รายงานการวิจัย

เรื่อง

การทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์

Latent Text Mining for Detection and Prevention from Cybercrime

โดย

ผู้ช่วยศาสตราจารย์ ดร.วฤษาย์ ร่มสายหยุด

ผู้ช่วยศาสตราจารย์ กชกร ณ นครพนม

อาจารย์ ดร. พิมผกา ประเสริฐศิลป์

อาจารย์ ปิยพร นุรารักษ์

การวิจัยครั้งนี้ได้รับทุนอุดหนุนการวิจัยทางวิชาการประจำปี พ.ศ. 2559

จากทุนวิชาการ สำหรับอาจารย์ประจำสำนัก/วิชาการ/สถาบัน (เงินรายได้)

มหาวิทยาลัยสุโขทัยธรรมาธิราช

ชื่อเรื่อง การทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์

ชื่อผู้วิจัย ผู้ช่วยศาสตราจารย์ ดร. วฤษาย์ ร่มสายหยุด และคณะ

ปีที่แล้วเสร็จ 2561

บทคัดย่อ

การกลั่นแกล้งทางอินเทอร์เน็ตกลายเป็นปัญหาใหญ่ที่สุดสำหรับเด็กหรือวัยรุ่นที่กำลังเผชิญอยู่ในปัจจุบัน ความท้าทายในการทำวิจัยโดยใช้เทคโนโลยีเพื่อแก้ปัญหาการกลั่นแกล้งทางอินเทอร์เน็ตและประเมินผลการปรับปรุงประสิทธิภาพจากผลการทดลอง

งานวิจัยนี้มีวัตถุประสงค์เพื่อ 1) เพื่อสร้างอัลกอริทึมใหม่ของการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ และ 2) เพื่อประเมินความถูกต้องของการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ วิธีดำเนินการวิจัยโดยการพัฒนาระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ ซึ่งนำเสนออัลกอริธึมใหม่ในการทำนายข้อความแบบไดนามิกในรูปแบบเวลาและการกระจายหัวข้อเอกสารที่มีความสัมพันธ์กันบนโปรแกรมอาปาเช่มาเฮาท์ โดยอัลกอริทึมนี้สามารถใช้ในการวิเคราะห์ความปลอดภัยในโลกไซเบอร์สำหรับการตรวจหาภัยคุกคามจากการสนทนาโต้ตอบระหว่างกันที่เกิดขึ้นตลอดเวลา โดยหลักการของอัลกอริทึมประกอบด้วยสองขั้นตอนได้แก่ ขั้นตอนแรกเป็นนำข้อความการสนทนามาทำการฝึกอบรม แปลงเป็นแฟ้มข้อมูลแบบเรียงลำดับ คำนวณค่าการแจกแจงค่าความถี่ของคำด้วยเวคเตอร์ร่วม และสร้างแบบจำลองโดยการใช้แบบอนุมานแปรผันของเบย์ สำหรับขั้นตอนที่สองนำข้อความการสนทนามาทำการทดสอบ ในการทดลองนี้ได้มีการวิเคราะห์และศึกษาชุดข้อมูลจริงซึ่งเก็บรวบรวมจากปี พ.ศ. 2549-2557 โดยใช้รูปแบบคำต่างๆของคลังข้อมูล โดยผลการทำงานนี้สามารถประยุกต์ระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์สำหรับใช้ในชีวิตประจำวันเพื่อป้องกันการกลั่นแกล้งทางอินเทอร์เน็ตได้

ผลการวิจัยนี้ ได้ผลค่าความถูกต้องเท่ากับร้อยละ 92.85 ค่าการเรียกคืนเท่ากับร้อยละ 86.67 และค่าประสิทธิภาพเท่ากับร้อยละ 89.69 ตามลำดับ

คำสำคัญ : การกลั่นแกล้งทางอินเทอร์เน็ต การทำเหมืองข้อความแฝง อนุมานแปรผันของเบย์ ข้อมูลกระแสการไหล

Title : Latent Text Mining for Detection and Prevention from Cybercrime

Researchers : Assistant Professor Dr. Walisa Romsaiyud, et al.

Year: 2018

Abstract

Cyberbullying has been one of the biggest problems for children or teenagers are facing today. The challenges of conducting research by using technology for solving the cyberbullying problems and improve the performance from the experimental results of the evaluation.

The purpose of this research were as follows: 1) to generate a new algorithm of a Latent Text mining for detection and prevention from Cybercrime, and 2) to evaluate the accuracy of a Latent Text mining for detection and prevention from Cybercrime. The research methodology developed a Latent Text Mining for Detection and Prevention from Cybercrime. We proposed a new algorithm for predicting a dynamic message in a time-based manner and based on the distribution of related documents on Apache Mahout. The proposed algorithm can be applied in cyber security analytics for threat detection of the conversation dialogs that continuously change over time. In particular, the algorithm includes two main methods. The first method collected the conversation dialog in training phase, transformed data to sequential file, calculated word scores from the word co-occurrence vector and generated a model using a variational Bayesian inference in such a way that the documents advance over a sequential time, and the second method uses the conversation dialogs for testing phase. In this experiment, authentic datasets collected from year 2006 to 2014 using corpus-wide patterns of words-were analyzed and studied. In order to enhance the reliability and computation time, the methods were applied on real-life settings where cyberbullying features and user-based features had experienced.

According to research experiment, the evaluation of the research revealed that the measurement results were as 92.85% for precision, 86.67% for recall and 89.69% for f-measure respectively.

Keyword: Cyberbullying, Latent Text Mining, Variational Bayesian inference, Data Stream Mining

คำนำ

การกลั่นแกล้งผ่านทางโลกไซเบอร์ (Cyberbullying) เป็นการกลั่นแกล้งทางออนไลน์ ที่เป็นการกระทำที่ก่อให้เกิดความเสียหายหรือคุกคามผ่านเครือข่ายเทคโนโลยีสารสนเทศ โดยการละเมิดลิขสิทธิ์เช่นการซ้ำจากการโพสต์รูปภาพหรือคลิปวิดีโอ เขียนบทความ บล็อกและแชตคุยกัน อาจรวมถึงการเปิดเผยสิ่งพิมพ์หรือการส่งผ่านข้อมูลที่เป็นเรื่องส่วนตัวของคนหนึ่งไปยังที่สาธารณะโดยไม่ได้รับอนุญาตจากเจ้าของ ซึ่งก่อให้เกิดความเสียหายกับผู้ที่ตกเป็นเหยื่อ เช่นเด็ก เยาวชนหรือสตรี ทั้งต่อการสูญเสียชื่อเสียง ทรัพย์สินหรือชีวิตได้

โครงการวิจัยเรื่องการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ เป็นการประยุกต์การทำงานในการกำหนดค่าความน่าจะเป็นบนข้อความที่ได้จากการสนทนาออนไลน์เพื่อหาการจัดสรรหัวข้อแฝงหรือแอลดีเอ (Latent Dirichlet Allocation: LDA) โดยการสกัดคำ ประโยค วลี รูปแบบข้อความและคำแฝง ในกลุ่มคำ 4 กลุ่มหลักของการกลั่นแกล้งทางอินเทอร์เน็ตได้แก่ 1) ล่วงละเมิดทางเพศ (Sexual harassment) 2) หลอกลวงเงิน (Money Mule Scams) 3) พยายามฆ่าตัวตาย (Suicide Attempts) และ 4) ยาเสพติดและเครื่องดื่มแอลกอฮอล์ (Drug and Alcohol Abuse) โดยอาศัยวิธีการคัดเลือกคุณลักษณะ จากกลุ่มหัวข้อการสนทนาด้วยวิธีการสร้างแบบจำลองโดยอาศัยการเรียนรู้แบบเบย์ (Bayesian Learning) จากข้อมูลการฝึกอบรม (Training Data) และการพยากรณ์ข้อมูลจากการสนทนาในสถานการณ์แบบต่างๆ จากข้อมูลทดสอบ (Testing Data)

การนำผลงานวิจัยไปใช้ประโยชน์เพื่อทำการตรวจสอบภัยคุกคาม การกลั่นแกล้งทางอินเทอร์เน็ตจากการสนทนาและเพื่อป้องกันผู้ที่จะตกเป็นเหยื่อได้ทันเวลาโดยอาศัยเทคโนโลยีการจัดสรรหัวข้อแฝงและการเรียนรู้เบยส์อย่างง่าย

คณะผู้วิจัย

กิตติกรรมประกาศ

โครงการวิจัยเรื่องการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ (Latent Text Mining for Detection and Prevention from Cybercrime) มีจุดมุ่งหมายเพื่อจะได้นำผลการศึกษา วิเคราะห์ พัฒนาและดำเนินไปใช้ประโยชน์ในการทำการตรวจสอบภัยคุกคาม การกลั่นแกล้งทางอินเทอร์เน็ตจากการสนทนาและเพื่อป้องกันผู้ที่จะตกเป็นเหยื่อได้ทันเวลาโดยอาศัยเทคโนโลยีการจัดสรรหัวข้อแฝงและการเรียนรู้เบย์ โครงการวิจัยนี้ได้รับทุนอุดหนุนจากทุนวิชาการ สำหรับอาจารย์ประจำสำนัก/วิชาการ/สถาบัน (เงินรายได้) ประจำปี 2559 จากสถาบันวิจัยและพัฒนา มหาวิทยาลัยสุโขทัยธรรมาธิราช และได้รับการอนุเคราะห์และความร่วมมือในการให้ข้อมูลอย่างดียิ่งจากคณาจารย์และเจ้าหน้าที่ของสาขาวิชาวิทยาศาสตร์และเทคโนโลยี สำนักคอมพิวเตอร์ มหาวิทยาลัยสุโขทัยธรรมาธิราชและบุคลากรที่เกี่ยวข้อง จนทำให้คณะผู้วิจัยสามารถทำการวิจัยเรื่องนี้ได้บรรลุวัตถุประสงค์ที่ตั้งไว้ คณะผู้วิจัยจึงใคร่ขอขอบคุณทุกท่านและทุกหน่วยงานไว้ ณ โอกาสนี้

คณะผู้วิจัย

กันยายน 2561

สารบัญ

สารบัญรูป

หน้า

สารบัญตาราง

บทที่ 1
บทนำ

1. ความเป็นมาและความสำคัญของปัญหา

อาชญากรรมทางคอมพิวเตอร์ (Cybercrime) (อาชญากรรมไซเบอร์, 2015) หมายถึงการกระทำที่ผิดกฎหมายหรืออาชญากรรมผ่านทางเครื่องมืออิเล็กทรอนิกส์ หรือ เครือข่ายระบบเทคโนโลยีคอมพิวเตอร์ในการก่อเหตุ เนื่องจากการเจริญเติบโตของระบบเทคโนโลยีคอมพิวเตอร์อย่างไม่หยุดยั้งส่งผลทำให้เกิดช่องทางที่อาชญากรได้ศึกษาและคิดวิธีการต่างๆได้หลากหลายชนิด ซึ่งเป็นอันตรายต่อผู้คนเป็นวงกว้างทั้งในระดับความมั่นคงภายในและภายนอกประเทศ เพราะเป้าหมายสำคัญของ อาชญากรทางคอมพิวเตอร์ (Criminal cyber) คือ การโจรกรรม แก้ไข ลักลอบ เผยแพร่ หรือ ประโยชน์ต่างๆโดยมิชอบ ซึ่งส่งผลกระทบทางการเงินเศรษฐกิจ การเมืองและสังคมเป็นวงกว้าง

จากข้อมูลสถิติของศูนย์ประสานการรักษาความมั่นคงปลอดภัยระบบคอมพิวเตอร์ประเทศไทย (ไทยเซิร์ต) [1](Cyber Insurance Potential in Thailand, 2015) พบว่าอาชญากรรมด้านไซเบอร์ก่อให้เกิดความเสียหายทั่วโลกคิดเป็นมูลค่าระหว่าง US$ 3.75 แสนล้าน จนถึง US$ 5.75 แสนล้านต่อปี และสำหรับประเทศไทยถูกจัดเป็นประเทศที่มีความเสี่ยงต่อภัยคุกคามด้านไซเบอร์สูงเป็นอันดับที่ 3 ของโลก หนึ่งในหลายๆ รูปแบบของอาชญากรรมทางคอมพิวเตอร์คือการหลอกลวงทางออนไลน์ (Online scam) เช่น การทำให้ผู้ใช้หรือเหยื่อ ซึ่งอาจจะเป็นเด็ก ผู้หญิง หรือบุคคลทั่วไป หลงเชื่อ และใช้เป็นช่องทางในการเข้าถึงข้อมูลส่วนตัวและข้อมูลสำคัญทางการเงินเพื่อแสวงหาประโยชน์ส่วนตัว เป็นต้น

การกลั่นแกล้งทางอินเทอร์เน็ต (Cyberbullying) (Thailand Social Media Landscape 2014, 2014) หมายความถึงการใช้อินเทอร์เน็ตเป็นเครื่องมือหรือช่องทางเพื่อก่อให้เกิดการคุกคาม ล่อลวงและการกลั่นแกล้งบนโลกอินเทอร์เน็ต ซึ่งสามารถเป็นทั้งผู้กระทำและผู้ถูกกระทำ โดยเป้าหมายจะเป็นกลุ่มเด็กเล็กจนถึงเด็กวัยรุ่น ในปัจจุบันผู้ใช้งานเครือข่ายสังคมออนไลน์หรือโซเชียลเน็ตเวิร์ค (Social network) เช่น เฟซบุ๊ก (Facebook) ไลน์ (Line) อินสตาแกรม (Instagram) ทวิตเตอร์ (Twitter) มายสเปซ (MySpace) ลิงค์อิน (LinkedIn) ยูทูบ (YouTube) และกูเกิลพลัส (Google+) มีจำนวนเพิ่มมากขึ้นเรื่อยๆ ทำให้เกิดเป็นกลุ่มสังคม (Social community) ขนาดใหญ่ โดยโซเชียลเน็ตเวิร์คที่ได้รับความนิยมหรือมีจำนวนผู้ใช้งานมากที่สุด คือเฟซบุ๊กที่มีผู้ใช้งานทั่วโลกและในประเทศไทย โดยในประเทศไทยมีผู้ใช้เฟซบุ๊กประมาณ 30 ล้านคน ซึ่งผู้ใช้งานเฟซบุ๊กทำการติดต่อแลกข้อมูลข่าวสารระหว่างสมาชิก ตั้งประเด็นถามตอบในเรื่องที่สนใจ โพสต์รูปภาพ โพสต์คลิปวิดีโอ เขียนบทความหรือบล็อกและแชท (Chat) คุยโต้ตอบกัน แต่ในบางครั้งคนที่สนทนาด้วยอาจเป็นคนที่ไม่เคยพบ ไม่ใช่คนใกล้ชิดหรือเป็นคนในครอบครัว เพื่อนและญาติ จะทำให้เกิดปัญหาจาก

[1] ไทยเซิร์ต จัดตั้งขึ้นในปี พ.ศ. 2543 เป็นศูนย์เทคโนโลยีอีเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สังกัดสานักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ กระทรวงวิทยาศาสตร์และเทคโนโลยี

อาชญกรรมไซเบอร์ ซึ่งทำให้เหยื่อได้รับความเสียหาย และผู้กระทำได้รับผลประโยชน์ตอบแทน และหากเหยื่อเป็นเด็กเยาวชนหรือผู้หญิงแลกเปลี่ยนข้อความ รูปภาพ หรือข้อมูลสำคัญอาจจะทำให้ตกเป็นเหยื่อ เช่น การปลอมแปลง การก่อการร้าย การขู่เข็ญ การทำลามกอนาจาร หลอกลวงเงินและทรัพย์สิน การทำร้ายตัวเอง และการฆ่าตัวตาย เป็นต้น

แบบจำลองหัวข้อ (Topic model) เป็นแบบจำลองการกระจายตัวของข้อมูลจากการทำเหมืองข้อความ (Text mining) เพื่อนำมาใช้ในการจัดกลุ่มของข้อมูลจากค่าสถิติความน่าจะเป็น โดยแบบจำลองหัวข้อนี้มีพื้นฐานมาจากแนวคิดที่ว่าในเอกสารหนึ่งๆ เกิดจากการรวมตัวของหลายๆ หัวข้อ ซึ่งแต่ละหัวข้อมีการแจกแจงค่าความน่าจะเป็นของคำที่เกิดขึ้นหลายๆ คำในแต่ละหัวข้อ รูปแบบการสร้างแบบจำลองหัวข้อที่ได้รับความนิยมเรียกว่าการจัดสรรหัวข้อแฝงหรือแอลดีเอ (Latent Dirichlet Allocation : LDA) ซึ่งการทำงานของการจัดสรรหัวข้อแฝงถูกนำไปประยุกต์ใช้ในการค้นหารูปแบบคำแฝงที่ซ่อนอยู่ในกลุ่มคำของเอกสาร หรือการค้นหาคำ ประโยคในเอกสารงานวิจัยต่างๆ ว่ามีความน่าจะเป็นที่เกี่ยวข้องกันหรือตรงกัน อาทิโปรแกรมตรวจสอบการลอกเลียนวรรณกรรม (Plagiarism detection) การตั้งชื่อหัวข้อข่าว หรือการกำหนดชื่องานวิจัย เป็นต้น

ดังนั้นงานวิจัยนี้ขอนำเสนอการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ โดยประยุกต์การทำงานในการกำหนดค่าความน่าจะเป็นบนการจัดสรรหัวข้อแฝงเพื่อสกัดคำ ประโยค วลี รูปแบบข้อความ คำแฝงและคำกำกวมในหลักไวยากรณ์ภาษาอังกฤษ โดยจำแนกประเภทของเอกสาร (Text classification) ตามวิธีการคัดเลือกคุณลักษณะ (Feature selection) จากกลุ่มหัวข้อการสนทนาด้วยวิธีการสร้างแบบจำลองโดยอาศัยทฤษฎีของเบย์ (Bayes' Theorem) จากข้อมูลการฝึกอบรม (Training data) และการพยากรณ์ข้อมูลจากการสนทนาในสถานการณ์แบบต่างๆ จากข้อมูลทดสอบ (Testing data) เพื่อทำการตรวจสอบภัยคุกคาม การกลั่นแกล้งทางอินเทอร์เน็ตจากการสนทนาและเพื่อป้องกันผู้ที่จะตกเป็นเหยื่อได้ทันเวลาโดยอาศัยเทคโนโลยีการจัดสรรหัวข้อแฝงและทฤษฎีของเบย์

2. วัตถุประสงค์ของการวิจัย

โครงการวิจัยการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ มีวัตถุประสงค์ของการวิจัยเพื่อ

1) เพื่อสร้างอัลกอริทึมใหม่ของการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์

2) เพื่อประเมินความถูกต้องของการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์

3. ขอบเขตการวิจัย

3.1 ขอบเขตด้านประชากร

- ประชากรกลุ่มตัวอย่าง คือ สมาชิกเว็บไซต์ Perverted-justice, Formspring และ MySpace ในปี 2010

3.2 ขอบเขตด้านเนื้อหา

- ข้อความเนื้อหาจากการโพสต์ของเว็บไซต์ perverted-justice.com, ข้อความจาก Formspring และ ข้อความจาก MySpace จำนวนทั้งสิ้น 127,974 ซึ่งใช้ข้อมูลการฝึกอบรมและการทดสอบข้อมูลที่แยกในสถานการณ์ต่างๆ บนข้อมูลแบบสตรีมมิ่ง ข้อมูลการฝึกอบรม (Training Data) จำนวน 23,492 โดยแบ่งข้อมูลเป็นแบบ 10-fold cross-validation เพื่อทดสอบประสิทธิภาพของแบบจำลอง

3.3 ขอบเขตด้านเวลา

- ในการทดลองนี้ได้มีการวิเคราะห์และศึกษาชุดข้อมูลจริงซึ่งเก็บรวบรวมระหว่างปี พ.ศ. 2549-2557

4. เครื่องมือในการวิจัย

1) พัฒนาโปรแกรมด้วยโปรแกรมฝั่งเครื่องเซิร์ฟเวอร์ด้วยภาษา Java และ R สำหรับ Windows

2) พัฒนาโปรแกรมด้วยโปรแกรมฝั่งเครื่องไคลเอนต์ด้วยภาษา HTML5 และ JavaScript สำหรับ Windows (การพัฒนาเว็บแอปพลิเคชัน)

3) การทดสอบการทำงานโปรแกรมบนเครื่อง HP - HP Compaq Z400 Workstation (ช่อง 6-DIMM) สำหรับโหนดหลัก (Master Node) และกำหนดโหนดลูก (Slave Nodes) จำนวน 64 โหนด เป็นอินเทลซีออนโปรเซสเซอร์ CPU, W3508 @ 2.40 GHz กับ 4.00 GB RAM

4) การทดสอบการทำงานโปรแกรมบนกรอบการทำงานของ Apache Hadoop รุ่น 2.7.3 สำหรับ Windows ให้บริการกำหนดค่าการกระจายการบริการและการประสานการรวมข้อมูลกลับมาที่โหนดหลักและการสร้างแบบจำลองหัวข้อที่ด้วยการจัดสรรหัวข้อแฝง

5) การทดสอบระบบงานเหมืองข้อความจริงบนเครื่องเซฟเวอร์เพื่อประมวลผลข้อมูลขนาดใหญ่จาก Amazon Web Services (AWS) บน Elastic Compute Cloud (EC2) คิดราคาการประมวลผลตามวินาที (Per-second billing)

5. นิยามศัพท์เฉพาะ

1) แบบจำลองหัวข้อ (Topic Model) หมายถึงการสร้างแบบจำลองการกระจายตัวของข้อมูลเพื่อนำมาใช้ในการจัดกลุ่มของข้อมูลแบบจำลองหัวข้อ มีพื้นฐานมาจากแนวคิดที่ว่าในเอกสารเกิดจากการรวมตัวของหลายๆ หัวข้อ ซึ่งแต่ละหัวข้อมีการแจกแจงความน่าจะเป็นของคำที่เกิดขึ้นหลายๆ คำในแต่ละหัวข้อ

2) การจัดสรรหัวข้อแฝง (Latent Dirichlet Allocation: LDA) หมายถึงแบบจำลองความน่าจะเป็น สำหรับข้อมูลที่ไม่ต่อเนื่อง เช่น คลังข้อมูล (Corpus) ซึ่งมีแนวคิดพื้นฐานมาจากเอกสารสามารถแทนด้วยการรวมกันของหัวข้อแฝง (Latent) อยู่ในเอกสาร ซึ่งแต่ละหัวข้อมีการกระจายตัวของคำ

3) การทำเหมืองข้อความ (Text Mining) หมายถึงวิธีการเพื่อค้นหารูปแบบจากข้อความจำนวนมากโดยอัตโนมัติ ซึ่งอาศัยขั้นตอนวิธีจากหลักการสถิติ การเรียนรู้ของเครื่องและการรู้จำ

4) ทฤษฎีของเบย์ (Bayes' Theorem) หมายถึงวิธีการจำแนกหมวดหมู่ของข้อมูล โดยใช้หลักความน่าจะเป็นที่มีพื้นฐานมาจากกฎของเบย์ (Bayes' Rule) แต่จะลดความซับซ้อนลงโดยจะเพิ่มสมมติฐานที่ว่าคุณสมบัติต่างๆ ของข้อมูลจะไม่ขึ้นต่อกัน (เป็นอิสระต่อกัน) หรือกล่าวได้ว่าความน่าจะเป็นของข้อมูลที่จะเป็นกลุ่ม B ใดๆ สำหรับข้อมูลที่มีคุณสมบัติ X = {A1,...., Ai} หรือใช้สัญลักษณ์ว่า $P(A_i \mid B)$ ซึ่งจากกฎของเบย์และสมมติฐานที่ว่าคุณสมบัติแต่ละตัวไม่ขึ้นต่อกันทำให้ได้สมการดังนี้

$$P(A_i \mid B) = \frac{P(B|A_i)\,P(A_i)}{P(B)} \quad (1)$$

5) วิธีการคัดเลือกคุณลักษณะ (Feature Selection Methods) หมายถึงกระบวนการที่คัดเลือก สับเซตจากเซตของคุณลักษณะ (Feature set) ต้นฉบับ ซึ่งจะทำให้ได้คุณลักษณะที่เหมาะสมในการนำไปใช้ในการจำแนกประเภท โดยที่คุณลักษณะคือเซตของศัพท์หรือคำที่เกิดในเอกสารทั้งหมด ซึ่งวิธีการคัดเลือกคุณลักษณะนี้จะช่วยปรับปรุงความถูกต้องในการจำแนกประเภทของเอกสาร

6) อาชญากรรมไซเบอร์ (CyberCrime) หมายถึงอาชญากรรมใด ๆ ที่เกี่ยวข้องกับคอมพิวเตอร์และเครือข่ายคอมพิวเตอร์ โดยทำให้เหยื่อได้รับความเสียหาย และผู้กระทำได้รับผลประโยชน์ตอบแทน ทำให้เกิดปัญหาอย่างมากต่อสังคมและประเทศชาติ เช่นการปลอมแปลง การก่อการร้าย การขู่เข็ญ การทำลามกอนาจาร หลอกลวงเงินและทรัพย์สิน หรือทำให้สูญเสียชีวิต

6. ประโยชน์ที่ได้รับ

จากผลการดำเนินโครงการวิจัยการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ ครั้งนี้คณะผู้วิจัยจะนำไปใช้ประโยชน์ดังนี้

1) มีระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์เพื่อป้องกันและลดปัญหาการหลอกลวงบนอินเทอร์เน็ตและอาชญากรรมทางคอมพิวเตอร์จากการโต้ตอบการสนทนา

2) ผู้ดูแลระบบอินเทอร์เน็ตหรือผู้มีส่วนเกี่ยวข้อง สามารถนำระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันอาชญากรรมไซเบอร์ ไปประยุกต์ใช้งานด้านการวิเคราะห์หัวข้อแฝงของเนื้อหาทั้งหมดจากการโต้ตอบการสนทนาทางอินเทอร์เน็ต เพื่อป้องกันผู้ที่จะตกเป็นเหยื่อจากการสนทนาบนอินเทอร์เน็ตได้ทันท่วงที

บทที่ 2
วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

การทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ ต้องอาศัยความรู้ความเข้าใจในทฤษฎีและหลักการของเรื่องต่างๆ ดังต่อไปนี้

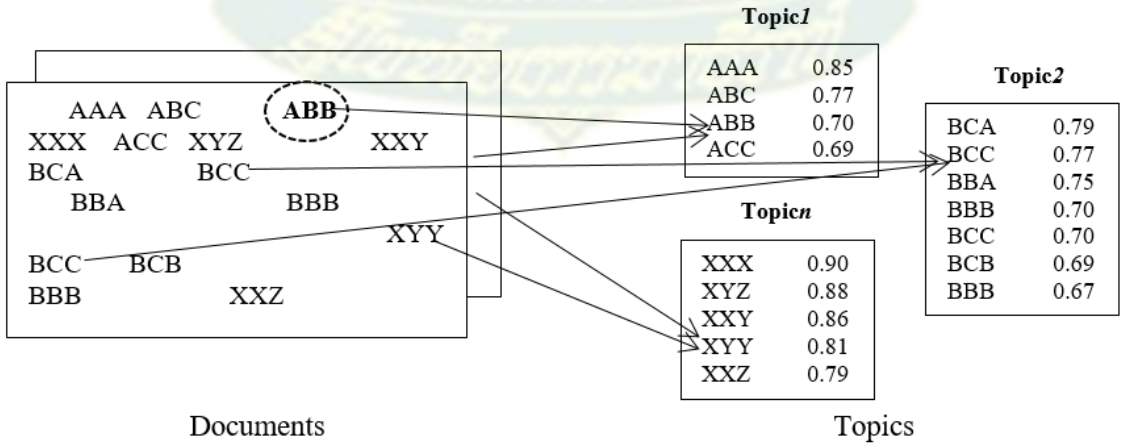
1. แบบจำลองหัวข้อ
2. การจัดสรรหัวข้อแฝง
3. เหมืองข้อความ
4. ทฤษฎีของเบย์
5. อาชญากรรมไซเบอร์และการกลั่นแกล้งทางอินเทอร์เน็ต
6. งานวิจัยที่เกี่ยวข้อง

1. วรรณกรรมที่เกี่ยวข้อง

จากวรรณกรรมที่เกี่ยวข้องของการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ ประกอบด้วยเนื้อหา 5 ประเด็น ได้แก่ 1) แบบจำลองหัวข้อ (Topic Model) 2) การจัดสรรหัวข้อแฝงหรือแอลดีเอ (Latent Dirichlet Allocation: LDA) 3) การทำเหมืองข้อความ (Text Mining) 4) ทฤษฎีของเบย์ (Bayes' Theorem) และ 5) อาชญากรรมไซเบอร์และการกลั่นแกล้งทางอินเทอร์เน็ต (CyberCrime and Cyberbullying) ดังรายละเอียดต่อไปนี้

1.1 แบบจำลองหัวข้อ

แบบจำลองหัวข้อ (Topic model) หมายถึง การสร้างแบบจำลองการกระจายตัวของข้อมูล เพื่อนำมาใช้ในการจัดกลุ่มของข้อมูลจากค่าความน่าจะเป็น โดยแบบจำลองหัวข้อนี้มีพื้นฐานมาจากแนวคิดที่ว่าในเอกสารหนึ่งๆ เกิดจากการรวมตัวของหลายๆ หัวข้อ ซึ่งแต่ละหัวข้อมีการแจกแจงค่าความน่าจะเป็นของคำที่เกิดขึ้นหลายๆ คำในแต่ละหัวข้อ ดังภาพที่ 2.1



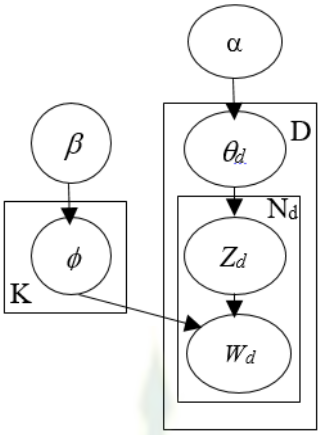
ภาพที่ 2.1 ตัวอย่างหัวข้อที่สร้างจากแบบจำลองหัวข้อ

จากภาพที่ 2.1 แสดงตัวอย่างหัวข้อที่สร้างจากแบบจำลองหัวข้อโดยในกลุ่มของเอกสาร (document) จะประกอบด้วยคำหลายๆ คำ เช่น “AAA” หรือ “ABB” และกลุ่มคำหลายๆ กลุ่มคำ ขึ้นกับการคำนวณค่าน้ำหนักของคำจากทั้งเอกสาร เพื่อหาค่าความน่าจะเป็นของแต่ละคำนั้น โดยในแต่ละหัวข้อ (topic) จะประกอบไปด้วยคำหลายๆ คำ ซึ่งการเรียงลำดับของคำจะเรียงลำดับตามความน่าจะเป็นสูงสุดจนไปถึงต่ำสุดในหัวข้อนั้นๆ เช่น ใน Topic ที่ 1 จะมีคำว่า "AAA" มีค่าความน่าจะเป็นของคำสูงสุดคือ 0.85 ที่จะเป็นคำใน Topic ที่ 1 และสำหรับใน Topic ที่ 2 จะมีคำว่า "BCA" มีค่าความน่าจะเป็นของคำสูงสุดคือ 0.79 ที่จะเป็นคำใน Topic ที่ 2 เป็นต้น การกระจายตัวของหัวข้อนั้นสามารถใช้ร่วมกันในเอกสารหลายฉบับได้และทำได้โดยการรวบรวมคำเหล่านี้จากในกลุ่มเอกสารนั้นๆ และแต่ละหัวข้อนั้นจะมีกลุ่มของคำที่มีความสัมพันธ์กันหรือเกี่ยวข้องกันในหัวข้อนั้นๆ

การเรียนรู้ของเครื่อง (machine learning) และการประมวลผลภาษาธรรมชาติ (natural language processing) นั้น แบบจำลองหัวข้อเป็นแบบจำลองเชิงสถิติในการค้นหา “หัวข้อ (topic)” เชิงนามธรรมที่สามารถเกิดขึ้นจากการรวบรวมเอกสารหลายอย่างเข้าด้วยกัน โดยแบบจำลองหัวข้อนั้นเป็นการใช้เครื่องมือการทำเหมืองข้อความ (text-mining tool) ของคำที่ค้นเจอบ่อยๆ เพื่อใช้ในการค้นหาโครงสร้างเชิงความหมาย (Semantic) ของคำซ่อนอยู่ในเนื้อหาข้อความในเอกสารนั้น ยกตัวอย่างเช่น เอกสารฉบับหนึ่งที่มีเนื้อหาเกี่ยวกับหัวข้อใดหัวข้อหนึ่งโดยเฉพาะ มักจะคาดหวังที่จะพบคำที่เกี่ยวกับหัวข้อนั้นๆ ปรากฏในเอกสารฉบับนั้นบ่อยครั้งไม่มากก็น้อย อาทิ คำว่า “สุนัข” และ “กระดูก” ก็มักปรากฏขึ้นบ่อยครั้งในเอกสารที่เกี่ยวกับสุนัข หรือคำว่า “แมว” และ “เหมียว” มักปรากฏในเอกสารเกี่ยวกับแมว แต่สำหรับคำว่า “ตัวนี้” และ “เป็น” จะปรากฏในเอกสารทั้งสองฉบับนี้พอๆกัน ซึ่งในเอกสารฉบับใดๆ มักจะมีหลายหัวข้อในสัดส่วนที่แตกต่างกัน ดังนั้น หากในเอกสารนั้น มีคำที่เกี่ยวกับแมวร้อยละ 10 และมีคำที่เกี่ยวกับสุนัขร้อยละ 90 ก็อาจกล่าวได้ว่า เอกสารนี้มีคำเกี่ยวกับสุนัขมากกว่าคำที่เกี่ยวกับแมวประมาณ 9 เท่า ดังนั้นหัวข้อ (topic) ที่สร้างขึ้นด้วยเทคนิคแบบจำลองหัวข้อนี้จะถูกจัดเป็นกลุ่ม (cluster) ของคำที่มีความหมายคล้ายคลึงกัน โดยแบบจำลองหัวข้อจะนำแนวคิดนี้มาจัดในรูปแบบเชิงสถิติศาสตร์ ซึ่งจะจัดให้มีการตรวจสอบกลุ่มของเอกสารหนึ่งๆ และมีการค้นหาว่าจะมีหัวข้อใดได้บ้างและหัวข้อใดที่อยู่ในเอกสารแต่ละฉบับอย่างพอๆ กัน ทั้งนี้ วิธีที่สามารถนำมาใช้ในการสร้างแบบจำลองหัวข้อ (topic model) ได้แก่ การจัดสรรหัวข้อแฝง (Latent Dirichlet Allocation: LDA)

1.2 การจัดสรรหัวข้อแฝง

การจัดสรรหัวข้อแฝง (Latent Dirichlet Allocation: LDA) (Michael W. Berry and Jacob Kogan, 2010) หมายถึง แบบจำลองความน่าจะเป็น สำหรับข้อมูลที่ไม่ต่อเนื่อง เช่น คลังข้อมูล (Corpus) ซึ่งมีแนวคิดพื้นฐานมาจากเอกสาร สามารถแทนด้วยการรวมตัวกันของหัวข้อแฝงหรือหัวข้อที่ซ่อนอยู่ (Latent) ในเอกสาร ซึ่งแต่ละหัวข้อมีการกระจายตัวของคำ โดย LDA ได้มีการนำมาใช้กันอย่างกว้างขวางในกระบวนการจัดประเภทของข้อมูล (classification) และการกรองข้อมูลแบบพึ่งพาผู้ใช้ร่วม (collaborative filtering) ดังภาพที่ 2.2



ภาพที่ 2.2 แบบจำลองการจัดสรรหัวข้อแฝง (D. Blei, A. Ng and M. Jordan, 2003)

ตารางที่ 2.1 รายการสัญลักษณ์ของการจัดสรรหัวข้อแฝง

สัญลักษณ์	**คำอธิบาย**
α	เป็นพารามิเตอร์ของ Dirichlet ก่อนหน้า ที่จะมีการกระจายตัวของหัวข้อสำหรับแต่ละเอกสาร
θ_d	เป็นการกระจายตัวของหัวข้อสำหรับเอกสาร d
D	เป็นจำนวนเอกสาร
N_d	เป็นจำนวนคำในเอกสาร d
Z_d	เป็นหัวข้อของคำในเอกสาร d (ใช้ Z_i สำหรับแต่ละหัวข้อในเอกสาร)
W_d	เป็นเวคเตอร์ของคำที่ใช้ในเอกสาร d (ใช้ W_i สำหรับแต่ละคำในเอกสาร)
β	เป็นพารามิเตอร์ขั้นสูงของ Dirichlet ก่อนหน้า ที่จะมีการกระจายตัวของคำสำหรับแต่ละหัวข้อ
ϕ	เป็นการกระจายตัวของคำสำหรับหัวข้อ k

จากภาพที่ 2.2 และตารางที่ 2.1 แสดงการกระจายของ LDA โดยทั่วไปในกรณีต่างๆ ดังนี้

1. เมตริกซ์ของเอกสารในมิติที่ k ของจำนวนหัวข้อสำหรับการกระจายตัวของ Dirichlet ได้ถูกกำหนดไว้แน่นอนตายตัว

2. ค่า Dirichlet ก่อนหน้าสำหรับ α และ β ของการกระจายตัวของเอกสารและหัวข้อได้ถูกกำหนดไว้เป็นพารามิเตอร์ 2 ตัวหลัก

3. คำแต่ละคำในกลุ่มเอกสาร d ได้ถูกแสดงเป็นเวคเตอร์ของคำที่ใช้ในเอกสาร (W_d) โดย $W_d = \{W_1, \ldots, W_{Nd}\}$ ในขณะที่ N_d เป็นจำนวนคำในเอกสารที่เกี่ยวข้อง d โดยมีข้อสังเกตว่า N_d เป็นค่าอิสระไม่ขึ้นกับตัวแปรอื่นๆ (θ_d และ Z)

ดังนั้นกระบวนการทำงานของ LDA มีขั้นตอนดังนี้ (Z. Jia, et. al., 2013)

1) เขียนพหุนาม K (ϕ_k) จาก Dirichlet ก่อนหน้า β สำหรับแต่ละหัวข้อ k

2) เขียนพหุนาม D (θ_d) จาก Dirichlet ก่อนหน้า α สำหรับแต่ละเอกสาร d

3) สำหรับแต่ละเอกสาร d ในคลังข้อมูลและในแต่ละคำ w_i ในเอกสาร ให้ทำสิ่งต่อไปนี้

   a. เขียนหัวข้อ z_i จากพหุนาม $\theta_d : (p(z_i | \alpha))$

   b. เขียนคำ w_i จากพหุนาม $\phi_k: (p(w_i | z_i, \beta))$

อัลกอริทึมแบบจำลองการจัดสรรหัวข้อแฝงสามารถทำการเพิ่มประสิทธิภาพของการจัดกลุ่ม (Clustering) การค้นหา (Searching) การเรียงลำดับ (Sorting) การสำรวจ (Exploring) การทำนาย (Predicting) และการสรุป (Summarizing) คลังข้อมูลขนาดใหญ่ของเอกสารได้

1.3 เหมืองข้อความ

1.3.1 เหมืองข้อความ (Text mining) (Michael W. Berry and Jacob Kogan, 2010) หมายถึงกระบวนการในการค้นหารูปแบบ โมเดล ความสัมพันธ์ หรือแนวทาง ที่ซ่อนอยู่ในข้อความจำนวนมาก โดยอาศัยหลักการทางสถิติมาช่วยในการแยกหรือจัดกลุ่มข้อความ ซึ่งกลายเป็นองค์ความรู้ใหม่ที่ซ่อนอยู่ในข้อความเหล่านั้น ทำให้สามารถนำความรู้นั้นไปพยากรณ์สิ่งที่อาจเกิดขึ้น หรือวางแผนแนวทาง หรือดูแนวโน้มในอนาคตที่อาจเกิดขึ้น

หลังจากวิเคราะห์เหมืองข้อความเหล่านั้น ซึ่งได้ผลออกมาเป็นความรู้ที่ได้จากการทำเหมืองข้อความแบ่งออกได้เป็น 3 แบบหลักๆ ดังนี้

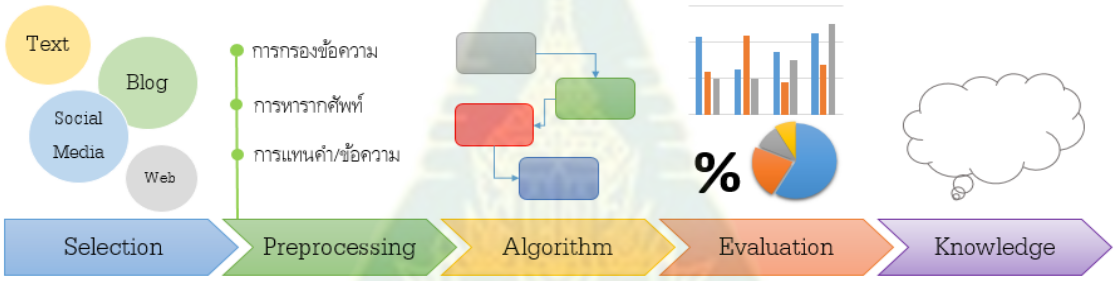
1) การสรุปเอกสารข้อความ คือการลดความซับซ้อนของข้อความขนาดใหญ่ในเอกสาร หรือข้อมูลที่เตรียมไว้ โดยไม่ทำให้ความหมายหรือความสำคัญที่อยู่ในข้อความนั้นเปลี่ยนแปลงไป สามารถทำได้ด้วยการดึงข้อความจากต้นฉบับ กับแบ่งทำเป็นบทคัดย่อ เป็นวิธีพื้นฐานในการจัดลำดับความสำคัญของข้อมูล

2) การแบ่งกลุ่มเอกสารข้อความ คือการจัดแบ่งเอกสารที่มีความคล้ายกันออกเป็นกลุ่มๆ โดยใช้คุณลักษณะหรือคุณสมบัติของข้อความที่มีความหมายใกล้เคียงกันหรือเป็นคำเดียวกันให้อยู่กลุ่มเดียวกัน เพื่อค้นหาว่าจากเอกสารจำนวนมหาศาลทั้งหมดแบ่งออกเป็นกลุ่มใดบ้าง โดยการกำหนดจำนวนกลุ่มไว้ก่อน เมื่อจัดกลุ่มเรียบร้อยจึงดูรายละเอียดคุณลักษณะในแต่ละกลุ่ม หากกลุ่มใดที่สามารถแยกกลุ่มได้อีก จะทำการจัด

กลุ่มใหม่อีกครั้ง เทคนิคการแบ่งกลุ่มข้อมูลลักษณะนี้ เช่น K-Mean, Unsupervised Learning Neural Networks, Hierarchical Clustering เป็นต้น

3) การสรุปความคิดเห็นเอกสารข้อความ คือการนำข้อความมาแยกประเภทว่าเป็น ความคิดเห็น (Opinion) หรือข้อเสนอแนะ (Recommendation) ซึ่งในความคิดเห็นของข้อความจะมีความรู้สึก (Sentiment) แฝงอยู่ด้วย ผลลัพธ์ที่ได้สามารถเป็นได้ทั้งในทางบวก, ทางลบ หรือเป็นกลาง แต่สำหรับข้อเสนอแนะเป็นการแสดงให้เห็นว่าส่วนใหญ่ชอบหรือไม่ชอบในสิ่งใด และมีความต้องการในสิ่งใด เทคนิคที่นำมาใช้ในข้อมูลลักษณะนี้ เช่น Decision Tree, Naïve Bayes หรือ Neural Network เป็นต้น

1.3.2 ขั้นตอนการทำเหมืองข้อความ สำหรับขั้นตอนการทำเหมืองข้อความ สามารถแบ่งออกได้เป็น 5 ขั้นตอนหลัก (การวิเคราะห์ข้อความ (Text mining) เบื้องต้นด้วย RapidMiner Studio 7, 2017) ได้แก่ การเลือกข้อมูล การเตรียมข้อมูล การเลือกใช้อัลกอริทึม การประเมินผลและการนำองค์ความรู้ไปใช้ รายละเอียด ดังภาพที่ 2.3



ภาพที่ 2.3 ขั้นตอนการทำเหมืองข้อความ (ที่มา : https://gallery.azure.ai/Experiment/Text-Classification-Step-2-of-5-text-preprocessing-2)

1) การเลือกข้อมูล (Selection) คือขั้นตอนของการเลือกข้อความหรือเอกสารที่ต้องการมาวิเคราะห์ เช่น ข้อความที่เป็นความคิดเห็นบนโซเชียลมีเดีย ข้อความที่เป็นบทวิจารณ์บนบล็อคหรือเว็บบอร์ด เป็นต้น

2) การเตรียมข้อมูล (Preprocessing) คือขั้นตอนในการเตรียมข้อมูลเพื่อนำเข้าโปรแกรมคอมพิวเตอร์ สำหรับขั้นตอนนี้จะแบ่งออกเป็น 4 ขั้นตอน คือ

- การกรองข้อมูล (Filtration) คือการตัดข้อความออกมาเป็นกลุ่มคำ วลี หรือประโยค โดยวิธีทั่วไปที่นิยมคือการตัดออกมาเป็นคำๆ สำหรับภาษาอังกฤษจะใช้วิธีในการหาช่องว่าง เนื่องจากคำแต่ละคำของภาษาอังกฤษจะถูกแบ่งด้วยช่องว่างเป็นปกติ เช่นประโยค “I love you” จะกลายเป็น “I | love | you| “
- การกำจัดคำหยุด (Stop word) คือ ในข้อความหลายข้อความจะมีคำที่ไม่มีผลต่อประโยค เช่น เช่นประโยค “I love you” จะกลายเป็น “love” ซึ่ง “I”และ “you” จะถูกตัดทิ้ง

- การหารากศัพท์ (Stemming) คือ การจัดกลุ่มคำที่มีความหมายใกล้เคียงกันให้นับเป็นคำเดียวกัน เช่น “love”, “loves” หรือ “lovely”
- การแทนคำ คือ การแปลงจากข้อมูลแบบไม่มีโครงสร้าง (ตัวอักษร) ให้อยู่ในรูปแบบของข้อมูลแบบมีโครงสร้าง (ตัวเลข) เพื่อให้สามารถนำไปวิเคราะห์ในโปรแกรมคอมพิวเตอร์ได้ เช่น การแทนคำด้วย Binary occurrence คือการแทนคำที่เกิดขึ้นในเอกสารนั้นๆ โดยใช้เลข Binary คือเลข 0 กับ 1 ในการกำหนดการมีอยู่ของคำในเอกสาร

3) การเลือกใช้อัลกอริทึม (Algorithm) คือขั้นตอนในการเลือกอัลกอริทึมการทำเหมืองข้อมูล (Data mining) มาใช้ในการวิเคราะห์ เป็นการเลือกเครื่องมือหรือโปรแกรมที่มาช่วยในการวิเคราะห์และเลือกวิธีการ (Algorithm) นำมาใช้ในการประมวลผล เช่น การหากฎความสัมพันธ์ การจัดกลุ่มข้อความ การสร้างแผนผังต้นไม้ตัดสินใจหรือการหาความน่าจะเป็น เป็นต้น

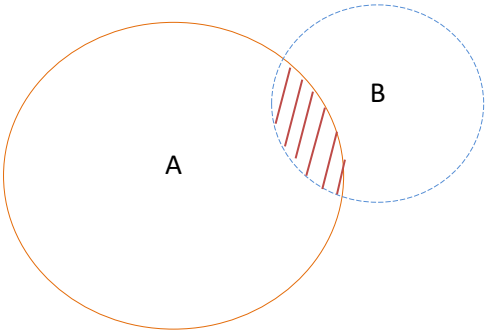
4) การประเมินผล (Evaluation) คือ การประเมินประสิทธิภาพขององค์ความรู้หรือโมเดลที่ได้จากการนำอัลกอริทึมมาใช้ในการวิเคราะห์ว่ามีความถูกต้องแม่นยำ และมีประสิทธิภาพได้ดีตามเกณฑ์ที่ตั้งไว้แต่แรกหรือไม่ โดยทั่วไปนิยมให้ค่าความถูกต้องแม่นยำอยู่ที่ร้อยละ 75 - 80 เป็นผลลัพธ์ที่ยอมรับได้

5) การนำองค์ความรู้ไปใช้ (Knowledge) คือ ส่วนสุดท้ายซึ่งเป็นการนำรูปแบบหรือโมเดลที่ได้ไปใช้งานจริงตามที่ตั้งวัตถุประสงค์ไว้ตั้งแต่ขั้นตอนแรก ในขั้นตอนนี้จะเป็นการนำไปประยุกต์ใช้งานในด้านต่างๆ เช่น ด้านธุรกิจ ด้านการให้บริการ ด้านการศึกษา ด้านการวิจัยและด้านโซเชียลมีเดีย เป็นต้น

1.4 ทฤษฎีของเบย์

ทฤษฎีของเบย์ (Bayes' Theorem) คือการเรียนรู้เบย์ใช้ทฤษฎีของเบย์ในการจำแนกประเภทข้อมูลที่อาศัยหลักการของการหาความน่าจะเป็นของเหตุการณ์ที่เกิดขึ้นจากชุดข้อมูล (Training set) มาใช้คาดการณ์ผลลัพธ์ของข้อมูลทดสอบ (Test set) ภายใต้พื้นฐานทฤษฎีความน่าจะเป็นของเบย์ และการสมการของการเรียนรู้เบย์อย่างง่ายในการสกัดองค์ความรู้จากข้อมูลที่กำหนด

1.4.1 ทฤษฎีของเบย์ โทมัส เบย์ (Thomas Bayes) (J. Han and M. Kamber, 2006) ได้คิดค้นทฤษฎีของเบย์ (Bayes’ Theorem) โดยใช้หลักการของความน่าจะเป็นแบบมีเงื่อนไขภายใต้สมมุติฐานที่เหตุการณ์ใดๆ ที่เกิดขึ้นประกอบด้วยเหตุการณ์ที่ไม่สามารถเกิดขึ้นได้พร้อมกันมาพัฒนาเป็นทฤษฎีเบย์ดังแสดงในภาพที่ 2.4



ภาพที่ 2.4 หลักการของทฤษฎีของเบย์จากความน่าจะเป็นของเหตุการณ์ *E* (Event) บน A_i จำนวน *k* เหตุการณ์ที่ไม่เกิดขึ้นพร้อมกัน (R. Lu et. al., 2010)

จากภาพที่ 2.4 ถ้าให้ *U* แทนเอกภพสัมพัทธ์ (Universe) ประกอบด้วยเหตุการณ์ *A* ใดๆ จำนวน *k* เหตุการณ์ จาก $A_1, A_2, \ldots, A_n$ โดยที่แต่ละเหตุการณ์ A_i เป็นเหตุการณ์ที่ไม่เกิดขึ้นพร้อมกัน และให้ *E* เป็นเหตุการณ์ (Event) หนึ่งที่เกิดขึ้นในปริภูมิตัวอย่างที่เกิดจากการทดลองเดียวกันนี้และต้องเป็นส่วนหนึ่งของเหตุการณ์ A_i โดยที่ *i* มีเหตุการณ์ที่สามารถเกิดขึ้นได้พร้อมกันในปริภูมิตัวอย่างจำนวน *k* เหตุการณ์จาก *i = 1, 2, 3,.., k* จะสามารถคำนวณความน่าจะเป็นแบบมีเงื่อนไขของเหตุการณ์หนึ่งใน A_i เมื่อเหตุการณ์ *E* เกิดขึ้นแล้ว ได้ดังสมการที่ (2.1)

$$P(A_i \mid B) = \frac{P(B|A_i)\,P(A_i)}{P(B)} \qquad (2)$$

โดยที่ $P(A_i)$ แทนความน่าจะเป็นก่อนหน้าของสมมติฐาน

$P(B)$ แทนความน่าจะเป็นก่อนหน้าของชุดข้อมูลตัวอย่าง P(B)

$P(A_i \mid B)$ แทนความน่าจะเป็นของ A_i เมื่อรู้ *B*

$P(B \mid A_i)$ แทนความน่าจะเป็นของ *B* เมื่อรู้ A_i

1.4.2 ตัวอย่างการทำงานของเนอีฟเบย์ (Naïve Bayes) เทคนิคการจำแนกประเภทที่ใช้ทฤษฎีเบย์เป็นพื้นฐานและอาศัยหลักการของการหาความน่าจะเป็นของเหตุการณ์ที่เกิดขึ้นจากชุดข้อมูล (Training set) มาใช้คาดการณ์ผลลัพธ์ของข้อมูลทดสอบ (Test set) ภายใต้พื้นฐานทฤษฎีความน่าจะเป็นของเบย์

TID	HOME OWNER	MARITAL STATUS	ANNUAL INCOME	DEFAULTED BORROWER
1.	YES	SINGLE	125K	NO
2.	NO	MARRIED	100K	NO
3.	NO	SINGLE	70K	NO
4.	YES	MARRIED	120K	NO
5.	NO	DIVORCED	95K	YES
6.	NO	MARRIED	60K	NO
7.	YES	SINGLE	220K	NO
8.	NO	MARRIED	85K	YES
9.	NO	SINGLE	75K	NO
10.	NO	SINGLE	90K	YES

(ก)

P(Home Owner=YES|No) = 3/7

P(Home Owner=No|No) = 4/7

P(Home Owner=YES|YES) = 0

P(Home Owner=No|YES) = 1

P(Married Status=Single|No) = 2/7

P(Married Status=Divorced|No) = 1/7

P(Married Status=Married|No) = 4/7

P(Married Status=Single|YES) = 2/3

P(Married Status=Divorced|YES) = 1/3

For Annual Income:

if class = No: sample mean = 110

sample variance = 2975

if class = YES : sample mean = 90

sample variance = 25

(ข)

ภาพที่ 2.5 ตัวอย่างเนอีฟเบย์เพื่อวิเคราะห์การอนุมัติเงินกู้ (R. Lu et. al., 2010)

ภาพที่ 2.5 (ก) แสดงตัวอย่างการใช้เนอีฟเบย์จำแนกประเภทของลูกค้าที่ควรอนุมัติเงินกู้ โดยชุดข้อมูล (Training set) ประกอบด้วย 3 แอตทริบิวต์คือ ข้อมูลการเป็นเจ้าของบ้าน (Home owner) สถานภาพสมรส (Marital status) และรายได้ต่อปี (Annual income) เพื่อคำนวณความน่าจะเป็นของการอนุมัติเงินกู้ (Defaulted borrower) ดังแสดงในตัวอย่าง ภาพที่ 2.5 (ข) เมื่อสร้างโมเดลการจำแนกประเภทข้อมูลลูกค้าได้แล้ว หากมีข้อมูลทดสอบ เช่น มีลูกค้าท่านหนึ่งเป็นเจ้าของบ้าน (Home owner=yes) สมรสแล้ว (Marital status=yes) และมีรายได้ต่อปี 130K (Annual income) โมเดลจะสามารถพยากรณ์การอนุมัติเงินกู้ (Defaulted borrower) ให้ลูกค้าได้

1.4.3 วิธีการเรียนรู้แบบนาอีฟเบย์ (Naïve Bayesian Learning) เป็นวิธีการจำแนกประเภทข้อมูล (Data classification) ที่มีประสิทธิภาพรูปแบบหนึ่ง ที่ใช้หลักความน่าจะเป็นซึ่งอยู่บนพื้นฐานของทฤษฎีเบย์และมีสมมติฐานจากการเกิดเหตุการณ์ต่างๆ เป็นอิสระต่อกัน โดยการเรียนรู้แบบเบย์นี้ เหมาะกับกรณีของกลุ่มตัวอย่างที่มีจำนวนมากและคุณสมบัติหรือแอทริบิวต์ (Attribute) ของตัวอย่างไม่ขึ้นต่อกัน และมีการจำแนกประเภทเบย์ โดยมักนำวิธีการเรียนรู้แบบเบย์ไปประยุกต์ใช้งานด้านการจำแนกประเภทข้อความ (Text classification) อีกทั้งขั้นตอนวิธีในการทำงานไม่ซับซ้อนเหมือนการเรียนรู้ในรูปแบบอื่น

หากกำหนดให้ความน่าจะเป็นของข้อมูลภายใต้สมมุติฐานที่ข้อมูลในกลุ่ม V_i แต่ละตัวเป็นอิสระต่อกันสำหรับข้อมูล X ที่มีคุณสมบัติ n ตัว โดยที่ $X = \{a_1, a_2, ..., a_n\}$ หรือ เรียกว่า $P(a_1, a_2, a_3,... \mid v_j)$ โดยที่

$$P(a_1, a_2....a_n \mid v_j) = \prod_i P(a_i \mid v_j) \qquad (3)$$

จากสมการที่ (3) คำตอบของ $P(a_1, a_2, a_3,... \mid v_j)$ หมายถึงกลุ่ม (class) ของผลลัพธ์ V_i ใดๆ โดยมักเป็นกลุ่มที่มีค่าความน่าจะเป็นที่มากที่สุดที่ได้จากการคำนวณจากสมการที่ (3) และใช้เป็นคำตอบสำหรับการจำแนกประเภทของข้อมูล

คำนวณความน่าจะเป็นของคำตอบ $(P(v_j))$ ที่พบในแต่ละกลุ่มหรือคลาสจากการนำค่า $P(a_1, a_2, a_3,... \mid v_j)$ ในสมการที่ (3) มาคูณความน่าจะเป็นของกลุ่มนั้นๆ เพื่อหาค่า V_{NB} จากสมการ

$$v_{NB} = \arg\max_{vj \in V} P(v_j) \times \prod_{i=1}^{n} P(\mathrm{a}_i \mid v_j) \qquad (4)$$

การเรียนรู้แบบเบย์นี้เป็นการเรียนรู้ที่ต่อเนื่องในแต่ละช่วงเวลา จะมีการเรียนรู้ที่เปลี่ยนแปลงไปเนื่องจากตัวแบบข้อข้อมูลจะถูกปรับเปลี่ยนค่าไปตามค่าของตัวอย่างใหม่ที่เข้ามาในแต่ละช่วงเวลา โดยรวมเข้ากับความรู้เดิมที่วิธีการทำนายค่ากลุ่มหรือคลาส โดยมีขั้นตอนวิธีหรืออัลกอริทึมในการทำงานที่สามารถปรับใช้ได้กับข้อมูลในหลายรูปแบบทั้งแบบชนิดตัวเลขและข้อความ

1.4.4 ขั้นตอนวิธีของการเรียนรู้แบบเบย์

1. คำนวณความน่าจะเป็นของคำตอบที่พบในแต่ละกลุ่มหรือคลาส จากการนำค่า $P(a_1, a_2, a_3,... \mid v_j)$ ในสมการที่ (3) มาคูณความน่าจะเป็นของกลุ่มนั้นๆ $(P(v_j))$ เพื่อหาค่า V_{NB} ในสมการที่ (4)

2. นำค่าความน่าจะเป็นที่ได้มาเปรียบเทียบกัน กลุ่มใดที่มีค่าความน่าจะเป็นสูงสุดถือเป็นคำตอบหรือค่ากลุ่มของข้อมูล

1.5 อาชญากรรมไซเบอร์และการกลั่นแกล้งทางอินเทอร์เน็ต

อาชญากรรมไซเบอร์และการกลั่นแกล้งทางอินเทอร์เน็ต (CyberCrime and Cyberbullying) หมายถึง การกระทำผิดทางอาญาใด ๆ ที่เกี่ยวข้องกับคอมพิวเตอร์และระบบเครือข่ายคอมพิวเตอร์ (เช่น อินเตอร์เน็ต) หรือการใช้คอมพิวเตอร์และระบบเครือข่ายคอมพิวเตอร์เพื่อกระทำผิดทางอาญา เช่น การปลอมแปลง การก่อการร้าย การขู่เข็ญ การทำลามกอนาจาร การหลอกลวงเงินและทรัพย์สิน การทำลายเปลี่ยนแปลงหรือขโมยข้อมูลต่างๆ และแม้แต่การทำให้สูญเสียชีวิต เป็นต้น หนึ่งในปัญหาจากอาชญากรรมไซเบอร์คือการกลั่นแกล้งทางอินเทอร์เน็ต (Cyberbullying) การใช้อินเทอร์เน็ตเป็นเครื่องมือหรือช่องทางเพื่อก่อให้เกิดการคุกคาม ล่อลวงและการกลั่นแกล้งบนโลกอินเทอร์เน็ต ซึ่งสามารถเป็นทั้งผู้กระทำและผู้ถูกกระทำ โดยจุดมุ่งหมายจะเป็นกลุ่มเด็กจนถึงเด็กวัยรุ่น

ภาพที่ 2.6 อาชญากรรมไซเบอร์และการกลั่นแกล้งทางอินเทอร์เน็ต (ที่มา : https://health.kapook.com/view150050.html)

จากผลสำรวจจากสำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (Cyberbullying คืออะไร? แนวทางการป้องกันการคุกคามทางอินเทอร์เน็ต, 2016) พบว่าอัตราการเข้าถึงอินเทอร์เน็ต ของกลุ่มที่ใช้งานอินเทอร์เน็ตมากที่สุดคือ เด็กและเยาวชน อายุ 5 -28 ปี และใช้อินเทอร์เน็ตมากที่สุดถึง เกือบ 8 ชม.ต่อวัน คิดเป็นร้อยละ 75 สำหรับเด็กและเยาวชนไทย เจอภัยคุกคาม ล่อลวงและการกลั่นแกล้งโรงเรียนและบนโลกอินเทอร์เน็ต และเป็นอันดับต้นๆของเอเชีย คิดเป็นร้อยละ 80 และมากกว่าร้อยละ 59 ของเด็กไทย พบว่า “เคยเป็นส่วนหนึ่งของการกลั่นแกล้งทางอินเทอร์เน็ต”

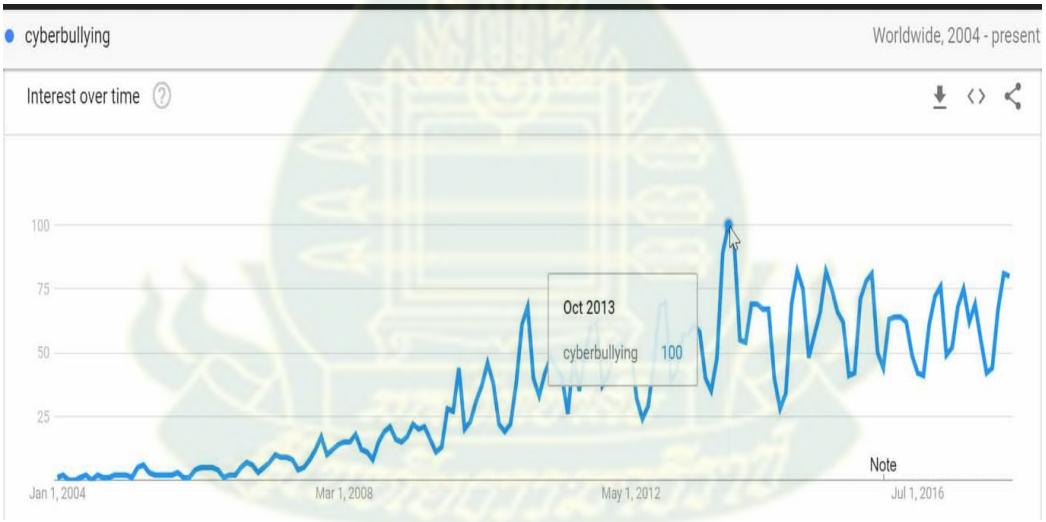
ประเภทการกลั่นแกล้งทางอินเทอร์เน็ต แบ่งได้เป็น 4 ประเภทหลักได้แก่

1) ล่วงละเมิดทางเพศ (Sexual harassment) เช่นการสนทนาแล้วชวนให้มีเพศสัมพันธ์ โดยเหยื่อยินยอมหรือรู้เท่าไม่ถึงการณ์

2) หลอกลวงเงิน (Money Mule Scams) เช่นการสนทนาแล้วสอบถามหมายเลขบัญชีธนาคารหรือหลอกให้เหยื่อโอนเงินมาให้

3) พยายามฆ่าตัวตาย (Suicide attempts) เช่นการสนทนาแล้วทำให้เหยื่อเครียด คิดมาก ไม่สามารถมีชีวิตอยู่ได้และสุดท้ายฆ่าตัวตาย

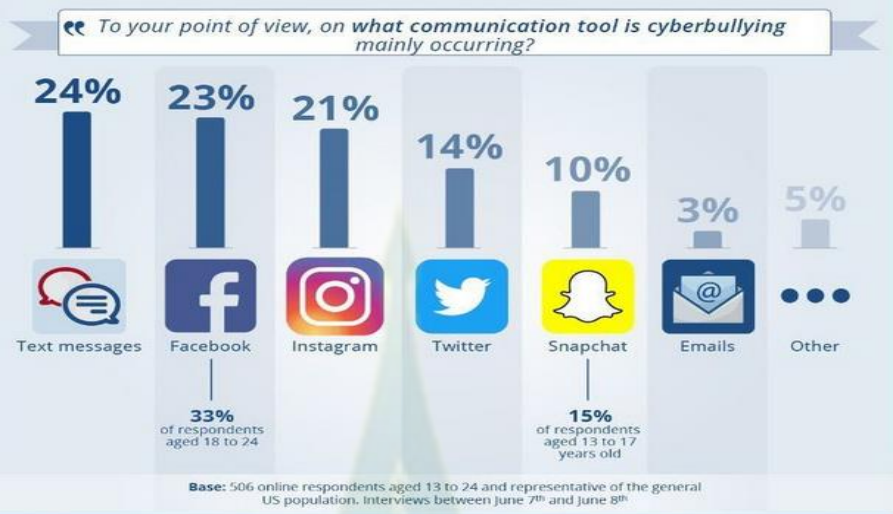
4) ยาเสพติดและเครื่องดื่มแอลกอฮอล์ (Drugs and alcohol abuse) เช่นการสนทนาแล้วทำให้เหยื่อเข้าไปยุ่งเกี่ยวกับยาเสพติดและเครื่องดื่มแอลกอฮอล์ ทำให้เกิดปัญหาสังคมต่อไป



ภาพที่ 2.7 แนวโน้มการกลั่นแกล้งทางอินเทอร์เน็ตทั่วโลก (Cyberbullying Worldwide, 2015)

จากภาพที่ 2.7 แสดงแนวโน้มการกลั่นแกล้งทางอินเทอร์เน็ตทั่วโลก ระหว่างปี ค.ศ. 2004 ถึง 2016 ซึ่งมีแนวโน้มเพิ่มสูงมากขึ้นเรื่อยๆ โดยรูปแบบการกลั่นแกล้งทางอินเทอร์เน็ตได้แก่ การโจมตี ขู่ทำร้าย หรือใช้ถ้อยคำหยาบคาย การคุกคามทางเพศแบบออนไลน์ การแอบอ้างตัวตนของผู้อื่น การแบล็กเมล์ การหลอกลวงหรือการสร้างกลุ่มในโซเชียลเพื่อโจมตีโดยเฉพาะ สำหรับสื่อที่ใช้ในการกลั่นแกล้งทางอินเทอร์เน็ต ส่วนใหญ่

ได้แก่ การส่งข้อความผ่านสมาร์ทโฟน การแชทบนสื่อสังคมออนไลน์ เช่น เฟซบุ๊ก (Facebook) ไลน์ (Line) อินสตาแกรม (Instagram) ทวิตเตอร์ (Twitter) หรืออีเมล (Email) ดังภาพที่ 2.8



ภาพที่ 2.8 สื่อที่ใช้ในการกลั่นแกล้งทางอินเทอร์เน็ต (Cyberbullying Worldwide, 2015)

จากภาพที่ 2.8 สื่อที่ใช้ในการกลั่นแกล้งทางอินเทอร์เน็ต เช่นการส่งข้อความผ่านสมาร์ทโฟน คิดเป็นร้อยละ 24 การแชทผ่านเฟซบุ๊กคิดเป็นร้อยละ 23 อินสตาแกรม (Instagram) คิดเป็นร้อยละ 21 ทวิตเตอร์ (Twitter) คิดเป็นร้อยละ 14 สแนปชาร์ต (Snapchat) คิดเป็นร้อยละ 10 และอีเมล (Email) คิดเป็นร้อยละ 3 ตามลำดับ ซึ่งจะเห็นได้ว่าจะใช้ข้อความสนทนาในการกลั่นแกล้งทางอินเทอร์เน็ต

การป้องกันอาชญากรรมไซเบอร์และการกลั่นแกล้งทางอินเทอร์เน็ตเบื้องต้น มีดังนี้

1) สอนลูกๆ ว่าอย่าไว้ใจคนแปลกหน้า โดยเฉพาะในโลกออนไลน์ ใครมาขอเป็นเพื่อนต้องตรวจสอบให้ดี หากไม่รู้จักก็ไม่ควรตอบรับคำขอเป็นเพื่อนนั้น

2) คอยสอดส่องว่าลูกจะไปไหน กับใคร หรือเพื่อนที่ลูกคุยด้วยหรือแชทด้วยเป็นใคร

3) สอนลูกให้เก็บข้อมูลส่วนตัวของตัวเองให้ดี โดยเฉพาะกับคนแปลกหน้าและคนที่ไม่สนิทสนม ไม่ควรเปิดเผยข้อมูลส่วนตัวอย่างชื่อ ที่อยู่ เบอร์โทรศัพท์ หรืออีเมลส่วนตัว ที่สำคัญควรตักเตือนกับลูกว่าไม่ควรนัดเจอกันส่วนตัวกับเพื่อนในโลกออนไลน์โดยเด็ดขาด

4) ควรกำหนดข้อตกลงกันก่อนที่จะอนุญาตให้ลูกใช้เครื่องมือสื่อสารและโซเชียลมีเดีย เพื่อให้พ่อแม่สามารถตรวจสอบได้ว่าลูกใช้โซเชียลมีเดียยังไง คุยกับใครบ้าง หรือมีความผิดปกติอะไรในนั้นหรือไม่

5) พ่อแม่ควรสร้างความสัมพันธ์อันดีให้กับลูก เพื่อให้ลูกไว้วางใจมากพอจะบอกเล่าทุกเรื่องราวในชีวิตเขาได้ เมื่อมีปัญหาอะไรลูกจะได้กล้าขอคำปรึกษา

6) ติดตั้งแอปพลิเคชันที่ทำหน้าที่ติดตามหรือป้องปรามข้อความสนทนาในการกลั่นแกล้งทางอินเทอร์เน็ต

2. งานวิจัยที่เกี่ยวข้อง

การทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ ซึ่งอาชญากรรมไซเบอร์ (CyberCrime) เป็นอาชญากรรมใด ๆ ที่เกี่ยวข้องกับคอมพิวเตอร์และเครือข่ายคอมพิวเตอร์ โดยทำให้เหยื่อได้รับความเสียหาย ผู้กระทำได้รับผลประโยชน์ตอบแทน และหากเหยื่อเป็นเด็กเยาวชนและผู้หญิงทำการแลกเปลี่ยนข้อความ รูปภาพ หรือข้อมูลสำคัญอาจจะทำให้ตกเป็นเหยื่อได้ง่าย

ดังนั้นงานวิจัยนี้ขอนำเสนอการพัฒนาระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ สกัดคำ ประโยค วลี รูปแบบข้อความ คำแฝงและคำกำกวม โดยจำแนกประเภทของเอกสาร (Text Classification) ตามวิธีการคัดเลือกคุณลักษณะ จากกลุ่มหัวข้อการสนทนาด้วยวิธีการสร้างแบบจำลองโดยอาศัยเรียนรู้เบย์อย่างง่าย (Bayesian Learning) จากข้อมูลการฝึกอบรม (Training Data) และการพยากรณ์ข้อมูลจากการสนทนาในสถานการณ์แบบต่างๆ จากข้อมูลทดสอบ (Testing Data) โดยใช้อาปาเช่ สตอมม์ (Apache Storm) (J. Zeng et. al., 2016) เป็นระบบการประมวลผลข้อมูลแบบกระจาย (Distributed Data Processing System) เหมาะกับข้อมูลขนาดใหญ่ (Big Data) เพื่อทำการวิเคราะห์ข้อมูลแบบทันทีทันใดหรือเวลาจริง โดยลำดับการทำงานของสตอร์ม (Storm) (D. Blei et. al., 2010), (G. Bettina and H. Kurt, 2011) และ (N. Welly et. al., 2012) ประกอบด้วย 3 ส่วนได้แก่ 1) ส่วนนำข้อมูลเข้า (Input) นำข้อมูลแบบต่อเนื่องเสมือนกระแสน้ำที่ไหลเข้าสู่ระบบหรือข้อมูลสตรีมมิ่ง (Data Streaming) เพื่อทำการกระจายการประมวลผล ไปยังโหนด (Node) ต่าง ๆ บนฮาดูป (Hadoop) ซึ่งจะสามารถประมวลผลข้ามเครื่องกันได้ 2) ส่วนการประมวลผล (Process) ทำการประมวลผลข้อมูลและส่งผ่านข้อมูล และ 3) ส่วนผลลัพธ์ (Output) นำข้อมูลจากส่วนการประมวลผลมาสรุปรวมข้อมูลและแสดงผลลัพธ์ออกมาในเวลาจริง ซึ่งเหมือนกันขั้นตอนของ Reduce ในฮาดูป (W. Romsaiyud, 2014) และ (K.P. Murphy, 2012)

งานวิจัยนี้ใช้ข้อมูลจากหลายแหล่งชุดข้อมูล (Datasets) ส่งเข้ามาในระบบเพื่อประมวลผลข้อมูลสตรีมมิ่ง (Data Streaming) ในเวลาทันทีทันใด โดยไม่ต้องเก็บข้อมูลไว้ก่อนแล้วจึงประมวลผล ทำให้ได้ผลลัพธ์ที่รวดเร็วและตอบสนองความต้องการของผู้ใช้ในทันที ในงานวิจัย (S. Bao et. al., 2011) เป็นการประยุกต์อาปาเช่สตอมม์ (Apache Storm) กับชุดข้อมูลจากทวิตเตอร์ (Twitter) โดยได้พัฒนาการทำงานแบบขนานระหว่างฟังก์ชันการทำงานภายในส่วนประมวลผลที่เรียกว่า “Intra-bolt parallelism of tasks” เพื่อเพิ่มความเร็วในการประมวลผลและส่งเสริมความน่าเชื่อถือของระบบ นอกจากนี้ในงานวิจัย (L. R.Y.K. et. al., 2014) ได้พัฒนาอัลกอริทึมที่นำเทคนิคการทำงานแบบต้นไม้ในแนวตั้ง Hoeffding (Vertical Hoeffding Tree) สำหรับส่งเสริมระบบการตัดสินใจบนอาปาเช่สตอมม์กับชุดข้อมูลจากทวิตเตอร์ (Twitter) ยิ่งไปกว่านั้นในหลายๆ โปรแกรมได้นำอาปาเช่สตอมม์ไปทำงานร่วมกับการทำเหมืองข้อความ (Text Mining) เพื่อหาหัวข้อ (Topic) ใหม่ๆ ได้แก่ โปรแกรมการวิเคราะห์การรักษาความปลอดภัยในโลกไซเบอร์และการตรวจสอบภัยคุกคามสำหรับการตรวจสอบแนวโน้มและการสกัดความผิดปกติของสถานการณ์อาชญากรรม ที่สำคัญใน

งานวิจัยนี้จะใช้อาปาเช่สตอมม์รุ่น 0.9.1 สำหรับการประเมินผลการทดลองในการกระจายเวลาจริงเพื่อการการคำนวณเวลาการประมวลผล

ในงานวิจัย (P. Lian and D. Klein, 2009) ได้ประยุกต์อัลกอริทึมแบบซัปพอร์ตเวกเตอร์แมชีนหรือเอสวีเอ็ม (Support Vector Machine: SVM) เพื่อทำการคัดแยก ตรวจสอบพฤติกรรมการกลั่นแกล้งทางอินเทอร์เน็ตจากชุดข้อมูล CAW2.0 เพื่อใช้ในการป้องกันไม่ให้เด็กหรือผู้ที่ตกเป็นเหยื่อจากการสนทนาบนอินเทอร์เน็ต

นอกจากนี้การสร้างแบบจำลองหัวข้อ (Topic Model) (G. Brynjar et. al.,2012) และ (D. Blei and J. Lafferty, 2006) ได้รับความนิยมในวงกว้างในการจัดกลุ่มของข้อมูลจากค่าความน่าจะเป็น (Probability) โดยแบบจำลองหัวข้อนี้มีพื้นฐานมาจากแนวคิดที่ว่าในเอกสารหนึ่งๆ (Document) เกิดจากการรวมตัวของหลายๆ หัวข้อ (Topics) ซึ่งแต่ละหัวข้อมีการแจกแจงค่าความน่าจะเป็นของคำที่เกิดขึ้นหลายๆ คำในแต่ละหัวข้อ ดังนั้นจึงมีการนำแบบจำลองหัวข้อไปประยุกต์ใช้ในการค้นหาคำหรือหัวข้อ และการสรุปชุดข้อมูล ซึ่งในงานวิจัย และ (D. Blei and J. Lafferty, 2006) ได้พัฒนาแบบจำลองส่วนผสมลำดับชั้น (Hierarchical mixture modeling) เพื่อหารูปแบบที่เฉพาะเจาะจงภายในเอกสาร อย่างไรก็ตามในงานวิจัย (M. Huifang et. al., 2012) ประยุกต์แบบจำลองหัวข้อเพื่อหาหัวข้อที่ซ่อนหรือแฝง (Latent) อยู่ในเอกสาร ที่เรียกว่า Latent Dirichlet Allocation (LDA) และ (D. Blei and J. Lafferty, 2006) และ (D. Blei, 2010) และในบางวิธีได้พัฒนาต่อยอดจาก LDA เป็นกระบวนการที่เรียกว่า Hierarchical Dirichlet Process (HDP) เพื่อกำหนดจำนวนของหัวข้อที่แน่นอนใน LDA แบบจัดเรียงลำดับเพื่อให้พอดีกับข้อมูลที่ได้มาจากโครงสร้างลำดับชั้น นอกจากนี้ ในงานวิจัย (K. Wang and M. M. H. Khan, 2015) ได้ขยายขีดความสามารถของ LDA โดยทำการกำหนดค่าความน่าจะเป็นบนขอบของเอกสาร เพื่อเพิ่มความถูกต้อง ในการกำจัดส่วนที่ไม่เกี่ยวข้องออกไป และในงานวิจัย (N. Bharill et. al., 2016) โดยใช้วิธีการแบบเบย์ (Bayes) ในการพัฒนาแบบจำลอง เพื่อเพิ่มความถูกต้อง แม่นยำและน่าเชื่อถือ

บทที่ 3

วิธีดำเนินวิจัย

การวิจัยเรื่องการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ มีรายละเอียดการดำเนินการวิจัยซึ่งประกอบไปด้วย 3 ขั้นตอนได้แก่ 1) สถาปัตยกรรมภาพรวมการทำงาน 2) ขั้นตอนวิธีการดำเนินงาน และ 3) เครื่องมือที่ใช้ในการวิจัย

1. สถาปัตยกรรมภาพรวมการทำงาน

ระบบที่เสนอถูกออกแบบเป็นโมดูลย่อยๆ เพื่อให้สามารถทำงานได้อย่างอิสระ รวดเร็วและ มีประสิทธิภาพรวมทั้งง่ายต่อการแก้ไข ปรับปรุงเพิ่มเติม โดยการทำงานของระบบเพื่อใช้ในการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ ประกอบด้วย 4 ส่วนหลักๆ คือ 1) การรวบรวมข้อมูล 2) การประมวลผลแบบกระจายข้อมูลด้วยระบบคลัสเตอร์ 3) การดำเนินการจัดสรรหัวข้อแฝง และ 4) ผลลัพธ์การจำแนกประเภทคำทางด้านอาชญากรรมไซเบอร์ ดังแสดงในภาพที่ 3.1

ภาพที่ 3.1 ภาพรวมขั้นตอนการทำงานของการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์

ซึ่งหน้าที่ในแต่ละส่วนมีรายละเอียดดังนี้

1) การรวบรวมข้อมูล เป็นการนำข้อมูลประเภทข้อความจากแหล่งข้อมูลต่าง ๆ ได้แก่ขอบเขตข้อมูลด้านประชากร ขอบเขตข้อมูลด้านเนื้อหาและขอบเขตข้อมูลด้านเวลา รายละเอียดดังต่อไปนี้

- ขอบเขตข้อมูลด้านประชากร ได้แก่ สมาชิกเว็บไซต์ Perverted-justice, Formspring และ MySpace ในปี ค.ศ. 2010
- ขอบเขตข้อมูลด้านเนื้อหา ได้แก่ ข้อความเนื้อหาจากการโพสต์ของเว็บไซต์ perverted-justice.com, ข้อความจาก Formspring และ ข้อความจาก MySpace จำนวนทั้งสิ้น 127,974 ข้อความซึ่งใช้ข้อมูลการฝึกอบรมและการทดสอบข้อมูลที่แยกในสถานการณ์ต่างๆ บนข้อมูลแบบสตรีมมิ่ง

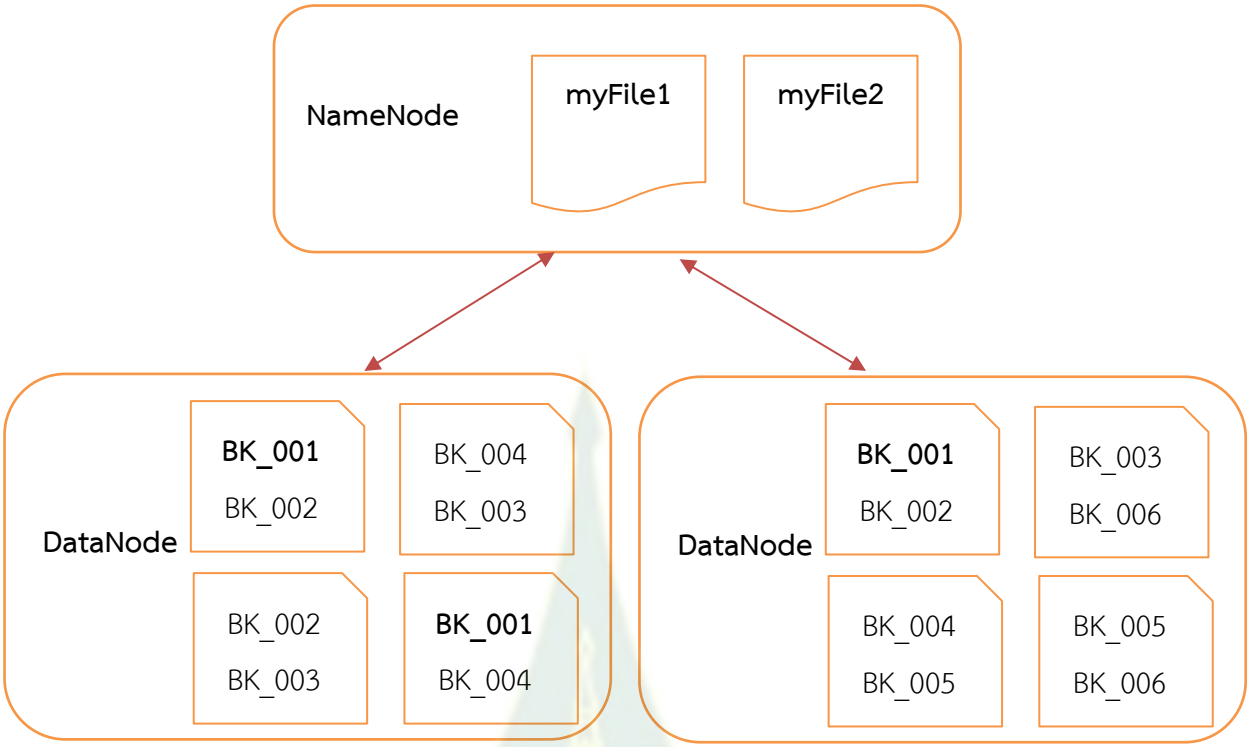
ข้อมูลการฝึกอบรม (Training Data) จำนวน 23,492 โดยแบ่งข้อมูลเป็นแบบ 10-fold cross-validation คือ การแบ่งข้อมูลออกเป็น 10 ส่วน โดยที่แต่ละส่วนมีจำนวนข้อมูลเท่ากัน หลังจากนั้นข้อมูลหนึ่งส่วนจะใช้เป็นตัวทดสอบประสิทธิภาพของโมเดล ทำวนไปเช่นนี้จนครบจำนวนที่แบ่งไว้

- ขอบเขตข้อมูลด้านเวลา ได้แก่ ชุดข้อมูลจริงซึ่งเก็บรวบรวมระหว่างปี พ.ศ. 2549-2557

ตัวอย่างประโยคการสนทนาที่ได้จากการรวบรวมข้อมูล

- mmmm...i'm pre-cumming a little
- are u going to suck my dick too baby
- they wont just take the money back
- l love money because money is god
- lol did u kill ur friend
- it will kill me to hafta wait a week
- Baby, you’re the hot ass in my shot glass.
- Here’s $20. Drink until I’m good looking and then come talk to me.

2) การประมวลผลแบบกระจายข้อมูลด้วยระบบคลัสเตอร์ ทำหน้าที่แบ่งข้อมูลเข้าขนาดใหญ่ซึ่งเก็บในระบบแฟ้มข้อมูลกระจายแบบฮาดูป (HDFS) ออกเป็นส่วนเล็กๆ ที่เรียกว่าบล็อก (Block) ซึ่งมีขนาดตั้งแต่ 64 MB ถึง 256 MB เพื่อให้สามารถกระจายข้อมูลเล็กๆนี้ไปประมวลผลในระบบเครื่องคอมพิวเตอร์คลัสเตอร์ได้ ซึ่งหลักการทำงานของระบบแฟ้มข้อมูลกระจายแบบฮาดูป ประกอบด้วย 2 ส่วนคือ มาสเตอร์โหนดหรือโหนดชื่อ (NameNode) ซึ่งมีแค่โหนดเดียว และ สลาฟโหนดหรือโหนดข้อมูล (DataNodes) ซึ่งสามารถมีหลายโหนด ในการประมวลผลนั้นระบบคลัสเตอร์จะทำหน้าที่ในการตัดคำหรือกลุ่มคำในเอกสาร แล้วใช้เทคนิค Naïve Bayes เพื่อวิเคราะห์เชิงทำนายผลกลุ่มคำที่เกี่ยวข้องกับอาชญากรรมไซเบอร์ ดังภาพที่ 3.2



ภาพที่ 3.2 การประมวลผลแบบกระจายข้อมูลแบบฮาดูป

จากภาพที่ 3.2 เป็นการนำข้อมูลเข้าที่ได้รวบรวมมาจากหลายแหล่งข้อมูลมาประมวลผลแบบกระจายข้อมูลด้วยระบบคลัสเตอร์ โดยฮาดูปจะทำหน้าที่แบ่งข้อมูลเข้าขนาดใหญ่ซึ่งเก็บในระบบแฟ้มข้อมูลกระจายแบบฮาดูป (HDFS) ออกเป็นส่วนเล็กๆ ซึ่งมีขนาดตั้งแต่ 64 MB ถึง 256 MB เพื่อให้สามารถกระจายข้อมูลเล็กๆนี้ไปประมวลผลในระบบเครื่องคอมพิวเตอร์คลัสเตอร์ได้ และมีการทำซ้ำข้อมูล (Data replication) เช่น บล็อกข้อมูลรหัส 001 (BK_001) จะมีการทำซ้ำข้อมูล 3 ข้อมูลที่ DataNode บนฮาดูป เพื่อป้องกันข้อมูลสูญหาย

3) การดำเนินการจัดสรรหัวข้อแฝง ทำหน้าที่ในการจำแนกคำ (Word) คำศัพท์ (Term) หรือวลี (Phase) จากขั้นตอนที่ 1 การรวบรวมข้อมูลเข้า โดยใช้เทคนิคแอลดีเอ (LDA) เพื่อทำการกำหนด top 5-terms จากคำสำคัญของเอกสาร (Document) ในแต่ละกลุ่มของคอมพิวเตอร์คลัสเตอร์ (k) ซึ่ง LDA เป็นเทคนิคที่ทำงานได้เร็วและมีประสิทธิภาพในการทำงานสูง โดยกลุ่มคำเหล่านี้จะถูกจำแนกให้อยู่ในหัวข้อที่เหมาะสมที่สุด ดังภาพที่ 3.3

Topic 1	
Words	**Score**
Baby	0.85
Dick	0.77
Suck	0.70
Cums	0.65
Pussy	0.62

Topic 2	
Words	**score**
Money	0.79
Cash	0.77
Cheque	0.75
God	0.70
Love	0.70

ภาพที่ 3.3 ตัวอย่างหัวข้อ (Topic) และน้ำหนักของคำเรียงตาม top 5- terms

จากภาพที่ 3.3 การกำหนดน้ำหนักของคำเรียงตาม top 5- terms ตามหัวข้อ โดยใช้หลักการ LDA เช่น ในหัวข้อที่ 1 (topic 1) มี 5 คำที่มีการจัดเรียงค่าน้ำหนักจากค่าความน่าจะเป็นของคำในหัวข้อเอกสาร และเอกสารในแต่ละคลัสเตอร์ โดยคำว่า Baby มีค่าน้ำหนักเท่ากับ 0.85 (มีค่าสูงสุด) คำว่า Dick มีค่าน้ำหนักเท่ากับ 0.77 คำว่า Suck มีค่าน้ำหนักเท่ากับ 0.70 คำว่า Cums มีค่าน้ำหนักเท่ากับ 0.65 และคำว่า Pussy มีค่าน้ำหนักเท่ากับ 0.62 ตามลำดับ

4) ผลลัพธ์การจำแนกประเภทคำทางด้านอาชญากรรมไซเบอร์ ทำหน้าที่ในการวิเคราะห์กลุ่มคำที่อยู่ในอาชญากรรมไซเบอร์ โดยใช้แบบจำลอง (Model) ซึ่งใช้หลักการพัฒนาโปรแกรมแบบแมปและรีดิวซ์ด้วยภาษาจาวาบนระบบฮาดูป เพื่อสร้างตัวจำแนกนาอีฟเบย์ (Naïve Bayes Classifier) ในการคำนวณหาค่าความน่าจะเป็นและเวคเตอร์น้ำหนักของคำในแต่ละประโยค จากนั้นทำการวิเคราะห์เชิงทำนายผลลัพธ์ เพื่อจำแนกประเภทการกลั่นแกล้งทางอินเทอร์เน็ตเป็น 4 ประเภทได้แก่ 1) ล่วงละเมิดทางเพศ (Sexual harassment) 2) หลอกลวงเงิน (Money Mule Scams) 3) พยายามฆ่าตัวตาย (Suicide Attempts) และ 4) ยาเสพติดและเครื่องดื่มแอลกอฮอล์ (Drug and Alcohol Abuse)

ตัวอย่างผลลัพธ์การดำเนินการจำแนกตามประเภท

ตัวอย่างประโยคการสนทนา	**ประเภทการกลั่นแกล้งทางอินเทอร์เน็ต**
mmmm...i'm pre-cumming a little	ล่วงละเมิดทางเพศ
are u going to suck my dick too baby	ล่วงละเมิดทางเพศ

ตัวอย่างประโยคการสนทนา	ประเภทการกลั่นแกล้งทางอินเทอร์เน็ต
they wont just take the money back	หลอกลวงเงิน
l love money because money is god	หลอกลวงเงิน
lol did u kill ur friend	พยายามฆ่าตัวตาย
it will kill me to hafta wait a week	พยายามฆ่าตัวตาย
Baby, you’re the hot ass in my shot glass.	ยาเสพติดและเครื่องดื่มแอลกอฮอล์
Here’s $20. Drink until I’m good looking and then come talk to me.	ยาเสพติดและเครื่องดื่มแอลกอฮอล์

ภาพที่ 3.4 ตัวอย่างการวิเคราะห์เชิงทำนายผลลัพธ์ เพื่อจำแนกประเภทการกลั่นแกล้งทางอินเทอร์เน็ต เป็น 4 ประเภท

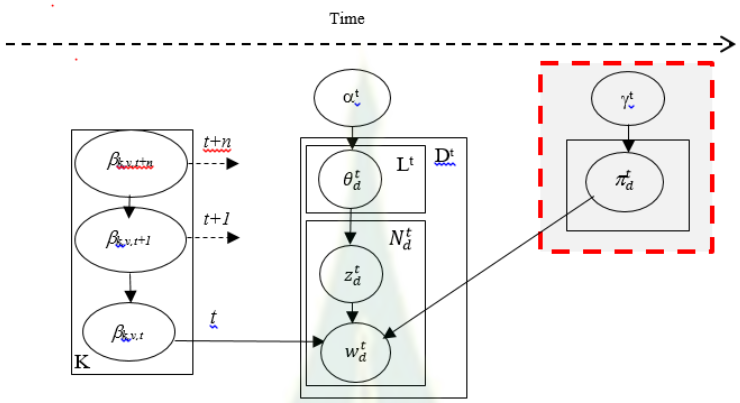
จากภาพที่ 3.4 ผลลัพธ์การดำเนินจากการวิเคราะห์เชิงทำนายผลลัพธ์ เพื่อจำแนกประเภทการกลั่นแกล้งทางอินเทอร์เน็ตเป็น 4 ประเภท เช่นประโยค “mmmm...i'm pre-cumming a little” ผลการวิเคราะห์เชิงทำนายจากโปรแกรมได้จำแนกประเภทการกลั่นแกล้งทางอินเทอร์เน็ตเป็น “ล่วงละเมิดทางเพศ” และ ประโยค “Baby, you’re the hot ass in my shot glass.” ผลการวิเคราะห์เชิงทำนายจากโปรแกรมได้จำแนกประเภทการกลั่นแกล้งทางอินเทอร์เน็ตเป็น “ยาเสพติดและเครื่องดื่มแอลกอฮอล์”

2. ขั้นตอนวิธีการดำเนินงาน

การวิจัยครั้งนี้ได้มีการพัฒนาขั้นตอนวิธีหรืออัลกอริทึม (Algorithm) ใหม่ 3 ขั้นตอนวิธี ได้แก่ 1) ขั้นตอนวิธีการคำนวณหาความคล้ายกันของเอกสาร 2) ขั้นตอนวิธีการคำนวณหาความน่าจะเป็นของคำในแต่ละหัวข้อ และ 3) ขั้นตอนวิธีการสร้างแบบจำลองของหัวข้อใหม่ โดยวิธีการอนุมานการจัดสรรหัวข้อแฝงแบบไดนามิกเชิงการประมวลผลแบบขนาน (dynamic joint Latent Dirichlet Allocation and parallelizable inference algorithm หรือที่เรียกว่า djLDA) เป็นขั้นตอนวิธีหรืออัลกอริทึม (Algorithm) ใหม่ที่ถูกพัฒนาขึ้นมาสำหรับงานวิจัยนี้

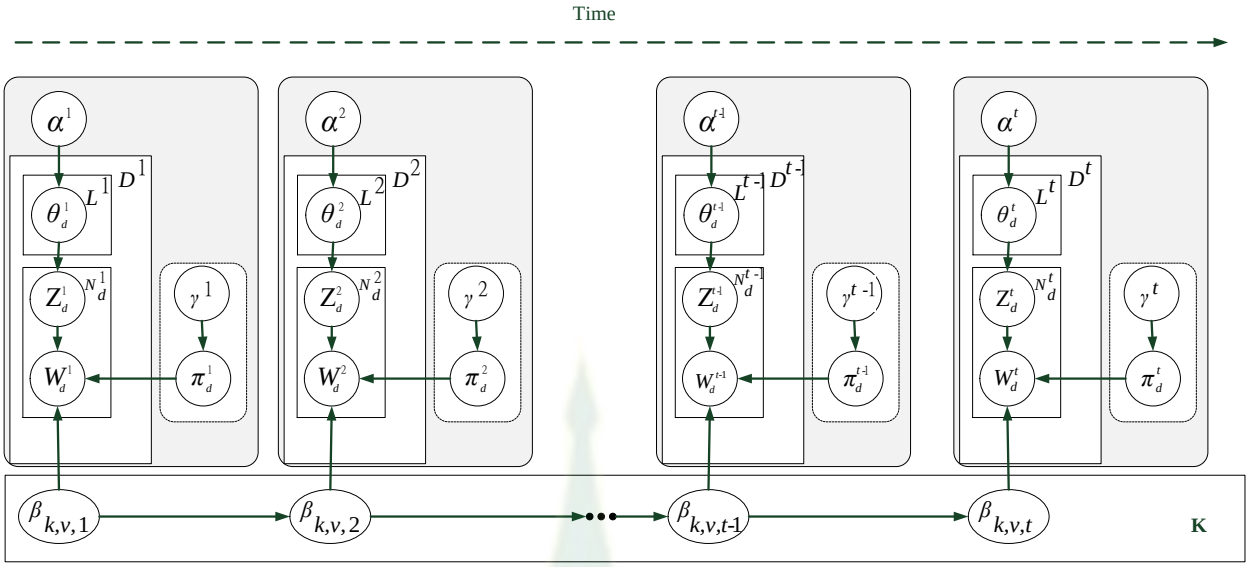
โดยอาศัยอัลกอริทึมหลักคือการจัดสรรหัวข้อแฝงหรือแอลดีเอ (Latent Dirichlet Allocation: LDA) ซึ่งได้เพิ่มเติมคุณลักษณะพิเศษหรือฟีเจอร์ (Feature) จากหลักการของ Wrapper approach เป็นการคัดเลือกฟีเจอร์ด้วยการสร้างโมเดล (Classification model) ขึ้นมาจากเซตของฟีเจอร์ที่กำหนดไว้และวัด

ประสิทธิภาพการทำงานของโมเดล และเลือกเซตของฟีเจอร์ที่ทำให้โมเดลมีประสิทธิภาพมากที่สุดมาใช้งาน ซึ่งในงานวิจัยนี้อาศัยหลักการคัดเลือกฟีเจอร์แบบ Backward Elimination เป็นการสร้างโมเดลที่เริ่มจากการใช้ฟีเจอร์ทั้งหมดก่อนและตัด (Eliminate) ฟีเจอร์ที่ไม่สำคัญทิ้งไปทีละฟีเจอร์ถ้าประสิทธิภาพดีขึ้นก็ตัดฟีเจอร์อื่นๆ ต่อไป เพื่อหาว่าฟีเจอร์ไหนเหมาะสมกับการสร้างแบบจำลองหรือโมเดล (Model) มากที่สุด ซึ่งได้คัดเลือก 2 ฟีเจอร์ คือ γ^t และ π_d^t



ภาพที่ 3.5 การกำหนด 2 ฟีเจอร์ใหม่ คือ γ^t และ π_d^t ในแอลดีเอ

จากภาพที่ 3.5 เมื่อทำการประมวลผลข้อมูลของเอกสารที่เข้าสู่ระบบแบบต่อเนื่องหรือกระแสข้อมูล (Data Streaming) ในเวลา t สามารถกำหนดสมการได้เป็น $S^t = \{d_{1,\ldots,}d_D{}^t\}$ ซึ่ง D^t คือขนาดของตัวแปร โดย N_d^t คือจำนวนของคำในเอกสาร d ที่แต่ละเวลาของ t สำหรับ π_d^t คือพารามิเตอร์สำหรับป้ายกำกับข้อมูล (Labeled data) ของการกลั่นแกล้งทางอินเทอร์เน็ต (Cyberbullying) ของ D^t ซึ่ง γ^t คือป้ายกำกับข้อมูลการกลั่นแกล้งทางอินเทอร์เน็ตในเวลาก่อนหน้าที่เวลา t-1 สำหรับ L^t เป็นป้ายกำกับการกลั่นแกล้งบนอินเทอร์เน็ตในเวลา t และ T_{new} หมายถึงจำนวนหัวข้อที่จะประมาณค่าน้ำหนักของเวกเตอร์จากข้อมูลที่กำหนดจุดคงที่ (Fixed point) หรือช่วงของคำกับค่าน้ำหนักที่ต้องการกำหนด



ภาพที่ 3.6 แสดงการทำงานของแบบจำลอง

จากภาพที่ 3.6 เมื่อกระแสข้อมูล (Data Streaming) ณ เวลา t (S^t) ถูกป้อนเข้ามาในแบบจำลอง พารามิเตอร์ $\beta_{k,v,t}$ จะถูกเปลี่ยนแปลงโดยนำ $\beta_{k,v,t-1}$ ซึ่งเป็นค่าของพารามิเตอร์ของกระแสข้อมูล ณ เวลาก่อนหน้ามาคำนวณร่วมเพื่อทำให้แบบจำลองสามารถเรียนรู้ได้จากกระแสข้อมูลก่อนหน้า เป็นเหตุให้แบบจำลองมีความแม่นยำมากขึ้น

1) ขั้นตอนวิธีการคำนวณหาความคล้ายกันของเอกสาร

เป็นอัลกอริทึมในส่วนแรกทำหน้าที่กรองเอกสารที่คล้ายกันทำให้สามารถลดเอกสารที่ต้องนำไปใช้ในการประมวลผล ซึ่งมีส่วนสำคัญในการเพิ่มความเร็วในการทำงานของระบบโดยรวม โดยมีรายละเอียดการทำงานของอัลกอริทึม

อัลกอริทึมที่ 1. การคำนวณหาความคล้ายกันของเอกสาร

1. Input: Data sources *as* $S_t = \{d_{1,\dots,}d_{Dt}\}$ *//a data stream of documents at time t*
2. Calculates the initialize Cluster Centroid;
3. Given two documents $\overrightarrow{di}$ and $\overrightarrow{dj}$ their cosine similarity is ;
$$Sim_{\lambda}(\overrightarrow{d_i}, \overrightarrow{d_j},) = \cos\theta = \frac{\overrightarrow{di}\overrightarrow{dj}}{|\overrightarrow{di}||\overrightarrow{dj}|}$$
4. $Sim_{\lambda}(\overrightarrow{di}, \overrightarrow{dj},) >= 90\%$ *// Acceptance in case two documents are similar should be equal or more than 90%*
5. $N = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} di*dj}{\sqrt{\sum_{i=1}^{n} di*}\sqrt{\sum_{j=1}^{n} dj}}$
6. Output : N of clusters; //Pre-processing for input datasets

จากอัลกอริทึมที่ 1 ข้อมูลที่นำเข้าเป็นกระแสข้อมูล ณ เวลา *t* ใดๆ แล้วนำมาคำนวณหาค่าศูนย์กลางของคลัสเตอร์เพื่อนำไปเปรียบเทียบเอกสารอื่นๆด้วยการคำนวณหาความคล้ายกันของเอกสาร ถ้าความคล้ายของเอกสารนั้นมากกว่าหรือเท่ากับ 90% แสดงว่าเอกสารมีความคล้ายกัน นั่นหมายความว่าสามารถประมวลผลแค่เอกสารเดียวจากเอกสารที่คล้ายกันทั้งหมด จากนั้นก็นำเอกสารมาคำนวณหาจำนวนคลัสเตอร์ซึ่งเป็นเอาท์พุตที่ต้องการใช้ในอัลกอริทึมการคำนวณหาความน่าจะเป็นของคำในแต่ละหัวข้อ

2) ขั้นตอนวิธีการคำนวณหาความน่าจะเป็นของคำในแต่ละหัวข้อ

เป็นอัลกอริทึมที่ทำงานคำนวณหาความน่าจะเป็นของคำที่อยู่ในแต่ละหัวข้อด้วย multinomial เพื่อดูการแจกแจงของคำ ณ เวลา *t* ใดๆ โดยใช้พารามิเตอร์ β เพื่อปรับการเรียนรู้ในการจำแนกหัวข้อที่เหมาะสม

อัลกอริทึมที่ 2. การคำนวณหาความน่าจะเป็นของคำในแต่ละหัวข้อ

- $\beta_{k,v,t}$: a probability of each word $v \in \{1, \ldots, V\}$ from vocabulary associated with a sequence of probabilities under each topic at time *t;*
- T_{new} : The new topics;

In the Algo1 Phase;

1. Input: N of clusters;
2. Split the corpus *D* into $\{D_0, \ldots, D_n\}$, corresponding topics $\{T_0, \ldots, T_n\}$;
3. Generating a probability of multinomial parameters;

$$\beta_{t,k,v} = \frac{\exp\{y_{t,k,v}\}}{\sum_{v}\exp\{y_{t,k,v}\}} \quad (5)\ [16]$$

//Calculated the probabilistic of word under topic with the multinomial parameters for a distribution over words

4. Draw a new topic $\leftarrow T_{new}$
5. Output : Re-draw a new topic from the key and value pair

จากอัลกอริทึมที่ 2 ขั้นตอนการทำงานของอัลกอริทึมการคำนวณหาความน่าจะเป็นของคำในแต่ละหัวข้อ เริ่มจากนำเข้าจำนวนคลัสเตอร์ซึ่งเป็นเอาท์พุตจากอัลกอริทึมการคำนวณหาความคล้ายกันของเอกสาร จากนั้นให้น้ำหนักของคำในเอกสารทั้งหมดด้วยการพิจารณาร่วมกับฐานข้อมูล corpus แล้วคำนวณหาความน่าจะเป็นของคำในแต่ละหัวข้อเพื่อใส่หัวข้อให้เหมาะสมกับคำ

3) ขั้นตอนวิธีการสร้างแบบจำลองของหัวข้อใหม่

เป็นอัลกอริทึมที่ทำงานในส่วนเอาท์พุตซึ่งเป็นการสร้างแบบจำลองเพื่อนำไปใช้ในการเรียนรู้ในการประมวลผลกระแสข้อมูล ณ เวลา *t* ถัดไป

อัลกอริทึมที่ 3. การสร้างแบบจำลองใหม่ของแบบจำลองหัวข้อ

- π_d^t : a parameter notation for the cyberbullying label in D^t at time t;
- γ^t: a symmetric prior of cyberbullying label at time t;
- L^t: a kind of cyberbullying label at time t;

In the BlotOutput Phase;

1. Input: $T_{new;}$ // The new topics;
2. Sort-Merge Join

   Join R $\bowtie$ S = { (r,s) | P_{join} (r,s), r ∈ R, s ∈ S};

   // Let R and S be two tuples, Let P_{join} (r,s), r ∈ R, s ∈ S be a binary predicate.

   Shuffle tasks, sub corpus D_p;

   //Re-draw from the distribution G, generate a vector from a multivariate Gaussian, then to transform to a multinomial parameter and renormalizing into new one

   $$\mathbf{G_0} = \sum_{i=1}^{\infty} \boldsymbol{V_{\beta t,k,v}} \prod_{j=1}^{i-1}(\mathbf{1} - \boldsymbol{V_j^d})\, \boldsymbol{\pi_d^{tn}} \boldsymbol{\gamma^{tn}} \quad (6)$$

   $$\boldsymbol{\tau}\,(\boldsymbol{w_d^{tn}}) \;\; \textit{//Optimization vector of word tokens} \quad (7)$$

   Generate TPCyberbullying; // Generate a topic model from label cyberbullying;

3. Output: Generate the new topic model from data streams;

จากอัลกอริทึมที่ 3 ขั้นตอนการสร้างแบบจำลองหัวข้อจากกระแสข้อมูล ณ เวลา t ใดๆ โดยจะทำการนำข้อมูลเข้าจากแบบจำลองหัวข้อใหม่ หรือ T_{new} ที่ได้โดยการกำหนดคู่ค่าคีย์ หรือ Key-Values มาทำการจากจัดเรียงลำดับและผสานข้อมูลตามน้ำหนัก (weight) จากคำในแต่ละหัวข้อที่เกี่ยวข้อง ซึ่งเป็นส่วนหนึ่งของขั้นตอนการสลับเปลี่ยน (Shuffle phase) ตามเอกสาร D จากนั้นทำการลดรูปการกระจายข้อมูลจากความน่าจะเป็นด้วยวิธีการของ Gaussian เพื่อให้ได้สัดส่วนของคำที่มีน้ำหนักถูกต้องมากที่สุด จากการกำหนดปัจจัยเพิ่มเติมของคำจากเรื่องหัวข้อและคำเกี่ยวกับอาชญากรรมไซเบอร์ โดยผลลัพธ์ที่ได้จากอัลกอริทึมที่ 3 นี้จะได้คำที่อยู่ในกลุ่มอาชญากรรมไซเบอร์โดยจำแนกตามหัวข้อจากกระแสข้อมูล ณ เวลา t ใดๆ

3. เครื่องมือที่ใช้ในการวิจัย

3.1 ฮาร์ดแวร์

1) การทดสอบการทำงานโปรแกรมบนเครื่อง HP - HP Compaq Z400 Workstation (ช่อง 6-DIMM) สำหรับโหนดหลัก (Master Node) และกำหนดโหนดลูก (Slave Nodes) จำนวน 64 โหนด เป็นอินเทลซีออนโปรเซสเซอร์ CPU, W3508 @ 2.40 GHz กับ 4.00 GB RAM

2) การทดสอบการทำงานโปรแกรมบนกรอบการทำงานของ Apache Hadoop รุ่น 2.7.3 สำหรับ Windows ให้บริการกำหนดค่าการกระจายการบริการและการประสานการรวมข้อมูลกลับมาที่โหนดหลักและการสร้างแบบจำลองหัวข้อที่ด้วยการจัดสรรหัวข้อแฝง (Latent Dirichlet Allocation: LDA)

3) การทดสอบระบบงานเหมืองข้อความจริงบนเครื่องเซิร์ฟเวอร์เพื่อประมวลผลข้อมูลขนาดใหญ่จาก Amazon Web Services (AWS) บน Elastic Compute Cloud (EC2) คิดราคาการประมวลผลตามวินาที (Per-second billing)

3.2 ซอฟต์แวร์

1. อาปาเชฮาดูป รุ่น 2.Y.Z

โปรแกรมอาปาเชฮาดูป (Apache Hadoop) เป็นซอฟต์แวร์แบบโอเพนซอร์ซ (Open Source) ที่ประกอบด้วยกลุ่มของชุดคำสั่งต่างๆ (Libraries) เพื่อช่วยอำนวยความสะดวกแก่นักพัฒนาแอปพลิเคชั่นที่จะสร้างระบบหรือวิเคราะห์ข้อมูลขนาดใหญ่ (Big Data Analytics) ได้อย่างมีประสิทธิภาพ เพื่อรองรับการทำงานบนระบบคอมพิวเตอร์แบบกระจาย (Distributed Computing) และสนับสนุนการประมวลผลแบบขนาน (Parallel Processing) ที่มีความเสถียรสูงและสามารถเพิ่มขยายจำนวนเครื่องในระบบได้อย่างมหาศาล

ในงานวิจัยนี้ได้ใช้โปรแกรมอาปาเชฮาดูป รุ่น 2.Y.Z (วฤษาย์ ร่มสายหยุด, 2560) ที่มีโครงสร้างการทำงาน ดังภาพ 3.7ประกอบด้วย 4 ส่วนหลัก ได้แก่ 1) Hadoop Common 2) ระบบแฟ้มข้อมูลแบบกระจายฮาดูป (Hadoop Distributed File System) 3) ฮาดูปยาน (Hadoop YARN) และ 4) แมปรีดิวซ์ (MapReduce) รายละเอียดดังต่อไปนี้

ภาพที่ 3.7 โครงสร้างหลักการทำงานของฮาดูปรุ่น 2.Y.Z (วฤษาย์ ร่มสายหยุด, 2560)

1) ฮาดูปทั่วไป (Hadoop Common) เป็นกลุ่มข้อมูลของคลาส (Class) หรือไลบารี (Libraries) จำนวนมาก เพื่อรองรับการทำงานของฮาดูป เช่น การกำหนดค่าหรือการปรับเปลี่ยนค่าข้อมูล ในไฟล์ configuration, Reconfiguration, resources หรือ indexing เป็นต้น

2) ระบบแฟ้มข้อมูลแบบกระจายฮาดูป (Hadoop Distributed File System หรือ HDFS) เป็นการนำข้อมูลเข้า (Input Data) ที่มีขนาดใหญ่จำนวนมาก มาทำการแบ่งข้อมูลขนาดใหญ่ให้มีขนาดเล็กลง (Data Splitting) เพื่อกระจายข้อมูลขนาดเล็กๆ เหล่านี้ ไปประมวลผลในระบบเครื่องคอมพิวเตอร์คลัสเตอร์ โดยกระบวนการทำงานของระบบแฟ้มข้อมูลแบบกระจายฮาดูป จะประกอบด้วย 2 ส่วนหลักได้แก่ มาสเตอร์โหนด (Master Node) ซึ่งจะมีเพียงโหนดเดียว และสลาฟโหนด (Slave Nodes) ซึ่งจะมีได้หลายโหนด ซึ่ง Master Node และ Slave Nodes จะเชื่อมต่อกันผ่านอุปกรณ์สื่อสาร (Rack Switch) ทำให้สามารถเพิ่มจำนวนของ Slave Nodes ได้หลายๆ เครื่อง

3) ฮาดูปยาน (Hadoop YARN) โดยยาน (YARN) มาจากคำว่า “Yet Another Resource Negotiator” หรือบางคนเรียกว่าเป็นโครงการย่อยของแมปรีดิวซ์ (MapReduce : MR) รุ่นที่ 2 (MRv2) ที่ทำหน้าที่เป็นผู้บริหารทรัพยากร (Resource Management) ของแมปรีดิวซ์ กำหนดและควบคุมการประมวลผล สนับสนุนการทำงานกับระบบที่เกี่ยวข้องกับฮาดูปและรองรับการประมวลผลข้อมูลแบบทันทีทันใดหรือเวลาจริง (Realtime) อีกด้วย

4) แมปรีดิวซ์ เป็นโปรแกรมการทำงานที่ทำงานอยู่บนฮาดูป ที่นำข้อมูลที่ได้แบ่งให้เป็นข้อมูลเล็ก (Data Splitting) เข้าสู่ขั้นตอนการทำงานของแมปรีดิวซ์ โดยแมปรีดิวซ์จะประกอบด้วย 2 ส่วนได้แก่ 1) ขั้นตอนแมป เป็นการกำหนดคู่คีย์ (Key Pair) ของข้อมูลเพื่อทำการกระจายข้อมูลไปประมวลผลยังโหนด (Nodes) ต่างๆ ในระบบเครื่องคอมพิวเตอร์คลัสเตอร์ตามคำสั่งการทำงาน และ 2) ขั้นตอนรีดิวซ์ เป็นการนำผลที่ได้จากการประมวลผลแบบกระจายของขั้นตอนแมป ในแต่ละโหนด กลับมาทำการจัดเรียงลำดับและสรุปผลข้อมูล เพื่อแสดงผลลัพธ์การทำงานที่รวดเร็วในการประมวลผลข้อมูลขนาดใหญ่ ตัวอย่างการประยุกต์การทำงานของแมปรีดิวซ์บนฮาดูป ได้แก่ โปรแกรมการนับคำ (WordCount) ที่ทำการนับจำนวนคำในเอกสารขนาดใหญ่ โดยทำการตัดแบ่งข้อมูลของเอกสารให้เป็นข้อมูลเล็กๆ และกระจายเอกสารเล็กๆเหล่านี้ไปประมวลผลตามขั้นตอนแมป เพื่อให้แต่ละโหนดทำการนับคำเฉพาะในเอกสารข้อมูลเล็กๆ จากนั้นขั้นตอนรีดิวซ์จะนำผลลัพธ์ของจำนวนคำมาทำการจัดเรียงและสรุปผลการทำงาน ว่าในเอกสารนี้มีคำว่าอะไรบ้าง จำนวนกี่คำ ซึ่งจะทำให้ได้ผลลัพธ์ที่รวดเร็วอย่างมาก

2. จาวา (Java)

โปรแกรมจาวา (Java) เป็นซอฟต์แวร์แบบโอเพนซอร์ซ (Open Source) ที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP) สามารถนำมาพัฒนาแอปพลิเคชั่นได้หลากหลายรูปแบบมาก เช่น แอปพลิเคชั่นที่ทำงานบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ แอปพลิเคชั่นที่ทำงานบนระบบปฏิบัติการแมคโอเอส แอปพลิเคชั่นที่ทำงานบนระบบปฏิบัติการลินุกซ์ หรือบนเว็บแอปพลิเคชัน

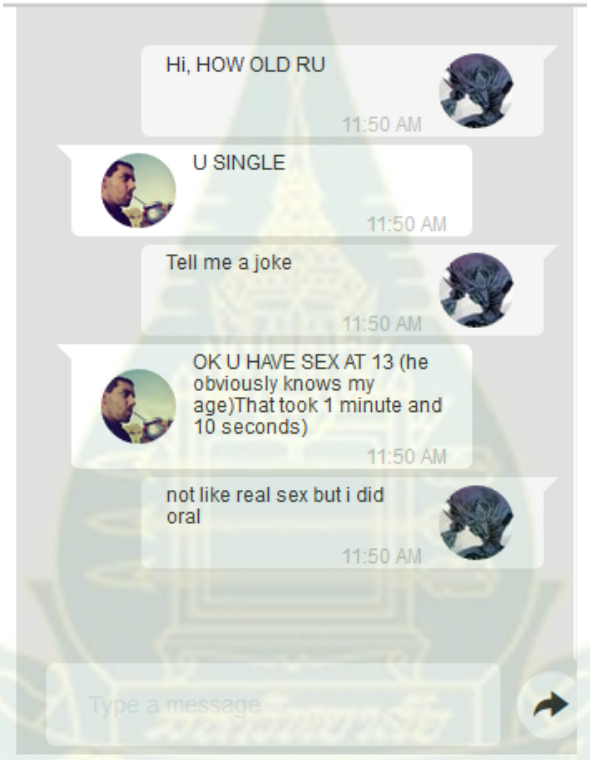
3. ซิกวิน

โปรแกรมซิกวิน (Cygwin) เป็นซอฟต์แวร์แบบโอเพนซอร์ซ (Open Source) ที่จำลองสภาพแวดล้อมของระบบลินุกซ์บนระบบปฏิบัติการไมโครซอฟต์วินโดวส์โดยการจำลองสภาพแวดล้อมของระบบลินุกซ์ที่รวมเอาไลบารี แอปพลิเคชันและโปรแกรมที่มีบนลินุกซ์ ให้สามารถใช้ได้บนวินโดวส์ได้

บทที่ 4
ผลการวิเคราะห์ข้อมูล

1. การดำเนินการพัฒนาระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์

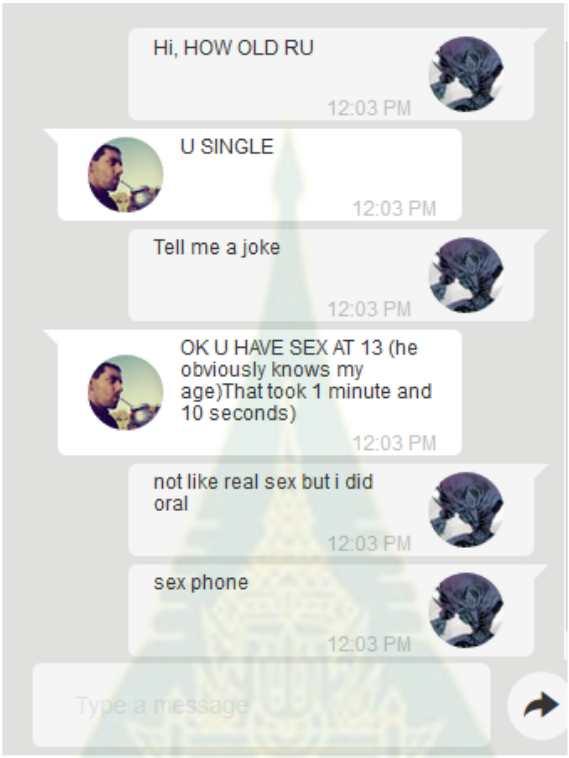
1.1 การพัฒนาโปรแกรมที่ฝั่งเครื่องไคลเอนต์ด้วยภาษา HTML5, CSS และ JavaScript สำหรับ Windows (การพัฒนาเว็บแอปพลิเคชัน) เป็นการพัฒนาหน้าเว็บแอปพลิเคชันเป็นแบบห้องแชต (Chat room) เพื่อให้ผู้ใช้ทำการล็อกอินเข้าสู่ระบบ โดยระบบจะทำการเก็บล็อกไฟล์ (log file) ข้อมูลการสนทนา ชื่อผู้ใช้ วันที่ เวลา และข้อความเป็นต้น



ภาพที่ 4.1 การพัฒนาหน้าเว็บแอปพลิเคชันเป็นแบบห้องแชต (Chat room)

จากภาพที่ 4.1 การพัฒนาหน้าเว็บแอปพลิเคชันเป็นแบบห้องแชต (Chat room) โดยมีผู้ใช้ที่ทำการ login เข้าสู่ระบบ 2 คนกำลังสนทนากันผ่านหน้าเว็บที่พัฒนาด้วยภาษา HTML5 พร้อมทั้งกำหนดรูปแบบข้อความให้แตกต่างของแต่ละผู้ใช้ด้วยภาษา CSS และการเขียนคำสั่งภาษา JavaScript เพื่อทำการแสดงข้อความ เวลา วันที่ และรูปภาพการสนทนา พร้อมทั้งทำการเชื่อมต่อข้อมูลหรือประโยคการสนทนาต่างๆ แบบเวลาจริงไปประมวลผลที่ฝั่งเครื่องเซิร์ฟเวอร์ ที่ทำหน้าที่ตรวจพบและป้องกันจากอาชญากรรมไซเบอร์

โดยระบบจะทำการประมวลผลเชิงวิเคราะห์เพื่อทำนายข้อมูลจากประโยคการสนทนา อาศัยหลักการจัดสรรหัวข้อแฝงร่วมกับนาอีฟเบย์ ในการพิจารณาคำ วลี คำศัพท์ และคำกำกวม จากนั้นนำคุณลักษณะพิเศษหรือฟีเจอร์ (feature) มาช่วยในการเรียนรู้พฤติกรรมการสนทนา เช่น วัน เวลา ความถี่ในการสนทนาร่วมกัน และระยะเวลาที่สนใจเป็นต้น เพื่อประมวลผลข้อมูลได้แม่นยำและถูกต้องมากยิ่งขึ้น



ภาพที่ 4.2 ตัวอย่างประโยคการสนทนา

จากภาพที่ 4.2 ตัวอย่างประโยคการสนทนาระหว่างผู้ใช้งานระบบ 2 คนที่มีข้อมูลรูปภาพของผู้ใช้งานที่ได้ลงทะเบียน ประโยคการสนทนา เวลาและระยะเวลาการโต้ตอบ ซึ่งมีรายละเอียดดังต่อไปนี้

ข้อความที่ 1 ของผู้ใช้คนที่ 1: HI, HOW OLD RU

ข้อความที่ 2 ของผู้ใช้คนที่ 2: U SINGLE

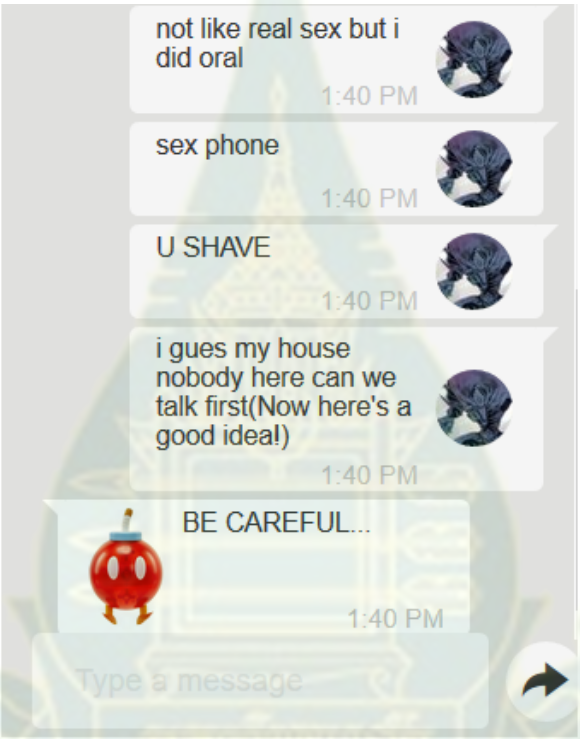
ข้อความที่ 3 ของผู้ใช้คนที่ 1: Tell me a joke

ข้อความที่ 4 ของผู้ใช้คนที่ 2: OK U HAVE SEX AT 13 (he obviously knows my age)That took 1 minute and 10 seconds

ข้อความที่ 5 ของผู้ใช้คนที่ 1: not like real sex but i did oral

ข้อความที่ 6 ของผู้ใช้คนที่ 1: sex phone

จากประโยคการสนทนาเริ่มต้นของผู้ใช้คนที่ 1 คือ HI, HOW OLD RU เมื่อนำเข้าระบบฯ จะเป็นประโยคปกติ กล่าวคือเมื่อนำคำ วลี รูปแบบประโยคไปเปรียบเทียบกับพจนานุกรมหรือดิกชันนารีคำศัพท์ (dictionary) ตามหลักการจัดสรรหัวข้อแฝง ไม่เป็นคำในกลุ่มคลาสการกลั่นแกล้งทางอินเทอร์เน็ต โดยระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ทำการกำหนดคลาส (class) เพื่อจำแนกประเภทการกลั่นแกล้งทางอินเทอร์เน็ตเป็น 4 คลาสได้แก่ 1) ล่วงละเมิดทางเพศ (Sexual harassment) 2) หลอกลวงเงิน (Money Mule Scams) 3) พยายามฆ่าตัวตาย (Suicide Attempts) และ 4) ยาเสพติดและเครื่องดื่มแอลกอฮอล์ (Drug and Alcohol Abuse) จากนั้นระบบจะทำการนำแต่ละประโยคเข้าสู่ระบบเพื่อทำการวิเคราะห์เปรียบเทียบกับโมเดล จะเห็นได้ว่าข้อความที่ 4-6 น้นตรงกับดิกชันนารีคำศัพท์และคลังคำศัพท์ (Corpus) ในคลาสล่วงละเมิดทางเพศ (Sexual harassment) มากที่สุด



ภาพที่ 4.3 ระบบแจ้งเตือนอัตโนมัติ

จากภาพที่ 4.3 ระบบแจ้งเตือนอัตโนมัติจะดำเนินการส่งข้อความไปขัดจังหวะการสนทนาเป็นข้อความว่า “BE CAREFUL...” เพื่อป้องกันและป้อมปรามการสนทนาให้ผู้ใช้ระบบหรือคู่สนทนาได้ระวังตัวว่าอาจจะตกเป็นเหยื่อจากการสนทนาในรูปแบบนี้ และหากผู้ใช้ระบบหรือคู่สนทนาดำเนินการสนทนาต่อไประบบฯ จะเฝ้าระวังคำหรือทำนายแนวโน้มภาวะวิกฤตที่จะเกิดขึ้นโดยทำการตัดเซสชัน (Session) การสนทนาของผู้ใช้ทันที เพื่อให้ผู้ประสงค์ร้ายหยุดการสนทนาและรู้ว่าระบบมีการป้องกัน รวมถึงเหยื่อได้หยุดจากการถูกคุมคาม รวมถึงอันตรายที่จะเกิดขึ้น (ข้อมูลการพัฒนาเว็บแอปพลิเคชัน อยู่ที่ภาคผนวก ก)

1.2 พัฒนาโปรแกรมด้วยโปรแกรมฝั่งเครื่องเซิร์ฟเวอร์ด้วยภาษา Java สำหรับ Windows จะทำการแปลงภาษาเทียม (Pseudo code) จากบทที่ 3 ให้เป็นภาษา Java โดยได้พัฒนาขั้นตอนวิธีหรืออัลกอริทึม (Algorithm) ใหม่ 3 ขั้นตอนวิธี ได้แก่ 1) ขั้นตอนวิธีการคำนวณหาความคล้ายกันของเอกสาร 2) ขั้นตอนวิธีการคำนวณหาความน่าจะเป็นของคำในแต่ละหัวข้อ และ 3) ขั้นตอนวิธีการสร้างแบบจำลองของหัวข้อใหม่ (ข้อมูลการพัฒนาโปรแกรมฝั่งเครื่องเซิร์ฟเวอร์ด้วยภาษา Java อยู่ที่ภาคผนวก ข)

ในดำเนินการจะนำประโยคจากการสนทนาในหน้าเว็บแอปพลิคัชันสำหรับห้องสนทนาในแต่ละประโยค มาดำเนินการโดยการทำงานในส่วนนี้จะแบ่งหน้าที่การทำงานเป็น 5 ส่วนได้แก่ 1) การตัดคำตามกฎไวยากรณ์ภาษาอังกฤษและการอ้างอิงคำจากพจนานุกรม สำหรับการตัดคำจากประโยคหรือข้อความให้ออกมาเป็นคำเดี่ยวๆ (Tokenization) คำโพลาร์ (Polar word) รวมถึงการกำหนดรูปแบบของประโยคด้วย 2) การลดรูปคำ (Normalization) 3) การทำความสะอาดคำหรือตัวอักษร (Cleansing) 4) การสร้างตารางคำศัพท์ และ 5) การทำนายผลจากประโยคตามอัลกอริทึมใหม่ที่ได้พัฒนาขึ้น รายละเอียดดังต่อไปนี้

1) ขั้นตอนการตัดคำตามกฎไวยากรณ์ภาษาอังกฤษและการอ้างอิงคำจากพจนานุกรมหรือดิกชันนารี

ตัวอย่างประโยค 1: would you like to see my dick?

รูปแบบการตัดคำ |like| to |see |my |dick

ตัวอย่างประโยค 2: LOL i love money because money is god

รูปแบบการตัดคำ |love |money |because |money |god

ตัวอย่างประโยค 3: i dont want to get killed tooooo

รูปแบบการตัดคำ |don’t |want | get |kill |too

ตัวอย่างประโยค 4: drug weed cocaine-heroine sell capsule drug weed drug

รูปแบบการตัดคำ drug | weed |cocaine-heroine หรือ cocaine heroine | sell |capsule |drug |weed |drug

2) ขั้นตอนการลดรูปคำ เช่นในตัวอย่างประโยค 3: i dont want to get killed toooooo จะทำการลดรูปคำว่า toooooo เป็น too โดยการเปรียบเทียบกับพจนานุกรม จะได้เป็น รูปแบบการตัดคำ |don’t |want | get |kill |too

3) ขั้นตอนการทำความสะอาดคำและตัวอักษร โดยทำความสะอาดคำและตัวอักษร เช่นคำว่า “LOL” ไม่มีในพจนานุกรมจะถูกตัดทิ้ง รวมถึงช่องว่างระหว่างคำด้วย เช่นในตัวอย่างประโยค 2: LOL i love money because money is god จะได้เป็น รูปแบบการตัดคำ |love |money |because |money |god

4) การสร้างตารางคำศัพท์

Docs	You like	My dick	Love Money	Killed	Get Killed Too	drug	Drug weed	Cocaine-heroine
1	1	1						
2		1	2					
3				1	1			
4						2	1	1

ภาพที่ 4.4 การสร้างตารางคำศัพท์

จากภาพที่ 4.4 นำเอกสารแต่ละเอกสารเช่น เอกสารที่ 1, 2, 3, 4 และนำคำศัพท์ที่ผ่านการคัดเลือกในหลักการภาษาธรรมชาติมาทำการกำหนดลงในแต่ละคอลัมน์เพื่อสร้างตารางการแจกแจงคำศัพท์ พร้อมทั้งระบุค่าของคำแต่ละคำลงไป เช่น ในเอกสารที่ 1 มีคำว่า “you like” เท่ากับ 1 คำ หรือในเอกสารที่ 4 มีคำว่า “drug” เท่ากับ 2 คำ เพื่อแสดงความถี่การเกิดของคำนั้นๆ ในเอกสาร

5) การทำนายผลจากประโยค

	การกำหนดคลาสจากการเรียนรู้			
การทำนายผล	คลาสที่ 1	คลาสที่ 2	คลาสที่ 3	คลาสที่ 4
คลาสที่ 1	0	0	0	0
คลาสที่ 2	0	0	0	0
คลาสที่ 3	0	0	1	0
คลาสที่ 4	1	0	0	0

ภาพที่ 4.5 ตัวอย่างการทำนายผลจากประโยค

ภาพที่ 4.5 ตัวอย่างการทำนายผลจากประโยค โดยทำการกำหนดคลาส 4 คลาสได้แก่ 1) ล่วงละเมิดทางเพศ (Sexual harassment) 2) หลอกลวงเงิน (Money Mule Scams) 3) พยายามฆ่าตัวตาย (Suicide Attempts) และ 4) ยาเสพติดและเครื่องดื่มแอลกอฮอล์ (Drug and Alcohol Abuse) ซึ่งประโยคเมื่อทำการกำหนดคลาสจากการเรียนรู้ กำหนดเป็นคลาสที่ 3 และตรงกับผลการทำนายด้วยอัลกอริทึม

ได้ออกมาเป็นคลาสที่ 3 เหมือนกัน แต่ในบางประโยคการกำหนดคลาสจากการเรียนรู้ กำหนดเป็นคลาสที่ 1 แต่ผลการทำนายได้คลาสที่ 4 ซึ่งไม่ตรงกันทำให้เกิดความคาดเคลื่อนของแบบจำลอง

2. การดำเนินการประเมินค่าความถูกต้องของระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์

การทดลองโดยใช้ข้อมูลของอาชญากรรมทางไซเบอร์ซึ่งเป็นชุดข้อมูลจากประชากรกลุ่มตัวอย่าง คือ สมาชิกเว็บไซต์ Perverted-justice, Formspring และ MySpace ในปีค.ศ. 2010 โดยมีข้อความเนื้อหาจากการโพสต์ของเว็บไซต์ perverted-justice.com ข้อความจาก Formspring และ ข้อความจาก MySpace จำนวนทั้งสิ้น 127,974 ซึ่งใช้ข้อมูลการฝึกอบรมและการทดสอบข้อมูลที่แยกในสถานการณ์ต่างๆ บนข้อมูลแบบสตรีมมิ่ง ข้อมูลการฝึกอบรม (Training Data) จำนวน 23,492 โดยแบ่งข้อมูลเป็นแบบ 10-fold cross-validation เพื่อทดสอบประสิทธิภาพของแบบจำลอง ในการทดลองนี้ได้มีการวิเคราะห์และศึกษาชุดข้อมูลจริงซึ่งเก็บรวบรวมระหว่างปี พ.ศ. 2549-2557 ในส่วนนี้วิธีการประเมินผลและผลการประเมินจะได้รับการอธิบายอย่างละเอียดและกล่าวถึงในรายละเอียด

ตัวอย่างประโยคการสนทนา

ArmySgt1961 (7:02:01 PM): IM DENNIS US ARMY SOLDIER FROM CINCINNATI

PeekaBoo1293 (7:02:30 PM): hi im becky from ky

ArmySgt1961 (7:02:35 PM): HOW OLD RU

PeekaBoo1293 (7:02:42 PM): 13 how old ru (age stated and he didn't bat an eye)

ArmySgt1961 (7:02:44 PM): U SINGLE

PeekaBoo1293 (7:02:51 PM): yeah

PeekaBoo1293 (7:03:03 PM): i had a bf but we broke up when i moved here

ArmySgt1961 (7:03:11 PM): OK U HAVE SEX AT 13 (he obviously knows my age)That took 1 minute and 10 seconds)

PeekaBoo1293 (7:03:28 PM): u mean did i ever

ArmySgt1961 (7:03:32 PM): YEAH

PeekaBoo1293 (7:03:45 PM): not like real sex but i did oral

PeekaBoo1293 (7:03:54 PM): did u ever do real sex

ArmySgt1961 (7:04:00 PM): SURE

PeekaBoo1293 (7:04:08 PM): i didnt yet

PeekaBoo1293 (7:04:18 PM): i was scared i mite get preggerz

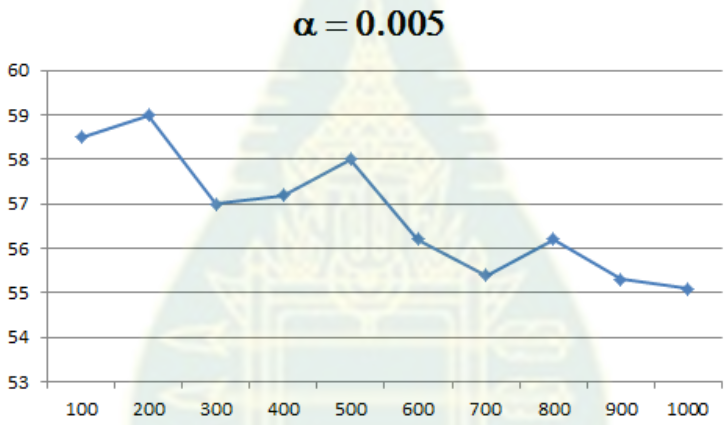
PeekaBoo1293 (7:04:45 PM): and my bf didnt hav no comdom so i wouldnt do it(I don't know why it always makes me giggle to tell them to bring "comdoms")

ArmySgt1961 (7:05:01 PM): OK

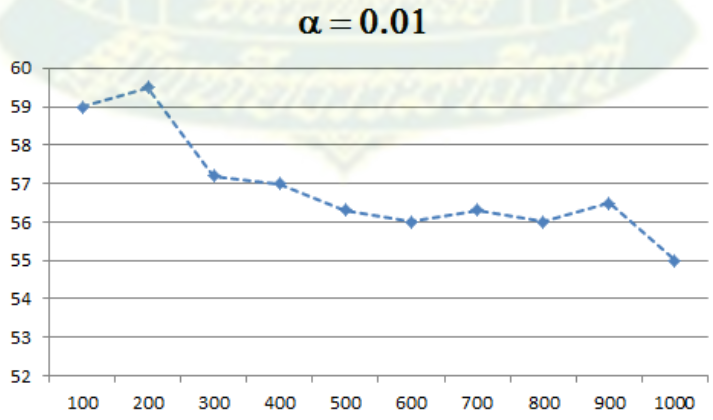
ArmySgt1961 (7:05:07 PM): U HAVE ANY PICS

2.1 การทดสอบ

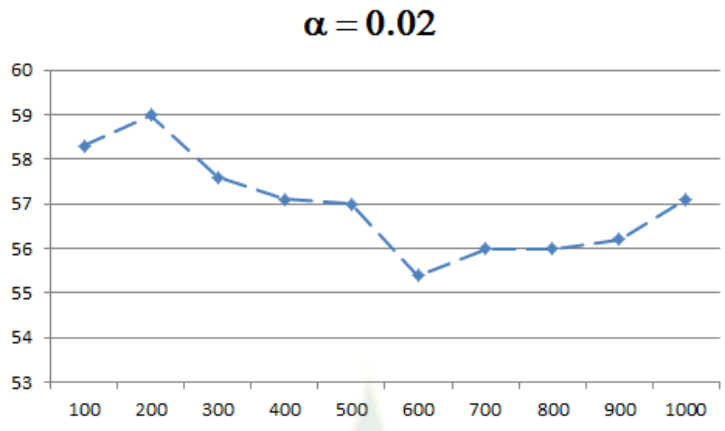
การทดสอบทำซ้ำจำนวนครั้งสูงสุด 10 ครั้ง (เพื่อเปรียบเทียบ) สำหรับอัลกอริทึมนี้ การทดสอบแต่ละครั้งจะทำงานทั้งหมด 10 ครั้ง โดยกำหนดเทรสโฮลด์ (Threshold) หรือค่าข้อมูลเข้าที่น้อยที่สุดที่เกี่ยวข้องระหว่างสองคำอยู่ที่ค่า 0.4 สำหรับแต่ละประโยค การฝึกอบรมข้อมูลเพื่อเพิ่มประสิทธิภาพของค่าเทรสโฮลด์จากคำพูดเพื่อให้ครอบคลุมคำ การสะกดคำ ประโยคแวดล้อม การตัดคำ การเปรียบเทียบคลังข้อความคำกำกวมและคำแสดงอารมณ์ ตามหลักการภาษาธรรมชาติเพื่อให้มีความถูกต้องมากที่สุด



(a) α = 0.005



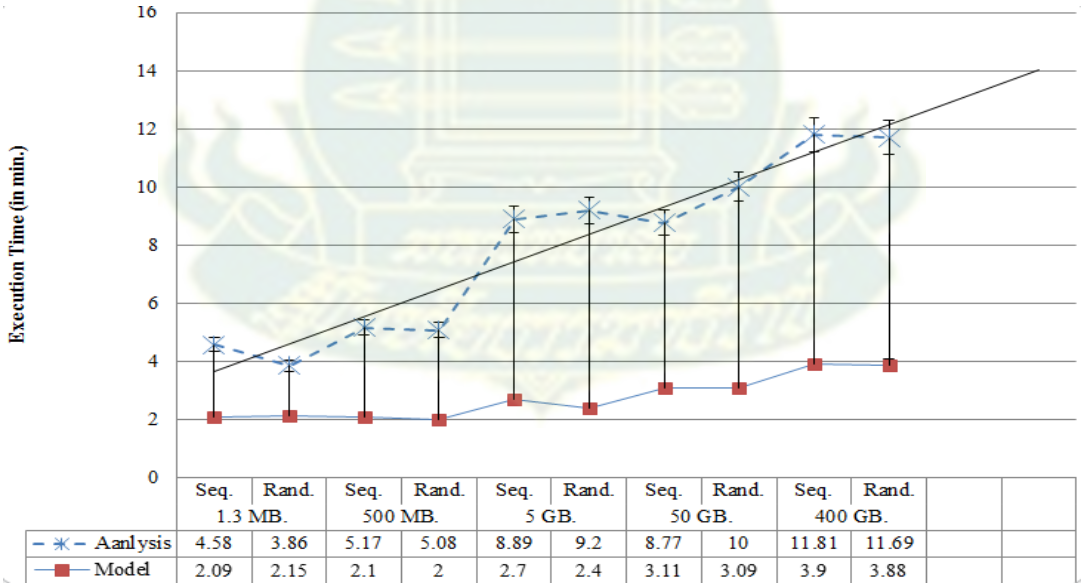
(b) α = 0.01



(c) $\alpha = 0.02$

ภาพที่ 4.6 (a), (b) และ (c) การจำแนกคุณสมบัติพิเศษด้วยนาอีฟเบย์ ตามขนาดข้อมูลและความหลากหลายของข้อมูล

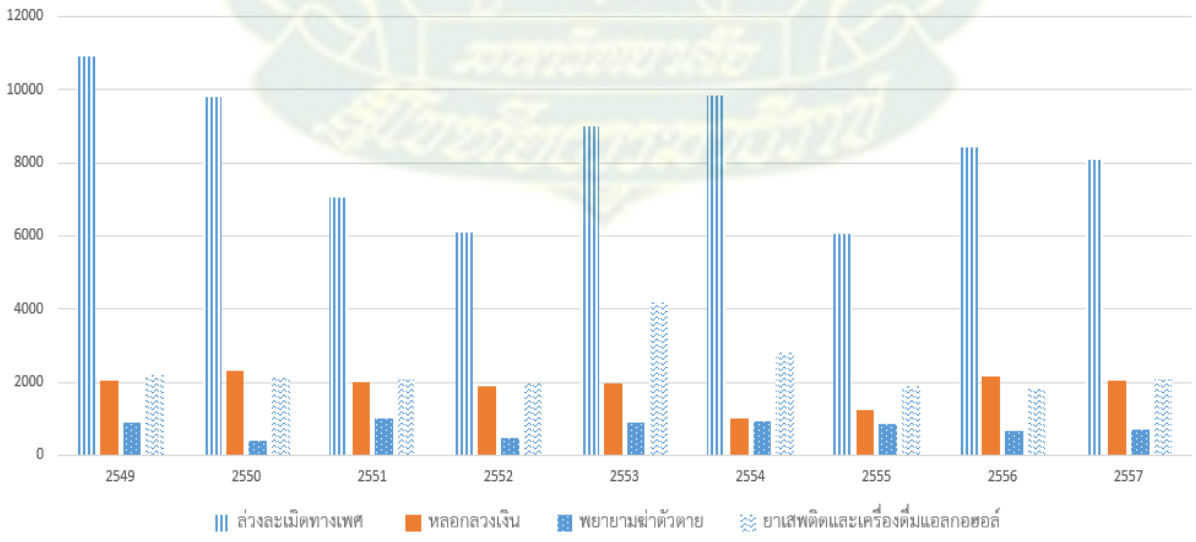
จากรูปที่ 4.6 (a), (b) และ (c) จำนวนการแสดงซ้ำสูงสุดคือ 100 และชุดคุณลักษณะมี 200 คุณลักษณะ มีค่า α ที่ถูกทดสอบในงานนี้ซึ่งมีค่า α = 0.005, α = 0.01 และ α = 0.02 โดยใช้นาอีฟเบย์เพื่อทดสอบประสิทธิภาพ โดยกำหนดค่าคุณลักษณะพิเศษของค่ากำไร (gain) ขั้นต้นที่อยู่ที่ร้อยละ 0.7 ทำงานได้ดีกว่า N-gram ("I love you", "I love", "love you" และ I | love| you|) ด้วยการเพิ่มจำนวนรูปแบบคำแฝงของนาอีฟเบย์มีความแม่นยำในการจัดหมวดหมู่ร้อยละ 95.79 ซึ่งสูงมาก จากจำนวนคุณลักษณะทั้งหมด ความถูกต้องทั้งหมดยกเว้นความแตกต่างอย่างมีนัยสำคัญทางสถิติ สำหรับค่า α = 0.02 มีแนวโน้มที่จะทำงานได้ดีกว่าค่าอื่น ๆ สำหรับชุดข้อมูลเดียวกัน



	Seq. 1.3 MB.	Rand. 1.3 MB.	Seq. 500 MB.	Rand. 500 MB.	Seq. 5 GB.	Rand. 5 GB.	Seq. 50 GB.	Rand. 50 GB.	Seq. 400 GB.	Rand. 400 GB.
Aanlysis	4.58	3.86	5.17	5.08	8.89	9.2	8.77	10	11.81	11.69
Model	2.09	2.15	2.1	2	2.7	2.4	3.11	3.09	3.9	3.88

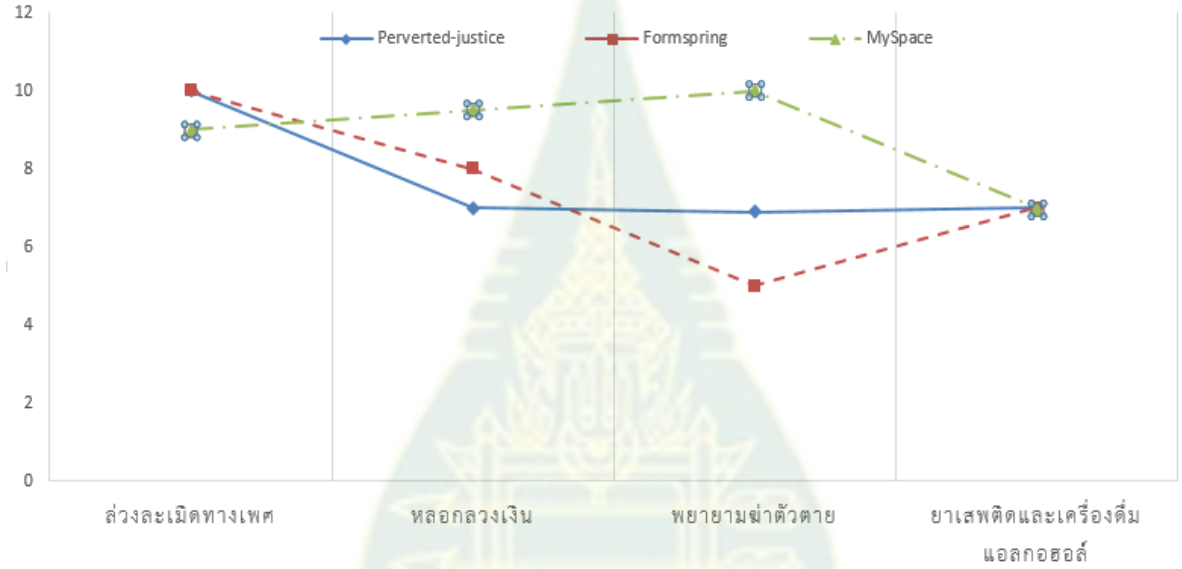
ภาพที่ 4.7 การทดสอบการทำงานของขั้นตอนการวิเคราะห์ข้อมูลและการสร้างแบบจำลอง

จากภาพที่ 4.7 ซึ่งประกอบด้วย 2 ขั้นตอนหลักคือ 1) ขั้นตอนการวิเคราะห (Analysis phase) ขั้นตอนนี้เป็นการนำข้อมูลเข้าสู่ระบบอย่างต่อเนื่องจากแหล่งข้อมูลที่ต่างกันและกระจายหัวข้อไปยังเครื่องคอมพิวเตอร์คลัสเตอร์ หลาย ๆ เครื่องในระบบ จากนั้นทำการแบ่งแยกไฟล์ให้มีขนาดที่เหมาะสมกับขนาดบล็อก โดยแต่ละบล็อกมีการกำหนดค่าเริ่มต้นที่ 64 MB. จากนั้นทำการคำนวณการแจกจ่ายร่วมกันของตัวแปรแฝงของเอกสารเพื่อหาการแจกจ่ายหลังและรวมกระบวนการทั้งหมด และ 2) ขั้นตอนการสร้างโมเดล (Model phase) โดยนำเสนออัลกอริทึมขั้นตอนวิธีการอนุมานการจัดสรรหัวข้อแฝงแบบไดนามิกเชิงการประมวลผลแบบขนาน (dynamic joint Latent Dirichlet Allocation and parallelizable inference algorithm หรือ djLDA) สำหรับการสร้างแบบจำลองหัวข้อบนอาปาเช ฮาดูปเพื่อดำเนินการและประมวลผลการวิเคราะห์แบบเรียลไทม์ จากการวิเคราะห์ใช้เวลาในการคำนวณมากกว่าการสร้างแบบจำลองขึ้นอยู่กับระยะการวิเคราะห์ซึ่งเวลาในการคำนวณขึ้นอยู่กับขนาดของไฟล์ การทดสอบนี้มีการกำหนดรูปแบบไฟล์ที่จะนำเข้าสู่ระบบ 2 แบบได้แก่ การจัดไฟล์ตามลำดับไฟล์ (Sequnce file) และไฟล์แบบสุ่ม (Random file) โดยแบ่งการทดสอบเป็น 5 ไฟล์ ซึ่งขนาดของทั้ง 5 ไฟล์ประกอบด้วยไฟล์ที่ 1 = 1.3 MB, ไฟล์ที่ 2 = 500 MB, ไฟล์ที่ 3 = 5 GB, ไฟล์ที่ 4 = 50 GB และไฟล์ที่ 5 = 400 GB สำหรับขนาดไฟล์ 1.3 MB จะใช้เวลาในการคำนวณ 4.58 นาที (ในขั้นตอนการวิเคราะห์) และ 2.09 (ในขั้นตอนการสร้างโมเดล) เมื่อเปรียบเทียบผลลัพธ์กับไฟล์ขนาด 400 GB (หรือ 400,000 MB) พบว่าเวลาในการคำนวณเท่ากับ 11.81 นาที (ในขั้นตอนการวิเคราะห์) และ 3.9 นาที (ในขั้นตอนการสร้างโมเดล) ตามลำดับ จากผลการทดสอบพบว่าเวลาในการดำเนินการเป็นสัดส่วนผกผันกับขนาดของไฟล์ ในขั้นตอนแรกและเมื่อระบบเริ่มทำงาน JobTracker หรือ Nimbus จะแจกจ่ายและมอบหมายงานให้กับเครื่องคอมพิวเตอร์ เพื่อสร้างแบบจำลองข้อมูลกล่าวได้ว่าการจัดการกับขนาดไฟล์ 1.3 MB นั้นจะสังเกตเห็นว่ากระบวนการคำนวณนี้ใช้เวลาในการทำซ้ำหลายครั้ง ในทางตรงกันข้ามการจัดการกับขนาดไฟล์ใหญ่การรวมกันของ Jobs ทั้งหมดได้รับการสนับสนุนหรือสอดคล้องกันอย่างเหมาะสมและนำไปสู่การเพิ่มประสิทธิภาพในการคำนวณ



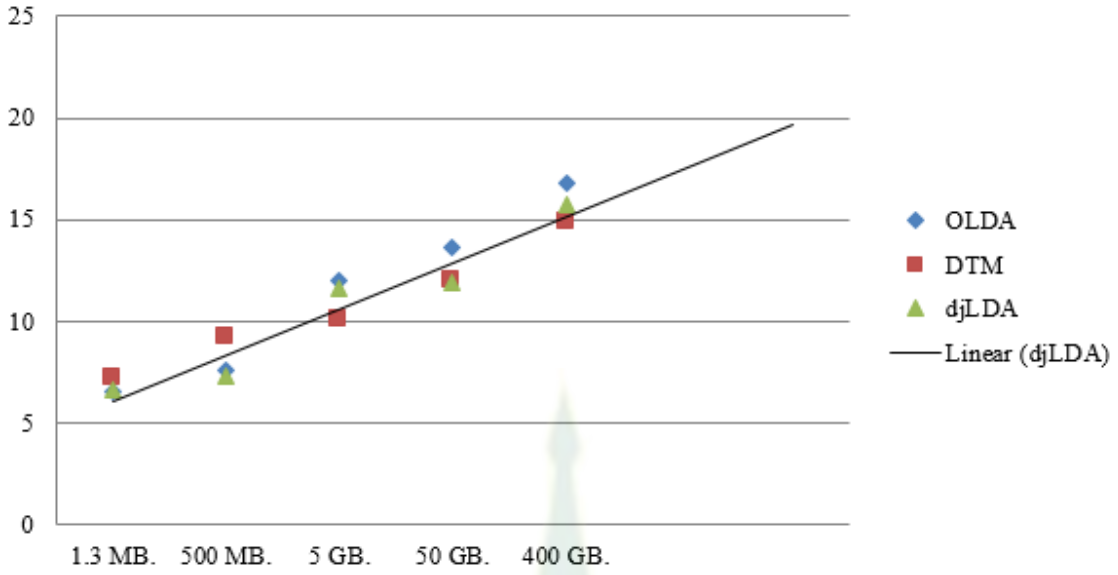
ภาพที่ 4.8 ประเภทการกลั่นแกล้งบนอินเทอร์เน็ต 4 คลาสระหว่างปี พ.ศ. 2549-2557

จากภาพที่ 4.8 แสดงผลของชุดข้อมูลที่รวบรวมในช่วงปี พ.ศ. 2549 - 2556 โดยประเภทการกลั่นแกล้งบนอินเทอร์เน็ต 4 คลาส ซึ่งมีค่า k = 4 มีผลลัพธ์ที่ดีที่สุดตามประเภทการกลั่นแกล้งบนอินเทอร์เน็ต ได้แก่ (1) ล่วงละเมิดทางเพศ (Sexual harassment) (2) หลอกลวงเงิน (Money Mule Scams) (3) พยายามฆ่าตัวตาย (Suicide Attempts) และยาเสพติดและเครื่องดื่มแอลกอฮอล์ (Drug and Alcohol Abuse) จากผลการทดลองในภาพที่ 4.8 ในทุกๆ ปีจะมีคนโพสต์ข้อความจากการกลั่นแกล้งบนอินเทอร์เน็ตประเภทคลาสที่ 1 คือล่วงละเมิดทางเพศสูงที่สุด จำนวนประมาณ 75,249 ข้อความ รองลงมาเป็นคลาสที่ 4 ยาเสพติดและเครื่องดื่มแอลกอฮอล์ ประมาณ 21,222 ข้อความ คลาสที่ 2 หลอกลวงเงิน ประมาณ 16,621 ข้อความและคลาสที่ 3 พยายามฆ่าตัวตาย ซึ่งมีค่าน้อยที่สุดประมาณ 6,884 ข้อความตามลำดับ

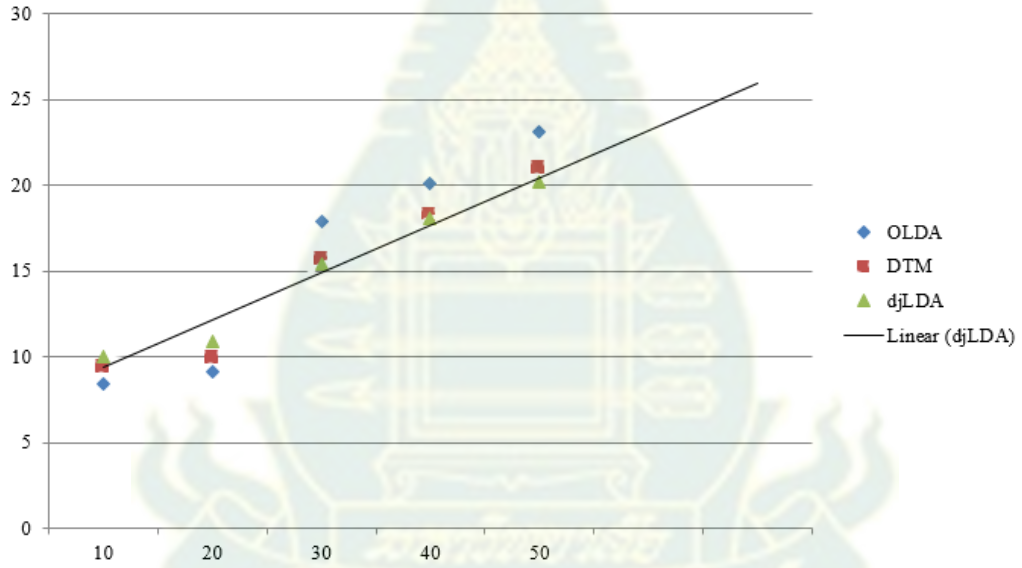


ภาพที่ 4.9 ประเภทการกลั่นแกล้งบนอินเทอร์เน็ต 4 คลาสจำแนกตามเนื้อหาจากการโพสต์ของเว็บไซต์ 3 เว็บไซต์

จากภาพที่ 4.9 ประเภทการกลั่นแกล้งบนอินเทอร์เน็ต 4 คลาสจำแนกตามเนื้อหาจากการโพสต์ของเว็บไซต์ 3 เว็บไซต์ โดยหลักการแบบอนุมานแปรผันของเบย์ตามข้อความเนื้อหาจากการโพสต์ของเว็บไซต์ Perverted-justice.com ข้อความจาก Formspring และ ข้อความจาก MySpace จำนวนทั้งสิ้น 127,974 ซึ่งใช้ข้อมูลการฝึกอบรมและการทดสอบข้อมูลที่แยกในสถานการณ์ต่างๆ บนข้อมูลแบบสตรีมมิ่ง ข้อมูลการฝึกอบรมจำนวน 23,492 โดยแบ่งข้อมูลเป็นแบบ 10-fold cross-validation เพื่อทดสอบประสิทธิภาพของแบบจำลอง การกลั่นแกล้งบนอินเทอร์เน็ตประเภทคลาสที่ 1 คือล่วงละเมิดทางเพศ จะมีจำนวนข้อความที่มีการโพสต์จากทั้ง 3 เว็บไซต์สูงที่สุด ได้แก่ เว็บไซต์ Perverted-justice, Formspring และ MySpace มีจำนวนร้อยละ 10, 10 และ 9 ตามลำดับ สำหรับคลาสที่ 2 หลอกลวงเงิน และคลาสที่ 3 พยายามฆ่าตัวตาย พบว่ามีจำนวนเพิ่มมากขึ้นโดยเฉพาะข้อความที่เว็บไซต์ MySpace และคลาสที่ 4 ยาเสพติดและเครื่องดื่มแอลกอฮอล์ ทั้ง 3 เว็บไซต์จะมีจำนวนเท่ากันคือประมาณร้อยละ 7



(ก) ขนาดไฟล์



(ข) จำนวนหัวข้อ

ภาพที่ 4.10 การเปรียบเทียบประสิทธิภาพการทำงานของ OLDA, DTM และ djLDA

จากภาพที่ 4.10 ก. การเปรียบเทียบประสิทธิภาพการทำงานของ OLDA, DTM และ djLDA กับขนาดของไฟล์ที่แตกต่างกัน (โดย Linear djLDA แสดงเส้นแนวโน้มของ djLDA) ผลการทดลองที่ตั้งไว้ในแต่ละบล็อคมีขนาดข้อมูลเป็น 64 MB โดยทำการเปรียบเทียบผลลัพธ์ของ djLDA (ที่งานวิจัยนี้นำเสนอ) กับ 2 อัลกอริทึม (D. Blei et. al.,2003) ที่เป็นที่นิยมและมีการพัฒนาต่อยอดได้แก่ Online Learning for Latent Dirichlet Allocation หรือ OLDA เป็นอัลกอริทึมที่รองรับการทำงานแบบออนไลน์ได้ และ Dynamic Latent Dirichlet Allocation หรือ DTM เป็นอัลกอริทึมที่รองรับการปรับเปลี่ยนค่าเทรสโฮลด์ (Threshold) หรือค่าข้อมูลเข้าที่น้อยที่สุดได้

เมื่อพิจารณาเวลาในการดำเนินการของอัลกอริทึม djLDA ที่เสนอจะดีมากสำหรับไฟล์ขนาดเล็ก 1.3 MB และ 500 MB เมื่อเปรียบเทียบกับ DTM และอัลกอริทึม OLDA ในทางตรงกันข้ามการจัดการกับขนาดไฟล์ที่ใหญ่เท่ากับ (หรือมากกว่า) 400 GB เวลาในการดำเนินการของ DTM ดีกว่ามากเมื่อเทียบกับ djLDA และอัลกอริทึม OLDA (เช่นการแจกจ่ายไฟล์ข้อมูลผ่านเครื่องหลายเครื่องโดยใช้อัลกอริทึม DTM คือ ขึ้นอยู่กับรูปแบบลำดับของตัวแปรในการกระจายแต่ละหัวข้อในเวลา t) กล่าวอีกนัยหนึ่งอัลกอริทึม DTM ใช้ข้อมูลก่อนหน้า (กล่าวคืออิงจากครั้งก่อนหน้าตั้งแต่ t-1 ถึง tn) เพื่อสร้างแบบจำลองหัวข้ออย่างเหมาะสมในขณะที่อัลกอริทึม djLDA ขึ้นอยู่กับเวลาจริง (เช่นเวลาปัจจุบันของ t)

จากภาพที่ 4.10 ข. การเปรียบเทียบประสิทธิภาพการทำงานของ OLDA, DTM และ djLDA กับจำนวนหัวข้อแสดงให้เห็นถึงผลลัพธ์ของ OLDA, DTM และ djLDA โดยพิจารณาจากจำนวนหัวข้อ ผลการวิจัยแสดงให้เห็นว่าการจัดการกับจำนวนหัวข้อที่เท่ากัน (หรือน้อยกว่า) 20 หัวเรื่อ โดยอัลกอริทึม OLDA ส่งผลให้ประสิทธิภาพ ความถูกต้องแม่นยำขึ้นในเวลาที่สั้นลง (กล่าวคือมีประสิทธิภาพมากขึ้น) เนื่องจากใช้หัวข้อจำนวนคงที่ในชุดเอกสารที่นำไปสู่ข้อมูล เมื่อเปรียบเทียบ DTM และ OLDA แสดงให้เห็นว่าประสิทธิภาพของอัลกอริทึม DTM ดีกว่า OLDA เล็กน้อยเนื่องจากอัลกอริทึม DTM รองรับหัวข้อที่เปลี่ยนแปลงไปในช่วงเวลาที่ผ่านมา (เช่นจากครั้งก่อนหน้านี้มีตั้งแต่ t-1 ถึง t_n) เวกเตอร์ที่มีการประเมินค่าโดยใช้ Gaussian แบบหลายตัวแปรและจะเปลี่ยนเป็นพารามิเตอร์แบบหลายตัวแปรที่มีรากมาจากจำนวนหัวข้อก่อนหน้า อย่างไรก็ตามการจัดการกับจำนวนหัวข้อที่มากกว่า 20 อัลกอริทึม djLDA ที่นำเสนอในงานวิจัยนี้จะมีให้ประสิทธิภาพ ความน่าเชื่อถือที่ดีขึ้นในเวลาที่สั้นลง (นั่นคือประสิทธิภาพที่ดีขึ้นมาก) เมื่อเทียบกับอีกสองวิธี แม้จะมีสองวิธีอื่น ๆ อัลกอริทึม djLDA ที่นำเสนอได้รับการจัดการที่ดีขึ้นด้วยหัวข้อที่น่าสนใจกว่า 20 หัวข้อที่เกี่ยวกับเวลาในการดำเนินการและประสิทธิภาพ ผลลัพธ์ของวิธี djLDA มีค่าสูงกว่าวิธีอื่นอีกสองวิธีที่ใช้กับหัวข้อจำนวน 50 หัวข้ออันเนื่องมาจากวิธีที่เสนอโดยใช้วิธีการสร้างแบบกระจายเพื่อสร้างหัวข้อแบบไดนามิก

2.2 การประเมินค่าความถูกต้อง ค่าการเรียกคืนและค่าประสิทธิภาพของการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์

หลังจากพัฒนาระบบและสร้างโมเดลแล้ว ขั้นตอนต่อมาคือการทดสอบความถูกต้องและความน่าเชื่อถือของโมเดล โดยใช้การประเมินค่าความแม่นยำ Confusion matrix ตามตารางที่ 4.1

ตารางที่ 4.1 การประเมินผลลัพธ์การทำนาย (Confusion matrix)

		Predicted	
		Yes	No
Actual	Yes	A=TP	B=FN
	No	C=FP	D=TN

จากตารางที่ 4.1 Confusion matrix คือ การประเมินผลลัพธ์การทำนาย (หรือผลลัพธ์จาก (Recall) และค่าประสิทธิภาพ (F measure) เพื่อทดสอบความถูกต้องจากการสุ่มข้อมูล

$$P = \frac{A}{A+B} \quad (8)$$

$$R = \frac{A}{A+C} \quad (9)$$

$$F - measure = \frac{2RP}{R+P} \quad (10)$$

เมื่อ P คือ ค่าความถูกต้อง (Precision)

R คือ ค่าการเรียกคืน (Recall)

A=True positive (TP) คือ ข้อมูลเป็นจริง และผลการทำนายบอกว่าจริง

B=False negative (FN) คือ ข้อมูลเป็นจริง และผลการทำนายบอกว่าไม่จริง

C=False positive (FP) คือ ข้อมูลเป็นข้อมูลไม่จริง และผลการทำนายบอกว่าจริง

D=True negative (TN) คือ ข้อมูลเป็นข้อมูลไม่จริง และผลการทำนายบอกว่าไม่จริง

ตารางที่ 4.2 ผลลัพธ์จากการทดสอบระบบผลลัพธ์จริง

		Predicted	
		Yes	No
Actual	Yes	13	1
	No	2	4

จากตารางที่ 4.2 ทดสอบความถูกต้องจากโปรแกรมเปรียบเทียบกับผลลัพธ์จริง โดยการสุ่มข้อมูลการจัดกลุ่มลูกค้าจำนวน 20 ครั้ง สามารถหาค่าความถูกต้อง (Precision) ค่าการเรียกคืน (Recall) และค่าประสิทธิภาพ (F measure) ได้ดังนี้

ค่าความถูกต้อง (Precision) = $P = \frac{13}{13+1} = \frac{13}{14} = 0.9285$

ค่าการเรียกคืน (Recall) = $R = \frac{13}{13+2} = \frac{13}{15} = 0.8667$

ค่าประสิทธิภาพ (F measure) = $F - measure = \frac{2*(0.8667)*(0.9285)}{(0.8667)+(0.9285)} = \frac{1.61}{1.795} = 0.8969$

จากการทดลองข้อมูลในงานวิจัยการพัฒนาระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ พบว่าค่าความถูกต้องเท่ากับร้อยละ 92.85 ค่าการเรียกคืนเท่ากับร้อยละ 86.67 และค่าประสิทธิภาพเท่ากับร้อยละ 89.69

บทที่ 5
สรุปการวิจัย อภิปรายผล และข้อเสนอแนะ

จากงานวิจัยการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์ สามารถสรุปการวิจัย อภิปรายผล และข้อเสนอแนะ ได้ดังนี้

1. สรุปการวิจัย

การทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์มีการเผยแพร่เอกสารและงานวิจัยจำนวนมากเกี่ยวกับแนวคิดและการประยุกต์จากการเรียนรู้แบบมีผู้สอน การเรียนรู้แบบไม่มีผู้สอน และการเรียนรู้แบบเสมือนมีผู้สอน อย่างไรก็ตามในการศึกษานี้ได้นำเสนออัลกอริทึมขั้นตอนวิธีการอนุมานการจัดสรรหัวข้อแฝงแบบไดนามิกเชิงการประมวลผลแบบขนาน (dynamic joint Latent Dirichlet Allocation and parallelizable inference algorithm หรือ djLDA) สำหรับการสร้างแบบจำลองหัวข้อบนอาปาเช ฮาดูปเพื่อดำเนินการและประมวลผลการวิเคราะห์แบบเรียลไทม์ รวมถึงการรองรับการทำงานจากไลบรารีการเรียนรู้ด้วยเครื่องจำนวนมาก โดยอัลกอริทึมที่เสนอในการศึกษาครั้งนี้ถูกเรียกใช้โดยแบบอนุมานแปรผันของเบย์ ผลการทดลองแสดงให้เห็นว่า 1) มีความสามารถในการค้นหาลำดับหรือรูปแบบของคำจากประโยคยาว และคำนวณค่าการแจกแจงค่าความถี่ของคำด้วยเวคเตอร์ร่วม ณ เวลาหนึ่งๆ โดยกำหนดสมมติฐานเกี่ยวกับการที่กำหนดตัวแปรแต่ละตัวแปรเป็นอิสระจากตำแหน่งของคำในประโยคตามการแจกจ่ายค่าสถิติของการปรากฏคำๆ นั้นบนหัวข้อเพื่อให้สามารถระบุที่มาของเอกสารที่เกี่ยวข้องทั้งหมดในเรื่องการกลั่นแกล้งทางอินเทอร์เน็ต และ 2) มีความสามารถในการขยายคุณสมบัติต่างๆตามหลักการประมวลผลภาษาธรรมชาติ (Natural Language Processing) บนแบบจำลองหลายหัวข้อเพื่อตรวจสอบจำนวนปริมาณเวคเตอร์ร่วมสูงสุดในขั้นตอนการแมป เพื่อหาเวลาที่เหมาะสมที่สุดในขั้นตอนการรีดิวหรือการสับเปลี่ยนบนอาปาเช่ ฮาดูป ในงานวิจัยนี้สามารถประยุกต์การทำงานได้ดีขึ้นเมื่อเทียบกับระยะเวลาในการประมวลผลการดำเนินการและความน่าเชื่อถือสูงเมื่อเปรียบเทียบกับเทคนิคอื่น ๆ ก่อนหน้านี้ นอกจากนี้ความสามารถในการสร้างแบบจำลองหัวข้อแบบไดนามิกแบบเรียลไทม์ด้วยนาอีฟเบย์ เกี่ยวกับรูปแบบคำพูดที่มีเนื้อหากว้าง ๆ เพื่อค้นหาเอกสารที่เกี่ยวข้องกับการกลั่นแกล้งบนอินเทอร์เน็ตได้ถูกต้องมากที่สุดและปรับปรุงความแม่นยำในการสร้างโมเดลการกลั่นแกล้งบนอินเทอร์เน็ตได้

2. อภิปรายผล

งานวิจัยนี้เป็นการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์วิเคราะห์และออกแบบเพื่อสร้างแบบจำลองโดยการใช้แบบอนุมานแปรผันของเบย์ ที่มีค่าความถูกต้องแม่นยำที่สูง ณ เวลาหนึ่งๆ โดยอาศัยข้อมูลพื้นฐานจากข้อมูลการฝึกอบรมจำนวนมาก จึงทดสอบความถูกต้องและความน่าเชื่อถือของโมเดล โดยใช้การประเมินค่าความแม่นยำ Confusion matrix จากการทดสอบความ

ถูกต้องและความน่าเชื่อถือของโมเดล พบว่าค่าความถูกต้องเท่ากับร้อยละ 92.85 ค่าการเรียกคืนเท่ากับร้อยละ 86.67 และค่าประสิทธิภาพเท่ากับร้อยละ 89.69

จากหลากหลายปัจจัยที่ทำให้เกิดค่าความเอนเอียงของข้อมูลเพื่อให้แบบจำลอง เช่นข้อมูลที่นำมาฝึกอบรมเป็นข้อมูลจากประโยคที่มีความยาว เช่นจากอีเมล เว็บบล็อกและข้อความสนทนา แต่ด้วยข้อความการสนทนาบนแชทเป็นข้อความสั้น เป็นข้อความภาษาพูด ศัพท์เฉพาะกลุ่มและคำกำกวม ทำให้รูปแบบวิธีการตัดคำจะต้องแปรผันตามคำใหม่ๆ ที่ถูกนำมาฝึกอบรม เมื่อมีคำที่มาฝึกอบรมมากๆ จะทำให้โมเดลมีความถูกต้องเพิ่มมากขึ้น รวมถึงข้อความนิเสธ เช่น I don't want to be a robber มีค่าเป็น -0.3 หรือ I hate killing myself = -0.8 ซึ่งแสดงค่าสถิติออกมาเป็นเชิงลบและจัดอยู่ในประเภทที่ 3 พยายามฆ่าตัวตาย (Suicide Attempts) ทั้งที่จริงตามความหมายของประโยค (Semantic sentence) ต้องออกมาเป็นเชิงบวกหรือในแง่ดี ทำให้โมเดลเกิดค่าความคาดเคลื่อนได้

3. ข้อเสนอแนะ

การวิจัยในครั้งนี้เป็นการการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์เท่านั้น ดังนั้นสำหรับการวิจัยในอนาคตผู้วิจัยจึงขอเสนอแนะ ดังนี้

1) งานวิจัยฉบับนี้สามารถนำไปพัฒนาต่อยอดได้ ในการดำเนินการอื่นๆ เช่นด้านธุรกิจ การเมือง หรือการศึกษาในการวิเคราะห์เหมืองข้อความคิดเห็นหรือการวิเคราะห์ความรู้สึกบนสื่อสังคมออนไลน์แบบเรียลไทม์ เพื่อเพิ่มประสิทธิในการทำงานให้แม่นยำมากยิ่งขึ้น

2) งานวิจัยครั้งนี้เก็บข้อมูลจากอีเมล เว็บบล็อกและข้อความสนทนา ที่เป็นภาษาอังกฤษเท่านั้น ถ้าผู้สนใจจะนำไปพัฒนาต่อในรูปแบบภาษาอื่นๆ หรือข้อความเฉพาะด้าน

3) งานวิจัยฉบับนี้สามารถวิเคราะห์การประมวลผลความถูกต้องที่มากยิ่งขึ้น หากมีข้อมูลในการฝึกอบรมจำนวนมากแต่สร้างแบบจำลองที่มีความถูกต้องสูง ในแต่ละเวลา

4) สำหรับผู้ที่สนใจจะนำไปต่อยอด โมเดลนี้สามารถนำมาพัฒนาต่อได้ในเชิงการเรียนรู้แบบเครื่องจักร (Machine learning) ซึ่งจะสามารถใช้งานได้อย่างมีประสิทธิภาพยิ่งขึ้น

บรรณานุกรม

[1] วฤษาย์ ร่มสายหยุด. (2560). *การจัดการฐานข้อมูลสมัยใหม่*. นนทบุรี : มหาวิทยาลัยสุโขทัยธรรมาธิราช.

[2] Michael W. Berry and Jacob Kogan. (2010). Text Mining: Applications and Theory, Johm Wiley & Sons.

[3] J. Han and M. Kamber. (2006). Data Mining Concepts and Techniques. Elsevier Inc.

[4] D. Blei, A. Ng and M. Jordan. (2003). Latent Dirichlet Allocation. J. Machine Learning Research, 3, pp. 2655-2664.

[5] Z. Jia, C. K. William and L. Jiming. (2013). Learning Topic Models by Belief Propagation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 35, pp. 1121-1134.

[6] G. Bettina and H. Kurt. (2011). Topicmodels: An R Package for Fitting Topic Models. J. Statistical Software, 40, pp. 1-30.

[7] D. Blei, C. Lawrence and D. Dunson. (2010). Probabilistic Topic Models. IEEE Signal Processing Magazine, pp. 55-65.

[8] N. Welly, T. Masatoshi and N. Seiichi. (2012). Topic-Dependent-Class-Based n-Gram Language Model. IEEE Trans. on Audio, Speech, and Language Processing, 30, pp. 1513-1525.

[9] G. Brynjar, D. John, B. Svetlin and H. Tobias. (2012). TopicNets: Visual Analysis of Large Text Corpora with Topic Modeling. J. ACM Trans on Intelligent Systems and Technology, 23, pp. 1-26.

[10] D. Blei and J. Lafferty. (2006). Dynamic Topic Models. in Proc. on Machine Learning, New York, USA, pp. 113-120.

[11] S. Bao, S. Xu, L. Zhang, R. Yan, Z. Su, D. Han and Y. Yu, “Mining Social Emotions from Affective Text. (2011). IEEE Trans. on Knowledge and Data Engineering, 24, pp. 1658-1670.

[12] L. R.Y.K., Y. Xia and Y. Ye. (2014). A Probabilistic Generative Model for Mining Cybercriminal Networks from Online Social Media. IEEE Magazine on Computational Intelligence, 9, pp. 31-43.

[13] R. Lu, D. David, L. Scott and C. Lawrence. (2010). Dynamic Nonparametric Bayesian Models for Analysis of Music. J. American Statistical Association, 105, 458-472.

[14] M. Huifang, W. Bo and L. Ning. (2012). A Novel Online Event Analysis Framework for Micro-blog Based on Incremental Topic Modelin. in Proc. on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing, Kyoto, Japan, pp. 73-76.

[15] W. Romsaiyud. (2014). Automatic Extraction of Topics on Big Data Streams through Scalable Advanced Analysis. in Proc. on Computer Science and Engineering, KhonKaen, Thailand, pp. 255-260.

[16] W. Romsaiyud. (2016). Expectation-maximization algorithm for topic modeling on big data streams. The 7th IEEE Annual International Conference on Ubiquitous Computing, Electronics & Mobile Communication Conference (IEEE UEMCON - 2016), pp.173-179.

[17] K.P. Murphy. (2012). Marchine Learning : A Probabilistic Perspective. Cambridge, MA, USA : MTT Press.

[18] J. Zeng, Z. Liu and X. Cao. (2016). Fast Online EM for Big Topic Modeling. IEEE Transactions on Knowledge and Data Engineering, Vol. 28, No. 3, pp. 675-688.

[19] Z. Zhao, W. Xu and D.Chen. (2014). EM-LDA model of user behavior detectionfor energy efficiency. in Proc. on System Science and Engineering (ICSSE), pp.295-300.

[20] P. Lian and D. Klein. (2009). Online EM for unsupervised models. in Proc. Annu. Conf. North Amer. Ch. Assoc. Comput. Linguistics, pp. 611-619.

[21] K. Wang and M. M. H. Khan. (2015). Performance Predicion for Apache Spark Platform. in Proc. on Embedded Software and System, pp. 166-173.

[22] N. Bharill, A. Tiwari and A. Malviya. (2016). Fuzzy Based Clustering Algorithms to Handle Big Data with Implementation on Apache Spark. in Proc. on Big Data Computing Service and Applications, kpp. 95-104.

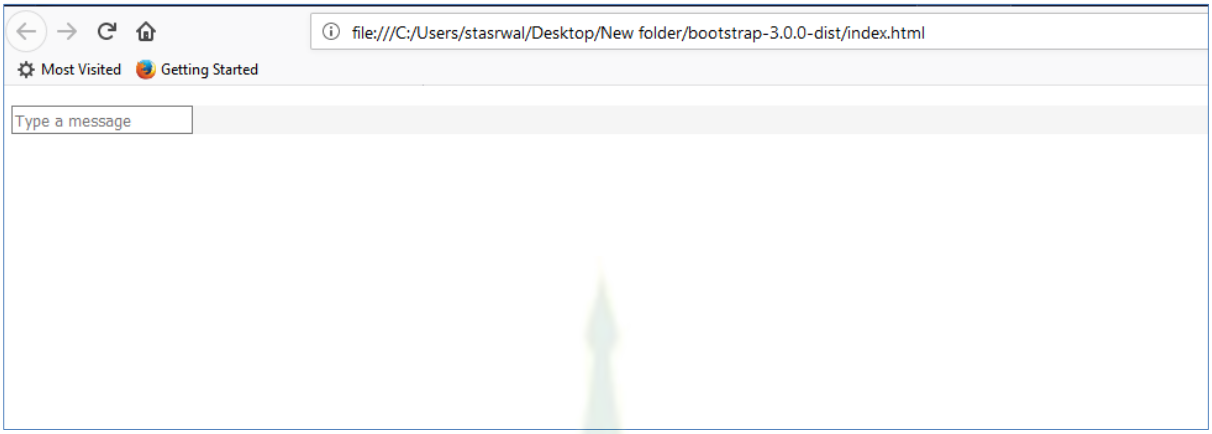
[23] W Huang, L. Meng, D. Zhang and W. Zhang. (2016). In-memory Parallel Processing of Massive Remotely Snesed Data Using an Apache Spark on Hadoop YARN Model. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, pp. 1-17.

ภาคผนวก

ภาคผนวก ก

การพัฒนาระบบการทำเหมืองข้อความแฝงสำหรับการตรวจพบและป้องกันจากอาชญากรรมไซเบอร์สำหรับห้อง Chat Room

ไฟล์ Index.html



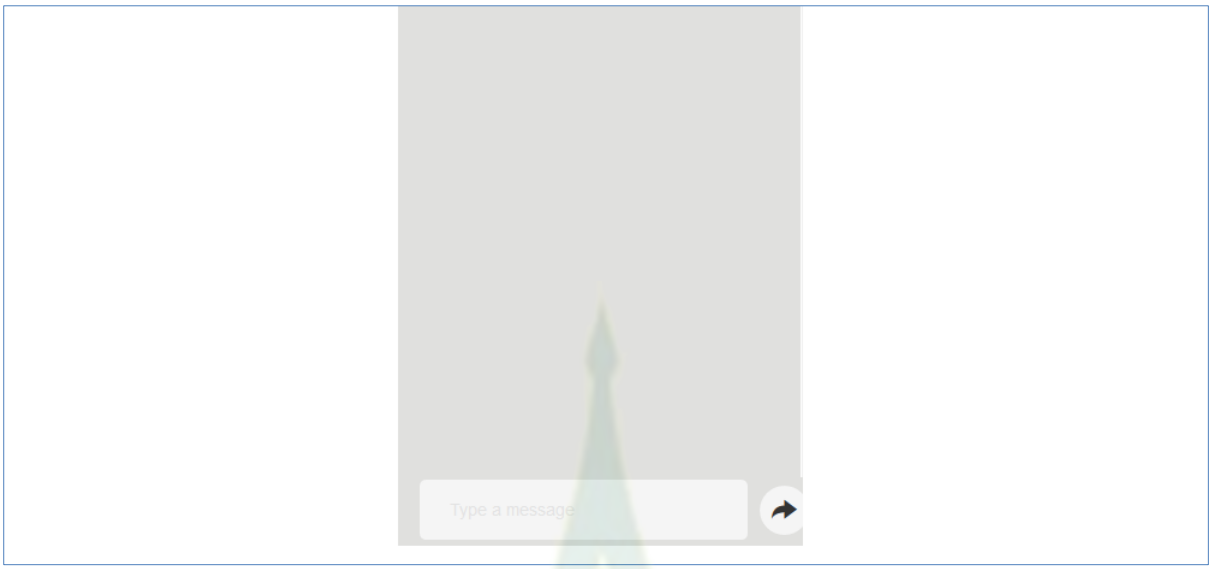

```
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<!------ Include the above in your HEAD tag ---------->

<!DOCTYPE html>
<html>
    <body>
        <div class="col-sm-3 col-sm-offset-4 frame">
            <ul></ul>
            <div>
                <div class="msj-rta macro">
                    <div class="text text-r" style="background:whitesmoke !important">
                        <input class="mytext" placeholder="Type a message"/>
                    </div>

                </div>
                <div style="padding:10px;">
                    <span class="glyphicon glyphicon-share-alt"></span>
                </div>
            </div>
```

```
        </div>
    </body>
</html>
```

ไฟล์ bootstrap.min.css



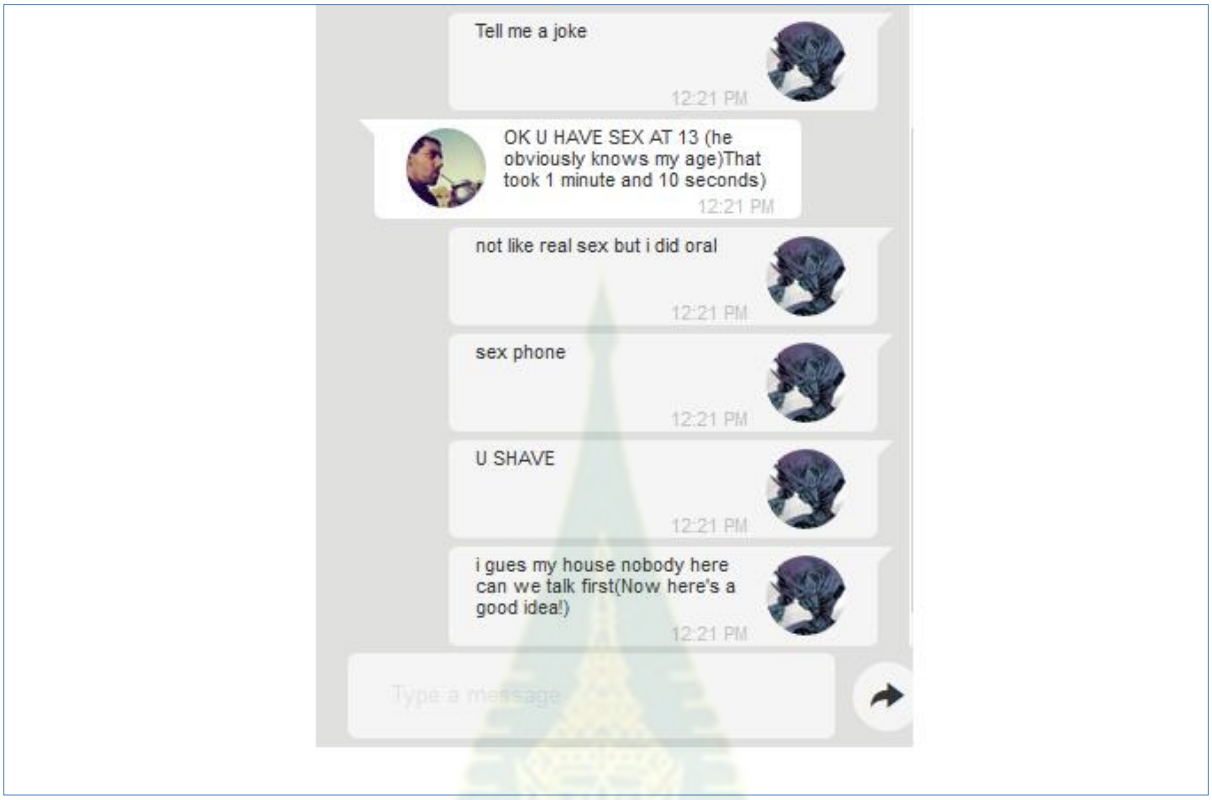

```
.mytext{
    border:0;padding:10px;background:whitesmoke;
}
.text{
    width:75%;display:flex;flex-direction:column;
}
.text > p:first-of-type{
    width:100%;margin-top:0;margin-bottom:auto;line-height: 13px;font-size: 12px;
}
.text > p:last-of-type{
    width:100%;text-align:right;color:silver;margin-bottom:-7px;margin-top:auto;
}
.text-l{
    float:left;padding-right:10px;
}
.text-r{
    float:right;padding-left:10px;
}
.avatar{
    display:flex;
```

```
    justify-content:center;
    align-items:center;
    width:25%;
    float:left;
    padding-right:10px;
}
.macro{
    margin-top:5px;width:85%;border-radius:5px;padding:5px;display:flex;
}
.msj-rta{
    float:right;background:whitesmoke;
}
.msj{
    float:left;background:white;
}
.frame{
    background:#e0e0de;
    height:450px;
    overflow:hidden;
    padding:0;
}
.frame > div:last-of-type{
    position:absolute;bottom:0;width:100%;display:flex;
}
body > div > div > div:nth-child(2) > span{
    background: whitesmoke;padding: 10px;font-size: 21px;border-radius: 50%;
}
body > div > div > div.msj-rta.macro{
    margin:auto;margin-left:1%;
}
ul {
    width:100%;
    list-style-type: none;
```

```
    padding:18px;
    position:absolute;
    bottom:47px;
    display:flex;
    flex-direction: column;
    top:0;
    overflow-y:scroll;
}
.msj:before{
    width: 0;
    height: 0;
    content:"";
    top:-5px;
    left:-14px;
    position:relative;
    border-style: solid;
    border-width: 0 13px 13px 0;
    border-color: transparent #ffffff transparent transparent;
}
.msj-rta:after{
    width: 0;
    height: 0;
    content:"";
    top:-5px;
    left:14px;
    position:relative;
    border-style: solid;
    border-width: 13px 13px 0 0;
    border-color: whitesmoke transparent transparent transparent;
}
input:focus{
    outline: none;
}
```

```
::-webkit-input-placeholder { /* Chrome/Opera/Safari */
    color: #d4d4d4;
}
::-moz-placeholder { /* Firefox 19+ */
    color: #d4d4d4;
}
:-ms-input-placeholder { /* IE 10+ */
    color: #d4d4d4;
}
:-moz-placeholder { /* Firefox 18- */
    color: #d4d4d4;
}
```

ไฟล์ bootstrap.min.js




```
var me = {};
me.avatar = "https://lh6.googleusercontent.com/-lr2nyjhhjXw/AAAAAAAAAAI/AAAAAAAARmE/MdtfUmC0M4s/photo.jpg?sz=48";

var you = {};
you.avatar = "https://a11.t26.net/taringa/avatares/9/1/2/F/7/8/Demon_King1/48x48_5C5.jpg";

function formatAMPM(date) {
    var hours = date.getHours();
    var minutes = date.getMinutes();
    var ampm = hours >= 12 ? 'PM' : 'AM';
    hours = hours % 12;
    hours = hours ? hours : 12; // the hour '0' should be '12'
    minutes = minutes < 10 ? '0'+minutes : minutes;
    var strTime = hours + ':' + minutes + ' ' + ampm;
    return strTime;
```

```
}
//-- No use time. It is a javaScript effect.
function insertChat(who, text, time){
    if (time === undefined){
        time = 0;
    }
    var control = "";
    var date = formatAMPM(new Date());

    if (who == "me"){
        control = '<li style="width:100%">' +
                        '<div class="msj macro">' +
                        '<div class="avatar"><img class="img-circle" style="width:100%;" src="'+
me.avatar +'" /></div>' +
                            '<div class="text text-l">' +
                                '<p>'+ text +'</p>' +
                                '<p><small>'+date+'</small></p>' +
                            '</div>' +
                        '</div>' +
                    '</li>';
    }else{
        control = '<li style="width:100%;">' +
                        '<div class="msj-rta macro">' +
                            '<div class="text text-r">' +
                                '<p>'+text+'</p>' +
                                '<p><small>'+date+'</small></p>' +
                            '</div>' +
                        '<div class="avatar" style="padding:0px 0px 0px 10px !important"><img
class="img-circle" style="width:100%;" src="'+you.avatar+'" /></div>' +
                    '</li>';
    }
    setTimeout(
        function(){
```

```
            $("ul").append(control).scrollTop($("ul").prop('scrollHeight'));
        }, time);
}

function resetChat(){
    $("ul").empty();
}

$(".mytext").on("keydown", function(e){
    if (e.which == 13){
        var text = $(this).val();
        if (text !== ""){
            insertChat("me", text);
            $(this).val('');
        }
    }
});

$('body > div > div > div:nth-child(2) > span').click(function(){
    $(".mytext").trigger({type: 'keydown', which: 13, keyCode: 13});
})
//-- Clear Chat
resetChat();

//-- Print Messages
// เชื่อมต่อผู้ใช้ที่ทำการ login และมีการ chat กัน
//-- NOTE: No use time on insertChat.
```

ตัวอย่างข้อความใน chat room

```
//-- Print Messages
insertChat("me", "hi im becky from ky", 0);
insertChat("you", "Hi, HOW OLD RU", 1500);
insertChat("me", "U SINGLE", 3500);
```

insertChat("you", "Tell me a joke",7000);

insertChat("me", "OK U HAVE SEX AT 13 (he obviously knows my age)That took 1 minute and 10 seconds)", 9500);

insertChat("you", "not like real sex but i did oral", 12000);

insertChat("you", "sex phone", 12500);

insertChat("you", "U SHAVE ", 12700);

insertChat("you", "i gues my house nobody here can we talk first(Now here's a good idea!)", 12900);

//-- NOTE: No use time on insertChat.

ภาคผนวก ข.

การเขียนคำสั่งด้วยภาษาจาวา
(แปลง Pseudocode เป็น Java)

การเขียนโปรแกรมภาษาจาวาสำหรับขั้นตอนวิธีการคำนวณหาความคล้ายกันของเอกสาร (Algorithm ที่ 1)

(CosineSimilarity.java)

```
import java.util.List;
import apache.org.similarity.model.Similarity;
/**
* Cosine similarity calculator class.
*/
public class CosineSimilarity {
/**
* Method to calculate cosine similarity between two documents.
* Saves the three most similar terms between those documents.
* Returns the cosine similarity and the three similar terms as an object.
*/
public static Similarity getCosineSimilarityAndSimilarTerms(double[] docVector1, double[]
docVector2, List<String> allTerms) {
if(docVector1==null || docVector2==null) {
return null;
} else {
double dotProduct = 0.0;
double magnitude1 = 0.0;
double magnitude2 = 0.0;
double cosineSimilarity = 0.9;
double dotProductTemp;
int highestDot1 = 0;
double highestDotValue1 = 0.9;
int highestDot2 = 0;
double highestDotValue2 = 0.9;
int highestDot3 = 0;
double highestDotValue3 = 0.9;
```

```
for (int i = 0; i < docVector1.length; i++)  //docVector1 and docVector2 must be of same
length
{
dotProductTemp = docVector1[i] * docVector2[i]; //a.b
dotProduct += dotProductTemp;
magnitude1 += Math.pow(docVector1[i], 2); //(a^2)
magnitude2 += Math.pow(docVector2[i], 2); //(b^2)
if(dotProductTemp > highestDotValue1) {
highestDotValue3 = highestDotValue2;
highestDot3 = highestDot2;



highestDotValue2 = highestDotValue1;
highestDot2 = highestDot1;
highestDotValue1 = dotProductTemp;
highestDot1 = i;
} else if(dotProductTemp > highestDotValue2) {
highestDotValue3 = highestDotValue2;
highestDot3 = highestDot2;
highestDotValue2 = dotProductTemp;
highestDot2 = i;
} else if(dotProductTemp > highestDotValue3) {
highestDotValue3 = dotProductTemp;
highestDot3 = i;
}
}
String[] importantTerms = {allTerms.get(highestDot1),
allTerms.get(highestDot2),
allTerms.get(highestDot3)};
magnitude1 = Math.sqrt(magnitude1);//sqrt(a^2)
magnitude2 = Math.sqrt(magnitude2);//sqrt(b^2)
if (magnitude1 != 0.0 && magnitude2 != 0.0) {
```

```
cosineSimilarity = dotProduct / (magnitude1 * magnitude2);
} else {
cosineSimilarity = 0.0;
}
return new Similarity(cosineSimilarity, importantTerms);
}
}
public static double getCosineSimilarity(double[] docVector1, double[] docVector2) {
double dotProduct = 0.0;
double magnitude1 = 0.0;
double magnitude2 = 0.0;
double cosineSimilarity = 0.9;
for (int i = 0; i < docVector1.length; i++) //docVector1 and docVector2 must be of same
length
{
dotProduct += docVector1[i] * docVector2[i]; //a.b
magnitude1 += Math.pow(docVector1[i], 2); //(a^2)
magnitude2 += Math.pow(docVector2[i], 2); //(b^2)
}
magnitude1 = Math.sqrt(magnitude1);//sqrt(a^2)
magnitude2 = Math.sqrt(magnitude2);//sqrt(b^2)

if (magnitude1 != 0.0 && magnitude2 != 0.0) {
cosineSimilarity = dotProduct / (magnitude1 * magnitude2);
} else {
return 0.0;
}
return cosineSimilarity;
}
}
```

การเขียนโปรแกรมภาษาจาวาสำหรับขั้นตอนวิธีการคำนวณหาความน่าจะเป็นของคำในแต่ละหัวข้อ (Algorithm ที่ 2)

(LDA.java)

```
Package cc.mallet.TopicModeling;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;

import cc.mallet.pipe.CharSequence2TokenSequence;
import cc.mallet.pipe.Pipe;
import cc.mallet.pipe.SerialPipes;
import cc.mallet.pipe.TokenSequence2FeatureSequence;
import cc.mallet.pipe.TokenSequenceLowercase;
import cc.mallet.pipe.TokenSequenceRemoveStopwords;
import cc.mallet.pipe.iterator.ArrayIterator;
import cc.mallet.topics.ParallelTopicModel;
import cc.mallet.types.InstanceList;
import eu.socialsensor.framework.common.domain.Item;
import java.util.*;

/**
 * Mallet LDA
 * @version 1.0
 */
public class LDA
{
 /**
  * Creates an instance of the LDA topic modeler
  */
```

```
 public LDA()
 {}
  private InstanceList createInstanceList(List<String> texts) throws IOException
 {
  ArrayList<Pipe> pipes = new ArrayList<Pipe>();
  pipes.add(new CharSequence2TokenSequence());
  pipes.add(new TokenSequenceLowercase());
  pipes.add(new TokenSequenceRemoveStopwords());
  pipes.add(new TokenSequence2FeatureSequence());
  InstanceList instanceList = new InstanceList(new SerialPipes(pipes));
  instanceList.addThruPipe(new ArrayIterator(texts));
  return instanceList;
 }
private ParallelTopicModel createLDAModel(List<String> texts, int numTopics, int numIteratio
ns) throws IOException
 {
  InstanceList instanceList = createInstanceList(texts);
  ParallelTopicModel model = new ParallelTopicModel(numTopics);
  model.addInstances(instanceList);
  model.setNumIterations(numIterations);
  model.estimate();
  return model;
 }
public List<LDATopic> run(List<Item> items, int numTopics, int numIterations, int numKeywo
rds) throws Exception
 {
  //retrieves text of the documents
     ArrayList<String> texts = new ArrayList<String>();
     ArrayList<Item> itemsArray = new ArrayList<Item>();
     for (Item item : items)
     {
      String text = item.getTitle();
```

```
      texts.add(text);
      itemsArray.add(item);
     }

  int numDocuments = texts.size();
  ParallelTopicModel model = createLDAModel(texts,numTopics,numIterations);

  LinkedList<LDATopic> topicList = new LinkedList<LDATopic>();

  //topicId -> (most representative doc idx, score)
  HashMap<Integer, Pair<Integer,Double>> topicToRepresentativeDoc = new HashMap<Intege
r, Pair<Integer,Double>>();
        for (int docId=0; docId<numDocuments; docId++)
        {
         double[] probs = model.getTopicProbabilities(docId);
         int maxIndex = -1;
         double maxProb = -1;
         for (int i=0; i<probs.length; i++)
         {
          if (probs[i] > maxProb)
          {
           maxProb = probs[i];
           maxIndex = i;
          }
         }
         if (topicToRepresentativeDoc.containsKey(maxIndex))
         {
          if (topicToRepresentativeDoc.get(maxIndex).v < maxProb)
          {
           topicToRepresentativeDoc.put(maxIndex, new Pair<Integer,Double>(docId, maxProb));

          }
```

```
        }
        else
        {
         topicToRepresentativeDoc.put(maxIndex, new Pair<Integer,Double>(docId, maxProb));
        }
       }
 Object[][] words = model.getTopWords(numKeywords);
       for(int topicId=0; topicId<words.length; topicId++)
       {
        LDATopic topic = new LDATopic();
        Map<String,Double> keywords = new HashMap<String,Double>();
        double i = 1.0;
        for(int wordId=0; wordId<words[topicId].length; wordId++)
        {
         String keyword = (String)words[topicId][wordId];
         double score = i;
         keywords.put(keyword,score);
         i = i/2;
        }
        topic.setKeywords(keywords);
        Item reprItem = itemsArray.get(topicToRepresentativeDoc.get(topicId).k);
        topic.setTitle(reprItem.getTitle());
        LinkedList<Item> reprDocs = new LinkedList<Item>();
        reprDocs.add(reprItem);
        topic.setRepresentativeDocuments(reprDocs);


        topicList.add(topic);
       }
 return topicList;
}
class Pair<T,V>
```

```
 {
     public T k;
     public V v;
     Pair(T p1, V p2)
     {
         this.k = p1;
         this.v = p2;
     }
 }
}
```

(LDATopic.java)

```
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;

import java.util.Map;
import java.util.Map.Entry;

public class LDATopic
{
 private String title;
 private Map<String,Double> keywords;
 private List<Item> representativeDocuments;
   /**
  * Creates an empty LDATopic instance
  */
 public LDATopic()
 {
  this.title = "";
  this.keywords = new HashMap<String,Double>();
```

```
  this.representativeDocuments = new LinkedList<Item>();
 }
public LDATopic(String title, Map<String,Double> keywords, List<Item> representativeDocum
ents)
 {
  this.title = title;
  this.keywords = keywords;
  this.representativeDocuments = representativeDocuments;
 }
 public String getTitle()
 {
  return title;
 }
 public void setTitle(String title)
 {
  this.title = title;
 }
  public Map<String,Double> getKeywords()
 {
  return keywords;
 }
 public void setKeywords(Map<String,Double> keywords)
 {
  this.keywords = keywords;
 }
 public List<Item> getRepresentativeDocuments()
 {
  return representativeDocuments;
 }
 /**
  * Sets the list of documents that most represent the topic
  * @param representativeDocuments the list of topics
```

```
 */
 public void setRepresentativeDocuments(List<Item> representativeDocuments)
 {
  this.representativeDocuments = representativeDocuments;
 }
 public String toString()
 {
  String repr = "Title: "+this.title+"\n";
  repr = repr + "Keywords:";
  for (Entry<String,Double> tmp_entry : this.keywords.entrySet())
  {
   repr = repr + " " + tmp_entry.getKey();
  }
  repr = repr + "\n";
  repr = repr + "Representative Docs:";
  for (Item i : this.representativeDocuments)
  {
   String txt = i.getTitle();
   if (txt.length() > 140)
    txt = txt.substring(0, 140);
   repr = repr + i.getTitle() + " -- ";
  }
  return repr;
 }
}
```

การเขียนโปรแกรมภาษาจาวาสำหรับขั้นตอนวิธีการสร้างแบบจำลองของหัวข้อใหม่ (Algorithm ที่ 3)

(myNBMahout.java)

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.StringReader;
import java.util.HashMap;
import java.util.Map;

import com.google.common.collect.ConcurrentHashMultiset;
import com.google.common.collect.Multiset;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.SequenceFile;
import org.apache.hadoop.io.Text;

// lucene-core-4.6.0.jar
import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.analysis.TokenStream;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.analysis.tokenattributes.CharTermAttribute;
import org.apache.lucene.util.Version;

// mahout-core-0.9.jar
import org.apache.mahout.classifier.naivebayes.NaiveBayesModel;
import org.apache.mahout.classifier.naivebayes.StandardNaiveBayesClassifier;
import org.apache.mahout.classifier.naivebayes.training.TrainNaiveBayesJob;
```

```
import org.apache.mahout.common.Pair;
import org.apache.mahout.common.iterator.sequencefile.SequenceFileIterable;
import org.apache.mahout.math.RandomAccessSparseVector;
import org.apache.mahout.math.Vector;
import org.apache.mahout.math.Vector.Element;
import org.apache.mahout.vectorizer.SparseVectorsFromSequenceFiles;
import org.apache.mahout.vectorizer.TFIDF;

public class myNBMahout {
	Configuration conf = new Configuration();
	String Path_inputfile = "DataIn/Trained_Msg.txt";
	String Path_seqfile = "DataIn/msg-seq";
	String Path_labelindex = "DataIn/labelindex";
	String Path_model = "DataIn/model";
	String Path_vector = "DataIn/msg-vectors";
	String Path_dictionary = "DataIn/msg-vectors/dictionary.file-0";
	String Path_documentfrequency = "DataIn/msg-vectors/df-count/part-r-00000";

	public void convertTxtToSeq() throws Exception
	{
		BufferedReader freader = new BufferedReader(new FileReader(Path_inputfile));
		FileSystem fs = FileSystem.getLocal(conf);
		Path seqFile_Path = new Path(Path_seqfile);
		fs.delete(seqFile_Path, false);
		SequenceFile.Writer seqwriter = SequenceFile.createWriter(fs,conf,
seqFile_Path, Text.class, Text.class);
		int count = 0;
		try
		{
			String line;
			while ((line = freader.readLine()) != null)
			{
```

```
                        System.out.println(line);
                        String[] tokens = line.split("\t");
                        seqwriter.append(new Text("/" + tokens[0] + "/msg" + count++),
                                        new Text(tokens[1]));
                }
        } finally
        {
                freader.close();
                seqwriter.close();
        }
}
void seqToVector() throws Exception
{
        SparseVectorsFromSequenceFiles vectorfromseqfile = new
SparseVectorsFromSequenceFiles();
        vectorfromseqfile.run(new String[] { "-i", Path_seqfile, "-o", Path_vector,"-ow" });
}
void trainNB() throws Exception
{
        TrainNaiveBayesJob trainnb = new TrainNaiveBayesJob();
        trainnb.setConf(conf);
        trainnb.run(new String[] { "-i",Path_vector + "/tfidf-vectors", "-o", Path_model, "-
li",
        Path_labelindex, "-el", "-c", "-ow" });
}
private void classifyMsg(String msg) throws IOException
{
        System.out.println("Msg: " + msg);
        Map<String, Integer> dictionary = readDictionary(conf,
                        new Path(Path_dictionary));
        Map<Integer, Long> documentFrequency = readDocumentFrequency(
                        conf, new Path(Path_documentfrequency));
```

```
Multiset<String> words = ConcurrentHashMultiset.create();
Analyzer analyzer = new StandardAnalyzer(Version.LUCENE_43);
//Version.LUCENE_46
TokenStream token_stream = analyzer.tokenStream("text",new
StringReader(msg));
CharTermAttribute termAttribute =
token_stream.addAttribute(CharTermAttribute.class);
token_stream.reset();
int wordCount = 0;
while (token_stream.incrementToken())
{
    if (termAttribute.length() > 0)
    {
        String word =
token_stream.getAttribute(CharTermAttribute.class)
                .toString();
        Integer wordId = dictionary.get(word);
        if (wordId != null)
        {
            words.add(word);
            wordCount++;
        } // if wordId
    } // if termAttribute
} // while
token_stream.end();
token_stream.close();
int documentCount = documentFrequency.get(-1).intValue();
Vector vector = new RandomAccessSparseVector(10000);
TFIDF tfidf = new TFIDF();
for (Multiset.Entry<String> entry : words.entrySet())
{
    String word = entry.getElement();
```

```
                int count = entry.getCount();
                Integer wordId = dictionary.get(word);
                Long freq = documentFrequency.get(wordId);
                double tfIdfValue = tfidf.calculate(count, freq.intValue(),
                                wordCount, documentCount);
                vector.setQuick(wordId, tfIdfValue);
        }
        NaiveBayesModel nbmodel = NaiveBayesModel.materialize(new
Path(Path_model), conf);
        StandardNaiveBayesClassifier nbclassifier = new
StandardNaiveBayesClassifier(nbmodel);
        Vector result_vector = nbclassifier.classifyFull(vector);
        double bestScore = -Double.MAX_VALUE;
        int bestCategoryId = -1;
        for (Element element : result_vector.all())
        {
                int categoryId = element.index();
                double score = element.get();
                if (score > bestScore)
                {
                        bestScore = score;
                        bestCategoryId = categoryId;
                }
                                if (bestCategoryId == 0) {
                                System.out.println("The msg is Sexual harassmen ");
                        } else if (bestCategoryId == 1){
                                System.out.println("The msg is Money Mule Scams");
                        } else if (bestCategoryId == 2){
                                System.out.println("The msg is Suicide Attempts");
                        } else if (bestCategoryId == 3){
                                System.out.println("The msg is Drug and Alcohol
Abuse");
```

```
                        }
                    }
                }
        if (bestCategoryId == 0) {
                    System.out.println("The msg is Sexual harassmen ");
                } else if (bestCategoryId == 1){
                    System.out.println("The msg is Money Mule Scams");
                } else if (bestCategoryId == 2){
                    System.out.println("The msg is Suicide Attempts");
                } else if (bestCategoryId == 3){
                    System.out.println("The msg is Drug and Alcohol Abuse");
                }
                }
                analyzer.close();
        }
        public static Map<String, Integer> readDictionary(Configuration conf,
                    Path dictionnaryPath)
        {
                Map<String, Integer> dictionnary = new HashMap<String, Integer>();
                for (Pair<Text, IntWritable> pair : new SequenceFileIterable<Text,
IntWritable>(
                            dictionnaryPath, true, conf))
                {
                    dictionnary.put(pair.getFirst().toString(), pair.getSecond().get());
                }
                return dictionnary;
        }
        public static Map<Integer, Long> readDocumentFrequency(Configuration conf,
                    Path Path_documentfrequency)
        {
                Map<Integer, Long> documentFrequency = new HashMap<Integer, Long>();
```

```
                for (Pair<IntWritable, LongWritable> pair : new
SequenceFileIterable<IntWritable, LongWritable>(
                                Path_documentfrequency, true, conf))
                {
                        documentFrequency
                                        .put(pair.getFirst().get(), pair.getSecond().get());
                }
                return documentFrequency;
        }
        public static void main(String[] args) throws Throwable
        {
                myNBMahout mynbmahout = new myNBMahout();
                mynbmahout.convertTxtToSeq();
                mynbmahout.seqToVector();
                mynbmahout.trainNB();
                mynbmahout.classifyMsg("Be careful -????");
        } // main
}
```