัน และมีโอกาสที่จะเปลี่ยนแปลงบ่อย ดังนั้น เพื่อให้ระบบการกรองมีความเป็นส่วนตัวสูง ตรงตามความต้องการของผู้ใช้แต่ละคน จึงออกแบบเอเจนท์กรองไปรษณีย์ที่เรียนรู้จากการรับคำสั่งจากผู้ใช้โดยตรง ได้แก่ การกำหนดกฎ และเงื่อนไขในการกรองของผู้ใช้ ทำให้เมื่อความต้องการของผู้ใช้เปลี่ยนแปลง เอเจนท์ก็จะสามารถปรับปรุงตัวเองให้สอดคล้องได้ทันที

รวมทั้ง ออกแบบเอเจนท์กรองไปรษณีย์ให้สามารถค้นหาคำ หรือข้อความในอีเมลในกล่องไปรษณีย์ได้อีกด้วย ช่วยให้ประหยัดเวลาในการค้นหา ผู้ใช้สามารถเข้าถึง และจัดการอีเมลของตนได้รวดเร็วยิ่งขึ้น

4.2 การออกแบบเอเจนท์กรองไปรษณีย์

การออกแบบเอเจนท์กรองไปรษณีย์ ได้ออกแบบ โดยแบ่งรูปแบบการกรองออกเป็น 2 ส่วน ดังนี้

4.2.1 การกรองอีเมลที่รับมาจากเมล์เซิร์ฟเวอร์

พิจารณาอีเมลที่รับมาจากเมล์เซิร์ฟเวอร์ ว่าเป็นอีเมลที่ผู้รับต้องการหรือไม่ โดยจะยังไม่เก็บลงฐานข้อมูลอีเมลของผู้ใช้ในทันที ต้องเปรียบเทียบกับเงื่อนไขที่ผู้ใช้กำหนดไว้ก่อน ซึ่งเงื่อนไขนั้นผู้ใช้สามารถกำหนดได้จากส่วนหัวของอีเมล และเนื้อหาของรายละเอียดในอีเมลที่ไม่ต้องการ ได้แก่

- ส่วนหัวของอีเมลที่เป็นอีเมลผู้ส่ง(email or domain) เช่น nong@hotmail.com หรือ
- ส่วนหัวของอีเมลที่เป็นชื่อผู้รับ(To) เช่น mai@hotmail.com

- ส่วนหัวของอีเมลที่เป็นชื่อผู้รับสำเนา (Cc) เช่น mai@hotmail.com
- ส่วนหัวของอีเมลที่เป็นชื่อเรื่อง(Subject) เช่น free, naked, money เป็นต้น
- ส่วนของรายละเอียดในอีเมล (Content in body) เช่น !!!, only $, be over 20 เป็นต้น

ถ้าพิจารณาแล้วว่าเป็นอีเมลที่อยู่ในความต้องการของผู้ใช้ นั่นคือ ไม่ตรงกับเงื่อนไขที่กำหนด ก็จะเก็บอีเมลนั้นลงฐานข้อมูลอีเมลของผู้ใช้ เมื่อจัดเก็บเสร็จเรียบร้อยแล้ว ค่อยส่งคำสั่งไปบอกยังเมล์เซิร์ฟว่าจัดเก็บอีเมลลงฐานข้อมูลเรียบร้อยแล้ว ให้เมล์เซิร์ฟเวอร์ลบอีเมลนั้นออกได้ แต่ถ้าเป็นอีเมลที่ไม่ต้องการก็จะส่งคำสั่งไปบอกยังเมล์เซิร์ฟเวอร์ เพื่อให้ลบอีเมลนั้นทิ้งไป

4.2.2 การกรองเพื่อแยกประเภทของอีเมล

การกรองในส่วนนี้พิจารณาอีเมลในฐานข้อมูลอีเมลของผู้ใช้ เพื่อแยกประเภทของอีเมล และจัดเก็บในกล่องไปรษณีย์ที่แตกต่างกันก่อนที่จะนำเสนอแก่ผู้ใช้ โดยนำข้อมูลอีเมลที่จัดเก็บมาเปรียบเทียบกับเงื่อนไขที่ผู้ใช้กำหนดไว้ ซึ่งเงื่อนไขนั้นผู้ใช้สามารถกำหนดจากส่วนหัวของอีเมลและเนื้อหาของรายละเอียดใน
อีเมล ได้แก่

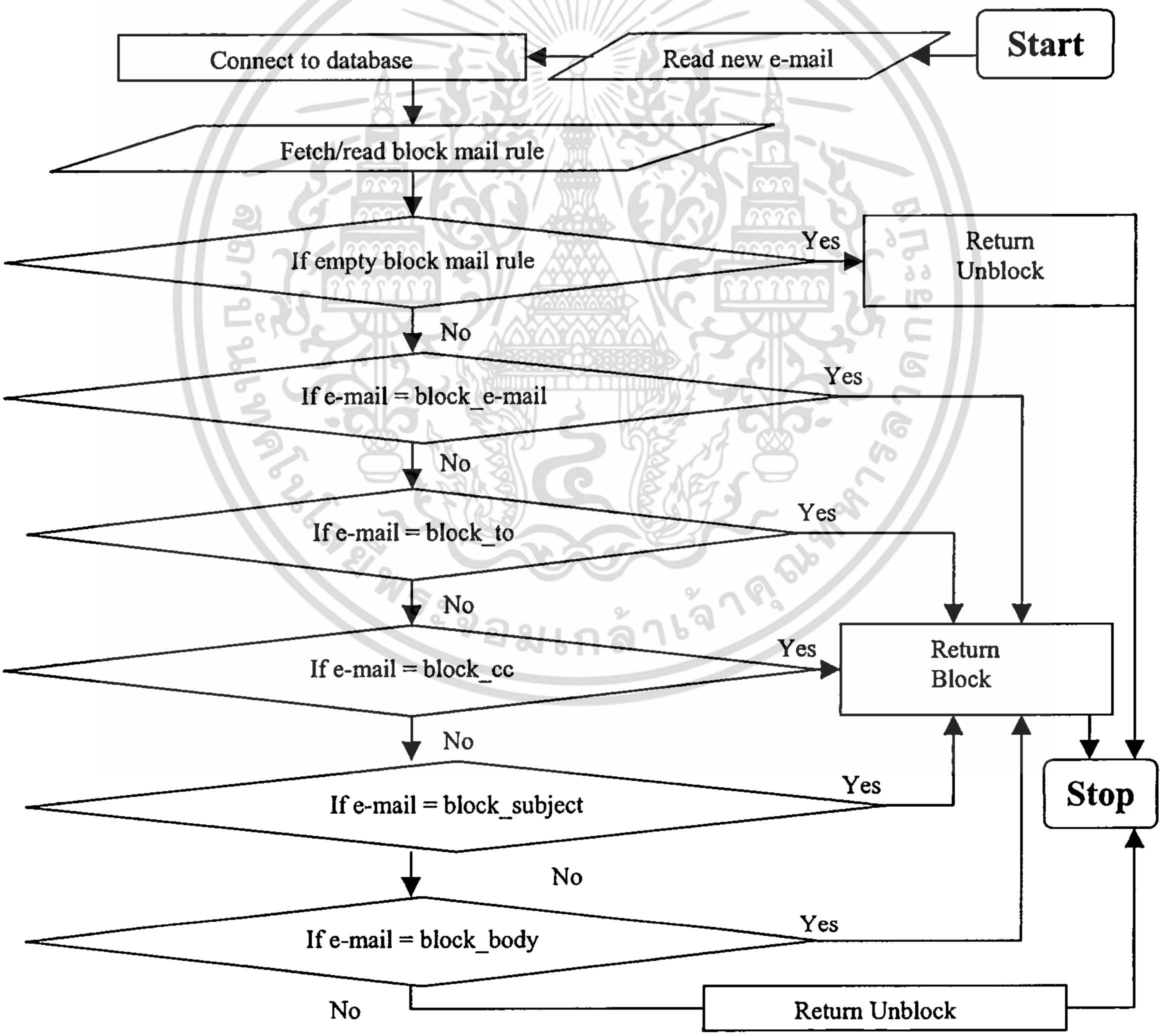
- ส่วนหัวของอีเมลที่เป็นอีเมลผู้ส่ง(From) เช่น kmitl11.it.ac.th
- ส่วนหัวของอีเมลที่เป็นอีเมลผู้รับ(To) เช่น iammai@mymail.com
- ส่วนหัวของอีเมลที่เป็นชื่อผู้รับสำเนา (Cc) เช่น ying@yahoo.com และจำนวนผู้รับสำเนาที่ต้องการ โดยสามารถกำหนดได้ว่าให้น้อยกว่าหรือ มากกว่าที่กำหนด
- ส่วนหัวของอีเมลที่เป็นชื่อเรื่อง(Subject) เช่น comprehensive, project เป็นต้น
- ส่วนของรายละเอียดในอีเมล (Content in body) เช่น รายละเอียดโครงการพัฒนาระบบงาน , หนังสือใหม่ห้องสมุด เป็นต้น ซึ่งผู้ใช้สามารถระบุเงื่อนไขที่คาดว่าจะปรากฏในรายละเอียดของอีเมลได้ 3 คำ สำหรับกฎ 1 ข้อ
- ขนาดของอีเมลที่น้อยกว่าหรือ มากกว่าจากที่กำหนด ซึ่งมีหน่วยเป็น kB
- เป็นอีเมลที่มีไฟล์แนบหรือไม่ ซึ่งสามารถกำหนดชนิด และขนาดของไฟล์แนบที่ต้องการ โดยสามารถกำหนดได้ว่าให้น้อยกว่าหรือ มากกว่าจากที่กำหนดได้

4.3 แนวคิดในการกรอง

การกรองไปรษณีย์นี้ใช้แนวคิดการกรองจากสิ่งที่เข้าใจ หรือรับรู้(Cognitive filtering) คือการกรองจากลักษณะเฉพาะของข้อความ ได้แก่ ส่วนหัว และเนื้อหาของอีเมล นั่นคือ From, to, cc, subject, body, attachment ของอีเมลที่รับมา ซึ่งมีส่วนของโปรแกรมที่ทำหน้าที่ค้นหาคำสำคัญ

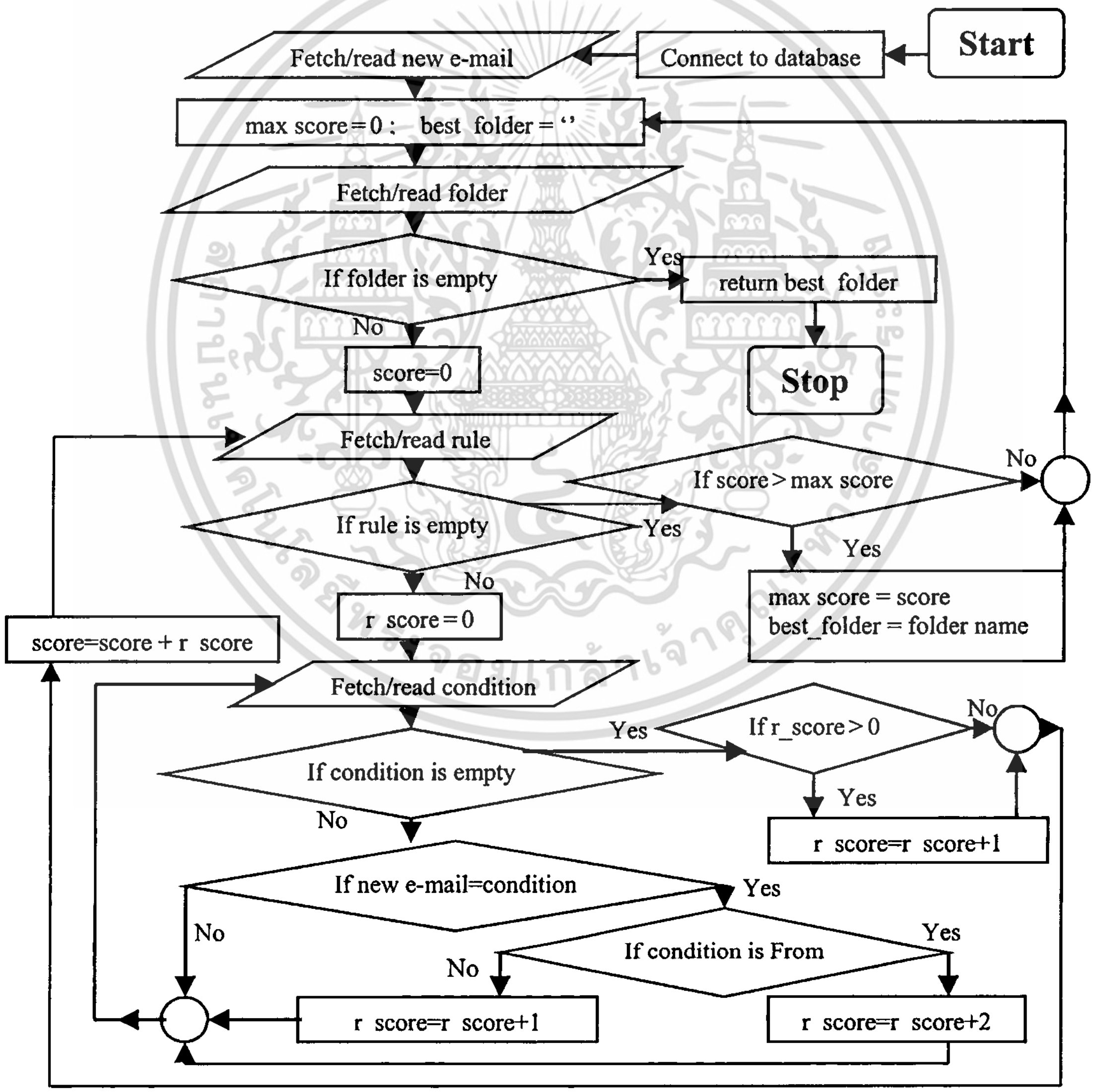
(keyword) หรือส่วนหนึ่งของข้อความ (วลี) ที่จะนำมาแบ่งประเภทของอีเมล และ Economic filtering ซึ่งการกรองด้วยวิธีการนี้ขึ้นอยู่กับการประเมินค่าใช้จ่ายในการจัดการอีเมล์ เช่น ความยาวอีเมล์, ขนาดของอีเมล์ เป็นต้น

วิธีการกรองอีเมลที่รับมาจากเมล์เซิร์ฟเวอร์ จะพิจารณาส่วนหัวของอีเมล หรือเนื้อหาของรายละเอียดในอีเมลที่แยกแยะรายละเอียดแล้ว ซึ่งถูกนำมาใช้เป็นคำสำคัญของอีเมล ว่ามีส่วนใดบ้างตรงกับเงื่อนไขที่กำหนด ถ้ามีคำสำคัญใดตรงกับเงื่อนไขที่กำหนดอย่างน้อย 1 ข้อ ก็จะถือว่าอีเมลนั้นเป็นอีเมลที่ถูกผู้รับปฏิเสธ ซึ่งผังงานในการกรองอีเมลที่รับมาจากเมล์เซิร์ฟเวอร์ แสดงในรูปที่ 4.1



รูปที่ 4.1 ผังงานในการกรองอีเมลที่รับมาจากเมล์เซิร์ฟเวอร์

ส่วนวิธีการกรองเพื่อแยกประเภทของอีเมล และนำไปจัดเก็บในกล่องไปรษณีย์ที่เหมาะสมนั้นกระทำโดย นำข้อมูลอีเมลที่ถูกแยกแยะรายละเอียดเรียบร้อยแล้ว ซึ่งจัดเก็บอยู่ในฐานข้อมูล ซึ่งถูกนำมาใช้เป็นคำสำคัญ หรือคำดรรชนีของอีเมล มาเปรียบเทียบกับเงื่อนไขที่กำหนดไว้ โดยคำสำคัญทุกตัวในอีเมล ที่ถูกใช้ในการอธิบายเนื้อหาของเอกสารมีค่าคะแนนไม่เท่ากัน ได้แก่ ส่วนของอีเมลผู้ส่งมีค่าน้ำหนัก 2 คะแนนและคำสำคัญอื่น ๆ จะมีค่าน้ำหนัก 1 คะแนน นั่นคือ ถ้าคำสำคัญตัวใดตรงกับเงื่อนไขที่กำหนดจะได้ score เพิ่มขึ้นตามเงื่อนไขนั้น ผังงานในการกรอง และพิจารณากล่องไปรษณีย์ที่เหมาะสมแสดงในรูปที่ 4.2

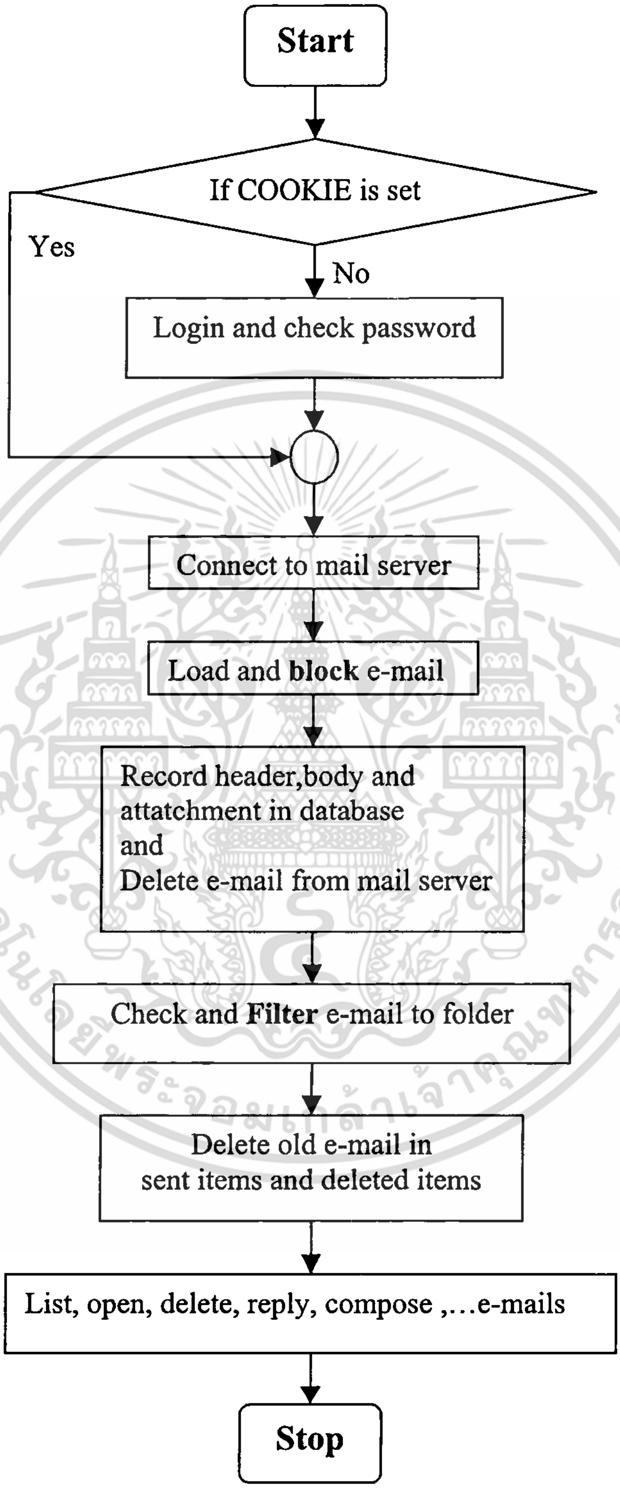


รูปที่ 4.2 ผังงานในการกรอง และพิจารณากล่องไปรษณีย์ที่เหมาะสม

การกรองเพื่อแยกประเภทของอีเมล และนำไปจัดเก็บในกล่องไปรษณีย์ที่ต้องการนั้นมีขั้นตอนในการกรอง ดังนี้

1. ผู้ใช้สร้างกล่องไปรษณีย์ กำหนดกฎ และเงื่อนไขสำหรับกฎนั้น ซึ่งกล่องไปรษณีย์หนึ่งตัว สามารถมีกฎได้หลายข้อ และกฎในแต่ละข้อ ก็สามารถกำหนดเงื่อนไขที่แตกต่างกัน ตามที่ผู้ใช้ต้องการ
2. เปรียบเทียบคำสำคัญกับเงื่อนไขที่กำหนด ถ้าเงื่อนไขข้อใดเป็นจริง ก็จะได้รับค่าน้ำหนักเพิ่มขึ้นตามค่าความสำคัญของแต่คำ (เริ่มต้นค่าน้ำหนักเท่ากับ 0)
3. เมื่อเปรียบเทียบเงื่อนไขของกฎข้อนั้นนั้นครบแล้ว ถ้าพบว่ามีค่าน้ำหนักมากกว่า 0 จะพิจารณาให้กฎข้อนั้นได้รับคะแนนเพิ่มขึ้นอีก 1 คะแนน เพื่อเป็นการให้คะแนนค่าความเชื่อมั่นกับกฎข้อนั้น
4. กระทำซ้ำข้อ 2 จนครบกฎทุกข้อของกล่องไปรษณีย์นั้น และกระทำกับทุกกล่องไปรษณีย์ ดังนั้น กล่องไปรษณีย์ที่มีเงื่อนไขตรงกับคำสำคัญมากที่สุด ก็จะเป็นกล่องไปรษณีย์ที่มีค่าหนักมากที่สุด
5. เมื่อได้ค่าน้ำหนักของกล่องไปรษณีย์ครบทุกกล่องแล้ว ก็จะพิจารณากล่องไปรษณีย์ที่เหมาะสมสำหรับอีเมลนั้น โดยเลือกกล่องอีเมลที่มีค่าน้ำหนักมากที่สุด อีเมลก็จะถูกพิจารณาให้ตกในกล่องไปรษณีย์นั้น ถ้าเกิดมีกล่องไปรษณีย์ที่มีค่าน้ำหนักเท่ากัน ก็จะพิจารณาให้อีเมลตกในกล่องแรกที่พบว่ามีคะแนนเท่ากัน และถ้ากล่องไปรษณีย์ทุกกล่องที่ผู้ใช้สร้างขึ้นมีค่าน้ำหนักเท่ากับ 0 อีเมลก็จะตกอยู่ในกล่อง inbox ตามเดิม

เอเจนท์กรองไปรษณีย์ เป็นเอเจนท์ที่ทำงานอยู่ระหว่างผู้ใช้ และเมล์เซิร์ฟเวอร์ ทำให้เอเจนท์สามารถจัดการอีเมลก่อนที่จะจัดส่งไปยังผู้ใช้ได้ ซึ่งแผนภาพการทำงานโดยรวมของเอเจนท์กรองไปรษณีย์มีการทำงานตามลำดับขั้นตอนดังรูปที่ 4.3



รูปที่ 4.3 แผนภาพการทำงานโดยรวมของเอเจนท์กรองไปรษณีย์

4.4 ฟังก์ชั่นเกี่ยวกับ IMAP

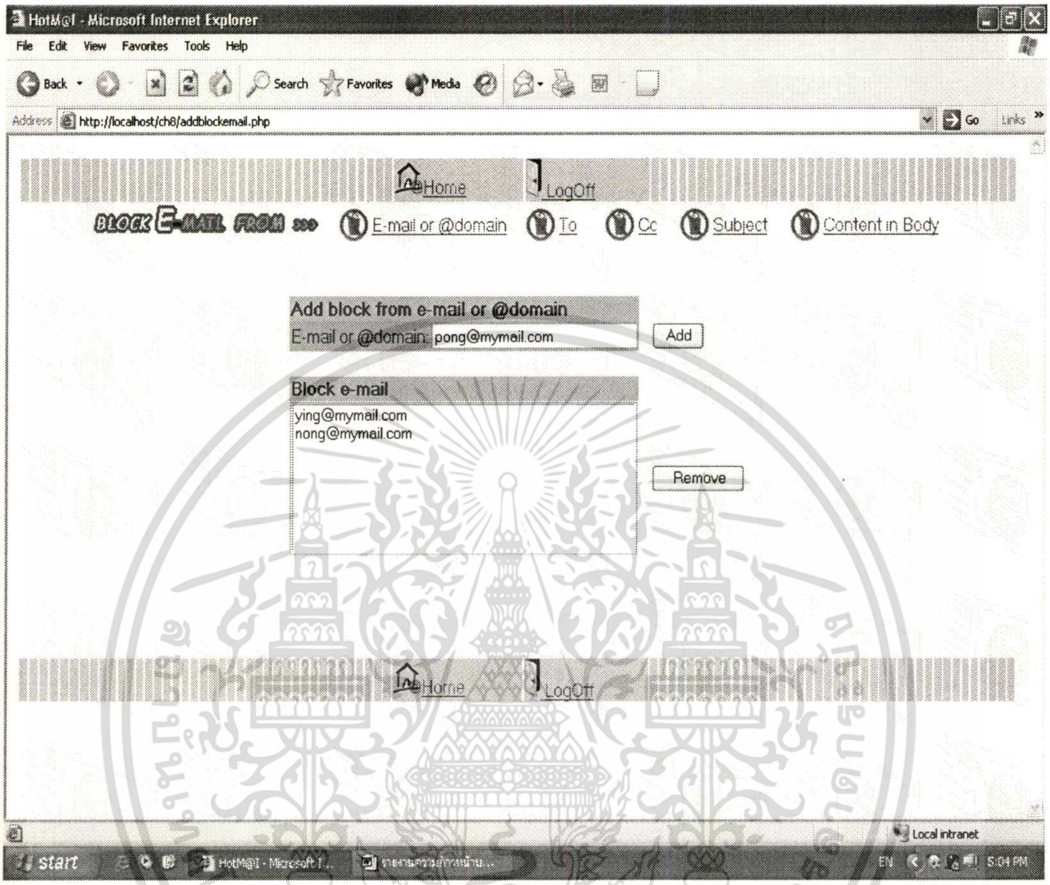
โปรแกรมรับ-ส่งอีเมลที่พัฒนาขึ้นนี้ เชื่อมต่อกับเมล์เซิร์ฟเวอร์ โดยการใช้ฟังก์ชัน IMAP ของ PHP เพื่อเปิดอีเมล POP3

ฟังก์ชั่นเกี่ยวกับ IMAP ส่วนใหญ่เริ่มมีตั้งแต่ PHP3 เป็นต้นมา บางฟังก์ชั่นรองรับ POP3 เซิร์ฟเวอร์ บางฟังก์ชั่นรองรับเฉพาะ IMAP เซิร์ฟเวอร์ ซึ่งในการพัฒนานี้ จะใช้ฟังก์ชั่น IMAP ที่ใช้กับ POP3 เท่านั้น ดังนี้

- imap_open เชื่อมต่อกับเมล์เซิร์ฟเวอร์ (เปิด mailbox เตรียมไว้)
- imap_last_error แสดง error message ล่าสุด (ถ้ามี)
- imap_close ยุติการเชื่อมต่อ (ปิด mailbox)
- imap_mailboxmsginfo อ่านข้อมูลของ mailbox เช่น จำนวนอีเมล, ขนาด inbox
- imap_num_msg นับจำนวนอีเมลในตู้ mailbox
- imap_headerinfo อ่านข้อมูลส่วนหัวของอีเมล
- imap_headers อ่านส่วนหัวของอีเมลทั้งหมดใน mailbox
- imap_fetchheader อ่านข้อมูลส่วนหัวของอีเมล
- imap_mime_header_decode ถอดรหัส MIME header เช่น เมล์ที่ส่งมาจาก Outlook
- imap_fetchstructure อ่านโครงสร้างของอีเมล
- imap_body อ่านเนื้อจดหมายอีเมล (message body)
- imap_fetchbody อ่านข้อมูลเนื้อจดหมายอีเมล ส่วนที่ต้องการ เช่น เนื้อหาข้อความ, ส่วนของไฟล์แนบ
- imap_qprint ถอดรหัสเปลี่ยนข้อความจดหมายให้อ่านได้ เช่น เมล์ที่ส่งมาจาก Outlook
- imap_delete ทำสัญลักษณ์อีเมลที่ต้องการลบ
- imap_expunge ลบอีเมลที่ทำสัญลักษณ์ไว้

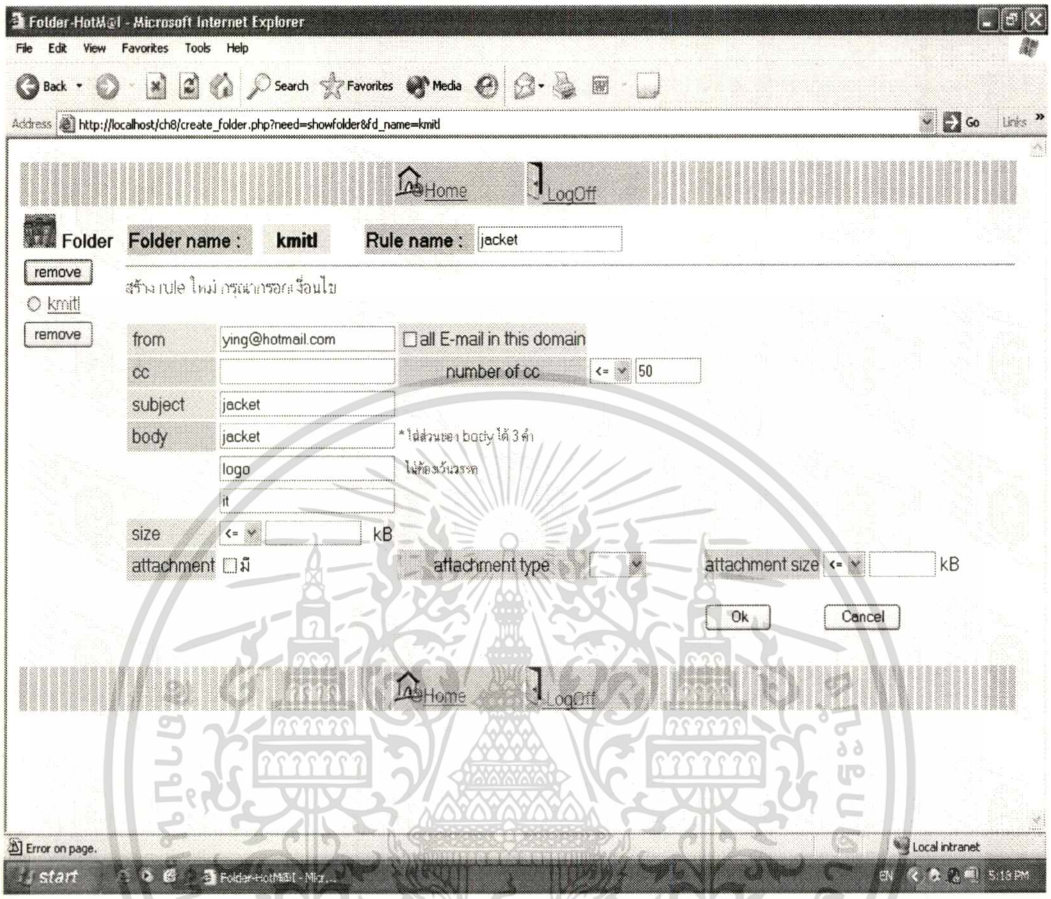
4.5 หน้าจอสำหรับรับข้อมูลจากผู้ใช้

ข้อมูลที่รับมาจากผู้ใช้ในที่นี้หมายถึง กฎในการกรองอีเมล ซึ่งผู้ใช้ต้องกรอกกฎในการกรองอีเมล 2 ส่วน ดังนั้น หน้าจอในการรับกฎจากผู้ใช้ จึงออกแบบเป็น 2 ส่วนเช่นกัน ซึ่งแบ่งเป็นหน้าจอสำหรับรับกฎ เพื่อกรองอีเมลที่รับมาจากเซิร์ฟเวอร์ และพิจารณาว่าเป็นอีเมลที่ถูกปฏิเสธหรือไม่ ตัวอย่างหน้าจอในการรับกฎที่กล่าวมาข้างต้น แสดงดังรูปที่ 4.4

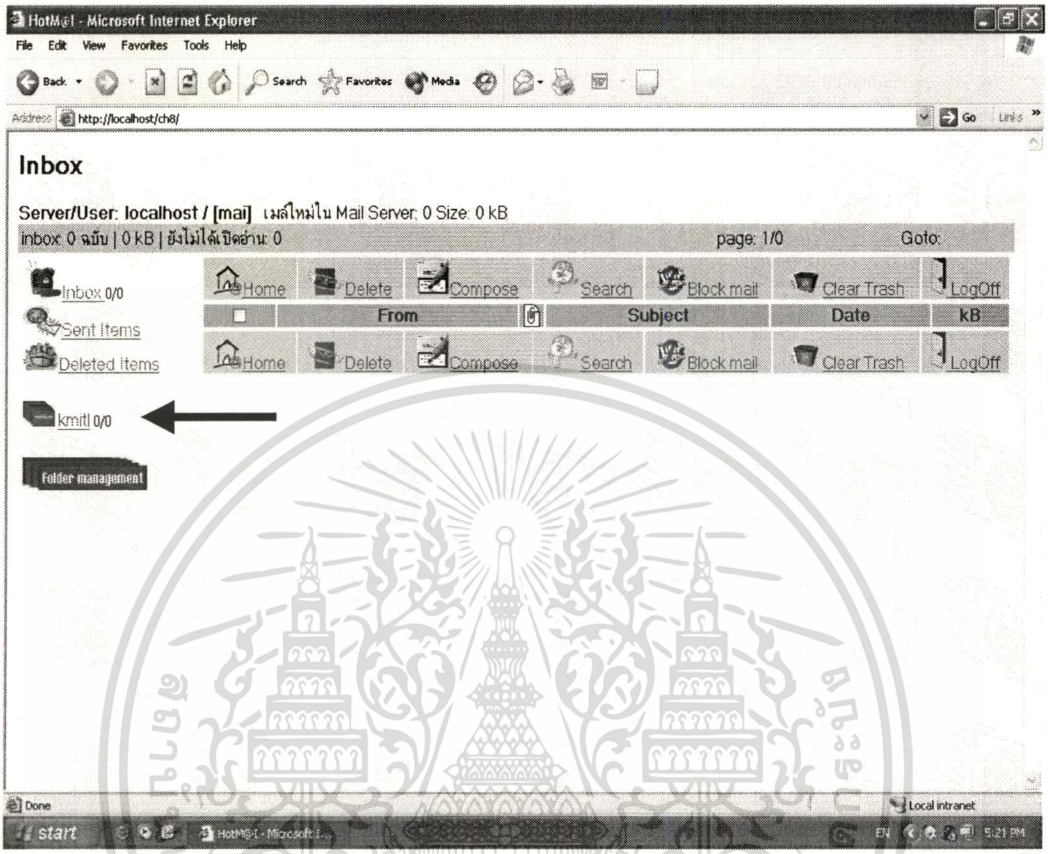


รูปที่ 4.4 ตัวอย่างหน้าจอการรับกฎจากผู้ใช้เพื่อกรองจากอีเมลผู้ส่ง

ส่วนการกรองเพื่อแยกประเภทของอีเมล และจัดเก็บในกล่องไปรษณีย์ที่เหมาะสม ตัวอย่างหน้าจอในการรับกฎจากผู้ใช้แสดงดังรูปที่ 4.5 ซึ่งด้านซ้ายมือจะแสดงชื่อกล่องไปรษณีย์ และชื่อกฎ ซึ่งผู้ใช้สามารถเพิ่ม ลดกล่องไปรษณีย์ และกฎ และสามารถคลิกเพื่อดูรายละเอียดของกฎได้ ส่วนด้านขวาจะแสดงรายละเอียดของกฎ เมื่อสร้างกล่องไปรษณีย์เสร็จเรียบร้อยแล้ว จะปรากฏชื่อของกล่องไปรษณีย์ที่เราสร้างขึ้นที่หน้าแรกของโปรแกรมดังรูปที่ 4.6



รูปที่ 4.5 ตัวอย่างหน้าจอการรับกฎและเงื่อนไขจากผู้ใช้ เพื่อจัดเก็บในกล่องไปรษณีย์ที่เหมาะสม



รูปที่ 4.6 แสดงชื่อกล่องไปรษณีย์ที่ผู้ใช้สร้างขึ้น

บทที่ 5

สรุปผลการพัฒนาและข้อเสนอแนะ

จากการศึกษา การวิเคราะห์ และออกแบบระบบเอเจนท์กรองไปรษณีย์ ที่ติดต่อกับเมลเซิร์ฟเวอร์จำลอง ผ่านทางโพรโทคอล POP3 โดยใช้ฟังก์ชันของ IMAP และโพรโทคอล SMTP โดยใช้ภาษา PHP และ JavaScript ในการพัฒนา ภายใต้สภาพแวดล้อมของ WindowsXP และทดสอบการรับอีเมลจากโปรแกรม Outlook Express

5.1 สรุปผลการพัฒนาเอเจนท์กรองไปรษณีย์

จากการศึกษา ออกแบบ และพัฒนาระบบเอเจนท์กรองไปรษณีย์ สามารถสรุปผลการศึกษาได้ดังนี้

1. เอเจนท์สามารถกรองอีเมลที่ไม่อยู่ในความต้องการของผู้ใช้ออกไป และสามารถแยกประเภทของอีเมลได้ตามเงื่อนไขที่ผู้ใช้กำหนด โดยเฉพาะอย่างยิ่ง ถ้าผู้ใช้กรอกเงื่อนไขได้ละเอียด ครอบคลุม และทราบข้อมูลของอีเมล์ที่จะรับล่วงหน้า ก็จะทำให้เอเจนท์นั้นกรองได้ถูกต้องแม่นยำยิ่งขึ้น
2. ในการพัฒนาอินเทลลิเจนท์เอเจนท์ ส่วนที่ยาก และ ซับซ้อนที่สุด คือ การสร้างฐานความรู้ การเรียนรู้ การแสดงความรู้ และการนำความรู้นั้นมาใช้ให้เกิดประโยชน์สูงสุด และยังต้องคำนึงถึง สถาปัตยกรรม และภาษาที่ใช้ด้วยเสมอ โดยเลือกให้เหมาะสมกับระบบที่พัฒนามากที่สุด
3. การพิจารณากล่องไปรษณีย์ที่เหมาะสม ในกรณีที่มีกล่องไปรษณีย์ตั้งแต่ 2 กล่องขึ้นไป มีค่าน้ำหนักเท่ากัน ในกรณีนี้ เอเจนท์จะเลือกกล่องแรกที่พบ ทำให้บางครั้งไม่ตรงตามความต้องการของผู้ใช้
4. จากการทดสอบการทำงานของโปรแกรมเอเจนท์กรองไปรษณีย์ พบว่า สามารถทำงานได้ตามขอบเขต และวัตถุประสงค์ที่วางไว้

5.2 ข้อจำกัดและข้อเสนอแนะ

จากการศึกษา ออกแบบ และพัฒนาระบบเอเจนท์กรองไปรษณีย์ สามารถสรุปข้อจำกัดและข้อเสนอแนะของระบบได้ดังนี้

1. การพัฒนาระบบเอเจนท์กรองไปรษณีย์ บนเมลเซิร์ฟเวอร์จำลอง มีข้อจำกัดหลายอย่าง เช่น ขนาดของอีเมลที่เซิร์ฟเวอร์จำลองสามารถรับได้มีขนาดจำกัด เมื่อเปรียบเทียบกับเมลเซิร์ฟเวอร์ที่ใช้งานจริง และความสามารถของเมลเซิร์ฟเวอร์เองก็ไม่หลากหลาย ทำให้ความสามารถ และประสิทธิภาพการทำงานของเอเจนทถูกจำกัดโดยประสิทธิภาพของเมลเซิร์ฟเวอร์จำลอง และการทำงานไม่ได้เชื่อมต่อกับอินเตอร์เนตจริง ๆ ทำให้ปัญหาบางอย่างไม่ได้ถูกนำมาพิจารณา ซึ่งอาจเกิดปัญหาขึ้นได้ในการใช้งานจริง เช่น การส่งอีเมลที่มีขนาดใหญ่
2. การพัฒนาเอเจนท์ในส่วนของอัลกอริทึมนั้น สามารถที่นำเอาหลักการการทำงานในลักษณะของปัญญาประดิษฐ์ (Artificial Intelligent) มาประยุกต์ใช้ ในการพัฒนา เพื่อทำให้เอเจนท์มีความสามารถในการสร้างฐานความรู้ การเรียนรู้ การแสดงความรู้ และการนำความรู้นั้นมาใช้ได้ด้วยตัวเอง ซึ่งเป็นส่วนที่ช่วยให้เอเจนท์ที่พัฒนามีความฉลาดมากขึ้น ตัวอย่างของหลักการที่สามารถนำมาประยุกต์ใช้ในการพัฒนา เช่น Fuzzy Logic, Neural Network เป็นต้น
3. สามารถเพิ่มเงื่อนไขในการกรองให้มีความละเอียด คลอบคลุมมากขึ้น และพิจารณาให้ตรงกับความต้องการของผู้ใช้ เช่น เพิ่มเงื่อนไขการกรองจากเวลาที่มีอีเมลเข้ามา
4. ความสามารถของเอเจนท์กรองไปรษณีย์ยังสามารถขยายให้เพิ่มได้อีกมาก เช่น การพัฒนาด้วยภาษาที่สนับสนุนการทำเอเจนท์ ได้แก่ Java , Visual C++ เป็นต้น
5. พัฒนาเอเจนท์กรองไปรษณีย์ ที่สามารถเรียนรู้จากการกระทำของผู้ใช้ เช่น เก็บสถิติอีเมลที่ผู้ใช้ไม่เคยเปิดอ่าน และถูกลบทิ้งเสมอ ก็อาจพิจารณาได้ว่า เป็นอีเมลขยะ เพื่อทำให้การทำงานของเอเจนท์เป็นอัตโนมัติ และเป็นอิสระจากผู้ใช้
6. สร้างกลไกการคำนวณค่าน้ำหนัก และค่าความเชื่อมั่นของอีเมล เพื่อกรองได้ถูกต้อง แม่นยำยิ่งขึ้น
7. ในปัจจุบัน มีอีเมลขยะ หรืออีเมลที่ไม่อยู่ในความต้องการของผู้ใช้เพิ่มขึ้นเป็นจำนวนมาก อีกทั้ง ผู้ส่งอีเมลเหล่านี้ยังพัฒนารูปแบบ และวิธีการส่งอีเมลอยู่ตลอดเวลา ดังนั้น เอเจนท์กรองไปรษณีย์ก็จะต้องมีการปรับปรุง และพัฒนารูปแบบในการกรองอยู่ตลอดเวลาเช่นกัน เพื่อที่จะสามารถกรองได้ถูกต้อง นำไปใช้ได้จริง และได้รับการยอมรับจากผู้ใช้
8. เพิ่มการรักษาความปลอดภัยให้กับผู้ใช้ เช่น scan virus
9. เพิ่มการทำงานของเอเจนท์เพื่อความสะดวกสบายแก่ผู้ใช้ เช่น address book

บรรณานุกรม

กอบเกียรติ สระอุบล. 2545ก. **การสร้างเวบเพจฉบับประยุกต์ เล่ม 1.** กรุงเทพ : บี อี แอนด์ ซี.

กอบเกียรติ สระอุบล . 2545ข. **การสร้างเวบเพจฉบับประยุกต์ เล่ม 2.** กรุงเทพ : บี อี แอนด์ ซี.

ยุทธนา ลีลาศวัฒนกุล. 2544. **คู่มือการเขียนและใช้งาน Visual C++ ฉบับโปรแกรมเมอร์.** กรุงเทพ : อินโฟเพรส

Bigus, Joseph P. and Bigus, Jennifer. 1998. **Constructing Intelligent Agents with Java.** New York, NY: John Wiley & Sons.

Chew, Siew-Kho. 2002. **Email Filtering Tool.** [online]. Available : http://www.dcs.shef.ac.uk/teaching/eproj/ug2002/pdf/u1sc.pdf

GROUP Technologies AG. 2001. **Rule-based E-mail Content Filtering SecuriQ Wall.** [online]. Available : http://www.group-technologies.com/en/home.nsf/id/F71C14D9C31B3F09C12592004188E9/$File/Whitepaper_securiQ_Wall-en.pdf.

Itskevitch, Julia. 2001. **Automatic Hierarchical E-mail Classification Using Association Rules.** [online]. Available: http://citeseer.nj.nec.com/cache/papers/cs/23077/http:zSzzSzfas.sfu.cazSzpubzSzcszSzthesesSzSz2001zSzJuliaItskevitchMSc.pdf/itskevitch01automatic.pdf

Lesnick, Lesnick L. and Moore, Ralph E. 1997. **Creation Cool Intelligent Agents for the Net.** Foster City, CA : IDG Book World.

Murch, Richard and Johnson, Tony. 1999. **Intelligent Software Agents.** Upper Saddle River, New Jersey: Prentice Hall.

Payne, Terry. 1994. **Learning Email Filtering Rules with Magi a Mail Agent Interface.** [online]. Available: http://www-2.cs.cmu.edu/~terryp/Pubs/msc_thesis.pdf.

Wall, Matthew. 2001. **The Sieve Language and a General Model for Deliver and Interoperable Filtering in Internet Mail.** [online]. Available : http://www.cyrusoft.com/sieve/sieve.whitepaper.pdf.

ประวัติผู้เขียน

ชื่อ	**นางสาวธิคารัตน์ ตันทะสุวรรณ**
ภูมิลำเนา	**เลขที่ 23/3 หมู่ 2 ถนนเพชรเกษม ตำบลน้ำจืด อำเภอกระบุรี จังหวัดระนอง**
วุฒิการศึกษา	**วท.บ.(วิทยาการคอมพิวเตอร์) คณะวิทยาศาสตร์ มหาวิทยาลัยกรุงเทพ**
อีเมล	**iammai@hotmail.com**