การคัดแยกสแปมเมล์ โดยอัลกอริทึมเบย์เซียน

Spam Filtering using Bayesian Algorithm

--

โดย

ธนรัฐ โชติพันธ์

รหัสประจำตัว 46066706

อาจารย์ที่ปรึกษา

ผศ.ดร. จันทร์บูรณ์ สถิตวิริยวงศ์

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2548

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การคัดแยกสแปมเมล์ โดยอัลกอริทึมเบย์เซียน
นักศึกษา	นายธนรัฐ โชติพันธ์
อาจารย์ที่ปรึกษา	ผศ.ดร. จันทร์บูรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

กว่า 80% ของอีเมล์ที่รับในปัจจุบันเป็นอีเมล์ที่ผู้รับไม่พึงประสงค์ที่จะรับ(สแปม) สร้างปัญหาสำหรับผู้ใช้งานอีเมล์ต้องพิจารณาคัดแยกอีเมล์ที่ใช้งานจริงโดยการพิจารณาจากการอ่านอีเมล์ที่รับมาทั้งหมดทีละฉบับด้วยตนเองซึ่งเสียเวลามาก วิธีการคัดแยกส่วนมากพบว่าเกิดการคัดแยกที่ไม่ถูกต้องอยู่ในลักษณะ false positive เป็นส่วนใหญ่ การนำทฤษฎีความน่าจะเป็นของ Bayes มาใช้พิจารณาอีเมล์โดยอ้างจากข้อมูลอีเมล์ที่ทำการสอนให้ระบบรู้จำมาประมวลผลโดยอัลกอริทึม Bayesian ว่าอีเมล์นั้นมีความน่าจะเป็นสแปมมากน้อยเพียงใด เพื่อลดปัญหา false positive และช่วยลดเวลาในการพิจารณาสแปมของผู้ใช้ได้ดีขึ้น

Title	Spam Filtering using Bayesian Algorithm
Student	Mr. Tanarat Chotiphan
Advisor	Asst.Prof.Dr. Chanboon Sathitwiriyawong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2005

ABSTRACT

Above 80% of Electronic Mail (E-mail) income are spam mail, it make many problems to user and use long time to classification. All technique for filtering spam can't solve spam's problem in 100% and it make many false positive filtering. Beyesian email filters take advantage of Bayes' theorem to classify emails into categories. This project try to use Bayesian algorithm for solve false positive problem and reduce time of classification by user better.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานนี้สำเร็จลุล่วงได้ด้วยการได้รับความช่วยเหลือและความกรุณาจากบุคคลต่าง ๆ เหล่านี้

1. ขอขอบพระคุณ บิดา มารดา ที่ให้โอกาสในการศึกษาเล่าเรียนอย่างเต็มที่
2. ขอขอบพระคุณ ผศ.ดร. จันทร์บูรณ์ สถิตวิริยวงศ์ อาจารย์ที่ปรึกษา ที่ได้กรุณาให้คำปรึกษา แนะนำ สละเวลา ให้การดูแลเอาใจใส่ ช่วยเหลือ ชี้นำ และแก้ไขในสิ่งบกพร่องต่างๆ สำหรับโครงการพัฒนาระบบงานนี้เป็นอย่างมาก
3. ขอขอบพระคุณคณาจารย์ทุกท่านที่ให้ความรู้มากมาย เพื่อนำความรู้มาใช้ประกอบในโครงการพัฒนาระบบงานนี้
4. ขอบคุณเจ้าหน้าที่ และคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง ที่อำนวยความสะดวกในสถานที่ในการศึกษา ค้นคว้า และปฏิบัติงาน โดยสะดวก และครบถ้วน
5. ขอบคุณน้องสาวที่คอยทักท้วงให้ทำงานได้บรรลุตามเป้าหมาย อีกทั้งช่วยไขข้อข้องใจในเนื้อหาบางส่วนยิ่งขึ้น
6. ขอขอบคุณเพื่อนๆ ทุกคนที่ให้คำปรึกษา ให้กำลังใจและช่วยเหลือในโครงการพัฒนาระบบงานนี้

นายธนรัฐ โชติพันธ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

หน้า

ตารางที่

สารบัญรูป

หน้า

รูปที่

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การสื่อสารที่พบเห็นในปัจจุบันมีหลากหลายวิธี ปัจจุบันระบบอิเล็กทรอนิกส์เข้ามามีบทบาทในการสื่อสารเป็นอย่างมากเพราะช่วยทำให้การสื่อสารนั้นทำได้อย่างรวดเร็วยิ่งขึ้น สอดคล้องกับการดำรงชีวิตในปัจจุบันผู้ที่สามารถรับและส่งข่าวสารได้อย่างรวดเร็วมักเป็นผู้ได้เปรียบ การรับส่งจดหมายอิเล็กทรอนิกส์ (Email) เป็นหนึ่งในทางเลือกในการสื่อสารที่มีความรวดเร็วและเป็นที่นิยมใช้กันอย่างแพร่หลายเนื่องจากมีค่าใช้จ่ายต่ำเมื่อเปรียบเทียบกับระยะทางการสื่อสารระหว่างผู้รับและผู้ส่ง อีกทั้งผู้ส่งสามารถส่งจดหมายผู้ส่งไปยังผู้รับได้ในระยะเวลาไม่กี่นาที เมื่อวิธีในการสื่อสารชนิดนี้ได้รับความนิยมเพิ่มขึ้นจนกล่าวได้ว่าแทบทุกคนหรือองค์กรจะมีกล่องรับจดหมายอิเล็กทรอนิกส์ (Mailbox) เพื่อใช้ในการรับและติดต่อสื่อสารแลกเปลี่ยนข้อมูลระหว่างกันทั้งสิ้น

เมื่อมีผู้ใช้อีเมล์มากเพิ่มขึ้น อีเมล์ก็เป็นช่องทางหนึ่งในการแข่งขันกันทางธุรกิจที่สามารถเข้าถึงกลุ่มคนที่ใช้อีเมล์ซึ่งมีจำนวนมากโดยการส่งอีเมล์กระจายไปยังผู้ใช้อีเมล์จำนวนมากเพื่อหวังผลทางการค้านั้นๆ หรือเพื่อโจมตีคู่แข่ง โดยไม่ได้คำนึงว่าผู้รับมีความต้องการที่จะรับอีเมล์นั้นหรือไม่ เราเรียกการส่งอีเมล์ลักษณะนี้ว่าเป็น สแปมเมล์ (SPAM)

การส่งสแปมเมล์ไปยังผู้ใช้อีเมล์จำนวนหลายแสนคนหรือหลายล้านคนเพื่อหวังว่าจะมีผู้ใช้อีเมล์จำนวนหนึ่งเปิดอ่านและยอมซื้อหรือใช้บริการสินค้าหรือเชื่อถือเนื้อหาสาระในอีเมล์นั้นก็ถือว่าผู้ส่งสแปมเมล์ (Spammer) ประสบความสำเร็จแล้วเพราะผลที่ผู้ส่งสแปมเมล์จะได้รับนั้นมีความคุ้มค่ากว่าค่าใช้จ่ายที่เสียไปในการส่งสแปมเมล์ในแต่ละครั้ง จึงไม่เป็นที่น่าแปลกใจหากเราจะรู้สึกได้ว่ามีสแปมเมล์เพิ่มมากขึ้นในปัจจุบัน

การส่งสแปมเมล์ที่มีปริมาณมากขึ้นทุกวัน ส่งผลกระทบให้เกิดปัญหาโดยตรงต่อผู้ให้บริการเครือข่ายอินเทอร์เน็ตคือมีสแปมเมล์จำนวนมากบนเครือข่ายที่ตนให้บริการทำให้สูญเสียแบนด์วิดธ์ไปจำนวนหนึ่งนั่นหมายถึงผู้ให้บริการอินเทอร์เน็ตต้องมีค่าใช้จ่ายเพิ่มขึ้นเพื่อขยายแบนด์วิดธ์เพื่อรองรับการใช้งานอื่นๆ และส่งผลต่อผู้ใช้เป็นจำนวนมากเพราะกว่า 80% ของอีเมล์ที่ได้รับเป็นสแปมเมล์ที่ผู้รับไม่ได้รู้จักกับผู้ส่งและไม่ได้ต้องการรับอีเมล์ฉบับนั้น ผู้ใช้ต้องมาทำการแยกอีเมล์ที่ตนต้องการออกจากสแปมเมล์ที่มีจำนวนมากซึ่งทำให้เสียเวลามาก

1.2 วัตถุประสงค์

- ศึกษาการคัดแยกสแปม โดยการใช้ทฤษฎีของ Bayes
- สร้างโปรแกรมในส่วนคัดแยกสแปมที่ใช้อัลกอริทึม Bayesian
- ทดลองคัดแยกสแปมด้วยอัลกอริทึม Bayesian
- ประเมินการคัดแยกอัลกอริทึม Bayesian
- คาดว่าการทดลองสามารถนำไปสู่การคัดแยกสแปมเมล์ออกจากอีเมล์ โดยใช้อัลกอริทึม Bayesian ได้ในระดับที่น่าพอใจ รวมถึงลดภาระในการคัดแยกหรือพิจารณาสแปมเอง

1.3 แนวคิดที่ใช้ในการพัฒนาระบบ

เครื่องมือที่ใช้ในการกรอกสแปมเมล์ออกจากอีเมล์ที่ทำออกมาในปัจจุบัน ยังไม่มีเครื่องมือใดสามารถแยกออกสแปมเมล์ออกจากอีเมล์ได้ 100% เนื่องจากสแปมเมอร์เองก็มีการพัฒนาวิธีการส่งสแปมเมล์เพื่อหลีกเลี่ยงการดักจับเพื่อจะเข้าถึงกลุ่มเป้าหมาย เครื่องมือดังที่มีส่วนใหญ่มุ่งเน้นใช้วิธีการพื้นฐานในการแยกประเภทของสแปมเมล์ เช่น การใช้เทคนิคบัญชีดำ (Blacklist) และบัญชีขาว (Whitelist) ซึ่งเป็นวิธีที่ดีเพราะสามารถแยกอีเมล์ที่เข้าข่ายว่าเป็นสแปมเมล์ได้เป็นจำนวนมากระดับหนึ่งโดยพิจารณาจาก mail address ของผู้ส่งและผู้รับที่ระบุไว้ในเฮดเดอร์ของอีเมล์ แต่อย่างไรก็ดีก็ยังมีสแปมเมล์แบบใหม่ๆ ที่สามารถหลุดรอดจากการดักจับลักษณะนี้ได้ด้วยการปลอมแปลงเฮดเดอร์ของอีเมล์

การพิจารณาสแปมเมล์จาก mail address ของผู้รับผู้ส่งคงไม่เพียงพอ ผู้เขียนให้ความเห็นว่าเราควรพิจารณาเนื้อหาของอีเมล์ด้วยว่าเข้าข่ายว่าเป็นสแปมเมล์หรือไม่ น่าจะสามารถลดจำนวนสแปมเมล์ได้มากขึ้น โดยผู้เขียนเลือกใช้เทคนิคของ Bayes คือใช้อัลกอริทึมของ Bayesian เพื่อเปรียบเทียบคำในเนื้อหาของอีเมล์เพื่อนำมาหาความน่าจะเป็นว่าคำต่างๆที่ปรากฏในเอกสารมีน้ำหนักว่าน่าจะอยู่อีเมล์ปกติเท่าใด มีน้ำหนักว่าน่าจะอยู่ในสแปมเมล์เท่าใด แล้วพิจารณาตามอัลกอริทึมว่าน้ำหนักของคำต่างๆ ที่ปรากฏอยู่ในอีเมล์นั้นรวมกันแล้วมีความน่าจะเป็นสแปมมากน้อยเท่าไร โดยการได้มาซึ่งน้ำหนักของคำต่างๆเกิดจากการเตรียมข้อมูล (Preprocessing) ด้วยการนับจำนวนของคำต่างๆ ที่ปรากฏในอีเมล์ปกติ (HAM) และสแปม (SPAM) ที่นำมาใช้ในการทดลอง มาหาความค่าความน่าจะเป็นของคำแต่ละคำว่ามีความน่าจะเป็นที่คำนั้นๆ จะเป็นสแปมมากน้อยเพียงใด

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาการโครงสร้างอีเมล์

1.4.2 ศึกษาอัลกอริทึม Bayesian

1.4.3 ออกแบบระบบและทดลองการคัดแยก

1.4.4 วัดผลการทดลอง และปรับปรุง

1.4.5 จัดทำเอกสารเพื่อเสนอผลงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถลดปริมาณของสแปมเมล์ของผู้ใช้นั้นได้อย่างมีประสิทธิภาพ และสามารถลดเวลาในการคัดแยกสแปมเมล์ออกจากอีเมล์ของผู้รับได้ดียิ่งขึ้น

บทที่ 2
แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 ความรู้พื้นฐานเกี่ยวกับการทำงานของจดหมายอิเล็กทรอนิกส์

จดหมายอิเล็กทรอนิกส์ (Email) จัดเป็นโปรแกรมประยุกต์บนอินเทอร์เน็ตที่นิยมในการสื่อสารมีลักษณะคล้ายกับจดหมายทั่วไป มีความแตกต่างตรงที่ มีความรวดเร็วในการส่งไปถึงผู้รับ มีค่าใช้จ่ายต่ำ และมีความง่ายในการใช้งาน นอกจากส่งข้อความที่มีลักษณะในรูปแบบเหมือนจดหมายแล้ว อีเมล์ยังสามารถที่จะส่ง Link เอกสาร HTML รูปภาพ เสียง รวมทั้งภาพเคลื่อนไหว

ระบบ Email ประกอบด้วย 3 ส่วนหลักคือ User agents, Mail server และ Simple Mail Transfer Protocol (SMTP) โดย

- User agent ซึ่งบางครั้งเรียกว่า mail reader เป็นโปรแกรมที่ ผู้ใช้ (user) ทำการอ่านและตอบจดหมายได้
- Mail server จะใช้ในการเก็บและจัดการกับตู้รับจดหมายอิเล็กทรอนิกส์ (mail box) ของ user แต่ละคน โดย Mailbox ทำหน้าที่ในการจัดการและเก็บรักษาจดหมายอิเล็กทรอนิกส์ (ในที่นี้จะใช้คำว่า message แทน) ที่ส่งมาถึงผู้รับ
- Simple Transfer Protocol (SMTP) ซึ่งเป็นโพรโทคอลที่ทำหน้าที่ส่งต่ออีเมล์จากระหว่างเครื่องต่าง ทำงานบนชั้น Application Layer

บนอินเทอร์เน็ตนั้น Email จะใช้บริการของ TCP ในการรับส่งเมล์จาก mail server ของผู้ส่งไปยัง mail server ของผู้รับโดยใช้เรียกใช้โพรโทคอล SMTP ซึ่ง mail server ที่ทำการส่งเมล์ ไปยัง mail server อื่นจะทำหน้าที่เป็น SMTP client และเมื่อ mail server ได้รับเมล์ จาก mail server อื่นก็จะทำหน้าที่เป็น SMTP server

2.1.1 SMTP

SMTP เป็นส่วนสำคัญในการใช้งาน Email โดยจะทำหน้าที่ส่ง message จาก mail server ฝั่งผู้ส่งไปยัง mail server ฝั่งผู้รับ โดยขั้นแรก SMTP Client จะทำการสร้าง TCP connection บน port 25 กับ SMTP server หลังจากนั้น SMTP client จะทำการบอก email address ของผู้ส่งและผู้รับต่อ SMTP server เมื่อ SMTP client และ server เมื่อดำเนินการเสร็จแล้ว SMTP Client จะทำ

การส่ง message ไป SMTP server โดย SMTP จะใช้บริการของ TCP เพื่อส่ง message ไปยัง server และถ้ามีการส่ง message อื่นๆ ไปยัง server อีก client ก็จะทำกระบวนการนี้ซ้ำใน TCP connection เดียวกันนี้ แต่ถ้าไม่มีการส่ง message อื่น จะทำการสั่งให้ TCP ทำการปิด connection ลักษณะการส่ง Email

จากรูปที่ 2.1 ต่อไปนี้ กำหนดให้ Host ของ client คือ crepes.fr และ Host ของ server คือ HAMburger.edu

```
S : 220 hamburger.edu
C : HELO crepes.fr
S : 250 Hello crepes.fr, pleased to meet you
C : MAIL FROM : alice@crepes.fr
S : 250 alice@crepes.fr...Sender OK
C : RCPT TO : bob@hamburger.edu
S : 250 bob@hamburger.edu ... Recipent OK
C : DATA
S : 354 Enter mail, end with "." On a line by itself
C : Do you like ketchup?
C : How about pickles?
C : .
S : 250 Message accepted for delivery
C : QUIT
S : 221 hamburger.edu closing connection
```

รูปที่ 2.1 แสดงตัวอย่างการส่งต่ออีเมล์จาก SMTP client ไปยัง SMTP server

จากตัวอย่าง client (C:) ส่ง message ("Do you like ketchup? How about pickles?") จาก mail server crespes.fr ไปยัง mail server (S:) HAMburger.edu client จะส่งคำสั่ง 5 คำสั่งคือ HELO, MAIL FROM, RCPT TO, DATA และ QUIT ไปยัง server และจะตอบกลับแต่ละคำส่งด้วย reply code และคำอธิบาย ในกรณีผู้ส่งต้องการส่งตั้งแต่ 2 message ไปยัง mail server ฝั่งผู้รับเดียวกัน จะสามารถส่ง message ทั้งหมดบน TCP connection เดียวกันได้

2.1.2 Mail Message Format and MIME

ตามโครงสร้างของอีเมล์จะประกอบไปด้วย 2 ส่วนหลักคือส่วนหัว (Header) สำหรับเก็บข้อมูลผู้รับ ผู้ส่ง เส้นทางการเดินทางของอีเมล์ รวมทั้งประเภทของข้อมูลที่อยู่ในส่วนเนื้อหาด้วย อีกส่วนคือส่วนเนื้อหา (Body) เก็บข้อมูลที่เป็นสาระสำคัญของอีเมล์นั้นในรูปแบบต่างๆ เช่น ข้อความ เสียง รูปภาพ หรือภาพเคลื่อนไหว เป็นต้น

ส่วน Header ของ mail ประกอบด้วยลำดับของ Header line โดย Header line และส่วน Body ของ message แยกกันด้วยบรรทัดว่าง โดยทุก Header ต้องมี From : Header line, TO : Header line และ Subject : Header line

Message ที่จะส่งไปใน TCP connection ประกอบด้วย ส่วนที่เป็น Header ของ message บรรทัดว่าง และ message body โดยบรรทัดสุดท้ายจะเป็นจุด 1 จุด เพื่อบอกว่าจบ message แล้ว

THE MIME Extension for Non-ASCII Data

เนื่องจาก Header ของ message ที่กำหนดอยู่ใน RFC822 สำหรับส่งข้อมูลแบบ ASCII นั้น ไม่สามารถทำการส่งข้อมูลประเภท multimedia เช่น รูปภาพ audio และ video ได้ทำให้ต้องทำการเพิ่ม Header พิเศษขึ้นมา เรียกว่า MIME (Multimedia mail extension) เพื่อส่งข้อมูลประเภท multimedia โดยต้องเพิ่มบรรทัดใน Header เพื่อประกาศ MIME content type

ประเภทของ MIME type มีดังนี้

- Text ใช้สำหรับชี้ให้ user agent ฝั่งผู้รับรู้ว่า body message ประกอบด้วยข้อมูลแบบ text เช่น text/plain
- Image ใช้สำหรับชี้ให้ user agent ฝั่งผู้รับรู้ว่า body message ประกอบด้วยรูปภาพ เช่น image/gif และ image/jpeg เมื่อ user agent ฝั่งผู้รับได้รับ
- Application ใช้สำหรับชี้ให้ user agent ฝั่งผู้รับรู้ว่าให้ทำการใดๆกับ body message ด้วยแอพพลิเคชันที่กำหนด เช่น application/msword

2.2 Mail Access Protocol

โพรโทคอล ที่ใช้ในการรับอีเมล์นิยมใช้กัน 2 โพรโทคอล ได้แก่ POP3 (Post Office Protocol – Version 3) และ IMAP (Internet Mail Access Protocol) โดย SMTP เป็นโพรโทคอลที่ใช้สำหรับส่งอีเมล์ระหว่าง user agent กับ mail server หรือระหว่าง mail server ด้วยกัน ส่วน POP3 และ IMAP จะใช้สำหรับรับเมล์จาก mail server จากฝั่งผู้รับไปยัง user agent ฝั่งผู้รับ

2.2.1 โพรโทคอล POP3

POP3 เป็นโพรโทคอลที่ใช้ในการรับเมล์อย่างง่าย โดยการทำงานจะเริ่มจาก user agent ทำการเปิด TCP connection ไปยัง mail server โดยใช้ port 110 เมื่อ TCP connection ถูกสร้างขึ้น POP3 จะมีขั้นตอนการทำงาน 3 ขั้นตอนด้วยกันได้แก่ authorization, transaction และ update ในขั้นตอนแรก user agent จะส่ง username และ password เพื่อทำการ authenticate ในขั้นตอนที่ 2

user agent จะทำการสำเนาอีเมล์และสามารถเลือกอีเมล์บน mail server ที่จะทำการลบได้ ในขั้นตอนที่ 3 จะเกิดหลังจาก client ส่งคำสั่ง quit เพื่อจบการทำงานของ POP ในเวลานี้ mail server จะลบอีเมล์ที่ทำการเลือกไว้ทิ้ง ดูได้จากตัวอย่างในรูปที่ 2.3

ในการทำงานของ POP3 นั้น user agent จะทำการส่งคำสั่งไป และทาง server จะตอบกลับแต่ละคำสั่งนั้นด้วย +OK เพื่อยืนยันว่าได้รับข้อมูลจาก client เป็นที่เรียบร้อยแล้ว หรือ –ERR เพื่อบอกว่าก่อนหน้านี้มีข้อผิดพลาดเกิดขึ้นตามรูปที่ 2.2

```
telnet mailServer 110
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on
```

รูปที่ 2.2 แสดงการติดต่อระหว่าง user agent กับ POP3 server

```
C : list
S : 1 498
S : 2 912
S : .
C : retr 1
S : (blah blah ...
S : .....
S : .....blah)
S : .
C : dele 1
C : retr 2
S : (blah blah ...
S : .....
S : .....blah)
S : .
C : dele 2
C : quit
S : +OK POP3 Server signing off
```

รูปที่ 2.3 แสดงตัวอย่างคำสั่งในขั้นตอน transaction และ update

หลังจากที่มีการส่งคำสั่ง Quit แล้ว POP3 server จะทำการอัพเดต Mailbox โดยการทำการลบ mail ที่เลือกเอาไว้แล้วนั้น จะเห็นว่าการทำงานแบบนี้เป็นลักษณะของการทำงานแบบ Download-and-delete mode ซึ่งก็คือ เมื่อทำการ download mail มาแล้วจะทำการลบ mail นั้นออกจาก Mailbox ทันที ซึ่งปัญหาที่ตามมาจากการทำงานของ mode นี้จะส่งผลต่อบางกรณีเช่น เมื่อ user

ทำการอ่าน mail จากที่บ้านแล้ว จะไม่สามารถทำการอ่าน mail เดิมนั้นได้อีกในที่อื่นๆ นอกจากที่บ้าน แต่ถ้ามีการทำงานที่เป็น Download-and-keep mode นั้น จะทำให้สามารถอ่าน mail นั้นจากหลายๆ ที่ได้ เพราะไม่มีการลบ mail นั้นออกจาก Mailbox

2.2.2 โพรโทคอล IMAP

เมื่อผู้รับทำการ Download message มายังเครื่องของตนเองโดยใช้ POP3 ผู้รับสามารถจัดการกับ message ที่เครื่องของตนเองเช่น สร้าง folder ที่ใช้เก็บ ลบ และเคลื่อนย้าย message ระหว่าง folder ได้ แต่ผู้รับไม่สามารถทำการจัดการในลักษณะที่กล่าวมานั้นได้ที่ remote server ดังนั้นเพื่อจัดการกับปัญหาตรงนี้ IMAP (Internet Mail Access Protocol) จึงถูกสร้างขึ้นมาซึ่งมีคุณสมบัติต่างๆ มากกว่า POP3 แต่ก็มีความซับซ้อนมากกว่า POP3 ด้วย

IMAP ถูกออกแบบมาเพื่ออนุญาตให้ผู้ใช้สามารถจัดการกับ remote Mailbox ได้โดยในการทำงานนั้น IMAP server จะต้องทำการเก็บข้อมูลสถานะของ folder ของ user แต่คนละคนไว้ ซึ่งจะตรงกันข้ามกับ POP3 คือจะไม่เก็บสถานะเกี่ยวกับ user เลย โดยมาก IMAP จะถูกนำมาใช้งานในองค์กรธุรกิจเนื่องจากอีเมล์ถูกจัดเก็บไว้ที่เดียวเพื่อช่วยในการจัดการเรื่องความมั่นคงของข้อมูลในอีเมล์ หากมีผู้ใดกระทำการใดๆ กับอีเมล์นั้นๆ ก็สามารถกระทำได้เพียงแต่ผู้ที่มีสิทธิเท่านั้น ผู้ใช้ที่ remote เข้ามาเพื่อใช้งานก็จะสามารถสืบค้นอีเมล์ได้ตามลักษณะการจัดเก็บที่คุ้นเคยทำให้ง่ายต่อการใช้งานไม่เกิดการสับสนในการค้นหาเมล์

2.3 การควบคุมสแปมเมล์

มาตรการควบคุมสแปมเมล์แต่หน่วยงานต่างๆ ใช้ควบคุมก็มีแตกต่างกันออกไป แต่จำแนกออกมาได้ 3 ลักษณะหลักดังนี้

- การออกข้อบังคับในองค์กร
- ออกมาตรการทางเศรษฐกิจโดยเพิ่มค่าใช้จ่ายในการรับและส่งอีเมล์
- การแก้ปัญหาด้วยวิธีทางเทคนิค

2.3.1 การออกข้อบังคับในองค์กร

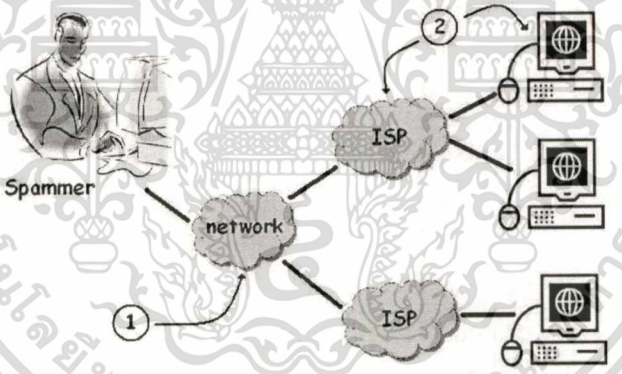
เป็นการแก้ปัญหาในเชิงบริหารโดยออกนโยบายในองค์กรเพื่อเป็นแนวทางในการปฏิบัติงานต่อการรับและส่งอีเมล์ หรือให้บริการเป็น relay mail โดยมากมักจะกล่าวในลักษณะจะไม่ให้บริการใดแก่ผู้ที่ส่งสแปมเมล์ ห้าม Spammer สามารถใช้บริการ relay mail ได้ เป็นต้น

2.3.2 ออกมาตรการทางเศรษฐกิจ

เนื่องจากการรับและส่งอีเมล์มีค่าใช้จ่ายที่ถูกมากต่อการรับส่งอีเมล์ในหนึ่งครั้ง ทำให้ผู้ใช้ไม่ให้ความสำคัญในการแก้ปัญหาสแปมเมล์ที่เกิดขึ้นกับตนเองมากนัก จึงมีแนวคิดในการในการที่จะทำให้ผู้ใช้ดังกล่าวเกิดความตระหนักในการรับและส่งอีเมล์ในแต่ละครั้ง และหมั่นตรวจสอบ Mailbox ของตนให้มีสแปมเมล์น้อยที่สุด โดยการเพิ่มค่าใช้จ่ายในการรับและส่งอีเมล์ในแต่ละครั้ง

2.3.3 การแก้ปัญหาโดยวิธีทางเทคนิค

เป็นการแก้ปัญหาทางเทคนิคและเป็นที่นิยมใช้มากที่สุดในการแก้ปัญหาสแปมเมล์ ตามที่สรุปการทำงานของสแปมเมล์ตามลักษณะดังกล่าว เราสามารถป้องกันสแปมเมล์ได้ 2 วิธีหลัก ขัดขวางไม่ให้ซอฟต์แวร์ส่งสแปมเมล์สามารถทำการส่งผ่านบนระบบเครือข่ายได้ หรือควบคุมการเข้าถึงเมล์เซิร์ฟเวอร์โดยสแปมเมอร์ในกระบวนการนี้จะมีเป้าหมายเพื่อตรวจจับและกำจัด สแปมเมล์



รูปที่ 2.4 แสดงตำแหน่งที่มักนิยมตรวจสอบสแปมเมล์

จากรูปที่ 2.4 แสดงถึงตำแหน่งที่ตรวจสอบอีเมล์ว่าเข้าขายสแปมเมล์หรือไม่ โดยจุดที่ 1 ในรูปมักมีลักษณะขัดขวางไม่ให้อีเมล์ที่เข้าข่ายว่าเป็นสแปมเมล์ส่งต่อผ่านระบบเครือข่ายไปยังถึงมือผู้รับได้ จุดที่ 2 จะเป็นการตรวจสอบซ้อนอีกครั้งจะมีรายละเอียดในการตรวจสอบมากขึ้น มีรูปแบบในการตรวจสอบเนื้อหาสาระสำคัญของอีเมล์ว่าจัดอยู่ในข่ายสแปมเมล์หรือไม่ ถ้าใช่ก็จะทำการกรองออก หรือรับไว้ทั้งหมดแต่แยกประเภทไว้ให้

2.3.3.1 การขัดขวางสแปมเมล์

เป็นวิธีการจัดการกับสแปมเมล์ได้ตรงจุดที่สุด แต่ตามหลักความเป็นจริงทำได้ยาก คือปิด open relay ทุกเมล์เซิร์ฟเวอร์ที่อยู่บนระบบเครือข่ายอินเทอร์เน็ต กำหนดให้โพรโทคอล SMTP ไม่อนุญาตให้มีการส่งต่อเมล์ที่มีเฮดเดอร์ปลอม และกำหนดให้มีการพิสูจน์ตัวจริงผู้ส่ง เพื่อง่ายต่อการตรวจสอบผู้ส่งด้วย แต่ในแนวการปฏิบัติจริงๆ คงไม่สามารถกระทำได้ เนื่องจากเราคงไม่สามารถควบคุมเมล์เซิร์ฟเวอร์ทุกเครื่องได้ ถึงแม้สามารถทำได้ สแปมเมอร์คงต้องสร้างเส้นทางใหม่ขึ้นมาเองก็ได้โดยอาจแฮคเครื่องคอมพิวเตอร์ของผู้ใช้คนอื่นเพื่อใช้ส่งสแปมเมล์อีกทอดก็ได้

2.3.3.2 การพิจารณาอีเมล์

ในการขัดขวางสแปมไม่ให้เข้าผ่านระบบนั้นสามารถกระทำได้เพียงในระดับหนึ่งเท่านั้น เนื่องจากพิจารณาเพียงที่มาที่ไปของอีเมล์เท่านั้น การจำแนกสแปมด้วยรายละเอียดที่มากขึ้น เช่น พิจารณาจากทั้งเนื้อความของอีเมล์ สามารถจำแนกว่าอีเมล์นั้นเป็นสแปมหรือไม่เป็นได้ดีกว่า ซึ่งจะใช้เทคนิคในการจำแนกกลุ่มของอีเมล์ปกติและกลุ่มที่คาดว่าจะเป็นสแปมเมล์ออกจากกันเพื่อลดปริมาณของสแปมเมล์ที่จะเข้าถึงผู้ใช้งาน หรือเพื่อลดเวลาในการพิจารณาเลือกอ่านอีเมล์ของผู้รับได้ดียิ่งขึ้น

2.4 การกรองสแปมเมล์

ในการแก้ปัญหาทางเทคนิคที่กล่าวถึงในข้อ 2.3.3 นั้นสามารถนำเทคนิคในการจำแนกประเภท เพื่อใช้จำแนกสแปมเมล์ออกจากอีเมล์ปกติซึ่งมีอยู่หลายวิธีที่นิยมใช้ แต่ที่มักกล่าวถึงสองลักษณะวิธี คือ

- Heuristic Filtering เป็นเรียนรู้ลักษณะของอีเมล์และคาดเดาว่าเมล์เหล่านั้นเป็นสแปมเมล์หรือไม่
- Cooperative Filtering เป็นการร่วมมือกันของกลุ่มผู้ใช้ กำหนดรูปแบบในการสื่อสารร่วมกันเพื่อแยกแยะอีเมล์ที่ส่งระหว่างกันเป็นอีเมล์ปกติ ในบางเทคนิคในกลุ่มผู้ใช้จะส่งข้อมูลของสแปมเมล์ในแต่ละสมาชิกเพื่อทำความเข้าใจตรงกันว่าหากมีอีเมล์ลักษณะเข้าข่ายผิดปกติตามที่ได้เข้าใจร่วมกันแล้วให้ถือว่าเป็นสแปมเมล์

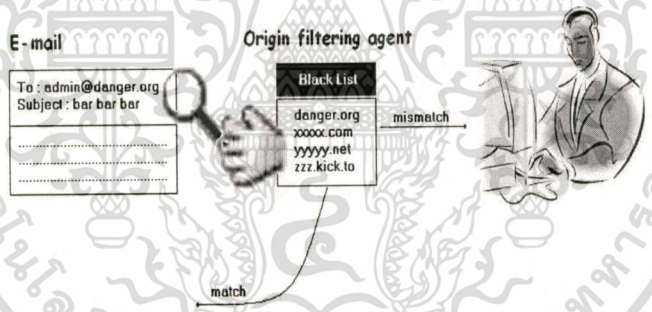
ในส่วนของเอกสารนี้จะกล่าวถึง Heuristic Filtering เนื่องจากทฤษฎีของ Bayes จัดอยู่ในการกรองกลุ่มนี้

2.4.1 Heuristic Filtering จำแนกออกเป็นสองลักษณะ

- Origin Filtering
- Message Filtering

2.4.1.1 Origin Filtering

เป็นการตรวจสอบก่อนที่จะทำการรับอีเมล์ โดยใช้ IP address, Domain name ในการตรวจสอบ เป็นการกรองโดยพิจารณาจากที่มาที่ไป และบางส่วนของอีเมล์ โดยกำหนดเงื่อนไขตั้งไว้ก่อน ว่าตรงกับที่กำหนดไว้หรือไม่ ถ้าตรงก็จะไม่ยอมให้อีเมล์นั้นส่งต่อไปถึงมือผู้รับได้เรียกว่าการทำบัญชีดำ (Blacklist) เป็นวิธีการที่ใช้ได้ตั้งแต่เมล์เซิร์ฟเวอร์อื่นๆที่ทำหน้าที่เป็น รีเลย์เมล์เซิร์ฟเวอร์จนถึงเมล์เซิร์ฟเวอร์ปลายทาง ซึ่งสามารถแยกการเชื่อมต่อทาง IP หรือ TCP ที่เป็นที่มาของสแปมเมล์ได้ ดังที่แสดงในรูปที่ 2.5 การทำบัญชีดำนิยมตรวจสอบรายชื่อผู้ส่ง หรือชื่อเรื่องของอีเมล์ แต่วิธีนี้ค่อนข้างจะมีปัญหาอยู่ตรงที่ สแปมเมอร์สามารถที่จะสุ่มชื่อผู้ส่งหรือชื่อเรื่องของอีเมล์ไม่ใช้ตรงกับบัญชีดำได้

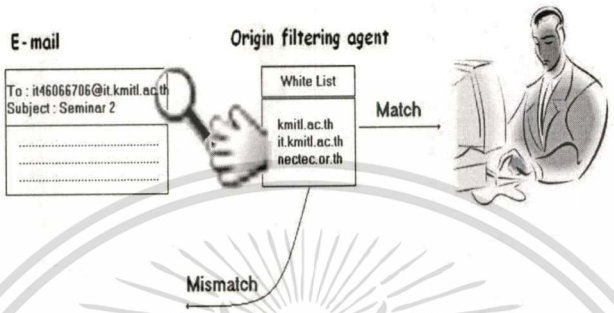


รูปที่ 2.5 แสดงหลักการกรองสแปมเมล์ด้วยเทคนิค Blacklist

อีกลักษณะหนึ่งที่จัดอยู่ในการกรองลักษณะนี้คือการกำหนดให้เซิร์ฟเวอร์ SMTP ทำการตรวจสอบกลับเพื่อเปรียบเทียบค่า IP ของอีเมล์ที่ส่งมา กับ IP ที่ทำการเชื่อมต่อกับเซิร์ฟเวอร์ มาตรวจสอบว่าตรงกันหรือไม่ ถ้าไม่ตรงกันก็จะทำการแยกออก ซึ่งลักษณะการทำงานนี้สามารถใช้กับรีเลย์เมล์

ในกรณีที่มีการสุ่มส่งสแปมจำนวนมากเพื่อให้สแปมนั้นอยู่นอกเหนือจากเงื่อนไขที่เมล์เซิร์ฟเวอร์ได้ตั้งไว้ เพื่อจะได้หลุดไปถึงผู้ใช้ ผู้ที่ดูแลเมล์เซิร์ฟเวอร์คงต้องกำหนดเงื่อนไขใหม่ๆ เป็นจำนวนมากเพื่อให้มีความสามารถเพียงพอในการกรองสแปม การทำบัญชีขาว (Whitelist) ก็เป็นอีกทางหนึ่งที่สามารถกรองอีเมล์หรือสแปมเมล์โดยทำหน้าที่กลับกันกับวิธี Blacklist คืออนุญาตให้ผ่านได้ก็ต่อเมื่ออีเมล์มีคุณสมบัติตรงกับเงื่อนไขที่ระบุไว้ เช่น กรองจากอีเมล์แอดเดรสของผู้ส่ง

หรือจากโดเมนที่ไม่รู้จัก เมื่อผู้ส่งหรือโดเมนดังกล่าวไม่มีอยู่ในรายการในบัญชีขาวจะไม่ให้ส่งผ่านไปได้ ข้อเสียของระบบนี้คือค่อนข้างยุ่งยากในการตรวจสอบอีเมล์ที่ไม่อยู่ในรายการอีกทั้งยังมีการตรวจสอบผู้ส่งรายใหม่เพื่อปรับปรุงลงในระบบด้วย และถ้าสแปมเมอร์สามารถที่จะสุ่มชื่อโดเมนให้ตรงกับที่มีอยู่ในระบบก็สามารถเข้าถึงเป้าหมายได้เช่นกัน แสดงตามรูปที่ 2.6



รูปที่ 2.6 แสดงหลักการกรองสแปมเมล์ด้วยเทคนิค Whitelist

2.4.1.2 Message Filtering

เป็นการตรวจสอบอีเมล์ที่รับมาแล้วว่าเป็นสแปมเมล์หรือไม่ โดยใช้บางส่วนของอีเมล์ เช่น IP address, Domain name, คำแต่ละคำ เป็นต้น ใช้เปรียบเทียบข้อความในอีเมล์เพื่อจำแนกประเภทการพิจารณาจากตัวของอีเมล์ เทคนิคนี้มีความจำเป็นต้องรู้ว่าสแปมเมอร์มักจะใช้ข้อความ หรือคำใด (Keyword) ในการตั้งชื่อเรื่องอีเมล์ หรือเขียนข้อความอีเมล์ คุณลักษณะอื่นๆของสแปมเมล์นำมาใช้ร่วมกับเทคนิคในการวิเคราะห์ความน่าจะเป็นว่าอีเมล์นั้นเป็นสแปมเมล์หรือไม่ นิยมใช้เทคนิค Naïve Bayesian filtering เพื่อหาค่าความน่าจะเป็นแล้วนำค่านั้นไปพิจารณาว่าเป็นสแปมเมล์หรือไม่ ก่อนที่จะใช้งานเทคนิคเราต้องสอนให้ agent (Preprocessing) ของเทคนิคนี้รู้จักเซตของคำที่อยู่ในข่ายสแปม และรู้จักเซตของคำที่ไม่ใช่สแปมเพื่อใช้ในการแยกประเภท จากสมการพิจารณาได้ว่าความน่าจะเป็นของข้อความ w ที่จะเป็นสแปม $(C = S)$ หรือที่ไม่เป็นสแปม $(C = H)$

$$P(W = w \mid C = S) = \frac{S(w)/N_S}{S(w)/N_S + H(w)/N_H}$$

เมื่อ $S(x)$ คือจำนวนของคำที่ตรงกับเซตของคำที่เป็น สแปม $H(x)$ คือจำนวนของคำที่ตรงกับเซตที่ไม่ใช่สแปม N_S และ N_H คือขนาดของเซตของคำที่เป็น สแปมและไม่ใช่สแปม

เมื่อข้อความ $M = \{w_1,...,w_N\}$ มาถึงจะเลือกเอาคำอยู่ในข่ายน่าสงสัยไปจำนวนหนึ่งโดยมีเงื่อนไขว่าคำที่เลือกไปนั้นต้องอยู่ข้อความนั้น $\{w1,...,wN\} \subseteq M$ โดยพิจารณาจาก $P(w_i) \approx 1$ หรือ $P(w_i) \approx 0$ นำคำที่เลือกนั้นมาหาค่าความไปที่ว่าข้อความ M จะเป็นสแปมตามกฎของ Bayes

$$P(C = S \mid \vec{W} = M) = \frac{P(C = S)\prod_i P(W = w_i \mid C = S)}{\sum_{k\in\{S,H\}} P(C = k)\prod_i P(W = w_i \mid C = k)}$$

เมื่อค่าความน่าจะเป็นเข้าใกล้ 1 ข้อความ M จะถูกจัดประเภทว่าเป็นสแปมเมล์

2.4.2 Cooperative Filtering

2.4.2.1 Content labeling

เป็นการเพิ่มส่วนพิเศษ (Label) รวมไว้ในอีเมล์ ซึ่ง label นี้จะแสดงถึงข้อมูลใดๆ ที่สามารถระบุ หรือรับรองได้ว่ามาจากผู้คนนั้นจริงๆ ข้อมูลใน label จะสามารถให้ผู้รับ หรือเครื่องคอมพิวเตอร์ฝั่งผู้รับสามารถแจกแจงได้ทันทีว่านั่นเป็นอีเมล์ปกติ หรือเป็นสแปมเมล์

เนื้อหาสาระของ label อาจจะเป็นการบ่งบอกถึงช่องทางในการติดต่อกันระหว่างผู้รับและผู้ส่ง ยกตัวอย่างเช่นอีเมล์ที่ส่งจากผู้ส่งสแปมเมล์จะได้โพรโทคอล SMTP TCP ที่ port 25 แต่การส่งจากผู้ใช้ในกลุ่มที่ตกลงกันอาจเพิ่ม label ที่ระบุ port number อื่นที่ผู้ส่งสแปมเมล์ไม่ได้รู้ด้วย

เทคนิคนี้ยังไม่สามารถลดจำนวนสแปมเมล์ไปได้เท่าที่ควรยังคงต้องใช้ควบคู่กับเทคนิคในการกรองเมล์ประเภทอื่นๆ นอกจากนี้ยังมีเทคนิคที่นำมาใช้ในการกรองสแปมเมล์อีกหลายเทคนิค เช่น genetic algorithms, neural network และอื่นๆ

2.5 การควบคุมสแปมเมล์ในลักษณะอื่นๆ

2.5.1 Accountability

การสื่อสารอีเมล์ผ่านเครือข่ายอินเทอร์เน็ตผู้ใช้มีความจำเป็นต้องติดต่อสื่อสารกับบุคคลที่เชื่อถือได้ มีตัวตนอยู่จริง และไม่สามารถปฏิเสธความรับผิดชอบต่อการรับและส่งอีเมล์นั้นได้ จึงมีการนำกลไกใบรับรองดิจิทอล (Digital Signature) เช่น S/MIME และ PGP/MIME มาใช้งาน โดยจำแนกออกเป็น 2 ลักษณะดังนี้

- End to End authentication
- First-Hop accountability

2.5.1.1 End to End authentication

เป็นลักษณะการพิสูจน์ตัวจริงผู้ส่งอีเมล์โดยพิจารณาจากใบรับรองดิจิทอลที่แนบมากับอีเมล์นั้นที่รับรองโดยหน่วยงานออกใบรับรองที่น่าไว้วางใจแห่งหนึ่ง ใบรับรองดิจิทอลจะแสดงรายละเอียดที่บ่งบอกถึงตัวผู้ส่ง จะถือได้ว่าอีเมล์ฉบับนั้นมีความน่าเชื่อถือได้เท่ากับที่ผู้รับชื่อถือที่รับรองใบรับรองดิจิทอลนั้น หน่วยงานที่ออกใบรับรองดิจิทอล (Certificate Authorities: CA) ได้แก่ ธนาคาร หน่วยงานรัฐบาล หรือองค์กรเอกชน เป็นต้น

2.5.1.2 First-Hop accountability

นำเอาแนวคิด End to End authentication มาปรับปรุงใช้ในการพิสูจน์ตัวจริงเซิร์ฟเวอร์ SMTP ในเบื้องต้นระบบจะทำการปฏิเสธอีเมล์ที่จากแหล่งที่มาที่ไม่รู้จัก ซึ่งเป็นการทำงานที่เกี่ยวข้องกับการติดต่อกันระหว่างสองส่วนคือ ระหว่าง First-Hop เซิร์ฟเวอร์ของผู้ให้บริการกับลูกค้า และระหว่างเซิร์ฟเวอร์ที่ทำหน้าที่เป็น relay

CA จะสร้างใบรับรองดิจิทอลที่เป็นที่น่าเชื่อถือต่อเซิร์ฟเวอร์ตัวแรก (First-Hop) ให้แก่โฮสต์ที่ทำการส่ง เพื่อเป็นการแสดงว่าผู้ส่งนั้นเป็นผู้ส่งที่ถูกรับรองโดย CA โดยใบรับรองดิจิทอลที่ถูกสร้างขึ้นมานี้จะมีอายุที่สั้นกว่าใบรับรองที่ออกให้กับผู้ใช้ (Client Certificate) ซึ่งถ้ามีการแก้ไขหรือทำการลอกเลียนแบบจะทำให้ระบบสามารถตรวจจับได้อย่างรวดเร็ว

ในการควบคุมในลักษณะนี้มุ่งเน้นที่จะตรวจสอบว่าผู้ส่งคือใครและมีอยู่จริงหรือไม่ ซึ่งก็มีหลายวิธีเช่น การตรวจสอบจากข้อมูลการเชื่อมต่อกับระบบว่ามาจากโฮสต์ใด ตรวจสอบจากชื่อโดเมนที่จดทะเบียนว่าเป็นของผู้ใด หรือเรียกร้องให้ต้องมีการล็อกอินต่อเซิร์ฟเวอร์ SMTP

บทที่ 3

การออกแบบและพัฒนาระบบงาน

3.1 ที่มาของการนำกฎของเบย์ (Bayesian) มาใช้

การพิจารณาคัดแยกสแปมเมล์ (SPAM) ออกจากอีเมล์ปกติหรืออีเมล์ที่ไม่ใช่สแปม (HAM) ด้วยวิธี Blacklist ซึ่งเป็นวิธีที่พบเห็นและนิยมใช้ทั่วไปในปัจจุบัน เนื่องจากมีค่าใช้จ่ายต่ำ และนำไปใช้งานง่าย แต่มีปัญหาในการคัดแยกว่าคัดแยกผิดเสียเป็นส่วนใหญ่ เนื่องจาก Blacklist จะพิจารณาอีเมล์ตามเงื่อนไขที่โปรแกรมหรือระบบตั้งไว้ตามลำดับ อีเมล์ฉบับเดียวกันอาจจะผ่านเงื่อนไขบางเงื่อนไข และไม่ผ่านเงื่อนไขบางเงื่อนไขได้ ทำให้อาจเกิดความผิดพลาดเพราะมีผลขึ้นอยู่กับลำดับของเงื่อนไขที่ตั้งไว้ รวมถึงข้อความหรือคำบางส่วนที่นำมาพิจารณา เช่น ชื่อเรื่องจดหมาย (Subject) อาจมีโอกาสเป็นได้ทั้ง SPAM และ HAM ก็ได้จึงสร้างปัญหาในการตั้งเงื่อนไขอีกทั้งบางส่วน หรือบางคำของข้อมูล อาจไม่สามารถนำมาตั้งเงื่อนไขได้เนื่องจากข้อจำกัดของตัวซอร์ฟแวร์เอง ทำให้ไม่ยุติธรรมต่อการคัดแยก มีผลทำให้การคัดแยกผิดพลาดได้

การคัดแยกควรพิจารณาจากส่วนต่างๆของอีเมล์ นำมาพิจารณาส่วนนั้นหรือคำนั้น เมื่อปรากฏรวมกันเป็นอีเมล์แล้วมีความน่าจะเป็น SPAM หรือ HAM มากน้อยเท่าไร

3.2 ทฤษฎีของเบย์ (Bayesian)

เนื่องด้วยทฤษฎีของเบย์ หรือ Bayesian ได้กล่าวถึงความน่าจะเป็นของการเกิดเหตุการณ์ใดๆ เมื่อรู้เหตุการณ์อื่นไว้พอสังเขปดังนี้ กำหนดให้ A และ B เป็นเหตุการณ์ใดๆ ความน่าจะเป็นของ A เมื่อรู้ B (ความน่าจะเป็นที่จะเกิดเหตุการณ์ A โดยมีเงื่อนไขว่าเหตุการณ์ B ได้เกิดขึ้นแล้ว) เขียนแทนด้วย P(A|B) สามารถคำนวณได้ด้วยทฤษฎีของเบย์ดังนี้

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

กล่าวคือความน่าจะเป็นของ A เมื่อรู้ B สามารถคำนวณได้จากผลคูณของความน่าจะเป็นของ B เมื่อรู้ A กับความน่าจะเป็นของ A หารด้วยความน่าจะเป็นของ B เราเรียก P(A) ว่าเป็นความน่าจะเป็นก่อน (prior probability) และเรียก P(A|B) เป็นความน่าจะเป็นภายหลัง (posterior probability)

ความน่าจะเป็นก่อนเป็นค่าที่ได้จากข้อมูลเบื้องต้น ส่วนความน่าจะเป็นภายหลังเป็นค่าความน่าจะเป็นก่อนที่ถูกปรับด้วยข้อมูลที่เพิ่มขึ้น

3.3 การนำทฤษฎีของเบย์มาใช้ในการคัดแยกสแปมเมล์

จากทฤษฎีในข้อ 3.2 สามารถนำไปใช้ในการเรียนรู้เพื่อคัดแยกแสปมเมล์โดยเขียนแทนการคำนวณดังนี้

$$P(spam \mid words) = \frac{P(words \mid spam)P(spam)}{P(words)}$$

แปลว่าความน่าจะเป็นแบบสุ่มที่คำที่เป็น SPAM จะอยู่ในเซตของเอกสารเท่ากับความน่าจะเป็นของคำที่ปรากฏอยู่ในเซตของคำที่เป็น SPAM กับความน่าจะเป็น SPAM หารด้วยความหน้าจะเป็นของคำนั้น และเมื่อความน่าจะเป็นของคำเท่ากับ ความน่าจะเป็นของคำที่ไปปรากฏอยู่ในเซตของ SPAM กับความน่าจะเป็น SPAM บวกด้วยความน่าจะเป็นของคำที่ปรากฏอยู่ในเซตของคำที่ไม่ใช่ SPAM (HAM) กับความน่าจะเป็นของคำที่ไม่ใช่ SPAM และเมื่อ $P(words)$ เขียนให้อยู่ในรูปอื่นได้ดังนี้

$$P(words) = P(words \mid spam)P(spam) + P(words \mid \overline{spam})P(\overline{spam})$$

เราสามารถเขียนใหม่ในรูปต่อไปนี้

$$P(spam \mid words) = \frac{P(words \mid spam)P(spam)}{P(words \mid spam)P(spam) + P(words \mid \overline{spam})P(\overline{spam})}$$

สามารถเขียนเป็นสมการในรูปได้ว่า

$$P(spam_i \mid words) = \frac{P(words \mid spam_i)P(spam_i)}{\sum_j P(words \mid spam_j)P(spam_j)}$$

ในการนำไปให้ยกตัวอย่างในกรณีที่เรารู้คำ 2 คำ แทนด้วย a,b จะถูกแทนค่าด้วย

$$P(a \mid b) = \frac{ab}{ab + (1-a)(1-b)}$$

และเมื่อเรารู้คำ 3 คำ แทนด้วย a,b,c จะถูกแทนด้วย

$$P(a \mid b \mid c) = \frac{abc}{abc + (1-a)(1-b)(1-c)}$$

ค่าความน่าจะเป็นที่ได้จะมีค่าตั้งแต่ 0- 1

เมื่อ 0 แสดงว่ามีความน่าจะเป็น SPAM 0%

เมื่อ 1 แสดงว่ามีความน่าจะเป็น SPAM 100%

จากกฎของ Bayes ข้างต้น บอกถึงแนวทางในการนำกฎนี้ไปใช้ได้ว่า เมื่อรับอีเมล์เข้ามาเราต้องแยกข้อความในอีเมล์นั้นออกเป็นคำๆ ก่อน เพื่อที่จะนำคำเหล่านั้นไปหาความน่าจะเป็นในเงื่อนไขที่ว่า ถ้ามีคำเหล่านี้ปรากฏในเอกสาร มีความน่าจะเป็นมากน้อยเพียงใดที่อีเมล์ฉบับนี้จะเป็น SPAM แต่เรื่องจากคำเหล่านี้ก่อนจะนำไปใช้กับกฎดังที่กล่าวมาจำเป็นต้องมีค่าตัวเลขที่แสดงถึงน้ำหนักของคำแต่ละคำนั้นว่า แต่ละคำนั้นมีน้ำหนักโอนเอียงไปทาง SPAM หรือ HAM โดยค่าดังกล่าวเราเรียกว่าค่า Weight ในระบบเราจะเรียกค่า Weight ว่าค่า SPAMCITY และค่า SPAMCITY ของแต่ละคำ หาได้จากสมการดังนี้

$$\text{SPAMCITY}_k = \frac{\text{Spam probability}_k}{\text{Ham probability}_k + \text{Spam probability}_k}$$

โดย $\text{Ham probability}_k = \frac{\text{tf}_k}{N_h}$

เมื่อ Ham probability_k หมายถึงความน่าจะเป็น HAM คำในลำดับที่ k (Term ที่ k)

tf_k = ความถี่ของคำที่ k ที่ปรากฏในเอกสารที่นำมาทดสอบทั้งหมด (N_h)

N_h = จำนวนเอกสารที่นำมาทำ Preprocessing ที่อยู่ใน HAM set

โดย $\text{Spam probability}_k = \frac{\text{tf}_k}{N_s}$

เมื่อ $\text{Spam probability}_k$ หมายถึงความน่าจะเป็น SPAM คำในลำดับที่ k (Term ที่ k)

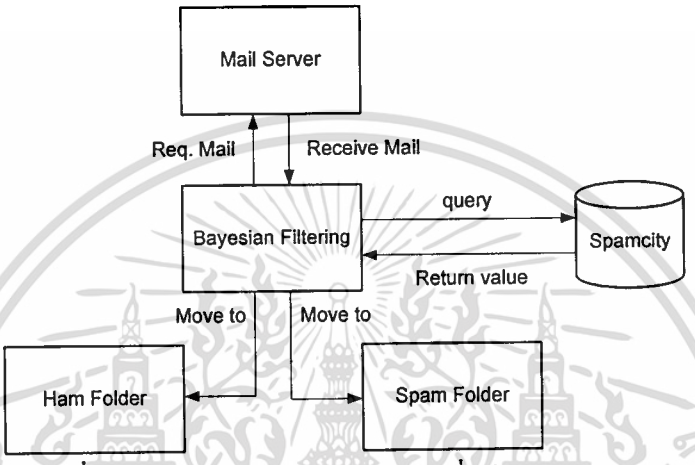
tf_k = ความถี่ของคำที่ k ที่ปรากฏในเอกสารที่นำมาทดสอบทั้งหมด (N_s)

N_s = จำนวนเอกสารที่นำมาทำ Preprocessing ที่อยู่ใน SPAM set

3.4 การออกแบบระบบ

3.4.1 โครงสร้างระบบ

ในการทดลองได้ออกแบบโครงสร้างของระบบโดยรวมเพื่อให้เข้าใจการทำงานโดยรวมดังรูปที่ 3.1

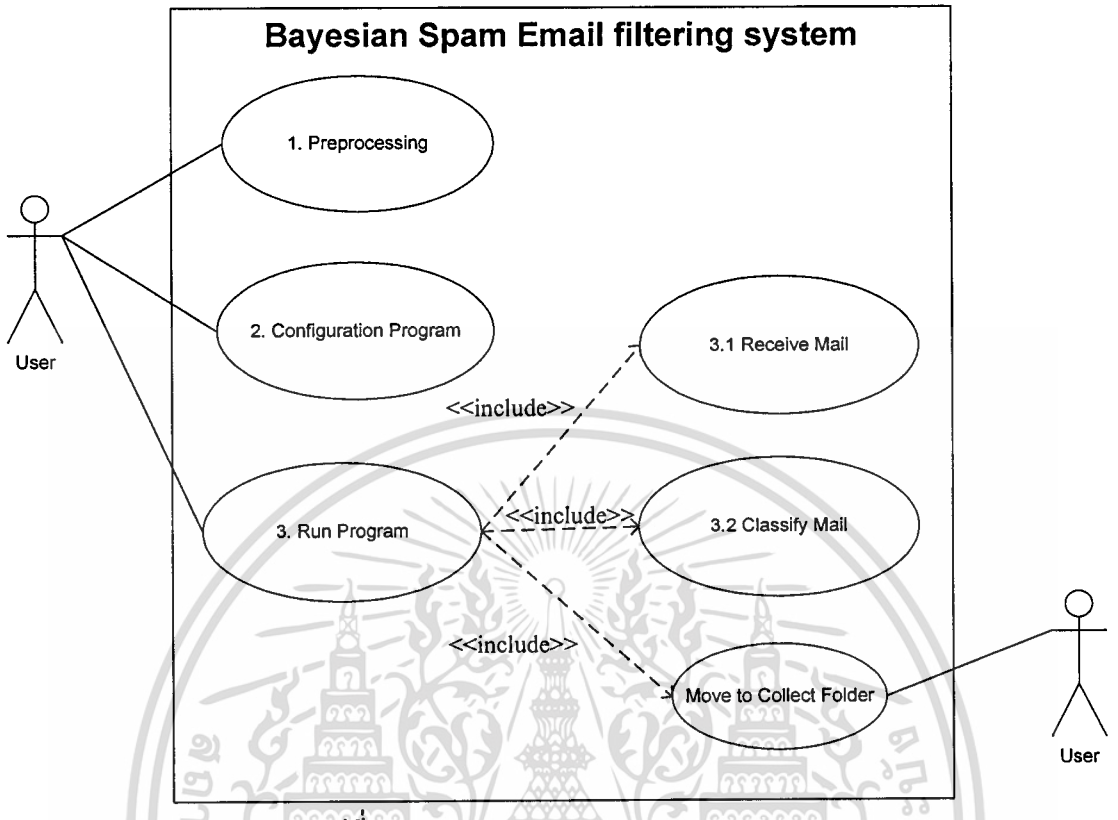


รูปที่ 3.1 แสดงโครงสร้างของระบบที่ใช้ในการทดลอง

โดยระบบจะถูกแทนโดย Bayesian Filtering ทำหน้าที่เข้าไปรับอีเมล์จาก mail server ด้วย account ที่กำหนดไว้ก่อน เมื่อรับเมล์สำเนาไว้แล้วจะทำการแยกออกเป็นคำ ที่ไม่ซ้ำกัน ใช้เป็นคำหลักที่นำไปสืบค้นเพื่อนำค่าของ SPAMCITY ที่ทำการหาไว้แล้วมาใช้ คำนวณตามอัลกอริทึม Bayesian เพื่อหาความน่าจะเป็นว่าอีเมล์แต่ละฉบับมีความน่าจะเป็น SPAM หรือไม่

3.4.2 Use Case Diagram

จากการศึกษาลักษณะการคัดแยกประเภทของอีเมล์ นำมาออกแบบระบบและเขียนเป็น Use Case Diagram ได้ดังรูปที่ 3.2 โดยระบบจะแยกออกเป็นสามส่วนหลัก คือส่วนที่หนึ่งทำการเตรียมข้อมูล (Preprocessing) เพื่อหาค่า SPAMCITY จากอีเมล์ที่ใช้นำมาทดลอง ส่วนที่สองเป็นการตั้งค่า mail account เพื่อใช้ในการรับเมล์มาจากเซิร์ฟเวอร์ และอีกส่วนคือส่วนที่ใช้ในการคำนวณความใกล้เคียงของอีเมล์กับ SPAM เมล์ ซึ่งเขียนไว้ใน Use Case ที่ 1-3 ตามลำดับ



รูปที่ 3.2 Use Case Diagram ของระบบงาน

จากรูปที่ 3.2 แสดงการทำงานโดยผู้ใช้ต้องทำการปรับแต่งค่าต่างๆ เช่น การระบุที่อยู่ของเมล์เซิร์ฟเวอร์ การระบุ mail account และ รหัสผ่าน เมื่อปรับแต่งค่าเหล่านี้เสร็จแล้วสามารถสั่งให้ระบบทำงาน จากนั้นระบบจะทำงานเองคือ ติดต่อกับเมล์เซิร์ฟเวอร์เพื่อรับอีเมล์ ทำการคัดแยกประเภทของอีเมล์ว่าเป็น SPAM หรือ HAM และทำการส่งไปอีเมล์ไปยังโฟลเดอร์ HAM และ SPAM ที่กำหนดไว้

3.4.2 Use Case Description

จาก Use Case Diagram สามารถเขียนอธิบายแต่ละ Use Case ดังตารางที่ 3.1 – 3.6

ตารางที่ 3.1 แสดงรายละเอียดของ Use Case ที่ 1

Use Case	1. Preprocess
Brief Description	เตรียมค่า SPAMCITY เพื่อนำไปใช้ในการคัดแยก
Actor	ผู้ใช้
Trigger	-
Pre-condition	-
Post-condition	ระบบมีค่าชื่อเริ่มต้นสำหรับใช้งาน
Primary scenario	1. กำหนด SPAM folder ที่เก็บ SPAM สำหรับการทดลอง (SPAM set) 2. กำหนด HAM folder ที่เก็บ HAM สำหรับการทดลอง (HAM set) 3. เปิดอีเมล์จากข้อ 1 และ 2 4. อ่านค่าส่วนหัวของอีเมล์แล้วบันทึกไว้ในหน่วยความจำ 5. อ่านค่าส่วนเนื้อความของอีเมล์เฉพาะในส่วนที่เป็น Text แล้วบันทึกเพิ่มจากข้อ 4 6. นำข้อความ Text ที่บันทึกไว้ในหน่วยความจำที่ได้จากข้อ 4 และ 5 มาทำให้อยู่ในรูปอักษรตัวเล็ก 7. แยกคำตามเงื่อนไขที่กำหนดไว้ 8. นับความถี่รวมของแต่ละคำที่ปรากฏใน SPAM set 9. นับความถี่รวมของแต่ละคำที่ปรากฏใน HAM set 10. หาความน่าจะเป็นของ SPAM ของแต่ละคำ 11. หาความน่าจะเป็นของ HAM ของแต่ละคำ 12. คำนวณค่า SPAMCITY 13. บันทึกไว้ในฐานข้อมูล
Alternatives	-

ตารางที่ 3.2 แสดงรายละเอียดของ Use Case ที่ 2

Use Case	2. Configuration System
Brief Description	ตั้งค่าเริ่มต้นระบบก่อนให้ระบบเริ่มทำงาน
Actor	ผู้ใช้
Trigger	ผู้ใช้แก้ไขไฟล์ .\panbot\system.dat ด้วยโปรแกรมจำพวก Text Editor
Pre-condition	-
Post-condition	ระบบมีค่าชื่อเริ่มต้นสำหรับใช้งาน
Primary scenario	1. ผู้ใช้เปิดไฟล์ <SYSDIR>\bin\panbot.py ด้วยโปรแกรม Text Editor 2. ระบุชื่อ username ของ mail account สำหรับเมล์เซิร์ฟเวอร์เครื่องที่ 1 และเครื่องต่อมา(ถ้ามี) ระหว่างแต่ละค่าคั่นด้วยเครื่องหมายคอมม่า (,) 3. ระบุรหัสผ่าน ของ mail account สำหรับ username คนที่ 1 และ username ของคนต่อมา(ถ้ามี) ระหว่างแต่ละค่าคั่นด้วยเครื่องหมายคอมม่า (,) 4. ระบุเมล์เซิร์ฟเวอร์แอดเดรสเครื่องที่ 1 และเครื่องต่อมา(ถ้ามี) ระหว่างแต่ละค่าคั่นด้วยเครื่องหมายคอมม่า (,) 5. ระบุเงื่อนไขการรับอีเมล์จากเมล์เซิร์ฟเวอร์ของเครื่องที่ 1 ด้วยเลข 0 เมื่อต้องการปล่อยให้อีเมล์คงอยู่ในเมล์เซิร์ฟเวอร์ หรือ 1 เมื่อต้องการให้ระบบทำการลบอีเมล์นี้ออกจากเมล์เซิร์ฟเวอร์ ระบุเงื่อไขการรับอีเมล์จากเมล์เซิร์ฟเวอร์ของเครื่องต่อมา(ถ้ามี) ระหว่างแต่ละค่าคั่นด้วยเครื่องหมายคอมม่า (,)
Alternatives	-

ตารางที่ 3.3 แสดงรายละเอียดของ Use Case ที่ 3

Use Case	3. Run Program
Brief Description	สั่งให้ระบบเริ่มทำงาน
Actor	ผู้ใช้
Trigger	พิมพ์คำสั่ง python <ไดร์ฟที่ติดตั้ง>:\<SYSDIR>\bin\panbot.py
Pre-condition	-
Post-condition	รับอีเมล์จากเมล์เซิร์ฟเวอร์ คัดแยกประเภทอีเมล์ และจัดส่งสู่ผู้รับ
Primary scenario	1. พิมพ์คำสั่งให้โปรแกรมทำงาน - python <ไดร์ฟที่ติดตั้ง>:\<SYSDIR>\bin\panbot.py
Alternatives	-

ตารางที่ 3.4 แสดงรายละเอียดของ Use Case ที่ 3.1

Use Case	3.1 Receive mail
Brief Description	ติดต่อ และรับอีเมล์จากเมล์เซิร์ฟเวอร์
Actor	-
Trigger	-
Pre-condition	-
Post-condition	รับอีเมล์มาได้อย่างถูกต้อง
Primary scenario	1. โปรแกรมทำการสร้าง connection ไปยังเมล์เซิร์ฟเวอร์ ตามเงื่อนไขการ Configuration System ที่อยู่ Use Case ที่ 1 2. โปรแกรมทำการพิสูจน์ตัวจริงสำหรับ mail account นั้น ตามเงื่อนไขการ Configuration System ที่อยู่ Use Case ที่ 1 3. รับอีเมล์จากเมล์เซิร์ฟเวอร์ 4. ลบอีเมล์จากเมล์เซิร์ฟเวอร์ ตามเงื่อนไขการ Configuration System ที่อยู่ Use Case ที่ 1 5. ยกเลิก connection ต่อเมล์เซิร์ฟเวอร์
Alternatives	-

ตารางที่ 3.5 แสดงรายละเอียดของ Use Case ที่ 3.2

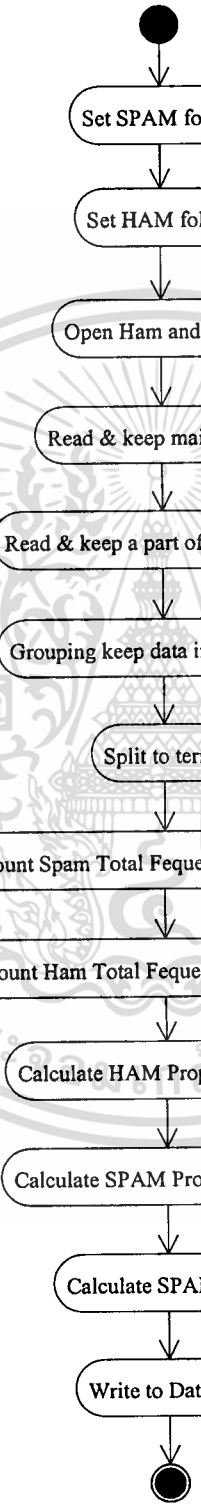
Use Case	3.2 Classify Mail
Brief Description	แยกประเภทอีเมล์จัดว่าเป็น SPAM หรือ HAM
Actor	-
Trigger	-
Pre-condition	-
Post-condition	แยกประเภทอีเมล์ได้ถูกต้องตามเงื่อนไขของโปรแกรม
Primary scenario	1. เปิดอีเมล์ที่รับมาแล้ว 2. อ่านค่าส่วนหัวของอีเมล์แล้วบันทึกไว้ในหน่วยความจำ 3. อ่านค่าส่วนเนื้อความของอีเมล์เฉพาะในส่วนที่เป็น Text แล้วบันทึกเพิ่มจากข้อ 2 4. นำข้อความ Text ที่อยู่ในหน่วยความจำที่ได้จากข้อ 2 และ 3 มาทำให้อยู่ในรูปอักษรตัวเล็ก 5. แยกคำตามเงื่อนไขที่กำหนดไว้ 6. กำจัดคำซ้ำ (คำเหล่านี้เรียกว่า Key term หรือ Index term) 7. นำคำที่เหลือไปสืบค้นค่า SPAMCITY ในฐานข้อมูล 8. พิจาณาค่า SPAMCITY ที่อยู่ในช่วงที่กำหนด 9. ตรวจสอบจำนวนคำที่มีค่า SPAMCITY ที่อยู่ในช่วงที่กำหนด 10. คำนวณเพื่อหาความน่าจะเป็นสแปมของอีเมล์ว่าจัดอยู่ในประเภทใด 11. เปลี่ยนชื่อไฟล์ให้ตรงกับประเภทที่จัดไว้
Alternatives	-

ตารางที่ 3.6 แสดงรายละเอียดของ Use Case ที่ 3.3

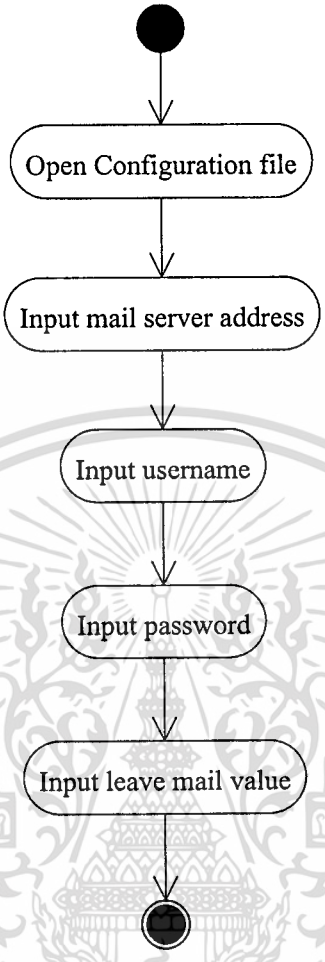
Use Case	3.3 Move to Collect Folder
Brief Description	จัดส่งอีเมล์ที่คัดแยกแล้วไปเก็บไว้ยังโฟลเดอร์แต่ละประเภท
Actor	-
Trigger	-
Pre-condition	-
Post-condition	เคลื่อนย้ายเมล์ที่แยกประเภทแล้วไปกับไว้ยังโฟล์เดอร์ที่คัดแยกไว้
Primary scenario	1. อ่านชื่อไฟล์ของอีเมล์ 2. ย้ายอีเมล์ไปเก็บในโฟลเดอร์ที่ตรงกับเงื่อนไขของการตั้งชื่อที่กำหนดไว้
Alternatives	

3.4.3 Activity Diagram

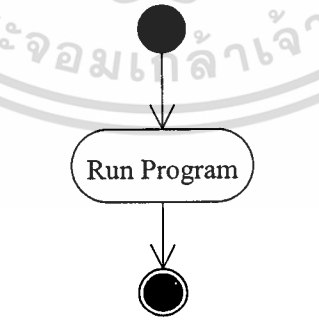
จาก Use Case Description นำมาเขียน Activity Diagram ดังรูปที่ 3.3 -3.9



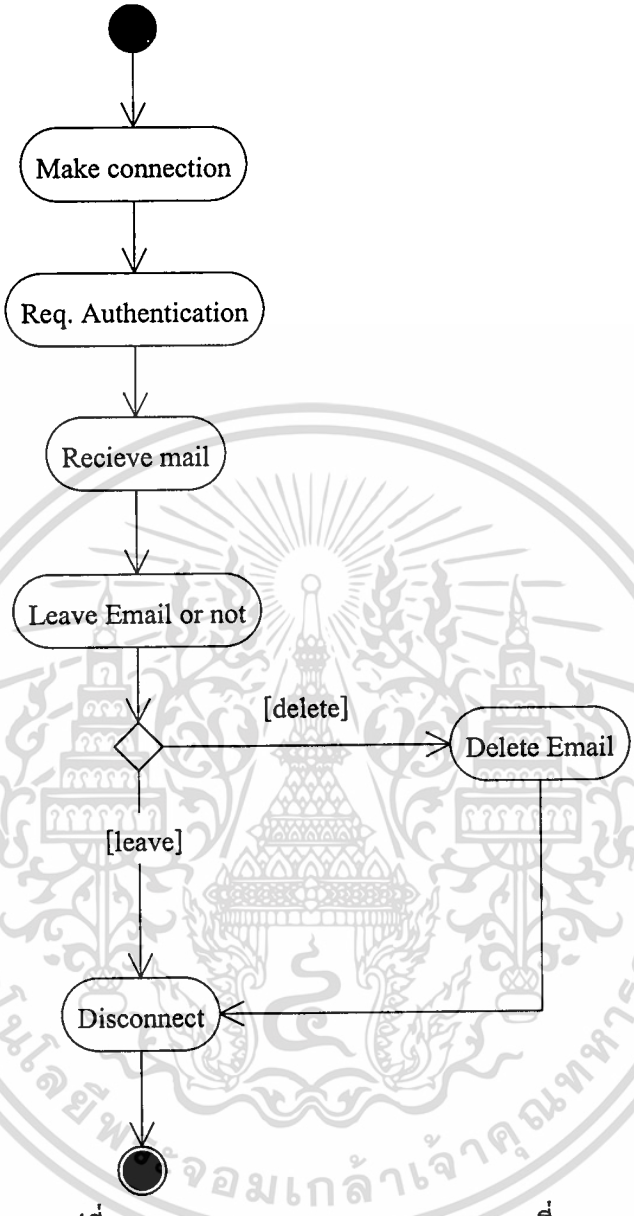
รูปที่ 3.3 Activity Diagram ของ Use Case ที่ 1



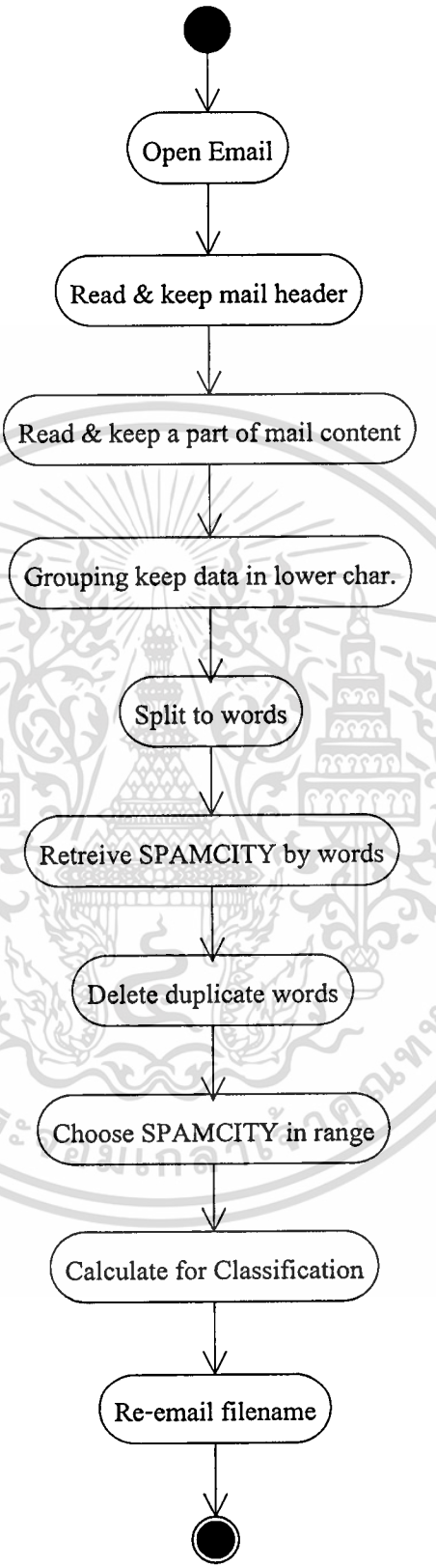
รูปที่ 3.4 Activity Diagram ของ Use Case ที่ 2



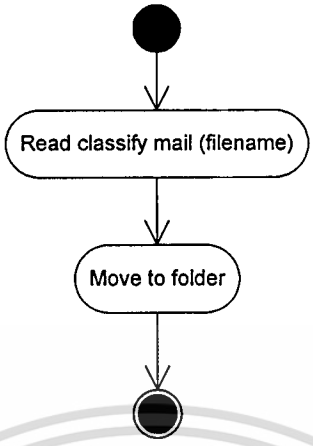
รูปที่ 3.5 Activity Diagram ของ Use Case ที่ 3



รูปที่ 3.6 Activity Diagram ของ Use Case ที่ 3.1



รูปที่ 3.7 Activity Diagram ของ Use Case ที่ 3.2



รูปที่ 3.8 Activity Diagram ของ Use Case ที่ 3.3

3.5 ข้อกำหนดอื่นๆ ที่ใช้ร่วมในระบบ

3.5.1 ขอบเขตของเนื้อหาที่ใช้พิจารณาและการแบ่งคำ

ขอบเขตการพิจารณาเนื้อความจดหมายอิเล็กทรอนิกส์เลือกพิจารณาครอบคลุมส่วนหัวของอีเมล์ และเนื้อความของจดหมายที่เป็นตัวอักษรเท่านั้น ไม่ได้พิจารณาไฟล์ที่แนบมาด้วยทุกชนิด

การพิจารณาแยกคีย์เวิร์ดจากอีเมล์ แยกตามคีย์เวิร์ดจากเครื่องหมายต่อไปนี้

(	เครื่องหมายวงเล็บเปิด
)	เครื่องหมายวงเล็บปิด
:	เครื่องหมายทวิภาค
@	เครื่องหมายแอต
<	เครื่องหมายน้อยกว่า
>	เครื่องหมายมากกว่า
\n	ตัวอักษรแบบ white space - ขึ้นบรรทัดใหม่
\r	ตัวอักษรแบบ white space - เครื่องหมาย Return
\t	ตัวอักษรแบบ white space - เครื่องหมาย Tab
"	เครื่องหมายอัญประกาศคู่
'	เครื่องหมายอัญประกาศเดี่ยว
,	เครื่องหมายจุลภาค
!	เครื่องหมายอัศเจรีย์
?	เครื่องหมายปรัศนีย์
#	เครื่องหมายนัมเบอร์ หรือ เครื่องหมายชาร์ป

$ เครื่องหมายดอลล่า

& เครื่องหมายแอมเพอร์แซนด์

* เครื่องหมายดอกจัน

+ เครื่องหมายบวก

/ เครื่องหมายหาร

{ วงเล็บปีกกาเปิด

} วงเล็บปีกกาปิด

[วงเล็บก้ามปูเปิด

] วงเล็บก้ามปูปิด

= เครื่องหมายเท่ากับ

; เครื่องหมายอัฒภาค

กล่าวคือเมื่อโปรแกรมเปิดอ่านเอกสารและพบเครื่องหมายดังรายการที่กำหนด จะทำการตัดคำโดยใช้เครื่องหมายดังกล่าวตัวแยก

3.5.2 การตั้งชื่อไฟล์อีเมล์

อีเมล์ที่รับมาทั้งหมดจะถูกจัดเก็บไว้ในโฟลเดอร์ <SYSDIR>\incoming และเมื่อทราบผลคัดแยกอีเมล์ที่รับเข้ามาว่ามีความน่าจะเป็น SPAM อยู่ที่ 50% ขึ้นไปจะถูกย้ายไปเก็บที่ <SYSDIR>\SPAM ในกรณีที่มีความน่าจะเป็น SPAM ต่ำกว่านั้นจะถูกย้ายไปเก็บที่ <SYSDIR>\HAM โดยรูปแบบของการบันทึกไฟล์ถูกกำหนดไว้ดังนี้

T280106180940_020.eml

คำอธิบาย

หลักที่ 1 หมายถึงสถานะหรือลักษณะของอีเมล์

T – เป็นเมล์ชั่วคราวที่อยู่ในระหว่างพิจารณาว่าเป็น SPAM หรือ HAM

S – บ่งบอกว่าอีเมล์ฉบับนี้เป็น SPAM จะถูกจัดเก็บอยู่ในโฟลเดอร์

H – บ่งบอกว่าอีเมล์ฉบับนี้เป็น HAM จะถูกจัดเก็บอยู่ในโฟลเดอร์

หลักที่ 2-7 หมายถึงวันที่รับอีเมล์ฉบับนี้มาจากเมล์เซิร์ฟเวอร์

หลักที่ 8-13 หมายถึงเวลาที่รับอีเมล์ฉบับนี้มาจากเมล์เซิร์ฟเวอร์

หลักที่ 15-18 หมายถึงลำดับของอีเมล์ที่รับเข้ามาแต่ละครั้ง

3.5.3 การระบุรายละเอียดของ mail account

ใน Use Case ที่ 2 มีส่วนที่อ้างถึงการ Configuration Programs เพื่อให้ระบุรายละเอียดของ mail account 4 อย่างคือ username, password, hostaddress และ delete mail flag สามารถกระทำได้โดยแก้ไขไฟล์โดยใช้โปรแกรมประเภท Text Editor ที่ <SYSDIR>\bin\panbot.py ที่ส่วนล่างสุดของไฟล์ ใน method ชื่อว่า main จะเก็บค่าของ mail account ไว้ แล้วบันทึกไว้ โดยมีรูปแบบดังนี้

t_users = ['username1','username2',...,'username N']

t_pass = ['password1', 'password2', ..., 'password N']

t_hostaddress = ['hostaddress1', 'hostaddress2', ..., 'hostaddress N']

t_deleFlag = [delete falg 1, delete falg 2, ..., delete falg N]

อย่างน้อยต้องมีการกำหนดไว้ 1 mail account ส่วนความหมายของ Delete Flag ให้ 0 มีความหมายว่า copy อีเมล์จากเซิร์ฟเวอร์ แต่ไม่ลบ และให้ 1 มีความหมายว่า copy อีเมล์จากเซิร์ฟเวอร์แล้วทำการลบออก ตัวอย่างการกำหนด mail account

```
t_users = ['postmaster']
t_pass = ['123']
t_hostaddress = ['127.0.0.1']
t_deleFlag = [0]
```

3.5.4 ขอบเขตของข้อมูลที่นำมาพิจารณา

จากการแบ่งคำ ในอีเมล์ที่มีความยาวมาก จะพบว่ามีคำที่แบ่งเป็นจำนวนมากซึ่งมีผลกับเวลาในการคำนวณ ในการทดลองมีการกำหนดขอบเขตของข้อมูลในตัวอีเมล์เพื่อใช้ในการพิจารณาเปรียบเทียบกันระหว่างความเร็วในการพิจารณาคัดแยก และความถูกต้อง โดยแบ่งออกเป็น 4 ลักษณะดังตารางที่ 3.7

ตารางที่ 3.7 แสดงการแยกขอบเขตของอีเมล์ที่นำมาใช้ทดสอบ

ขอบเขตที่	ชื่อขอบเขตบนโปรแกรม	ความหมาย
1	Full	พิจารณาข้อความแบบ Text ทั้งหมด
2	Some Content	พิจารณาบางส่วนของ Header เฉพาะบางคีย์ คือ Received, To, Bcc, Cc, From และ Subject ร่วมกับส่วนของ Body ที่มีข้อมูลเป็นข้อความ Text
3	Body	พิจารณาส่วนของ Body ที่มีข้อมูลเป็นข้อความ Text
4	Header	พิจารณาบางส่วนของ Header เฉพาะบางคีย์ คือ Received, To, Bcc, Cc, From และ Subject

จากขอบเขตที่กำหนดไว้ในตารางที่ 3.7 จะถูกนำไปแบ่งเป็นประเภทของข้อมูลที่ใช้ในการทดลอง มีรูปแบบแสดงดังตารางที่ 3.8

ตารางที่ 3.8 แสดงความหมายของแต่ละประเภทจากอีเมล์ที่นำมาใช้ในการทดลอง

ประเภท	ความหมาย
HAM-1	อีเมล์ที่อยู่ใน HAM Set มีขอบเขตในการทดสอบตามขอบเขตที่ 1 ตามตารางที่ 3.7
HAM-2	อีเมล์ที่อยู่ใน HAM Set มีขอบเขตในการทดสอบตามขอบเขตที่ 2 ตามตารางที่ 3.7
HAM-3	อีเมล์ที่อยู่ใน HAM Set มีขอบเขตในการทดสอบตามขอบเขตที่ 3 ตามตารางที่ 3.7
HAM-4	อีเมล์ที่อยู่ใน HAM Set มีขอบเขตในการทดสอบตามขอบเขตที่ 4 ตามตารางที่ 3.7
SPAM-1	อีเมล์ที่อยู่ใน SPAM Set มีขอบเขตในการทดสอบตามขอบเขตที่ 1 ตามตารางที่ 3.7
SPAM-2	อีเมล์ที่อยู่ใน SPAM Set มีขอบเขตในการทดสอบตามขอบเขตที่ 2 ตามตารางที่ 3.7
SPAM-3	อีเมล์ที่อยู่ใน SPAM Set มีขอบเขตในการทดสอบตามขอบเขตที่ 3 ตามตารางที่ 3.7
SPAM-4	อีเมล์ที่อยู่ใน SPAM Set มีขอบเขตในการทดสอบตามขอบเขตที่ 4 ตามตารางที่ 3.7

3.6 ความต้องการของระบบ

3.6.1 ความต้องการของระบบทางฮาร์ดแวร์

- เครื่อง PC CPU ความเร็วในการประมวลผลที่ 1GHz. ขึ้นไป
- หน่วยความจำหลัก 128 MB. ขึ้นไป
- พื้นที่ว่างบนหน่วยความจำหลัก 30 MB

3.6.2 ความต้องการของระบบทางซอฟต์แวร์

- ระบบปฏิบัติการ MS-Windows และสามารถเชื่อมต่อ ODBC ได้
- โปรแกรม Python เวอร์ชั่น 2.4 เพื่อใช้เป็นตัว Interpreter Source Code
- โปรแกรมเสริมของ PyWin 2.0 เพื่อทำให้ python สามารถติดต่อ ODBC ได้
- โปรแกรม POP3 Mail Server (ใช้จำลองเมล์เซิร์ฟเวอร์เพื่อทดลองใช้งาน)
- โปรแกรม PANBOT (โปรแกรมของโครงการพิเศษนี้)

3.7 การติดตั้งระบบ

1) ติดตั้งโปรแกรม PANBOT โดย Copy โปรแกรมทั้งโฟลเดอร์ ลงในไดรฟ์ C:\
2) ติดตั้งโปรแกรม Python 2.4
3) ติดตั้งโปรแกรมเสริม PyWin 2.0
4) ติดตั้งและปรับค่าการใช้งานของ Mail Server ให้สามารถใช้งานรับส่งเมล์ภายในได้
5) สร้าง System Data Source บน ODBC โดยให้ชื่อว่า SPAMCITY และอ้างถึงไฟล์ C:\PANBOT\SPAMDATA.MDB

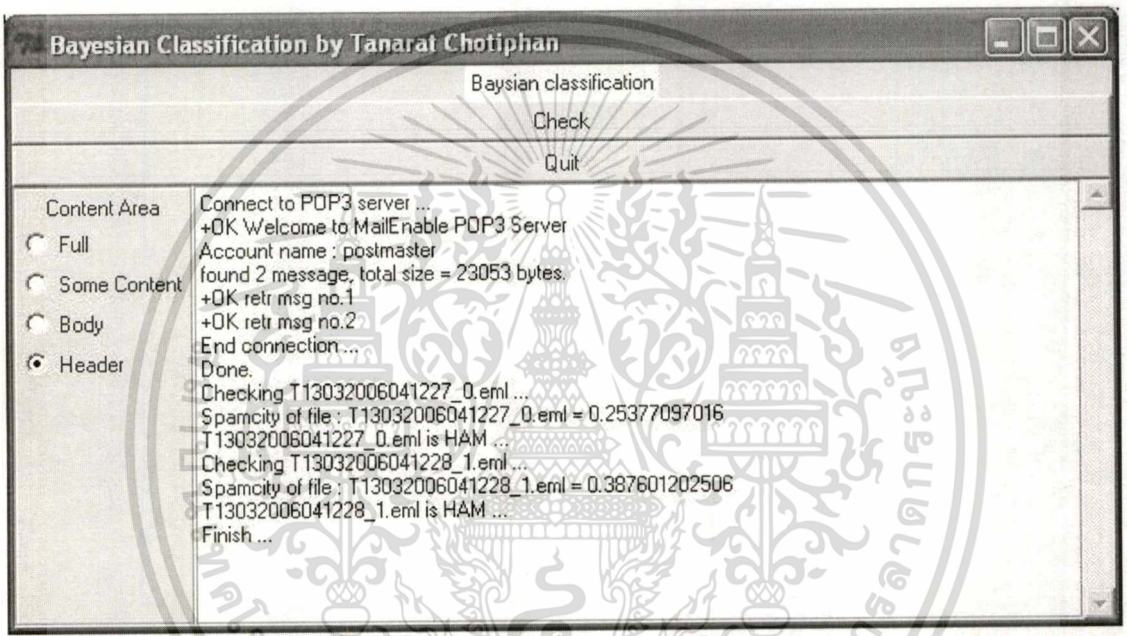
3.8 การเรียกใช้งานระบบ

การเรียกใช้โปรแกรมเรียกจาก Command line ด้วยคำสั่ง python c:\panbot\baygui.py โปรแกรมให้เราเลือกขอบเขตของข้อมูลแล้วทำกดปุ่ม Check เพื่อรับอีเมล์จากเมล์เซิร์ฟเวอร์มาพิจารณา แล้วรายงานผล

3.9 ตัวอย่างการทำงาน

เพื่อความเข้าใจที่ดียิ่งขึ้นในการทำงานโดยอ่านจากกรณีตัวอย่างของโปรแกรมดังต่อไปนี้ เริ่มทดลองโดยการเรียกใช้โปรแกรม จากรูปภาพที่ 3.9 โปรแกรมจะแสดงหน้าต่างรอให้ผู้ใช้กำหนดประเภทของขอบเขตของข้อมูล ตามที่กำหนดไว้ในข้อ 3.5.4 จากตัวอย่างเลือกเฉพาะในส่วนของ Header บางส่วน เมื่อเลือกขอบเขต แล้วกดปุ่ม Check เพื่อทำการรับอีเมล์ตามที่กำหนดไว้ใน ข้อ 3.5.3 และคัดแยก เมื่อเชื่อมต่อกับเมล์เซิร์ฟเวอร์ได้แล้วจะแจ้ง Welcome message คือ *+OK +OK Welcome to MailEnable POP3 Server* กลับมายังเครื่องของผู้ใช้เพื่อแสดงว่าโปรแกรมได้ทำการเชื่อมต่อกับเมล์เซิร์ฟเวอร์เรียบร้อยแล้ว จากนั้นโปรแกรมจะกระทำการยืนยันว่าเป็นผู้ใช้งานเมล์เซิร์ฟเวอร์ตัวจริงเพื่อขอรับเมล์ ตามชื่อผู้ใช้และรหัสที่กำหนดไว้ ในหน้าต่างจะแสดง *Account name : postmaster* เพื่อแจ้งให้ผู้ใช้ทราบว่าขณะนี้กำลังร้องขอเพื่อรับเมล์โดยใช้ mail account ชื่อว่า

postmaster เมื่อผ่านการพิสูจน์ตัวจริงจากเมล์เซิร์ฟเวอร์แล้วโปรแกรมจะรายงานว่ามีอีเมล์ของ mail account นี้ จำนวนทั้งหมด รวมขนาดทั้งสิ้นแสดงเป็นหน่วยเป็น byte และทำการรับอีเมล์จากเมล์เซิร์ฟเวอร์ แสดงเป็นสถานะการรับทีละฉบับจนครบ เนื่องจากการระบุค่า t_deleFlag = 0 จึงไม่ทำการสั่งให้เมล์เซิร์ฟเวอร์ลบอีเมล์ออกจากเมล์เซิร์ฟเวอร์ แต่หากต้องการให้ลบอีเมล์ทิ้ง ก็ให้เปลี่ยนค่า t_deleFlag = 1 สำหรับ mail account นั้น เมื่อรับเมล์ครบทุกฉบับแล้วโปรแกรมจะทำการสิ้นสุดการเชื่อมต่อและทำการติดต่อไปยังเมล์เซิร์ฟเวอร์อื่นๆ ที่กำหนดไว้จนครบ



รูปที่ 3.9 แสดงหน้าต่างการทำงานของโปรแกรม

หลังจากที่โปรแกรมได้ทำการรับอีเมล์มารอไว้เพื่อการพิจารณาว่าเป็น SPAM หรือ HAM โปรแกรมจะทำงานต่อโดยการนำอีเมล์ที่รับมา แยกออกเป็นคำๆ เพื่อใช้วิเคราะห์ความน่าจะเป็นที่แต่ละคำที่ปรากฏอยู่ในอีเมล์ว่าเป็น SPAM เมล์หรือไม่ เนื่องจากอีเมล์มีการรับส่งข้อมูลกันระหว่างเมล์เซิร์ฟเวอร์กับเมล์เซิร์ฟเวอร์ หรือเมล์ไคลเอนต์กับเมล์เซิร์ฟเวอร์ อยู่ในรูปแบบ text file ไฟล์แนบที่ใช้ในอีเมล์ก็จะถูกแปลงให้อยู่ในรูปของ text ด้วย ตามมาฐาน RFC 2822 การพิจารณาของโปรแกรมจะเลือกพิจารณาเฉพาะส่วนหัว และส่วนเนื้อความของอีเมล์ และละเว้นไม่พิจารณาไฟล์ที่แนบมาด้วย โดยเงื่อนไขในการแยกคำหรือคีย์เวิร์ดจะดำเนินการตามข้อ 3.5.1 โดยตัดหรือแยกคำเมื่อพบเครื่องหมายดังกล่าวข้างต้น เช่น ถ้าในอีเมล์มีข้อความต่อไปนี้

Hmm - I've been using MH for a long time (since well before there were sequences) and I don't think I've ever seen a "pseq" ...

ข้อความจะถูกแปลงให้เป็นตัวพิมพ์เล็ก และแยกคีย์เวิร์ดตามเครื่องหมายที่พิจารณาในการแยกคำตามเงื่อนไขที่แจ้งไว้ในข้อ 3.5.1 จะแยกได้ดังนี้

hmm, i, ve, been , using, mh, for, a, long, time, since, well, before, there, were, sequences, and, i, don, t, think, i, ve, ever, seen, a, pseq, และ ...

หลังจากนั้นโปรแกรมจะทำการกำจัดคำซ้ำออกไป เพื่อนำคำเหล่านี้มาเป็นคีย์เวิร์ด จะเหลือเพียง

hmm, i, ve, been , using, mh, for, a, long, time, since, well, before, there, were, sequences, and, don, t, think, ever, seen, pseq, และ ...

คีย์เวิร์ดที่ได้ทั้งหมดจะถูกนำไปสืบค้นเพื่อเรียกค่า Weight ของแต่ละคีย์เวิร์ด (SPAMCITY) ออกมา ซึ่งค่า SPAMCITY จะถูกเก็บไว้ใน <SYSDIR>\spamdata.mdb ซึ่งได้เชื่อมต่อไว้แล้วด้วย ODBC Driver ตามที่ระบุไว้ในข้อ 3.7 ค่าที่บันทึกอยู่ใน spamdata.mdb นี้ ได้มาจากการทำ Preprocessing ตาม Use Case ที่ 1

คีย์เวิร์ดที่ได้จากตัวอย่างข้างต้นจะถูกนำไปสืบค้นทีละตัวเพื่อสืบค้นค่า SPAMCITY ถ้าพบค่า SPAMCITY จะตรวจสอบก่อนว่าค่านั้นอยู่ในช่วงที่โปรแกรมระบุหรือไม่ (ในโปรแกรมระบุช่วงของค่า SPAMCITY ที่จะนำมาใช้ในการพิจารณาอยู่ระหว่าง 0.3 >= N > 0 และ 1 > N >= 0.7) เหตุที่เลือก SPAMCITY ได้มาจากการทดลองจริงในบทที่ 4 เพราะคีย์เวิร์ดที่อยู่ในช่วงดังกล่าวมีอำนาจจำแนกสูง มีน้ำหนักจะพิจารณาได้ว่าอีเมล์ที่ประกอบด้วยคำเหล่านั้นมีโอกาสเป็น SPAM หรือไม่ อย่างไรก็ตามการเลือกช่วงของค่า SPAMCITY ในการพิจารณาจะส่งผลต่อความแม่นยำของการคัดแยกด้วย ซึ่งคงแล้วแต่ความเหมาะสมแต่ละกรณี

ในกรณีที่ไม่พบว่ามีคีย์เวิร์ดดังกล่าวปรากฏอยู่ในฐานข้อมูล spamdata.mdb ซึ่งในกรณีทดลองนี้จะไม่เกิดขึ้นเนื่องจากอีเมล์ที่ใช้ทดลองเป็นอีเมล์ส่วนหนึ่งของอีเมล์ทั้งหมดที่ใช้ทำการ Preprocessing แต่ถ้าเกิดกรณีนี้ โปรแกรมจะกำหนดให้ค่า SPAMCITY เป็น 0.4 แสดงว่าคำหรือคีย์เวิร์ดนี้ไม่สามารถที่มีส่วนช่วยให้จำแนกได้ว่าอีเมล์นี้เป็น SPAM หรือไม่ การกำหนดค่าให้เป็น 0.4 เพียงเพื่อจะไม่ให้ค่า SPAMCITY อยู่ในช่วงที่จะนำไปพิจารณา จากตัวอย่างที่กล่าวมาเราจะได้ค่า SPAMCITY ดังตารางที่ 3.9

ตารางที่ 3.9 แสดงค่า SPAMCITY ของตัวอย่างคีย์เวิร์ด

Keywords	SPAMCITY	Keywords	SPAMCITY
hmm	0	were	.233877233877234
i	.3831610110570	sequences	0
ve	.3155216284987	and	.486538641937256
been	.4060618642308	don	.417391304347826
using	.2819717338848	t	.340560005879327
mh	.0421052631578	think	.220324508966695
for	.4268425315284	ever	.566893424036281
a	.4038983188499	seen	.464285714285714
long	.4586699813548	pseq	0
time	.4891774891774	…	.756756756756757
since	.2868998221695	before	.411934552454283
well	.361328125	there	.323932046841498

ในบางครั้งโปรแกรมอาจได้รับค่า SPAMCITY เป็น 0 เนื่องจากจำนวนคีย์เวิร์ดนั้นไม่มีปรากฏอยู่ใน SPAM Set ที่นำมาทำ preprocessing เลย หรือ 1 เนื่องจากจำนวนคีย์เวิร์ดนั้นไม่มีปรากฏอยู่ HAM Set ที่นำมาทำ preprocessing เลยเช่นกัน ค่า SPAMCITY สามารถบ่งบอกถึงน้ำหนักของคีย์เวิร์ดว่าเข้าใกล้ HAM หรือ SPAM เช่น ค่า 0.82 จะเข้าใจได้ว่าสัดส่วนของคำนั้นปรากฏอยู่ในอีเมล์ที่เป็น SPAM มากกว่า HAM ในกรณีนี้ ค่า 0 และ 1 ก็ไม่นำมาพิจารณาตามเงื่อนไข เพราะจะมีผลทำให้ค่าความน่าจะเป็นที่ออกมาเป็น 0 หรือ 1 เท่านั้น หากพิจารณาตามช่วงของ SPAMCITY ที่กำหนดเราจะได้ที่มาใช้ในการคำนวณดังตารางที่ 3.10

ตารางที่ 3.10 แสดงคำที่อยู่ในช่วงที่เลือกมาพิจารณา

Keywords	SPAMCITY
mh	0.042105263
think	0.220324509
were	0.233877234
using	0.281971734

ตารางที่ 3.10 (ต่อ)

Keywords	SPAMCITY
since	0.286899822
...	0.756756757

จากนั้นโปรแกรมนำค่าเหล่านี้มาคำนวณมาทฤษฎีของเบย์ ได้ผลดังนี้

$$\frac{(0.042105)(0.220325)(0.233877)(0.281972)(0.2869)\ (0.756757)}{(0.042105)(0.220325)(0.233877)(0.281972)(0.2869)\ (0.756757) + (1 - 0.042105)(1 - 0.220325)(1 - 0.233877)(1 - 0.281972)(1 - 0.2869)(1 - 0.756757)}$$

ได้ค่าเป็น

$$\frac{0.000132824841646615}{0.0713956291905389}$$

มีความน่าจะเป็นที่อีเมล์ตัวอย่างนี้จะเป็น SPAM = 0.00186040578607599 หรือ 0.0018%

เมื่อโปรแกรมพิจารณาความน่าจะเป็นของอีเมล์ที่รับมาแต่ละฉบับแล้วจะสำเนาส่งไปยัง HAM folder และ SPAM folder ถือว่าจบกระบวนการ

บทที่ 4

การทดลองและผลการดำเนินการ

4.1 วัตถุประสงค์การทดลอง

- เพื่อทดลองการคัดแยกอีเมล์ด้วยอัลกอริทึม Bayesian สามารถลดปัญหาการคัดแยกผิด (False Positive) โดยคัดแยกถูกต้อง 90% ขึ้นไป
- เพื่อหาข้อสรุปถึงในเรื่องความถูกต้องในการพิจารณากับความเร็วในการดำเนินการ

4.2 เงื่อนไขในการทดลอง

4.2.1 ข้อมูลที่ใช้ในการทำ Preprocessing

ข้อมูลตัวอย่างที่นำมาใช้ในการทดลองมาจาก SPAMAssassin public mail Corpus [9] มีอีเมล์ทั้งสิ้นจำนวน 9,349 ฉบับ แบ่งเป็นอีเมล์ปกติ (HAM) จำนวน 6,951 ฉบับ และเป็นสแปม (SPAM) จำนวน 2,398 ฉบับ

4.2.2 ข้อมูลที่ใช้ในการทดลอง

a. เป็นอีเมล์ส่วนหนึ่งที่ใช้ในการทำ Preprocessing (ข้อ 4.2.1) โดยสุ่มมาใช้ เป็นจำนวน 4,000 ฉบับ แบ่งเป็น HAM จำนวน 2,000 ฉบับ และเป็น SPAM จำนวน 2,000 ฉบับ

b. เป็นอีเมล์ที่จาก untroubled.org เป็นคนละชุดกับข้อ 4.2.1 สุ่มทดสอบจำนวน 2,000 ฉบับ

4.3 วิธีการทดลอง

- ทดลองคัดแยกสแปมจากข้อมูลในข้อ 4.2.2 ข้อย่อย a. ที่ SPAMCITY ในช่วง $0 < N <= 0.2$ และ $0.8 <= N < 1$ บนขอบเขตข้อมูลทั้ง 4 ประเภท
- ทดลองคัดแยกสแปมจากข้อมูลในข้อ 4.2.2 ข้อย่อย a. ที่ SPAMCITY กับข้อมูลบนขอบเขตข้อมูลที่เป็นบางส่วนของ Header อย่างเดียว ในหลายช่วง คือ

  $0 < N <= 0.1$ และ $0.9 <= N < 1$

  $0 < N <= 0.2$ และ $0.8 <= N < 1$

  $0 < N <= 0.3$ และ $0.7 <= N < 1$

  $0 < N <= 0.4$ และ $0.6 <= N < 1$

- ทดลองคัดแยกสแปมจากข้อมูลในข้อ 4.2.2 ข้อย่อย b. ที่ SPAMCITY ในช่วง 0 < N <= 0.3 และ 0.7 <= N < 1 บนประเภท SPAM-1 และ SPAM-4

4.4 สภาพแวดล้อมในการทดลอง

เครื่อง PC ที่ใช้ในการทดสอบมีความสามารถดังนี้

- CPU : Intel Pentium 4 1.6 GHz
- RAM 256 MB

4.5 ผลการทดลอง

ทดลองคัดแยกสแปมจากข้อมูลในข้อ 4.2.2 ที่ SPAMCITY ในช่วง 0 < N <= 0.2 และ 0.8 <= N < 1 บนขอบเขตข้อมูลทั้ง 4 ประเภท เพื่อหาผลการทดลองว่า การคัดแยกสแปมบนขอบเขตข้อมูลได้จะได้ผลการคัดแยกที่ดีที่สุด แสดงผลดังตารางที่ 4.1 และตารางที่ 4.2

ตารางที่ 4.1 แสดงผลการคัดแยก HAM ในแต่ละขอบเขตข้อมูล

ประเภท	คัดแยก	ฉบับ	เปอร์เซ็นต์
HAM-1	ถูก	1,886	94.3
	ผิด	114	5.7
HAM-2	ถูก	1,863	93.15
	ผิด	137	6.85
HAM-3	ถูก	1,847	92.35
	ผิด	153	7.62
HAM-4	ถูก	1,933	96.65
	ผิด	67	3.35

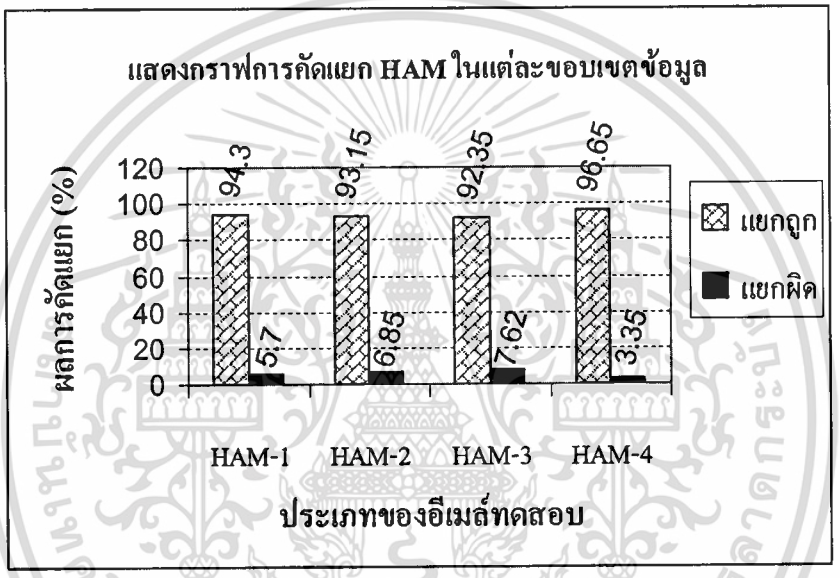
ตารางที่ 4.2 แสดงผลการคัดแยก SPAM ในแต่ละขอบเขตข้อมูล

ประเภท	คัดแยก	ฉบับ	เปอร์เซ็นต์
SPAM-1	ถูก	1,946	97.3
	ผิด	54	2.7
SPAM-2	ถูก	1,972	98.6
	ผิด	28	1.4

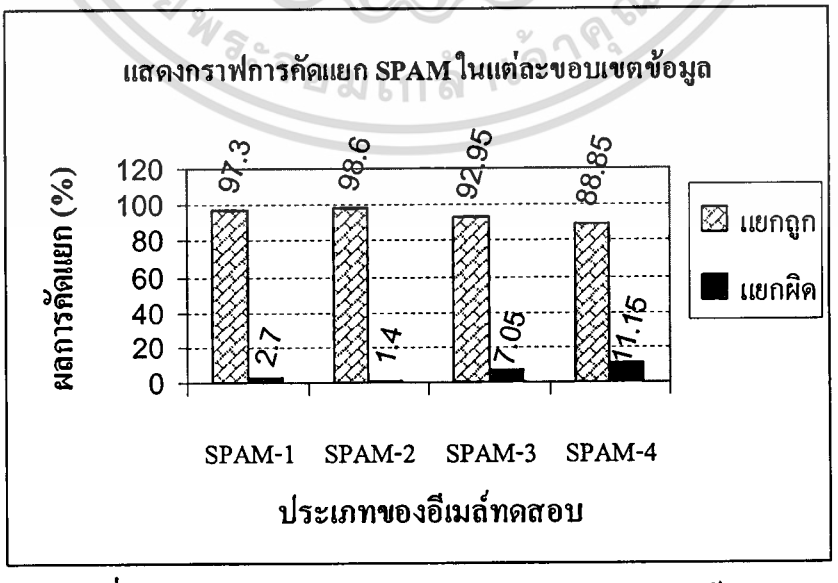
ตารางที่ 4.2 (ต่อ)

SPAM-3	ถูก	1,859	92.95
	ผิด	141	7.05
SPAM-4	ถูก	1,777	88.85
	ผิด	223	11.15

ทั้งตารางที่ 4.1 และ 4.2 แสดงเป็นกราฟได้ตามรูปที่ 4.1 และ รูปที่ 4.2



รูปที่ 4.1 แสดงกราฟการคัดแยก HAMในแต่ละขอบเขตข้อมูล



รูปที่ 4.2 แสดงกราฟการคัดแยก SPAMในแต่ละขอบเขตข้อมูล

จากการทดลองพบว่าจำนวนคำที่อยู่ในอีเมล์ส่งผลต่อการคัดแยก เมื่อจำนวนคำมากก็ใช้เวลาในการคัดแยกมากเช่นกัน จากกราฟจะเห็นได้ว่าขอบเขตข้อมูลชุดที่ 1 (พิจารณาทั้งหมด) ที่ใช้เวลาในการคัดแยกต่อฉบับนานที่สุด และชุดที่ 4 (พิจารณาเพียงบางส่วนของ Header) ใช้เวลาในการคัดแยกต่อฉบับน้อยที่สุด ไม่ส่งต่อการคัดแยก HAM มากนัก แต่ในทางตรงข้ามกลับส่งผลต่อการคัดแยก SPAM นั่นหมายความว่าหากเราเลือกพิจารณาในขอบเขตข้อมูลชุดที่ 4 ประสิทธิภาพการพิจารณา SPAM จะลดลงแต่ระดับการคัดแยกถูกก็ยังอยู่ในระดับ 88% ซึ่งถือว่าสามารถใช้งานได้ในเกณฑ์พอใช้เมื่อเทียบกับระยะเวลาที่ใช้ในการคัดแยก แต่เมื่อพิจารณาในกรณีที่ผู้ส่ง SPAM ใช้เทคนิคในการปลอม Header ทำให้การพิจารณาเฉพาะในส่วนของ Header มีความน่าเชื่อถือน้อยลง หรืออาจไม่มีความน่าเชื่อถือเลยในกรณีที่สามารถปลอม Header ได้อย่างสมบูรณ์

ทดลองคัดแยกสแปมจากข้อมูลในข้อ 4.2.2 ข้อย่อย a. ที่ SPAMCITY กับข้อมูลบนขอบเขตข้อมูลที่เป็นบางส่วนของ Header อย่างเดียวชุดเดียวกัน ในหลายขอบเขต Weight เพื่อหาช่วงของขอบเขต Weight ที่ดีที่สุดในการคัดแยก แสดงได้ดังตารางที่ 4.3 และตารางที่ 4.4

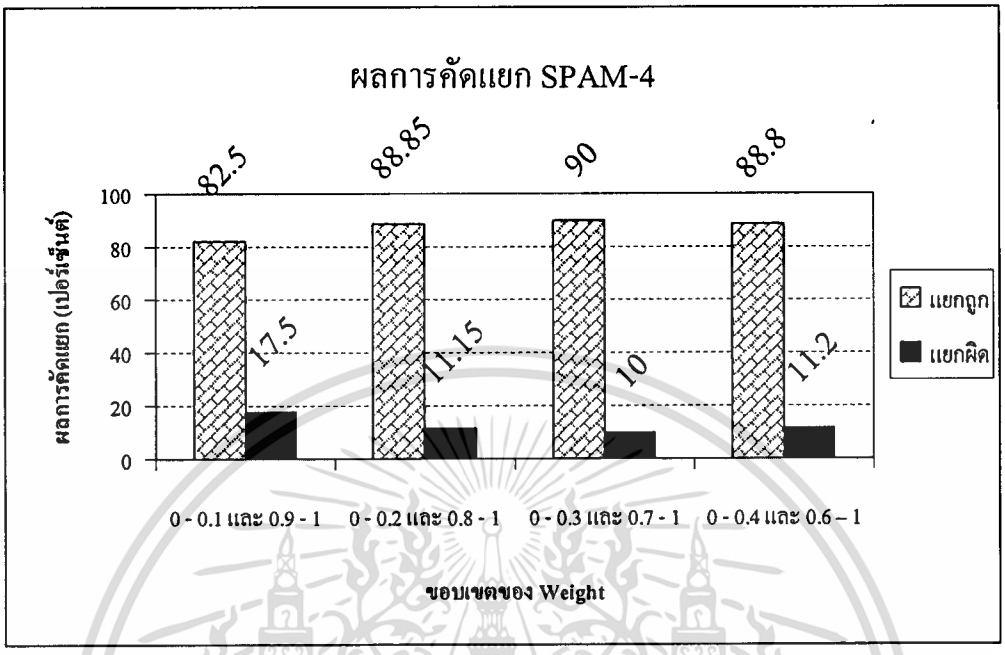
ตารางที่ 4.3 แสดงผลการคัดแยก SPAM-4 ในแต่ละขอบเขตของ Weight

ผลการคัดแยก SPAM-4		
ขอบเขต Weight	ผลการคัดแยก (%)	
	แยกถูก	แยกผิด
(0, 0.1] และ [0.9, 1)	82.50	17.50
(0, 0.2] และ [0.8, 1)	88.85	11.15
(0, 0.3] และ [0.7, 1)	90.00	10.00
(0, 0.4] และ [0.6, 1)	88.80	11.20

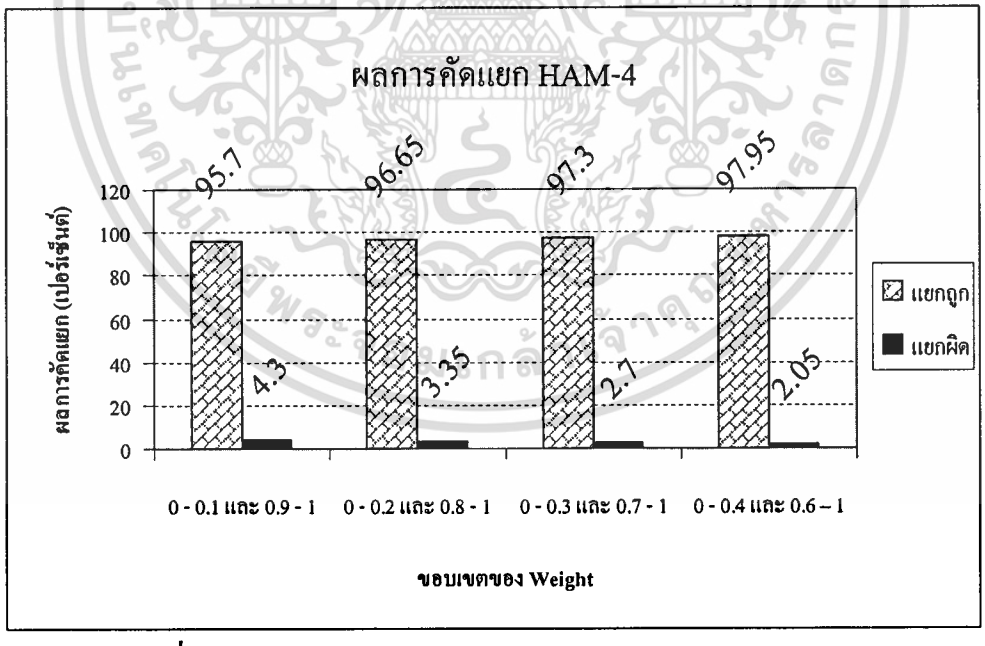
ตารางที่ 4.4 แสดงผลการคัดแยก HAM-4 ในแต่ละขอบเขตของ Weight

ผลการคัดแยก HAM-4		
ขอบเขต Weight	ผลการคัดแยก (%)	
	แยกถูก	แยกผิด
(0, 0.1] และ [0.9, 1)	95.70	4.30
(0, 0.2] และ [0.8, 1)	96.65	3.35
(0, 0.3] และ [0.7, 1)	97.30	2.70
(0, 0.4] และ [0.6, 1)	97.95	2.05

จากตารางที่ 4.3 และตารางที่ 4.4 นำมาเขียนเป็นกราฟดังรูปที่ 4.3 และ รูปที่ 4.4



รูปที่ 4.3 แสดงความถูกต้องในการคัดแยก SPAM-4 ในแต่ละช่วง



รูปที่ 4.4 แสดงความถูกต้องในการคัดแยก HAM-4 ในแต่ละช่วง

จากรูปที่ 4.3 และรูปที่ 4.4 แสดงผลการคัดแยกคิดเป็นเปอร์เซ็นต์เมื่อขอบเขตของ SPAMCITY ที่กำหนดมีการปรับ 4 ระดับ พบว่าในระยะที่ 0- 0.3 และ 0.7 – 1 เป็นช่วงที่เหมาะสมในการนำไปใช้คัดแยกเนื่องจากมีประสิทธิภาพในการคัดแยก SPAM และ HAM สูง ถึงแม้ว่า ในช่วงที่ 0 – 0.4 และ

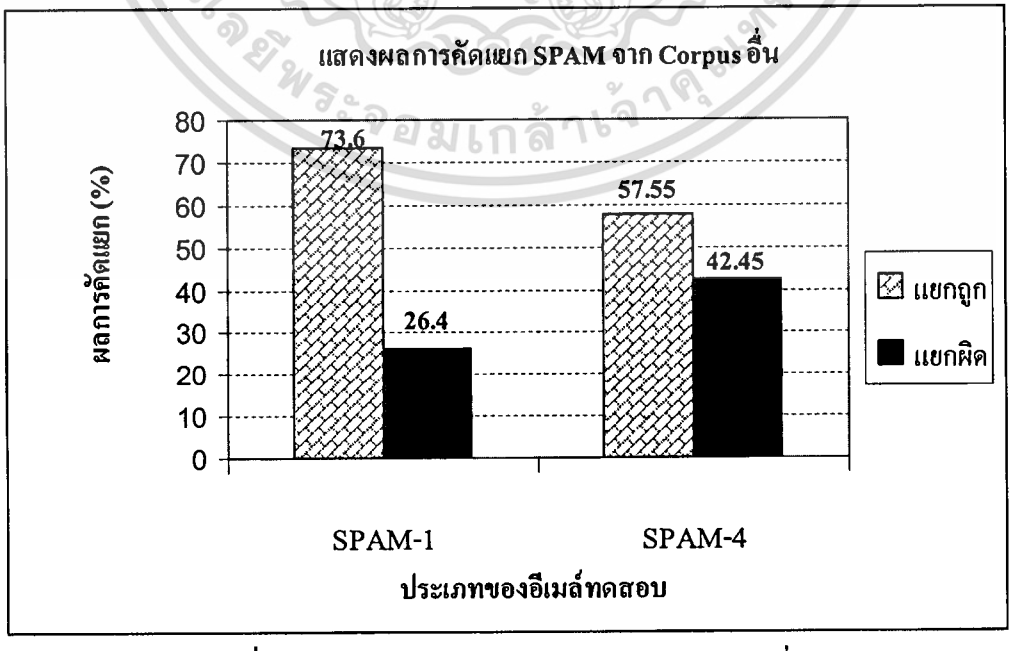
0.6 – 1 มีประสิทธิภาพในการคัดแยก HAM สูงกว่า แต่เนื่องด้วยงานพิจารณาการลดการคัดแยก SPAM ผิดเป็นหลักจึงควรใช้ช่วง 0 - 0.3 และ 0.7 – 1 จะมีประสิทธิภาพมากกว่า

ทดลองคัดแยกสแปมจากข้อมูลในข้อ 4.2.2 ข้อย่อย b. ที่ SPAMCITY ในช่วง 0 < N <= 0.3 และ 0.7 <= N < 1 บนประเภท SPAM-1 และ SPAM-4 เพื่อหาความแตกต่างของการคัดแยกสแปมจาก Corpus อื่นกับการทดลองคัดแยกสแปมใน Corpus ที่ทำ Preprocessing แสดงได้ดังตารางที่ 4.5

ตารางที่ 4.5 แสดงผลการคัดแยก SPAM-1 และ SPAM-4 บนอีเมล์ตามข้อ 4.2.2 ข้อย่อย b

ประเภท	คัดแยก	ฉบับ	เปอร์เซ็นต์
SPAM-1	ถูก	1,472	73.6
	ผิด	528	26.4
SPAM-4	ถูก	1,152	57.55
	ผิด	849	42.45

จากผลการคัดแยกในตารางที่ 4.5 แสดงผลได้ว่าเมื่อนำอีเมล์จาก Corpus อื่นๆ ที่ไม่เกี่ยวข้องกันกับ Copus ที่ใช้ในการทำ Preprocessing นี้จะทำให้ผลการคัดแยกลดลงแสดงผลเป็นกราฟได้ดังรูปที่ 4.5



รูปที่ 4.5 แสดงผลการคัดแยก SPAM จาก Corpus อื่น

จากรูปที่ 4.5 แสดงให้เห็นว่าการคัดแยก SPAM จากอีเมล์อื่นที่ไม่ได้อยู่ใน Corpus ที่ใช้ในการทำ Preprocessing นั้นจะทำให้ผลการคัดแยก SPAM ได้ถูกต้องลดลง ทั้งนี้จากการทดลองพบว่าการพิจารณาอีเมล์เฉพาะบางส่วนของ Header มีการคัดแยกผิดพลาดมากขึ้นจากเดิมที่เคยทดลองกับ Corpus เดียวกันกับที่ทำ Preprocessing 88% ลดลงเหลือ 57% โดยประมาณ แสดงให้เห็นว่าความน่าเชื่อถือในการคัดแยกเฉพาะบางส่วนของ Header นั้นมีน้อยลงจนแทบจะใช้ไม่ได้ ในการคัดแยกอีเมล์ใหม่ จำเป็นต้องเลือกพิจารณาอีเมล์ที่เป็นข้อความแบบ Text ทั้งฉบับ เพื่อคงความถูกต้องในการคัดแยกไว้ แม้ความถูกต้องในการคัดแยกของอีเมล์ใหม่ที่ไม่เกี่ยวข้องกับข้อมูลอีเมล์ที่ทำการ Preprocessing เลย แต่ก็ยังคงความถูกต้องในการคัดแยกไว้อย่างน้อยประมาณ 70%

4.6 สรุปผลการทดลอง

จากการทดลองสรุปได้ว่าจำนวนคำที่นำมาใช้ในการพิจารณามีผลต่อการคัดแยก การเลือกคำหรือกลุ่มคำที่มีค่าอำนาจจำแนกสูงจะสามารถลดเวลาในการคัดแยกและยังคงประสิทธิภาพไว้ได้หรือลดลงเพียงเล็กน้อยเท่านั้น

ในการคัดแยกด้วยอัลกอริทึม Bayesian จากการทดลองสามารถคัดแยก SPAM ที่มาจาก Corpus เดียวกันกับที่นำไปเตรียมข้อมูล (Preprocessing) เฉลี่ยถูกต้อง 90% และมีความถูกต้องลดลงเมื่อทดลองคัดแยก SPAM ที่มาจาก Corpus อื่น คาดว่าเราสามารถปรับในเรื่องของประสิทธิภาพการคัดแยกให้เพิ่มขึ้นได้อีกโดยพิจารณาถึงวิธีเลือกคำที่ดีในการคัดแยก

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

การคัดแยกประเภทของอีเมล์โดยใช้อัลกอริทึม Bayesian นั้นให้ผลในการลดปัญหา False Positive เป็นที่น่าพอใจคัดแยก SPAM ถูกต้องประมาณ 90% บน Corpus เดียวกัน แต่ถ้าอีเมล์ที่มาจาก Corpus อื่นคัดแยกถูกต้องลดลงเหลือประมาณ 70% โดยประมาณ แต่ต้องใช้เวลาในการพิจารณามาก เนื่องจากการพิจารณาที่ได้ผลถูกต้องมากที่สุดต้องพิจารณาจากข้อความแบบ Text ทั้งหมดของอีเมล์นั้น หากสามารถลดเวลาในการพิจารณาแยกประเภท และสามารถทำงานได้ดีคงที่ หรือดียิ่งขึ้น โดยพัฒนากระบวนเลือกคำหรือขอบเขตที่ใช้ในพิจารณาที่เหมาะสมของอีเมล์ให้มีความรวดเร็วและถูกต้องมากยิ่งขึ้น จะได้รับความนิยมเพิ่มขึ้นอย่างแน่นอน

5.2 ข้อเสนอแนะ

5.2.1 แนวทางการประยุกต์ใช้งาน

โปรแกรมที่จัดทำได้จัดทำในลักษณะเป็น local mail server ที่สามารถคัดแยก SPAM เมล์สู่เมล์โฟลเดอร์ของผู้รับ (mail box) โดยใช้อัลกอริทึม Bayesian สามารถนำไปเพิ่มในส่วนของ GUI ด้านไคลเอ็นต์เพื่อใช้ในการเข้าถึงอีเมล์ของผู้รับนั้นๆ ในองค์กรได้ หรือจะปรับให้เป็นเมล์ไคลเอ็นต์สำหรับผู้คนเดียว

5.2.2 ข้อจำกัดของโครงงานพิเศษนี้

5.2.2.1 ขนาดของอีเมล์มีผลต่อความเร็วในการทำงาน

อีเมล์ที่มีขนาดใหญ่จะใช้เวลาในการพิจารณาแยกคำ การตัดคำที่ซ้ำซ้อนเพื่อใช้ทำเป็นคีย์เวิร์ด และถ้ากรณีที่ใช้เวลานานที่สุดคืออีเมล์ที่จำนวนคีย์เวิร์ดที่ไม่ซ้ำกันเลยหรือมีจำนวนเข้าใกล้กับขนาดของอีเมล์ จะทำให้ใช้เวลานานในการนำคีย์เวิร์ดไปเทียบในฐานข้อมูลเพื่อหาค่า SPAMCITY ทีละตัวจนครบ หากมี 500 ตัว จากเงื่อนไขของสิ่งแวดล้อมของฮาร์ดแวร์ในข้อ 4.4 จะพบว่าใช้เวลาประมาณ 2 นาทีขึ้นไป ในการสืบค้นค่า SPAMCITY ทั้งหมด เพื่อนำมาใช้ในการคำนวณหาความน่าจะเป็น

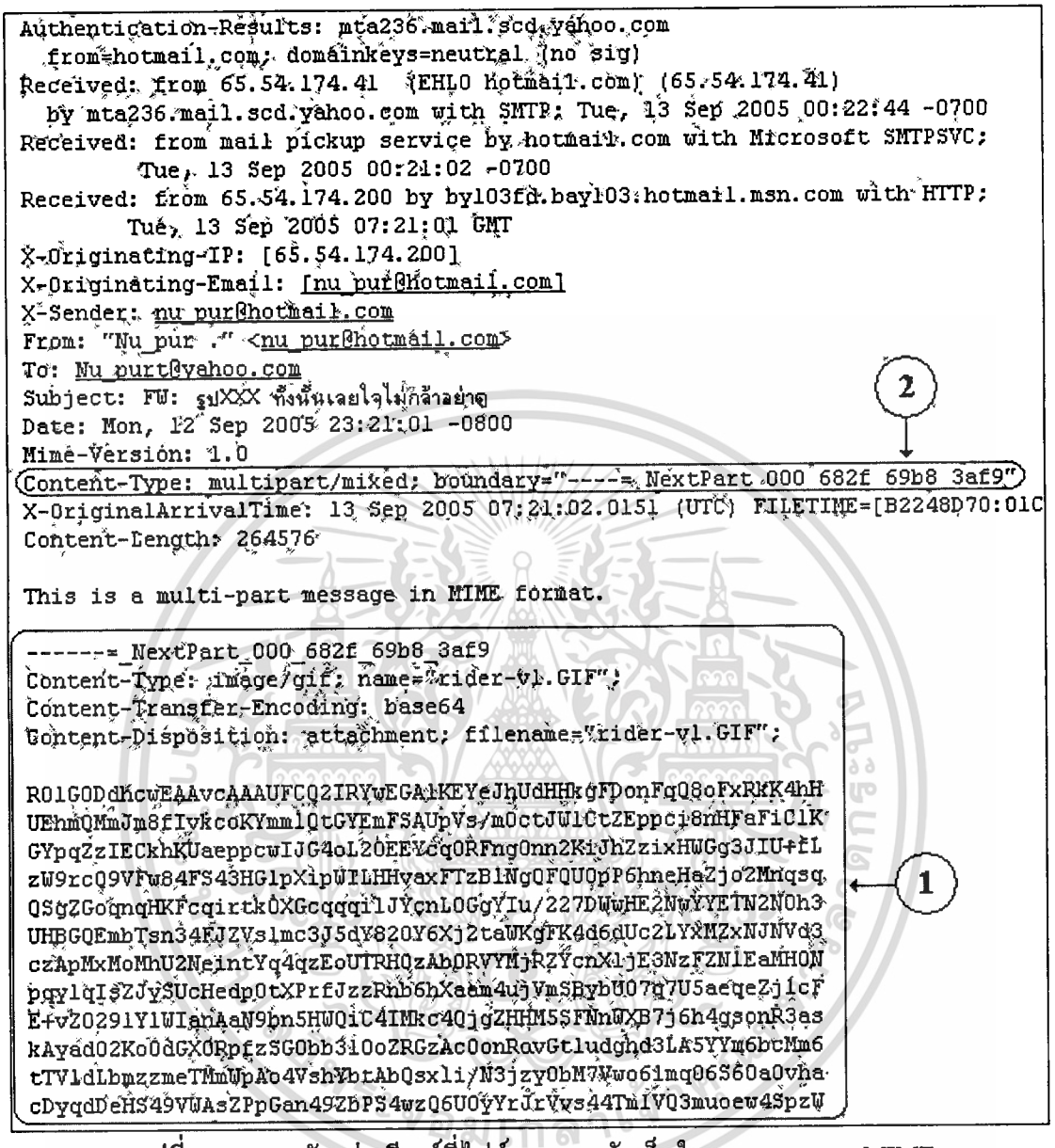
5.2.2.2 การพิจารณาอีเมล์ที่การ Unicode ด้วยรหัสที่ต่างกัน

อีเมล์จะถูกแปลงให้อยู่ในรูปของ text ก่อนรับและส่ง ใน PC แต่ละการแปลงอีเมล์ให้อยู่ในรูปแบบไฟล์ text จะอ้างอิงกับ Unicode ที่เครื่องนั้นระบุไว้ ในกรณีที่โปรแกรม PANBOT ถูกติดตั้งไว้ที่ PC ที่ใช้ Unicode ไม่ตรงกับอีเมล์นั้น จะพบปัญหาไม่สามารถแปลงตัวอักษรได้ถูกต้องตรงกัน เช่น ในเครื่องของผู้ส่งแสดง a passes the station แต่เมื่อรับ □passes the station ทำให้มีผลต่อการแยกคีย์เวิร์ดและวิเคราะห์ความน่าจะเป็นของเอกสารทำให้ส่งผลให้เกิดความคลาดเลื่อนในการวิเคราะห์ได้

5.2.2.3 ปัญหาที่มาจาก Forward อีเมล์

ในอีเมล์ที่มีไฟล์แนบมาด้วยจะถูกแปลงไฟล์ที่แนบมานั้นให้อยู่ในรูปแบบของ Text ก่อนตามมาตรฐาน Multipurpose Internet Mail Extensions (MIME) ดังรูปที่ 5.1 แสดงตัวอย่างอีเมล์ที่มีไฟล์แนบมาด้วยดูได้จากในส่วนที่ 1 ในรูปจะเป็นไฟล์รูปภาพชื่อ rider-V1.GIF ที่แปลงอยู่ในรูปแบบของ Text เรียบร้อยแล้ว ไฟล์ที่แนบมานี้จะถูกหั่นออกเป็นส่วนๆ โดยมี boundary ในส่วนที่ 2 ของรูปเป็นตัวแบ่ง

ในบางอีเมล์ไคลเอ็นต์ การ forward อีเมล์ จะไม่ตรวจสอบก่อนว่าอีเมล์นั้นมีส่วนใดเป็น Text จริงๆ และส่วนใดเป็น Text ที่อยู่ในรูปแบบของ MIME เมื่อไม่สนใจจึงใส่เนื้อความทั้งหมดในอีเมล์ในลักษณะของ Text ธรรมดาแล้วส่งต่อ ผู้อ่านอาจจะพบเห็น forward อีเมล์ในลักษณะนี้อยู่บ้าง กล่าวคือจะพบว่ามีอีเมล์ที่ส่งต่อมาถึงผู้อ่าน แต่ไม่ได้ถูกแสดงว่าเป็นอีเมล์ที่มีไฟล์แนบเป็นรูปภาพเห็นเป็นอีเมล์ที่เต็มไปด้วยข้อความ Text ที่มีส่วนที่ 1 ในรูปที่ 5.1 ปะปนมาด้วย โปรแกรม PANBOT จะพิจารณาตัดคำเพื่อหาคีย์เวิร์ดนั้นพิจารณาส่วนที่ Forward มาด้วย ทำให้มีจำนวนคีย์เวิร์ดมากขึ้น แต่ไม่มีค่าอำนาจจำแนกพอที่จะพิจารณาได้ว่าอีเมล์ที่มีคีย์เวิร์ดนี้ปรากฏจะเป็น SPAM หรือไม่



รูปที่ 5.1 แสดงตัวอย่างอีเมล์ที่ไฟล์แนบและจัดเก็บในตามมาตรฐาน MIME

5.2.2.4 การคัดแยกอีเมล์ภาษาไทยไม่น่าเชื่อถือ

เนื่องจากรูปแบบการเขียนของภาษาไทยคำแต่ละคำติดกัน เมื่อหมดประโยคจึงเว้นวรรคทำให้ไม่สามารถจะพิจารณาตัดคำได้เหมือนกับในภาษาอังกฤษ เมื่อมีอีเมล์ภาษาไทยเข้ามาก็จะพิจารณาตัดคำตามเงื่อนไขที่กำหนดไว้ในข้อ 3.5.1 ซึ่งก็จะทำให้ได้คีย์เวิร์ดเช่นกันแต่จะได้คำที่ยาวและมีความถี่ที่ซ้ำกันน้อยมากจนถึงไม่ซ้ำ มีผลทำให้การคัดแยกไม่น่าเชื่อถือตามความหมาย SPAM ของผู้ใช้นั้นได้

5.2.3 มุมมองในการพัฒนาต่อ

ในระหว่างการพัฒนาผู้พัฒนาพบปัญหาหลายรูปแบบที่น่าสนใจ เห็นว่ามีความน่าสนใจในการพัฒนา เช่น

- จะทำอย่างไรให้การคัดแยกอีเมล์ที่ใช้ Unicode ต่างกันให้มีความผิดพลาดน้อยที่สุดในการคัดแยก
- ในส่วนหัวข้ออีเมล์จะประกอบไปด้วย Keys ต่างๆ ที่ใช้ในการอ้างถึง values นั้นๆ ตามมาตรฐาน RFC 822 เพื่อการใช้งานของแต่ละเมล์เซิร์ฟเวอร์ เช่น Key – "To:" ใช้อ้างถึง value ที่ส่งถึงผู้รับ เป็นต้น พบว่าในส่วนหัวของอีเมล์นั้นประกอบไปด้วย Keys คำนวณมาก ขึ้นอยู่กับการใช้งานของแต่ละเมล์เซิร์ฟเวอร์ ในการพิจารณาคัดแยก SPAM เมล์ในโปรแกรมนี้นั้นพิจารณาตัว Keys ด้วย ทำให้มีปัญหาว่า Keys เหล่านี้มีผลต่อการคัดแยกมากน้อยเพียงใด มีความความจำเป็นต้องพิจารณาหรือไม่
- ในการคัดแยกในโปรแกรมนี้ได้เลือกคัดแยกเฉพาะส่วนที่เป็น Text นั่นรวมถึงส่วนที่เป็น HTML ด้วย หากการคัดแยกสามารถแยก tag ของ HTML ออกจากการคัดแยกน่าจะส่งผลต่อความแม่นยำในการคัดแยกอีเมล์ที่มีข้อความที่เป็น HTML ประกอบอยู่ด้วย
- ในบางกรณีเชื่อว่าการคัดแยกอาจไม่สามารถแยก SPAM ออกมาโดยพิจารณาเพียงในส่วนของ Text เท่านั้น หากมีวิธีการพิจารณาในส่วนของไฟล์ที่แนบมาด้วยนั้นน่าจะทำให้คัดแยกได้ดียิ่งขึ้น

บรรณานุกรม

อรรยา สิงห์สงบ. 2003. **ความพยายามทางกฎหมายกับการแก้ไขปัญหาจดหมายอิเลคทรอนิสก์ขยะ.** [Online]. Available: http://legalaid.bu.ac.th/files/articles/spammail.pdf.

Androutsopoulos, I. et al. 2000. "An Evaluation of Naive Bayesian Anti-Spam Filtering." **Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML).** 11(1): 9-17.

Baeza-Yates Ricardo, Ribeiro-Neto Berthier. 1999. **Modern Information Retrieval.** Boston: Addison-Wesley.

Garcia D. Flavio. 2004. "SPAM FILTER ANALYSIS." **IFIP TC11 19th International Information Security Conference (SEC2004).** 19(1): 395-410.

Hoffman Paul. 1998. **Unsolicited Bulk Email: Mechanisms for Control.** [Online]. Available: http://www.imc.org/ube-sol.html

Kurose F James, Keith W. Ross. 2004. **Computer Networking: A Top Down Approach Featuring the Internet.** 2nd edition. Boston: Addison-Wesley.

Network World Fusion. **Anti-spam Compare-o-matic.** [Online]. Available: http://www. nwfusion .com/bg/2003/spam/compare.jsp.

Process Software, Inc. **Bayesian Filtering Example.**[Online]. Available: http://process.com

Spamassassin. **SpamAssassin public mail corpus.** [Online]. Available: http://spamassassin. apache.org/publiccorpus/

Wikipedia. 2005. **Email spam.** [Online]. Available: http://en.wikipedia.org/wiki/Email_spam

ประวัติผู้เขียนโครงงาน

ชื่อผู้จัดทำโครงงาน นายธนรัฐ โชติพันธ์

วันเดือนปีเกิด 19 กรกฎาคม 2519

สถานที่เกิด รพ. สรรพสิทธิประสงค์ จังหวัดอุบลราชธานี

อิเล็กทรอนิกส์เมล์ t_chotipun@hotmail.com

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนมูลนิธิวัดศรีอุบลรัตนาราม
มัธยมศึกษาตอนต้น	โรงเรียนอัสสัมชัญอุบลราชธานี
มัธยมศึกษาตอนปลาย	โรงเรียนเบ็ญจะมะมหาราช
อุดมศึกษา	ศึกษาระดับปริญญาตรี ครุศาสตร์บัณฑิต สาขาคอมพิวเตอร์ศึกษา คณะครุศาสตร์ สถาบันราชภัฏสุรินทร์ ในปีการศึกษา 2540