ตัวกำจัดเมล์ขยะ

Spam mail killer

โดย

นายชัยณรงค์ รุจิเสถียรทรัพย์ 44010101

นายณธัช เลี่ยวชวลิต 44010130

อาจารย์ที่ปรึกษา

อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา

ดร. อรัญญา วลัยรัชต์ อาจารย์ที่ปรึกษา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ตัวกำจัดเมล์ขยะ

Spam mail killer

ผู้จัดทำ

1. นายชัยณรงค์ รุจิเสถียรทรัพย์ รหัสประจำตัวนักศึกษา 44010101
2. นายณธัช เลี่ยวชวลิต รหัสประจำตัวนักศึกษา 44010130

..... อาจารย์ที่ปรึกษา

(อาจารย์ ธนา หงษ์สุวรรณ)

..... อาจารย์ที่ปรึกษา

(ดร.อรัญญา วลัยรัชต์)

ตัวกำจัดเมล์ขยะ

นายชัยณรงค์ รุจิเสถียรทรัพย์ รหัส 44010101
นายณธัช เลี่ยวชวลิต รหัส 44010130
อาจารย์ ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ดร. อรัญญา วลัยรัชต์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

ในปัจจุบันนี้การใช้อีเมล์เป็นช่องทางหนึ่งในการสื่อสารที่รวดเร็วและเสียค่าใช้จ่ายน้อย ซึ่งเป็นลู่ทางให้ผู้หวังผลประโยชน์ทางธุรกิจใช้การสื่อสารนี้ในการโฆษณาหรือกระจายข่าวที่ไม่เป็นที่ต้องการของผู้รับที่รู้จักในชื่อว่า อีเมล์ขยะ(Spam Mail) จึงทำให้เกิดการพัฒนาโปรแกรมที่ช่วยกำจัดอีเมล์ขยะ (Spam Mail Killer)

ตัวกำจัดอีเมล์ขยะที่ดีจะต้องสามารถจัดการอีเมล์ขยะได้อย่างถูกต้องและมีประสิทธิภาพ สำหรับโครงงานนี้ได้นำวิธีของ Bayesian ซึ่งเป็นวิธีการคำนวณข้อความทางสถิติมาประยุกต์ใช้ในการสร้างโปรแกรมกำจัดอีเมล์ขยะ

Spam Mail Killer

Mr. Chainarong Rujisatainsap
Mr .Natuch Leochavalit
Mr .Thana Hongsuwan Advisor
Dr. Aranya Warairacht Advsisor

ABSTRACT

In this day , email is used for communication between people because it has low-cost . So there are many people use it to find benefit for them such as advertisement , bring information for user and etc . But the most emial which called spam mail sent by these people , user do not want to receive them . In this way there are many anti-spam software developer create email filtered that can classify spam mail from good email.

The efficient email filtered should classify spam mail from good email correctly . For this project , we use the efficient statistical algorithm which called bayesian to make the spam mail filtered.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ ดร. อรัญญา วลัยรัชต์ และอาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมาในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

นายชัยณรงค์ รุจิเสถียรทรัพย์
นายณชัช เลี่ยวชวลิต

สารบัญ

สารบัญ(ต่อ)

สารบัญตาราง

หน้าที่

สารบัญรูป

หน้าที่

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากในปัจจุบันมีการใช้งานอีเมล์ในการติดต่อสื่อสารกันมากขึ้น ทำให้มีบุคคลบางกลุ่มหรืออาจกล่าวได้ว่าเป็นบุคคลที่ไม่ประสงค์ดีได้ใช้อีเมล์เป็นช่องทางในการในการกระจายข่าวสารที่เขาต้องการสื่อแต่ผู้รับไม่ต้องการไปสู่ผู้รับมากมาย อาจจะเป็นการโฆษณาขายสินค้า หรือว่าข้อความชวนเชื่อต่างๆ ซึ่งการใช้ช่องทางของอีเมล์เป็นช่องทางที่ไม่ต้องลงทุนเยอะ ได้ผลตอบแทนที่สูง และสามารถทำได้ง่าย สิ่งเหล่านี้จะก่อให้เกิดความเบื่อหน่าย เมื่อผู้ใช้งานอีเมล์ต้องคอยมานั่งลบอีเมล์ที่เขาไม่ต้องการจะได้รับและอาจส่งผลถึงความปลอดภัยของข้อมูลของผู้ใช้ ดังนั้นสิ่งที่ผู้ใช้อีเมล์ต้องการก็คือตัวกรองที่มีประสิทธิภาพในการกรองอีเมล์ที่ไม่ต้องการ (Spam Mail)

ดังนั้นหากเราพัฒนาตัวกรองอีเมล์ขยะจะช่วยทำให้ผู้ใช้งานอีเมล์ได้รับความปลอดภัยที่มากขึ้น และไม่ต้องคอยเสียเวลากับการลบอีเมล์ขยะ ซึ่งเมื่อเราต้องการจะสร้างตัวกรองที่มีประสิทธิภาพ เราจำเป็นจะต้องรู้เกี่ยวกับพฤติกรรมของสแปมเมอร์ให้ดี ทั้งวิธีการส่งอีเมล์ของสแปมเมอร์ การใช้ลูกเล่นในอีเมล์และ การที่สแปมเมอร์สามารถล่วงรู้เมล์ของคุณ

จากการศึกษาพบว่าได้มีหลักการที่มีประสิทธิภาพ ในการป้องกันอีเมล์ขยะอยู่หลายหลักการ ซึ่งไม่มีหลักการใดที่ถือว่าได้ผลในการป้องกัน 100 % แต่ในที่นี้ได้ศึกษาแนวทางหลักการที่คิดว่ามีประสิทธิภาพในระดับหนึ่ง ได้แก่ การใช้ทฤษฎีของ Bayesian ซึ่งเป็นทฤษฎีในทางสถิติ และความน่าจะเป็นในการจัดการอีเมล์ขยะ

1.2 วัตถุประสงค์

1.2.1 เพื่อสร้างตัวกรองอีเมล์ที่มีประสิทธิภาพบนเครื่องผู้ใช้งานอีเมล์

1.2.2 เพื่อใช้ในการป้องกันความปลอดภัยทางด้านข้อมูลและทรัพย์สินให้แก่ผู้ใช้งานอีเมล์

1.2.3 เพื่อลดจำนวนของอีเมล์ขยะที่มีอยู่ในระบบ

1.2.4 เพื่อเป็นทางเลือกหนึ่งให้แก่ผู้ใช้งานอีเมล์ได้ใช้งานตัวกรองที่ผลิตขึ้นโดยไม่มีค่าใช้จ่ายใดๆ

1.2.5 เพื่อเป็นแนวทางในการพัฒนาให้แก่ผู้ที่สนใจ และผู้ที่ต้องการจะเพิ่มเติมความสามารถของตัวกรองได้

1.3 ขอบเขตของการพัฒนา

ในการพัฒนาโครงงานนี้ได้มีการสร้างในส่วนของซอฟแวร์ โดยแบ่งออกเป็น 2 ส่วนคือ ส่วนของโปรแกรมที่ใช้ในการติดต่อระหว่างอีเมล์ที่เครื่องผู้ใช้และอีเมล์เซิร์ฟเวอร์ และส่วนที่ใช้สำหรับกรองอีเมล์ขยะ

1.3.1 คุณสมบัติของโปรแกรมที่ใช้ในการติดต่อระหว่างอีเมล์ที่เครื่องผู้ใช้และอีเมล์เซิร์ฟเวอร์

1.3.1.1 สามารถติดต่อกับโปรแกรมอีเมล์ไคลเอนต์และเมล์เซิร์ฟเวอร์ได้

1.3.1.2 สามารถติดต่อเมล์เซิร์ฟเวอร์ที่สนับสนุน POP3

1.3.2 คุณสมบัติของตัวกรองอีเมล์

1.3.2.1 สามารถจัดจำแนกระหว่างอีเมล์ทั่วไป และอีเมล์ขยะได้

1.3.2.2 สามารถให้ผู้ใช้เป็นผู้ร่วมตัดสินใจในการกรองอีเมล์ขยะ

1.3.2.3 สามารถเกิดการเรียนรู้อีเมล์ และปรับปรุงฐานข้อมูลได้

บทที่ 2
แนวคิดและทฤษฎี

แนวคิดและทฤษฎีในบทนี้จะแบ่งเป็นสองส่วนคือ ส่วนทฤษฎีของอีเมล์ และทฤษฎี Bayesian ซึ่งมีรายละเอียดดังต่อไปนี้

2.1 ทฤษฎีที่เกี่ยวกับอีเมล์

อีเมล์ได้เริ่มใช้งานมานานแล้ว ตั้งแต่ยุคของเครื่องเมนเฟรมหรือมินิคอมพิวเตอร์ ซึ่งไอบีเอ็มได้พัฒนาระบบอีเมล์ที่เรียกว่า PROFS ออกมาใช้งาน นอกจากนี้ก็มีบนระบบ UNIX ต่อมาหลายๆค่ายก็ได้พัฒนาระบบอีเมล์ของตนขึ้นมา โดยส่วนใหญ่จะเป็นองค์ประกอบในแอพพลิเคชั่นที่ทำงานบนระบบเครือข่าย เช่น Microsoft Mail และ cc:Mail เป็นต้น ซึ่งต่างก็ใช้เทคโนโลยีของตนเองและเป็นระบบปิด ดังนั้นการส่งเมล์ไปยังอีกผู้ใช้ที่ใช้ระบบเมล์คนละค่ายกันจึงเป็นเรื่องยุ่งยาก

ในยุคต่อมาที่ระบบเครือข่ายทั้ง LAN และ WAN ต่างมีมาตรฐาน และเป็นระบบเปิดมาขึ้น ก็ได้ปรับเปลี่ยนการทำงานของอีเมล์มาเป็นแบบไคลเอนต์เซิร์ฟเวอร์ที่เป็นพื้นฐานแบบที่ใช้กันในระบบ UNIX และมีการพัฒนาอีเมล์เซิร์ฟเวอร์ขึ้นมาโดยเฉพาะ เช่น Exchange Server หรือ Note Server เป็นต้น ซึ่งผู้ใช้จะติดต่อเข้าสู่อีเมล์เซิร์ฟเวอร์ได้ทั้งโดยการผ่านระบบ LAN หรือใช้โมเด็มเข้ามาจาก WAN ทำให้ผู้ใช้จะไม่เห็นไฟล์ในฮาร์ดดิสก์บนเซิร์ฟเวอร์เลย ดังนั้นความปลอดภัยของระบบจึงมีมากขึ้น จนในปัจจุบันได้พัฒนาขึ้นมาเป็นระบบ Workflow ที่ใช้อีเมล์เป็นพื้นฐาน

การทำงานของอีเมล์ไคลเอนต์และอีเมล์เซิร์ฟเวอร์มีส่วนประกอบดังนี้

- User Agent เป็นโปรแกรมทางด้านผู้ใช้งาน แบ่งเป็น 2 ส่วนคือ ส่วนของผู้ส่งและส่วนของผู้รับ โดย User Agent จะติดต่อเข้าสู่เซิร์ฟเวอร์ของตนโดยผ่านระบบ LAN หรือ Dial-up ซึ่งในส่วนของ User Agent นี้จะเป็นส่วนที่ผู้ใช้ติดตั้งโปรแกรม Email Client เพื่อเรียกใช้บริการอีเมล์ เช่น Outlook Express , Microsoft Outlook , หรือ Eudora เป็นต้น

- MTA (Mail Transfer Agent) เป็นโปรแกรมที่ส่งอีเมล์จากต้นทางไปยังผู้รับปลายทาง ซึ่งจะต้องผ่านเครื่องจำนวนมากที่เชื่อมต่อกันในเครือข่าย โดยโปรแกรมเหล่านี้จะช่วยกันส่งต่ออีเมล์เป็นทอดๆจนไปถึงเครื่องที่มี account หรือเมล์บ็อกซ์ของผู้รับ และหากไม่สามารถส่งอีเมล์ถึงผู้รับได้ ยังทำหน้าที่ส่ง erroer mail กลับมายังผู้ส่งได้อีก ซึ่งเครื่องที่มี MTA อยู่มักจะมีเมล์บ็อกซ์ของผู้ใช้อยู่ด้วย ซึ่งเป็น primary mailbox และเรียกเครื่องนี้ว่า Mail Servar

2.1.1 โปรโตคอลและประเภทการใช้งาน

การทำงานทั่วๆไปของอีเมล์โดยสรุปมีเพียง 2 ประเภทคือ การส่งอีเมล์ และการรับอีเมล์ โดยโปรโตคอล SMTP (Simple Mail Transfer Protocol) จะใช้ขณะที่ User Agent ส่งอีเมล์มาที่ MTA (เฉพาะแบบ Offline) และใช้ขณะรับและส่งอีเมล์ระหว่าง MTA ด้วยกัน สำหรับการใช้อีเมล์แบบ Offline คือเครื่องที่ผู้ใช้ใช้อ่านเมล์ไม่ได้ต่อกับเครื่องที่มีเมล์บ็อกซ์ตลอดเวลา อาจเลือกดาวน์โหลดเมล์มาเก็บไว้ที่เครื่องของตัวเอง โปรโตคอลสำหรับการรับอีเมล์ที่ใช้งานกันอย่างแพร่หลายมีอยู่ 2 แบบคือ โปรโตคอล POP (Post Office Protocol) และ IMAP (Internet Message Access Protocol) ซึ่งจะทำหน้าที่ดาวน์โหลดหรืออัพโหลดอีเมล์จากเครื่องผู้ใช้ไปยังเครื่องที่มี MTA อยู่

รูปแบบของข้อมูลที่ใช้ในโปรโตคอลต่างๆของอีเมล์ถูกกำหนดไว้ใน RFC 822 ซึ่งแบ่งส่วนประกอบภายในอีเมล์เป็น 2 ส่วนคือ ส่วนที่เป็นจ่าหน้าอีเมล์ และข้อมูลอีเมล์ ในส่วนของจ่าหน้าอีเมล์นี้มีไว้เป็นข้อมูลเพื่อให้ส่งไปถึงผู้รับ รูปแบบของข้อมูลจะเป็นข้อความหรือเท็กซ์ นำหน้าด้วยคำสำคัญ เช่น From , To โดยแต่ละบรรทัดจะปิดท้ายด้วย Carriage Return และ/หรือ Line Feed ขึ้นอยู่กับระบบปฏิบัติการที่ใช้

ส่วนข้อมูลของอีเมล์จะแบ่งเป็น 2 ส่วนคือ ส่วนหัว (Header) และส่วนเนื้อหาของอีเมล์ ส่วนหัวนี้จะถูกสร้างขึ้นโดยอัตโนมัติโดย User Agent ของผู้ส่ง เพื่อให้ MTA ต่างๆระหว่างทางที่ส่งผ่านอีเมล์ฉบับนั้นได้อ่านไปใช้งาน ซึ่งประกอบด้วยข้อมูลต่างๆหลายประเภท เช่น Message Header , วันที่และเวลาที่ส่ง เป็นต้น ส่วนที่เป็นเนื้อหาของอีเมล์นั้นจะเป็นบรรทัดที่อยู่แยกจากส่วนหัว โดยถูกคั่นด้วยบรรทัดว่าง และในแต่ละบรรทัดของข้อความจะสิ้นสุดบรรทัดด้วย Carriage Return และ/หรือ Line Feed

ตามข้อกำหนดของ RFC 822 ในการส่งอีเมล์ผ่านอินเตอร์เน็ตนั้น แต่ละบรรทัดจะมีขนาดยาวได้ไม่เกิน 1000 ไบต์ และขนาดของอีเมล์แต่ละครั้งจะไม่เกิน 64 กิโลไบต์ ซึ่งผู้ส่งไม่จำเป็นต้องสนใจว่าอีเมล์ที่ส่งไปนั้นจะผ่านไปที่ MTA ใดบ้าง เนื่องจากอีเมล์จะถูกเข้ารหัสและส่งไปถึงยัง User Agent ของผู้รับปลายทางและผ่านการถอดรหัสโดยอัตโนมัติ

การใช้งานอีเมล์ในปัจจุบันซึ่งทำงานแบบไคลเอนต์เซิร์ฟเวอร์สามารถทำงานได้ 3 แบบคือ

- แบบ Offline หรือเรียกว่า Download and Delete ซึ่งเป็นรูปแบบมาตรฐานทั่วไปในการใช้งานกับอีเมล์ของอินเตอร์เน็ต ซึ่งใช้โปรโตคอล POP หรือ IMAP โดย User Agent ของผู้รับจะดาวน์โหลดอีเมล์ทั้งหมดมาจากเมล์เซิร์ฟเวอร์และลบอีเมล์เหล่านั้นออกไป (ในโปรแกรมไคลเอนต์ของอีเมล์บางโปรแกรมสามารถให้เลือกได้ว่าต้องการลบอีเมล์ที่ดาวน์โหลดมาแล้วทางเซิร์ฟเวอร์นั้นทิ้งหรือไม่) ทำให้ผู้ใช้สามารถอ่านอีเมล์นั้นได้ตลอดเวลาโดยไม่ต้องติดต่อเซิร์ฟเวอร์อีก แต่ User Agent จะไม่รู้ว่ามีอีเมล์เข้ามาใหม่จนกว่าจะติดต่อเข้าไปยังเมล์เซิร์ฟเวอร์และดาวน์โหลดอีเมล์มาใหม่

- แบบ Online เป็นแบบที่อีเมล์ค้าน User Agent ของผู้รับจะต้องติดต่อกับเมล์เซิร์ฟเวอร์ของผู้รับเองตลอดเวลาที่ใช้อีเมล์ ซึ่งระบบที่ให้บริการอีเมล์แบบนี้จะสามารถเปิดแชร์เมล์บ็อกซ์ที่เซิร์ฟเวอร์ได้ตลอดเวลา เช่น NFS (Network File System) หรือ CIFS (Common Internet File System)

- แบบ Disconnected เป็นแบบผสมผสานระหว่างแบบ Offline และแบบ Online โดยอาศัยเมล์เซิร์ฟเวอร์ของผู้รับเป็นที่หลักในการจัดเก็บข้อมูลของอีเมล์ และในส่วนเนื้อที่ของ User Agent นี้จะเป็นที่เก็บอีเมล์สำรอง โดยเมื่อมีการดาวน์โหลดอีเมล์มาก็จะทำงานในแบบของ Offline เพื่อลดภาระที่ต้องติดต่อกับเมล์เซิร์ฟเวอร์ตลอดเวลา แต่ข้อมูลอีเมล์จะไม่ถูกลบออกจากเมล์เซิร์ฟเวอร์ ผู้ใช้สามารถโหลดอีเมล์ที่แก้ไขแล้วกลับไปยังเมล์เซิร์ฟเวอร์ในภายหลังได้ เช่น การแก้ไขหรือตอบกลับอีเมล์ที่ส่งมา เป็นต้น ซึ่งโปรโตคอลที่สามารถตอบสนองการใช้งานแบบนี้ได้ก็คือ IMAP

2.1.2 POP3 (Post Office Protocol : RFC 1939)

โปรโตคอลของ POP3 นี้จะทำงานในแบบของไคลเอนต์เซิร์ฟเวอร์ คือมีโปรแกรม POP Server ในเมล์เซิร์ฟเวอร์ และ POP Client ในเครื่องผู้รับ ซึ่งปกติจะฝังอยู่ในโปรแกรมที่เป็น User Agent เลย โปรแกรมทั้งสองจะติดต่อกันโดยใช้ชุดคำสั่งที่เป็นรหัส ASCII คือเมื่อด้านที่รับทำคำสั่งก็จะทำงานตามคำสั่งนั้น แล้วตอบกลับมาโดยมีค่าเป็น +OK หมายถึงทำงานได้เรียบร้อย หรือ –ERR หมายถึงเกิดปัญหาขึ้นทำงานไม่ได้ ซึ่งในคำสั่งที่ต้องมีการตอบกลับและส่งข้อมูลกลับมา โดยประกอบด้วยข้อมูลหลายๆบรรทัดนั้น POP3 จะให้บรรทัดสุดท้ายเป็นเครื่องหมายจุด (.) ตามด้วย Carriage Retum และ Line Feed หมายถึงการสิ้นสุดข้อมูล แต่ในกรณีที่ข้อมูลบรรทัดสุดท้ายมีข้อมูลที่เป็นจุดด้วยจะใช้เทคนิคที่เรียกว่า Character Stuffing เพื่อแก้ปัญหา โดยจะเติมจุดลงไปอีกหนึ่งตัวเพื่อเป็นตัวบ่งชี้ว่าข้อมูลนั้นเป็นจุด ซึ่งจะแตกต่างจากสัญลักษณ์แสดงการสิ้นสุดของข้อมูล

การทำงานของ POP3 จะทำร่วมกับโปรโตคอล TCP โดยทั่วไปจะใช้พอร์ต 110 ในการติดต่อ ขั้นตอนการทำงานของ POP3 ประกอบด้วย 3 สถานะคือ สถานะขออนุมัติ สถานะรับส่งรายการ และสถานะปรับปรุงข้อมูล ซึ่งในแต่ละสถานะจะรับรู้คำสั่งต่างๆของโปรโตคอลแตกต่างกัน โดยมีรายละเอียดต่างๆดังนี้

- สถานะขออนุมัติ (Authorization State) เมื่อเริ่มต้นติดต่อกับเซิร์ฟเวอร์จะเป็นการเข้าสู่สถานการณ์ขออนุมัติ โดยไคลเอนต์จะต้องแจ้งชื่อผู้ใช้และรหัสผ่าน เพื่อขออนุมัติจากเซิร์ฟเวอร์ก่อน โดยไคลเอนต์จะใช้คำสั่ง USER เพื่อระบุชื่อผู้ใช้ หรือคำสั่ง PASS เพื่อกำหนด Password แต่ในกรณีที่ชื่อและ Password ถูกเข้ารหัสไว้ และไม่ได้เป็นค่า ASCII ทั่วไป ไคลเอนต์จะใช้คำสั่ง APOP ทำงานแทนคำสั่ง USER และ PASS

- สถานะรับส่งรายการ (Transaction State) หลังจากที่ได้รับอนุมัติจากเซิร์ฟเวอร์แล้ว ก็จะเข้าสู่สถานะที่ใช้คำสั่งในการทำงานต่างๆ

- สถานะปรับปรุงข้อมูล (Update State) เมื่อ User Agent เลิกใช้งานด้วยคำสั่ง QUIT ของ POP3 เซิร์ฟเวอร์ก็จะเข้าสู่สถานะปรับปรุงข้อมูล เพื่อลบอีเมล์ที่ดาวน์โหลดเรียบร้อยแล้วออกไป จากนั้นก็จะเข้าสู่สถานะขออนุมัติใหม่โดยอัตโนมัติ เพื่อรอรับการทำงานครั้งต่อไป

คำสั่ง	พารามิเตอร์	สถานะ	รายละเอียด
USER	ชื่อผู้ใช้งาน	ขออนุมัติ	แจ้งชื่อผู้ใช้ และระบุเมล์บ็อกซ์ที่จะใช้
PASS	Password	ขออนุมัติ	ใช้ระบุ Password โดยจะใช้ต่อจากคำสั่ง USER
APOP	ชื่อ , Password	ขออนุมัติ	ทำหน้าที่เหมือนคำสั่ง USER และ PASS รวมกัน แต่ข้อมูลถูกเข้ารหัสก่อนส่งไป
STAT	ไม่ระบุ	รับส่งรายการ	ใช้ตรวจสอบสภาพเซิร์ฟเวอร์ เช่น จำนวนอีเมล์ในเซิร์ฟเวอร์ ขนาดของอีเมล์ที่จะดาวน์โหลด
UIDL	หมายเลขข้อความ	รับส่งรายการ	ใช้ตรวจสอบหมายเลขประจำของอีเมล์
LIST	หมายเลขข้อความ	รับส่งรายการ	ใช้ตรวจสอบหมายเลขของอีเมล์ และขนาดของอีเมล์
RETR	ข้อความ	รับส่งรายการ	เป็นคำสั่งที่ใช้ส่งข้อมูลของอีเมล์
DELE	ข้อความ	รับส่งรายการ	เป็นการระบุเครื่องหมายการลบลงในอีเมล์ที่จะลบ และอีเมล์เหล่านั้นจะถูกลบออกจากเมล์บ็อกซ์เมื่อใช้คำสั่ง

			QUIT เมื่อสิ้นสุดการทำงาน
RSET	ไม่ระบุ	รับส่งรายการ	คำสั่งนี้จะยกเลิกเครื่องหมายการลบอีเมล์ ที่เคยกำหนดไว้ด้วยคำสั่ง DELE ออกไปทุกๆอีเมล์
TOP	หมายเลขข้อความ , จำนวนบรรทัด	รับส่งรายการ	เซิร์ฟเวอร์จะส่งข้อมูลย้อนกลับไปเท่ากับจำนวนบรรทัดที่กำหนด
NOOP	ไม่ระบุ	รับส่งรายการ	เป็นคำสั่ง No Operation
QUIT	ไม่ระบุ	รับส่งรายการและขออนุมัติ	ใช้เมื่อจบการทำงาน หากมีอีเมล์ซึ่งทำเครื่องหมายว่าจะลบไว้ อีเมล์เหล่านั้นจะถูกลบจากเมล์บ็อกซ์ในขั้นตอนนี้

ตารางที่ 2-1 รายละเอียดในคำสั่งต่างๆของ POP3

2.1.3 การเข้ารหัสและ MIME (RFC 1341)

ในการรับส่งอีเมล์ผ่านเครือข่ายนั้น คอมพิวเตอร์ที่เชื่อมต่ออยู่ในเครือข่ายมักจะมีหลากหลายชนิด ดังนั้นข้อมูลที่ส่งผ่านจึงจะต้องเป็นข้อมูลที่อยู่ในรูปแบบกลางๆ ซึ่งคอมพิวเตอร์จะรับรู้และเข้าใจได้เหมือนกัน เพื่อไม่ให้ข้อมูลที่รับและส่งเหล่านั้นผิดเพี้ยนไปจากความจริง และสามารถส่งข้อมูลทั้งที่เป็นข้อความและไม่เป็นข้อความ (เช่น ข้อมูลที่เป็นรูปภาพและเสียง) รวมกันไปในอีเมล์ฉบับเดียวกันได้ ดังนั้นจึงได้นำเทคนิคการเข้ารหัสที่เรียกว่า MIME มาใช้เพื่อเข้ารหัสและถอดรหัสในการรับส่งอีเมล์โดยทั่วไป

เทคนิคของ MIME (Multipurpose Internet Mail Extensions) นี้เป็นเทคนิคที่แปลงรหัส ASCII ทั่วไป ซึ่งมี 8 บิต ให้เป็นค่า 7 บิต (ให้บิตที่ 0 มีค่าเป็น 0 เสมอ) โดยที่เทคนิคของ MIME นี้จะสามารถใช้รับส่งข้อมูลได้ทุกๆแบบ ไม่ว่าจะเป็นข้อมูลของอีเมล์หรือไฟล์ประเภทต่างๆที่แนบไปกับอีเมล์ ซึ่งอีเมล์บนอินเทอร์เน็ตในยุคแรกๆในกรณีที่ต้องการรับส่งข้อมูลที่มีรูปแบบไฟล์แตกต่างไปจากค่า ASCII โดยทั่วไป ผู้ส่งจะต้องแปลงรหัสข้อมูลก่อนส่งด้วยคำสั่ง UUENCODE เพื่อแปลงข้อมูลให้อยู่ในรูปแบบของ MIME ในด้านผู้รับก็จะต้องถอดรหัสข้อมูลกลับมาอยู่ในรูปแบบเดิม โดยใช้คำสั่ง UUDECODE ซึ่งทั้งสองคำสั่งนี้เริ่มพัฒนาขึ้นมาพร้อมกับระบบปฏิบัติการ Unix และภายหลังจึงมีให้แพร่หลายในระบบปฏิบัติการอื่นๆ แต่ในปัจจุบันโปรแกรมที่ทำหน้าที่รับส่งอีเมล์จะทำหน้าที่แปลงและถอดรหัสข้อมูลให้โดยอัตโนมัติ โดยไม่จำเป็นต้องอาศัยคำสั่ง UUENCODE และ UUDECODE อีกต่อไปแล้ว

ลักษณะข้อมูลของ MIME ประกอบด้วย 2 ส่วนคือ ส่วนหัว หรือเรียกว่า Content Transfer Encoding ซึ่งจะเก็บรายละเอียดของไฟล์ที่เข้ารหัสเอาไว้ เช่น ประเภทของไฟล์ เป็นต้น ส่วนที่สองเป็นส่วนของข้อมูลที่เข้ารหัสแล้ว การเข้ารหัสและถอดรหัสของ MIME นี้ถูกระบุไว้ในส่วนหัวเพื่อให้ผู้รับและผู้ส่งเข้าใจตรงกันว่าอีเมล์นี้เข้ารหัสและถอดรหัสด้วยวิธีใด ซึ่งมีอยู่ด้วยกัน 6 วิธี คือ

- **Quoted-Printable** เทคนิคการเข้ารหัสนี้จะแปลงข้อมูลให้อยู่ในลักษณะที่อ่านได้เสมอ ซึ่งหากข้อมูลเป็น ASCII 7 บิตอยู่แล้วก็จะไม่มีการแปลงข้อมูล แต่ถ้าเป็นค่าบิตที่ศูนย์มีค่าเป็น 1 ข้อมูลจะถูกแปลงให้มาอยู่ในรูปค่าของเลขฐาน 16 และนำหน้าด้วยเครื่องหมายเท่ากับ (=) ตัวอย่างเช่น ข้อมูลที่เข้ารหัสแล้วมีค่าเป็น =A1 หมายถึงข้อมูลที่ค่า ASCII เป็น 161 หรือค่า Hex เป็น A1 เป็นต้น

- **Base64** เป็นเทคนิคการเข้ารหัสโดยจะแปลงข้อมูลจำนวน 24 บิต (ข้อมูล 8 บิตจำนวน 3 ไบต์) ออกเป็นข้อมูล 6 บิตจำนวน 4 ชุด โดยหลังจากที่เข้ารหัสแล้ว ข้อมูลจะถูกแปลงให้อยู่ในรูปของตัวอักษร 64 ตัว มี

ค่าตามตาราง Base64 Alphabet แต่ข้อมูลดังกล่าวจะไม่เปลี่ยนแปลงค่าของ Carriage Return และ Line Feed และปิดท้ายข้อมูลด้วยเครื่องหมาย = ซึ่งเรียกว่า PAD

- **Binary** เป็นข้อมูลที่ต่อเนื่องกันเป็นค่าไบต์นารี ไม่แบ่งออกเป็นบรรทัด ซึ่งข้อมูลประเภทนี้จะส่งโดยไม่มีการเข้ารหัสข้อมูล
- **Seven-Bit** เป็นข้อมูลที่มีค่า ASCII 7 บิต ซึ่งข้อมูลประเภทนี้จะส่งโดยไม่มีการเข้ารหัสข้อมูล
- **Eight-Bit** เป็นข้อมูลที่มีค่า ASCII 8 บิต ข้อมูลประเภทนี้จะส่งโดยไม่มีการเข้ารหัสข้อมูล
- **X-Token** เป็นเทคนิคการเข้ารหัสที่ต้องมีการติดต่อและตกลงกันระหว่างด้านผู้ส่งและผู้รับของ SMTP เซิร์ฟเวอร์ก่อน

2.1.4 ความหมายของอีเมล์ขยะ (Spam mail)

อีเมล์ขยะคืออีเมล์ที่ถูกส่งมาจากผู้ที่เรียกว่า Spammer ที่มีลักษณะเหมือนๆกัน และมีการส่งต่อตัวเองไปบนอินเทอร์เน็ตเป็นจำนวนมาก ซึ่งจะส่งกระจายไปให้กับผู้ใช้งานอีเมล์ โดยผู้รับไม่รู้เลยว่าเคยให้ความสนใจกับสิ่งที่อีเมล์ขยะได้ส่งมา อีเมล์ขยะส่วนมากจะเป็นพวกที่เกี่ยวกับธุรกิจ โฆษณา หรือเป็นพวกสื่อลามกต่างๆ ซึ่งจะมีรูปแบบที่ไม่แน่นอน และมีรูปแบบที่ชวนให้เกิดความน่าสงสัย อย่างเช่น โฆษณาชวนเชื่อที่สามารถให้คุณรวยเร็ว หรือเกี่ยวกับภาพโป๊ หรือเรื่องเซ็ก ซึ่งสิ่งที่ทำให้ผู้ส่งต้องการส่งอีเมล์ขยะเหล่านี้ออกมาก็คือต้นทุนที่ใช้ในการโฆษณาเพียงน้อยนิดแต่ผลตอบแทนที่ได้มีมากกว่า แต่ว่าฝ่ายผู้รับจะต้องเสียค่าใช้จ่ายเป็นจำนวนมากเพื่อป้องกันอีเมล์ขยะเหล่านี้

ตัวอย่างบางส่วนที่แสดงถึงปริมาณของอีเมล์ขยะในแต่ละรูปแบบอีเมล์

- 92%: อีเมล์เกี่ยวกับเรื่องทางเพศ เซ็ก และสื่อลามกต่างๆ
- 89%: อีเมล์ที่มีกิจกรรมเกี่ยวกับการเงิน
- 76%: อีเมล์ที่เกี่ยวกับความเชื่อต่างๆ เช่น การเมือง , กฎหมาย และอื่นๆ
- 32%: อีเมล์ที่เกี่ยวข้องกับการค้า การเสนอขายสินค้าต่างๆ

2.1.5 ผลกระทบจากการที่ได้รับอีเมล์ขยะ

2.1.5.1 สิ้นเปลืองเวลาของผู้ใช้ในการจัดการคัดแยกและลบอีเมล์ขยะทิ้ง ผลของอีเมล์ขยะอาจดูไม่ได้ร้ายแรงอะไรถ้าคุณเป็นผู้ใช้ตามบ้าน เพียงแค่ถ้ามีอีเมล์ขยะมาคุณก็แค่ลบเมล์นั้นทิ้งก็ไม่มีอะไรเกิดขึ้น แต่ถ้าเป็นองค์กรธุรกิจ การที่พนักงานต้องคอยมานั่งลบอีเมล์ขยะคงไม่ใช่เรื่องที่น่ายินดีนัก มีผลทำให้การทำงานขององค์กรล่าช้า และส่งผลเสียแก่องค์กรไม่น้อย ตัวอย่างเช่น ถ้าบริษัทหนึ่งให้บริการออนไลน์ แล้วมีการส่งอีเมล์ขยะมาสักวันละ 100000 ฉบับ ซึ่งอาจทำให้เมล์บ๊อกซ์ของบริษัทเต็มส่งผลให้พนักงานแต่ละคนต้องเสียเวลาในการลบเมล์ขยะเมล์ละ 2 วินาที ซึ่งเมื่อมาคิดแล้วเวลารวมที่พนักงานในบริษัทต้องเสียไปกับการลบอีเมล์ขยะถึง 555 ชั่วโมงในการลบหมด

2.1.5.2 สิ้นเปลืองbandwidth ทำให้เมล์เซิร์ฟเวอร์ต้องเสียbandwidth ไปเป็นจำนวนมากพอๆกับขนาดของอีเมล์ขยะนั้น หากbandwidth มีอยู่อย่างจำกัดก็จะส่งให้อีเมล์อื่นที่เข้ามาจะต้องใช้เวลานานกว่าปกติหรืออาจจะไม่ได้รับในที่สุด

2.1.5.3 สิ้นเปลืองการประมวลผลของCPU ที่เมล์เซิร์ฟเวอร์

2.1.5.4 สิ้นเปลืองเนื้อที่ในเมล์บ็อกซ์ ซึ่งจะมีผลกระทบมากหากเมล์บ็อกซ์มีการจำกัดเนื้อที่ของผู้ใช้ หากผู้ใช้ทิ้งไว้ไม่ได้มาตรวจสอบบ่อยๆก็จะทำให้เนื้อที่หมดไปได้ หากเป็นอีเมล์ทางธุรกิจที่สำคัญก็จะทำให้เสียหายต่อธุรกิจได้

2.1.5.5 ผลต่อผู้ให้บริการเซิร์ฟเวอร์ที่มีการตั้งค่าในการrelayไว้ไม่จำกัดกลุ่มที่แน่นอน คือจะทำให้สแปมเมอร์สามารถใช้เซิร์ฟเวอร์นั้นทำการส่งอีเมล์ขยะออกไป ซึ่งเมื่อมีการสืบค้นตอของ MTAก็จะทำให้เซิร์ฟเวอร์นั้นถูกบล็อกทำให้ไม่สามารถส่งอีเมล์ได้

2.2 ทฤษฎี Bayesian

ในอดีตอีเมล์ขยะยังไม่ค่อยจะมีผลกระทบต่อระบบคอมพิวเตอร์มากนัก การกรองอีเมล์ขยะในอดีตจึงมีการใช้กฎเกณฑ์อย่างง่ายๆ ในการจำแนก ซึ่งวิธีการที่ใช้ เช่น การตั้งกฎในการจำแนกรูปแบบของอีเมล์ขยะ (pattern-matching) โดยมีการตั้งกฎที่ระบุว่าอีเมล์ลักษณะไหนที่เป็นอีเมล์ขยะ แต่เนื่องจากในปัจจุบันผู้ส่งอีเมล์ขยะ สามารถส่งอีเมล์ขยะที่มีความสามารถในการหลบเลี่ยงตัวกรอง จึงทำให้วิธีการจำแนกรูปแบบของอีเมล์ขยะเริ่มที่จะไม่มีประสิทธิภาพ เพราะว่าวิธีการนี้เป็นวิธีการที่มีการตั้งกฎเกณฑ์ที่ตายตัว ซึ่งเมื่อผู้ส่งอีเมล์ขยะมีการเปลี่ยนแปลงรูปแบบในการส่งก็จะต้องมีการเพิ่มกฎเข้าไปใหม่เรื่อยๆ ทำให้เกิดความยากลำบากต่อการเปลี่ยนแปลงตามความสามารถในการลบเลี่ยงที่เปลี่ยนไปของผู้ส่งอีเมล์ขยะ จึงมีการคิดค้นวิธีการที่มีความสามารถเพิ่มขึ้นเพื่อกรองอีเมล์ขยะ โดยมีการนำทฤษฎีต่างๆเข้ามาเพื่อใช้การพัฒนาตัวกรองอีเมล์ขยะเพื่อให้มีความสามารถที่เพิ่มขึ้น

หนึ่งในวิธีปัจจุบันที่มีประสิทธิภาพก็คือการนำการวิเคราะห์ทางสถิติเข้ามามีส่วนช่วยในการเรียนรู้ของอีเมล์ ซึ่งเป็นวิธีการที่มีประสิทธิภาพในการจำแนกประเภทของอีเมล์ขยะคือการจำแนกโดยใช้ทฤษฎีBayesian

ตัวกรองอีเมล์ขยะที่มีการพัฒนาขึ้นโดยการใช้ทฤษฎี Bayesian จะมีพื้นฐานของการนำความน่าจะเป็นของอีเมล์นำมาคำนวณ ซึ่งเป็นที่วิธีการในการคัดเลือก keywords เพื่อนำมาใช้ในการคำนวณ โดยการคัดเลือก keywords จะขึ้นกับหลักการในการจำแนกคำ รวมทั้งการตั้งกฎของการจำแนกคำโดยผู้พัฒนาเป็นผู้จัดการ

การใช้หลักการทางสถิติมาใช้ในการสร้างตัวกรองเป็นสิ่งที่ค่อนข้างสมเหตุสมผล เพราะว่าความแตกต่างกันระหว่างอีเมล์ทั่วไป กับอีเมล์ขยะ ค่อนข้างที่จะแยกออกจากกันได้ยาก ซึ่งอีเมล์ขยะทั่วไปจะมีรูปแบบหรือว่าหลักการที่คล้ายคลึงกันกับอีเมล์ทั่วไป ซึ่งสุดท้ายต้องขึ้นกับผู้ใช้เป็นผู้ระบุเองว่าอีเมล์ฉบับไหนที่เป็นอีเมล์ขยะ หรือว่าอีเมล์ทั่วๆไป และผู้ใช้แต่ละคนอาจจะระบุได้แตกต่างกัน ซึ่งอีเมล์ฉบับหนึ่งเป็นอีเมล์ขยะของผู้ใช้คนหนึ่ง แต่ในขณะที่ผู้ใช้อีกคนอาจจะบอกว่าเป็นอีเมล์ปกติ

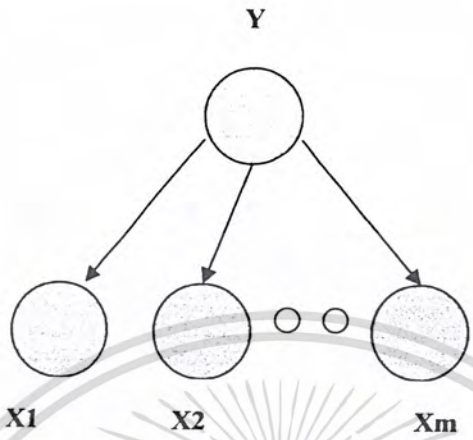
จากการที่ผู้ใช้เป็นผู้กำหนดนิยามของอีเมล์ขยะเองทำให้สังเกตได้ว่าสิ่งหนึ่งที่แตกต่างกันระหว่างอีเมล์ทั่วไปกับอีเมล์ขยะก็คือ เนื้อหาที่อยู่ภายในอีเมล์

วิธีการทางด้านสถิติจะมีข้อเสียก็คือการใช้งานตัวกรองที่สร้างขึ้นจะต้องให้อีเมล์มีการเรียนรู้ในรูปแบบต่างๆ ซึ่งต้องใช้ระยะเวลาหนึ่งการสอนอีเมล์ให้แก่ระบบตัวกรอง

ซึ่งทฤษฎี Bayesian จะมีรายละเอียดดังต่อไปนี้

2.2.1 การจัดจำแนกโดยวิธี Bayesian

Bayesian เป็นวิธีการทางด้านสถิติในการจำแนกสิ่งต่างๆออกเป็นกลุ่มๆ โดยแต่ละกลุ่มจะมีความเป็นอิสระต่อกัน ดังรูป



รูปที่ 2-1 โมเดลการจัดจำแนกของ Bayesian

กำหนดให้แต่ละอีเมล์ถูกแทนที่ด้วยเวกเตอร์ X = (X1,X2,X3,....Xm) โดยแต่ละ X1,X2,...Xm คือค่า multinomial random (ค่า K) แต่ละ Xj จะมีค่า K หนึ่งค่า

ใน class Y เราจะสนใจแค่ binary Bayesian โดยที่ 0 แทนอีเมล์ทั่วไป และ 1 แทนอีเมล์ขยะ

ให้ θ แสดงถึง กลุ่มของตัวแปรสำหรับโมเดล

$$P(x,y|\theta)=P(y|\pi)\prod_{j=1}^{m}P(x_j \mid y,\theta_j) \quad (1)$$

$$P(x \mid Y=i,n)=\prod_{j}\prod_{k}\eta_{ijk}^{x_j^k} \quad (2)$$

x_j^k ตัวแปรที่ใช้ระบุค่า k ที่เป็นของค่า X ในลำดับที่ j

η_{ijk} = P(x_j^k = 1 | Y = i ,n) เป็นความน่าจะเป็นที่มีเงื่อนไข(conditional probability) ซึ่งจะกำหนดให้ Y = i(เป็นได้ทั้ง 0 หรือ 1)

ค่าความน่าจะเป็นที่มาในภายหลัง(posterior probability) เช่น ค่าความน่าจะเป็นของอีเมล์ที่ถูกตัดสินใจว่าเป็นอีเมล์ขยะสำหรับการจำแนกโดยวิธี Bayesian ค่าความน่าจะเป็นที่มาในภายหลังแสดงดังสมการ

$$P(Y=1 \mid x,\eta) = \frac{\pi \prod_j \prod_k \eta_{1jk}^{x_j^k}}{\pi \prod_j \prod_k \eta_{1jk}^{x_j^k} + (1-\pi)\prod_j \prod_k \eta_{0jk}^{x_j^k}}$$

$$= \frac{\exp\{\log\pi + \sum_j \sum_k x_j^k \log\eta_{1jk}}{\exp\{\log\pi + \sum_j \sum_k x_j^k \log\eta_{1jk}\} + \exp\{\log(1-\pi) + \sum_j \sum_k x_j^k \log\eta_{0jk}}$$

$$= \frac{e^{\beta_1^T x}}{e^{\beta_1^T x} + e^{\beta_0^T x}} \qquad (3)$$

ซึ่ง x และ β ถูกนิยามดังต่อไปนี้

$$x = [1 \quad x_1^1 \ldots x_1^K \ldots\ldots x_m^1 \ldots\ldots x_m^K]^T \qquad (4)$$

$$\beta_0 = [\log(1-\pi) \quad \log\eta_{011} \ldots\ldots \eta_{01k} \ldots\ldots \eta_{0m1} \ldots\ldots \eta_{0mk}]^T \qquad (5)$$

$$\beta_1 = [\log\pi \quad \log\eta_{111} \ldots\ldots \eta_{11k} \ldots\ldots \eta_{1m1} \ldots\ldots \eta_{1mk}]^T \qquad (6)$$

ถ้าเราทำการหารเศษและส่วนด้วยตัวเศษของสมการที่ 3 จะได้รูปแบบทางตรรกะของ ค่าความน่าจะเป็นที่มาในภายหลัง (posterior probability) ดังนี้

$$P(Y=1 \mid x, \theta) = 1 \qquad \text{ซึ่ง } \theta = \beta_1 - \beta_0 \qquad (7)$$

สุดท้ายจะได้ กลุ่มของการเทรนนิ่ง D ประกอบไปด้วย จำนวนอีเมล์ N ฉบับ : $D = \{(x_n, y_n); n = 1,\ldots,N\}$ ซึ่งแต่ละแฟลกจะบ่งชี้ว่าเป็นอีเมล์ขยะหรือว่าเป็นอีเมล์ทั่วไป ดังนั้นค่าความน่าจะเป็นของ log จะเป็นดังนี้

$$\iota(\theta \mid D) = \sum_{n=1}^{N} \log P(y_n \mid \pi) + \sum_{n=1}^{N} \sum_{j=1}^{m} \log P(x_{j,n} \mid y_n, \eta) \qquad (8)$$

เมื่อต้องการจะสอนให้ตัวจัดจำแนกเรียนรู้ จะต้องมีการเสนอกลุ่มของข้อมูลสัญลักษณ์ให้แก่ตัวจัดจำแนก ซึ่งกลุ่มของสัญลักษณ์จะขึ้นอยู่กับการคำนวณค่า $\theta = (\pi, \eta)$ ซึ่งค่านี้จะทำให้ค่าความน่าจะเป็นของ log (สมการที่ 8) มีค่าสูงสุด

จากการที่มีการระบุตัวแปร ทำให้ตัวจัดจำแนกมีความพร้อมในการจำแนกอีเมล์ใหม่ที่เข้ามา โดยการจำแนกจะขึ้นกับการคำนวณหาค่าความน่าจะเป็นที่มาภายหลังโดยสมการที่ 7

จากการแตกสมการที่ 8 ออกมา และมีการใช้ Lagrange multipiers (สมการที่ 11) และจะได้ค่าที่มากที่สุดสำหรับตัวแปรดังสมการ

$$\pi = \frac{\sum_{n=1}^{N} y_n^1}{N} \quad (9)$$

$$n_{ijk} = \frac{\sum_{n=1}^{N} x_{j,n}^{k} y_n^i}{\sum_{n=1}^{N} y_n^i} \quad (10)$$

การประมาณค่าของความน่าจะเป็นที่มากที่สุดสำหรับ π เป็นตัวอย่างของอัตราส่วนของอีเมล์ทั้งหมดที่เป็นของคลาส 1 (1 ค่าที่บ่งบอกว่าเป็นอีเมล์ขยะ) และค่าความน่าจะเป็นที่มากที่สุดของ η_{ijk} เป็นตัวอย่างของอัตราส่วนของอีเมล์ในลำดับของคลาสที่ i ซึ่งอ้างถึงลักษณะในลำดับที่ j ที่ขึ้นกับค่าในลำดับที่ k

2.2.2 ตัวกรองอีเมล์โดยวิธี Bayesian

จากการจัดจำแนกโดยวิธี Bayesian พบว่าไม่สมบูรณ์ในการกรองอีเมล์ขยะ จึงควรจะมีการเพิ่มรายละเอียดต่อไปนี้เข้าไป ซึ่งมีอยู่ 2 ขั้นตอน

2.2.2.1 กลุ่มของลักษณะเฉพาะที่ใช้เพื่อแสดงลักษณะของแต่ละอีเมล์ มี 2 วิธี

วิธีที่ 1 การจัดจำแนกโดเมนทั่วๆไปของข้อความ

ในอีเมล์แต่ละอีเมล์คำส่วนมากที่ใช้ในการทำให้ระบบเรียนรู้ จะอยู่ที่ในส่วนของเนื้อความ(body) ในอีเมล์นั้นๆ ในที่นี้จะเสนอลักษณะเฉพาะที่จะได้รับการคัดเลือก

การเลือกลักษณะเฉพาะนี้จะมีการกำหนดค่าเป็น 1 ในกรณีที่อีเมล์มีคำอยู่ในส่วนที่เป็นเนื้อความ(body) และกำหนดค่าเป็น 0 ในกรณีอื่นๆของอีเมล์

เราจะคำนวณหาค่า mutual information(MI) สำหรับแต่ละลักษณะเฉพาะ(X) ดังนี้

$$MI(X;Y) = \sum_{x\in\{0,1\},y\in\{0,1\}} P(X=x,Y=y).\log\frac{P(X=x,Y=y)}{P(X=x).P(Y=y)} \quad (11)$$

ข้อดีของการเลือกกลุ่มลักษณะเฉพาะโดยวิธีนี้คือการเข้าถึงแบบอัตโนมัติสามารถทำได้ง่าย , ลักษณะเฉพาะที่ถูกเลือกและค่า MI สามารถเก็บไว้ได้และสามารถทำการเพิ่มเติมข้อมูลที่เข้ามาใหม่ๆได้ และรูปแบบใหม่ๆของอีเมล์ทั่วไป และอีเมล์ขยะ สามารถเรียนรู้ได้แบบอัตโนมัติโดยตัวกรองได้

วิธีที่2 การจัดจำแนกโดยระบุโดเมนที่เข้าใจอยู่แล้ว

วิธีที่สองนี้จะต่างจากแบบแรกตรงที่ แบบแรกเป็นการเลือกกลุ่มของโดเมนทั่วๆไป แต่แบบที่สองจะมีการระบุว่าต้องการจะกระทำกับโดเมนในกลุ่มนั้นยังไง

สังเกตว่ารูปแบบโดยทั่วไปของอีเมล์ขยะจะมีความแตกต่างกับอีเมล์ที่ดี ซึ่งอาจจะมีหลักเกณฑ์ในการแยกได้หลากหลายแบบ ได้แก่

ผู้ส่งอีเมล์ขยะต้องการจะขายของบางอย่าง และส่วนมากจะทำการส่งข้อมูลของราคามากับอีเมล์ ซึ่งจากการที่ต้องการจะขายของอีเมล์จะมีรูปแบบที่ทำให้ดูน่าสนใจ โดยการใช้ HTML หรือว่ามีการใส่ url เพื่อให้ผู้อ่านได้เข้าไปดู

การที่ผู้ส่งอีเมล์ขยะต้องการจะทำให้ถูกต้องตามกฎในการส่งอีเมล์เพื่อที่จะสามารถลบเลี่ยง แล้วถูกมองว่าเป็นอีเมล์ที่ถูกต้อง ก็โดยการที่มีการทำให้รูปแบบอีเมล์มีรูปแบบที่สมบูรณ์แบบ เช่น การที่ผู้ส่งอีเมล์ขยะทำให้เป็นรูปแบบอีเมล์ที่สมบูรณ์โดยการหลอกใช้อีเมล์แอดเดรสที่ถูกต้อง และหลอกส่วนหัวอีเมล์

การที่ผู้ส่งอีเมล์ขยะทำให้หัวข้อของอีเมล์มีเนื้อความที่สั้นๆ จากหัวข้อของอีเมล์ปกติ

2.2.2.2 มาตรฐานในการตัดสินใจ

ในการจะหาค่าความน่าจะเป็นของอีเมล์เพื่อใช้ในการระบุว่าอีเมล์นั้นเป็นอีเมล์ขยะหรือไม่ จำเป็นที่จะต้องมีการตัดสินใจที่แน่นอนว่าอีเมล์นั้นเป็นอีเมล์ขยะหรือไม่ ดังตัวอย่างสมการที่ 12 แสดงค่าความน่าจะเป็นที่ใช้ในการระบุว่าอีเมล์นั้นจะเป็นอีเมล์ขยะได้อย่างไร

$$P(Y = 1 \mid x) >= \tfrac{1}{2} \tag{12}$$

และสิ่งที่เราต้องการก็คือการที่ความน่าจะเป็นในการจำแนกมีความผิดพลาดที่น้อยที่สุด โดยมีการกำหนด L->S คือความผิดพลาดในการจำแนกว่าอีเมล์ที่ดีเป็นอีเมล์ขยะ และ S->L คือความผิดพลาดในการจำแนกว่าอีเมล์ขยะเป็นอีเมล์ที่ดี ซึ่งความน่าจะเป็นของความผิดพลาดในการจำแนกแสดงดังสมการ 13

$$P_{err} = P(L->S) + P(S->L) \tag{13}$$

จากสมการที่ 13 เราให้ความน่าจะเป็นของการจำแนกทั้งสองอย่างความน่าจะเป็นที่เท่าเทียมกันแต่ในความเป็นจริงแล้วผู้คนทั่วไปจะต้องการความน่าจะเป็นของความผิดพลาดในการจำแนกอีเมล์ที่ดีเป็นอีเมล์ขยะที่ผิดพลาดน้อยกว่าความผิดพลาดในการจำแนกอีเมล์ขยะเป็นอีเมล์ดี ดังนั้นเราจึงต้องมีการให้น้ำหนักของอีเมล์ที่ดีเพื่อจะได้ไม่ทำให้เกิดความผิดพลาดมากนักของความผิดพลาดในการจำแนกอีเมล์ดีเป็นอีเมล์ขยะ ความน่าจะเป็นของความผิดพลาดที่มีการใส่น้ำหนักแสดงดังสมการ 14

$$P_{wrr} = \lambda \,.\, P(L->S)+P(S->L) \tag{14}$$

ค่าความน่าจะเป็นของความผิดพลาดที่มีการใส่น้ำหนักจะมีค่าน้อยที่สุดจะถูกทำโดยการทำให้เป็นเศษส่วนกันของค่าความน่าจะเป็นที่เป็นอีเมล์ขยะ กับค่าความน่าจะเป็นของอีเมล์ที่ดี

$$P(spam \mid x) / P(legitimate \mid x) > \lambda \tag{15}$$

$$P(spam \mid x) > t \quad ; t = \lambda / 1+\lambda \tag{16}$$

2.2.3 การทดสอบประสิทธิภาพตัวกรอง

เมื่อจัดทำตัวกรองแล้วจะต้องมีการทดสอบประสิทธิภาพ เพื่อทดสอบดูว่าตัวกรองที่ทำขึ้นมาสามารถกรองอีเมล์ได้มีประสิทธิภาพขนาดไหน

ประสิทธิภาพของตัวกรองจะดูจากความผิดพลาดในการจำแนกอีเมล์ทั้งชนิดที่เป็นอีเมล์ดี และอีเมล์ไม่ดี โดยจะมีค่าต่างๆที่สามารถบอกประสิทธิภาพได้ ได้แก่ spam recall(SR) , span precision(SP) และ ค่า total cost ratio(TCR)

Spam Recall (SR)

$$SR = \frac{n_{S->S}}{n_{S->S} + n_{S->L}} \quad (17)$$

$n_{S->S}$ จำนวนของอีเมล์ขยะที่จำแนกได้ถูกต้อง

$n_{S->L}$ จำนวนความผิดพลาดของการจำแนกอีเมล์ขยะ (จำแนกอีเมล์ขยะเป็นอีเมล์ดี)

Spam Precision(SP)

$$SP = \frac{n_{S->S}}{n_{S->S} + n_{L->S}} \quad (18)$$

$n_{S->S}$ จำนวนของอีเมล์ขยะที่จำแนกได้ถูกต้อง

$n_{L->S}$ จำนวนความผิดพลาดของการจำแนกอีเมล์ดี (จำแนกอีเมล์ดีเป็นอีเมล์ขยะ)

ค่า Spam Recall ที่สูงจะบอกว่ามีความผิดพลาดน้อยในการจำแนกอีเมล์ขยะ

ค่า Spam Precision ที่สูงจะบอกว่ามีความผิดพลาดน้อยในการจำแนกอีเมล์ดี

แต่ว่าค่า SP และ SR บอกได้เพียงความผิดพลาดที่เกิดขึ้นในแต่ละการจำแนกอีเมล์ แต่ไม่ได้บอกถึงความแตกต่างของความผิดพลาดทั้งสองอย่าง ซึ่งค่าจึงต้องมีการหาค่า Total Cost Ratio (TCR) เพื่อแก้ไขปัญหานี้

การจะหาค่า TCR เราจำเป็นจะต้องทำการหาค่าต่างๆ ต่อไปนี้ก่อน

accuracy(ACC) , error rate(Err)

$$Acc = \frac{n_{L->L} + n_{S->S}}{N_L + N_S} \qquad Err = \frac{n_{L->S} + n_{S->L}}{N_L + N_S} \quad (19)$$

N_L, N_S จำนวนของอีเมล์ดี และอีเมล์ขยะตามลำดับ

$n_{L->L}$ จำนวนของอีเมล์ดีที่จำแนกได้ถูกต้อง

จากสมการที่ 19 ไม่ได้มีการใส่ค่าน้ำหนักให้กับสมการ จึงมีการใส่น้ำหนักให้กับสมการ

$$\text{W.Acc} = \frac{\lambda . n_{L->L} + n_{S->S}}{\lambda N_L + N_S} \qquad \text{W.Err} = \frac{\lambda . n_{L->S} + n_{S->L}}{\lambda N_L + N_S} \tag{20}$$

เพื่อนำไปหาค่า TCR ควรจะมีการหาค่า accuracy และ err ก่อนที่จะมีการสร้างตัวกรองอีเมล์ขยะดัง สมการที่ 21

$$\text{W.Acc}^{b} = \frac{\lambda N_L}{N_L + N_S} \qquad \text{W.Err}^{b} = \frac{N_S}{\lambda N_L + N_S} \tag{21}$$

จะได้สมการ TCR ดังสมการที่ 22

$$\text{TCR} = \frac{WErr^{b}}{WErr} \tag{22}$$

ค่า TCR ที่สูงจะสามารถบ่งบอกถึงประสิทธิภาพที่ดีของตัวกรองอีเมล์ขยะ แต่ถ้าค่า TCR < 1 แสดงว่า ตัวกรองอีเมล์ขยะไม่มีประสิทธิภาพในการกรอง เพราะไม่สามารถกรองอีเมล์ขยะได้เลย

2.2.4 การประยุกต์ใช้งานทฤษฎี Bayesian

จากการศึกษาการใช้ทฤษฎี Bayesian ในการจัดการอีเมล์ขยะสามารถแบ่งแยกออกเป็นขั้นตอนในการสร้างตัวกรองอีเมล์ขยะออกเป็น 2 ขั้นตอน

ขั้นตอนที่ 1 การเก็บค่าความน่าจะเป็นของคำเอาไว้เป็นฐานความรู้

ในขั้นตอนนี้จะเป็นขั้นตอนในการจัดการกับคำในอีเมล์โดยใช้หลักการในการตัดคำ โดยสังเกตรูปแบบของคำต่างๆ จากอีเมล์ทั้งที่เป็นอีเมล์ขยะ และอีเมล์ทั่วๆไป และตั้งกฎขึ้นมาเพื่อที่จะตัดคำ จากนั้นจึงทำการระบุความน่าจะเป็นของแต่ละคำว่ามีค่าเป็นเท่าไหร่ แล้วเก็บค่าความน่าจะเป็นของคำไว้ใช้เป็นฐานความรู้ โดยจะเก็บลงไฟล์ที่เป็น hash table ซึ่งง่ายต่อการค้นหา โดยการทำงานจะใช้ข้อมูลของอีเมล์ที่เรามีอยู่แล้วโดยการจัดจำแนกออกเป็น อีเมล์ที่ดี และ อีเมล์ขยะ โดยจะต้องเป็นผู้ที่ตัดสินใจเองว่าอีเมล์ฉบับไหนเป็นอีเมล์ที่ดี และฉบับไหนเป็นอีเมล์ขยะ

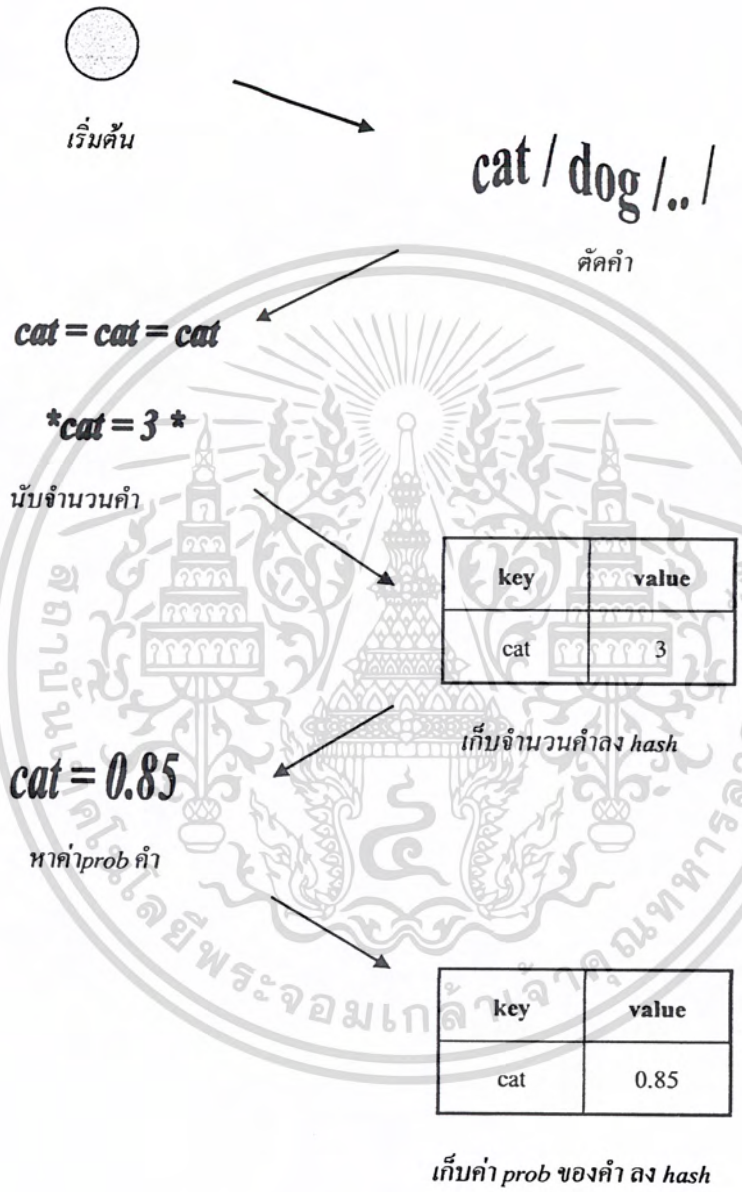
ขั้นตอนที่ 2 จัดจำแนกอีเมลใหม่

เมื่อได้ค่าความน่าจะเป็นของคำของอีเมล์ในขั้นตอนแรกที่เก็บลงในฐานความรู้ ซึ่งจำเป็นจะต้องทำให้ฐานความรู้มีข้อมูลของความน่าจะเป็นของคำในจำนวนหนึ่งแล้ว จึงจะนำค่าความน่าจะเป็นของคำมาเพื่อใช้ในการอ้างอิงถึงค่าความน่าจะเป็นโดยรวมของอีเมล์ฉบับใหม่ที่กำลังเข้ามาในตัวกรองเพื่อใช้ในการทดสอบว่าอีเมล์ที่กำลังจะเข้ามานั้นมีค่าความน่าจะเป็นเป็นเท่าไหร่ ซึ่งค่าความน่าจะเป็นจะใช้ในการบ่งชี้ว่าอีเมล์ฉบับนั้นจะเป็นอีเมล์ที่ดี หรือว่าเป็นอีเมล์ขยะ

เมื่อได้ทดสอบอีเมล์ฉบับต่างๆ ที่เข้ามาในระบบอย่างอัตโนมัติโดยตัวกรองแล้ว ว่าเป็นอีเมล์ที่ดี หรือว่าเป็นอีเมล์ขยะ แล้วจะนำอีเมล์ฉบับที่ได้ทำการทดสอบมาทำการเรียนรู้เพิ่มเติมให้แก่ฐานความรู้อย่างอัตโนมัติเพื่อนำไปใช้เป็นฐานความรู้ในการอ้างอิงการทำงานในครั้งต่อๆไป

การทำงานของแต่ละขั้นตอนจะมีรายละเอียดปลีกย่อยดังต่อไปนี้

ขั้นตอนที่ 1 การใช้ทฤษฎี Bayesian ในการเก็บค่าความน่าจะเป็นของคำไว้ในฐานความรู้



รูปที่ 2-2 ขั้นตอนการเตรียมฐานความรู้

คำอธิบายขั้นตอนที่ 1 การเก็บค่าความน่าจะเป็นไว้ในฐานข้อมูล

วิธีการตัดคำ

ขั้นตอนเริ่มต้นในการเตรียมฐานข้อมูล จะทำการตัดคำในอีเมล์ที่จะนำมาใช้ในการเตรียมฐานความรู้ โดยอีเมล์ที่นำมาใช้ตัดคำจะใช้ทั้งอีเมล์ขยะ และอีเมล์ทั่วๆไป โดยวิธีการตัดคำจะมีเงื่อนไขของการตัดคำเพื่อจะพยายามที่จะไม่เอาคำที่ไม่สื่อความหมายออกไปจากอีเมล์ต้นฉบับ

จุดประสงค์ของการตัดคำที่ไม่สื่อความหมายออกไป เพราะต้องการได้คำเดี่ยวๆที่มีความหมายสมบูรณ์อยู่ในตัวของคำอยู่แล้ว ซึ่งวิธีการหรือเงื่อนไขที่ใช้ในการตัดคำที่ไม่ต้องการออกไปมีเงื่อนไขมากมายในที่นี้จะกล่าวถึงเงื่อนไขที่ทดลองตัดคำดูแล้วและเห็นว่าสามารถตัดคำที่ไม่สื่อความหมายออกไปได้จำนวนหนึ่ง

เงื่อนไขในการตัดคำมีรายละเอียดดังต่อไปนี้

เงื่อนไขที่ 1 ตัดตามการเว้นวรรคของคำ

ในภาษาอังกฤษจะมีการใช้เว้นวรรค เพื่อแบ่งคำต่างๆ ออกด้วยกัน ซึ่งเราต้องการคำเดี่ยวๆ ที่มีความหมายในตัวเองจึงทำการตัดเว้นวรรคของคำ เพื่อให้ได้คำเดี่ยวๆ

เงื่อนไขที่ 2 ตัดเครื่องหมายต่างๆ ออกจากคำ

ในเนื้อความของอีเมล์การจะแยกคำออกเป็นคำๆ ที่สื่อความหมายโดยการใช้การเว้นวรรคเพียงอย่างเดียวเป็นเรื่องที่ยาก เพราะคำบางคำในอีเมล์อาจจะมีเครื่องหมายที่ใส่อยู่ในคำเพื่อใช้ในการแยกคำ หรือว่าเป็นลิงค์ที่ใช้เพื่อต่อไปยังอินเทอร์เนต เช่น ***hello*** , Advertise_here , http://www.ce.kmitl.ac.th และอื่นๆ

มีความจำเป็นอย่างมากที่จะต้องคำการตัดเครื่องหมายต่างๆ เหล่านี้ออกเพื่อแยกคำออกเป็นคำเดี่ยวๆ เพราะถ้าไม่มีการแบ่งแยกคำออกเป็นคำเดี่ยวๆ แล้วอาจเป็นช่องทางที่ผู้ส่งอีเมล์ขยะ สามารถจะมาก่อกวนตัวกรองได้โดยการใส่คำที่มีเครื่องหมายแปลกๆ เข้ามาด้วยทำให้ตัวกรองไม่สามารถแยกแยะได้ว่าคำๆนั้นเป็นคำที่ดีที่สื่อความหมาย หรือว่าเป็นคำที่ไม่ดี

เครื่องหมายที่มักพบบ่อยๆ ในอีเมล์ที่ควรจะทำการตัดก็ได้แก่

[.] , _ , * , { , } , < , > , (,) , : , ; , ? , ! , \ , / , % , + , > , < , @ , " , ' , = , &

เงื่อนไขที่ 3 เก็บค่าไอพี และ ที่อยู่ของผู้ให้บริการอีเมล์

ค่าไอพี และ ที่อยู่ของผู้ให้บริการอีเมล์ที่ระบุในอีเมล์ บางครั้งก็สามารถที่จะใช้ในการระบุความเป็นอีเมล์ขยะได้

บางครั้งค่าไอพีเช่น 161.246.6.21 ที่อยู่ในอีเมล์โดยที่อยู่ในส่วนหัวของอีเมล์ก็อาจใช้ระบุประเภทของอีเมล์ได้ เพราะบางครั้งอีเมล์ขยะอาจจะถูกส่งมาจากไอพีเดียวกันตลอด ซึ่งจากการส่งมาจากไอพีเดียวกันตลอดค่าไอพีนั้นก็จะแสดงค่าความน่าจะเป็นที่สูงที่สามารถระบุได้ว่าเป็นอีเมล์ขยะ

ในส่วนของที่อยู่ผู้ให้บริการอีเมล์เช่น @yahoo.co.uk ก็สามารถใช้บอกได้ว่าอีเมล์นั้นถูกส่งมาจากเซิร์ฟเวอร์ไหน ถ้าอีเมล์ขยะนั้นถูกส่งมาจากผู้ให้บริการอีเมล์เดิมบ่อยๆ ค่าความน่าจะเป็นของที่อยู่ผู้ให้บริการนั้นก็จะสูงตามไปด้วยทำให้สามารถระบุความเป็นอีเมล์ขยะได้

เงื่อนไขที่ 4 ตัดตัวเลข

ตัวเลขโดดๆ เช่น 1 , 100 ที่ปรากฏอยู่ในอีเมล์ไม่ได้ช่วยในการระบุความเป็นอีเมล์ขยะหรือว่าอีเมล์ทั่วไปได้มากนัก แต่ว่าตัวเลขเช่น 10-20, 200$ ตัวเลขเหล่านี้ควรจะเก็บไว้เพราะอาจบอกความเป็นอีเมล์ขยะได้

เงื่อนไขที่ 5 ตัดคำที่ไม่น่าจะระบุอะไร (Stop words)

เมื่อเราได้คำเป็นคำโดดๆ แล้ว แต่ว่าคำโดดๆบางคำก็ไม่ได้ใช้ในการระบุความเป็นอีเมล์ขยะ ได้มากนัก เพราะว่าคำเหล่านั้นสามารถพบได้ทั่วไปในอีเมล์ปกติ เช่น การตัดคอมเมนต์ต่างๆของ HTML ออก , การตัดคำในส่วนของส่วนหัวของอีเมล์เช่น from ,to เพราะคำพวกนี้พบได้ในทุกอีเมล์ และคำอื่นๆที่เห็นว่าพบได้ทั่วไป

วิธีการนับจำนวนคำ

จากขั้นตอนของการตัดคำในอีเมล์ จะได้คำโดดๆ ซึ่งคำเหล่านี้จะมีจำนวนที่มากในระดับหนึ่ง จึงต้องทำการรวบรวมคำ โดยหลักในการรวบรวมคำทำได้โดยการนับจำนวนคำที่ซ้ำกัน เช่น เมื่อทำการตัดคำในอีเมล์แล้ว จะได้คำว่า cat จะทำการนับจำนวนคำว่า cat ว่าในอีเมล์ฉบับนั้นมีอยู่จำนวนเท่าไหร่

วิธีการเก็บจำนวนคำลง Hash Table

จากขั้นตอนการนับคำจะได้ ว่าในอีเมล์ฉบับนั้นแต่ละคำจะมีจำนวนอยู่เท่าไหร่ แต่ว่าจะต้องรู้ว่าคำที่ได้ตัดแล้ว และนับแล้วมาจากอีเมล์ประเภทไหน เพื่อจะได้ใช้ในการแยกแยะว่ากลุ่มของคำที่ได้นับแล้วเป็นของอีเมล์ประเภทใด

เมื่อได้ทำการแยกประเภทของคำของอีเมล์แล้ว จะต้องนำคำและจำนวนที่นับได้เก็บลงใน hash table ซึ่ง hash table จะแยกออกเป็นสอง hash table อันหนึ่งใช้สำหรับเก็บคำและจำนวนของอีเมล์ที่ดี ส่วนอีก hash table ใช้เก็บคำและจำนวนของอีเมล์ขยะ

ถ้านำคำที่นับได้ของอีเมล์มาเก็บลงใน hash table จะต้องทำการตรวจสอบใน hash table ว่ามีการเก็บคำนั้นไว้แล้วหรือเปล่า และเก็บเป็นจำนวนเท่าไหร่ จากนั้นจะนำจำนวนคำที่ตัดได้ไปเพิ่มจำนวนของคำๆนั้นให้ใน hash table

วิธีการคำนวณความน่าจะเป็นของแต่ละคำ

ขั้นตอนนี้จะเป็นขั้นตอนที่ใช้ในการคำนวณค่าความน่าจะเป็นของคำ โดยจะคำนวณคำทุกคำที่อยู่ทั้งใน hash table ของอีเมล์ดี และ Hash Table ของอีเมล์ขยะ

การคำนวณค่าความน่าจะเป็นของคำเพื่อใช้เป็นฐานความรู้เพื่อให้สามารถระบุอีเมล์ที่เข้ามาใหม่ได้ว่าเป็นอีเมล์ขยะ หรือว่าเป็นอีเมล์ทั่วๆไป

วิธีการคำนวณค่าความน่าจะเป็นของคำแสดงดังรูปที่ 2-3

```
( let (( g ( * 2 ( or (gethash word good ) 0 )))
      ( b ( or (gethash word bad ) 0 )))
      (unless ( < (+ g b ) 5 )
            (max . 01
      (min .99 (float (/ (min 1 (/ b  nbad))
            (+ (min 1(/ g ngood))
            (min 1 (/ b nbad ))))))))
```

รูปที่ 2-3 การคำนวณความน่าจะเป็นของคำ

word good , word bad แทนจำนวนคำของแต่ละคำที่อยู่ในแต่ละ hash table โดยเป็นคำๆ เดียวกัน แต่ว่าอยู่ต่าง hash table โดย word good จะแทนด้วยจำนวนคำที่อยู่ใน hash table ของอีเมล์ที่ดี ในขณะที่ word bad จะแทนด้วยจำนวนคำที่อยู่ใน hash table ของอีเมล์ขยะ เช่น

Hash table ของอีเมล์ดีมีคำว่า Advertise จำนวนเท่ากับ 2

Hash Table ของอีเมล์ขยะมีคำว่า Advertise จำนวนเท่ากับ 3

word good (Advertise = 2) , word bad (Advertise = 3)

ngood , nbad จำนวนของอีเมล์ โดย ngood แทนจำนวนอีเมล์ที่ดี nbad แทนจำนวนอีเมล์ขยะ

การคำนวณการเก็บข้อมูลในฐานความรู้

นำ **word good** และ **word bad** มาจาก hash table โดยเลือกคำที่เหมือนกัน

word good นำไปคำนวณหาค่า **g = (word good or 0) * 2**

word bad นำไป **b = (word bad or 0)**

ถ้า **g + b** น้อยกว่า 5 ก็ให้ค่าความน่าจะเป็นของคำมีค่าเป็น 0.0

g + b มากกว่า 5 ทำในขั้นตอนต่อไป

นำ g กับ b ไปคำนวณ ดังนี้

max .01 (min .99 ,

(min (1 , (b/nbad))) /

[(min (1 , (g/ngood))) + (min (1,(b/nbad)))]

หมายเหตุ *** **function min(x,y)**

ถ้า **x>y min(x,y) = y** ***

*** **function max(x,y)**

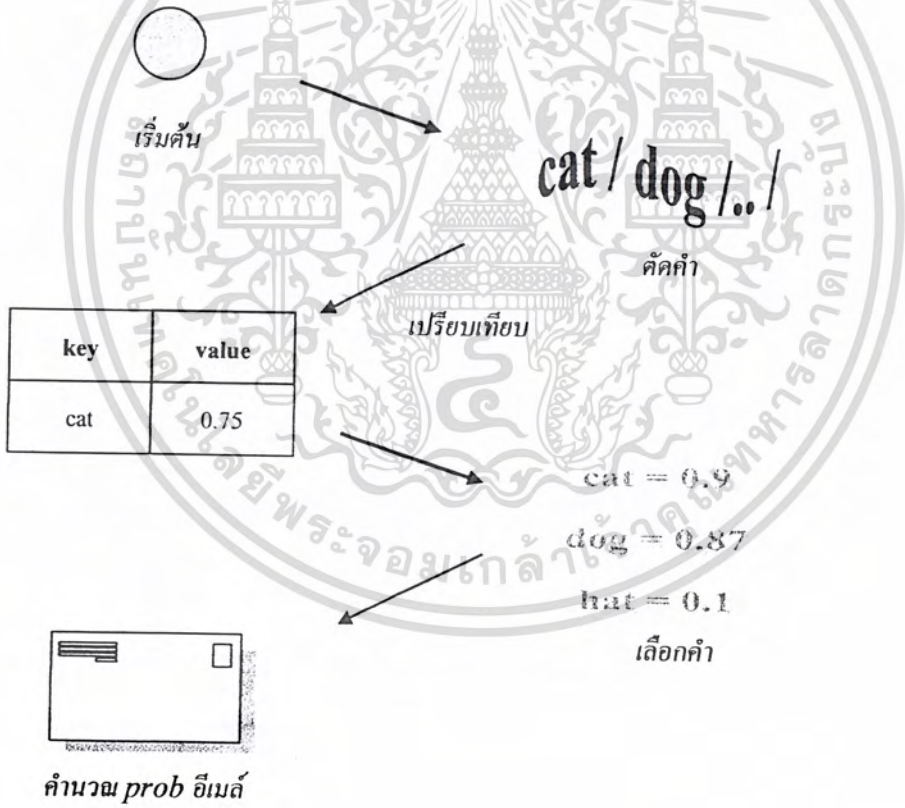
ถ้า **x>y max(x,y) = x** ***

วิธีการเก็บค่าความน่าจะเป็นของคำลง Hash Table

นำค่าความน่าจะเป็นของคำที่คำนวณได้ในขั้นตอนที่ 4 มาเก็บลงใน hash table โดยที่ hash table ที่ใช้เก็บจะเป็น hash table อีก hash table หนึ่งที่สร้างขึ้นมา โดย hash table นี้จะมีค่าความน่าจะเป็นของคำทุกคำที่อยู่ใน hash table ก่อนหน้านี้ โดยค่าความน่าจะเป็นของคำนี้จะสามารถปรับปรุงได้ตลอดเวลา

ในขั้นตอนของการเตรียมฐานความรู้ควรจะมีการเตรียมฐานความรู้ให้ตัวกรองอีเมล์ มีความสามารถในระดับหนึ่งก่อนจึงจะสามารถนำไปใช้เปรียบเทียบเพื่อระบุว่าอีเมล์ที่เข้ามาเป็นอีเมล์ขยะ หรือว่าเป็นอีเมล์ปกติ ในขั้นตอนที่ 2

ขั้นตอนที่ 2 การจัดจำแนกอีเมล์ใหม่



รูปที่ 2-4 การจัดจำแนกอีเมล์ใหม่

คำอธิบายขั้นตอนที่ 2 การจัดจำแนกอีเมล์ใหม่

วิธีการตัดคำ

เมื่อมีอีเมล์ใหม่เข้ามา นำอีเมล์ใหม่มาตัดคำด้วยกฎเกณฑ์เดียวกับในขั้นตอนการเตรียมฐานความรู้

วิธีการเปรียบเทียบ

นำคำที่ได้จากขั้นตอนการตัดคำนำไปเปรียบเทียบกับฐานความรู้ที่ได้ทำไว้แล้ว โดยนำคำที่ตัดไปเปรียบเทียบดูว่าคำนั้นๆของอีเมล์ที่เข้ามาในตัวกรอง มีค่าความน่าจะเป็นเท่าไหร่

วิธีการเลือกคำ

เป็นขั้นตอนที่ได้ค่าความน่าจะเป็นของคำออกมาแล้วมาทำการเลือก เพื่อจะได้นำคำที่เลือกไปคำนวณหาค่าความน่าจะเป็นของอีเมล์ฉบับนั้นๆ

การเลือกคำจะเลือกคำที่มีค่าความน่าจะเป็นของคำในช่วงมากๆ อาจจะมากกว่า 0.75 ซึ่งคำที่มีค่าความน่าจะเป็นมากๆนี้บอกได้ว่าเป็นคำไม่ดี และในช่วงที่น้อยๆ เช่นในช่วงที่น้อยกว่า 0.15 ช่วงนี้จะบอกว่าคำนั้นเป็นคำที่ดี ซึ่งการที่เลือกค่าความน่าจะเป็นในช่วงนี้ก็เพราะค่าความน่าจะเป็นของคำในช่วงกลาง จะไม่ค่อยบอกอะไรมากนักว่าอีเมล์ฉบับนั้นประกอบไปด้วยคำที่เป็นคำไม่ดี หรือว่าคำที่ดี

วิธีการคำนวณค่าความน่าจะเป็นของอีเมล์

การคำนวณค่าความน่าจะเป็นของอีเมล์จะคำนวณจากค่าความน่าจะเป็นของคำที่ได้เลือกไว้แล้วซึ่งการคำนวณจะสามารถบอกได้ว่าอีเมล์ฉบับนั้นเป็นอีเมล์ขยะหรือว่าเป็นอีเมล์ทั่วๆไป

วิธีการคำนวณค่าความน่าจะเป็นของอีเมล์มีดังนี้

$$\frac{a_1 * a_2 * a_3 * \ldots\ldots a_n}{(a_1 * a_2 * a_3 * \ldots\ldots a_n) + [(1-a_1) * (1-a_2) * (1-a_3) * \ldots\ldots (1-a_n)]}$$

a_j = ค่าความน่าจะเป็นของคำแต่ละคำที่นำมาคำนวณ

สุดท้ายจะได้ค่าความน่าจะเป็นของอีเมล์ฉบับนั้นซึ่งจะสามารถบอกได้ว่าเป็นอีเมล์ขยะหรือไม่โดยอาจจะกำหนดว่าถ้าค่าความน่าจะเป็นของอีเมล์มีค่า > 0.85 ก็จะระบุว่าอีเมล์ฉบับนั้นเป็นอีเมล์ขยะ

วิธีการเรียนรู้อย่างอัตโนมัติของตัวกรอง

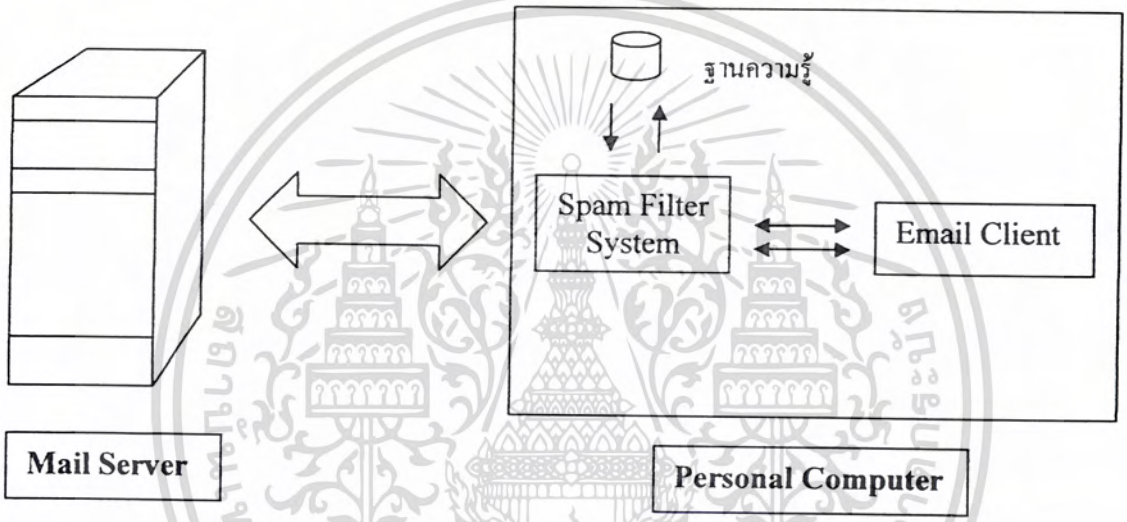
ขั้นตอนนี้เป็นขั้นตอนการเรียนรู้หลังจากที่มีการคำนวณค่าความน่าจะเป็นของอีเมล์ และมีการจำแนกอีเมล์โดยตัวกรองอย่างอัตโนมัติ ซึ่งตัวกรองจะทำนำอีเมล์ที่ได้ทำการแยกประเภทไว้อย่างอัตโนมัติแล้วไปทำการเรียนรู้เพิ่มเติมให้กับตัวกรองเอง

ซึ่งขั้นตอนในการทำการเรียนรู้ให้แก่ตัวกรองอีเมล์ขยะจะใช้ขั้นตอนที่เหมือนกับขั้นตอนของการเตรียมฐานความรู้แต่ว่าขั้นตอนนี้ผู้ใช้ไม่จำเป็นต้องทำการระบุอีเมล์เองว่าเป็นอย่างไร เพราะตัวกรองได้กระทำให้อย่างอัตโนมัติ

บทที่ 3

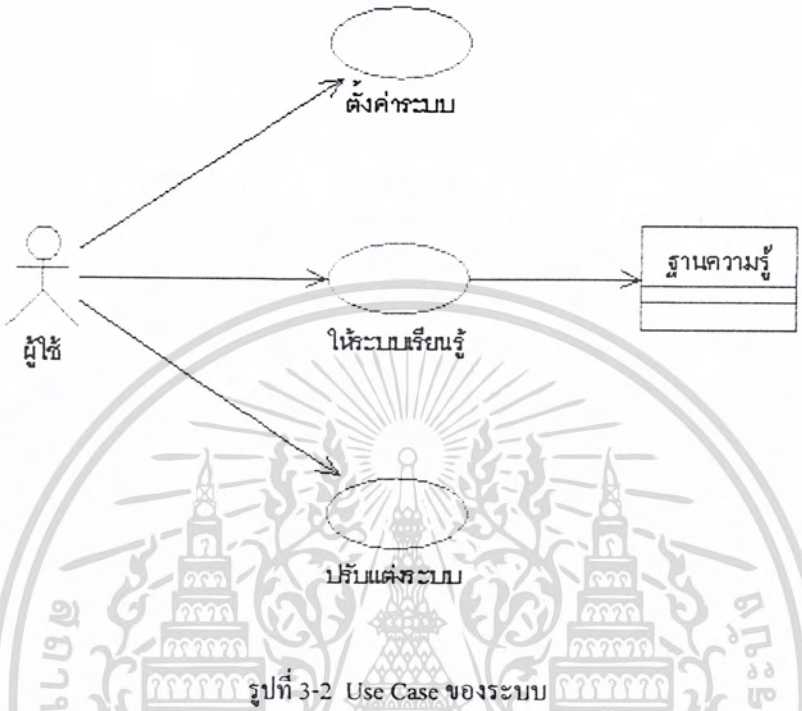
การออกแบบระบบ

ระบบกรองอีเมล์ขยะถูกออกแบบไว้ให้ทำงานอยู่บนเครื่องผู้ใช้งาน โดยจะทำงานอยู่ระหว่างเมล์เซิร์ฟเวอร์และโปรแกรมอีเมล์ไคลเอนต์ เช่น Outlook Express , Microsoft Outlook , Eudora โดยเมื่อมีอีเมล์เข้ามาใหม่และระบบพร้อมที่จะทำการจัดจำแนกอีเมล์ ระบบกรองอีเมล์ขยะจะทำการจัดจำแนกอีเมล์จากนั้นจึงส่งอีเมล์นั้นไปยังโปรแกรมอีเมล์ไคลเอนต์ ดังรูปที่ 3- 1



รูปที่ 3-1 แสดงตำแหน่งที่ระบบกรองอีเมล์ขยะทำงานอยู่

3.1 Use Case Diagram ของระบบ

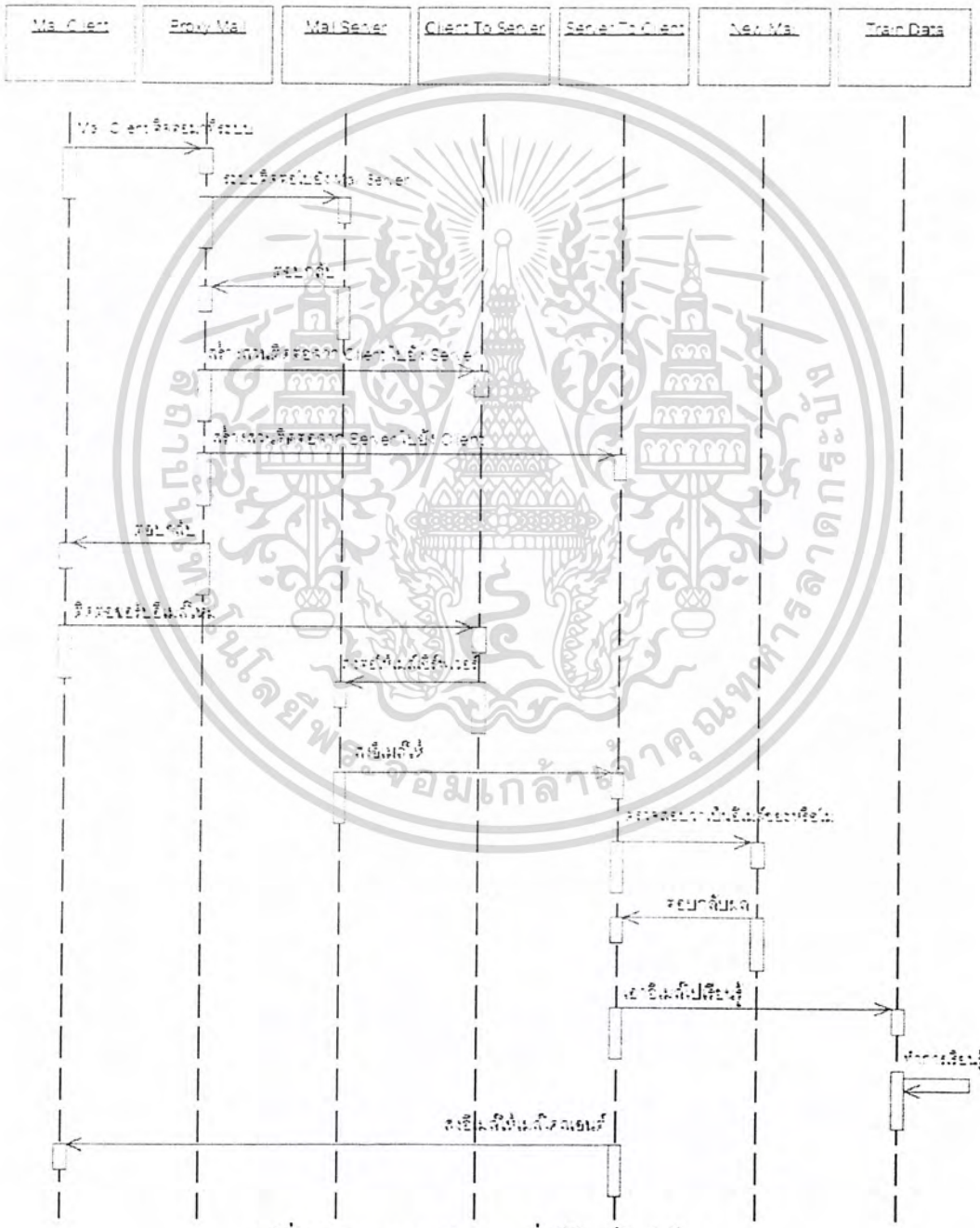


รูปที่ 3-2 Use Case ของระบบ

3.2 Sequence Diagram

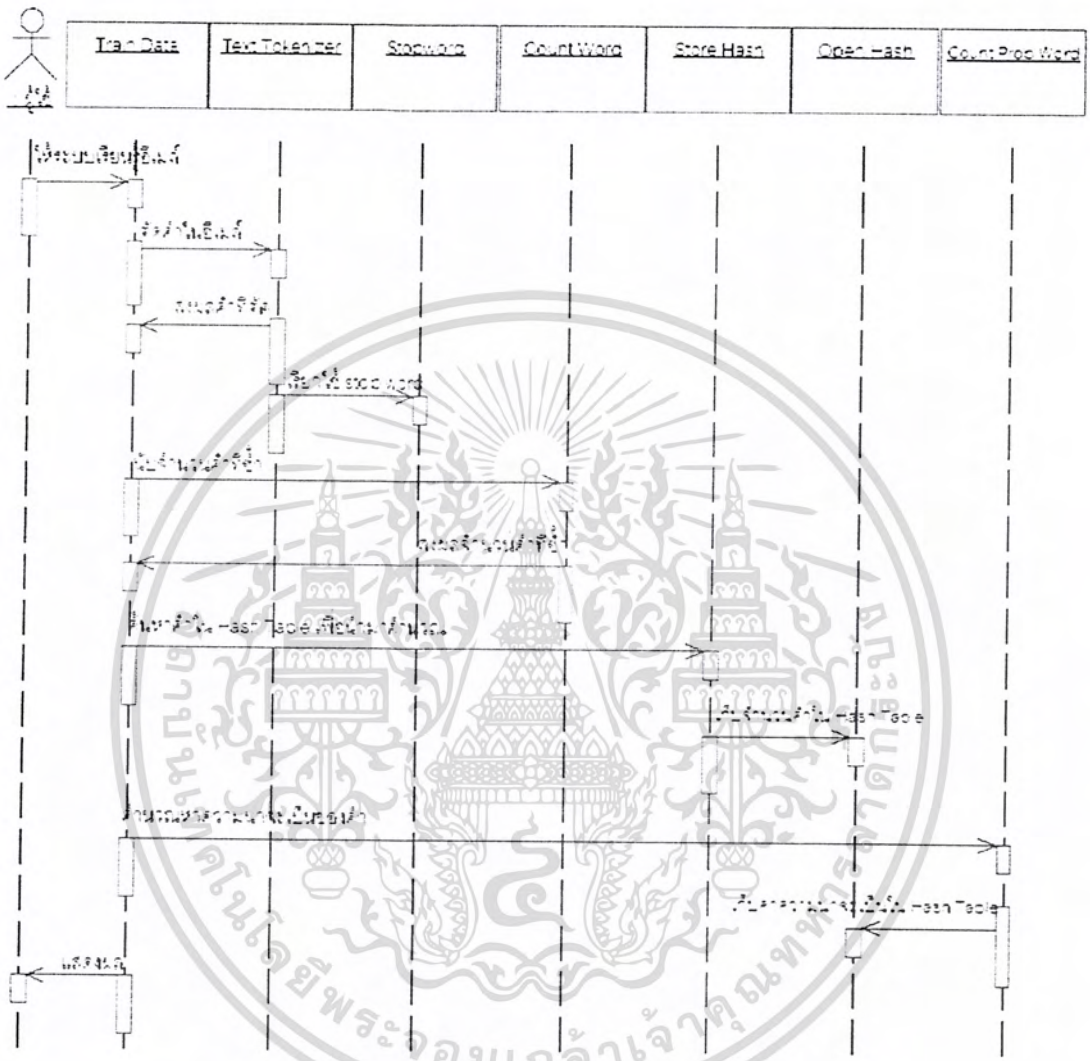
การทำงานของระบบกรองอีเมล์ขยะแสดงโดย Sequence Diagram ทั้ง 3 คือ ส่วนที่โปรแกรมอีเมล์ไคลเอนต์ติดต่อขอรับอีเมล์ใหม่มาจากเมล์เซิร์ฟเวอร์ ซึ่งถ้าระบบกรองอีเมล์ขยะพร้อมใช้งานก็จะทำการจัดจำแนกอีเมล์ ส่วนที่ผู้ใช้ทำการให้ระบบเรียนรู้อีเมล์เมื่ออีเมล์เป็น Unclassified และส่วนที่ผู้ใช้ทำการ Reclassified อีเมล์ที่ถูกจัดจำแนกไปแล้ว

3.2.1 Sequence Diagram เมื่อมีอีเมล์ใหม่เข้ามาในระบบ ซึ่งอยู่บนสมมุติฐานที่ว่าระบบพร้อมที่จะทำการจัดจำแนกอีเมล์



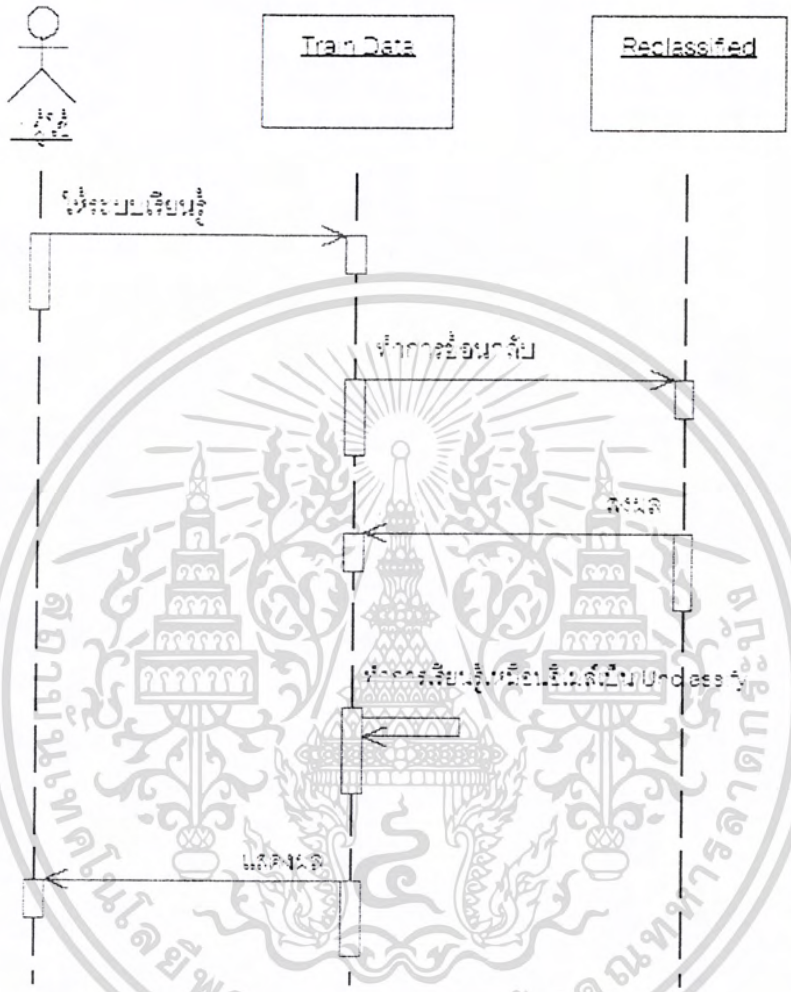
รูปที่ 3-3 Sequence Diagram เมื่อมีอีเมล์ใหม่เข้ามา

3.2.2 Sequence Diagram ของการให้ระบบเรียนรู้ เมื่ออีเมล์เป็น Unclassify



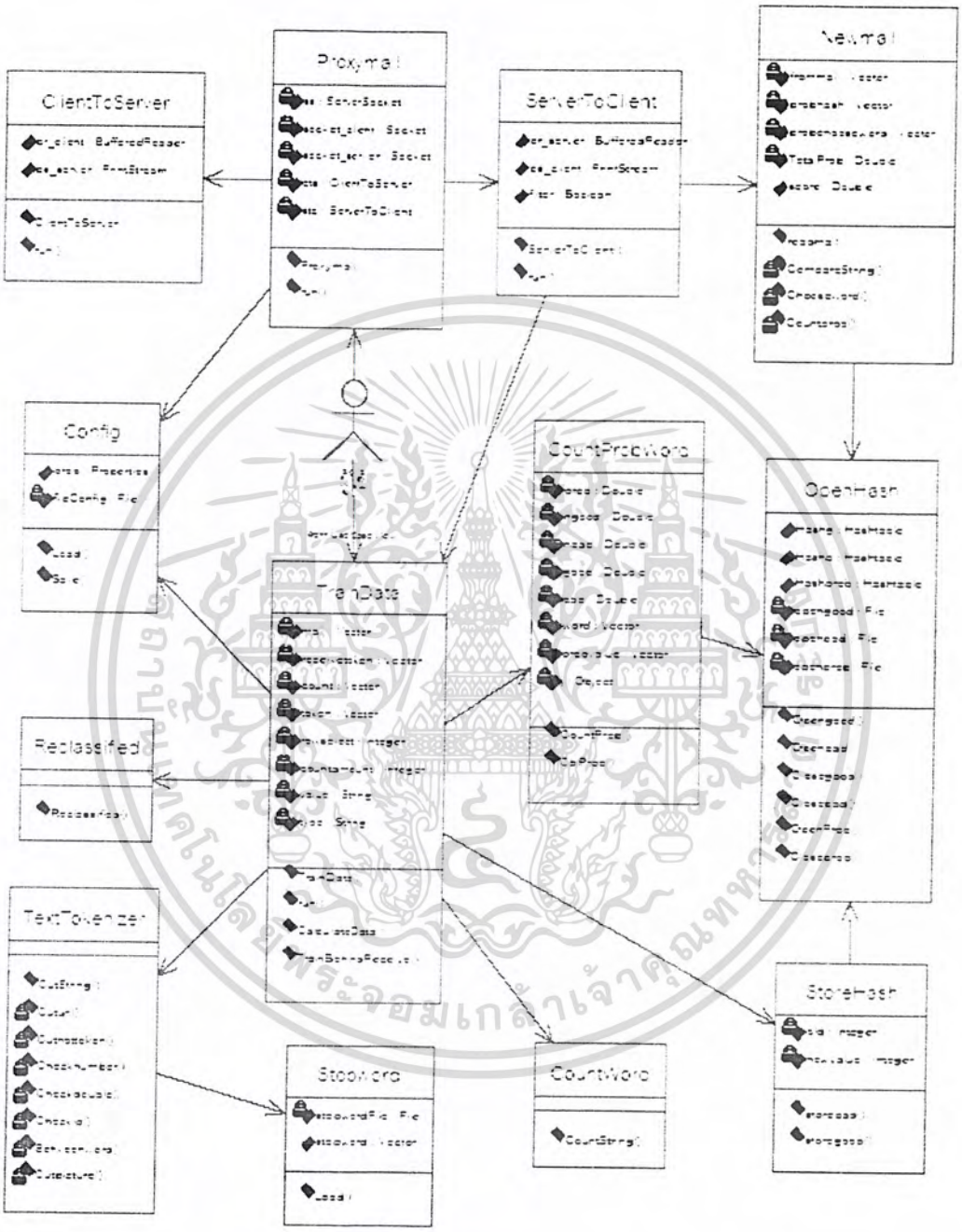
รูปที่ 3-4 Sequence Diagram ของการให้ระบบเรียนรู้

3.2.3 Sequence Diagram ของการให้ระบบเรียนรู้ เมื่ออีเมล์ถูก Classify ไปแล้ว



รูปที่ 3-5 Sequence Diagram ของการให้ระบบเรียนรู้ เมื่ออีเมล์ถูก Classify ไปแล้ว

3.3 Class Diagram ของระบบ



รูปที่ 3-6 Class Diagram ของระบบ

3.4 เครื่องมือที่ใช้ในการพัฒนาโปรแกรม

3.4.1 j2sdk 1.4.5_02

3.4.2 JCreator Pro v3.0

บทที่ 4
ผลการดำเนินงาน

4.1 ขั้นตอนในการดำเนินงาน

4.1.1 ศึกษาวิธีการในการรับส่งอีเมล์ และรูปแบบของอีเมล์ทั่วๆไป

4.1.2 ศึกษาหารูปแบบของอีเมล์ขยะว่ามีรูปแบบแตกต่างจากอีเมล์ทั่วไปอย่างไร และทำการเก็บรูปแบบทั่วไปของอีเมล์ขยะ

4.1.3 ศึกษารูปแบบของวิธีการต่างๆที่ผู้ส่งอีเมล์ขยะส่งอีเมล์ขยะให้แก่ผู้ใช้

4.1.4 ศึกษาวิธีการป้องกันอีเมล์ขยะ โดยศึกษาหาข้อดี และข้อเสียของแต่ละวิธีการ

4.1.5 ศึกษาทฤษฎี Bayesian เพราะเป็นวิธีที่ใช้ในการป้องกันอีเมล์ขยะ

4.1.6 ศึกษาการทำงานของโพรโตคอล pop 3

4.1.7 ศึกษาการเขียนภาษา java ในการสร้างระบบการกรอง

4.1.8 เก็บรวมรวมอีเมล์ทั้งอีเมล์ที่ดี และอีเมล์ขยะ เพื่อใช้ในการทดสอบระบบ

4.1.9 ทดสอบประสิทธิภาพโดยรวมของระบบ

4.1.10 สร้างโปรแกรมตัวกรอง

4.2 ขั้นตอนในการทดสอบประสิทธิภาพของระบบ

ภายหลังจากการสร้างระบบในการกรองอีเมล์ขยะ โดยวิธี Bayesian แล้วจึงทำการจำลองระบบที่จะทำการทดสอบความมีประสิทธิภาพของตัวกรอง ซึ่งขั้นตอนในการทดสอบดังต่อไปนี้

4.2.1 จัดตั้งระบบของตัวกรองอีเมล์ขยะ

ตัวกรองอีเมล์ขยะจะทำงานอยู่ระหว่างเซิร์ฟเวอร์ที่ให้บริการอีเมล์ และโปรแกรมอีเมล์ที่ผู้ใช้ จึงต้องทำให้ตัวกรองอีเมล์ขยะเป็นได้ทั้งโปรแกรมอีเมล์ของผู้ใช้ และเป็นเสมือนอีเมล์เซิร์ฟเวอร์

โดยการจัดตั้งระบบจะต้องทำให้ตัวกรองสามารถติดต่อกับอีเมล์เซิร์ฟเวอร์ได้เพื่อรับอีเมล์มาจากอีเมล์เซิร์ฟเวอร์ ซึ่งอีเมล์เซิร์ฟเวอร์ต้องมีการให้บริการอีเมล์โพรโตคอล pop3 และตัวกรองต้องสามารถที่จะส่งอีเมล์ให้กับโปรแกรมอีเมล์ของผู้ใช้

4.2.2 รวบรวมอีเมล์เพื่อนำมาใช้เพื่อทำให้ตัวกรองได้เรียนรู้

รวบรวมอีเมล์ในประเภทที่เป็นอีเมล์ที่ดี และอีเมล์ขยะ โดยผู้พัฒนาเป็นผู้ตัดสินว่าอีเมล์ฉบับไหนอยู่ในกลุ่มของอีเมล์ประเภทไหน ซึ่งจำเป็นต้องเก็บรวบรวมอีเมล์ไว้ให้มีจำนวนที่มากในปริมาณหนึ่ง

4.2.3 ให้ตัวกรองได้เรียนรู้อีเมล์

นำอีเมล์ที่ได้รวบรวมไว้มาทำการเรียนรู้ให้กับตัวกรองเพื่อตัวกรองจะสามารถเก็บค่าความน่าจะเป็นของคำแต่ละคำเพื่อใช้ในการจำแนกอีเมล์ต่อไป

4.2.4 ทดสอบประสิทธิภาพของตัวกรอง

นำอีเมล์มาจำนวนหนึ่งซึ่งเป็นอีเมล์ที่ได้จำแนกโดยผู้พัฒนาแล้วว่าเป็นอีเมล์ประเภทใด จากนั้นจึงนำอีเมล์ที่ได้ทำการจำแนกแล้วมาทดสอบระบบดูว่าระบบมีประสิทธิภาพที่จะสามารถกรองอีเมล์มากเท่าไหร่ โดยวัดประสิทธิภาพโดยการใช้ค่าต่างๆในการประเมินผล

4.3 ผลการทดสอบ

จากการทดสอบความสามารถในการกรองของตัวกรองอีเมล์ขยะ โดยการใช้อีเมล์ที่ดีจำนวน 600 อีเมล์ และอีเมล์ขยะจำนวน 500 อีเมล์ โดยทำการทดสอบโดยทดสอบอีเมล์จำนวน 5 ครั้ง โดยแต่ละครั้งจะทดสอบแบบแบ่งอีเมล์แต่ละชนิดออกเป็น 5 ส่วน โดยใช้ 1 ส่วนในการทดสอบระบบ และอีก 4 ส่วนที่เหลือใช้ในการทำให้ตัวกรองเรียนรู้ โดยส่วนที่ใช้ทดสอบระบบจะมีการเลือกคำจากอีเมล์ที่นำมาทดสอบโดยทำการเลือกคำโดยมีการให้น้ำหนักของค่าความน่าจะเป็นของคำที่ต้องการในช่วงต่างๆ และมีการปรับปรุงฐานความรู้เพิ่มเติมโดยการเพิ่มอีเมล์ที่ใช้ในการปรับปรุงฐานความรู้เข้าไปโดยใช้อีเมล์ที่ใช้ในการทดสอบเพิ่มเติมให้กับฐานความรู้ และทำการทดสอบอีเมล์ใหม่อีกครั้งหนึ่ง

ซึ่งจากการทดสอบประสิทธิภาพของระบบจะทำให้ได้ค่าต่างๆ ซึ่งแสดงประสิทธิภาพของระบบดังต่อไปนี้

Spam Recall(SR) , Spam Precision (SP) , WAcc , WErr , $WAcc^{b}$, $WErr^{b}$, TCR

ใช้อีเมล์ที่ดีจำนวน 480 ฉบับ และ อีเมล์ขยะจำนวน 400 ฉบับ เป็นฐานความรู้

มีการให้ค่าน้ำหนักสำหรับการทดสอบดังนี้ (เลือกช่วงค่าความน่าจะเป็นของคำ)

1. มากกว่า 0 แต่น้อยกว่าเท่ากับ 0.15 หรือ มากกว่าเท่ากับ 0.85 แต่น้อยกว่า 1

(0< x <= 0.15 || 0.85 <= x < 1)

2. มากกว่า 0 แต่น้อยกว่าเท่ากับ 0.15 หรือ มากกว่าเท่ากับ 0.90 แต่น้อยกว่า 1

(0< x <= 0.15 || 0.90 <= x < 1)

3. มากกว่า 0 แต่น้อยกว่าเท่ากับ 0.15 หรือ มากกว่าเท่ากับ 0.75 แต่น้อยกว่า 1

(0< x <= 0.15 || 0.75 <= x < 1)

4. มากกว่า 0 แต่น้อยกว่าเท่ากับ 0.05 หรือ มากกว่าเท่ากับ 0.85 แต่น้อยกว่า 1

(0< x <= 0.05 || 0.85 <= x < 1)

5. มากกว่า 0 แต่น้อยกว่าเท่ากับ 0.25 หรือ มากกว่าเท่ากับ 0.85 แต่น้อยกว่า 1

(0< x <= 0.25 || 0.85 <= x < 1)

การทดสอบครั้งที่ 1

ช่วงค่าความน่าจะเป็นที่เลือก	ฐานความรู้เริ่มต้น				ฐานความรู้ปรับปรุง			
	อีเมล์ดี		อีเมล์ขยะ		อีเมล์ดี		อีเมล์ขยะ	
	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น
1	117	3	84	16	120	0	88	12
2	117	3	79	21	120	0	80	20
3	117	3	87	13	120	0	90	10
4	115	5	88	12	118	2	91	9
5	118	2	64	36	120	0	65	35

ตารางที่ 4-1 การทดสอบตัวกรองครั้งที่ 1

ฐานความรู้เริ่มต้น(ครั้งที่ 1)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	$WAcc^b$	$WErr^b$	TCR
1	0.84	0.965	0.913	0.086	0.545	0.454	5.279
2	0.79	0.963	0.890	0.109	0.545	0.454	4.165
3	0.87	0.966	0.927	0.072	0.545	0.454	6.305
4	0.88	0.946	0.922	0.077	0.545	0.454	5.896
5	0.64	0.969	0.827	0.172	0.545	0.454	2.639

ตารางที่ 4-2 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้เริ่มต้นครั้งที่ 1

ฐานความรู้ปรับปรุง(ครั้งที่ 1)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	$WAcc^b$	$WErr^b$	TCR
1	0.88	1	0.945	0.054	0.545	0.454	8.407
2	0.80	1	0.909	0.090	0.545	0.454	5.044
3	0.90	1	0.954	0.045	0.545	0.454	10.080
4	0.91	0.978	0.95	0.05	0.545	0.454	9.08
5	0.65	1	0.840	0.159	0.545	0.454	2.855

ตารางที่ 4-3 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้ปรับปรุงครั้งที่ 1

การทดสอบครั้งที่ 2

ช่วงค่าความน่าจะเป็นที่เลือก	ฐานความรู้เริ่มต้น				ฐานความรู้ปรับปรุง			
	อีเมล์ดี		อีเมล์ขยะ		อีเมล์ดี		อีเมล์ขยะ	
	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น
1	120	0	92	8	120	0	96	4
2	120	0	90	10	120	0	93	7
3	120	0	94	6	120	0	98	2
4	120	0	95	5	120	0	99	1
5	120	0	87	13	120	0	90	10

ตารางที่ 4-4 การทดสอบตัวกรองครั้งที่ 2

ฐานความรู้เริ่มต้น(ครั้งที่ 2)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	$WAcc^b$	$WErr^b$	TCR
1	0.92	1	0.963	0.036	0.545	0.454	12.61
2	0.90	1	0.954	0.045	0.545	0.454	10.08
3	0.94	1	0.972	0.027	0.545	0.454	16.81
4	0.95	1	0.977	0.022	0.545	0.454	20.63
5	0.87	1	0.940	0.059	0.545	0.454	7.69

ตารางที่ 4-5 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้เริ่มต้นครั้งที่ 2

ฐานความรู้ปรับปรุง(ครั้งที่ 2)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	$WAcc^b$	$WErr^b$	TCR
1	0.96	1	0.981	0.018	0.545	0.454	25.22
2	0.93	1	0.968	0.031	0.545	0.454	14.64
3	0.98	1	0.990	0.009	0.545	0.454	50.44
4	0.99	1	0.995	0.004	0.545	0.454	113.5
5	0.90	1	0.954	0.045	0.545	0.454	10.08

ตารางที่ 4-6 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้ปรับปรุงครั้งที่ 2

การทดสอบครั้งที่ 3

ช่วงค่าความน่าจะเป็นที่เลือก	ฐานความรู้เริ่มต้น				ฐานความรู้ปรับปรุง			
	อีเมล์ดี		อีเมล์ขยะ		อีเมล์ดี		อีเมล์ขยะ	
	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น
1	120	0	81	19	120	0	85	15
2	120	0	80	20	120	0	84	16
3	120	0	84	16	120	0	89	11
4	120	0	86	14	120	0	90	10
5	120	0	79	21	120	0	81	19

ตารางที่ 4-7 การทดสอบตัวกรองครั้งที่ 3

ฐานความรู้เริ่มต้น(ครั้งที่ 3)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	$WAcc^b$	$WErr^b$	TCR
1	0.81	1	0.913	0.086	0.545	0.454	5.27
2	0.80	1	0.909	0.090	0.545	0.454	5.04
3	0.84	1	0.927	0.072	0.545	0.454	6.30
4	0.86	1	0.936	0.063	0.545	0.454	7.20
5	0.79	1	0.904	0.095	0.545	0.454	4.77

ตารางที่ 4-8 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้เริ่มต้นครั้งที่ 3

ฐานความรู้ปรับปรุง(ครั้งที่ 3)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	$WAcc^b$	$WErr^b$	TCR
1	0.85	1	0.931	0.068	0.545	0.454	6.67
2	0.84	1	0.927	0.072	0.545	0.454	6.305
3	0.89	1	0.95	0.050	0.545	0.454	9.08
4	0.90	1	0.954	0.045	0.545	0.454	10.88
5	0.81	1	0.913	0.086	0.545	0.454	5.27

ตารางที่ 4-9 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้ปรับปรุงครั้งที่ 3

การทดสอบครั้งที่ 4

ช่วงค่าความน่าจะเป็นที่เลือก	ฐานความรู้เริ่มต้น				ฐานความรู้ปรับปรุง			
	อีเมล์ดี		อีเมล์ขยะ		อีเมล์ดี		อีเมล์ขยะ	
	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น
1	120	0	91	9	120	0	92	8
2	120	0	89	11	120	0	90	10
3	120	0	92	8	120	0	94	6
4	120	0	94	6	120	0	97	3
5	120	0	86	14	120	0	87	13

ตารางที่ 4-10 การทดสอบตัวกรองครั้งที่ 4

ฐานความรู้เริ่มต้น(ครั้งที่ 4)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	$WAcc^b$	$WErr^b$	TCR
1	0.91	1	0.959	0.040	0.545	0.454	11.35
2	0.89	1	0.950	0.050	0.545	0.454	9.08
3	0.92	1	0.963	0.036	0.545	0.454	12.61
4	0.94	1	0.972	0.027	0.545	0.454	16.81
5	0.86	1	0.936	0.063	0.545	0.454	7.20

ตารางที่ 4-11 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้เริ่มต้นครั้งที่ 4

ฐานความรู้ปรับปรุง(ครั้งที่ 4)

ค่าน้ำหนัก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	$WAcc^b$	$WErr^b$	TCR
1	0.92	1	0.963	0.036	0.545	0.454	12.61
2	0.90	1	0.954	0.045	0.545	0.454	10.08
3	0.94	1	0.972	0.027	0.545	0.454	16.81
4	0.97	1	0.986	0.014	0.545	0.454	32.43
5	0.87	1	0.940	0.059	0.545	0.454	7.69

ตารางที่ 4-12 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้ปรับปรุงครั้งที่ 4

การทดสอบครั้งที่ 5

ช่วงค่าความน่าจะเป็นที่เลือก	ฐานความรู้เริ่มต้น				ฐานความรู้ปรับปรุง			
	อีเมล์ดี		อีเมล์ขยะ		อีเมล์ดี		อีเมล์ขยะ	
	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น	ตรงประเด็น	ไม่ตรงประเด็น
1	120	0	95	5	120	0	97	3
2	120	0	93	7	120	0	95	5
3	120	0	96	4	120	0	97	3
4	120	0	97	3	120	0	99	1
5	120	0	90	10	120	0	92	8

ตารางที่ 4-13 การทดสอบตัวกรองครั้งที่ 5

ฐานความรู้เริ่มต้น(ครั้งที่ 5)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	WAcc^b	WErr^b	TCR
1	0.95	1	0.977	0.022	0.545	0.454	20.63
2	0.93	1	0.968	0.032	0.545	0.454	14.19
3	0.96	1	0.981	0.018	0.545	0.454	25.22
4	0.97	1	0.986	0.014	0.545	0.454	32.43
5	0.90	1	0.954	0.045	0.545	0.454	10.08

ตารางที่ 4-14 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้เริ่มต้นครั้งที่ 5

ฐานความรู้ปรับปรุง(ครั้งที่ 5)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	WAcc^b	WErr^b	TCR
1	0.97	1	0.986	0.014	0.545	0.454	32.43
2	0.95	1	0.977	0.022	0.545	0.454	20.63
3	0.97	1	0.986	0.014	0.545	0.454	32.43
4	0.99	1	0.995	0.004	0.545	0.454	113.5
5	0.92	1	0.963	0.036	0.545	0.454	12.61

ตารางที่ 4-15 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้ปรับปรุงครั้งที่ 5

ฐานความรู้เริ่มต้น(เฉลี่ย 5 ครั้ง)

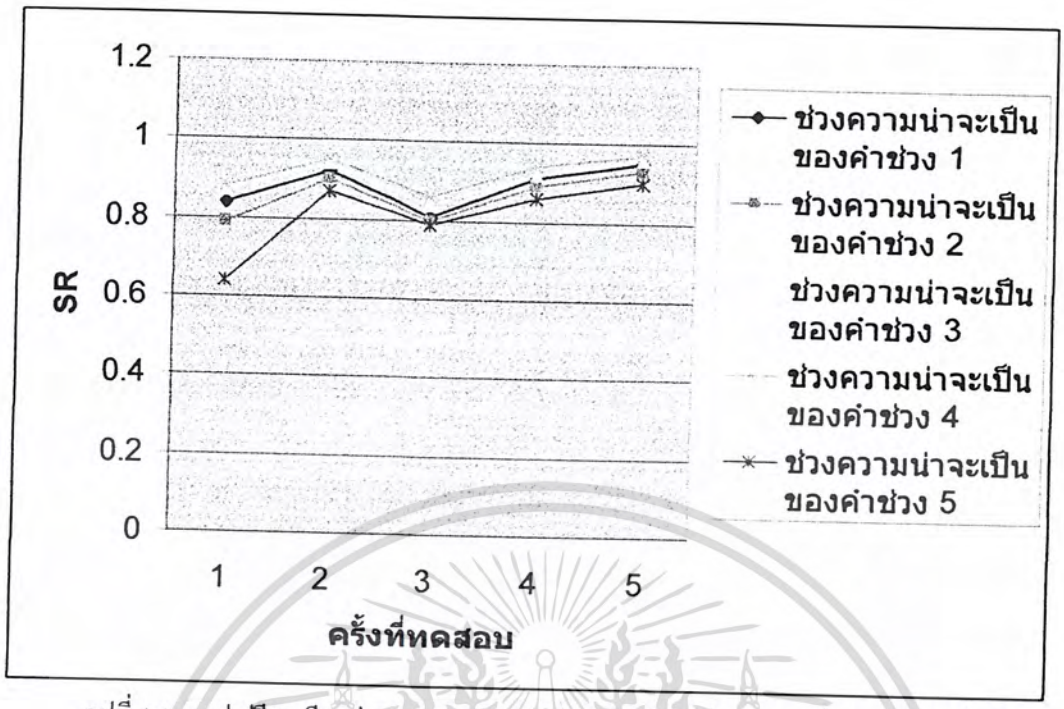
ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	WAcc^b	WErr^b	TCR
1	0.89	0.993	0.945	0.054	0.545	0.454	8.41
2	0.86	0.992	0.934	0.065	0.545	0.454	6.98
3	0.90	0.993	0.954	0.045	0.545	0.454	10.08
4	0.92	0.989	0.958	0.040	0.545	0.454	11.35
5	0.81	0.993	0.912	0.086	0.545	0.454	5.28

ตารางที่ 4-16 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้เริ่มต้นเฉลี่ย 5 ครั้ง

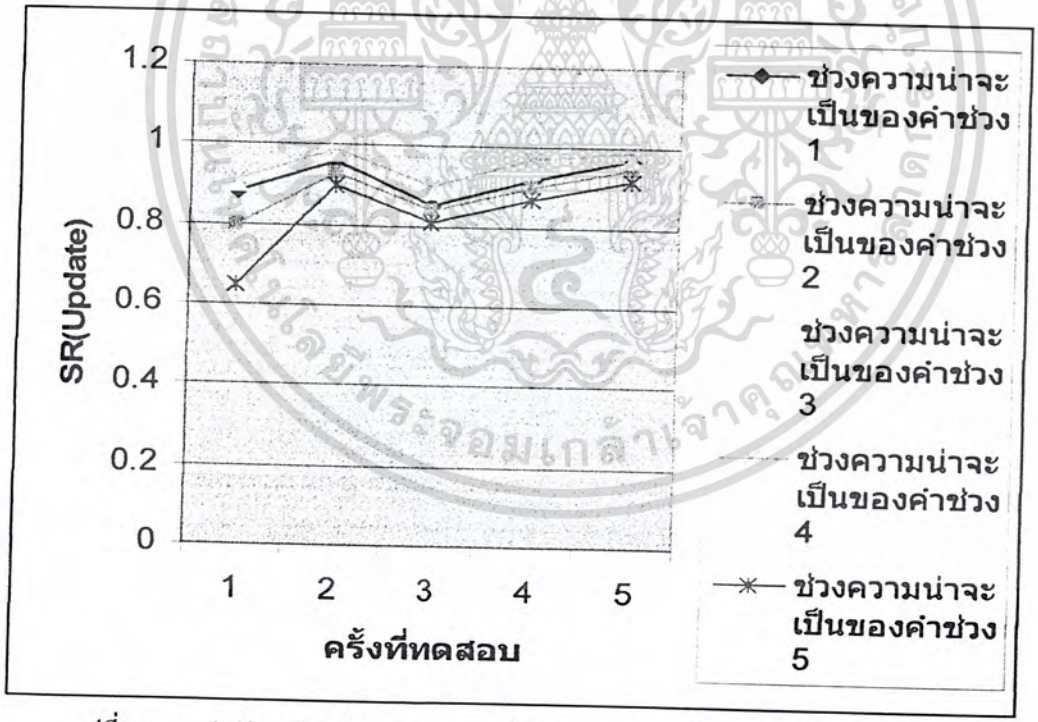
ฐานความรู้ปรับปรุง(เฉลี่ย 5 ครั้ง)

ช่วงค่าความน่าจะเป็นที่เลือก	ค่าแสดงประสิทธิภาพ						
	spam recall(SR)	spam precision(SP)	WAcc	WErr	WAcc^b	WErr^b	TCR
1	0.91	1	0.961	0.038	0.545	0.454	11.94
2	0.88	1	0.947	0.052	0.545	0.454	8.73
3	0.93	1	0.970	0.029	0.545	0.454	15.66
4	0.95	0.995	0.976	0.023	0.545	0.454	19.74
5	0.83	1	0.922	0.077	0.545	0.454	5.89

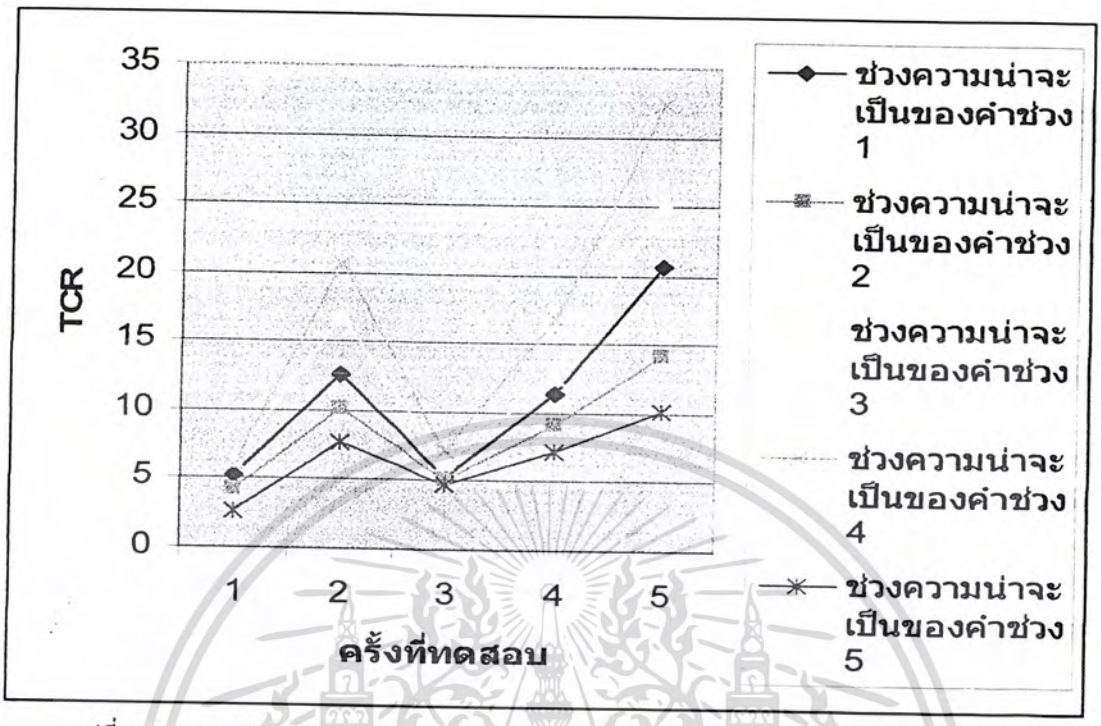
ตารางที่ 4-17 ค่าแสดงประสิทธิภาพของตัวกรองเมื่อใช้ฐานความรู้ปรับปรุงเฉลี่ย 5 ครั้ง



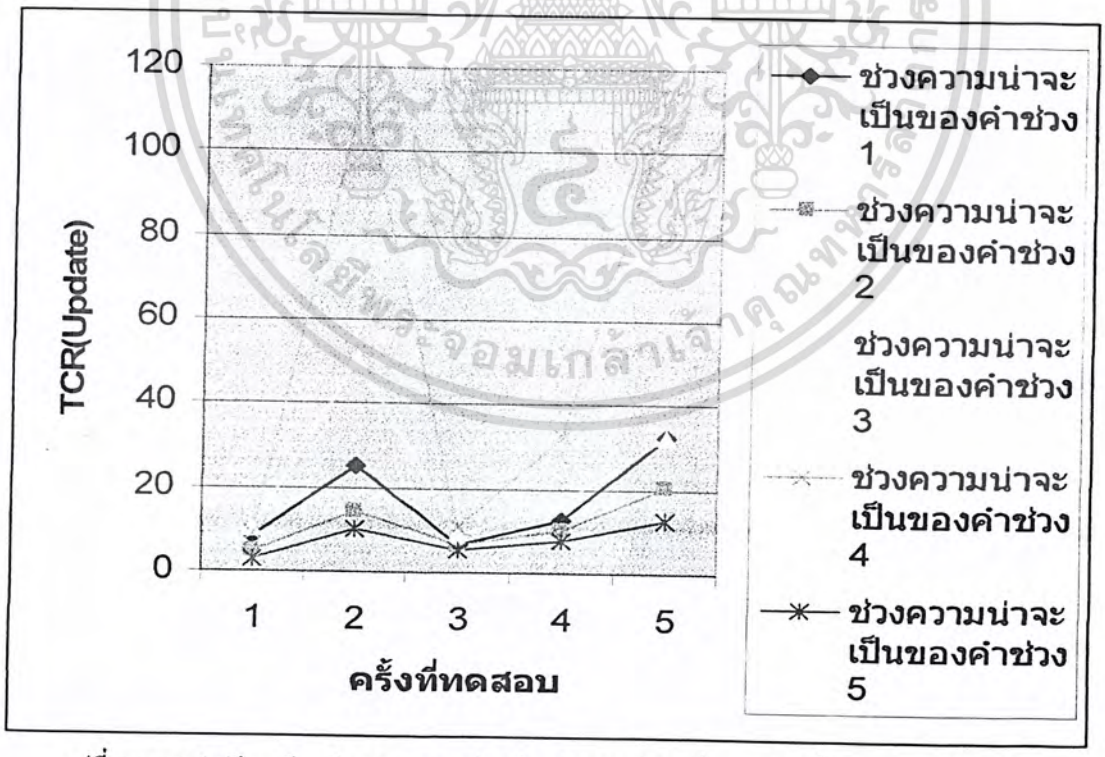
รูปที่ 4-1 กราฟเปรียบเทียบค่า SR ของแต่ละช่วงค่าความน่าจะเป็นของคำ เมื่อฐานความรู้เริ่มต้น



รูปที่ 4-2 กราฟเปรียบเทียบค่า SR ของแต่ละช่วงค่าความน่าจะเป็นของคำ เมื่อฐานความรู้ปรับปรุง



รูปที่ 4-3 กราฟเปรียบเทียบค่า TCR ของแต่ละช่วงค่าความน่าจะเป็นของคำ เมื่อฐานความรู้เริ่มต้น



รูปที่ 4-4 กราฟเปรียบเทียบค่า TCR ของแต่ละช่วงค่าความน่าจะเป็นของคำ เมื่อฐานความรู้ปรับปรุง

4.4 วิเคราะห์ผลการทดสอบ

จากการทดสอบระบบตัวกรองพบว่าการเลือกช่วงของค่าความน่าจะเป็นของคำที่ใช้ในการทดสอบอีเมล์ใหม่ที่เข้ามาในระบบมีผลต่อความสามารถในการจัดจำแนกอีเมล์

ซึ่งการเลือกช่วงค่าความน่าจะเป็นของคำควรจะหลีกเลี่ยงต่อความผิดพลาดในการจำแนกอีเมล์ดีเป็นอีเมล์ขยะ เพราะเป็นสิ่งที่ผู้ใช้งานไม่ต้องการจะให้เกิดความผิดพลาด และควรจะเลือกช่วงที่มีประสิทธิภาพในการกรองอีเมล์ขยะที่ดีด้วย จากการทดสอบช่วงความน่าจะเป็นของคำในช่วงต่างๆ พบว่าช่วงที่มีประสิทธิภาพในการกรองอีเมล์ขยะได้ดีที่สุดก็คือช่วง 0<x<=0.05 or 0.85<=x<1 เพราะเป็นช่วงที่มีการเลือกอีเมล์ดีมาคำนวณน้อย แต่ว่าในช่วงนี้จะทำให้การจำแนกอีเมล์ดีมีประสิทธิภาพลดน้อยลงไปด้วย

แต่ว่ามีช่วงหนึ่งซึ่งคือช่วง 0<x<=0.15 or 0.75<=x<1 ช่วงนี้ประสิทธิภาพในการกรองอีเมล์ขยะจะน้อยกว่าช่วง 0<x<=0.05 or 0.85<=x<1 เพียงเล็กน้อยแต่ประสิทธิภาพในการกรองอีเมล์ดีมีประสิทธิภาพที่มากกว่า และเมื่อนำช่วงนี้ไปเปรียบเทียบกับช่วงอื่นๆ ก็มีประสิทธิภาพในการกรองที่มากกว่า

จากการทดสอบยังสามารถสรุปอีกได้ว่าถ้ามีการให้ตัวกรองได้เรียนรู้อีเมล์เพิ่มเติมจะทำให้ประสิทธิภาพของตัวกรองมีมากขึ้น

4.5 สรุปผลการทดสอบ

จากการพัฒนาตัวกรองพบว่าการเลือกช่วงของค่าความน่าจะเป็นของคำเพื่อนำมาทดสอบอีเมล์ จะมีผลต่อประสิทธิภาพในการจำแนกอีเมล์ และการให้ตัวกรองได้เรียนรู้อีเมล์เพิ่มเติมขึ้นเรื่อยๆ ก็จะมีผลทำให้ตัวกรองประสิทธิภาพที่เพิ่มขึ้นตามไปด้วย

บทที่ 5
บทวิจารณ์และสรุปผล

5.1 วิจารณ์โครงงาน

5.1.1 การพัฒนาโปรแกรม

ภาษา Java ที่ใช้ในการพัฒนาโปรแกรมมีความสามารถในพัฒนาโปรแกรม เพราะเป็นภาษาที่สามารถนำไปใช้ได้กับหลายแพลตฟอร์ม และเนื่องจากเป็นภาษาที่เป็นแบบออบเจค ทำให้ง่ายต่อการออกแบบและสร้างโปรแกรม รวมทั้งยังมีชบวนการที่มีความสามารถทางด้านการตัดคำซึ่งเป็นสิ่งสำหรับโครงงานนี้

5.1.2 การศึกษาวิธีการจำแนกอีเมล์

เนื่องจากมีอยู่หลากหลายวิธีที่สามารถนำมาใช้ในการจำแนกอีเมล์ แต่ว่าบางวิธีก็ไม่สามารถที่จะจำแนกอีเมล์ได้อย่างดีพอ และบางวิธีที่โปรแกรมของอีเมล์ของผู้ใช้สามารถที่จะตั้งค่าเพื่อใช้ในการกรองได้อยู่แล้วจึงทำการศึกษาหาวิธีการที่ค่อนข้างมีประสิทธิภาพ

จากการหาวิธีที่สามารถกรองอีเมล์ขยะได้อย่างมีประสิทธิภาพ พบว่าวิธีการทางสถิติจะสามารถนำมาประยุกต์และมีประสิทธิภาพเป็นอย่างดี แต่ว่าวิธีการทางสถิติมีข้อเสียที่ต้องมีการทำให้ตัวกรองเรียนรู้ข้อมูลซึ่งต้องใช้เวลาในการทำให้ตัวกรองเรียนรู้เป็นระยะเวลาหนึ่ง ตัวกรองจึงจะมีประสิทธิภาพในระดับหนึ่ง

ซึ่งวิธี Bayesian เป็นวิธีทางสถิติที่คิดว่ามีประสิทธิภาพและสามารถหาข้อมูลได้ไม่ยากนัก

5.2 สรุปผลโครงงาน

โครงงานที่พัฒนาสามารถนำไปใช้ได้จริงบนเครื่องผู้ใช้ที่ต้องการระบบกรองอีเมล์ขยะ ซึ่งสามารถใช้ร่วมกับโปรแกรมอีเมล์ไคลเอนต์ที่ติดต่อกับเมล์เซิร์ฟเวอร์ที่เป็น POP3

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อไป

5.3.1 ถ้าผู้ที่สนใจที่จะทำโครงงานเกี่ยวกับการกรองอีเมล์ขยะ ควรจะติดตามการพัฒนาการของอีเมล์ขยะซึ่งมีการพัฒนารูปแบบที่มีความสามารถขึ้นเรื่อยๆ

5.3.2 มีวิธีการสถิติที่นอกเหนือจาก Bayesian ซึ่งน่าสนใจที่จะนำวิธีการเหล่านั้นลองมาประยุกต์ใช้เพื่อสร้างตัวกรองอีเมล์ขยะ ซึ่งอาจจะประสิทธิภาพที่คาดไม่ถึง

5.3.3 วิธีการ Bayesian เป็นวิธีการที่จัดการกับคำ โดยใช้เงื่อนไขในการเลือกคำ ซึ่งอาจจะมีเงื่อนไขต่างๆที่ผู้พัฒนาได้มองข้ามไป ซึ่งเงื่อนไขเหล่านั้นอาจจะมีผลต่อการจำแนกอีเมล์ขยะที่ดีขึ้น จึงควรจะพัฒนาเงื่อนไขต่างๆให้ครอบคลุมมากยิ่งขึ้น

5.4 ปัญหาและอุปสรรค

5.4.1 โครงงานมีการใช้สถิติในการคำนวณความน่าจะเป็นของคำ และมีการใช้ความน่าจะเป็นของคำมาทำการคัดเลือกคำเพื่อจะใช้ในการจำแนกอีเมล์ จึงทำให้การเลือกช่วงของความน่าจะเป็นของคำทำได้ยากเพราะมีหลากหลายวิธีในการเลือกค่าความน่าจะเป็นของคำ

5.4.2 โครงงานที่ทำอยู่จะต้องใช้ฐานความรู้ที่เกี่ยวกับอีเมล์ทั่วไป และอีเมล์ขยะเป็นจำนวนมากซึ่งทำให้ต้องใช้ระยะเวลาที่พอสมควรในการรวบรวมอีเมล์ตัวอย่าง

5.4.3 โครงงานตั้งอยู่บนพื้นฐานของคำ ซึ่งการจะตัดคำจากอีเมล์จะต้องใช้การสังเกตรูปแบบทั่วๆไปของอีเมล์ทำให้ต้องใช้ระยะเวลานานในการศึกษารูปแบบของอีเมล์

บรรณานุกรม

[1] ดร. วีระศักดิ์ ซึ่งถาวร (2543) : "Java Programming Volume 1"

[2] ดร. วีระศักดิ์ ซึ่งถาวร (2545) : "Java Programming Volume 2"

[3] ดร. วีระศักดิ์ ซึ่งถาวร (2547) : "Java Programming Volume 3"

[4] Kai wei(2003) : "A Naive Bayes Spam Filter"

[5] Paul Wolfe , Charlie Scott , Mike W. Erwin (2004) : "Anti Spam Toolkit"

[6] RFC 1939 : "pop3"

เว็บไซต์อ้างอิง

http://www.paulgraham.com

http://spamassassin.org

http://www.sourceforge.net

http://www.rfc.net