ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)

ปริญญา

วิศวกรรมคอมพิวเตอร์	วิศวกรรมคอมพิวเตอร์
สาขา	**ภาควิชา**

เรื่อง การระบุเว็บโฮสต์ไทยโดยใช้เครื่องจักรเรียนรู้

Machine Learning Based Thai Web Host Identification

นามผู้วิจัย นายปุณณวัฒน์ ธาดาภาคย์

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก .....

(ผู้ช่วยศาสตราจารย์อานนท์ รุ่งสว่าง, Ph.D.)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม .....

(ผู้ช่วยศาสตราจารย์อรรถสิทธิ์ สุรฤกษ์, Dr.Inf.)

หัวหน้าภาควิชา .....

(ผู้ช่วยศาสตราจารย์ภุชงค์ อุทโยภาศ, Ph.D.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

.....

(รองศาสตราจารย์กัญจนา ธีระกุล, D.Agr.)

คณบดีบัณฑิตวิทยาลัย

วันที่ ..... **เดือน** ..... **พ.ศ.** .....

วิทยานิพนธ์

เรื่อง

การระบุเว็บโฮสต์ไทยโดยใช้เครื่องจักรเรียนรู้

Machine Learning Based Thai Web Host Identification

โดย

นายปุณณวัฒน์ ธาดาภาคย์

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)

พ.ศ. 2554

ปุณณวัฒน์ ธาดาภาคย์ 2554: การระบุเว็บโฮสต์ไทยโดยใช้เครื่องจักรเรียนรู้ ปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์) สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก: ผู้ช่วยศาสตราจารย์ อานนท์ รุ่งสว่าง, Ph.D. 79 หน้า

เว็บเพจบนอินเทอร์เน็ตถือว่าเป็นแหล่งความรู้ขนาดใหญ่ และมีการเพิ่มจำนวนขึ้นอย่างต่อเนื่อง จนทำให้ไม่สามารถเก็บรวบรวมเว็บเพจทั้งหมดได้ภายใต้ทรัพยากรระบบที่มีอยู่อย่างจำกัด เช่น ระยะเวลา พื้นที่เก็บข้อมูล หรือแบนด์วิดท์ เป็นต้น ด้วยเหตุนี้การใช้เว็บคราวเลอร์แบบเจาะจงภาษาจึงเป็นทางเลือกหนึ่งที่จะมาช่วยให้สามารถเลือกเก็บเฉพาะเว็บเพจที่เขียนด้วยภาษาตามที่ต้องการได้ ตัวอย่างการนำไปใช้งานเช่น ผู้ให้บริการระบบสืบค้นในแต่ละประเทศมักจะแสดงเว็บเพจผลลัพธ์ตามภาษาประจำชาติของผู้ใช้ เพื่อให้ผู้ใช้เกิดความพึงพอใจต่อระบบ หรือการสร้างคลังเว็บเพจประจำชาติ เพื่อเก็บรักษาเว็บเพจที่มีข้อมูลสำคัญด้านสังคม วัฒนธรรม วิถีการดำเนินชีวิตให้กับคนรุ่นต่อไป

วิทยานิพนธ์ฉบับนี้นำเสนอระบบต้นแบบเว็บคราวเลอร์เจาะจงเว็บเพจภาษาไทยแบบแยกเก็บตามไซต์ ซึ่งประกอบด้วยส่วนของการทำนายหรือระบุภาษาของโฮสต์ และส่วนของเว็บคราวเลอร์แบบแยกเก็บตามไซต์ โดยที่ส่วนของการทำนายภาษาใช้เทคนิคทางเครื่องจักรเรียนรู้แบบหลายตัว ซึ่งเครื่องจักรเรียนรู้แต่ละตัวจะเรียนรู้จากคุณลักษณะทั้งตัวอย่างของเว็บโฮสต์ที่เกี่ยวข้อง และเว็บโฮสต์ที่ไม่เกี่ยวข้อง ด้วยเทคนิคทางเครื่องจักรเรียนรู้ที่แตกต่างกัน และส่วนทำนายภาษายังมีหน้าที่ช่วยระบุภาษาของเว็บโฮสต์เป้าหมายพร้อมทั้งให้คะแนนความมั่นใจอีกด้วย แต่สำหรับเว็บคราวเลอร์แบบแยกเก็บตามไซต์มีหน้าที่เก็บเว็บเพจภายใต้เว็บไซต์ที่กำหนดเรียงลำดับตามคะแนนความมั่นใจที่ได้จากส่วนการทำนายภาษา จากผลการทดลองแสดงให้เห็นว่าประสิทธิภาพการเก็บเว็บเพจจากอินเทอร์เน็ตของวิธีการที่นำเสนอดีกว่าวิธีการของเว็บคราวเลอร์เจาะจงภาษาในระดับเว็บเพจที่ผ่านมา ซึ่งวัดผลจากอัตราการเก็บเกี่ยวเว็บเพจภาษาไทย

_____ _____ ___ / ___ / ___

ลายมือชื่อนิสิต ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

Punnawat Tadapak 2011: Machine Learning Based Thai Web Host Identification. Master of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Assistant Professor Arnon Rungsawang, Ph.D. 79 pages.

Web pages serve as the largest knowledge bases for humans in all areas, and are continuously increased. Current limited resources, such as time, data storage, network bandwidth, make us unable to collect all web pages from the Internet. Therefore, a language specific web crawler (LSWC) is an alternative approach to help gathering the needed web pages written in a specific language. Some examples usage which apply the LSWC are; the search engine providers in the country that need to collect as most as web pages written in their specified native language to satisfy their own people's need of information, the national web archive project that needs to long term preserve their social heritage, cultural, way of life, for future generations.

This thesis proposes a prototype of the Thai language specific web site crawler. It composes of two main components; a language predictor and a web site crawler. The former employs the classifier emsemble based predictors that each different machine learning based classifier has been trained from the sample sets of host features, both positive and negative examples. This component then takes responsibility to identify the relevant web hosts in which web pages are written in Thai, including with some confident scores. The later, i.e. the web site crawler, chooses to collect the web pages from the prioritized list of relevant web hosts provided by the former. The experimental result from the real Internet data set shows that the crawling performance in term of the standard harvest rate of the proposed approach is better than those traditional LSWCs reported in the literature.

_____ _____ ___ / ___ / ___

Student's signature Thesis Advisor's signature

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณผู้ช่วยศาสตราจารย์อานนท์ รุ่งสว่าง อาจารย์ที่ปรึกษาวิทยานิพนธ์หลักที่คอยช่วยให้คำแนะนำต่างๆ ในการทำวิจัย ตลอดจนการตรวจสอบแก้ไขข้อบกพร่องต่างๆ ทั้งในบทความทางวิชาการและวิทยานิพนธ์ฉบับนี้จนเสร็จสมบูรณ์ นอกจากนี้ข้าพเจ้าขอขอบคุณพี่ๆ น้องๆ ในห้องปฏิบัติการวิจัยวิศวกรรมข้อมูล และฐานความรู้ขนาดใหญ่ (MIKE) ที่ช่วยให้คำแนะนำในงานวิจัย และให้ความช่วยเหลือแก่ข้าพเจ้าเป็นอย่างดี โดยเฉพาะนายธนพล สืบเชื้อ ที่ช่วยพัฒนาโปรแกรมเว็บคราวเลอร์แบบเจาะจงภาษาจนสามารถใช้งานได้จริง และนายเอกสิทธิ์ ศรีสุขะ ที่ได้ถ่ายทอดประสบการณ์จากการทำวิจัยด้านเว็บคราวเลอร์เจาะจงภาษาไทย

ท้ายที่สุดนี้ขอขอบคุณบิดา และมารดาที่คอยให้การสนับสนุนด้านการศึกษา ค่าใช้จ่ายต่างๆ และให้กำลังใจ จนวิทยานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงได้ด้วยดี

ปุณณวัฒน์ ธาดาภาคย์

กันยายน 2554

สารบัญ

หน้า

สารบัญตาราง

สารบัญตาราง (ต่อ)

ตารางผนวกที่ **หน้า**

สารบัญภาพ

สารบัญภาพ (ต่อ)

ภาพผนวกที่ **หน้า**

การระบุเว็บโฮสต์ไทยโดยใช้เครื่องจักรเรียนรู้

Machine Learning Based Thai Web Host Identification

คำนำ

เว็บเพจบนอินเทอร์เน็ตถือว่าเป็นแหล่งความรู้ขนาดใหญ่และมีจำนวนเพิ่มขึ้นอย่างต่อเนื่อง ทำให้ผู้ใช้ต้องอาศัยเครื่องมือที่เรียกว่า ระบบสืบค้นข้อมูล (search engine) เช่น Yahoo (2011); Google (2011); Bing (2011) เป็นต้น เพื่อค้นหากลุ่มของเว็บเพจที่สนใจตามคำค้นที่กำหนด (query) แต่ด้วยจำนวนเว็บเพจที่พบในปัจจุบันมีมากกว่า 18.77 พันล้านเว็บเพจ (Kunder, 2011) จึงกลายเป็นสิ่งที่ท้าทายต่อทีมนักวิจัยและผู้ให้บริการระบบสืบค้นเป็นอย่างมาก เนื่องจากการเก็บเว็บเพจทั้งหมดต้องใช้ทั้งระยะเวลาและทรัพยากรระบบอย่างมหาศาล ดังนั้นจึงมีนักวิจัยนำเสนอวิธีการต่างๆ เพื่อเลือกเก็บเว็บเพจจากอินเทอร์เน็ต เช่น Chakarbati *et al.* (1999) นำเสนอวิธีการเก็บเว็บเพจตามหัวเรื่องที่สนใจ Gomes *et al.* (2008) เสนอวิธีการสร้างคลังเว็บเพจประจำชาติ (national web archive) โดยการเก็บเว็บเพจตามโดเมนระดับบนสุดของประเทศที่สนใจ และยังมีงานวิจัยที่นำเสนอวิธีการเลือกเก็บเว็บเพจแบบเจาะจงภาษาที่สนใจ เช่น เว็บเพจภาษาไทย (Somboonviwat *et al.* (2006); Srisukha *et al.* (2008)) และเว็บเพจภาษาอาราบิค (Alabbad *et al.*, 2009) เป็นต้น

โดยทั่วไปแล้วผู้ให้บริการระบบสืบค้นข้อมูลในแต่ละประเทศมุ่งเน้นที่จะหาวิธีการเก็บเว็บเพจแบบเจาะจงภาษาท้องถิ่น เพื่อตอบสนองต่อความต้องการของผู้ใช้ ตัวอย่างเช่น ระบบสืบค้นข้อมูล Baidu (2011) ของประเทศจีน เป็นต้น ซึ่งการเก็บเว็บเพจแบบเจาะจงภาษานั้นมีวิธีการอย่างง่าย คือ การเลือกเก็บเว็บเพจภายใต้โดเมนระดับบนสุดของประเทศที่ใช้ภาษานั้นเป็นหลัก เช่น .th สำหรับเว็บเพจภาษาไทย หรือ .jp สำหรับเว็บเพจภาษาญี่ปุ่น แต่อย่างไรก็ตามวิธีนี้ยังไม่ครอบคลุมถึงการเก็บเว็บเพจที่อยู่ตามโดเมนทางธุรกิจ เช่น .com .org .net ฯลฯ ซึ่งเนื้อหาของเว็บเพจในกลุ่มนี้ก็มีความสำคัญไม่น้อยไปกว่าเว็บเพจที่อยู่ภายใต้โดเมนของประเทศเช่นกัน จากปัญหาดังกล่าวจึงจำเป็นต้องอาศัยโปรแกรมเว็บคราวเลอร์แบบเจาะจงภาษา เพื่อค้นหาและเก็บเว็บเพจตามภาษาที่ต้องการได้อย่างมีประสิทธิภาพ ทั้งในด้านการใช้ทรัพยากรระบบและระยะเวลาในการเก็บเว็บเพจ

การทำงานของเว็บคราวเลอร์เจาะจงภาษาแบบดั้งเดิม จะพิจารณาในระดับของเว็บเพจ กล่าวคือ เว็บคราวเลอร์จะพิจารณาตามลิงค์ที่สกัดได้จากเว็บเพจที่เพิ่งเก็บมาหรือไม่ ขึ้นอยู่กับการพิจารณาจากคุณลักษณะของเว็บเพจต้นทาง เช่น ภาษาของเว็บเพจต้นทาง ภาษาที่ใช้บน

แองเคอร์เท็กซ์ (anchor text) ระยะห่างจากเว็บเพจภาษาที่ต้องการ เป็นต้น ซึ่งในปัจจุบันจำนวนเว็บเพจมีแนวโน้มเพิ่มขึ้นอย่างต่อเนื่อง การพิจารณาทีละเว็บเพจอาจไม่เหมาะสมนัก นอกจากนี้จากการศึกษาและวิเคราะห์เว็บกราฟของกลุ่มเว็บเพจภาษาไทย (Somboonviwat *et al.*, 2006) พบว่าเว็บเพจที่ถูกเขียนด้วยภาษาเดียวกัน มักจะอยู่ใกล้ๆ กัน และมีการเชื่อมโยงกันอย่างหนาแน่น หรือมีคุณลักษณะของ language locality นั่นเอง

วิทยานิพนธ์นี้นำเสนอระบบต้นแบบเว็บไซต์คราวเลอร์แบบเจาะจงเว็บเพจภาษาไทยที่มุ่งเน้นหาโฮสต์ที่ให้บริการเว็บเพจภาษาไทยเป็นหลัก ซึ่งระบบต้นแบบประกอบด้วย 2 ส่วน ได้แก่ เว็บคราวเลอร์แบบแยกเก็บตามไซต์ และส่วนการทำนายภาษาของเว็บไซต์เป้าหมาย ซึ่งในส่วนของเว็บคราวเลอร์แบบแยกเก็บตามไซต์มีหน้าที่ค้นหาเส้นทางและเก็บรวบรวมเว็บเพจภาษาไทยภายใต้เว็บไซต์ที่กำหนดได้อย่างมีประสิทธิภาพ แต่ส่วนการทำนายภาษาของเว็บไซต์เป้าหมายมีหน้าที่พิจารณาเว็บไซต์เป้าหมาย ว่ามีโอกาสที่จะให้บริการเว็บเพจภาษาไทยหรือไม่ ซึ่งในส่วนนี้อาศัยเทคนิคทางเครื่องจักรเรียนรู้ที่เรียนรู้จากกลุ่มเว็บไซต์ตัวอย่างด้วยคุณลักษณะทั้ง 4 กลุ่ม ได้แก่ กลุ่มคุณลักษณะโดเมน ที่ตั้งของเว็บเซิร์ฟเวอร์ จำนวนของเว็บไซต์ต้นทาง และระดับความเป็นไทยของเว็บไซต์ต้นทาง ในส่วนของการทดลองได้เปรียบเทียบประสิทธิภาพการทำงานของเว็บคราวเลอร์โดยให้เว็บคราวเลอร์แต่ละแบบเก็บเว็บเพจจากอินเทอร์เน็ต ซึ่งจากผลการทดลองพบว่า การระบุหาเว็บไซต์ที่มีโอกาสให้บริการเว็บเพจภาษาที่ต้องการก่อนกระบวนการเก็บเว็บเพจ มีส่วนช่วยให้ระบบต้นแบบเว็บไซต์คราวเลอร์ที่นำเสนอมีอัตราการเก็บเกี่ยว (harvest rate) เว็บเพจภาษาไทยที่ดีกว่าเว็บคราวเลอร์เจาะจงภาษาแบบดั้งเดิม

วัตถุประสงค์

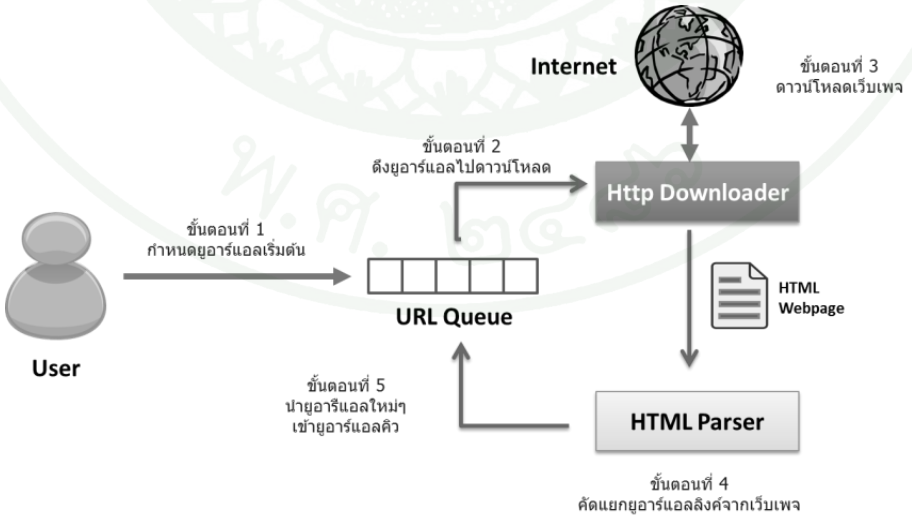
พัฒนาระบบต้นแบบไซต์คราวเลอร์แบบเจาะจงเว็บเพจภาษาไทย โดยอาศัยเทคนิคทางเครื่องจักรเรียนรู้ เพื่อค้นหาเว็บไซต์ที่ให้บริการเว็บเพจภาษาไทยบนอินเทอร์เน็ตได้อย่างมีประสิทธิภาพ

การตรวจเอกสาร

งานวิจัยด้านเว็บคราวเลอร์เจาะจงภาษาต้องอาศัยความรู้พื้นฐาน ได้แก่ หลักการทำงานของเว็บคราวเลอร์อย่างคร่าวๆ การออกแบบเว็บคราวเลอร์เจาะจงภาษา และเทคนิคทางเครื่องจักรเรียนรู้แบบหลายตัว เพื่อนำมาสร้างระบบทำนายภาษาของเว็บไซต์เป้าหมายได้อย่างมีประสิทธิภาพ

หลักการทำงานของเว็บคราวเลอร์

เว็บคราวเลอร์ (web crawler) (Cho and Garcia-Molina, 2002) เป็นส่วนหนึ่งของระบบสืบค้นข้อมูลเว็บ (web search engine) โดยเป็นโปรแกรมที่มีหน้าที่ค้นหา และเก็บเว็บเพจจากอินเทอร์เน็ต ซึ่งมีขั้นตอนการทำงานอย่างคร่าวๆ ดังภาพที่ 1 เริ่มต้นจากการที่ผู้ใช้ (user) กำหนดกลุ่มของยูอาร์แอลเริ่มต้น (seed URLs) ให้กับยูอาร์แอลคิว (URL Queue) จากนั้นในขั้นตอนที่ 2 เว็บคราวเลอร์จะดึงยูอาร์แอลจากคิวเพื่อไปเก็บเว็บเพจตามที่ระบุไว้ในยูอาร์แอลนั้นดังขั้นตอนที่ 3 ซึ่งถ้าเก็บเว็บเพจมาได้สำเร็จในขั้นตอนที่ 4 ส่วนคัดแยกยูอาร์แอลลิงค์ (HTML Parser) จะทำการสกัดหายูอาร์แอลใหม่ๆ จากเว็บเพจที่เพิ่งเก็บมา และในขั้นตอนสุดท้ายจะทำการเพิ่มยูอาร์แอลที่พบใหม่ลงไปในยูอาร์แอลคิว และจะทำงานแบบนี้ซ้ำไปเรื่อยๆ จนกว่าผู้ดูแลเว็บคราวเลอร์จะสั่งให้หยุดทำงาน หรือในยูอาร์แอลคิวไม่มียูอาร์แอลแล้ว หรือเนื้อที่สำหรับบันทึกข้อมูลเต็ม ซึ่งรายละเอียดส่วนประกอบต่างๆ ของเว็บคราวเลอร์มีดังนี้



ภาพที่ 1 การทำงานของเว็บคราวเลอร์อย่างคร่าวๆ

1. ยูอาร์แอลคิว (URL queue)

ยูอาร์แอลคิวนั้นเป็นโครงสร้างข้อมูลชนิดหนึ่ง ที่ใช้เก็บยูอาร์แอลของเว็บเพจที่เว็บคราวเลอร์ต้องการไปเก็บ โดยก่อนที่เพิ่มยูอาร์แอลลงไปในยูอาร์แอลคิวนั้น จะต้องตรวจสอบก่อนว่ายูอาร์แอลนั้นเคยไปเก็บมาแล้วเหรือไม่ เพื่อป้องการไม่ให้เว็บคราเลอร์ไปรบกวนเว็บเซิร์ฟเวอร์ปลายทางมากจนเกินไป และลดการเก็บข้อมูลเว็บเพจที่ซ้ำซ้อนจนทำให้เสียทรัพยากรระบบโดยเปล่าประโยชน์นั่นเอง นอกจากนี้เว็บคราวเลอร์บางชนิดได้มีการออกแบบยูอาร์แอลคิวให้สามารถกำหนดลำดับของยูอาร์แอลตามระดับความสำคัญได้ (priority URL queue) แทนที่จะใช้วิธีการค้นหาที่นิยม ได้แก่ การค้นหาแบบกว้างก่อน หรือการค้นหาแบบลึกก่อน เป็นต้น ตัวอย่างของเว็บคราวเลอร์ที่ใช้คิวลำดับสำคัญ เช่น เว็บคราเลอร์เจาะจงภาษา ที่ออกแบบให้ยูอาร์แอลคิวสามารถจัดลำดับยูอาร์แอลได้ โดยที่ยูอาร์แอลของเว็บเพจภาษาที่สนใจจะอยู่ในลำดับต้นๆ ของคิวนั้นเสมอ

2. ส่วนเก็บเว็บเพจ (HTTP downloader)

เป็นส่วนประกอบที่ใช้ในการติดต่อรับข้อมูลจากเว็บเซิร์ฟเวอร์ปลายทาง โดยเมื่อได้รับยูอาร์แอลจากคิวมาแล้ว จะแยกยูอาร์แอลที่ได้ออกเป็น 4 ส่วนคือ ชื่อเครื่องเซิร์ฟเวอร์ (Server), โปรโตคอล (Protocol), พอร์ต (Port) และพาธของไฟล์ข้อมูล (Path) ดังตัวอย่างตารางที่ 1

ตารางที่ 1 ตัวอย่างการแยกส่วนประกอบจากยูอาร์แอล

ยูอาร์แอล	โปรโตคอล	เซิร์ฟเวอร์	พอร์ต	พาธ
http://www.ku.ac.th	http	www.ku.ac.th	80	/
http://www.ku.ac.th/index.html	http	www.ku.ac.th	80	/index.html
http://www.ku.ac.th:8080/	http	www.ku.ac.th	8080	/

หลังจากที่แยกส่วนประกอบของยูอาร์แอลได้แล้ว ส่วนเก็บเว็บเพจก็จะนำชื่อของเซิร์ฟเวอร์มาแปลงเป็นหมายเลขไอพีแอดเดรส (IP Address) ด้วยการติดต่อไปยังดีเอ็นเอสเซิร์ฟเวอร์ (DNS server) เสียก่อน เช่น www.ku.ac.th จะถูกแปลงเป็นไอพี 158.108.216.5 เป็นต้น จากนั้นจึงนำเลขไอพีแอดเดรสที่แปลงได้ มาใช้ติดต่อกับเซิร์ฟเวอร์ปลายทางผ่านทางพอร์ต และโปรโตคอลของเซิร์ฟเวอร์ที่กำหนดตามยูอาร์แอล เพื่อร้องขอ (Request) แฟ้มข้อมูลตามพาธที่ระบุ

ไว้ เมื่อเว็บเซิร์ฟเวอร์ได้รับคำร้องขอดังกล่าว เว็บเซิร์ฟเวอร์จะค้นหาแฟ้มเอกสารตามพาธที่ได้รับ ซึ่งถ้าพบก็จะส่งข้อมูลดังกล่าวเป็นกระแสข้อมูล (Stream) กลับมายังส่วนเก็บเว็บเพจเพื่อจัดเก็บ และส่งข้อมูลดังกล่าวต่อไปยังส่วนคัดแยกยูอาร์แอลลิงค์ (HTML Parser) ต่อไป

3. ส่วนคัดแยกยูอาร์แอลลิงค์ (HTML Parser)

ข้อมูลเอกสารเว็บเพจนั้นจะถูกเขียนขึ้นมาด้วยภาษาเอชทีเอ็มแอล (HTML) ซึ่งจะประกอบไปด้วยแท็ก (tag) คุณลักษณะของแท็ก (attribute) และเนื้อหา (content) ที่แทรกอยู่ระหว่างแท็ก เมื่อเวลาผู้ใช้เรียกดูหน้าเว็บเพจผ่านเว็บบราวเซอร์ เว็บบราวเซอร์ก็จะแปลงแท็กและเนื้อหาเหล่านั้นให้อยู่ในรูปกราฟฟิกและแสดงผลกลับไปยังผู้ใช้ ซึ่งหน้าที่หลักของส่วนคัดแยกยูอาร์แอลลิงค์ ก็คือ สกัดยูอาร์แอลลิงค์ออกจากแท็ก (Tag) ที่ใช้ระบุถึงการเชื่อมโยงไปยังเว็บเพจอื่น แล้วนำยูอาร์แอลที่คัดแยกมาได้ไปตรวจสอบหาว่าเคยให้เว็บคราวเลอร์ไปเก็บข้อมูลมาหรือยัง ถ้ายังไม่เคยก็จะนำไปเพิ่มลงในยูอาร์แอลคิวต่อไป สำหรับแท็กที่ถูกพิจารณาเพื่อสกัดหาลิงค์การเชื่อมโยงนั้น ได้แก่ A, FRAME, META, และ AREA ดังตัวอย่างในตารางที่ 2

ตารางที่ 2 แท็กที่นิยมใช้เพื่อเชื่อมโยงเว็บเพจ

แท็ก	ตัวอย่างการใช้แท็ก
A	<A href=”http://www.ku.ac.th”>KU</A>
AREA	<AREA shape=”0,0,10,10” href=”http://www.ku.ac.th”>
FRAME	<FRAME src=”http://www.ku.ac.th”>
META	<META http-equive=”refresh” content=”1;url=http://www.ku.ac.th”>

การออกแบบเว็บคราวเลอร์เจาะจงภาษา

เว็บคราวเลอร์เจาะจงภาษาถูกออกแบบมาเพื่อใช้สำหรับค้นหาและเก็บรวบรวมเว็บเพจในภาษาที่สนใจ โดยไม่ขึ้นอยู่กับโดเมนของประเทศที่ใช้ภาษานั้น ตัวอย่างเช่น เว็บเพจภาษาไทย มีกระจายอยู่ทั้งในโดเมนของประเทศไทย (.th) และโดเมนทางธุรกิจอื่นๆ (.com, .net, .org) เป็นต้น เพราะฉะนั้นนักออกแบบเว็บคราวเลอร์เจาะจงภาษาจึงให้ความสำคัญกับการคัดเลือกหาเส้นทางที่จะนำไปสู่เว็บเพจภาษาที่สนใจมากที่สุดก่อน ซึ่งส่วนประกอบที่สำคัญของเว็บคราวเลอร์เจาะจง

ภาษา จะมีอยู่ 2 ส่วนที่สำคัญ ได้แก่ ส่วนตรวจสอบภาษาของเว็บเพจ และส่วนทำนายภาษาหรือส่วนคัดเลือกเส้นทางนั่นเอง โดยมีรายละเอียดดังต่อไปนี้

1. ส่วนตรวจสอบภาษา (language identification) เป็นส่วนประกอบที่สำคัญ เพราะถ้าส่วนตรวจสอบภาษาของเว็บเพจทำงานผิดพลาด จะส่งผลกระทบต่อการทำงานของส่วนทำนายภาษาเป็นอย่างมาก เพราะวิธีการทำนายว่าลิงค์ที่ปรากฏบนเว็บเพจมีโอกาสที่จะชี้ไปยังเว็บเพจที่สนใจหรือไม่ มักจะพิจารณาจากคุณลักษณะ (feature) ของเว็บเพจต้นทางเป็นหลัก เช่น ระดับภาษาของเว็บเพจต้นทาง ระดับภาษาที่ปรากฏบนแองเคอร์แท็กซ์ (anchor text) เป็นต้น ซึ่งวิธีการตรวจสอบภาษาของเว็บเพจมีด้วยกันหลายวิธี ได้แก่

1.1 การตรวจสอบด้วยรหัสภาษาของเว็บเพจ (encoding) ซึ่งมักจะมีการประกาศไว้ที่แท็ก META ตัวอย่างการประกาศรหัสภาษาเช่น <meta http-equiv="Content-Type" content= "text/html; charset=ISO-8859-1" /> จากตัวอย่างได้ประกาศใช้รหัสภาษาเป็น ISO-8859-1 แต่สำหรับรหัสภาษาของเว็บเพจภาษาไทย มีด้วยกัน 3 แบบ คือ UTF-8, Windows-874 และ TIS-620

1.2 การตรวจสอบโดยการใช้วิธีการดูลำดับการเรียงของตัวอักษร n ตัว (n-gram) ตัวอย่างเช่น คำว่า weather ถ้าใช้วิธีการ 3-gram จะได้การแบ่งข้อความดังนี้ _we, wea, eat, ath, the, her, er_ โดยที่ _ แทนช่องว่าง 1 ช่อง ซึ่งวิธีการตรวจสอบภาษาวิธีนี้จะใช้วิธีการจดจำแบบการเรียงตัวอักษรของภาษาที่ต้องการ ตัวอย่างเช่น ถ้ามีรูปแบบเป็น ery_ และ eux_ มีความน่าจะเป็นที่คำนั้นเขียนด้วยภาษาอังกฤษ และภาษาฝรั่งเศสตามลำดับ (Baykan *et al.*, 2008) นอกจากนี้ยังมีโปรแกรมที่ใช้ตรวจสอบภาษาโดยใช้หลักการนี้ ได้แก่ โปรแกรมเท็กซ์เคท (Cavnar and Trenkle, 1994)

1.3 การตรวจสอบภาษา โดยนับจำนวนคำที่ปรากฏบนหน้าเว็บเพจ และยังพบในพจนานุกรมของภาษาที่ต้องการด้วย ซึ่งข้อจำกัดของวิธีการนี้คือ จำเป็นต้องเข้ารหัสภาษา (encoding) ของเว็บเพจให้ถูกต้องก่อนการตรวจสอบภาษา ตัวอย่างโปรแกรมที่ใช้เทคนิคนี้ ได้แก่ โปรแกรมเล็กซ์โต (LexTo, 2011)

2. ส่วนทำนายภาษา หรือส่วนคัดเลือกเส้นทาง (language predictor) ในส่วนนี้จะสกัดคุณลักษณะที่เป็นประโยชน์จากเว็บเพจต้นทาง เพื่อมาใช้คัดเลือกเส้นทางที่น่าจะเป็นไปได้ จากลิงค์ทั้งหมดที่ค้นพบในหน้าเว็บเพจนั้น เพื่อมุ่งเน้นให้เว็บคราวเลอร์เก็บเฉพาะเว็บเพจที่สนใจให้มากที่สุดเท่าที่จะเป็นไปได้ ตัวอย่างการคัดเลือกเส้นทาง เช่น การกำหนดเงื่อนไขต่างๆ หรือสร้างกฎ

ฮิวริสติกส์ (heurestic) เพื่อใช้ในการเลือกเส้นทาง หรือยกเลิกเส้นทางนั้น (Somboonviwat *et al.*, 2006) หรืออีกวิธีการหนึ่งคือการใช้เทคนิคทางเครื่องจักรเรียนรู้ (Srisukha *et al.*, 2008) เป็นต้น

จากการตรวจเอกสารพบว่า มีงานวิจัยเกี่ยวกับเว็บคราวเลอร์เจาะจงภาษา ทั้งภาษาไทย และภาษาอื่นถูกตีพิมพ์อยู่เรื่อยๆ ตั้งแต่ พ.ศ. 2549 จนกระทั่งถึงปัจจุบัน โดยมีประวัติการนำเสนองานวิจัยที่เกี่ยวข้องกับเว็บคราวเลอร์เจาะจงภาษา โดยเรียงลำดับตามปีที่ตีพิมพ์ ดังตารางที่ 3 และพบว่าเว็บคราวเลอร์เจาะจงภาษาไทยมีงานวิจัยออกมาอย่างต่อเนื่อง

ตารางที่ 3 ประวัติการตีพิมพ์ผลงานวิจัยเกี่ยวกับเว็บคราวเลอร์เจาะจงภาษา

ปี ค.ศ.	ภาษาที่สนใจ	วิธีการที่ใช้เพื่อค้นหาเว็บเพจที่สนใจ	เป้าหมายของเว็บคราวเลอร์
Somboonviwat *et al.* (2005)	ภาษาไทย	กฎฮิวริสติกส์	ระดับเว็บเพจ
Medelyan *et al.* (2006)	ภาษาเยอรมัน	-	ระดับเว็บเพจ
Tamura *et al.* (2007)	ภาษาไทย	กฎฮิวริสติกส์	ระดับเว็บเพจ
Srisukha *et al.* (2008)	ภาษาไทย	เครื่องจักรเรียนรู้	ระดับเว็บเพจ
ธนพล และคณะ (2553)	ภาษาไทย	เครื่องจักรเรียนรู้	ระดับเว็บไซต์
Tadapak *et al.* (2010)	ภาษาไทย	เครื่องจักรเรียนรู้	ระดับเว็บไซต์
Alabbad *et al.* (2009)	ภาษาอารบิค	กฎฮิวริสติกส์	ระดับเว็บเพจ
Abdeen *et al.* (2010)	ภาษาอารบิค	กฎฮิวริสติกส์	ระดับเว็บเพจ
Chan *et al.* (2010)	ภาษาจีน, ญี่ปุ่น และเกาหลี	กฎฮิวริสติกส์	ระดับเว็บเพจ
Azimzadeh *et al.* (2010)	ภาษาเปอร์เซีย	กฎฮิวริสติกส์	ระดับเว็บเพจ
Mon *et al.* (2011)	ภาษาพม่า	กฎฮิวริสติกส์	ระดับเว็บเพจ

สถิติฐานข้อมูลเว็บภาษาไทย

ในปี พ.ศ. 2546 Sanguanpong *et al.* (2003) ได้เก็บรวบรวมสถิติเว็บเพจภาษาไทย โดยเริ่มต้นเก็บในเดือนมีนาคม เป็นเวลาทั้งหมด 7 วัน โดยมีการเลือกเก็บเว็บเพจภาษาไทย โดยการ

พิจารณาจากโดเมนของเว็บไซต์ต้องเป็นโดเมนของประเทศไทย (.th) หรือถ้ามีโดเมนอื่นๆ ให้พิจารณาจากที่ตั้งของเว็บเซิร์ฟเวอร์ต้องตั้งอยู่ในประเทศไทย ซึ่งข้อมูลทางสถิติของเว็บเพจภาษาไทยแสดงในตารางที่ 4 ซึ่งจะพบว่าจำนวนเว็บเพจส่วนใหญ่จะอยู่ในโดเมน .ac.th และโดเมน .com พบเว็บไซต์ที่ตั้งอยู่ประเทศไทยมากที่สุด

ตารางที่ 4 สถิติเว็บเพจภาษาไทย ในเดือนมีนาคม พ.ศ. 2546 (Sanguanpong *et al.,* 2003)

โดเมน	จำนวนเว็บเพจ	จำนวนเว็บไซต์
.ac.th	1,093,388	2,979
.com	977,478	10,385
.go.th	313,109	839
.co.th	279,532	6,159
.or.th	236,342	651
.org	107,188	481
.net	87,700	764
โดเมนอื่นๆ	56,361	388
.net.th	55,169	102
.in.th	36,658	748
.edu	20,500	561
.mi.th	14,563	67
รวม	3,277,988	24,124

ตารางที่ 5 สถิติเว็บเพจภาษาไทยในปี พ.ศ. 2549 (Somboonviwat *et al.,* 2006)

โดเมน	จำนวนเว็บเพจ
.th	588,082
.com	903,792
.net	70,777
โดเมนอื่นๆ	143,587
รวม	1,706,238

ต่อมาในปี พ.ศ. 2549 Somboonviwat *et al.* (2006) ได้เสนอเว็บคราวเลอร์เจาะจงเว็บเพจภาษาไทย ซึ่งได้นำเสนอสถิติเว็บภาษาไทย ดังตารางที่ 5 พบว่าจากจำนวนเว็บเพจที่จำแนกได้ว่าเป็นเว็บเพจภาษาไทยประมาณ 1.7 ล้านเว็บเพจนั้น เว็บเพจภาษาไทยส่วนใหญ่อยู่ที่โดเมน .com และ .th ตามลำดับ

สถิติล่าสุดในเดือนมิถุนายน พ.ศ. 2554 จากเว็บไซต์ http://all.in.th/ ซึ่งเป็นเว็บไซต์ที่เก็บรวบรวมการจดทะเบียนเว็บไซต์ระดับบนสุดเป็น .th แสดงดังตารางที่ 6 แต่จำนวนเว็บไซต์ที่แสดงทั้งหมดนี้ ยังไม่ได้ผ่านการตรวจสอบว่าเป็นเว็บไซต์ภาษาไทยหรือไม่

ตารางที่ 6 สถิติเว็บเพจภาษาไทยในเดือนมิถุนายน พ.ศ. 2554

โดเมน	จำนวนเว็บไซต์
.ac.th	5,138
.co.th	25,075
.go.th	5,245
.in.th	11,984
.mi.th	27
.net.th	28
.or.th	982
รวม	48,479

มาตรวัดประสิทธิภาพของเว็บคราวเลอร์

โดยทั่วไปแล้วการเปรียบเทียบประสิทธิภาพของเว็บคราวเลอร์อย่างง่าย สามารถวัดได้จากการหาอัตราส่วนระหว่างจำนวนเว็บเพจที่สนใจกับจำนวนเว็บเพจทั้งหมดที่เก็บมา หรือที่เรียกว่าอัตราการเก็บเกี่ยว (harvest rate) นั่นเอง นอกจากนี้ยังมีมาตรวัดที่สำคัญอีกหลายแบบ เพื่อช่วยในการเปรียบเทียบประสิทธิภาพของเว็บคราวเลอร์ดังนี้

1. ความแม่นยำ (accuracy) เพื่อใช้ในการเปรียบเทียบความแม่นยำของระบบทำนายต่างๆ ว่าทำนายผลได้ถูกต้องหรือไม่

$$\text{ความแม่นยำ} = \frac{\text{จำนวนตัวอย่างที่ทำนายได้ถูกต้อง}}{\text{จำนวนตัวอย่างทั้งหมด}} \quad (1)$$

2. อัตราการเก็บเกี่ยว (harvest rate) เพื่อเปรียบเทียบอัตราส่วนระหว่างจำนวนเว็บเพจที่สนใจกับจำนวนเว็บเพจทั้งหมด เพราะฉะนั้นถ้ามีค่าเข้าใกล้ 1 มากๆ แสดงว่าเว็บคราวเลอร์นั้นมีโอกาสเก็บเว็บเพจที่ไม่เกี่ยวข้องน้อยมากๆ

$$\text{อัตราการเก็บเกี่ยว} = \frac{\text{จำนวนเว็บเพจที่สนใจ}}{\text{จำนวนเว็บเพจที่เก็บมาทั้งหมด}} \quad (2)$$

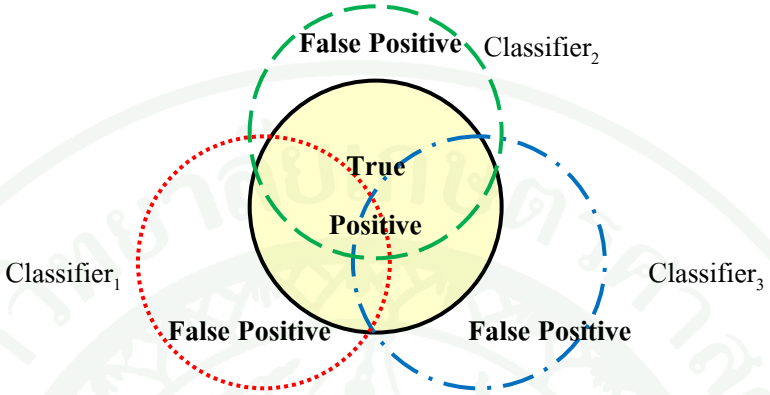
3. ความครอบคลุม (coverage) เพื่อเปรียบเทียบดูว่า เว็บคราวเลอร์ที่ออกแบบนั้นสามารถเก็บเว็บเพจที่สนใจ มาจากกลุ่มของเว็บเพจที่สนใจที่มีอยู่ทั้งหมดมาได้มากน้อยเพียงใด ซึ่งถ้าต้องการออกแบบให้เว็บคราวเลอร์มีประสิทธิภาพ นอกจากจะสนใจปรับปรุงให้เว็บคราวเลอร์มีอัตราการเก็บเกี่ยวที่สูง แล้วยังต้องคำนึงถึงความครอบคลุมด้วย เพื่อให้สามารถเก็บเว็บเพจที่สนใจได้อย่างครบถ้วน

$$\text{ความครอบคลุม} = \frac{\text{จำนวนเว็บเพจที่สนใจที่เก็บมาได้}}{\text{จำนวนเว็บเพจที่สนใจที่มีอยู่ทั้งหมด}} \quad (3)$$

เทคนิคเครื่องจักรเรียนรู้แบบหลายตัว

การใช้เครื่องจักรเรียนรู้เพื่อจำแนกประเภทของเอกสาร ส่วนใหญ่จะเลือกใช้เครื่องจักรเรียนรู้ที่ให้ประสิทธิภาพที่ดีที่สุดเพียงหนึ่งตัว เพื่อความสะดวกและรวดเร็วในการสร้างระบบจำแนกเอกสาร ทำให้เมื่อนำเครื่องจักรเรียนรู้ไปใช้งานจริง มีโอกาสที่ทำนายผิดพลาดได้สูง เพราะพบกับข้อมูลที่ไม่เคยเห็นมาก่อน ซึ่งวิธีการปรับปรุงประสิทธิภาพเครื่องจักรเรียนรู้เพียงหนึ่งตัวให้สามารถทำนายผลได้ถูกต้องทุกกรณีทำได้ยากมาก ดังนั้นเพื่อให้เครื่องจักรเรียนรู้มีความทนทานต่อการนำไปใช้งาน แนวทางหนึ่งคือ การสร้างระบบเครื่องจักรเรียนรู้แบบหลายตัว เปรียบเสมือนการเพิ่มผู้เชี่ยวชาญให้กับระบบทำนาย ตามภาพที่ 2 สมมติให้คลาสตำตอบมีสองคลาส คือ คลาสบวก (+) และคลาสลบ (-) และให้เส้นทึบเป็นเซ็ตของคำตอบที่ถูกต้องของคลาส + เพราะฉะนั้น ถ้าระบบทำนายมี classifier_1 เพียงตัวเดียว ระบบนี้ก็สามารถจดจำลักษณะของคลาส + ได้เพียงบางส่วนเท่านั้น แต่เมื่อเพิ่ม classifier_2 และ classifier_3 เข้าไป พบว่า ถ้าใช้ classifier ทั้งสามตัวจะสามารถจำแนกข้อมูลของคลาส + ได้ครอบคลุมเซ็ตคำตอบทั้งหมด แต่ผลกระทบที่ตามมาคือ มี

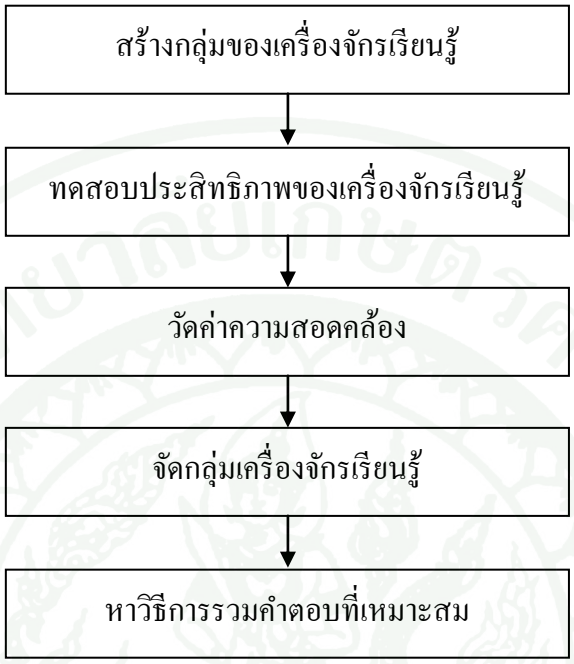
ความผิดพลาดเพิ่มมากขึ้น ดังจะเห็นได้จากพื้นที่ของ False Positive ที่เพิ่มขึ้น ซึ่งสามารถกรองเฉพาะคำตอบที่ถูกต้องได้ โดยการเลือกใช้วิธีการรวมคำตอบ (fusion method) ให้เหมาะสมกับกลุ่มของเครื่องจักรเรียนรู้ที่เลือกนำมาใช้งาน



ภาพที่ 2 ขอบเขตความสามารถของเครื่องจักรเรียนรู้แต่ละตัว

การสร้างระบบเครื่องจักรเรียนรู้แบบหลายตัว ในกรณีที่มีเครื่องจักรเรียนรู้ตัวเดิมอยู่แล้วแต่ต้องการเพิ่มเครื่องจักรเรียนรู้ตัวใหม่เข้าไปในระบบ จากแนวทางของ Goeble (2004) มีขั้นตอนคร่าวๆ ตามภาพที่ 3 ดังนี้ เริ่มต้นจากการสร้างเครื่องจักรเรียนรู้เตรียมไว้ล่วงหน้า โดยอาจจะให้เครื่องจักรเรียนรู้แต่ละตัว เรียนรู้จากจำนวนคุณลักษณะ (feature) ที่แตกต่างกัน (เช่น เครื่องจักรเรียนรู้บางตัวใช้ 5 คุณลักษณะ หรือบางตัวใช้ 10 คุณลักษณะในการเรียนรู้) หรือใช้เทคนิคทางเครื่องจักรเรียนรู้ที่แตกต่างกัน (เช่น Naïve Bayes, SVM, Decision Tree C4.5) เนื่องจากเทคนิคทางเครื่องจักรเรียนรู้แต่ละแบบ มีความสามารถในการเรียนรู้ตัวอย่างข้อมูลไม่เท่ากัน ส่งผลให้คำตอบที่เครื่องจักรเรียนรู้ทำนายในบางกรณีมีความแตกต่างกัน ซึ่งในบางกรณีที่เครื่องจักรเรียนรู้ตัวหนึ่งให้ผลการทำนายที่ผิดพลาด แต่เครื่องจักรเรียนรู้อีกตัวอาจจะทำนายผลได้ถูกต้องก็มีโอกาสเป็นไปได้ หลังจากได้จำนวนเครื่องจักรเรียนรู้ตามที่ต้องการแล้ว จากนั้นในขั้นตอนที่ 2 ทำการทดสอบประสิทธิภาพของเครื่องจักรเรียนรู้แต่ละตัวด้วยวิธีการตรวจสอบไขว้กลับ n พับ (n-fold cross validation) กับชุดข้อมูลฝึกสอน และในขั้นตอนที่ 3 เป็นการจัดกลุ่มเครื่องจักรเรียนรู้เพื่อหาเครื่องจักรเรียนรู้ที่ทำงานร่วมกันแล้ว ทำให้มีประสิทธิภาพมากกว่าการใช้เครื่องจักรเรียนรู้เพียงตัวเดียว แต่ปัญหาที่ตามมา ก็คือ จำนวนกลุ่มของเครื่องจักรเรียนรู้มีจำนวนมากมาย ดังนั้นเราจึงต้องหาวิธีการเพื่อจัดกลุ่มของเครื่องจักรเรียนรู้ที่เหมาะสม ซึ่ง Goeble (2004) เสนอให้ใช้มาตรวัดความสอดคล้อง (correlation measure) โดยจะกล่าวรายละเอียดในหัวข้อต่อไป หลังจากที่

ได้กลุ่มของเครื่องจักรเรียนรู้ที่เหมาะสมเรียบร้อยแล้ว ในขั้นตอนสุดท้าย คือ การทดสอบหาวิธีการรวมคำตอบที่เหมาะสมให้กับแต่ละกลุ่มของเครื่องจักรเรียนรู้และนำไปใช้งานต่อไปได้



ภาพที่ 3 ภาพรวมของขั้นตอนการสร้างระบบเครื่องจักรเรียนรู้แบบหลายตัว

1. มาตรวัดความสอดคล้อง (correlation measure)

ในการสร้างระบบเครื่องจักรเรียนรู้แบบหลายตัว ต้องพยายามหลีกเลี่ยงการเลือกเครื่องจักรเรียนรู้ที่ให้คำตอบซ้ำซ้อนกัน กล่าวคือ สมมติถ้ามีเครื่องจักรเรียนรู้สองตัวในระบบเครื่องจักรเรียนรู้แต่ละตัว ควรที่จะให้ผลการทำนายที่แตกต่างกัน เนื่องจากจะได้มั่นใจได้ว่าในทุกๆ กรณีจะมีเครื่องจักรเรียนรู้หนึ่งตัวให้ผลการทำนายที่ถูกต้องเสมอ แต่ถ้าหากเครื่องจักรเรียนรู้ทั้งสองตัวให้ผลการทำนายเหมือนกันทุกครั้ง ก็เลือกใช้งานเครื่องจักรเรียนรู้ตัวใดตัวหนึ่งก็เพียงพอแล้ว ดังนั้นการคัดเลือกเครื่องจักรเรียนรู้ตัวใหม่เข้ามาในระบบก็มีความสำคัญเช่นกัน ซึ่งวิธีการอย่างง่าย สามารถทำได้โดย ทดลองจับกลุ่มเครื่องจักรเรียนรู้ทุกกรณีที่เป็นไปได้ ซึ่งจำนวนกลุ่มจะเพิ่มขึ้นอย่างมาก เมื่อจำนวนเครื่องจักรเรียนรู้เพิ่มขึ้น เพื่อเป็นการหลีกเลี่ยงวิธีการจัดกลุ่มในทุกๆ กรณีที่เป็นไปได้ เราจึงจำเป็นต้องใช้มาตรวัดความสอดคล้อง (correlation measure) เพื่อหาความสอดคล้องระหว่างเครื่องจักรเรียนรู้ทั้งสองตัว โดยวิเคราะห์จากเมตริกซ์ความสัมพันธ์ (correlation analysis matrix) ที่นำเสนอโดย (Petrakos, 2000) ดังภาพที่ 4 โดยที่ N^{TT} คือจำนวนตัวอย่างที่เครื่องจักรเรียนรู้ทั้งสองตัวทำนายผลถูกต้อง, N^{FF} คือจำนวนตัวอย่างที่เครื่องจักรเรียนรู้ทั้ง

สองตัวทำนายผิด, N^{TF} คือจำนวนตัวอย่างที่ $classifier_i$ ทำนายถูกต้อง แต่ $classifier_j$ ทำนายผิด และ N^{FT} จำนวนตัวอย่างที่ $classifier_i$ ทำนายผิด แต่ $classifier_j$ ทำนายได้ถูกต้อง

		Classifier$_j$
Classifier$_i$	$\mathbf{N^{TT}}$	$\mathbf{N^{TF}}$
	$\mathbf{N^{FT}}$	$\mathbf{N^{FF}}$

ภาพที่ 4 เมตริกซ์ความสัมพันธ์ (Correlation Analysis Matrix)

มาตรวัดความสอดคล้องที่นิยมโดยทั่วไป มีดังนี้

1.1 Q Statistic (Kuncheva, 2004) ใช้วิเคราะห์ความแตกต่างของเครื่องจักรเรียนรู้ ถ้าค่า **Q** เท่ากับ **0** แสดงว่า เครื่องจักรเรียนรู้ทั้งสองตัวให้คำตอบแตกต่างกันอย่างสิ้นเชิง แต่ถ้า Q มีค่าเข้าใกล้ 1 มากๆ กรณีนี้เลือกใช้เพียงเครื่องจักรเรียนรู้ตัวใดตัวหนึ่งก็เพียงพอแล้ว

$$Q_{i,j} = \frac{N^{TT}N^{FF} - N^{TF}N^{FT}}{N^{TT}N^{FF} + N^{TF}N^{FT}} \quad (4)$$

1.2 ρ-correlation (Petrakos, 2000) วิเคราะห์หากรณีที่เครื่องจักรเรียนรู้ทั้งสองตัวให้ผลการทำนายที่ผิดพลาดเหมือนกัน

$$\rho = \frac{2N^{FF}}{N^{TF} + N^{FT} + 2N^{FF}} \quad (5)$$

1.3 Disagreement Measure (Kuncheva, 2004) วิเคราะห์หาจำนวนตัวอย่างที่เครื่องจักรเรียนรู้ทั้งสองตัว ให้ผลการทำนายที่แตกต่างกัน ซึ่งสมมุติฐานของมาตรวัดนี้ ก็คือ ถ้าเครื่องจักรเรียนรู้ทั้งสองตัวให้คำตอบไม่เหมือนกันเลย แสดงว่า ต้องมีเครื่องจักรเรียนรู้ตัวใดตัวหนึ่งที่ทำนายได้ถูกต้อง เพราะฉะนั้นเราจึงควรเลือกเครื่องจักรเรียนรู้คู่ที่มีค่ามากกว่า 0

$$D_{i,j} = N^{TF} + N^{FT} \quad (6)$$

1.4 Double-Fault Measure (Kuncheva, 2004) หาจำนวนครั้งที่เครื่องจักรเรียนรู้ทั้งสองตัวให้ผลการทำนายผิดพร้อมกัน เพราะฉะนั้นเราจึงควรเลือกเครื่องจักรเรียนรู้คู่ที่มีค่าเท่ากับหรือเข้าใกล้ 0

$$DF_{i,j} = N^{FF} \quad (7)$$

2. วิธีการรวมคำตอบ (fusion method)

หลังจากที่เราได้กลุ่มของเครื่องจักรเรียนรู้เรียบร้อยแล้ว ก็มาถึงขั้นตอนเพื่อหาวิธีการรวมคำตอบที่เหมาะสม ในกรณีที่เครื่องจักรเรียนรู้แต่ละตัวให้คำตอบที่แตกต่างกัน เราจำเป็นต้องกำหนดเงื่อนไข เพื่อหาตัวแทนคำตอบของระบบ ซึ่งวิธีการรวมคำตอบแบ่งเป็น 2 แบบ ตามลักษณะผลลัพท์ที่ได้จากเครื่องจักรเรียนรู้ (Ruta, 2000) คือ

2.1 เครื่องจักรเรียนรู้ให้คำตอบมาเป็น class label ตัวอย่างวิธีการรวมคำตอบเช่น Majority Vote เพื่อเลือกคำตอบที่เครื่องจักรเรียนรู้ตอบมามากที่สุด หรือ Weight Majority Vote คล้ายๆ กับ majority vote แต่มีการให้น้ำหนักหรือความน่าเชื่อถือกับเครื่องจักรเรียนรู้แต่ละตัว

2.2 เครื่องจักรเรียนรู้ให้ค่าสนับสนุนของแต่ละคลาสเป็นตัวเลขจำนวนจริง ที่มีช่วงอยู่ระหว่าง 0 ถึง 1 หรือที่เรียกว่า soft output ซึ่งเครื่องจักรเรียนรู้ในกลุ่มนี้สามารถเลือกใช้วิธีการรวมคำตอบได้หลากหลาย เช่น วิธีการรวมอย่างง่าย (Simple Summary Function) (Ruta and Gabrys, 2000) ได้แก่ MAX (เลือกคลาสคำตอบที่มีผลรวมของค่าสนับสนุนสูงที่สุดเป็นคำตอบสุดท้าย) MIN (เลือกคลาสคำตอบที่มีผลรวมของค่าสนับสนุนน้อยที่สุดเป็นคำตอบสุดท้าย) AVERAGE (เลือกคลาสคำตอบที่มีค่าเฉลี่ยของค่าสนับสนุนในแต่ละคลาสสูงที่สุดเป็นคำตอบสุดท้าย) PRODUCT (เลือกคลาสคำตอบที่มีผลคูณของค่าสนับสนุนสูงที่สุดเป็นคำตอบสุดท้าย) เป็นต้น

อุปกรณ์และวิธีการ

อุปกรณ์

1. อุปกรณ์ฮาร์ดแวร์ (Hardware Equipment)

ในขั้นตอนการพัฒนาและทดสอบระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษา ใช้เครื่องคอมพิวเตอร์แม่ข่ายของภาควิชาวิศวกรรมคอมพิวเตอร์ จำนวน 1 เครื่อง โดยมีรายละเอียดดังนี้

1.1 หน่วยประมวลผลกลาง (CPU) Intel ® Xeon ™ 2.80 GHz จำนวน 4 แกน

1.2 หน่วยความจำหลัก (RAM) ขนาด 4 กิกะไบต์

1.3 ฮาร์ดดิสก์ ขนาด 500 กิกะไบต์

1.4 ระบบเครือข่ายด้วยอัตราเร็ว 1 กิกะบิตต่อวินาที

2. อุปกรณ์ซอฟต์แวร์ (Software Equipment)

2.1 ระบบปฏิบัติการ (Operating System) Linux CentOS รุ่น 5.1

2.2 Java Software Development Kit (JDK) รุ่น 6.0 ใช้คอมไพล์โปรแกรมภาษาจาวา

2.3 โปรแกรม Eclipse รุ่น 3.3 ใช้พัฒนาโปรแกรม

2.4 โปรแกรม InetAddressLocator รุ่น 2.23 ใช้ระบุที่ตั้งของเครื่องเซิร์ฟเวอร์

2.5 โปรแกรม jerhicho HTMLParser รุ่น 3.1 ใช้สกัดคุณลักษณะจากหน้าเว็บเพจ

2.6 โปรแกรมเล็กโต (LexTo) ใช้ตรวจสอบภาษาของเว็บเพจ

2.7 ระบบฐานข้อมูลเบิร์กเลย์ (Berkeley DB) รุ่น 4.0

2.8 โปรแกรมเวกา (Weka) ใช้ฝึกสอนและทดสอบเครื่องจักรเรียนรู้

2.9 โปรแกรมเว็บคราวเลอร์ Heritrix ใช้สร้างฐานข้อมูลเว็บภาษาไทย

2.10 โปรแกรม Apache Tomcat รุ่น 7.0.16 ใช้เป็นเว็บเซิร์ฟเวอร์

วิธีการ

ในส่วนนี้จะอธิบายถึงขั้นตอนการสร้างระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาไทย ตั้งแต่การให้คำนิยามของเว็บเพจภาษาไทยและเว็บไซต์ภาษาไทย การเก็บตัวอย่างเว็บไซต์จากอินเทอร์เน็ต เพื่อมาสร้างเป็นฐานข้อมูลเว็บภาษาไทยสำหรับใช้ฝึกสอนเครื่องจักรเรียนรู้ ขั้นตอนการออกแบบระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษา และส่วนประกอบต่างๆ ทั้งส่วนของ

เว็บคราวเลอร์แบบแยกเก็บตามไซต์ (site crawler) และส่วนของการทำนายภาษาของเว็บไซต์เป้าหมาย (language predictor)

1. นิยามเว็บเพจภาษาไทย และเว็บไซต์ภาษาไทย

1.1 เว็บเพจภาษาไทย (Srisukha *et al.*, 2008) คือ เว็บเพจที่หลังจากกรองแท็ก HTML ออกไปแล้ว พบว่า มีอัตราส่วนของจำนวนคำภาษาไทยที่ปรากฏอยู่ในพจนานุกรมเล็กซิตรอน (LEXiTRON, 2011) อย่างน้อยร้อยละ 10 ของจำนวนคำทั้งหมดที่พบในหน้าเว็บเพจนั้น

1.2 เว็บไซต์ภาษาไทย คือ เว็บไซต์ที่มีจำนวนเว็บเพจภาษาไทยต่อจำนวนเว็บเพจทั้งหมดที่เก็บมาจากเว็บไซต์นั้นมากกว่าร้อยละ 25

2. การสร้างฐานข้อมูลเว็บภาษาไทย

เนื่องจากจำนวนเว็บเพจบนอินเทอร์เน็ตมีจำนวนมาก และมีความหลากหลายทั้งในด้านประเภทของเนื้อหาและภาษาของเว็บเพจ เราจึงต้องหาวิธีการเก็บซับเซ็ตของเว็บเพจจากอินเทอร์เน็ต เพื่อนำมาศึกษาคุณลักษณะของเว็บไซต์ และใช้ในการออกแบบระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาไทย ซึ่งจากงานวิจัยของ Somboonviwat *et al.* (2006) แสดงให้เห็นว่าเว็บเพจภาษาไทยด้วยกันมักจะมีการเชื่อมโยงถึงกัน และอยู่ใกล้ๆ กันในเว็บกราฟ หรือมีคุณสมบัติ language locality นั่นเอง เราจึงเริ่มต้นการสร้างฐานข้อมูลเว็บภาษาไทยจากการคัดเลือกกลุ่มเว็บไซต์เริ่มต้นที่ให้บริการเว็บเว็บเพจภาษาไทยเป็นส่วนใหญ่ โดยที่แต่ละเว็บไซต์มีหัวเรื่องที่แตกต่างกัน ได้แก่

- http://www.ku.ac.th/ (ตัวอย่างเว็บไซต์สถาบันการศึกษา)
- http://www.nectec.or.th/ (ตัวอย่างเว็บไซต์ด้านองค์กร)
- http://www.thairath.co.th/ (ตัวอย่างเว็บไซต์หนังสือพิมพ์)
- http://www.railway.co.th/ (ตัวอย่างเว็บไซต์ธุรกิจ)
- http://www.dmoz.org/world/Thai/ (ตัวอย่างเว็บไซต์ที่จัดหมวดหมู่เว็บไซต์ของประเทศไทย)

เมื่อได้กลุ่มของเว็บไซต์เริ่มต้นเรียบร้อยแล้ว ขั้นตอนต่อไปเป็นการเลือกวิธีการเก็บเว็บเพจ

ของเว็บคราวเลอร์ ซึ่งวิธีที่นิยมมี 2 วิธี คือ วิธีการเก็บแบบกว้างก่อน (breadth-first search) และวิธีการเก็บแบบลึกก่อน (depth-first search) ซึ่งในการสร้างฐานข้อมูลชุดนี้เลือกวิธีการเก็บแบบกว้างก่อน เนื่องจากต้องการให้การเชื่อมโยงของเว็บกราฟภายในแต่ละเว็บไซต์มีความสมบูรณ์มากที่สุด และได้เลือกใช้เว็บคราวเลอร์ Heritrix (2011) มาเป็นเครื่องมือในการเก็บเว็บเพจ โดยมีการตั้งค่าให้ทำงานแบบแยกเก็บตามไซต์ ซึ่งมีความแตกต่างจากการทำงานของเว็บคราวเลอร์แบบดั้งเดิม ตรงที่เว็บคราวเลอร์แบบแยกเก็บตามไซต์จะเริ่มต้นเก็บเว็บเพจหลัก (entry page) ของทุกๆ เว็บไซต์ที่กำหนดไว้ก่อน และค่อยเลือกตามลิงค์ที่ชี้ไปยังเว็บเพจที่อยู่ภายใต้เว็บไซต์ที่กำลังทำงานอยู่เท่านั้น แต่การทำงานของเว็บคราวเลอร์แบบดั้งเดิมจะตามเก็บเว็บเพจทุกๆ ลิงค์ที่พบ นอกจากนี้ยังกำหนดให้เว็บคราวเลอร์เก็บเว็บเพจมาไม่เกินเว็บไซต์ละ 150 เว็บเพจ และกำหนดให้เว็บคราวเลอร์เลือกเก็บเฉพาะเว็บไซต์ที่มีระยะห่างจากเว็บไซต์เริ่มต้นไม่เกิน 4 hop เท่านั้น ซึ่งจากการปล่อยให้เว็บคราวเลอร์เก็บเว็บเพจจากอินเทอร์เน็ตในเดือนกันยายน พ.ศ. 2552 ได้ค่าสถิติของฐานข้อมูลเว็บภาษาไทย ดังตารางที่ 7

ตารางที่ 7 สถิติของฐานข้อมูลเว็บภาษาไทย เดือนกันยายน พ.ศ. 2552

โดเมนระดับบนสุด	ภาษาไทย		ภาษาอื่น	
	จำนวน (เว็บเพจ)	จำนวน (เว็บไซต์)	จำนวน (เว็บเพจ)	จำนวน (เว็บไซต์)
.com	226,295	1,075	333,294	6,656
.net	31,741	157	40,178	767
.org	50,517	165	79,682	1,611
.th	1,677,409	23,229	800,161	8,227
.info	1,851	12	2,942	64
โดเมนอื่นๆ	108,936	476	192,567	3,477
รวม	2,096,749	25,114	1,448,824	20,802

จากตารางที่ 7 พบว่า เว็บไซต์ที่ให้บริการเว็บเพจภาษาไทยส่วนใหญ่มีโดเมนระดับบนสุดเป็นโดเมนของประเทศไทย (.th) ประมาณ 92.49% และยังพบเว็บไซต์ที่ให้บริการเว็บเพจภาษาไทยกระจายอยู่ตามโดเมนทางธุรกิจอื่นๆ อีกประมาณ 7.51% แต่อย่างไรก็ตามภายใต้โดเมน .th ยังพบเว็บไซต์ที่ให้บริการเว็บเพจภาษาอื่นๆ จำนวนไม่น้อยเช่นกัน (39.55%) นอกจากนี้ค่าสถิติของ

ฐานข้อมูลเว็บภาษาไทยชุดนี้ยังมีคุณลักษณะใกล้เคียงกับงานวิจัยของ Somboonviwat *et al.* (2006) ทั้งๆ ที่สร้างจากกลุ่มของเว็บไซต์เริ่มต้นที่แตกต่างกัน

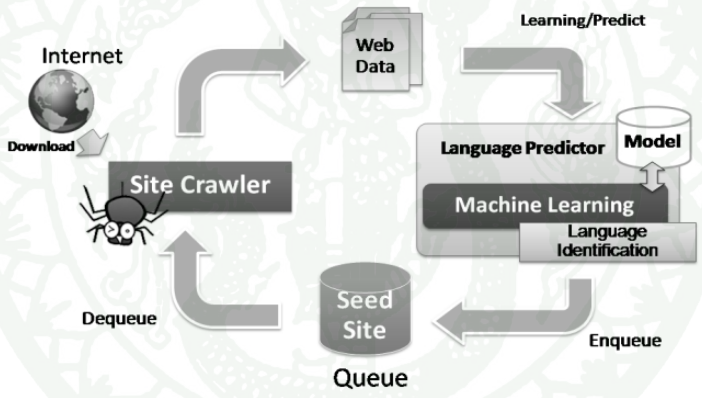
3. ระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษา

การเชื่อมโยงเว็บเพจบนอินเทอร์เน็ตมีความเป็นอิสระ เราจึงไม่สามารถระบุขอบเขตที่ชัดเจนของกลุ่มเว็บเพจที่สนใจได้ ไม่เหมือนกับการหาเส้นแบ่งอาณาเขตของแต่ละประเทศ ซึ่งจากการศึกษาเว็บกราฟของฐานข้อมูลเว็บภาษาไทย พบว่า เว็บเพจภาษาไทยมีการกระจายอยู่ทั้งในและนอกโดเมนของประเทศไทย (.th) ด้วยเหตุนี้การสร้างเว็บคราวเลอร์แบบเจาะจงภาษาที่มีประสิทธิภาพเพื่อรองรับเว็บเพจจำนวนมากบนอินเทอร์เน็ตได้นั้นจึงเป็นงานที่ท้าทาย โดยเมื่อเร็วๆ นี้ มีทีมนักวิจัยได้ออกแบบเว็บคราวเลอร์เจาะจงเว็บเพจภาษาไทยในระดับเว็บเพจ (Somboonviwat *et al.*, 2006; Srisukha *et al.*, 2008) โดยนำเสนอวิธีการจัดเรียงลำดับยูอาร์แอล (URL) ในยูอาร์แอลคิว (URL queue) ด้วยการพิจารณาจากคุณลักษณะของเว็บเพจต้นทาง เพื่อหาความน่าจะเป็นของเว็บเพจปลายทางว่าเป็นเว็บเพจที่สนใจหรือไม่ โดยจัดให้ยูอาร์แอลของเว็บเพจปลายทางที่คาดว่าจะเป็นเว็บเพจภาษาไทยอยู่ในลำดับต้นๆ ของยูอาร์แอลคิว ซึ่งจากวิธีการนำเสนอที่ผ่านมา เว็บคราวเลอร์จำเป็นต้องพิจารณาทุกๆ การเชื่อมโยง (hyperlink) ที่ปรากฏบนหน้าเว็บเพจต้นทาง และยังต้องจัดเรียงยูอาร์แอลคิวใหม่ทุกครั้งที่เพิ่มยูอาร์แอลใหม่เข้าไปในคิว ทำให้เว็บคราวเลอร์เหล่านั้นอาจจะพบปัญหาเมื่อเก็บเว็บเพจจำนวนมากๆ ได้ ดังนั้นงานวิจัยนี้จึงเสนอวิธีการแก้ปัญหาดังกล่าวด้วยการนำเสนอระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาในระดับเว็บไซต์แทน ซึ่งจากการวิเคราะห์โฮสต์กราฟของฐานข้อมูลเว็บภาษาไทย พบว่า นอกจากเว็บเพจภาษาไทยที่มีคุณลักษณะของ language locality แล้ว ในระดับเว็บไซต์ภาษาไทยก็มีคุณสมบัตินี้อีกด้วย ดังตารางที่ 8 แสดงให้เห็นว่า เว็บไซต์ภาษาไทยมักจะเชื่อมโยงไปยังเว็บไซต์ภาษาไทยด้วยกันมากถึง 330,984 การเชื่อมโยง หรือประมาณ 64.95% ในทางกลับกันเว็บไซต์ภาษาอื่นก็เชื่อมโยงไปหาเว็บไซต์ภาษาอื่นจำนวน 136,549 การเชื่อมโยง หรือประมาณ 82.60%

ตารางที่ 8 การเชื่อมโยงระดับเว็บไซต์ระหว่างเว็บไซต์ภาษาไทยและเว็บไซต์ภาษาอื่น

	เว็บไซต์ภาษาไทย (จำนวนการเชื่อมโยง)	เว็บไซต์ภาษาอื่น (จำนวนการเชื่อมโยง)
เว็บไซต์ภาษาไทย	330,984	178,603
เว็บไซต์ภาษาอื่น	28,772	136,549

ระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาไทยแบบแยกเก็บตามไซต์ ประกอบด้วย 2 ส่วนหลักๆ ได้แก่ ส่วนการทำนายภาษาของเว็บไซต์เป้าหมาย (language predictor) และเว็บคราวเลอร์แบบแยกเก็บตามไซต์ (site crawler) ตามภาพที่ 5 ซึ่งหลังจากที่ผู้ดูแลได้เพิ่มรายชื่อกลุ่มเว็บไซต์เริ่มต้นเข้าไปในคิว (seed site queue) แล้ว เว็บคราวเลอร์แบบแยกเก็บตามไซต์จะเริ่มเก็บเว็บเพจจากแต่ละเว็บไซต์ตามจำนวนที่ผู้ดูแลระบบกำหนดไว้ จนครบทุกเว็บไซต์ที่มีในคิว เว็บคราวเลอร์จึงหยุดการทำงาน จากนั้นก็จะเริ่มสกัดคุณลักษณะของเว็บไซต์ปลายทางเพื่อส่งต่อให้ส่วนการทำนายภาษาของเว็บไซต์ซึ่งใช้เทคนิคทางเครื่องจักรเรียนรู้มาทำนายว่า เว็บไซต์เป้าหมายมีโอกาสให้บริการเว็บเพจภาษาไทยหรือไม่ พร้อมทั้งให้คะแนนความมั่นใจเพื่อใช้จัดเรียงลำดับเว็บไซต์เป้าหมายในคิวและในขั้นตอนสุดท้ายจะทำการเพิ่มรายชื่อเว็บไซต์ปลายทางที่เกี่ยวข้องลงไปในคิว เพื่อให้เว็บคราวเลอร์ทำการเก็บเว็บเพจในรอบต่อๆไป ซึ่งรายละเอียดการออกแบบส่วนประกอบต่างๆ มีดังนี้



ภาพที่ 5 ระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาโดยใช้เครื่องจักรเรียนรู้

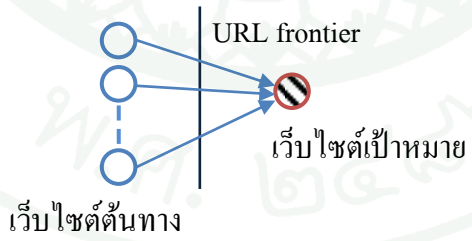
3.1 ส่วนการทำนายภาษาของเว็บไซต์ (language predictor)

ส่วนการทำนายภาษาของเว็บไซต์เป็นส่วนประกอบที่สำคัญ เนื่องจากถ้าทำนายผลผิดพลาด จะส่งผลกระทบอย่างมากต่ออัตราการเก็บเกี่ยวโดยรวมทั้งระบบ ตัวอย่างเช่น ในกรณีที่ส่วนการทำนายภาษาให้ผลการทำนายว่า เว็บไซต์เป้าหมายเป็นเว็บไซต์ที่ให้บริการเว็บเพจภาษาไทย แต่จริงๆ แล้วเว็บไซต์นั้นไม่ได้ให้บริการเว็บเพจภาษาไทย ทำให้ต้องสูญเสียทั้งเวลาแบนด์วิดท์ และเนื้อที่เก็บข้อมูล เพื่อเก็บเว็บเพจภาษาอื่นจำนวนมาก จนกระทั่งเว็บคราวเลอร์หยุดเก็บเว็บเพจจากเว็บไซต์นั้น ในทางกลับกันถ้าส่วนทำนายภาษาให้ผลการทำนายเว็บไซต์ภาษาไทยว่า เป็นเว็บไซต์ภาษาอื่น ก็จะทำให้พลาดเว็บเพจภาษาไทยจำนวนมากเช่นกัน แต่อย่างไรก็ตาม

คุณลักษณะของเว็บไซต์ที่ให้บริการเว็บเพจภาษาไทยมีความหลากหลายมาก เราจึงเลือกใช้วิธีการทางเครื่องจักรเรียนรู้ เพื่อมาเรียนรู้คุณลักษณะของเว็บไซต์ที่ให้บริการเว็บเพจภาษาไทย แทนที่จะใช้วิธีการสร้างกฎฮิวริสติก (Heuristic) เนื่องจาก Srisukha *et al.* (2008) ได้เปรียบเทียบให้เห็นว่าการใช้เครื่องจักรเรียนรู้มาช่วยเลือกเส้นทางหาเว็บเพจภาษาไทยนั้นให้ผลอัตราการเก็บเกี่ยวที่ดีกว่าการใช้กฎฮิวริสติก (Heuristic) ที่เสนอโดย Somboonviwat *et al.* (2006)

กระบวนการออกแบบส่วนทำนายภาษาประกอบด้วยหลายขั้นตอน ตั้งแต่ขั้นตอนการคัดเลือกและสกัดคุณลักษณะจากโฮสต์กราฟ ขั้นตอนการเตรียมชุดข้อมูลฝึกสอนและทดสอบเครื่องจักรเรียนรู้ ขั้นตอนการฝึกสอนเครื่องจักรเรียนรู้ และขั้นตอนการทดสอบประสิทธิภาพของส่วนทำนายภาษาเบื้องต้นก่อนนำไปรวมเข้ากับระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษา ซึ่งมีรายละเอียดดังต่อไปนี้

3.1.1 ขั้นตอนการคัดเลือกและสกัดคุณลักษณะจากโฮสต์กราฟ เนื่องจากได้เปลี่ยนมุมมองปัญหาจากการค้นหาเว็บเพจแบบเจาะจงภาษา มาเป็นการค้นหาในระดับเว็บไซต์ ทำให้สามารถสกัดคุณลักษณะ (feature) จากเว็บไซต์ต้นทางได้มากขึ้น เมื่อเทียบกับการสกัดคุณลักษณะจากเว็บเพจต้นทาง โดยการสกัดคุณลักษณะแต่ละครั้งจะเริ่มต้นหลังจากที่เว็บคราวเลอร์เก็บเว็บเพจเสร็จเรียบร้อยแล้ว ทำให้เราสามารถหาการเชื่อมโยงทั้งหมดของเว็บไซต์เป้าหมายที่ต้องการจากโฮสต์กราฟได้ (ภาพที่ 6) ซึ่งคุณลักษณะในระดับของเว็บไซต์ที่ใช้ในงานวิจัยนี้ จัดได้เป็น 4 กลุ่ม ดังนี้



ภาพที่ 6 โฮสต์กราฟแสดงการเชื่อมโยงระหว่างเว็บไซต์ต้นทางไปยังเว็บไซต์ปลายทาง

ก. กลุ่มคุณลักษณะสถานที่ตั้งของเว็บเซิร์ฟเวอร์

ผู้พัฒนาเว็บไซต์ส่วนใหญ่มักจะเลือกผู้ให้บริการเว็บเซิร์ฟเวอร์ที่ตั้งอยู่ภายในประเทศ เนื่องจากความสะดวกในการติดต่อผู้ให้บริการ ด้านค่าใช้จ่าย และยิ่งไปกว่านั้น

ยังคำนึงถึงความเร็วในการรับส่งข้อมูลระหว่างเว็บเซิร์ฟเวอร์กับผู้ใช้ เพราะการรับส่งข้อมูลโดยใช้เครือข่ายภายในประเทศจะได้ความเร็วที่ดีกว่าการรับส่งข้อมูลกับเครื่องแม่ข่ายที่อยู่นอกประเทศ นอกจากจะเลือกเว็บเซิร์ฟเวอร์ที่มีสถานที่ตั้งอยู่ในประเทศแล้ว จากการศึกษาของ Rauber *et al.* (2002) พบว่าเว็บเซิร์ฟเวอร์ส่วนใหญ่จะตั้งอยู่ใกล้กับศูนย์โทรคมนาคมของประเทศอีกด้วย จากการค้นหาที่ตั้งของรายชื่อเว็บไซต์ในฐานข้อมูลเว็บภาษาไทย (ตารางที่ **7**) โดยการป้อนรายชื่อเว็บไซต์หรือหมายเลขไอพีให้กับโปรแกรม InetAddressLocator (2011) พบว่า เว็บไซต์ที่ให้บริการเว็บเพจภาษาไทยส่วนใหญ่มีที่ตั้งอยู่ ณ ประเทศไทย ประมาณ 91.98% ตามมาด้วยสหรัฐอเมริกา จีน อังกฤษ และญี่ปุ่น ตามลำดับ (ตารางที่ 9) ด้วยเหตุนี้สถานที่ตั้งของเว็บเซิร์ฟเวอร์จึงเป็นคุณลักษณะที่สำคัญ เพราะถ้าเว็บไซต์ปลายทางใดๆ มีการเชื่อมโยงมาจากเว็บไซต์ต้นทางที่ตั้งในประเทศไทยจำนวนมากแล้ว ก็น่าจะมีความน่าจะเป็นสูงที่เว็บไซต์ปลายทางนั้นจะให้บริการเว็บเพจภาษาไทย

ตารางที่ 9 สถานที่ตั้งของเว็บไซต์ในฐานข้อมูลเว็บภาษาไทย

ประเทศที่ตั้งเว็บเซิร์ฟเวอร์	จำนวนเว็บไซต์ภาษาไทย	จำนวนเว็บไซต์ภาษาอื่น	รวม
ประเทศไทย	23,101	7,885	30,986
ประเทศจีน	106	964	1,070
ประเทศญี่ปุ่น	14	443	475
ประเทศอังกฤษ	23	349	372
ประเทศสหรัฐอเมริกา	479	7,438	7,917
ประเทศอื่นๆ	1,391	3,723	5,114

การสกัดคุณลักษณะในกลุ่มนี้ จะใช้วิธีหาที่ตั้งของเว็บเซิร์ฟเวอร์ปลายทางร่วมกับการนับจำนวนของเว็บเซิร์ฟเวอร์ต้นทาง แยกตามประเภทของเว็บไซต์ และสถานที่ตั้งของเว็บเซิร์ฟเวอร์ ซึ่งจะมีรูปแบบของฟีเจอร์เวคเตอร์ (feature vector) ตามภาพที่ 7 และสามารถสกัดคุณลักษณะได้ทั้งหมด 13 คุณลักษณะดังนี้

- ที่ตั้งอยู่ของเว็บไซต์ปลายทาง ซึ่งมีเซ็ตคำตอบดังนี้ {“TH”, “CH”, “JP”, “UK”, “USA”, “OTHER”}
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาไทยและตั้งอยู่ที่ประเทศไทย
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาไทยและตั้งอยู่ที่ประเทศจีน

- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาไทยและตั้งอยู่ที่ประเทศญี่ปุ่น
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาไทยและตั้งอยู่ที่ประเทศอังกฤษ
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาไทยและตั้งอยู่ที่ประเทศสหรัฐอเมริกา
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาไทยและตั้งอยู่ที่ประเทศอื่นๆ
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาอื่นและตั้งอยู่ที่ประเทศไทย
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาอื่นและตั้งอยู่ที่ประเทศจีน
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาอื่นและตั้งอยู่ที่ประเทศญี่ปุ่น
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาอื่นและตั้งอยู่ที่ประเทศอังกฤษ
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาอื่นและตั้งอยู่ที่ประเทศสหรัฐอเมริกา
- จำนวนเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาอื่นและตั้งอยู่ที่ประเทศอื่นๆ

TH, 4, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0

ภาพที่ 7 ตัวอย่างฟีเจอร์เวคเตอร์ของกลุ่มคุณลักษณะที่ตั้งของเว็บเซิร์ฟเวอร์

จากภาพที่ 7 สามารถอธิบายได้ว่า เว็บไซต์เป้าหมายตั้งอยู่ที่ประเทศไทย โดยมีการเชื่อมโยงจากเว็บไซต์ภาษาไทยจำนวน 5 การเชื่อมโยง (4 การเชื่อมโยงมาจากเครื่องแม่ข่ายที่ตั้งอยู่ในประเทศไทย และ 1 การเชื่อมโยงจากเครื่องแม่ข่ายที่ตั้งอยู่ในประเทศสหรัฐฯ) และการเชื่อมโยงจากเว็บไซต์ภาษาอื่นอีก 1 การเชื่อมโยง (จากเครื่องแม่ข่ายที่ตั้งอยู่ในประเทศไทย)

ข. กลุ่มคุณลักษณะโดเมนระดับบนสุดของเว็บไซต์

จากการตรวจเอกสารพบว่า งานวิจัยที่มีวัตถุประสงค์เพื่อสร้างคลังเว็บ (web archive) มักจะใช้วิธีการเลือกเก็บเว็บเพจตามโดเมนของประเทศที่ต้องการ เช่น การสร้างคลังเว็บประจำชาติโปรตุเกส Gomes *et al.* (2008) เลือกที่จะเก็บเว็บเพจภายใต้โดเมน .pt เท่านั้น แทนที่จะออกแบบเว็บคราวเลอร์เจาะจงภาษา เป็นต้น นอกจากนี้เมื่อวิเคราะห์โฮสต์กราฟจากฐานข้อมูลเว็บภาษาไทย (ตารางที่ 7) พบว่า เว็บไซต์ต้นทางที่มีโดเมนแตกต่างกันจะมีค่าสถิติของการเชื่อมโยงไป

ยังเว็บไซต์ปลายทางที่เป็นเว็บไซต์ภาษาไทยไม่เท่ากัน ดังตารางที่ 10 จากการสังเกต พบว่า เว็บไซต์ต้นทางและเว็บไซต์ปลายทางที่มีโดเมนเป็น .th มีความน่าจะเป็นที่เว็บไซต์ปลายทางจะเป็นเว็บไซต์ภาษาไทยมากกว่าโดเมนอื่น นอกจากนี้ยังพบว่า เว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่นและมีโดเมนเป็น .com และ .org มีความน่าจะเป็นที่เว็บไซต์ปลายทางจะเป็นเว็บไซต์ภาษาไทย ตัวอย่างเว็บไซต์ต้นทาง เช่น http://www.dmoz.org/ ซึ่งตามนิยามนั้น เว็บไซต์นี้จัดเป็นเว็บไซต์ภาษาอื่น แต่พบการเชื่อมโยงจากเว็บไซต์นี้ไปยังเว็บไซต์ภาษาไทยจำนวนมาก

ตารางที่ 10 โครงสร้างการเชื่อมโยงระดับเว็บไซต์ของฐานข้อมูลเว็บภาษาไทย โดยแยกตามโดเมนระดับบนสุด

โดเมนระดับบนสุดของเว็บไซต์ต้นทาง	ประเภทเว็บไซต์ของเว็บไซต์ต้นทาง	เว็บไซต์ปลายทาง	
		จำนวนเว็บไซต์ภาษาไทย	จำนวนเว็บไซต์ภาษาอื่น
.com	เว็บไซต์ภาษาไทย	15,297	3,116
	เว็บไซต์ภาษาอื่น	29,633	16,166
.net	เว็บไซต์ภาษาไทย	3,496	1,213
	เว็บไซต์ภาษาอื่น	3,790	2,222
.org	เว็บไซต์ภาษาไทย	3,397	808
	เว็บไซต์ภาษาอื่น	12,062	7,798
.th	เว็บไซต์ภาษาไทย	371,294	117,093
	เว็บไซต์ภาษาอื่น	31,234	10,274
.info	เว็บไซต์ภาษาไทย	51	21
	เว็บไซต์ภาษาอื่น	261	143
โดเมนอื่น	เว็บไซต์ภาษาไทย	2,816	821
	เว็บไซต์ภาษาอื่น	29,381	5,563

การสกัดคุณลักษณะในกลุ่มนี้ จะใช้วิธีหาโดเมนระดับบนสุดทั้งของเว็บไซต์ปลายทาง ร่วมกับการนับจำนวนของโดเมนระดับบนสุดของเว็บไซต์ต้นทาง แยกตามประเภทของเว็บไซต์ ซึ่งจะมีรูปแบบของฟีเจอร์เวคเตอร์ ตามภาพที่ 8 และจะสามารถสกัดได้ทั้งหมด 13 คุณลักษณะ ได้แก่

- โดเมนของเว็บไซต์ปลายทาง ซึ่งมีเซ็ตคำตอบดังนี้ {“COM”, “NET”, “ORG”, “TH”, “INFO”, “OTHER”}
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาไทย และมีโดเมนเป็น .com
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาไทย และมีโดเมนเป็น .net
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาไทย และมีโดเมนเป็น .org
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาไทย และมีโดเมนเป็น .th
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาไทย และมีโดเมนเป็น .info
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาไทย และมีโดเมนอื่นๆ
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่น และมีโดเมนเป็น .com
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่น และมีโดเมนเป็น .net
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่น และมีโดเมนเป็น .org
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่น และมีโดเมนเป็น .th
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่น และมีโดเมนเป็น .info
- จำนวนของเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่น และมีโดเมนอื่นๆ

TH, 1, 0, 0, 1, 0, 0, 10, 0, 0, 0, 0, 0

ภาพที่ 8 ตัวอย่างฟีเจอร์เวคเตอร์ของกลุ่มคุณลักษณะโดเมนระดับบนสุดของเว็บไซต์

จากภาพที่ 8 สามารถอธิบายได้ว่า เว็บไซต์เป้าหมายมีโดเมนระดับบนสุดเป็น .th และมีการเชื่อมโยงจากเว็บไซต์ภาษาไทยที่มีโดเมนระดับบนสุดเป็น .com และ .th โดเมนละ 1 การเชื่อมโยง นอกจากนี้ยังมีการเชื่อมโยงจากเว็บไซต์ภาษาอื่นที่มีโดเมนระดับบนสุดเป็น .com อีกจำนวน 10 การเชื่อมโยง

ค. กลุ่มคุณลักษณะระดับภาษาของเว็บไซต์

คุณลักษณะกลุ่มนี้มีแนวคิดมาจากคุณลักษณะ language locality ที่ว่าเว็บเพจภาษาเดียวกันมักจะมีการเชื่อมโยงถึงกัน และอยู่ใกล้ๆ กันในเว็บกราฟ ซึ่งจากตารางที่ 8 พบว่า

เว็บไซต์ภาษาไทยก็มีคุณสมบัติ language locality ด้วย ซึ่งเราคาดว่าเว็บไซต์ต้นทางที่มีระดับภาษาของเว็บไซต์เป็นภาษาไทยมากๆ โอกาสที่จะมีการเชื่อมโยงไปยังเว็บไซต์ภาษาไทยก็มากขึ้นตามไปด้วย เราจึงสกัดคุณลักษณะในกลุ่มนี้ ด้วยการคำนวณหาค่าระดับภาษาของเว็บไซต์ต้นทาง จากอัตราส่วนของเว็บเพจภาษาไทย ต่อจำนวนเว็บเพจทั้งหมดที่เก็บมาจากเว็บไซต์นั้น ซึ่งจะมีรูปแบบของฟีเจอร์เวคเตอร์ ตามภาพที่ 9 และจะสามารถสกัดได้อีก 5 คุณลักษณะ ดังต่อไปนี้

- ค่าระดับภาษาของเว็บไซต์ต้นทางที่มีค่ามากที่สุด เป็นอันดับที่ 1
- ค่าระดับภาษาของเว็บไซต์ต้นทางที่มีค่ามากที่สุด เป็นอันดับที่ 2
- ค่าระดับภาษาของเว็บไซต์ต้นทางที่มีค่ามากที่สุด เป็นอันดับที่ 3
- ค่าระดับภาษาของเว็บไซต์ต้นทางที่มีค่ามากที่สุด เป็นอันดับที่ 4
- ค่าระดับภาษาของเว็บไซต์ต้นทางที่มีค่ามากที่สุด เป็นอันดับที่ 5

99.5, 90.0, 75.8, 0.0, 0.0

ภาพที่ 9 ตัวอย่างฟีเจอร์เวคเตอร์ของกลุ่มคุณลักษณะระดับภาษาของเว็บไซต์

จากภาพที่ 9 สามารถอธิบายได้ว่า เว็บไซต์เป้าหมายมีค่าระดับภาษาไทยของเว็บไซต์ต้นทางเป็น 99.5, 99.0 และ 75.8 ตามลำดับ ซึ่งค่าระดับภาษาไทยที่เท่ากับ 0.0 ไม่สามารถสรุปได้ว่ามาจากค่าระดับภาษาไทยของเว็บไซต์ต้นทางที่มีค่าเท่ากับ 0.0 (เว็บไซต์นั้นไม่ได้ให้บริการเว็บเพจภาษาไทย) หรือเป็นเพราะเว็บไซต์เป้าหมายนี้มีจำนวนเว็บไซต์ต้นทางเท่ากับ 3 เว็บไซต์

ง. กลุ่มคุณลักษณะการเชื่อมโยงระหว่างเว็บไซต์

คุณลักษณะกลุ่มนี้เกิดจากแนวคิดของ language locality เช่นกัน เพราะถ้าเว็บไซต์ปลายทางใดๆ ที่มีเว็บไซต์ต้นทางเป็นเว็บไซต์ภาษาไทยชี้เข้ามาจำนวนมาก ก็น่าจะมีความน่าจะเป็นสูงที่เว็บไซต์ปลายทางจะเป็นเว็บไซต์ภาษาไทย และในทางกลับกัน เว็บไซต์ปลายทางที่มีลิงค์ชี้เข้ามาจากเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่น เว็บไซต์ปลายทางนั้นก็มีโอกาสที่จะเป็นเว็บไซต์ภาษาอื่นมากเช่นกัน เพราะฉะนั้นการสกัดคุณลักษณะในกลุ่มนี้ จะนับจำนวนของเว็บไซต์ต้นทาง แยกตามประเภทของเว็บไซต์ต้นทาง ดังนั้นคุณลักษณะในกลุ่มนี้จะมี

รูปแบบของฟีเจอร์เวคเตอร์ ตามภาพที่ 10 และจะสามารถสกัดคุณลักษณะได้เพียง 2 คุณลักษณะ ได้แก่

- จำนวนเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาไทย
- จำนวนเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่น

10, 1

ภาพที่ 10 ตัวอย่างฟีเจอร์เวคเตอร์ของกลุ่มคุณลักษณะการเชื่อมโยงระหว่างเว็บไซต์

จากภาพที่ 10 สามารถอธิบายได้ว่า เว็บไซต์เป้าหมายมีจำนวนเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาไทย จำนวน 10 เว็บไซต์ และเว็บไซต์ต้นทางที่เป็นเว็บไซต์ภาษาอื่นอีก 1 เว็บไซต์

3.1.2 การเตรียมชุดข้อมูลฝึกสอน และชุดข้อมูลทดสอบ

ในงานวิจัยนี้ได้เลือกใช้เทคนิคทางเครื่องจักรเรียนรู้แบบมีผู้สอน (supervised learning) มาใช้เป็นส่วนทำนายภาษาของเว็บไซต์เป้าหมาย เพราะฉะนั้นจึงจำเป็นต้องสกัดคุณลักษณะตามหัวข้อ 3.1.1 จากโฮสต์กราฟของฐานข้อมูลเว็บภาษาไทยไว้ล่วงหน้า จากนั้นหาจำนวนตัวอย่างที่เหมาะสมเพื่อมาฝึกสอนเครื่องจักรเรียนรู้ ซึ่งถ้าเว็บไซต์ใดถูกคัดเลือกให้เป็นตัวอย่างสำหรับฝึกสอนแล้ว จะไม่นำมาใช้เป็นชุดข้อมูลทดสอบอีก

ขั้นตอนการเตรียมชุดข้อมูลฝึกสอนและทดสอบอย่างง่าย เริ่มต้นจากกำหนดสัดส่วนในการสุ่มกลุ่มตัวอย่างฝึกสอน โดยเริ่มที่จำนวนตัวอย่าง 10% ของจำนวนเว็บไซต์ในฐานข้อมูลเว็บภาษาไทย และเพิ่มขึ้นทีละ 10% จนกระทั่งถึง 90% โดยสุ่มเลือกอัตราส่วนละ 3 ครั้ง ส่วนชุดข้อมูลทดสอบนั้นจะใช้ตัวอย่างในส่วนที่เหลือที่ไม่ได้ถูกสุ่มเลือกขึ้นมา จากนั้นใช้เครื่องจักรเรียนรู้แบบเบย์อย่างง่าย (Naïve Bayes) สร้างแบบแบบจำลอง (model) จากการเรียนรู้ตัวอย่างชุดข้อมูลฝึกสอน และใช้ทดสอบการทำนายภาษาของเว็บไซต์เป้าหมายที่อยู่ในชุดข้อมูลทดสอบต่อไป ซึ่งรายละเอียดการเตรียมชุดข้อมูลฝึกสอนและทดสอบ แสดงดังตารางที่ 11

จากตารางที่ 11 พบว่า อัตราส่วนที่เหมาะสมระหว่างจำนวนตัวอย่างของชุดข้อมูลฝึกสอนต่อจำนวนตัวอย่างชุดทดสอบ คือ 30 ต่อ 70 ตามลำดับ โดยที่เครื่องจักรเรียนรู้แบบเบย์อย่างง่าย (Naïve Bayes) สามารถเรียนรู้จากชุดข้อมูลฝึกสอน และสามารถทำนายภาษาของเว็บไซต์เป้าหมายที่อยู่ในชุดข้อมูลทดสอบ ได้ความแม่นยำถึง 80.90%

ตารางที่ 11 ผลการสุ่มคัดเลือกกลุ่มตัวอย่างเว็บไซต์มาสร้างเป็นชุดข้อมูลฝึกสอน และชุดข้อมูลทดสอบ

อัตราส่วน (ชุดข้อมูลฝึกสอน : ชุดข้อมูลทดสอบ)	ครั้งที่	ชุดข้อมูลฝึกสอน		ชุดข้อมูลทดสอบ		ความแม่นยำ (%)
		จำนวนเว็บไซต์ภาษาไทย	จำนวนเว็บไซต์ภาษาอื่น	จำนวนเว็บไซต์ภาษาไทย	จำนวนเว็บไซต์ภาษาอื่น	
10 : 90	1	2,549	2,042	22,565	18,760	37.86
	2	2,523	2,068	22,591	18,734	34.79
	3	2,492	2,099	22,622	18,703	40.19
20 : 80	1	5,011	4,172	20,103	16,630	45.74
	2	5,009	4,174	20,105	16,628	40.61
	3	5,006	4,177	20,108	16,625	48.34
30 : 70	1	7,389	6,385	17,725	14,417	80.90
	2	7,602	6,172	17,512	14,630	54.26
	3	7,536	6,238	17,578	14,564	80.36
40 : 60	1	10,082	8,284	15,032	12,518	80.55
	2	10,095	8,271	15,019	12,531	49.51
	3	10,077	8,289	15,037	12,513	45.69
50 : 50	1	12,567	10,391	12,547	10,411	46.96
	2	12,501	10,457	12,613	10,345	40.31
	3	12,573	10,385	12,541	10,417	26.42
60 : 40	1	15,076	12,473	10,038	8,329	48.39
	2	10,085	8,282	15,029	12,520	47.48
	3	15,118	12,431	9,996	8,371	47.91

ตารางที่ 11 (ต่อ)

อัตราส่วน (ชุดข้อมูลฝึกสอน : ชุดข้อมูลทดสอบ)	ครั้งที่	ชุดข้อมูลฝึกสอน		ชุดข้อมูลทดสอบ		ความแม่นยำ (%)
		จำนวนเว็บไซต์ภาษาไทย	จำนวนเว็บไซต์ภาษาอื่น	จำนวนเว็บไซต์ภาษาไทย	จำนวนเว็บไซต์ภาษาอื่น	
70 : 30	1	17,539	14,602	7,575	6,200	47.30
	2	17,536	14,605	7,578	6,197	80.07
	3	17,551	14,590	7,563	6,212	41.07
80 : 20	1	20,107	16,625	5,007	4,177	40.81
	2	2,009	16,642	5,024	4,160	46.52
	3	20,102	16,630	5,012	4,172	45.37
90 : 10	1	22,577	18,747	2,537	2,055	80.62
	2	22,590	18,734	2,524	2,068	71.84
	3	22,636	18,688	2,478	2,114	39.33

ตารางที่ 12 สถิติของชุดข้อมูลฝึกสอน และชุดข้อมูลทดสอบ ที่สุ่มตัวอย่างมาจากฐานข้อมูลเว็บภาษาไทย ในอัตราส่วน (30 : 70)

โดเมนระดับบนสุด	ชุดข้อมูลฝึกสอน		ชุดข้อมูลทดสอบ	
	จำนวนเว็บไซต์ภาษาไทย	จำนวนเว็บไซต์ภาษาอื่น	จำนวนเว็บไซต์ภาษาไทย	จำนวนเว็บไซต์ภาษาอื่น
.com	331	2,086	744	4,570
.net	41	231	116	536
.org	57	471	108	1,140
.th	6,826	2,503	16,403	5,724
.info	3	21	9	43
โดเมนอื่นๆ	131	1,073	343	2,404
รวม	7,389	6,385	17,725	14,417

ตารางที่ 13 รายการชุดข้อมูลฝึกสอนที่ใช้วิธีการจัดหมู่ (combination) กับกลุ่มคุณลักษณะทั้ง 4 กลุ่ม

ลำดับที่	กลุ่มคุณลักษณะสถานที่ตั้งของเว็บเซิร์ฟเวอร์	กลุ่มคุณลักษณะโดเมนระดับบนสุดของเว็บไซต์	กลุ่มคุณลักษณะระดับภาษาของเว็บไซต์	กลุ่มคุณลักษณะการเชื่อมโยงระหว่างเว็บไซต์
1	✓			
2		✓		
3			✓	
4				✓
5	✓	✓		
6	✓		✓	
7	✓			✓
8		✓	✓	
9		✓		✓
10			✓	✓
11	✓	✓	✓	
12	✓	✓		✓
13	✓		✓	✓
14		✓	✓	✓
15	✓	✓	✓	✓
16	✓ (เฉพาะโดเมน .th)	✓ (เฉพาะโดเมน .th)	✓ (เฉพาะโดเมน .th)	✓ (เฉพาะโดเมน .th)
17	✓ (นอกโดเมน .th)	✓ (นอกโดเมน .th)	✓ (นอกโดเมน .th)	✓ (นอกโดเมน .th)

ถึงแม้ว่าจะได้ชุดข้อมูลฝึกสอน และชุดข้อมูลทดสอบที่เหมาะสมแล้ว แต่เรามีความจำเป็นที่จะต้องเพิ่มความหลากหลายให้กับชุดข้อมูลทดสอบ เนื่องจากเทคนิคทางเครื่องเรียนรู้แต่ละแบบมีความสามารถในการเรียนรู้และจดจำคุณลักษณะที่แตกต่างกัน ตัวอย่างเช่น SVM อาจจะให้ความแม่นยำในการทำนายสูงสุด ถ้าใช้เพียงคุณลักษณะจากกลุ่มโดเมน และที่ตั้งของเว็บเซิร์ฟเวอร์เท่านั้น เป็นต้น ด้วยเหตุนี้เราจึงใช้วิธีการจัดหมู่ (combination) กับกลุ่มคุณลักษณะทั้ง 4 กลุ่ม ซึ่งจะได้ความหลากหลายของชุดข้อมูลฝึกสอนทั้งหมด 16 แบบ แสดงในตารางที่ 13 นอกจากนี้ยังได้เพิ่มวิธีการสร้างชุดข้อมูลฝึกสอนอีก 1 โดยมีสมมติฐานจากเว็บไซต์ภาษาไทยที่มี

โดเมนระดับบนสุดเป็น .th น่าจะมีคุณลักษณะที่แตกต่างจากเว็บไซต์ภาษาไทยที่อยู่นอกโดเมน .th เป็นแบบที่ 16 และ 17 ในตารางที่ 13 ตามลำดับ

3.1.3 ขั้นตอนการฝึกสอนเครื่องจักรเรียนรู้ และทดสอบประสิทธิภาพของส่วนทำนายภาษาเบื้องต้น

ในส่วนนี้จะอธิบายการออกแบบส่วนทำนายภาษาของเว็บไซต์เป้าหมาย เพื่อช่วยลดภาระงานแก่เว็บคราวเลอร์แบบแยกเก็บตามไซต์ โดยมีการทำนายหาเว็บไซต์ที่มีความน่าจะเป็นที่จะให้บริการเว็บเพจภาษาไทยก่อนที่จะให้เว็บคราวเลอร์ทำงาน จากงานวิจัยของ Srisukha *et al.* (2008) ได้ใช้เทคนิคเครื่องจักรเรียนรู้แบบเบย์อย่างง่าย (Naïve Bayes) ในการสร้างเว็บคราวเลอร์เจาะจงภาษาไทยในระดับของเว็บเพจ ดังนั้นในการทดลองเบื้องต้นเราจึงเลือกใช้เครื่องจักรเรียนรู้แบบเบย์อย่างง่าย (Naïve Bayes) เพื่อเปรียบเทียบประสิทธิภาพอัตราการเก็บเกี่ยว (harvest rate) เว็บเพจภาษาไทย ระหว่างเว็บคราวเลอร์เจาะจงภาษาในระดับเว็บเพจ กับเว็บคราวเลอร์เจาะจงภาษาในระดับเว็บไซต์ นอกจากนี้ยังทดลองใช้เทคนิคเครื่องจักรเรียนรู้แบบหลายตัว (classifier ensemble) มาเพิ่มความแม่นยำให้กับส่วนทำนายภาษาอีกด้วย

เทคนิคเครื่องจักรเรียนรู้มีหลายแบบ แต่ที่เลือกมาใช้ในงานวิจัยนี้ มีทั้งหมด 7 เทคนิค ได้แก่ Naïve Bayes, bayes network, radial basic fuction (RBS) network, support vector machine (SVM), k-nearest neighbor (k-NN), decision tree และ random forest จากนั้นให้เทคนิคเครื่องจักรเรียนรู้เหล่านี้ เรียนรู้จากชุดข้อมูลฝึกสอนทั้ง 17 แบบ (ตารางที่ 13) โดยมีการแบ่งชุดข้อมูลฝึกสอนแต่ละชุดออกเป็น 10 ส่วนเท่าๆ กัน และทำการตรวจสอบไขว้ 10 พับ (10-fold cross validation) ด้วยโปรแกรมเวก้า (Weka) (Hall *et al.*, 2009) ซึ่งผลการทดสอบประสิทธิภาพเบื้องต้นแสดงดังตารางที่ 14 พบว่า เทคนิคเครื่องจักรเรียนรู้ที่ได้ค่าความแม่นยำสูงสุดถึง 81.60% คือ decision tree ทั้งจากข้อมูลฝึกสอนชุดที่ 11 และชุดที่ 15 แต่สำหรับข้อมูลฝึกสอนชุดที่ 17 ที่ได้ค่าความแม่นยำถึง 93.27% นั้นเพราะว่ามีจำนวนตัวอย่างฝึกสอนน้อยกว่าข้อมูลฝึกสอนชุดอื่นๆ และเวลานำไปใช้งานจริงจำเป็นต้องใช้งานร่วมกับเครื่องจักรเรียนรู้ที่ได้ฝึกสอนจากข้อมูลฝึกสอนชุดที่ 16 อีกด้วย ซึ่งเมื่อรวมเครื่องจักรเรียนรู้ทั้งสองชุดข้อมูลนี้ ทำให้ค่าความแม่นยำเพิ่มสูงขึ้นเป็น 82.12% เมื่อใช้เครื่องจักรเรียนรู้แบบ k-NN สำหรับทำนายเว็บไซต์เป้าหมายที่มีโดเมนเป็น .th (ข้อมูลฝึกสอนชุดที่ 16) และใช้เครื่องจักรเรียนรู้แบบ decision tree สำหรับทำนายเว็บไซต์เป้าหมายที่มีโดเมนอื่นที่ไม่ใช่ .th (ข้อมูลฝึกสอนชุดที่ 17) ด้วยเหตุนี้จึงมีความเป็นไปได้ว่า ถ้าเราใช้

เครื่องจักรเรียนรู้แบบหลายตัว (classifier ensemble) น่าจะช่วยเพิ่มความแม่นยำให้กับส่วนทำนายภาษาได้เพิ่มขึ้น

ตารางที่ 14 ผลการทดสอบประสิทธิภาพเบื้องต้นของเครื่องจักรเรียนรู้แบบต่างๆ ที่มีความแม่นยำสูงที่สุดในแต่ละชุดข้อมูลฝึกสอน

ข้อมูลฝึกสอนชุดที่	เทคนิคเครื่องจักรเรียนรู้	จำนวนตัวอย่างที่ทำนายถูกต้อง	ความแม่นยำ (Accuracy) (%)
1	Decision Tree	10,779	78.26
1	Random Forest	10,779	78.26
2	Random Forest	10,753	78.07
3	SVM	10,556	76.64
4	SVM	7,460	54.16
5	Random Forest	10,958	79.56
6	Decision Tree	11,143	80.90
7	Random Forest	10,781	78.27
8	Decision Tree	11,009	79.93
9	Random Forest	10,754	78.07
10	SVM	10,549	76.59
11	Decision Tree	11,239	81.60
12	Decision Tree	10,955	79.53
12	Random Forest	10,954	79.53
13	Decision Tree	11,143	80.90
14	Decision Tree	11,010	79.93
15	Decision Tree	11,239	81.60
16	10-NN	7,165	76.80
17	Decision Tree	4,146	93.27

3.1.3 ขั้นตอนการสร้างระบบเครื่องจักรเรียนรู้แบบหลายตัว (classifier ensemble)

จากผลการทดลองในตารางที่ 14 แสดงให้เห็นว่า เครื่องจักรเรียนรู้เพียงตัวเดียวไม่สามารถทำนายประเภทของเว็บไซต์เป้าหมายได้ถูกต้องทุกกรณี และนอกจากนี้ยังพบอีกว่าการใช้เครื่องจักรเรียนรู้มากกว่าหนึ่งตัว มีโอกาสเพิ่มความแม่นยำให้กับส่วนทำนายอีกด้วย เพราะฉะนั้นเราจึงสนใจที่จะเลือกใช้เทคนิคเครื่องจักรเรียนรู้แบบหลายตัว (classifier ensemble) เพื่อมาเพิ่มประสิทธิภาพให้กับส่วนทำนายภาษาของเว็บไซต์เป้าหมาย โดยมีขั้นตอนตามแนวทางที่เสนอโดย Goeble *et al.* (2004) แสดงดังภาพที่ 3 ซึ่งมีทั้งหมด 5 ขั้นตอน ดังนี้

ขั้นตอนที่หนึ่งการสร้างกลุ่มของเครื่องจักรเรียนรู้ และขั้นตอนที่สองที่ให้ทดสอบเครื่องจักรเรียนรู้ ด้วยวิธีการทดสอบแบบไขว้กลับ 10 พับนั้นได้ดำเนินการไว้แล้วในหัวข้อ 3.1.2 ขั้นตอนที่สามจำเป็นที่จะต้องคัดเลือกหาเครื่องจักรเรียนรู้ตั้งต้น ซึ่งเราใช้วิธีการคัดเลือกจากเครื่องจักรเรียนรู้ที่ให้ค่าความแม่นยำสูงที่สุด คือ เครื่องจักรเรียนรู้ที่มีความแม่นยำเท่ากับ 81.60% ซึ่งมีด้วยกัน 2 ตัว ดังนั้นจึงใช้มาตรวัดความสัมพันธ์แบบสถิติคิว (Q statistic correlation measure) และพบว่าได้ค่าเท่ากับ 1 ซึ่งสามารถตีความได้ว่าเครื่องจักรเรียนรู้ทั้งสองตัวนี้ซ้ำซ้อนกัน และให้คำตอบเหมือนกันในทุกๆ กรณี เราจึงเลือกเครื่องจักรเรียนรู้แบบ decision tree ที่ใช้ชุดข้อมูลฝึกสอนชุดที่ 15 เนื่องจากชุดนี้ได้ใช้กลุ่มคุณลักษณะครบทั้ง 4 กลุ่ม จากนั้นทำการวัดค่าความสอดคล้อง (correlation) ระหว่างเครื่องจักรเรียนรู้ตั้งต้นกับเครื่องจักรเรียนรู้ที่เหลือ ด้วยมาตรวัดทั้ง 4 แบบ ได้แก่ Q Statistic, ρ-correlation, Disagreement Measure และ Double-Fault Measure เพื่อคัดเลือกหาสมาชิกใหม่ของระบบจากเครื่องจักรเรียนรู้ที่เหลือ โดยจะเลือกจากเครื่องจักรเรียนรู้ที่มีค่าความสอดคล้องน้อยที่สุดเมื่อเทียบกับเครื่องจักรเรียนรู้ตั้งต้นเข้าไปในระบบทำนายทีละตัว ซึ่งตัวอย่างผลการคำนวณค่ามาตรวัดความสอดคล้องดังตารางที่ 15

ตารางที่ 15 ตัวอย่างผลการคำนวณค่ามาตรวัดความสอดคล้องทั้ง 4 แบบ

ข้อมูลฝึกสอนชุดที่	เทคนิคเครื่องจักรเรียนรู้	มาตรวัดความสอดคล้อง			
		Q	ρ	D	DF
1	Decision Tree	0.98	0.84	2,166	5,547
2	Random Forest	0.95	0.74	3,427	4,965
3	SVM	0.89	0.64	5,000	4,383
4	SVM	0.64	0.45	11,287	4,671

ตารางที่ 15 (ต่อ)

ข้อมูลฝึกสอนชุดที่	เทคนิคเครื่องจักรเรียนรู้	มาตรวัดความสอดคล้อง			
		Q	ρ	D	DF
5	Random Forest	0.99	0.86	1,894	5,785
6	Decision Tree	1.00	0.95	617	6,036
7	Random Forest	0.98	0.84	2,160	5,545
8	Decision Tree	0.98	0.84	2,025	5,390
9	Random Forest	0.95	0.74	3,421	4,965
10	SVM	0.88	0.63	5,151	4,327
12	Decision Tree	0.99	0.88	1,639	5,887
13	Decision Tree	1.00	0.95	617	6,036
14	Decision Tree	0.98	0.84	2,025	5,390
16	k-NN	0.94	0.71	3,674	4,541
17	Decision Tree				

ในขั้นตอนการจัดกลุ่มเครื่องจักรเรียนรู้ จากการทดลองพบว่ามาตรวัดความสอดคล้องที่เหมาะสมกับการจัดกลุ่มในการทดลองครั้งนี้คือ มาตรวัดแบบ Double-Fault ให้ค่าความแม่นยำสูงที่สุด โดยประกอบไปด้วยเครื่องจักรเรียนรู้ทั้งหมด 10 ตัว ได้แก่ decision tree บนชุดข้อมูลฝึกสอนที่ 1, 15 และ 17, SVM บนชุดข้อมูลฝึกสอนที่ 3, 4 และ 10, k-NN บนชุดข้อมูลฝึกสอนที่ 16 และ random forest บนชุดข้อมูลฝึกสอนที่ 2 และ 7 โดยใช้วิธีการรวมคำตอบแบบ majority vote ซึ่งได้ความแม่นยำเท่ากับ 81.49% และมีค่าความแม่นยำมากกว่าเครื่องจักรเรียนรู้แบบหนึ่งตัว ซึ่งได้เท่ากับ 80.17% ดังตารางที่ 16

ตารางที่ 16 ตัวอย่างการจัดกลุ่มระบบเครื่องจักรเรียนรู้แบบหลายตัว และผลการทดสอบบนชุดข้อมูลทดสอบหลังจากใช้วิธีการรวมคำตอบแบบต่างๆ ด้วยมาตรวัดความแม่นยำ

กลุ่มเครื่องจักรเรียนรู้	วิธีการรวมคำตอบ (fusion method)					
	MAX	MIN	PRODUCT	Basian Combination	Average	Majority vote
DT (15)	80.17					
DT (15), SVM (10)	80.09	80.09	80.09	80.09	80.09	78.10
DT (15), SVM (10), SVM (3)	80.01	80.01	79.54	78.68	78.68	77.35
DT (15), SVM (10), SVM (3), K-NN (16) + DT (17)	79.01	79.01	80.54	81.00	81.00	78.55
DT (15), SVM (10), SVM (3), K-NN (16) + DT (17), SVM (4)	79.00	79.00	80.69	81.09	81.09	78.55
DT (15), SVM (10), SVM (3), K-NN (16) + DT (17), SVM (4), RF (2)	79.09	79.09	79.76	80.25	80.25	78.30
DT (15), SVM (10), SVM (3), K-NN (16) + DT (17), SVM (4), RF (2), DT (14)	79.09	79.09	79.86	80.14	80.14	81.46
DT (15), SVM (10), SVM (3), K-NN (16) + DT (17), SVM (4), RF (2), DT (14), RF (7)	79.12	79.12	79.75	80.09	80.09	81.40
DT (15), SVM (10), SVM (3), K-NN (16) + DT (17), SVM (4), RF (2), DT (14), RF (7), DT (1)	79.11	79.11	78.98	80.98	80.98	81.49
DT (15), SVM (10), SVM (3), K-NN (16) + DT (17), SVM (4), RF (2), DT (14), RF (7), DT (1), RF (5)	79.12	79.12	78.95	80.57	80.57	79.93
DT (15), SVM (10), SVM (3), K-NN (16) + DT (17), SVM (4), RF (2), DT (14), RF (7), DT (1), DT (12)	79.12	79.12	78.96	79.97	79.97	79.90

3.2 ส่วนของเว็บคราวเลอร์แบบแยกเก็บตามไซต์ (site crawler)

เว็บคราวเลอร์แบบแยกเก็บตามไซต์ (site crawler) มีหน้าที่เก็บเว็บเพจภายใต้เว็บไซต์ที่กำหนดไว้ในยูอาร์แอลคิว ซึ่งรายชื่อเว็บไซต์เริ่มต้นจะได้รับจากส่วนทำนายภาษา (language predictor) และมีการจัดเรียงตามความน่าจะเป็นที่เว็บไซต์ปลายทางนั้นจะมีโอกาสให้บริการเว็บเพจภาษาไทยจากมากไปหาน้อย แต่ด้วยจำนวนเว็บไซต์ที่มีอยู่มากมายบนอินเทอร์เน็ต จึงทำให้

ไม่สามารถใช้วิธีการออกแบบเว็บคราวเลอร์แบบดั้งเดิมได้ ดังนั้นในงานนี้จึงออกแบบให้เป็นเว็บคราวเลอร์แบบขนานในโหมดไฟร์วอลล์ (firewall mode) (Cho *et al.*, 2002) เพื่อให้สามารถเก็บเว็บเพจได้จำนวนมากที่สุดเท่าที่จะเป็นไปได้ ภายใต้ระยะเวลาที่กำหนด และป้องกันปัญหาการเก็บเว็บเพจซ้ำซ้อนได้อีกด้วย เนื่องจากในโหมดนี้จะให้เว็บคราวเลอร์ 1 ตัวรับผิดชอบ 1 เว็บไซต์เท่านั้น จึงเหมาะสมกับกับงานของเว็บคราวเลอร์แบบแยกเก็บตามไซต์เป็นอย่างยิ่ง นอกจากนี้เว็บคราวเลอร์แบบแยกเก็บตามไซต์จะเริ่มต้นเก็บเว็บเพจหลัก (entry page) เช่น index.html, index.php เป็นต้นของแต่ละเว็บไซต์ก่อนเป็นลำดับแรก ซึ่งแตกต่างจากเว็บคราวเลอร์แบบดั้งเดิมที่มักจะตามลิงค์จากเว็บเพจหนึ่งไปสู่อีกเว็บเพจหนึ่ง ทำให้พลาดเว็บเพจหลักที่น่าจะมีข้อมูลสำคัญของแต่ละเว็บไซต์

นอกจากกจะออกแบบให้เว็บคราวเลอร์ทำงานเก็บเว็บเพจได้อย่างรวดเร็วแล้ว ยังต้องคำนึงอัตราการเก็บเกี่ยวเว็บเพจภาษาไทยของแต่ละเว็บไซต์ด้วย แต่เนื่องจากระบบต้นแบบมีส่วนทำนายภาษาเพื่อช่วยกรองหาเว็บไซต์เป้าหมายที่คาดว่าจะให้บริการเว็บเพจภาษาไทยแล้ว ดังนั้นในส่วนของไซต์คราวเลอร์ จึงเลือกใช้วิธีอย่าง่ายในการคัดเลือก และค้นหาเส้นทางของเว็บเพจภาษาไทย 2 วิธี การปรับปรุงจากอัลกอริทึมการค้นหาแบบกว้างก่อน และการปรับปรุงอัลกอริทึมที่นำเสนอโดย Tamura *et al.* (2007)

3.2.1 วิธีการเก็บเว็บเพจโดยปรับปรุงจากอัลกอริทึมการค้นหาแบบกว้างก่อน วิธีนี้จะทำการสกัดลิงค์และการเพิ่มยูอาร์แอลใหม่ลงไปในคิวตามวิธีการค้นหาแบบกว้าง ซึ่งคล้ายกับเว็บคราวเลอร์แบบดั้งเดิม แต่จะมีการเพิ่มเงื่อนไขเพื่อให้เว็บคราวเลอร์หยุดการเก็บเว็บเพจจากเว็บไซต์นั้น เมื่อมีการเก็บเว็บเพจภาษาอื่นติดต่อกันเกินค่าที่กำหนดไว้ ซึ่งสามารถสรุปเป็นขั้นตอนการทำงานของเว็บคราวเลอร์อย่างคร่าวๆ (ภาพที่ 11) เมื่อเว็บคราวเลอร์เก็บเว็บเพจมาแล้ว จะทำการสกัดลิงค์จากหน้าเว็บเพจนั้น (บรรทัดที่ 3) ถ้าพบว่ามีลิงค์ที่เชื่อมโยงไปเว็บไซต์อื่น จะเพิ่มชื่อเว็บไซต์นั้นลงไปใน newServerList (บรรทัดที่ 4-5) แต่ถ้าเป็นยูอาร์แอลของเว็บเพจภายในเว็บไซต์นั้นจะเพิ่มลงไปใน nonVisitedQ แทน ซึ่งเว็บคราวเลอร์จะหยุดการทำงานก็ต่อเมื่อไม่สามารถดึงยูอาร์แอลจาก nonVisitedQ (ในบรรทัดที่ 20) ได้แล้ว หรือในอีกกรณีคือเก็บเว็บเพจภาษาอื่นติดต่อกันเกินกว่าค่า τ ที่กำหนดไว้ (บรรทัดที่ 15 - 16)

3.2.2 วิธีการเก็บเว็บเพจโดยปรับปรุงจากอัลกอริทึมที่นำเสนอโดย Tamura *et al.* (2007) วิธีนี้เป็นการกำหนดกฎฮิวริสติกส์เพื่อเก็บเว็บเพจภาษาไทย และใช้คิวแบบมีลำดับความสำคัญ (priority queue) ด้วย วิธีการทำงานโดยสรุป แสดงดังภาพที่ 11

```
1:  Dequeue the highest priority server from the seed site's queue
2:  Download the web page
3:  Extract links
4:  Unless links from the same server
5:      Enqueue new servers found to newServerList
6:  Else
7:      Enqueue new URLs to nonVisitedQ
8:  Parse HTML to extract charset attribute in META tag
9:  If charset found (e.g., tis-620, windows-874)
10:     Convert text with that charset and send to LexTo
11: Else
12:     Convert text with UTF8 and send to LexTo
13: If LexTo returns nonThai
14:     nonThai_count++
15:     If nonThai_count > defined_Threshold τ
16:         StopThisProcess_and_EXIT
17: Else
18:     nonThai_count = 0
19: If nonVisitedQ not empty
20:     Dequeue a URL from nonVisitedQ
21:     Goto 2
```

ภาพที่ 11 วิธีการเก็บเว็บเพจภาษาไทยที่ปรับปรุงมาจากวิธีการค้นหาแบบกว้างก่อน

```
1:  URL = Dequeue the highest priority server from the site's queue
2:  Download the web page
3:  links = Extract links
4:  If links from the different server
5:      Enqueue new servers found to new_server_queue
6:  Parse HTML to extract charset attribute in META tag
7:  If charset found (e.g., tis-620, windows-874)
8:      Convert text with that charset and send to LexTo
9:  Else
10:     Convert text with that UTF-8 and send to LexTo
11: If LexTo returns Thai
12:     Number_of_NonThai_Page = 0
13:     Distance_from_Thai_page = 0
14:     URL_score = 1.0
15: Else
16:     Number_of_NonThai_Page++
17:     Distance_from_Thai_page = Distance_from_Thai_page(URL) + 1
18:     URL_score = 1.0 / Distance_from_Thai_page
19: End If
20: Enqueue new URLs to non_visit_queue with Distance_from_Thai_page
    and URL_score
21: URL = Dequeue the highest URL_score from the non_visit_queue
22: If (Distance_from_Thai_page(URL) > Threshold d) OR
     (Number_of_NonThai_Page  > Threshold τ)
23:     Clear the non_visit_queue
24:     Number_of_NonThai_Page = 0
25:     Goto 1
26: Else
27:     Goto 2
```

ภาพที่ 12 วิธีการเก็บเว็บเพจภาษาไทยที่ปรับปรุงมาจากอัลกอริทึมที่นำเสนอโดย Tamura *et al.* (2007)

จากภาพที่ 12 วิธีนี้จะประกอบด้วยสามคิว ได้แก่ site_queue, non_visit_queue และ new_server_queue ซึ่ง site_queue (บรรทัดที่ 1) เก็บรายชื่อเว็บไซต์ที่มีความน่าจะเป็นจะให้บริการเว็บเพจภาษาไทยโดยได้รับมาจากส่วนทำนายภาษา ส่วน non_visit_queue (บรรทัดที่ 20) เป็นยูอาร์แอลคิวของแต่ละเว็บไซต์ที่กำลังดาวน์โหลดอยู่ ซึ่งมีลักษณะเป็นคิวที่สามารถจัดลำดับได้โดยเรียงค่า URL_socre จากมากไปน้อย และคิวตัวสุดท้ายคือ new_server_queue (บรรทัดที่ 5) คิวตัวนี้จะเก็บรายชื่อเว็บไซต์ที่พบใหม่ เพื่อส่งต่อให้ส่วนทำนายภาษาหลังจากเสร็จสิ้นการเก็บเว็บเพจในแต่ละรอบ ซึ่งหนึ่งรอบการทำงานคือการที่ไซต์คราวเลอร์ดาวน์โหลดเว็บเพจจาก site_queue จนครบทั้งหมดแล้ว และในแต่ละเว็บไซต์จะหยุดการดาวน์โหลดเว็บเพจก็ต่อเมื่อ 1) non_visit_queue ไม่มียูอาร์แอลแล้ว 2) ยูอาร์แอลที่กำลังจะดาวน์โหลดมีระยะห่างจากเว็บเพจภาษาไทย เกินกว่าค่า threshold d (บรรทัดที่ 22) ที่กำหนดไว้ ซึ่งระยะห่างจากเว็บเพจภาษาไทย จะนับจากจำนวนเว็บเพจภาษาอื่นที่ดาวน์โหลดติดต่อกัน 3) เมื่อดาวน์โหลดเว็บเพจภาษาอื่นติดต่อกันเกินค่า threshold τ (บรรทัดที่ 22) ที่กำหนดไว้ ซึ่งมีความแตกต่างจากค่า Distance_from_Thai_page ตรงที่ Number_of_NonThai_Page จะนับขณะที่คราวเลอร์กำลังดาวน์โหลดซึ่งจะขึ้นอยู่กับลำดับยูอาร์แอลใน non_visit_queue ส่วนค่า Distance_from_Thai_page ได้รับมาจากเว็บเพจต้นทางที่ชี้มายังเว็บเพจนั้น

ผลและวิจารณ์

ในส่วนนี้นำเสนอผลการทดลองของระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาไทยแบบแยกเก็บตามไซต์ ซึ่งจะแบ่งการทดลองออกเป็น 2 ส่วน คือ การทดสอบระบบต้นแบบบนชุดข้อมูลทดสอบ โดยการเก็บเว็บเพจภายในฐานข้อมูลเว็บภาษาไทย และอีกการทดลองหนึ่งคือการทดสอบระบบต้นแบบโดยให้เก็บเว็บเพจจริงจากอินเทอร์เน็ต

ผล

1. การทดสอบบนฐานข้อมูลเว็บภาษาไทย

เพื่อหลีกเลี่ยงการรบกวนเว็บเซิร์ฟเวอร์ปลายทางบนอินเทอร์เน็ตมากเกินความจำเป็น จึงได้ออกแบบวิธีการทดสอบระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาไทยแบบแยกเก็บตามไซต์เบื้องต้นบนฐานข้อมูลเว็บภาษาไทยก่อน ซึ่งสามารถทดสอบได้หลายครั้งจนกว่าจะพึงพอใจกับประสิทธิภาพการทำงาน โดยผ่านโปรแกรมเว็บพร็อกซี่ที่พัฒนาขึ้นเอง (รายละเอียดในภาคผนวก ค.) ซึ่งในการทดลองนี้จะทดสอบประสิทธิภาพของเว็บคราวเลอร์ด้วยมาตรวัดทั้งอัตราการเก็บเกี่ยว (harvest rate) และความครอบคลุม (coverage) บนชุดข้อมูทดสอบที่ได้แบ่งไว้แล้วตามตารางที่ 12 โดยรายละเอียดที่นำมาเปรียบเทียบมีดังนี้

1.1 เว็บคราวเลอร์แบบสมบูรณ์ (perfect crawler) เป็นเว็บคราวเลอร์ที่ทราบเส้นทางที่ดีที่สุดในการเก็บเว็บเพจภาษาไทย โดยอาจจะเก็บเว็บเพจภาษาอื่นมาบ้าง เพราะมีบางเว็บเพจภาษาไทยที่สามารถเข้าถึงได้ก็ต่อเมื่อต้องแกะลิงค์จากเว็บเพจภาษาอื่นเท่านั้น นอกจากเว็บคราวเลอร์นี้จะเก็บเว็บเพจภาษาไทยจากชุดข้อมูลทดสอบได้มากที่สุดแล้ว ยังสามารถใช้เปรียบเทียบประสิทธิภาพกับเว็บคราวเลอร์แบบอื่นๆ ได้อีกด้วย เพราะเว็บคราวเลอร์ที่มีประสิทธิภาพที่ดี จะมีเส้นกราฟทั้งอัตราการเก็บเกี่ยว และความครอบคลุมเข้าใกล้เส้นกราฟของเว็บคราวเลอร์แบบสมบูรณ์

1.2 เว็บคราวเลอร์ที่ใช้การค้นหาแบบกว้างก่อน (BFS crawler) เป็นเว็บคราวเลอร์ที่ใช้วิธีการเก็บเว็บเพจแบบวิธีการค้นหาแบบกว้างก่อน (breadth-first search) เพราะฉะนั้นเว็บคราวเลอร์นี้จะทำให้ทราบถึงสัดส่วนของจำนวนเว็บเพจภาษาไทยต่อจำนวนเว็บเพจภาษาอื่นในชุดข้อมูลทดสอบ

1.3 เว็บคราวเลอร์เจาะจงภาษาในระดับของเว็บเพจ (LSWC) เลือกใช้เว็บคราวเลอร์เจาะจงภาษาในระดับเว็บเพจ ตามแนวทางของ Somboonviwat *et al.* (2006) ที่เสนอไว้ แต่ได้ปรับเปลี่ยนส่วนการตรวจสอบภาษาจากเดิมที่ใช้โปรแกรม TextCat (2011) มาเป็นโปรแกรมเล็กซ์โต (LexTo, 2011) เพื่อให้การจำแนกประเภทของเว็บเพจตรงกันตามนิยามของเว็บเพจภาษาไทย ส่วนค่าตัวแปร T ที่ใช้เป็นเงื่อนไขให้เว็บคราวเลอร์หยุดดาวน์โหลดเว็บเพจจากเว็บไซต์เมื่อเก็บเว็บเพจภาษาอื่นจากเว็บไซต์นั้นเกินกว่าค่า T ที่ได้กำหนดไว้ และในการทดลองนี้ทำให้ T เท่ากับ 20 ซึ่งเป็นค่าที่เหมาะสมจากผลการงานวิจัยของ Somboonviwat *et al.* (2006)

1.4 เว็บคราวเลอร์เจาะจงภาษาแบบแยกเก็บตามไซต์ โดยใช้วิธีการเก็บเว็บเพจที่ได้จากการปรับปรุงจากอัลกอริทึมการค้นหาแบบกว้างก่อน (LS-1) โดยที่กำหนดค่า τ=20

1.5 เว็บคราวเลอร์เจาะจงภาษาแบบแยกเก็บตามไซต์ โดยใช้วิธีการเก็บเว็บเพจที่ปรับปรุงอัลกอริทึมที่นำเสนอโดย Tamura *et al.* (2007) (LS-2) โดยที่กำหนดค่า d = 3 และ τ=5

1.6 เว็บคราวเลอร์เจาะจงภาษาแบบแยกเก็บตามไซต์ โดยส่วนทำนายภาษาใช้เทคนิคเครื่องจักรเรียนรู้แบบหนึ่งตัว (ML1-LSWC) และส่วนของเว็บคราวเลอร์แบบแยกเก็บตามไซต์จะใช้วิธีการเก็บเว็บเพจที่ปรับปรุงมาจากอัลกอริทึมการค้นหาแบบกว้างก่อน

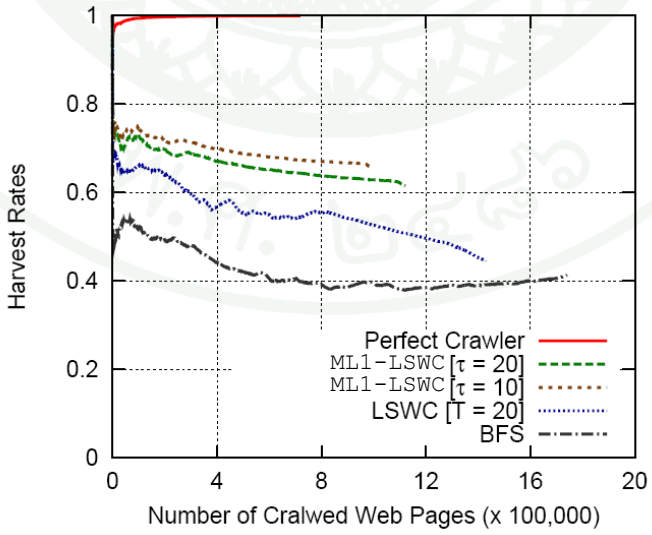
1.7 เว็บคราวเลอร์เจาะจงภาษาแบบแยกเก็บตามไซต์ โดยส่วนทำนายภาษาใช้เทคนิคเครื่องจักรเรียนรู้แบบหลายตัว (CE-LSWC) และส่วนของเว็บคราวเลอร์แบบแยกเก็บตามไซต์จะใช้วิธีการเก็บเว็บเพจที่ปรับปรุงมาจากอัลกอริทึมที่นำเสนอโดย Tamura *et al*. (2007)

การทดลองในส่วนนี้จะมีอยู่ 2 การทดลองย่อย ได้แก่ การทดลองที่ 1 เป็นการตรวจสอบสมมติฐานของปัญหาที่ได้เปลี่ยนมุมมองจากการการออกแบบเว็บคราวเลอร์ระดับเว็บพจ มาเป็นการออกแบบเว็บคราวเลอร์ในระดับของเว็บไซต์แทน ส่วนการทดลองที่ 2 เพื่อเปรียบเทียบประสิทธิภาพวิธีการเก็บเว็บเพจของเว็บคราวเลอร์แบบแยกเก็บตามไซต์ทั้ง 2 วิธี

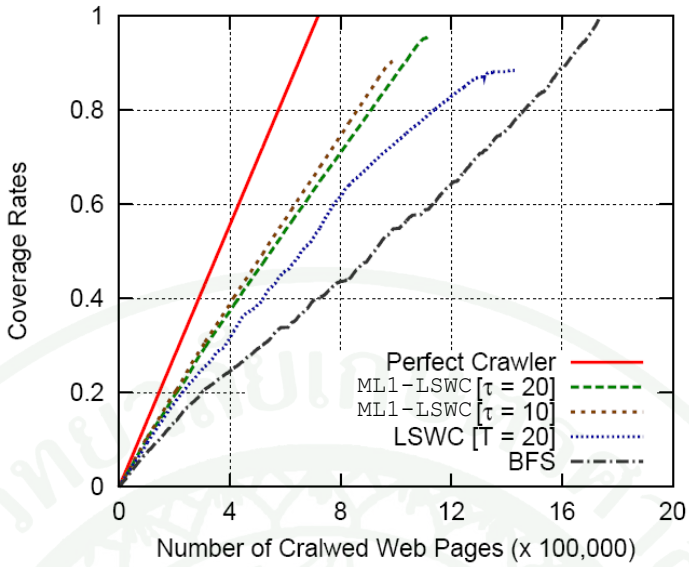
การทดลองที่ 1 มีจุดประสงค์เพื่อยืนยันสมมติฐานที่ว่า นอกจากเว็บภาษาไทยจะมีคุณสมบัติ language locality แล้ว เว็บไซต์ภาษาไทยก็มีคุณสมบัตินี้ด้วย ซึ่งถ้าสมมติฐานที่ตั้งไว้เป็นจริง จะทำให้การเปลี่ยนมุมมองของปัญหาในการออกแบบเว็บคราวเลอร์เจาะจงภาษานั้นจะเปลี่ยนจากมุมมองในระดับของเว็บเพจ มาเป็นมุมมองในระดับของเว็บไซต์ได้ อีกทั้งยังช่วยให้

การค้นหาเว็บไซต์ที่ให้บริการภาษาไทยได้ง่ายขึ้น ส่งผลให้ได้จำนวนเว็บเพจภาษาที่มากขึ้นตามไปด้วย สำหรับวิธีการทดลองจะเริ่มต้นจากกำหนดกลุ่มของเว็บไซต์เริ่มต้น ได้แก่ http://www.truehits.net/ และให้เว็บคราวเลอร์ทั้ง 5 ตัว คือ เว็บคราวเลอร์แบบสมบูรณ์ (หัวข้อ 1.1) เว็บคราวเลอร์พื้นฐาน (หัวข้อ 1.2) เว็บคราวเลอร์ในหัวข้อ 1.6 ที่กำหนดค่าตัวแปร τ = 10 และ 20 ตามลำดับ และเว็บคราวเลอร์ในหัวข้อ 1.3 ซึ่งเว็บคราวเลอร์ทั้ง 5 ตัว จะสกัดและตามเก็บเว็บเพจที่อยู่ภายใต้เว็บไซต์ในชุดข้อมูลทดสอบเท่านั้น ซึ่งจากการทดสอบด้วยเว็บคราวพื้นฐานพบว่า จำนวนเว็บเพจภาษาไทยในชุดข้อมูลทดสอบที่สามารถเก็บได้เท่ากับ 1,021,456 เว็บเพจ ซึ่งมาจากเว็บไซต์ภาษาไทย 11,219 เว็บไซต์ และเป็นเว็บเพจภาษาอื่นจำนวน 718,907 เว็บเพจ ซึ่งมาจากเว็บไซต์ภาษาอื่น 13,281 เว็บไซต์ โดยผลการทดสอบประสิทธิภาพของเว็บคราวเลอร์ทั้ง 5 ตัว ด้วยอัตราการเก็บเกี่ยว และความครอบคลุม แสดงดังภาพที่ 13 และภาพที่ 14 ตามลำดับ

จากผลการทดลองที่ 1 พบว่า ส่วนทำนายภาษาของ ML1-LSWC มีส่วนช่วยให้อัตราการเก็บเกี่ยวดีกว่า LSWC เนื่องจากงานวิจัยของ Somboonviwat *et al.* (2006) ได้ทดสอบแล้วว่ากฎฮิวริสติกส์ที่นำเสนอให้อัตราการเก็บเกี่ยวที่ดีกว่าเว็บคราวเลอร์พื้นฐาน (BFS) และยังได้ความครบถ้วนของเว็บเพจมากกว่าอีกด้วย ในกรณีที่กำหนดค่า τ = 20 ดังนั้นเราสามารถเลือกใช้ ML1-LSWC [τ=20] มาเป็นเว็บคราวเลอร์เบื้องต้นเพื่อใช้ในการเปรียบเทียบกับเว็บคราวเลอร์เจาะจงภาษา แบบใช้เครื่องจักรเรียนรู้หลายตัว เพื่อทดสอบเก็บเว็บเพจบนอินเทอร์เน็ตในการทดลองต่อไป

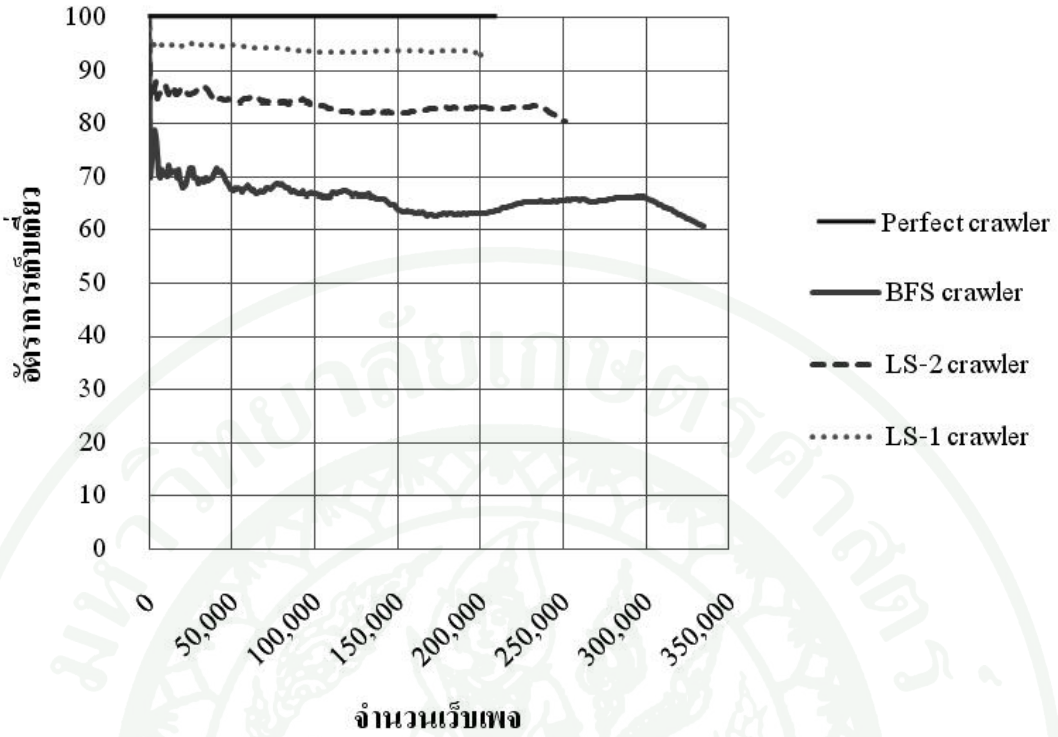


ภาพที่ 13 อัตราการเก็บเกี่ยวเว็บเพจของเว็บคราวเลอร์ทั้ง 5 ตัว จากการเก็บเว็บเพจจากชุดข้อมูลทดสอบในฐานข้อมูลเว็บภาษาไทย

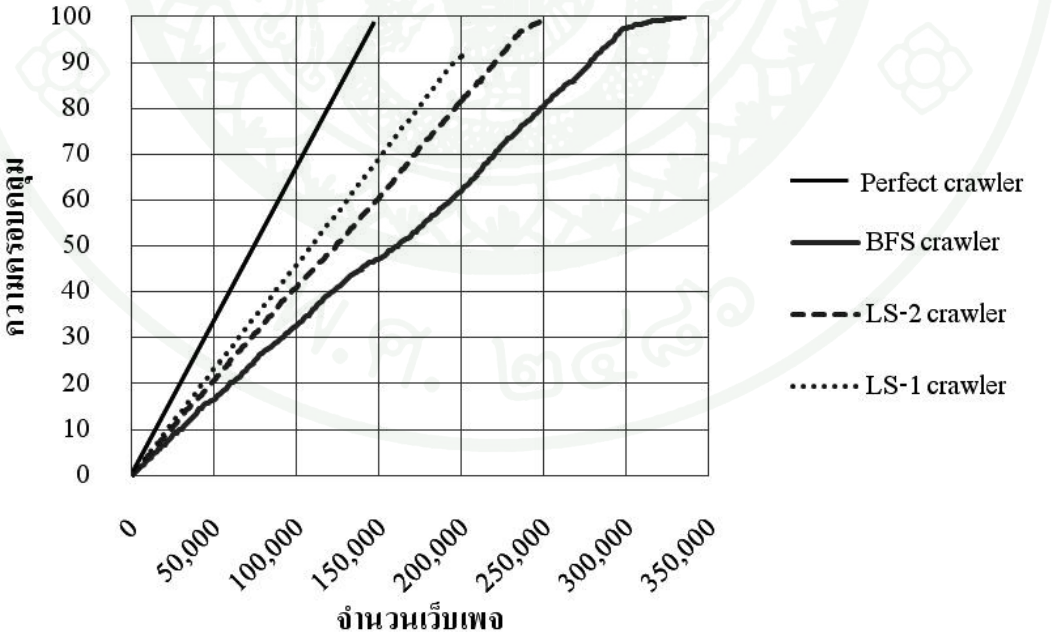


ภาพที่ 14 ความครอบคลุมของเว็บเพจของเว็บคราวเลอร์ทั้ง 5 ตัว จากการเก็บเว็บเพจจากชุดข้อมูลทดสอบในฐานข้อมูลเว็บภาษาไทย

จากการทดลองที่ 1 แสดงให้เห็นว่าการออกแบบเว็บคราวเลอร์เจาะจงภาษาในระดับของเว็บไซต์ทำให้อัตราการเก็บเกี่ยวที่ดีกว่า เว็บคราวเลอร์เจาะจงภาษาในระดับของเว็บเพจ ซึ่งในการทดลองที่ 2 จะทำการหาวิธีการเก็บเว็บเพจที่เหมาะสมต่อไป โดยเปรียบเทียบประสิทธิภาพวิธีการเก็บเว็บเพจของเว็บคราวเลอร์แบบแยกเก็บตามไซต์ที่นำเสนอไว้ทั้ง 2 วิธี ได้แก่ เว็บคราวเลอร์ LS-1 (หัวข้อที่ 1.4) และ LS-2 (หัวข้อที่ 1.5) นอกจากนี้ยังเปรียบเทียบกับเว็บคราวเลอร์แบบสมบูรณ์ (หัวข้อ 1.1) และเว็บคราวเลอร์พื้นฐาน (หัวข้อ 1.2) อีกด้วย โดยวิธีการทดลองเริ่มต้นจากการกำหนดกลุ่มของเว็บไซต์เริ่มต้นจำนวน 10,000 เว็บไซต์ ด้วยวิธีการสุ่มเลือกรายชื่อเว็บไซต์จากฐานข้อมูลเว็บภาษาไทย มาประเภทละ 5,000 เว็บไซต์ ซึ่งจากการทดลองปล่อยเว็บคราวเลอร์พื้นฐาน (หัวข้อ 1.2) พบว่าจากกลุ่มรายชื่อเว็บไซต์ที่สุ่มมา มีจำนวนเว็บเพจทั้งหมด 334,701 เว็บเพจ และเป็นเว็บเพจภาษาไทยจำนวน 203,219 เว็บเพจ จากนั้นวัดประสิทธิภาพการทำงานของเว็บคราวเลอร์ทั้ง 4 ตัวด้วยมาตรวัดอัตราการเก็บเกี่ยว และความครอบคลุม ได้ผลการทดสอบ ตามภาพที่ 15 และภาพที่ 16 ตามลำดับ



ภาพที่ 15 อัตราการเก็บเกี่ยวของเว็บคราวเลอร์ทั้ง 4 ตัวเพื่อเปรียบเทียบอัลกอริทึมในการเก็บเว็บเพจ



ภาพที่ 16 ความครอบคลุมของเว็บคราวเลอร์ทั้ง 4 ตัวเพื่อเปรียบเทียบอัลกอริทึมในการเก็บเว็บเพจ

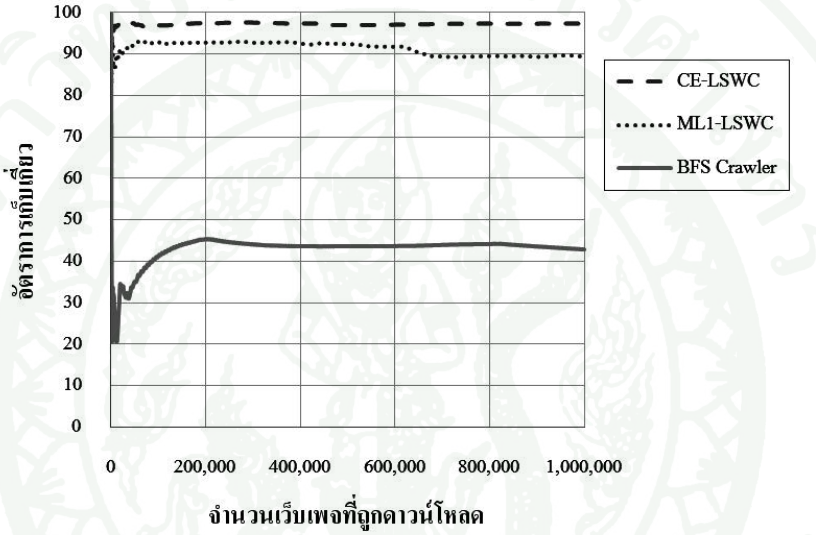
จากผลการทดลองที่ 2 สรุปได้ว่าเว็บคราวเลอร์แบบแยกเก็บตามไซต์ที่เลือกใช้วิธีการเก็บเว็บเพจที่ปรับปรุงจากวิธีการที่นำเสนอโดย Tamura *et al.* (2007) หรือ LS-2 ให้อัตราการเก็บเกี่ยวเว็บเพจภาษาไทยที่ดีกว่าเว็บคราวเลอร์ที่ปรับปรุงวิธีการเก็บเว็บเพจมาจากวิธีการค้นหาแบบกว้าง หรือ LS-1 แต่อย่างไรก็ตามเว็บคราวเลอร์ที่ปรับปรุงวิธีการเก็บเว็บเพจมาจากวิธีการค้นหาแบบกว้างกลับได้จำนวนเว็บเพจภาษาไทยที่มีความครบถ้วนมากกว่า ดังนั้นถ้าผู้ดูแลระบบเว็บคราวเลอร์ต้องการเก็บเว็บเพจภาษาไทย โดยมีแบนด์วิชดิ์ และเนื้อที่ในการเก็บข้อมูลจำกัด ควรจะเลือกใช้เว็บคราเวลอร์ LS-2 (ในหัวข้อ 1.5) แต่ถ้าผู้ดูแลระบบเว็บคราวเลอร์สนใจความครบถ้วนของเว็บเพจภาษาไทยควรจะเลือกใช้เว็บคราวเลอร์ LS-1 (ในหัวข้อ 1.4)

2. การทดสอบโดยเก็บเว็บเพจจริงจากอินเทอร์เน็ต

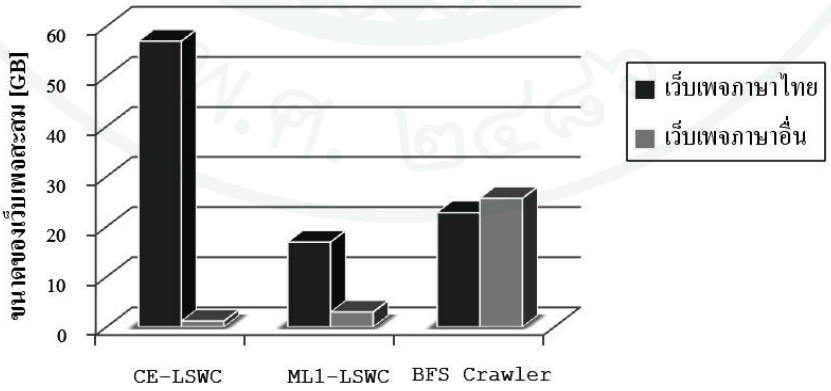
การทดสอบโดยเก็บเว็บเพจจริงจากอินเทอร์เน็ตจะเป็นการทดสอบประสิทธิภาพของระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษา แบบแยกเก็บตามไซต์ ระหว่างระบบต้นแบบที่ส่วนทำนายภาษาใช้เครื่องจักรเรียนรู้หนึ่งตัว (ML1-LSWC) กับส่วนทำนายภาษาที่ใช้เครื่องจักรเรียนรู้แบบหลายตัว (CE-LSWC) โดยในการทดสอบครั้งนี้จะใช้เพียงแค่มาตรวัดอัตราการเก็บเกี่ยวเท่านั้น เนื่องจากไม่ทราบจำนวนเว็บเพจภาษาไทยที่มีอยู่จริงบนอินเทอร์เน็ต

การทดลองเริ่มต้นให้เว็บคราวเลอร์ทั้งสามตัว ได้แก่ ML1-LSWC, CE-LSWC และ BFS Crawler เก็บเว็บเพจจากอินเทอร์เน็ต โดยได้กำหนดเว็บไซต์เริ่มต้น เป็นเว็บไซต์ภาษาไทยที่ได้รับความนิยม ได้แก่ http://truehits.net เป็นไดเรคทอรี่, http://www.kapook.com เป็นเว็บไซต์ให้บริการข้อมูลด้านข่าวสารและความบันเทิง และ http://www.pantip.com เป็นเว็บไซต์คอมมูนิตี้ นอกจากนี้ยังกำหนดให้เว็บคราวเลอร์แต่ละตัวสามารถเก็บเว็บเพจมาได้ไม่เกิน 1,000 เว็บเพจต่อเว็บไซต์ และหยุดการทดลองเมื่อเว็บคราวเลอร์แต่ละตัวเก็บเว็บเพจได้ครบ 1 ล้านเว็บเพจแล้ว ซึ่งผลการวัดประสิทธิภาพของคราวเลอร์ทั้งสามด้วยอัตราการเก็บเกี่ยว แสดงดังภาพที่ 17 พบว่าเว็บคราวเลอร์พื้นฐาน (BFS crawler) หลังจากดาวน์โหลดเว็บเพจครบ 1 ล้านเว็บเพจแล้วได้อัตราการเก็บเกี่ยวเท่ากับ 42.91% ประกอบด้วยเว็บเพจภาษาไทยจำนวน 429,134 เว็บเพจ และเว็บเพจภาษาอื่นอีก 570,867 เว็บเพจ ส่วน ML1-LSWC ซึ่งเป็นเว็บคราวเลอร์เจาะจงภาษาแบบแยกเก็บตามไซต์ แต่ส่วนทำนายภาษาใช้เทคนิคเครื่องจักรเรียนรู้แบบหนึ่งตัวนั้น ได้อัตราการเก็บเกี่ยวเท่ากับ 89.25% โดยมีเว็บเพจภาษาไทยจำนวน 892,497 เว็บเพจ และสำหรับระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาแบบแยกเก็บตามไซต์ โดยใช้เครื่องจักรเรียนรู้แบบหลายตัวเป็นส่วนทำนายภาษา และใช้วิธีการเก็บเว็บเพจที่ปรับปรุงมาจากวิธีการที่นำเสนอโดย Tamura *et al.* (2007) ให้

อัตราการเก็บเกี่ยวมากที่สุด ถึง 97.34% ซึ่งได้จำนวนเว็บเพจภาษาไทยถึง 973,417 เว็บเพจ นอกจากนี้เมื่อนำเว็บเพจทั้งหมดมาวิเคราะห์เพื่อหาจำนวนแบนด์วิดช์ที่ใช้ไปในการทดลอง พบว่า เว็บคราวเลอร์พื้นฐาน (BFS craweler) ดาวน์โหลดข้อมูลเว็บเพจภาษาไทย 22 กิกะไบต์ และเว็บเพจภาษาอื่นอีก 25 กิกะไบต์ ส่วน ML1-LSWC ใช้แบนด์วิดท์ในการเก็บข้อมูลเว็บเพจภาษาไทย จำนวน 17 กิกะไบต์ ส่วนเว็บเพจภาษาอื่นอีก 3 กิกะไบต์ และเว็บคราวเลอร์ตัวสุดท้ายที่นำเสนอ (CE-LSWC) ใช้แบนด์วิดท์ในการเก็บเว็บเพจภาษาไทยทั้งหมด 57 กิกะไบต์ และมีเว็บเพจภาษาอื่นเพียง 1 กิกะไบต์เท่านั้น ตามภาพที่ 18



ภาพที่ 17 อัตราการเก็บเกี่ยวของเว็บคราวเลอร์ทั้ง 3 ตัว โดยที่เก็บเว็บเพจจากอินเทอร์เน็ต



ภาพที่ 18 แสดงปริมาณข้อมูลที่เว็บคราวเลอร์ทั้ง 3 ตัวเก็บมาจากอินเทอร์เน็ต แยกตามประเภทของเว็บไซต์

หลังจากจบการทดลองที่ 2 แล้ว ได้ทดลองปล่อยให้ระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาโดยใช้เครื่องจักรเรียนรู้แบบหลายตัว ทำการเก็บเว็บเพจจนครบทุกเว็บไซต์ที่ได้ทำนายไว้ว่า มีความน่าจะเป็นที่เว็บไซต์ปลายทางจะให้บริการเว็บเพจภาษาไทย พบค่าสถิติของฐานข้อมูลเว็บภาษาไทยใหม่ ซึ่งเริ่มเก็บในเดือนเมษายน 2554 ตามตารางที่ 17 ซึ่งพบเว็บไซต์ภาษาไทยภายใต้โดเมนระดับบนสุดที่เป็น .com มากที่สุดถึง 20,575 เว็บไซต์ รองลงมาคือ .th จำนวน 4,650 เว็บไซต์

ตารางที่ 17 สถิติของฐานข้อมูลเว็บภาษาไทยในเดือนเมษายน 2554

โดเมนระดับบนสุด	จำนวนเว็บไซต์ภาษาไทย	จำนวนเว็บไซต์ภาษาอื่น
.com	20,575	3,545
.net	957	121
.org	670	171
.th	4,650	782
.info	52	11
โดเมนอื่นๆ	99	47
รวม	27,003	4,677

จากการตรวจสอบฐานข้อมูลเว็บภาษาไทยที่เพิ่งสร้างในเดือนเมษายน พ.ศ. 2554 พบว่ามีบางเว็บไซต์มีอยู่ในฐานข้อมูลเว็บภาษาไทยที่สร้างในเดือนกันยายน พ.ศ. 2552 ด้วย ยิ่งไปกว่านั้นยังพบว่า ประเภทของเว็บไซต์มีการเปลี่ยนแปลง เช่น จากเดิมในปี 2552 เป็นเว็บไซต์ภาษาไทย แต่มาเก็บเว็บเพจใหม่ในปี พ.ศ. 2554 กลายเป็นเว็บไซต์ภาษาอื่น จำนวน 459 เว็บไซต์ ในทางกลับกัน ก็มีบางเว็บไซต์ที่เป็นเว็บไซต์ภาษาอื่นจากการเก็บเว็บเพจในปี 2552 แต่กลายมาเป็นเว็บไซต์ภาษาไทยในปี 2554 จำนวน 198 เว็บไซต์ ส่วนรายละเอียดการเปลี่ยนแปลงประเภทของเว็บไซต์ แยกตามโดเมนระดับบนสุด แสดงในตารางที่ 18 แต่อย่างไรก็ตาม ก็พบเว็บไซต์ที่มีประเภทเว็บไซต์เหมือนเดิมเช่นกัน แสดงในตารางที่ 19

ตารางที่ 18 สถิติการเปลี่ยนแปลงประเภทของเว็บไซต์จากการเก็บเว็บเพจในปี 2552 และ 2554

โดเมนระดับบนสุด	เปลี่ยนจากเว็บไซต์ภาษาอื่นมาเป็นเว็บไซต์ภาษาไทย	เปลี่ยนจากเว็บไซต์ภาษาไทยมาเป็นเว็บไซต์ภาษาอื่น
.com	22	59
.net	3	12
.org	3	8
.th	170	377
.info	0	2
โดเมนอื่นๆ	0	1
รวม	198	459

ตารางที่ 19 สถิติของประเภทเว็บไซต์ที่ไม่เปลี่ยนแปลงทั้งจากการเก็บเว็บเพจในปี 2552 และ 2554

โดเมนระดับบนสุด	เว็บไซต์ภาษาไทย	เว็บไซต์ภาษาอื่น
.com	528	25
.net	110	1
.org	53	4
.th	3,082	135
.info	1	0
โดเมนอื่นๆ	8	3
รวม	3,782	168

สุดท้ายนี้จากการสร้างฐานข้อมูลเว็บภาษาไทยทั้งสองครั้งในปี 2552 และ 2554 เมื่อรวมฐานข้อมูลกันทำให้ได้รายชื่อเว็บไซต์ทั้งภาษาไทย 46,636 เว็บไซต์ และรายชื่อเว็บไซต์ภาษาอื่นอีก 23,017 เว็บไซต์ ดังแสดงในตารางที่ 20

ตารางที่ 20 สถิติของการรวมฐานข้อมูลเว็บภาษาไทย ในปี 2552 และ 2554

โดเมนระดับบนสุด	เว็บไซต์ภาษาไทย	เว็บไซต์ภาษาอื่น
.com	18,717	9,649
.net	885	842
.org	639	1,689
.th	25,686	7,358
.info	56	68
โดเมนอื่นๆ	653	3,411
รวม	46,636	23,017

วิจารณ์

ถึงแม้ว่าระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาแบบแยกเก็บตามไซต์ โดยใช้เครื่องจักรเรียนรู้แบบหลายตัว จะมีขั้นตอนที่ยุ่งยาก ซับซ้อน และจำเป็นต้องใช้ชุดข้อมูลฝึกสอน และชุดข้อมูลทดสอบจำนวนมาก เพื่อมาสร้างเป็นระบบต้นแบบนั้น ทำให้นักออกแบบต้องสูญเสียทั้งเวลา และทรัพยากรระบบอย่างมาก แต่ผลลัพธ์ที่ได้กลับมาก็ถือว่าคุ้มค่า ซึ่งจะเห็นได้จากผลการทดลองให้ระบบต้นแบบเก็บเว็บเพจจากอินเทอร์เน็ตหนึ่งล้านเว็บเพจ พบว่า วิธีการที่นำเสนอมีอัตราการเก็บเกี่ยวเป็นที่น่าพอใจอย่างมาก เมื่อเทียบกับเว็บคราวเลอร์ตัวอื่นๆ ยิ่งไปกว่านั้นจำนวนเว็บไซต์บนอินเทอร์เน็ตนับวันจะมีแต่เพิ่มจำนวนมากขึ้น และข้อมูลเว็บเพจมีการเปลี่ยนแปลงตลอดเวลา ดังนั้นถ้าเราไม่ใช้เทคนิคทางเครื่องจักรเรียนรู้มาช่วยกรองเว็บไซต์ที่ไม่เกี่ยวข้องออกไปบ้าง จะทำให้ไม่สามารถเก็บรวบรวมเว็บเพจที่สนใจจำนวนมากภายใต้ระยะเวลาที่จำกัดได้ สุดท้ายนี้จากการวิเคราะห์ฐานข้อมูลเว็บภาษาไทยในปี 2552 และปี 2554 ถึงแม้ว่าการสร้างฐานข้อมูลทั้งสองครั้งจะเริ่มต้นจากลุ่มเว็บไซต์ที่แตกต่างกัน แต่เมื่อปล่อยให้เว็บคราวเลอร์ทำงานก็กลับพบเว็บไซต์ที่เป็นสมาชิกร่วมกันจากทั้งสองฐานข้อมูลถึง 4,607 เว็บไซต์

สรุปและข้อเสนอแนะ

สรุป

วิทยานิพนธ์ฉบับนี้นำเสนอระบบต้นแบบเว็บคราวเลอร์เจาะจงภาษาแบบแยกเก็บตามไซต์ เพื่อมุ่งเน้นหาเว็บไซต์ที่ให้บริการเว็บเพจภาษาไทย โดยมีส่วนประกอบหลัก 2 ส่วน คือ ส่วนของการทำนายภาษาของเว็บไซต์เป้าหมาย ซึ่งส่วนนี้จะใช้เทคนิคทางเครื่องจักรเรียนรู้แบบหลายตัว เพื่อช่วยความแม่นยำในการทำนายหาเว็บไซต์เป้าหมายที่จะให้บริการเว็บเพจภาษาไทย และอีกส่วนหนึ่งคือ ส่วนของเว็บคราวเลอร์แบบแยกเก็บตามไซต์ ซึ่งมีหน้าที่เก็บเว็บเพจภายใต้เว็บไซต์ที่กำหนด ซึ่งได้ปรับปรุงวิธีการเก็บเว็บเพจให้มีประสิทธิภาพโดยอาศัยกฎฮิวริสติกส์อย่างง่าย ในส่วนของการทดลองให้เว็บคราวเลอร์เก็บเว็บเพจจริงจากอินเทอร์เน็ต และได้ใช้มาตรวัดอัตราการเก็บเกี่ยว เพื่อเปรียบเทียบประสิทธิภาพของระบบต้นแบบที่นำเสนอกับเว็บคราวเลอร์เจาะจงภาษาแบบดั้งเดิม โดยผลการทดลองชี้ให้เห็นว่า ระบบที่นำเสนอมีประสิทธิภาพที่ดีกว่าวิธีการของเว็บคราวเลอร์เจาะจงภาษาที่ผ่านมา นอกจากนี้ยังได้รายชื่อเว็บไซต์ที่ถูกระบุว่าเป็นเว็บไซต์ภาษาไทยทั้งหมด 46,636 เว็บไซต์อีกด้วย

ข้อเสนอแนะ

สำหรับงานวิจัยต่อเนื่องยังมีอีกหลายเรื่องที่น่าสนใจ เช่น การปรับปรุงฐานความรู้ของเครื่องจักรเรียนรู้ในส่วนของการทำนายภาษา โดยใช้เว็บเพจที่เก็บมาแล้วให้เป็นประโยชน์ หรือการค้นหาคุณลักษณะใหม่ที่สามารถสกัดได้จากโฮสต์กราฟ ตัวอย่างเช่น การหาเส้นทางการเชื่อมโยงของเว็บไซต์ปลายทางในระดับก่อนหน้า คล้ายกับการหาระยะทางจากเว็บเพจภาษาไทยล่าสุด เป็นต้น นอกจากนี้อาจจะหาวิธีการตรวจหาเว็บไซต์ที่เป็นขอบของภาษาที่ต้องการ เนื่องจากลิงค์ที่สกัดได้จากเว็บไซต์ที่อยู่บริเวณขอบของภาษา จะมีโอกาสน้อยมากที่ลิงค์จะชี้ไปยังเว็บไซต์ภาษาที่สนใจ

เอกสารและสิ่งอ้างอิง

ธนพล สืบเชื้อ, ปุณณวัฒน์ ธาดาภาคย์ และอานนท์ รุ่งสว่าง. 2553. เว็บคราวเลอร์เจาะจงเว็บเพจภาษาไทยแบบแยกเก็บตามไซต์. *ใน***การประชุมทางวิชาการของมหาวิทยาลัยเกษตรศาสตร์ ครั้งที่ 48**. มหาวิทยาลัยเกษตรศาสตร์, กรุงเทพฯ.

Abdeen, M. and Tolba, M.F. 2010. Challenges and design issues of an Arabic web crawler, pp. 203-206. In **Proceedings of the International Conference on Computer and System.**

Alabbad, S.H. and Alanazi S. 2009. Language Based Crawling: Crawling the Arabic Content of the Web, pp. 83-88. In **Proceedings of the International Conference on Internet Computing.**

Azimzadeh, M., Yari, A. and Kargar, M.J. 2010. Language specific crawling based on web pages features. In **Proceesings of International Conference on Multimedia Computing and Information Technology.**

Baidu. 2011. **Chinese Web Search Engine**. Available Source: http://www.baidu.com/, August 8, 2011.

Baykan, E., Henzinger, M. and Weber, I. 2008. Web page language identification based on URLs. **VLDB Endow 1** (1): 176-187.

Bing. 2011. **Bing Web Search Engine**. Available Source: http://www.bing.com/, August 8, 2011.

Cavnar, W. and Trenkle J. 1994. N-Gram-Based Text Categorization, pp. 161-175. In **Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval.**

Chakrabarti, S., M. van den Berg and B. Dom. 1999. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. **Computer Networks: The International Journal of Computer and Telecommunications Networking** 31 (11-16): 1623-1640.

Chan, S. and Yamana H. 2010. The Method of Improving the Specific Language Focused Crawler. In **Proceedings of the Joint Conference on Chinese Language Processing.**

Cho, J. and H. Garcia-Molina. 2002. Parallel Crawlers. In **Proceedings of the 11th international conference on World Wide Web.** ACM, New York, USA.

Goebel, K.F. and W. Yan. 2004. Choosing Classifiers for Decision Fusion, pp. 563-568. In **Proceedings of the Seventh International Conference on Information Fusion.**

Gomes, D., Nogueira A., Miranda J. and Costa M. 2008. Introducing the Portuguese Web Archive Initiative, In **8th International Web Archiving Workshop.** Denmark.

Google**.** 2011. **Google Web Search Engine**. Available Source: http://www.google.com/, August 8, 2011.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten. 2009. The Weka Data Mining Software: An Update. **SIGKDD Explorations 2009** (1): 10-18.

Heritrix**.** 2011**. Internet Archive Web Crawler Project.** Available Source: http://crawler.archive.org/, August 8, 2011.

Kuncheva, L.I. 2004. **Combining Pattern Classifiers: Methods and Algorithms.** Wiley-Interscience, New Jersey, Canada.

Kunder M. 2011. **World Wide Web Size**. Available Source: http://www.worldwidewebsize.com/, August 8, 2011.

InetAddressLocator. 2011. **API for discovers geographical location from IP addresses.** Available Source: http://javainetlocator.sourceforge.net/, August 8, 2011.

LEXiTRON. 2011. **Thai-English Electronic Dictionary.** Available Source : http://lexitron.nectec.or.th/, August 8, 2011.

LexTo. 2011. **Thai Lexeme Tokenizer.** Available Source: http://sansarn.com/lexto/, August 8, 2011.

Medelyan O., Schulz S., Paetzold J., Poprat M. and Marko K. 2006. Language specific and topic focused web crawling. In **Proceedings of the Language Resouces Conference.** Italy.

Mon P., Choong C. and Mikami Y. 2011. Language Specific Crawler for Myanmar Web Pages. **Computer Science 8** (2): 127-135.

Rauber, A., Aschenbrenner, A., Witvoet, O. 2002. Austrian on-line archive processing: analyzing archives of the world wide web, pp. 16-31. In **Proceedings of the 6th European Conference on Digital Libraries.**

Ruta, D. and B. Gabrys. 2000. An Overview of Classifier Fusion Methods. **Computing and Information Systems,** 7 (1): 1-10.

Sanguanpong, S. and K. Koht-Arsa. 2003. Structure Properties of the Thai WWW: The 2003 Survey. In **The Conference on Internet Technology (CIT2003).** Asian Institute of Technology, Pathumthani, Thailand.

Somboonviwat, K., M. Kitsuregawa and T. Tamura. 2005. Simulation Study of Language Specific Web Crawling, pp. 1254-1258. In **Proceedings of the 21st International Conference on Data Engineering Workshops.** IEEE Computer Society.

Somboonviwat, K., T. Tamura and M. Kitsuregawa. 2006. Finding Thai Web Pages in Foreign Web Spaces, pp. 135. In **Proceedings of the 22nd International Conference on Data Engineering Workshops.** IEEE Computer Society.

Srisukha, E., S. Jinarat, C. Haruechaiyasak and A. Rungsawang. 2008. Naïve Bayes Based Language-Specific Web Crawling, pp. 113-116. In **Proceedings of the 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology.** IEEE Press.

Tadapak, P., T. Suebchua and A. Rungsawang. 2010. A Machine Learning Based Language Specific Web Site Crawler, pp. 155-161. In **Proceedings of the 13th International Conference on Network-Based Information Systems.**

Tamura, T., K. Somboonviwat and M. Kitsuregawa. 2007. A Method for Language-Specific Web Crawling and Its Evaluation. **Systems and Computers in Japan** 38 (2): 10-20.

Yahoo. 2011. **Yahoo! Web Search Engine.** Available Source: http://www.yahoo.com/, August 8, 2011.

ภาคผนวก

ภาคผนวก ก

ผลการทดลองเพิ่มเติม

ผลการทดลองเพิ่มเติม

ในส่วนนี้จะแสดงผลการทดลองทั้งหมดที่ได้ทดสอบไว้อย่างละเอียด เนื่องจากในส่วนของวิธีการ และผลการทดลองจำเป็นต้องสรุป และคัดเลือกนำเสนอเฉพาะกรณีที่สำคัญ เพื่อให้เห็นถึงภาพรวมอย่างคร่าวๆ ของวิธีการ หรือผลการทดลองที่นำเสนออย่างสังเขป

ตารางผนวกที่ ก1 ผลการฝึกสอน และทดสอบเครื่องจักรเรียนรู้แบบต่างๆ บนชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ

ข้อมูลฝึกสอนชุดที่	เทคนิคเครื่องจักรเรียนรู้	ความแม่นยำบนชุดข้อมูลฝึกสอน โดยการตรวจสอบไขว้ 10 พับ	ความแม่นยำบนชุดข้อมูลทดสอบ
1	Naïve Bayes	78.25	78.62
	Bayes Network	78.25	78.63
	RBS Network	78.18	78.53
	SVM	78.23	78.63
	10-NN	78.15	78.57
	Decision Tree	78.26	78.58
	Random Forest	78.26	78.60
2	Naïve Bayes	77.88	78.18
	Bayes Network	77.89	78.19
	RBS Network	77.83	78.07
	SVM	76.52	65.18
	10-NN	77.88	78.13
	Decision Tree	78.00	78.26
	Random Forest	78.07	78.28
3	Naïve Bayes	73.16	73.09
	Bayes Network	73.17	73.12
	RBS Network	73.37	74.46
	SVM	76.64	77.01
	10-NN	76.34	76.41

ตารางผนวกที่ ก1 (ต่อ)

ข้อมูลฝึกสอน ชุดที่	เทคนิคเครื่องจักรเรียนรู้	ความแม่นยำบนชุดข้อมูลฝึกสอน โดยการตรวจสอบไขว้ 10 พับ	ความแม่นยำบนชุดข้อมูลทดสอบ
	Random Forest	75.12	75.56
4	Naïve Bayes	54.11	55.59
	Bayes Network	54.10	55.59
	RBS Network	54.12	55.58
	SVM	54.16	55.65
	10-NN	54.04	55.54
	Decision Tree	54.10	55.64
	Random Forest	54.13	55.65
5	Naïve Bayes	77.33	77.76
	Bayes Network	77.33	77.75
	RBS Network	78.63	78.12
	SVM	78.12	68.60
	10-NN	79.49	78.08
	Decision Tree	79.53	78.10
	Random Forest	79.56	77.94
6	Naïve Bayes	78.44	77.97
	Bayes Network	78.52	78.04
	RBS Network	77.90	77.00
	SVM	80.30	78.91
	10-NN	80.49	80.09
	Decision Tree	80.90	80.36
	Random Forest	79.74	78.69
7	Naïve Bayes	78.24	78.59
	Bayes Network	78.24	78.59

ตารางผนวกที่ ก1 (ต่อ)

ข้อมูลฝึกสอนชุดที่	เทคนิคเครื่องจักรเรียนรู้	ความแม่นยำบนชุดข้อมูลฝึกสอน โดยการตรวจสอบไขว้ 10 พับ	ความแม่นยำบนชุดข้อมูลทดสอบ
	RBS Network	78.19	78.49
	SVM	78.22	78.62
	10-NN	78.16	78.56
	Decision Tree	78.26	78.59
	Random Forest	78.27	78.61
8	Naïve Bayes	77.76	77.75
	Bayes Network	77.78	77.80
	RBS Network	77.40	78.51
	SVM	79.47	76.47
	10-NN	79.66	78.96
	Decision Tree	79.93	80.00
	Random Forest	78.74	78.60
9	Naïve Bayes	77.84	78.17
	Bayes Network	77.85	78.18
	RBS Network	77.81	78.07
	SVM	76.41	65.13
	10-NN	77.89	78.14
	Decision Tree	78.00	78.26
	Random Forest	78.07	78.30
10	Naïve Bayes	73.44	73.29
	Bayes Network	73.46	73.32
	RBS Network	73.52	74.72
	SVM	76.59	76.88
	10-NN	76.35	76.44
	Decision Tree	76.00	76.78

ตารางผนวกที่ ก1 (ต่อ)

ข้อมูลฝึกสอน ชุดที่	เทคนิคเครื่องจักรเรียนรู้	ความแม่นยำบนชุดข้อมูลฝึกสอน โดยการตรวจสอบไขว้ 10 พับ	ความแม่นยำบนชุดข้อมูลทดสอบ
	Random Forest	75.11	75.61
11	Naïve Bayes	80.05	79.99
	Bayes Network	80.13	80.06
	RBS Network	79.46	78.83
	SVM	80.64	77.81
	10-NN	81.47	80.08
	Decision Tree	81.60	80.17
	Random Forest	80.62	79.30
12	Naïve Bayes	77.31	77.75
	Bayes Network	77.30	77.74
	RBS Network	78.63	78.10
	SVM	78.06	68.51
	10-NN	79.49	78.08
	Decision Tree	79.53	78.10
	Random Forest	79.53	77.85
13	Naïve Bayes	78.50	78.00
	Bayes Network	78.54	78.08
	RBS Network	77.99	76.97
	SVM	80.33	78.71
	10-NN	80.49	80.07
	Decision Tree	80.90	80.36
	Random Forest	79.86	78.89
14	Naïve Bayes	77.76	77.73
	Bayes Network	77.81	77.77
	RBS Network	77.52	78.51

ตารางผนวกที่ ก1 (ต่อ)

ข้อมูลฝึกสอนชุดที่	เทคนิคเครื่องจักรเรียนรู้	ความแม่นยำบนชุดข้อมูลฝึกสอน โดยการตรวจสอบไขว้ 10 พับ	ความแม่นยำบนชุดข้อมูลทดสอบ
	SVM	79.50	76.48
	10-NN	79.66	78.96
	Decision Tree	79.93	80.00
	Random Forest	78.72	78.65
15	Naïve Bayes	80.07	79.98
	Bayes Network	80.11	80.06
	RBS Network	79.45	78.80
	SVM	80.59	77.80
	10-NN	81.48	80.07
	Decision Tree	81.60	80.17
	Random Forest	80.41	79.17
16	Naïve Bayes	72.82	73.45
	Bayes Network	72.94	73.54
	RBS Network	74.96	76.04
	SVM	76.60	77.20
	10-NN	76.80	77.29
	Decision Tree	76.14	77.16
	Random Forest	74.92	75.35
17	Naïve Bayes	87.49	84.95
	Bayes Network	87.20	85.11
	RBS Network	90.60	86.31
	SVM	92.24	83.26
	10-NN	91.29	87.24
	Decision Tree	**93.27**	86.45
	Random Forest	91.74	86.58

ภาคผนวก ข

การใช้งานโปรแกรมเว็บคราวเลอร์ Heritrix เบื้องต้น

การใช้งานโปรแกรมเว็บคราวเลอร์ Heritrix เบื้องต้น

โปรแกรม Heritrix เป็นส่วนหนึ่งในโครงการ Internet Archive ซึ่งเป็นโปรแกรมเว็บคราวเลอร์แบบโอเพนซอร์สที่ถูกพัฒนาด้วยภาษาจาวา โดยที่ทีมพัฒนาได้ออกแบบให้โปรแกรม Heritrix สามารถรองรับการเก็บเว็บเพจจำนวนมากๆ ได้ และคำนึงถึงการเข้าเยี่ยมชมเว็บเพจที่เคารพกฎ robots.txt ด้วย นอกจากนี้ยังมีความสามารถสกัดลิงค์จาก JavaScript และไฟล์ Adobe Flash จากเว็บเพจได้อีกด้วย ในส่วนของการใช้งานได้พัฒนาส่วนติดต่อผู้ใช้ผ่านเว็บ ซึ่งสามารถตั้งค่าพื้นฐานต่างๆ ของเว็บคราวเลอร์ได้ เช่น การเลือกวิธีการค้นหาเว็บเพจได้ทั้งแบบกว้าง (breadth-first search) และแบบลึก (depth-first search) การกำหนดจำนวนเว็บเพจสูงสุดที่จะเก็บต่อเว็บไซต์หรือต่อครั้ง สามารถกำหนดชนิดของเอกสารที่ต้องการได้ เช่น text/html เป็นต้น ซึ่งส่วนติดต่อผู้ใช้แบ่งการแสดงผลออกเป็น 7 ส่วน ได้แก่

ส่วนที่ 1 Console เป็นหน้าหลักที่แสดงสถานะการทำงานของเว็บคราวเลอร์ หน่วยความจำที่ถูกใช้งาน ขนาดหน่วยความจำที่จองไว้เริ่มต้น

ส่วนที่ 2 Job แสดงรายการงานทั้งหมด

ส่วนที่ 3 Profiles แสดงรายการโปรไฟล์ทั้งหมด โดยโปรไฟล์จะสามารถเก็บรายละเอียดการตั้งค่าของเว็บคราวเลอร์ตามที่ต้องการได้

ส่วนที่ 4 Logs แสดงรายการ log ของเว็บคราวเลอร์ขณะที่กำลังทำงานอยู่

ส่วนที่ 5 Reports แสดงรายงานของเว็บคราวเลอร์ทั้งงานที่กำลังทำอยู่กับงานที่เว็บคราวเลอร์ทำงานเสร็จเรียบร้อยแล้ว โดยมีรายงานอยู่ 5 แบบ ได้แก่ crawl report seed report frontier report processors report และ toe thread report

ส่วนที่ 6 Setup เป็นส่วนที่เอาไว้จัดการ instance ของ Heritrix

ส่วนที่ 7 Help ประกอบด้วยคู่มือการใช้งานโปรแกรมทั้งในมุมมองของผู้ใช้งานทั่วไป และมุมมองของนักพัฒนา แสดงลิงค์สำหรับแจ้งปัญหาการใช้งาน แสดงลิงค์ไปยังกระดานสนทนาของ

กลุ่มผู้ที่ใช้งานโปรแกรม Heritrix และการอธิบายรหัสสถานะ HTTP ต่างๆ ที่ถูกแสดงไว้ในส่วนของ Logs

ขั้นตอนการติดตั้งโปรแกรมเว็บคราวเลอร์ Heritrix

1. ติดตั้ง Java Software Development Kit (JDK) เวอร์ชั่น 6.0 และกำหนดค่าตัวแปร JAVA_HOME และ JAVA_OPTS ให้กับระบบปฏิบัติการ

1.1 ตัวแปร JAVA_HOME เพื่อระบุ Path ที่ติดตั้ง JDK เช่น JAVA_HOME=/opt/jdk1.6.0_04

1.2 ตัวแปร JAVA_OPTS เพื่อให้ Java virtual machine จองหน่วยความจำเริ่มต้นให้ตามที่ต้องการ เพื่อป้องกันปัญหาหน่วยความจำไม่เพียงพอ (Java heap space) ในขณะที่กำลังทำงาน เช่น ต้องการให้ Java virtual machine จองหน่วยความจำเริ่มต้น 1024 MB ทำได้โดย JAVA_OPTS=-Xmx1024M

2. ติดตั้งโปรแกรม Apache Tomcat เวอร์ชั่น 7.0.16

สร้างบัญชีผู้ใช้ เนื่องจากโปรแกรม Heritrix จะมาตรวจสอบสิทธิ์การใช้งานจากไฟล์ $TOMCAT_HOME/conf/tomcat-users.xml โดยกำหนด roles เป็น admin เท่านั้น (ภาพผนวกที่ ข1)

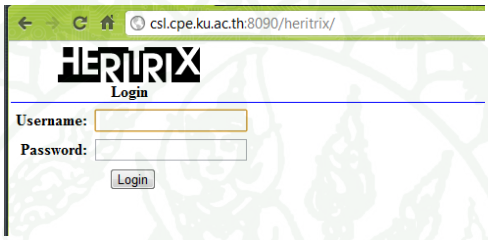
```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
    <role rolename="manager"/>
    <role rolename="admin"/>
    <user username="tomcat_admin" password="1a2s3d4f" roles="manager"/>
    <user username="pun_admin" password="1a2s3d4f" roles="admin"/>
</tomcat-users>
```

ภาพผนวกที่ ข1 ตัวอย่างไฟล์ tomcat-users.xml

3. ติดตั้งโปรแกรม Heritrix โดยนำไฟล์ heritrix.war ไปวางที่ $TOMCAT_HOME/webapps ซึ่งไฟล์ heritrix.war ได้มาจากการคอมไพล์ซอร์สโค้ดที่ได้ดาวน์โหลดมาใหม่ หรือดาวน์โหลดได้โดยตรงจากเว็บไซต์http://builds.archive.org:8080/cruisecontrol/buildresults/HEAD-heritrix

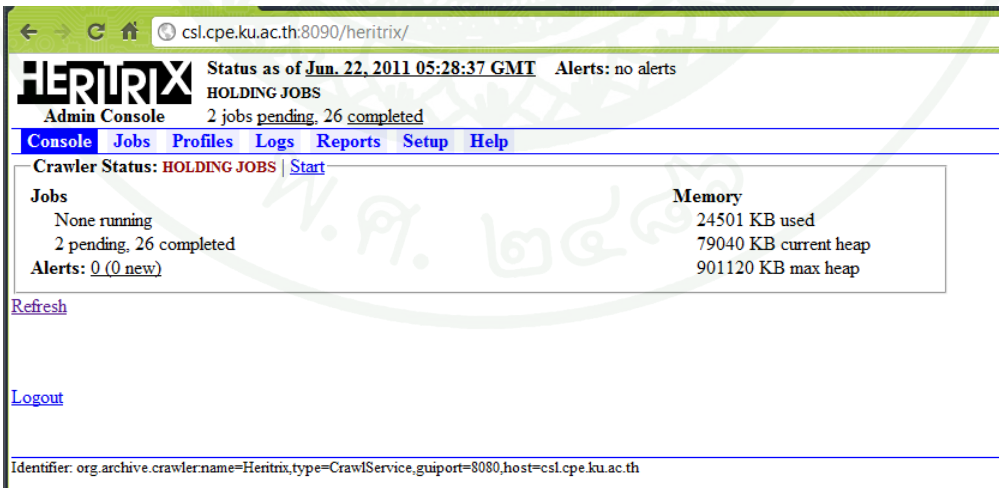
4. สั่งให้ Apache Tomcat เริ่มทำงาน โดยไปที่ $TOMCAT_HOME/bin/startup.sh (Linux) หรือ $TOMCAT_HOME/bin/startup.bat (Windows)

5. เปิดเว็บบราวเซอร์แล้วพิมพ์ URL http://localhost:8080/heritrix จะได้หน้าเว็บตามภาพผนวกที่ ข2



ภาพผนวกที่ ข2 เว็บเพจหน้า Login ของโปรแกรม Heritrix

6. Login เข้าโปรแกรมโดยใช้ Username และ password ที่กำหนดไว้ตามขั้นตอนที่ 3 จากนั้นจะเว็บเพจตามภาพผนวกที่ ข3 ถือว่าติดตั้งโปรแกรมเสร็จสมบูรณ์



ภาพผนวกที่ ข3 เว็บเพจหน้า Console ของโปรแกรม Heritrix

ตัวอย่างการใช้โปรแกรมเว็บคราวเลอร์ Heritrix

ในส่วนนี้จะแสดงตัวอย่างการใช้งานโปรแกรม Heritrix โดยมีการตั้งค่าให้เว็บคราวเลอร์ทำงานแบบแยกเก็บตามไซต์ และกำหนดให้เว็บคราวเลอร์เลือกเก็บเฉพาะเว็บเพจ (text/html) โดยที่สามารถเก็บเว็บเพจได้ไม่เกิน 300 เว็บเพจต่อเว็บไซต์ ซึ่งประกอบด้วยขั้นตอนดังต่อไปนี้

1. การสร้างงาน (job) มีรายละเอียดดังนี้

1.1 เข้าใช้งานโปรแกรม Heritrix จากนั้นไปที่แท็บ “Jobs”

1.2 ในส่วนของ Create new job ให้เลือก With defaults จะปรากฏเว็บเพจตามภาพผนวกที่ ข4

1.3 ตั้งชื่องาน และคำอธิบาย ในช่อง Name of new job และ Description ตามลำดับ

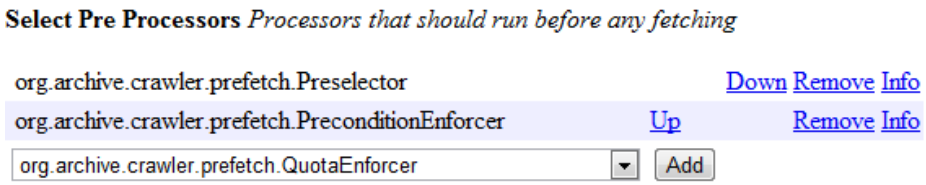
1.4 กำหนดเว็บไซต์เริ่มต้นในช่อง Seeds

1.5 คลิกที่ปุ่ม “Modules”

ภาพผนวกที่ ข4 หน้าเว็บเพจของการสร้างงานใหม่

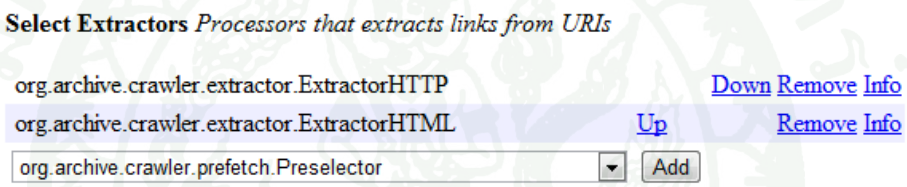
2. การตั้งค่าใน Modules เพื่อเพิ่มความสามารถให้กับเว็บคราวเลอร์ ได้แก่

2.1 การกำหนดจำนวนเว็บเพจสูงสุดที่จะต้องเก็บ โดยไปที่ส่วนของ “Select Pre Processors” และเลือก org.archive.crawler.prefetch.QuataEnforcer จากนั้นกดปุ่ม Add (ภาพผนวกที่ ข5)



ภาพผนวกที่ ข5 โมดูลในส่วนของ Select Pre Processors

2.2 การกำหนดให้เว็บคราวเลอร์สกัดลิงค์จากเว็บเพจ (text/html) เท่านั้น โดยค่าปริยายของโปรแกรมจะกำหนดไว้ให้สามารถลิงค์ทั้งจากเว็บเพจ, css, JavaScript และ Adobe Flash เพราะฉะนั้นต้องไปที่ส่วนของ “Select Extractors” แล้วลบโมดูล โดยคลิกที่ Remove ให้เหลือโมดูล ExtractorHTTP และ ExtractorHTML ตามภาพผนวกที่ ข6



ภาพผนวกที่ ข6 โมดูลในส่วนของ Select Extractors

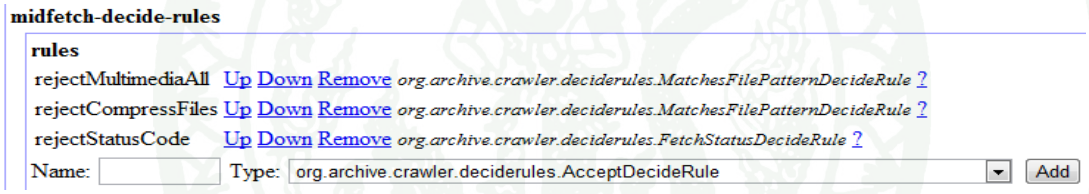
2.3 คลิก “Submodules” ซึ่งอยู่ตำแหน่งล่างสุดของเว็บเพจ เพื่อไปกำหนดรายละเอียดของโมดูลที่เลือกไว้

3. การตั้งค่าของเว็บคราวเลอร์อย่างคร่าวๆ ใน Submodules มีดังนี้

3.1 กำหนดให้เว็บคราวเลอร์ปฏิเสธยูอาร์แอลที่ไม่ต้องการ เช่น รูปภาพ ไฟล์เสียง หรือไฟล์วีดีโอ เป็นต้น ให้ทำการเงื่อนไขทั้งในส่วนของ decide-rules (ภาพผนวกที่ ข7) และ midfetch-decide-rules (ภาพผนวกที่ ข8) โดยกำหนดชื่อตามที่ต้องการในช่อง Name และเลือก Type เป็น org.archive.crawler.deciderules.MatchesFilePatternDecideRule เพื่อให้สามารถกำหนดรูปแบบของยูอาร์แอลที่ไม่ต้องการได้ จากภาพผนวกที่ ข7 และภาพผนวกที่ ข8 ได้เพิ่มเงื่อนไข 2 เงื่อนไข คือ ให้มีการปฏิเสธยูอาร์แอลที่เป็นไฟล์มัลติมีเดีย (รูปภาพ วีดีโอ เอ็มพี 3) และให้มีการปฏิเสธยูอาร์แอลที่เป็นไฟล์บีบอัดต่างๆ อีกด้วย ซึ่งชื่อของเงื่อนไขที่เพิ่มเข้าไปคือ rejectMultimedaAll และ rejectCompressFiles ตามลำดับ นอกจากนี้ในส่วนของ midfetch-decide-rules สามารถเพิ่มให้

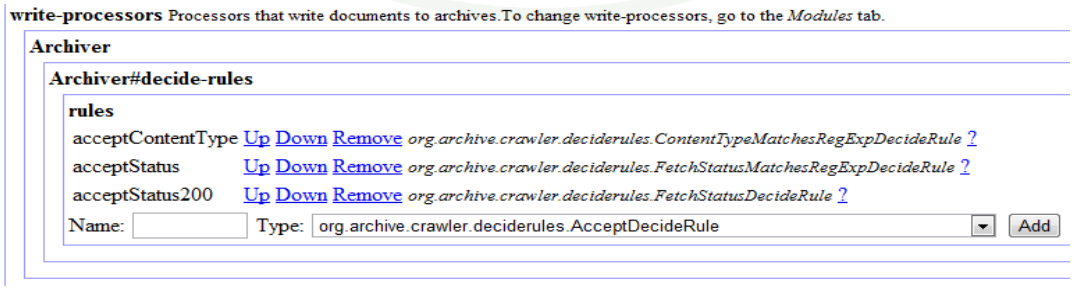
ปฏิเสธเว็บเพจที่มีรหัสสถานะ (HTTP status code) ตามที่ต้องการได้อีกด้วย ซึ่งในตัวอย่างนี้จะปฏิเสธเว็บเพจที่มีรหัสสถานะ เป็น 404 ทั้งหมด ซึ่งจะมีการตั้งค่าในหัวข้อ 4

ภาพผนวกที่ ข7 ซับโมดูลในส่วนของ decide-rules



ภาพผนวกที่ ข8 ซับโมดูลในส่วนของ midfetch-decide-rules

3.2 กำหนดเงื่อนไขการเก็บเว็บเพจลงแฟ้มข้อมูล โดยจะบันทึกข้อมูลก็ต่อเมื่อเงื่อนไขทั้ง 2 เงื่อนไขเป็นจริง ได้แก่ เงื่อนไขที่ 1 ชนิดของเอกสารเป็นเว็บเพจ (text/html) และเงื่อนไขที่ 2 รหัสสถานะ (HTTP status code) ต้องเป็น 200 โดยเมื่อเพิ่มเงื่อนไขในส่วนของ write-processors แล้วจะได้ตามภาพผนวกที่ ข9



ภาพผนวกที่ ข9 ซับโมดูลในส่วนของ write-processors

3.3 คลิก “Settings” ซึ่งอยู่ตำแหน่งล่างสุดของเว็บเพจ เพื่อไปกำหนดรายละเอียดของเว็บคราวเลอร์ทั้งหมด

4. การตั้งค่าของเว็บคราวเลอร์ในส่วนของ Settings ที่จำเป็น มีดังตารางผนวกที่ ข1 จากนั้นเมื่อแก้ไขเสร็จแล้ว ให้คลิกที่ “Submit job” ซึ่งอยู่ตำแหน่งล่างสุดของเว็บเพจ

ตารางผนวกที่ ข1 รายการตั้งค่าเว็บคราวเลอร์ที่จำเป็นในหน้า Settings

ตัวแปร	คำอธิบาย
Meta data (Crawl Operator)	ชื่อผู้ดูแลระบบเว็บคราวเลอร์
Meta data (Crawl Organization)	ชื่อองค์กรของผู้ดูแลระบบเว็บคราวเลอร์
crawl-order (max-toe-threads)	จำนวนเทรดสูงสุดที่สามารถดาวน์โหลดข้อมูลได้พร้อมๆ กัน
scope (max-trans-hops) และ scope (max-speculative-hops)	เป็นการกำหนดให้เว็บคราวเลอร์ตามลิงค์ที่สกัดได้ใหม่หรือไม่ ซึ่งถ้าต้องการให้เป็นเว็บคราวเลอร์แบบแยกเก็บตามไซต์ ต้องกำหนดค่าเท่ากับ 0 ทั้งสองตัวแปร เพื่อให้เก็บเฉพาะเว็บเพจภายใต้เว็บไซต์ที่กำหนดเท่านั้น
scope (max-path-depth)	ค่าความลึกของยูอาร์แอลที่อนุญาต
scope (rejectMultimediaAll)	เพื่อปฏิเสธลิงค์ที่เป็นเอกสารประเภทมัลติมีเดีย โดยจะต้องกำหนด decision เป็น reject และ use-preset-pattern เป็น All ซึ่งมีรายการที่ถูกปฏิเสธดังนี้ รูปภาพ: .bmp, .gif, .jp(e)g, .png, .tif(f) เสียง: .mid, .mp2, .mp3, .mp4, .wav วีดีโอ: .avi, .mov, .mpeg, .ram, .rm, .smil, .wmv เอกสาร: .doc, .pdf, .ppt, .swf
scope (rejectCompressFiles)	เพื่อปฏิเสธลิงค์ที่เป็นเอกสารประเภทบีบอัด และไฟล์อื่นๆ ที่ยังไม่ครอบคลุมในเงื่อนไข rejectMultimediaAll โดยจะต้องกำหนด decision เป็น reject และ use-preset-pattern เป็น Custom และใส่ค่า regexp เป็นดังนี้ .*(?i)(\.(css|js|zip|gz|rar|tar|ico|xls|7z|jar|war|xml| |docx|xlsx|xls|pttx))$

ตารางผนวกที่ ข1 (ต่อ)

ตัวแปร	คำอธิบาย
http-headers(user-agent)	ตั้งชื่อให้กับเว็บคราวเลอร์ โดยจะต้องเปลี่ยน "PROJECT_URL_HERE" เป็นยูอาร์แอลของเว็บไซต์ของผู้ดูแลระบบเว็บคราวเลอร์ ซึ่งถ้าไม่กำหนด หรือใส่ไม่ถูกต้องเว็บคราวเลอร์จะไม่ทำงาน
http-headers (from)	E-mail ติดต่อผู้ดูแลระบบเว็บคราวเลอร์ ซึ่งถ้าไม่กำหนด หรือใส่ไม่ถูกต้องเว็บคราวเลอร์จะไม่ทำงาน
frontier (min-delay-ms)	เป็นการกำหนดระยะเวลาอย่างน้อยที่เว็บคราวเลอร์ต้องรอหลังจากที่ไปเก็บเว็บเพจจากเว็บไซต์นั้นมาแล้ว
frontier (total-bandwidth-usage-KB-sec)	กำหนดแบนด์วิดท์สูงสุดที่ให้ใช้งานได้ ซึ่งถ้าไม่จำกัดแบนด์วิดท์ ก็กำหนดให้เท่ากับ 0
write-processors (prefix)	กำหนดชื่อต้นของแฟ้มข้อมูลที่โปรแกรมบันทึก
write-processors (max-size-bytes)	ขนาดสูงสุดของแฟ้มข้อมูลที่บันทึก
write-processors (path)	Path ที่เก็บไฟล์ที่จะต้องบันทึก

5. สั่งให้เว็บคราวเลอร์เริ่มทำงาน โดยคลิกที่ "Start" ในส่วนของ Crawler status ในหน้าเพจแรก (ภาพผนวกที่ ข3)

วิเคราะห์การทำงาน

โปรแกรม Heritrix อำนวยความสะดวกในการให้คำอธิบายการทำงานของโปรแกรม ทั้งการออกรายงานและแฟ้มลงบันทึก (log files) ที่สามารถเปิดดูได้ขณะที่โปรแกรมกำลังทำงาน และหลังจากที่เว็บคราวเลอร์ทำงานเสร็จแล้ว

1. รายงานค่าสถิติการเก็บเว็บเพจในแต่ละงาน เช่น จำนวนเว็บเพจต่อโฮสต์ จำนวนเว็บเพจแยกตามรหัสสถานะ (HTTP code status) ฯลฯ ซึ่งผู้ใช้สามารถเรียกดูรายงานต่างๆ ได้จากแท็บ "Report"

2. แฟ้มลงบันทึก (log files) มีรูปแบบการจัดเก็บเหตุการณ์และข้อผิดพลาดต่างๆ ทั้งหมด 5 ชนิด โดยที่ผู้ใช้สามารถเข้าดูรายละเอียดของแฟ้มลงบันทึกได้จากแท็บที่ 4 (Logs) ตามภาพผนวกที่ ข3

2.1 crawl.log จะมีบันทึกทุกเหตุการณ์ที่เว็บคราวเลอร์ร้องขอข้อมูลไปยังเว็บเซิร์ฟเวอร์ โดยตัวอย่างการเก็บบันทึกแสดงดังภาพผนวกที่ ข10 ซึ่งมีการเก็บข้อมูลทั้งหมด 10 เขตข้อมูล (field) ได้แก่ ตราเวลา (timestamp) ณ ตอนที่เกิดเหตุการณ์ในรูปแบบ ISO8601 สถานะของดึงข้อมูล (ตารางผนวกที่ ข2) ขนาดของไฟล์ที่เก็บมาในหน่วนไบต์ ยูอาร์แอลของเอกสารที่เก็บมา รหัสอ้างอิงแหล่งที่มาของเอกสาร (ตารางผนวกที่ ข3) ยูอาร์แอลของเอกสารต้นทาง ประเภทของเอกสารที่เก็บมา รหัสเทรด (Tread ID) ที่ไปเก็บเอกสาร ตราเวลา (timestamp) ที่เริ่มติดต่อเครื่องแม่ข่ายของเอกสารซึ่งถ้าสำเร็จจะระบุเวลาที่ใช้ในการรับส่งข้อมูลในหน่วยมิลลิวินาทีไว้หลังเครื่องหมาย + และเข้ารหัสแบบ SHA1 กับเนื้อหาของเอกสารที่เก็บมา ตามลำดับ

```
2004-07-21T23:29:40.438Z   200        310
http://127.0.0.1:9999/selftest/Charset/charsetselftest_end.html LLLL
http://127.0.0.1:9999/selftest/Charset/shiftjis.jsp text/html #000 20040721232940401+10
M77KNTBZH2IU6V2SIG5EEG45EJICNQNM -
```

ภาพผนวกที่ ข10 แสดงตัวอย่างการเก็บบันทึกในแฟ้ม crawl.log

2.2 local-errors.log บันทึกข้อมูลเกี่ยวกับข้อผิดพลาดต่างๆ ในขณะที่เว็บคราวเลอร์กำลังทำงาน ซึ่งโดยปกติจะเป็นเรื่องเกี่ยวกับระบบเครือข่าย เช่น ปัญหาในการรับข้อมูลจากเว็บเซิร์ฟเวอร์ปลายทาง

2.3 progress-statistocs.log บันทึกค่าทางสถิติต่างๆ ของเว็บคราวเลอร์ขณะทำงาน โดยแบ่งเป็นคอลัมน์ต่างๆ ดังนี้ ตราเวลา (timestamp) ณ ตอนที่เกิดเหตุการณ์ในรูปแบบ ISO8601 จำนวนของยูอาร์แอลที่สกัดพบ จำนวนยูอาร์แอลในคิว ณ ขณะนั้น จำนวนเอกสารที่เก็บมาแล้ว ค่าเฉลี่ยจำนวนการเก็บเอกสารต่อวินาทีตั้งแต่เริ่มทำงาน แบนด์วิดท์เฉลี่ยตั้งแต่เริ่มทำงานในหน่วยของกิโลบิตต่อวินาที จำนวนเอกสารที่ไม่สามารถเก็บมาได้ จำนวนเทรดที่ทำงาน ณ ขณะนั้น จำนวนหน่วนความจำที่ใช้ ณ ขณะนั้นในหน่วยของกิโลบิต

2.4 runtime-error.log บันทึกข้อบกพร่องที่เกิดระหว่างการทำงานที่เกิดขึ้นจากการทำงานของโปรแกรม เช่น ปัญหาหน่วยความจำไม่เพียงพอ ปัญหาที่เกิดจากข้อจำกัดของอุปกรณ์ฮาร์ดแวร์

2.5 uri-errors.log บันทึกการตรวจพบยูอาร์แอลที่ผิดรูปแบบ

ตารางผนวกที่ ข2 แสดงรายการสถานะการดึงข้อมูลที่กำหนดโดยโปรแกรม Heritrix

รหัสสถานะ	คำอธิบาย
1	Successful DNS lookup
0	Fetch never tried (perhaps protocol unsupported or illegal URI)
-1	DNS lookup failed
-2	HTTP connect failed
-3	HTTP connect broken
-4	HTTP timeout (before any meaningful response received)
-5	Unexpected runtime exception; see runtime-errors.log
-6	Prerequisite domain-lookup failed, precluding fetch attempt
-7	URI recognized as unsupported or illegal
-8	Multiple retries all failed, retry limit reached
-50	Temporary status assigned URIs awaiting preconditions; appearance in logs may be a bug
-60	Failure status assigned URIs which could not be queued by the Frontier (and may in fact be unfetchable)
-61	Prerequisite robots.txt-fetch failed, precluding a fetch attempt
-62	Some other prerequisite failed, precluding a fetch attempt
-63	A prerequisite (of any type) could not be scheduled, precluding a fetch attempt
-3000	Severe Java 'Error' conditions (OutOfMemoryError, StackOverflowError, etc.) during URI processing.
-4000	'chaff' detection of traps/content of negligible value applied
-4001	Too many link hops away from seed
-4002	Too many embed/transitive hops away from last URI in scope
-5000	Out of scope upon reexamination (only happens if scope changes during crawl)
-5001	Blocked from fetch by user setting
-5002	Blocked by a custom processor
-5003	Blocked due to exceeding an established quota

ตารางผนวกที่ ข2 (ต่อ)

รหัสสถานะ	คำอธิบาย
-5004	Blocked due to exceeding an established runtime
-6000	Deleted from Frontier by user
-7000	Processing thread was killed by the operator (perhaps because of a hung condition)
-9998	Robots.txt rules precluded fetch

ตารางผนวกที่ ข3 แสดงรหัสอ้างอิงแหล่งที่มาของเอกสารที่ใช้ในโปรแกรม Heritrix

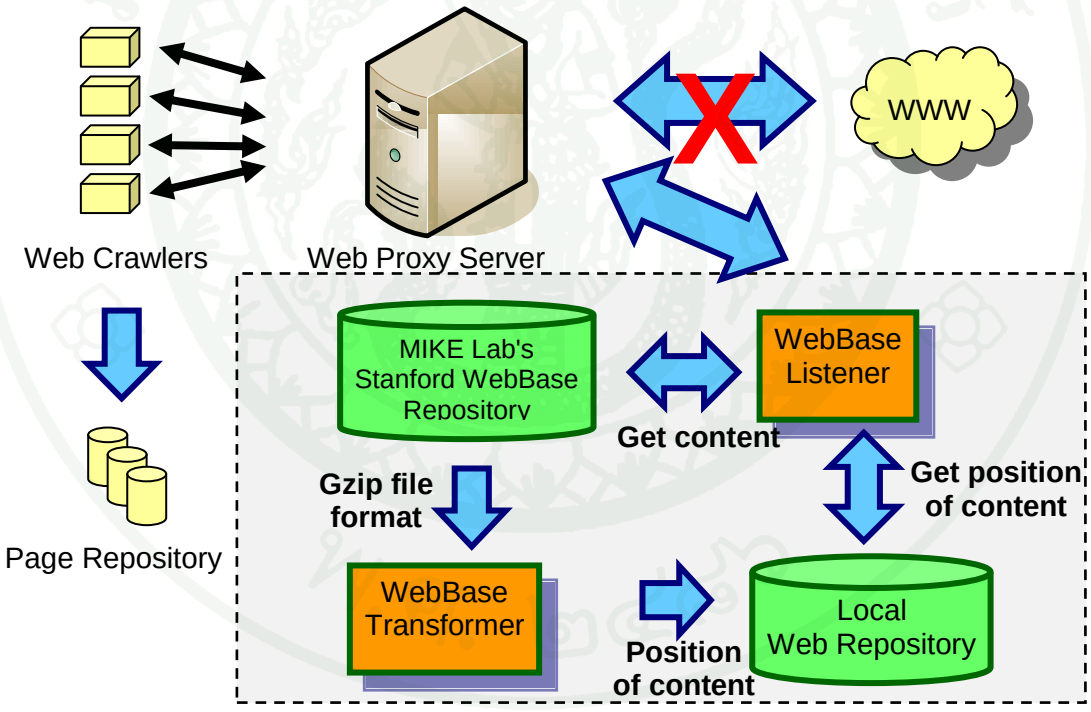
รหัสอ้างอิงแหล่งที่มา	คำอธิบาย
R	Redirect
E	Embed
X	Speculative embed (aggressive/Javascript link extraction)
L	Link
P	Prerequisite (as for DNS or robots.txt before another URI)

ภาคผนวก ค

โปรแกรมเว็บพร็อกซี่สำหรับฐานข้อมูลเว็บเบสสแตนฟอร์ด

โปรแกรมเว็บพร็อกซี่สำหรับฐานข้อมูลเว็บเบสสแตนฟอร์ด

การออกแบบ และทดสอบอัลกอลิทึมของเว็บคราวเลอร์นั้น นอกจากมีความท้าทายทางด้านวิศวกรรมคอมพิวเตอร์แล้ว ยังต้องการการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตอีกด้วย ซึ่งจะทำให้การทดสอบประสิทธิภาพของอัลกอริทึม หรือสถาปัตยกรรมของเว็บคราวเลอร์ที่ออกแบบต้องใช้เวลานาน สิ้นเปลืองแบนด์วิดช์ (Bandwidth) และรบกวนเครื่องแม่ข่ายปลายทางอีกด้วย ดังนั้นเราจึงนำเสนอการออกแบบเว็บพร็อกซี่จำลอง (Web proxy simulator) โดยใช้ข้อมูลเว็บเพจจากฐานข้อมูลเว็บเบสของมหาวิทยาลัยสแตนฟอร์ด (Stanford WebBase) เป็นข้อมูลนำเข้า ทำให้นักออกแบบเว็บคราวเลอร์สามารถทดสอบประสิทธิภาพการทำงานของเว็บคราวเลอร์ได้โดยไม่ต้องเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตอีกต่อไป ลดภาระการรบกวนเครื่องแม่ข่ายที่ถูกทดสอบ อีกทั้งยังจะช่วยหลีกเลี่ยงปัญหาเรื่องช่องว่างของเวลา (time gap) ของเว็บเพจได้อีกด้วย



ภาพผนวกที่ ค1 โครงสร้างของเว็บพร็อกซี่เพื่อใช้งานฐานข้อมูลเว็บเบสจากมหาวิทยาลัยสแตนฟอร์ด

วิธีการ

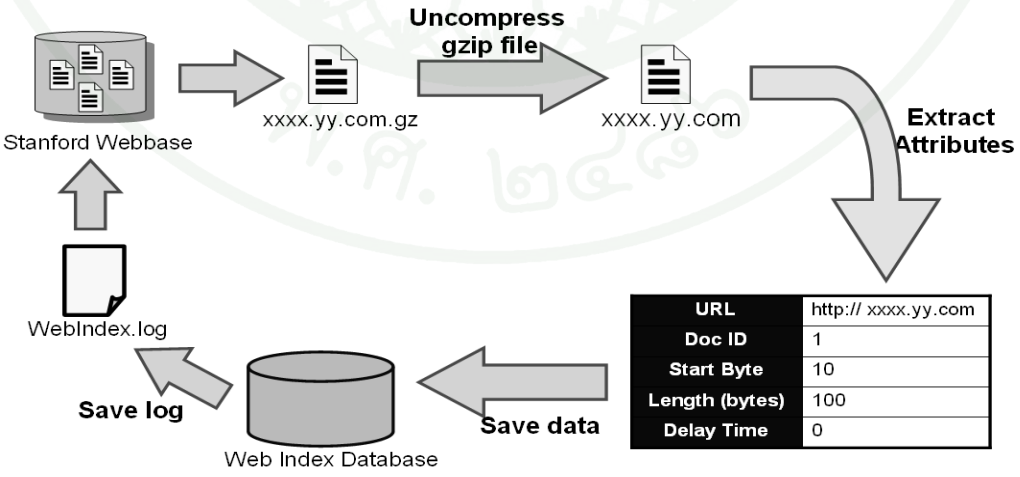
การสร้างเว็บพร็อกซี่เพื่อใช้งานฐานข้อมูลเว็บเบสจากมหาวิทยาลัยสแตนฟอร์ด มีส่วนประกอบหลักๆ ดังError! Reference source not found.ค2 และมีรายละเอียดดังนี้

1. WebBase Listener ทำหน้าที่ตอบสนองการร้อขอข้อมูลจากเว็บคราวเลอร์

2. Local Web Repository เป็นฐานข้อมูลดรรชนีเว็บเพจที่มีทั้งหมด ประกอบไปด้วย

   - URL: ยูอาร์แอลของเว็บเพจ
   - Doc ID: รหัสของเว็บเพจ
   - Start Byte: ตำแหน่งของ byte เริ่มต้นของเนื้อหาเว็บเพจนั้น
   - Length: ขนาดของเนื้อหาเว็บเพจ
   - Delay Time(ms): ค่าหน่วงเวลาของเว็บเซิร์ฟเวอร์
   - Percent Thai: ระดับภาษาไทยของเว็บเพจ

3. WebBase Transformer มีหน้าที่สร้างดรรชนี จากข้อมูลเว็บเพจของมหาวิทยาลัยสแตนฟอร์ด แล้วนำดรรชนีที่ได้มาเก็บไว้ที่ local Web Repository โดยมีขั้นตอนการทำงานตามภาพผนวกที่ ค2



ภาพผนวกที่ ค2 แผนภาพแสดงการทำงานของ WebBase Transformer

4. MIKE Lab's Stanford WebBase Repository เป็นฐานข้อมูลเว็บเพจจากมหาวิทยาลัยสแตนฟอร์ด

การทดสอบ

เริ่มต้นโดยการให้เว็บพร็อกซี่พร้อมทำงาน และกำหนดให้มีจำนวนเทรดที่รอรับคำร้อง (request) เท่ากับ 100 เทรด จากนั้นการทดสอบจะใช้โปรแกรม Apache Benchmark (ab) จำลองการร้องขอซึ่งมีวิธีการใช้งาน ดังภาพผนวกที่ ค3 โดยที่

ab คือคำสั่งเรียกใช้งานโปรแกรม Apache Benchmark

-n คือ จำนวนคำร้อง (request) ที่จะส่งไปให้กับเว็บเซิร์ฟเวอร์ปลายทาง

-c คือ จำนวนคำร้อง (request) ที่จะส่งไปให้กับเว็บเซิร์ฟเวอร์ปลายทางพร้อมๆกัน (concurrent)

-g คือ กำหนดแฟ้มข้อมูลที่จะให้เขียนผลการทดสอบลงไป

-X คือ เว็บเซิร์ฟเวอร์ปลายทาง หรือเว็บพร็อกซี่ที่ใช้ทดสอบ

URL คือ ยูอาร์แอลที่ต้องการร้องข้อให้เว็บเซิร์ฟเวอร์ปลายทางส่งข้อมูลมาให้

```
ab -n 100 -c 5 -g c:\result\ab_n100_c5.txt -X csl.cpe.ku.ac.th:8081 http://cpe.ku.ac.th/
```

ภาพผนวกที่ ค3 ตัวอย่างการใช้งานโปรแกรม Apache Benchmark

ผล

ผลการทดสอบด้วยโปรแกรม Apache Benchmark แสดงดังตารางผนวกที่ ค1 โดยการทดลองจะเริ่มต้นส่งจำนวนคำร้องขอที่ 100 คำร้อง และสิ้นสุดที่ 5,000 คำร้อง ซึ่งจากผลการทดสอบ สรุปได้ว่า ควรกำหนดจำนวนเทรดของเว็บพร็อกซี่ให้สอดคล้องกับจำนวนของคำร้องขอ (request) ที่เข้ามาพร้อมกัน ซึ่งในการทดลองนี้ได้กำหนดให้เว็บพร็อกซี่มีจำนวนเทรดที่รอรับคำร้อง (request) เท่ากับ 100 เทรด

ตารางผนวกที่ ค1 ผลการทดสอบประสิทธิภาพของเว็บพร็อกซี่ด้วยโปรแกรม Apache Benchmark

No.	# requests	# concurrency requests	transfer rate (Kbytes/sec)	time (s)
1	100	50	134.11	13.27
2	100	100	173.56	10.25
3	500	50	160.90	55.30
4	500	150	152.49	58.34
5	500	250	146.38	60.78
6	500	500	144.81	61.44
7	1,000	50	109.73	162.16
8	1,000	150	123.45	144.14
9	1,000	250	121.25	146.75
10	1,000	500	124.01	143.48
11	5,000	50	46.54	1,911.63
12	5,000	150	46.06	1,931.80
13	5,000	250	43.98	2,023.14

วิธีการใช้งาน

1. ปรับแต่งค่าเริ่มต้นของโปรแกรม โดยแก้ไขไฟล์ configuration.properties ซึ่งประกอบด้วย 4 ส่วน ดังนี้

- webbase.path ระบุตำแหน่งของฐานข้อมูลเว็บ
- webbase.db.path ตำแหน่งของฐานข้อมูลดรรชนีเว็บเพจที่สร้างขึ้นใหม่
- proxy.port พอร์ตของเว็บพร็อกซี่
- proxy.maxThread กำหนดให้พร็อกซี่ทำงานพร้อมกันกี่เทรด

ตัวอย่างเช่น

webbase.path = /mike/data/csl_web_pub/pub/webbase/th/crawl_09052009_l3/

webbase.db.path = /home/punnawat/pun_spider/proxy_index/crawl_09052009_l3/

proxy.port = 8081

proxy.maxThread = 150

2. การนำข้อมูลจากฐานข้อมูลเว็บเบสจากมหาวิทยาลัยสแตนฟอร์ด มาสร้างดรรชนีสำหรับเว็บพร็อกซี่ ใช้คำสั่งดังนี้

java -Xmx512M -jar proxy.jar import

3. การสั่งให้เว็บพร็อกซี่เริ่มทำงาน ใช้คำสั่งดังนี้

java -Xmx512M -jar proxy.jar start&

4. การใช้งานเว็บพร็อกซี่ เครื่องปลายทาง ต้อง set proxy มาที่ csl.cpe.ku.ac.th และใช้ port เป็น 8081 หรืออื่นๆ ตามที่กำหนดไว้ในไฟล์ configuration.properties

ประวัติการศึกษาและการทำงาน

ชื่อ	นายปุณณวัฒน์ ธาดาภาคย์
เกิดวันที่	5 ธันวาคม 2527
สถานที่เกิด	อำเภอเมือง จังหวัดพิษณุโลก
ประวัติการศึกษา	วศ.บ. (วิศวกรรมคอมพิวเตอร์) มหาวิทยาลัยนเรศวร
ตำแหน่งปัจจุบัน	วิศวกร ระดับ 4
สถานที่ทำงานปัจจุบัน	การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย
ผลงานดีเด่นและ/หรือรางวัลทางวิชาการ	-
ทุนการศึกษาที่ได้รับ	-