การตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ

นายเกียรติคุณ ชอบธรรม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

LINK FARM DETECTION USING GRAPH GRAMMAR

Mr.Kiattikun Chobtham

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2008

หัวข้อวิทยานิพนธ์	การตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ
โดย	นายเกียรติคุณ ชอบธรรม
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม)

..... กรรมการภายนอกมหาวิทยาลัย
(ดร.กฤษณ์ โกสวัสดิ์)

เกียรติคุณ ชอบธรรม : การตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ
(LINK FARM DETECTION USING GRAPH GRAMMAR) อาจารย์ที่ปรึกษา
วิทยานิพนธ์หลัก: ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษา
วิทยานิพนธ์ร่วม: ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง, 55 หน้า

งานวิจัยในการตรวจจับลิงก์ฟาร์มโดยทั่วไปมีแนวคิดในการหาอัลกอริทึมในการตรวจจับให้มีความถูกต้องเพียงอย่างเดียวโดยไม่ได้คำนึงถึงโครงสร้างลิงก์ฟาร์ม ดังนั้นจึงมีการพัฒนาไวยากรณ์กราฟมาใช้อธิบายตัวแบบของลิงก์ฟาร์ม และพัฒนาไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์ม ซึ่งในกระบวนการตรวจจับลิงก์ฟาร์มนั้นมีการนับจำนวนของการใช้โปรดักชันจากข้อมูลสอน และมีอัลกอริทึมตรวจจับลิงก์ฟาร์มซึ่งใช้กฎดรรกศาสตร์ในการจำแนกความเป็นสแปมโฮสจากระดับตรวจจับทั้งหมด 20 ระดับ ผลการทดลองพบว่าเมื่อเปรียบเทียบกับงานวิจัยที่เกี่ยวข้องประสิทธิภาพในการตรวจจับลิงก์ฟาร์มนั้นได้ผลที่ดี ดังนั้นการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟที่นำเสนอนั้นเมื่อใช้จำนวนการใช้โปรดักชันทั้งในเว็บกราฟระดับเว็บเพจและโฮสพิจารณาร่วมกันสามารถนำมาใช้ในการตรวจจับลิงก์ฟาร์มได้เป็นอย่างดี

ภาควิชา: วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต: เกียรติคุณ ชอบธรรม
สาขาวิชา: วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก:
ปีการศึกษา: 2551 ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม:

4970234021: MAJOR COMPUTER SCIENCE
KEY WORD: GRAPH GRAMMAR / CONTEXT-FREE GRAPH GRAMMAR / PARSING ALGORITHM / LINK FARM / WEB SPAM / SPAM DETECTION

KIATTIKUN CHOBTHAM: LINK FARM DETECTION USING GRAPH GRAMMAR. THESIS PRINCIPAL ADVISOR: ASST. PROF. ATHASIT SURARERKS, Ph.D., THESIS CO-ADVISOR: ASST. PROF. ARNON RUNGSAWANG, Ph.D., 55 pp.

There are many link farm detection techniques proposed in the literature review. These techniques only involve designing algorithms for detection with high precision without considering the structure of link farm. In our work, we introduce a new graph grammar model for expressing the structure of a link farm and a graph grammar for the link farm detection. Supervised graph grammar induction is modified to fit the training data with the number of applying production rules. Link farm detection algorithm is proposed and it uses logical rule to classify target hosts with 20 steps of detection. Compared with the related works, graph grammar in the experiments can effectively recognize link farms from web spam dataset. The comparison between the frequency of usage of some productions of spam and those of normal hosts indicates that graph grammar seem to be a good mechanism for detecting link farm.

Department: Computer Engineering Student's signature: .....
Field of study: Computer Science Thesis principal advisor's signature: .....
Academic year: 2008 Thesis co-advisor's signature: .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์ และความช่วยเหลืออย่างยิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ และผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง อาจารย์ที่ปรึกษา ซึ่งเป็นผู้ให้ข้อคิด แนวทาง และคำปรึกษา ตลอดจนเป็นผู้ตรวจทานแก้ไข ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง ขอขอบพระคุณเป็นอย่างสูงที่ให้ความเมตตา ช่วยเหลือ รวมทั้งโอกาสและสิ่งที่ดีแก่ผู้วิจัยเสมอมา

ขอขอบพระคุณศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา ผู้ช่วยศาสตราจารย์ ดร. เศรษฐา ปานงาม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ดร.กฤษณ์ โกสวัสดิ์ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ประธานกรรมการและกรรมการสอบวิทยานิพนธ์ ที่ได้กรุณาให้คำแนะนำในการแก้ไขวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น และขอขอบพระคุณคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัยทุกท่านที่ประสิทธิประสาทความรู้อันมีค่ายิ่งแก่ผู้วิจัย

ท้ายนี้ขอขอบพระคุณ บิดา มารดา ที่เป็นกำลังใจสำคัญ และขอขอบคุณ เพื่อนๆ พี่ๆ และน้องๆ ทุกคน ที่เปรียบเสมือนแรงผลักดันและให้ความช่วยเหลือในทุกๆ ด้านจนผู้วิจัยสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง

สารบัญ

หน้า

บทที่ หน้า

สารบัญตาราง

สารบัญภาพ

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการค้นคืนเว็บเพจให้ให้ได้ผลลัพธ์ตรงกับความต้องการของผู้ใช้ที่มีจำนวนมากบนอินเทอร์เน็ต ปัจจุบันได้มีการใช้บริการโปรแกรมค้นหา (search engine) อย่างแพร่หลาย เนื่องจากมีการพัฒนาผลลัพธ์ให้ถูกต้องและรวดเร็ว ซึ่งในปัจจุบันอาจกล่าวได้ว่าโปรแกรมค้นหาเปรียบเสมือนประตูไปสู่เว็บเพจ (web page) ต่างๆ ในโลกของเวิลด์ไวด์เว็บ (World Wide Web) โดยปกติแล้วผลลัพธ์ที่ได้จากการสืบค้นทั้งหมดจากคำค้นหาใดๆ จะมีเพียงผลคำตอบในอันดับต้นหรือลำดับหน้าต้นๆ เท่านั้นที่ผู้ใช้งานโปรแกรมค้นหาจะเปิดอ่านรายละเอียดภายในเว็บเพจ ดังนั้นด้วยบทบาทที่มากขึ้นของโปรแกรมค้นหา กลุ่มเว็บด้านธุรกิจจึงมีความต้องการให้เว็บเพจของตัวเองปรากฏอยู่ในหน้าแรกหรืออันดับต้นของผลการค้นคืนจากคำค้นหาต่างๆ จากผู้ใช้ และมีความพยายามในการสร้างความสำคัญให้กับเว็บเพจของตนเองโดยที่ไม่คำนึงถึงความเหมาะสมด้านคุณภาพของเนื้อหาตามความเป็นจริง เพื่อที่จะให้จำนวนของผู้ใช้เปิดเข้ามาอ่านหน้าเว็บเพจของตัวเองให้มากที่สุด ความพยายามเหล่านี้เรียกว่า การทำเว็บสแปม (web spam) ซึ่งเทคนิคที่นิยมทำในขั้นตอนการสร้างเว็บสแปมคือเทคนิคคอนเทนท์สแปม (content spam) และเทคนิคลิงก์สแปม (link spam)

ในการทำคอนเทนท์สแปมมีการสร้างหรือบิดเบือนเนื้อหาของหน้าเว็บเพจให้ตรงกับคำค้นหา (key word) ให้มากที่สุดเช่น การเพิ่มคำค้นหาจำนวนมากลงในเนื้อหาของเว็บเพจเป็นต้น ส่วนลิงก์สแปมเป็นการเปลี่ยนแปลงโยกย้ายลิงก์ระหว่างเว็บเพจเพื่อที่จะให้เว็บเพจเป้าหมาย (target page) ได้รับคะแนนจำนวนมากจากเว็บเพจอื่นทำให้อันดับของผลลัพธ์ในโปรแกรมค้นหาเพิ่มขึ้นจากการคิดคะแนนความสำคัญ (ranking score) ของเว็บเพจ ซึ่งความสำคัญของเว็บเพจนั้นนิยมสร้างโดยอัลกอริทึมการคิดคะแนนจากลิงก์ที่เชื่อมโยงกันแต่ละเว็บเพจเช่นเพจแรงค์ (PageRank) เป็นต้น ทำให้ปัจจุบันผู้สร้างสแปมมุ่งเน้นทำเว็บสแปมโดยใช้เทคนิคแบบลิงก์สแปมเพราะการทำคอนเทนท์สแปมสามารถตรวจจับได้ง่ายโดยมนุษย์ แต่การเพิ่มคะแนนเพจแรงค์ให้เว็บเพจใดๆ โดยการจงใจสร้างโครงสร้างที่มีสมบัติเฉพาะตัวนั้นตรวจจับได้ยากกว่า ทำให้มีนักวิจัยเสนอวิธีในการค้นหากลุ่มของเว็บเพจที่มีการสร้างขึ้นแบบอัตโนมัติจากผู้สร้างสแปมและมีโครงสร้างที่อยู่กันอย่างหนาแน่นเรียกว่าลิงก์ฟาร์ม (link farm) ซึ่งจุดประสงค์หลักของลิงก์ฟาร์มคือการสร้างกลไกในการเพิ่มคะแนนเพจแรงค์ให้กับเว็บเพจเป้าหมายที่ต้องการ โดยให้อัลกอริทึมการคิดคะแนนเพจแรงค์มีความเอนเอียงจากจำนวนลิงก์ที่หนาแน่นซึ่งทำให้คะแนนเพจแรงค์ของเว็บเพจเป้าหมายมีคะแนนสูงกว่าความเป็นจริง และในปัจจุบันมีนักวิจัยค้นพบว่าจำนวนลิงก์ฟาร์มมีเพิ่มมากขึ้นเรื่อยๆ และอีกทั้งยังตรวจจับได้ยาก เกิดปัญหาทำให้ประสิทธิภาพในการค้นคืนของโปรแกรมค้นหาลดลง เพราะผลลัพธ์ของ

โปรแกรมค้นหาประกอบไปด้วยเว็บเพจที่ไม่เกี่ยวข้อง หรือเป็นโฆษณาแอบแฝงที่ไม่เป็นประโยชน์ต่อผู้ใช้งาน

ในการตรวจจับลิงก์ฟาร์มแต่ละเว็บเพจโดยใช้มนุษย์ตรวจสอบจะต้องใช้เวลาและค่าใช้จ่ายเป็นจำนวนมาก ดังนั้นการตรวจจับลิงก์ฟาร์มอัตโนมัติจึงเป็นสิ่งที่ท้าทายในการเพิ่มประสิทธิภาพและความถูกต้องของการค้นคืน โดยมีงานวิจัยที่เกี่ยวกับการตรวจจับลิงก์ฟาร์ม อาทิเช่น การนำเสนอวิธีการคำนวณคะแนนของลิงก์ฟาร์ม [1,2,3] ซึ่งจะต้องให้มนุษย์ทำการตัดสินใจในการสร้างชุดข้อมูลเริ่มต้นก่อนและใช้ทฤษฎีความน่าจะเป็นในการคำนวณหาคะแนนจากชุดข้อมูลเว็บกราฟทั้งหมด และงานวิจัยซึ่งใช้วิธีการตรวจจับลิงก์ฟาร์มโดยการพิจารณาเว็บกราฟย่อยที่อยู่รวมตัวกันอย่างหนาแน่น [4,5,6,7] และในงานวิจัยล่าสุดมีวิธีการใช้สมบัติต่างๆ ของเว็บเพจมาใช้ในการทำต้นไม้ตัดสินใจหรือซัพพอร์ตเวกเตอร์แมชชีน [8,9] ซึ่งจะต้องใช้ชุดข้อมูลตัวอย่างสอนจำนวนหนึ่งเพื่อให้ได้ผลที่น่าพอใจ ดังนั้นแนวคิดในงานวิจัยนี้จึงมีแนวทางในการศึกษาโครงสร้างของลิงก์ฟาร์มและนำเสนอโครงสร้างลิงก์ฟาร์มนั้นด้วยตัวแบบจำลองทางภาษาซึ่งจะช่วยให้สามารถตรวจจับและจัดการกำจัดลิงก์ฟาร์มได้โดยใช้วิธีพิจารณาการตรวจจับให้อยู่ในรูปแบบของปัญหาการตัดสินใจ (decision problem)

ปัญหาการตัดสินใจนั้นมีการศึกษาปัญหาในรูปของแบบจำลองที่เรียกว่าภาษาและความรู้ในทางไวยากรณ์เป็นส่วนหนึ่งของการอธิบายความซับซ้อนของปัญหานั้น จากงานวิจัยทางด้านไวยากรณ์กราฟ (graph grammar) ที่ผ่านมานั้น [10,11,12] ถูกพัฒนาขึ้นเพื่อนำมาอธิบายภาษากราฟ เช่น โครงสร้างโปรตีน สารประกอบเคมี และภาษาวิชวล (visual language) เป็นต้น และได้ผลดีในการใช้ไวยากรณ์กราฟมาช่วยแก้ปัญหา เนื่องจากปกติแล้วอัลกอริทึมที่ใช้ในการตรวจจับลิงก์ฟาร์มโดยทั่วไปไม่ได้คำนึงถึงโครงสร้างของตัวลิงก์ฟาร์มที่เป็นสมบัติในการเพิ่มคะแนนเพจแรงค์ให้กับเว็บเพจเป้าหมาย ดังนั้นจึงนำไวยากรณ์กราฟมาใช้เพราะมีกลไกในการสร้างและเปลี่ยนแปลงกราฟซึ่งสามารถนำมาอธิบายโครงสร้างที่แสดงถึงการเชื่อมโยงกันของลิงก์ฟาร์มในเครือข่ายเวิลด์ไวด์เว็บช่วยทำให้สามารถตอบปัญหาตัดสินใจที่เป็นปัญหาการตรวจจับลิงก์ฟาร์มได้

งานวิจัยนี้ได้พัฒนาไวยากรณ์กราฟในการนำเสนอแบบจำลองของลิงก์ฟาร์ม ไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์ม และอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ ซึ่งใช้หลักการของการแจงส่วนของไวยากรณ์กราฟ นอกจากนั้นยังมีการนับจำนวนการใช้โปรดักชันต่างๆ ในไวยากรณ์กราฟเพื่อสามารถแยกแยะเว็บเพจที่สแปมกับเว็บเพจที่เป็นเว็บเพจปกติ ซึ่งมีเป้าหมายเพื่อตรวจจับลิงก์ฟาร์มได้ความถูกต้องที่สูง และมีแนวทางใหม่ในการแก้ปัญหาในเชิงเครือข่ายเวิลด์ไวด์เว็บโดยใช้ความรู้ทางด้านไวยากรณ์กราฟ

1.2 วัตถุประสงค์ของงานวิจัย

1) พัฒนาตัวแบบทางภาษาที่เรียกว่าไวยากรณ์กราฟในการนำเสนอโครงสร้างของลิงก์ฟาร์มในรูปแบบทั่วไป
2) สร้างไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มบนเว็บกราฟที่แสดงถึงการถ่ายเทคะแนนเพจแรงค์ภายในลิงก์ฟาร์มที่ทำให้เว็บเพจเป้าหมายมีคะแนนเพจแรงค์ที่สูง
3) นำเสนออัลกอริทึมตรวจจับลิงก์ฟาร์มโดยใช้หลักการแจงส่วนของไวยากรณ์กราฟกับชุดข้อมูลเว็บกราฟ
4) เปรียบเทียบผลอัลกอริทึมการตรวจจับลิงก์ฟาร์มที่ใช้ไวยากรณ์กราฟกับงานวิจัยอื่นที่นำเสนอ โดยสามารถพัฒนาคุณภาพในการค้นคืนเว็บเพจทำให้โปรแกรมค้นหาสามารถตรวจจับลิงก์ฟาร์มที่เป็นโครงสร้างหนึ่งในการทำเว็บสแปมได้

1.3 ขอบเขตของงานวิจัย

1) ศึกษาแนวทางในการตรวจจับลิงก์ฟาร์ม โดยอาศัยจำนวนและความสัมพันธ์ของจำนวนของการใช้โปรดักชันต่างๆ ของไวยากรณ์กราฟ
2) เสนอไวยากรณ์กราฟที่แสดงถึงโครงสร้างของลิงก์ฟาร์มที่ทำให้เว็บเพจเป้าหมายคะแนนเพจแรงค์เพิ่มสูงขึ้น
3) อัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟจะนำเสนอในระดับโฮสและเว็บเพจของชุดข้อมูล
4) ชุดข้อมูลที่ใช้ในการทดสอบระบบจะเป็นเว็บกราฟมาตรฐานที่เก็บข้อมูลมาจากโดเมนอังกฤษ (UK domain)
5) การทดสอบประสิทธิภาพการทำงานจะเปรียบเทียบกับอัลกอริทึมของงานวิจัยที่เกี่ยวข้อง

1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

2.1) ศึกษาข้อมูลเอกสารและงานวิจัยที่เกี่ยวข้องกับไวยากรณ์กราฟ การคิดคะแนนเพจแรงค์ การทำลิงก์ฟาร์ม และอัลกอริทึมการตรวจจับลิงก์ฟาร์ม
2.2) ออกแบบหลักการของตัวแบบไวยากรณ์กราฟ และอัลกอริทึมการตรวจจับลิงก์ฟาร์ม
2.3) พัฒนาตัวแบบไวยากรณ์กราฟในการเสนอรูปแบบของลิงก์ฟาร์มและตีพิมพ์ผลการวิจัย
2.4) สร้างไวยากรณ์กราฟในการตรวจจับลิงก์ฟาร์ม
2.5) พัฒนาอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ

2.6) ทดสอบและตรวจสอบประสิทธิภาพของอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ

2.7) ปรับปรุงตัวแบบไวยากรณ์กราฟ และปรับปรุงการทำงานของอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ

2.8) วิเคราะห์ผลการทดลอง สรุปผลการวิจัยและตีพิมพ์ผลการวิจัย

2.9) เรียบเรียงและจัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1) ได้แบบจำลองไวยากรณ์กราฟสำหรับการนำเสนอโครงสร้างลิงก์ฟาร์มในรูปทั่วไป
2) นำเสนอไวยากรณ์กราฟที่ใช้ในการตรวจจับลิงก์ฟาร์ม
3) สามารถตรวจจับลิงก์ฟาร์มโดยใช้อัลกอริทึมแจงส่วนไวยากรณ์กราฟกับชุดข้อมูลเว็บกราฟ
4) สามารถทราบถึงโครงสร้างของเว็บเพจในการทำลิงก์ฟาร์ม เพื่อให้เว็บเพจเป้าหมายได้คะแนนเพจแรงค์สูงขึ้น
5) สามารถเลือกตรวจจับลิงก์ฟาร์มเฉพาะเว็บเพจเป้าหมายที่มีคะแนนเพจแรงค์สูงได้
6) สามารถจัดการ และป้องกันการทำลิงก์ฟาร์ม ไม่ให้คุณภาพในโปรแกรมค้นหาลดลง

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นผลงานวิชาการในหัวข้อเรื่องดังต่อไปนี้

1) “Link Farm Representation using Graph Grammars” โดย เกียรติคุณ ชอบธรรม อานนท์ รุ่งสว่าง และอรรถสิทธิ์ สุรฤกษ์ ในการประชุมวิชาการ The 11th National Computer Science and Engineering Conference (NCSEC2007) ณ โรงแรมมิราเคิลแกรนด์ กรุงเทพ ระหว่างวันที่ 19 -21 พฤษจิกายน พ.ศ.2550

2) “Formalization of Link Farm Structure using Graph Grammar” โดย เกียรติคุณ ชอบธรรม อานนท์ รุ่งสว่าง และอรรถสิทธิ์ สุรฤกษ์ ในการประชุมวิชาการ The 22nd IEEE International Conference on Advanced Information Networking and Applications (AINA2008) ณ เมืองโอกินาวา ประเทศญี่ปุ่น ระหว่างวันที่ 25-28 มีนาคม พ.ศ.2551

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

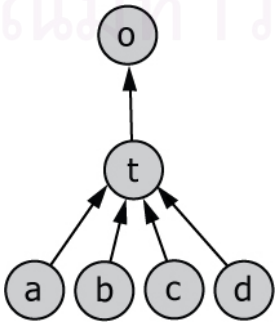
2.1.1 เว็บกราฟ

เวิลด์ไวด์เว็บเป็นเครือข่ายของเว็บเพจจำนวนมากในอินเทอร์เน็ตที่เชื่อมโยงด้วยไฮเปอร์ลิงก์ (hyperlink) หรือลิงก์ จากการศึกษาถึงความสัมพันธ์ระหว่างเว็บเพจที่มีความซับซ้อนและขนาดใหญ่พบว่ามีการใช้แบบจำลองของเว็บกราฟในงานวิจัยต่างๆ เช่น เพจแรงค์ [13] และเทคนิคการทำลิงก์ฟาร์ม ดังนั้นจึงมีการนิยามเว็บกราฟเพื่อช่วยในการให้แก้ปัญหาต่างๆ ดังนี้

นิยามที่ 2.1 *เว็บกราฟ (web graph) คือกราฟระบุทิศทาง* $G = (V, E)$ *โดยที่* V *หมายถึงเซตของโหนด (node) ที่แสดงถึงหน้าเว็บเพจหรือโฮส (host) และ* E *หมายถึงเซตของเส้นเชื่อม ที่แสดงว่ามีลิงก์อย่างน้อย 1 เส้น แทนสัญลักษณ์ด้วย* $(x, y) \in E$ *โดยที่มีเว็บเพจต้นทาง x ชี้ไปยังเว็บเพจปลายทาง y ซึ่งเงื่อนไขที่ว่าเส้นเชื่อมแต่ละเส้นไม่มีน้ำหนักและไม่มีวงวนชี้เข้าโหนดตัวเอง (self loop)*

ในแต่ละโหนดนั้นอาจจะมีโหนดที่มีลิงก์ชี้เข้ามา ซึ่งเรียกว่า อินลิงก์ (inlink) และลิงก์ที่ชี้ออกไปเรียกว่า เอาท์ลิงก์ (outlink) ส่วนจำนวนของอินลิงก์ของโหนดใดๆ จะเรียกว่า อินดีกรี (indegree) และจำนวนของเอาท์ลิงก์เรียกว่าเอาท์ดีกรี (outdegree)

ตัวอย่างที่ 2.1 *เว็บกราฟในระดับเว็บเพจดังรูปที่ 2.1 มีจำนวนโหนดทั้งหมด 6 โหนดและเส้นเชื่อมทั้งหมด 5 เส้น ในเว็บเพจ t มี อินลิงก์คือ {a,b,c,d} มีอินดีกรีเป็น 4 เอาท์ลิงก์คือ {o} และมีเอาท์ดีกรีเป็น 1* □

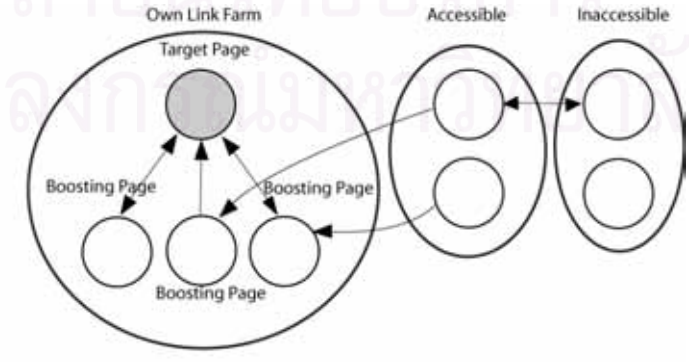


รูปที่ 2.1 แสดงเว็บกราฟในระดับเว็บเพจ

2.1.2 ลิงก์ฟาร์ม

ลิงก์ฟาร์ม[14] คือกลุ่มของเว็บเพจซึ่งอยู่ภายใต้การควบคุมอย่างสมบูรณ์ของผู้สร้างสแปมซึ่งมีจำนวนลิงก์อยู่กันอย่างหนาแน่นและมีจำนวนของเว็บเพจจำกัดเนื่องด้วยข้อจำกัดทางด้านค่าใช้จ่าย ผู้ใช้ที่คลิกเข้ามายังลิงก์ฟาร์มจะมีความน่าจะเป็นน้อยที่จะเดินทางออกจากเครือข่ายของลิงก์ฟาร์มนี้ได้ ทำให้เมื่อคำนวณค่าคะแนนความสำคัญของเว็บเพจเป้าหมายของลิงก์ฟาร์มด้วยวิธีการคำนวณโดยใช้คะแนนเพจแรงค์จะได้ค่าสูงกว่าปกติ ทำให้ได้ผลการค้นคืนในอันดับที่สูงขึ้นจากความเป็นจริงในโปรแกรมค้นหาบนอินเทอร์เน็ต การสร้างลิงก์ฟาร์มเป็นที่แพร่หลายในปัจจุบันเพราะด้วยความนิยมในการใช้โปรแกรมค้นหาและมีเพียงจำนวนผลลัพธ์การค้นคืนต้นๆ เท่านั้นที่ผู้ใช้จะเปิดหน้าเว็บเพจเข้าชม ซึ่งการสร้างลิงก์ฟาร์มมักจะใช้โปรแกรมอัตโนมัติในการสร้างเว็บเพจเสมือนที่มักจะไม่มีเนื้อหาที่เป็นประโยชน์ต่อผู้อ่านหรือเป็นเพียงข้อความที่ไม่สามารถอ่านโดยมนุษย์ได้ และจากการศึกษางานวิจัย [2,8] พบว่าเว็บเพจปกติมักจะไม่ชี้ไปหาเว็บเพจเป็นลิงก์ฟาร์ม แต่ลิงก์ฟาร์มมีอิสระในการชี้ไปยังเว็บใดๆ ก็ได้ ส่วนวิธีการเพิ่มคะแนนของลิงก์ฟาร์มยังมีเทคนิคในการรวมรวมลิงก์อื่นๆ มาจากอินเทอร์เน็ตซึ่งเทคนิคการทำลิงก์สแปมโดยทั่วๆ ไปนั้นสามารถนำเสนอในรูปเว็บกราฟทั่วไปได้ดังรูปที่ 2.2 ซึ่งสามารถแยกส่วนประกอบของเว็บเพจใดๆ ตามมุมมองของผู้สร้างสแปมได้ 3 กลุ่มคือ

1. **กลุ่มเว็บที่ไม่สามารถเข้าถึงได้** (inaccessible webpages) คือกลุ่มของเว็บเพจซึ่งผู้สร้างสแปมไม่สามารถเข้าถึงและเปลี่ยนลิงก์ให้ชี้มายังลิงก์ฟาร์มของตนได้
2. **กลุ่มเว็บที่เข้าถึงได้** (accessible webpages) คือกลุ่มของเว็บเพจที่สามารถเข้าถึงได้โดยผู้สร้างสแปมซึ่งสามารถโพสข้อความที่เป็นลิงก์เข้ามาสู่ลิงก์ฟาร์มของตน มักเป็นบล็อก (blog) วิกิพีเดีย (wikipedia) สมุดเยี่ยมหรือเว็บบอร์ด (web board)
3. **กลุ่มเว็บที่เป็นของผู้สร้างสแปม** คือกลุ่มของเว็บเพจที่เรียกว่า ลิงก์ฟาร์ม ที่อยู่ภายใต้การควบคุมอย่างสมบูรณ์ของผู้สร้างสแปม ลิงก์ฟาร์มใดๆ จะประกอบด้วยเว็บเพจเป้าหมายและบูซเพจ (boosting page)

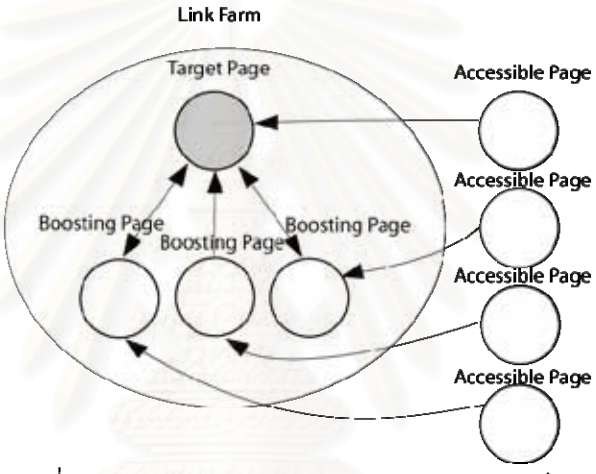


รูปที่ 2.2 แบบจำลองของเว็บกราฟในการทำลิงก์สแปม

นอกจากนี้การศึกษาถึงโครงสร้างของลิงก์ฟาร์มได้มีงานวิจัย [15] ซึ่งกล่าวว่าลิงก์ฟาร์มจะเรียกว่าออพติมอลสแปมฟาร์ม (optimal spam farm) ถ้าหากว่าสามารถทำให้คะแนนของเพจแรงค์ของเว็บเพจเป้าหมายสูงที่สุดเท่าที่จะทำได้บนเว็บกราฟดังแสดงแบบจำลองไว้ในรูปที่ 2.3 โดยเสนอเป็นทฤษฎีบทดังต่อไปนี้

ทฤษฎีบทที่ 2.1 *ลิงก์ฟาร์มจะเรียกว่าเป็น ออพติมอลสแปมฟาร์ม ก็ต่อเมื่อ*

1. *เว็บเพจที่ทำหน้าที่เพิ่มคะแนนเรียกว่า บูซเพจ จะต้องชี้ไปหาเว็บเพจเป้าหมายเพียงอย่างเดียวเท่านั้น*
2. *เว็บเพจเป้าหมายจะต้องชี้ไปบางบูซเพจเพียงอย่างเดียว*
3. *เว็บเพจที่เข้าถึงได้ต้องชี้ไปหาเว็บเพจเป้าหมายและทุกๆ บูซเพจ*



รูปที่ 2.3 แบบจำลองของออพติมอลสแปมฟาร์ม

2.1.3 เพจแรงค์

เพจแรงค์ [13] คืออัลกอริทึมในการคิดคะแนนของเว็บเพจจากโครงสร้างของเว็บกราฟที่แสดงถึงความสำคัญของเว็บเพจใดๆ เพื่อใช้ในการจัดอันดับผลการค้นคืนจากคำค้นหา ซึ่งค่าคะแนนเพจแรงค์มากย่อมแสดงว่าหน้าเว็บเพจนั้นมีความสำคัญสูง และมีโอกาสที่จะได้อันดับการค้นคืนต้นๆ ในหน้าแรกของผลลัพธ์ ซึ่งในปัจจุบันเพจแรงค์ได้รับความนิยมมาใช้ในการจัดอันดับเว็บเพจในโปรแกรมค้นหาเช่น กูเกิ้ล (google) เป็นต้น ซึ่งวิธีการคำนวณมาจากอินดีกรี เอาท์ดีกรีและความสำคัญของเว็บเพจที่ชี้เข้าหาเว็บเพจนั้น อัลกอริทึมการคิดคะแนนเพจแรงค์มีแนวคิดมาจากแบบจำลองการเดินสุ่ม (random walk) ตามทฤษฎีของมาร์คอฟเชน (markov chain) ซึ่งเปรียบเสมือนที่ผู้ใช้เริ่มต้นคลิกจากเว็บเพจต้นทางไปตามเอาท์ลิงก์ด้วยความน่าจะเป็นเท่าๆ กัน บวกกับการคลิกสุ่มกระโดด (random jump) ไปยังเว็บเพจอื่นทำให้คะแนนเพจแรงค์เป็นคะแนนโดยรวมสำหรับทุกๆ เว็บเพจ (global score) ซึ่งคิดคะแนนจากทั้งหมดในเว็บกราฟทำให้ยากในการเบี่ยงเบนคะแนนด้วยจำนวนเว็บเพจที่สร้างขึ้นเองจำนวนไม่มาก

นิยามที่ 2.2 *เพจแรงค์คือค่าคะแนนที่เกิดจากการคำนวณในเว็บกราฟโดยมาจากคะแนนเพจแรงค์ของอินลิงก์หารด้วยเอาท์ดีกรีของเว็บเพจในสมาชิกของอินลิงก์ของเว็บเพจนั้น บวกกับคะแนนที่มาจากการสุ่มกระโดดไปยังเว็บเพจอื่นๆ โดยที่*

$$Rank(u) = d\sum_{v\in B_u}\frac{Rank(v)}{N_v} + \frac{(1-d)}{N}$$

เมื่อ $Rank(u)$ *คือค่าคะแนนเพจแรงค์ของหน้าเว็บเพจ u*

$Rank(v)$ *คือค่าคะแนนเพจแรงค์ของหน้าเว็บเพจ v*

d *คือค่าคงที่ (damping factor) ซึ่งนิยมใช้ 0.85*

B_u *คือเซตของอินลิงก์ของเว็บเพจ u*

N_v *คือเอาท์ดีกรีของเว็บเพจ v*

N *คือจำนวนเว็บเพจทั้งหมดในเว็บกราฟ*

ในการคำนวณคะแนนเพจแรงค์ทั้งหมดในเว็บกราฟ จะพิจารณาสถานะ (state) ซึ่งแทนด้วยโหนดและพิจารณาทรานสิชัน (transition) ซึ่งเป็นเส้นเชื่อมของโหนดในเว็บกราฟ โดยสถานะและทรานสิชันนั้นสามารถแทนด้วยทรานสิชันเมทริกซ์ (transition matrix) ในระหว่างคำนวณจะใช้เวกเตอร์ขนาด Nx1 ซึ่งแทนด้วยค่าคะแนนเพจแรงค์ของเว็บเพจทั้งหมดของเว็บกราฟในรอบการคำนวณแต่ละรอบ โดยการคำนวณจะสิ้นสุดเมื่อค่าลู่เข้าหรือผลต่างของเวกเตอร์แต่ละรอบมีค่าน้อยกว่าค่าเริ่มเปลี่ยน (threshold)

นิยามที่ 2.3 *คะแนนเพจแรงค์ทั้งหมดในเว็บกราฟคือค่าที่เกิดจากการคำนวณความน่าจะเป็นของทรานสิชันเมทริกซ์ P โดยที่มีสูตรการคำนวณคือ*

$$X(t) = d\left(P^t + V\right)X(t-1) + \frac{(1-d)}{N}1_N$$

เมื่อ $X(t)$ *คือเมทริกซ์ขนาด Nx1 ซึ่งเป็นค่าคะแนนเพจแรงค์ของหน้าเว็บเพจทั้งหมดของเว็บกราฟในรอบ t*

d *คือค่าคงที่ (damping factor) ซึ่งนิยมใช้ 0.85*

P *คือทรานสิชันเมทริกซ์ของเว็บกราฟโดยที่*

p_{ij} *คือ* $\frac{1}{N_i}$ *ถ้าหากมีลิงก์จากเว็บเพจ i ไปยังเว็บเพจ j และ*

p_{ij} *คือ 0 ถ้าไม่มีลิงก์ระหว่างเว็บเพจ*

N_i *คือเอาท์ดีกรีของเว็บเพจ i*

V *คือเมทริกซ์* $\frac{1}{N}1_N[r_1, r_2, r_3, ..., r_N]$ *เมื่อ*

r_i *มีค่าเป็น 1 เมื่อเว็บเพจที่ i ไม่มีลิงก์ชี้ไปหาเว็บเพจอื่น*

r_i *มีค่าเป็น 0 ในกรณีอื่นๆ*

1_N *คือเมทริกซ์ขนาด* $N \times 1$ *ที่ทุกแถวมีค่าเป็น 1*

N *คือจำนวนเว็บเพจทั้งหมดในเว็บกราฟ*

จากสูตรการคำนวณคะแนนเพจแรงค์ข้างต้นสามารถแสดงรหัสเทียม [16] ได้ดังนี้

อัลกอริทึมที่ 2.1 อัลกอริทึมคำนวณคะแนนเพจแรงค์

PageRank algorithm

Input : web graph, τ (threshold), c (damping factor)

Output : PageRank

1: **Begin**

2: N ← **Count all web pages in web graph**

3: $\forall_s$Source[s]=1/N

4: **While (**residual>τ**)**

5: $\forall_d$Dest[d]=0

6: **While (**link.eof() is false**)**

7: (source,n,$dest_1$,$dest_2$,$dest_3$,…,$dest_n$) ← **Read link**

8: **For j=1 to n**

9: Dest[$dest_j$]= Dest[$dest_j$]+Source[source]/n

10: **End for loop**

11: **End while**

12: $\forall_d$Dest[d]=c×Dest[d]+(1-c)/N

13: residual=||Source-Dest||

14: Source=Dest

15: **End while**

16: PageRank= Dest

17: **End**

จากอัลกอริทึมคำนวณคะแนนเพจแรงค์จะนำเข้าคือเว็บกราฟ ค่าเริ่มเปลี่ยนและค่าคงที่ ซึ่งจะให้ผลลัพธ์ออกมาเป็นคะแนนเพจแรงค์ของแต่ละเว็บเพจในเว็บกราฟ ซึ่งแต่ละขั้นตอนสามารถอธิบายการทำงานอย่างละเอียดได้ดังนี้

บรรทัดที่ 2 : นับจำนวนเว็บเพจทั้งหมดในเว็บกราฟ

บรรทัดที่ 3 : ให้คะแนนเริ่มต้นของทุกเว็บเพจเท่าๆ กัน

บรรทัดที่ 4-15 : คำนวณหาคะแนนเพจแรงค์ในแต่รอบการอ่านลิงก์ในเว็บกราฟจนกว่าค่าคะแนนเพจแรงค์จะลู่เข้า หรือมีค่าเริ่มเปลี่ยนระหว่างรอบน้อยกว่าค่าที่กำหนด

บรรทัดที่ 5 : ให้คะแนนในรอบต่อไปเป็น 0
บรรทัดที่ 6-11 : คำนวณหาคะแนนในแต่รอบจนกว่าจะสิ้นสุดอ่านข้อมูลในเว็บกราฟ
บรรทัดที่ 7 : อ่านลิงก์ในแต่รอบออกมาเป็นเว็บต้นทาง เอาท์ดีกรีและเอาท์ลิงก์
บรรทัดที่ 8-10 : กระจายคะแนนเว็บเพจต้นทางให้กับเอาท์ลิงก์ตามจำนวนเอาท์ดีกรี
บรรทัดที่ 12 : คำนวณคะแนนเพจแรงค์ตามนิยาม
บรรทัดที่ 13 : คำนวณหาค่าเริ่มเปลี่ยน
บรรทัดที่ 14 : ให้คะแนนที่คำนวณได้เป็นค่าคะแนนในรอบต่อไป
บรรทัดที่ 16 : ให้ผลลัพธ์คะแนนที่ได้เป็นคะแนนเพจแรงค์

2.1.4 ไวยากรณ์กราฟ

ไวยากรณ์กราฟคือตัวแบบที่อธิบายถึงความซับซ้อนของภาษากราฟมีพื้นฐานมาจากทฤษฎีภาษารูปนัย (formal language) โดยที่สามารถสร้างและเปลี่ยนแปลงกราฟได้ ระบบในการสร้างไวยากรณ์กราฟโดยทั่วไปประกอบด้วยระบบที่สร้างขึ้น (induction) จากผู้เชี่ยวชาญหรือโปรแกรมอัตโนมัติ [17,18] โดยมีชุดของข้อมูลสอนหรือแบบจำลองซึ่งจะถูกสร้างให้เป็นไวยากรณ์กราฟซึ่งสามารถเปลี่ยนแปลงแก้ไขได้ภายหลังหากมีการรับตัวอย่างใหม่เข้ามาและระบบรู้จำตัวอย่างข้อมูลทดสอบเรียกว่าตัวแจงส่วน (parser) [19] เพื่อใช้ในการตอบคำถามว่าข้อมูลทดสอบที่รับเข้ามาเป็นสมาชิกในภาษาหรือไม่

ไวยากรณ์กราฟไม่พึ่งบริบท (context-free graph grammar) [20] มีความคล้ายกับไวยากรณ์ไม่พึ่งบริบทของสายอักขระเพียงแต่มีโปรดักชัน (production) ที่มีด้านซ้ายเป็นกราฟที่มีโหนดเพียงตัวเดียวส่วนทางด้านขวาของโปรดักชันเป็นกราฟใด ๆ ในการสร้างกราฟจากโปรดักชันนั้นมีความซับซ้อนมากกว่าการสร้างสายอักขระธรรมดา เพราะว่าโหนดที่แทนที่ด้วยกราฟย่อยอันใหม่จะต้องเชื่อมกับกราฟเดิมโดยใช้เส้นเชื่อมใหม่ รูปแบบคำสั่งที่บอกถึงวิธีการในการสร้างเส้นเชื่อมใหม่นั้นเรียกว่า กฎฝังตัว (embedding rule)

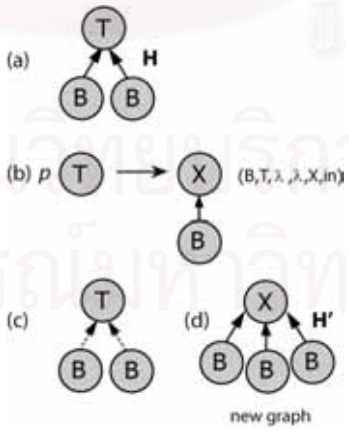
นิยามที่ 2.4 *ไวยากรณ์กราฟไม่พึ่งบริบทคืออันดับ* $G=(\Sigma_n,\Sigma_t,\Gamma_n,\Gamma_t,S,P)$ *โดย*

Σ_n *คือชุดของอักขระที่ไม่ใช่โหนดปลายทาง*
Σ_t *คือชุดของอักขระที่เป็นโหนดปลายทาง*
Γ_n *คือชุดอักษรของเส้นเชื่อม*
Γ_t *คือชุดอักษรของเส้นเชื่อมปลายทาง*
S *คือโหนดเริ่มต้น*
P *คือเซตของโปรดักชันที่อยู่ในรูปของ* $p:X\rightarrow D,E$ *ซึ่ง* X *คือกราฟที่มีโหนดเพียง โหนดเดียว,* D *คือกราฟ และ* E *คือกฎฝังตัวมีรูปแบบคำสั่ง* (n,X,γ,ρ,y,D_E)

รูปแบบคำสั่ง $(n, X, \gamma, \rho, y, D_E)$ มีความหมายคือ ถ้ามีโหนด $n \in \Sigma_n \cup \Sigma_t$ ในกราฟเดิมที่เชื่อมต่อโหนด X ด้วยเส้นเชื่อม γ เมื่อมีการกระทำโปรดักชันแล้วจะถูกเปลี่ยนให้เชื่อมต่อกับโหนด $y \in \Sigma_n \cup \Sigma_t$ ในกราฟทางด้านขวาของโปรดักชันนั้นและมีเส้นเชื่อมใหม่ชื่อว่า ρ โดยมีทิศทาง $D_E = \{in, out\}$ กรณีระบุว่า *in* เส้นเชื่อมจะมีทิศทางชี้เข้าหาโหนด y และกรณีระบุว่า *out* เส้นเชื่อมมีทิศทางชี้ไปออกจากโหนด y

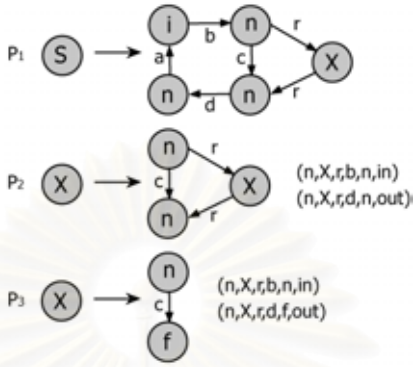
วิธีการสร้างกราฟโดยไวยากรณ์กราฟในภาษาจะใช้ส่วนของการประยุกต์ทางด้านซ้าย (left application) และการตรวจสอบสมาชิกของภาษาจะใช้การประยุกต์ทางด้านขวา (right application) การประยุกต์ทางด้านซ้ายเป็นวิธีการที่กราฟเปลี่ยนรูปจากกราฟ H ไปยัง H' โดยโปรดักชัน p แทนด้วยสัญลักษณ์ $H => H'$ และแต่ละขั้นตอนเรียกว่า ขั้นตอนการแปลง (derivation step) เมื่อทำขั้นตอนการแปลงเสร็จสิ้นจะได้ ลำดับการแปลง (derivation) ส่วนวิธีการที่กราฟเปลี่ยนรูปจากกราฟ H' ไปยัง H โดยโปรดักชัน p ซึ่งเกิดจากการกลับด้านซ้ายและด้านขวาของโปรดักชันหรือการทำย้อนกลับ เรียกว่าการประยุกต์ทางด้านขวา ซึ่งแต่ละขั้นตอนเรียกว่า ขั้นตอนการลดทอน (reduction step) ส่วนลำดับของขั้นตอนการลดทอนนั้นเรียกว่า ลำดับการแจงส่วน (parsing sequence)

ตัวอย่างที่ 2.2 *แสดงถึงตัวอย่างการใช้กฎฝังตัวโดยการประยุกต์ทางด้านซ้ายดังแสดงรูปที่ 2.4 กำหนดให้กราฟ H ใน 2.4 (a) และโปรดักชัน p ใน 2.4 (b) ขณะที่ T และ B คือโหนดในกราฟในการทำการแปลงตามนิยามที่ 2.4 ของไวยากรณ์กราฟไม่พึ่งบริบท T จะถูกเปลี่ยนให้เป็นกราฟทางด้านขวามือของโปรดักชัน และตามนิยามของกฎฝังตัว เส้นเชื่อมระหว่าง B และ T จะถูกลบออก 2.4 (c) และเส้นเชื่อมใหม่ระหว่าง B และ X จะถูกสร้างขึ้น ซึ่งสัญลักษณ์ λ นั้นบอกถึงเส้นเชื่อมที่ไม่มีอักษรกำกับ ดังนั้นกราฟใหม่ที่ได้ จะกลายเป็น H' 2.4 (d)* □



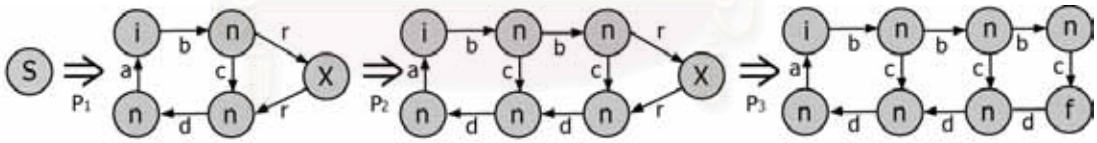
รูปที่ 2.4 ตัวอย่างของการใช้กฎฝังตัวที่สร้างโดยนิยามของไวยากรณ์กราฟ

ตัวอย่างที่ 2.3 *แสดงถึงตัวอย่างการประยุกต์ทางด้านซ้ายและการประยุกต์ทางด้านขวาเพื่อสร้างลำดับการแปลงและลำดับการแจงส่วนของกราฟ H โดยไวยากรณ์กราฟเมื่อ* $\Sigma_n = \{S, X\}$, $\Sigma_t = \{i, n, f\}$, $\Gamma_n = \{a, b, c, r\}$, $S = \{S\}$ *และ P คือเซตของ 3 โปรดักชันที่มีกฎฝังตัวดังแสดงในรูปที่ 2.5*



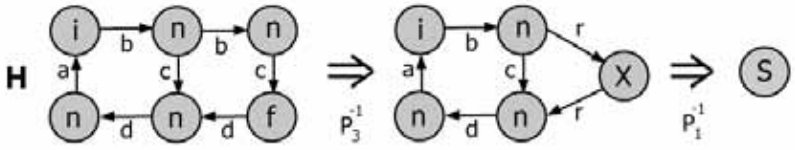
รูปที่ 2.5 โปรดักชันแสดงไวยากรณ์กราฟของตัวอย่างที่ 2.3

จากตัวอย่างไวยากรณ์กราฟสามารถสร้างการประยุกต์ทางด้านซ้ายเพื่อสร้างลำดับการแปลงได้คือ เริ่มต้นจากโหนดเริ่มต้น S มีการประยุกต์ทางด้านซ้ายจะได้กราฟทางด้านขวาของโปรดักชันที่ 1 ต่อมามีการใช้โปรดักชันที่ 2 โหนด X จะถูกให้เปลี่ยนเป็นกราฟทางด้านขวามือของโปรดักชัน ซึ่งมีกฎฝังตัวระบุไว้คือ โหนด n ที่เชื่อมกับโหนด X ด้วยเส้นเชื่อม r ในกราฟเดิมจะเปลี่ยนให้โหนด n มาเชื่อมต่อกับโหนด n ในกราฟที่สร้างขึ้นใหม่ด้วยเส้นเชื่อม b ซึ่งมีลักษณะเช่นเดียวกับกฎฝังตัวอีกข้อหนึ่ง เมื่อทำโปรดักชันที่ 2 เสร็จสิ้นจึงทำโปรดักชันที่ 3 ซึ่งลำดับการแปลงสามารถแสดงได้ดังรูปที่ 2.6



รูปที่ 2.6 ลำดับการแปลงไวยากรณ์กราฟของตัวอย่างที่ 2.3

และจากไวยากรณ์กราฟตัวอย่างสามารถใช้การประยุกต์ทางด้านขวาเพื่อสร้างลำดับการแจงส่วนของกราฟ H ได้คือ เริ่มต้นจากการค้นหาโปรดักชันที่สามารถทำการลดทอนจนสามารถได้เป็นโหนด S เริ่มต้นเพียงโหนดเดียว ซึ่งลำดับการแจงส่วนแสดงได้ดังรูปที่ 2.7 □



รูปที่ 2.7 ลำดับการแจงส่วนไวยากรณ์กราฟของตัวอย่างที่ 2.3

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยตรวจจับเว็บเพจที่เป็นลิงก์ฟาร์ม [4]

เบาว์นิ่ง วู และเบรน เดวิสัน (Baoning Wu, Brain D. Davison) พัฒนาอัลกอริทึมในการตรวจจับลิงก์ฟาร์มแบบอัตโนมัติในปี 2005 โดยนำเสนอว่าลิงก์ฟาร์มนั้นมีการเชื่อมโยงกันอย่างหนาแน่นและมีลิงก์เชื่อมโยงร่วมกัน ดังนั้นวิธีการค้นหาลิงก์ฟาร์มโดยเริ่มต้นจากการสร้างเซตเริ่มต้นจากการพิจารณาการมีลิงก์ร่วมกันระหว่างอินลิงก์และเอาท์ลิงก์ ต่อมามีการขยายขอบเขตของเซตลิงก์ฟาร์มโดยที่เว็บเพจที่พิจารณาว่าเป็นลิงก์ฟาร์มนั้นจะมีเอาท์ลิงก์ชี้ไปหาเว็บเพจอื่นๆ ในลิงก์ฟาร์ม โดยใช้วิธีการนับจำนวนลิงก์ที่ชี้ไปหาเซตเริ่มต้นของลิงก์ฟาร์ม และกระบวนการสุดท้ายคือปรับน้ำหนักกับกลุ่มของลิงก์ฟาร์มที่ตรวจจับได้

ขั้นตอนการทำงานของอัลกอริทึม แบ่งเป็น 3 ขั้นตอนคือ

1. การค้นหาเซตของลิงก์ฟาร์มเริ่มต้น
    a. ในเว็บเพจ p ให้เก็บโฮสของอินลิงก์ของเว็บเพจนั้นในเซต INdomain(p)
    b. เว็บเพจ p ให้เก็บโฮสของเอาท์ลิงก์ของเว็บเพจนั้นใส่ในเซต OUTdomain(p)
    c. ถ้าจำนวนสมาชิกของเซต INdomain(p) $\cap$ OUTdomain(p) มากกว่าหรือเท่ากับค่าที่กำหนด (T_{io}) ให้เว็บ p เป็นเว็บที่ถือว่าน่าสงสัย (bad page)
    d. วนซ้ำทำเว็บเพจทั้งหมด โดยเว็บเพจที่น่าสงสัยนั้นกำหนดให้ค่าเป็น 1 ในแถวลำดับ A[n] ส่วนเว็บเพจอื่นๆ เซตค่าเป็น 0
2. ขั้นตอนการขยายขอบเขต
    a. เว็บเพจ p ที่เป็นสมาชิกใน A[n] ที่มีค่าเป็น 0 ให้เก็บเอาท์ลิงก์ไว้ในเซต OUT(p)
    b. ตั้งค่า badnum=0
    c. ในแต่ละเว็บเพจ k ในเซต OUT(p) ถ้า A[k] เป็น 1 ให้เพิ่มค่า badnum ทีละ 1
    d. ถ้าค่า badnum มากกว่าค่า T_{pp} จึงให้แถวลำดับ A[p]=1
    e. วนซ้ำเว็บเพจ p จนกระทั่งค่า A ไม่เปลี่ยนแปลง
3. ขั้นตอนการปรับน้ำหนัก เป็นการลดค่าคะแนนของการจัดอันดับโดยคูณด้วยน้ำหนักให้ลดลงตามจำนวนลิงก์ที่ชี้ออกไปยังลิงก์ฟาร์ม

ผลลัพธ์ที่ได้พบว่าอัลกอริทึมสามารถตรวจจับลิงก์ฟาร์มส่วนใหญ่ในชุดข้อมูลทดสอบได้ และอันดับของผลการค้นหาดีขึ้น แต่ไม่สามารถแยกแยะโครงสร้างการเชื่อมต่อของเว็บเพจภายในโฮสเดียวกัน และอาจจะมีกลุ่มชุมชนของเว็บ (web community) ที่เป็นเว็บเพจที่ไม่ใช่ลิงก์ฟาร์มแต่มีโครงสร้างที่อัลกอริทึมตรวจจับได้

2.2.2 งานวิจัยตรวจจับเว็บสแปมโดยใช้โครงสร้างลิงก์ [9]

ในปี 2006 ลูก้า บีแชทตี้และคาร์ลอส คาสทิลโล (Luca Beechetti, Carlos Castillo) ได้นำเสนอวิธีการค้นหาลิงก์ฟาร์มโดยใช้ตัวจำแนก (classifier) ซึ่งมีการพิจารณาเว็บเพจช่วยเหลือ (supporter) ในระยะทางจำกัดค่าหนึ่งและใช้วิธีการคำนวณทรังเคทเพจแรงค์ (Truncated PageRank) ที่มีความคล้ายคลึงกับวิธีการคำนวณเพจแรงค์เพียงแต่จะไม่มีการคิดเว็บเพจช่วยเหลือในระยะทางใกล้ค่าหนึ่ง นอกจากนั้นยังมีการเพิ่มคุณลักษณะเพิ่มเติมให้กับตัวจำแนกเช่น เอาท์ดีกรี อินดีกรี และค่าคะแนนเพจแรงค์ ผลการทดลองโดยใช้ต้นไม้ตัดสินใจพบว่าสามารถตรวจจับเว็บเพจที่เป็นสแปมได้ถึง 80.4 % ของข้อมูลที่ใช้ในการทดสอบ

2.2.3 งานวิจัยค้นหาลิงก์ฟาร์มโดยใช้แบบจำลองการเดิมสุ่มจากเซตเริ่มต้น [1]

เบาว์นิ่ง วูและ คูมาร์ เชลล์ลาพิลา (Baoning Wu, Kumar Chellapilla) นำเสนอการตรวจจับลิงก์ฟาร์มในปี 2007 โดยการเริ่มต้นจากเซตของเว็บเพจเริ่มต้นที่เป็นลิงก์ฟาร์มโดยใช้มนุษย์ตัดสินใจและใช้การขยายขอบเขตจากแบบจำลองการเดินสุ่มโดยมีการวนรอบการคำนวณของความน่าจะเป็นโดยที่

$$p(t+1)=\frac{1}{2}\left(I+AD^{t}\right)p(t)$$

เมื่อ A คือเมทริกซ์ประชิดของเว็บกราฟ G

I คือเมทริกซ์เอกลักษณ์

D คือทรานสิชันเมทริกซ์

ซึ่งความน่าจะเป็นเริ่มต้นของโหนดในลิงก์ฟาร์มเริ่มต้น (S) คือ

$$p_0(i)=\begin{cases}1/|S| & : \text{ถ้า } i\in S\\ 0 & : \text{ถ้า } i\notin S\end{cases}$$

โดยอัลกอริทึมที่นำเสนอจะมีการกำหนดเซตของลิงก์ฟาร์มเริ่มต้นโดยมนุษย์ก่อนและใช้การคำนวณความน่าจะเป็นในรอบถัดไปในชุดข้อมูลทั้งเว็บกราฟจำนวนหลายๆ รอบจนกระทั่งลู่เข้าสู่ค่าหนึ่ง ซึ่งลิงก์ฟาร์มที่ตรวจได้คือ กลุ่มของเว็บเพจที่มีความน่าจะเป็นที่สูงกว่ากลุ่มเว็บเพจอื่น ในผลการทดลองพบว่าเมื่อใช้เซตของลิงก์ฟาร์มจำนวน 73 โฮสพบว่าสามารถให้ค่าความแม่นยำได้ถึง 95.12% ของข้อมูลที่ใช้ในการทดสอบ

2.2.4 การตรวจจับเว็บสแปมโดยใช้วิธีการแอนตี้ทรัสต์ (Anti-trust) [2]

วิธีการแอนตี้ทรัสต์ถูกนำเสนอในปี 2006 ซึ่งเป็นวิธีในการกระจายความน่าจะเป็นของลิงก์ฟาร์มในเซตเริ่มต้น ออกไปทางอินลิงก์ที่ชี้เข้ามาหาลิงก์ฟาร์ม เมื่อกำหนดให้เว็บเพจเริ่มต้นนั้นเป็นสแปมที่ตรวจสอบโดยใช้มนุษย์ ซึ่งการคำนวณหาคะแนนแอนตี้ทรัสต์นั้นจะใช้อัลกอริทึมเพจแรงค์แบบเอนเอียงที่พัฒนามาจากอัลกอริทึมการคิดคะแนนเพจแรงค์ โดยมีการเพิ่มน้ำหนักให้กับการคิดคะแนนเมื่อพบว่าเว็บเพจนั้นเป็นสแปมโดยการทำงานของอัลกอริทึมนั้นสามารถแสดงได้ดังนี้

ขั้นตอนการทำงานของอัลกอริทึมแอนตี้ทรัสต์

1. นำเข้าเซตของเว็บเพจเริ่มต้นที่เป็นสแปมและตรวจสอบจากมนุษย์ โดยเลือกจากเว็บเพจที่มีคะแนนเพจแรงค์ที่สูง
2. คำนวณเมทริกซ์สลับเปลี่ยนของเมทริกซ์ประชิด T (transpose of the binary webgraph matrix)
3. เริ่มต้นคำนวณอัลกอริทึมเพจแรงค์แบบเอนเอียง (biased PageRank) กับเมทริกซ์ T

$$X(t) = d \cdot T \cdot X(t-1) + (1-d) \cdot \alpha$$

เมื่อ $X(t)$ คือเมทริกซ์ขนาด $N \times 1$ ซึ่งเป็นค่าคะแนนเพจแรงค์แบบเอนเอียงของเว็บกราฟในรอบ t

d คือค่าคงที่ซึ่งนิยมใช้ 0.85

T คือเมทริกซ์สลับเปลี่ยนของเมทริกซ์ประชิด T' โดยที่

t'_{ij} คือ 1 ถ้าหากมีลิงก์จากเว็บเพจ i ไปยังเว็บเพจ j และ

t'_{ij} คือ 0 ถ้าไม่มีลิงก์ระหว่างเว็บเพจ i กับเว็บเพจ j

α คือค่าความน่าจะเป็นที่เอนเอียงให้กับกลุ่มเว็บเพจเริ่มต้นที่เป็นสแปม

4. จัดเรียงเว็บเพจตามคะแนนเพจแรงค์ และรับค่าเริ่มเปลี่ยนในการตัดสินใจของคะแนนเพจแรงค์ที่เป็นสแปม

ผลการทดลองจากเว็บเพจเริ่มต้นจำนวน 40 เว็บเพจที่เป็นสแปมซึ่งมีคะแนนเพจแรงค์มากที่สุด พบว่าสามารถตรวจจับเว็บเพจที่เป็นสแปมได้จำนวน 39 เว็บเพจจากจำนวน 100 เว็บเพจที่จัดเรียงตามคะแนนแอนตี้ทรัสต์ และเมื่อทดลองจากเว็บเพจเริ่มต้นจำนวน 80 เว็บเพจที่เป็นสแปม พบว่าสามารถตรวจจับเว็บเพจที่เป็นสแปมได้อย่างถูกต้องทั้ง 100 เว็บเพจที่จัดเรียงตามคะแนนแอนตี้ทรัสต์ ดังนั้นอัลกอริทึมแอนตี้ทรัสต์สามารถตรวจจับเว็บสแปมได้อย่างมีประสิทธิภาพเมื่อกำหนดให้เว็บเพจเริ่มต้นที่มีคะแนนเพจแรงค์ที่สูงและทำการเลือกจำนวนของเว็บเพจเริ่มต้นที่เป็นสแปมจำนวนมาก

2.2.5 การตรวจจับลิงก์สแปมแบบทรานส์ดักทิฟ (Transductive) [8]

ในปี 2007 มีการนำเสนอวิธีในการใช้ซัพพอร์ตเวกเตอร์แมชชีน มาช่วยในการตรวจจับลิงก์ฟาร์ม การวัดผลอัลกอริทึมนั้นมีการแบ่งชุดข้อมูลเว็บกราฟ (Webgraph 2006) ออกเป็นชุดสอนและชุดทดสอบ และใช้ฟังก์ชันในการทำนายชุดทดสอบให้ผลลัพธ์ออกมาเป็นเว็บปกติหรือเป็นสแปม ซึ่งการทำงานของอัลกอริทึมนั้นสามารถแสดงได้ดังนี้

ขั้นตอนการทำงานของอัลกอริทึมทรานส์ดักทิฟ

ให้เว็บกราฟ $G=(V,E,w)$ เป็นกราฟที่มีน้ำหนักโดยที่ V คือเซตของโหนด E คือเซตของเส้นเชื่อมและเว็บเพจ $S\subset V$ เป็นเว็บเพจที่ติดฉลากว่าปกติหรือเป็นสแปม โดยสมมติให้กราฟนั้นเป็นโครงสร้างกราฟที่หนาแน่น (strongly connected) ถ้าหากไม่ใช่โครงสร้างดังกล่าวให้ทำการแยกส่วนกราฟให้อยู่ในรูปของโครงสร้างที่เป็นกราฟย่อยที่หนาแน่น ดังนั้นเว็บเพจที่เหลือที่ไม่ได้ติดฉลากนั้นสามารถจำแนกได้โดยกระบวนการดังนี้

1. กำหนดฟังก์ชันการเดินสุ่ม ซึ่งสามารถแสดงฟังก์ชันความน่าจะเป็นคือ

$$p(u,v)=\frac{w(v,u)}{d^{-}(u)}$$

เมื่อ โหนด u,v ใดๆ คือโหนดในกราฟ $w(v,u)$ คือ น้ำหนักของเส้นเชื่อม (v,u)
$d^{-}(u)$ คือ ผลรวมของน้ำหนักของเส้นเชื่อมที่มีโหนดปลายทางเป็น u
กำหนดให้ Π คือเวกเตอร์โดยที่

$$\sum_{u\in V}\Pi(u)p(u,v)=\Pi(v)$$

2. กำหนดให้ P คือเมทริกซ์ที่มีแถวและคอลัมน์เป็น $p(u,v)$ และ Π คือเมทริกซ์ทแยงมุมที่มีสมาชิกในแนวเส้นทแยงมุมเป็น $\Pi(u)$ ดังนั้นจึงสร้างเมทริกซ์ L โดยที่

$$L=\Pi-\alpha\frac{\Pi P+P^{T}\Pi}{2}$$ เมื่อ α คือค่าคงที่อยู่ในช่วง $]0,1[$

3. กำหนดฟังก์ชัน y บนโหนดในกราฟ โดยที่ $y(v)=1$ หรือ $y(v)=-1$ ถ้าหากเว็บเพจนั้นติดฉลากว่าเป็นเว็บเพจปกติหรือสแปมตามลำดับ และให้ $y(v)=0$ ถ้าหากว่าเว็บเพจนั้นยังไม่ได้ทำการจำแนกว่าเป็นสแปม ดังนั้นแก้สมการเชิงเส้น

$$L\varphi=\Pi y$$

และทำการจำแนกเว็บเพจที่เหลือด้วยฟังก์ชันในการจำแนกด้วยฟังก์ชัน $\varphi(v)$

สำหรับผลการทดลองของอัลกอริทึมทรานส์ดักทิฟสามารถให้ค่าความแม่นยำสูงในระดับค่าเรียกคืนที่ต่ำและมีประสิทธิภาพในการทำงานที่ดีกว่าอัลกอริทึมแอนตี้ทรัสต์

บทที่ 3

การตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ

งานวิจัยนี้มีจุดประสงค์เพื่อพัฒนาไวยากรณ์กราฟซึ่งเป็นตัวแบบทางภาษาที่มีความสามารถในการอธิบายและตรวจสอบความเป็นสมาชิกของภาษามาประยุกต์ใช้ในการนำเสนอโครงสร้างของลิงก์ฟาร์มและตรวจจับลิงก์ฟาร์มโดยมีแนวทางปรับปรุงอัลกอริทึมการแจงส่วนของไวยากรณ์กราฟเดิมที่มีข้อจำกัดคือไม่สามารถแจงส่วนกับข้อมูลเว็บกราฟได้โดยตรง ไวยากรณ์กราฟที่ใช้ตรวจจับลิงก์ฟาร์มนั้นจะใช้ผู้เชี่ยวชาญในการสร้างไวยากรณ์กราฟจากตัวแบบจำลองของลิงก์ฟาร์มที่นำเสนอไว้ในงานวิจัยอื่น เมื่อได้ไวยากรณ์กราฟลิงก์ฟาร์มในรูปทั่วไปแล้ว จึงใช้การวิเคราะห์เพิ่มเติมเพื่อสร้างไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์ม ซึ่งจำเป็นจะต้องทำการแจงส่วนเทียบกับข้อมูลของโฮสติดฉลากที่เป็นข้อมูลสอนเพื่อหาค่าจำนวนการใช้โปรดักชันซึ่งเป็นลักษณะเฉพาะของลิงก์ฟาร์ม เพื่อใช้ในการแยกแยะกลุ่มโฮสที่เป็นสแปมกับโฮสปกติ ผลลัพธ์ที่ได้จะอยู่ในรูปของกฎตรรกศาสตร์และสามารถนำมาใช้ทดสอบกับชุดข้อมูลทดสอบโดยใช้อัลกอริทึมตรวจจับลิงก์ฟาร์มด้วยไวยากรณ์กราฟต่อไป เนื่องจากคะแนนเพจแรงค์มีอิทธิพลต่อลำดับการแสดงผลลัพธ์ของการค้นคืนเว็บเพจทำให้เว็บเพจที่มีคะแนนเพจแรงค์สูงมีโอกาสแสดงผลลัพธ์ของการค้นคืนในลำดับต้นๆ หรือหน้าแรก และจากสมมติฐานที่ว่าเว็บเพจปกติมักจะไม่ชี้ไปหาเว็บเพจที่เป็นลิงก์ฟาร์ม ทำให้ความสำคัญของข้อมูลที่มาทดสอบแตกต่างกัน ดังนั้นงานวิจัยนี้จึงทำการแบ่งระดับการตรวจจับ (Step of detection) ออกเป็น 20 ระดับซึ่งได้จากการออกแบบการทดลองในบทที่ 4 เป้าหมายเพื่อตรวจจับลิงก์ฟาร์มในกลุ่มที่มีความสำคัญให้ได้ค่าความแม่นยำที่สูง ดังนั้นจากที่กล่าวข้างต้นนี้ทำให้สามารถเสนอแนวทางใหม่ในการแก้ปัญหาในเครือข่ายเวิลด์ไวด์เว็บโดยใช้ความรู้ด้านไวยากรณ์กราฟ

3.1 บทกล่าวนำ

ปัจจุบันงานวิจัยที่สนใจในการศึกษาการตรวจจับลิงก์ฟาร์มแบ่งแนวคิดออกเป็นกลุ่มงานวิจัยที่ตรวจสอบโครงสร้างลิงก์ฟาร์มที่มีสมบัติเชื่อมโยงกันอย่างหนาแน่น กลุ่มงานวิจัยที่ศึกษาการคำนวณคะแนนความน่าจะเป็นลิงก์ฟาร์มของเว็บกราฟทั้งหมด และกลุ่มงานวิจัยในการแก้ปัญหาตรวจจับลิงก์ฟาร์มโดยใช้หลักการเรียนรู้ของเครื่อง ปัญหาของแนวคิดเดิมคือในการตรวจจับโครงสร้างของลิงก์ฟาร์มที่อยู่กันอย่างหนาแน่นนั้น บางครั้งกลุ่มเว็บดังกล่าวอาจไม่ใช่ลิงก์ฟาร์มเพราะอาจเป็นกลุ่มเว็บชุมชนต่างๆ หรือโครงสร้างของเว็บเพจปกติที่มีโครงสร้างคล้ายคลึงกับลิงก์ฟาร์มโดยบังเอิญ ส่วนกลุ่มงานวิจัยที่ใช้แบบจำลองการเดินสุ่มโดยการคำนวณคะแนนความน่าจะเป็นทั้งเว็บกราฟจำนวนหลายๆ รอบจนค่าคะแนนลู่เข้าสู่ค่าที่ยอมรับได้นั้นจะต้องมีการกำหนดเซตของเว็บเพจที่เป็นสแปมเริ่มต้นก่อนจากการตัดสินใจของผู้เชี่ยวชาญจึงจะสามารถให้ผลลัพธ์ที่ถูกต้อง ซึ่งวิธีการนี้สิ้นเปลืองทรัพยากรในการคำนวณเพราะข้อมูลเว็บ

กราฟในระดับเว็บเพจนั้นมีขนาดใหญ่มากและจำเป็นต้องคำนวณหลายรอบจึงจะได้ผลลัพธ์ที่น่าพอใจ และกลุ่มงานวิจัยที่ใช้การแก้ปัญหาการเรียนรู้ของเครื่องนั้นมีการใช้ตัวอย่างสอนที่กำหนดโดยผู้เชี่ยวชาญจำนวนมากจึงจะได้ผลดีในการเรียนรู้

จากกลุ่มงานวิจัยที่กล่าวมาข้างต้นนี้เป็นแนวคิดที่พยายามตอบปัญหาการตรวจจับลิงก์ฟาร์มให้ถูกต้องเพียงอย่างเดียวโดยไม่ได้คำนึงถึงโครงสร้างลิงก์ฟาร์มที่ตรวจจับนั้นมีลักษณะและคุณสมบัติอย่างไร และลิงก์ฟาร์มที่ตรวจจับได้นั้นไม่มีความแตกต่างกันในเรื่องของความสำคัญในคะแนนเพจแรงค์ของลิงก์ฟาร์มที่ตรวจจับได้ ดังนั้นผู้วิจัยจึงมีแนวคิดในการพัฒนาไวยากรณ์กราฟมาใช้อธิบายตัวแบบของลิงก์ฟาร์ม และพัฒนาไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์ม ซึ่งในกระบวนการตรวจจับลิงก์ฟาร์มเริ่มต้นจากการศึกษาถึงตัวแบบลิงก์ฟาร์มและกลุ่มของเว็บเพจอื่นๆ ที่เกี่ยวข้อง โดยพิจารณาโครงสร้างของเว็บเพจทั้ง 3 กลุ่มคือ กลุ่มเว็บเพจที่เข้าถึงได้ กลุ่มเว็บเพจที่เป็นลิงก์ฟาร์ม และกลุ่มเว็บเพจภายนอกที่เป็นเอาท์ลิงก์ของลิงก์ฟาร์ม โดยโครงสร้างทั้ง 3 กลุ่มนี้มีลักษณะโครงสร้างที่มีการถ่ายเทคะแนนเพจแรงค์ให้กับเว็บเพจเป้าหมายเพิ่มสูงขึ้น จากนั้นจะนำเสนอให้อยู่ในรูปไวยากรณ์กราฟโดยวิธีการแปลงลิงก์ฟาร์มด้วยไวยากรณ์กราฟ (graph grammar induction) ซึ่งจะได้ไวยากรณ์กราฟลิงก์ฟาร์มที่สามารถนำเสนอลิงก์ฟาร์มได้ในรูปทั่วไปได้ด้วยกฎที่เรียกว่า โปรดักชัน โดยโปรดักชันบางตัวมีลักษณะพิเศษคือสามารถใช้แบบวนซ้ำได้

จากนั้นมีการนำเสนอไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มโดยแจงส่วนกับชุดข้อมูลสอนทั้งในเว็บกราฟระดับโฮสและเว็บเพจ เมื่อพิจารณาข้อมูลสอนนั้นเป็นโฮสและเว็บเพจเป้าหมาย และใช้การนับจำนวนของการใช้โปรดักชันมาพิจารณาความสัมพันธ์ของการใช้โปรดักชันต่างๆ จากกฎตรรกศาสตร์ที่สร้างขึ้นเพื่อตัดสินใจว่าโฮสนั้นเป็นสแปมหรือปกติ กระบวนการสุดท้ายเป็นการสร้างอัลกอริทึมตรวจจับลิงก์ฟาร์มด้วยไวยากรณ์กราฟกับข้อมูลเว็บกราฟโดยแจงส่วนกับชุดข้อมูลทดสอบเมื่อให้ข้อมูลทดสอบนั้นเป็นโฮสและเว็บเพจเป้าหมายและทำการนับจำนวนการใช้โปรดักชันเพื่อทดสอบกับกฎตรรกศาสตร์ที่ได้จากข้อมูลสอนโดยมีการกำหนดข้อมูลทดสอบออกเป็น 20 ระดับการตรวจจับเพื่อจัดลำดับความสำคัญของข้อมูล

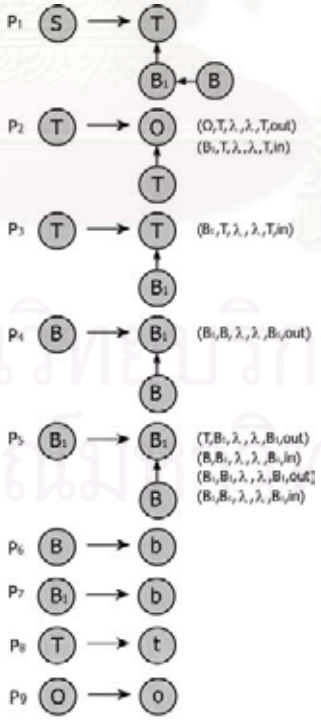
จากที่กล่าวมาข้างต้นทำให้ผู้วิจัยสนใจการพัฒนาไวยากรณ์กราฟในการนำเสนอลิงก์ฟาร์ม การค้นหาไวยากรณ์กราฟที่เหมาะสมในการตรวจจับลิงก์ฟาร์ม และการนำเสนออัลกอริทึมการตรวจจับลิงก์ฟาร์มด้วยไวยากรณ์กราฟ กระบวนการทั้งหมดนี้สามารถแบ่งเป็นขั้นตอนในการพิจารณาอย่างละเอียดได้ดังนี้

1. การนำเสนอรูปแบบจำลองลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ
2. ไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์ม
3. อัลกอริทึมการตรวจจับลิงก์ฟาร์ม
4. การวัดและทดสอบประสิทธิภาพการทำงาน

3.2 การนำเสนอแบบจำลองลิงก์ฟาร์มด้วยไวยากรณ์กราฟ (link farm representation using graph grammar)

การสร้างไวยากรณ์กราฟสำหรับลิงก์ฟาร์มใดๆ มีกระบวนการพิจารณาดังนี้ โครงสร้างย่อยที่ซ้ำกันในเว็ปกราฟจะถูกแทนที่ด้วยอักขระที่ไม่ใช่โหนดปลายทางจนกระทั่งลิงก์ฟาร์มทั้งหมดแทนที่ด้วยอักขระที่ไม่ใช่โหนดปลายทางเพียงโหนดเดียว ดังนั้นตัวแบบลิงก์ฟาร์มทั่วไปดังรูปที่ 2.2 สามารถนำเสนอด้วยไวยากรณ์กราฟตามกระบวนการคือ เว็บเพจเป้าหมายในลิงก์ฟาร์มใดๆ จะถูกเปลี่ยนให้เป็นอักขระ T และเนื่องจากจะต้องมีบูซเพจในการเพิ่มคะแนนให้กับเว็บเพจเป้าหมายซึ่งมีอยู่จำนวนหนึ่ง หรืออาจจะมีลิงก์จากเว็บที่เข้าถึงได้ชี้เข้าหาเว็บเพจเป้าหมายในระยะทางเท่ากับ 1 ไวยากรณ์กราฟจะเปลี่ยนเว็บเพจเหล่านี้ให้เป็น B_1 นอกจากนั้นอาจจะมีบูซเพจหรือเว็บที่เข้าถึงได้ชี้ไปยังบูซเพจซึ่งชี้เข้าหาเว็บเพจเป้าหมายในระยะทางมากกว่า 1 ดังนั้นเว็บเพจนี้จะถูกเปลี่ยนให้เป็น B ซึ่งจำนวนของเว็บเพจจะถูกสร้างขึ้นแบบวนซ้ำหรือสร้างเว็บเพจที่ชี้ไปหาเว็บเพจเป้าหมายด้วยระยะทางที่เพิ่มขึ้น ดังนั้นไวยากรณ์กราฟที่ได้จะแสดงไว้ในทฤษฎีบทที่ 3.1

ทฤษฎีบทที่ 3.1 *ตัวแบบลิงก์ฟาร์มที่นำเสนอ (รูปที่ 2.2) สามารถแปลงให้เป็นไวยากรณ์กราฟลิงก์ฟาร์มโดยที่* $\Sigma_n = \{S, B_1, B, O, T\}$, $\Sigma_t = \{t, b, o\}$, $\Gamma_t = \Gamma_n = \{\lambda\}$, $S = \{S\}$ *และ P คือเซตของ 9 โปรดักชันที่มีกฎฝังตัวดังรูปที่ 3.1*

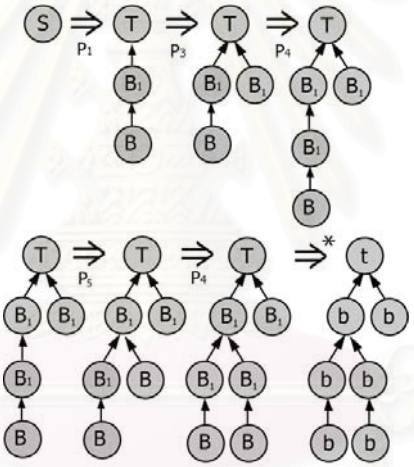


รูปที่ 3.1 โปรดักชันแสดงไวยากรณ์กราฟลิงก์ฟาร์ม

พิสูจน์ จะแสดงว่าไวยากรณ์กราฟลิงก์ฟาร์มที่เสนอ (รูปที่ 2.2) สามารถอธิบายตัวแบบลิงก์ฟาร์มได้และจะเสนอความสัมพันธ์ระหว่างโปรดักชันและโครงสร้างของลิงก์ฟาร์ม

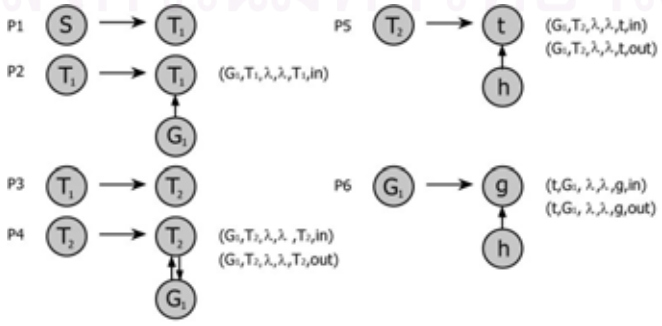
ลิงก์ฟาร์มใดๆ จะต้องประกอบด้วยเว็บเพจเป้าหมายอย่างน้อยหนึ่งเว็บเพจซึ่งนำเสนอด้วยโปรดักชัน P_1 ซึ่งเว็บเพจเป้าหมายนั่นก็คือโหนด T ถูกสร้างขึ้นแล้วเช่นเดียวกัน ขณะเดียวกันผู้สร้างลิงก์ฟาร์มจะต้องสร้างจำนวนบูซเพจจำนวนหนึ่งซึ่งมีลิงก์ชี้ไปยังเว็บเพจเป้าหมายเพื่อทำการเพิ่มคะแนนเพจแรงค์ให้เว็บเพจเป้าหมาย ดังนั้นทุกๆ บูซเพจนั่นก็คือโหนด B สามารถอธิบายได้ด้วยโปรดักชัน P_3 และ P_5 เนื่องจากบูซเพจหรือเว็บเพจที่เข้าถึงได้สามารถเข้าถึงเว็บเพจเป้าหมายได้ในระยะทางมากกว่าหรือเท่ากับ 1 ดังนั้นโปรดักชัน P_4 จึงสามารถที่จะใช้เพื่อเพิ่มระยะทางของบูซเพจหรือเว็บเพจที่เข้าถึงได้ ส่วนโปรดักชันที่เหลือนั้นจะใช้ในการเปลี่ยนอักขระให้เป็นอักขระปลายทาง ■

ตัวอย่างที่ 3.1 *ลิงก์ฟาร์มสามารถแสดงได้โดยไวยากรณ์กราฟซึ่งมีลำดับการแปลงดังรูปที่ 3.2 ซึ่งมีบูซเพจที่ชี้ไปยังเว็บเพจเป้าหมาย 3 ระดับรวมทั้งหมด 6 บูซเพจ* □



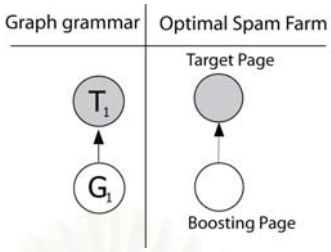
รูปที่ 3.2 ลำดับการแปลงของไวยากรณ์กราฟลิงก์ฟาร์ม

ทฤษฎีบทที่ 3.2 *รูปแบบออพติมอลสแปมฟาร์มจากทฤษฎีบทที่ 2.1 ทั้งสามข้อนั้นสามารถแปลงให้อยู่ในรูปไวยากรณ์กราฟออพติมอลสแปมฟาร์มโดยที่* $\Sigma_n = \{T_1, T_2, G_1\}$, $\Sigma_t = \{t, g, h\}$, $\Gamma_t = \Gamma_n = \{\lambda\}$, $S = \{S\}$ *และ* P *คือเซตของ 6 โปรดักชันที่มีกฎฝังตัวดังรูปที่ 3.3*



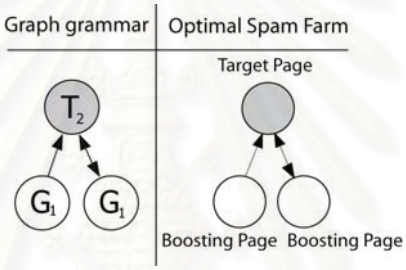
รูปที่ 3.3 โปรดักชันแสดงไวยากรณ์กราฟออพติมอลสแปมฟาร์ม

พิสูจน์ การสร้างไวยากรณ์กราฟตัวแบบของออพติมอลสแปมฟาร์มในรูปที่ 3.3 จะเริ่มต้นจากการใช้โปรดักชันที่ 1 และถ้าตามด้วยใช้โปรดักชันที่ 2 เมื่อจบการสร้างลำดับการแปลงแล้วจะเห็นว่ามีบูซเพจชี้ไปหาเว็บเพจเป้าหมายเพียงอย่างเดียวเท่านั้นดังแสดงในรูปที่ 3.4



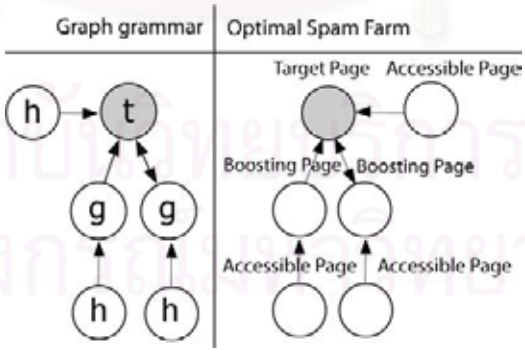
รูปที่ 3.4 แสดงกราฟเมื่อใช้โปรดักชันที่ 2 ของไวยากรณ์กราฟออพติมอลสแปมฟาร์ม

เมื่อมีการใช้โปรดักชันที่ 3 และตามด้วยโปรดักชันที่ 4 จะมีเว็บเพจเป้าหมายที่ชี้ไปหาบูซเพจด้วยดังแสดงในรูปที่ 3.5



รูปที่ 3.5 แสดงกราฟเมื่อใช้โปรดักชันที่ 4 ของไวยากรณ์กราฟออพติมอลสแปมฟาร์ม

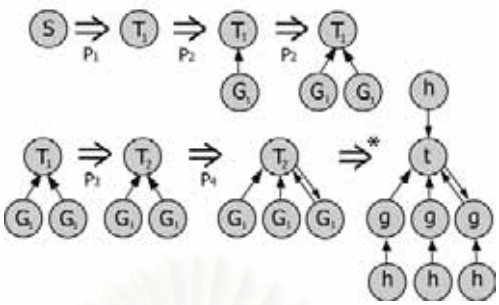
และท้ายสุดจะต้องมีการบังคับใช้โปรดักชันที่ 5 และ 6 เป็นการสิ้นสุดการสร้างลำดับการแปลงโดยเป็นการบังคับว่าต้องมีเว็บภายนอกชี้เข้าหาเว็บเพจเป้าหมายและทุกๆ บูซเพจ ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 แสดงกราฟเมื่อใช้โปรดักชันที่ 5 และ 6 ของไวยากรณ์กราฟออพติมอลสแปมฟาร์ม

ดังนั้นแสดงว่าไวยากรณ์กราฟนั้นสามารถสร้างลิงก์ฟาร์มที่มีสมบัติของออพติมอลสแปมฟาร์มครบทั้ง 3 ข้อตามทฤษฎีบทที่ 2.1 ในทางกลับกัน ออพติมอลสแปมฟาร์มก็สามารถสร้างได้จากโปรดักชันทั้ง 6 โปรดักชันเช่นเดียวกัน ■

ตัวอย่างที่ 3.2 *ออพติมอลสแปมฟาร์มสามารถแสดงได้โดยไวยากรณ์กราฟซึ่งมีลำดับการแปลงดังรูปที่ 3.7 ซึ่งมีบูซเพจที่ชี้ไปยังเว็บเพจเป้าหมาย 3 เว็บเพจ* □



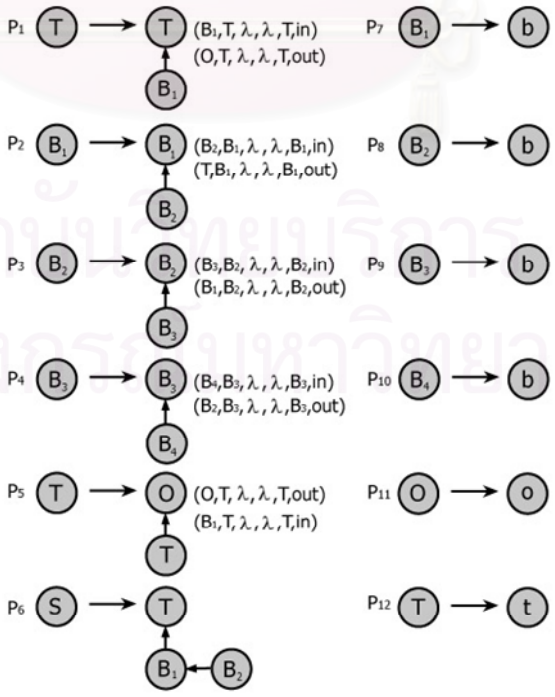
รูปที่ 3.7 ลำดับการแปลงของไวยากรณ์กราฟออพติมอลสแปมฟาร์ม

3.3 ไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์ม

จากไวยากรณ์ลิงก์ฟาร์มข้างต้นเป็นไวยากรณ์กราฟที่ใช้สำหรับนำเสนอลิงก์ฟาร์มในรูปทั่วไป ดังนั้นจึงมีการศึกษาถึงโปรดักชันต่าง ๆ และจำนวนการใช้โปรดักชันของไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มในการตรวจสอบด้วยกฎตรรกศาสตร์ที่เหมาะสม

3.3.1 ไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์ม

ไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มเป็นไวยากรณ์ที่พัฒนาจากไวยากรณ์ลิงก์ฟาร์มซึ่งสนใจบูซเพจหรือกลุ่มเว็บที่เข้าถึงได้ในระยะทางน้อยกว่าหรือเท่ากับ 4 ซึ่งมาจากผลการทดลองในกระบวนการเลือกโปรดักชันที่เหมาะสม โดยที่ $\Sigma_n = \{S, B_1, B_2, B_3, B_4, O, T\}$, $\Sigma_t = \{t, b, o\}$, $\Gamma_t = \Gamma_n = \{\lambda\}$, $S = \{S\}$ และ P คือเซตของ 12 โปรดักชันที่มีกฎฝังตัวดังรูปที่ 3.8



รูปที่ 3.8 ไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์ม

3.3.2 การเลือกโปรดักชันในการตรวจจับลิงก์ฟาร์มและจำนวนการใช้ โปรดักชัน (ค่าพารามิเตอร์) ที่เหมาะสมในการสร้างกฎตรรกศาสตร์

เนื่องจากไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มนั้นสามารถแจงส่วนสำเร็จได้ทั้งเว็บเพจที่เป็นลิงก์ฟาร์มและเว็บเพจปกติ ดังนั้นจำนวนการใช้โปรดักชันบางโปรดักชันจึงเป็นประโยชน์ในการแยกความแตกต่างระหว่างลิงก์ฟาร์มและเว็บเพจปกติ ดังนั้นจากไวยากรณ์กราฟที่พัฒนาขึ้นจำเป็นจะต้องทำการแจงส่วนในข้อมูลสอนเพื่อทำการเลือกโปรดักชัน และค่าพารามิเตอร์ (parameter) ที่เหมาะสมในการสร้างกฎตรรกศาสตร์ร่วมกัน

วิธีการเลือกโปรดักชันในการตรวจจับลิงก์ฟาร์ม

1. พิจารณาโฮสที่เป็นชุดสอน และนำมาแจงส่วนด้วยไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์ม กำหนดให้โฮสนั้นเป็นโฮสเป้าหมายสำหรับการแจงส่วนด้วยข้อมูลเว็บกราฟระดับโฮสและเว็บเพจที่มีคะแนนเพจแรงค์มากที่สุดในโฮสนั้นเป็นเว็บเพจเป้าหมายสำหรับการแจงส่วนด้วยข้อมูลเว็บกราฟระดับเว็บเพจ เมื่อแจงส่วนสำเร็จทำการเก็บค่าจำนวนการใช้โปรดักชันต่างๆไว้
2. สร้างกราฟระหว่างจำนวนการใช้โปรดักชันต่างๆ กับจำนวนโฮสที่ใช้จำนวนโปรดักชันนั้น โดยแยกเส้นกราฟระหว่างโฮสที่เป็นสแปมกับโฮสที่เป็นปกติออกจากกัน
3. เลือกโปรดักชันที่สามารถแยกแยะระหว่างเส้นกราฟของโฮสที่เป็นสแปมและโฮสปกติได้ดี ซึ่งนั่นคือโปรดักชันที่มีความสามารถในการตรวจจับลิงก์ฟาร์ม

วิธีการเลือกค่าพารามิเตอร์ที่เหมาะสมในการสร้างกฎตรรกศาสตร์

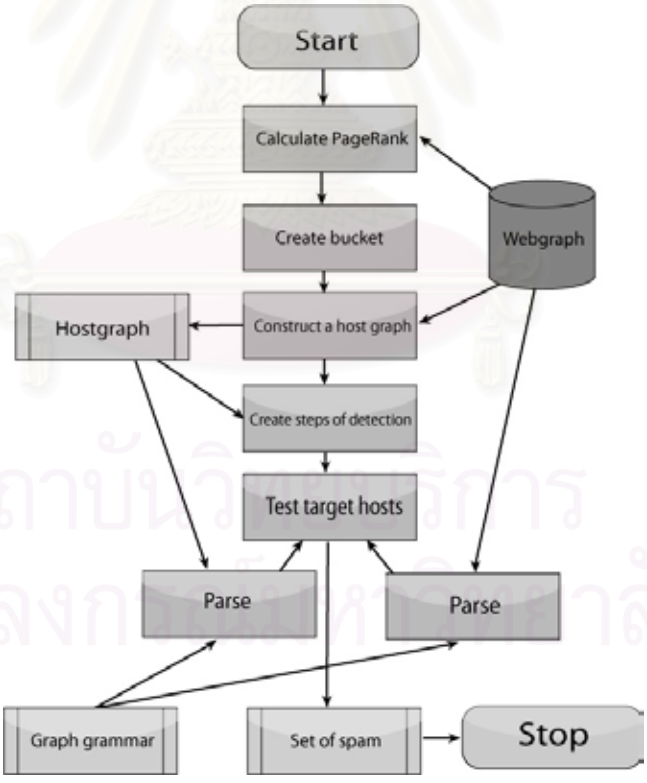
1. เลือกจำนวนการใช้โปรดักชันของโปรดักชันต่างๆ ที่เหมาะสมจากวิธีการเลือกโปรดักชันในการตรวจจับลิงก์ฟาร์มมาเป็นพารามิเตอร์ในการทดสอบกับชุดข้อมูลสอนด้วยกฎตรรกศาสตร์ที่สร้างขึ้น
2. ปรับพารามิเตอร์ที่ใช้สำหรับอัลกอริทึมทดสอบลิงก์ฟาร์มให้ได้ค่าเรียกคืนเป็น 0.1-1.0 จำนวน 10 ระดับค่าเรียกคืน
3. ทำการเลือกชุดพารามิเตอร์ที่มีค่าความแม่นยำที่สุดในแต่ละระดับค่าเรียกคืน

3.4 อัลกอริทึมการตรวจจับลิงก์ฟาร์ม

อัลกอริทึมการตรวจจับลิงก์ฟาร์มจะใช้วิธีนับจำนวนการใช้โปรดักชันจากการแจงส่วนของไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มกับข้อมูลเว็บกราฟในระดับโฮสและเว็บเพจ โดยตัดสินใจร่วมกันจากกฎตรรกศาสตร์ และค่าพารามิเตอร์ที่ได้มาจากข้อมูลชุดสอน นอกจากนั้นอัลกอริทึมจะมีการจัดลำดับความสำคัญในการทดสอบตัวอย่าง โดยแบ่งชุดข้อมูลที่เป็นชุดข้อมูลทดสอบออกเป็น 20 ระดับการตรวจจับ ซึ่งอัลกอริทึมการตรวจจับลิงก์ฟาร์มนั้นการทำงานจะประกอบไปด้วยอัลกอริทึมย่อยอีก 5 อัลกอริทึมโดยที่รายละเอียดการทำงานของอัลกอริทึมต่างๆ นั้นสามารถแสดงได้ดังนี้

3.4.1 อัลกอริทึมการตรวจจับลิงก์ฟาร์ม

อัลกอริทึมการตรวจจับลิงก์ฟาร์มเป็นอัลกอริทึมหลักในการทำงานของกระบวนการตรวจจับลิงก์ฟาร์ม โดยเรียกการทำงานของอัลกอริทึมอื่นๆ ตามลำดับและสามารถแสดงผังงานในรูปที่ 3.9 และรหัสเทียมได้ดังนี้



รูปที่ 3.9 ผังงานแสดงอัลกอริทึมการตรวจจับลิงก์ฟาร์ม

อัลกอริทึมที่ 3.1 อัลกอริทึมการตรวจจับลิงก์ฟาร์ม

Link farm detection algorithm

Input : web graph, training set, test set, graph grammar, i (step of detection), parameter

Output : set of spam

1: **Begin**

2: **PageRank**(web graph, τ, c)

3: **Bucket**(web graph, pagerank)

4: host graph ← **Construct a host graph from web graph**

5: **Step of detection (**host graph, training set, test set, bucket**)**

6: target host set ← **All hosts in step of detection** i

7: target page set ← **Find a maximum PageRank page in target host**

8: **For each target host**

9: **Spamic** ← **Test(**target host,target page,host graph,web graph,graph grammar,parameter**)**

10: **End for loop**

11: Set of spam ← **Group all target hosts that spamic >0.5**

12: **End**

จากอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟจะนำเข้าคือเว็บกราฟ ชุดข้อมูลสอน ชุดข้อมูลทดสอบ ไวยากรณ์กราฟ ระดับการตรวจจับและค่าพารามิเตอร์ ซึ่งเป็นจำนวนการใช้โปรดักชันที่ได้จากชุดข้อมูลสอน และจะให้ผลลัพธ์ออกมาเป็นเซตของโฮสที่เป็นสแปม ซึ่งแต่ละขั้นตอนสามารถอธิบายการทำงานอย่างละเอียดได้ดังนี้

บรรทัดที่ 2 : คำนวณคะแนนเพจแรงค์จากเว็บกราฟ
บรรทัดที่ 3 : แบ่งถังข้อมูลของโฮสออกเป็น 10 ถัง
บรรทัดที่ 4 : สร้างโฮสกราฟจากเว็บกราฟระดับเว็บเพจ
บรรทัดที่ 5 : แบ่งระดับการตรวจจับออกเป็น 20 ระดับ
บรรทัดที่ 6 : ให้โฮสทั้งหมดในระดับการตรวจจับที่รับเข้าเป็นโฮสเป้าหมาย
บรรทัดที่ 7 : ให้เว็บเพจที่มีคะแนนเพจแรงค์สูงที่สุดภายในโฮสเป็นเว็บเพจเป้าหมาย
บรรทัดที่ 8-10 : ทดสอบความเป็นสแปมในแต่โฮสเป้าหมายและเว็บเพจเป้าหมาย
บรรทัดที่ 11 : ให้ผลลัพธ์เป็นสแปมเมื่อมีความน่าจะเป็นมากกว่า 0.5

3.4.2 อัลกอริทึมคำนวณคะแนนเพจแรงค์

การคำนวณคะแนนเพจแรงค์สามารถคำนวณได้จากอัลกอริทึมที่ 2.1

3.4.3 อัลกอริทึมแบ่งถังข้อมูล

ถังข้อมูลจะมีประโยชน์ในการจัดกลุ่มระดับคะแนนเพจแรงค์ โดยแบ่งทั้งหมดออกเป็น 10 ถัง ซึ่งแต่ละถังจะมีข้อมูลโฮสจำนวนเท่าๆ กัน โดยมีการเรียงโฮสตามคะแนนเพจแรงค์ที่มากที่สุดของเว็บเพจภายในโฮสจากมากไปหาน้อย ดังแสดงรหัสเทียมดังนี้

อัลกอริทึมที่ 3.2 อัลกอริทึมแบ่งถังข้อมูล

Bucket algorithm

Input : web graph, PageRank

Output : bucket

1: **Begin**

2: Number of host ← **Count all hosts in web graph**

3: n ← $\lceil$ Number of hosts/10 $\rceil$

4: **For** j=1 **to** Number of host

5: Max PR of host[j] ← **Select maximum PageRank in host[j]**

6: **End for loop**

7: **host** ← **Sort descending Max PR of host[j]**

8: **For** i=1 **to** 10

9: bucket[i] = $\bigcup_{k=(i-1)n+1}^{i\times n}$ host[k]

10: **End for loop**

11: **End**

จากอัลกอริทึมแบ่งถังข้อมูลจะนำเข้าข้อมูลเว็บกราฟ และคะแนนเพจแรงค์ ซึ่งจะให้ผลลัพธ์ออกมาเป็นถังข้อมูลของโฮสจำนวน 10 ถัง ซึ่งแต่ละขั้นตอนสามารถอธิบายการทำงานอย่างละเอียดได้ดังนี้

บรรทัดที่ 2 : นับจำนวนโฮสทั้งหมดจากชื่อเว็บเพจในเว็บกราฟ

บรรทัดที่ 3 : แบ่งจำนวนโฮสในแต่ละถังให้เท่าๆ กันจำนวน 10 ถัง

บรรทัดที่ 4-6 : หาคะแนนเพจแรงค์ที่มีค่ามากที่สุดในแต่ละโฮส

บรรทัดที่ 7 : จัดเรียงโฮสตามคะแนนเพจแรงค์ของเว็บเพจในโฮสนั้นจากมากไปหาน้อย

บรรทัดที่ 8-10 : แบ่งโฮสที่จัดเรียงคะแนนแล้วออกเป็น 10 ถังเท่าๆ กัน

3.4.4 อัลกอริทึมแบ่งระดับการตรวจจับ

การแบ่งชุดข้อมูลทดสอบจำนวน 20 ระดับการตรวจจับเพื่อจัดลำดับความสำคัญในการตรวจจับลิงก์ฟาร์ม จากสมมติฐานที่ว่าเว็บเพจปกติมักจะไม่ชี้ไปหาเว็บเพจเป็นลิงก์ฟาร์ม ดังนั้นระดับการตรวจจับระดับแรกจะประกอบด้วยโฮสที่มีเอาท์ลิงก์เป็นโฮสในชุดข้อมูลสอนที่เราทราบว่าเป็นสแปม และระดับการตรวจจับระดับท้าย จะเป็นโฮสที่มีเว็บเพจเป้าหมายที่มีคะแนนเพจแรงค์ต่ำ ดังนั้นข้อมูลทดสอบที่แบ่งไว้ในแต่ละระดับการตรวจจับนั้นแสดงได้ในตารางที่ 3.1

ตารางที่ 3.1 แสดงถึงข้อมูลทดสอบที่แบ่งไว้ตามแต่ละระดับการตรวจจับ

ระดับการตรวจจับ	ชุดข้อมูล
1	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปมที่อยู่ในถังที่ 1
2	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปมที่อยู่ในถังที่ 2 และถังที่ 1
3	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปมที่อยู่ในถังที่ 3 ถึงถังที่ 1
4	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปมที่อยู่ในถังที่ 4 ถึงถังที่ 1
5	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปมที่อยู่ในถังที่ 5 ถึงถังที่ 1
6	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปมที่อยู่ในถังที่ 6 ถึงถังที่ 1
7	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปมที่อยู่ในถังที่ 7 ถึงถังที่ 1
8	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปมที่อยู่ในถังที่ 8 ถึงถังที่ 1
9	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปมที่อยู่ในถังที่ 9 ถึงถังที่ 1
10	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม

ระดับการตรวจจับ	ชุดข้อมูล
11	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 1
12	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 2
13	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 3
14	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 4
15	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 5
16	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 6
17	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 7
18	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 8
19	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 9
20	ชุดข้อมูลทดสอบที่มีเอาท์ลิงก์เป็นข้อมูลชุดสอน ซึ่งข้อมูลชุดสอนนั้นเป็นสแปม และชุดข้อมูลทดสอบในถังที่ 10

ชุดข้อมูลทดสอบทั้ง 20 ระดับการตรวจจับสามารถแบ่งได้โดยใช้อัลกอริทึมแบ่งระดับการตรวจจับ ดังแสดงเป็นรหัสเทียมได้ดังนี้

อัลกอริทึมที่ 3.3 อัลกอริทึมแบ่งระดับการตรวจจับ

```
Step of detection algorithm
Input :   host graph, training set, test set, bucket
Output :  Step (step of detection)
1: Begin
2:     For i=1 to 20
3:         If i<=10
4:           Inlink ⟵ inlinks of spam hosts in training set which hosts are in bucket <= i
5:           Step[i]= Inlink ∩ test set
6:         Else
7:           Step[i]= Step[10] ∪ test set which hosts are in bucket <= (i-10)
8:     End if
9:     End for loop
10: End
```

จากอัลกอริทึมแบ่งระดับการตรวจจับจะนำเข้าคือเว็บกราฟ ชุดข้อมูลสอน ชุดข้อมูลทดสอบ และถังข้อมูล ซึ่งจะให้ผลลัพธ์ออกมาเป็นชุดข้อมูลทดสอบจำนวน 20 ชุดเรียกว่าระดับการตรวจจับ ซึ่งแต่ละขั้นตอนสามารถอธิบายการทำงานอย่างละเอียดได้ดังนี้

บรรทัดที่ 2-9 : สร้างระดับการตรวจจับที่ i โดยมีระดับทั้งหมด 20 ระดับ

บรรทัดที่ 3-5 : ตรวจสอบเงื่อนไขให้ 10 ระดับแรกเป็นชุดข้อมูลทดสอบที่เป็นอินลิงก์ของโฮสที่เป็นสแปมในชุดข้อมูลสอนและอยู่ในถังข้อมูลที่ 1 ถึงถังข้อมูลที่ i

บรรทัดที่ 6-7 : ตรวจสอบเงื่อนไขให้ 10 ระดับหลังเป็นชุดข้อมูลทดสอบอยู่ในถังข้อมูลที่ 1 ถึงถังข้อมูลที่ i รวมกับชุดข้อมูลระดับการตรวจจับที่ 10

3.4.5 อัลกอริทึมทดสอบตัวอย่าง

สำหรับชุดตัวอย่างที่เป็นโหนดเป้าหมายในการทดสอบ จะรับข้อมูลซึ่งเป็นเว็บกราฟในระดับโฮส และระดับเว็บเพจ โดยมีไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มในการแจงส่วนเทียบกับเว็บกราฟในระดับโฮสและเว็บเพจตามลำดับ ซึ่งผลลัพธ์ที่ได้คือผลจากการตัดสินใจจากกฎตรรกศาสตร์ที่ได้จากผลการทดลองดังแสดงในตารางที่ 4.1 และค่าพารามิเตอร์จากชุดข้อมูลสอนที่เราเลือกไว้จากผลการทดลอง โดยจะให้คำตอบจากอัลกอริทึมทดสอบตัวอย่างออกมาเป็น 0 ถ้าหากโฮสนั้นเป็นโฮสปกติและคำตอบออกมาเป็น 1 ถ้าหากโฮสนั้นเป็นสแปม

อัลกอริทึมที่ 3.4 อัลกอริทึมทดสอบตัวอย่าง

Test algorithm

Input : target host, target page, host graph, web graph, graph grammar, training set, parameter

Output : spamic

1: **Begin**

2: ($p2_{training}$, $p3_{training}$, $p4_{training}$, $P3_{training}$, $P4_{training}$) = parameter

3: (p1,…, p12) = **parse(**web graph, graph grammar, target page**)**

4: (P1,…, P12) = **parse(**host graph, graph grammar, target host**)**

5: **rule** ← **{(**p2<= $p2_{training}$**) or (**p3<= $p3_{training}$**) or (**p4<= $p4_{training}$**)}**

6: **and {(**P1<= $P1_{training}$**) and (**P2<= $P2_{training}$**)}**

7: **If(**rule is true**)**

8: spamic ← 1

9: **Else**

10: spamic ← 0

11: **End if**

12: **End**

จากอัลกอริทึมทดสอบตัวอย่างจะนำเข้าคือโฮสเป้าหมาย เว็บเพจเป้าหมาย เว็บกราฟระดับเว็บเพจและโฮส ไวยากรณ์กราฟ ชุดข้อมูลสอนและค่าพารามิเตอร์ ซึ่งเป็นจำนวนการใช้โปรดักชัน และจะให้ผลลัพธ์ออกมาว่าโฮสเป็นสแปมหรือไม่ ซึ่งแต่ละขั้นตอนสามารถอธิบายการทำงานอย่างละเอียดได้ดังนี้

บรรทัดที่ 2 : รับค่าพารามิเตอร์ซึ่งเป็นจำนวนการใช้โปรดักชันต่างๆ

บรรทัดที่ 3 : ทำการแจงส่วนไวยากรณ์กราฟบนเว็บกราฟระดับเว็บเพจ

บรรทัดที่ 4 : ทำการแจงส่วนไวยากรณ์กราฟบนเว็บกราฟระดับโฮส

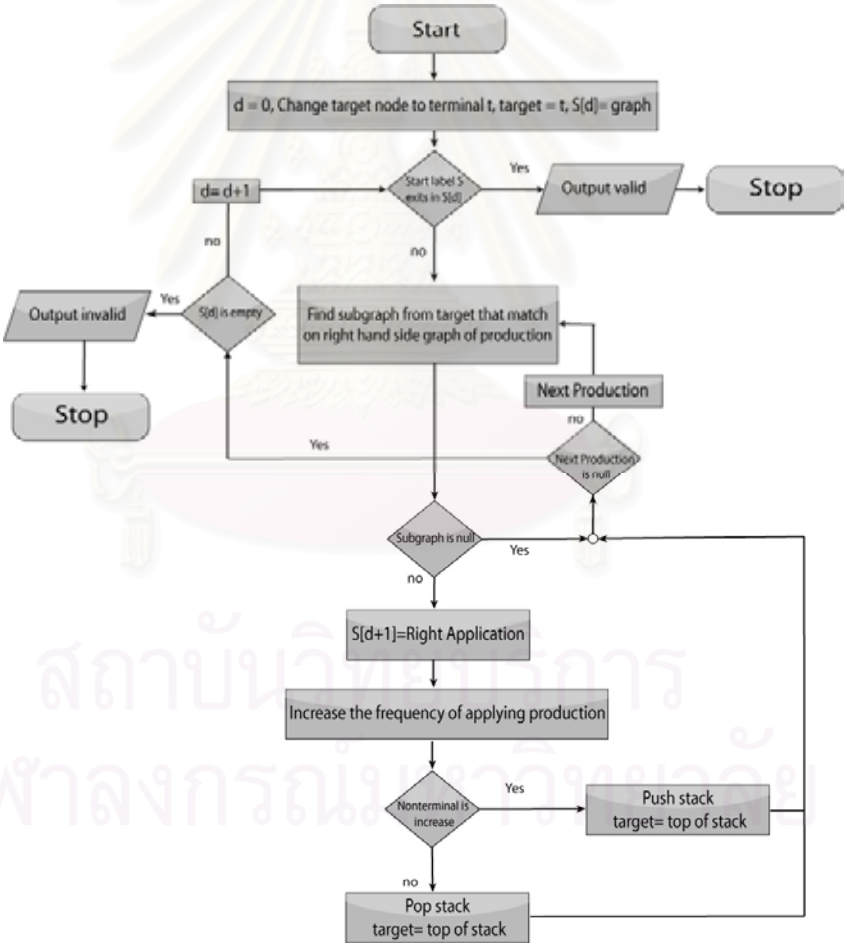
บรรทัดที่ 5 : ทดสอบกับกฎตรรกศาสตร์ว่าเป็นสแปมหรือไม่

บรรทัดที่ 7-8 : ตรวจสอบเงื่อนไขจากกฎตรรกศาสตร์ว่าเป็นสแปม พร้อมกับให้ผลลัพธ์

บรรทัดที่ 9-10 : ตรวจเงื่อนไขจากกฎตรรกศาสตร์ว่าไม่เป็นสแปม พร้อมกับให้ผลลัพธ์

3.4.6 อัลกอริทึมแจงส่วนไวยากรณ์กราฟ

ในอัลกอริทึมการแจงส่วนนี้มีความแตกต่างจากการแจงส่วนของกราฟทั่วๆ ไป เนื่องจากจะต้องทำงานในเว็บกราฟที่มีขนาดใหญ่ และการหาไวยากรณ์กราฟนั้นไม่ใช่ลดรูปกราฟทั้งหมดให้เหลือเพียงโหนดเริ่มต้นเพียงโหนดเดียว แต่เป็นการลดรูปกราฟย่อยในเว็บกราฟให้ได้โหนดเริ่มต้นจึงจะถือว่าแจงส่วนสำเร็จ ส่วนวิธีการค้นหากราฟนั้นจำเป็นจะต้องใช้โหนดเป้าหมายในการเลือกโหนดในการทำงาน อีกประการหนึ่งโหนดปลายทางของไวยากรณ์กราฟนั้นสามารถเปลี่ยนไปเป็นชื่อโหนดใดๆ ในเว็บกราฟก็ได้ ดังนั้นโดยวิธีการจะเริ่มต้นจากการค้นหาโหนดเริ่มต้นในกราฟก่อนแล้วจึงค้นหากราฟที่ตรงกับกราฟทางด้านขวาของโปรดักชัน เมื่อเจอกราฟย่อยที่ตรงกันแล้วทำการประยุกต์ทางด้านขวา เพื่อให้ได้กราฟสุดท้ายเป็นตัวแปรเริ่มต้นเพียงโหนดเดียวจึงจะถือว่าทำการแจงส่วนสำเร็จ ซึ่งแสดงผังงานดังรูปที่ 3.10 และรหัสเทียมของอัลกอริทึมได้ดังนี้



รูปที่ 3.10 ผังงานแสดงอัลกอริทึมแจงส่วนไวยากรณ์กราฟ

อัลกอริทึมที่ 3.5 อัลกอริทึมแจงส่วนไวยากรณ์กราฟ

```
Parse algorithm
Input :  graph, graph grammar, target node
Output :  "Valid" or "Invalid" , Array P (frequency of applying productions)
1: Begin
2:   d ← 0
3:   Change target node to terminal t in graph
4:   target ← t
5:   s[d] ← graph
6:          While(Start label not exits in S[d])
7:              For all production p in graph grammar
8:                  redex ← FindRedex (S[d], production p, target)
9:                  If (redex !=null)
10:                 S[d+1] ← Right application(S[d], production p, redex)
11:                 P ← (the frequency of applying production p) + 1
12:                     IF (There is Non-terminal in S[d+1] but not in S[d])
13:                         Non-terminal ← Non-terminal in S[d+1] but not in S[d]
14:                         Push stack (Non-terminal)
15:                         target ← Top of stack
16:                     End if
17:                     IF (There is Non-terminal in S[d] but not in S[d+1])
18:                         Non-terminal ← Non-terminal in S[d] but not in S[d+1]
19:                         Pop stack
20:                         target ← Top of stack
21:                     End if
22:                 End if
23:             End for loop
24:         If (S[d] is empty)
25:             output("Invalid") and exit
26:          End iF
27:          d ← d+1
28:         End while loop
29: End
```

จากอัลกอริทึมแจงส่วนไวยากรณ์กราฟจะนำเข้าคือกราฟ ไวยากรณ์กราฟ และโหนดเป้าหมาย และจะให้ผลลัพธ์ออกมาเป็นผลการแจงส่วน และจำนวนการใช้โปรดักชัน ถ้าหากทำการแจงส่วนสำเร็จ ซึ่งแต่ละขั้นตอนสามารถอธิบายการทำงานอย่างละเอียดได้ดังนี้

บรรทัดที่ 2 : กำหนดรอบการทำงานเริ่มต้นเป็น 0

บรรทัดที่ 3 : เปลี่ยนโหนดเป้าหมายเป็นอักขระ t ในกราฟนำเข้า

บรรทัดที่ 4 : ให้เป้าหมายเป็นอักขระ t ในการทำงาน

บรรทัดที่ 5 : กำหนดกราฟเริ่มต้นการทำงาน

บรรทัดที่ 6-28 : ตรวจสอบเงื่อนไขการแจงส่วนสำเร็จเมื่อปรากฏโหนดเริ่มต้นในกราฟ

บรรทัดที่ 7-23 : วนลูปทุกๆ จำนวนโปรดักชันของไวยากรณ์กราฟ

บรรทัดที่ 8: ค้นหากราฟทางด้านขวาของโปรดักชันที่ตรงกับกราฟในขณะทำงานโดยเริ่มต้นค้นหาจากเป้าหมายก่อน

บรรทัดที่ 9-22: ตรวจสอบเงื่อนไขว่ามีกราฟที่มีโครงสร้างตรงกันกับกราฟทางด้านขวามือของโปรดักชันหรือไม่

บรรทัดที่ 10: ทำการประยุกต์ทางด้านขวากับกราฟในรอบการทำงานนั้นด้วยกราฟที่มีโครงสร้างตรงกันกับกราฟทางด้านขวามือของโปรดักชัน

บรรทัดที่ 12-16: ตรวจสอบเงื่อนไข ในกรณีที่มีการเพิ่มอักขระที่ไม่ใช่ปลายทางเข้าไปในการประยุกต์ทางด้านขวา ให้เก็บอักขระนั้นเข้ากองซ้อน และเปลี่ยนเป้าหมายเป็นอักขระที่อยู่ชั้นบนสุดของกองซ้อน

บรรทัดที่ 17-21: ตรวจสอบเงื่อนไข ในกรณีที่มีการลดอักขระที่ไม่ใช่ปลายทางเข้าไปในการประยุกต์ทางด้านขวา ให้ถอดกองซ้อนออก และเปลี่ยนเป้าหมายเป็นอักขระที่อยู่ชั้นบนสุดของกองซ้อน

บรรทัดที่ 24-26: ตรวจสอบเงื่อนไข ถ้าหากว่าไม่สามารถทำการประยุกต์ทางด้านขวาได้ และก็ไม่มีกราฟเกิดขึ้นในการทำงานรอบต่อไป ให้ยกเลิกการทำงานและแจงส่วนไม่สำเร็จ

บรรทัดที่ 27: เพิ่มรอบการทำงาน

3.5 การวัดและทดสอบประสิทธิภาพการทำงาน

ในการวัดประสิทธิภาพของการตรวจจับลิงก์ฟาร์มนั้นจะใช้ข้อมูลเว็บกราฟและชุดข้อมูลของโฮสที่มีการติดฉลากว่าเป็นโฮสสแปมหรือโฮสปกติ ซึ่งแบ่งออกเป็นข้อมูลสอนในการหาค่าพารามิเตอร์กับกฎตรรกศาสตร์ และข้อมูลทดสอบซึ่งใช้ตรวจสอบโดยอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ โดยค่าสำหรับวัดประสิทธิภาพการทำงานและวิธีการทดสอบประสิทธิภาพจะแสดงได้ดังนี้

3.5.1 ค่าสำหรับวัดประสิทธิภาพการทำงาน

ค่าความแม่นยำ คือค่าที่ใช้ในการวัดจำนวนของโฮสสแปมที่อัลกอริทึมตรวจพบได้อย่างถูกต้องเมื่อเทียบกับโฮสที่ตรวจจับได้ ซึ่งคำนวณได้ดังนี้

$$\text{ค่าความแม่นยำ} = \frac{\text{จำนวนโฮสสแปมที่ตรวจว่าเป็นสแปมจริง}}{\text{จำนวนโฮส ทั้งหมดที่ตรวจว่าเป็นโฮสสแปม}}$$

ค่าเรียกคืน คือค่าที่ใช้ในการวัดจำนวนของโฮสสแปมที่อัลกอริทึมสามารถตรวจจับได้เทียบกับโฮสสแปมที่มีอยู่ในชุดข้อมูลทั้งหมดซึ่งคำนวณได้ดังนี้

$$\text{ค่าเรียกคืน} = \frac{\text{จำนวนโฮสสแปมที่ตรวจว่าเป็นสแปมจริง}}{\text{จำนวนโฮส ที่เป็นโฮสสแปมทั้งหมด}}$$

3.5.2 วิธีการวัดประสิทธิภาพการทำงานของอัลกอริทึม

1. การเปรียบเทียบประสิทธิภาพทำงานของอัลกอริทึม

ในการเปรียบเทียบประสิทธิภาพของอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟกับอัลกอริทึมการตรวจจับลิงก์ฟาร์มที่ใช้วิธีทรานส์ดักทิฟและแอนตี้ทรัสต์ จะทำการทดสอบตามสภาพแวดล้อมการทดลองเดียวกันและใช้ข้อมูลเว็บกราฟเดียวกัน ซึ่งมีการเปรียบเทียบสองประเด็นคือการเปรียบเทียบประสิทธิภาพในด้านค่าความแม่นยำในระดับสัดส่วนของข้อมูลสอนต่างๆ กัน เมื่อกำหนดให้เรียกคืนเป็น 0.5 โดยเปรียบเทียบสัดส่วนของข้อมูลสอนตั้งแต่ 10% 20% 30% 40% และ 50% จากการสุ่มข้อมูลที่ติดฉลากเพื่อใช้สำหรับการหาค่าพารามิเตอร์กับกฎตรรกศาสตร์ ในการวัดผลเพื่อไม่ให้เกิดความเอนเอียงในการทดสอบจะทำการเก็บค่าเฉลี่ยทั้งหมด 10 รอบซึ่งจะแบ่งข้อมูลที่ติดฉลากออกเป็น 10 ชุดอย่างสุ่ม ซึ่งการทดลองจะใช้ชุดข้อมูลทดสอบที่สลับชุดข้อมูลทีละชุดไปเรื่อยๆ จนครบ 10 รอบ ตามวิธีการตรวจสอบแบบไขว้ (10-Fold cross validation) และนำมาสร้างกราฟระหว่างสัดส่วนข้อมูลสอนและค่าความแม่นยำ

ประเด็นที่สองคือการเปรียบเทียบประสิทธิภาพของอัลกอริทึมในด้านค่าความแม่นยำเมื่อกำหนดให้ค่าเรียกคืนต่างๆ กัน ซึ่งจะวัดผลโดยทำการเก็บค่าเฉลี่ยทั้งหมด 10 รอบกับชุดมูลที่เป็นแบบไขว้ เมื่อกำหนดให้สัดส่วนของข้อมูลสอนเป็น 50% โดยเปรียบเทียบค่าเรียกคืนตั้งแต่ 0.5-1.0 และนำมาสร้างกราฟระหว่างค่าเรียกคืนและค่าความแม่นยำ ซึ่งชุดข้อมูลทดสอบที่ใช้ในการทำการตรวจสอบแบบไขว้นั้นแสดงในตารางที่ 3.2

ตารางที่ 3.2 แสดงข้อมูลทดสอบในแต่ละรอบ

รอบที่	ข้อมูลทดสอบ (ชุดข้อมูล)				
	สัดส่วนข้อมูลสอน 10%	สัดส่วนข้อมูลสอน 20%	สัดส่วนข้อมูลสอน 30%	สัดส่วนข้อมูลสอน 40%	สัดส่วนข้อมูลสอน 50%
1	2,3,4,5,6,7,8,9,10	3,4,5,6,7,8,9,10	4,5,6,7,8,9,10	5,6,7,8,9,10	6,7,8,9,10
2	3,4,5,6,7,8,9,10,1	4,5,6,7,8,9,10,1	5,6,7,8,9,10,1	6,7,8,9,10,1	7,8,9,10,1
3	4,5,6,7,8,9,10,1,2	5,6,7,8,9,10,1,2	6,7,8,9,10,1,2	7,8,9,10,1,2	8,9,10,1,2
4	5,6,7,8,9,10,1,2,3	6,7,8,9,10,1,2,3	7,8,9,10,1,2,3	8,9,10,1,2,3	9,10,1,2,3
5	6,7,8,9,10,1,2,3,4	7,8,9,10,1,2,3,4	8,9,10,1,2,3,4	9,10,1,2,3,4	10,1,2,3,4
6	7,8,9,10,1,2,3,4,5	8,9,10,1,2,3,4,5	9,10,1,2,3,4,5	10,1,2,3,4,5	1,2,3,4,5
7	8,9,10,1,2,3,4,5,6	9,10,1,2,3,4,5,6	10,1,2,3,4,5,6	1,2,3,4,5,6	2,3,4,5,6
8	9,10,1,2,3,4,5,6,7	10,1,2,3,4,5,6,7	1,2,3,4,5,6,7	2,3,4,5,6,7	3,4,5,6,7
9	10,1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	2,3,4,5,6,7,8	3,4,5,6,7,8	4,5,6,7,8
10	1,2,3,4,5,6,7,8,9	2,3,4,5,6,7,8,9	3,4,5,6,7,8,9	4,5,6,7,8,9	5,6,7,8,9

2. การวัดประสิทธิภาพเฉพาะกลุ่มลิงก์ฟาร์มที่มีคะแนนเพจแรงค์ที่สูง

เนื่องจากพบว่าในผลการค้นคืนของโปรแกรมค้นหาบนอินเทอร์เน็ตมีการเรียงลำดับตามคะแนนเพจแรงค์ซึ่งผู้ใช้มักจะสนใจผลลัพธ์ในหน้าแรกเท่านั้น การทดลองเพิ่มเติมจึงทำการวัดประสิทธิภาพของไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มกับอัลกอริทึมทดสอบตัวอย่างโดยการใช้ข้อมูลในถังมาเป็นข้อมูลสอนสำหรับการหาค่าพารามิเตอร์ และข้อมูลทดสอบสำหรับการทดสอบด้วยอัลกอริทึมทดสอบตัวอย่าง เมื่อกำหนดให้สัดส่วนของข้อมูลสอนเป็น 50% จากการสุ่มข้อมูลที่ติดฉลากในถังนั้น

บทที่ 4

ผลการทดลอง

ในผลการทดลองจะอธิบายถึงชุดข้อมูลและเครื่องมือที่ใช้ในการทดลอง ผลในการเลือกโปรดักชันและกฎตรรกศาสตร์ที่เหมาะสม และค่าพารามิเตอร์ต่างๆ ในกระบวนการตรวจจับลิงก์ฟาร์ม เนื่องจากงานวิจัยมีการแบ่งกลุ่มของเว็บเพจตามคะแนนเพจแรงค์ออกเป็น 10 ถัง [1,4,21] และสมมติฐานที่ว่าเว็บเพจปกติมักจะไม่ชี้ไปหาเว็บเพจที่เป็นลิงก์ฟาร์ม ดังนั้นจึงกำหนดให้มีระดับการตรวจจับเป็น 20 ระดับกับข้อมูลที่นำมาทดสอบ และจะแสดงผลในการเปรียบเทียบประสิทธิภาพของอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟกับอัลกอริทึมทรานส์ดักทิฟ และอัลกอริทึมแอนตี้ทรัสต์ ซึ่งเป็นงานวิจัยที่เกี่ยวข้องและมีการทดลองที่ใช้ชุดข้อมูลเว็บกราฟเดียวกัน โดยจะวัดประสิทธิภาพในด้านสัดส่วนข้อมูลสอน ค่าความแม่นยำ และค่าเรียกคืน ทั้งนี้ผลการทดลองจะครอบคลุมถึงประสิทธิภาพในด้านของเวลาการทำงานของอัลกอริทึม ลำดับท้ายสุดผลการทดลองเพิ่มเติมจะแสดงผลการตรวจจับเฉพาะกลุ่มลิงก์ฟาร์มที่มีคะแนนเพจแรงค์ที่สูง ซึ่งรายละเอียดดังที่กล่าวมานั้นแสดงได้ดังนี้

4.1 ชุดข้อมูลและเครื่องมือที่ใช้ในการทดลอง

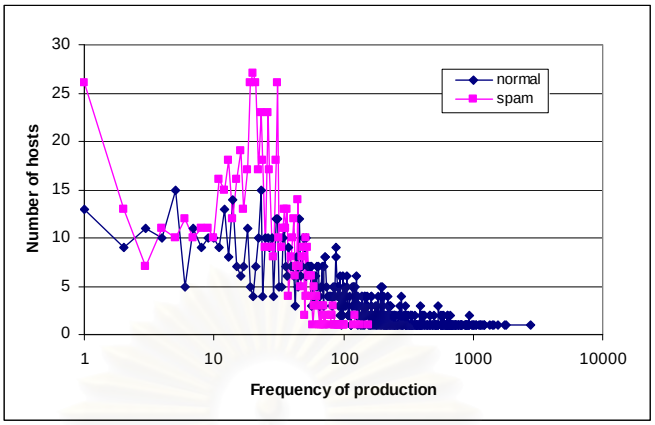
ชุดข้อมูลที่ใช้ในการทดลอง เป็นชุดข้อมูลเว็บกราฟมาตรฐานที่เก็บข้อมูลมาจากโดเมนในประเทศอังกฤษในปี 2006 โดยฝ่ายวิจัยยาฮู [22] (Yahoo Research) โดยข้อมูลทั้งหมดประกอบด้วย 77,741,046 เว็บเพจ 2,965,197,340 ไฮเปอร์ลิงก์ รวมทั้งหมด 11,402 โฮส และมีชุดตัวอย่างที่ติดฉลากโดยใช้อาสาสมัครจำนวน 33 คนในการตรวจสอบหน้าเว็บเพจทั้งหมดเพื่อตัดสินใจว่าโฮสใดเป็นสแปมหรือไม่ โดยใช้ข้อมูลแนะนำจากผู้เชี่ยวชาญประกอบการตรวจสอบ เครื่องมือที่ใช้ในการทดลองคือโปรแกรมเนทบีนส์ (NetBeans IDE) เวอร์ชัน 5.5.1 และชุดไลบรารีเว็บกราฟ 2.0 (Webgraph 2.0) ซึ่งใช้ภาษาจาวาในการพัฒนาโดยใช้เครื่องคอมพิวเตอร์ ระบบปฏิบัติการ Windows XP Service Pack 2 หน่วยประมวลผล Pentium 4 2.6 GHz หน่วยความจำ 760 MB

4.2 โปรดักชันและกฎตรรกศาสตร์ที่เหมาะสมในการตรวจจับลิงก์ฟาร์ม

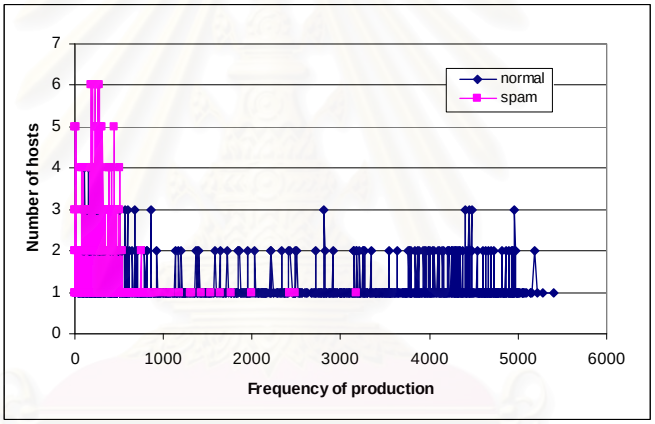
ข้อมูลโฮสที่ติดฉลากจะถูกสุ่มแบ่งออกเป็นข้อมูลสอนและข้อมูลทดสอบจำนวนเท่าๆ กัน เนื่องจากผลการทดลองแจงส่วนกับชุดข้อมูลสอนพบว่าบูซเพจหรือเว็บเพจที่เข้าถึงได้ระยะทางมากกว่า 4 ไม่สามารถแยกแยะระหว่างเส้นกราฟของโฮสสแปมและโฮสปกติได้ดังนั้นจึงนำเสนอไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มดังแสดงในรูปที่ 3.8

จากวิธีการเลือกโปรดักชันในการตรวจจับลิงก์ฟาร์มโดยข้อมูลสอนถูกนำมาใช้ในการหาโปรดักชันที่เหมาะสมในการตรวจจับลิงก์ฟาร์ม ผลการทดลองสามารถนำมาเขียนกราฟ

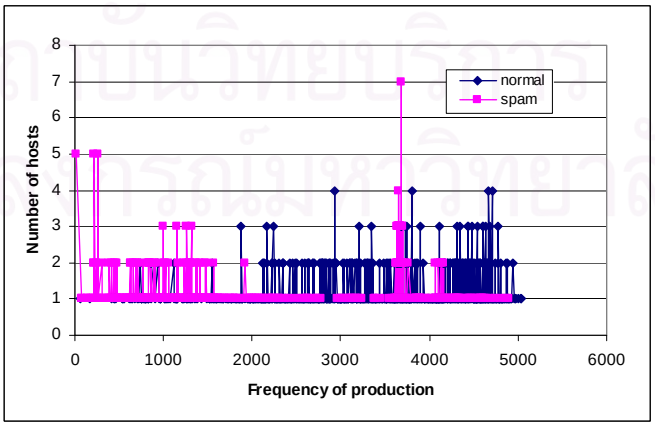
ระหว่างจำนวนการใช้โปรดักชันต่างๆ กับจำนวนโฮสสแปมและโฮสปกติที่ใช้จำนวนโปรดักชันที่แจงส่วนกับข้อมูลเว็บกราฟในระดับโฮสและเว็บเพจ ดังแสดงได้ในรูปที่ 4.1-4.9



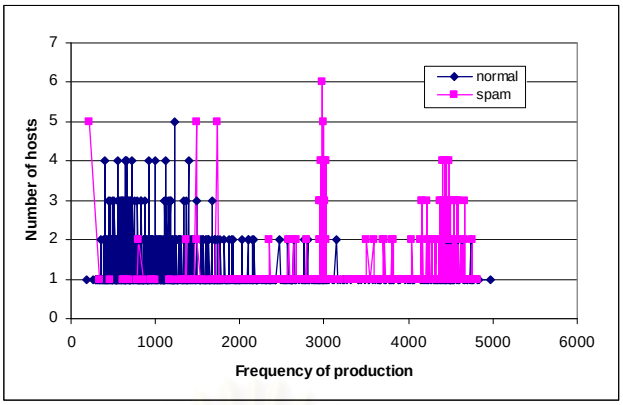
รูปที่ 4.1 กราฟแสดงจำนวนการใช้โปรดักชันที่ 1 ที่แจงส่วนจากเว็บกราฟในระดับโฮสกับจำนวนโฮสที่เป็นสแปมกับโฮสปกติ



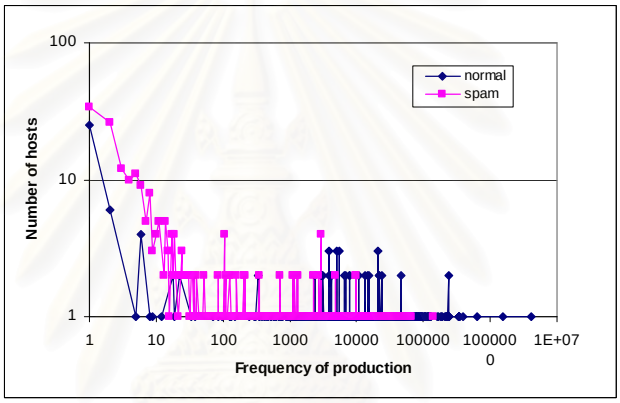
รูปที่ 4.2 กราฟแสดงจำนวนการใช้โปรดักชันที่ 2 ที่แจงส่วนจากเว็บกราฟในระดับโฮสกับจำนวนโฮสที่เป็นสแปมกับโฮสปกติ



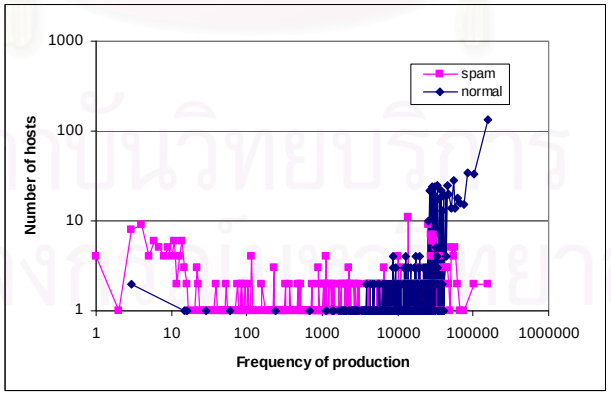
รูปที่ 4.3 กราฟแสดงจำนวนการใช้โปรดักชันที่ 3 ที่แจงส่วนจากเว็บกราฟในระดับโฮสกับจำนวนโฮสที่เป็นสแปมกับโฮสปกติ



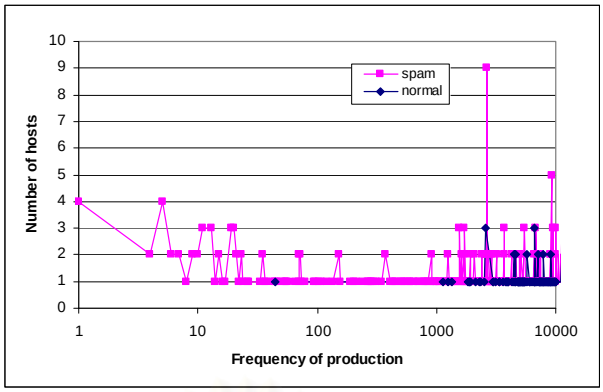
รูปที่ 4.4 กราฟแสดงจำนวนการใช้โปรดักชันที่ 4 ที่แจงส่วนจากเว็บกราฟในระดับโฮสกับจำนวนโฮสที่เป็นสแปมกับโฮสปกติ



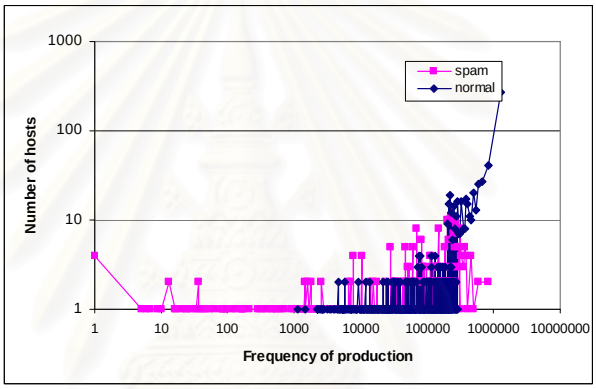
รูปที่ 4.5 กราฟแสดงจำนวนการใช้โปรดักชันที่ 1 ที่แจงส่วนจากเว็บกราฟในระดับเว็บเพจกับจำนวนโฮสที่เป็นสแปมกับโฮสปกติ



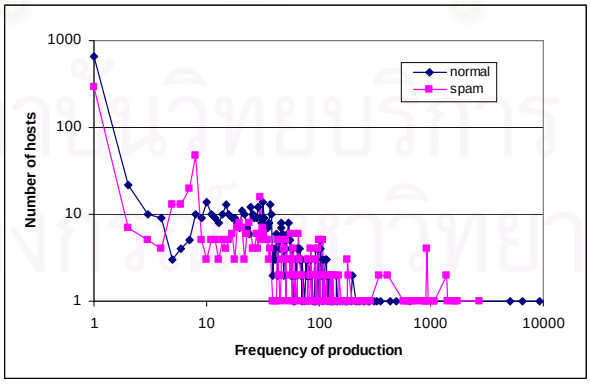
รูปที่ 4.6 กราฟแสดงจำนวนการใช้โปรดักชันที่ 2 ที่แจงส่วนจากเว็บกราฟในระดับเว็บเพจกับจำนวนโฮสที่เป็นสแปมกับโฮสปกติ



รูปที่ 4.7 กราฟแสดงจำนวนการใช้โปรดักชันที่ 3 ที่แจงส่วนจากเว็บกราฟในระดับเว็บเพจกับจำนวนโฮสที่เป็นสแปมกับโฮสปกติ



รูปที่ 4.8 กราฟแสดงจำนวนการใช้โปรดักชันที่ 4 ที่แจงส่วนจากเว็บกราฟในระดับเว็บเพจกับจำนวนโฮสที่เป็นสแปมกับโฮสปกติ



รูปที่ 4.9 กราฟแสดงจำนวนการใช้โปรดักชันที่แสดงถึงบูซเพจในระยะทางที่ 5 แจงส่วนจากเว็บกราฟในระดับเว็บเพจกับจำนวนโฮสที่เป็นสแปมกับโฮสปกติ

ซึ่งจากผลการทดลองพบว่าจำนวนการใช้โปรดักชันที่ 2 3 และ 4 ของไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มกับข้อมูลเว็บกราฟระดับเว็บเพจ (p2, p3, p4) และจำนวนการใช้โปรดักชันที่ 1 และ 2 ของไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มกับข้อมูลเว็บกราฟในระดับโฮส (P1,P2) มีความเหมาะสมในการนำมาตรวจจับลิงก์ฟาร์ม เพราะสามารถแยกแยะระหว่างเส้นกราฟของโฮสสแปมและโฮสปกติได้เป็นอย่างดี ดังนั้นจึงมีการสร้างกฎตรรกศาสตร์เพื่อใช้ในการตรวจสอบกับข้อมูลและทำการเปรียบเทียบประสิทธิภาพของกฎซึ่งพบว่ากฎตรรกศาสตร์ที่ 1 ให้ค่าความแม่นยำสูงที่สุดเมื่อปรับค่าพารามิเตอร์ในชุดข้อมูลสอน และนำมาทดสอบกฎด้วยพารามิเตอร์ที่ได้กับชุดข้อมูลทดสอบ โดยกำหนดค่าเรียกคืนเท่ากับ 1 ดังแสดงได้ตามตารางที่ 4.1 เมื่อกำหนดให้ $P1_{training}$ และ $P2_{training}$ คือค่าพารามิเตอร์ซึ่งเป็นจำนวนการใช้โปรดักชันที่ 1 และ 2 ของเว็บกราฟระดับโฮสที่ได้จากชุดข้อมูลสอนเช่นเดียวกับกรณี $p2_{training}$ $p3_{training}$ และ $p4_{training}$ คือค่าพารามิเตอร์ซึ่งเป็นจำนวนการใช้โปรดักชันของเว็บกราฟในระดับเว็บเพจ และจะบรรจุกฎตรรกศาสตร์นี้ไว้ในอัลกอริทึมที่ 3.4

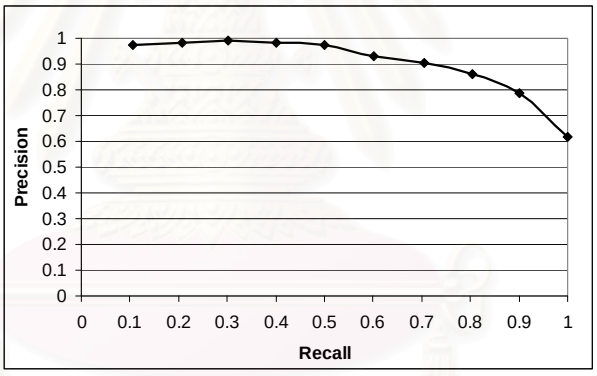
ตารางที่ 4.1 แสดงค่าความแม่นยำของกฎตรรกศาสตร์ที่เลือกใช้ทั้งในข้อมูลสอนและข้อมูลทดสอบเมื่อกำหนดค่าเรียกคืนเท่ากับ 1

กฎที่	กฎตรรกศาสตร์	ค่าความแม่นยำ (ข้อมูลสอน)	ค่าความแม่นยำ (ข้อมูลทดสอบ)
1	((p2<= $p2_{training}$) or (p3<= $p3_{training}$) or (p4<= $p4_{training}$)) and (P1<= $P1_{training}$) and (P2<= $P2_{training}$)	0.619301	0.631361
2	(p2<= $p2_{training}$) and (p3<= $p3_{training}$) and (p4<= $p4_{training}$) and ((P1<= $P1_{training}$) or (P2<= $P2_{training}$))	0.601495	0.619988
3	(p2<= $p2_{training}$) or (p3<= $p3_{training}$) or (p4<= $p4_{training}$) or (P1<= $P1_{training}$) or (P2<= $P2_{training}$)	0.523785	0.534301
4	(p2<= $p2_{training}$) and (p3<= $p3_{training}$) and (p4<= $p4_{training}$) and (P1<= $P1_{training}$) and (P2<= $P2_{training}$)	0.535988	0.627474

และจากวิธีการเลือกค่าพารามิเตอร์ที่เหมาะสมในการสร้างกฎตรรกศาสตร์ในชุดข้อมูลสอนจะได้ค่าพารามิเตอร์ที่เหมาะสมในแต่ละระดับค่าเรียกคืนดังแสดงในตารางที่ 4.2 และแสดงถึงกราฟระหว่างค่าความแม่นยำและค่าเรียกคืน ดังรูปที่ 4.10

ตารางที่ 4.2 แสดงค่าวัดประสิทธิภาพแต่ละจำนวนการใช้โปรดักชันที่เลือกใช้ในชุดข้อมูลสอน

ค่าพารามิเตอร์ ($p2_{training}$, $p3_{training}$, $p4_{training}$, $P1_{training}$, $P2_{training}$)	ค่าเรียกคืน	ค่าความแม่นยำ
25,25,25,84,500	0.105606	0.975904
350,350,350,100,500	0.207301	0.981482
1500,1500,1500,150,410	0.302477	0.991453
2000,2000,2000,150,1000	0.401565	0.980892
4000,4000,4000,150,600	0.500652	0.974619
10000,10000,10000,80,600	0.601043	0.929436
15000,15000,15000,80,650	0.704042	0.903010
25000,25000,25000,100,600	0.804433	0.861732
30000,30000,30000,160,950	0.900913	0.787016
158989,317980,817668,156,3175	1.000000	0.619301



รูปที่ 4.10 กราฟค่าความแม่นยำในแต่ละระดับค่าเรียกคืนของข้อมูลชุดสอน

4.3 ผลการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟโดยแจงส่วนกับเว็บกราฟในระดับเว็บเพจร่วมกับระดับโฮส

จากวิธีการวัดประสิทธิภาพของอัลกอริทึมจะแบ่งข้อมูลทั้งหมดออกมาเป็น 2 ชุดที่มีจำนวนเท่าๆ กัน คือ ชุดข้อมูลสอน และข้อมูลทดสอบ ตามสภาพแวดล้อมการทดลองตามอัลกอริทึมที่ใช้วิธีทรานส์ดักทิฟและแอนตี้ทรัสต์ ซึ่งในผลการทดลองจะทำการเก็บค่าเฉลี่ยทั้งหมด 10 รอบ ดังแสดงในตารางที่ 4.3-4.12 ส่วนในตารางที่ 4.13 จะแสดงถึงค่าเฉลี่ยของประสิทธิภาพในการใช้อัลกอริทึมกับข้อมูลชุดทดสอบ ส่วนในกราฟรูปที่ 4.11 แสดงถึงประสิทธิภาพในการตรวจจับด้วยอัลกอริทึมกับข้อมูลทดสอบเปรียบเทียบกับข้อมูลสอน

ตารางที่ 4.3 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 3 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 25000,25000,25000,100,600

รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	3	0.168697	0.967742
2	3	0.171374	0.983784
3	3	0.162724	0.971751
4	3	0.156102	0.959302
5	3	0.154066	0.947059
6	3	0.152963	0.963855
7	3	0.157944	0.954023
8	3	0.161931	0.966102
9	3	0.160984	0.955056
10	3	0.167603	0.978142
ค่าเฉลี่ย =		**0.161439**	**0.964682**

ตารางที่ 4.4 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 6 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 25000,25000,25000,100,600

รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	6	0.308341	0.948127
2	6	0.325800	0.966480
3	6	0.316934	0.965418
4	6	0.276253	0.951140
5	6	0.247846	0.925000
6	6	0.244741	0.927536
7	6	0.237868	0.943396
8	6	0.236742	0.936330
9	6	0.272727	0.947368
10	6	0.300561	0.946903
ค่าเฉลี่ย =		**0.276781**	**0.945770**

ตารางที่ 4.5 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 8 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 25000,25000,25000,100,600

รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	8	0.359887	0.950495
2	8	0.379472	0.966427
3	8	0.368022	0.967662
4	8	0.373699	0.961071
5	8	0.351196	0.943445
6	8	0.348948	0.945596
7	8	0.329210	0.955801
8	8	0.324810	0.950139
9	8	0.332386	0.948649
10	8	0.348314	0.948980
ค่าเฉลี่ย =		**0.351594**	**0.953826**

ตารางที่ 4.6 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 10 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 30000,30000,30000,160,950

รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	10	0.404873	0.937093
2	10	0.432203	0.952282
3	10	0.420056	0.948718
4	10	0.431409	0.944099
5	10	0.410526	0.928571
6	10	0.400573	0.924945
7	10	0.377735	0.934118
8	10	0.390151	0.930023
9	10	0.387310	0.927438
10	10	0.392322	0.929047
ค่าเฉลี่ย =		**0.404716**	**0.935633**

ตารางที่ 4.7 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 15 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 30000,30000,30000,160,950

รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	15	0.471415	0.914546
2	15	0.503767	0.930435
3	15	0.492904	0.932021
4	15	0.509934	0.932526
5	15	0.488995	0.922383
6	15	0.48566	0.918626
7	15	0.453853	0.915547
8	15	0.46875	0.908257
9	15	0.457386	0.899441
10	15	0.469101	0.909256
ค่าเฉลี่ย =		**0.480177**	**0.918304**

ตารางที่ 4.8 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 18 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 25000,25000,25000,100,600

รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	18	0.628865	0.829419
2	18	0.640301	0.850000
3	18	0.637653	0.853165
4	18	0.641438	0.851759
5	18	0.634449	0.845663
6	18	0.611854	0.842105
7	18	0.599429	0.818182
8	18	0.607954	0.817834
9	18	0.602272	0.819588
10	18	0.610486	0.826363
ค่าเฉลี่ย =		**0.621470**	**0.835408**

ตารางที่ 4.9 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 19 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 25000,25000,25000,100,600

รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	19	0.694470	0.812500
2	19	0.709981	0.829483
3	19	0.704825	0.831473
4	19	0.711447	0.829107
5	19	0.702392	0.818283
6	19	0.684512	0.818286
7	19	0.666983	0.800228
8	19	0.677083	0.799776
9	19	0.672348	0.802260
10	19	0.677902	0.812570
ค่าเฉลี่ย =		**0.690194**	**0.815397**

ตารางที่ 4.10 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 20 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 25000,25000,25000,100,600

รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	20	0.788191	0.795648
2	20	0.796610	0.803419
3	20	0.792809	0.802682
4	20	0.795648	0.803247
5	20	0.790430	0.792706
6	20	0.773422	0.788499
7	20	0.764985	0.780583
8	20	0.768939	0.781521
9	20	0.766098	0.780888
10	20	0.771535	0.791547
ค่าเฉลี่ย =		**0.780867**	**0.792074**

ตารางที่ 4.11 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 20 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 30000,30000,30000,160,950

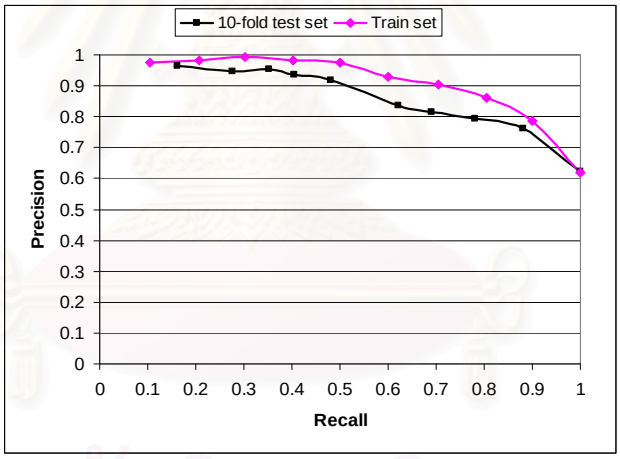
รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	20	0.878163	0.768663
2	20	0.890772	0.776046
3	20	0.890255	0.773849
4	20	0.892147	0.773585
5	20	0.890909	0.760000
6	20	0.881453	0.756358
7	20	0.868696	0.748975
8	20	0.869318	0.751228
9	20	0.867424	0.751436
10	20	0.868913	0.765045
ค่าเฉลี่ย =		**0.879805**	**0.762518**

ตารางที่ 4.12 แสดงประสิทธิภาพของการตรวจจับในระดับการตรวจจับที่ 20 ในแต่ละรอบเมื่อใช้ค่าพารามิเตอร์ คือ 158989,317980,817668,156,3175

รอบที่	ระดับการตรวจจับ	ค่าเรียกคืน	ค่าความแม่นยำ
1	20	1.000000	0.631361
2	20	1.000000	0.630641
3	20	1.000000	0.626928
4	20	1.000000	0.625814
5	20	1.000000	0.617612
6	20	1.000000	0.616382
7	20	1.000000	0.617146
8	20	1.000000	0.620811
9	20	1.000000	0.621908
10	20	1.000000	0.630089
ค่าเฉลี่ย =		**1.000000**	**0.623869**

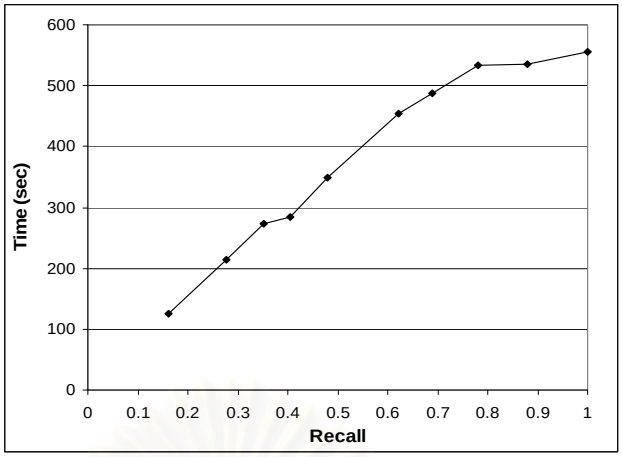
ตารางที่ 4.13 แสดงประสิทธิภาพของอัลกอริทึมกับข้อมูลชุดทดสอบโดยเฉลี่ยในแต่ละรอบ

ค่าพารามิเตอร์ ($p2_{training}$, $p3_{training}$, $p4_{training}$, $P1_{training}$, $P2_{training}$)	ระดับการตรวจจับ	ค่าเรียกคืนเฉลี่ย	ค่าความแม่นยำเฉลี่ย
25000,25000,25000,100,600	3	0.161439	0.964682
25000,25000,25000,100,600	6	0.276781	0.945770
25000,25000,25000,100,600	8	0.351594	0.953826
30000,30000,30000,160,950	10	0.404716	0.935633
30000,30000,30000,160,950	15	0.480177	0.918304
25000,25000,25000,100,600	18	0.621471	0.835408
25000,25000,25000,100,600	19	0.690194	0.815397
25000,25000,25000,100,600	20	0.780867	0.792074
30000,30000,30000,160,950	20	0.879805	0.762518
158989,317980,817668,156,3175	20	1.000000	0.623869



รูปที่ 4.11 กราฟค่าความแม่นยำในแต่ละระดับค่าเรียกคืนของข้อมูลชุดทดสอบเฉลี่ยในแต่ละรอบการทำงานเปรียบเทียบกับประสิทธิภาพจากชุดสอน

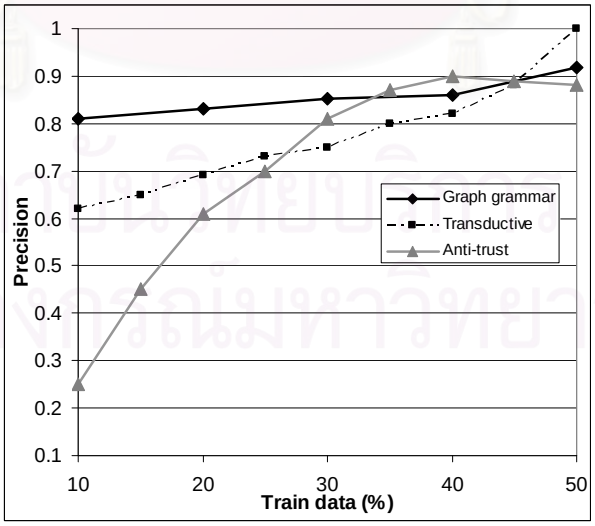
ในการทดลองวัดประสิทธิภาพของเวลาในการทำงานอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟ จะทำจับเวลาทำงานเฉพาะอัลกอริทึมทดสอบตัวอย่าง (อัลกอริทึมที่ 3.4) เมื่อกำหนดระดับตรวจสอบมาให้ตามตารางที่ 4.3-4.12 และเปรียบเทียบเวลาในการทำงานเฉลี่ยทั้ง 10 รอบเป็นหน่วยวินาที ซึ่งสามารถเขียนกราฟระหว่างเวลากับระดับค่าเรียกคืนได้ในรูปที่ 4.12



รูปที่ 4.12 กราฟแสดงเวลาเมื่อเทียบกับค่าเรียกคืนระดับต่างๆ

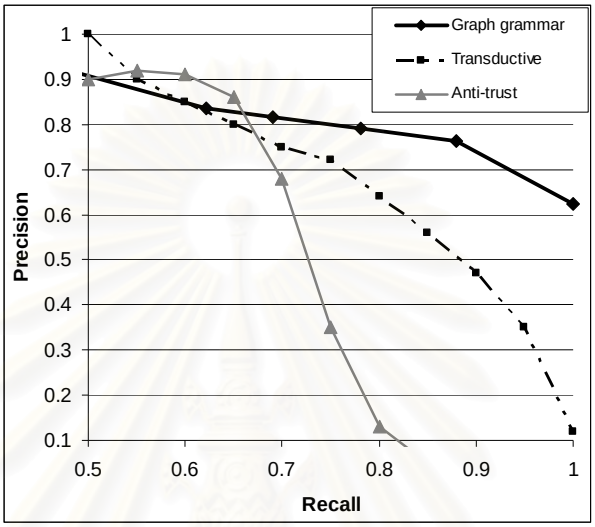
4.4 ผลการเปรียบเทียบประสิทธิภาพของการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟเทียบกับผลงานวิจัยที่เกี่ยวข้อง

ในการเปรียบเทียบประสิทธิภาพของอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟกับอัลกอริทึมการตรวจจับลิงก์ฟาร์มที่ใช้วิธีทรานส์ดักทิฟและแอนตี้ทรัสต์ ผลการทดลองเปรียบเทียบประสิทธิภาพของอัลกอริทึมในด้านค่าความแม่นยำเมื่อกำหนดให้สัดส่วนข้อมูลสอนต่างๆ กัน ซึ่งจะวัดผลโดยทำการเก็บค่าเฉลี่ยทั้งหมด 10 รอบกับชุดมูลที่เป็นแบบไขว้ เมื่อกำหนดให้ค่าเรียกคืนเป็น 0.5 และนำมาสร้างกราฟระหว่างสัดส่วนข้อมูลสอนกับค่าความแม่นยำดังแสดงในรูปที่ 4.13



รูปที่ 4.13 กราฟแสดงค่าความแม่นยำกับสัดส่วนข้อมูลสอนเทียบกับอัลกอริทึมอื่น

ผลการทดลองเปรียบเทียบประสิทธิภาพของอัลกอริทึมในด้านค่าความแม่นยำเมื่อกำหนดให้ค่าเรียกคืนต่างๆ กัน ซึ่งจะวัดผลโดยทำการเก็บค่าเฉลี่ยทั้งหมด 10 รอบกับชุดมูลที่เป็นแบบไขว้ เมื่อกำหนดให้สัดส่วนของข้อมูลสอนเป็น 50% โดยเปรียบเทียบค่าเรียกคืนตั้งแต่ 0.5-1.0 และนำมาสร้างกราฟระหว่างสัดส่วนข้อมูลสอนและค่าความแม่นยำดังแสดงในรูปที่ 4.14



รูปที่ 4.14 กราฟแสดงค่าความแม่นยำกับค่าเรียกคืนเทียบกับอัลกอริทึมอื่น

4.5 ประสิทธิภาพของการตรวจจับลิงก์ฟาร์มที่มีคะแนนเพจแรงค์ในระดับที่สูงโดยใช้ไวยากรณ์กราฟ

ผลทดลองจากวิธีการวัดประสิทธิภาพเฉพาะกลุ่มลิงก์ฟาร์มที่มีคะแนนเพจแรงค์ที่สูงโดยการใช้ข้อมูลในถังแต่ละถัง มาเป็นชุดข้อมูลสอนและทดสอบ เมื่อกำหนดให้สัดส่วนของข้อมูลสอนเป็น 50% สุ่มจากข้อมูลที่ติดฉลาก ดังนั้นประสิทธิภาพในการตรวจจับลิงก์ฟาร์มที่แสดงถึงค่าเรียกคืนและค่าความแม่นยำนั้นสามารถแสดงในตารางที่ 4.14

ตารางที่ 4.14 แสดงประสิทธิภาพในการตรวจจับลิงก์ฟาร์มที่มีคะแนนเพจแรงค์ในถังที่ 1-5

ถังที่	ค่าพารามิเตอร์ ($p2_{training}$, $p3_{training}$, $p4_{training}$, $P1_{training}$, $P2_{training}$)	ค่าเรียกคืน	ค่าความแม่นยำ
1	80000,150000,400000,80,1500	0.007429	0.909090
2	40000,100000,230000,35,400	0.019316	0.962963
3	20000,150000,250000,60,1000	0.054977	0.860465
4	35000,100000,300000,80,800	0.048291	0.844155
5	30000,100000,170000,50,500	0.069093	0.808695

4.6 วิเคราะห์ผลการทดลอง

จากผลการทดลองพบว่าจำนวนการใช้โปรดักชันมีผลต่อโครงสร้างและสมบัติของลิงก์ฟาร์ม ซึ่งจำนวนการใช้โปรดักชันแต่ละโปรดักชันจะพิจารณาถึงจำนวนของบูซเพจที่ทำหน้าที่เพิ่มคะแนนให้กับเว็บเพจเป้าหมายและจำนวนโฮสที่ชี้ไปยังโฮสเป้าหมาย เนื่องจากผู้สร้างสแปมไม่สามารถเพิ่มโฮสให้ชี้มายังลิงก์ฟาร์มได้อย่างอิสระ ดังนั้น จำนวนโฮส จำนวนบูซเพจและระดับขั้นของบูซเพจนั้นเป็นลักษณะสำคัญในการแบ่งแยกโฮสสแปมและโฮสปกติ

ขณะเดียวกันลำดับการเลือกโฮสมาทดสอบมีความสำคัญต่อประสิทธิภาพของอัลกอริทึมการตรวจจับลิงก์ฟาร์มด้วยไวยากรณ์กราฟเพราะเนื่องจากคะแนนเพจแรงค์มีผลต่ออันดับของผลการค้นคืนในโปรแกรมค้นหา และการเลือกโฮสที่มีลิงก์ชี้ไปยังเว็บที่เป็นสแปมยังช่วยเพิ่มความถูกต้องให้กับการทดสอบอีกด้วย เพราะเว็บที่ปกตินั้นมักจะเป็นเว็บที่มีเนื้อหาสาระต่อผู้ใช้งานและย่อมจะแนะนำเว็บเพจที่ดีต่อผู้ใช้งานด้วยเช่นกัน

ในการเปรียบเทียบผลการทดลองกับงานวิจัยที่เกี่ยวข้อง ไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มสามารถนำเสนอลักษณะสำคัญของลิงก์ฟาร์มด้วยจำนวนการใช้โปรดักชันที่เหมาะสม ดังนั้นด้วยตัวอย่างสอนที่น้อยกว่าผลการทดลองจึงพบว่าสามารถให้ค่าความแม่นยำในระดับที่สูงกว่าเมื่อเปรียบเทียบกับอัลกอริทึมทั้งสอง

บทที่ 5

สรุปผลงานวิจัยและข้อเสนอแนะ

5.1 สรุปผลงานวิจัย

งานวิจัยนี้พัฒนาไวยากรณ์กราฟในการนำเสนอลิงก์ฟาร์ม ซึ่งเป็นโครงสร้างที่ผู้สร้างสแปมสร้างขึ้นอย่างจงใจเพื่อเพิ่มคะแนนเพจแรงค์ให้กับเว็บเพจเป้าหมาย โดยไวยากรณ์กราฟลิงก์ฟาร์มเป็นไวยากรณ์ที่สร้างขึ้นจากแบบจำลองของลิงก์ฟาร์มในงานวิจัยที่นำเสนอก่อนหน้านี้ ไวยากรณ์กราฟลิงก์ฟาร์มที่สร้างขึ้นมีลักษณะพิเศษคือ มีการใช้โปรดักชันแบบวนซ้ำได้ ทำให้สามารถใช้โปรดักชันต่างๆ ในการสร้างและเปลี่ยนแปลงลิงก์ฟาร์มได้ตามโครงสร้างที่ต้องการ ส่วนกระบวนการตรวจจับลิงก์ฟาร์มนั้นมีการพัฒนาไวยากรณ์กราฟในการตรวจจับลิงก์ฟาร์ม ซึ่งมีการเลือกโปรดักชันและจำนวนการใช้โปรดักชันที่เหมาะสมในการสร้างกฎตรรกศาสตร์ และมีการพัฒนาอัลกอริทึมในตรวจจับลิงก์ฟาร์มกับชุดข้อมูลเว็บกราฟโดยมีการแบ่งระดับการตรวจจับออกเป็น 20 ระดับตามความสำคัญของโฮสที่จะใช้ในการตรวจสอบ

ในส่วนของผลการทดลองพบว่าจำนวนการใช้โปรดักชันบางตัวของไวยากรณ์กราฟที่แจงส่วนในข้อมูลเว็บกราฟระดับโฮสและระดับเว็บเพจ สามารถนำมาใช้ในการแยกแยะระหว่างโฮสปกติและโฮสที่เป็นสแปม และเมื่อใช้การตัดสินใจร่วมกันระหว่างจำนวนการใช้โปรดักชันเหล่านั้นพบว่าค่าความแม่นยำของการตรวจจับลิงก์ฟาร์มมีค่าที่สูง ในการเปรียบเทียบอัลกอริทึมการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟกับงานวิจัยที่เกี่ยวข้องพบว่าประสิทธิภาพในการตรวจจับได้ผลดีเทียบเท่ากับงานวิจัยที่ใช้วิธีทรานส์ดักทิฟและอัลกอริทึมแอนตี้ทรัสต์ในระดับค่าเรียกคืนที่ต่ำ และมีค่าความแม่นยำมากกว่าในระดับค่าเรียกคืนที่สูง และถ้าหากเทียบสัดส่วนข้อมูลสอนในระดับที่น้อย ประสิทธิภาพการตรวจจับลิงก์ฟาร์มด้วยไวยากรณ์กราฟสามารถให้ค่าความแม่นยำที่สูงกว่างานวิจัยที่เปรียบเทียบ ส่วนผลการทดลองเพิ่มเติมเกี่ยวการตรวจจับลิงก์ฟาร์มที่มีเว็บเพจเป้าหมายคะแนนเพจแรงค์สูงพบว่าโดยเลือกค่าจำนวนการใช้โปรดักชันจากชุดข้อมูลสอนในถังข้อมูลนั้นๆ ที่แบ่งไว้ตามระดับคะแนนเพจแรงค์ของเว็บเพจเป้าหมาย พบว่าในผลการทดลองด้วยอัลกอริทึมการทดสอบตัวอย่างกับข้อมูลทดสอบในถังนั้นๆ มีค่าความแม่นยำที่สูง ดังนั้นเราจึงสามารถเลือกตรวจจับเฉพาะลิงก์ฟาร์มของเว็บเพจเป้าหมายที่มีคะแนนเพจแรงค์สูงได้

ดังนั้นการตรวจจับลิงก์ฟาร์มโดยใช้ไวยากรณ์กราฟที่นำเสนอ มีประสิทธิภาพที่ดีในการตรวจจับลิงก์ฟาร์มเพราะเราสามารถตรวจจับลิงก์ฟาร์มที่มีอิทธิพลสูงต่อลำดับผลการค้นคืนในโปรแกรมค้นหาบนอินเทอร์เน็ต และเนื่องจากโครงสร้างของลิงก์ฟาร์มมีลักษณะเฉพาะตัวที่แตกต่างจากโครงสร้างของเว็บเพจที่เป็นเว็บเพจปกติ ดังนั้นถ้าหากพัฒนาไวยากรณ์กราฟที่แสดงถึงโครงสร้างของลิงก์ฟาร์มนั้นได้ เราก็สามารถเข้าใจถึงลักษณะและกระบวนในการสร้าง

ลิงก์ฟาร์ม ทำให้สามารถที่จะพัฒนาคุณภาพของโปรแกรมค้นหาบนอินเทอร์เน็ตให้ดีขึ้นได้โดยใช้กระบวนการตรวจจับลิงก์ฟาร์มด้วยไวยากรณ์กราฟ

5.2 ข้อเสนอแนะ

1. ถ้าหากใช้ไวยากรณ์กราฟที่มีความซับซ้อนที่สูงขึ้นเช่นไวยากรณ์กราฟพึ่งบริบท (context-sensitive graph grammar) ในการสร้างไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มจะช่วยให้สามารถอธิบายโครงสร้างของลิงก์ภายในบูซเพจได้เพิ่มขึ้น น่าจะช่วยให้สามารถตรวจจับลิงก์ฟาร์มได้ถูกต้องมากขึ้น แต่จะต้องใช้ทรัพยากรในการตรวจสอบที่มากขึ้น เนื่องจากอัลกอริทึมแจงส่วนของไวยากรณ์กราฟพึ่งบริบทมีความซับซ้อนสูง

2. ในการตัดสินใจว่าเป็นลิงก์ฟาร์มโดยใช้กฎตรรกศาสตร์ อาจจะเปลี่ยนเป็นการเรียนรู้ของเครื่องอย่างอัตโนมัติเช่นเครือข่ายประสาทเทียม ต้นไม้ตัดสินใจ หรือขั้นตอนวิธีเชิงพันธุกรรม ซึ่งให้ผลลัพธ์ออกมาเป็นความน่าจะเป็นของลิงก์ฟาร์มเพื่อความสะดวกในการถ่วงน้ำหนักของลิงก์ฟาร์มที่ค้นพบกับคะแนนเพจแรงค์ทั้งหมดของเว็บเพจภายในลิงก์ฟาร์ม ทำให้ลิงก์ฟาร์มที่ค้นพบไม่ต้องถูกลบออกจากเว็บกราฟเพียงแต่ถูกปรับลดคะแนนเพจแรงค์ลง

3. การเพิ่มเติมในส่วนของการพัฒนาระบบในการสร้างไวยากรณ์กราฟสำหรับตรวจจับลิงก์ฟาร์มแบบอัตโนมัติ จะช่วยให้ความสามารถในการตรวจจับลิงก์ฟาร์มสูงขึ้น เนื่องจากไม่มีมนุษย์มากำหนดแบบจำลองของลิงก์ฟาร์ม โดยอาจจะพัฒนาอัลกอริทึมสร้างไวยากรณ์กราฟ (inference graph grammar) จากตัวอย่างบวกที่เป็นโครงสร้างลิงก์ฟาร์ม และตัวอย่างลบที่เป็นเว็บเพจปกติ

รายการอ้างอิง

[1] B. Wu and K. Chellapilla. Extracting Link Spam using Biased Random Walks from Spam Seed Sets. In the Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web. (2007).

[2] V. Krishnan and R. Raj. Web Spam Detection with Anti-Trust Rank. In the Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web. (2006).

[3] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating Web Spam with TrustRank. In the Proceedings of the 30th International Conference on Very Large Data Bases. (2004).

[4] B. Wu and B. D. Davison. Identifying Link Farm Pages. In the Proceedings of the 14th International World Wide Web Conference. (2005).

[5] D. Gibson, R. Kumar, and A. Tomkins. Discovering Large Dense Subgraphs in Massive Graphs. In the Proceedings of the 31st International conference on Very Large Data Bases. (2005).

[6] H. Saito, M. Toyoda, M. Kitsuregawa, and K. Aihara. A Large-Scale Study of Link Spam Detection by Graph Algorithms. In the Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web. (2007).

[7] P.T. Metaxas and J. DeStefano. Web Spam, Propaganda and Trust. In the Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web. (2005).

[8] D. Zhou, C.Burges and T.Tao. Transductive Link Spam Detection. Proceedings of the 3rd international workshop on Adversarial Information Retrieval on the Web. (2007).

[9] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Link-based characterization and detection of Web Spam. In Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web. (2006).

[10] B. Shan. Stochastic Context-Free Graph Grammars for Glycoprotein Modeling. Implementation and Application of Automata. (2004).

[11] D. Zhang, K. Zhang, and J. Cao. A Context Sensitive Graph Grammar Formalism for the Specification of Visual Languages. The Computer Journal 44 (2001) : 186-200.

[12] I. Jonyer and L. Holder, and D. Cook. MDL-Based Context-Free Graph Grammar Induction and Applications. International Journal of Artificial Intelligence Tools 13 (2004) : 65-79.

[13] L. Pages, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bring Order to the Web. Technical report, Stanford Digital Libraries. (1998).

[14] Z. Gyöngyi and H. Garcia-Molina. Web Spam Taxonomy. In the Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web. (2005).

[15] Y. Du, Y. Shi, and X. Zhao. Using Spam Farm to Boost PageRank. In the Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web. (2007).

[16] T. H. Haveliwala. Effcient Computation of PageRank. Technical report, Stanford Digital Libraries. (1999).

[17] J.P. Kukluk, L.B. Holder, and D.J. Cook. Inference of Node Replacement Recursive Graph Grammars. In the Proceedings of the 6th SIAM International conference on Data Mining. (2006).

[18] K. Ates, J.P. Kukluk, L.B. Holder, D.J. Cook, and K. Zhang. Graph Grammar Induction on Structural Data for Visual Programming. In the Proceedings of the 18th IEEE International Conference of Tools with AI. (2006).

[19] Y.Adachi and Y.Nakajima. A Context-Sensitive NCE Graph Grammar and its Parsability. Proceedings of the 2000 IEEE International Symposium on Visual Languages. (2000)

[20] J. Engelfriet and V. Oostrom. Regular Description of Context-Free Graph Languages. Journal of Computer and System Sciences 53 (1996) : 556-574

[21] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your Neighbors: Web Spam Detection using the Web Topology. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. (2007).

[22] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna. Reference Collection for Web Spam. ACM SIGIR Forum. (2006).

ประวัติผู้เขียนวิทยานิพนธ์

นายเกียรติคุณ ชอบธรรม เกิดเมื่อวันที่ 31 มีนาคม พ.ศ.2527 ที่จังหวัดน่าน และเรียนจบการศึกษาระดับประถมศึกษาจากโรงเรียนราชานุบาล ระดับมัธยมศึกษาจากโรงเรียนศรีสวัสดิ์วิทยาคาร อ.เมือง จ.น่าน ได้รับการคัดเลือกให้เป็นนักเรียนทุนที่ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยมหิดล และสำเร็จการศึกษาระดับปริญญาบัณฑิตเกียรตินิยมอันดับหนึ่งในปี 2548 ต่อมาได้เข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2549