การปรับปรุงระบบกรองอีเมลสแปมสำหรับภาษาไทยด้วยวิธีการทางสถิติ

นายเฉลิมพล ณ สงขลา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2552

ENHANCING SPAM EMAIL FILTER SYSTEM FOR THAI USING STATISTICAL METHOD

Mr. Chalermpol Na Songkhla

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2009

หัวข้อวิทยานิพนธ์	การปรับปรุงระบบกรองอีเมลสแปมสำหรับภาษาไทยด้วยวิธีการทางสถิติ
โดย	นายเฉลิมพล ณ สงขลา
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	อาจารย์ ดร. เกริก ภิรมย์โสภา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(อาจารย์ ดร.เกริก ภิรมย์โสภา)

..... กรรมการ
(อาจารย์ ดร. นัทที นิภานันท์)

..... กรรมการภายนอกมหาวิทยาลัย
(อาจารย์ ดร.ธนัสนี เพียรตระกูล)

เฉลิมพล ณ สงขลา : การปรับปรุงระบบกรองอีเมลสแปมสำหรับภาษาไทยด้วยวิธีการทางสถิติ. (ENHANCING SPAM EMAIL FILTER SYSTEM FOR THAI USING STATISTICAL METHOD) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : อ.ดร.เกริก ภิรมย์โสภา, 109 หน้า.

วิทยานิพนธ์นี้ได้ทำการศึกษาปัญหาอีเมลสแปมและวิธีการแก้ไขปัญหาอีเมลสแปม โดยมุ่งศึกษาวิธีการแก้ไขปัญหาอีเมลสแปมสำหรับภาษาไทย วิธีการแก้ไขปัญหาอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์โดยทั่วไปนั้นกรองอีเมลสแปมภาษาไทยได้ไม่มีประสิทธิภาพ เนื่องจากภาษาไทยมีลักษณะเฉพาะ ยกตัวอย่างเช่น ไม่มีขอบเขตแบ่งคำที่แน่นอน เป็นต้น จึงจำเป็นต้องใช้โปรแกรมตัดคำไทยเพื่อช่วยประมวลผลคำไทย ส่วนหนึ่งของวิทยานิพนธ์นี้ได้นำเสนอวิธีการปรับปรุงระบบกรองอีเมลที่มีระบบการเรียนรู้แบบเบย์สำหรับภาษาไทย ผลการทดสอบแสดงให้เห็นว่าระบบกรองอีเมลที่มีระบบการเรียนรู้แบบเบย์และใช้โปรแกรมตัดคำไทยนั้นมีประสิทธิภาพสูงขึ้น อย่างไรก็ตามความรู้ที่ระบบการเรียนรู้ได้เรียนรู้นั้นไม่สามารถนำมาใช้ร่วมกันระหว่างเครื่องอีเมลแม่ข่ายได้ จุดประสงค์ของวิทยานิพนธ์นี้ได้นำเสนอวิธีการสร้างกฎด้วยวิธีการทางสถิติ ซึ่งเป็นวิธีการซึ่งรวมข้อดีของวิธีการแก้ไขปัญหาอีเมลสแปมด้วยกฎและวิธีการแก้ไขปัญหาอีเมลสแปมที่มีระบบการเรียนรู้เข้าด้วยกัน กฎที่สร้างได้สามารถนำมาใช้ร่วมกันระหว่างเครื่องแม่ข่ายอีเมล และสามารถรับมือกับรูปแบบอีเมลสแปมที่หลากหลายได้ ผลการทดสอบแสดงให้เห็นว่าวิธีการที่นำเสนอสามารถปรับตัวเพื่อกรองอีเมลสแปมภาษาไทยได้

ภาควิชา : .....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต : .....เฉลิมพล ณ สงขลา.....
สาขาวิชา : ..วิศวกรรมคอมพิวเตอร์... ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก : .....
ปีการศึกษา : ....2552....

5170272321 : MAJOR COMPUTER ENGINEERING

KEYWORDS: SPAM DETECTION / THAI

CHALERMPOL NA SONGKHLA : ENHANCING SPAM EMAIL FILTER SYSTEM FOR THAI USING STATISTICAL METHOD. THESIS ADVISOR : KRERK PIROMSOPA, Ph.D., 109 pp.

This thesis studies the spam-email problems and the anti-spam solutions by focusing on anti-spam solutions for Thai-spam email. The general Bayesian-learning-anti-spam solution filters Thai-spam email ineffectively. Since Thai language has specific characteristics (i.e. no word boundary), word segmentation should be applied in order to process the Thai words correctly. One part of this thesis is to enhance Bayesian learning for Thai spam detection. The result of this part shows that Bayesian learning spam detection with Thai word segmentation program can filter Thai spam more effectively. However, the knowledge cannot be shared among mail servers. The goal of this thesis is to generate rules from statistical method which combines the advantage of rule-based method and the advantage of learning method. The generated rules can be shared among mail servers and can keep up with the variations of spam email. The result shows that our proposed method can adaptively filter Thai email spam.

Department: ..Computer Engineering.. Student's Signature : Chalermpol Na Songkhla.....
Field of Study :.Computer Engineering. Advisor's Signature :.....K. Piroj.....
Academic Year : ...2009....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์อย่างยิ่งของอาจารย์ ดร. เกริก ภิรมย์โสภา อาจารย์ที่ปรึกษาซึ่งท่านได้ให้ความรู้ คำปรึกษา เสนอแนะแนวทางอันเป็นประโยชน์ต่องานวิจัยนี้และสนับสนุนเป็นอย่างดีจนทำให้การวิจัยในครั้งนี้สำเร็จเสร็จสมบูรณ์ออกมาด้วยดี

ขอขอบพระคุณอาจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์ อาจารย์ ดร. นัททที นิภานันท์ และอาจารย์ ดร. ธนัสนี เพียรตระกูลกรรมการสอบวิทยานิพนธ์ที่กรุณาเสียสละเวลาให้คำแนะนำตรวจสอบและชี้จุดบกพร่องที่ควรแก้ไขวิทยานิพนธ์ฉบับนี้

สุดท้ายนี้ผู้เสนอวิทยานิพนธ์ขอกราบขอบพระคุณคุณพ่อ คุณแม่และญาติ ผู้คอยให้กำลังใจ สนับสนุนอย่างเต็มที่เสมอมาและเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์สำเร็จได้ด้วยดี

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

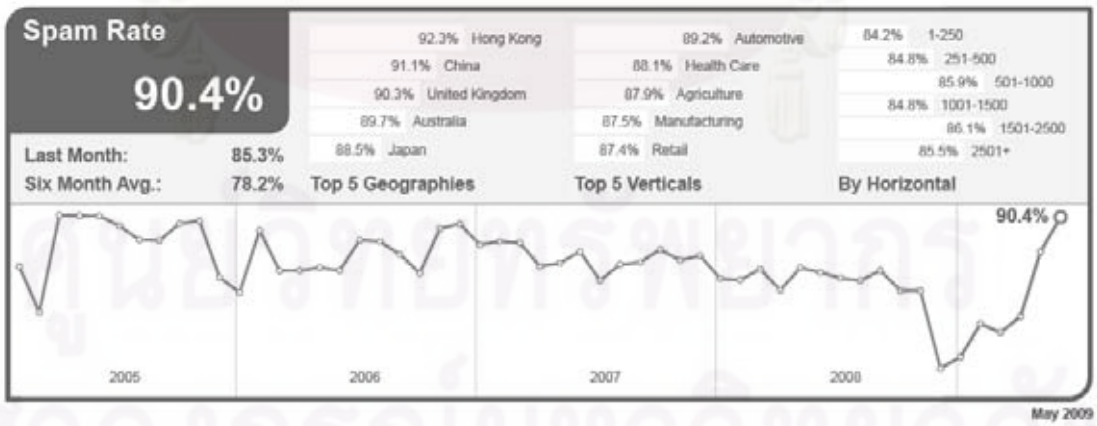
บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันอีเมลเป็นการติดต่อสื่อสารที่มีความสะดวก รวดเร็ว และมีค่าใช้จ่ายน้อย จากคุณสมบัติดังกล่าวทำให้มีผู้ใช้งานอีเมลบางกลุ่มใช้อีเมลในการติดต่อสื่อสารเพื่อผลประโยชน์ทางการค้า เช่น การส่งอีเมลโฆษณาขายสินค้า เป็นต้น ซึ่งเรียกว่าการส่งอีเมลสแปม

อีเมลสแปมมีลักษณะสำคัญ คือ เป็นอีเมลที่ถูกส่งไปหาผู้รับอีเมลจำนวนมาก (Bulk Email) เป็นอีเมลที่ผู้รับอีเมลไม่ต้องการ (Unsolicited Email) และเป็นอีเมลที่มีเนื้อหาเกี่ยวกับการโฆษณาขายสินค้า (Commercial Email)

จากรูปที่ 1 รายงานข้อมูลอีเมลสแปมของบริษัทไซแมนเทคพบว่าในเดือนพฤษภาคม ปี 2009 มีจำนวนอีเมลสแปมคิดเป็น 90.4 เปอร์เซ็นต์ของจำนวนอีเมลที่ถูกส่งออกมาทั้งหมด (Global Email) [1] และจากรายงานข้อมูลของสำนักบริการสำนักเทคโนโลยีสารสนเทศภาครัฐ (Government Information Technology Services) ซึ่งเป็นหน่วยงานของรัฐบาลไทยที่ให้บริการข้อมูลและบริการด้านเทคโนโลยีสารสนเทศพบว่ามีอีเมลสแปมจำนวนมากคิดเป็น 77.22 เปอร์เซ็นต์ของจำนวนอีเมลที่ถูกส่งออกมาทั้งหมดในช่วงเดือนมกราคมปี 2551 [2] จำนวนอีเมลสแปมจำนวนมากที่ถูกส่งออกมาจะทำให้เกิดความรำคาญต่อผู้รับอีเมล และเกิดผลเสียต่อระบบมากมาย



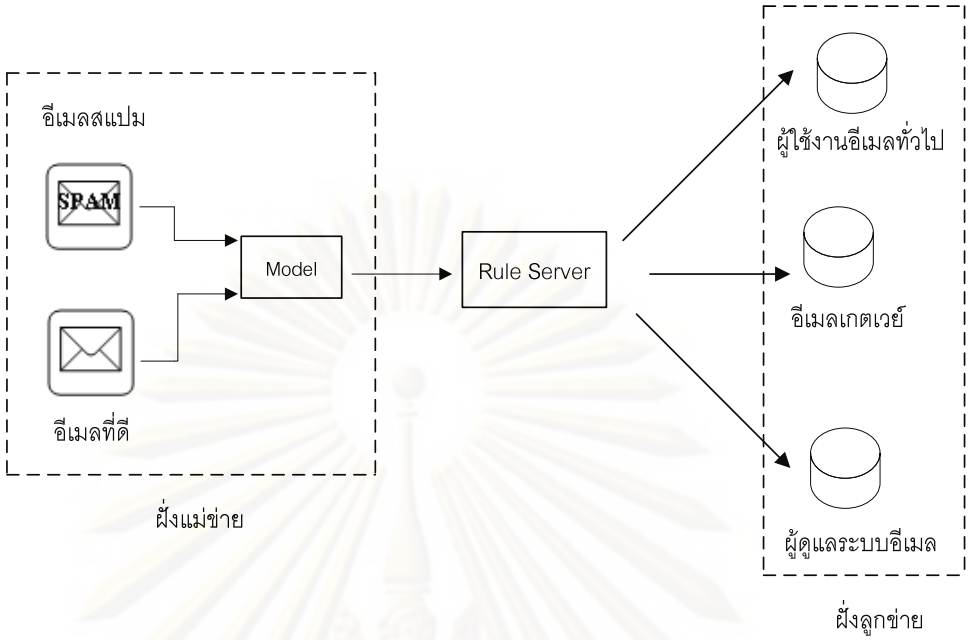
รูปที่ 1 ปริมาณอีเมลสแปมทั้งหมดในระบบ

ปัจจุบันมีการพัฒนาวิธีการแก้ปัญหาอีเมลสแปมขึ้นหลายประเภท ยกตัวอย่างเช่น วิธีการป้องกันการส่งอีเมลสแปมบนฝั่งของผู้ส่งอีเมล (Sender Side) ซึ่งวิธีการในประเภทนี้ไม่ได้ผลลัพธ์ที่ดีในทางปฏิบัติ จึงมีการพัฒนาวิธีการแก้ปัญหาอีเมลสแปมบนฝั่งของผู้รับอีเมล (Receiver Side) ตัวอย่างเช่น วิธีการค้นหาลักษณะสำคัญหรือการสร้างกฎขึ้นมาป้องกันอีเมลสแปมซึ่งเป็นวิธีการที่มีประสิทธิภาพวิธีการหนึ่งของการแก้ปัญหาอีเมลสแปมบนฝั่งของผู้รับ ข้อดีของการสร้างกฎขึ้นมาคือสามารถนำกฎมาใช้ร่วมกันระหว่างเครื่องแม่ข่ายได้ง่าย แต่ข้อเสียที่สำคัญคือผู้ส่งอีเมลสแปม (Spammer) จะทำการหลีกเลี่ยงกฎที่สร้างไว้ตลอดเวลา ทำให้จำเป็นที่จะต้องดูแลให้กฎทันสมัยอยู่ตลอดเวลา จึงมีการพัฒนาวิธีการแก้ปัญหาโดยใช้ระบบการเรียนรู้ ตัวอย่างเช่น ระบบการเรียนรู้แบบเบย์ซึ่งเป็นวิธีทางสถิติ ข้อดีของวิธีการใช้ระบบการเรียนรู้คือสามารถเรียนรู้รูปแบบใหม่ๆ ของอีเมลสแปมได้ง่าย

แต่วิธีการแก้ปัญหาอีเมลสแปมโดยใช้ระบบการเรียนรู้แบบเบย์ที่มีอยู่ในปัจจุบันนั้นมีประสิทธิภาพด้อยลงเมื่อวิเคราะห์คำไทย โดยพบว่าระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์ทั่วไปนั้น จะตัดคำภาษาไทยออกมาเป็นคำที่อ่านไม่รู้เรื่อง ทำให้ได้ระบบกรองอีเมลที่ไร้ประสิทธิภาพ ส่วนหนึ่งของวิทยานิพนธ์นี้จึงจัดทำขึ้นเพื่อนำเสนอระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีระบบการเรียนรู้แบบเบย์และใช้ระบบตัดคำไทย เพื่อเพิ่มประสิทธิภาพกรองอีเมลสแปมภาษาไทย

ผู้เสนอวิทยานิพนธ์ได้แสดงรายละเอียดการออกแบบระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีระบบการเรียนรู้แบบเบย์และใช้ระบบตัดคำไทยไว้ในบทที่ 3 และแสดงรายละเอียดการพัฒนาระบบไว้ในบทที่ 5

แต่ข้อเสียที่สำคัญของวิธีการแก้ปัญหาอีเมลสแปมโดยใช้ระบบการเรียนรู้ คือ เมื่อระบบการเรียนรู้ได้เรียนรู้ข้อมูลขนาดใหญ่ ฐานข้อมูลจะมีขนาดใหญ่มาก ทำให้สิ้นเปลืองเนื้อที่ในการเก็บอย่างมาก และการนำความรู้ (Knowledge) ที่ระบบได้เรียนรู้นั้นมาใช้ร่วมกันระหว่างเครื่องแม่ข่ายอีเมลเป็นไปได้ยาก วิทยานิพนธ์นี้จึงจัดทำขึ้นเพื่อพัฒนาวิธีการแก้ปัญหาอีเมลสแปมภาษาไทยที่สามารถสร้างกฎภาษาไทยด้วยวิธีการทางสถิติ ซึ่งเป็นการนำข้อดีของวิธีการแก้ปัญหาอีเมลสแปมแบบการสร้างกฎและข้อดีของวิธีการแก้ปัญหาแบบระบบการเรียนรู้เข้าด้วยกัน



รูปที่ 2 ระบบศูนย์กลางการป้องกันอีเมลสแปมภาษาไทย

จากรูปที่ 2 การพัฒนาวิธีการแก้ปัญหาอีเมลสแปมที่สามารถสร้างกฎภาษาไทยด้วยวิธีการทางสถิตินี้มีความสามารถที่จะเป็นระบบศูนย์กลางการป้องกันอีเมลสแปมภาษาไทยได้โดยทำการรับข้อมูลอีเมลที่ดี และข้อมูลอีเมลสแปมภาษาไทยซึ่งมาจากแหล่งข้อมูลต่างๆ เช่น การรายงานจากผู้ใช้งานทั่วไป อีเมลเกตเวย์ (Email Gateway) เป็นต้น จากนั้นระบบจะประมวลผลข้อมูลและสร้างกฎป้องกันอีเมลสแปมภาษาไทยออกมาให้ผู้ใช้งานหรืออีเมลเกตเวย์สามารถนำกฎไปติดตั้งเพื่อเพิ่มประสิทธิภาพการกรองอีเมลสแปมภาษาไทย

ตัวระบบ (Model) ในรูปที่ 2 นั้น ผู้เสนอวิทยานิพนธ์ได้แสดงรายละเอียดการออกแบบกระบวนการต่างๆ ภายในตัวระบบ คือ กระบวนการเตรียมข้อมูล (Preprocessing) กระบวนการตัดคำ (Tokenizing) กระบวนการคัดเลือกคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปม (Selecting spam-liked pattern) และกระบวนการกำหนดคะแนนให้กับกฎ (Scoring) ไว้ในบทที่ 4 และได้แสดงรายละเอียดของการพัฒนาตัวระบบไว้ในบทที่ 6

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาวิธีการแก้ปัญหาอีเมลสแปมภาษาไทยแบบวิธีการเรียนรู้แบบเบย์
2. เพื่อศึกษาวิธีการแก้ปัญหาอีเมลสแปมแบบวิธีการสร้างกฎโดยใช้วิธีการทางสถิติ (Statistical Ruled-base)
3. เพื่อพัฒนาวิธีการแก้ปัญหาอีเมลสแปมที่เป็นภาษาไทยได้อย่างมีประสิทธิภาพ

4. เพื่อวัดประสิทธิภาพของกฎภาษาไทยที่สร้างขึ้นมา

1.3 ขอบเขตของการวิจัย

1. วิทยานิพนธ์นี้ทำการทดลองกับระบบกรองอีเมลที่มีระบบการเรียนรู้แบบเบย์และทดสอบกับโปรแกรมกรองอีเมลสแปมแอสแซสซิน (SpamAssassin) เท่านั้น
2. วิทยานิพนธ์นี้ทำการสร้างกฎป้องกันอีเมลสแปมภาษาไทยให้กับโปรแกรมกรองอีเมลสแปมแอสแซสซินเท่านั้น
3. ข้อมูลตัวอย่างที่นำมาทดสอบในวิทยานิพนธ์นี้จะสนใจเพียงส่วนเนื้อหา (Content) หรือคำที่ปรากฏในอีเมลมิได้สนใจส่วนหัวเรื่อง (Header)
4. กฎที่สร้างขึ้นมีรูปแบบอักขระ (Character Encoding) เป็น tis-620 และจำนวนกฎที่นำมาทดสอบ คือ กฎในส่วนหัวข้อเรื่องจำนวน 100 กฎและกฎในส่วนเนื้อหาจำนวน 100 กฎ

1.4 ข้อกังวลเกี่ยวกับข้อมูลส่วนบุคคล

เนื่องจากในวิทยานิพนธ์นี้ข้อมูลตัวอย่างจะมาจากแหล่งข้อมูลที่หลากหลาย แต่อย่างไรก็ตามการทดลองในวิทยานิพนธ์นี้จะสนใจเพียงส่วนเนื้อหาเท่านั้นมิได้สนใจส่วนหัวเรื่องและการทดลองต่างๆ กระทำโดยเครื่องคอมพิวเตอร์จะไม่มีการเปิดเผยข้อมูลในส่วนเนื้อหาและส่วนหัวเรื่องออกมา

1.5 ขั้นตอนและวิธีดำเนินการวิจัย

1.5.1 ขั้นตอนศึกษางานวิจัยที่เกี่ยวข้อง

ศึกษาปัญหาอีเมลสแปม และงานวิจัยที่เกี่ยวข้อง เช่น วิธีการแก้ไขปัญหาอีเมลสแปม เป็นต้น

1.5.2 ขั้นตอนศึกษาการใช้เครื่องมือที่ใช้ในงานวิจัย

ในวิทยานิพนธ์นี้จำเป็นต้องสร้างระบบต่างๆ ขึ้นมา เช่น ระบบการส่ง-รับอีเมล, ระบบการป้องกันอีเมลสแปม เป็นต้น ซึ่งเครื่องมือที่ใช้สร้างระบบดังกล่าวนั้นมีความซับซ้อนและจะต้องมีการปรับแต่งค่าต่างๆ เพื่อให้ระบบสามารถทำงานได้

1.5.3 ขั้นตอนเตรียมการและออกแบบการทดลอง

ขั้นตอนนี้จะทำการเก็บรวบรวมอีเมลสแปมภาษาไทย เพื่อเป็นชุดข้อมูลตัวอย่างที่ใช้สำหรับเป็นข้อมูลฝึก และข้อมูลทดสอบ

ทำการปรับปรุงระบบกรองอีเมลสแปมที่มีการเรียนรู้แบบเบย์ทั่วไปให้เป็นระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีการเรียนรู้แบบเบย์ และใช้โปรแกรมตัดคำไทย

ทำการออกแบบและพัฒนาตัวแบบ (Model) สำหรับการสร้างกฎภาษาไทยด้วยวิธีทางสถิติ

1.5.4 ขั้นตอนทดลองเพื่อทดสอบประสิทธิภาพของระบบที่สร้างขึ้น

เมื่อได้เตรียมความพร้อมในการทดลองแล้ว ทั้งระบบกรองอีเมลสแปมสำหรับภาษาไทยที่เป็นระบบการเรียนรู้แบบเบย์ซึ่งใช้โปรแกรมตัดคำไทยและกฎภาษาไทยที่สร้างขึ้นมาจะถูกทดสอบกับชุดตัวอย่างข้อมูลด้วยวิธีการทดสอบไขว้ k กลุ่ม (k-fold Cross Validation) บันทึกผลการทดลองนั้น เพื่อนำไปสรุป

1.5.5 ขั้นตอนสรุปผลการทดลอง ผลที่ได้รับจากงานวิจัยรวมถึงข้อเสนอแนะ และจัดทำวิทยานิพนธ์

ในขั้นตอนนี้จะนำผลการทดลองที่ได้มาสรุปผลและข้อเสนอแนะ เพื่อนำมาจัดทำวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจปัญหาอีเมลสแปม และวิธีการแก้ปัญหาอีเมลสแปมรูปแบบต่างๆ
2. ได้รับความรู้เกี่ยวกับการแก้ปัญหาอีเมลสแปมแบบที่มีระบบการเรียนรู้ และการแก้ปัญหาอีเมลสแปมด้วยวิธีการสร้างกฎ รวมทั้งข้อดีและข้อเสีย
3. ได้ระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีประสิทธิภาพ
4. สามารถนำความรู้จากผลการวิจัยนี้ไปประยุกต์ใช้จริงต่อไปในอนาคต

1.7 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 8 บทดังต่อไปนี้ บทที่ 1 บทนำ บทที่ 2 ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง บทที่ 3 หลักการสร้างและออกแบบระบบกรองอีเมลสแปมภาษาไทยที่มีระบบการเรียนรู้แบบเบย์และใช้โปรแกรมตัดคำไทย บทที่ 4 หลักการสร้างกฎด้วยวิธีการทางสถิติ บทที่ 5 การพัฒนาระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีระบบการเรียนรู้แบบเบย์ บทที่ 6 การพัฒนาระบบการสร้างกฎด้วยวิธีการทางสถิติ บทที่ 7 การทดสอบประสิทธิภาพของระบบ บทที่ 8 บทสรุปงานวิจัย

บทที่ 1 บทนำกล่าวถึงความเป็นมาและความสำคัญของปัญหาอีเมลสแปม รวมถึงวัตถุประสงค์ของการวิจัย

บทที่ 2 ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องกล่าวถึงพื้นฐานความรู้ของระบบอีเมล ความหมายของอีเมลสแปม ลักษณะสำคัญของอีเมลสแปม อธิบายวิธีการแก้ไขปัญหาอีเมลสแปมรูปแบบต่างๆ รวมทั้งข้อดีและข้อเสีย นอกจากนี้ได้กล่าวถึงงานวิจัยที่เกี่ยวข้อง

บทที่ 3 หลักการสร้างและการออกแบบระบบกรองอีเมลสแปมภาษาไทยที่มีระบบการเรียนรู้แบบเบย์และใช้โปรแกรมตัดคำไทยกล่าวถึงพื้นฐานความรู้การเรียนรู้แบบเบย์ การนำหลักการเรียนรู้แบบเบย์มาประยุกต์ใช้ในการกรองอีเมลสแปม ปัญหาภาษาไทยกับระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์ และการออกแบบระบบกรองอีเมลสแปมภาษาไทยที่มีระบบการเรียนรู้แบบเบย์และใช้โปรแกรมตัดคำไทยเพื่อปรับปรุงประสิทธิภาพการกรองอีเมลสแปมภาษาไทย

บทที่ 4 หลักการสร้างกฎด้วยวิธีการทางสถิติกล่าวถึงกฎในโปรแกรมกรองอีเมลสแปมแอสแซสซิน หลักการสร้างกฎขึ้นเอง และกล่าวถึงหลักการสร้างกฎด้วยวิธีการทางสถิติซึ่งแสดงให้เห็นรายละเอียดกรอบการทำงานของระบบการสร้างกฎด้วยวิธีการทางสถิติ

บทที่ 5 การพัฒนาระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์สำหรับภาษาไทยกล่าวถึงการนำหลักการสร้างและออกแบบระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์สำหรับภาษาไทยและใช้โปรแกรมตัดคำไทยในบทที่ 3 มาใช้ในการสร้างระบบ

บทที่ 6 การพัฒนาระบบสร้างกฎด้วยวิธีการสถิติกล่าวถึงการนำหลักการสร้างกฎด้วยวิธีการทางสถิติในบทที่ 4 มาใช้ในการสร้างระบบ

บทที่ 7 การทดสอบประสิทธิภาพของระบบกล่าวถึงการทดสอบประสิทธิภาพของระบบที่ได้พัฒนาขึ้น คือ ระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์สำหรับภาษาไทย และระบบการสร้างกฎด้วยวิธีการทางสถิติ

บทที่ 8 บทสรุปงานวิจัยกล่าวถึงสิ่งที่ได้จากการวิจัย ประโยชน์ของงานวิจัย และแนวทางวิจัยต่อ

1.8 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตอบรับให้ตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง *"การเพิ่มประสิทธิภาพการจัดจำแนกอีเมลสแปมภาษาไทยด้วยโปรแกรมตัดคำไทยคูวส์: Enhancing Thai Spam Detection with CUWS"* โดย เฉลิมพล ณ สงขลา และเกริก ภิรมย์โสภา, ในงานประชุมวิชาการ "The Conference on Knowledge and Smart Technologies (KST-2009)" ณ มหาวิทยาลัยบูรพา จังหวัดชลบุรี ระหว่างวันที่ 24-25 กรกฎาคม 2552 ในหัวข้อเรื่อง *"Statistical Rules for Thai Spam Detection"* โดย เฉลิมพล ณ สงขลา และเกริก ภิรมย์โสภา, ในงานประชุมวิชาการนานาชาติ "The 2nd International Conference on Future Network (ICFN

2010)" ณ โรงแรม Yuhai Int'l Resort Apartment & SPA เมือง Sanya ประเทศจีน ระหว่างวันที่ 22-24 มกราคม 2553

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 อีเมล (E-mail)

ไปรษณีย์อิเล็กทรอนิกส์หรือจดหมายอิเล็กทรอนิกส์ (Electronic mail: E-mail) เป็นการส่งหรือรับข้อความผ่านทางเครือข่ายเชื่อมโยงระบบอิเล็กทรอนิกส์ เช่น อินเตอร์เน็ท เป็นต้น

โพรโทคอลหลักที่ใช้ในการรับหรือส่งอีเมลในเครือข่ายอินเตอร์เน็ท คือ โพรโทคอลเอสเอ็มทีพี (Simple Mail Transfer Protocol: SMTP) ซึ่งถูกนำเสนอขึ้นมาในปี 1982 [3] นอกจากนี้ยังมีโพรโทคอลสำหรับเครื่องลูกข่ายที่ใช้ในการอ่านอีเมล คือ โพรโทคอลพีโอพี (Post Office Protocol: POP) [4] และโพรโทคอลไอเอ็มเอพี (Internet Message Access Protocol: IMAP) [5] ซึ่งถูกนำเสนอในปี 1984 และปี 1996 ตามลำดับ

2.1.2 องค์ประกอบของระบบอีเมล

องค์ประกอบของระบบอีเมลมี 2 ส่วนดังนี้

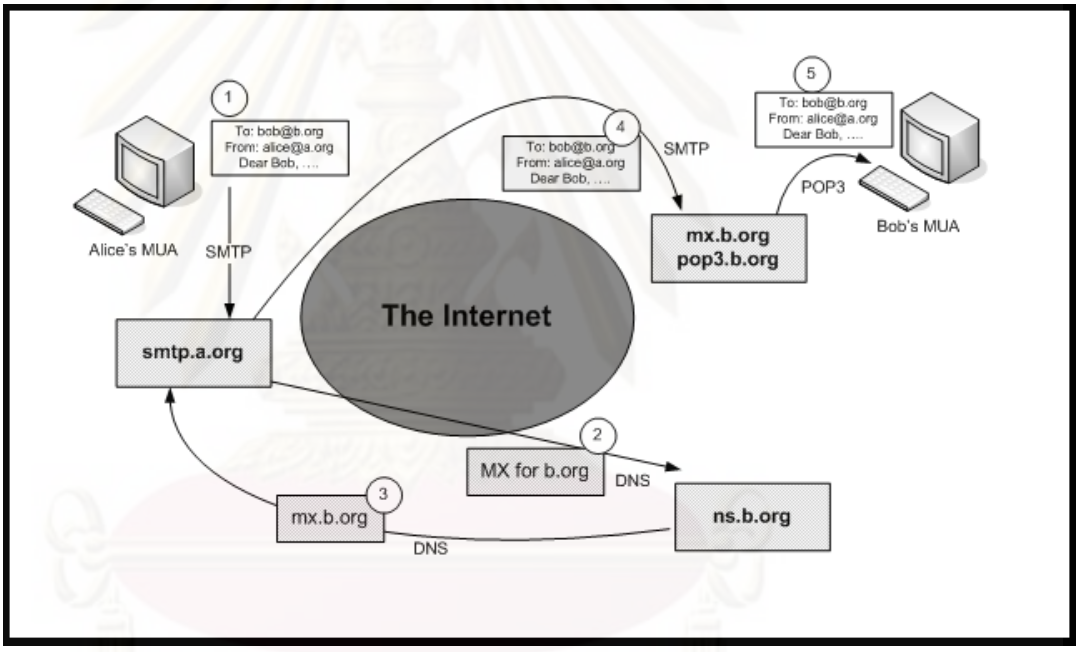
1. Mail User Agent (MUA) คือ ส่วนที่ติดต่อกับผู้ใช้และอำนวยความสะดวกในการเขียน แก้ไข เปิดอ่าน และส่งอีเมลรวมถึงการจัดเก็บอีเมลเพื่อนำมาใช้ภายหลัง ยกตัวอย่าง MUA เช่น Eudora, KMail, Outlook Express, Thunderbird เป็นต้น นอกจากนี้ยังมี MUA แบบบนเว็บไซต์หรือที่เรียกว่าเว็บเมล (Web-mail) เช่น SquirrelMail, AIM Mail, Gmail, Hotmail เป็นต้น
2. Mail Transfer Agent (MTA) คือ ส่วนที่ทำหน้าที่หาเส้นทางและส่งจดหมายไปถึงปลายทาง การตรวจสอบความถูกต้องของอีเมลทั้งของฝั่งผู้รับและผู้ส่ง ยกตัวอย่าง MTA เช่น Sendmail, Postfix, IBM, Exim เป็นต้น

2.1.3 การทำงานของระบบอีเมล

การติดต่อสื่อสารโดยใช้อีเมลระหว่างผู้ส่งและผู้รับแสดงในรูปที่ 3 มีหลักการทำงานดังนี้

1. ผู้ส่งทำการเขียนอีเมลผ่านทาง MUA จากเครื่องคอมพิวเตอร์ของตนเองโดย MUA จะจัดรูปแบบข้อความให้เป็นรูปแบบ Internet Message Format และใช้โพรโทคอลเอสเอ็มทีพี เพื่อส่งข้อความไปยัง MTA (smtp.a.org)

2. MTA ของผู้ส่งจะทำการค้นหา Mail Exchanger Records (MX records) ของโดเมนผู้รับ (b.org) ผ่านทางเครื่องแม่ข่ายดีเอ็นเอส ซึ่ง MX records จะมีข้อมูลของ MTA ของผู้รับซึ่งมีหน้าที่รับอีเมลที่ถูกส่งเข้ามา
3. เครื่องแม่ข่ายดีเอ็นเอสจะส่งค่า MX records ของโดเมนผู้รับมายัง MTA ของผู้ส่ง
4. MTA ของผู้ส่งจะทำการส่งข้อความไปยังเครื่องแม่ข่ายอีเมลของผู้รับโดยผ่านทางโพโทคอลเอสเอ็มทีพี และ MTA ของผู้รับจะนำข้อความไปเก็บในกล่องจดหมายของแต่ละคน
5. ผู้รับจะทำการนำข้อความที่เก็บไว้ในกล่องจดหมายมาเก็บไว้ที่เครื่องของผู้รับผ่านทางโพรโทคอลพีโอพี3 (POP3) หรือโพรโทคอลไอเอ็มเอพี (IMAP)



รูปที่ 3 การทำงานของระบบอีเมล

2.1.4 Internet Email Format

อีเมลมีโครงสร้างตามรูปแบบ Internet Message Format และมีส่วนประกอบดังนี้

1. ข้อมูลส่วนหัว (Header) คือข้อมูลต่างๆ ที่เกี่ยวกับอีเมล เช่น ชื่อผู้ส่ง, ชื่อผู้รับ เป็นต้น ข้อมูลในส่วนหัวนี้ มี 2 แบบ คือ ข้อมูลส่วนหัวที่อีเมลทุกฉบับจำเป็นต้องมี และข้อมูลส่วนหัวที่ถูกใช้ในอีเมลบางฉบับ ดังแสดงในตารางที่ 1

ตารางที่ 1 ตัวอย่างข้อมูลส่วนหัวที่อีเมลทุกฉบับจำเป็นต้องมี และข้อมูลส่วนหัวที่ถูกใช้ในอีเมลบางฉบับตามโครงสร้างรูปแบบ Internet Message Format
(From, to, Subject, Date คือ ข้อมูลส่วนหัวที่อีเมลทุกฉบับจำเป็นต้องมี)

ข้อมูลส่วนหัว	ความหมาย
From	ที่อยู่อีเมลของผู้ส่งอีเมล
To	ที่อยู่อีเมลของผู้รับอีเมล
Subject	หัวข้อเรื่องของอีเมล
Date	วันเวลาที่อีเมลถูกส่งออกมา
CC	ที่อยู่อีเมลที่ทำการส่งสำเนาอีเมล
BCC	ที่อยู่อีเมลที่ทำการส่งสำเนาอีเมลซ่อน
Content-Type	ข้อมูลเกี่ยวกับรูปแบบของอีเมล

2. ข้อมูลส่วนเนื้อหา (Body) คือเนื้อหาของอีเมล ในรูปที่ 4 แสดงถึงตัวอย่างอีเมลตามรูปแบบโครงสร้าง Internet Message Format

From: Johnny Smith
Date: 12-Sep-2001
Subject: Dinner Tonight?
Body:
Dear Cindy,
We will have dinner tonight. Would you like to come?

รูปที่ 4 ตัวอย่างอีเมลตามรูปแบบโครงสร้าง Internet Message Format

2.1.5 **โพรโทคอลเอสเอ็มทีพี** (SMTP)

โพรโทคอลเอสเอ็มทีพีถูกออกแบบมาเพื่อเป็นโพรโทคอลในการรับส่งอีเมลที่มีความน่าเชื่อถือ และมีประสิทธิภาพ

ขั้นตอนวิธีการรับส่งอีเมลนั้นเริ่มต้นที่เครื่องลูกข่ายเอสเอ็มทีพี (SMTP Client) จะทำการสร้างการเชื่อมต่อกับเครื่องแม่ข่ายเอสเอ็มทีพี (SMTP Server) ที่พอร์ท 25 โดยเครื่องลูกข่ายเอสเอ็มทีพีจะต้องรับผิดชอบหน้าที่ในการส่งข้อความอีเมลไปยังเครื่องแม่ข่ายเอสเอ็มทีพีให้สำเร็จ

และเครื่องแม่ข่ายเอสเอ็มทีพีจะต้องรับผิดชอบหน้าที่ในการรายงานข้อผิดพลาดหากไม่สามารถส่งข้อความอีเมลไปยังเครื่องแม่ข่ายอื่นๆ ต่อได้

โพรโทคอลเอสเอ็มทีพีนี้มีคำสั่งของโพรโทคอลเอสเอ็มทีพี (SMTP Command) เพื่อรับส่งข้อความอีเมลระหว่างเครื่องลูกข่ายเอสเอ็มทีพีและเครื่องแม่ข่ายเอสเอ็มทีพี ตัวอย่างคำสั่งในโพรโทคอลเอสเอ็มทีพีแสดงให้ตารางที่ 2

ตารางที่ 2 ตัวอย่างคำสั่งในโพรโทคอลเอสเอ็มทีพี

คำสั่ง	**คำอธิบาย**
HELO	คำสั่งนี้ใช้สำหรับเริ่มทำการติดต่อระหว่างผู้ส่งกับเครื่องแม่ข่าย
Mail	คำสั่งนี้ใช้เริ่มต้นรายละเอียดการส่งอีเมลซึ่งข้อมูลตั้งแต่ส่วนนี้จะถูกส่งเข้ากล่องจดหมาย
RECIPIENT (RCPT)	คำสั่งนี้ใช้สำหรับระบุผู้รับของอีเมล (ถูกเรียกหนึ่งครั้งสำหรับผู้รับหนึ่งคน)
DATA	คำสั่งนี้ใช้บอกว่าข้อความที่อยู่หลังคำสั่งนี้ทั้งหมดจะถือเป็นเนื้อหาของอีเมล
RESET (RSET)	คำสั่งนี้ใช้สำหรับทำการยกเลิกการส่งอีเมล

2.1.6 **อีเมลสแปม** (Spam Mail) [6]

อีเมลสแปมหรืออีเมลที่ผู้รับอีเมลไม่ต้องการรับซึ่งมีคำศัพท์อื่นๆ ที่รู้จักอย่างเป็นทางการของอีเมลสแปม ตัวอย่างเช่น Unsolicited Bulk Email (UBE) คือ อีเมลที่ถูกส่งไปหาผู้รับจำนวนมากโดยที่ผู้รับไม่ต้องการ Unsolicited Commercial Email (UCE) คือ อีเมลที่มีเนื้อหาเกี่ยวกับการโฆษณาขายสินค้าถูกส่งไปหาผู้รับโดยที่ผู้รับไม่ต้องการ เป็นต้น

ในวิทยานิพนธ์นี้ได้กำหนดลักษณะสำคัญของอีเมลสแปม คือ เป็นอีเมลที่ถูกส่งไปหาผู้รับอีเมลจำนวนมาก (Bulk Email) เป็นอีเมลที่ผู้รับอีเมลไม่ต้องการ (Unsolicited Email) และเป็นอีเมลที่มีเนื้อหาเกี่ยวกับการโฆษณาขายสินค้า (Commercial Email) ตัวอย่างอีเมลสแปมภาษาไทยแสดงดังรูปที่ 5

Subject: มาดูวิธีลดน้ำหนักแบบง่าย 3-4 kg

Body: **ลดน้ำหนัก**
ที่ได้รับการยอมรับจากหลายประเทศทั่วโลก

จะดีไหม ถ้า**ลดน้ำหนัก** ที่ได้ใช้
เป็นวิธีที่แพทย์ทั่วโลกแนะนำ

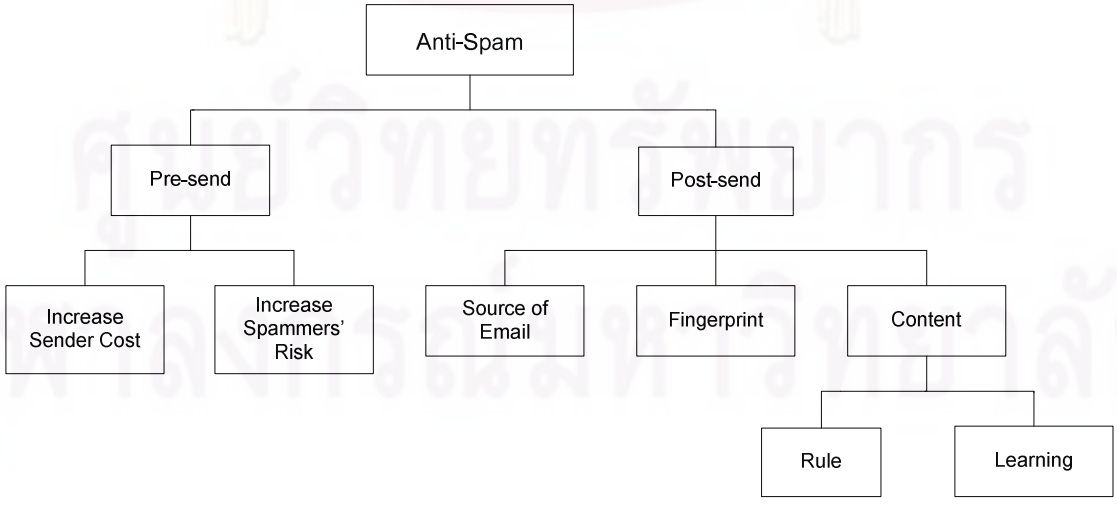
- มีอย. จาก64 ประเทศ รับรองความปลอดภัย
- การรับรองผล ภายใน 30 วัน
- ไม่ต้องใช้ยา ไม่ต้องอดอาหาร ให้เกิดผลเสียกับร่างกาย
- มีเครื่องหมายและรางวัลการรับรอง
รับประผลลัพธ์ที่ดีเยี่ยม
- แล้วถ้า เห็นผลภายใน 7 วัน**คลิกเพื่อดูข้อมูลเพิ่มเติม**

รูปที่ 5 ตัวอย่างอีเมลสแปมภาษาไทย

อีเมลสแปมที่ถูกส่งออกมาจำนวนมากก่อให้เกิดความรำคาญต่อผู้รับอีเมลทำให้สิ้นเปลืองแบนด์วิดท์ของระบบสำหรับรับอีเมลที่ไม่ต้องการจำนวนมากต่อวัน สิ้นเปลืองเนื้อที่ในการเก็บอีเมลสแปม สิ้นเปลืองเวลาในการประมวลผลของเครื่องแม่ข่ายอีเมลและบางครั้งยังมีการแนบไวรัสมาอีกด้วย

2.1.7 **วิธีการแก้ไขอีเมลสแปม** (Anti-Spam Methods)

ปัจจุบันมีการพัฒนาการแก้ปัญหาอีเมลสแปมขึ้นมาหลายประเภทซึ่งสามารถแบ่งวิธีการได้ตามรูปที่ 6



รูปที่ 6 วิธีการแก้ไขปัญหาอีเมลสแปม

จากรูปที่ 6 จะเห็นได้ว่าสามารถแบ่งออกเป็น 2 ประเภทใหญ่ๆ ดังนี้

2.1.7.1 **การแก้ไขปัญหาอีเมลสแปมบนฝั่งของผู้ส่ง** (Sender Side)

การแก้ไขปัญหาอีเมลสแปมบนฝั่งของผู้ส่ง คือ วิธีการต่างๆ นั้นจะกระทำก่อนการส่งอีเมลออกไปหรือวิธีที่จะป้องกันก่อนที่จะส่งอีเมลสแปมออกไป วิธีการรูปแบบนี้สามารถแบ่งออกเป็นรูปแบบย่อยได้อีก 2 รูปแบบย่อย ดังนี้

2.1.7.1.1 **การเพิ่มค่าใช้จ่ายในการส่งอีเมลให้กับผู้ส่งอีเมล** (Increasing Sender Costs)

วิธีการเพิ่มค่าใช้จ่ายในการส่งอีเมลให้กับผู้ส่งอีเมล ส่วนใหญ่อยู่บนพื้นฐานของเวลาการคำนวณของหน่วยประมวลผล (CPU) ผู้ส่งอีเมลจะต้องมีการคำนวณฟังก์ชันราคา (Pricing Function) ก่อนการส่งอีเมลออกไป ทำให้เกิดการหน่วงเวลาก่อนการส่งอีเมลออกไปแต่ละฉบับ โดยทั่วไปแล้วการติดต่อสื่อสารโดยใช้อีเมล ไม่มีความจำเป็นที่จะต้องสื่อสารแบบ Real Time การหน่วงเวลาก่อนส่งอีเมลออกไปจึงไม่มีผลกระทบต่อผู้ใช้อีเมลทั่วไปมากนัก แต่จะส่งผลกระทบอย่างมากต่อผู้ส่งอีเมลสแปม

ข้อดีของวิธีการนี้ คือ ไม่จำเป็นต้องแก้ไขโพรโทคอลเอสเอ็มทีพี แต่ปัญหาสำคัญของวิธีการนี้ คือ การหาฟังก์ชั่นที่ทำให้เกิดการหน่วงเวลาบนเครื่องคอมพิวเตอร์เก่าและเครื่องคอมพิวเตอร์ใหม่มีความเท่าเทียมกัน

ตัวอย่างงานวิจัยของวิธีการนี้ คือ Hashcash [7], งานวิจัย Pricing via Processing or Combatting Junk Mail [8], CAMRAM [9]

2.1.7.1.2 **การเพิ่มความเสี่ยงให้กับผู้ส่งอีเมลสแปม** (Increasing Spammers' Risk)

วิธีการเพิ่มความเสี่ยงให้กับผู้ส่งอีเมลสแปมเป็นวิธีการที่ใช้กฎหมายควบคุมการส่งอีเมลสแปม ผู้ใช้งานอีเมลบางกลุ่มมีความคิดว่าการส่งอีเมลสแปมเพื่อโฆษณาขายสินค้านั้น เป็นเสรีภาพในการโฆษณาขายสินค้า แต่ผลกระทบของการส่งอีเมลสแปมทำให้เกิดความรำคาญต่อผู้รับอีเมล ซึ่งผู้ใช้งานอีเมลอีกกลุ่มหนึ่งมีความคิดว่าเป็นการละเมิดสิทธิส่วนบุคคลของผู้รับ

ประเทศสหรัฐอเมริกาได้ออกกฎหมายเพื่อเป็นข้อกำหนดควบคุมการส่งอีเมลสแปม โดยในสหรัฐอเมริกาได้ออกกฎหมายการควบคุมการล่วงละเมิดด้วยอีเมลที่ไม่พึงประสงค์ (Controlling the Assault of Non-Solicited Pornography and Marketing Act of 2003: CAN-SPAM 2003) [10]

กฎหมายการควบคุมการล่วงละเมิดด้วยอีเมลที่ไม่พึงประสงค์ในประเทศสหรัฐอเมริกาเป็นข้อกำหนดสำหรับบุคคลที่ส่งอีเมลเพื่อการพาณิชย์และกำหนดบทลงโทษหากมีการละเมิดกฎหมายข้อกำหนดเบื้องต้นต่างๆ ตัวอย่างข้อกำหนดต่างๆ ดังนี้

- ผู้ส่งอีเมลสแปมจะต้องมีระบบยกเลิกการรับอีเมลสแปม หรือที่เรียกว่า Opt-out mechanism หมายถึง ผู้รับอีเมลสามารถยกเลิกการรับอีเมลสแปมได้ และผู้ส่งอีเมลสแปมจะต้องนำชื่อของผู้รับออกจากรายชื่อทั้งหมดภายใน 10 วัน และจะต้องไม่ส่งอีเมลสแปมให้ผู้รับอีกในอนาคต
- อีเมลสแปมต้องมีหัวข้อเรื่อง ข้อมูลส่วนหัว รวมทั้งข้อมูลที่มา-ที่ไปของอีเมล (Routing Information) ที่ถูกต้องชัดเจน
- อีเมลสแปมต้องมีที่อยู่ที่แท้จริงของผู้ส่งอีเมลแสดงอยู่
- อีเมลสแปมที่ไม่เหมาะสมต้องมีหัวข้อเรื่องสแปมชัดเจน

ตัวอย่างการบังคับใช้กฎหมายการควบคุมการล่วงละเมิดด้วยอีเมลที่ไม่พึงประสงค์ในประเทศสหรัฐอเมริกา คือ กรณีของเจเรมี่เจนเนส [11] ซึ่งเป็นสแปมเมอร์ที่มีชื่อเสียงมากที่สุดคนหนึ่งของโลกถูกพิพากษาจำคุก 9 ปีโดยศาลของรัฐเวอร์จิเนียในข้อหากระทำผิดการส่งอีเมลสแปม นายเจนเนสได้ทำการปลอมแปลงส่วนหัวของอีเมลสแปมและส่งอีเมลสแปมจำนวนมากออกไปหาผู้รับซึ่งถือเป็นความผิดทางอาญาของรัฐเวอร์จิเนีย

นอกจากนี้ประเทศในแถบยุโรปได้มีการออกกฎหมายควบคุมการส่งอีเมลสแปมด้วย [12]

อย่างไรก็ตามข้อกำหนดเบื้องต้นอาจจะเป็นผลเสียให้กับผู้รับเนื่องจากผู้ส่งอีเมลสแปมจะสามารถรู้ที่อยู่อีเมลที่มีผู้รับอยู่จริงส่งผลทำให้ผู้รับอีเมลอาจจะได้รับอีเมลสแปมจำนวนมากกว่าเดิม

โดยปกติแล้วผู้ส่งอีเมลสแปมสามารถปลอมแปลงส่วนหัวเรื่องของอีเมลเพื่อปกปิดตัวตนที่แท้จริงของอีเมลสแปมทำให้ไม่สามารถนำผู้กระทำผิดกฎหมายมาลงโทษได้ และวิธีการนี้ยังจำเป็นต้องใช้การร่วมมือกันระหว่างประเทศเพื่อออกกฎหมายควบคุมการส่งอีเมลสแปมในทุกประเทศได้

2.1.7.2 **การแก้ไขปัญหาอีเมลสแปมบนฝั่งของผู้รับ** (Receiver Side)

การแก้ไขปัญหาอีเมลสแปมบนฝั่งของผู้รับ คือ วิธีการต่างๆ นั้นจะกระทำหลังจากที่ได้ส่งอีเมลออกไปแล้วหรือวิธีที่จะกรองอีเมลสแปมซึ่งถูกส่งเข้ามาก่อนที่จะไปถึงกล่องจดหมายของผู้รับซึ่งวิธีการรูปแบบนี้สามารถแบ่งออกเป็นรูปแบบย่อยได้อีก 3 รูปแบบย่อย ดังนี้

2.1.7.2.1 **วิธีการวิเคราะห์ที่มาของอีเมล** (Source of Email)

วิธีการวิเคราะห์ที่มาของอีเมลเป็นการตรวจสอบที่มาของอีเมล ข้อมูลส่วนใหญ่ คือ ที่อยู่ไอพีของเครื่องลูกข่ายหรือโดเมนของผู้ส่งอีเมล วิธีการในประเภทนี้มีหลายแบบ ยกตัวอย่างได้ดังนี้

- SPF (Sender Policy Framework) [13] มีหลักการทำงาน โดยเมื่อเครื่องแม่ข่ายอีเมลปลายทางได้รับการติดต่อเข้ามาจากผู้ส่งอีเมล ซึ่งตามโพรโทคอลเอสเอ็มทีพี ผู้ส่งอีเมลจะมีการใช้คำสั่ง HELO และ MAIL FROM และอีกหลายคำสั่ง ข้อมูลที่เครื่องแม่ข่ายเมลปลายทางได้รับนี้ จะทำให้ตรวจสอบได้ว่าอีเมลที่ถูกส่งมา ถูกส่งมาจากต้นทางที่เหมาะสมหรือไม่ โดยการตรวจสอบไปยัง SPF Record บนเครื่องแม่ข่ายโดเมนเนม แล้วตรวจสอบข้อมูลที่ได้รับมาว่าถูกต้องหรือไม่ สังเกตได้ว่าวิธีการ SPF นี้ต้องการความร่วมมือจากโดเมนต่างๆ ให้ระบุค่า SPF Record บนเครื่องแม่ข่ายโดเมนเนม
- การตรวจสอบที่มาของอีเมลโดยตรวจสอบจากรายชื่อผู้ส่งอีเมลที่ดี (Whitelist) ซึ่งเป็นรายชื่อผู้ส่งอีเมลที่ผู้รับอีเมลมีความไว้วางใจ เช่น ผู้ส่งอีเมลที่ผู้รับติดต่อสื่อสารด้วยกันเป็นประจำเป็นต้น อีเมลที่มีชื่อผู้ส่งปรากฏอยู่ในรายชื่อผู้ส่งอีเมลที่ดีนี้จะถูกนำส่งไปยังกล่องจดหมาย (ไม่ถูกกรองออก)
- การตรวจสอบที่มาของอีเมลโดยตรวจสอบจากรายชื่อผู้ส่งอีเมลสแปม (Blacklist) จะทำให้อีเมลที่มีชื่อผู้ส่งปรากฏอยู่ในรายชื่อผู้ส่งอีเมลสแปมนี้จะถูกกรองออก โดยทั่วไปข้อมูลรายชื่อผู้ส่งอีเมลสแปมนี้จะมาจากฐานข้อมูลของที่อยู่ไอพี (IP Address) ที่เป็นแหล่งที่มาของการส่งอีเมลสแปม รวมถึงไอพีที่เป็นรีเลย์ในการส่งอีเมลสแปมอีกด้วย ฐานข้อมูลประเภทนี้มีการดูแลให้มีความทันสมัยอยู่ตลอดเวลาและมีชื่อว่า Domain Name System Blackhole List (DNSBL) โดย DNSBL มีอยู่หลายแบบ เช่น Real-Time Blackhole List (RBL), Mail Abuse Prevention System (MAPS) [14], Spamhaus [15]

- การตรวจสอบที่มาของอีเมลโดยตรวจสอบโดยใช้เทคนิค Greylisting หลักการคือเมื่อมีอีเมลใหม่ถูกส่งเข้ามายังเครื่องแม่ข่ายอีเมล จากนั้นเครื่องแม่ข่ายอีเมลจะใช้ข้อมูลบางอย่างของอีเมลใหม่ที่เข้ามานั้นในการตรวจสอบว่าเป็นอีเมลที่ไม่เคยส่งมาก่อนหรือไม่ เช่น ไอพีของผู้ส่ง เป็นต้น หากเครื่องแม่ข่ายอีเมลพบว่าอีเมลที่เข้ามาใหม่นั้นถูกส่งมาเป็นครั้งแรก เครื่องแม่ข่ายอีเมลจะทำการปฏิเสธการรับอีเมลใหม่นี้ชั่วคราว โดยถ้าอีเมลใหม่นี้ถูกส่งมาจากเครื่องแม่ข่ายที่ดี (Legitimate Mail Server) จะพยายามทำการส่งอีเมลมาใหม่อีกรอบ และจะทำให้เครื่องแม่ข่ายอีเมลของผู้รับยอมรับอีเมลใหม่ที่ส่งมาในครั้งที่สอง แต่ถ้าอีเมลใหม่นี้ถูกส่งมาจากสแปมเมอร์ซึ่งจะทำการส่งอีเมลจำนวนมากต่อครั้งโดยใช้โปรแกรมช่วยส่งอีเมลสแปมซึ่งโดยปกติแล้วจะไม่ส่งอีเมลมาใหม่ และจะทำให้เครื่องแม่ข่ายอีเมลของผู้รับไม่ได้รับอีเมลสแปม

ตัวอย่างวิธีการวิเคราะห์ที่มาของอีเมลแบบอื่นๆ เช่น Caller-ID [16], DKIM [17]

ข้อเสียของวิธีการนี้ คือ จำเป็นต้องมีการแก้ไขโพรโทคอลเอสเอ็มทีพีบ้างเล็กน้อย

2.1.7.2.2 **วิธีการวิเคราะห์ลักษณะเฉพาะของอีเมล** (Fingerprints)

วิธีการวิเคราะห์ลักษณะเฉพาะของอีเมล มีหลักการทำงานโดยเครื่องแม่ข่ายอีเมลทำการคำนวณค่า Checksum ของอีเมลที่ได้รับเข้ามา และส่งค่า Checksum ที่คำนวณได้ไปยังเครื่องแม่ข่ายศูนย์กลาง ตัวอย่างเช่น เครื่องแม่ข่าย DCC ซึ่งมีหน้าที่เก็บค่า Checksum ที่ได้รับรายงานเข้ามาลงในฐานข้อมูล

เมื่อเครื่องแม่ข่ายอีเมลต้องการตรวจสอบว่าอีเมลที่เข้ามาใหม่เป็นอีเมลสแปมหรือไม่ เครื่องแม่ข่ายอีเมลจะทำการส่งค่า Checksum ของอีเมลที่เข้ามาใหม่นั้นให้กับเครื่องแม่ข่าย DCC เพื่อเปรียบเทียบค่า Checksum ที่อยู่บนฐานข้อมูล แล้วตอบกลับข้อมูลบางอย่างไปยังเครื่องแม่ข่ายอีเมล เช่น จำนวนครั้งที่อีเมลถูกรายงานเข้ามา เป็นต้น

เมื่อเครื่องแม่ข่ายอีเมลได้รับข้อมูลจากเครื่องแม่ข่าย DCC แล้ว เครื่องแม่ข่ายอีเมลจะตัดสินว่าอีเมลฉบับนั้นเป็นอีเมลสแปมหรือไม่จะขึ้นอยู่กับนโยบายของผู้ดูแลระบบที่กำหนดไว้

ตัวอย่างงานวิจัยของวิธีการนี้ คือ DCC [18], Vipul's Razor [19], Pyzor [20]

ข้อเสียของวิธีการนี้ คือ ต้องใช้ระยะเวลาในการรวบรวมการรายงานค่า Checksum ของอีเมลต่างๆ จากเครื่องแม่ข่ายอีเมลจึงจะสามารถป้องกันอีเมลสแปมได้ ส่งผลทำให้ระบบเกิดความเสียหายขึ้นแล้วจึงจะสามารถป้องกันอีเมลสแปมได้

2.1.7.2.3 **วิธีการวิเคราะห์เนื้อหาของอีเมล** (Content of Email)

วิธีการวิเคราะห์เนื้อหาของอีเมลเป็นวิธีการวิเคราะห์ข้อความของอีเมลเพื่อค้นหาลักษณะบางอย่างที่สำคัญ ซึ่งวิธีการรูปแบบนี้สามารถแบ่งออกเป็นรูปแบบย่อยได้อีก 2 วิธี ดังนี้

- วิธีการวิเคราะห์คำหรือวลีที่สำคัญหรือการสร้างกฎ (Rule Based)

วิธีการวิเคราะห์คำหรือวลีที่สำคัญหรือวิธีการสร้างกฎ เป็นการวิเคราะห์ข้อความของอีเมล ทั้งในส่วน Body และในส่วน Subject โดยค้นหาคำสำคัญที่บ่งบอกว่าเป็นอีเมลสแปม ยกตัวอย่างเช่น Viagra เป็นต้น

ข้อดีของวิธีนี้ คือ กฎที่สร้างขึ้นสามารถใช้ร่วมกันระหว่างเครื่องแม่ข่ายอีเมลได้ แต่เนื่องจากผู้ส่งอีเมลสแปมสามารถเปลี่ยนแปลงรูปแบบของอีเมล เพื่อหลบเลี่ยงกฎต่างๆ อยู่ตลอดเวลา ทำให้อีเมลสแปมถูกส่งเข้ามายังกล่องจดหมายของผู้รับ วิธีการนี้อาจจะไม่ได้ผลลัพธ์ที่ดีมาก หากผู้ดูแลระบบไม่สร้างกฎให้มีความทันสมัยอยู่ตลอดเวลา

ตัวอย่างโปรแกรมของวิธีการนี้ คือ โปรแกรมกรองอีเมลสแปมแอสแซสซิน (SpamAssassin) [21], โปรแกรมกรองอีเมลดีสแปม (DSPAM) [22]

- วิธีการใช้ระบบการเรียนรู้ (Learning)

วิธีการใช้ระบบการเรียนรู้เป็นวิธีการทางสถิติที่อาศัยเทคนิคการจัดจำแนกของข้อมูล ซึ่งเป็นกระบวนการแบ่งข้อมูลออกเป็นหมวดหมู่และจัดจำแนกประเภทให้กับข้อมูลที่ไม่ทราบประเภท

ตัวอย่างของวิธีการนี้ คือ โปรแกรมกรองอีเมลสแปมแอสแซสซิน (SpamAssassin), โปรแกรมกรองอีเมลดีสแปม (DSPAM), โปรแกรมกรองอีเมลสแปมเบย์ (SpamBayes) [23], โปรแกรมกรองอีเมลสแปมโพลบ (SpamProbe) [24]

วิธีการนี้จะนำเทคนิคการจัดจำแนกประเภทของข้อมูลมาใช้ในการจำแนกประเภทของอีเมลว่าเป็นอีเมลที่ดีหรือเป็นอีเมลสแปม (Spam Filter) หลักการของ

ระบบการเรียนรู้คือ การใช้ข้อมูลฝึก (ซึ่งมีทั้งตัวอย่างบวก และตัวอย่างลบ) สอนระบบให้สามารถรู้จำได้ จากนั้นจึงให้ระบบจำแนกข้อมูลที่ไม่ทราบประเภทได้ โดยมุ่งเน้นที่ระบบการเรียนรู้แบบเบย์

ระบบการเรียนรู้แบบเบย์จะเก็บรวบรวมความถี่ของคำแต่ละคำที่ปรากฏบนทั้งอีเมลที่ดี และอีเมลสแปม จากนั้นจะบันทึกลงบนฐานข้อมูล ซึ่งเรียกว่าความรู้ (Knowledge)

เครื่องแม่ข่ายอีเมลฝั่งผู้รับสามารถกรองอีเมลที่เข้ามาได้ วิธีการคือระบบกรองอีเมลสแปมจะคำนวณความน่าจะเป็นที่จะเป็นอีเมลสแปมของอีเมลฉบับนั้น โดยวิเคราะห์คำที่ปรากฏบนอีเมลฉบับนั้นๆ เทียบกับฐานข้อมูล

วิธีการนี้สามารถรับมือกับอีเมลสแปมที่มีรูปแบบที่หลากหลายได้ โดยทำการเรียนรู้รูปแบบใหม่ๆ เข้าไปเพิ่ม (Retrain) และตัวอย่างโปรแกรมของวิธีการนี้ คือ โปรแกรมกรองอีเมลสแปมแอสแซสซิน, โปรแกรมกรองอีเมลดีสแปม

แต่จากการวิเคราะห์พบว่าภาษาไทยมีลักษณะที่ยากต่อการประมวลผลด้วยคอมพิวเตอร์ เนื่องจากไม่มีการเว้นวรรคแบ่งคำ ไม่มีเครื่องหมายบอกการสิ้นสุดของประโยค เป็นต้น ทำให้ระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์ทั่วไปนั้น จะตัดคำภาษาไทยออกมาเป็นคำที่อ่านไม่รู้เรื่อง ทำให้ได้ระบบกรองอีเมลที่ไร้ประสิทธิภาพ ส่วนหนึ่งของวิทยานิพนธ์นี้จึงจัดทำขึ้นเพื่อนำเสนอระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีระบบการเรียนรู้แบบเบย์ และใช้ระบบตัดคำไทย

แต่ข้อเสียที่สำคัญคือเมื่อระบบการเรียนรู้ได้เรียนรู้ข้อมูลขนาดใหญ่ จะทำให้ฐานข้อมูลมีขนาดใหญ่มากทำให้สิ้นเปลืองเนื้อที่ และการนำฐานข้อมูลไปใช้ร่วมกันเป็นไปได้ยาก

ในวิทยานิพนธ์นี้นำเสนอวิธีการแก้ปัญหาอีเมลสแปมสำหรับภาษาไทยที่สามารถสร้างกฎด้วยวิธีการทางสถิติ (Statistical Rule Based) ซึ่งเป็นการรวมข้อดีของวิธีการสร้างกฎและวิธีการทางสถิติที่ใช้ในระบบการเรียนรู้ กฎที่ได้จะสามารถกรองอีเมลสแปมภาษาไทย และสามารถดูแลให้มีความทันสมัยได้ง่าย

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 โปรแกรมกรองอีเมลสแปมแอสแซสซิน (SpamAssassin) [25]

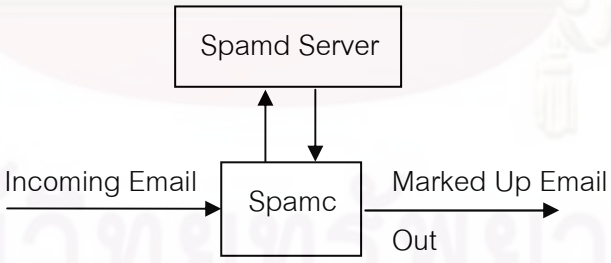
โปรแกรมกรองอีเมลสแปมแอสแซสซิน ถูกพัฒนาด้วยภาษาเพิร์ล (Perl) โดยสถาบันพัฒนาซอฟต์แวร์อะพาเช่ (Apache Software Foundation) เป็นโปรแกรมกรองอีเมลที่เป็นแบบซอฟต์แวร์เสรี (Open source Software) และเป็นแบบฮิวริสติก (Heuristic Based) ซึ่งภายใน

โปรแกรมจะมีการทดสอบหลายวิธีเพื่อตรวจสอบอีเมลที่เข้ามายังเครื่องแม่ข่ายอีเมล ผลลัพธ์ของแต่ละการทดสอบได้ผลออกมาเป็นคะแนนความเป็นอีเมลสแปม และคะแนนของแต่ละการทดสอบจะถูกรวมกันเพื่อพิจารณาว่าอีเมลที่เข้ามานั้นเป็นอีเมลที่ดีหรือเป็นอีเมลสแปม โดยจะมีเกณฑ์ของคะแนนความเป็นอีเมลสแปม ซึ่งจะเป็นตัวกำหนดว่าอีเมลที่เข้ามานั้นจะเป็นอีเมลสแปม เมื่อได้รับผลรวมของคะแนนมากกว่าเกณฑ์ของคะแนนความเป็นอีเมลสแปม (ค่ามาตรฐานเกณฑ์ของคะแนน คือ 5 คะแนน)

การทดสอบที่มีอยู่ในโปรแกรมกรองอีเมลสแปมแอสแซสซิน ตัวอย่างเช่น การตรวจสอบการยืนยันตัวตนของผู้ส่ง (Authentication of Sender) การวิเคราะห์คำหรือวลีที่สำคัญ (Rule Based) การกรองอีเมลสแปมโดยใช้ระบบการเรียนรู้แบบเบย์ (Bayesian Learning) เป็นต้น

โปรแกรมกรองอีเมลสแปมแอสแซสซินสามารถจัดการกับปัญหาการใช้เวลาการประมวลผลอีเมลจำนวนมากได้ โดยโปรแกรมจะทำการสั่งให้ส่วนสแปมแอสแซสซินดีมอน (SpamAssassin Daemon: Spamd) ทำงานในตอนเริ่มต้นของระบบ และส่วนสแปมแอสแซสซินดีมอนสามารถสั่งให้สแปมแอสแซสซินเพิร์ลมอดูล (SpamAssassin Perl Module) ทำการตรวจสอบอีเมลที่เข้ามาบนเครื่องแม่ข่ายอีเมลได้

อีเมลแต่ละฉบับจะถูกส่งไปยังส่วนสแปมแอสแซสซินลูกข่าย (SpamAssassin Client: Spamc) และจะส่งต่ออีเมลต่อไปให้กับส่วนสแปมแอสแซสซินดีมอน เมื่ออีเมลฉบับนั้นๆ ถูกประมวลและได้รับคะแนนความเป็นสแปมแล้ว จะส่งผลลัพธ์มาให้กับส่วนสแปมแอสแซสซินลูกข่ายต่อไป ดังแสดงในรูปที่ 7



รูปที่ 7 ส่วนประกอบของโปรแกรมกรองอีเมล (SpamAssassin Component)

วิธีการนี้จะมีประสิทธิภาพมากกว่าการใช้คำสั่งโปรแกรมสแปมแอสแซสซิน (SpamAssassin Script) ในการตรวจสอบอีเมลแต่ละฉบับ

2.2.2 The SpamAssassin Rule Emporium (SARE) [26]

วิธีการสร้างกฎเป็นวิธีการแก้ปัญหาอีเมลสแปมวิธีการหนึ่งที่มีประสิทธิภาพโดยจะค้นหาคำหรือวลีที่สำคัญซึ่งสามารถบ่งบอกความเป็นอีเมลสแปมได้ทั้งในส่วนหัวข้อเรื่อง (Subject) และส่วนเนื้อหา (Body) กฎแต่ละกฎสามารถเพิ่มหรือลดคะแนนความเป็นอีเมลสแปมของอีเมลได้ ตัวอย่างกฎในโปรแกรมสแปมแอสแซสซินถูกแสดงในรูปที่ 8

```
body        Check_viagra  /^V*i*a*g*r*a*/
describe    Check_viagra  Checking Viagra pattern.
score       3.5
```

รูปที่ 8 กฎในโปรแกรมกรองอีเมลสแปมแอสแซสซิน

จากตัวอย่างกฎในรูปที่ 8 ถ้าอีเมลที่เข้ามาบนเครื่องแม่ข่ายอีเมลมีคำว่า 'V_i_a_g_r_a' ปรากฏในส่วนเนื้อหาจะทำให้สอดคล้องกับ Regular Expression ของตัวอย่างกฎ และอีเมลฉบับนี้จะถูกเพิ่มคะแนนความเป็นอีเมลสแปมขึ้นอีก 3.5 คะแนน

โปรแกรมกรองอีเมลสแปมแอสแซสซินนั้นจะมีกฎมาตรฐานติดตั้งมาด้วยจำนวนหนึ่งและสามารถที่จะสร้างกฎขึ้นมาเองได้ (Custom Rules)

SARE เป็นเว็บไซต์ที่รวบรวมกฎที่สร้างขึ้นมาเอง ซึ่งได้รวบรวมกฎหลายประเภท ยกตัวอย่างเช่น กฎที่ถูกเขียนขึ้นมาสำหรับป้องกันอีเมลสแปมรูปแบบใหม่ๆ ของผู้ส่งอีเมลสแปม กฎสำหรับป้องกันอีเมลที่ไม่เหมาะสม เป็นต้น ซึ่งจะมีทั้งกฎสำหรับส่วนเนื้อหาและส่วนหัวข้อด้วย

กฎต่างๆ ในเว็บไซต์ SARE นี้ดูแลโดย Christ Santerre ซึ่งเขาเป็นผู้เขียนกฎแต่ละกฎด้วยตัวเอง และได้ประกาศไว้บนเว็บไซต์ เพื่อให้บุคคลทั่วไปสามารถนำกฎไปใช้ได้ นอกจากนี้ผู้ดูแลเครื่องแม่ข่ายอีเมลยังสามารถใช้คำสั่งในการ download กฎของ SARE ได้อัตโนมัติเมื่อกฎมีการเปลี่ยนแปลง

กฎที่เขาสร้างขึ้น สามารถทำให้อีเมลสแปมได้รับคะแนนเพิ่มขึ้นจาก 3 คะแนนเป็น 35 คะแนนได้แต่กฎในเว็บไซต์ SARE ไม่มีกฎสำหรับป้องกันอีเมลสแปมภาษาไทยและขณะนี้เว็บไซต์ SARE ไม่มีการพัฒนาสร้างกฎเพิ่มเติม

2.2.3 Real-Time Statistical Rules for Spam Detection [27]

เป็นโครงงานวิจัยที่ถูกพัฒนาในปี 2006 เป็นการนำเสนอวิธีการสร้างกฎสำหรับระบบกรองอีเมลสแปมภาษาจีนด้วยวิธีการทางสถิติโดยเป็นการนำเอาข้อดีของวิธีการสร้างกฎและวิธีการใช้ระบบการเรียนรู้ซึ่งเป็นวิธีการทางสถิติรวมเข้าด้วยกัน มีทั้งกฎสำหรับส่วนเนื้อหาและกฎสำหรับส่วนหัวข้อในงานวิจัยนี้ได้เสนอแนะว่าสามารถนำวิธีการไปประยุกต์ใช้กับการป้องกันอีเมลสแปมภาษาอื่นๆ ได้จึงได้นำมาประยุกต์และใช้เป็นข้อมูลอ้างอิงในการพัฒนาการสร้างกฎสำหรับป้องกันอีเมลสแปมภาษาไทยต่อไป

2.2.4 Vietnamese Spam Detection based on Language Classification [28]

เป็นโครงงานวิจัยที่ถูกพัฒนาในปี 2008 เป็นการนำเสนอระบบกรองอีเมลสแปมสำหรับภาษาเวียดนามที่เป็นระบบการเรียนรู้แบบเบย์และมีการใช้ระบบตัดคำภาษาเวียดนามซึ่งได้นำเสนอลักษณะเฉพาะของภาษาเวียดนามที่มีความแตกต่างจากภาษาอังกฤษทำให้ระบบกรองอีเมลที่มีระบบการเรียนรู้แบบเบย์ทั่วไปนั้นกรองอีเมลสแปมภาษาเวียดนามได้ไม่มีประสิทธิภาพจึงจำเป็นต้องใช้ระบบตัดคำภาษาเวียดนาม เพื่อเพิ่มประสิทธิภาพให้กับระบบกรองอีเมลสแปมสำหรับภาษาเวียดนามที่มีระบบการเรียนรู้แบบเบย์

อย่างไรก็ตามผู้วิจัยของโครงงานวิจัยนี้ชี้ให้เห็นว่าการใช้วิธีการทางสถิติหรือระบบการเรียนรู้นั้นสามารถรับมือกับรูปแบบของอีเมลสแปมที่หลากหลายรูปแบบได้โดยการเรียนรู้รูปแบบใหม่ๆ เข้าไปเพิ่มแต่ไม่สามารถนำความรู้ที่ได้มาใช้ร่วมกันระหว่างเครื่องแม่ข่ายอีเมลได้ง่าย งานวิจัยในอนาคตของโครงงานวิจัยนี้จะทำการสร้างกฎภาษาเวียดนามให้กับระบบกรองอีเมลสแปมแอสแซสซินด้วยวิธีการทางสถิติ เพื่อเพิ่มประสิทธิภาพให้กับระบบกรองอีเมลสแปมสำหรับภาษาเวียดนาม

บทที่ 3
หลักการสร้างและออกแบบระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีระบบการเรียนรู้แบบเบย์ และใช้โปรแกรมตัดคำไทย

ระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์ทั่วไปนั้น เป็นวิธีการแก้ไขปัญหาอีเมลสแปมที่มีประสิทธิภาพเมื่อใช้วิเคราะห์คำในภาษาอังกฤษ แต่เมื่อนำมาวิเคราะห์คำภาษาไทยพบว่า ไม่มีประสิทธิภาพเท่าที่ควร เนื่องจากระบบกรองอีเมลสแปมแบบเบย์ทั่วไปไม่สามารถวิเคราะห์คำภาษาไทยได้ถูกต้อง จึงจำเป็นต้องใช้โปรแกรมตัดคำไทยช่วยในการวิเคราะห์คำภาษาไทย ในบทนี้จะกล่าวถึงพื้นฐานการเรียนรู้แบบเบย์ ปัญหาลักษณะสำคัญของภาษาไทย โปรแกรมตัดคำไทยคูวส์ และการออกแบบระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีระบบการเรียนรู้แบบเบย์ และใช้โปรแกรมตัดคำไทย

3.1 การเรียนรู้แบบเบย์

การจัดจำแนกประเภทของอีเมลโดยวิเคราะห์เนื้อหาของอีเมลถูกนำเสนอโดย Paul Graham [29] ซึ่งเป็นวิธีการทางสถิติที่มีงานวิจัยศึกษามาก่อนหน้าแล้วนั้น [30], [31] ซึ่งวิธีการของ Graham ง่าย มีประสิทธิภาพ และเป็นที่นิยมใช้

ระบบกรองอีเมลที่ใช้วิธีการทางสถิติของ Graham นั้นใช้ทฤษฎีของเบย์ซึ่งเป็นวิธีการทางคณิตศาสตร์เพื่อคำนวณความน่าจะเป็นของเหตุการณ์หนึ่งโดยที่กำหนดว่ามีเหตุการณ์อื่นที่เกิดขึ้นอย่างอิสระต่อกันซึ่งทฤษฎีของเบย์แสดงได้ดังสมการต่อไปนี้

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)} \quad (1)$$

หลักการการเรียนรู้ของเครื่อง คือ ผู้ดูแลระบบจะต้องใช้ข้อมูลฝึกซึ่งเป็นอีเมลที่ดีและเป็นอีเมลสแปมที่ถูกส่งเข้ามายังเครื่องแม่ข่ายเพื่อสอนเครื่องให้สามารถจดจำ วิเคราะห์ และสร้างฐานข้อมูลที่เก็บจำนวนครั้งของคำแต่ละคำที่ปรากฏทั้งบนอีเมลที่ดีและอีเมลสแปม

ตัวอย่างเช่น ถ้าหากระบบกรองอีเมลได้รับข้อมูลฝึกที่เป็นอีเมลสแปมซึ่งมีคำว่า ‘Viagra’ จำนวนมากปรากฏอยู่ ระบบจะสามารถจำแนกอีเมลใหม่ซึ่งมีคำว่า ‘Viagra’ ปรากฏอยู่เป็นอีเมลสแปม ผลที่ได้คือระบบกรองอีเมลที่สามารถเรียนรู้ได้เองอย่างอัตโนมัติ

3.1.1 การคำนวณความน่าจะเป็นที่จะเป็นอีเมลสแปมให้กับอีเมลที่เข้ามาใหม่

หลังจากทำการสอนระบบกรองอีเมลด้วยข้อมูลฝึกแล้ว เมื่ออีเมลใหม่เข้ามาบนเครื่องแม่ข่ายจะถูกวิเคราะห์เนื้อหาของอีเมลโดยแบ่งเนื้อหาออกเป็นคำและค้นหาคำสำคัญซึ่งเป็นคำที่

สามารถบ่งบอกได้ว่าอีเมลเป็นอีเมลที่ดีหรือเป็นอีเมลสแปมโดยคำนวณจากกฎของเบย์ (1) จะได้ว่า

$$P(Spam|Token)=\frac{P(Token|Spam)*P(Spam)}{P(Token)} \quad (2)$$

กำหนดให้ คือ P(Spam|Token) ความน่าจะเป็นแบบมีเงื่อนไขที่จะเป็นอีเมลสแปม เมื่อมีคำ (Token) ปรากฏอยู่

P(Token|Spam) คือ ความน่าจะเป็นแบบมีเงื่อนไขของอีเมลสแปมหนึ่งฉบับที่จะมีคำ (token) ปรากฏอยู่

P(Spam) คือ ความน่าจะเป็นก่อนหน้าที่อีเมลหนึ่งฉบับจะเป็นอีเมลสแปม

P(Token) คือ ความน่าจะเป็นก่อนหน้าที่อีเมลใดใดจะพบคำ (Token)

จากข้อมูลตัวอย่างในตารางที่ 3 ความน่าจะเป็นก่อนหน้าของอีเมลหนึ่งฉบับจะเป็นอีเมลสแปมจะได้ว่า

P(Spam) = 0.6

ความน่าจะเป็นก่อนหน้าของอีเมลใดใดจะพบคำ (Token) จะได้ว่า

P(ขาย) = 0.7

P(ถูก) = 0.1

ตารางที่ 3 ข้อมูลตัวอย่าง

อีเมล	จำนวนอีเมลที่ดี	จำนวนอีเมลสแปม	ผลรวม
อีเมลทั้งหมด	400	600	1000
อีเมลที่มีคำว่า 'ขาย'	200	500	700
อีเมลที่มีคำว่า 'ถูก'	10	90	100

ความน่าจะเป็นแบบมีเงื่อนไขของอีเมลสแปมหนึ่งฉบับที่จะมีคำ (Token) ปรากฏอยู่จะได้ว่า

$$P(ขาย|Spam) = \frac{500}{600} = 0.83$$

$$P(ถูก|Spam) = \frac{90}{600} = 0.15$$

ดังนั้นสามารถคำนวณความน่าจะเป็นที่จะเป็นอีเมลสแปมตามกฎของเบย์ได้ ดังนี้

$$P(Spam|ขาย) = \frac{0.83*0.6}{0.7} = 0.71$$

$$P(Spam|ถูก) = \frac{0.15*0.6}{0.1} = 0.9$$

พบว่าความน่าจะเป็นที่จะเป็นอีเมลสแปมของอีเมลที่มีคำว่า “ขาย” ปรากฏอยู่มีค่า 0.71 และความน่าจะเป็นที่จะเป็นอีเมลสแปมของอีเมลที่มีคำว่า “ถูก” ปรากฏอยู่มีค่า 0.9

เมื่อได้ความน่าจะเป็นที่จะเป็นอีเมลสแปมของอีเมลของแต่ละคำที่ปรากฏบนอีเมลแล้ว เราสามารถรวมความน่าจะเป็นให้กับอีเมลฉบับนั้น ได้จากความน่าจะเป็นของคำแต่ละคำโดยจะใช้ทฤษฎีนาอีฟเบย์ (Naive Bayesian) ได้ว่า

$$P(\text{Spam}|\text{ขาย},\text{ถูก}) = \frac{P(\text{ขาย},\text{ถูก}|\text{Spam})*P(\text{Spam})}{P(\text{ขาย},\text{ถูก})}$$

$$P(\text{Ham}|\text{ขาย},\text{ถูก}) = \frac{P(\text{ขาย},\text{ถูก}|\text{Ham})*P(\text{Ham})}{P(\text{ขาย},\text{ถูก})}$$

จะได้ว่า

$$1 = \frac{P(\text{ขาย},\text{ถูก}|\text{Spam})*P(\text{Spam})}{P(\text{ขาย},\text{ถูก})} + \frac{P(\text{ขาย},\text{ถูก}|\text{Ham})*P(\text{Ham})}{P(\text{ขาย},\text{ถูก})}$$

จากสมมติฐาน ที่ว่าคำแต่ละคำที่ปรากฏบนอีเมล เป็นอิสระต่อกันอย่างมีเงื่อนไข จะได้ว่า

$$X = P(\text{ขาย},\text{ถูก}|\text{Spam})*P(\text{Spam})$$
$$= P(\text{ขาย}|\text{Spam})*P(\text{ถูก}|\text{Spam})*P(\text{Spam})$$
$$Y = P(\text{ขาย},\text{ถูก}|\text{Ham})*P(\text{Ham}) = P(\text{ขาย}|\text{Ham})*P(\text{ถูก}|\text{Ham})*P(\text{Ham})$$

ดังนั้น

$$P(\text{Spam}|\text{ขาย},\text{ถูก}) = \frac{X}{X+Y}$$

$$P(\text{Ham}|\text{ขาย},\text{ถูก}) = \frac{Y}{X+Y}$$

เมื่อแทนค่าจากฐานข้อมูลจะได้ว่า

$$X = 0.83*0.15*0.6$$
$$= 0.075$$
$$Y = 0.5*0.025*0.4$$
$$= 0.005$$

ดังนั้น

$$P(\text{Spam}|\text{ขาย},\text{ถูก}) = \frac{0.075}{(0.075+0.005)}$$

$$= 0.9375$$

คำแต่ละคำจะถูกคำนวณค่าความน่าจะเป็นออกมาและจะถูกรวมคะแนนเพื่อคำนวณคะแนนให้กับอีเมลฉบับนั้นๆ ซึ่งในวิธีการของ Paul Graham จะรวมค่าความน่าจะเป็นของคำที่มีความสำคัญมากที่สุด 15 อันดับแรกเท่านั้น คำที่ปรากฏอยู่ทั้งอีเมลที่ดีและปรากฏอยู่บนอีเมลสแปมนั้น จะไม่นำมารวมค่าความน่าจะเป็น เพราะไม่มีความสำคัญต่อการจัดจำแนก

ความสามารถที่สำคัญของระบบการเรียนรู้แบบเบย์ สามารถทำให้เรียนรู้อีเมลสแปมใหม่ได้โดยไม่ต้องการให้ผู้ดูแลระบบเป็นผู้จัดการ เมื่อพบว่าอีเมลที่ถูกระบบกรองอีเมลจัดจำแนกประเภทผิด เนื่องมาจากผู้ส่งอีเมลสแปมทำการเปลี่ยนแปลงรูปแบบคำของอีเมลสแปมก็สามารถเรียนรู้คำใหม่เข้าไปได้

3.2 การตัดคำภาษาไทย

ระบบกรองอีเมลถูกพัฒนาขึ้นมากมายด้วยวิธีการเรียนรู้แบบเบย์มีประสิทธิภาพสูงเมื่อวิเคราะห์คำภาษาอังกฤษ แต่เมื่อวิเคราะห์คำไทย ประสิทธิภาพที่ได้กลับด้อยลง เพราะภาษาไทยเป็นภาษาที่ไม่มีการเว้นวรรคคำ การตัดคำที่ใช้สำหรับตัดคำในภาษาอื่นๆ เช่น ภาษาอังกฤษ เป็นต้น ไม่เหมาะสมที่จะนำมาตัดไทย การหาขอบเขตคำหรือการตัดคำเป็นสิ่งสำคัญ จึงนำการตัดคำไทยคูวส์มาใช้ เพื่อให้ได้ผลการวิเคราะห์คำได้อย่างถูกต้องแม่นยำ

ภาษาไทยมีลักษณะเฉพาะที่แตกต่างกับภาษาทั่วไป ดังแสดงในตารางที่ 4

ตารางที่ 4 ลักษณะเฉพาะของภาษาไทย

ลักษณะเฉพาะ	ภาษาไทย	ภาษาอังกฤษ
ขอบเขตสิ้นสุดของคำ	ไม่มี	มีการเว้นวรรคแบ่งคำ
การขึ้นต้นของประโยค	ไม่มี	ตัวอักษรตัวพิมพ์ใหญ่
การบอกจุดสิ้นสุดของประโยค	ไม่มี	จุด (.)

ตัวอย่างฐานข้อมูลคำของระบบกรองอีเมลสที่มีระบบการเรียนรู้แบบเบย์ทั่วไป ซึ่งแสดงในรูปที่ 9 จะเห็นได้ว่าระบบกรองอีเมลที่มีระบบการเรียนรู้แบบเบย์ทั่วไปนั้นทำการตัดคำออกมาเพียงคำละ 2-3 ตัวอักษร

```
Subject: คุณเบื่อวันจันทร์ชอบวันศุกร์มั้ย


[14450] dbg: bayes: DB journal sync: last sync: 1151465861
[14450] dgb: bayes: corpus size: nspam = 72450, nham = 37282
[14450] dbg: bayes: 8:คุ
[14450] dbg: bayes: 8:ณเ
[14450] dbg: bayes: 8:บื
[14450] dbg: bayes: 8:อ
[14450] dbg: bayes: 8:วั
[14450] dbg: bayes: 8:นจ
[14450] dbg: bayes: 8:นั
[14450] dbg: bayes: 8:ทร
[14450] dbg: bayes: 8:ช์
```

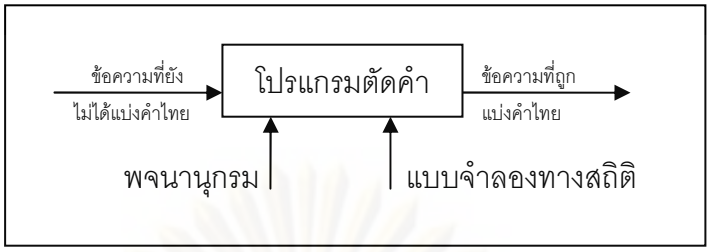
รูปที่ 9 ตัวอย่างฐานข้อมูลคำของระบบกรองอีเมลที่มีระบบการเรียนรู้แบบเบย์ทั่วไป

3.3 โปรแกรมตัดคำไทยคูวส์

โปรแกรมตัดคำไทยคูวส์ [32] เป็นโปรแกรมตัดคำไทยที่ได้รับรางวัลในการแข่งขันการพัฒนาโปรแกรมแบ่งคำภาษาไทย (BEST2009) จุดมุ่งหมายของการพัฒนาโปรแกรมตัดคำไทยคูวส์เพื่อนำเสนอขั้นตอนวิธี (Algorithm) ที่สามารถแบ่งคำไทยได้ถูกต้องมากที่สุดโดยใช้เวลาประมวลผลอย่างเหมาะสม

เทคนิคที่ใช้ในการพัฒนาคือเทคนิคการแบ่งคำที่มีอยู่แล้ว ยกตัวอย่างเช่น การตัดคำแบบยาวที่สุด การตัดคำแบบสอดคล้องมากที่สุด แบบจำลองเอ็นแกรม (N-Grams) แบบจำลองฮิดเด็นมาร์คอฟ (Hidden Markov) เป็นต้น

จากรูปที่ 10 ข้อมูลนำเข้าคือข้อความที่ยังไม่ได้ถูกแบ่งคำระหว่างประมวลผลโปรแกรมแบ่งคำจะอาศัยพจนานุกรมและแบบจำลองทางสถิติและจะส่งข้อความที่ถูกตัดคำแล้วเป็นข้อมูลออก



รูปที่ 10 โครงสร้างของโปรแกรมตัดคำไทยคูวส์

ประสิทธิภาพของคูวส์ เมื่อทดสอบด้วยวิธีการทดสอบไขว้ k กลุ่ม (k-Fold Cross Validation) ได้ความถูกต้องเฉลี่ยเกินกว่า 99%

ผู้พัฒนาโปรแกรมตัดคำไทยคูวส์ได้พัฒนาโปรแกรมขึ้นด้วยภาษาจาวา (JAVA) และได้พัฒนาไลบรารีให้ผู้ใช้งานทั่วไปสามารถใช้โปรแกรมตัดคำไทยคูวส์ได้ผ่านทางเว็บเซอร์วิส (Web Service)

ไลบราลีที่สำคัญในการใช้งานโปรแกรมตัดคำไทยคูวส์ผ่านทางเว็บเซอร์วิส คือ commons-httpclient-3.1.jar, commons-logging-1.1.1.jar, commons-codec-1.3.jar, log4j-1.2.15.jar ซึ่งสามารถ download ได้ในเว็บไซต์ http://oracle.cp.eng.chula.ac.th/me/cuws ตัวอย่างการใช้งานโปรแกรมตัดคำไทยคูวส์ผ่านทางเว็บเซอร์วิสบนภาษาจาวาดังข้างล่างนี้

```
import com.cuws.api.CUWSWebService;
static void Main(string[] args)
{       string key = "xxxx"; // Key from registration
        CUWSWebService cuws = new CUWSWebService(key);
        string text = "สวัสดีปีใหม่";
        cuws.CUWS cuws = new cuws.CUWS();
        string x = cuws.cut(key, text);
        Console.WriteLine(x);       }
```

ผลลัพธ์ของโปรแกรมดังข้างล่างนี้

```
สวัสดี | ปี | ใหม่
```

3.4 แบบจำลองตัวแทนข้อความ (Text Representation Model)

แบบจำลองตัวแทนข้อความมีอยู่หลายแบบ โดยทั่วไปคือแบบจำลองตัวแทนข้อความเวกเตอร์สเปซ (Vector Space Model) ซึ่งมีรูปแบบลักษณะดังนี้

กำหนดให้ในฐานข้อมูลมีเอกสาร (D) จำนวน m เอกสาร

$D = \{d_1, d_2, \ldots d_m\}$

ในแต่ละเอกสาร แสดงด้วยเวกเตอร์ T

$T = \{t_1, t_2, \ldots t_n\}$ เมื่อ n คือ จำนวนของคำที่ปรากฏในเอกสาร

$W_{ij} = \{w_{ij}\}$ โดย W_{ij} เป็นเมทริกซ์ถ่วงน้ำหนักซึ่งถูกถ่วงน้ำหนักโดยค่า t_i และค่า d_j

ตารางที่ 5 แบบจำลองตัวแทนข้อความเวกเตอร์สเปซ

Terms	Documents				
	d_1	d_2	d_3	…	d_m
t_1	w_{11}	w_{12}	w_{13}	…	w_{1m}
t_2	w_{21}	w_{22}	w_{23}	…	w_{2m}
…	...	…	…	…	…
t_n	w_{n1}	w_{n2}	w_{n3}	…	w_{nm}

แบบจำลองตัวแทนข้อความเวกเตอร์สเปซนั้นจะมีวิธีการคำนวณค่าเมทริกซ์ถ่วงน้ำหนัก (Wij) หลายวิธีการดังนี้

1) แบบจำลอง Term Frequency (TF)

กำหนดให้ f_{ij} คือ ค่าความถี่ของ t_i ที่ปรากฏในเอกสาร d_j

$w_{ij} = f_{ij}$

2) แบบจำลอง Inverse Document Frequency (IDF)

กำหนดให้ h คือ จำนวนเอกสารที่มี t_i ปรากฏอยู่

$w_{ij} = \log(\frac{m}{h})$ เมื่อ t_i ปรากฏอยู่ในเอกสาร d_j

$w_{ij} = 0$ เมื่อ t_i ไม่ปรากฏอยู่ในเอกสาร d_j

3) แบบจำลอง Term Frequency–Inverse Document Frequency (TF-IDF)

$w_{ij} = f_{ij} * \log(\frac{m}{h})$ เมื่อ t_i ปรากฏอยู่ในเอกสาร d_j

$w_{ij} = 0$ เมื่อ t_i ไม่ปรากฏอยู่ในเอกสาร d_j

4) แบบจำลอง Boolean Vector Space

$w_{ij} = 1$ เมื่อ t_i ปรากฏอยู่ในเอกสาร d_j

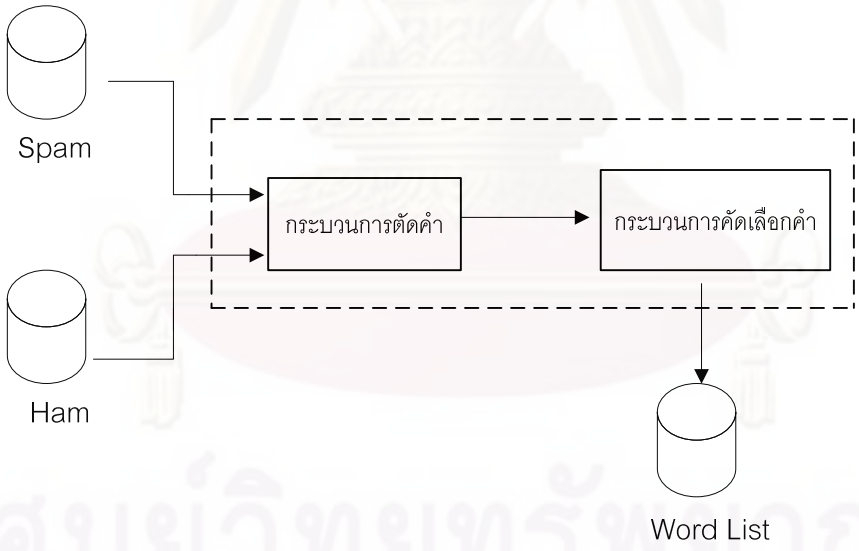
$w_{ij} = 0$ เมื่อ t_i ไม่ปรากฏอยู่ในเอกสาร d_j

ในวิทยานิพนธ์นี้ใช้แบบจำลองตัวแทนข้อความเวกเตอร์สเปซแบบตรรกะ (Boolean Vector Space Model) โดยจะมีค่าของเวกเตอร์มีค่าเป็น 1 เมื่อคำค้นหา n ปรากฏอยู่ในเอกสาร และจะมีค่าเป็น 0 เมื่อไม่พบคำค้นหา n ปรากฏอยู่ในเอกสาร

3.5 การออกแบบระบบ

เครื่องแม่ข่ายอีเมลจะมีการติดตั้งระบบกรองอีเมลซึ่งมีอยู่มากมายหลายประเภทแต่ในวิทยานิพนธ์นี้หมายถึงระบบกรองอีเมลที่มีการเรียนรู้แบบเบย์

ในระบบกรองอีเมลแบบเบย์ทั่วไปนั้นระบบจำเป็นต้องเรียนรู้ตัวอย่างอีเมลที่ดีและอีเมลสแปม ในระบบการเรียนรู้อีเมลทั้งสองประเภทจะเข้าสู่กระบวนการตัดคำเพื่อคัดเลือกคำ คำนวณความน่าจะเป็นของคำแต่ละคำที่ปรากฏบนอีเมลที่ดีและอีเมลสแปมเก็บลงฐานข้อมูล เพื่อเก็บไว้ใช้ในการจำแนกอีเมลต่อไปดังรูปที่ 11

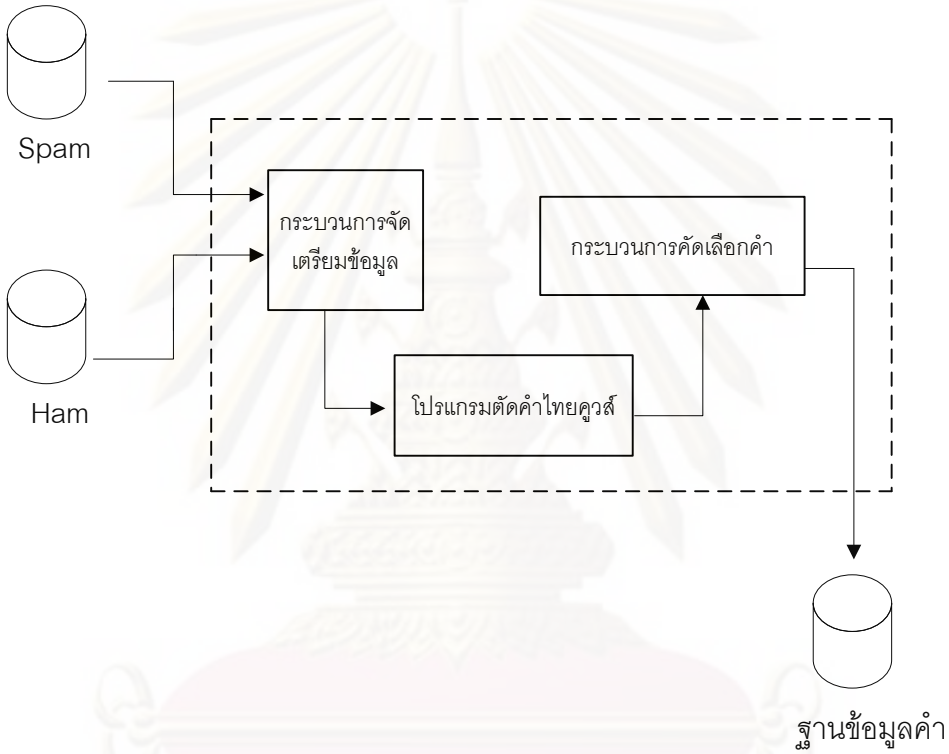


รูปที่ 11 การเรียนรู้ของระบบกรองอีเมลแบบเบย์ทั่วไป

แต่ระบบกรองอีเมลแบบเบย์ที่มีการใช้โปรแกรมตัดคำภาษาไทยคูวส์ นั้นจะมีขั้นตอนการทำงานดังรูปที่ 12

จากรูปที่ 12 เมื่ออีเมลถูกส่งเข้ามายังระบบกรองอีเมลอันดับแรกจะมีกระบวนการจัดเตรียมข้อมูล เช่น การตัดคำสั่งในภาษาเอชทีเอ็มแอล (HTML Tags) ออก เพื่อให้ได้เฉพาะ

ข้อความเท่านั้น การเปลี่ยนแปลงรูปแบบอักขระ (Character Encoding) ให้เหมาะสมกับโปรแกรมตัดคำไทยคูวส์ เป็นต้น หลังจากนั้นข้อความภาษาไทยจะถูกส่งเป็นข้อมูลเข้าให้กับโปรแกรมตัดคำไทยคูวส์ เมื่อได้ข้อมูลออกจากโปรแกรมตัดคำแล้วกระบวนการคัดเลือกคำสำคัญจะทำการเลือกคำที่มีความสำคัญซึ่งหมายถึงคำที่ปรากฏอยู่ในอีเมลแล้วสามารถบ่งบอกได้ว่าอีเมลฉบับนั้นเป็นอีเมลที่ดีหรืออีเมลสแปม หลังจากนั้นคำสำคัญจะถูกเก็บลงฐานข้อมูลในรูปแบบแบบจำลองตัวแทนข้อความเวกเตอร์สเปซแบบตรรกะ



รูปที่ 12 การเรียนรู้ของระบบกรองอีเมลแบบเบย์ที่มีการใช้โปรแกรมตัดคำไทยคูวส์

เมื่อมีอีเมลใหม่เข้ามายังเครื่องแม่ข่ายเมล ระบบกรองอีเมลที่ได้พัฒนาขึ้นจะเรียกโปรแกรมตัดคำไทยคูวส์ทำการแบ่งคำไทยออกมา จากนั้นจะนำคำแต่ละคำไปคำนวณค่าความน่าจะเป็นที่จะเป็นอีเมลสแปมจากฐานข้อมูลคำที่ได้เรียนรู้ไว้และคำนวณคะแนนความเป็นอีเมลสแปมให้กับอีเมลใหม่เครื่องแม่ข่ายเมลจะสามารถคัดกรองอีเมลที่เข้ามาใหม่ว่าเป็นอีเมลสแปมหรือไม่จากคะแนนความเป็นอีเมลสแปมของอีเมล

บทที่ 4
หลักการสร้างกฎด้วยวิธีการทางสถิติ

วิธีการแก้ปัญหาอีเมลสแปมโดยใช้ระบบการเรียนรู้นั้นยังมีข้อเสียที่สำคัญ คือ ความรู้ที่ได้เรียนรู้นั้นมาใช้ร่วมกันระหว่างเครื่องแม่ข่ายอีเมลเป็นไปได้ยาก และเมื่อระบบเรียนรู้ข้อมูลขนาดใหญ่ ฐานข้อมูลจะมีขนาดใหญ่มาก ทำให้สิ้นเปลืองเนื้อที่ในการเก็บ ผู้เสนอวิทยานิพนธ์ได้ทำการออกแบบวิธีการสร้างกฎด้วยวิธีการทางสถิติซึ่งเป็นการนำข้อดีของวิธีการแก้ปัญหาอีเมลสแปมด้วยวิธีการสร้างกฎและวิธีการแก้ปัญหาอีเมลสแปมด้วยวิธีการเรียนรู้เข้าด้วยกัน ในบทนี้จะกล่าวถึงตัวอย่างกฎสำหรับการป้องกันอีเมลสแปมในโปรแกรมสแปมแอสแซสซิน และหลักการสร้างกฎด้วยวิธีการทางสถิติ

4.1 กฎของโปรแกรมกรองอีเมลสแปมแอสแซสซิน

โปรแกรมกรองอีเมลสแปมแอสแซสซินมีวิธีการทดสอบอีเมลว่าเป็นอีเมลสแปมหรือเป็นอีเมลที่ดีอยู่หลายวิธีการทดสอบ การทดสอบด้วยกฎเป็นวิธีการทดสอบที่มีประสิทธิภาพวิธีการทดสอบหนึ่งของโปรแกรมกรองอีเมลสแปมแอสแซสซิน

กฎในโปรแกรมกรองอีเมลสแปมแอสแซสซิน ส่วนใหญ่จะมีส่วนประกอบพื้นฐานที่สำคัญดังนี้

1) ชื่อของกฎ (Rule name)
2) คำสั่ง (Directive) เช่น body, describe, score เป็นต้น
3) คะแนนของกฎ (Score)

```
line1:  body      Check_token     /ลดความอ้วน/
line2:  describe  Check_token     Checking ลดความอ้วน.
line3:  score     3.5
```

รูปที่ 13 ตัวอย่างกฎในโปรแกรมกรองอีเมลสแปมแอสแซสซิน

จากรูปที่ 13 แสดงถึงตัวอย่างกฎในโปรแกรมกรองอีเมลสแปมแอสแซสซินที่มีส่วนประกอบพื้นฐานที่สำคัญครบสมบูรณ์ ดังนี้

บรรทัดที่ 1

body คือ คำสั่งที่บ่งบอกว่ากฎข้อนี้ทำการตรวจสอบเนื้อหาในส่วน Body ของอีเมล

Check_token คือ ชื่อของกฎ

/ลดความอ้วน/ คือ Regular Expression ที่ใช้เป็นรูปแบบในการตรวจสอบข้อความในอีเมลของกฎข้อนี้

บรรทัดที่ 2

describe คือ คำสั่งที่ใช้สำหรับกำหนดคำอธิบายให้กับกฎ

Check_token คือ ชื่อของกฎที่จะถูกกำหนดคำอธิบาย

Checking ลดความอ้วน คือ คำอธิบายของกฎข้อนี้ ซึ่งเป็นคำอธิบายที่สามารถอ่านเข้าใจได้ (Human-readable description) และคำอธิบายนี้จะถูกนำมาแสดงในรายงานเมื่ออีเมลที่ถูกตรวจสอบมีข้อความสอดคล้องกับกฎข้อนี้

บรรทัดที่ 3

score คือ คำสั่งที่ใช้สำหรับกำหนดคะแนนให้กับกฎ

3.5 คือ คะแนนของกฎข้อนี้ โดยถ้าค่าของคะแนนมีค่าสูงจะหมายถึงอีเมลที่มีข้อความสอดคล้องกับ Regular Expression ของกฎข้อนี้มีแนวโน้มจะเป็นอีเมลสแปมสูง

ตัวอย่างกฎในรูปที่ 13 นั้น จะทำให้อีเมลที่มีข้อความสอดคล้องกับ Regular Expression ได้รับคะแนนความเป็นอีเมลสแปมเพิ่มขึ้น 3.5 คะแนน

กฎต่างๆ ของโปรแกรมกรองอีเมลสแปมแอสแซสซินสามารถแบ่งได้เป็น 2 ประเภทดังนี้

1) กฎซึ่งถูกติดตั้งมาพร้อมกับโปรแกรมกรองอีเมลสแปมแอสแซสซิน (Default Rules)

กฎในประเภทนี้จะเป็นกฎที่ถูกติดตั้งมาพร้อมกับโปรแกรมกรองอีเมลสแปมแอสแซสซิน ซึ่งอยู่ในแฟ้ม /usr/share/spamassassin โดยจะทำการตรวจสอบคำหรือข้อความในส่วนต่างๆ ของอีเมล ยกตัวอย่างกฎบางข้อในประเภทนี้ ดังนี้

กฎ FROM_STARTS_WITH_NUMS จะทำการตรวจสอบข้อความที่ปรากฏที่หัวข้อ From ในส่วน Header ของอีเมล ถ้าหากอีเมลมีข้อความในหัวข้อ From ที่เริ่มต้นด้วยตัวเลข 2 หลัก จะทำให้ได้รับคะแนนความเป็นอีเมลสแปมเพิ่มขึ้น 0.390 คะแนน รายละเอียดของกฎ FROM_STARTS_WITH_NUMS แสดงดังข้างล่างนี้

```
header    FROM_STARTS_WITH_NUMS    From =~ /^\d\d/
describe  FROM_STARTS_WITH_NUMS    From: starts with nums
score     FROM_STARTS_WITH_NUMS    0.390
```

กฎ HTML_WIN_OPEN จะทำการตรวจสอบข้อความในส่วน body โดยถ้าหากอีเมลมีข้อความ HTML ซึ่งทำการเปิดหน้าต่างขึ้นมาใหม่ด้วยจาวาสคริปต์ (JavaScript) จะได้รับคะแนนความเป็นอีเมลสแปมเพิ่มขึ้น 0.5 คะแนน รายละเอียดของกฎ HTML_WIN_OPEN แสดงดังข้างล่างนี้

body	HTML_WIN_OPEN	eval:html_test ('window_open')
describe	HTML_WIN_OPEN	Javascript to open a new window
score	HTML_WIN_OPEN	0.5

2) กฎซึ่งสามารถเขียนขึ้นเองได้ (Custom Rules)

ผู้ใช้งานโปรแกรมกรองอีเมลสแปมแอสแซสซินสามารถเขียนกฎเพื่อป้องกันอีเมลสแปมได้เอง หรือทำการ download กฎที่ผู้ใช้งานคนอื่นๆ สร้างขึ้นมาติดตั้งได้

หลักการเขียนกฎขึ้นเองมีดังนี้

1) การเขียนกฎขึ้นเองนั้นต้องกำหนดชื่อของกฎ และคำอธิบายของกฎที่สามารถอธิบายกฎได้ชัดเจน
2) กำหนดว่ากฎที่เขียนขึ้นเองนั้นจะทำการตรวจสอบข้อความอีเมลในส่วนใดของอีเมลโดยใช้คำสั่งต่างๆ ในการกำหนด ดังแสดงในตารางที่ 6

ตารางที่ 6 ตัวอย่างคำสั่งสำหรับการเขียนกฎในโปรแกรมกรองอีเมลสแปมแอสแซสซิน

คำสั่ง	**ส่วนที่ตรวจสอบ**
Header	ข้อความในส่วนหัว (Header) ของอีเมล
Body	ข้อความส่วนเนื้อหา (Body) ของอีเมล
Uri	ข้อความที่มีส่วนเชื่อมโยง (link) ไปยังเวปอื่น

3) สำหรับผู้ใช้งานบางคนที่เขียนกฎขึ้นมาแล้วต้องการให้กฎที่สร้างขึ้นมาทำงานภายใต้เงื่อนไขที่กำหนด ผู้ใช้งานสามารถทำได้โดยใช้คำสั่ง tflag ซึ่งเป็นการกำหนดค่า Test flag โดยจะมีหลายแบบดังแสดงในตารางที่ 7

ตารางที่ 7 ตัวอย่าง Test flag แบบต่างๆ

Test flag	ความหมาย
net	กำหนดให้กฎทำงานเมื่อระบบทำการทดสอบกับ network-test เท่านั้น
Learn	กำหนดให้ระบบต้องมีการเรียนรู้ก่อนจึงจะสามารถใช้กฎข้อนี้ได้
Nice	กำหนดให้กฎทำการให้คะแนนความเป็นอีเมลสแปมเป็นค่าลบ

ยกตัวอย่างกฎที่เขียนขึ้นมาเองดังนี้

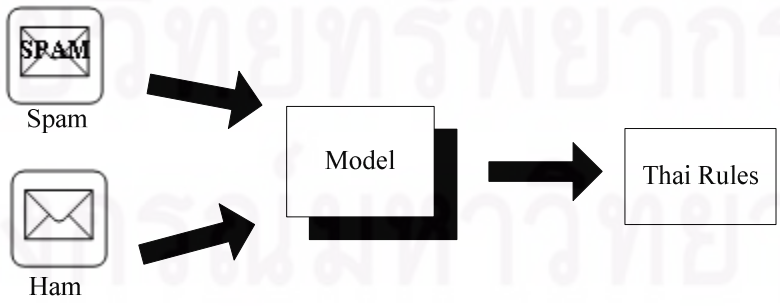
กฎ SARE_SPEC_ROLEX_BRANDS จะทำการตรวจสอบข้อความที่หัวข้อเรื่อง (Subject) ในส่วน Header ของอีเมล ถ้าหากอีเมลมียี่ห้อที่มีชื่อเสียงของสินค้าชนิดต่างๆ จะได้รับคะแนนความเป็นอีเมลสแปมเพิ่มขึ้น 1.11 คะแนน

```
header    SARE_SPEC_ROLEX_BRANDS Subject =~
/(\b(cartier|piaget|omega|longines|vuitton|r(?:[0o@]|aw)lex)\b.{0,30}){3,}/i
describe    SARE_SPEC_ROLEX_BRANDS    Spammer subject - multiple
brands
score       SARE_SPEC_ROLEX_BRANDS    1.1
```

4.2 หลักการสร้างกฎด้วยวิธีการทางสถิติ

ในการสร้างกฎด้วยวิธีทางสถิติจำเป็นต้องใช้ทั้งตัวอย่างอีเมลที่ดีและตัวอย่างอีเมลสแปม ซึ่งเก็บรวบรวมมาจากเครื่องแม่ข่ายอีเมลจำนวนหลายเครื่องหรือการรายงานจากผู้ใช้งานอีเมล

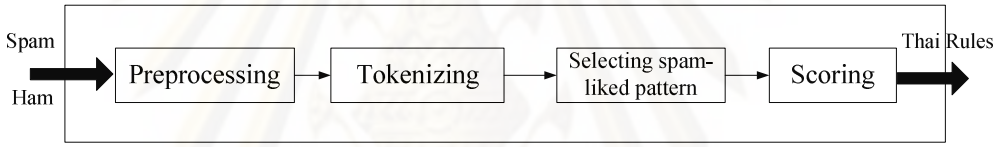
ตัวอย่างอีเมลสแปมที่ดีจะถูกแยกไว้ที่แฟ้มกล่องจดหมายขาเข้า (Inbox Folder) และตัวอย่างอีเมลสแปมจะถูกแยกไว้ที่แฟ้มจดหมายขยะ (Junk Folder) โดยที่ตัวระบบ (Model) จะทำการดึงข้อมูลตัวอย่างทั้งสองแบบโดยตรงจากแฟ้มทั้งสอง



รูปที่ 14 กรอบการทำงานของระบบการสร้างกฎด้วยวิธีการทางสถิติ

การทำงานของระบบดังแสดงในรูปที่ 14 ผลลัพธ์ที่ได้คือกฎสามารถแก้ไขปัญหาอีเมลสแปมภาษาไทย ซึ่งเป็นกฎทำการตรวจสอบข้อความในส่วนของหัวข้อเรื่อง (Subject) และในส่วนของเนื้อหา (Body) และเมื่อมีตัวอย่างอีเมลรูปแบบใหม่ๆ เข้ามา ก็สามารถเรียนรู้อีเมลรูปแบบใหม่ๆ ได้ ทำให้กฎที่ได้มีความทันสมัย

ชุดตัวอย่างข้อมูลของอีเมลทั้งสองแบบ ซึ่งก็คือฐานข้อมูล (Databases) จะถูกส่งเข้าไปประมวลผลในส่วนตัวระบบ (Model) ซึ่งภายในตัวระบบจะมีกระบวนการหลายขั้นตอน ยกตัวอย่างเช่น กระบวนการเตรียมข้อมูล (Preprocessing) กระบวนการตัดคำ (Tokenizing) กระบวนการคัดเลือกคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปม (Selecting spam-liked pattern) และกระบวนการกำหนดคะแนนให้กับกฎ (Scoring) ซึ่งกระบวนการตัดคำ กระบวนการคัดเลือกคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปม และกระบวนการกำหนดคะแนนให้กับกฎอ้างอิงจากงานวิจัย [27] ตัวอย่างรายละเอียดภายในตัวระบบแสดงในรูปที่ 15



รูปที่ 15 รายละเอียดภายในตัวระบบ (Model)

4.2.1 **กระบวนการเตรียมข้อมูล** (Preprocessing)

ในกระบวนการนี้ระบบจะอ่านข้อมูลตัวอย่างอีเมลที่ดีและตัวอย่างอีเมลสแปมจากแฟ้มกล่องจดหมายขาเข้าและแฟ้มจดหมายขยะ แต่ข้อมูลตัวอย่างทั้งสองแบบยังไม่มีความเหมาะสมที่จะประมวลผลได้

ในอันดับแรกข้อมูลตัวอย่างทั้งสองประเภทจะถูกเก็บอยู่ในรูปแบบไฟล์ที่ถูกเข้ารหัส ยกตัวอย่างเช่น Base64 เป็นต้น ตัวอย่างข้อความที่ถูกเข้ารหัส Base64 ดังแสดงข้างล่างนี้

```
PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFkPgogIDxib2R5PgogICAgPHA+VGh
pcyBpcyB0aGUgYm9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8
L2h0bWw+Cg==
```

ภายในกระบวนการเตรียมข้อมูลนี้จะทำการถอดรหัสข้อความรหัสให้ได้ข้อความที่สามารถอ่านเข้าใจได้ (Human Reading) ตัวอย่างข้อความรหัสซึ่งถูกถอดรหัสเรียบร้อยแสดงดังข้างล่างนี้

```
<html>
 <head>
 </head>
 <body>
  <p>This is the body of the message.</p>
 </body>
</html>
```

หลังจากที่ได้ข้อมูลตัวอย่างของทั้งสองประเภทที่สามารถอ่านเข้าใจได้แล้ว ข้อมูลตัวอย่างต้องถูกแก้ไขลบคำ (Token) ที่ไม่มีประโยชน์ต่อการประมวลผล เช่น คำสั่งในภาษาเอชทีเอ็มแอล (HTML tags) เป็นต้น ข้อมูลตัวอย่างที่ถูกลบคำที่ไม่มีประโยชน์ออกแล้ว แสดงดังข้างล่างนี้

```
This is the body of the message.
```

4.2.2 กระบวนการตัดคำ (Tokenizing)

ในกระบวนการนี้ระบบจะนำข้อมูลตัวอย่างทั้งสองแบบที่มีความเหมาะสมต่อการประมวลผลมาค้นหารูปแบบคำหรือวลีที่สำคัญที่สามารถบ่งบอกได้ว่าเป็นอีเมลสแปม (Spam-liked Pattern)

แต่ปัญหาสำคัญ คือ การประมวลผลภาษาไทยซึ่งภาษาไทยมีความแตกต่างกับภาษาอื่นๆ เช่น ภาษาอังกฤษ เป็นต้น จากลักษณะความแตกต่างที่สำคัญระหว่างภาษาไทยและภาษาอังกฤษถูกแสดงอยู่ในตารางที่ 2 ทำให้จำเป็นต้องใช้โปรแกรมตัดคำ (Word Segmentation) มาช่วยในการประมวลผลภาษาไทย

ผู้เสนอวิทยานิพนธ์ได้นำโปรแกรมตัดคำไทยที่มีประสิทธิภาพและเป็นที่รู้จัก คือ โปรแกรมตัดคำไทยคูวส์ (CUWS) โปรแกรมตัดคำไทย Java BreakIterator [33] โปรแกรมตัดคำไทย DictionaryBasedBreakIterator [34] โปรแกรมตัดคำไทย Swath [35] และโปรแกรมตัดคำไทย Cttex [36] มาทดสอบประสิทธิภาพโดยให้ทำการตัดคำไทยบนชุดข้อมูลทดสอบ BEST2010 ซึ่งเป็นข้อมูลทดสอบที่สร้างโดยหน่วยงานของรัฐ โดยมุ่งเน้นเพื่อนำข้อมูลทดสอบนี้มาทดสอบประสิทธิภาพโปรแกรมตัดคำไทยในการแข่งขันแบ่งคำไทย BEST2010 [37] จึงเป็นชุดข้อมูลที่มีความน่าเชื่อถือและใช้ทดสอบประสิทธิภาพของโปรแกรมตัดคำไทยได้

ตารางที่ 8 การเปรียบเทียบประสิทธิภาพโปรแกรมตัดคำไทย

โปรแกรมตัดคำไทย	**F-measure**
CUWS	93.562%
Java BreakIterator	83.859%
DictionaryBasedBreakIterator	83.657%
Swath	82.64%
Cttex	56.57%

จากตารางที่ 8 แสดงให้เห็นว่าโปรแกรมตัดคำไทยคูวส์นั้นมีประสิทธิภาพมากที่สุด

ผู้เสนอวิทยานิพนธ์ได้ใช้โปรแกรมตัดคำไทยคูวส์ (CUWS) เพื่อช่วยประมวลผลคำไทยซึ่งมีรายละเอียดแสดงในหัวข้อ 3.3

4.2.3 กระบวนการคัดเลือกคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปม (Selecting spam-liked pattern)

ในกระบวนการนี้จะทำการคัดเลือกคำหรือวลีที่สำคัญซึ่งบ่งบอกว่าเป็นอีเมลสแปมซึ่งปรากฏทั้งในส่วนหัวข้อเรื่องและส่วนเนื้อหา

กำหนดให้คำ คือ ตัวแปร t โดยจะทำการคำนวณความสัมพันธ์ระหว่าง t กับความเป็นอีเมลที่ดี (V_{th}) และความสัมพันธ์ระหว่าง t กับความเป็นอีเมลสแปม (V_{ts}) จากนั้นจะทำการหาค่าอัตราส่วน R_t ดังสมการต่อไปนี้

$$R_t = \frac{V_{ts}}{V_{th}} \tag{3}$$

คำหรือวลีที่สำคัญที่สามารถบ่งบอกได้ว่าเป็นอีเมลสแปมจะถูกคัดเลือกจากคำมีค่าอัตราส่วน R_t มากที่สุดจำนวน N อันดับแรก โดยที่ N คือจำนวนของกฎ

มีการนำเสนอวิธีการคัดเลือกคำหรือวลีที่สำคัญที่สามารถบ่งบอกได้ว่าเป็นอีเมลสแปมมากมายหลายวิธี ซึ่งคือสูตรที่ใช้ในการคำนวณค่า V_{ts} และค่า V_{th} ดังแสดงต่อไปนี้

กำหนดให้ E คือ สมมติฐานที่อีเมลจะเป็นอีเมลสแปม

H คือ สมมติฐานที่อีเมลจะมี t ปรากฏอยู่

- วิธีการ Document Frequency (DF)

$$V_{ts} = P(H|E) = \frac{P(E \bigcap H)}{P(E)} \tag{4}$$

$$V_{th} = P(H|\bar{E}) = \frac{P(\bar{E} \bigcap H)}{P(\bar{E})} \quad (5)$$

- วิธีการ Conditional Probabilities and Bayes's Theorem (CP)

$$V_{ts} = P(E|H) = \frac{P(E \bigcap H)}{P(H)} \quad (6)$$

$$V_{th} = P(\bar{E}|H) = \frac{P(\bar{E} \bigcap H)}{P(H)} \quad (7)$$

- วิธีการ Mutual Information (MI)

$$V_{ts} = \log\left(\frac{P(E \bigcap H)}{P(E)P(H)}\right) \quad (8)$$

$$V_{th} = \log\left(\frac{P(\bar{E} \bigcap H)}{P(\bar{E})P(H)}\right) \quad (9)$$

- วิธีการ Information Gain (IG)

$$V_{ts} = -P(E)\log\left(P(E)\right) + P(E \bigcap H)\log\left(\frac{P(E \bigcap H)}{P(H)}\right) \quad (10)$$

$$V_{th} = -P(\bar{E})\log\left(P(\bar{E})\right) + P(\bar{E} \bigcap H)\log\left(\frac{P(\bar{E} \bigcap H)}{P(H)}\right) \quad (11)$$

- วิธีการ Kullback-Leibler divergence (KL)

$$V_{ts} = \frac{P(E \bigcap H)}{P(H)}\log\left(\frac{P(E \bigcap H)}{P(E)P(H)}\right) \quad (12)$$

$$V_{th} = \frac{P(\bar{E} \bigcap H)}{P(H)}\log\left(\frac{P(\bar{E} \bigcap H)}{P(\bar{E})P(H)}\right) \quad (13)$$

วิธีการหาค่าค่างๆ ในสมการที่ (4) – (13) นั้น สามารถคำนวณได้ดังนี้

กำหนดให้ A คือ จำนวนครั้งที่อีเมลสแปมมี t ปรากฏอยู่

B คือ จำนวนครั้งที่อีเมลที่ดีมี t ปรากฏอยู่

C คือ จำนวนครั้งที่อีเมลสแปมไม่มี t ปรากฏอยู่

D คือ จำนวนครั้งที่อีเมลที่ดีไม่มี t ปรากฏอยู่

$$P(E) = \frac{A+C}{A+B+C+D} \quad (14)$$

$$P(\bar{E}) = \frac{B+D}{A+B+C+D} \quad (15)$$

$$P(H) = \frac{A+B}{A+B+C+D} \quad (16)$$

$$P(E \bigcap H) = \frac{A}{A+B+C+D} \quad (17)$$

$$P(\bar{E} \bigcap H) = \frac{B}{A+B+C+D} \quad (18)$$

จากการศึกษาของงานวิจัยก่อนหน้า [27] ซึ่งทำการศึกษาและเปรียบเทียบวิธีการคัดเลือกคำหรือวลีที่สำคัญมาสร้างเป็นกฎ คือ วิธีการ DF วิธีการ CP วิธีการ MI วิธีการ IG และวิธีการ KL พบว่าวิธีการ CP นั้นเป็นวิธีการคัดเลือกคำมาสร้างเป็นกฎได้มีประสิทธิภาพสูงที่สุด

มีงานวิจัยมากมายที่สนับสนุนแนวคิดการนำวิธีการ Bayesian มาประยุกต์ใช้ในการคัดกรองอีเมลสแปม [31],[38],[39],[40],[41],[42] และชี้ให้เห็นว่าวิธีการ Bayesian เป็นวิธีการที่มีประสิทธิภาพสูงในการคัดกรองอีเมล ยกตัวอย่างเช่น งานวิจัย [31] ชี้ให้เห็นถึงประสิทธิภาพของวิธีการ Naive Bayesian ซึ่งมีประสิทธิภาพคัดกรองอีเมลสแปมได้ถึง 98.3 เปอร์เซ็นต์ เป็นต้น

นอกจากนี้ยังมีงานวิจัย [43] ที่ทำการเปรียบเทียบประสิทธิภาพวิธีการเรียนรู้ของเครื่อง (Machine Learning) ที่นำมาประยุกต์ใช้ในการกรองอีเมลสแปม คือ วิธีการ Naive Bayesian วิธีการ Term Frequency-Inverse Document Frequency วิธีการ K-nearest neighbor และวิธีการ Support Vector Machine ซึ่งผลการทดลองชี้ให้เห็นว่าวิธีการ Bayesian และวิธีการ Term Frequency-Inverse Document Frequency มีประสิทธิภาพอยู่ในเกณฑ์ดี และในบางชุดข้อมูลพบว่าวิธีการ Beyasian มีประสิทธิภาพกรองอีเมลสแปมสูงที่สุด

ดังนั้นในวิทยานิพนธ์นี้จึงได้ใช้วิธีการ Conditional Probabilities and Bayes' Theorem ในการคัดเลือกคำมาสร้างกฎ

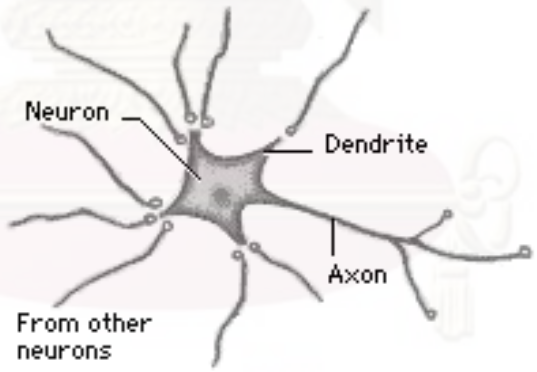
4.2.4 กระบวนการกำหนดคะแนนให้กับกฎ (Scoring)

เมื่อได้คำหรือวลีที่สำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปมแล้วในกระบวนการนี้จะทำการกำหนดคะแนนให้กับคำสำคัญแต่ละคำ คะแนนนั้นจะมีทั้งเป็นค่าจำนวนจริงบวกและค่าจำนวนจริงลบ

คะแนนที่มีค่าเป็นจำนวนจริงบวกนั้นจะให้คะแนนกับคำสำคัญที่เกิดขึ้นบ่อยบนกลุ่มอีเมลสแปม ส่วนคะแนนที่มีค่าเป็นจำนวนจริงลบนั้นจะให้คะแนนกับคำสำคัญที่เกิดขึ้นบ่อยบนกลุ่มอีเมลที่ดี วิทยานิพนธ์นี้จะทำการให้คะแนนเฉพาะคำที่สามารถบ่งบอกความเป็นอีเมลสแปมได้เท่านั้น

ก่อนหน้านี้วิธีการให้คะแนนกับกฎของโปรแกรมสแปมแอสแซสซินจะใช้วิธีการทางพันธุกรรมแต่เนื่องจากวิธีนี้ใช้เวลาการประมวลผลนานมาก [27] ทำให้มีผู้พัฒนาโปรแกรมการให้คะแนนกับกฎของโปรแกรมสแปมแอสแซสซินชื่อว่า The Fast SpamAssassin Score Learning Tool [44] ซึ่งเป็นโปรแกรมที่สามารถทำการให้คะแนนกับกฎได้แต่ใช้เวลาการประมวลผลเร็วกว่ามาก โปรแกรมนี้ใช้วิธีโครงข่ายประสาทเทียม (Aritificial Neural Network)

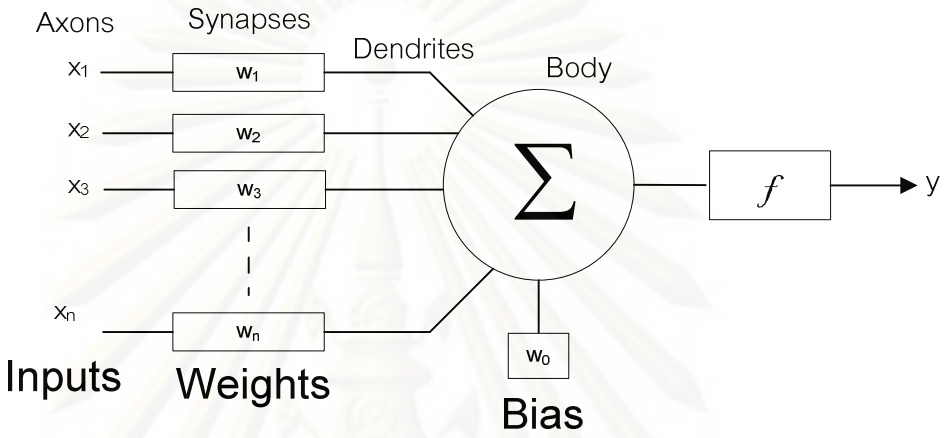
วิธีโครงข่ายประสาทเทียมเป็นระบบการคำนวณที่เลียนแบบการทำงานของสมองมนุษย์ โครงสร้างเซลล์ประสาทของสมองมนุษย์แสดงในรูปที่ 16



รูปที่ 16 โครงสร้างของเซลล์ประสาทในสมองมนุษย์

จากรูปที่ 16 เซลล์ประสาทของมนุษย์จะประกอบด้วยเซลล์ (Cell Body) และมีแขนงยื่นออกจากตัวเซลล์ คือ เดนไดรท์ (Dendrite) ซึ่งใช้รับสัญญาณไฟฟ้าเข้าสู่เซลล์ และแอกซอน (Axon) มีกิ่งก้านน้อยกว่าเดนไดรท์ แต่มีความยาวมากกว่าเดนไดรท์ ทำหน้าที่นำสัญญาณออกจากเซลล์ประสาท ส่วนปลายของแอกซอนมีลักษณะเป็นปมและจ่ออยู่จนเกือบสัมผัสกับปลายของเดนไดรท์

ส่วนที่ต่อเชื่อมกันระหว่างปลายของเดนไดรท์และปลายของแอกซอน คือ ไซแนปส์ (Synapse) เซลล์ประสาทจะรับข้อมูลนำเข้า (Input) ผ่านทางไซแนปส์ เมื่อปลายของแอกซอนได้รับสัญญาณไฟฟ้าจะทำให้เกิดการกระตุ้นให้เกิดการส่งผ่านสัญญาณเชิงเคมีผ่านไซแนปส์ เดนไดร์ทจะตีความสัญญาณเชิงเคมีนั้นเป็นสัญญาณไฟฟ้าวิ่งเข้าสู่เซลล์ประสาทต่อไป

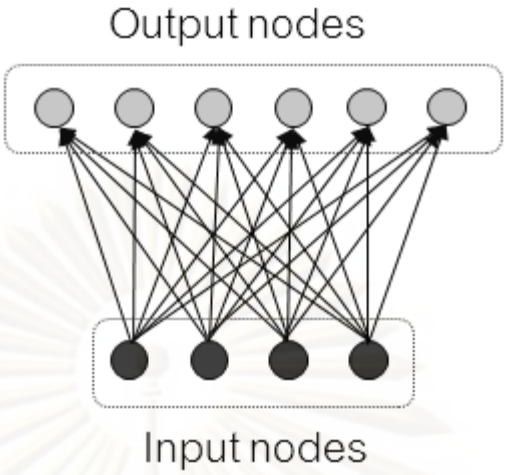


รูปที่ 17 โครงสร้างของเซลล์ประสาทเทียม

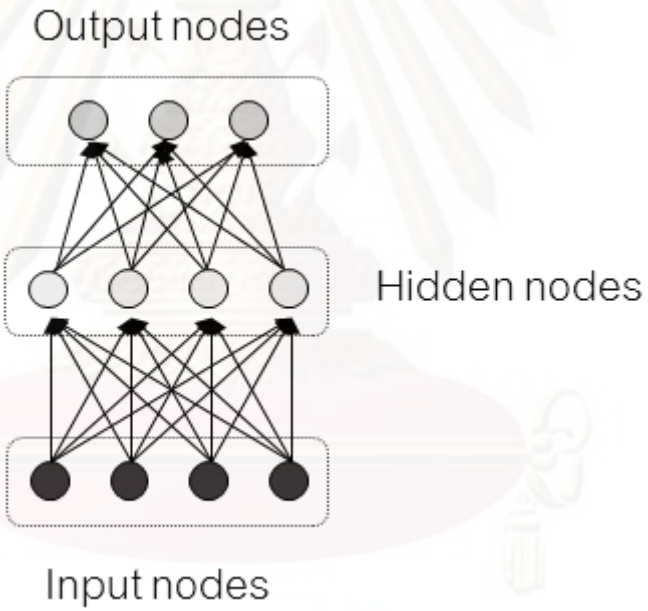
จากรูปที่ 17 เซลล์ประสาทเทียมประกอบด้วยสัญญาณนำเข้า (Inputs) คือ x_i ซึ่งเปรียบเทียบได้กับสัญญาณที่เข้ามายังเดนไดรท์ของเซลล์ประสาทของมนุษย์ ค่าถ่วงน้ำหนัก (Weights) คือ w_i ซึ่งเปรียบเทียบได้กับไซแนปส์ซึ่งเป็นส่วนที่ส่งผ่านสัญญาณเข้าสู่เซลล์ และสัญญาณส่งออกคือ y

สัญญาณนำเข้าที่ถูกส่งเข้ามายังเซลล์ประสาทเทียมจะถูกรวมสัญญาณ และถูกส่งไปยังฟังก์ชันกระตุ้น (Activation Function) สำหรับการกำหนดสัญญาณส่งออกซึ่งเปรียบเทียบได้กับเมื่อสัญญาณนำเข้าที่ถูกส่งเข้ามาทางเดนไดรท์ แล้วเข้าสู่เซลล์ประสาทของมนุษย์ หากรวมสัญญาณที่เข้ามายังเซลล์ประสาทแล้วผลรวมมีค่าเกินค่าระดับ (Threshold) สัญญาณไฟฟ้าจะถูกส่งออกมาจากเซลล์ประสาทผ่านทางแอกซอน

เซลล์ประสาทเทียมหลายๆ ตัวจะเชื่อมต่อกันเกิดเป็นลักษณะโครงข่ายจัดอยู่ในรูปของชั้น (Layer) โครงสร้างการเชื่อมต่อโครงข่ายประสาทเทียมจะแบ่งออกเป็นโครงข่ายประสาทแบบชั้นเดียว (Single Layer Perceptron Networks) ดังแสดงในรูปที่ 18 และโครงข่ายประสาทเทียมแบบหลายชั้น (Multi Layer Perceptron Networks) ดังแสดงในรูปที่ 19



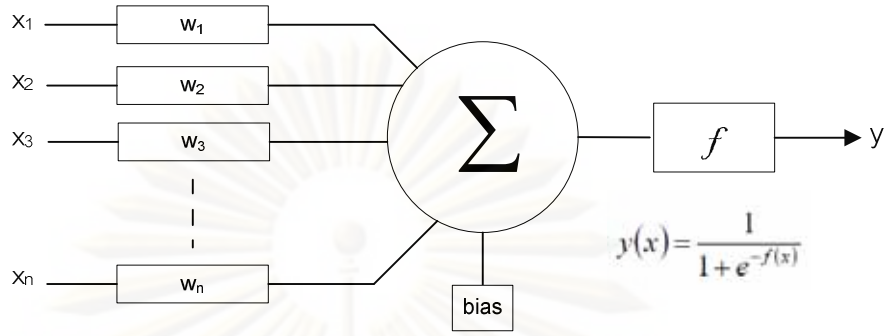
รูปที่ 18 โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer Perceptron)



รูปที่ 19 โครงข่ายประสาทเทียมแบบหลายชั้น (Multi Layer Perceptron Network)

จากรูปที่ 18 โครงข่ายประสาทเทียมแบบชั้นเดียวจะประกอบด้วยชั้นข้อมูลนำเข้า (Input Layer) และชั้นข้อมูลส่งออก (Output Layer) และจากรูปที่ 19 โครงข่ายประสาทเทียมแบบหลายชั้นจะประกอบด้วยชั้นข้อมูลนำเข้า (Input Layer) ชั้นที่อยู่ตรงกลาง คือ ชั้นซ่อน (Hidden Layer) ซึ่งอาจจะมีมากกว่า 1 ชั้นก็ได้ และชั้นข้อมูลส่งออก (Output Layer)

โปรแกรม The Fast SpamAssassin Score Learning Tool นั้นใช้โครงสร้างประสาทเทียมแบบชั้นเดียว (Single Layer Perceptron) มีรายละเอียดแสดงในรูปที่ 20



รูปที่ 20 โครงสร้างประสาทเทียมของโปรแกรม The Fast SpamAssassin Score Learning Tool

จากรูปที่ 20 ภายในโครงสร้างประสาทเทียมของโปรแกรม The Fast SpamAssassin Score Learning Tool จะมีฟังก์ชันถ่ายโอน (Transfer Function) เป็นฟังก์ชันถ่ายโอนเชิงเส้น (Linear Transfer Function) ดังสมการที่ 19 และมีฟังก์ชันกระตุ้น (Activation Function) เป็นฟังก์ชันซิกมอยด์แบบลอการิทึม (Log-Sigmoid Transfer Function) ดังสมการที่ 20

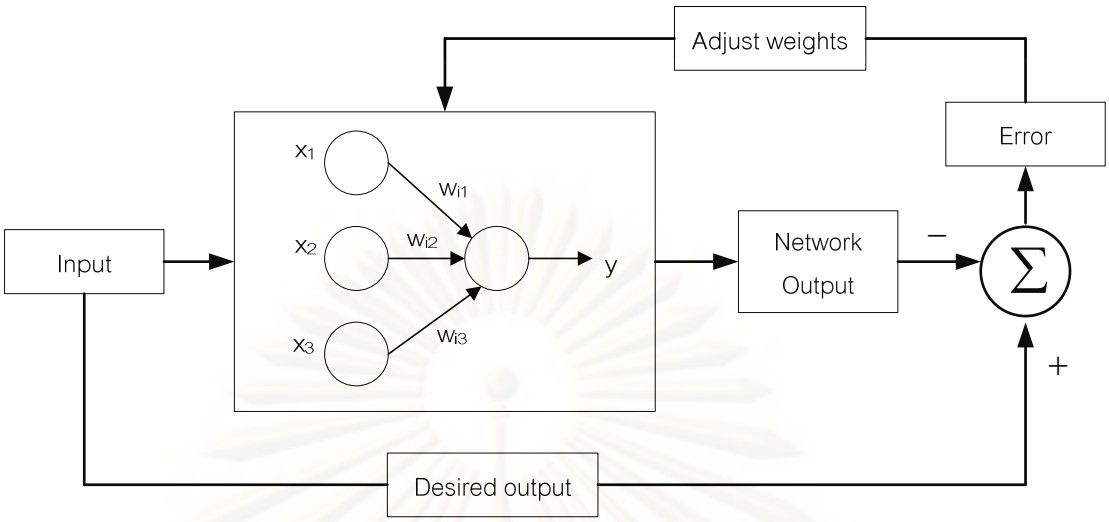
$$f(x) = bias + \sum_{i=1}^{N} x_i \, w_i \tag{19}$$

$$y(x) = \frac{1}{1 + e^{-f(x)}} \tag{20}$$

ค่า w_i คือคะแนนของกฎข้อที่ i

ค่า x_i คือค่าตรรกะแบบบูล (Boolean) จะเป็น 1 เมื่อกฎข้อที่ i ถูกกระตุ้นโดยตัวอย่างข้อความที่นำมาเรียนรู้ และจะเป็น 0 เมื่อกฎข้อที่ i ไม่ถูกกระตุ้นโดยตัวอย่างข้อความที่นำมาเรียนรู้

การเรียนรู้โครงข่ายประสาทเทียมของโปรแกรม The Fast SpamAssassin Score Learning Tool เป็นแบบมีผู้ฝึกสอน (Supervised Learning)



รูปที่ 21 การเรียนรู้ของโปรแกรม The Fast SpamAssassin Score Learning Tool

จากรูปที่ 21 เมื่อข้อมูลนำเข้า (Input) ถูกป้อนให้กับโครงข่าย โครงข่ายจะทำการประมวลผลจนได้คำตอบและค่าถ่วงน้ำหนักมาชุดหนึ่ง สำหรับคำตอบที่ได้จากโครงข่าย (Network Output) จะถูกนำมาคำนวณค่าความคลาดเคลื่อนโดยวัดเป็นระยะห่างจากคำตอบที่ต้องการ ในโปรแกรมนี้ใช้ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Mean Squared Error) ดังสมการที่ 21

$$E(x) = y(x)*(1-y(x))*(y_{desired}-y(x)) \quad (21)$$

ถ้าหากมีความคลาดเคลื่อนสูงก็จะทำการปรับค่าถ่วงน้ำหนัก ดังสมการที่ 22 (α คือ อัตราการเรียนรู้)

$$w_i = w_i + \alpha * E(x) * x_i \quad (22)$$

โปรแกรมจะทำการสอนต่อไปจนกว่าค่าความคลาดเคลื่อนระหว่างคำตอบที่ได้จากโครงข่ายกับคำตอบที่ต้องการมีค่าน้อยถึงระดับที่ยอมรับได้จึงหยุดสอน

เมื่อโปรแกรมหยุดสอนแล้ว ค่าถ่วงน้ำหนักจะถูกแปลงเป็นคะแนนของกฎ ดังสมการที่ 23

$$\text{score(weight)} = \text{-threshold} * \text{weight} / \text{bias} \quad (23)$$

บทที่ 5
การพัฒนาระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์สำหรับภาษาไทย

จากหลักการสร้างและออกแบบระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์สำหรับภาษาไทยและใช้โปรแกรมตัดคำไทยในบทที่ 3 ผู้เสนอวิทยานิพนธ์ได้นำมาใช้ในการพัฒนาระบบซึ่งจะกล่าวถึงรายละเอียดการพัฒนาในบทนี้ ซึ่งประกอบด้วย สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา และขั้นตอนในการพัฒนาระบบ

5.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนาระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีระบบการเรียนรู้แบบเบย์และใช้โปรแกรมตัดคำไทย

การพัฒนาระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีระบบการเรียนรู้แบบเบย์และใช้โปรแกรมตัดคำไทยถูกพัฒนาขึ้นภายใต้สภาพแวดล้อมทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ของเครื่องที่ใช้พัฒนาระบบดังต่อไปนี้

<u>ฮาร์ดแวร์</u>

1. หน่วยประมวลผลกลาง (CPU) อินเทล คอร์ 2 ดูโอ 2.66 กิกะเฮิรตซ์ (Intel Core2Duo 2.66 Ghz)
2. หน่วยความจำ (RAM) 4 กิกะไบต์ (4GB)
3. จานบันทึกข้อมูล (Hard Disk) 160 กิกะไบต์ (160GB)

<u>ซอฟต์แวร์</u>

1. ระบบปฏิบัติการ อูบันตูสำหรับเดสก์ท็อป เวอร์ชัน 8.04 (Ubuntu Destop 8.04)
2. โปรแกรมกรองอีเมลสแปมแอสแซสซิน เวอร์ชัน 3.2.4 (SpamAssassin 3.2.4)
3. ตัวแปลภาษาเพิร์ล (Perl Interpreter)
4. โปรแกรมตัดคำไทยคูวส์ (Chulalongkorn University Word Segmentation: CUWS)
5. โปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค (Postfix)

5.2 การติดตั้งซอฟต์แวร์ในการสร้างเครื่องแม่ข่ายอีเมล (Mail Server)

เมื่อเตรียมเครื่องมือสำหรับการพัฒนาระบบเรียบร้อยแล้ว จึงทำการติดตั้งและปรับแต่งเครื่องมือต่างๆ เพื่อพัฒนาระบบกรองอีเมลสแปมสำหรับภาษาไทยที่มีระบบการเรียนรู้แบบเบย์

และใช้โปรแกรมตัดคำไทย โดยในส่วนแรกจะทำการสร้างเครื่องแม่ข่ายอีเมลโดยติดตั้งซอฟต์แวร์ต่างๆ ดังนี้

5.2.1 **ติดตั้งระบบปฏิบัติการอูบันตูสำหรับเดสก์ท็อป เวอร์ชัน** 8.04 (Ubuntu Desktop 8.04)

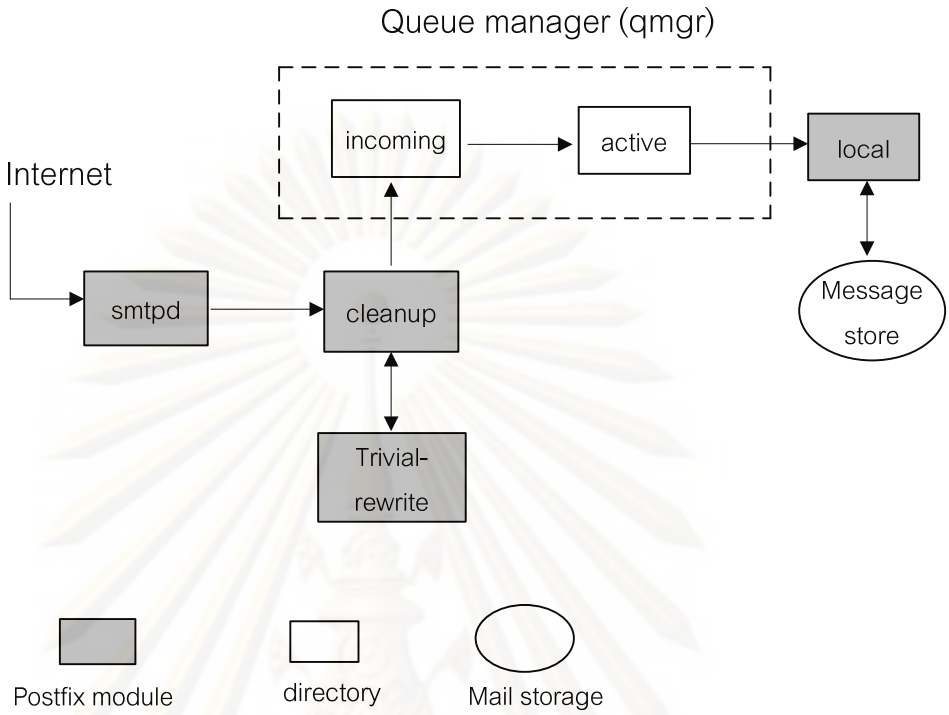
ระบบปฏิบัติการอูบันตูสำหรับเดสก์ท็อป (Ubuntu Desktop) เป็นระบบปฏิบัติการซึ่งทำงานบนระบบลินุกซ์และแยกตัวออกมาจาก Debian ซอฟต์แวร์ต่างๆ ที่รวมมาอยู่ในอูบันตูนั้นเป็นซอฟต์แวร์เสรีทั้งหมด ผู้เสนอวิทยานิพนธ์เลือกระบบปฏิบัติการอูบันตูสำหรับเดสก์ท็อป เนื่องจากเป็นซอฟต์แวร์เสรี มีการใช้งานง่าย มีคู่มือประกอบการใช้งานมากมาย และยังสามารถติดตั้งเครื่องแม่ข่ายอีเมล (Mail Server) และโปรแกรมกรองอีเมลต่างๆ เช่น โปรแกรมสแปมแอสแซสซิน เป็นต้น ได้ง่ายเหมาะแก่การนำมาพัฒนา สามารถดูรายละเอียดการติดตั้งได้ในภาคผนวก

5.2.2 **ติดตั้งโปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค** (Postfix)

โปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค (Postfix) เป็น MTA (Mail Transfer Agent) บนระบบยูนิกซ์และเป็นซอฟต์แวร์เสรี ถูกพัฒนาขึ้นมาเพื่อให้เป็นโปรแกรมเครื่องแม่ข่ายอีเมลที่มีคุณภาพและมีคุณสมบัติสำคัญด้านต่างๆ ยกตัวอย่างดังต่อไปนี้

- เป็นเครื่องแม่ข่ายอีเมลที่มีความน่าเชื่อถือโดยเมื่ออยู่ในสภาพที่พื้นที่ไม่พอก็ยังสามารถทำงานได้
- เป็นเครื่องแม่ข่ายอีเมลที่มีประสิทธิภาพโดยโพสฟิคมีการใช้เทคนิคการจำกัดการสร้างโพสเซสใหม่ขึ้นและลดการเข้าถึงไฟล์ต่างๆ ในระหว่างการทำงานลง
- เป็นเครื่องแม่ข่ายอีเมลที่มีความยืดหยุ่นโดยแบ่งแยกส่วนต่างๆ ออกเป็นมอดูลทำให้ง่ายต่อการแก้ไขดัดแปลง
- เป็นเครื่องแม่ข่ายอีเมลที่มีการใช้งานง่ายทั้งในการติดตั้งและการดูแลระบบ โดยผ่านคอนฟิคไฟล์ (Config file) ที่สามารถอ่านและเข้าใจได้ง่าย

ตัวอย่างโครงสร้างและการทำงานของโปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค (Postfix) ระหว่างรับข้อความที่ถูกส่งเข้ามาจากเครือข่ายอินเตอร์เน็ทแสดงในรูปที่ 22



รูปที่ 22 ตัวอย่างโครงสร้างและการทำงานของโปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิคระหว่างรับข้อความจากอินเตอร์เน็ท

ที่มา : Alan Schwartz. SpamAssassin. O'Reilly, 2004

การทำงานของโปรแกรมนั้น จะมีดีมอนมาสเตอร์ (Master Daemon) คอยทำหน้าที่ประสานงานระหว่างมอดูลต่างๆ ภายในโปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค เมื่อมีอีเมลเข้ามายังเครื่องแม่ข่ายอีเมลดีมอนเอสเอ็มทีพี (SMTP Daemon) ทำการตรวจสอบอีเมลที่เข้ามาจากเครือข่ายอินเตอร์เน็ทโดยค้นหาจากฐานข้อมูลของที่อยู่ไอพีซึ่งเป็นแหล่งที่มาของการส่งอีเมลสแปม (DNSBL) หลังจากนั้นจะส่งอีเมลเข้าสู่มอดูล Cleanup

ในส่วนมอดูล Cleanup จะทำการตรวจสอบรูปแบบส่วนหัวของอีเมลในฟิลด์ต่างๆ เช่น From, Date, Recipien Address เป็นต้น เพื่อให้เป็นไปตามมาตรฐาน RFC822 และยังมีการทำงานร่วมมอดูล Trivial-rewrite เพื่อทำการเขียนที่อยู่ให้กับอีเมลในรูปแบบ Fully Qualified Domain Name เมื่อทำการตรวจสอบในมอดูล Cleanup เรียบร้อยแล้วครับ จะนำอีเมลส่งไปยังคิว Incoming

ในคิว Incoming จะทำการเรียกดีมอนคิวเอ็มจีอาร์ (QMGR Daemon) ให้ทำงาน และดีมอนคิวเอ็มจีอาร์จะแปลงสถานะของอีเมลจากคิว Incoming เป็นคิว Active (สถานะพร้อมส่ง) หลังจากนั้นจะทำการเรียกมอดูล Local ให้นำอีเมลส่งไปยังกล่องข้อความของผู้รับที่อยู่ภายใน

เครือข่ายเดียวกัน (Local) แต่จะทำการเรียกมอดูลเอสเอ็มทีพีแทนมอดูลโลคัลถ้าหากผู้รับอยู่คนละเครือข่าย

นอกจากนี้โปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิคยังถูกบรรจุอยู่ในชุดซอฟต์แวร์เสรีในคลังโปรแกรมของอูบันตู (Repository) ทำให้สามารถติดตั้งโพสฟิคได้ง่ายบนระบบปฏิบัติการอูบันตู สามารถดูรายละเอียดได้ในภาคผนวก

5.3 การติดตั้งซอฟต์แวร์กรองอีเมลให้กับเครื่องแม่ข่ายอีเมล

เมื่อติดตั้งระบบในส่วนแรก คือ เครื่องแม่ข่ายอีเมลเรียบร้อยแล้ว จึงทำการติดตั้งซอฟต์แวร์ในส่วนที่สอง คือ ระบบกรองอีเมลให้กับเครื่องแม่ข่ายอีเมลโดยติดตั้งซอฟต์แวร์ต่างๆ ดังนี้

5.3.1 ติดตั้งตัวแปรภาษาเพิร์ล (Perl Interpreter)

ตัวแปลภาษาเพิร์ล (Perl Interpreter) เป็นโปรแกรมที่ทำให้สามารถเรียกใช้คำสั่งจากไลบรารี (Libraly) หรือมอดูลของภาษาเพิร์ลให้ทำงานได้โดยโปรแกรมกรองอีเมลที่ใช้ในวิทยานิพนธ์นี้ คือ โปรแกรมกรองอีเมลสแปมแอสแซสซินมีความจำเป็นต้องใช้ตัวแปรภาษาเพิร์ลจึงจะสามารถทำงานได้

5.3.2 ติดตั้งโปรแกรมกรองอีเมลสแปมแอสแซสซิน

โปรแกรมกรองอีเมลสแปมแอสแซสซิน (SpamAssassin) เป็นซอฟต์แวร์เสรี (Open source Software) ที่ถูกเขียนขึ้นด้วยภาษาเพิร์ล (Perl) โดยสถาบันพัฒนาซอฟท์แวร์อะพาเช่ (Apache Software Foundation) เพื่อกรองอีเมลที่เข้ามายังเครื่องแม่ข่ายอีเมลว่าเป็นอีเมลที่ดีหรือเป็นอีเมลสแปม

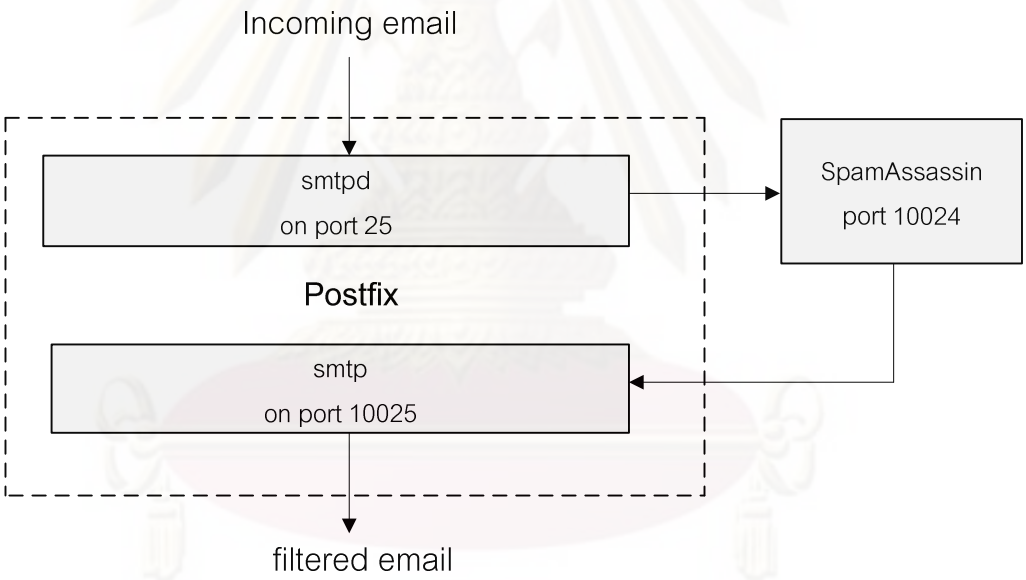
โปรแกรมกรองอีเมลสแปมแอสแซสซินมีข้อดีที่สำคัญหลายประการดังนี้

- ใช้วิธีการตรวจสอบอีเมลที่เข้ามายังเครื่องแม่ข่ายอีเมลหลายวิธี (Multi-technique Approach) ยกตัวอย่างเช่น การตรวจสอบการยืนยันตัวตนของผู้ส่ง (SPF) การตรวจสอบที่มาของอีเมลจากรายชื่อผู้ส่งที่ดี (Whitelist) การตรวจสอบที่มาของอีเมลจากรายชื่อผู้ส่งอีเมลสแปม (Blacklist) การตรวจสอบอีเมลจากฐานข้อมูลเครื่องแม่ข่าย DCC (Fingerprints) การวิเคราะห์เนื้อหาของอีเมล (Content Analysis) รวมถึงวิธีการกรองอีเมลสแปมโดยใช้ระบบการเรียนรู้แบบเบย์ (Bayesian Learning)
- สามารถแก้ไขปรับแต่งวิธีการทดสอบอีเมลต่างๆ ได้ง่าย และสามารถเขียนกฎขึ้นมาป้องกันอีเมลสแปมเองได้ง่าย

- เป็นโปรแกรมกรองอีเมลที่มีชื่อเสียง และได้รับรางวัลต่างๆ มากมาย ยกตัวอย่างเช่น รางวัล Best of Open Source in Security จาก Infoworld [45] รางวัล Product of The Year 2006 จาก Datamation.com [46] รางวัล Best Linux-based Anti-spam Solution ในปี 2006 จาก Linux New Media Awards 2006 [47] ซึ่งเป็นรางวัลที่มาจากการได้รับคะแนนเสียงจากผู้ใช้งานโปรแกรมกรองอีเมลสแปมมาเป็นอันดับหนึ่งมากถึง 69% ทิ้งห่างจากอันดับสองคือ Bogo Filter ที่ได้รับคะแนนเพียง 11% เป็นต้น

ดังนั้นในวิทยานิพนธ์นี้จึงนำโปรแกรมกรองอีเมลสแปมแอสแซสซินมาแก้ไขในส่วนการกรองอีเมลสแปมที่ใช้ระบบการเรียนรู้แบบเบย์ เพื่อเพิ่มประสิทธิภาพการกรองอีเมลสแปมภาษาไทย

โปรแกรมกรองอีเมลสแปมแอสแซสซินสามารถนำมาติดตั้งเพื่อทำงานร่วมกับโปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค (Postfix) ได้ โดยมีลักษณะการเชื่อมต่อกันดังรูปที่ 23



รูปที่ 23 การเชื่อมต่อโปรแกรมกรองอีเมลสแปมแอสแซสซินกับโปรแกรมกรองอีเมลโพสฟิค

เมื่อโปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค (Postfix) ได้รับอีเมลเข้ามาที่พอร์ท 25 จะทำการนำอีเมลนั้นส่งตรวจสอบกับโปรแกรมกรองอีเมลสแปมแอสแซสซินที่พอร์ท 10024 เมื่ออีเมลถูกตรวจสอบเสร็จแล้ว หากอีเมลไม่มีความน่าสงสัยที่จะเป็นอีเมลสแปมก็จะถูกส่งกลับมายังพอร์ท 10025 เพื่อส่งอีเมลให้กับโปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค (Postfix) แต่หากอีเมลมีความน่าสงสัยที่จะเป็นอีเมลสแปม โปรแกรมกรองอีเมลสแปมแอสแซสซินจะทำตามข้อกำหนดที่ได้กำหนด

ไว้ เช่น ลบทิ้งอีเมลออกไป หรือแก้ไขหัวข้อเรื่องเพื่อทำให้รู้ว่าอีเมลมีความน่าสงสัยที่จะเป็นอีเมลสแปม เป็นต้น แล้วจึงส่งกลับมาให้กับโปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค (Postfix) ที่พอร์ท 10025

ติดตั้งโปรแกรมกรองอีเมลสแปมแอสแซสซินให้กับเครื่องแม่ข่ายอีเมลโพสฟิคบนระบบปฏิบัติการอูบันตูมีขั้นตอนดังนี้

1. เมื่อระบบได้ทำการติดตั้งระบบปฏิบัติการอูบันตู โปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิคและโปรแกรมอื่นๆ ที่ช่วยให้โปรแกรมกรองอีเมลสแปมแอสแซสซินทำงานได้แล้ว เช่น ตัวแปลภาษาเพิร์ล เป็นต้น จากนั้นทำการติดตั้งโปรแกรมกรองอีเมลสแปมแอสแซสซินโดยใช้คำสั่งข้างล่างนี้

```
sudo apt-get install spamassasssin
```

2. ทำการปรับแต่งไฟล์คอนฟิคเบื้องต้นเพื่อทำให้ดีมอนสแปมแอสแซสซิน (Spamassassin daemon) ทำงานซึ่งไฟล์คอนฟิคอยู่ที่ไดเรกทอรี /etc/default/spamassassin โดยแก้ไขตามข้างล่างนี้

```
ENABLED=1
```

หลังจากนั้นทำการสั่งให้โปรแกรมกรองอีเมลสแปมแอสแซสซินทำงานโดยคำสั่งข้างล่างนี้

```
sudo etc/init.d/spamassassin start
```

3. ทำการแก้ไขไฟล์คอนฟิคของโปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิคให้ทำงานเชื่อมต่อกับโปรแกรมกรองอีเมลสแปมแอสแซสซินโดยแก้ไขไฟล์คอนฟิคอยู่ที่ไดเรกทอรี /etc/postfix/master.cf

```
content_filter=spamassassin
spamassassin pipe user=spamfilter argv=/usr/bin/spamc -f -e
/usr/sbin/sendmail -oi -f ${sender} ${recipient}
```

5.4 การแก้ไขระบบการเรียนรู้แบบเบย์ของโปรแกรมกรองอีเมลสแปมแอสแซสซินเพื่อเพิ่มประสิทธิภาพการกรองอีเมลสแปมภาษาไทย

เมื่อได้เตรียมระบบเครื่องแม่ข่ายอีเมลและติดตั้งระบบกรองอีเมลที่มีระบบการเรียนรู้แบบเบย์ให้กับเครื่องแม่ข่ายอีเมลเรียบร้อยแล้ว จากนั้นจะทำการแก้ไขระบบการเรียนรู้แบบเบย์ของโปรแกรมกรองอีเมลสแปมแอสแซสซินเพื่อเพิ่มประสิทธิภาพการกรองอีเมลสแปมภาษาไทย โดยแก้ไขไฟล์ Bayes.pm ซึ่งอยู่ในไดเรกทดรี /usr/share/perl5/Mail/SpamAssassin

ไฟล์ Bayes.pm เป็นมอดูลเพิร์ล (Perl Module) ที่ทำการหาค่าความเป็นอีเมลสแปมของอีเมลที่เข้ามายังเครื่องแม่ข่ายอีเมลในโปรแกรมกรองอีเมลสแปมแอสแซสซินโดยใช้วิธีการเรียนรู้

แบบเบย์ซึ่งมีการทำงานตามหลักการของวิธีการเรียนรู้แบบเบย์ซึ่งถูกนำเสนอโดย Paul Graham [29] สามารถแบ่งได้เป็น 2 ส่วนดังนี้

1. ส่วนเรียนรู้ข้อมูลตัวอย่างอีเมลที่ดีและตัวอย่างอีเมลสแปม เริ่มต้นจะทำการวิเคราะห์ข้อความเนื้อหาในตัวอย่างอีเมลทั้งสองแบบ และทำการแบ่งข้อความออกมาเป็นคำ เก็บคำต่างๆ บันทึกลงฐานข้อมูล
2. ส่วนหาค่าความเป็นอีเมลสแปมของอีเมลที่เข้ามาใหม่ โดยทำการวิเคราะห์ข้อความเนื้อหาในอีเมลที่เข้ามาใหม่ฉบับนั้น และทำการแบ่งข้อความออกมาเป็นคำ จากนั้นจึงทำการหาค่าความน่าจะเป็นที่จะเป็นอีเมลสแปมของคำต่างๆ จากฐานข้อมูลที่ได้เก็บข้อมูลจากข้อมูลตัวอย่างทั้งอีเมลที่ดีและอีเมลสแปม เมื่อหาค่าความน่าจะเป็นที่จะเป็นอีเมลสแปมของคำต่างๆ ที่ปรากฏบนอีเมลฉบับนั้นครบแล้ว ก็จะทำการหาค่าความน่าจะเป็นที่จะเป็นอีเมลสแปมของอีเมลฉบับนั้นได้โดยคิดคำนวณจากการรวมค่าความน่าจะเป็นที่จะเป็นอีเมลสแปมของคำที่ปรากฏบนอีเมลฉบับนั้น

ภายใน Bayes.pm นั้นจะมีส่วนของโค้ดต้นฉบับ (Source Code) ของการตัดคำของระบบการเรียนรู้แบบเบย์เมื่อวิเคราะห์ข้อความที่เป็นภาษาในภูมิภาคเอเชีย ดังข้างล่างนี้

```
        if (TOKENIZE_LONG_8BIT_SEQS_AS_TUPLES && $token =~ /[\xa0-\xff]{2}/)
{
        # Matt sez: "Could be asian? Autrijus suggested doing character ngrams,
        # but I'm doing tuples to keep the dbs small(er)."  Sounds like a plan
        # to me! (jm)
        while ($token =~ s/^(..?)//) {
          push (@rettokens, "8:$1");
        }
        next;
}
```

สังเกตจากโค้ดต้นฉบับส่วนการตัดคำของ Bayes.pm ได้ว่า วิธีการตัดคำที่ใช้เมื่อวิเคราะห์ข้อความเนื้อหาอีเมลที่เป็นภาษาในภูมิภาคเอเชีย เป็นแบบเอ็นแกรม (N-Grams) โดยจะทำการตัดคำออกมาเป็นทีละ N ตัวอักษร เช่น N=2 จะตัดคำออกมาทีละ 2 ตัว เป็นตัว ดังตัวอย่างข้างล่างนี้

Subject: คุณเบื่อวันจันทร์ชอบวันศุกร์มั๊ย

[14450] dbg: bayes: 8:คุ

[14450] dbg: bayes: 8:ณเ

[14450] dbg: bayes: 8:บื

[14450] dbg: bayes: 8:่อ

[14450] dbg: bayes: 8:วั

[14450] dbg: bayes: 8:นจ

[14450] dbg: bayes: 8:ัน

[14450] dbg: bayes: 8:ทร

[14450] dbg: bayes: 8:์ช

[14450] dbg: bayes: 8:อบ

[14450] dbg: bayes: 8:วั

[14450] dbg: bayes: 8:นศ

[14450] dbg: bayes: 8:ุก

[14450] dbg: bayes: 8:ร์

[14450] dbg: bayes: 8:มั

[14450] dbg: bayes: 8:๊ย

[14450] dbg: bayes: 8:คุ

[14450] dbg: bayes: 8:ณเ

[14450] dbg: bayes: 8:บื

[14450] dbg: bayes: 8:่อ

[14450] dbg: bayes: 8:วั

[14450] dbg: bayes: 8:นจ

[14450] dbg: bayes: 8:ัน

[14450] dbg: bayes: 8:ทร

[14450] dbg: bayes: 8:์ช

[14450] dbg: bayes: 8:อบ

[14450] dbg: bayes: 8:วั

[14450] dbg: bayes: 8:นศ

[14450] dbg: bayes: 8:ุก

```
[14450] dbg: bayes: 8:ร์
[14450] dbg: bayes: 8:มั
[14450] dbg: bayes: 8:ัย
```

วิธีการตัดคำแบบเอ็นแกรม (N-Grams) ของระบบการเรียนรู้แบบเบย์ในโปรแกรมกรองอีเมลสแปมแอสแซสซินนั้นจะตัดคำในภาษาไทยออกมาเป็นคำที่ไม่มีความหมายในภาษาไทย ทำให้วิเคราะห์เนื้อหาข้อความในอีเมลภาษาไทยได้ไม่มีประสิทธิภาพ

ผู้เสนอวิทยานิพนธ์จึงทำการแก้ไขวิธีการตัดคำจากเดิมที่ใช้วิธีการตัดคำแบบเอ็นแกรม (N-Grams) มาเป็นการใช้โปรแกรมตัดคำไทยคูวส์ (CUWS) ตัดคำแทน ผู้เสนอวิทยานิพนธ์ได้แก้ไข Bayes.pm ให้สามารถเรียกใช้โปรแกรมตัดคำไทยคูวส์ผ่านทางเว็บเซอร์วิส (Web Service) โดยใช้จาวาเอพีไอ (Java API) ดังข้างล่างนี้

```
use Inline (
    Java => 'STUDY',
     CLASSPATH=> $ENV{CLASSPATH},
     #STUDY  =>  ['com-cuws-api','commons-codec-1.3','commons-httpclient-
3.1','commons-logging-1.1.1','log4j-1.2.15'],
     STUDY => ['com.cuws.api.CUWSWebService','Pod_1'],
     AUTOSTUDY => 1
);

#use Java Class , Cutter is used for cutting word
my $cutter = new Mail::SpamAssassin::Bayes::Pod_1() ;
```

เมื่อมีการวิเคราะห์คำในภาษาภูมิภาคเอเชีย จะทำการเรียกใช้โปรแกรมตัดคำไทยคูวส์เพื่อทำการแบ่งคำไทยอย่างมีประสิทธิภาพ ได้คำไทยออกมาเป็นคำที่มีความหมายในภาษาไทย ผู้เสนอวิทยานิพนธ์ได้ทำการแก้ไข Bayes.pm ดังข้างล่างนี้

```
if (TOKENIZE_LONG_8BIT_SEQS_AS_TUPLES && $token =~ /[\xa0-\xff]{2}/)
        {
                $keeptoken = $tis2utf->convert($token);
                $segmentedword = $cutter->cut($keeptoken);

                @separatedword = split('\|', $segmentedword);
                foreach my $buffer (@separatedword) {
                $buffer = $utf2tis->convert($buffer);

                 push (@rettokens, "8:$buffer");
                  }
          next;
          }
```

บทที่ 6
การพัฒนาระบบสร้างกฎด้วยวิธีการทางสถิติ

จากหลักการสร้างกฎด้วยวิธีการทางสถิติในบทที่ 4 ผู้เสนอวิทยานิพนธ์ได้นำมาใช้ในการพัฒนาระบบซึ่งจะกล่าวถึงรายละเอียดการพัฒนาในบทนี้ ซึ่งประกอบด้วย สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา และขั้นตอนในการพัฒนาระบบ

6.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนาระบบสร้างกฎด้วยวิธีการทางสถิติ

การพัฒนาระบบสร้างกฎด้วยวิธีการทางสถิติถูกพัฒนาขึ้นภายใต้สภาพแวดล้อมทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ของเครื่องที่ใช้พัฒนาระบบดังต่อไปนี้

ฮาร์ดแวร์

1. หน่วยประมวลผลกลาง (CPU) อินเทล คอร์ 2 ดูโอ 2.66 กิกะเฮิรตซ์ (Intel Core2Duo 2.66 Ghz)
2. หน่วยความจำ (RAM) 4 กิกะไบต์ (4GB)
3. จานบันทึกข้อมูล (Hard Disk) 160 กิกะไบต์ (160GB)

ซอฟต์แวร์

1. ระบบปฏิบัติการ อูบันตูสำหรับเดสก์ท็อป เวอร์ชัน 8.04 (Ubuntu Destop 8.04)
2. โปรแกรมกรองอีเมลสแปมแอสแซสซิน เวอร์ชัน 3.2.4 (SpamAssassin 3.2.4)
3. ตัวแปลภาษาเพิร์ล (Perl Interpreter)
4. โปรแกรมตัดคำไทยคูวส์ (Chulalongkorn University Word Segmentation: CUWS)
5. โปรแกรมเครื่องแม่ข่ายอีเมลโพสฟิค (Postfix)
6. โปรแกรมอีคลิปซ์ (Eclipse)
7. โปรแกรมเวก้า (Weka)

6.2 การติดตั้งซอฟต์แวร์ในการสร้างเครื่องแม่ข่ายอีเมล (Mail Server)

แสดงในหัวข้อ 5.2

6.3 การติดตั้งซอฟต์แวร์กรองอีเมลให้กับเครื่องแม่ข่ายอีเมล

แสดงในหัวข้อ 5.3

6.4 การติดตั้งโปรแกรมอีคลิปซ์ (Eclipse)

ในการพัฒนาระบบการสร้างกฎด้วยวิธีการทางสถิติ ผู้เสนอวิทยานิพนธ์ใช้โปรแกรมภาษาจาวาในการพัฒนาระบบ เนื่องจากภาษาจาวาเป็นเครื่องมือที่แจกฟรีไม่เสียค่าใช้จ่าย และมีคลาส (Class) ให้เลือกใช้จำนวนมากทำให้ผู้เสนอวิทยานิพนธ์สามารถพัฒนาระบบได้รวดเร็วมากขึ้นโดยการพัฒนาต่อจากคลาสเดิมที่มีอยู่แล้ว

การเขียนโปรแกรมภาษาจาวาจำเป็นต้องมีจาวาแพลตฟอร์ม (Java Platform) ประกอบด้วยจาวาเวอร์ชวลแมชีน (Java Virtual Machine) และไลบรารีมาตรฐานจาวา (Java Standard Libraly) สำหรับการใช้งานโปรแกรมภาษาจาวา

โปรแกรมอีคลิปซ์ (Eclipse) เป็นโปรแกรมที่รวมองค์ประกอบต่างๆ ที่ช่วยเหลือผู้พัฒนาโปรแกรมภาษาจาวา (Java Integrated Development Environment: Java IDE) เพื่อให้เกิดความรวดเร็ว ถูกต้องและแม่นยำในการพัฒนาโปรแกรมภาษาจาวา นอกจากนี้ยังสามารถตรวจสอบระบบที่พัฒนาขึ้นได้

โปรแกรมอีคลิปซ์เป็นซอฟต์แวร์เสรี ภายในประกอบด้วยองค์ประกอบที่สำคัญในการใช้งานโปรแกรมภาษาจาวา ตัวอย่างเช่น โปรแกรมแก้ไขข้อความ (Editor), ตัวแปลภาษาจาวา (Java Compiler), จาวาเวอร์ชวนแมชีน (Java Virtual Machine) เป็นต้น

6.5 โปรแกรมเวก้า (Weka)

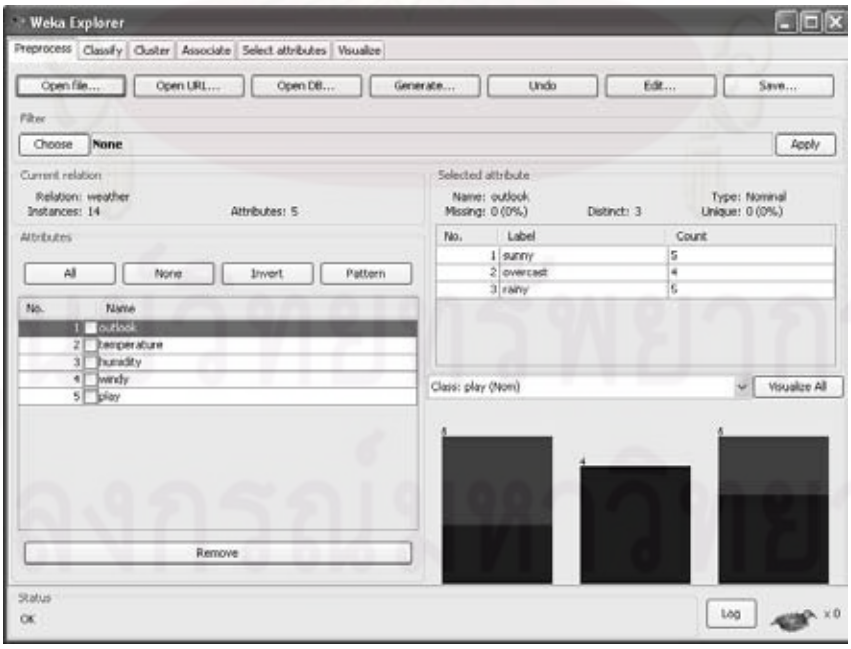
โปรแกรมเวก้าเป็นซอฟต์แวร์เสรีที่รวบรวมขั้นตอนวิธีของการเรียนรู้ของเครื่อง (Machine Learning Algorithms) เพื่อประมวลผลงานทางด้านเหมืองข้อมูล (Data Mining Tasks) วิธีการเรียกใช้งานขั้นตอนวิธีการเรียนรู้ของเครื่องนี้สามารถเรียกใช้งานกับชุดข้อมูล (Data Sets) ได้โดยตรง หรือเรียกใช้งานบนโค้ดต้นฉบับ (Source Code) ในภาษาจาวาก็ได้

ตัวอย่าง weather.arff ซึ่งเป็นชุดข้อมูลที่โปรแกรมเวก้ารับเป็นข้อมูลนำเข้าเพื่อประมวลผล แสดงดังข้างล่างนี้

```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
```

```
@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

ตัวอย่างการเรียกใช้งานขั้นตอนวิธีต่างๆ ในโปรแกรมเวก้ากับชุดข้อมูลโดยตรง ผ่านส่วนต่อประสานผู้ใช้แบบกราฟิก (Graphical User Interface: GUI) แสดงในรูปที่ 24



รูปที่ 24 ส่วนต่อประสานกับผู้ใช้แบบกราฟิกของโปรแกรมเวก้า

6.6 การพัฒนาระบบสร้างกฎด้วยวิธีการทางสถิติ

ในการพัฒนาระบบสร้างกฎด้วยวิธีการทางสถิติจากรูปที่ 14 (แสดงในบทที่ 4) ผู้เสนอวิทยานิพนธ์ได้รวบรวมข้อมูลตัวอย่างอีเมลที่ดีเก็บลงในแฟ้มกล่องจดหมายที่ดี และข้อมูลตัวอย่างอีเมลสแปมเก็บลงในแฟ้มจดหมายขยะ โดยที่อยู่ของแฟ้มกล่องจดหมายที่ดีของเครื่องที่ใช้พัฒนาระบบแสดงดังข้างล่างนี้

/home/tle55/Maildir/.haminbox

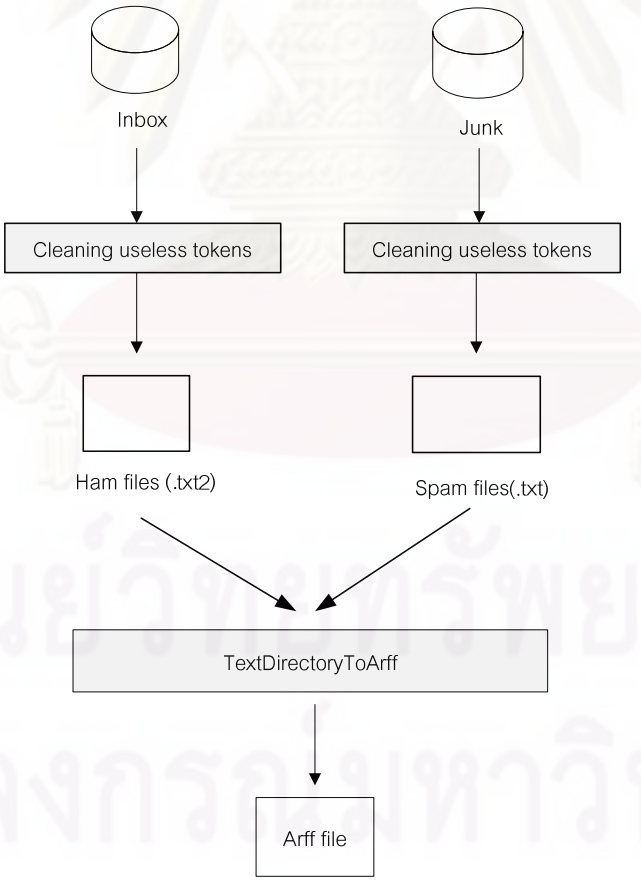
ที่อยู่ของแฟ้มกล่องจดหมายขยะของเครื่องที่ใช้พัฒนาระบบแสดงดังข้างล่างนี้

/home/tle55/Maildir/cur

จากรูปที่ 15 (แสดงในบทที่ 4) แสดงตัวระบบ (Model) ที่ข้อมูลตัวอย่างทั้งสองแบบจะถูกส่งไปประมวลผล โดยภายในตัวระบบมีรายละเอียดในการพัฒนาดังนี้

6.6.1 การพัฒนาระบบในกระบวนการเตรียมข้อมูล (Preprocessing)

กระบวนการเตรียมข้อมูลเป็นกระบวนการแรกในตัวระบบ (Model) ซึ่งรับข้อมูลตัวอย่างอีเมลที่ดีและตัวอย่างอีเมลขยะมาประมวลผล



รูปที่ 25 รายละเอียดภายในกระบวนการเตรียมข้อมูล

จากรูปที่ 25 เริ่มต้นข้อมูลตัวอย่างอีเมลที่ดีและอีเมลสแปมยังไม่เหมาะสมที่จะประมวลผล ข้อมูลทั้งสองแบบจึงถูกส่งไปลบข้อความหรือคำต่างๆ ที่ไม่มีประโยชน์ต่อการประมวลผล เช่น ลบคำสั่งในภาษาเอชทีเอ็มแอล (HTML tags) ออก เป็นต้น

เนื่องจากไฟล์อีเมลที่เก็บอยู่ในแฟ้มกล่องจดหมายที่ดี และแฟ้มกล่องจดหมายขยะจะถูกเข้ารหัสไว้ ผู้เสนอวิทยานิพนธ์จึงได้ใช้คลาส JavaMaildir ในการถอดรหัสอีเมลเพื่ออ่านข้อความในอีเมล

โค้ดต้นฉบับ (Source Code) สำหรับอ่านข้อความในอีเมลส่วนหัวข้อเรื่อง (Subject) แสดงดังข้างล่างนี้

```
Session session = Session.getInstance(new Properties());
String user = "tle55";
String absolute_url = "maildir:/home/" + user + "/Maildir";
String absolute_url2 = "maildir:////home/" + user + "/Maildir";
String relative_url = "maildir:///testhome/Maildir";
String url = absolute_url;
Store store = session.getStore(new URLName(url));
store.connect();
/*spam msg*/
Folder inbox = store.getFolder("inbox");
inbox.open(Folder.READ_WRITE);
int msgtype;
msgtype = 1;
int start;
Writer output = null;
Writer output2 = null;
String pathforwrite;
for ( start = 1 ; start<=inbox.getMessageCount() ; start++ )
{
        Message msg = inbox.getMessage(start);
        String temp_string1=msg.getSubject();
        pathforwrite = "class2/spam"+start+".txt";
```

```
                File file = new File(pathforwrite);
                 output = new BufferedWriter(new FileWriter(file));
                output.write(temp_string1);
                 output.close();
            }
        /* ham msg*/
        Folder haminbox = store.getFolder("haminbox");
        haminbox.open(Folder.READ_WRITE);
        msgtype = 2;
        for ( start = 1 ; start<=haminbox.getMessageCount() ; start++ )
        {
                 Message msgham = haminbox.getMessage(start);
                 pathforwrite = "class2/ham"+start+".txt2";
                 File file2 = new File(pathforwrite);
                 output2 = new BufferedWriter(new FileWriter(file2));
                 String temp_string3=msgham.getSubject();
                output2.write(temp_string3);
                output2.close();
        }
```

ในการอ่านข้อความในอีเมลส่วนเนื้อหา (Content) นั้น จำเป็นต้องใช้คลาส Simpleread เพื่อช่วยในการอ่านข้อความในอีเมลส่วนเนื้อหา โค้ดต้นฉบับ (Source Code) แสดงดังข้างล่างนี้

```
import javax.mail.*;
import java.util.List;
import java.util.Properties;
import java.io.*;
import javax.mail.internet.*;
import java.io.UnsupportedEncodingException;
import java.nio.charset.Charset;
public class simpleread {
```

```
public static void updateTextInMessage(Message msg, int msgnumber,
                int msgtype) throws UnsupportedEncodingException,
MessagingException, IOException {
        Object content = msg.getContent();
        if (content instanceof String) {
                updateTextInMessage((Part) msg);
                Writer output = null;
                String text = content.toString();
                String pathforwrite;
                if (msgtype == 1) {
                        pathforwrite = "class2/spam" + msgnumber + ".txt";
                } else {
                        pathforwrite = "class2/ham" + msgnumber + ".txt2";
                }
                File file = new File(pathforwrite);
                output = new BufferedWriter(new FileWriter(file));
                output.write(text);
                 output.close();

                 FileReader reader = new FileReader(pathforwrite);
                 StringBuffer before_cut_html = new StringBuffer();
                 List<String> lines = HTMLUtils.extractText(reader);

                 for (String line : lines)
                 {
                     before_cut_html.append(line);
                 }

                 Writer output2 = null;
                 output2 = new BufferedWriter(new FileWriter(file));
```

```
                        output2.write(before_cut_html.toString());
                        output2.close();
                } else {
                        updateTextInMessage((Multipart) content, msgnumber,
msgtype, des_subject, des_backup);
                        msg.setContent((Multipart) content);
                }
                msg.saveChanges();
        }

        public static void updateTextInMessage(Multipart multipart, int msgnumber,
                        int msgtype, String des_subject,String des_backup) throws
UnsupportedEncodingException, MessagingException,
                        IOException, ParseException  {

                        /* msgnumber and msgtype are used for writing file */
                        /* msgtype 1 is spam file */
                        /* msgtype 2 is ham file */

                        int partsCount = multipart.getCount();
                        for (int i = 0; i < partsCount; i++) {
                        BodyPart bodyPart = multipart.getBodyPart(i);
                        String disposition=null;
                        disposition = bodyPart.getDisposition();

                        if (disposition == null && bodyPart instanceof MimeBodyPart)
                        {
                                MimeBodyPart mimeBodyPart = (MimeBodyPart)
bodyPart  ;
```

```
if (i == 0) {
System.out.println(mimeBodyPart.getContent());
 Writer output = null;
String text = mimeBodyPart.getContent().toString();
String pathforwrite;

if (msgtype == 1) {
pathforwrite = "class2/spam" + msgnumber + ".txt";
} else {
pathforwrite = "class2/ham" + msgnumber + ".txt2";
}

File file = new File(pathforwrite);
output = new BufferedWriter(new FileWriter(file));
String des_show_type = "Multipart";
String temp_string2;
output.write(text);
output.close();
FileReader reader = new FileReader(pathforwrite);
StringBuffer before_cut_html = new StringBuffer();
List<String> lines = HTMLUtils.extractText(reader);

    for (String line : lines)
    {
       before_cut_html.append(line);
    }
 Writer output2 = null;
output2 = new BufferedWriter(new FileWriter(file));
output2.write(before_cut_html.toString());
output2.close();
```

```
                                }

                                Object content = mimeBodyPart.getContent();
                                if (content instanceof Multipart)
                                {
                                   updateTextInMessage((Multipart) content,
msgnumber,
                                   msgtype, des_subject, des_backup);
                                   mimeBodyPart.setContent((Multipart) content);
                                } else if (mimeBodyPart.isMimeType("text/plain"))
                                        updateTextInMessage((Part) bodyPart);
                        }
                }
        }

        public static void updateTextInMessage(Part textPart)
                        throws MessagingException, IOException {
                String body = (String) textPart.getContent();
                body = "\r\n\r\nAdded text.\r\n";
                textPart.setContent(body, "text/plain");
        }

        public static void main(String args[]) throws Exception {
                /*Test*/
                Session session = Session.getInstance(new Properties());
                String user = "tle55";
                String absolute_url = "maildir:/home/" + user + "/Maildir";
                String absolute_url2 = "maildir:////home/" + user + "/Maildir";
                String relative_url = "maildir:///testhome/Maildir";
                String url = absolute_url;
```

```
                Store store = session.getStore(new URLName(url));
                store.connect();
                Folder inbox = store.getFolder("inbox");
                inbox.open(Folder.READ_WRITE);
                int msgtype;
                msgtype = 1; /* spam msg */
                int start;
                for (start = 1; start <= inbox.getMessageCount(); start++) {
                        Message msg = inbox.getMessage(start);
                        updateTextInMessage(msg, start, msgtype);
                }

                /* Writing ham file */
                Folder haminbox = store.getFolder("haminbox");
                haminbox.open(Folder.READ_WRITE);
                msgtype = 2;
                for (start = 1; start <= haminbox.getMessageCount(); start++) {
                        Message msgham = haminbox.getMessage(start);
                        updateTextInMessage(msgham, start, msgtype);
                }
        }
}
```

โค้ดต้นฉบับ (Source Code) สำหรับอ่านข้อความในอีเมลส่วนเนื้อหา (Content) แสดงดังข้างล่างนี้

```
Session session = Session.getInstance(new Properties());
String user = "tle55";
String absolute_url = "maildir:/home/" + user + "/Maildir";
String absolute_url2 = "maildir:////home/" + user + "/Maildir";
```

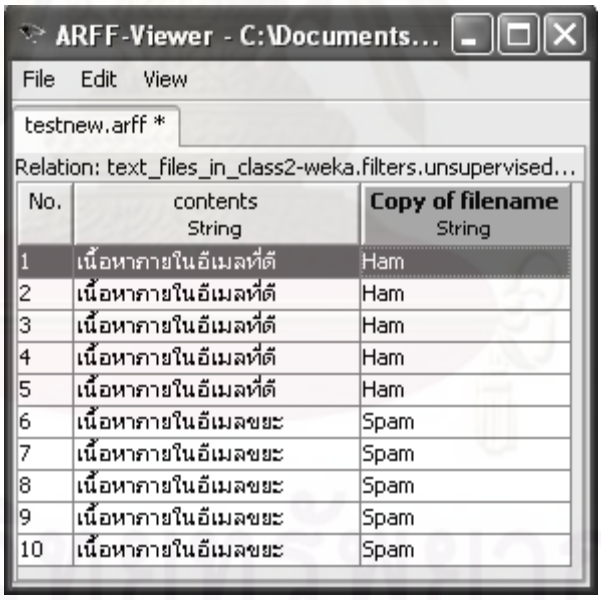
```
String relative_url = "maildir:///testhome/Maildir";
String url = absolute_url;
Store store = session.getStore(new URLName(url));
store.connect();
Folder inbox = store.getFolder("inbox");
inbox.open(Folder.READ_WRITE);
int msgtype;
msgtype = 1; /* spam msg */
int start;
simpleread simread = new simpleread();

for ( start = 1 ; start<=inbox.getMessageCount() ; start++ )
{
    Message msg = inbox.getMessage(start);
    simread.updateTextInMessage(msg, start, msgtype);
}

Folder haminbox = store.getFolder("haminbox");
haminbox.open(Folder.READ_WRITE);
msgtype = 2; /* ham msg */
for ( start = 1 ; start<=haminbox.getMessageCount() ; start++ )
{
    Message msgham = haminbox.getMessage(start);
    simread.updateTextInMessage( msgham , start , msgtype);
}
```

การทำงานของโค้ดต้นฉบับสำหรับอ่านข้อความในส่วนหัวข้อเรื่องของอีเมลและส่วนเนื้อหาของอีเมล เมื่อได้ถอดรหัสอีเมลและทำให้ข้อความของอีเมลในส่วนหัวข้อและเนื้อหามีความเหมาะสมที่จะประมวลผลแล้ว จากนั้นจะทำการเขียนข้อความของอีเมลแต่ละฉบับออกมาเป็นไฟล์ข้อความ (Text file) แต่ละไฟล์ซึ่งตัวอย่างอีเมลที่ดีจะเขียนเป็นไฟล์นามสกุล txt2 และตัวอย่างอีเมลสแปมจะเขียนเป็นไฟล์นามสกุล txt

เมื่อเขียนไฟล์ข้อความของอีเมลที่ดีและอีเมลสแปมแต่ละฉบับลงเรียบร้อยแล้ว จากนั้นจะทำการแปลงไฟล์ข้อความของอีเมลทั้งสองแบบเป็นไฟล์ Arff ซึ่งเป็นข้อมูลนำเข้า (Input Data) ที่โปรแกรมเวก้าสามารถนำไปประมวลผลทางสถิติได้ โค้ดต้นฉบับสำหรับแปลงไฟล์ข้อความเป็นไฟล์ Arff แสดงดังข้างล่างนี้

```
String name = “class2”; //folder that stores text file
TextDirectoryToArff tdta = new TextDirectoryToArff();
Instances dataset = tdta.createDataset(name);
//write dataset to arff file
ArffSaver saver_before = new ArffSaver();
saver_before.setInstances(dataset);
saver_before.setFile(new File("./data/testnew.arff"));
saver_before.writeBatch();
```

ตัวอย่างข้อมูลภายในไฟล์ Arff ที่ได้ เมื่อสร้างมาจากตัวอย่างอีเมลที่ดี 5 ฉบับและตัวอย่างอีเมลสแปม 5 ฉบับ แสดงดังรูปที่ 26



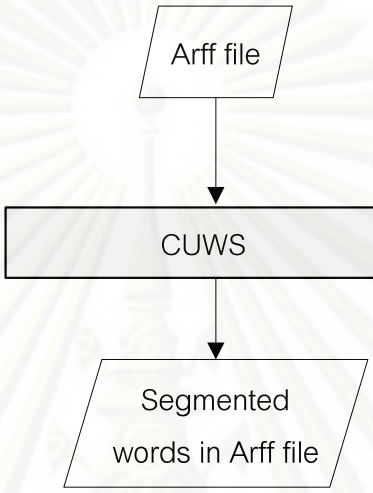
No.	contents String	Copy of filename String
1	เนื้อหาภายในอีเมลที่ดี	Ham
2	เนื้อหาภายในอีเมลที่ดี	Ham
3	เนื้อหาภายในอีเมลที่ดี	Ham
4	เนื้อหาภายในอีเมลที่ดี	Ham
5	เนื้อหาภายในอีเมลที่ดี	Ham
6	เนื้อหาภายในอีเมลขยะ	Spam
7	เนื้อหาภายในอีเมลขยะ	Spam
8	เนื้อหาภายในอีเมลขยะ	Spam
9	เนื้อหาภายในอีเมลขยะ	Spam
10	เนื้อหาภายในอีเมลขยะ	Spam

รูปที่ 26 ตัวอย่างข้อมูลภายในไฟล์ Arff ที่สร้างมาจากชุดข้อมูลตัวอย่าง

เมื่อสร้างไฟล์ Arff เรียบร้อยแล้ว จากนั้นก็จะเข้าสู่กระบวนการตัดคำ (Tokenizing) ต่อไป

6.6.2 การพัฒนาระบบในกระบวนการตัดคำ (Tokenizing)

กระบวนการตัดคำจะนำข้อมูลตัวอย่างอีเมลที่ดีและข้อมูลตัวอย่างอีเมลสแปมซึ่งอยู่ในรูปแบบไฟล์ Arff มาประมวลผลคำเพื่อให้ได้คำหรือวลีที่สำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปม (Spam-liked Pattern) ได้



รูปที่ 27 รายละเอียดภายในกระบวนการตัดคำ

จากรูปที่ 27 ข้อมูลตัวอย่างอีเมลที่ดีและตัวอย่างอีเมลสแปมที่อยู่ในรูปแบบไฟล์ Arff ซึ่งสร้างจากกระบวนการก่อนหน้านั้น จะถูกนำมาเป็นข้อมูลนำเข้าในกระบวนการนี้ โดยไฟล์ Arff จะถูกส่งเข้าไปตัดคำโดยโปรแกรมตัดคำไทยคูวส์ (CUWS) เมื่อได้ผ่านการตัดคำแล้ว ข้อมูลที่ได้คือไฟล์ Arff ที่ถูกตัดคำเรียบร้อยแล้ว

เนื่องจากการประมวลผลภาษาไทยจำเป็นต้องใช้โปรแกรมตัดคำ (Word Segmentation) เพื่อช่วยในการประมวลผลภาษาไทย ผู้เสนอวิทยานิพนธ์ได้ใช้โปรแกรมตัดคำไทยคูวส์ (CUWS) สำหรับการตัดคำไทย

เมื่อข้อมูลตัวอย่างอีเมลที่ดีและตัวอย่างอีเมลสแปมซึ่งอยู่ในรูปแบบ Arff ได้ผ่านการตัดคำเรียบร้อยแล้ว ก็จะนำข้อมูลที่ถูกตัดคำแล้วมาผ่านตัวกรอง (Filters) ต่างๆ เพื่อปรับรูปแบบให้เหมาะสมต่อการประมวลผลในกระบวนการต่อไป

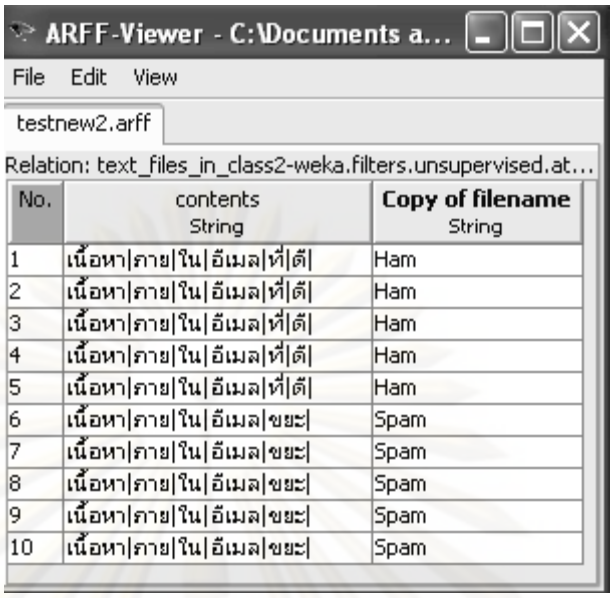
โค้ดต้นฉบับสำหรับกระบวนการตัดคำแสดงดังข้างล่างนี้

```
//dataset is Arff file
m_Training = dataset;
```

```
        String key = "3fb5e00c155c83f9"; // register to get the key //
        CUWSWebService cuws = new CUWSWebService(key);
        String before_change;
        String after_change;
        ThaiUtils change_encode;
        // Using StringToNominal for 2nd attribute
       m_Filter0 = (Filter)
Class.forName("weka.filters.unsupervised.attribute.StringToNominal").newInstance();
        String[] options_fil0={"-R","2"};
        if (m_Filter0 instanceof OptionHandler)
        ((OptionHandler) m_Filter0).setOptions(options_fil0);
        m_Filter0.setInputFormat(m_Training);
            m_Training = Filter.useFilter(m_Training, m_Filter0);
int temp_instances=0;
         for (temp_instances=0 ; temp_instances < m_Training.numInstances() ;
temp_instances++)
        {
                before_change =
m_Training.instance(temp_instances).stringValue(0);
                after_change = ThaiUtils.ASCII2Unicode(before_change);
                m_Training.instance(temp_instances).setValue(0,
cuws.cut(after_change) );
        }
   // Using StringToWordVector for content attribute
   m_Filter1 = (Filter)
Class.forName("weka.filters.unsupervised.attribute.StringToWordVector").newInstance();
   String[] options_fil1={"-R","first-last","-W","1000","-prune-rate","-1.0","-N","0","-
stemmer","weka.core.stemmers.NullStemmer","-M","1","-
tokenizer","weka.core.tokenizers.WordTokenizer -delimiters \" \\r \\t.,;:\\\'\\\"()?!|\""};
   if (m_Filter1 instanceof OptionHandler)
```

```
      ((OptionHandler) m_Filter1).setOptions(options_fil1);
    m_Filter1.setInputFormat(m_Training);
    m_Training = Filter.useFilter(m_Training, m_Filter1);
     // Using Copy
     m_Filter2 = (Filter)
Class.forName("weka.filters.unsupervised.attribute.Copy").newInstance();
     String[] options_fil2={"-R","1"};
     if (m_Filter2 instanceof OptionHandler)
       ((OptionHandler) m_Filter2).setOptions(options_fil2);
       m_Filter2.setInputFormat(m_Training);
      m_Training = Filter.useFilter(m_Training, m_Filter2);
      // Using Remove
      m_Filter3 = (Filter)
Class.forName("weka.filters.unsupervised.attribute.Remove").newInstance();
     String[] options_fil3={"-R","1"};
     if (m_Filter3 instanceof OptionHandler)
      ((OptionHandler) m_Filter3).setOptions(options_fil3);
      m_Filter3.setInputFormat(m_Training);
      m_Training = Filter.useFilter(m_Training, m_Filter3);
      m_Training.setClassIndex(m_Training.numAttributes() - 1);
     ArffSaver saver = new ArffSaver();
     saver.setInstances(m_Training);
     saver.setFile(new File("./data/test.arff"));
     saver.writeBatch();
```

ข้อมูลตัวอย่างอีเมลที่ดีและอีเมลสแปมภายในไฟล์ Arff ที่ได้ผ่านการตัดคำแล้ว (แต่ยังไม่ได้ผ่านตัวกรอง) แสดงในรูปที่ 28


ARFF-Viewer - C:\Documents a...

File Edit View

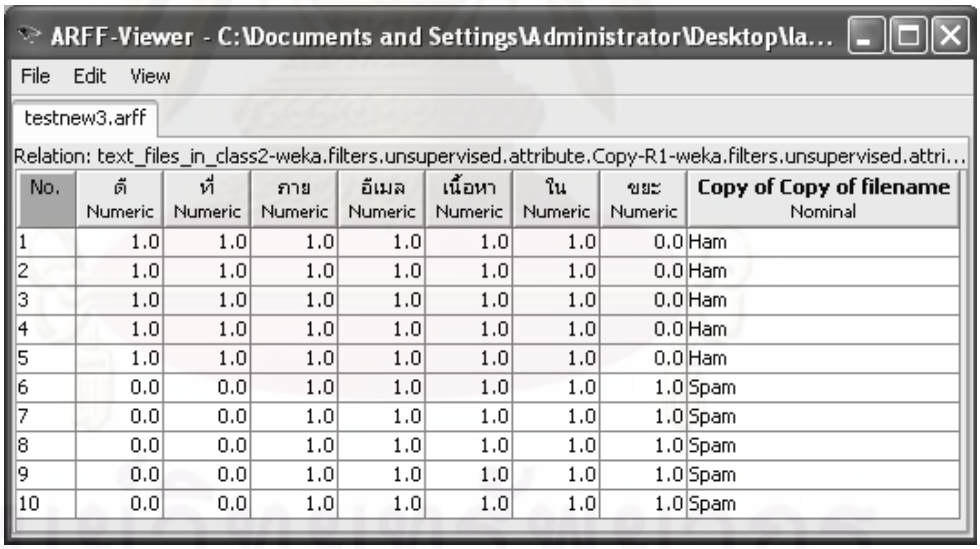
testnew2.arff

Relation: text_files_in_class2-weka.filters.unsupervised.at...

No.	contents String	Copy of filename String
1	เนื้อหา|ภาย|ใน|อีเมล|ที่|ดี|	Ham
2	เนื้อหา|ภาย|ใน|อีเมล|ที่|ดี|	Ham
3	เนื้อหา|ภาย|ใน|อีเมล|ที่|ดี|	Ham
4	เนื้อหา|ภาย|ใน|อีเมล|ที่|ดี|	Ham
5	เนื้อหา|ภาย|ใน|อีเมล|ที่|ดี|	Ham
6	เนื้อหา|ภาย|ใน|อีเมล|ขยะ|	Spam
7	เนื้อหา|ภาย|ใน|อีเมล|ขยะ|	Spam
8	เนื้อหา|ภาย|ใน|อีเมล|ขยะ|	Spam
9	เนื้อหา|ภาย|ใน|อีเมล|ขยะ|	Spam
10	เนื้อหา|ภาย|ใน|อีเมล|ขยะ|	Spam

รูปที่ 28 ตัวอย่างข้อมูลอีเมลที่ดีและอีเมลสแปมภายในไฟล์ Arff ที่ได้ผ่านการตัดคำแล้วแต่ยังไม่ได้ผ่านตัวกรองต่างๆ

ข้อมูลตัวอย่างอีเมลที่ดีและอีเมลสแปมภายในไฟล์ Arff ที่ได้ผ่านการตัดคำแล้ว และผ่านตัวกรองต่างๆ แสดงในรูปที่ 29


ARFF-Viewer - C:\Documents and Settings\Administrator\Desktop\la...

File Edit View

testnew3.arff

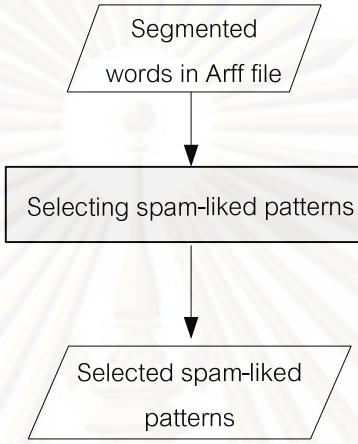
Relation: text_files_in_class2-weka.filters.unsupervised.attribute.Copy-R1-weka.filters.unsupervised.attri...

No.	ดี Numeric	ที่ Numeric	ภาย Numeric	อีเมล Numeric	เนื้อหา Numeric	ใน Numeric	ขยะ Numeric	Copy of Copy of filename Nominal
1	1.0	1.0	1.0	1.0	1.0	1.0	0.0	Ham
2	1.0	1.0	1.0	1.0	1.0	1.0	0.0	Ham
3	1.0	1.0	1.0	1.0	1.0	1.0	0.0	Ham
4	1.0	1.0	1.0	1.0	1.0	1.0	0.0	Ham
5	1.0	1.0	1.0	1.0	1.0	1.0	0.0	Ham
6	0.0	0.0	1.0	1.0	1.0	1.0	1.0	Spam
7	0.0	0.0	1.0	1.0	1.0	1.0	1.0	Spam
8	0.0	0.0	1.0	1.0	1.0	1.0	1.0	Spam
9	0.0	0.0	1.0	1.0	1.0	1.0	1.0	Spam
10	0.0	0.0	1.0	1.0	1.0	1.0	1.0	Spam

รูปที่ 29 ตัวอย่างข้อมูลอีเมลที่ดีและอีเมลสแปมภายในไฟล์ Arff ที่ได้ผ่านการตัดคำแล้วและผ่านตัวกรองต่างๆ

6.6.3 การพัฒนาระบบในกระบวนการคัดเลือกคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปม (Selecting spam-liked pattern)

ในกระบวนการคัดเลือกคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปมนั้น จะนำข้อมูลซึ่งอยู่ในรูปแบบไฟล์ Arff ที่ได้จากกระบวนการก่อนหน้ามาประมวลผลต่อในกระบวนการนี้



รูปที่ 30 รายละเอียดภายในกระบวนการคัดเลือกคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปม

จากรูปที่ 30 ข้อมูลตัวอย่างอีเมลที่ดีและตัวอย่างอีเมลสแปมที่อยู่ในรูปแบบไฟล์ Arff และถูกตัดคำเรียบร้อยแล้ว จะถูกส่งเข้าไปค้นหาคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปมได้ และข้อมูลที่ได้คือคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปมซึ่งจะนำมาสร้างเป็นกฎต่อไป

จากหัวข้อที่ 4.2.3 ในกระบวนการนี้จะทำการคำนวณความสัมพันธ์ระหว่างคำหรือตัวแปร t กับความเป็นอีเมลที่ดี (V_{th}) และความสัมพันธ์ระหว่างตัวแปร t กับความเป็นอีเมลสแปม (V_{ts}) และทำการหาค่าอัตราส่วน R_t ดังสมการ (3)

วิธีการคัดเลือกคำหรือวลีที่สำคัญที่สามารถบ่งบอกได้ว่าเป็นอีเมลสแปม คือ วิธีการ Conditional Probability and Bayes' Theorem ดังสมการ (6), (7) และมีวิธีการคำนวนหาค่าต่างๆ ดังสมการ (14), (15), (16), (17), (18)

โค้ดต้นฉบับสำหรับกระบวนการคัดเลือกคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปมแสดงดังข้างล่างนี้

```
name = "test.arff";
 m_TrainingFile = name;
 m_Training     = new Instances( new BufferedReader(new
```

```
FileReader(m_TrainingFile) ) );
m_Training.setClassIndex(m_Training.numAttributes() - 1);
int temp=0;
int temp2;
int[] containpattern = new int[2];
int[] notcontainpattern = new int[2];
float sum_A_B_C_D=0;
float P_E=0;
float P_E_Bar=0;
float P_H=0;
float P_E_H=0;
float P_E_Bar_H=0;
float Vts= 0;
float Vth= 0;
float[][] Result = new float[5][m_Training.numAttributes()];
String[] Attribute_word = new String[m_Training.numAttributes()];
// A
containpattern[0]=0;
// B
containpattern[1]=0;
// C
notcontainpattern[0]=0;
// D
notcontainpattern[1]=0;
for (temp=0 ; temp < m_Training.numAttributes() ; temp++)
{
        // Loop for all instances
        for ( temp2=0 ; temp2 < m_Training.numInstances() ; temp2++ )
        {
       if (  m_Training.instance(temp2).value(temp) == 0 )
```

```
        {
                if (
m_Training.instance(temp2).stringValue(m_Training.classIndex()).equals("A") )
                {
                        notcontainpattern[0]++;
                }
                else if(
m_Training.instance(temp2).stringValue(m_Training.classIndex()).equals("B") )
                {
                        notcontainpattern[1]++;
                }
        }
        else
        {
           if (
m_Training.instance(temp2).stringValue(m_Training.classIndex()).equals("A") )
                  {
                  containpattern[0]++;
                  }
                  else if(
m_Training.instance(temp2).stringValue(m_Training.classIndex()).equals("B") )
                  {
                        containpattern[1]++;
                  }
        }
}
// Calculate A , B , C , D
sum_A_B_C_D=containpattern[0]+containpattern[1]+notcontainpattern[0]+notc
ontainpattern[1];
P_E=(containpattern[0]+notcontainpattern[0])/sum_A_B_C_D;
```

```
P_E_Bar=(containpattern[1]+notcontainpattern[1])/sum_A_B_C_D;
P_H=(containpattern[0]+containpattern[1]) / sum_A_B_C_D;
P_E_H=(containpattern[0]) / sum_A_B_C_D;
P_E_Bar_H=(containpattern[1]) / sum_A_B_C_D;
/* Keep Attribute Word */
Attribute_word[temp] = m_Training.attribute(temp).name();
 Vts=0;
 Vth=0;
 Vts= (P_E_H) / P_H;
 Vth= (P_E_Bar_H) / P_H;
Result[1][temp]=(Vts)/(Vth);
//Clear Every Value for each attribute
containpattern[0]=0;
containpattern[1]=0;
notcontainpattern[0]=0;
notcontainpattern[1]=0;
sum_A_B_C_D=0;
P_E=0;
P_E_Bar=0;
P_H=0;
P_E_H=0;
P_E_Bar_H=0;
Vth=0;
Vts=0;
}
```

เมื่อได้คำที่สามารถบ่งบอกความเป็นอีเมลสแปมซึ่งสามารถนำมาสร้างเป็นกฎได้แล้ว ก็จะทำการเขียนลงบนไฟล์ข้อความ (Text file) ตามรูปแบบของกฎสำหรับโปรแกรมสแปมแอสแซสซิน โค้ดต้นฉบับสำหรับเขียนไฟล์ข้อความเพื่อเป็นกฎสำหรับส่วนหัวข้อของอีเมล (Subject rules) และกฎสำหรับส่วนเนื้อหาของอีเมล (Body rules) ดังข้างล่างนี้

```
WekaDemo.bubbleSort1( Result[1] , Attribute_word );
Writer output2 = null;
String text2;
String text3;
String text4;
String pathwrite2 = "df/newrule.cf";
int number_flag = 1;
int trim_flag = 1;
String temp_buffer;
File file2 = new File(pathwrite2);
output2 = new BufferedWriter(new FileWriter(file2));
System.out.println("-----------------------------------------------");
for ( temp = m_Training.numAttributes()-1 ; temp>0 ; temp-- )
{
        temp_buffer = Attribute_word[temp];
        temp_buffer = temp_buffer.trim();
        int looptoken;
        StringBuffer buffer_cut_slash = new StringBuffer() ;
        for ( looptoken = 0 ; looptoken<Attribute_word[temp].length() ; looptoken++ )
        {
            if ( Attribute_word[temp].charAt(looptoken) == '/' )
            {
                buffer_cut_slash.append('\\');
            }
            buffer_cut_slash.append(Attribute_word[temp].charAt(looptoken));
        }
        Attribute_word[temp] = buffer_cut_slash.toString();
       //Subject
       text2 = "\nheader "+"TH_SUBJECT_"+temp+" Subject =~
"+"/"+Attribute_word[temp]+"/\n";
```

```
        text3 = "describe "+"TH_SUBJECT_"+temp+" "+"Subject contains
'"+Attribute_word[temp]+"'\n";
        //Body
         text2 = "\nbody "+"TH_BODY_"+temp+" "+"/"+Attribute_word[temp]+"/\n";
         text3 = "describe "+"TH_BODY_"+temp+" "+"Body contains
'"+Attribute_word[temp]+"'\n";
        output2.write(text2);
        output2.write(text3);
}
```

ตัวอย่างกฎสำหรับส่วนหัวข้อเรื่อง (Subject rule) ที่ระบบได้สร้างขึ้นมา แสดงดังข้างล่างนี้

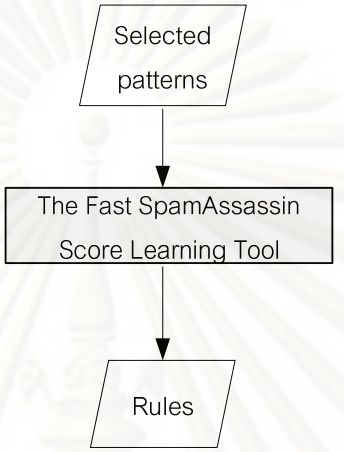
```
Subject TH_subject_1 /คลิก/
describe TH_subject_1 subject contains ‘คลิก’
```

ตัวอย่างกฎสำหรับส่วนเนื้อหา (Content rule) ที่ระบบได้สร้างขึ้นมา แสดงดังข้างล่างนี้

```
body TH_body_1 /เงิน/
describe TH_body_1 body contains ‘เงิน’
```

6.6.4 **การพัฒนาระบบในกระบวนการกำหนดคะแนนให้กับกฎ** (Scoring)

เมื่อได้คัดเลือกคำที่สามารถบ่งบอกความเป็นอีเมลสแปมซึ่งสามารถนำมาสร้างเป็นกฎแล้ว ในกระบวนการนี้จะทำการกำหนดคะแนนให้กับคำสำคัญแต่ละคำ



รูปที่ 31 รายละเอียดภายในกระบวนการกำหนดคะแนนให้กับกฎ

จากรูปที่ 31 คำที่สามารถบ่งบอกความเป็นอีเมลสแปมได้นั้นจะถูกส่งเข้าสู่กระบวนการกำหนดคะแนน และข้อมูลที่ได้คือกฎพร้อมทั้งคะแนนความเป็นอีเมลสแปม

ผู้เสนอวิทยานิพนธ์ใช้โปรแกรมสำหรับการให้คะแนนกับกฎของโปรแกรมสแปมแอสแซสซินซึ่งชื่อว่า The Fast SpamAssassin Score Learning Tool ซึ่งใช้วิธี Artificial Neural Network

หลักการทำงานของโปรแกรม The Fast SpamAssassin Score Learning Tool คือ ผู้ใช้ต้องกำหนดชุดข้อมูลตัวอย่างอีเมลที่ดี และตัวอย่างอีเมลสแปมให้กับกฎที่จะนำมาให้คะแนน

ผู้เสนอวิทยานิพนธ์ได้ใช้ตัวอย่างอีเมลที่ดีและตัวอย่างอีเมลสแปมซึ่งเป็นตัวอย่างที่ได้ใช้ในการหาคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลมาเป็นชุดข้อมูลตัวอย่างในการให้คะแนนกับกฎ

คำสั่งที่สำคัญในโปรแกรม The Fast SpamAssassin Score Learning Tool นั้น มีดังนี้

mass-check คือ คำสั่งตรวจสอบข้อมูลตัวอย่างกับกฎต่างๆ โดยจะเขียนผลลัพธ์ออกมาในไฟล์ข้อความ คือ spam.log และ ham.log เป็นผลสรุปของแต่ละอีเมลว่ามีกฎข้อใดบ้างที่สามารถตรวจสอบอีเมลฉบับนั้นได้ ดังตัวอย่างข้างล่างนี้

```
1 /path/to/mbox:<5.1.0.14.2.20011004073932.05f4fd28@localhost>
TRACKER_ID,BALANCE_FOR_LONG
```

logs-to-c คือ คำสั่งที่แปลง spam.log และ ham.log เป็นโค้ดต้นฉบับภาษาซี (C source files) เพื่อทำให้สามารถป้อนเป็นข้อมูลนำเข้าให้กับคำสั่ง Perceptron ซึ่งเป็นคำสั่งที่จะทำการประมวลผลต่อไป

ขั้นตอนในการสั่งให้โปรแกรม The Fast SpamAssassin Score Learning Tool ทำงาน แสดงดังข้างล่างนี้

```
./mass-check --progress ham:dir:/home/tle55/Maildir/.haminbox/cur/
spam:dir:/home/tle55/Maildir/cur/
./logs-to-c --count --cffile=../rules
make
./perceptron
```

ผลลัพธ์ที่ได้คือคะแนนของกฎต่างๆ แสดงดังข้างล่างนี้

```
score    TH_subject_1    1.080
```

บทที่ 7
การทดสอบประสิทธิภาพของระบบ

ในบทนี้จะทำการทดสอบประสิทธิภาพของระบบที่ได้พัฒนาขึ้น คือ ระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์สำหรับภาษาไทย และระบบการสร้างกฎด้วยวิธีการทางสถิติ

7.1 การทดสอบประสิทธิภาพของระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์สำหรับภาษาไทย

ในการทดสอบประสิทธิภาพระบบกรองอีเมลสแปมที่มีระบบการเรียนรู้แบบเบย์สำหรับภาษาไทยและใช้โปรแกรมตัดคำไทยคูวส์จะใช้เครื่องที่ใช้ในการพัฒนาระบบเป็นเครื่องที่ใช้ในการทดสอบระบบ

ข้อมูลที่ใช้ในการทดสอบนั้นผู้เสนอวิทยานิพนธ์ได้ทำการเก็บรวบรวมเองจากกล่องจดหมายของผู้เสนอวิทยานิพนธ์ โดยผู้เสนอวิทยานิพนธ์ได้กระจายชื่ออีเมลของผู้เสนอวิทยานิพนธ์ในเว็บไซต์สาธารณะต่างๆ รวมทั้งเวปบอร์ดที่สามารถกระจายชื่ออีเมลออกไปเพื่อทำให้ผู้ส่งอีเมลสแปมได้เห็นชื่ออีเมลของผู้เสนอวิทยานิพนธ์และส่งอีเมลสแปมมาให้ นอกจากนี้ยังรวบรวมข้อมูลจากเครื่องแม่ข่ายอีเมลของภาควิชาวิศวกรรมคอมพิวเตอร์และเครื่องแม่ข่ายอีเมลของคณะวิศวกรรมศาสตร์อีกด้วย ผู้เสนอวิทยานิพนธ์ได้คัดแยกเฉพาะอีเมลทั้งสองประเภทที่เป็นภาษาไทยเท่านั้น

ในการทดลองข้อมูลถูกแยกออกเป็นข้อมูลฝึกและข้อมูลทดสอบในจำนวนที่เท่ากัน จำนวนตัวอย่างที่ใช้ คือ อีเมลสแปมจำนวน 500 ฉบับ อีเมลที่ดีจำนวน 500 ฉบับ

ผู้เสนอวิทยานิพนธ์ได้ทำการทดสอบเปรียบเทียบประสิทธิภาพระหว่างระบบกรองอีเมลแบบเบย์ธรรมดากับระบบกรองอีเมลแบบเบย์ที่มีการใช้โปรแกรมตัดคำไทยคูวส์

เมื่อระบบได้คำนวณความน่าจะเป็นของอีเมลจากหลักการของเบย์แล้วจะแปลงความน่าจะเป็นไปเป็นคะแนนความเป็นอีเมลสแปมโดยพิจารณาเป็นช่วงจากค่าที่กำหนดไว้ล่วงหน้าดังแสดงในตารางที่ 9

ตารางที่ 9 การแปลงค่าความน่าจะเป็นที่จะเป็นอีเมลสแปมของอีเมลเป็นคะแนนความเป็นอีเมลสแปมของอีเมล

ช่วงความน่าจะเป็นที่จะเป็นสแปม	คะแนนความเป็นที่อีเมลสแปม
0.00 – 0.01	-2.6
0.01 – 0.05	-1.1
0.05 – 0.20	-0.7
0.20 – 0.40	-0.2
0.40 – 0.60	0.0
0.60 – 0.80	1.0
0.80 – 0.95	2
0.95 – 0.99	3
0.99 - 1	3.5

จากตารางที่ 9 หากค่าความน่าจะเป็นอยู่ระหว่าง 0.00-0.01 คิดเป็น -2.6 คะแนน เป็นต้น เกณฑ์ที่ใช้วัดประสิทธิภาพคือ ค่าผลต่างของคะแนนเฉลี่ยความเป็นอีเมลสแปมและอีเมลที่ดีที่ได้มาจากระบบการเรียนรู้แบบเบย์

ตารางที่ 10 ผลต่างของคะแนนเฉลี่ยของทั้งสองระบบ

ชุดข้อมูล	Bayes	Bayes with CUWS
ชุดที่ 1	3.555	3.615
ชุดที่ 2	3.760	4.026
ชุดที่ 3	3.786	3.976
ชุดที่ 4	4.189	4.257

*ผลต่างมากคือประสิทธิภาพสูง

จากผลการทดลองในตารางที่ 10 พบว่าระบบกรองอีเมลแบบเบย์ที่มีการใช้โปรแกรมตัดคำไทยคูวส์ให้ผลต่างระหว่างอีเมลที่ดีและอีเมลสแปมมากกว่าเดิม โดยโปรแกรมสแปมแอสแซสซินจะให้คะแนนที่ได้จากกรณีของสแปมมีแนวโน้มสูงขึ้นและให้คะแนนการคัดกรองที่ดีต่ำกว่าระบบกรองอีเมลแบบเบย์ธรรมดา

7.2 การทดสอบประสิทธิภาพของระบบการสร้างกฎด้วยวิธีการทางสถิติ

ในการทดสอบประสิทธิภาพระบบการสร้างกฎด้วยวิธีการทางสถิติจะใช้เครื่องที่ใช้ในการพัฒนาระบบเป็นเครื่องที่ใช้ในการทดสอบระบบ

ข้อมูลที่ใช้ในการทดสอบนั้นผู้เสนอวิทยานิพนธ์ได้ทำการเก็บรวบรวมจากกล่องจดหมายของผู้เสนอวิทยานิพนธ์ จากเครื่องแม่ข่ายอีเมลของภาควิชาวิศวกรรมคอมพิวเตอร์ และจากเครื่องแม่ข่ายอีเมลของคณะวิศวกรรมศาสตร์จำนวน 370,000 ฉบับ ผู้เสนอวิทยานิพนธ์ได้คัดแยกเฉพาะอีเมลทั้งสองประเภทที่เป็นภาษาไทยเท่านั้น เป็นอีเมลที่ดีภาษาไทย 1,000 ฉบับ เป็นอีเมลสแปมภาษาไทย 1,000 ฉบับ

ข้อมูลที่ใช้ทดสอบถูกแยกออกเป็นข้อมูลฝึกเพื่อใช้สำหรับการสร้างกฎ และข้อมูลทดสอบเพื่อใช้สำหรับการทดสอบประสิทธิภาพของกฎ ผู้เสนอวิทยานิพนธ์ได้ตรวจสอบผลด้วยวิธีการ 10-fold Cross Validation

ตัวอย่างการทดสอบคือใช้ข้อมูลฝึกที่เป็นอีเมลที่ดีภาษาไทยจำนวน 900 ฉบับและเป็นอีเมลสแปมภาษาไทยจำนวน 900 ฉบับสำหรับการสร้างกฎ และใช้ข้อมูลทดสอบที่เป็นอีเมลที่ดีจำนวน 100 ฉบับและเป็นอีเมลสแปมจำนวน 100 ฉบับสำหรับการทดสอบประสิทธิภาพของกฎ

ในการทดสอบจะใช้วิธีการ Conditional Probability (CP) ในการหาคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปมและนำมาสร้างเป็นกฎ ซึ่งกฎที่นำมาทดสอบจะใช้กฎในส่วนหัวข้อจำนวน 100 กฎและกฎในส่วนของเนื้อหาจำนวน 100 กฎ และเกณฑ์ของคะแนนความเป็นอีเมลสแปมเท่ากับ 5

การทดลองที่ 1 ทำการเปรียบเทียบผลรวมคะแนนความเป็นอีเมลสแปมเฉลี่ยระหว่างระบบกรองอีเมลสแปมแอสแซสซินแบบปริยาย (Default SpamAssassin) กับระบบกรองอีเมลสแปมแอสแซสซินที่ได้ติดตั้งกฎภาษาไทยที่ได้สร้างขึ้น (SpamAssassin with Thai rules)

จุดประสงค์ของการทดลองที่ 1 นี้เพื่อพิจารณาว่ากฎภาษาไทยที่ได้สร้างขึ้นนั้นส่งผลกระทบอย่างไรต่อระบบกรองอีเมลสแปมแอสแซสซินแบบปริยาย

ตารางที่ 11 การเปรียบเทียบผลรวมของคะแนนความเป็นอีเมลสแปมเฉลี่ยระหว่างระบบกรองอีเมลสแปมแอสแซสซินแบบปริยาย (Default SpamAssassin) และระบบกรองอีเมลสแปมแอสแซสซินที่ได้ติดตั้งกฎภาษาไทยที่สร้างขึ้น (SpamAssassin with Thai rules)

ชุดข้อมูล	Default SpamAssassin		SpamAssassin with Thai rules	
	Ham	Spam	Ham	Spam
1	0.17	2.21	0.79	10.56
2	0.40	2.85	0.91	12.97
3	0.41	2.81	0.79	12.75
4	0.46	2.18	1.31	11.19
5	0.44	2.80	1.01	14.06
6	0.91	2.36	0.83	12.09
7	0.14	2.2	0.50	9.34
8	0.18	3.02	0.67	11.63
9	0.17	2.25	1.33	13.98
10	0.19	2.52	0.90	13.38

จากตารางที่ 11 แสดงให้เห็นว่าระบบกรองอีเมลสแปมแอสแซสซินที่ได้ติดตั้งกฎภาษาไทยที่ได้สร้างขึ้นนั้นสามารถเพิ่มผลรวมคะแนนความเป็นอีเมลสแปมได้ในทุกชุดข้อมูล ทำให้ระบบสามารถกรองอีเมลสแปมภาษาไทยได้มีประสิทธิภาพมากขึ้น

การทดลองที่ 2 ทำการเปรียบเทียบร้อยละความถูกต้องการคัดกรองอีเมลสแปม (Spam Recall) และร้อยละความผิดพลาดของการคัดกรองอีเมลที่ดี (Ham Error) ระหว่างระบบกรองเมลสแปมแบบปริยาย (Default SpamAssassin) กับระบบกรองเมลสแปมแอสแซสซินที่ได้ติดตั้งกฎภาษาไทยที่สร้างขึ้น (SpamAssassin with Thai rules)

ค่าต่างๆ ที่สำคัญในการคำนวณเกณฑ์สำหรับการประเมินประสิทธิภาพระบบกรองอีเมลทั่วไปซึ่งเป็นที่รู้จัก มีรายละเอียดดังนี้

True Positive (TP) คือ จำนวนของอีเมลสแปมซึ่งถูกจำแนกว่าเป็นอีเมลสแปม

True Negative (TN) คือ จำนวนของอีเมลที่ดีซึ่งถูกจำแนกว่าเป็นอีเมลที่ดี

False Positive (FP) คือ จำนวนของอีเมลที่ดีซึ่งถูกจำแนกว่าเป็นอีเมลสแปม

False Negative (FN) คือ จำนวนของอีเมลสแปมซึ่งถูกจำแนกว่าเป็นอีเมลที่ดี

เกณฑ์ที่ใช้สำหรับประเมินประสิทธิภาพของระบบโดยทั่วไป คือ Spam Recall, Spam Error, Ham Recall และ Ham Error มีรายละเอียดดังนี้

Spam Recall หรือ Sensitivity คือ ค่าความถูกต้องของการคัดกรองอีเมลสแปม ซึ่งสามารถคำนวณได้ดังสมการที่ 21

$$\text{Spam Recall} = \frac{TP}{(TP+FN)} \tag{21}$$

Spam Error หรือ Type II Error คือ ค่าความผิดพลาดของการคัดกรองอีเมลสแปม ซึ่งสามารถคำนวณได้ดังสมการที่ 22

$$\text{Spam Error} = \frac{FN}{(FN+TP)} \tag{22}$$

Ham Recall หรือ Specificity คือ ค่าความถูกต้องของการคัดกรองที่ดี ซึ่งสามารถคำนวณได้ดังสมการที่ 23

$$\text{Ham Recall} = \frac{TN}{(TN+FP)} \tag{23}$$

Ham Error หรือ Type I Error คือ ค่าความผิดพลาดของการคัดกรองอีเมลที่ดี ซึ่งสามารถคำนวณได้ดังสมการที่ 24

$$\text{Ham Error} = \frac{FP}{(FP+TN)} \tag{24}$$

จุดประสงค์ของการทดลองที่ 2 นี้เพื่อทดสอบความถูกต้องในการคัดกรองเมลของกฎภาษาไทยที่สร้างขึ้นมา

เกณฑ์ที่ใช้สำหรับประเมินประสิทธิภาพของระบบในการทดลองที่ 2 คือ Spam Recall และ Ham Error

ตารางที่ 12 เปรียบเทียบร้อยละความถูกต้องของการคัดกรองอีเมลสแปม (Spam recall rate percentages) และร้อยละความผิดพลาดของการคัดกรองอีเมลที่ดี (Ham error rate percentages) ระหว่างระบบกรองอีเมลสแปมแอสแซสซินแบบปริยาย (Default SpamAssassin) กับระบบกรองอีเมลสแปมแอสแซสซินที่ได้ติดตั้งกฎภาษาไทยที่สร้างขึ้น (SpamAssassin with Thai rules)

ชุดข้อมูล	Default SpamAssassin		SpamAssassin with Thai rules	
	Ham Error	Spam Recall	Ham Error	Spam Recall
1	0%	17.6%	3%	80.8%
2	0%	20%	2%	83.2%
3	0%	22.4%	3%	82.4%
4	1%	14.4%	5%	82.4%
5	0%	25.6%	2%	86.4%
6	0%	26.4%	2%	77.6%
7	0%	12.8%	0%	76.8%
8	0%	25.6%	1%	81.6%
9	0%	16.8%	2%	84%
10	0%	20.8%	5%	80%

จากตารางที่ 12 แสดงให้เห็นว่าระบบกรองเมลสแปมแอสแซสซินที่ได้ติดตั้งกฎภาษาไทยที่สร้างขึ้นมีค่าความถูกต้องของการคัดกรองอีเมลสแปมที่สูงขึ้นในทุกชุดข้อมูล กฎภาษาไทยที่สร้างขึ้นนี้ทำให้ระบบกรองเมลสแปมแอสแซสซินกรองอีเมลสแปมได้แม่นยำมากขึ้น

บทที่ 8
บทสรุปงานวิจัย

8.1 สิ่งที่ได้จากการวิจัย (Contribution)

สิ่งที่ได้จากงานวิจัย

1. อธิบายความหมายอีเมลสแปม ลักษณะอีเมลสแปม และปัญหาอีเมลสแปม
2. การรวบรวมและแบ่งแยกรูปแบบวิธีการแก้ปัญหาอีเมลสแปมที่มีอยู่ในปัจจุบัน โดยนำเสนอหลักการวิธีการแก้ไขปัญหารูปแบบต่างๆ ข้อดีและข้อเสียของแต่ละรูปแบบ และตัวอย่างของซอร์ฟแวร์ที่เป็นการแก้ไขปัญหารูปแบบต่างๆ
3. นำเสนอหลักการวิธีการปรับปรุงประสิทธิภาพการแก้ไขปัญหาอีเมลสแปมแบบใช้ระบบการเรียนรู้แบบเบย์ (Bayesian Learning) สำหรับภาษาไทย ซึ่งได้นำโปรแกรมตัดคำไทยมาช่วยในการประมวลผลคำไทยให้กับระบบการเรียนรู้แบบเบย์ทั่วไป ทำให้ได้ระบบกรองเมลที่มีระบบการเรียนรู้แบบเบย์ที่มีประสิทธิภาพเพิ่มมากขึ้น
4. นำเสนอหลักการวิธีการปรับปรุงประสิทธิภาพระบบกรองอีเมลสแปมด้วยวิธีการทางสถิติ ซึ่งเป็นการนำข้อดีของการแก้ปัญหาอีเมลสแปมแบบการสร้างกฎและข้อดีของวิธีการแก้ปัญหาแบบระบบการเรียนรู้เข้าด้วยกัน ทำให้ได้กฎภาษาไทยที่สามารถนำไปใช้ร่วมกันระหว่างเครื่องแม่ข่ายอีเมล และสามารถรับมือกับรูปแบบของอีเมลสแปมที่หลากหลายได้ ทำให้ระบบกรองเมลสแปมแอสแซสซินสามารถกรองอีเมลสแปมภาษาไทยได้มีประสิทธิภาพมากขึ้น

8.2 ประโยชน์ของการสร้างกฎด้วยวิธีการทางสถิติ

การสร้างกฎด้วยวิธีการทางสถิติเป็นวิธีการสร้างกฎที่ทำให้ได้กฎที่สามารถนำมาใช้ร่วมกันระหว่างเครื่องแม่ข่ายอีเมลได้ และกฎที่ได้สามารถรับมือกับรูปแบบของอีเมลสแปมที่หลากหลายได้ เมื่อนำกฎที่สร้างขึ้นไปติดตั้งบนระบบกรองอีเมลก็ทำให้ได้ระบบกรองอีเมลที่มีประสิทธิภาพ

8.3 แนวทางการวิจัยต่อ

1. วิธีการสร้างกฎด้วยวิธีการทางสถิติจะทำการสร้างกฎออกมา ซึ่งภายในกฎแต่ละกฎจะประกอบด้วยคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปม โดยวิธีการที่นำเสนอนี้คำสำคัญมีลักษณะเป็นคำสั้น ในงานวิจัยในอนาคตจะพัฒนาวิธีการคัดเลือกคำสำคัญที่สามารถบ่งบอกความเป็นอีเมลสแปมที่สามารถคัดเลือกคำที่มีลักษณะเป็นคำยาว ซึ่งจะทำให้สามารถลดค่า False Positive ที่เกิดขึ้นได้

2. ในงานวิจัยในอนาคตจะทำการพิจารณาจำนวนกฎที่เหมาะสมที่ทำให้ระบบกรองอีเมลสแปมมีค่าความถูกต้องของการคัดกรองอีเมลสูงสุด และค่าความผิดพลาดของการคัดกรองอีเมลที่ดีต่ำสุด

รายการอ้างอิง

[1] PC World. 90 Percent of Email Is Spam [Online]. Available from: www.pcworld.com/article/165533/90_percent_of_email_is_spam_symantec_says.html, [2009, September 15]

[2] Government Information Technology Services. Mail Cleaner Statistics – 2008 [Online]. Available from: http://www.inet.co.th/event/endpoint_songkhla/09-11-5INET-hadyai-2-c.pdf, [2009, December 20]

[3] Internet FAQ Archives. RFC2821 – Simple Mail Transfer Protocol [Online]. Available from: http://www.faqs.org/rfcs/rfc2821.html, [2009, December 20]

[4] The Internet Engineering Task Force(IETF). Post Office Protocol [Online]. Available from: http://www.ietf.org/rfc/rfc1939.txt, [2009, December 20]

[5] Internet FAQ Archives. RFC3501 – Internet Message Access Protocol [Online]. Available from: http://www.faqs.org/rfcs/rfc3501.html, [2009, December 20]

[6] Wikipedia. E-mail spam – Wikipedia [Online]. Available from: http://en.wikipedia.org/wiki/Spam_mail, [2009, December 20]

[7] Adam Back. Hashcash – A Denial of Service Counter Measure [Online]. Available from: http://www.hashcash.org/papers/hashcash.pdf, [2009, December 20]

[8] Cynthia Dwork, Moni Naor. Pricing via Processing or Combatting Junk Mail, Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, pp.139-147. London, United Kingdom : Springer-Verlag, 1992.

[9] Sourceforge. Camram antispam system [Online]. Available from: http://sourceforge.net/projects/camram, [2009, September 20]

[10] Federal Law in USA. CAN-SPAM Act of 2003. USA, 2005.

[11] BBC News. Man gets nine years for spamming [Online]. Available from : http://news.bbc.co.uk/2/hi/americas/4426949.stm, [2009, September 20]

[12] Spam Laws. Anti-Spam Laws and the European Union [Online]. Available from : http://www.spamlaws.com/eu.shtml, [2009, September 20]

[13] Sender Policy Framework. SPF: Project Overview [Online]. Available from : http://www.openspf.org, [2009, September 20]

[14] TREND MICRO. MAPS – Stopping Spam at its Source [Online], Available from : http://www.mail-abuse.com, [2009, September 25]

[15] Spamhaus. The Spamhaus Project [Online], Available from : http://www.spamhaus.org, [2009, September 25]

[16] Microsoft. Caller ID for E-mail: The Next Step to Deterring Spam. Microsoft Corporation, White Paper. USA, 2004.

[17] DKIM. DomainKeys Identified Mail (DKIM) [Online]. Available from : http://www.dkim.org, [2009, September 25]

[18] Rhyolite Software. Distributed Checksum Clearinghouses [Online]. Available from : http://www.rhyolite.com/dcc, [2009, September 20]

[19] Sourceforge, Vipul's Razor [Online], Available from : http://razor.sourceforge.net, [2009, September 20]

[20] Sourceforge, Pyzor [Online], Available from : http://sourceforge.net/apps/trac/pyzor, [2009, September 30]

[21] Apache SpamAssassin Project, SpamAssassin : Welcome to SpamAssassin [Online], Available from : http://spamassassin.apache.org, [2009, September 30]

[22] The DSPAM Project, DSPAM Project Homepage [Online], Available from : http://www.nuclearelephant.com/index.php, [2009, September 20]

[23] Sourceforge, SpamBayes: Bayesian anti-spam classifier written in Python [Online], Available from : http://spambayes.sourceforge.net, [2010, February 1]

[24] Sourceforge, Welcome to SpamProbe – A Fast Bayesian Spam Filter [Online], Available from : http://spamprobe.sourceforge.net, [2010, February 1]

[25] Alan Schwartz. SpamAssassin. CA : O'Reilly, 2004.

[26] SA Rules Emporium, SARE – SpamAssassin Rules Emporium [Online], Available from : http://www.rulesemporium.com, [2010, February 1]

[27] Tran Quang Anh, Haixin Duan, Xing Li. Real-time statistical rules for spam detection, International Journal of Computer Science and Network Security 6 (February 2006) : 178-184.

[28] Nguyen Tuan Anh, Tran Quang Anh, Nguyen Ngoc Binh. Vietnamese Spam Detection based on Language Classification, Proceedings of the 2nd International Conference on Communications and Electronics, pp.74-79. Hanoi, Vietnam : IEEE Computer Society, 2008.

[29] Paul Graham. A plan for spam [Online], Available from : http://www.paulgraham.com/spam.html, [2010, February 1]

[30] Patrick Pantel, Dekang Lin. SpamCop – A Spam Classification & Organization Program, Proceedings of AAAI-98 Workshop on Learning for Text Categorization, pp.95-98. Wisconsin, USA : AAAI Press, 1998.

[31] Mehran Sahami, Susan Dumais, David Heckerman, Eric Horvitz. A Bayesian Approach to Filtering Junk E-mail, Proceedings of AAAI-98 Workshop on Learning for Text Categorization, pp.55-62. Wisconsin, USA : AAAI Press, 1998.

[32] Vit Niennattrakul, Pairote Leelaphattarakij, Jirat Srisawat. CUWS: Thai word segmentation software [Online], Available from : http://oracle.cp.eng.chula.ac.th/me/cuws, [2010, February 1]

[33] Sun Microsystems, BreakIterator (Java Platform SE 6) [Online], Available from : http://java.sun.com/javase/6/docs/api/java/text/BreakIterator.html, [2010, April 7]

[34] IBM, ICU Homepage [Online], Available from : http://site.icu-project.org, [2010, April 7]

[35] NECTEC, SWATH [Online], Available from : http://www.hlt.nectec.or.th/products/swath.php, [2010, April 7]

[36] Hui, Cttex [Online], Available from : http://vuthi.blogspot.com/2004/07/cttex.html, [2010, April 7]

[37] NECTEC, BEST | Bring out your BEST [Online], Available from : http://thailang.nectec.or.th/best, [2010, April 7]

[38] Androutsopoulos I, Koutsias J, Chandrinos KV, Spyropoulos CD. An Evaluation of Naive Bayesian Anti-Spam Filtering, Proceedings of the 11th European Conference on Machine Learning, pp.9-17. Barcelona, Spain : Springer, 2000.

[39] Androutsopoulos I, Koutsias J, Chandrinos KV, Spyropoulos CD. An Experimental Comparison of Naive Bayesian and Keyword-based Anti-Spam Filtering Personal E-mail Messages, Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.160-167. New York, USA : ACM Press, 2000.

[40] Androutsopoulos I, Paliouras G, Karkaletsis V, Sakkis G, Spyropoulos C, Stamatopoulos P. Learning to Filter Spam E-mail: A Comparison of a Naive Bayesian and a Memory-based Approach, Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp.1-13. Lyon, France : Springer, 2000.

[41] Chan J, Koprinska I, Poon J. Co-training on textual documents with a single natural feature set, Proceedings of the 9th Australasian Document Computing Symposium, pp.586-589, Melbourne, Australia : IEEE Computer Society, 2004.

[42] Paul Graham. Better Bayesian Filtering [Online], Available from : http://www.paulgraham.com/better.html, [2010, April 7]

[43] Chih-Chin Lai, Ming-Chi Tsai. An Empirical Performance Comparison of Machine Learning Methods for Spam E-mail Categorization, Proceedings of the 4th International Conference on Hybrid Intelligent Systems, pp.44-48, Kitakyushu, Japan : IEEE Computer Society, 2004.

[44] Henry Stern. Fast SpamAssassin Score Learning Tool [Online], Available from : http://cpansearch.perl.org/src/FELICITY/Mail-SpamAssassin-3.0.6/masses/README.perceptron, [2009, September 20]

[45] Infoworld, Best of open source in security – Infoworld [Online], Available from : http://www.infoworld.com/d/security-central/best-open-source-in-security-122?page=0,1, [2010, April 7]

[46] Datamation.com, Datamation.com Announces Product of the Year Winners [Online], Available from : http://itmanagement.earthweb.com/article.php/3586381, [2010, April 7]

[47] Linux New Media, Linux New Media Awards 2006 [Online], Available from : http://www.linux-magazine.com/w3/issue/74/Linux_New_Media_Awards_2006.pdf, [2010, April 7]

ภาคผนวก

ภาคผนวก
ความรู้พื้นฐานที่ใช้

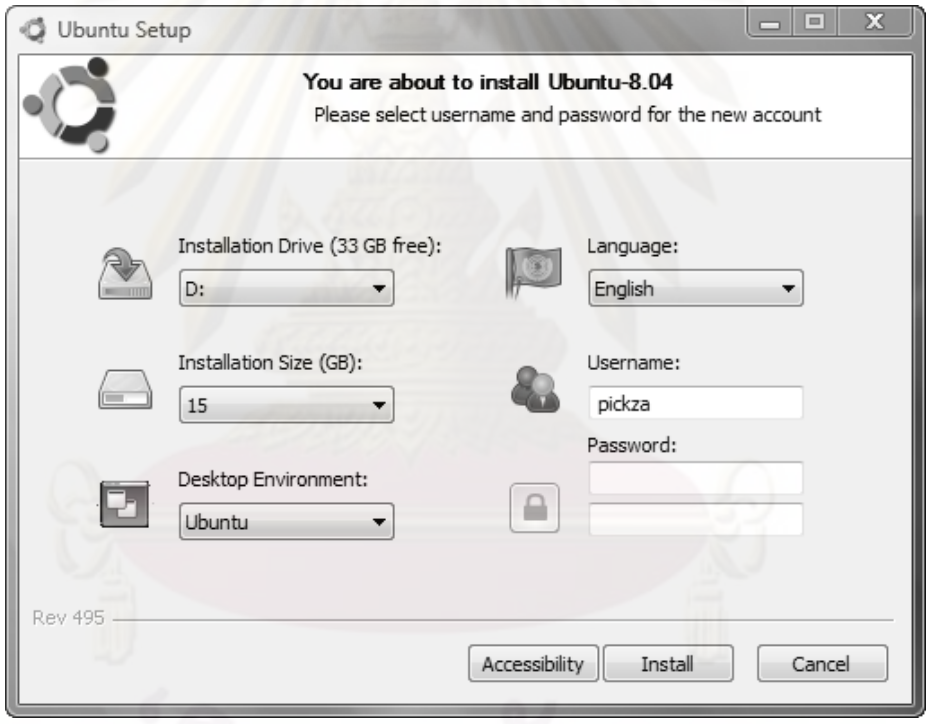
การติดตั้งระบบปฏิบัติการ

เตรียมความพร้อมก่อนทำการติดตั้งระบบปฏิบัติการ Ubuntu

1. โปรแกรม Ubuntu Desktop 8.10 (Download จาก http://www.ubuntu.com/getubuntu/download)
2. พื้นที่ว่างสำหรับการติดตั้ง ไม่ควรน้อยกว่า 10 GB

การติดตั้งระบบปฏิบัติการ

1. ให้ Double click ที่ wuxi.exe (ตัวติดตั้งระบบปฏิบัติการ Ubuntu) จะแสดงหน้าต่าง ดังรูปที่ 23



รูปที่ 32 การติดตั้งระบบปฏิบัติการ

2. Installation Drive : ให้เลือกไดรฟ์ที่ต้องการติดตั้งระบบปฏิบัติการ Ubuntu
   Installation Size : เลือกติดตั้งที่ 15 GB
   Desktop Environment : ให้เลือกเป็นของ Ubuntu
   Language : ให้เลือกระบบการติดตั้ง English
   Username : เป็นการตั้งชื่อ Login ของระบบปฏิบัติการ Ubuntu
   Password : เป็นรหัสที่ใช้สำหรับเข้าระบบปฏิบัติการ Ubuntu

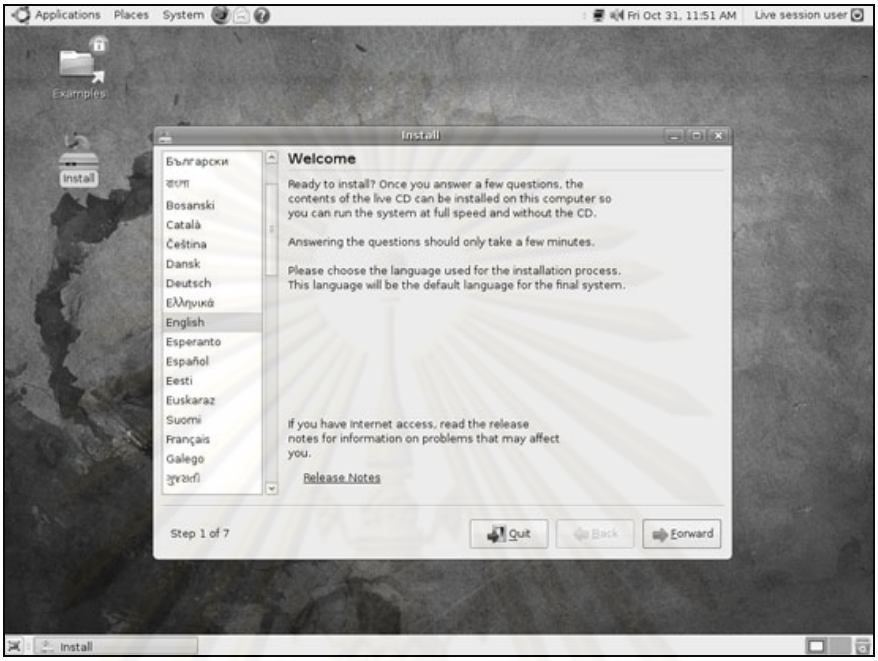
เมื่อใส่ครบทุกขั้นตอนแล้วให้ click ที่ปุ่ม Install ระบบก็จะทำการติดตั้งระบบต่างๆ ภายในเครื่อง ใช้เวลาประมาณ 1 ชั่วโมง หลังจากนั้นเครื่องจะทำการ restart เครื่องใหม่

รูปที่ 33 การเข้าสู่ระบบปฏิบัติการ Ubuntu

3. เมื่อเข้าสู่หน้าต่างของระบบปฏิบัติการ Ubuntu ให้ double click ที่สัญรูป install

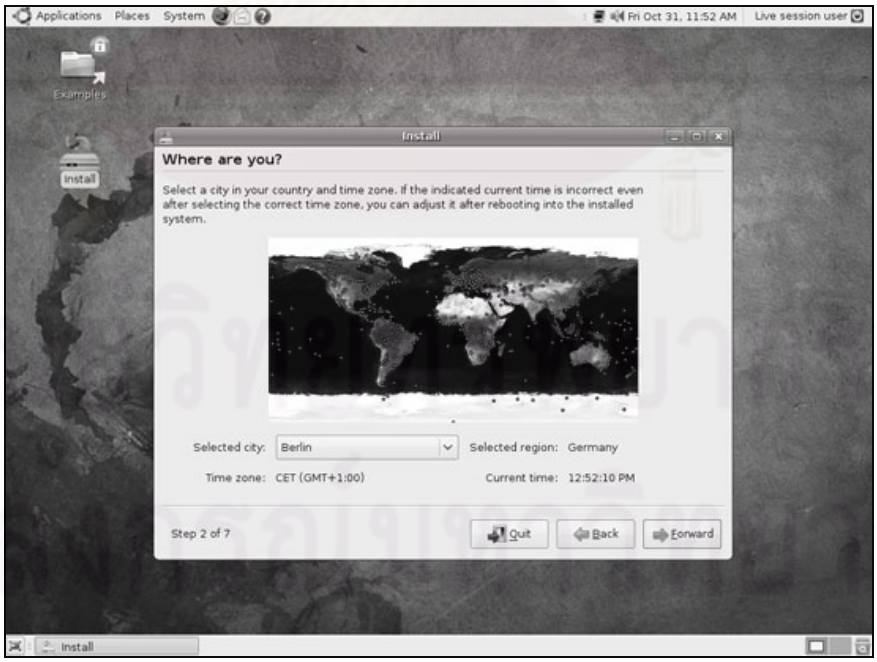
รูปที่ 34 การติดตั้งระบบปฏิบัติการ Ubuntu ลงบนฮาร์ดไดร์ฟ

4. แสดงหน้าจอให้เลือกภาษาในการติดตั้งระบบ เลือกเป็นภาษาอังกฤษ (English) แล้วทำการ click ที่ปุ่ม Forward เพื่อทำรายการต่อไป



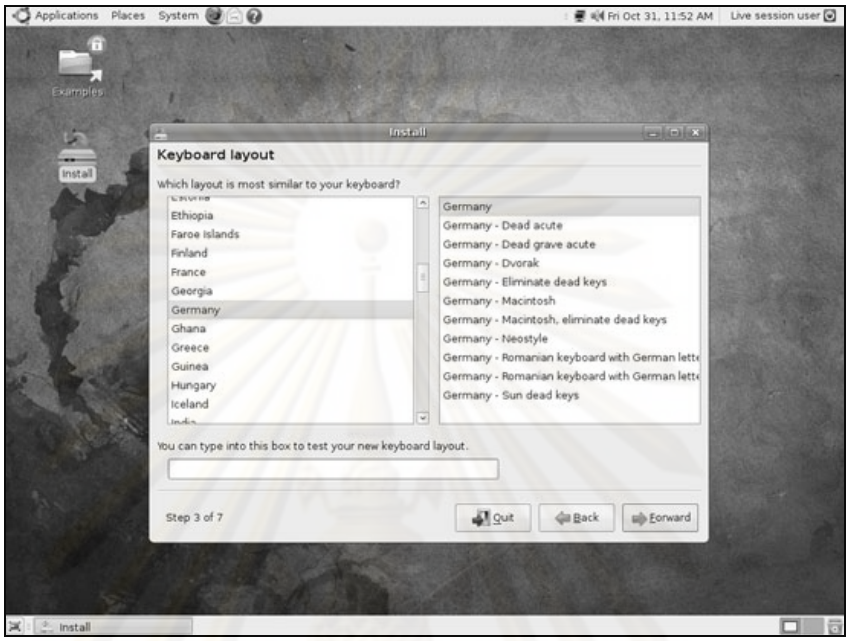
รูปที่ 35 เลือกภาษาในการติดตั้งระบบ

5. ให้ทำการเลือกเมืองที่อยู่ และเวลาของเมือง ณ ขณะที่ทำการติดตั้งระบบ แล้ว click ที่ปุ่ม Forward เพื่อทำรายการต่อไป



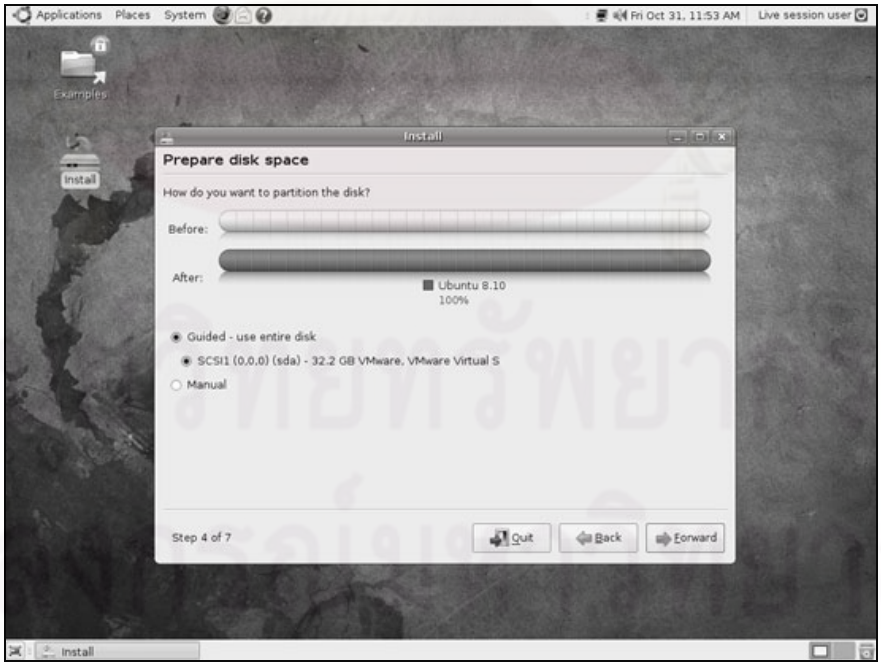
รูปที่ 36 ตั้งค่าเมืองและเวลาให้กับระบบ

6. ทำการเลือกภาษาให้ keyboard ที่ต้องการใช้โดยสามารถทดลองพิมพ์ข้อความเพื่อตรวจสอบความถูกต้อง จากนั้น click ที่ปุ่ม Forward เพื่อทำรายการต่อไป



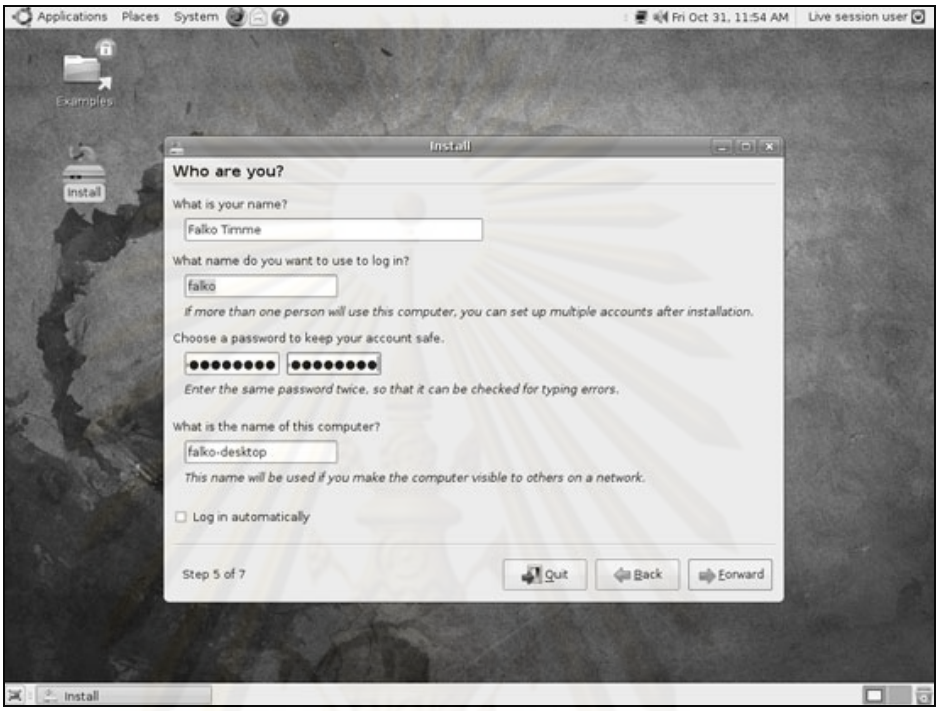
รูปที่ 37 ตั้งค่าภาษาให้กับระบบ

7. ทำการแบ่ง partition ให้กับ hard disk ในที่นี้เลือกเป็นแบบ Guided เพื่อให้ระบบทำการแบ่งการติดตั้งให้โดยอัตโนมัติ



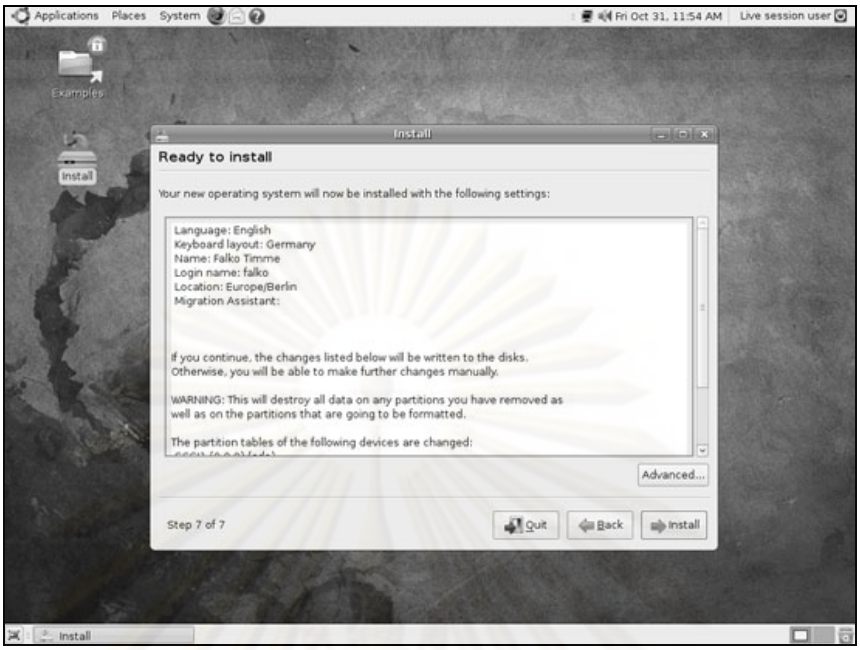
รูปที่ 38 การแบ่ง Partition ให้กับระบบ

8. ทำการใส่ชื่อ-นามสกุลของผู้ใช้ให้กับระบบ จากนั้นให้ใส่ Login name และ Password เพื่อใช้สำหรับกรอกข้อมูลยืนยันการใช้งานก่อนเข้าใช้ระบบปฏิบัติการ Ubuntu และสุดท้ายให้ใส่ชื่อเครื่องคอมพิวเตอร์ด้วย

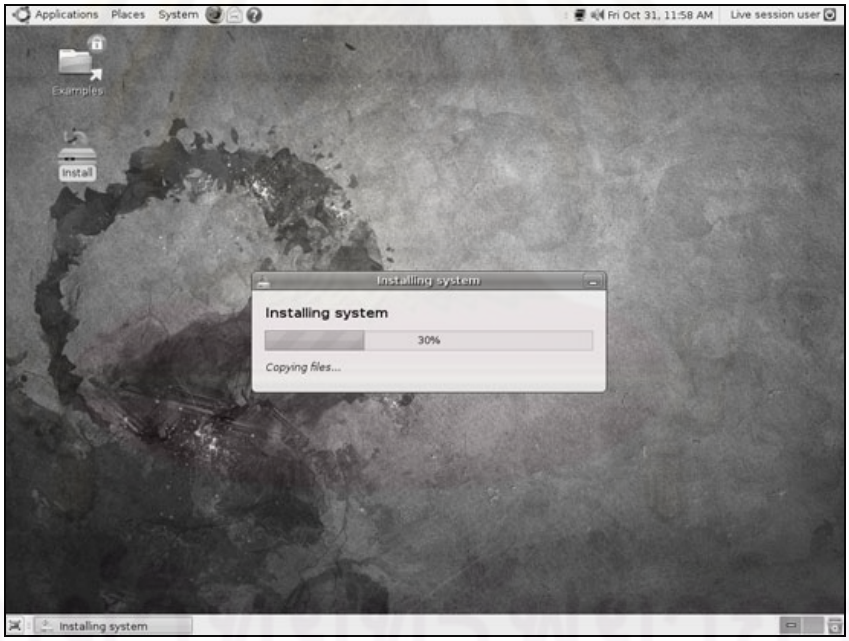


รูปที่ 39 ตั้งค่าข้อมูลเพื่อให้สำหรับการยืนยันเข้าใช้ระบบ

9. แสดงรายละเอียดข้อมูลการติดตั้งระบบทั้งหมด ถ้ามีข้อมูลที่ต้องการแก้ไขสามารถ click ที่ปุ่ม Back กลับไปเพื่อทำการแก้ไขหัวข้อที่ต้องการได้ เมื่อตรวจสอบข้อมูลครบถ้วนเรียบร้อยให้ click ที่ปุ่ม Install เพื่อทำการติดตั้งระบบปฏิบัติการ Ubuntu ลงเครื่อง

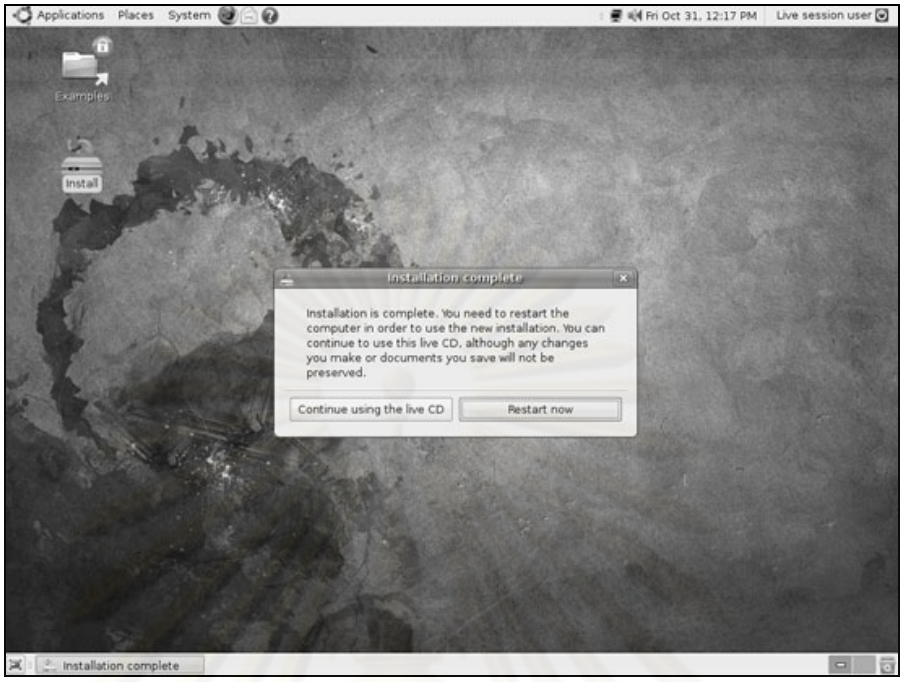


รูปที่ 40 รายละเอียดข้อมูลก่อนติดตั้งระบบ



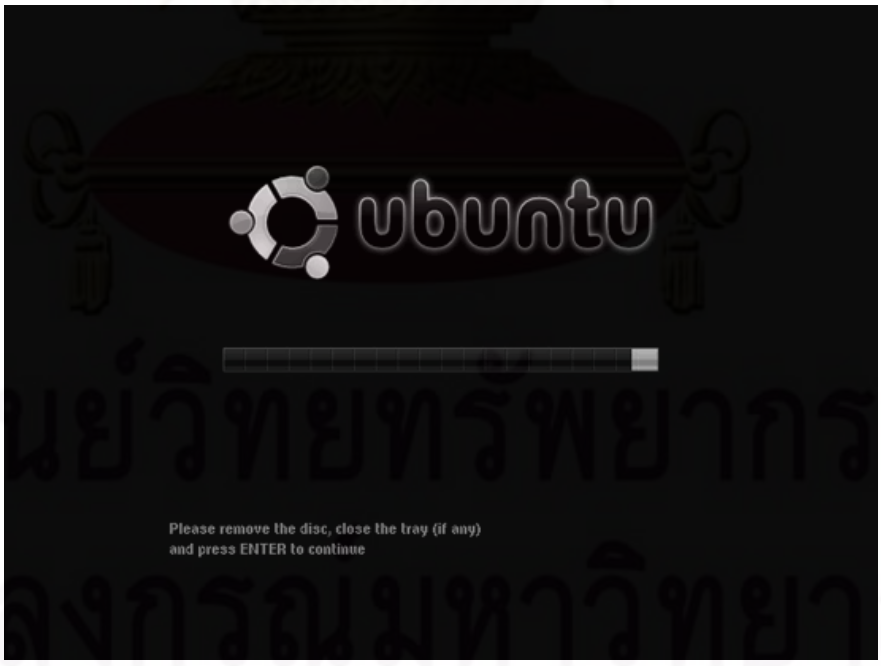
รูปที่ 41 สถานะการติดตั้งระบบ

10. เมื่อระบบได้ทำการติดตั้งเสร็จเรียบร้อยแล้ว ให้ทำการ restart เครื่องใหม่ก่อนเริ่มการใช้งาน โดย click ที่ปุ่ม Restart now



รูปที่ 42 ข้อความให้ restart เครื่องใหม่หลังการติดตั้งระบบเสร็จ

11. เมื่อ restart เครื่องขึ้นมาแล้ว จะมีหน้าต่างให้กรอกข้อมูล Login โดยใส่ Username และ Password ที่ได้กรอกข้อมูลในขั้นตอนที่ 8 (ถ้าใส่ข้อมูลผิดจะไม่สามารถเข้าใช้ระบบปฏิบัติการ Ubuntu ได้)



รูปที่ 43 สถานะการตรวจสอบข้อมูลก่อนเข้าสู่ระบบ

รูปที่ 44 หน้าต่างให้กรอก username เพื่อเข้าสู่ระบบ

รูปที่ 45 หน้าต่างให้กรอก Password

12. เข้าสู้ระบบปฏิบัติการ Ubuntu ที่พร้อมใช้งานได้

รูปที่ 46 หน้าจอ Desktop ของระบบ

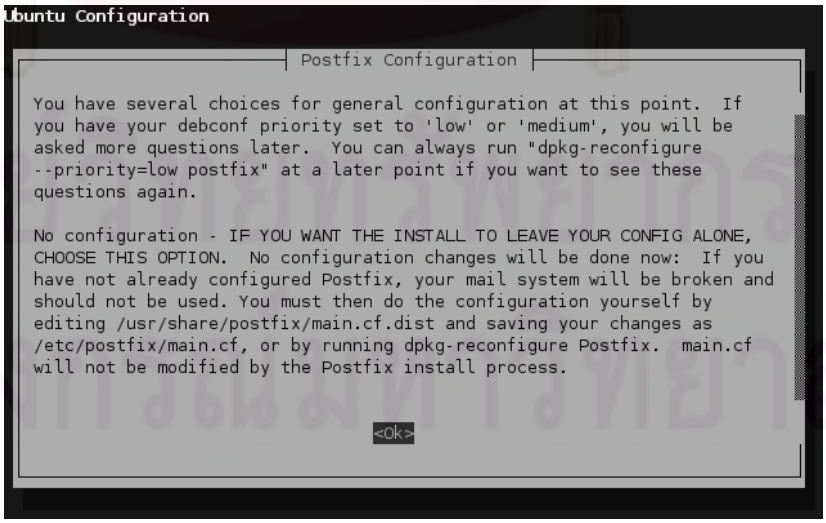
<u>การติดตั้งเครื่องแม่ข่ายอีเมล Postfix บนระบบปฏิบัติการ Ubuntu</u>

การติดตั้งเครื่องแม่ข่ายอีเมลได้มี 4 ขั้นตอนดังนี้

1. Mail Transfer Agent : Postfix เป็น MTA ที่ทำหน้าที่ในการรับส่งเมล์
   a. ทำการติดตั้ง postfix กับ SMTP-AUTH และ TLS โดยทำการติดตั้ง postfix packade ดังนี้

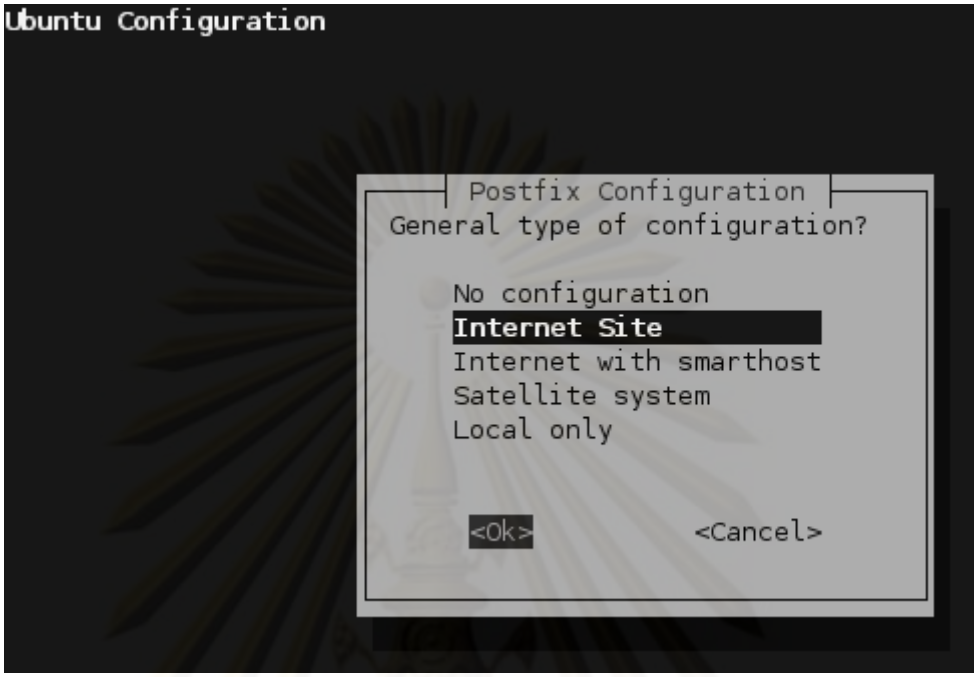
```
sudo aptitude install postfix
```

   b. จะแสดงหน้าต่าง Postfix Configuration ให้ click ที่ปุ่ม <ok>



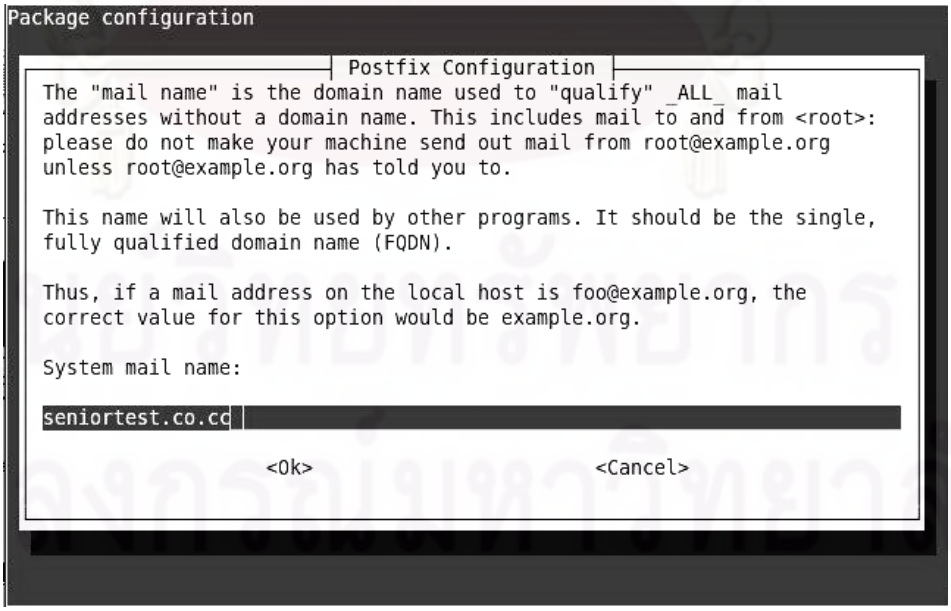
รูปที่ 47 หน้าต่าง Postfix Configuration

c. เลือกชนิดของการ config เป็นแบบ Internet Site แล้วกดที่ปุ่ม <ok>



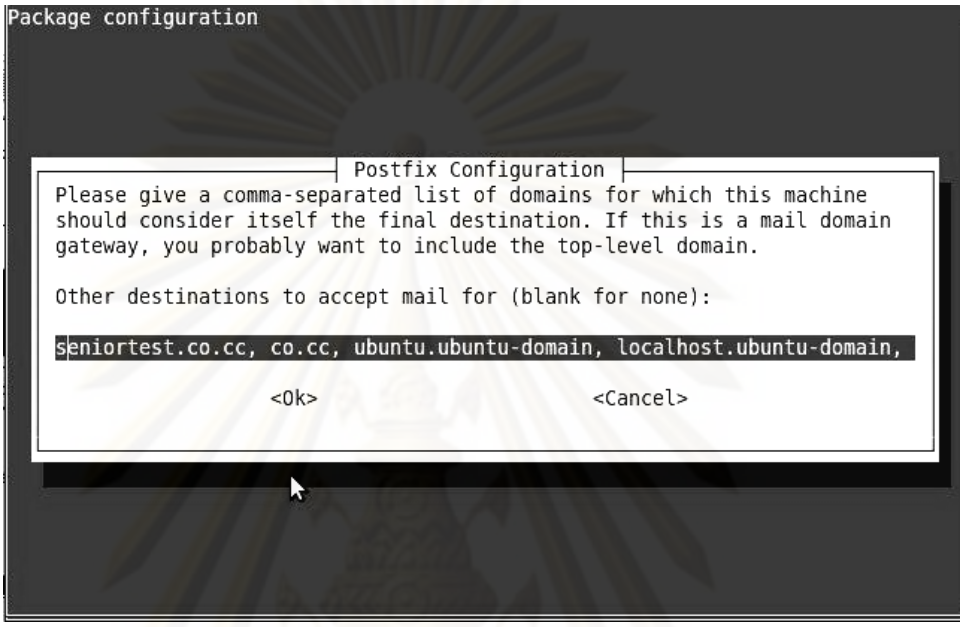
รูปที่ 48 เลือกชนิดการติดตั้งโปรแกรม Postfix

d. ใส่ชื่อ domain name (ที่จะให้แสดงหลังเครื่องหมาย @ ของ Email address) ซึ่งในที่นี้ได้มีการสมัครทดลองบน domain จริงของ co.cc โดยมีการเพิ่ม MX Record บน DNS Server ให้ชี้ไปที่ seniortest.co.cc จึงทำการใส่ข้อความ ดังรูปภาพ แล้ว click ที่ปุ่ม <ok>



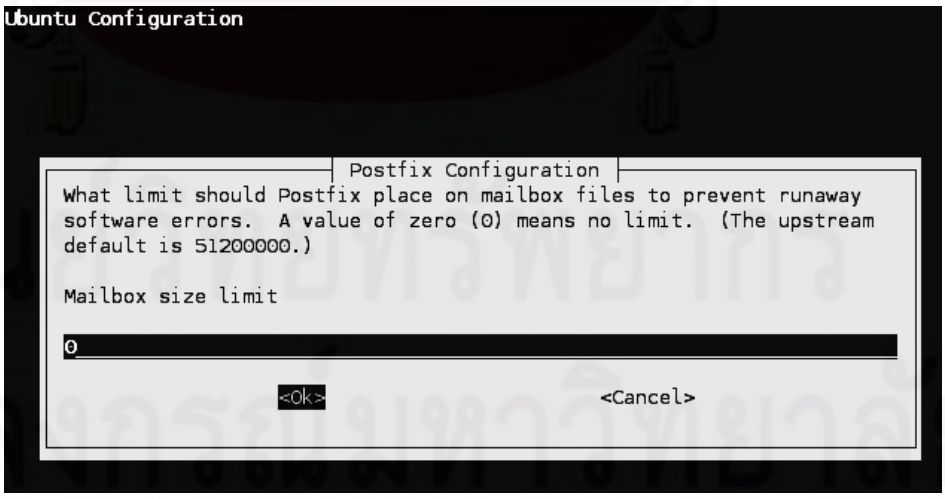
รูปที่ 49 การใส่ชื่อ Domain Name

e. ให้ทำการใส่ชื่อ domain ปลายทาง เพื่อเป็นการระบุว่าจะยอมรับ domain ใดบ้าง ที่จะให้เข้ามาใน mail server ให้ทำการเพิ่มเติม domain แล้ว click ที่ปุ่ม <ok>



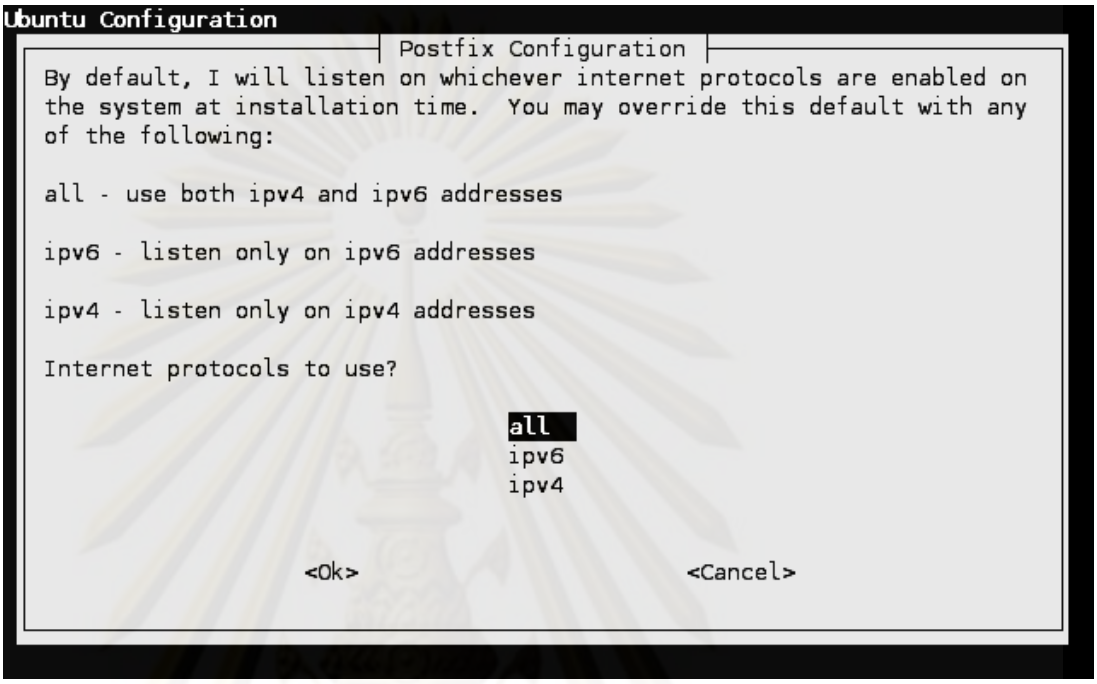
รูปที่ 50 การกำหนด Domain ปลายทาง

f. การจำกัด mailbox files ให้เครื่องสามารถกำหนดพื้นที่จัดเก็บให้กับ mail ที่เข้ามาใน server ในที่นี้กำหนดให้เป็น 0 คือหมายถึงการรับแบบไม่มีขีดจำกัด แล้ว click ที่ปุ่ม <ok>



รูปที่ 51 การกำหนดพื้นที่ในการรับอีเมล (E-mail)

g. ทำการเลือกชนิดของ protocol ที่ใช้กับระบบ ในที่นี้เลือกเป็น all เพื่อรองรับการทำงานทั้งระบบเก่าและระบบใหม่ คือ ipv6 และ ipv4 แล้ว click ที่ปุ่ม <ok>



รูปที่ 52 กำหนด Protocol

h. เมื่อติดตั้งเสร็จเรียบร้อยแล้ว จะมีหน้าต่างแสดงรายละเอียดของข้อมูล ดังรูปที่ 44

```
root@ubuntu:~# sudo dpkg-reconfigure postfix
 * Stopping Postfix Mail Transport Agent postfix                    [ OK ]
setting synchronous mail queue updates: false
setting myorigin
setting destinations: seniortest.co.cc, co.cc, ubuntu.ubuntu-domain, localhost.u
buntu-domain, localhost, hotmail.com, gmail.com
setting relayhost:
setting mynetworks: 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
setting mailbox_size_limit: 0
setting recipient_delimiter: +
setting inet_interfaces: all
setting inet_protocols: all
WARNING: /etc/aliases exists, but does not have a root alias.

Postfix is now set up with the changes above.  If you need to make changes, edit
/etc/postfix/main.cf (and others) as needed.  To view Postfix configuration
values, see postconf(1).

After modifying main.cf, be sure to run '/etc/init.d/postfix reload'.

Running newaliases
 * Stopping Postfix Mail Transport Agent postfix                    [ OK ]
 * Starting Postfix Mail Transport Agent postfix                    [ OK ]
root@ubuntu:~#
```

รูปที่ 53 รายละเอียดข้อมูลการติดตั้ง Postfix

i. สามารถทำการเปิดไฟล์ configuration เพื่อทำการปรับแต่งค่าเพิ่มเติมตามความเหมาะสมในภายหลังได้ โดยเข้าไปที่ /etc/postfig/main.cf

j. เมื่อปรับแต่งค่าต่างๆ เรียบร้อยแล้วให้ทำการ restart การทำการของ postfix ใหม่ด้วยคำสั่ง

```
sudo /etc/init.d/postfix restart
```

k. ทำการตรวจสอบการทำงานของ postfix ว่าสามารถทำงานได้หรือไม่ จากคำสั่ง

```
telnet localhost 25
```

จะแสดงหน้าต่างขึ้นมาดังรูปที่ 45

```
root@ubuntu:~# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.nonspam.test ESMTP Postfix (Ubuntu)
```

รูปที่ 54 การตรวจสอบการใช้งานโปรแกรม Postfix

2. Mail Delivery Agent : Dovecot ทำหน้าที่รับและจัดส่งเมล์ (IMAP/POP3) ของผู้ใช้แต่ละคน ที่มีอยู่ในรายการฐานข้อมูล

a. ติดตั้ง IMAP สำหรับรองรับการทำงานในระบบ online ด้วยคำสั่ง

```
sudo aptitude install dovecot-imapd
```

b. ติดตั้ง POP3 สำหรับรองรับการทำงานในระบบ online ด้วยคำสั่ง

```
sudo aptitude install dovecot-pop3d
```

c. ทำการ config ค่าของ dovecot โดยเข้าไปที่ /etc/dovecot/dovecot.conf ให้มีค่าที่ต้อง ดังนี้

```
protocols = pop3 pop3s imap imaps
```

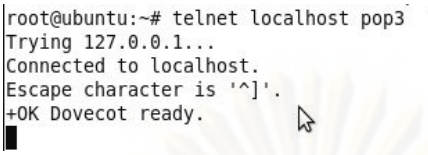
d. เมื่อติดตั้งเรียบร้อยแล้วให้ทำการ restart dovecot ใหม่ ด้วยคำสั่ง

```
sudo /etc/init.d/dovecot.restart
```

e. ทำการตรวจสอบการทำงานของ IMAP/POP3 ด้วยคำสั่ง

```
telnet locohost pop3
```

ถ้า dovecot ทำงาน จะแสดงผลดังรูปที่ 46

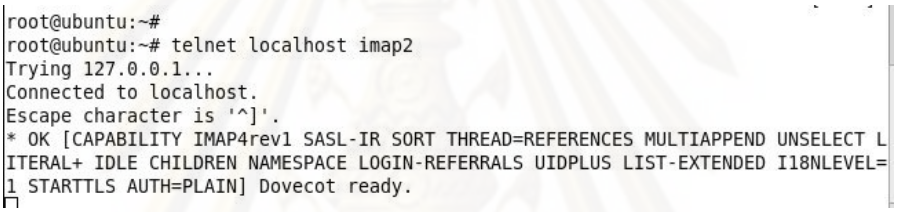


รูปที่ 55 การตรวจสอบการใช้งาน Dovecot POP3

หรือ

```
telnet locohost imap2
```

ถ้า dovecot ทำงาน จะแสดงผลดังรูปที่ 47



รูปที่ 56 การตรวจสอบการใช้งาน Dovecot IMAP

3. Webmail : Squirrelmail เป็น webmail browser ที่ช่วยให้สามารถตรวจสอบ mail ผ่านทาง web browser ได้ ซึ่งก่อนจะทำการติดตั้ง squirrelmail ได้ ต้องทำการติดตั้งโปรแกรมก่อน ดังนี้

   3.1 Mail Transfer Agent (ได้ทำการติดตั้งไปแล้วจากขั้นตอนที่ 1)

   3.2 Mail Delivery Agent (ได้ทำการติดตั้งไปแล้วจากขั้นตอนที่ 2)

   3.3 Apache2 Web Server

   a) ติดตั้ง apache2 ด้วยคำสั่ง

```
sudo aptitude install apache2
```

   b) ทำการติดตั้ง php5 ด้วยคำสั่ง

```
sudo aptitude install php5
```

```
sudo aptitude install libapache2-mod-php5
```

c) หลังจากทำการติดตั้ง apache2 เรียบร้อยแล้ว ให้ทำการ restart ใหม่ ด้วยคำสั่ง

```
sudo /etc/init.d/apache2 restart
```

d) ตรวจสอบการทำงานของ apache2 ด้วยการเปิด web browser แล้วพิมพ์ http://localhost หรือ http://127.0.0.1

e) ติดตั้ง web mail คือ squirrelmain ด้วยคำสั่ง

```
sudo aptitude install squirrelmail
```

f) ทำการ config ค่าใน squirrelmail ให้เป็นค่าตามที่กำหนด

```
sudo squirrelmail-configure
```

เลือกเมนูที่ 2 แล้วทำการเปลี่ยนค่า ดังนี้

```
A.  Update IMAP Settings   : localhost:143 (other)
```

เลือกเมนูที่ 4 ให้เปลี่ยนค่าตรง option 11 ให้เป็น enable

```
A.  Update IMAP Settings   : localhost:143 (other)
```

g) ทำการ congic apache2 โดยการคัดลอกไฟล์จาก squirrelmail ไปไว้ที่ apache2 ด้วยคำสั่ง

```
sudo cp /etc/squirrelmail/apache.conf /etc/apache2/sites-
sudo ln -s /etc/apache2/sites-available/squirrelmail /etc/apache2/sites-
```

h) ทำการ restart apache2 ก่อนการตรวจสอบการทำงานของ squirrelmail ด้วยคำสั่ง

```
sudo /etc/init.d/apache2 force-reload
```

i) ทำการเรียกใช้งาน web mail โดยการเปิด web browser แล้วพิมพ์ http://localhost/squirrelmail

ประวัติผู้เขียนวิทยานิพนธ์

นายเฉลิมพล ณ สงขลา เกิดเมื่อวันที่ 6 กันยายน พ.ศ. 2528 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ในปีการศึกษา 2550 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2551