รายงานวิจัยฉบับสมบูรณ์

โครงการ การทำเหมืองเว็บไทยโดยเทคนิคการเรียนรู้ของเครื่องและการโปรแกรมตรรกะเชิงอุปนัย

Thai Web Mining Using Machine Learning and Inductive Logic Programming

โดย บุญเสริม กิจศิริกุล

พฤศจิกายน 2546

สัญญาเลขที่ RSA/08/2544

รายงานวิจัยฉบับสมบูรณ์

โครงการ การทำเหมืองเว็บไทยโดยเทคนิคการเรียนรู้ของเครื่องและการโปรแกรมตรรกะเชิงอุปนัย

Thai Web Mining using Machine Learning and Inductive Logic Programming

บุญเสริม กิจศิริกุล
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

(ความเห็นในรายงานนี้เป็นของผู้วิจัย สกว.ไม่จำเป็นต้องเห็นด้วยเสมอไป)

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณสำนักงานกองทุนสนับสนุนการวิจัยที่ได้ให้ทุนตลอดระยะเวลา 3 ปี (1 ธค. 2543 — 30 พย. 2546) สำหรับโครงการ “การทำเหมืองเว็บไทยโดยเทคนิคการเรียนรู้ของเครื่องและการโปรแกรมตรรกะเชิงอุปนัย” และขอขอบคุณ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ให้โอกาส สถานที่ ตลอดจนทรัพยากรต่างๆ ที่ใช้ในงานวิจัย ขอขอบคุณผู้ช่วยวิจัยและผู้มีส่วนร่วมในโครงการนี้ทุกท่าน

บุญเสริม กิจศิริกุล

พฤศจิกายน 2546

บทคัดย่อ

รหัสโครงการ: RSA/08/2544

ชื่อโครงการ: การทำเหมืองเว็บไทยโดยเทคนิคการเรียนรู้ของเครื่องและการโปรแกรมตรรกะเชิงอุปนัย

ชื่อนักวิจัย: นายบุญเสริม กิจศิริกุล

E-mail Address: boonserm.k@chula.ac.th

ระยะเวลาโครงการ: 1 ธค. 2543 – 30 พย. 2546

ปัจจุบันการเติบโตของอินเตอร์เน็ตเป็นไปอย่างรวดเร็วมาก มีเว็บเพจจำนวนหลายพันล้านเพจที่เข้าถึงได้บนอินเตอร์เน็ตและมีเว็บเพจหลายล้านเพจเกิดขึ้นใหม่ทุกวัน ผู้ใช้ข้อมูลบนอินเตอร์เน็ตต้องใช้เวลาและความพยายามอย่างมากในการค้นหาเอกสารที่ต้องการ ระบบค้นหาเว็บในปัจจุบันครอบคลุมเอกสารเพียงบางส่วนของเอกสารทั้งหมดบนอินเตอร์เน็ต ระบบค้นหาเว็บเหล่านี้มักค้นคืนได้เอกสารที่ไม่ตรงกับความต้องการของผู้ใช้เพราะใช้การค้นหาตามคำสำคัญ ส่วนระบบไดเร็กทอรีโครงข่าย เช่น Yahoo! จัดโครงสร้างของเว็บเพจแยกตามหมวดหมู่ของเว็บเพจ ทำให้สามารถค้นคืนเอกสารได้ตรงกับความต้องการของผู้ใช้มากกว่า อย่างไรก็ดี ระบบไดเร็กทอรีโครงข่ายก็มีข้อจำกัดที่ปริมาณเว็บเพจที่ครอบคลุมจะน้อยมาก เนื่องจากต้องใช้แรงงานคนจำนวนมากในการแบ่งหมวดหมู่ของเอกสาร

ในงานวิจัย เรานำเสนอวิธีการเพื่อแก้ปัญหานี้โดยการจำแนกประเภทของเว็บเพจออกเป็นหมวดหมู่อย่างอัตโนมัติ วิธีการที่นำเสนอนี้ใช้เทคนิคของการเรียนรู้ของเครื่องและการโปรแกรมตรรกะเชิงอุปนัย หัวข้อสำคัญที่เน้นทำวิจัยคือ (1) การวิจัยพื้นฐานเพื่อเพิ่มประสิทธิภาพของการโปรแกรมตรรกะเชิงอุปนัย และ (2) การวิจัยเทคนิคการเรียนรู้ของเครื่องที่สามารถใช้ประโยชน์จากข้อมูลแบบไม่มีฉลาก

การโปรแกรมตรรกะเชิงอุปนัย (ไอแอลพี) สามารถนำมาประยุกต์ใช้กับการจำแนกเว็บเพจเป็นหมวดหมู่โดยอัตโนมัติได้ และมีจุดเด่นอยู่ที่ผู้สอนสามารถป้อนความรู้เบื้องต้นในรูปแบบของโปรแกรมตรรกะอันดับที่หนึ่งได้ ซึ่งจะช่วยให้การจำแนกข้อมูลทำได้อย่างมีประสิทธิภาพยิ่งขึ้น ไอแอลพีจะให้เอ้าท์พุตเป็นเซตของกฎที่สอดคล้องกับตัวอย่างสอน อย่างไรก็ดีไอแอลพีมีข้อด้อยที่กฎที่สร้างได้อาจไม่ตรงพอดีกับตัวอย่างทดสอบโดยเฉพาะข้อมูลที่มีสัญญาณรบกวน และทำให้ข้อมูลเหล่านี้ไม่สามารถจำแนกหมวดหมู่ได้อย่างถูกต้อง ในกรณีเช่นนี้ เราจำเป็นต้องใช้วิธีการที่สามารถหากฎที่ตรงกับข้อมูลมากที่สุด ในงานวิจัยนี้ เราใช้กระบวนการดึงลักษณะสำคัญและวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก เพื่อหากฎที่ตรงกับข้อมูลมากที่สุด ผลการทดลองที่ได้แสดงให้เห็นว่า วิธีการดึงลักษณะสำคัญและแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กทำให้การโปรแกรมตรรกะเชิงอุปนัยมีประสิทธิภาพสูงขึ้น

นอกจากนั้นในงานวิจัยนี้เรายังได้นำเสนอวิธีการเรียนรู้แบบใหม่ที่เรียกว่า การสอนไขว้แบบวนซ้ำ ซึ่งสามารถใช้ประโยชน์จากข้อมูลไม่มีฉลากได้ วิธีการนี้มีข้อดีกว่าการเรียนรู้แบบสอนทั่วไปที่ต้องใช้ข้อมูลมีฉลากทั้งหมดและต้องอาศัยแรงงานคนจำนวนมากเพื่อติดฉลากให้กับข้อมูล แนวคิดของการสอนไขว้แบบวนซ้ำคือการรวมตัวแยกแยะย่อยสองตัว ซึ่งจะสอนโต้ตอบกันเองไปมาเพื่อปรับปรุงประสิทธิภาพของระบบโดยรวม เมื่อให้ข้อมูลไม่มีฉลากสองเซต แต่ละเซตสำหรับตัวแยกแยะย่อยแต่ละตัว ตัวแยกแยะจะติดฉลากให้กับตัวแยกแยะอีกตัว ด้วยการโต้ตอบที่ดีระหว่างตัวแยกแยะทั้งสอง ทำให้ประสิทธิภาพของระบบโดยรวมค่อยๆ ดีขึ้น ผลการทดลองแสดงให้เห็นว่าวิธีการที่นำเสนอสามารถใช้ประโยชน์จากข้อมูลที่ไม่มีฉลากได้อย่างมีประสิทธิภาพ เรายังได้ปรับปรุงประสิทธิภาพของการสอนไขว้แบบวนซ้ำโดยการใช้การโปรแกรมตรรกะเชิงอุปนัยมาเป็นตัวแยกแยะย่อย ผลที่ได้พบว่า อัลกอริทึมการสอนไขว้แบบวนซ้ำชนิดการโปรแกรมตรรกะเชิงอุปนัยสามารถเพิ่มประสิทธิภาพของการสอนไขว้แบบวนซ้ำแบบดั้งเดิมได้อย่างมาก และมีประสิทธิภาพสูงกว่าวิธีการอื่นๆ ทุกวิธีที่นำทดสอบรวมทั้งการเรียนรู้แบบสอนซึ่งใช้ข้อมูลแบบมีฉลากทั้งหมดอีกด้วย

คำหลัก: การทำเหมืองเว็บ, การโปรแกรมตรรกะเชิงอุปนัย, การเรียนรู้ของเครื่อง

Abstract

Project Code: RSA/08/2544

Project Title: Thai Web Mining using Machine Learning and Inductive Logic Programming

Investigator: Boonserm Kijsirikul

E-mail Address: boonserm.k@chula.ac.th

Project Period: December 1, 2000 – November 30, 2003

With the explosive growth of the Internet, today there are billions Web page accessible on the Internet with several million pages being added daily. The user must spend a great deal of time and effort looking for document he needs. Web search engines available today cover only some fraction of all documents in the Internet, and because of the use of keyword search, they also return documents not related to the real user-interest. Net directory systems, such as Yahoo!, organize their Web resources in category-specific style and thus can provide documents better matching the user needs. However, these systems have the limitations that the number of Web pages covered by the systems is even small as they need a lot of human effort to categorize the documents.

In this research, we propose an approach to solving this problem by automatically classifying Web pages into categories. The proposed approach employs the techniques of machine learning and inductive logic programming. Two main issues are studied in this research; (1) a basic research for improving techniques of inductive logic programming, and (2) machine learning techniques that can make use of unlabeled data.

Inductive Logic Programming (ILP) can be applied to automatic classification of Web pages. It has advantage that the user can provide background knowledge in the form of first-order logic programs. This makes more efficient classification of data. Given training examples and background knowledge as the input, ILP outputs a set of rules that is consistent with training examples. However, ILP has disadvantage that the obtained rules may not be exactly match with test data, especially noisy data, and thus the data cannot be correctly classified. In such a case, we need a method that finds the best matching rule. In this work, we employ first-order feature extraction and backpropagation neural networks to find the best matching rule. The experimental results show that feature extraction and the neural network improve the performance of ILP alone.

In the research, we also propose a new learning method, called Iterative Cross-Training (ICT) which can make use of unlabeled data. The method has advantage over traditional supervised learning which needs all labeled data and requires a lot of human effort to label the data. The idea of ICT is to combine two sub-classifiers which iteratively train each other for improving the performance of the whole system. Given two sets of unlabeled data, each of which is for each classifier, the classifier labels data for the other. With good interaction between two classifiers, the performance of the whole system is increasingly improved. The experimental results show that the proposed method can effectively use unlabeled data. We then further improve the performance of ICT by employing an ILP system as a sub-classifier. The results show that ICT with ILP gives significant improvement over the original ICT, and performs better than other methods tested in our experiment including the supervised learning method which uses all labeled data.

Keywords: Web Mining, Inductive Logic Programming, Machine Learning

II. เนื้อหางานวิจัย

ปัจจุบันการเติบโตของเว็บไซต์เป็นไปอย่างรวดเร็วมาก ในวันหนึ่งๆ มีเว็บเพจ (Web page) เกิดขึ้นใหม่ประมาณ 1.5 ล้านเพจ และมีเว็บเพจทั้งหมดมากกว่า 1,000 ล้านเพจ สำรวจในปี 2000 [Pierre, 2000] ทำให้ผู้ใช้ข้อมูลบนอินเตอร์เน็ตประสบความยากลำบากในการค้นหาข้อมูลที่ต้องการ ระบบค้นหาเว็บ (Web search engine) หนึ่งๆ ไม่สามารถจะค้นหาและเก็บข้อมูลได้ทั้งหมด เนื่องจากจำนวนเว็บเพจที่มีอยู่มากมายมหาศาลและการเพิ่มปริมาณที่รวดเร็วของเว็บเพจในแต่ละวัน สาเหตุหนึ่งที่ผู้ใช้มักไม่ได้ข้อมูลที่ต้องการก็เพราะระบบค้นหาส่วนใหญ่ เช่น Google[Google], Excite[Excite], AltaVista[AltaVista] ใช้การค้นหาตามคำสำคัญ (keyword) ทำให้เว็บเพจ (Web page) ที่ค้นคืนมาให้กับผู้ใช้มีเพจที่ไม่ตรงกับความต้องการของผู้ใช้จำนวนมาก แม้ว่าเว็บเพจนั้นจะมีคำสำคัญตรงกับที่ผู้ใช้ป้อนให้ก็ตาม Lawrence และ Giles [Lawrence & Giles, 1998] ได้ศึกษาพบว่า ระบบค้นหาเว็บที่มีชื่อเสียงเช่น Hotbot[Hotbot], AltaVista, Excite, infoseek[Infoseek] ค้นหาข้อมูลโดยได้การครอบคลุม (coverage) เพียงแค่ 57.5%, 46.5%, 23.1% และ 16.5% ตามลำดับ นอกจากอัตราการครอบคลุมที่ต่ำแล้ว ความแม่นยำ (precision) ก็น้อยด้วย กล่าวคือจะมีเพจที่ไม่เกี่ยวข้องติดออกมาจำนวนมาก

ตัวอย่างเช่น ผู้ใช้คนหนึ่งต้องการศึกษาว่าเสือ Jaguar มีน้ำหนักโดยเฉลี่ยเท่าไร และใช้คำสำคัญคือ “jaguar” และ “weight” เพื่อค้นหาข้อมูลบนอินเตอร์เน็ตจากระบบค้นหาเว็บ AltaVista ซึ่งมีเป็นระบบค้นหาเว็บที่มีการรวบรวมเว็บเพจไว้ทั้งสิ้นประมาณ 30 ล้านเพจ (สำรวจเมื่อปี 1996 [Ma, et al., 1996]) AltaVista จะให้เว็บเพจที่ค้นคืนมาได้ประมาณ 900 เอกสาร ซึ่งในจำนวนนี้มีเอกสารที่ไม่ตรงกับความต้องการจำนวนมาก เช่น มีข้อมูลของรถยนต์ยี่ห้อ Jaguar มีข้อมูลของทีมฟุตบอลที่ชื่อ Jacksonville Jaguars มีข้อมูลของเครื่องเล่นเกมส์ที่ชื่อ Atari Jaguar และอื่นๆ อีกมากมาย ทำให้ผู้ใช้ต้องสูญเสียเวลาจำนวนมากในการเลือกหาเอกสารที่ต้องการ ระบบค้นหาเว็บเหล่านี้แม้จะมีข้อเสียที่ให้เอกสารที่ไม่ตรงกับความต้องการจำนวนมาก แต่ก็มีข้อดีที่การสร้างระบบทำได้ค่อนข้างง่ายโดยไม่ต้องอาศัยแรงงานคนมากนัก ซึ่งจะใช้หุ่นยนต์เว็บ (Web robot) ที่เป็นโปรแกรมแบบหนึ่งทำหน้าที่รวบรวมเอกสารที่อยู่บนเครือข่ายอย่างอัตโนมัติ

นอกจากระบบค้นหาเว็บแล้ว ผู้ใช้ยังสามารถใช้ระบบไดเร็กทอรีโครงข่าย (net directory system) เพื่อค้นหาข้อมูลบนอินเตอร์เน็ตได้เช่นกัน ตัวอย่างที่มีชื่อเสียงของระบบประเภทนี้คือ Yahoo! ซึ่งสามารถค้นหาข้อมูลที่ตรงกับความต้องการได้มากกว่า เนื่องจากข้อมูลได้ถูกจัดแยกเป็นโครงสร้างตามหมวดหมู่ แต่ก็มีข้อเสียที่ปริมาณเว็บเพจที่ครอบคลุมจะน้อยกว่าระบบค้นหาเว็บมาก เนื่องจากต้องใช้แรงงานคนจำนวนมากในการจัดทำไดเร็กทอรีโครงข่าย (มีประมาณ 9 แสนเว็บเพจในการสำรวจเมื่อปี 1999 [Mladenic & Grobelnik, 1999])

ดังนั้นจะพบว่าปัญหาสำคัญ 2 ปัญหาที่เกิดขึ้นคือ (1) การครอบคลุมต่ำกล่าวคือระบบค้นหาเว็บหรือระบบไดเร็กทอรีโครงข่ายมีจำนวนเอกสารที่ค้นคืนมาได้ไม่ครอบคลุมเพียงพอ และ (2) ความแม่นยำน้อยกล่าวคือระบบค้นหาเว็บค้นคืนเอกสารที่ไม่ตรงกับความต้องการจำนวนมาก

งานวิจัยนี้มุ่งเน้นที่จะศึกษาวิจัยถึงแนวทางในการแก้ปัญหานี้ ซึ่งรวมข้อดีของระบบทั้งสองประเภทด้านบนโดยการจัดหมวดหมู่ของเว็บเพจให้ได้อย่างอัตโนมัติ เทคนิคที่จะนำมาใช้คือการเรียนรู้ของเครื่องและการโปรแกรมตรรกะเชิงอุปนัย และพัฒนาเทคนิคใหม่ของการเรียนรู้ของเครื่องที่สามารถเรียนรู้จากตัวอย่างที่ไม่มีฉลาก (unlabeled example) เนื่องจากพบว่าในโดเมนของเว็บเพจนั้น เรามีตัวอย่างที่ไม่มีฉลาก (ในกรณีนี้คือเว็บเพจ) จำนวนมากที่จะนำมาสอนระบบทำเหมืองเว็บ แต่ตัวอย่างที่มีฉลากนั้น ผู้สอนต้องทำการติดฉลากกับตัวอย่าง เช่น ถ้าต้องการสอนระบบทำเหมืองเว็บให้เรียนรู้ว่า อะไรคือเพจภาษาไทย อะไรคือเพจภาษาอื่น ผู้สอนต้องนำตัวอย่างที่ไม่มีฉลากมาติดฉลากลงไป ซึ่งเสียแรงงานจำนวนมาก และวิธีการเรียนรู้ของเครื่องที่จะ

นำมาใช้กับระบบทำเหมืองเว็บนั้น ส่วนมากต้องการตัวอย่างสอนจำนวนมากจึงจะเรียนรู้ได้อย่างถูกต้องแม่นยำ วิธีวิจัยที่จะทำในงานนี้ จึงมุ่งเน้นที่จะใช้ประโยชน์จากตัวอย่างที่ไม่มีฉลากให้มากที่สุด

งานวิจัยนี้นำเสนอโมเดลของการสอนแบบใหม่ที่เรียกว่า การสอนไขว้แบบวนซ้ำซึ่งใช้ประโยชน์จากตัวแยกแยะ (classifier) 2 ตัวคือ ตัวแยกแยะราคาแพง (expensive classifier) และตัวแยกแยะราคาถูก (cheap classifier) และจากชุดตัวอย่าง 2 ชุดคือ ชุดตัวอย่างที่ 1 และ ชุดตัวอย่างที่ 2 ซึ่งตัวอย่างทั้งสองชุดเป็นแบบไม่มีฉลากทั้งสิ้น ตัวแยกแยะทั้งสองจะมีพารามิเตอร์ที่ต้องเรียนจากตัวอย่าง ซึ่งถ้าเป็นตัวอย่างที่มีฉลากก็สามารถเรียนพารามิเตอร์ได้โดยตรงจากตัวอย่าง แต่ในกรณีที่เราสนใจคือตัวอย่างไม่มีฉลากนั้น ตัวแยกแยะทั้งสองก็น่าจะสามารถเรียนรู้พารามิเตอร์ได้เช่นเดียวกัน โดยอาศัยการโต้ตอบและการสอนระหว่างกันเอง เนื่องจากว่าเราต้องการใช้ประโยชน์จากตัวอย่างที่ไม่มีฉลาก ดังนั้นถ้าตัวแยกแยะไม่มีความรู้เบื้องต้นเกี่ยวกับโดเมนที่จะเรียนรู้เลย การเรียนรู้ก็ไม่สามารถทำได้ เราจึงให้ความรู้เบื้องต้นกับตัวแยกแยะด้วย ตัวอย่างเช่น ถ้าต้องการให้เรียนรู้ว่าเว็บเพจใดเป็นภาษาไทย เพจใดเป็นภาษาอื่น เราก็จะให้ความรู้เบื้องต้นกับตัวแยกแยะราคาแพงในรูปของพจนานุกรมภาษาไทย ตัวเรียนรู้ราคาแพงในที่นี้จึงหมายถึงตัวแยกแยะที่มีความรู้เกี่ยวกับโดเมนอยู่บ้าง และใช้การคำนวณมาก ส่วนตัวแยกแยะราคาถูกไม่จำเป็นต้องมีความรู้เบื้องต้นเกี่ยวกับโดเมนและใช้การคำนวณต่ำ ในงานวิจัยนี้ใช้ตัวแยกแยะแบบเบย์ (Bayes classifier) เป็นตัวแยกแยะราคาถูก ส่วนตัวแยกแยะราคาแพงใช้ตัวแยกแยะที่สามารถใช้ความรู้เบื้องต้นเกี่ยวกับโดเมนได้ เช่น อาจเป็นตัวแยกแยะตัดคำที่มีความรู้เบื้องต้นในรูปของพจนานุกรม หรือ อาจเป็นการโปรแกรมตรรกะเชิงอุปนัย – ไอแอลพี (Inductive Logic Programming – ILP) มีคุณสมบัติที่เหนือกว่าการเรียนรู้ของเครื่องชนิดอื่นที่สามารถรับความรู้เบื้องต้นจากผู้สอนได้ และจากการโต้ตอบและสอนกันเองระหว่างตัวแยกแยะทั้งสอง ซึ่งอาจต้องวนสอนกันเองหลายรอบจนกระทั่งพารามิเตอร์ของทั้งสองลู่เข้า การเรียนรู้ก็จะสิ้นสุด ข้อดีของตัวแยกแยะราคาถูกคือใช้การคำนวณต่ำดังนั้นเมื่อเรียนรู้สำเร็จแล้ว จึงเหมาะที่จะนำไปให้กับหุ่นยนต์เว็บใช้สำหรับงานที่ต้องการได้โดยไม่สิ้นเปลืองเวลาในการคัดเลือกเว็บเพจจากอินเตอร์เน็ต

วัตถุประสงค์

วัตถุประสงค์ของการวิจัยในโครงการนี้ได้แก่

(1) เพื่อทำการวิจัยพื้นฐานเพื่อเพิ่มประสิทธิภาพของการโปรแกรมตรรกะเชิงอุปนัย

(2) เพื่อวิจัยเทคนิคการเรียนรู้ของเครื่องที่สามารถใช้ประโยชน์จากข้อมูลแบบไม่มีฉลาก

(3) เพื่อปรับปรุงประสิทธิภาพของการสอนไขว้แบบวนซ้ำด้วยการโปรแกรมตรรกะเชิงอุปนัย

ระเบียบวิธีวิจัยและผลที่ได้

1. การวิจัยพื้นฐานเพื่อเพิ่มประสิทธิภาพของการโปรแกรมตรรกะเชิงอุปนัย

ในการจำแนกเว็บเพจออกเป็นหมวดหมู่โดยอัตโนมัตินั้น สามารถทำได้หลายวิธีการด้วยกัน วิธีการหนึ่งที่น่าสนใจคือ การโปรแกรมตรรกะเชิงอุปนัยหรือไอแอลพี (Inductive Logic Programming – ILP) เนื่องจากวิธีการนี้ ผู้สอนสามารถป้อนความรู้เบื้องต้นในรูปแบบของโปรแกรมตรรกะอันดับที่หนึ่งได้ ซึ่งจะช่วยให้การจำแนกข้อมูลทำได้อย่างมีประสิทธิภาพยิ่งขึ้น ในหัวข้อนี้จะกล่าวถึงการวิจัยพื้นฐานเพื่อเพิ่มประสิทธิภาพของไอแอลพี

การโปรแกรมตรรกะเชิงอุปนัย [Quinlan, 1990; Muggleton, 1991; Muggleton & De Raedt, 1994] โดยทั่วไปมักถูกนำไปใช้กับปัญหาที่มีลักษณะเป็นสองกลุ่ม (two-class concept) มีจุดหมายเพื่อจำแนกตัวอย่างออกเป็นสองกลุ่ม (class) คือ กลุ่มตัวอย่างบวก และกลุ่มตัวอย่างลบ โดยการสร้างกฎเพื่ออธิบายกลุ่มตัวอย่างบวก ดังนั้นเมื่อนำกฎที่สร้างได้ไปจำแนกตัวอย่างใหม่ ตัวอย่างที่ตรงกับกฎจะถูกจำแนกเป็นกลุ่มตัวอย่างบวก

ส่วนตัวอย่างที่ไม่ตรงกับกฎจะถูกจำแนกเป็นกลุ่มตัวอย่างลบ แต่ในกรณีที่ต้องการนำระบบไอแอลพีไปใช้จำแนกตัวอย่างออกเป็นหลายกลุ่ม (multi-class concept) นั้น อาจมีตัวอย่างบางตัวโดยเฉพาะตัวอย่างที่มีสัญญาณรบกวน (noisy data) ที่ไม่ตรงกับกฎข้อใดข้อหนึ่งในเซตของกฎทั้งหมด ซึ่งในกรณีเช่นนี้ระบบไอแอลพีแต่เพียงอย่างเดียวไม่สามารถจำแนกตัวอย่างในลักษณะนี้ได้ จึงจำเป็นต้องมีวิธีการอื่นเข้ามาช่วยเพื่อทำให้ระบบไอแอลพีสามารถจำแนกตัวอย่างออกเป็นหลายกลุ่มได้ ตัวอย่างของวิธีการเหล่านี้ได้แก่ วิธีการจำแนกตามกลุ่มหลัก (Majority class method) [Clark & Niblett, 1989;Dzeroski, et al., 1996] การใช้วิธีการของเบย์โดยระบบ 1BC [Flach & Lachiche, 1999] การสร้างต้นไม้ตัดสินใจที่สามารถใช้แทนกฎลำดับที่หนึ่ง (First-Order Rule) โดยระบบ TILDE [Blockeel & Raedt, 1997] ฯลฯ

อย่างไรก็ดียังไม่มีวิธีการใดที่สามารถเลือกกฎในกรณีที่ตัวอย่างนั้นไม่ตรงกับกฎข้อใดข้อหนึ่งได้ ดังนั้นในงานวิจัยนี้จึงเป็นการมุ่งศึกษาหาวิธีการ ซึ่งสามารถนำมาใช้จำแนกตัวอย่างเพื่อใช้ร่วมกับกฎที่ได้จากระบบไอแอลพี เราได้ใช้กระบวนการดึงลักษณะสำคัญและวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก (Backpropagation Neural Network — BNN) เพื่อประมาณกฎสำหรับเลือกกลุ่มที่ใกล้เคียงในกรณีดังกล่าว กระบวนการดึงลักษณะสำคัญใช้เพื่อหาลักษณะสำคัญจากกฎลำดับที่หนึ่ง รูปแบบของปัญหาเดิมจะเปลี่ยนไปอยู่ในรูปของค่าคุณลักษณะ (attribute value) ซึ่งเป็นปัญหาที่มีลักษณะเป็นตรรกศาสตร์ประพจน์ (propositional logic) โดยใช้ค่าความจริงจากลักษณะสำคัญจัดเป็นอินพุตเวกเตอร์ (input vector) เพื่อป้อนให้แก่นิวรอล-เน็ตเวิร์กเรียนรู้และทดสอบ นิวรอลเน็ตเวิร์กเป็นวิธีการเรียนรู้ของเครื่องแบบหนึ่งซึ่งใช้กันอย่างแพร่หลาย เนื่องจากนิวรอลเน็ตเวิร์กสามารถเรียนรู้เพื่อปรับค่าน้ำหนักของเส้นเชื่อมภายในโครงสร้าง ซึ่งเป็นการกำหนดความสำคัญให้แก่เส้นเชื่อมต่างๆ ดังนั้นหากนำลักษณะสำคัญมาป้อนเป็นอินพุตให้แก่นิวรอลเน็ตเวิร์ก เมื่อผ่านกระบวนการเรียนรู้แล้ว นิวรอลเน็ตเวิร์กจะสามารถกำหนดได้ว่าลักษณะสำคัญข้อใดมีความสำคัญมากกว่าลักษณะสำคัญข้ออื่น สาเหตุสำคัญอีกประการหนึ่งที่เลือกใช้วิธีแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก คือ นิวรอล-เน็ตเวิร์กสามารถจำแนกตัวอย่างที่มีลักษณะเป็นหลายกลุ่มได้ ดังนั้นด้วยการนำกระบวนการดึงลักษณะสำคัญและแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กมาใช้ร่วมกับกฎจากระบบไอแอลพี จะทำให้เราสามารถนำกฎที่ได้จากระบบไอแอลพีมาจำแนกตัวอย่างในกรณีที่ตัวอย่างนั้นไม่ตรงกับกฎข้อใดข้อหนึ่งพอดีได้ อันเป็นการพัฒนาความสามารถของระบบไอแอลพีให้จัดการกับปัญหาที่มีลักษณะเป็นหลายกลุ่ม หรือใช้จำแนกตัวอย่างซึ่งไม่ตรงกับกฎข้อใดในเซตของกฎพอดีได้

การประมาณกฎจากระบบไอแอลพีด้วยวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก แบ่งเป็น 3 ขั้นตอนหลัก คือ (1) การดึงลักษณะสำคัญ (Feature Extraction) (2) การสร้างโครงสร้างนิวรอลเน็ตเวิร์กจากกฎและลักษณะสำคัญ และ (3) การสอนนิวรอลเน็ตเวิร์ก ซึ่งมีวิธีการดังจะกล่าวต่อไปนี้

1.1 การดึงลักษณะสำคัญ

ปัญหาสำคัญของกฎลำดับที่หนึ่ง คือในกรณีปัญหาที่มีลักษณะเป็นหลายกลุ่ม เมื่อจำแนกตัวอย่างทดสอบจะพบว่าจะมีตัวอย่างบางตัวอย่างที่ไม่ตรงกับกฎข้อใดเลย ระบบไอแอลพีเพียงลำพังจะไม่สามารถบอกได้ว่าตัวอย่างนั้นควรจะถูกจำแนกเป็นกลุ่มใด ในการวิจัยครั้งนี้มีแนวคิดว่าเมื่อไม่มีกฎข้อใดที่ตรงกับตัวอย่างพอดี เราสามารถใช้การครอบคลุมเพียงบางส่วน (partially cover) จากกฎแต่ละข้อเมื่อนำกฎไปเทียบกับตัวอย่าง เพื่อจำแนกตัวอย่างออกเป็นกลุ่มได้ การครอบคลุมเพียงบางส่วนอาจครอบคลุมคุณสมบัติบางอย่างที่สำคัญของกฎในแต่ละข้อ ซึ่งนำไปสู่การจำแนกตัวอย่างนั้นๆ ได้ ในกรณีที่ไม่สามารถหากฎที่ตรงกับตัวอย่างพอดี กฎที่ควรจะตรงกับตัวอย่างที่สุดควรเป็นกฎที่มีลักษณะสำคัญตรงกับตัวอย่างนั้นมากกว่ากฎข้ออื่นๆ โดยส่วนที่ไม่ตรงกับกฎควรเป็นลักษณะที่ไม่สำคัญ ในการทดลองครั้งนี้ได้ใช้แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กมาทำการเรียนรู้ เพื่อกำหนดน้ำหนักซึ่งเป็นการให้ความสำคัญกับลักษณะสำคัญแต่ละข้อของกฎ

ตัวอย่างเช่น กฎลำดับที่หนึ่งซึ่งทุกสัญพจน์ (literal) ภายในกฎข้อนั้นมีแต่ตัวแปรซึ่งอยู่ในส่วนหัวของกฎ เราสามารถดึงลักษณะสำคัญ โดยใช้แต่ละสัญพจน์ในกฎข้อนั้นเป็นลักษณะสำคัญของกฎได้ ตัวอย่างเช่น

```
mesh(A,1) :- not_important(A), not_loaded(A).
```

จากตัวอย่างซึ่งเป็นกฎลำดับที่หนึ่งข้างต้นจะเห็นว่า สัญพจน์แต่ละตัวมีแต่ตัวแปรซึ่งอยู่ในส่วนหัวทั้งสิ้น และไม่มีตัวแปรใหม่ปรากฏอยู่ในสัญพจน์นั้นเลย เราจึงสามารถใช้สัญพจน์แต่ละตัวเป็นลักษณะสำคัญได้ และเรียกสัญพจน์ที่มีลักษณะแบบนี้ว่า ลักษณะสำคัญเดี่ยว (singleton feature)

แต่โดยปกติแล้วกฎในลำดับที่หนึ่งที่ได้จากระบบไอแอลพีจะมีความซับซ้อนกว่าตัวอย่างข้างต้นมาก ดังนั้นจึงต้องหาวิธีการที่สามารถดึงลักษณะสำคัญที่ประกอบด้วยตัวแปรใหม่ในแต่ละสัญพจน์ให้ได้ ในระบบไอแอลพีโดยทั่วไปสัญพจน์ที่มีตัวแปรใหม่จะไม่สามารถอยู่โดดเดี่ยวโดยปราศจากสัญพจน์อื่นซึ่งสร้างตัวแปรนี้ขึ้นมา และสัญพจน์ส่วนใหญ่ที่สร้างตัวแปรใหม่ จะถูกรวมเข้าอยู่ในกฎลำดับที่หนึ่งเพื่อทำหน้าที่ส่งผ่านตัวแปรใหม่นั้นไปยังสัญพจน์อื่นซึ่งทำหน้าที่ตรวจสอบคุณสมบัติเฉพาะของตัวอย่างนั้นๆ ดังนั้นในงานวิจัยนี้จึงได้นำลักษณะการเกิดตัวแปรใหม่ในสัญพจน์มาสร้างเป็นลำดับของสัญพจน์ เพื่อสร้างเป็นลักษณะสำคัญจากกฎลำดับที่หนึ่ง

เราได้คำนิยามสายสัญพจน์ปิด (closed chain) และสายสัญพจน์เปิด (open chain) เพื่อใช้ในกระบวนการดึงลักษณะสำคัญดังนี้

นิยามที่ 1 (สายสัญพจน์ปิด) *สัญพจน์ใดๆ จะเป็นส่วนหนึ่งของสายสัญพจน์ปิดถ้าตัวแปรใหม่ทุกตัวในสัญพจน์นั้นซึ่งไม่อยู่ในส่วนหัวของกฎปรากฏในสัญพจน์อย่างน้อยสองสัญพจน์ และปรากฏในสัญพจน์อย่างน้อยหนึ่งสัญพจน์ร่วมกับตัวแปรในส่วนหัวของกฎหรือตัวแปรซึ่งเคยมีอยู่แล้วในสายสัญพจน์นั้นๆ* □

กล่าวอีกนัยหนึ่งคือ สายสัญพจน์ปิดจะเริ่มจากตัวแปรซึ่งอยู่ในส่วนหัวของกฎ แล้วตัวแปรเหล่านี้จะไปสร้างตัวแปรใหม่ในสัญพจน์ซึ่งอยู่ในกฎ ตัวแปรใหม่ที่ถูกสร้างขึ้นไปสร้างตัวแปรใหม่ขึ้นอีก เป็นลำดับอย่างนี้ไปจนสิ้นสุดที่สัญพจน์ซึ่งไม่มีการสร้างตัวแปรใหม่อีก สัญพจน์ที่ไม่มีการสร้างตัวแปรใหม่นี้โดยส่วนใหญ่จะเป็นสัญพจน์ที่บอกลักษณะพิเศษของตัวอย่างนั้น ตัวอย่างเช่น จากกฎ

```
mesh(A,1) :- short(A), neighbour_yz_l(B,A), usual(B).
```

จะเห็นว่า ในส่วนหัวของกฎมีตัวแปร `A` ปรากฏอยู่ สัญพจน์ `short(A)` ซึ่งมีแต่ตัวแปรที่อยู่ในส่วนหัวเท่านั้นจะถูกจัดเป็นลักษณะสำคัญที่เป็นลักษณะสำคัญเดี่ยว นอกจากลักษณะสำคัญนี้แล้ว ลักษณะสำคัญอีกอันหนึ่งที่ถูกสร้างขึ้นในลักษณะของสายสัญพจน์ปิด คือ

```
neighbour_yz_l(B,A), usual(B).
```

ซึ่งเป็นลักษณะสำคัญที่เริ่มจากตัวแปร `A` ในส่วนหัวของกฎ จากนั้นที่สัญพจน์ `neighbour_yz_l(B,A)` ตัวแปร `A` สร้างตัวแปรใหม่ `B` ดังนั้นสัญพจน์ `neighbour_yz_l(B,A)` จะถูกรวมเข้าไว้ในสายสัญพจน์ จากนั้นตัวแปร `B` ซึ่งถูกสร้างขึ้นในสัญพจน์ `neighbour_yz_l(B,A)` ปรากฏอีกครั้งในสัญพจน์ `usual(B)` และในสัญพจน์ `usual(B)` นี้มีแต่ตัวแปรที่เคยเกิดขึ้นแล้ว และไม่มีการสร้างตัวแปรใหม่ ทำให้สายสัญพจน์นี้สิ้นสุดที่สัญพจน์ `usual(B)` และเป็นสายสัญพจน์ปิด จะเห็นได้ว่า จากตัวอย่างของสายสัญพจน์ปิดข้างต้น สัญพจน์ `neighbour_yz_l(B,A)` ปรากฏอยู่ในกฎข้อนี้เพื่อทำหน้าที่เพียงสร้างตัวแปร `B` ขึ้นจากตัวแปร `A` และส่งต่อตัวแปร `B` ไปยังสัญพจน์อื่น ในขณะที่สัญพจน์ `usual(B)` เป็นสัญพจน์ที่ใช้ระบุลักษณะเฉพาะของตัวอย่าง สัญพจน์ `usual(B)` จะปรากฏอยู่ในลักษณะสำคัญโดยไม่มีสัญพจน์ `neighbour_yz_l(B,A)` ร่วมอยู่ไม่ได้ ดังนั้นจึงต้องรวมทั้งสองสัญพจน์ไว้ในสายสัญพจน์ปิดด้วยกัน

นิยามที่ 2 (สายสัญพจน์เปิด) *สัญพจน์ใดๆ จะเป็นส่วนหนึ่งของสายสัญพจน์เปิด ถ้ามีตัวแปรใหม่ในสัญพจน์นั้นซึ่งไม่อยู่ในส่วนหัวของกฎ ปรากฏในสัญพจน์เพียงสัญพจน์เดียวเท่านั้นร่วมกับตัวแปรในส่วนหัวของกฎหรือตัวแปรซึ่งเคยมีอยู่แล้วในสายสัญพจน์นั้นๆ* □

ลักษณะการเกิดสายสัญพจน์เปิดในกฎลำดับที่หนึ่งจะคล้ายกับการเกิดสัญพจน์ปิด มีการนำลักษณะการเกิดตัวแปรใหม่มาใช้เช่นเดียวกันกับสายสัญพจน์ปิด โดยสัญพจน์ที่มีตัวแปรใหม่จะถูกรวมเข้าไว้ในสายสัญพจน์เช่นกัน เพียงแต่ในสายสัญพจน์เปิดจะมีตัวแปรใหม่เกิดขึ้นและตัวแปรใหม่นี้ไม่ปรากฏในสัญพจน์อื่น นั่นคือตัวแปรใหม่ที่เกิดขึ้นนี้ไม่ได้ทำหน้าที่สร้างตัวแปรอื่น หรือกำหนดลักษณะเฉพาะของตัวอย่าง ต่างจากในสายสัญพจน์ปิดซึ่งตัวแปรใหม่ทุกตัวที่เกิดขึ้นจะต้องทำหน้าที่อย่างใดอย่างหนึ่ง คือ สร้างตัวแปรอื่นหรือกำหนดลักษณะเฉพาะของตัวอย่าง ตัวอย่างของสายสัญพจน์เปิด เช่น

จากกฎ

```
mesh(A,2) :- short(A), neighbour_xy_l(B,A), neighbour_zx_l(C,B).
```

ลักษณะสำคัญหนึ่งที่ถูกสร้างขึ้นในลักษณะของสายสัญพจน์เปิด คือ

```
neighbour_xy_l(B,A), neighbour_zx_l(C,B).
```

จากตัวอย่างจะเห็นว่า ตัวแปร `A` ซึ่งอยู่ภายในส่วนหัวของกฎ ทำหน้าที่สร้างตัวแปรใหม่ `B` ขึ้นภายในสัญพจน์ `neighbour_xy_l(B,A)` และตัวแปร `B` ที่เกิดขึ้นใหม่ ทำหน้าที่สร้างตัวแปรใหม่ `C` ที่สัญพจน์ `neighbour_zx_l(C,B)` จะเห็นได้ว่าสัญพจน์ `neighbour_xy_l(B,A)` ทำหน้าที่ส่งต่อตัวแปร `B` เพื่อไปสร้างตัวแปรใหม่ `C` ยังสัญพจน์ถัดไป ดังนั้นสัญพจน์ `neighbour_xy_l(B,A)` จึงถูกรวมในสายสัญพจน์เปิด ส่วนในสัญพจน์ `neighbour_zx_l(C,B)` ตัวแปร `C` ที่ถูกสร้างขึ้นไม่ปรากฏอยู่ในสัญพจน์อื่นอีกเลย ดังนั้นสายสัญพจน์นี้จึงเป็นสายสัญพจน์เปิด

<u>ตัวอย่างสายสัญพจน์ปิดและสายสัญพจน์เปิด</u>

จากกฎ

```
p(A,B) :- q1(A), q2(A,C), q3(C), q4(C,D), q5(D), q6(A,E,F), q7(E,G),
          q8(E,H), q9(E), q10(F,I), q11(I,B).
```

จะได้สายสัญพจน์ปิดดังต่อไปนี้

```
1) q2(A,C),q3(C)
2) q2(A,C),q4(C,D),q5(D)
3) q2(A,C),q3(C),q4(C,D),q5(D)
4) q6(A,E,F),q9(E),q10(F,I),q11(I,B)
```

และจะได้สายสัญพจน์เปิดดังต่อไปนี้

```
1) q6(A,E,F),q7(E,G)
2) q6(A,E,F),q8(E,H)
3) q6(A,E,F),q7(E,G),q8(E,H)
4) q6(A,E,F),q7(E,G),q9(E)
5) q6(A,E,F),q8(E,H),q9(E)
6) q6(A,E,F),q7(E,G),q8(E,H),q9(E)
```

จากตัวอย่างจะเห็นว่า สัญพจน์บางตัวไม่สามารถปรากฏเพียงลำพังในสายสัญพจน์ได้ กล่าวคือ สัญพจน์ที่มีตัวแปรใหม่จะต้องปรากฏร่วมกับสัญพจน์อื่นที่มีตัวแปรซึ่งเคยปรากฏมาแล้ว หรือเป็นตัวแปรในส่วนหัวของกฎ หรือกล่าวอีกนัยหนึ่งว่า สัญพจน์ที่มีตัวแปรใหม่จะต้องปรากฏร่วมกับสัญพจน์ที่ทำหน้าที่สร้างตัวแปรนั้นซึ่งอาจเป็นสัญพจน์ในส่วนหัวของกฎหรือเป็นสัญพจน์ในตัวกฎก็ได้ เพราะถ้าสัญพจน์ซึ่งมีตัวแปรใหม่ปรากฏโดยไม่

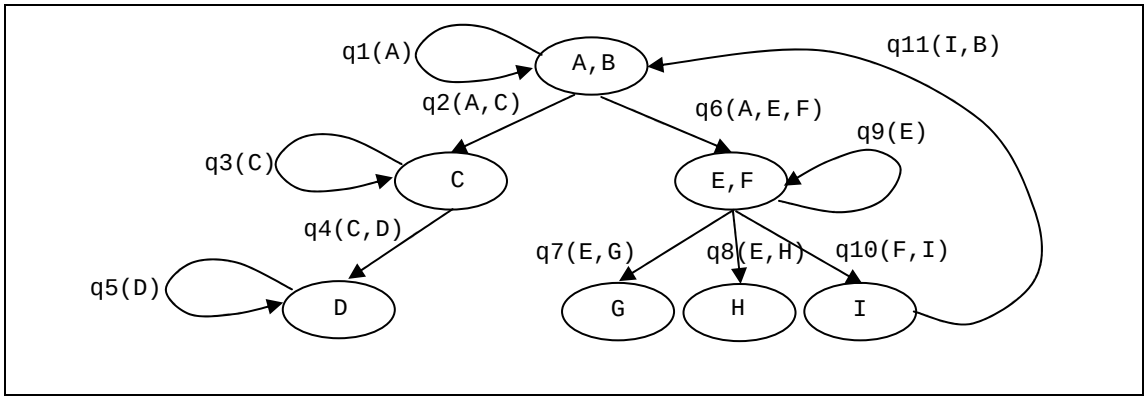
มีสัญพจน์ซึ่งทำหน้าที่สร้างตัวแปรนั้น จะทำให้ไม่สามารถหาการแทนที่ตัวแปรนั้นได้ ตัวอย่างของสายสัญพจน์เหล่านี้ เช่น “q3(C)”, “q1(A), q3(C)”, “q5(D)”, “q1(A), q5(D)” ฯลฯ ตามคำนิยามของทั้งสายสัญพจน์ปิดและสายสัญพจน์เปิดจะจำกัดไม่ให้มีสายสัญพจน์เหล่านี้เกิดขึ้นอยู่แล้ว เนื่องจากจากคำนิยามของสายสัญพจน์ทั้งสองแบบ สัญพจน์ใดๆ จะถูกเข้ารวมเข้าไว้ในสายสัญพจน์ ถ้าในสัญพจน์นั้นมีตัวแปรใหม่ปรากฏร่วมกับตัวแปรที่มีอยู่แล้วในส่วนหัวของกฎหรือเคยปรากฏร่วมกับตัวแปรอื่นในสายสัญพจน์นั้นๆ

1.1.1 อัลกอริทึมการดึงลักษณะสำคัญ

จากคำนิยามของสายสัญพจน์ปิดและสายสัญพจน์เปิดจะเห็นได้ว่า สายสัญพจน์ทั้งสองมีแนวคิดมาจากการเกิดตัวแปรใหม่ภายในสัญพจน์ ดังนั้นเพื่อให้สามารถเข้าใจวิธีการดึงลักษณะสำคัญได้ง่าย ในงานวิจัยนี้จึงแทนกฎลำดับที่หนึ่งด้วยกราฟ โดยให้โนด (node) ของกราฟแทนตัวแปรใหม่ โนดเริ่มต้น (initial node) แทนตัวแปรซึ่งอยู่ในส่วนหัวของกฎ และเส้นเชื่อม (edge) แทนสัญพจน์ซึ่งมีตัวแปรนั้นๆ พิจารณากฎลำดับที่หนึ่งดังตัวอย่างต่อไปนี้

```
p(A,B) :- q1(A), q2(A,C), q3(C), q4(C,D), q5(D), q6(A,E,F), q7(E,G),
          q8(E,H), q9(E), q10(F,I), q11(I,B).
```

สัญพจน์ส่วนหัวคือ `p(A,B)` ดังนั้นโนดซึ่งแทนตัวแปร `A` และ `B` จะถูกจัดเป็นโนดเริ่มต้นของกราฟ สัญพจน์ `q1(A)` ไม่ได้ทำหน้าที่สร้างตัวแปรใหม่ มีเพียงตัวแปร `A` ปรากฏอยู่ภายในสัญพจน์ จึงมีเส้นเชื่อมซึ่งแทนสัญพจน์ `q1(A)` ออกและวนกลับเข้าที่โนดนั้น จากนั้นที่สัญพจน์ `q2(A,C)` ตัวแปร `A` ทำหน้าที่สร้างตัวแปรใหม่ `C` ดังนั้นจึงมีเส้นเชื่อมซึ่งแทนสัญพจน์ `q2(A,C)` ออกจากโนดเริ่มต้นไปยังโนดใหม่ซึ่งแทนตัวแปร `C` จากกฎข้างต้น เราสามารถแทนด้วยกราฟได้ดังรูปที่ 1



รูปที่ 1 ตัวอย่างของกราฟซึ่งใช้แทนกฎลำดับที่หนึ่ง

อัลกอริทึมการดึงลักษณะสำคัญมีขั้นตอนดังต่อไปนี้

(1) หาเส้นเชื่อมทุกเส้นซึ่งเริ่มต้นและสิ้นสุดที่โนดเริ่มต้น ใช้สัญพจน์เหล่านั้นเป็นลักษณะสำคัญเดี่ยว

(2) หาสายสัญพจน์ปิดทุกสายที่เป็นไปได้ซึ่งเริ่มต้นจากโนดเริ่มต้น สายสัญพจน์ที่ได้ในขั้นตอนนี้เป็นสายสัญพจน์ปิด

(3) หาทางเดินที่เป็นไปได้ทุกทางซึ่งเริ่มต้นจากโนดเริ่มต้นแล้วไปสิ้นสุดที่โนดสิ้นสุด (terminal node) ซึ่งเป็นโนดที่ไม่มีเส้นเชื่อมออกจากตัวเอง สายสัญพจน์ที่ได้ในขั้นตอนนี้เป็นสายสัญพจน์เปิด

(4) หาการรวม (combination) ทุกแบบที่เป็นไปได้ของสายสัญพจน์เปิดซึ่งได้จากขั้นที่สาม ที่มีตัวแปรใหม่ร่วมกัน

1.1.2 ตัวอย่างกระบวนการดึงลักษณะสำคัญ

พิจารณากฎด้านล่างนี้

```
p(A,B) :- q1(A), q2(A,C), q3(C), q4(C,D), q5(D), q6(A,E,F), q7(E,G),
          q8(E,H), q9(E), q10(F,I), q11(I,B).
```

จากกฎข้างต้นสามารถดึงลักษณะสำคัญได้ดังต่อไปนี้

ขั้นที่ 1

1) q1(A)

ขั้นที่ 2

2) q2(A,C),q3(C)

3) q2(A,C),q4(C,D),q5(D)

4) q2(A,C),q3(C),q4(C,D),q5(D)

5) q6(A,E,F),q9(E),q10(F,I),q11(I,B)

ขั้นที่ 3

6) q6(A,E,F),q7(E,G)

7) q6(A,E,F),q8(E,H)

8) q6(A,E,F),q9(E),q7(E,G)

9) q6(A,E,F),q9(E),q8(E,H)

ขั้นที่ 4

10) q6(A,E,F),q7(E,G),q8(E,H)

11) q6(A,E,F),q7(E,G),q8(E,H),q9(E)

จากตัวอย่างจะเห็นว่า อัลกอริทึมไม่ได้สร้างสายสัญพจน์เปิดทุกสายสัญพจน์ที่เป็นไปได้ โดยสายสัญพจน์เปิดที่ไม่ได้ถูกสร้างขึ้น ได้แก่ สายสัญพจน์เปิดที่เป็นส่วนหนึ่งของสายสัญพจน์ปิด เช่น “q2(A,C)”, “q2(A,C),q4(C,D)” และ “q6(A,E,F),q9(E),q10(F,I)” ฯลฯ เนื่องจาก โดยปกติแล้วการสร้างตัวแปรใหม่ในกฎลำดับที่หนึ่ง มีจุดมุ่งหมายเพื่อใช้ตัวแปรใหม่นั้นในสัญพจน์อื่นซึ่งทำหน้าที่ตรวจสอบลักษณะเฉพาะของตัวอย่างนั้นๆ หรือทำหน้าที่สร้างตัวแปรใหม่ตัวอื่น ดังนั้นสายสัญพจน์ทุกสายสัญพจน์ที่สร้างขึ้นจึงควรสิ้นสุดที่สัญพจน์ซึ่งไม่มีตัวแปรใหม่เกิดขึ้น แต่ในบางกรณีที่สร้างสายสัญพจน์ปิดไม่ได้จึงจำเป็นต้องสร้างสายสัญพจน์เปิดด้วย

จากลักษณะสำคัญที่ได้ เราสามารถเปลี่ยนรูปแบบของกฎเพื่อนำไปใช้สร้างนิวรอลเน็ตเวิร์กโดยใช้ลักษณะสำคัญเดี่ยว สายสัญพจน์ปิด และสายสัญพจน์เปิด ประกอบเข้าไปในกฎเพื่อทำให้สามารถหาค่าความจริงของแต่ละลักษณะสำคัญได้โดยไม่ทำให้ความหมายเดิมของกฎเปลี่ยนไป ตัวอย่างเช่น จากตัวอย่างของกระบวนการดึงลักษณะสำคัญข้างต้น กฎที่ผ่านการเปลี่ยนรูปแบบแล้วจะเป็นดังรูปที่ 2

```
p(A,B) :- q1(A),
          q2(A,C),q3(C),
          q2(A,C),q4(C,D),q5(D),
          q2(A,C),q3(C),q4(C,D),q5(D),
          q6(A,E,F),q9(E),q10(F,I),q11(I,B),
          q6(A,E,F),q7(E,G),
          q6(A,E,F),q8(E,H),
          q6(A,E,F),q9(E),q7(E,G),
          q6(A,E,F),q9(E),q8(E,H),
          q6(A,E,F),q7(E,G),q8(E,H),
          q6(A,E,F),q7(E,G),q8(E,H),q9(E).
```

รูปที่ 2 ตัวอย่างของลักษณะสำคัญที่อยู่ภายในกฎ

1.2 การสร้างโครงสร้างนิวรอลเน็ตเวิร์กจากกฎและลักษณะสำคัญ

เมื่อผ่านกระบวนการดึงลักษณะสำคัญแล้ว ลักษณะสำคัญที่ได้จะอยู่ในรูปของลักษณะสำคัญเดี่ยว สายสัญพจน์ปิด และสายสัญพจน์เปิดดังรูปที่ 2 จากนั้นเราจะสร้างนิวรอลเน็ตเวิร์กโดยกำหนดให้ประกอบด้วย 3 ชั้น คือ ชั้นอินพุต (input layer) ชั้นซ่อน (hidden layer) และชั้นเอ๊าท์พุต (output layer) นิวรอนในชั้นอินพุตแทนลักษณะสำคัญของกฎแต่ละข้อ ดังนั้นจำนวนนิวรอนที่มีในชั้นอินพุตจะเท่ากับจำนวนลักษณะสำคัญที่มีอยู่ในเซตของกฎ นิวรอนในชั้นซ่อนแทนกฎแต่ละข้อ จำนวนนิวรอนในชั้นนี้จึงมีค่าเท่ากับจำนวนกฎ นิวรอนในชั้นอินพุตซึ่งแทนลักษณะสำคัญภายในกฎแต่ละข้อจะมีเส้นเชื่อมมายังนิวรอนในชั้นซ่อนซึ่งแทนกฎข้อนั้นๆ นิวรอนในชั้นเอ๊าท์พุตแทนกลุ่มของตัวอย่าง การเชื่อมต่อจากชั้นซ่อนมายังชั้นเอ๊าท์พุตเป็นการเชื่อมต่อแบบทั้งหมด (fully connected) จำนวนนิวรอนในชั้นเอ๊าท์พุตขึ้นกับจำนวนกลุ่มของตัวอย่าง ในกรณีของปัญหาแบบหลายกลุ่ม จำนวนนิวรอนในชั้นเอ๊าท์พุตจะมีค่าเท่ากับจำนวนกลุ่ม ส่วนในกรณีของปัญหาแบบสองกลุ่ม จำนวนนิวรอนจะมีค่าเท่ากับสอง นิวรอนตัวหนึ่งจะแทนกลุ่มที่เป็นตัวอย่างบวก ในขณะที่นิวรอนอีกตัวหนึ่งแทนกลุ่มที่เป็นตัวอย่างลบ

ตัวอย่างโครงสร้างของนิวรอลเน็ตเวิร์ก

สมมติว่าในเซตของกฎประกอบด้วยกฎ 4 ข้อ คือ {*C1*, *C2*, *C3*, *C4*} และตัวอย่างแบ่งออกเป็น 3 กลุ่ม คือ {1, 2, 3} กฎข้อ *C1* เป็นกฎสำหรับรู้จำตัวอย่างกลุ่ม 1 กฎข้อ *C2* และ *C3* เป็นกฎสำหรับรู้จำตัวอย่างกลุ่ม 2 และ กฎข้อ *C4* เป็นกฎสำหรับรู้จำตัวอย่างกลุ่ม 3 ดังด้านล่างนี้

```
C1: mesh(A,1) :- not_important(A), not_loaded(A).
C2: mesh(A,2) :- short(A), opposite_l(B,A).
C3: mesh(A,2) :- usual(A), neighbour_yz_r(A,B), cont_loaded(B).
C4: mesh(A,3) :- short(A), neighbour_zx_r(A,B), opposite_r(A,C),
                 short(C).
```

เมื่อผ่านกระบวนการดึงลักษณะสำคัญแล้วจะได้ลักษณะสำคัญดังต่อไปนี้

F1C1: `not_important(A)`
F2C1: `not_loaded(A)`

F1C2: `short(A)`
F2C2: `opposite_l(B,A)`

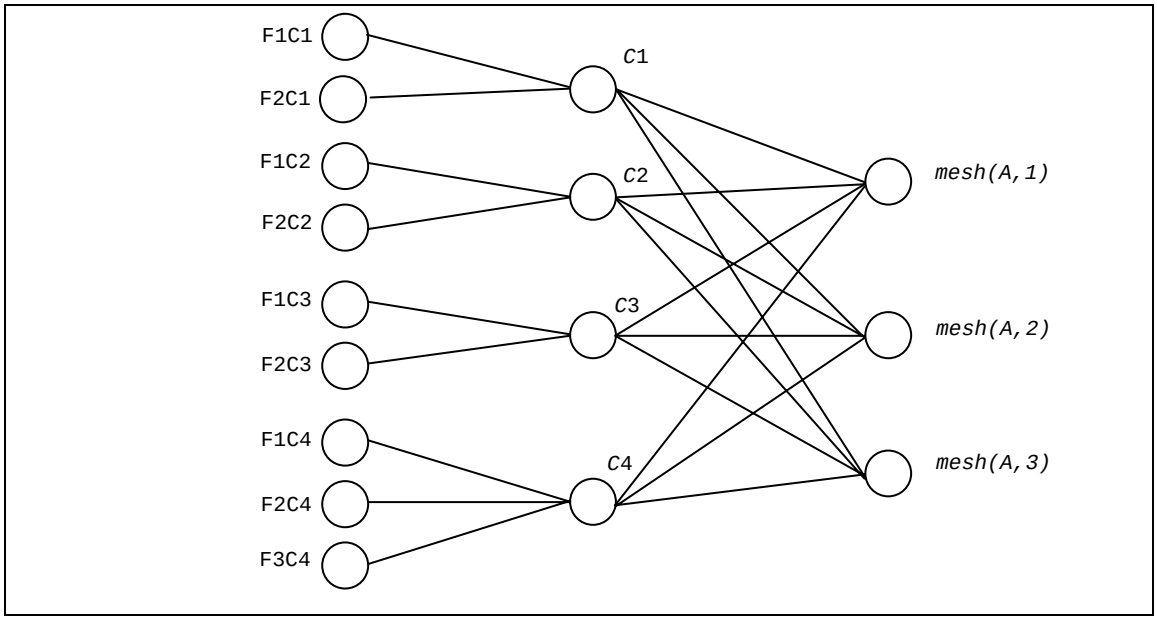
F1C3: `usual(A)`
F2C3: `neighbour_yz_r(A,B), cont_loaded(B)`

F1C4: `short(A)`
F2C4: `opposite_r(A,C), short(C)`
F3C4: `neighbour_zx_r(A,B)`

โดยที่ *FiCj* แทนลักษณะสำคัญที่ *i* ในกฎข้อที่ *j*

ในตัวอย่างด้านบนนี้ ลักษณะสำคัญ *F1C1*, *F2C1*, *F1C2*, *F1C3* และ *F1C4* เป็นลักษณะสำคัญเดี่ยว *F2C3* และ *F2C4* เป็นสายสัญพจน์ปิด ในขณะที่ *F2C2*, *F3C4* เป็นสายสัญพจน์เปิด ลักษณะสำคัญเดี่ยวเป็นการตรวจสอบคุณสมบัติของตัวอย่างโดยใช้เพียงสัญพจน์เดียว เช่น ในลักษณะสำคัญ *F2C1* "`not_loaded(A)`" เป็นการตรวจสอบว่า ตัวอย่าง `A` มีลักษณะเป็น `not_loaded(A)` หรือไม่ ตัวอย่างของสายสัญพจน์ปิด เช่น ลักษณะสำคัญ *F2C4* "`opposite_r(A,C), short(C)`" เป็นการตรวจสอบคุณสมบัติของตัวอย่าง `A` ว่ามีคุณสมบัติตรงตามสัญพจน์ทั้งสองหรือไม่ คือ `A` และ `C` มีความสัมพันธ์ `opposite_r(A,C)` กัน และ `C` ยังมีคุณสมบัติเป็น `short(C)` อีกด้วย จะเห็นในสายสัญพจน์นี้ได้ว่า ตัวแปร `C` ถูกสร้างขึ้นจากตัวแปร `A` ในสัญพจน์ `opposite_r(A,C)` ทำให้สัญพจน์ `short(C)` ไม่สามารถใช้เป็นลักษณะสำคัญโดยไม่มีสัญพจน์ที่สร้างมันขึ้นมา คือ `opposite_r(A,C)` ประกอบอยู่ด้วย ส่วนตัวอย่างของสายสัญพจน์เปิดในกรณีนี้คือ ลักษณะสำคัญ *F2C2* ซึ่งจำเป็นต้องใช้เป็นลักษณะสำคัญข้อหนึ่งด้วยเนื่องจาก `opposite_l(B,A)` เป็นการตรวจสอบคุณสมบัติว่าตัวอย่าง `A` มีคุณสมบัติ `opposite_l(B,A)` หรือไม่

ตัวอย่างนี้เป็นการจำแนกตัวอย่างออกเป็น 3 กลุ่ม ประกอบด้วย `mesh(A,1)`, `mesh(A,2)` และ `mesh(A,3)` ตามลำดับ ดังนั้นโครงสร้างของนิวรอลเน็ตเวิร์กจะเป็นตามรูปที่ 3 นั่นคือ เน็ตเวิร์กประกอบด้วย 3 ชั้น ชั้นอินพุต ชั้นซ่อน และชั้นเอ๊าท์พุต ชั้นอินพุตประกอบด้วย 9 นิวรอน แต่ละนิวรอนแทนลักษณะสำคัญแต่ละข้อ (*F1C1*, *F2C1*, *F1C2*, *F2C2*, *F1C3*, *F2C3*, *F1C4*, *F2C4* และ *F3C4*) ชั้นซ่อนประกอบด้วย 4 นิวรอน แต่ละนิวรอนแทนกฎแต่ละข้อ (*C1*, *C2*, *C3* และ *C4*) ลักษณะสำคัญที่มาจากกฎข้อเดียวกันจะถูกเชื่อมต่อไปยังนิวรอนซึ่งแทนกฎข้อนั้นๆ เช่น นิวรอนซึ่งแทนลักษณะสำคัญ *F1C1* และ *F2C1* จะถูกเชื่อมต่อไปยังนิวรอนซึ่งแทนกฎ *C1* นิวรอนซึ่งแทนลักษณะสำคัญ *F1C2* และ *F2C2* จะถูกเชื่อมต่อไปยังนิวรอนซึ่งแทนกฎ *C2* เป็นต้น ส่วนที่ชั้นเอ๊าท์พุต ประกอบด้วย 3 นิวรอน แต่ละนิวรอนแทนกลุ่มแต่ละกลุ่ม นิวรอนจากชั้นซ่อนและชั้นเอ๊าท์พุตถูกเชื่อมต่อแบบทั้งหมด



รูปที่ 3 ตัวอย่างโครงสร้างของนิวรอลเน็ตเวิร์ก

1.3 การสอนนิวรอลเน็ตเวิร์ก

เมื่อกำหนดโครงสร้างของนิวรอลเน็ตเวิร์กได้แล้ว ขั้นตอนต่อไปคือ การสอนนิวรอลเน็ตเวิร์ก อินพุตเวกเตอร์ (input vector) ที่จะป้อนให้กับนิวรอลเน็ตเวิร์กได้มาจากการนำตัวอย่างที่ใช้สอนมาเทียบกับลักษณะสำคัญ เพื่อหาค่าความจริงของลักษณะสำคัญแต่ละข้อ โดยทั่วไปแล้วค่าความจริงของลักษณะสำคัญสามารถมีได้หลายแบบขึ้นกับการแทนที่ตัวแปรด้วยค่าคงที่ ในการวิจัยครั้งนี้ได้เลือกการแทนที่ที่ทำให้มีจำนวนลักษณะสำคัญที่มีค่าความจริงเป็นจริงมากที่สุด โดยกำหนดให้ค่าของอินพุตนิวรอนสำหรับลักษณะสำคัญที่มีค่าความจริงเป็นจริงเป็น 1 และลักษณะสำคัญที่มีค่าความจริงเป็นเท็จมีค่าเป็น –1 สำหรับส่วนของเอ้าท์พุตเวกเตอร์กำหนดให้เอ้าท์พุตนิวรอนที่แทนกลุ่มของตัวอย่างนั้นมีค่าเท่ากับ 1 นอกจากนั้นให้มีค่าเท่ากับ 0

ตัวอย่างการสร้างอินพุตเวกเตอร์และเอ้าท์พุตเวกเตอร์

จากตัวอย่างโครงสร้างของนิวรอลเน็ตเวิร์กข้างต้น กำหนดให้กลุ่มความรู้ภูมิหลังที่ใช้สร้างกฎเป็นดังนี้

```
not_important(a1).              not_loaded(a2).
short(a1).                      not_loaded(a3).
short(a2).                      cont_loaded(a2).
neighbour_yz_r(a1,a2).          cont_loaded(a3).
neighbour_zx_r(a1,a2).          opposite_r(a1,a2).
usual(a2).                      opposite_r(a1,a3).
```

เมื่อผ่านกระบวนการดึงลักษณะสำคัญในตัวอย่างข้างต้นแล้ว จะสามารถแปลงกฎแต่ละข้อเป็นดังนี้

```
C1:     mesh(A,1) :-
F1C1:           (not_important(A)),
F2C1:           (not_loaded(A)).

C2:     mesh(A,2) :-
F1C2:           (short(A)),
F2C2:           (opposite_l(B,A)).
```

```
C3:          mesh(A,2) :-
F1C3:        (usual(A)),
F2C3:        (neighbour_yz_r(A,B), cont_loaded(B)).

C4:      mesh(A,3) :-
F1C4:        (short(A)),
F2C4:        (neighbour_zx_r(A,B)),
F3C4:        (opposite_r(A,C), short(C)).
```

สมมติว่าใช้ตัวอย่าง mesh(a1,3) เป็นตัวอย่างที่ใช้เรียนรู้ เมื่อนำไปเทียบกับลักษณะสำคัญแต่ละข้อจะเป็นดังนี้

```
C1:      mesh(a1,1) :-
F1C1:        (not_important(a1)),                          TRUE
F2C1:        (not_loaded(a1)).                             FALSE

C2:      mesh(a1,2) :-
F1C2:        (short(a1)),                                  TRUE
F2C2:        (opposite_l(a2,a1)).                          FALSE

C3:      mesh(a1,2) :-
F1C3:        (usual(a1)),                                  FALSE
F2C3:        (neighbour_yz_r(a1,a2), cont_loaded(a2)).     TRUE

C4:      mesh(a1,3) :-
F1C4:        (short(a1)),                                  TRUE
F2C4:        (neighbour_zx_r(a1,a2)),                      TRUE
F3C4:        (opposite_r(a1,a2), short(a2)).               TRUE
```

เมื่อนำไปจัดเป็นอินพุตเวกเตอร์ให้กับนิวรอลเน็ตเวิร์กเพื่อทำการเรียนรู้ จะได้อินพุตเวกเตอร์เป็น <1, -1, 1, -1, -1, 1, 1, 1, 1> ส่วนของเอ๊าท์พุตเวกเตอร์จะเป็น <0, 0, 1> เนื่องจากตัวอย่างนี้เป็นตัวอย่างของกลุ่ม 3 ซึ่งแทนด้วยเอ๊าท์พุตนิวรอนตัวสุดท้าย ดังนั้นเอ๊าท์พุตนิวรอนตัวสุดท้ายจึงมีค่าเป็น 1 นอกจากนั้นมีค่าเป็น 0 จะสังเกตจากตัวอย่างได้ว่า ในส่วนของลักษณะสำคัญจากกฎข้อ *C4* จะสามารถใช้การแทนที่ได้อีกแบบหนึ่งซึ่งให้ค่าความจริงไม่เหมือนกัน คือ

```
C4:      mesh(a1,3) :-
F1C4:        (short(a1)),                                  TRUE
F2C4:        (neighbour_zx_r(a1,a2)),                      TRUE
F3C4:        (opposite_r(a1,a3), short(a3)).               FALSE
```

ลักษณะสำคัญ *F3C4* สามารถใช้ค่าคงที่ a3 แทนตัวแปร C ได้ในสัญพจน์ opposite_r(A,C) ซึ่งทำให้ตัวแปร C ในสัญพจน์ short(C) ต้องถูกแทนที่ด้วยค่าคงที่ a3 ด้วยเช่นกัน ทำให้ลักษณะสำคัญ *F3C4* ที่ถูกแทนที่ด้วยค่าคงที่แล้วเปลี่ยนเป็น (opposite_r(a1,a3), short(a3)) ซึ่งให้ค่าความจริงเป็นเท็จ เมื่อเทียบกับการแทนที่โดยใช้ค่าคงที่ a2 แทนตัวแปร C ในตัวอย่างก่อนหน้านี้ที่ให้ค่าความจริงของลักษณะสำคัญในกฎข้อนี้เป็นจริงทั้งหมด ดังนั้นในตัวอย่างนี้จึงเลือกใช้การแทนที่ตัวแปร C ด้วยค่าคงที่ a2

ตัวอย่างที่ใช้ในกระบวนการเรียนรู้เพื่อสร้างกฎจากระบบไอแอลพี จะถูกนำมาเทียบกับลักษณะสำคัญเพื่อสร้างอินพุตเวกเตอร์และเอ๊าท์พุตเวกเตอร์ดังตัวอย่างข้างต้น แต่ในระบบไอแอลพีโดยทั่วไปกฎที่ถูกสร้างขึ้นจะไม่ครอบคลุมตัวอย่างที่ใช้เรียนรู้ทุกตัว จะมีตัวอย่างบางตัวที่ไม่ตรงพอดีกับกฎข้อใดเลย ดังนั้นในการเลือกตัวอย่างที่ใช้เรียนรู้สำหรับนิวรอลเน็ตเวิร์ก จะมีเงื่อนไขดังต่อไปนี้

1) สำหรับปัญหาที่มีลักษณะเป็นหลายกลุ่ม จะเลือกเฉพาะตัวอย่างที่ถูกครอบคลุมด้วยกฎ หรือตัวอย่างที่ตรงพอดีกับกฎมาเป็นตัวอย่างสำหรับนิวรอลเน็ตเวิร์ก

2) ในปัญหาที่มีลักษณะเป็นสองกลุ่ม ตัวอย่างที่ใช้สำหรับสร้างนิวรอลเน็ตเวิร์กจะเป็นตัวอย่างบวกที่ตรงพอดีกับกฎและกลุ่มตัวอย่างลบ

กระบวนการเรียนรู้ของนิวรอลเน็ตเวิร์กจะเริ่มตั้งแต่ทำการสุ่มน้ำหนัก (weight) ของเส้นเชื่อม และค่าไบแอส (bias) ของโนด จากนั้นทำการเรียนรู้โดยทำการปรับน้ำหนักแบบเพิ่มขึ้น (incremental) โดยปรับน้ำหนักเมื่อป้อนตัวอย่างทีละตัว ด้วยขั้นตอนวิธีแบ็กพรอพาเกชัน [Rumelhart, et al., 1986] และทดสอบการลู่เข้า โดยนับจำนวนตัวอย่างที่ถูกติดกันทั้งหมดให้มากกว่าค่าที่กำหนดไว้ในตอนเริ่มทำการทดลอง นิวรอล-เน็ตเวิร์กที่สร้างได้จะถูกนำไปใช้รู้จำตัวอย่างต่อไป

1.4 ผลการทดลอง

ในส่วนนี้กล่าวถึงระบบไอแอลพีที่นำมาทดสอบ ชุดข้อมูลที่ใช้ในการทดลอง ผลการทดลองเปรียบเทียบระหว่างวิธีการประมาณกฎด้วยวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก (Backpropagation Artificial Neural Network for Approximating Rules: BANNAR) กับวิธีการอื่นๆ ดังต่อไปนี้

1.4.1 ระบบไอแอลพีที่ใช้ในการทดลอง

เราได้ทดลองเปรียบเทียบระบบ BANNAR ซึ่งเราพัฒนาขึ้นกับระบบไอแอลพีอื่นๆ คือ PROGOL, GOLEM, TILDE, 1BC, LINUS และ FOSSIL ซึ่งรายละเอียดของแต่ละระบบมีดังนี้

1. PROGOL

ระบบ PROGOL [Muggleton, 1995] เป็นระบบไอแอลพีที่มีประสิทธิภาพระบบหนึ่งที่ใช้กันอย่างแพร่หลาย รับอินพุตเป็นเซตของตัวอย่างบวก ตัวอย่างลบ และกลุ่มความรู้ภูมิหลัง ผู้ใช้จะต้องประกาศลักษณะการทำงานของสัญพจน์ (mode declaration) เพื่อให้ระบบรู้ว่าแต่ละอาร์กิวเมนต์ของสัญพจน์จะมีลักษณะเป็นอย่างไร โดยอาร์กิวเมนต์ของสัญพจน์สามารถเป็นได้ทั้งตัวแปรอินพุต ตัวแปรเอ๊าท์พุต และค่าคงที่ นอกจากหน้าที่ของอาร์กิวเมนต์ต่างๆ แล้ว การประกาศลักษณะการทำงานของสัญพจน์นี้ยังสามารถระบุได้ถึงจำนวนครั้งที่ต้องการให้สัญพจน์นี้ปรากฏในกฎที่สร้างได้จากระบบอีกด้วย การสร้างกฎของระบบ PROGOL จะเริ่มตั้งแต่สุ่มตัวอย่างสร้างอนุประโยคที่เฉพาะเจาะจงที่สุด (most-specific) สำหรับตัวอย่างนั้น แล้วทำการค้นหาด้วยวิธีการแบบ A* (A*-like search) [Nilsson, 1980] โดยมีจุดมุ่งหมายเพื่อให้ได้การบีบอัด (compression) สูงสุดในการค้นหานั้น ทำให้ระบบ PROGOL ใช้เวลาและเนื้อที่หน่วยความจำมากในการสร้างกฎเมื่อเทียบกับระบบอื่นๆ ที่ใช้ในการทดลองครั้งนี้

เนื่องจากระบบ PROGOL เป็นระบบไอแอลพีที่ทำงานกับปัญหาที่มีลักษณะเป็นสองกลุ่ม คือ กลุ่มตัวอย่างบวก และกลุ่มตัวอย่างลบ ดังนั้นในการสร้างกฎสำหรับปัญหาที่มีลักษณะเป็นหลายกลุ่ม จึงสร้างกฎโดยสร้างครั้งละกลุ่ม กำหนดให้ตัวอย่างบวกในการเรียนรู้คือตัวอย่างในกลุ่มนั้น และตัวอย่างลบคือ ตัวอย่างของกลุ่มอื่นๆ ทำซ้ำจนครบทุกกลุ่ม จะได้กฎสำหรับทุกๆ กลุ่ม

2. GOLEM

ระบบ GOLEM [Muggleton & Feng, 1990] เป็นระบบไอแอลพีที่ทำงานกับปัญหาที่มีลักษณะเป็นสองกลุ่มเช่นเดียวกับระบบ PROGOL แต่แตกต่างกันในวิธีการค้นหาอนุประโยค โดยระบบ GOLEM รับอินพุตเป็นตัวอย่างบวก ตัวอย่างลบ และกลุ่มความรู้ภูมิหลัง ระบบเริ่มสร้างอนุประโยคด้วยการสุ่มเลือกตัวอย่างขึ้นมาเป็นคู่ หาอาร์แอลจีจีของตัวอย่างคู่นั้น เลือกผลการทำอาร์แอลจีจีที่ครอบคลุมตัวอย่างบวกมากที่สุด จากนั้นจึงวางนัยทั่วไปโดยนำอนุประโยคที่ได้มาทำอาร์แอลจีจีกับตัวอย่างบวกตัวใหม่ซึ่งสุ่มเลือกมาจากเซตของตัวอย่างบวกจนกระทั่งไม่สามารถหาอนุประโยคที่ครอบคลุมตัวอย่างได้มากขึ้น

ระบบ GOLEM อนุญาตให้ผู้ใช้ระบุจำนวนคู่ของตัวอย่างที่จะสุ่มมาทำอาร์แอลจีจีเพื่อหาอนุประโยคที่ครอบคลุมตัวอย่างมากที่สุดได้ ซึ่งจำนวนคู่ที่ระบบกำหนดไว้คือ 8 คู่ ถ้ากำหนดให้มีค่ามากขึ้น โอกาสที่ระบบจะพบอนุประโยคที่ครอบคลุมตัวอย่างได้มากจะสูงขึ้นตามไปด้วย ซึ่งในการทดลองครั้งนี้ได้ใช้ค่าที่ระบบกำหนดไว้ให้ คือ 8 คู่ในการสร้างกฎ และเนื่องจากระบบ GOLEM เป็นระบบที่ถูกสร้างขึ้นมาให้ใช้กับปัญหาที่เป็นสองกลุ่ม ดังนั้นในการสร้างกฎสำหรับกลุ่มใดๆ จะใช้ตัวอย่างของกลุ่มนั้นเป็นตัวอย่างบวกและใช้ตัวอย่างของกลุ่มอื่นๆ ที่เหลือเป็นตัวอย่างลบ เช่นเดียวกับขั้นตอนการสร้างกฎของระบบ PROGOL

3. TILDE

ระบบ TILDE [Blockeel & Raedt, 1997] จะทำการเรียนรู้เพื่อสร้างต้นไม้ตัดสินใจ (decision tree) ที่สามารถแทนกฎลำดับที่หนึ่งได้ หลักการทำงานของระบบจะคล้ายกับอัลกอริทึมการสร้างต้นไม้ตัดสินใจแบบทั่วไปที่ใช้กับค่าตรรกศาสตร์ประพจน์ โดยเริ่มจากสร้างโนดขึ้นมาทดสอบเพื่อเปรียบเทียบค่าฮิวริสติก (heuristic) ของแต่ละโนด ระบบจะเลือกโนดทดสอบที่มีค่าฮิวริสติกดีที่สุดในขณะนั้นมาสร้างเป็นโนดจริงในต้นไม้ จากนั้นจะนำตัวอย่างทั้งหมดมาทำการทดสอบกับโนดนี้เพื่อแบ่งตัวอย่างออกเป็นสองกลุ่ม คือ กลุ่มที่ให้ค่าความจริงเป็นจริงและให้ค่าความจริงเป็นเท็จเมื่อเทียบกับโนดปัจจุบัน จากนั้นจึงสร้างโนดทดสอบขึ้นมาใหม่สำหรับทดสอบกับตัวอย่างทั้งสองกลุ่มแล้วทำการทดสอบไปเรื่อยๆ จนกระทั่งมีตัวอย่างเพียงกลุ่มเดียวเท่านั้นที่ตรงกับโนดนั้น ระบบ TILDE นี้จะต่างจากระบบ PROGOL และ GOLEM ตรงที่ระบบ TILDE สามารถรู้จำตัวอย่างที่มีลักษณะเป็นหลายกลุ่มได้ เนื่องจากที่โนดใบ (leaf node) ของต้นไม้ตัดสินใจที่สร้างขึ้นจะแทนตัวอย่างแต่ละกลุ่ม เมื่อตัวอย่างไปตกอยู่ที่กลุ่มใด ก็จะถูกจำแนกเป็นกลุ่มนั้น

4. 1BC

1BC [Flach & Lachiche, 1999] เป็นตัวแยกแยะแบบเบย์ (Bayesian Classifier) สามารถใช้กับตรรกะลำดับที่หนึ่งได้ ระบบนี้ใช้แนวคิดจากวิธีการของเบย์กับตรรกศาสตร์ประพจน์ ซึ่งใช้ค่าทางสถิติของตัวอย่างมาจำแนกตัวอย่างใหม่ เมื่อนำแนวคิดของเบย์มาใช้กับตรรกะลำดับที่หนึ่งจึงต้องมีการปรับเปลี่ยนวิธีการรู้จำ ระบบ 1BC แบ่งลักษณะของสัญพจน์ออกเป็นสองกลุ่มด้วยกันคือ กลุ่มที่เป็นสัญพจน์โครงสร้าง และกลุ่มที่เป็นสัญพจน์ที่บอกลักษณะ ซึ่งสัญพจน์ทั้งสองกลุ่มนี้ผู้ใช้ต้องเป็นผู้กำหนดให้ในตอนเริ่มเรียนรู้ จากนั้นระบบจะสร้างลักษณะสำคัญจากสัญพจน์ทั้งสองกลุ่ม แล้วจึงเรียนรู้ด้วยอัลกอริทึมเบย์อย่างง่าย (naive Bayes algorithm) นอกจากลักษณะของสัญพจน์ที่ผู้ใช้เป็นผู้กำหนดแล้ว ระบบ 1BC ยังอนุญาตให้ผู้ใช้กำหนดลักษณะของลักษณะสำคัญที่สร้างขึ้นได้ โดยระบุจำนวนสัญพจน์และจำนวนตัวแปรที่สามารถปรากฏได้ในลักษณะสำคัญข้อหนึ่งๆ ซึ่งในการทดลองครั้งนี้ใช้ค่าปกติของระบบคืออนุญาตให้มีจำนวนสัญพจน์และจำนวนตัวแปรเท่ากับ 3 ทั้งสองค่า แต่ในการทดลองกับปัญหาการวิเคราะห์ไฟไนต์เอลิเมนต์เกิดปัญหาหน่วยความจำไม่พอ เนื่องจากมีสัญพจน์ที่มีค่าทั้งสองเป็น 3 อยู่เป็นจำนวนมาก จึงกำหนดให้ค่าทั้งสองมีค่าเป็น 2

5. LINUS

ระบบ LINUS [Lavrac & Dzeroski, 1994] เป็นระบบไอแอลพีที่สามารถใช้กับปัญหาที่มีลักษณะเป็นหลายกลุ่มและมีเป็นตรรกะลำดับที่หนึ่งได้ โดยระบบจะเปลี่ยนรูปแบบของปัญหาจากตรรกะลำดับที่หนึ่งให้เป็นค่าคุณสมบัติ (attribute value) การทำงานของระบบจะเริ่มโดยการสร้างลักษณะสำคัญจากตัวอย่างและความรู้ภูมิหลังซึ่งอยู่ในรูปตรรกะลำดับที่หนึ่ง ใช้ลักษณะสำคัญเหล่านั้นเป็นค่าคุณสมบัติของตัวอย่างแต่ละตัว จากนั้นจึงใช้ระบบที่สามารถรู้จำตัวอย่างจากค่าคุณสมบัติได้ เช่น ASSISTANT, CN2, NEWGEM และ C4.5 ฯลฯ ในการทดลองเปรียบเทียบครั้งนี้ได้สร้างลักษณะสำคัญโดยระบบ LINUS แล้วนำลักษณะสำคัญซึ่งอยู่ในรูปค่าคุณสมบัติไปทำการรู้จำด้วยระบบ C4.5 [Quinlan, 1993] เนื่องจากระบบ C4.5 สามารถจำแนกตัวอย่างซึ่งมีลักษณะเป็นหลายกลุ่มได้

1.4.2 ชุดข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูลหรือกลุ่มของปัญหาที่ใช้ในการทดลองครั้งนี้มี 4 ชุด คือ การรู้จำภาพตัวพิมพ์อักษรไทย การวิเคราะห์ไฟไนต์เอลิเมนต์ การวิเคราะห์ความสามารถก่อกลายพันธุ์ และการวิเคราะห์ตำแหน่งตัวหมากรุก รายละเอียดของกลุ่มปัญหาทั้งหมดเป็นดังนี้

1. การรู้จำภาพตัวพิมพ์อักษรไทย (Thai Optical Character Recognition: TCR)

กลุ่มตัวอย่างในข้อมูลชุดนี้ประกอบด้วย พยัญชนะ สระ วรรณยุกต์ และตัวเลขไทย รวมทั้งหมดเป็นตัวพิมพ์อักษรไทย 77 ตัวอักษร ใช้ตัวอักษร 2 แบบ 7 ขนาด รวมเป็นชุดตัวอย่างที่ใช้เรียนรู้ทั้งสิ้น 1,078 ตัว ภาพตัวอย่างถูกสแกนด้วยความละเอียด 300 จุดต่อนิ้ว (dpi) จากนั้นนำมาผ่านกระบวนการกำจัดสัญญาณรบกวน กระบวนการทำภาพให้บางเพื่อหาเวกเตอร์พื้นฐานซึ่งประกอบกันเป็นภาพตัวอักษรนั้น นำเวกเตอร์พื้นฐาน บริเวณที่เป็นจุดร่วมของเส้น บริเวณที่เป็นรอยหยักขึ้นและรอยหยักลง จัดเป็นตัวอย่างที่ใช้เรียนรู้ กลุ่มความรู้ภูมิหลังที่ใช้ในการทดลองประกอบด้วยกฎซึ่งอยู่ในรูปกฎลำดับที่หนึ่งทั้งหมด 55 ข้อ กฎแต่ละข้อเป็นลักษณะของเส้นที่ใช้แบ่งแยกตัวอักษร เช่น `headzone(A,B)` คือ ตัวอักษรที่มีเวกเตอร์พื้นฐาน A มีส่วนหัวอยู่บริเวณ B หรือ `headprim(A,B)` หมายถึง ส่วนหัวของภาพตัวอักษรที่มีเวกเตอร์พื้นฐาน A มีเวกเตอร์พื้นฐาน B เป็นส่วนหัวของตัวอักษร ฯลฯ ซึ่งรายละเอียดของตัวอย่างที่ใช้เรียนรู้และความรู้ภูมิหลังปรากฏอยู่ใน [Kijsirikul & Sinthupinyo, 1999] กลุ่มตัวอย่างที่ใช้ทดสอบเป็นภาพตัวอักษรกลุ่มเดียวกับตัวอย่างที่ใช้เรียนรู้ ซึ่งได้รับการเพิ่มสัญญาณรบกวนโดยนำไปถ่ายเอกสารแบบเข้มและแบบจาง จากนั้นจึงนำไปผ่านกระบวนการเช่นเดียวกับตัวอย่างที่ใช้เรียนรู้ นำเวกเตอร์พื้นฐาน บริเวณที่เป็นจุดร่วมของเส้น บริเวณที่เป็นรอยหยักขึ้นและรอยหยักลง มาจัดเป็นตัวอย่างเพื่อใช้ทดสอบ

2. การวิเคราะห์ไฟไนต์เอลิเมนต์ (Finite Element Mesh Design: FEM)

จุดมุ่งหมายของปัญหา FEM [Dolsak & Muggleton, 1992] คือ การสร้างกฎเพื่อวิเคราะห์ไฟไนต์เอลิเมนต์ในโครงสร้าง โดยอาศัยกลุ่มความรู้ภูมิหลังเป็นลักษณะต่างๆ ของโครงสร้าง ประกอบด้วยลักษณะของเส้นเชื่อม เช่น `long`, `short` และ `usual` ฯลฯ เงื่อนไขขอบเขต เช่น `free` และ `one_side_fixed` ฯลฯ โลด เช่น `cont_loaded` และ `one_sideloaded` ฯลฯ กลุ่มตัวอย่างประกอบด้วยโครงสร้าง 5 แบบ แบ่งตัวอย่างออกเป็น 13 กลุ่ม แต่ละกลุ่มคือ จำนวนองค์ประกอบ (element) ที่เหมาะสมของโครงสร้างนั้น โดยตัวอย่างแต่ละตัวอย่างถูกจัดอยู่ในรูปแบบ `mesh(Edge, Number)` เมื่อ `Edge` คือ ชื่อโครงสร้าง และ `Number` คือ จำนวนองค์ประกอบภายในโครงสร้างนั้น รวมจำนวนตัวอย่างทั้งหมด 278 ตัวอย่าง

3. การวิเคราะห์ความสามารถก่อกลายพันธุ์ (Mutagenesis: MUTA)

จุดมุ่งหมายของปัญหา MUTA [Srinivasan, et al.,1996] คือ การสร้างกฎเพื่อวิเคราะห์ความสามารถก่อกลายพันธุ์ของโมเลกุล โดยอาศัยกลุ่มความรู้ภูมิหลังเป็นลักษณะต่างๆ ภายในโมเลกุลนั้น ประกอบด้วยลักษณะของอะตอมและลักษณะโครงสร้างของโมเลกุล เช่น `benzene`, `carbon_6_ring`, `carbon_5_aromatic_ring`, `hetero_aromatic_6_ring`, `hetero_aromatic_5_ring`, `ring_size_6`, `ring_size_5`, `nitro`, `methyl`, `anthracene`, `phenanthrene` และ `ball3` ฯลฯ กลุ่มตัวอย่างประกอบด้วยตัวอย่างที่เป็นข้อมูลของโมเลกุลทั้งหมด 188 โมเลกุล แบ่งเป็น 125 โมเลกุลอยู่ในกลุ่มตัวอย่างบวก และ 63 โมเลกุลอยู่ในกลุ่มตัวอย่างลบ

4. การวิเคราะห์ตำแหน่งตัวหมากรุกสากล (King-Rook-King Chess Endgame: KRK)

ในข้อมูลชุด KRK [Muggleton, et al., 1989] เป็นการวิเคราะห์ตำแหน่งของตัวหมากบนกระดานหมากรุกสากลที่ไม่สามารถเกิดขึ้นได้ในขณะที่ฝ่ายสีขาวเป็นผู้เดินและมีหมากเหลืออยู่บนกระดาน 3 ตัว คือ ขุนฝ่ายขาว เรือฝ่ายขาว และ ขุนฝ่ายดำ ตัวอย่างที่ใช้อยู่ในรูป `illegal(WKf, WKr, WRf, WRr, BKf, BKr)`

เมื่อ `WKf`, `WKr`, `WRf`, `WRr`, `BKf` และ `BKr` คือ แถว (file) ของขุนฝ่ายขาว หลัก (rank) ของขุนฝ่ายขาว แถวของเรือฝ่ายขาว หลักของเรือฝ่ายขาว แถวของขุนฝ่ายดำ และหลักของขุนฝ่ายดำ กลุ่มความรู้ภูมิหลังคือความสัมพันธ์ของตำแหน่งบนกระดาน ในชุดข้อมูลนี้ใช้ความสัมพันธ์ 2 แบบ คือ `adj(X,Y)` และ `lt(X,Y)` ซึ่งแสดงถึงตำแหน่ง `X` และ `Y` ที่อยู่ติดกัน และตำแหน่งที่มีค่าน้อยกว่ากัน ตามลำดับ จำนวนตัวอย่างในข้อมูลชุดนี้ประกอบด้วยตัวอย่างทั้งหมด 10,000 ตัวอย่าง แบ่งเป็นตัวอย่างบวก 3,361 ตัว และตัวอย่างลบ 6,639 ตัว

1.4.3 ผลการทดลองที่ได้

การทดลองกับชุดข้อมูล FEM, MUTA และ KRK ใช้การทดลองโดยแบ่งตัวอย่างออกเป็น 3 เซตย่อยแบบสุ่ม แล้วทำการทดลองโดยใช้เซตย่อยหนึ่งเซตเป็นเซตตัวอย่างที่ใช้ทดสอบและใช้เซตตัวอย่างที่เหลืออีกสองเซตเป็นเซตตัวอย่างที่ใช้เรียนรู้ จากนั้นวนทำซ้ำโดยให้เซตตัวอย่างทุกเซตเป็นตัวอย่างทดสอบหนึ่งครั้ง แล้วหาค่าเฉลี่ยของผลที่ได้ (3-fold cross-validation: 3CV) สำหรับชุดข้อมูล TCR ใช้การทดสอบโดยสร้างเซตตัวอย่างที่ใช้ทดสอบจากเซตตัวอย่างที่ใช้เรียนรู้ แต่เพิ่มสัญญาณรบกวนโดยผ่านการถ่ายเอกสารแบบเข้มและแบบจาง การทดลองเริ่มด้วยสร้างกฎจากตัวอย่างที่ใช้เรียนรู้ด้วยระบบ PROGOL หรือ GOLEM ตามเซตตัวอย่างที่ใช้เรียนรู้ โดยในชุดข้อมูล FEM และ KRK ใช้ระบบ GOLEM ในการสร้างกฎ ส่วนในชุดข้อมูล TCR และ MUTA ใช้ระบบ PROGOL ในการสร้างกฎ จากนั้นนำกฎที่ได้มาหาลักษณะสำคัญเพื่อนำไปเทียบกับตัวอย่างสำหรับเรียนรู้และทดสอบ แล้วสร้างเป็นอินพุตเวกเตอร์และเอ๊าท์พุตเวกเตอร์ นำอินพุตเวกเตอร์และเอ๊าท์พุตเวกเตอร์มาเรียนรู้และทดสอบ ผลการทดลองเปรียบเทียบระหว่างระบบ BANNAR กับระบบอื่นเป็นดังตารางที่ 1

ตารางที่ 1 ผลการทดลองเปรียบเทียบระหว่าง BANNAR กับระบบไอแอลพีอื่นๆ

ชุดข้อมูล	จำนวนตัวอย่างที่ใช้เรียนรู้	จำนวนตัวอย่างที่ใช้ทดสอบ	จำนวนกลุ่ม	BANNAR	PROGOL หรือ GOLEM	TILDE	1BC	LINUS
TCR	1,074	2,143	77	94.40	72.00^{3}	88.57^{3}	77.23^{3}	66.54^{3}
FEM	278	3CV	13	64.45	57.80^{2}	58.02^{2}	46.73^{2}	60.45^{1}
MUTA	188	3CV	2	83.58	82.01^{0}	68.94^{1}	77.72^{0}	74.41^{1}
KRK	10,000	3CV	2	99.93	99.89^{0}	69.80^{3}	87.12^{3}	99.89^{0}

หมายเหตุ: ใช้การทดสอบค่าทีแบบทางเดียว (one-tailed paired t-test) ตัวเลขยกแสดงระดับความเชื่อมั่น (confidence level) ตัวเลข 1, 2 และ 3 แสดงถึงระดับความเชื่อมั่นที่มากกว่า 90.0%, 99.0% และ 99.5% ตามลำดับ ส่วนตัวเลข 0 แสดงถึงระดับความเชื่อมั่นที่ต่ำกว่า 90%

เนื่องจากระบบ PROGOL และ GOLEM เป็นระบบไอแอลพีที่ใช้กับปัญหาที่มีลักษณะเป็นสองกลุ่ม ดังนั้นเมื่อนำไปใช้กับปัญหาที่มีลักษณะเป็นหลายกลุ่ม และตัวอย่างที่ใช้ทดสอบไม่ตรงกับกฎข้อใดในเซตของกฎ PROGOL และ GOLEM ไม่สามารถทำการจำแนกตัวอย่างนั้นได้ จึงต้องอาศัยวิธีการอื่นมาใช้จำแนกตัวอย่างเหล่านั้น ในการทดลองครั้งนี้ได้เลือกใช้วิธีการจำแนกตามกลุ่มหลักมาใช้ในกรณีดังกล่าว ผลการทดลองที่ปรากฏในตารางที่ 1 ในส่วนของ PROGOL และ GOLEM เป็นผลการทดลองที่ใช้วิธีการจำแนกตามกลุ่มหลักมาใช้ โดยในกรณีของปัญหาที่เป็นสองกลุ่ม คือ MUTA และ KRK ตัวอย่างที่ไม่ตรงกับกฎข้อใดเลย จะถูกจำแนกเป็นกลุ่มลบ ตัวอย่างที่ตรงกับกฎตั้งแต่หนึ่งข้อขึ้นไป จะถูกจำแนกว่าเป็นตัวอย่างบวก แต่ในกรณีปัญหาที่เป็นหลายกลุ่ม จะเลือกกลุ่มโดยใช้วิธีจำแนกตามกลุ่มหลัก กล่าวคือ เมื่อตัวอย่างที่ใช้ทดสอบไม่ตรงกับกฎข้อใดในเซตของกลุ่ม จะทำการจำแนกเป็นกลุ่มที่มีจำนวนมากที่สุดในกลุ่มตัวอย่างที่ใช้เรียนรู้ ในส่วนของระบบ LINUS

ที่นำมาใช้ทดลองเปรียบเทียบในการทดลองครั้งนี้ ใช้ระบบ LINUS สร้างลักษณะสำคัญของชุดตัวอย่าง จากนั้นนำลักษณะสำคัญที่ได้ซึ่งอยู่ในรูปตรรกศาสตร์ประพจน์ไปเรียนรู้และทดสอบด้วยระบบ C4.5

ผลการทดลองในตารางที่ 1 แสดงให้เห็นว่าเปอร์เซ็นต์ความถูกต้องของระบบ PROGOL หรือ GOLEM เมื่อนำไปใช้กับปัญหาที่มีลักษณะเป็นสองกลุ่ม คือ MUTA และ KRK ให้ผลการรู้จำสูงกว่า TILDE ในขณะที่เมื่อนำไปใช้กับปัญหาที่มีลักษณะเป็นหลายกลุ่ม คือ TCR และ FEM TILDE ซึ่งโดยปกติจะใช้งานกับปัญหาที่มีลักษณะเป็นหลายกลุ่มอยู่แล้ว ให้ผลการรู้จำสูงกว่า PROGOL หรือ GOLEM เมื่อเทียบผลการรู้จำของ PROGOL หรือ GOLEM กับ 1BC พบว่า PROGOL หรือ GOLEM ให้ผลการรู้จำสูงกว่า 1BC ในชุดข้อมูล FEM, MUTA และ KRK ให้ผลการรู้จำต่ำกว่า 1BC ในชุดข้อมูล TCR เทียบผลการรู้จำของ PROGOL หรือ GOLEM กับระบบ LINUS ปรากฏว่า PROGOL หรือ GOLEM ให้ผลการรู้จำสูงกว่า LINUS ในชุดข้อมูล TCR และ MUTA ให้ผลการรู้จำเท่ากันในชุดข้อมูล KRK และสูงกว่าในชุดข้อมูล TCR

จากผลการทดลองเมื่อเทียบเปอร์เซ็นต์ความถูกต้องของ PROGOL หรือ GOLEM กับระบบอื่น 3 ระบบ (TILDE, 1BC และ LINUS) ใน 4 ชุดข้อมูล รวมเปรียบเทียบ 12 การทดลอง พบว่า ในกลุ่มปัญหาที่มีลักษณะเป็นหลายกลุ่ม (TCR และ FEM) PROGOL หรือ GOLEM ให้ผลการรู้จำสูงกว่าระบบอื่น 2 การทดลองจาก 6 การทดลอง ต่ำกว่า 4 การทดลองจาก 6 การทดลอง และเมื่อเปรียบเทียบในกลุ่มปัญหาที่มีลักษณะเป็นสองกลุ่ม (MUTA และ KRK) PROGOL หรือ GOLEM ให้ผลการรู้จำสูงกว่าระบบอื่น 5 การทดลองจาก 6 การทดลอง และให้ผลการรู้จำเท่ากัน 1 การทดลองจาก 6 การทดลอง จะเห็นได้ว่า กฎที่ได้จากระบบ PROGOL หรือ GOLEM จะสามารถใช้ได้ดีในกลุ่มปัญหาที่มีลักษณะเป็นสองกลุ่ม และใช้ได้ไม่ดีในกลุ่มปัญหาที่มีลักษณะเป็นหลายกลุ่ม เมื่อนำวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กมาใช้ในระบบ BANNAR ทำให้เปอร์เซ็นต์ความถูกต้องสูงขึ้น และเมื่อเทียบเปอร์เซ็นต์ความถูกต้องของ BANNAR กับระบบอื่น พบว่า BANNAR ให้เปอร์เซ็นต์ความถูกต้องสูงกว่าระบบอื่นทุกระบบในทุกชุดข้อมูล และให้เปอร์เซ็นต์ความถูกต้องสูงกว่าแบบมีนัยสำคัญทางสถิติเมื่อเทียบกับระบบ PROGOL หรือ GOLEM ใน 2 ชุดข้อมูล (TCR และ FEM) ให้เปอร์เซ็นต์ความถูกต้องสูงกว่าแบบมีนัยสำคัญทางสถิติเมื่อเทียบกับ TILDE ในทุกชุดข้อมูล ให้เปอร์เซ็นต์ความถูกต้องสูงกว่าแบบมีนัยสำคัญทางสถิติเมื่อเทียบกับ 1BC ใน 3 ชุดข้อมูล (TCR, FEM และ KRK) และให้เปอร์เซ็นต์ความถูกต้องสูงกว่าแบบมีนัยสำคัญทางสถิติเมื่อเทียบกับ LINUS ใน 3 ชุดข้อมูล (TCR, FEM และ MUTA)

1.5 สรุปผลการวิจัยพื้นฐานเพื่อเพิ่มประสิทธิภาพของการโปรแกรมตรรกะเชิงอุปนัย

ผลการวิจัยที่ได้แสดงให้เห็นว่าการนำวิธีการดึงลักษณะสำคัญมาใช้ร่วมกับแบ็กพรอพาเกชันนิวรอล-เน็ตเวิร์กทำให้ประสิทธิภาพของการโปรแกรมตรรกะเชิงอุปนัยดีขึ้น สามารถจำแนกตัวอย่างที่ไม่ตรงพอดีกับกฎดั้งเดิมได้เป็นอย่างดี และผลการทดลองที่ได้แสดงให้เห็นถึงเปอร์เซ็นต์ความถูกต้องที่เพิ่มขึ้นของระบบ BANNAR เมื่อเทียบกับกฎเดิมซึ่งได้จากระบบ PROGOL หรือ GOLEM และเมื่อเทียบกับระบบอื่นอีก 4 ระบบ ใน 4 ชุดข้อมูล รวมเปรียบเทียบ 16 ครั้ง ผลปรากฏว่า ระบบ PROGOL หรือ GOLEM ให้เปอร์เซ็นต์ความถูกต้องต่ำกว่าระบบ TILDE และ LINUS ในชุดข้อมูลที่เป็นแบบหลายกลุ่ม และให้เปอร์เซ็นต์ความถูกต้องสูงกว่าระบบ TILDE และ LINUS ในชุดข้อมูลที่มีลักษณะเป็นสองกลุ่ม ผลการทดลองนี้แสดงให้เห็นว่ากฎที่ได้จากระบบ PROGOL และ GOLEM สามารถใช้ได้เป็นอย่างดีในปัญหาที่มีลักษณะเป็นสองกลุ่ม แต่ให้เปอร์เซ็นต์ความถูกต้องไม่ดีเมื่อใช้กับปัญหาที่มีลักษณะเป็นหลายกลุ่ม เมื่อใช้กระบวนการดึงลักษณะสำคัญและนิวรอล-เน็ตเวิร์กในระบบ BANNAR ทำให้เปอร์เซ็นต์ความถูกต้องสูงขึ้นและสูงกว่าระบบอื่นในทุกชุดข้อมูล โดยให้ความถูกต้องสูงกว่าอย่างมีนัยสำคัญทางสถิติสูงกว่า 90.0% จำนวน 12 ครั้ง จากการเปรียบเทียบ 16 ครั้ง และให้เปอร์เซ็นต์ความถูกต้องสูงกว่าอย่างมีนัยสำคัญทางสถิติต่ำกว่า 90.0% จำนวน 4 ครั้ง จากการเปรียบเทียบ 16 ครั้ง

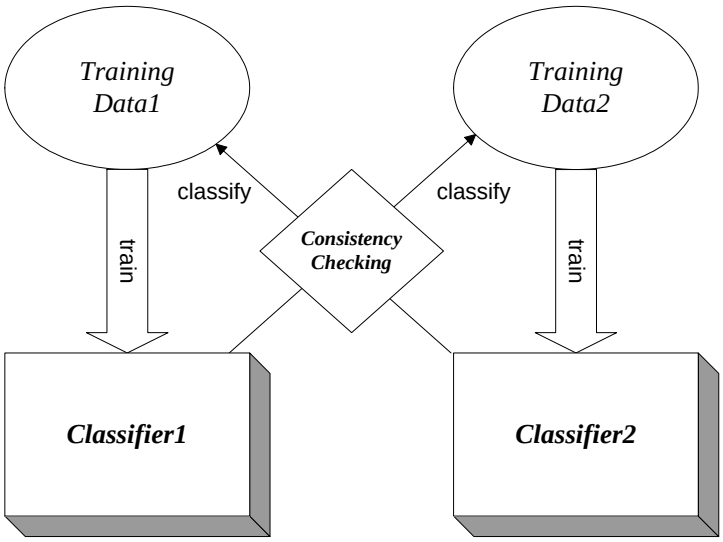
2. การวิจัยเทคนิคการเรียนรู้ของเครื่องที่สามารถใช้ประโยชน์จากข้อมูลแบบไม่มีฉลาก

วิธีที่นิยมใช้ในการจำแนกข้อมูลที่อยู่บนอินเตอร์เน็ตให้เป็นหมวดหมู่นั้น มักใช้วิธีการเรียนรู้โดยการสอน (supervised learning) ซึ่งสามารถทำได้โดยการติดฉลากเพื่อบอกหมวดหมู่ให้กับแต่ละเว็บเพจ และนำเว็บเพจเหล่านั้นไปสอนให้กับอัลกอริทึมในการเรียนรู้ แม้ว่าวิธีการเช่นนี้จะทำให้ได้ระบบจำแนกเว็บเพจที่มีความถูกต้องสูง แต่ก็จำเป็นต้องอาศัยแรงงานคนจำนวนมากเพื่อติดฉลากให้กับเว็บเพจ โดยเฉพาะอย่างยิ่งเว็บเพจบนอินเตอร์เน็ตมีการเปลี่ยนแปลงที่รวดเร็วมากทั้งในเชิงปริมาณที่เพิ่มขึ้นอย่างรวดเร็ว และในเชิงเนื้อหาที่มีการแก้ไขปรับปรุงกันอยู่เสมอ จึงเป็นเหตุให้การใช้แรงงานคนไม่สามารถทำได้อย่างมีประสิทธิภาพ

ในการวิจัยนี้เราต้องการทำให้ระบบค้นหาสามารถแยกหมวดหมู่ของเว็บเพจได้อย่างอัตโนมัติ โดยที่พยายามใช้ประโยชน์จากข้อมูลไม่มีฉลากให้มากที่สุด เพื่อลดแรงงานมนุษย์ เราจึงมุ่งเน้นที่จะหาวิธีการที่สามารถจำแนกประเภทของเว็บเพจอย่างอัตโนมัติ

งานวิจัยนี้นำเสนอเทคนิคใหม่ในการจำแนกเว็บเพจ วิธีการนี้มีชื่อว่า การสอนไขว้แบบวนซ้ำ (Iterative Cross-Training - ICT) วิธีการที่นำเสนอนี้สามารถใช้ประโยชน์จากข้อมูลที่ไม่มีฉลากได้อย่างมีประสิทธิภาพ เหมาะกับการใช้งานกับข้อมูลบนอินเตอร์เน็ตซึ่งมีเว็บเพจที่ไม่มีฉลากอยู่จำนวนมาก

2.1 อัลกอริทึมการสอนไขว้แบบวนซ้ำ



รูปที่ 4 การสอนไขว้แบบวนซ้ำ

รูปที่ 4 แสดงโมเดลของการสอนไขว้แบบวนซ้ำ ซึ่งประกอบด้วยตัวแยกแยะ (classifier) 2 ตัวคือ *Classifier1* และ *Classifier2* และชุดตัวอย่าง 2 ชุดคือ *TrainingData1* และ *TrainingData2* ซึ่งเป็นตัวอย่างที่ไม่มีฉลาก โดยการให้ความรู้เบื้องต้นเกี่ยวกับโดเมนที่จะเรียนรู้ หรือให้ตัวอย่างมีฉลากเริ่มต้นจำนวนน้อยๆ ตัวแยกแยะทั้งสองจะประมาณค่าของพารามิเตอร์ที่ต้องเรียนจากชุดตัวอย่าง โดยอาศัยการโต้ตอบและการสอนระหว่างกันเอง ตัวอย่างของความรู้เบื้องต้นก็อย่างเช่น ถ้าต้องการให้เรียนรู้ว่าเว็บเพจใดเป็นภาษาไทย เพจใดเป็นภาษาอื่น เราก็จะให้ความรู้เบื้องต้นในรูปของพจนานุกรมภาษาไทย ในกรณีที่เราไม่มีความรู้เบื้องต้นเกี่ยวกับโดเมนที่จะเรียนรู้ เราจะให้ตัวอย่างมีฉลากเริ่มต้นจำนวนน้อยๆ เพื่อให้โมเดลเริ่มกระบวนเรียนรู้ สำหรับชุดตัวอย่าง *TrainingData1* และ *TrainingData2* นั้นได้มาจากการทำสำเนาข้อมูลที่ผู้ใช้ให้มา ให้ θ_1 เป็นชุดพารามิเตอร์ (parameter set) ของ *Classifier1* และ θ_2 เป็นชุดพารามิเตอร์ของ *Classifier2* ชุดข้อมูล *TrainingData1* ใช้สำหรับสอน *Classifier1* ให้เรียนรู้พารามิเตอร์ ส่วน *TrainingData2* ใช้สำหรับสอน *Classifier2* อัลกอริทึมของการสอนไขว้แบบวนซ้ำแสดงในตารางที่ 2 ต่อไปนี้

ตารางที่ 2 อัลกอริทึมการสอนไขว้แบบวนซ้ำ

อินพุต:

- *TrainingData1* และ *TrainingData2* เป็นชุดของตัวอย่างไม่มีฉลาก

(1) กำหนดค่าเริ่มต้นของชุดพารามิเตอร์สำหรับ *Classifier1* เป็น θ_{10}

$$\theta_1 \leftarrow \theta_{10}$$

(2) กำหนดค่าเริ่มต้นของชุดพารามิเตอร์สำหรับ *Classifier2* เป็น θ_{20}

$$\theta_2 \leftarrow \theta_{20}$$

(3) วนซ้ำจนกระทั่ง θ_1 ไม่เปลี่ยนแปลงหรือจำนวนรอบเกินกว่าค่าขีดแบ่ง

- ใช้ *Classifier1* และชุดพารามิเตอร์ θ_1 เพื่อติดฉลากให้กับข้อมูลทุกตัวใน *TrainingData2* ให้เป็นตัวอย่างบวกและตัวอย่างลบ พร้อมกับตรวจเช็คความสอดคล้องของการแยกแยะกับ *Classifier2* ถ้าจำเป็น
- สอน *Classifier2* โดยใช้ตัวอย่างมีฉลากใน *TrainingData2* เพื่อประมาณค่าของชุดพารามิเตอร์ θ_2 ของ *Classifier2*
- ใช้ *Classifier2* และชุดพารามิเตอร์ θ_2 เพื่อติดฉลากให้กับข้อมูลทุกตัวใน *TrainingData1* ให้เป็นตัวอย่างบวกและตัวอย่างลบ พร้อมกับตรวจเช็คความสอดคล้องของการแยกแยะกับ *Classifier1* ถ้าจำเป็น
- สอน *Classifier1* โดยใช้ตัวอย่างมีฉลากใน *TrainingData1* เพื่อประมาณค่าของชุดพารามิเตอร์ θ_1 ของ *Classifier1*

แนวคิดของอัลกอริทึมด้านบนคือ หากเราสามารถดึงเอาข้อมูลทางสถิติที่น่าเชื่อถือออกมาจาก *TrainingData2* ได้ก็น่าจะมีประโยชน์ในการจำแนก *TrainingData1* ถ้าหากว่าชุดพารามิเตอร์เริ่มต้นของ *Classifier1*(θ_{10}) มีคุณสมบัติที่จะติดฉลากข้อมูลได้ถูกต้องมากกว่าติดฉลากผิดพลาดสำหรับ *TrainingData2* แล้ว เราก็น่าจะได้ข้อมูลทางสถิติที่แฝงอยู่ในตัวอย่างที่ทำนายถูก ซึ่งข้อมูลทางสถิตินี้เองที่จะช่วยให้ *Classifier2* สามารถติดฉลากได้อย่างถูกต้องให้กับตัวอย่างใน *TrainingData1* ที่มีคุณลักษณะคล้ายกัน และถ้า *TrainingData1* ที่มีการกำหนดฉลากใหม่นี้สามารถให้ θ_1 ที่ดีกว่า θ_{10} แล้ว เราก็น่าจะได้ชุดพารามิเตอร์ที่น่าเชื่อถือมากยิ่งๆ ขึ้นเมื่อผ่านไปในแต่ละรอบของการสอน ในขั้นตอนแรกของอัลกอริทึมนั้น เราต้องกำหนดค่าเริ่มต้นให้กับชุดพารามิเตอร์ของ *Classifier1* และ *Classifier2* ซึ่งทำได้โดยการสอนตัวแยกแยะด้วยข้อมูลจำนวนหนึ่งที่มีฉลาก (ถ้าหากเรามีข้อมูลเหล่านั้น) หรือถ้าเราไม่มีข้อมูลที่มีฉลากอยู่เลย ก็อาจกำหนดให้เป็นค่าใดๆ ที่มีการกำหนดไว้ล่วงหน้าหรืออาจกำหนดค่าโดยการสุ่มก็ได้

ในขณะที่ตัวแยกแยะที่กำลังทำงาน (active classifier) จะติดฉลากให้กับข้อมูล ตัวแยกแยะนั้นสามารถที่จะถามเพื่อยืนยันจากตัวแยกแยะอีกตัวได้ว่าตัวอย่างที่สนใจอยู่นั้นควรจัดอยู่ในกลุ่ม (class) ใด ถ้าหากตัวแยกแยะทั้งคู่เห็นพ้องกัน ตัวอย่างนั้นก็จะถูกกำหนดฉลาก แต่ถ้าหากว่าตัวแยกแยะทั้งคู่เห็นแตกต่างกันและตัวแยกแยะที่กำลังทำงานมีค่าความมั่นใจ (confident value) มากกว่า ตัวอย่างนั้นก็จะถูกกำหนดฉลากตามตัวแยกแยะที่กำลังทำงาน ถ้าไม่เช่นนั้นแล้วตัวอย่างนั้นก็จะไม่ถูกกำหนดฉลาก จุดมุ่งหมายของการตรวจเช็คความสอดคล้องกันนี้มีเพื่อให้การกำหนดฉลากของตัวอย่างมีความน่าเชื่อถือมากขึ้น อย่างไรก็ดีการตรวจเช็คนี้จะทำให้การทำงานของระบบช้าลง

เราได้นำวิธีการสอนไขว้แบบวนซ้ำประยุกต์ใช้กับปัญหาการจำแนกเว็บเพจ 2 ปัญหาคือ (1) การจำแนกประเภทเว็บเพจว่าเป็นเพจภาษาไทยหรือไม่ใช่ และ (2) การจำแนกประเภทเว็บเพจออกเป็นหมวดหมู่ ส่วน

ต่อไปจะกล่าวถึงรายละเอียดในปัญหาทั้งสองนี้ จากนั้นจะอธิบายถึงผลกระทบของข้อมูลสัญญาณรบกวนกับปัญหาการจำแนกประเภทเว็บเพจ

2.2 การจำแนกประเภทเว็บเพจภาษาไทยและภาษาอื่น

ในปัญหาของการจำแนกประเภทเว็บเพจภาษาไทยและไม่ใช่ภาษาไทยที่จะกล่าวในหัวข้อนี้ มีจุดมุ่งหมายคือเราต้องการจัดประเภทของเว็บเพจออกเป็นสองกลุ่ม คือ กลุ่มที่เป็นเว็บเพจภาษาไทยและกลุ่มที่ไม่ใช่ ซึ่งสามารถนำไปประยุกต์ใช้ในการสร้างหุ่นยนต์เว็บให้สามารถค้นหาเฉพาะเว็บเพจที่เป็นภาษาไทย เพื่อนำไปสร้างระบบค้นหาสำหรับเว็บเพจไทยได้ สำหรับปัญหานี้ตัวแยกแยะย่อยตัวแรกคือ *Classifier1* จะได้รับความรู้เบื้องต้นในรูปของพจนานุกรมไทย และใช้พจนานุกรมเพื่อช่วยในการจำแนกเว็บเพจว่าเป็นภาษาไทยหรือไม่ อัลกอริทึมที่ใช้ในตัวแยกแยะย่อยนี้คืออัลกอริทึมตัดคำภาษาไทย ส่วนตัวแยกแยะย่อยตัวที่สอง *Classifier2* จะไม่ได้รับความรู้เบื้องต้นและจะใช้อัลกอริทึมเบย์อย่างง่าย (Naïve Bayes) ดังจะกล่าวต่อไปนี้

2.2.1 ตัวแยกแยะในการจำแนกประเภทเว็บเพจภาษาไทยและภาษาอื่น

(1) ตัวแยกแยะตัดคำภาษาไทย

วิธีที่ง่ายที่สุดที่จะตรวจสอบว่าเว็บเพจเป็นเว็บเพจภาษาไทยหรือไม่ ทำได้โดยตรวจสอบคำในเว็บเพจว่าเป็นคำไทยหรือไม่ ถ้ามีคำ*จำนวนมาก*ปรากฏในพจนานุกรม ก็แสดงว่าเว็บเพจนั้นน่าจะเป็นภาษาไทย อย่างไรก็ดีเราคงคาดหวังไม่ได้ว่าคำทุกคำในเว็บเพจนั้นต้องปรากฏในพจนานุกรมทั้งหมด เนื่องจากว่าในเว็บเพจไทยนั้นมักจะมีคำในภาษาอื่น เช่น ภาษาอังกฤษ ปะปนอยู่ด้วย นอกจากนั้นแล้วมักจะมีคำเฉพาะ เช่น ชื่อคน ชื่อสถานที่ต่าง ๆ ปรากฏอยู่ด้วย ซึ่งคำเฉพาะเหล่านี้มักจะไม่มีอยู่ในพจนานุกรม ดังนั้นจึงจำเป็นต้องกำหนดให้ได้ว่าควรจะมีคำไทยปรากฏมากน้อยเท่าไร จึงจะจัดว่าเป็นเว็บเพจไทย ปัญหานี้มีความยุ่งยากมากขึ้นโดยเฉพาะในภาษาไทย ซึ่งคำเขียนต่อเนื่องกันโดยไม่มีเครื่องหมายวรรคตอนคั่น ด้านล่างนี้อธิบายวิธีการตัดคำภาษาไทยที่ใช้ในงานวิจัยนี้

กำหนดให้เอกสาร d มีตัวอักษร n ตัว $(c_1, c_2,\ldots,c_n)$ ตัวแยกแยะตัดคำของเราจะสร้างการตัดคำทุกแบบที่เป็นไปได้และเลือกการตัดคำที่ดีที่สุดซึ่งให้ค่าฟังก์ชันด้านล่างนี้น้อยที่สุด

$$\text{argmin} \sum_{i=1}^{m} \text{cost}(w_i) \qquad (1)$$

โดยที่ $\text{cost}(w_i) = \eta_1$ ถ้า w_i เป็นคำที่ปรากฏในพจนานุกรม

$= \eta_2$ ถ้า w_i เป็นสายอักขระที่ไม่ปรากฏในพจนานุกรม

สำหรับการทดลองในหัวข้อที่ 2.2.3 นั้น เรากำหนดให้ η_1 และ η_2 มีค่าเท่ากับ 1 และ 2 ตามลำดับ

หลังจากที่ได้การตัดคำที่ดีที่สุดแล้ว เอกสารจะประกอบด้วย (1) คำที่ปรากฏในพจนานุกรม และ (2) สายอักขระที่ไม่ปรากฏในพจนานุกรม เว็บเพจไทยควรจะเป็นเพจที่มีคำไทยจำนวนมากและสายอักขระที่ไม่รู้จักจำนวนน้อย เราจึงนิยาม *WordRatio* ให้มีค่าเท่ากับ

$$\frac{\text{จำนวนตัวอักษรที่อยู่ในคำทั้งหมด}}{\text{จำนวนตัวอักษรทั้งหมดที่อยู่ในเอกสาร}} \qquad (2)$$

เมื่อให้เซตของตัวอย่างบวกและตัวอย่างลบ ตัวแยกแยะตัดคำจะหาค่าขีดแบ่งของ *WordRatio* ที่ทำให้ตัวอย่างบวกและลบซึ่งถูกจัดจำพวกอย่างถูกต้องมีจำนวนสูงสุด เอกสารใดที่มี *WordRatio* มากกว่าค่าขีดแบ่งจะถูกจัดจำพวกให้เป็นตัวอย่างบวก (เว็บเพจไทย) แต่เอกสารที่มี *WordRatio* น้อยกว่าค่าขีดแบ่งจะเป็นตัวอย่างลบ (ไม่ใช่เว็บเพจไทย) และเราใช้ค่าขีดแบ่งของ *WordRatio* เป็นพารามิเตอร์ของตัวแยกแยะตัดคำ

จากการใช้ค่าขีดแบ่งของ *WordRatio* เพียงตัวเดียวเป็นพารามิเตอร์ เราสามารถหา θ_{10} ซึ่งติดฉลากให้กับตัวอย่างได้ถูกต้องมากกว่าติดผิดพลาดได้ เว็บเพจไทยควรมีค่า *WordRatio* สูงและเว็บเพจที่ไม่ใช่ภาษาไทยควรให้ค่าต่ำ ถ้าจำนวนเว็บเพจไทยและไม่ใช่ไทยในเซตของตัวอย่างมีจำนวนเท่ากัน เราจะได้ว่าทุกค่าของค่าขีดแบ่ง *WordRatio* จะจำแนกตัวอย่างได้ถูกต้องมากกว่าจำแนกผิดพลาด (ยกเว้น $\theta_{10} = 0.0$ และ $\theta_{10} = 1.0$ ซึ่งให้จำนวนตัวอย่างที่จำแนกถูกและผิดเท่ากัน) ในกรณีที่เว็บเพจไทยมีจำนวนน้อยกว่าเว็บเพจที่ไม่ใช่ภาษาไทยนั้น θ_{10} ที่มีค่ามาก (เช่น 0.7, 0.8 หรือ 0.9) จะจำแนกตัวอย่างได้ถูกต้องมากกว่าจำแนกผิดพลาด θ_{10} ที่มีค่าน้อยจะเหมาะกับกรณีที่เว็บเพจไทยมีจำนวนมากกว่าเพจที่ไม่ใช่ไทย

เราสามารถปรับค่า θ_1 ใหม่ได้หลังจากที่ *Classifier2* ติดฉลากให้กับข้อมูลใน *TrainingData1* ดังนี้

ให้ *SP* เป็นค่าต่ำสุดของ *WordRatio* ที่ได้จากตัวอย่างบวกทั้งหมด และ *LN* เป็นค่าสูงสุดของ *WordRatio* ที่ได้จากตัวอย่างลบทั้งหมด ในกรณีที่ $SP \geq LN$ เราสามารถปรับค่าของ θ_1 ได้โดยใช้สมการด้านล่างนี้

$$\theta_1 = \frac{SP+LN}{2} \tag{3}$$

พิจารณากรณีที่ $SP<LN$ ให้ $V_1=SP$, $V_n=LN$ และ $V_2,\ldots,V_{n-1}$ เป็นค่าที่อยู่ระหว่าง V_1 กับ V_n ($V_1 \leq V_2 \leq \cdots \leq V_{n-1} \leq V_n$) เราสามารถปรับค่า θ_1 ใหม่ได้ดังนี้

$$\theta_1 = \frac{V_{i*}+V_{i*+1}}{2} \tag{4}$$

$$V_{i*} = \text{argmin (no. of } V_j \text{ + no. of } V_k)$$

โดยที่ V_k เป็นค่า *WordRatio* ของตัวอย่างที่ถูกติดฉลากเป็นบวก และ V_j เป็นค่าของตัวอย่างที่ถูกติดฉลากเป็นลบ และ $V_1 \leq V_k \leq V_i$, $V_{i+1} \leq V_j \leq V_n$

ถ้า *SP* มากกว่าหรือเท่ากับ *LN*, θ_1 จะสามารถแยกตัวอย่างบวกออกจากตัวอย่างลบได้โดยสมบูรณ์ แต่ถ้า *SP* น้อยกว่า *LN*, θ_1 จะให้จำนวนตัวอย่างที่แยกผิดพลาดน้อยที่สุด

(2) ตัวแยกแยะเบย์แบบง่าย

สำหรับการแยกแยะข้อความนั้น ตัวแยกแยะเบย์อย่างง่ายเป็นวิธีการที่ได้รับความนิยมและเป็นวิธีที่มีประสิทธิภาพมากที่สุดวิธีหนึ่ง [Mitchell, 1997] วิธีนี้จะใช้ "*ถุงคำ (bag-of-words)*" เพื่อแทนข้อความ ในปัญหาการแยกเว็บเพจออกเป็นภาษาไทยและไม่ใช่นั้น เราจะใช้ "*ถุงตัวอักษร (bag-of-characters)*" แทน ซึ่งถุงตัวอักษรจะเหมาะกับปัญหานี้ เพื่อให้หุ่นยนต์เว็บสามารถค้นหาเว็บเพจไทยได้อย่างมีประสิทธิภาพ เพราะว่าไม่ต้องอาศัยการตัดคำ ดังนั้นการทำงานจะรวดเร็วมาก แม้ว่าการแทนข้อความด้วยถุงตัวอักษรจะง่ายกว่าการแทนด้วยถุงคำก็ตาม แต่การทดลองในหัวข้อต่อไปได้แสดงให้เห็นว่าวิธีการนี้ทำงานได้อย่างดี

กำหนดให้ $L = \{l_1,l_2,\ldots,l_m\}$ เป็นเซตของฉลากและ $d=(c_1,c_2,\ldots,c_n)$ เป็นเอกสารที่มี n ตัวอักษร ฉลากแสดงกลุ่มที่น่าจะเป็นที่สุด (l^*) สามารถประมาณค่าได้โดยใช้เทคนิคของเบย์อย่างง่าย ดังสมการด้านล่างนี้

$$\begin{aligned} l^* &= \underset{l_j}{\text{argmax}} \; Pr(l_j \mid c_1,\ldots,c_n) \\ &= \underset{l_j}{\text{argmax}} \; \frac{Pr(l_j)Pr(c_1,\ldots,c_n \mid l_j)}{Pr(c_1,\ldots,c_n)} \\ &= \underset{l_j}{\text{argmax}} \; Pr(l_j)Pr(c_1,\ldots,c_n \mid l_j) \end{aligned} \tag{5}$$

ในปัญหาของการจำแนกประเภทของเว็บเพจไทยหรือไม่ใช่ไทยนี้ L คือเซตของฉลากภาษาไทยและไม่ใช่ภาษาไทย ส่วน $d = (c_1,c_2,\ldots,c_n)$ โดยทั่วไปแล้วจะมีค่าที่เป็นไปได้ทั้งหมดมากมายมหาศาล ทำให้การคำนวณ $Pr(c_1,c_2,\ldots,c_n \mid l_j)$ ต้องใช้ข้อมูลจำนวนมหาศาลเพื่อให้ได้ความน่าจะเป็นที่น่าเชื่อถือได้ ดังนั้นเพื่อที่จะลดจำนวนของข้อมูลและปรับปรุงการประมาณค่าความน่าจะเป็นให้มีความน่าเชื่อถือมากยิ่งขึ้น เรานิยมใช้สมมติฐานของเบย์ดังนี้คือ (1) สมมติฐานเกี่ยวกับการไม่ขึ้นต่อกันอย่างมีเงื่อนไข (conditional independent assumption) กล่าวคือ การปรากฏของตัวอักษรตัวหนึ่งๆ จะไม่ขึ้นกับตัวอักษรอื่นๆ ทั้งหมดในเอกสารเมื่อรู้กลุ่มของตัวอย่าง และ (2) สมมติฐานที่ว่าตำแหน่งของตัวอักษรในเอกสารหรือเว็บเพจไม่มีความสำคัญ กล่าวคือ การพบตัวอักษร “ก” อยู่ที่ตำแหน่งแรกสุดของเอกสารมีค่าเหมือนกับการพบ “ก” ที่ท้ายเอกสาร แน่นอนว่าสมมติฐานเหล่านี้มักจะไม่เป็นจริงในทางปฏิบัติ แต่ผลการทดลองของตัวแยกแยะเบย์อย่างง่ายได้แสดงให้เห็นถึงประสิทธิภาพที่สูงของตัวแยกแยะนี้ในงานประเภทการแยกแยะข้อความ [Joachims, 1998; McCallum, et al., 1998; Yang & Pederson, 1997]

จากสมมติฐานด้านบนทำให้เราสามารถเขียนสมการด้านบนใหม่ดังต่อไปนี้

$$l^* = \underset{l_j}{\operatorname{argmax}} \; Pr(l_j) \prod_{i=1}^{n} Pr(c_i \mid l_j, c_1,\ldots,c_{i-1})$$

$$= \underset{l_j}{\operatorname{argmax}} \; Pr(l_j) \prod_{i=1}^{n} Pr(c_i \mid l_j) \qquad (6)$$

ค่าความน่าจะเป็น $Pr(l_j)$ และ $Pr(c_i \mid l_j)$ คือชุดพารามิเตอร์ θ_2 สำหรับตัวแยกแยะเบย์อย่างง่ายและประมาณค่าได้จากเซตของตัวอย่างสอน $Pr(l_j)$ ประมาณค่าได้จากอัตราส่วนระหว่างจำนวนตัวอย่างที่อยู่ในกลุ่ม l_j กับจำนวนตัวอย่างทั้งหมด ส่วน $Pr(c_i \mid l_j)$ หรือความน่าจะเป็นที่เราจะเห็นตัวอักษร c_i เมื่อรู้ว่ากลุ่มคือ l_j ประมาณค่าได้ดังต่อไปนี้

$$Pr(c_i \mid l_j) = \frac{1 + N(c_i,l_j)}{T + N(l_j)} \qquad (7)$$

โดยที่ $N(c_i,l_j)$ คือจำนวนครั้งที่ตัวอักษร c_i ปรากฏในเซตของตัวอย่างสอนจากกลุ่ม l_j
$N(l_j)$ คือ จำนวนตัวอักษรทั้งหมดในเซตของตัวอย่างสอนจากกลุ่ม l_j และ
T คือจำนวนตัวอักษรที่แตกต่างกันทั้งหมดในเซตของตัวอย่างสอน

2.2.2 อัลกอริทึมที่นำมาเปรียบเทียบในการจำแนกประเภทเว็บเพจภาษาไทยและภาษาอื่น

ในการทดลองเพื่อประเมินประสิทธิภาพของการสอนไขว้แบบวนซ้ำนั้น เราได้เปรียบเทียบกับวิธีการต่อไปนี้

(1) ตัวแยกแยะตัดคำแบบสอน (Supervised Word Segmentation Classifier)
(2) ตัวแยกแยะเบย์อย่างง่ายแบบสอน (Supervised Naïve Bayes Classifier)
(3) อัลกอริทึมโคเทรนนิง (CoTraining Algorithm) และ
(4) อัลกอริทึมอีเอ็ม (EM Algorithm)

ตัวแยกแยะตัดคำแบบสอนและตัวแยกแยะเบย์อย่างง่ายแบบสอนมีอัลกอริทึมเหมือนกันกับตัวแยกแยะที่อธิบายไว้ในหัวข้อที่ 2.2.1 ยกเว้นว่าตัวแยกแยะในหัวข้อนี้จะใช้ข้อมูลที่มีฉลากทั้งหมดในการสอน ส่วนอัลกอริทึมโคเทรนนิงและอัลกอริทึมอีเอ็มเป็นดังต่อไปนี้

อัลกอริทึมโคเทรนนิง

อัลกอริทึมโคเทรนนิงถูกนำเสนอใน [Blum & Mitchell, 1998] แนวคิดของวิธีการนี้คือตัวอย่างสามารถมองได้สองมุมมอง เช่น เว็บเพจสามารถพิจารณาได้จากคำที่อยู่ในเนื้อหาของเพจนั้น หรือสามารถพิจารณาได้จากคำที่อยู่ในไฮเปอร์ลิงค์ของเพจที่ชี้มายังเพจนั้น และมีสมมติฐานที่ว่าแต่ละมุมมองเพียงพอต่อการเรียนรู้เพื่อจำแนกประเภทของตัวอย่าง อัลกอริทึมจะประกอบด้วยตัวแยกแยะย่อยสองตัว แต่ละตัวเรียนรู้จากคนละมุมมอง

จากแนวคิดนี้ เราได้ทดลองเขียนโปรแกรมตามอัลกอริทึมโคเทรนนิงดังแสดงในตารางที่ 3 ตัวแยกแยะย่อยในตารางมีอัลกอริทึมเหมือนกับของการสอนไขว้แบบวนซ้ำ ในการใช้อัลกอริทึมโคเทรนนิงในกรณีของปัญหาการจำแนกประเภทเว็บเพจภาษาไทยและไม่ใช่นั้น เรามองเว็บเพจแต่ละเพจเป็นเซตของคำที่ปรากฏในเพจนั้น หรือ เซตของตัวอักษรที่ปรากฏในเพจนั้น และใช้ตัวแยกแยะตัดคำ (*Classifier1*) เพื่อเรียนรู้จากเซตของคำ และใช้ตัวแยกแยะเบย์อย่างง่าย (*Classifier2*) เพื่อเรียนรู้จากเซตของตัวอักษรในเพจนั้น การปรับค่า θ_1 และ θ_2 ของตัวแยกแยะทั้งสองมีวิธีการเหมือนกับวิธีการที่อธิบายไว้ในหัวข้อที่ 2.2.1

ตารางที่ 3 อัลกอริทึมโคเทรนนิง

อินพุต:

- เซตของตัวอย่างที่มีฉลาก *LE* และ
- เซตของตัวอย่างที่ไม่มีฉลาก *UE*

(1) สร้างเซต *UE'* โดยเลือกตัวอย่าง *u* ตัวจากเซต *UE*

(2) วนซ้ำจนกระทั่งไม่มีตัวอย่างเหลืออยู่ใน *UE*

- ใช้ *UE* เพื่อประมาณค่าของชุดพารามิเตอร์ θ_1 ของ *Classifier1*
- ใช้ *UE* เพื่อประมาณค่าของชุดพารามิเตอร์ θ_2 ของ *Classifier2*
- ใช้ *Classifier1* พร้อมด้วยชุดพารามิเตอร์ θ_1 เพื่อติดฉลากให้กับตัวอย่างบวก *p* ตัวและตัวอย่างลบ *n* ตัวจาก *UE'*
- ใช้ *Classifier2* พร้อมด้วยชุดพารามิเตอร์ θ_2 เพื่อติดฉลากให้กับตัวอย่างบวก *p* ตัวและตัวอย่างลบ *n* ตัวจาก *UE'*
- นำตัวอย่างที่ติดฉลากใหม่นี้เพิ่มเข้าไปใน *LE*
- เลือกตัวอย่าง *2p+2n* ตัวอย่างสุ่มจาก *UE* และนำไปเพิ่มใน *UE'*

อัลกอริทึมอีเอ็ม

อัลกอริทึมอีเอ็ม (Expectation-Maximization – EM algorithm) ถูกนำเสนอใน [Dempster et al. 1977] อัลกอริทึมนี้เป็นอัลกอริทึมประเภทวนซ้ำแบบหนึ่ง เพื่อใช้แก้ปัญหาของข้อมูลไม่สมบูรณ์ (incomplete data) เมื่อกำหนดโมเดลของการสร้างข้อมูลและข้อมูลที่มีค่าบางส่วนหายไป อัลกอริทึมนี้จะใช้โมเดลปัจจุบัน เพื่อประมาณค่าที่หายไป แล้วใช้ค่าที่ประมาณได้เพื่อปรับโมเดลให้ดีขึ้น อัลกอริทึมจะคำนวณค่าความน่าจะเป็น (likelihood) ของพารามิเตอร์ของโมเดลที่ดีที่สุดแบบท้องถิ่น (local) เพื่อประมาณค่าที่หายไป ในกรณีที่เรานำอัลกอริทึมอีเอ็มมาใช้ในการเรียนรู้เว็บเพจที่ไม่มีฉลากนั้น ฉลากจะถูกมองว่าเป็นค่าที่หายไป อัลกอริทึมอีเอ็มแสดงในตารางที่ 4

ตารางที่ 4 อัลกอริทึมอีเอ็มสำหรับตัวแยกแยะเบย์อย่างง่าย

อินพุต:

- เซตของตัวอย่างที่ไม่มีฉลาก UE

(1) กำหนดค่าเริ่มต้นของชุดพารามิเตอร์ $Pr(c_i|l_j)$ และ $Pr(l_j)$ สำหรับ *Classifier* โดยเรียนจากตัวอย่างมีฉลากเริ่มต้น แล้วนำพารามิเตอร์เหล่านี้ไปติดฉลากให้กับตัวอย่างทั้งหมดใน UE

(2) ใช้ตัวอย่างที่ติดฉลากแล้ว ไปประมาณค่าพารามิเตอร์ $Pr(c_i|l_j)$ และ $Pr(l_j)$ ใหม่ของ *Classifier* โดยที่ $Pr(l_j|d) \in \{0,1\}$

(3) วนซ้ำจนกระทั่งพารามิเตอร์ไม่เปลี่ยนแปลงหรือจำนวนรอบเกินกว่าค่าขีดแบ่ง

—(E-step) ประมาณค่าฉลากของกลุ่มข้อมูลแบบถ่วงน้ำหนักตามความน่าจะเป็นให้เท่ากับ $Pr(l_j|d)$ สำหรับทุกเอกสาร ตามสมการที่ 10

—(M-step) ใช้ฉลากของกลุ่มข้อมูลที่ประมาณค่าได้ตาม $Pr(l_j|d)$ เพื่อคำนวณค่าพารามิเตอร์ใหม่โดยใช้เอกสารทั้งหมดตามสมการที่ 8 และ 9

ในงานวิจัยนี้โมเดลของการสร้างข้อมูลจะใช้ตัวแยกแยะเบย์อย่างง่าย อัลกอริทึมนี้ประกอบไปด้วย 2 ขั้นตอนคือ ขั้นตอนอี (E-step) และขั้นตอนเอ็ม (M-step) ขั้นตอนอีจะคำนวณฉลากกลุ่มข้อมูลแบบถ่วงน้ำหนักตามความน่าจะเป็น (probabilistically weighted class labels) สำหรับทุกเอกสารโดยใช้ตัวแยกแยะ และขั้นตอนเอ็มจะประมาณค่าพารามิเตอร์ใหม่โดยใช้เอกสารทั้งหมดที่ถูกคำนวณฉลากแบบถ่วงน้ำหนักแล้ว กระบวนการของขั้นตอนอีและขั้นตอนเอ็มจะวนซ้ำจนกระทั่งพารามิเตอร์มีค่าไม่เปลี่ยนแปลง Nigam และคณะ [Nigam et al. 1999] ได้ใช้อัลกอริทึมอีเอ็มสำหรับปัญหาการแยกแยะข้อความ

ดังที่แสดงในตารางที่ 4 ขั้นตอนแรกของอัลกอริทึมอีเอ็มคือ การประมาณค่าพารามิเตอร์ของตัวแยกแยะเบย์อย่างง่าย โดยเรียนจากตัวอย่างมีฉลากเริ่มต้น จากนั้นตัวแยกแยะจะใช้ค่าพารามิเตอร์ที่ได้นำไปติดฉลากให้กับตัวอย่างไม่มีฉลากทุกตัว หลังจากนั้นกระบวนการเรียนรู้จะวนทำขั้นตอนอี (E-step) และขั้นตอนเอ็ม (M-step) จนกระทั่งอัลกอริทึมลู่เข้า การประมาณค่าฉลากของกลุ่มข้อมูลแบบถ่วงน้ำหนักสามารถคำนวณได้ ดังต่อไปนี้

กำหนดให้ $L = \{l_1,l_2,\ldots,l_m\}$ เป็นเซตของฉลากและ $d = (c_1,c_2,\ldots,c_n)$ เป็นเอกสารที่มีตัวอักษร n ตัว จากชุดข้อมูลสอน D, $Pr(l_j|d) \in \{0, 1\}$ เป็นฉลากของเอกสาร d ค่าประมาณความน่าจะเป็นที่ตัวอักษร c_i จะอยู่ในฉลาก l_j คือ

$$Pr(c_i \mid l_j) = \frac{1+\sum_{d\in D} N(c_i,d)Pr(l_j \mid d)}{T+\sum_{k=1}^{T}\sum_{d\in D} N(c_k,d)Pr(l_j \mid d)} \qquad (8)$$

โดยที่ T คือจำนวนตัวอักษรที่แตกต่างกันทั้งหมดในเซตของตัวอย่างสอน

$N(c_i,d)$ คือจำนวนครั้งที่ตัวอักษร c_i ปรากฏในเอกสาร d

ส่วนความน่าจะเป็นของฉลากหนึ่งๆ ที่จะเกิดขึ้นมีค่าตามสมการที่ (9) ต่อไปนี้

$$Pr(l_j) = \frac{1 + \sum_{d \in D} Pr(l_j \mid d)}{|L| + |D|} \quad (9)$$

โดยที่ $|L|$ และ $|D|$ คือจำนวนของฉลากที่แตกต่างกันและจำนวนเอกสารในชุดข้อมูลสอนตามลำดับ เมื่อกำหนดเอกสาร $d = (c_1, c_2, \ldots, c_n)$ ที่ไม่มีฉลากซึ่งมีตัวอักษร n ตัวให้ ตัวแยกแยะเบย์อย่างง่ายจะประมาณค่าความน่าจะเป็นที่เอกสารนี้มีฉลากเป็น l_j โดยใช้สมการ (10) ด้านล่างนี้

$$Pr(l_j \mid d) = \frac{Pr(l_j) Pr(d \mid l_j)}{Pr(d)} = \frac{Pr(l_j) \prod_{i=1}^{n} Pr(c_i \mid l_j)}{\sum_{k=1}^{|L|} Pr(l_k) \prod_{i=1}^{n} Pr(c_i \mid l_k)} \quad (10)$$

สังเกตว่าในสมการนี้ $Pr(l_j|d)$ เป็นค่าถ่วงน้ำหนักตามความน่าจะเป็น กล่าวคือแต่ละเอกสาร d จะถูกพิจารณาว่ามีฉลากเป็น l_j ด้วยความน่าจะเป็นเท่ากับ $Pr(l_j|d)$

2.2.3 ผลการทดลองการจำแนกประเภทเว็บเพจภาษาไทยและภาษาอื่น

ชุดข้อมูลและตั้งค่าการทดลองในปัญหาการแยกแยะเว็บเพจภาษาไทยและไม่ใช่

เรารวบรวมข้อมูลเพื่อทำการทดลองโดยเริ่มจากเว็บเพจ 4 เพจ คือ เว็บเพจภาษาญี่ปุ่น 1 เพจ[1] เว็บเพจภาษาไทย 2 เพจ[2] และ เว็บเพจภาษาอังกฤษ 1 เพจ[3] เริ่มต้นจากเว็บเพจทั้งสี่ เราให้หุ่นยนต์เว็บวิ่งตามไฮเปอร์ลิงค์ภายในเพจเหล่านั้น เพื่อเข้าถึงเว็บเพจอื่นๆ จนกระทั่งหุ่นยนต์เว็บรวบรวมเว็บเพจได้ 450 เพจสำหรับเว็บเพจเริ่มต้นแต่ละเพจ ดังนั้นเว็บเพจที่รวบรวมได้ทั้งสิ้นประกอบด้วยเว็บเพจภาษาไท*ยประมาณ* 900 เพจ เนื่องจากเพจภาษาไทยอาจจะชี้ไปยังเพจที่เป็นภาษาอังกฤษหรือภาษาอื่นๆ อีก ในทำนองเดียวกัน จะได้เว็บเพจที่เป็นภาษาญี่ปุ่นและภาษาอังกฤษอย่างละประมาณ 450 เพจ จากนั้นเรานำเว็บเพจทั้งหมดมารวมกันแล้วแบ่งออกเป็น 3 เซตคือ เซต A, B และ C แต่ละเซตมีเว็บเพจทั้งหมด 600 เพจ (เพจไทย อังกฤษและญี่ปุ่นประมาณ 300, 150 และ 150 ตามลำดับ) เราใช้วิธีการทดลองแบบ 3-fold cross-validation เพื่อวัดผลการทดลอง ในการทดลองแบบนี้ แต่ละเซตจะถูกนำมาเป็นชุดทดสอบเซตละหนึ่งครั้งโดยที่ใช้เซตที่เหลือเป็นชุดสอน แล้วหาค่าเฉลี่ยที่ได้เป็นผลการทดลอง

การตั้งค่าสำหรับอัลกอริทึมเป็นดังต่อไปนี้

- สำหรับการสอนไขว้แบบวนซ้ำนั้น ไม่มีการให้ตัวอย่างที่มีฉลากเริ่มต้นกับอัลกอริทึมการสอนไขว้แบบวนซ้ำ ค่าพารามิเตอร์เริ่มต้น θ_1 กำหนดให้มีค่าเป็น 0.7
- เนื่องจากอัลกอริทึมโคเทรนนิง (แสดงด้วย CoTraining ในตาราง) ต้องการตัวอย่างมีฉลากสำหรับเริ่มทำงาน ดังนั้นเราให้ตัวอย่างมีฉลากเริ่มต้นทั้งหมด 18 ตัว ในการทดลองพารามิเตอร์อื่นๆ ถูกกำหนดค่าดังต่อไปนี้ $|UE|$, p, n และ u มีค่าเป็น 1182, 3, 3 และ 115 ตามลำดับ
- สำหรับอัลกอริทึมอีเอ็ม กำหนดให้ค่าพารามิเตอร์เริ่มต้น θ_1 กำหนดให้มีค่าเป็น 0.7 จากนั้นใช้ตัวแยกแยะตัดคำ กำหนดฉลากให้กับตัวอย่างสอนของอัลกอริทึมอีเอ็ม หลังจากนั้นใช้อัลกอริทึมอีเอ็มด้วยตัวแยกแยะเบย์อย่างง่ายเพื่อเรียนรู้จนลู่เข้า

[1] http://www.yahoo.co.jp
[2] http://www.sanook.com, http://www.pantip.com
[3] http://www.javasoft.com

ผลที่ได้

ในการวัดประสิทธิภาพของวิธีการที่ทดสอบ เราใช้ความแม่นยำ (precision : P) การค้นคืน (recall : R) และ ตัววัด F_1 (F1-measure : F_1) ซึ่งนิยามดังนี้

$$P = \frac{\text{จำนวนตัวอย่างบวกที่ทำนายได้อย่างถูกต้อง}}{\text{จำนวนตัวอย่างที่ทำนายว่าเป็นบวก}} \quad (11)$$

$$R = \frac{\text{จำนวนตัวอย่างบวกที่ทำนายได้อย่างถูกต้อง}}{\text{จำนวนตัวอย่างทั้งหมด}} \quad (12)$$

$$F_1 = \frac{2PR}{P+R} \quad (13)$$

ผลการทดลองแสดงในตารางที่ 4 ในตารางนี้ “CoTraining (Bayes)” และ “CoTraining (Word)” เป็นผลการทดลองของตัวแยกแยะเบย์อย่างง่ายและตัวแยกแยะตัดคำของ CoTraining ตามลำดับ ส่วน “ICT (Bayes)” และ “ICT (Word)” เป็นตัวแยกแยะเบย์อย่างง่ายและตัวแยกแยะตัดคำของอัลกอริทึมการสอนไขว้แบบวนซ้ำ ตามลำดับ ส่วน U-Bayes-EM S-Bayes และ S-Word เป็นอัลกอริทึมอีเอ็ม ตัวแยกแยะเบย์อย่างง่ายแบบสอน และตัวแยกแยะตัดคำแบบสอน ตามลำดับ

ตารางที่ 5 ผลการเปรียบเทียบอัลกอริทึมสำหรับปัญหาการแยกแยะเว็บเพจภาษาไทยและไม่ใช่

ตัวแยกแยะ	P (%)	R (%)	F_1
ICT(Word)	99.78	100.00	99.89
S-Bayes	100.00	99.00	99.50
ICT(Bayes)	100.00	98.89	99.44
CoTraining(Bayes)	100.00	98.89	99.44
U-Bayes-EM	100.00	98.78	99.39
S-Word	99.08	99.61	99.34
CoTraining(Word)	100.00	98.66	99.33

ดังแสดงในตารางที่ 5 ICT(Word) มีประสิทธิภาพสูงสุดตามค่า F_1 ตามมาด้วย S-Bayes ตัวแยกแยะ ICT(Bayes) มีประสิทธิภาพใกล้เคียงกับ CoTraining(Bayes) ส่วน S-Word และ CoTraining(Word) มีประสิทธิภาพต่ำกว่าวิธีการอื่น ผลการทดลองแสดงให้เห็นว่า ICT ทำงานได้อย่างดีมีประสิทธิภาพทัดเทียมหรือมากกว่า S-Bayes ซึ่งเป็นอัลกอริทึมที่ใช้ตัวอย่างมีฉลากทั้งหมด และทำงานได้ดีกว่า S-Word มาก ผลการทดลองแสดงให้เห็นถึงประสิทธิภาพของ ICT ในการใช้ประโยชน์จากข้อมูลไม่มีฉลากได้อย่างดี

2.3 การจำแนกประเภทเว็บเพจออกเป็นหมวดหมู่

หัวข้อนี้กล่าวถึงการจำแนกประเภทเว็บเพจออกเป็นหมวดหมู่ โดยจะกล่าวถึงลักษณะสำคัญที่นำมาใช้ในการจำแนกเว็บเพจ ชุดข้อมูลที่นำมาทดลอง และผลการทดลองตามลำดับ

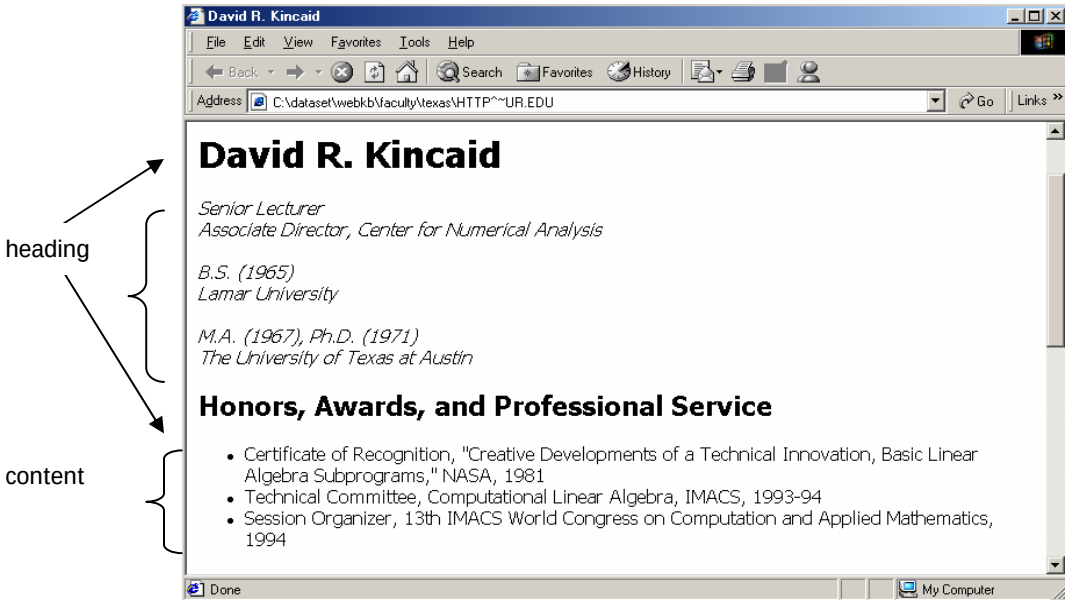
2.3.1 เซตลักษณะสำคัญ (feature set)

ในปัญหาการจำแนกเว็บเพจนั้น ประสิทธิภาพของตัวแยกแยะจะขึ้นกับเทคนิคที่ใช้ในการเรียนรู้และเซตของลักษณะสำคัญที่ใช้แทนตัวอย่างสอน เซตลักษณะสำคัญที่เหมาะสมจะช่วยให้ความถูกต้องของตัวแยกแยะสูง เซตลักษณะสำคัญที่เราเลือกใช้ในงานวิจัยนี้คือ เซตของคำที่ปรากฏในหัวข้อ (heading) ของเว็บเพจและเซตของคำที่ปรากฏในเนื้อหา (content) ของเว็บเพจ

- **เนื้อหา:** เนื้อหาของเว็บเพจให้ข้อมูลและรายละเอียดของเว็บเพจกับผู้ใช้เว็บเพจ ตัวอย่างเช่นเว็บเพจในรูปที่ 5 มีหัวข้อหนึ่งเป็น “Honors, Awards, and Professional Service” และมีเนื้อหาเป็นรายการของกิจกรรมอย่างละเอียด เราดึงเอาคำที่ปรากฏในเนื้อหาเพื่อใช้เป็นเซตของลักษณะสำคัญ ซึ่งเซตของคำในเนื้อหานี้จะใช้สำหรับสอน *Classifier2* ของอัลกอริทึมการสอนไขว้แบบวนซ้ำ
- **หัวข้อ:** สมมติฐานของเราคือคำที่อยู่ในหัวข้อจะแสดงไอเดียหลักของเนื้อหาที่ตามมา เราจึงดึงเอาคำที่ปรากฏในหัวข้อเพื่อใช้เป็นเซตของลักษณะสำคัญอีกเซตหนึ่ง และเซตของคำในหัวข้อนี้จะใช้สำหรับสอน *Classifier1* ของอัลกอริทึมการสอนไขว้แบบวนซ้ำ
- **หัวข้อและเนื้อหา:** หลังจากที่อัลกอริทึมการสอนไขว้แบบวนซ้ำเรียนรู้จบแล้ว ตัวแยกแยะผสม (combined classifier) จะทำนายฉลากหรือกลุ่ม (class) ของตัวอย่างโดยรวมเอ๊าท์พุตจาก *Classifer1* ที่เรียนรู้จากหัวข้อ และ *Classifier2* ที่เรียนรู้จากเนื้อหา สมมติให้ $L = \{l_1, l_2, \ldots, l_m\}$ เป็นเซตของฉลาก ตัวแยกแยะผสมทำนายตัวอย่างหนึ่งๆ ว่ามีฉลากเป็น l_j ตามค่าความน่าจะเป็น $Pr(l_j \mid d_i)$ ในสมการด้านล่างนี้

$$Pr(l_j \mid d_i) = Pr(l_j \mid x_1)\, Pr(l_j \mid x_2) \qquad (14)$$

โดยที่ x_1, x_2 เป็นเซตลักษณะสำคัญจากคำในหัวข้อและคำในเนื้อหาของเอกสาร d_i ส่วน $Pr(l_j \mid x_1)$ และ $Pr(l_j \mid x_2)$ เป็นค่าความน่าจะเป็นที่เอกสารนั้นจะเป็นฉลาก l_j ที่ทำนายโดย *Classifier1* และ *Classifier2* ตามลำดับ



รูปที่ 5 หัวข้อและเนื้อหาของเว็บเพจ

2.3.2 ตัวแยกแยะย่อยของอัลกอริทึมการสอนไขว้แบบวนซ้ำและอัลกอริทึมที่นำมาเปรียบเทียบ

ตัวแยกแยะย่อยของอัลกอริทึมการสอนไขว้แบบวนซ้ำ

ตัวแยกแยะย่อยของอัลกอริทึมการสอนไขว้แบบวนซ้ำในกรณีนี้เหมือนกับตัวแยกแยะเบย์อย่างง่ายที่ได้อธิบายไว้ในหัวข้อที่ 2.2.1 เกือบทั้งหมด ต่างกันตรงที่ในกรณีนี้เราใช้ “*ถุงคำ (bag-of-words)*” เพื่อแทนข้อความในการทดลองในหัวข้อ 2.3.5 เราใช้ตัวแยกแยะเบย์อย่างง่ายเป็น *Classifier1* และ *Classifier2* ในตารางที่ 2

ซึ่งจะเรียนรู้จากเซตลักษณะสำคัญที่ต่างกัน โดย *Classifier1* จะเรียนจากเซตของคำในหัวข้อ ส่วน *Classifier2* จะเรียนจากเซตของคำในเนื้อหา

อัลกอริทึมที่นำมาเปรียบเทียบในการทดลอง

ในการทดลองเพื่อประเมินประสิทธิภาพของการสอนไขว้แบบวนซ้ำในปัญหาการจำแนกประเภทเวบเพจออกเป็นหมวดหมู่นั้น เราได้เปรียบเทียบกับวิธีการต่อไปนี้

(1) ตัวแยกแยะเบย์อย่างง่ายแบบสอน (Supervised Naïve Bayes Classifier – S-Bayes)

(2) อัลกอริทึมโคเทรนนิง (CoTraining Algorithm) และ

(3) อัลกอริทึมอีเอ็ม (EM Algorithm)

อัลกอริทึมโคเทรนนิงและอัลกอริทึมอีเอ็มได้กล่าวแล้วในหัวข้อที่ 2.2.2 แต่ในกรณีนี้เราใช้ "*ถุงคำ (bag-of-words)*" เพื่อแทนข้อความ

2.3.3 ชุดข้อมูลที่ใช้ในการทดลอง

เราได้นำการสอนไขว้แบบวนซ้ำไปใช้กับปัญหาการจำแนกเว็บเพจแบบหลายกลุ่ม (multi-class Web page categorization) โดยทำการทดลองกับชุดข้อมูลทั้งหมด 3 ชุดคือ

(1) ชุดข้อมูล WebKb

(2) ชุดข้อมูล WebClass และ

(3) ชุดข้อมูล DrugUsage

- ชุดข้อมูล WebKb ประกอบด้วยเว็บเพจที่เกี่ยวกับมหาวิทยาลัย ชุดข้อมูลนี้ได้จาก ftp จากมหาวิทยาลัย Carnegie Mellon University [WebKB] ชุดข้อมูลนี้ประกอบด้วยเว็บเพจทั้งหมด 981 เพจ รวบรวมจากเว็บไซต์ของ computer science department ของมหาวิทยาลัย 4 แห่ง: Cornell University, University of Washington, University of Wisconsin, และ University of Texas เว็บเพจทั้งหมดนี้ถูกติดฉลากโดยใช้แรงงานคน และถูกแบ่งออกเป็น 4 กลุ่มคือ course homepages, faculty homepages, project homepages และ student homepages มีจำนวนเว็บเพจในแต่ละกลุ่มเท่ากับ 220 เพจ 147 เพจ 81 เพจ และ 533 เพจตามลำดับ
- ชุดข้อมูล WebClass ได้มาจาก machine learning research group ในประเทศ Italy [WebClass] ชุดข้อมูลนี้เกี่ยวข้องกับงานอดิเรก ประกอบด้วยเว็บเพจจำนวน 192 เพจ แบ่งออกเป็น 4 กลุ่มคือ astronomy, jazz, auto และ motorcycle แต่ละกลุ่มมีเว็บเพจจำนวน 48 เพจ ข้อมูลในสองกลุ่มแรกค่อนข้างแตกต่างกัน ส่วนข้อมูลในสองกลุ่มหลังคือ auto และ motorcycle ค่อนข้างคล้ายกัน
- ชุดข้อมูล DrugUsage ได้มาจาก research group at Sirindhorn International Institute of Technology [DrugUsage] ชุดข้อมูลนี้ประกอบด้วยเว็บเพจจำนวน 353 เพจ แบ่งออกเป็น 5 กลุ่มในโดเมนของยา กลุ่มข้อมูลทั้งห้าคือ adverse, clinical pharmacology, overdose, patient information และ warning เว็บเพจในกลุ่ม adverse จะอธิบายเกี่ยวกับผลข้างเคียงของยา เว็บเพจในกลุ่ม clinical pharmacology จะอธิบายเกี่ยวกับการใช้ยา เว็บเพจในกลุ่ม overdose จะให้ข้อมูลเกี่ยวกับอาการผู้ป่วยที่ได้รับยาเกินปกติ เว็บเพจในกลุ่ม patient information จะเกี่ยวข้องกับข่าวสารข้อมูลสำหรับผู้ป่วยเกี่ยวกับการใช้ยา ส่วนเว็บเพจในกลุ่ม warning อธิบายคำเตือนของยาให้กับผู้ป่วย

2.3.4 การประมวลผลล่วงหน้า

ก่อนเริ่มใช้อัลกอริทึมทำการเรียนรู้นั้น เราจะทำการประมวลผลล่วงหน้าสำหรับเว็บเพจทุกเพจ โดยการลบ html tag, ตัดคำหยุด (stop word) และทำการหารากคำ (word stemming) ดังนี้

- ตัดคำหยุด: เราลบคำที่ไม่มีความสำคัญในการจำแนกเว็บเพจออกจากเพจ คำที่ถูกลบได้แก่ auxiliary verb, preposition, pronoun, possessive pronoun, phone number, digit sequence, date และ special character
- หารากคำ: คำหลายคำมีรากคำตัวเดียวกัน เช่น "teached", "teaching", "teach", "teaches", และ "teacher" ต่างก็มีรากคำเดียวกันคือ "teach" โดยการนำ word stemming มาใช้ คำทั้งห้าคำข้างต้นจะถูกแทนที่ด้วย "teach" ในการทดลองด้านล่างนี้เราใช้อัลกอริทึม Porter Stemming [Porter 1980] สำหรับการหารากคำ

2.3.5 ผลการทดลองเปรียบเทียบกับอัลกอริทึมอื่นในการจำแนกประเภทเว็บเพจออกเป็นหมวดหมู่

เราได้ทำการทดลองเพื่อวัดประสิทธิภาพของการสอนไขว้แบบวนซ้ำ โดยเปรียบเทียบอัลกอริทึมการสอนไขว้แบบวนซ้ำ (ICT) กับ อัลกอริทึมเบย์อย่างง่ายแบบสอน (Supervised Naïve Bayes – S-Bayes) ที่มีการใช้งานอย่างกว้างขวางในปัญหาการจำแนกข้อความ แต่อัลกอริทึมเบย์อย่างง่ายแบบสอนเป็นอัลกอริทึมที่ต้องอาศัยข้อมูลแบบมีฉลากทั้งหมดจึงจะทำงานได้ และอัลกอริทึมที่นำมาเปรียบเทียบอีก 2 ตัวคือ อัลกอริทึมโคเทรนนิง (CoTraining) และอัลกอริทึมอีเอ็ม (EM-algorithm) ซึ่งสามารถใช้ข้อมูลแบบไม่มีฉลากร่วมกับข้อมูลที่มีฉลากเริ่มต้นจำนวนน้อยๆ ได้เช่นกัน ในการวัดประสิทธิภาพของอัลกอริทึมนั้น เราใช้ความแม่นยำ (precision : P) การคืนคืน (recall : R) และ ตัววัด F_1 (F_1-measure) (เหมือนที่นิยามไว้ในหัวข้อที่ 2.2.3)

$$P = \frac{\text{จำนวนตัวอย่างบวกที่ทำนายได้อย่างถูกต้อง}}{\text{จำนวนตัวอย่างที่ทำนายว่าเป็นบวก}}$$

$$R = \frac{\text{จำนวนตัวอย่างบวกที่ทำนายได้อย่างถูกต้อง}}{\text{จำนวนตัวอย่างทั้งหมด}}$$

$$F_1 = \frac{2PR}{P+R}$$

ผลการทดลองสำหรับชุดข้อมูล WebKb

การตั้งค่าต่างๆ ของการทดลองเป็นดังต่อไปนี้

- ในกรณีของอัลกอริทึมการสอนไขว้แบบวนซ้ำ เราเลือกตัวอย่างจำนวน 30% ของตัวอย่างทั้งหมดแบบสุ่มสำหรับแต่ละกลุ่ม เพื่อใช้เป็นชุดตัวอย่างสอนมีฉลากเริ่มต้น และใช้ 30% ของตัวอย่างทั้งหมดเป็นชุดตัวอย่างสอนไม่มีฉลาก ที่เหลือ 40% เป็นตัวอย่างทดสอบ
- ในกรณีของอัลกอริทึมโคเทรนนิง เราใช้ตัวอย่างสอนมีฉลากเริ่มต้นชุดเดียวกับของอัลกอริทึมการสอนไขว้แบบวนซ้ำ นอกจากนั้นชุดตัวอย่างสอนมีฉลากและชุดทดสอบก็เหมือนกับของอัลกอริทึมการสอนไขว้แบบวนซ้ำ พารามิเตอร์ p และ n ในตารางที่ 3 ถูกตั้งค่าให้มีค่าเท่ากับ 1 และ 3 ตามลำดับ
- ในกรณีของอัลกอริทึมเบย์อย่างง่ายแบบสอน เราใช้ตัวอย่างจำนวน 60% ของตัวอย่างทั้งหมดเป็นชุดตัวอย่างสอนมีฉลาก ส่วนที่เหลือ 40% เป็นตัวอย่างทดสอบ
- ในกรณีของอัลกอริทึมอีเอ็ม ชุดตัวอย่างสอนมีฉลากเริ่มต้น ชุดตัวอย่างสอนไม่มีฉลากและชุดตัวอย่างทดสอบเหมือนกันกับของอัลกอริทึมการสอนไขว้แบบวนซ้ำ

ผลการทดลองเปรียบเทียบอัลกอริทึมการสอนไขว้แบบวนซ้ำกับอัลกอริทึมอื่นๆ สำหรับชุดข้อมูล WebKb แสดงในตารางที่ 6 – ตารางที่ 9 ในการทดลอง เราใช้คำที่อยู่ในหัวข้อ (heading) ในเว็บเพจสำหรับสอน *Classifier1* และใช้คำที่อยู่ในเนื้อหา (content) ในเว็บเพจสำหรับสอน *Classifier2* ของอัลกอริทึมการสอนไขว้แบบวนซ้ำ ผลที่ได้สำหรับ *Classifier1* และ *Classifier2* แสดงด้วย Heading-based Classifier และ Content-based Classifier ตามลำดับ ส่วน Heading+content-based Classifer เป็นตัวแยกแยะที่รวมผลการแยกแยะจาก *Classifier1* และ *Classifer2* ในทำนองเดียวกันเราใช้คำที่อยู่ในหัวข้อและเนื้อหาเพื่อสอนตัวแยกแยะทั้งสองตัวของโคเทรนนิง (โคเทรนนิงใช้ตัวแยกแยะ 2 ตัวเช่นกัน) ส่วนอัลกอริทึมเบย์อย่างง่ายแบบสอนและอัลกอริทึมอีเอ็มนั้นใช้ตัวแยกแยะตัวเดียวในการเรียนรู้ ดังนั้น Heading-based Classifier และ Content-based Classifier สำหรับอัลกอริทึมทั้งสอง คือตัวแยกแยะที่ได้จากการสอนอัลกอริทึมด้วยคำในหัวข้อและคำในเนื้อหาแยกกันตามลำดับ ส่วน Heading+content-based Classifer เป็นตัวแยกแยะที่รวมผลของตัวแยกแยะเดี่ยวๆ 2 ตัวเช่นเดียวกับกรณีของการสอนไขว้แบบวนซ้ำ

ผลการทดลองในตารางที่ 6 – ตารางที่ 9 แสดงให้เห็นว่าประสิทธิภาพวัดโดยค่าเฉลี่ย F_1 ของ Heading-based Classifier ของอัลกอริทึมการสอนไขว้แบบวนซ้ำมีค่าเท่ากับ 78.25% ซึ่งสูงกว่าของอัลกอริทึมโคเทรน-นิงและอีเอ็ม ในทำนองเดียวกัน ค่าเฉลี่ย F_1 ของ Content-based Classifier ของอัลกอริทึมการสอนไขว้แบบวนซ้ำก็มีค่าสูงกว่าของอัลกอริทึมโคเทรนนิงและอีเอ็ม อย่างไรก็ดีประสิทธิภาพของการสอนไขว้แบบวนซ้ำต่ำกว่าของอัลกอริทึมเบย์อย่างง่ายแบบสอนเล็กน้อย สาเหตุที่เป็นเช่นนี้เนื่องจากอัลกอริทึมการสอนไขว้แบบวนซ้ำได้รับตัวอย่างมีฉลากเพียงครึ่งเดียวของตัวอย่างมีฉลากที่ใช้สอนอัลกอริทึมเบย์อย่างง่ายแบบสอนส่วนตัวแยกแยะที่ใช้เซตลักษณะสำคัญทั้งสองร่วมกัน (Heading+content-based Classifier) ของอัลกอริทึมการสอนไขว้แบบวนซ้ำก็มีประสิทธิภาพสูงกว่าของอัลกอริทึมโคเทรนนิงและอีเอ็ม นอกจากนั้นเราพบว่าอัลกอริทึมการสอนไขว้แบบวนซ้ำใช้เวลาในการเรียนรู้น้อยกว่าของอัลกอริทึมโคเทรนนิงและอีเอ็ม โดยใช้เวลาประมาณ 3 นาที ส่วนอัลกอริทึมโคเทรนนิงและอีเอ็มใช้เวลามากกว่า 20 นาที ทุกอัลกอริทึมเขียนด้วยโปรแกรมภาษาจาวารันบนระบบปฏิบัติการไมโครซอฟต์วินโดวส์

ผลการทดลองสำหรับชุดข้อมูล WebClass และ DrugUsage

การตั้งค่าต่างๆ ของการทดลองสำหรับ WebClass และ DrugUsage เป็นดังต่อไปนี้

- ในกรณีของอัลกอริทึมการสอนไขว้แบบวนซ้ำ อัลกอริทึมโคเทรนนิง และอัลกอริทึมอีเอ็ม เราเลือกตัวอย่างจำนวน 33% ของตัวอย่างทั้งหมดแบบสุ่มสำหรับแต่ละกลุ่ม เพื่อใช้เป็นชุดตัวอย่างสอนมีฉลากเริ่มต้น และใช้ 33% ของตัวอย่างทั้งหมดเป็นชุดตัวอย่างสอนไม่มีฉลาก ที่เหลือ 34% เป็นตัวอย่างทดสอบ
- ในกรณีของอัลกอริทึมเบย์อย่างง่ายแบบสอน เราใช้ตัวอย่างจำนวน 66% ของตัวอย่างทั้งหมดเป็นชุดตัวอย่างสอนมีฉลาก ส่วนที่เหลือ 34% เป็นตัวอย่างทดสอบ

ผลการทดลองเปรียบเทียบอัลกอริทึมการสอนไขว้แบบวนซ้ำกับอัลกอริทึมอื่นๆ สำหรับชุดข้อมูล WebClass แสดงในตารางที่ 10 – ตารางที่ 13 ส่วนผลการทดลองสำหรับชุดข้อมูล DrugUsage แสดงในตารางที่ 14 – ตารางที่ 17 ผลการทดลองที่ได้สำหรับชุดข้อมูลทั้งสองนี้ ให้ผลในทำนองเดียวกันกับชุดข้อมูล WebKb คือ อัลกอริทึมการสอนไขว้แบบวนซ้ำมีประสิทธิภาพสูงกว่าอัลกอริทึมโคเทรนนิงและอัลกอริทึมอีเอ็ม แต่ให้ผลด้อยกว่าอัลกอริทึมเบย์อย่างง่ายแบบสอน และพบว่าประสิทธิภาพของทุกอัลกอริทึมมีค่าน้อยลงในชุดข้อมูล DrugUsage ที่เป็นเช่นนี้เนื่องจากในชุดข้อมูลนี้เว็บเพจในกลุ่มต่างๆ มีเนื้อหาที่เกี่ยวข้องกับยาทุกกลุ่ม ทำให้เนื้อหาใกล้เคียงกันกว่าชุดข้อมูลอื่นๆ ทำให้การจำแนกกลุ่มทำได้ยากขึ้น และในชุดข้อมูลนี้มีจำนวนกลุ่ม 5 กลุ่มซึ่งมากกว่าในชุดข้อมูลอื่นๆ และก็เป็นสาเหตุหนึ่งที่ทำให้การจำแนกกลุ่มยากมากขึ้นอีก

ตารางที่ 6 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมการสอนไขว้แบบวนซ้ำสำหรับชุดข้อมูล WebKb

ICT	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Course	67.69	100.00	80.73	66.15	97.73	78.90	85.71	95.45	90.32
Faculty	24.79	96.67	39.46	22.39	100.00	36.59	20.98	100.00	34.68
Project	35.00	87.50	50.06	23.26	62.50	33.90	27.03	62.50	37.74
Student	92.45	92.45	92.45	87.00	82.08	84.47	90.10	85.85	87.92
Average	71.85	94.39	78.25	67.23	86.73	71.76	73.39	88.27	76.22

ตารางที่ 7 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมเบย์อย่างง่ายแบบสอนสำหรับชุดข้อมูล WebKb

S-Bayes	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Course	69.35	97.73	81.13	88.89	90.91	89.89	86.09	92.27	89.03
Faculty	39.22	68.97	50.00	37.93	75.86	50.57	42.84	75.49	54.58
Project	50.00	62.50	55.56	47.37	56.25	51.43	49.35	55.00	51.33
Student	90.74	91.59	91.16	87.04	87.85	87.44	90.63	86.90	88.63
Average	74.99	87.24	79.91	76.95	84.18	79.60	78.92	83.76	80.46

ตารางที่ 8 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมโคเทรนนิงสำหรับชุดข้อมูล WebKb

Co Training	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Course	68.25	97.73	80.37	89.74	79.55	84.34	85.71	95.45	90.32
Faculty	40.63	86.67	55.32	51.52	56.67	53.97	32.88	80.00	46.60
Project	24.14	87.50	37.84	25.53	75.00	38.09	37.50	18.75	25.06
Student	93.26	78.30	85.13	91.67	51.89	66.27	75.74	96.26	84.77
Average	73.95	84.69	75.64	79.69	60.72	66.14	68.29	87.26	75.30

ตารางที่ 9 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมอีเอ็มสำหรับชุดข้อมูล WebKb

EM	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Course	63.19	98.64	77.00	83.00	80.91	81.33	84.63	80.45	81.68
Faculty	26.51	90.44	40.94	35.09	89.03	48.16	28.10	97.26	42.91
Project	39.22	75.00	51.26	37.38	57.50	41.70	24.41	87.50	37.19
Student	87.23	91.58	89.20	91.95	69.89	77.05	91.68	71.02	77.91
Average	68.62	91.64	75.98	76.78	74.28	70.70	74.87	78.50	70.08

ตารางที่ 10 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมการสอนไขว้แบบวนซ้ำสำหรับชุดข้อมูล WebClass

ICT	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Astro	93.33	100.00	96.55	100.00	92.86	96.30	100.00	92.86	96.30
Auto	60.87	100.00	75.68	93.33	100.00	96.55	76.47	92.86	83.87
Jazz	100.00	64.29	78.26	93.33	100.00	96.55	100.00	100.00	100.00
Motor	91.67	78.57	84.62	93.33	100.00	96.55	93.33	100.00	96.55
Average	86.47	85.72	83.78	95.00	98.22	96.49	92.45	96.43	94.18

ตารางที่ 11 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมเบย์อย่างง่ายแบบสอนสำหรับชุดข้อมูล WebClass

S-Bayes	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Astro	87.97	83.33	84.08	100.00	92.86	96.20	100.00	95.24	97.53
Auto	74.22	97.62	83.98	93.61	97.62	95.39	95.56	97.62	96.47
Jazz	100.00	66.67	79.34	97.78	100.00	98.85	97.78	100.00	98.85
Motor	75.23	92.86	81.27	81.18	100.00	89.50	82.54	100.00	90.39
Average	84.36	85.12	82.17	93.14	97.62	94.99	93.97	98.21	95.81

ตารางที่ 12 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมโคเทรนนิงสำหรับชุดข้อมูล WebClass

Co Training	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Astro	65.00	92.86	76.47	100.00	92.86	96.30	100.00	95.24	97.53
Auto	77.78	100.00	87.50	77.78	100.00	87.50	87.71	97.62	92.32
Jazz	100.00	78.57	88.04	93.33	100.00	96.55	87.78	100.00	92.97
Motor	53.85	100.00	70.04	60.87	100.00	75.68	75.00	95.24	82.91
Average	74.16	92.86	80.51	83.00	98.22	89.01	87.62	97.02	91.43

ตารางที่ 13 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมอีเอ็มสำหรับชุดข้อมูล WebClass

EM	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Astro	83.45	97.62	89.51	100.00	95.24	97.53	100.00	97.62	98.77
Auto	46.13	50.00	47.94	84.26	100.00	91.39	77.31	100.00	86.94
Jazz	62.86	78.57	65.83	91.39	100.00	95.48	95.83	100.00	97.78
Motor	80.00	52.38	52.72	74.73	100.00	84.86	76.11	100.00	85.35
Average	68.11	69.64	64.00	87.60	98.81	92.31	87.31	99.40	92.21

ตารางที่ 14 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมการสอนไขว้แบบวนซ้ำสำหรับชุดข้อมูล DrugUsage

ICT	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
Category	P	R	F_1	P	R	F_1	P	R	F_1
Adverse	72.30	98.55	83.41	57.56	96.71	72.17	69.70	95.83	80.70
Clinical	83.33	50.72	63.06	88.28	85.82	87.03	66.67	91.67	77.20
Overdose	44.86	92.99	60.53	44.10	48.19	46.05	34.62	75.00	47.37
Patient	38.00	69.44	49.12	48.53	29.17	36.44	46.43	54.17	50.00
Warning	64.21	91.61	75.50	47.24	91.61	62.33	54.76	95.83	69.69
Average	60.54	80.66	69.17	57.14	70.30	63.04	54.44	82.50	64.99

ตารางที่ 15 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมเบย์อย่างง่ายแบบสอนสำหรับชุดข้อมูล DrugUsage

S-Bayes	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Adverse	97.10	94.32	95.69	82.70	95.71	88.73	82.14	100.00	90.19
Clinical	80.86	84.42	82.60	91.54	88.71	90.10	90.91	83.33	86.96
Overdose	65.59	95.71	77.84	51.84	88.77	65.45	51.16	91.67	65.67
Patient	64.38	95.83	77.02	67.73	70.83	69.25	75.00	87.50	80.77
Warning	70.76	93.06	80.39	50.24	91.61	64.89	53.66	95.65	68.75
Average	75.74	92.67	83.35	68.81	87.12	76.89	70.57	91.63	78.47

ตารางที่ 16 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมโคเทรนนิงสำหรับชุดข้อมูล DrugUsage

Co Training	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Adverse	78.71	88.71	83.41	59.42	91.36	72.01	67.74	91.30	77.77
Clinical	73.15	59.18	65.43	77.10	83.03	79.95	86.36	79.17	82.61
Overdose	42.40	59.18	49.40	37.00	75.91	49.75	38.89	87.50	53.85
Patient	18.06	12.50	14.77	34.50	43.06	38.30	31.25	41.67	35.72
Warning	65.21	93.06	76.69	44.24	94.44	60.25	38.33	100.00	55.42
Average	55.51	62.52	58.81	50.45	77.56	61.14	52.51	79.93	61.07

ตารางที่ 17 ประสิทธิภาพของตัวแยกแยะในอัลกอริทึมอีเอ็มสำหรับชุดข้อมูล DrugUsage

EM	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
category	P	R	F_1	P	R	F_1	P	R	F_1
Adverse	88.59	67.75	76.78	69.28	98.55	81.36	70.59	100.00	82.76
Clinical	70.61	33.88	45.79	67.23	83.15	74.35	68.97	83.33	75.47
Overdose	16.40	11.17	13.29	24.11	56.58	33.82	19.35	25.00	21.82
Patient	17.06	31.94	22.25	37.62	48.61	42.42	27.42	70.83	39.53
Warning	72.41	87.50	79.25	32.05	95.83	48.04	33.33	95.83	49.46
Average	53.02	46.45	49.52	46.06	76.55	57.51	43.93	75.00	55.41

ตารางที่ 18 – ตารางที่ 20 สรุปผลการทดลองเปรียบเทียบอัลกอริทึมการสอนไขว้แบบวนซ้ำกับอัลกอริทึมอื่นๆ สำหรับชุดข้อมูล WebKb, WebClass และ DrugUsage ตามลำดับ เราสามารถสรุปผลจากการทดลองได้ดังนี้คือ ประสิทธิภาพของอัลกอริทึมการสอนไขว้แบบวนซ้ำสูงกว่าโคเทรนนิงและอีเอ็มสำหรับทุกชุดข้อมูล ซึ่งแสดงว่าวิธีการที่นำเสนอสามารถใช้ประโยชน์จากข้อมูลที่ไม่มีฉลากได้อย่างมีประสิทธิภาพ อย่างไรก็ดี ประสิทธิภาพของการสอนไขว้แบบวนซ้ำต่ำกว่าของอัลกอริทึมเบย์อย่างง่ายแบบสอน สาเหตุที่เป็นเช่นนี้เนื่องจากตัวอย่างที่ใช้สอนอัลกอริทึมเบย์อย่างง่ายแบบสอนเป็นตัวอย่างที่ติดฉลากทั้งหมด ซึ่งต่างจากตัวอย่างที่ใช้สอนการสอนไขว้แบบวนซ้ำเป็นตัวอย่างที่ไม่มีฉลากเป็นส่วนใหญ่ ผลที่ได้โดยรวมแสดงให้เห็นว่า การสอนไขว้แบบวนซ้ำสามารถใช้งานได้อย่างดีสำหรับการจำแนกประเภทเว็บเพจ โดยรับข้อมูลที่มีฉลากเพียงเล็กน้อย สามารถใช้ประโยชน์จากตัวอย่างที่ไม่มีฉลากได้อย่างมีประสิทธิภาพ และให้ความถูกต้องสูงกว่าอัลกอริทึมที่รับข้อมูลไม่มีฉลากตัวอื่นๆ

ตารางที่ 18 ผลการทดลองเปรียบเทียบอัลกอริทึมต่างๆสำหรับชุดข้อมูล WebKb

	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
	P	R	F_1	P	R	F_1	P	R	F_1
ICT	71.85	94.39	78.25	67.23	86.73	71.76	73.39	88.27	76.22
S-Bayes	74.99	87.24	79.91	76.95	84.18	79.60	78.92	83.76	80.46
CoTraining	73.95	84.69	75.64	79.69	60.72	66.14	68.29	87.26	75.30
EM	68.62	91.64	75.98	76.78	74.28	70.70	74.87	78.50	70.08

ตารางที่ 19 ผลการทดลองเปรียบเทียบอัลกอริทึมต่างๆสำหรับชุดข้อมูล WebClass

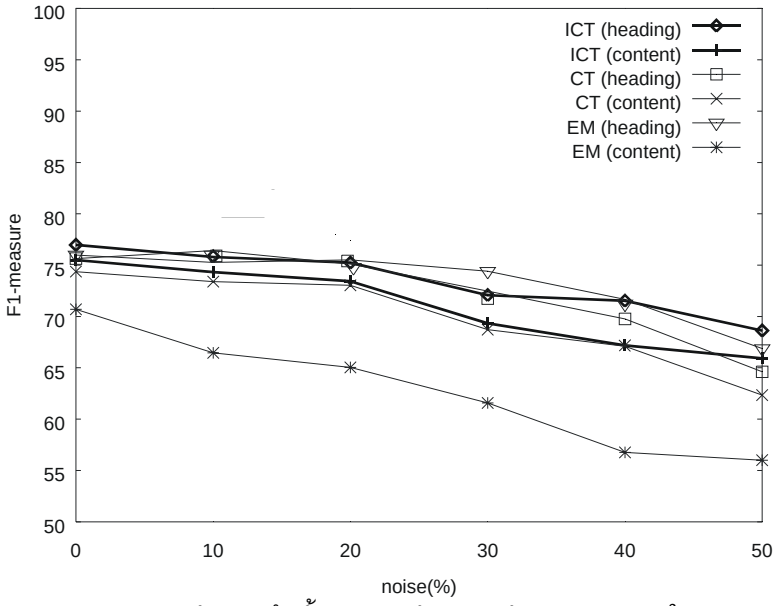
	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
	P	R	F_1	P	R	F_1	P	R	F_1
ICT	86.47	85.72	83.78	95.00	98.22	96.49	92.45	96.43	94.18
S-Bayes	84.36	85.12	82.17	93.14	97.62	94.99	93.97	98.21	95.81
CoTraining	74.16	92.86	80.51	83.00	98.22	89.01	87.62	97.02	91.43
EM	68.11	69.64	64.00	87.60	98.81	92.31	87.31	99.40	92.21

ตารางที่ 20 ผลการทดลองเปรียบเทียบอัลกอริทึมต่างๆสำหรับชุดข้อมูล DrugUsage

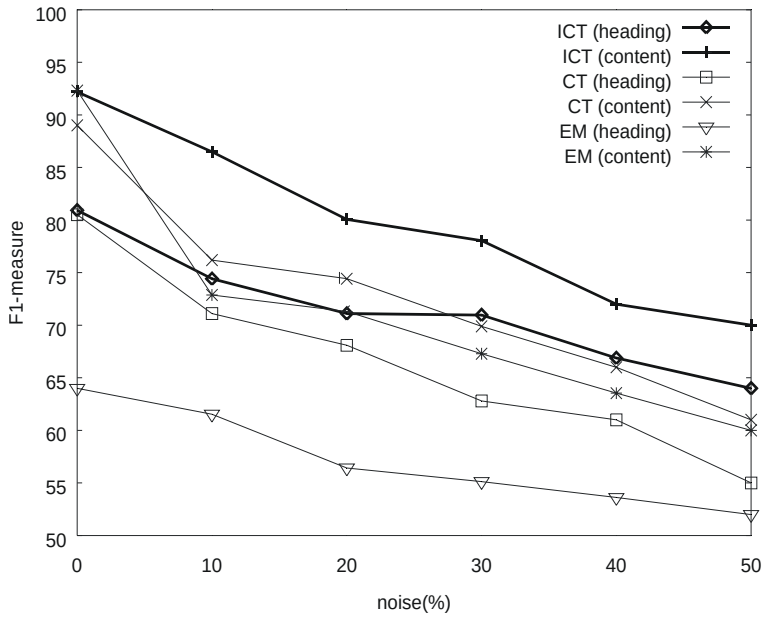
	Heading-based Classifier			Content-based Classifier			Heading+content-based Classifier		
	P	R	F_1	P	R	F_1	P	R	F_1
ICT	60.54	80.66	69.17	57.14	70.30	63.04	54.44	82.50	64.99
S-Bayes	75.74	92.67	83.35	68.81	87.12	76.89	70.57	91.63	78.47
CoTraining	55.51	62.52	58.81	50.45	77.56	61.14	52.51	79.93	61.07
EM	53.02	46.45	49.52	46.06	76.55	57.51	43.93	75.00	55.41

2.4 ผลกระทบของข้อมูลสัญญาณรบกวนในปัญหาการจำแนกเว็บเพจออกเป็นหมวดหมู่

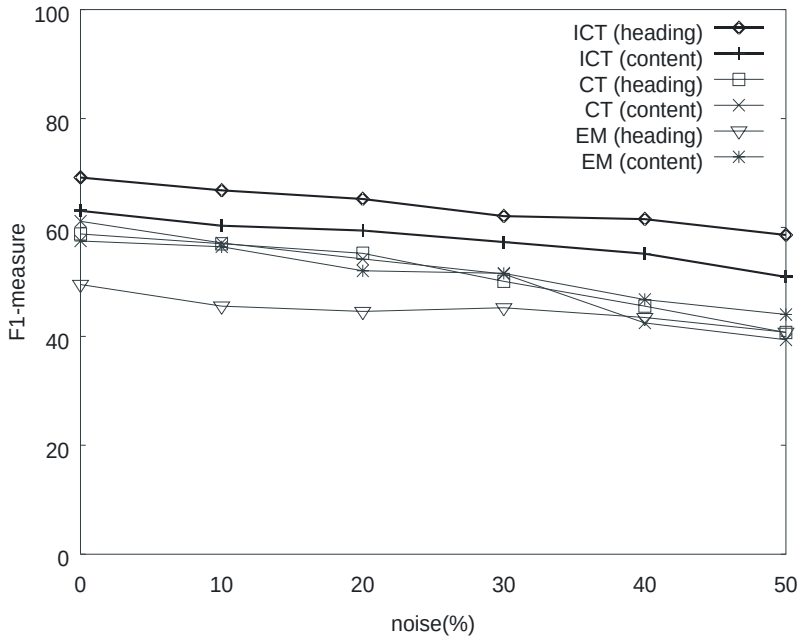
หัวข้อนี้ศึกษาผลกระทบของข้อมูลที่มีสัญญาณรบกวน (noisy data) ต่ออัลกอริทึมการสอนไขว้แบบวนซ้ำในการใช้งานจริง ข้อมูลสัญญาณรบกวนมักเกิดขึ้นบ่อยครั้ง อาจเนื่องมาจากการบันทึกข้อมูลที่ผิดพลาด ในกรณีของการจำแนกเว็บเพจอาจเกิดขึ้นจากความผิดพลาดของผู้สอนในการติดฉลากให้กับเว็บเพจ ดังนั้น เราจึงศึกษาว่าอัลกอริทึมจะมีประสิทธิภาพลดลงมากน้อยเพียงไรหรือมีความทนทานต่อข้อมูลสัญญาณรบกวนได้มากน้อยเท่าไร ในการทดลองด้านล่างนี้ เราเพิ่มสัญญาณรบกวนเข้าไปในข้อมูลมีฉลาก โดยการเปลี่ยนค่าของกลุ่มข้อมูลจากกลุ่มหนึ่งไปเป็นกลุ่มอื่นที่ไม่ถูกต้อง และให้สัดส่วนของข้อมูลที่มีสัญญาณรบกวนต่อข้อมูลที่มีฉลากทั้งหมดเป็น 10% ถึง 50% แล้วทำการทดลองกับชุดข้อมูลทั้งสามชุด อัลกอริทึมที่นำมาเปรียบเทียบทั้งหมดคือ อัลกอริทึมการสอนไขว้แบบวนซ้ำ อัลกอริทึมโคเทรนนิง และอัลกอริทึมอีเอ็ม ผลที่ได้แสดงในรูปที่ 6 – รูปที่ 8



รูปที่ 6 ประสิทธิภาพของอัลกอริทึมทั้งสามสำหรับข้อมูลสัญญาณรบกวนในชุดข้อมูล WebKb



รูปที่ 7 ประสิทธิภาพของอัลกอริทึมทั้งสามสำหรับข้อมูลสัญญาณรบกวนในชุดข้อมูล WebClass



รูปที่ 8 ประสิทธิภาพของอัลกอริทึมทั้งสามสำหรับข้อมูลสัญญาณรบกวนในชุดข้อมูล DrugUsage

ผลการทดลองในรูปที่ 6 – รูปที่ 8 แสดงประสิทธิภาพของอัลกอริทึมการสอนไขว้แบบวนซ้ำ (แสดงโดย ICT ในรูป) เปรียบเทียบกับอัลกอริทึมโคเทรนนิง (แสดงโดย CT ในรูป) และอัลกอริทึมอีเอ็ม (แสดงโดย EM ในรูป) ICT (heading) และ ICT (content) ในรูปแสดงผลที่ได้ของตัวแยกแยะย่อยที่ใช้เซตของคำในหัวข้อและตัวแยกแยะย่อยที่ใช้เซตของคำในเนื้อหาของอัลกอริทึมการสอนไขว้แบบวนซ้ำตามลำดับ ในทำนองเดียวกัน CT (heading) และ CT (content) เป็นตัวแยกแยะย่อยของอัลกอริทึมโคเทรนนิง ส่วน EM (heading) และ EM (content) เป็นตัวแยกแยะย่อยของอัลกอริทึมอีเอ็ม

ผลที่ได้สำหรับชุดข้อมูลทั้งสามมีลักษณะเดียวกัน คือ อัลกอริทึมการสอนไขว้แบบวนซ้ำมีความทนทานต่อข้อมูลสัญญาณรบกวนมากกว่าอัลกอริทึมอื่นๆ ซึ่งดูได้จากการลดลงของประสิทธิภาพของตัวแยกแยะในอัลกอริทึมการสอนไขว้แบบวนซ้ำที่ลดลงน้อยกว่าของอัลกอริทึมอื่นๆ ตัวอย่างเช่นในกรณีของชุดข้อมูล WebKb ในรูปที่ 6 เมื่อเราเพิ่มสัญญาณรบกวนถึง 50% ตัวแยกแยะที่ใช้คำในหัวข้อและตัวแยกแยะที่ใช้คำในเนื้อหาของอัลกอริทึมการสอนไขว้แบบวนซ้ำมีประสิทธิภาพลดลง 10.85% และ 12.70% ตามลำดับ ในขณะที่ตัวแยกแยะที่ใช้คำในหัวข้อและตัวแยกแยะที่ใช้คำในเนื้อหาของอัลกอริทึมโคเทรนนิง มีประสิทธิภาพลดลง 14.85% และ 16.17% ตามลำดับ ส่วนตัวแยกแยะที่ใช้คำในหัวข้อและตัวแยกแยะที่ใช้คำในเนื้อหาของอัลกอริทึมอีเอ็มมีประสิทธิภาพลดลง 11.99% และ 20.79% ตามลำดับ ส่วนผลสำหรับชุดข้อมูลอื่นๆ ก็ให้ผลในทำนองเดียวกัน สาเหตุหนึ่งที่อัลกอริทึมการสอนไขว้แบบวนซ้ำมีความทนทานต่อสัญญาณรบกวนมากกว่าอัลกอริทึมอื่น ก็เนื่องมาจากการที่อัลกอริทึมการสอนไขว้แบบวนซ้ำใช้การติดฉลากใหม่หมดทุกรอบการเรียนรู้จนกระทั่งการเรียนรู้ลู่เข้า ดังนั้นถ้าหากว่าในรอบหลังๆ ความถูกต้องของตัวแยกแยะมีมากขึ้น ก็จะทำให้ติดฉลากให้กับตัวอย่างไม่มีฉลากได้ถูกต้องมากขึ้นในทุกๆ รอบ แม้ว่าในรอบแรกๆ ผลของสัญญาณรบกวนอาจทำให้การติดฉลากผิดพลาด แต่ในรอบหลังๆ การติดฉลากก็สามารถทำได้ดีขึ้น ตัวอย่างเดิมที่เคยติดฉลากผิดในรอบแรกๆ ก็อาจถูกต้องได้ในรอบหลังๆ ซึ่งวิธีการนี้ต่างกับอัลกอริทึมโคเทรนนิงซึ่งใช้การติดฉลากข้อมูลไม่มีฉลากทีละน้อย และไม่มีการติดฉลากใหม่ ทำให้ผลของสัญญาณรบกวนส่งผลให้การติดฉลากผิดพลาดได้ และไม่สามารถแก้ไขได้

2.5 สรุปผลการวิจัยเทคนิคการเรียนรู้ของเครื่องและวิธีใช้ประโยชน์จากข้อมูลแบบไม่มีฉลาก

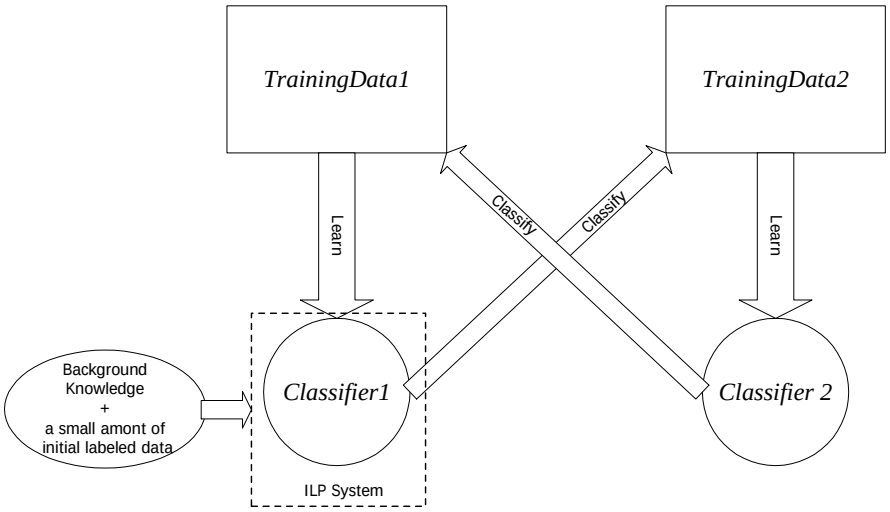
เราได้ใช้การสอนไขว้แบบวนซ้ำกับปัญหาการจำแนกประเภทเว็บเพจว่าเป็นเพจภาษาไทยหรือไม่ใช่และปัญหาการจำแนกประเภทเว็บเพจออกเป็นหมวดหมู่ ผลการวิจัยแสดงให้เห็นว่าอัลกอริทึมการสอนไขว้แบบวน-ซ้ำมีประสิทธิภาพใกล้เคียงกับอัลกอริทึมเบย์อย่างง่ายแบบสอน แต่มีข้อดีที่สามารถใช้ประโยชน์จากข้อมูลไม่มีฉลากได้ ทำให้จำนวนข้อมูลมีฉลากที่ต้องการน้อยลงได้ และพบว่าอัลกอริทึมการสอนไขว้แบบวนซ้ำมีประสิทธิภาพมากกว่าอัลกอริทึมอื่นๆ เช่น อัลกอริทึมโคเทรนนิง อัลกอริทึมอีเอ็ม ที่ใช้ข้อมูลไม่มีฉลากในการเรียนรู้ นอกจากนั้นจากการศึกษาถึงผลกระทบของข้อมูลสัญญาณรบกวนที่มีต่ออัลกอริทึม เราพบว่าอัลกอริทึมการสอนไขว้แบบวนซ้ำมีความทนทานต่อข้อมูลมีสัญญาณรบกวนได้ดีกว่าอัลกอริทึมอื่นๆ

3. การปรับปรุงประสิทธิภาพของการสอนไขว้แบบวนซ้ำด้วยการโปรแกรมตรรกะเชิงอุปนัย

อัลกอริทึมการสอนไขว้แบบวนซ้ำที่กล่าวในหัวข้อที่แล้วสามารถใช้จำแนกประเภทเว็บเพจและให้ผลเป็นที่น่าพอใจ อย่างไรก็ดีพบว่ายังมีข้อจำกัดในด้านประสิทธิภาพของการจำแนกประเภท ในหัวข้อนี้จะกล่าวถึงการใช้การโปรแกรมตรรกะเชิงอุปนัยมาช่วยเพิ่มประสิทธิภาพของการสอนไขว้แบบวนซ้ำ

3.1 อัลกอริทึมการสอนไขว้แบบวนซ้ำโดยใช้การโปรแกรมตรรกะเชิงอุปนัย

อัลกอริทึมการสอนไขว้แบบวนซ้ำประกอบด้วยตัวแยกแยะย่อย 2 ตัว ในหัวข้อนี้เรานำการโปรแกรมตรรกะเชิงอุปนัยมาเป็นตัวแยกแยะย่อยตัวที่หนึ่งและใช้ตัวแยกแยะเบย์อย่างง่ายเป็นตัวแยกแยะย่อยตัวที่สอง ดังแสดงในรูปที่ 9 สำหรับการโปรแกรมตรรกะเชิงอุปนัยนั้น เราได้เลือกใช้ โปรกอล (PROGOL) [Muggleton, 1995] และได้สร้างความรู้ภูมิหลังสำหรับเว็บเพจแต่ละประเภท และกำหนดเว็บเพจตัวอย่างเพื่อการเรียนรู้จำนวนหนึ่งให้กับอัลกอริทึมการสอนไขว้แบบวนซ้ำ



รูปที่ 9 อัลกอริทึมการสอนไขว้แบบวนซ้ำโดยใช้การโปรแกรมตรรกะเชิงอุปนัย

ตัวแยกแยะที่หนึ่ง (*Classifier1*) จะทำการเรียนรู้จากตัวอย่างที่มีฉลากและตัวอย่างที่ไม่มีฉลาก การทำงานจะเริ่มจากการนำความรู้ภูมิหลังและตัวอย่างที่มีฉลากมาสร้างกลุ่มของกฎ และนำกฎที่ได้มาจำแนกประเภทตัวอย่างเว็บเพจที่ไม่มีฉลาก (*TrainingData1*) หลังจากนั้นจะนำเว็บเพจที่ผ่านการจำแนกประเภทไปให้ตัวแยกแยะที่สองทำการเรียนรู้โดยใช้ตัวแยกแยะเบย์อย่างง่าย เมื่อตัวแยกแยะที่สองเรียนรู้ได้สำเร็จก็จะทำการจำแนกประเภทเว็บเพจใน *TrainingData2* ต่อมาตัวแยกแยะที่หนึ่งจะทำการเรียนรู้โดยใช้การโปรแกรมตรรกะเชิงอุปนัย และจะทำเช่นนี้ไปจนกระทั่งลู่เข้า

3.2 การสกัดลักษณะสำคัญของเว็บเพจและความรู้ภูมิหลัง

หลังจากที่ทำการตัดคำหยุดและหารากคำเรียบร้อยแล้ว เราได้ทำการสร้างเพรดิเคตเพื่อใช้เป็นลักษณะสำคัญ (feature) ให้กับระบบไอแอลพีใช้ในการเรียนรู้ เพรดิเคตที่ถูกสร้างขึ้นแบ่งเป็น 3 กลุ่มดังนี้

- เพรดิเคต *has_title(P,Word)* หมายถึง เว็บเพจ *P* มีคำ *Word* อยู่ที่ชื่อเว็บเพจ
- เพรดิเคต *has_head(P,Word)* หมายถึง เว็บเพจ *P* มีคำ *Word* อยู่ที่หัวข้อย่อยของเว็บเพจ
- เพรดิเคต *has_link(P,Word)* หมายถึง เว็บเพจ *P* มีคำ *Word* อยู่ที่ไฮเปอร์ลิงค์ของเว็บเพจ

นอกจากนี้เราได้สร้างความรู้ภูมิหลังให้แก่ประเภทของเว็บเพจโดยอยู่ในรูปของเพรดิเคต ตัวอย่างเช่น

classMaterial(textbook).
classMaterial(slide).
classMaterial(syllabus).
assignment(project).
assignment(homework).

เมื่อตัวแยกแยะที่หนึ่งทำการเรียนรู้เสร็จสิ้นลงจะสามารถผลิตกฎตรรกะลำดับที่หนึ่งได้ เช่น

coursehomepage(A) :- has_link(A,B) , assignment(B).

ซึ่งหมายถึง เว็บเพจ *A* จะจัดอยู่ในประเภท *course homepage* เมื่อมีลิงค์ไปยังเพจ *B* และมีคำที่เกี่ยวข้องกับ *assignment* อยู่ที่ไฮเปอร์ลิงค์ซึ่งก็คือ คำว่า *project* และ *homework* เป็นต้น

3.3 การทดลอง

เราได้ดำเนินการทดลองบนข้อมูล 2 ชุด คือ WebKb และ DrugUsage โดยทำการวัดประสิทธิภาพของอัลกอริทึมโดยใช้ค่าความแม่นยำ (P) ค่าเรียกคืน (R) และ ค่า F_1 (F_1) ซึ่งนิยามไว้ในหัวข้อที่ 2.2.3 ในการทดลองด้านล่างนี้เราได้ทำการทดลองเปรียบเทียบกับอัลกอริทึมการสอนไขว้แบบวนซ้ำที่ใช้ไอแอลพี (ICT-ILP) กับอัลกอริทึมการสอนไขว้แบบวนซ้ำดั้งเดิม (ICT-NB) และยังได้ทดลองเปรียบเทียบกับอัลกอริทึมโคเทรนนิง (CoTraining) [Blum & Mitchell, 1998] อีเอ็ม (EM) [Dempster et al. 1977] และตัวแยกแยะเบย์อย่างง่ายแบบสอน (S-Bayes)

3.3.1 ผลการทดลองกับชุดข้อมูล WebKb

ในการทดลองกับชุดข้อมูล WebKb นี้ เราได้กำหนดสัดส่วนของตัวอย่างที่มีฉลากโดยสุ่มเลือกมาเป็นจำนวน 30% สำหรับเว็บเพจแต่ละประเภท และกำหนดให้ตัวอย่างในการเรียนรู้ที่ไม่มีฉลากมีจำนวน 30% ส่วนที่ใช้ในการทดสอบวัดประสิทธิภาพมีจำนวน 40% และได้ทำการทดลอง 5 ครั้งและนำผลการทดลองมาเฉลี่ย (5-fold cross validation) ผลการทดลองแสดงในตารางที่ 21

ตารางที่ 21 ประสิทธิภาพการทำงานเฉลี่ยบนชุดข้อมูล WebKb

อัลกอริทึม	ตัวแยกแยะที่ 1			ตัวแยกแยะที่ 2		
	P	R	F_1	P	R	F_1
ICT-ILP	80.00	81.82	80.90	82.61	86.36	84.44
ICT-NB	71.85	94.39	78.25	67.23	86.73	71.76
S-Bayes	74.99	87.24	79.91	76.95	84.18	79.60
CoTraining	73.95	84.69	75.64	79.69	60.72	66.14
EM	68.62	91.64	75.98	76.78	74.28	70.70

ผลการทดลองพบว่าตัวแยกแยะที่หนึ่งของอัลกอริทึมการสอนไขว้แบบวนซ้ำได้ถูกพัฒนาให้มีความสามารถวัดโดยค่า F_1 เพิ่มขึ้นจากเดิม 78.25% (ICT-NB) เป็น 80.90% (ICT-ILP) และเมื่อเปรียบเทียบกับอัลกอริทึมเบย์อย่างง่ายแบบสอนซึ่งได้รับข้อมูลตัวอย่างที่มีฉลากมากกว่าอัลกอริทึมการสอนไขว้แบบวนซ้ำถึง 2 เท่า (60%) พบว่าอัลกอริทึมเบย์อย่างง่ายแบบสอนมีประสิทธิภาพด้อยกว่าอัลกอริทึมการสอนไขว้แบบวนซ้ำ (79.91%) นอกจากนี้อัลอริทึมการสอนไขว้แบบวนซ้ำชนิดการโปรแกรมตรรกะเชิงอุปนัยนี้ยังให้ประสิทธิภาพการทำงานที่สูงกว่าโคเทรนนิงและอีเอ็มอีกด้วย

เมื่อพิจารณาตัวแยกแยะที่สอง พบว่าอัลกอริทึมการสอนไขว้แบบวนซ้ำชนิดการโปรแกรมตรรกะเชิงอุปนัยสามารถเพิ่มประสิทธิภาพให้แก่ตัวแยกแยะที่สอง โดยได้ค่า F_1 เป็น 84.44% ซึ่งสูงกว่าวิธีเดิม 12.68% จะเห็นได้ว่าการโปรแกรมตรรกะเชิงอุปนัยได้มีส่วนช่วยให้ตัวแยกแยะที่สอง (ตัวแยกแยะเบย์อย่างง่าย) ทำงานได้มีประสิทธิภาพสูงขึ้น อย่างไรก็ดีอัลกอริทึมการสอนไขว้แบบวนซ้ำชนิดการโปรแกรมตรรกะเชิงอุปนัยจะใช้เวลาการเรียนรู้นานกว่าแบบเดิม

3.3.2 ผลการทดลองกับชุดข้อมูล DrugUsage

ในการทดลองกับชุดข้อมูล DrugUsage นี้ เราได้กำหนดสัดส่วนของตัวอย่างที่มีฉลากโดยสุ่มเลือกมา เป็นจำนวน 33% สำหรับเว็บเพจแต่ละประเภท และกำหนดให้ตัวอย่างในการเรียนรู้ที่ไม่มีฉลากมีจำนวน 33% ส่วนที่ใช้ในการทดสอบวัดประสิทธิภาพมีจำนวน 34% และได้ทำการทดลอง 3 ครั้งและนำผลการทดลองมาเฉลี่ย (3-fold cross validation) ผลการทดลองแสดงในตารางที่ 22

ตารางที่ 22 ประสิทธิภาพการทำงานเฉลี่ยบนชุดข้อมูล DrugUsage

อัลกอริทึม	ตัวแยกแยะที่ 1			ตัวแยกแยะที่ 2		
	P	R	F_1	P	R	F_1
ICT-ILP	82.37	98.32	89.90	56.03	88.20	65.39
ICT-NB	60.54	80.66	69.17	57.14	70.30	63.04
S-Bayes	75.74	92.67	83.35	68.81	87.12	76.89
CoTraining	55.51	62.52	58.81	50.45	77.56	61.14
EM	53.02	46.45	49.52	46.06	76.55	57.51

จากผลการทดลองพบว่าอัลกอริทึมการสอนไขว้แบบวนซ้ำชนิดการโปรแกรมตรรกะเชิงอุปนัย (ICT-ILP) มีประสิทธิภาพการทำงานสูงสุดโดยเพิ่มจาก 69.17% เป็น 89.90% สำหรับตัวแยกแยะที่สองของ ICT-ILP ก็พบว่าได้ประสิทธิภาพที่สูงเมื่อเปรียบเทียบกับอัลกอริทึมอื่นๆ

3.4 สรุปผลการใช้การโปรแกรมตรรกะเชิงอุปนัยเพื่อเพิ่มประสิทธิภาพของการสอนไขว้แบบวนซ้ำ

เราพบว่าการโปรแกรมตรรกะเชิงอุปนัยได้มีส่วนช่วยให้ประสิทธิภาพของอัลกอริทึมการสอนไขว้แบบวนซ้ำทำงานได้ดียิ่งขึ้น ทั้งนี้เป็นเพราะการนำความรู้ภูมิหลังที่เหมาะสมใส่ให้แก่ระบบเพื่อใช้ในการสร้างกฎที่สามารถใช้ในการจำแนกประเภทของข้อมูลได้อย่างถูกต้อง โดยกฎที่ได้จากการเรียนรู้สามารถใช้เป็นตัวแทนของกลุ่มข้อมูลได้เป็นอย่างดีอีกทั้งยังมีความหมายที่อ่านเข้าใจได้ง่ายอีกด้วย

4. สรุปผลการวิจัย

ในงานวิจัยนี้ เราได้นำเสนอวิธีการสำหรับการจำแนกประเภทเว็บเพจออกเป็นหมวดหมู่โดยอัตโนมัติ หัวข้อที่เน้นทำวิจัย คือ (1) การวิจัยพื้นฐานเพื่อเพิ่มประสิทธิภาพของการโปรแกรมตรรกะเชิงอุปนัย และ (2) การวิจัยเทคนิคการเรียนรู้ของเครื่องที่สามารถใช้ประโยชน์จากข้อมูลแบบไม่มีฉลาก สำหรับการเพิ่มประสิทธิภาพของการโปรแกรมตรรกะเชิงอุปนัยนั้น เราได้นำเสนอวิธีการใหม่สำหรับหากฎที่ตรงกับข้อมูลมากที่สุด ซึ่งวิธีการนี้ใช้กระบวนการดึงลักษณะสำคัญร่วมกับนิวรอลเน็ตเวิร์ก ผลการวิจัยแสดงให้เห็นว่าวิธีการที่นำเสนอมีประสิทธิภาพสูงกว่าวิธีการอื่นๆ ของการโปรแกรมตรรกะเชิงอุปนัยที่นำมาเปรียบเทียบ นอกจากนี้เรายังได้นำเสนอวิธีการเรียนรู้แบบใหม่ที่เรียกว่า การสอนไขว้แบบวนซ้ำ ซึ่งสามารถใช้ประโยชน์จากข้อมูลไม่มีฉลากได้ วิธีการนี้เหมาะสำหรับการจำแนกประเภทเว็บเพจบนอินเตอร์เน็ตที่มีข้อมูลมากมายมหาศาลที่ไม่มีฉลาก การสอนไขว้แบบวน-ซ้ำมีจุดเด่นที่ใช้ตัวแยกแยะย่อยสองตัว ที่สามารถโต้ตอบและสอนกันเองไปมาจนกระทั่งการเรียนรู้สำเร็จ ผลที่ได้แสดงให้เห็นว่า การสอนไขว้แบบวนซ้ำสามารถใช้ประโยชน์จากข้อมูลไม่มีฉลากได้เป็นอย่างดี และมีประสิทธิภาพใกล้เคียงกับการเรียนรู้แบบสอนที่ต้องใช้ข้อมูลมีฉลากทั้งหมด เรายังได้เพิ่มประสิทธิภาพของการสอนไขว้แบบวนซ้ำ โดยการนำการโปรแกรมตรรกะเชิงอุปนัยมาเป็นตัวแยกแยะย่อย ซึ่งเพิ่มประสิทธิภาพของการสอนไขว้แบบวนซ้ำให้สูงกว่าเดิมได้อย่างมาก และมีประสิทธิภาพสูงกว่าวิธีการอื่นๆ ทุกวิธีที่นำมาทดสอบ รวมทั้งการเรียนรู้แบบสอนอีกด้วย

เอกสารอ้างอิง

Blockeel, H. and Raedt, L. D. (1997) Experiments with top-down induction of logical decision trees. Technical Report CW247. Department of Computer Sciences, K.U.Leuven.

Blum, A. and Mitchell, T. (1998) Combining labeled and unlabeled data with co-training, In *Proceeding of the Eleventh Annual Conference on Computational Learning Theor*y.

Clark, P. and Niblett, T. (1989) The CN2 induction algorithm. *Machine Learning* 3(4): 261-283.

Dempster, A.P., Laird, N. M., and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B 39:1-38.

Dolsak, B. and Muggleton, S. (1992) The application of inductive logic programming to finite element mesh design. In S. Muggleton (Ed.), *Inductive Logic Programming*, pp. 453-472, Academic Press.

Dzeroski, S., Schulze-Kremer, S., Heidtke, K. R., Siems, K., and Wettschereck, D. (1996) Applying ILP to diterpene structure elucidation from 13C NMR spectra. In *Proceedings of the Ninth International Workshop on Inductive Logic Programming*.

Flach, P. and Lachiche, N. (1999) 1BC: A first-order Bayesian classifier. In *Proceedings of the Ninth International Workshop on Inductive Logic Programming*, pp. 92-103, Springer LNAI 1634.

Joachims, T. (1998) Text categorization with support vector machines: Learning with many relevant feature, In *Proceedings of Tenth European Conference on Machine Learning*, Springer Verlag.

Kijsirikul, B. and Sinthupinyo, S. (1999) Approximate ILP rules by backpropagation neural network: A result on Thai character recognition. In *Proceedings of the Ninth International Workshop on Inductive Logic Programming*, pp. 162-173, Springer-Verlag.

Lavrac, N. and Dzeroski, S. (1994) *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood.

Lawrence, S. and Giles, C. L. (1998) Searching the World Wide Web. *Scienc*e, 280(5360):98–100.

Ma, Y., Liu, S.W. and Huang, T.W. (1996) WWW search engines. *High Performance Communication Network Course Report, University of California at Berkeley*.

McCallum, A., Rosenfeld, R., Mitchell, T. and Nigam, A. (1998) Improving text classification by shrinkage in a hierarchy of cl*asse*s, In *Proceedings of the Fifteenth International Conference on Machine Learnin*g, pp. 350-358, Morgan Kaufmann.

Mitchell, T. (1997) *Machine Learning*, pp.180-184, McGraw-Hill. New York.

Mladenic, D. and Grobelnik, M. (1999) Assigning keywords to documents using machine learning. In *Proceedings of the Tenth International Conference on Information and Intelligent Systems IIS-99, Varazdin, Croatia*.

Muggleton, S. (1991) Inductive logic programming, *New Generation Computing*, 8(4), 295-318.

Muggleton, S. (1995) Inverse entailment and PROGOL. *New Generation Computing*, 13, 245-286.

Muggleton, S., Bain, M., Hayes-Michie, J. and Michie, D. (1989) An experimental comparison of human and machine learning algorithms. In *Proceedings of the Sixth International Workshop on Machine Learning*, pp. 113-118, Morgan Kaufmann.

Muggleton, S. and Feng, C. (1990) Efficient induction of logic programs. In *Proceedings of the First Conference on Algorithmic Learning Theory*, pp. 368-381, Ohmsha, Tokyo.

Muggleton, S. and De Raedt, L. (1994) Inductive logic programming: Theory and methods, *Journal of Logic Programming*, 19:20, 629-679.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. (1999) Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39(2):103-134.

Nilsson, N. J. (1980) *Principles of artificial intelligence.* Tioga, Palo Alto, CA.

Pierre, J.M. (2000) Practical issues for automated categorization of Web sites. In *Proceedings of Conference on Sematic Web*.

Porter, M.F. (1980) An algorithm for suffix stripping. *Program* 14(3): 130-137.

Quinlan, J.R. (1990) Learning logical definitions from relations. *Machine Learning* 5(3):239-266.

Quinlan, J. R. (1993) *C4.5:Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.

Srinivasan, A., Muggleton, S., Sternberg, M. J. E. and King, R. D. (1996) Theories for mutagenicity: A study in first-order and feature-based induction. *Artificial Intelligence*, 85, 277-299.

Yang, Y. and Pederson, J. (1997) Feature selection in statistical learning of text categorization, In *Proceeding of the Fourteenth International Conference on Machine Learning* (pp. 412-420).

AltaVista, http://www.altavista.com

DrugUsage. http://www.kindcu.siit.ac.th

Excite, http://www.excite.com

Google, http://www.google.com

Hotbot, http://www.hotbot.com

Infoseek, http://infoseek.go.com

WebClass. http://www.di.uniba.it/~malerba/software/webclass/webclass.html

WebKb. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/course-cotrain-data.tar.gz

III. ผลที่ได้จากโครงการ

งานวิจัยที่ตีพิมพ์ในวารสารวิชาการระดับนานาชาติ

1. Boonserm Kijsirikul, Sukree Sinthupinyo and Kongsak Chongkasemwongse, “Approximate Match of Rules Using Backpropagation Neural Networks”, Machine Learning Journal, Vol. 44 (3), pp.273-299, September, 2001.
2. Nuanwan Soonthornphisaj, Boonserm Kijsirikul, “Iterative Cross-Training: An Algorithm for Web Page Categorization”, Intelligent Data Analysis, Vol. 7(3), pp.233-253, July 2003.
3. Nuanwan Soonthornphisaj, Boonserm Kijsirikul, “Iterative Cross-Training: An Algorithm for Learning from Unlabeled Web Pages”, International Journal of Intelligent Systems, 2003 (to appear).

Book Chapter

4. Nuanwan Soonthornphisaj, and Boonserm Kijsirikul, “The Effects of Different Feature Sets on Web Page Categorization Problem using Iterative Cross-Training Algorithm”, In Enterprise Information Systems III, J. Filipe, B. Sharp and P. Miranda (Eds.), Kluwer, pp. 132-138, 2002.

รางวัลวิจัย

ผลงานของโครงการวิจัยนี้ได้รับรางวัลวิจัยดังต่อไปนี้

5. รางวัลผลงานวิจัยดี จากกองทุนรัชดาภิเษกสมโภช ประเภทอาจารย์และนักวิจัยประจำสถาบัน ประจำปี 2545 จากจุฬาลงกรณ์มหาวิทยาลัย
6. รางวัลบทความวิจัยยอดเยี่ยม (best paper award) จาก The Second International Conference on Intelligent Technologies (InTech-2001) บทความเรื่อง “An Evaluation of the Incremental Iterative Cross-Training Approach on Web Page Classification”, November 27-29, 2001.

การนำเสนอผลงานในการประชุมวิชาการนานาชาติ

7. Sukree Sinthupinyo and Boonserm Kijsirikul “Combining Neural Networks with Inductive Logic Programming for Predicting Unseen and Noisy Data”, In Proc. of International Conference on Intelligent Technologies (InTech-2000), Bangkok, Thailand, December 12-14, 2000.
8. Nuanwan Soonthornphisaj and Boonserm Kijsirikul “Iterative Cross-Training: An Algorithm for Learning from Unlabeled Web Pages”, In Proc. of International Conference on Intelligent Technologies (InTech-2000), Bangkok, Thailand, December 12-14, 2000.
9. Nuanwan Soonthornphisaj and Boonserm Kijsirikul, “The effects of different feature sets on Web Page Categorization Problem using Iterative Cross-Training Algorithm”, International Conference on Enterprise Information Systems (ICEIS-2001), Portugal, 7-10 July, 2001.

การนำเสนอผลงานในการประชุมวิชาการนานาชาติ (ต่อ)

10. Nuanwan Soonthornphisaj and Boonserm Kijsirikul, “An Evaluation of the Incremental Iterative Cross-Training Approach on Web Page Classification”, In Proc. of the Second International Conference on Intelligent Technologies (InTech-2001), Bangkok, 27-29, November 2001. (Best Paper Award).
11. Nuanwan Soonthornphisaj, Pisit Chartbanchachai, Thanapol Pratheeptham, and Boonserm Kijsirikul, “Web Page Categorization Using Hierarchical Headings Structure”, International Conference on Information Technology Interface (ITI2002), Croatia, June 24-27, 2002.

การผลิตบัณฑิต

โครงการนี้ได้ผลิตบัณฑิตในระดับดุษฎีบัณฑิตและมหาบัณฑิตที่ทำวิทยานิพนธ์เกี่ยวข้องกับโครงการดังต่อไปนี้

1. นายสุกรี สินธุภิญโญ	ระดับดุษฎีบัณฑิต
2. นางนวลวรรณ สุนทรภิษัช	ระดับดุษฎีบัณฑิต
3. นายก้องศักดิ์ จงเกษมวงศ์	ระดับมหาบัณฑิต
4. นายอดุลย์ ตันธุวนิตย์	ระดับมหาบัณฑิต