ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)

ปริญญา

วิศวกรรมคอมพิวเตอร์	วิศวกรรมคอมพิวเตอร์
สาขา	**ภาควิชา**

เรื่อง การเก็บเว็บเพจแบบเฉพาะเจาะจงภาษาไทยโดยใช้ฐานความรู้

Knowledge Based Thai Language-Specific Web Crawling

นามผู้วิจัย นายเอกสิทธิ์ ศรีสุขะ

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก .....

(ผู้ช่วยศาสตราจารย์อานนท์ รุ่งสว่าง, Ph.D.)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม .....

(รองศาสตราจารย์สุรศักดิ์ สงวนพงษ์, M. Eng.)

หัวหน้าภาควิชา .....

(ผู้ช่วยศาสตราจารย์เขมะฑัต วิภาตะวนิช, Ph.D.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

.....

(รองศาสตราจารย์กัญจนา ธีระกุล, D.Agr.)

คณบดีบัณฑิตวิทยาลัย

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

วิทยานิพนธ์

เรื่อง

การเก็บเว็บเพจแบบเฉพาะเจาะจงภาษาไทยโดยใช้ฐานความรู้

Knowledge Based Thai Language-Specific Web Crawling

โดย

นายเอกสิทธิ์ ศรีสุขะ

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)

พ.ศ. 2552

เอกสิทธิ์ ศรีสุขะ 2552: การเก็บเว็บเพจแบบเฉพาะเจาะจงภาษาไทยโดยใช้ฐานความรู้ ปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์) สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ ประธานกรรมการที่ปรึกษา: ผู้ช่วยศาสตราจารย์อานนท์ รุ่งสว่าง, Ph.D. 90 หน้า

เว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาคือเครื่องมือที่ใช้สำหรับเก็บรวบรวมเว็บเพจที่เขียนขึ้นจากภาษาที่ผู้ใช้กำหนด สำหรับการเก็บรวบรวมเว็บเพจภาษาไทยนั้น วิธีที่ง่ายที่สุดคือกำหนดให้เว็บคราวเลอร์เก็บรวบรวมเว็บเพจตามอินเตอร์เน็ตโดเมนท้องถิ่นประจำประเทศไทย เช่น .th เป็นต้น อย่างไรก็ตามปัญหาหลักคือเว็บเพจภาษาไทยส่วนใหญ่จำนวนมากกว่าครึ่งหนึ่งอยู่ภายนอกขอบเขตของโดเมนดังกล่าว ดังนั้นวิธีการนี้จึงยังไม่ครอบคลุมถึงปริมาณเว็บเพจภาษาไทยส่วนใหญ่ที่มีอยู่ทั้งหมดได้ วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการค้นหาเว็บเพจภาษาไทยซึ่งจะคำนวณเพื่อหาค่าความน่าจะเป็นที่ลิงค์ถัดไปจะเป็นเว็บเพจภาษาไทยโดยใช้หลักทฤษฎีความน่าจะเป็นแบบเบส์อย่างง่ายมาคำนวณกับฐานความรู้ที่ได้จากการเก็บรวบรวมเว็บเพจมาในครั้งแรก คะแนนความน่าจะเป็นดังกล่าวจะถูกนำมาใช้ในการจัดลำดับของยูอาร์แอลในยูอาร์แอลคิวเพื่อให้เว็บคราวเลอร์เลือกเก็บเว็บเพจจากยูอาร์แอลที่มีคะแนนสูงสุดก่อนนั่นเอง จากผลการทดลองพบว่าวิธีการที่นำเสนอดังกล่าวให้ผลในด้านการเก็บเกี่ยวและความครอบคลุมได้ดีกว่าวิธีการอื่น

_____ _____ ___ / ___ / ___

ลายมือชื่อนิสิต ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

Ekkasit Srisukha 2009: Knowledge Based Thai Language-Specific Web Crawling. Master of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Assistant Professor Arnon Rungsawang, Ph.D. 90 pages.

Language-specific web crawler is a tool used for gathering web pages that is written in a specific language. In case of gathering the Thai web pages, the simplest way is to setup a web crawler to follow only web pages according to a national domain name restriction, e.g. by specifying the ".th" domain. However, the main problem of this method is that more than half of Thai web pages are outside the ".th" domain. Therefore, the web crawler still misses many Thai web pages. This thesis proposed a language-specific web page finding method which calculates the probability scores of next links that are likely to find Thai's web pages, by using a Naïve Bayes theory to compute with the knowledge base that has been built from the first crawling. The output probability scores are then used to reorder the URLs in the crawler's priority queue, so that, the web crawler will collect the highest probability scores web pages first. According to the evaluation results, the proposed strategy achieves better harvest rate and crawling coverage than the other approaches proposed in the literature.

_____ _____ ___ / ___ / ___

Student's signature Thesis Advisor's signature

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. อานนท์ รุ่งสว่าง ประธานกรรมการที่ปรึกษาที่ได้ปลูกฝังระเบียบวิธีวิจัยเพื่อให้ข้าพเจ้าได้คิดอย่างเป็นระบบและมีแบบแผนที่ดี รวมไปถึงได้ให้ความรู้และคำแนะนำที่มีประโยชน์เป็นอย่างยิ่งต่อการทำวิทยานิพนธ์ฉบับนี้ ขอขอบพระคุณ ดร. ชูชาติ หฤไชยะศักดิ์ อาจารย์ที่ปรึกษาร่วมที่สละเวลาอันมีค่าของท่านเพื่อรับฟังรายงานความก้าวหน้า และให้ความรู้ คำปรึกษาตลอดจนชี้แนะข้อผิดพลาดที่เกิดขึ้นจนทำให้วิทยานิพนธ์นี้สำเร็จไปได้ด้วยดี และขอขอบพระคุณรองศาสตราจารย์สุรศักดิ์ สงวนพงษ์ กรรมการที่ปรึกษาวิชารองที่กรุณาให้คำปรึกษาเกี่ยวกับการนำเสนองานวิจัย รวมไปถึงให้คำถามที่เป็นประโยชน์ต่อการปรับปรุงวิทยาพิพนธ์ฉบับนี้

ขอขอบคุณ คุณสุภัคพงศ์ จินารัตน์ที่ได้ให้คำปรึกษาที่ดีมาโดยตลอดทั้งในด้านของการถ่ายทอดความรู้และประสบการณ์ที่พบในการทำวิจัย และขอบคุณสมาชิกห้องปฏิบัติการวิจัยวิศวกรรมข้อมูลและฐานความรู้ขนาดใหญ่ (MIKE) ทุกท่านที่สร้างความสุขด้วยเสียงหัวเราะและให้ความช่วยเหลือด้วยดีเสมอมา

ขอบคุณเจ้าหน้าที่ธุรการโครงการปริญญาโทที่ช่วยเหลือในด้านการประสานงานต่าง ๆ ให้งานสำเร็จลุล่วงไปด้วยดี และขอบคุณกำลังใจจากพี่น้องและเพื่อน ๆ ทุกท่านที่ทำให้ข้าพเจ้าทำงานจนสำเร็จลงได้

สุดท้ายนี้ขอขอบคุณสถาบันบัณฑิตวิทยาศาสตร์และเทคโนโลยีไทย (TGIST) ที่ได้มอบทุนวิจัยแก่ข้าพเจ้าเพื่อจัดซื้อวัสดุ-อุปกรณ์ที่จำเป็นในการทำวิทยานิพนธ์นี้ คุณงามความดีหรือประโยชน์อันใดที่เกิดจากวิทยาพินธ์ฉบับนี้ ข้าพเจ้าขออุทิศให้บิดา มารดา บุพการี คณาจารย์และผู้มีพระคุณทุกท่าน

เอกสิทธิ์ ศรีสุขะ

สิงหาคม 2552

สารบัญ

สารบัญตาราง

สารบัญตาราง (ต่อ)

ตารางผนวกที่

สารบัญภาพ

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

ภาพผนวกที่

การเก็บเว็บเพจแบบเฉพาะเจาะจงภาษาไทยโดยใช้ฐานความรู้

Knowledge Based Thai Language-Specific Web Crawling

คำนำ

ปัจจุบันข้อมูลข่าวสารที่เผยแพร่ผ่านเว็บเพจบนอินเทอร์เน็ตมีจำนวนมหาศาลและมีปริมาณเพิ่มขึ้นอย่างรวดเร็วตลอดเวลา ดังนั้นการเก็บรวบรวมเว็บเพจทั้งหมดเพื่อให้ผลลัพธ์ที่ได้จากการสืบค้นข้อมูลมีความครอบคลุมกับความต้องการของผู้ใช้ล้วนแล้วแต่เป็นสิ่งที่ทำได้อย่างยากยิ่ง ทั้งยังพบว่าเว็บเพจที่ได้เก็บมาแล้วจำนวนมากยังมีเนื้อหาที่ไม่ทันสมัยและข้อมูลที่ได้รับมาไม่มีประโยชน์ เป็นการสิ้นเปลืองเวลาและสูญเสียทรัพยากรคอมพิวเตอร์โดยไม่คุ้มค่า ด้วยเหตุนี้นักวิจัยแต่ละประเทศจึงมุ่งเน้นพัฒนาวิธีการเก็บรวบรวมเว็บเพจส่วนหนึ่งที่ตรงกับภาษาประจำชาติออกมาจากอินเทอร์เน็ต เพื่อนำเว็บเพจที่เก็บมาได้สร้างเป็นระบบคลังข้อมูลเว็บเพจขนาดใหญ่สำหรับภาษานั้น ห้องสมุดดิจิตอล หรือระบบสืบค้นข้อมูลเฉพาะภาษา โดยมุ่งหวังให้ผลลัพธ์จากการสืบค้นข้อมูลมีความครอบคลุมและตรงกับความต้องการของผู้ใช้ในประเทศนั้นมากยิ่งขึ้น

เทคโนโลยีหลักที่นำมาใช้ในการเก็บรวบรวมเว็บเพจตามภาษาเรียกว่า เว็บคราวเลอร์แบบเฉพาะเจาะจงภาษา (Somboonviwat *et al.,* 2005) ซึ่งจะทำหน้าที่เก็บรวบรวมเว็บเพจตามภาษาที่ผู้ใช้กำหนด โดยเริ่มเก็บเว็บเพจจากกลุ่มเว็บเพจเริ่มแรกและเดินทางไปตามลิงค์เพื่อเก็บเว็บเพจอื่นๆในภาษานั้นต่อไป

วิธีการพื้นฐานที่เว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาเลือกใช้ คือการกำหนดขอบเขตการค้นหาตามระบบอินเตอร์เน็ตโดเมนระดับบนสุดประจำประเทศนั้น เช่น เมื่อต้องการเก็บรวบรวมเว็บเพจภาษาไทยก็ให้ทำการค้นหาที่โดเมน ".th" เป็นต้น แต่จากงานวิจัยของ Somboonviwat และคณะ (Somboonviwat *et al.,* 2005) พบว่าเว็บเพจภาษาไทยส่วนใหญ่จำนวนมากกว่าครึ่งหนึ่งอยู่ภายนอกขอบเขตของโดเมนดังกล่าว ดังนั้นวิธีการนี้จึงยังไม่ครอบคลุมถึงปริมาณเว็บเพจภาษาไทยส่วนใหญ่ที่มีอยู่ทั้งหมดได้

วิทยานิพนธ์นี้จะนำเสนอวิธีการค้นหาเว็บเพจภาษาไทย นอกเหนือจากขอบเขตการค้นหาตามอินเทอร์เน็ตโดเมน โดยจะคำนวณหาค่าความน่าจะเป็นที่ลิงค์ถัดไปจะเป็นเว็บเพจภาษาไทยด้วยการนำหลักทฤษฎีความน่าจะเป็นแบบเบส์อย่างง่ายมาคำนวณหาค่าความน่าจะเป็นกับฐานความรู้ที่ถูกสร้างมาแล้วจากการสกัดฟีเจอร์ที่บ่งบอกถึงลักษณะของเว็บเพจภาษาไทยจากกลุ่มของเว็บเพจที่ได้เก็บมาแล้วในครั้งแรก ค่าความน่าจะเป็นนี้ถ้าปรากฏในลิงค์ใดและมีค่ามากจะหมายความว่ามีความเป็นไปได้สูงมากที่ลิงค์นั้นจะเป็นเว็บเพจภาษาไทย หลังจากนั้นเราจะใช้ค่าคะแนนดังกล่าวมาใช้ในการจัดลำดับของยูอาร์แอลในยูอาร์แอลคิวเพื่อให้เว็บคราวเลอร์เลือกเก็บเฉพาะเว็บเพจจากยูอาร์แอลที่มีคะแนนสูงสุดก่อนนั่นเอง

วัตถุประสงค์

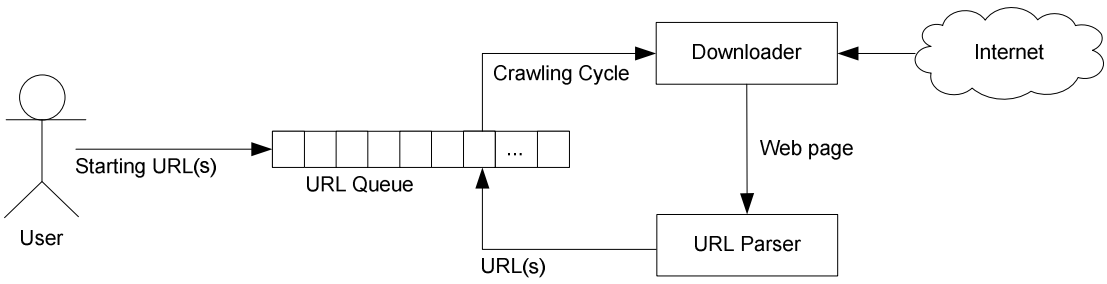
พัฒนาวิธีการค้นหาเว็บเพจภาษาไทยให้ได้มากที่สุดอย่างมีประสิทธิภาพโดยใช้หลักความน่าจะเป็นแบบเบส์อย่างง่ายมาคำนวณกับฐานความรู้เพื่อหาค่าความน่าจะเป็นที่ลิงค์ถัดไปใด ๆ จะเป็นเว็บเพจภาษาไทย

การตรวจเอกสาร

ใจความสำคัญของบทนี้จะกล่าวถึงทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้องกับเว็บคราวเลอร์ อันได้แก่ แบบจำลองรายละเอียดแต่ละส่วนประกอบของเว็บคราวเลอร์ การเก็บเว็บเพจแบบเฉพาะเจาะจงหัวเรื่อง วิธีการวัดประสิทธิภาพและหน่วยวัดที่ใช้ การค้นหาเว็บเพจแบบเฉพาะเจาะจงภาษา เป็นต้น และในหัวข้อสุดท้ายจะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษา

ความรู้เบื้องต้นเกี่ยวกับเว็บคราวเลอร์

เว็บคราวเลอร์ (Web crawler) คือ โปรแกรมที่พัฒนาขึ้นมาเพื่อใช้ประโยชน์ในการเก็บรวบรวมเว็บเพจจากอินเทอร์เน็ต เว็บคราวเลอร์ตัวแรกถูกพัฒนาขึ้นในปี 1993 โดย Gray (1993) จนถึงปัจจุบันมีงานวิจัยอยู่มากมายต่างมุ่งเน้นที่จะออกแบบพัฒนาให้เว็บคราวเลอร์มีประสิทธิภาพสูงสุด อย่างไรก็ตามการออกแบบและหลักการทำงานยังคงยึดอยู่บนพื้นฐานเดียวกันและถูกปรับปรุงเปลี่ยนแปลงมาจากแบบจำลองเดียวกัน สำหรับในหัวข้อนี้จะอธิบายถึงแบบจำลองพื้นฐานของเว็บคราวเลอร์

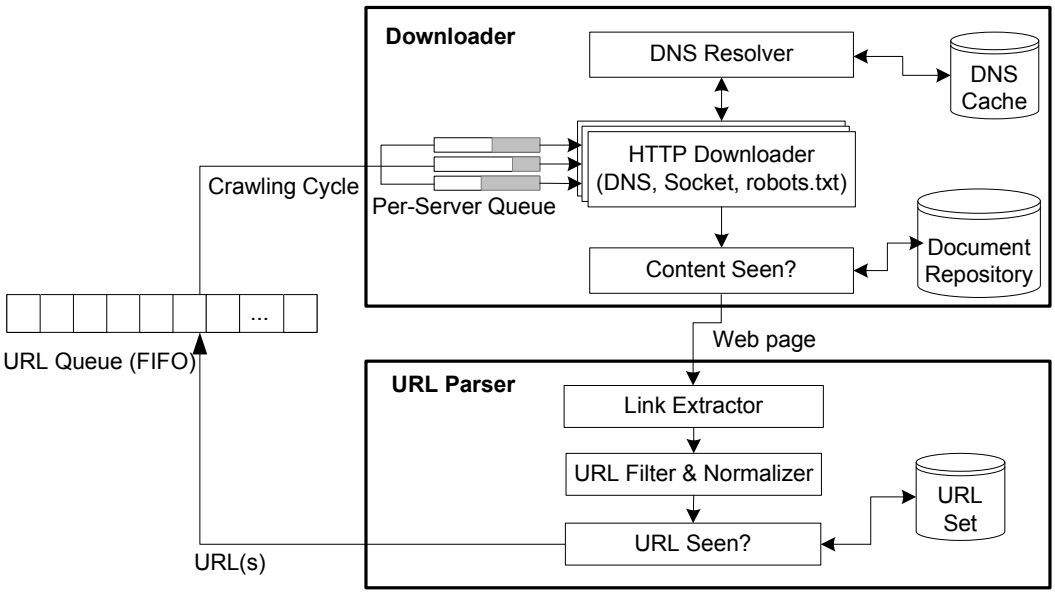


ภาพที่ 1 แบบจำลองพื้นฐานของเว็บคราวเลอร์

เว็บคราวเลอร์เป็นโปรแกรมที่เก็บรวมรวมเว็บเพจได้โดยอัตโนมัติ โดยเริ่มเก็บเว็บเพจจากยูอาร์แอลเริ่มต้น (Seed URLs) ที่ผู้ใช้กำหนดโดยใช้ตัวเก็บเว็บเพจ (Downloader) จากนั้นตัวคัดแยกยูอาร์แอล (URL parser) จะทำการคัดแยกยูอาร์แอลใหม่ออกจากเว็บเพจที่ดาวน์โหลดมาแล้วโดยจะนำยูอาร์แอลใหม่ที่ได้ไปตรวจสอบว่าก่อนหน้านี้เคยเก็บเว็บเพจจากยูอาร์แอลนี้มาก่อนหรือไม่ หากพบว่าเคยเก็บมาแล้วจะไม่เก็บเว็บเพจนั้นซ้ำซ้อนอีก แต่หากพบว่าไม่เคยเก็บมาก่อนตัวคัดแยกยูอาร์

แอลจะนำยูอาร์แอลดังกล่าวใส่เข้าไปในยูอาร์แอลคิว (URL queue) เพื่อเป็นลำดับรายการของยูอาร์แอลในทำการเก็บเว็บเพจในรอบถัดไป ขั้นตอนการทำงานเหล่านี้จะทำงานเป็นวงรอบ (Crawling cycle) จนกว่าจะสิ้นสุดการทำงานซึ่งเกิดขึ้นได้ 2 กรณีคือ เว็บคราวเลอร์เก็บเว็บเพจได้ครบตามจำนวนที่กำหนด หรือไม่พบยูอาร์แอลที่สามารถเก็บต่อไปได้ รายละเอียดขั้นตอนดังกล่าวแสดงในแบบจำลองดังภาพที่ 1

แบบจำลองพื้นฐานที่กล่าวมาเป็นเพียงแนวทางการออกแบบเว็บคราวเลอร์และการทำงานในเบื้องต้นเท่านั้น อย่างไรก็ตามเมื่อพิจารณาแต่ละส่วนประกอบล้วนแต่มีรายละเอียดในเชิงเทคนิคอยู่มากมาย ในหัวข้อถัดไปจะอธิบายรายละเอียดในแต่ละส่วนประกอบ รวมไปถึงชี้แนะแนวทางในการเพิ่มประสิทธิภาพที่ได้รวบรวมมาจากงานวิจัยที่เกี่ยวข้อง ตามแบบจำลองดังแสดงในภาพที่ 2



ภาพที่ 2 แบบจำลองรายละเอียดเชิงเทคนิคในแต่ละส่วนประกอบของเว็บคราวเลอร์

1. ยูอาร์แอลคิว (URL Queue)

ยูอาร์แอลคิวทำหน้าที่เป็นแหล่งที่เก็บยูอาร์แอลที่เว็บคราวเลอร์สกัดมาได้จากในเว็บเพจแต่ยังไม่ได้ทำการเก็บรวบรวมมา โดยยูอาร์แอลคิวในแบบจำลองพื้นฐานนี้จะเป็นแบบเข้าก่อนออกก่อน (FIFO: First-In First-Out) กล่าวคือ เว็บคราวเลอร์จะทำเก็บเว็บเพจตามลำดับยูอาร์แอลที่เข้ามาก่อน (Breadth-First search)

ยูอาร์แอลคิวโดยทั่วไปมักสร้างในหน่วยความจำชั่วคราวเพื่อให้ความเร็วในการเข้าถึงข้อมูลมีสูง อย่างไรก็ตามเนื่องจากปริมาณความจุของหน่วยความจำมีอยู่อย่างจำกัด ไม่เพียงพอต่อจำนวนยูอาร์แอลที่มีปริมาณมหาศาลได้ จึงมีงานวิจัยส่วนหนึ่งที่มุ่งเน้นวิธีการออกแบบยูอาร์แอลคิว ยกตัวอย่างเช่น Heydon and Najork (1999) ได้สร้างยูอาร์แอลคิวโดยใช้พื้นที่ของดิสก์สำหรับเก็บยูอาร์แอลที่ใช้ไม่บ่อยนัก ส่วนยูอาร์แอลที่ถูกใช้บ่อยจะมีการสลับมาไว้ในหน่วยความจำชั่วคราว ด้วยวิธีการดังกล่าวทำให้ระบบสามารถรองรับปริมาณยูอาร์แอลได้เป็นอย่างมากและยังคงมีความเร็วสูงในการเข้าถึงข้อมูล สำหรับงานวิจัยของ Koht-Arsa and Sanguanpong (2001) ได้นำโครงสร้างข้อมูลเอวีแอลทรี (AVL Tree) (Sedgewick and Flajolet, 1995) มาเก็บส่วนประกอบของยูอาร์แอล โดยเก็บส่วนประกอบที่เหมือนกันไว้ในที่เดียวกันและสิ่งที่แตกต่างกันจะแยกเก็บไว้ในโครงสร้างต้นไม้ย่อยอันอื่นคล้ายกับกรรมวิธีในการบีบอัดข้อมูล ด้วยวิธีการดังกล่าวสามารถประหยัดพื้นที่ในการจัดเก็บไปได้มาก ส่วนงานวิจัยของ Pant *et al.* (2002) ไม่ได้มุ่งเน้นถึงประสิทธิภาพของเว็บคราวเลอร์ จึงป้องกันหน่วยความจำไม่เพียงพอโดยจำกัดจำนวนยูอาร์แอลในคิวไว้ประมาณหนึ่งแสนยูอาร์แอลเท่านั้น

2. ตัวเก็บเว็บเพจ (Downloader)

ทำหน้าที่เก็บเว็บเพจจากอินเทอร์เน็ต ประกอบไปด้วยกระบวนการย่อยได้แก่ ตัวเก็บเว็บเพจโดยใช้โปรโตคอลเอชทีทีพี (HTTP downloader) ทำหน้าที่เก็บเว็บเพจเฉพาะยูอาร์แอลที่เป็นโปรโตคอลเอชทีทีพี ซึ่งเป็นโปรโตคอลหลักของเว็บเพจบนอินเทอร์เน็ต ดีเอ็นเอสแคช (DNS cache) เป็นหน่วยความจำสำหรับเก็บหมายเลขไอพีแอดเดรสที่ได้จากการแปลงชื่อเซิร์ฟเวอร์ในยูอาร์แอลให้เป็นหมายเลขไอพีแอดเดรส (DNS resolving) เพื่อลดภาระในการติดต่อกับดีเอ็นเอสเซิร์ฟเวอร์ (DNS server) คิวการดาวน์โหลดต่อเซิร์ฟเวอร์ (Per-server queue) ทำหน้าที่จัดลำดับยูอาร์แอลใหม่ เพื่อเป็นการป้องกันไม่ให้ตัวเก็บเว็บเพจเข้ามาเก็บเว็บเพจเป็นจำนวนมากในเซิร์ฟเวอร์เดียวกันและในช่วงเวลาเดียวกัน การจัดเก็บเอกสาร (Document repository) อธิบายถึงวิธีการจัดเก็บเว็บเพจลงในแหล่งที่เก็บข้อมูล และหัวข้อสุดท้ายคือการตรวจสอบเนื้อหาของเว็บเพจที่เก็บมาแล้ว (Content seen test) จะตรวจสอบความซ้ำซ้อนของเนื้อหาเว็บเพจ และจะทิ้งเนื้อหานั้นถ้าซ้ำกันกับเนื้อหาของเว็บเพจที่ได้เคยเก็บรวบรวมไปแล้ว ทั้งนี้เพื่อให้ใช้พื้นที่ดิสก์ของแหล่งที่เก็บข้อมูลได้อย่างคุ้มค่ามากที่สุด ซึ่งรายละเอียดของหัวข้อทั้งหมดจะได้กล่าวถึงในหัวข้อถัดไป

2.1 ตัวเก็บเว็บเพจโดยใช้โปรโตคอลเอชทีทีพี (HTTP Downloader)

ทำหน้าที่เก็บเว็บเพจเฉพาะยูอาร์แอลที่เป็นโปรโตคอลเอชทีทีพี ซึ่งเป็นโปรโตคอลหลักของเว็บเพจบนอินเทอร์เน็ต โดยยูอาร์แอลที่ใช้เป็นตัวระบุเว็บเพจจะถูกแบ่งออกเป็น 4 ส่วน ได้แก่ โปรโตคอล (Protocol) ชื่อเซิร์ฟเวอร์ (Server) หมายเลขพอร์ต (Port) และพาธ (Path) ตัวอย่างการแบ่งส่วนประกอบของยูอาร์แอลจากตารางที่ 1

ตารางที่ 1 การแยกส่วนประกอบของยูอาร์แอล

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	/
https://ftp.ku.ac.th/pub/boot.im	ftp	ftp.ku.ac.th	443	/pub/boot.im

กระบวนการที่ตัวเก็บเว็บเพจร้องขอเว็บเพจจากเว็บเซิร์ฟเวอร์นั้น ขั้นแรกจะทำการติดต่อกับดีเอ็นเอสเซิร์ฟเวอร์ (DNS server) เพื่อขอแปลงชื่อเซิร์ฟเวอร์ให้เป็นหมายเลขไอพีแอดเดรส เช่น www.ku.ac.th จะถูกแปลงเป็น 158.108.216.5 เป็นต้น จากนั้นจะเชื่อมต่อไปยังเว็บเซิร์ฟเวอร์ปลายทาง โดยใช้หมายเลขไอพีที่ได้พร้อมระบุหมายเลขพอร์ตและโปรโตคอล เมื่อการเชื่อมต่อสำเร็จเว็บเซิร์ฟเวอร์จะตอบรับการเชื่อมต่อ จึงเริ่มเข้าสู่กระบวนการร้องขอไฟล์จากเว็บเซิร์ฟเวอร์ตามพาธที่ระบุไว้ เว็บเซิร์ฟเวอร์จะไปค้นหาไฟล์และส่งกลับมาเป็นกระแสอักขระกลับมายังตัวเก็บเว็บเพจที่ร้องขอจนสิ้นสุดสายอักขระจึงเป็นอันจบสิ้นกระบวนการ

โดยทั่วไปตัวเก็บเว็บเพจจะเสียเวลาในการรอคอยการติดต่อกับเว็บเซิร์ฟเวอร์ บางครั้งอาจรอการตอบกลับจนหมดเวลาโดยไม่ได้รับข้อมูล ทำให้การจัดเก็บเว็บเพจเกิดความล่าช้าได้ วิธีหนึ่งที่ถูกนำมาใช้ คือ เธรด (Thread) ซึ่งเป็นกลไกหนึ่งที่ภาษาโปรแกรมระดับสูงได้เตรียมเอพีไอ (API) ไว้บริการอยู่แล้ว ด้วยวิธีนี้ทำให้ตัวเก็บเว็บเพจหลาย ๆ ตัวทำการเก็บเว็บเพจหลาย ๆ เว็บเพจไปพร้อม ๆ กันได้ในขณะเวลาหนึ่ง ๆ ยังผลให้อัตราการเก็บเว็บเพจต่อช่วงเวลา (download rate) เพิ่มขึ้น เนื่องจากในขณะที่รอคอยการเก็บเว็บเพจหนึ่งเว็บเพจอื่นอาจจะถูกเก็บเรียบร้อยไปแล้ว ดังนั้นวิธีการนี้จึงเป็นที่นิยมในการเพิ่มประสิทธิภาพของเว็บคราวเลอร์อย่างกว้างขวาง

การเรียกใช้ซ็อคเก็ตแบบไม่ปิดกั้นข้อมูล (Non-blocking socket) เป็นอีกวิธีการหนึ่งที่ช่วยเพิ่มประสิทธิภาพของตัวเก็บเว็บเพจให้รับข้อมูลจากเว็บเซิร์ฟเวอร์ได้ทันทีและบันทึกข้อมูลลงฐานข้อมูลเว็บเพจ (Document repository) ได้ในเวลาเดียวกันด้วย โดยจะมอบภาระในการปิดการรับส่งข้อมูลให้เป็นหน้าที่ของระบบเครือข่ายเองที่ต้องคอยสอบถาม (Polling) กับเว็บเซิร์ฟเวอร์อยู่ตลอดเวลาว่าข้อมูลสิ้นสุดแล้วหรือไม่ หากสายอักขระตัวสุดท้ายของข้อมูลสิ้นสุดลงระบบเครือข่ายจะปิดการรับข้อมูลโดยอัตโนมัติ

2.2 ดีเอ็นเอสแคส (DNS Cache)

ดีเอ็นเอสแคช ถูกออกแบบมาเพื่อใช้เก็บข้อมูลหมายเลขไอพีแอสเดรสที่ได้รับมาจากดีเอ็นเอสเซิร์ฟเวอร์ เนื่องจากการเปลี่ยนจากชื่อเซิร์ฟเวอร์เป็นหมายเลขไอพีแอดเดรสเป็นกระบวนการที่ใช้เวลาค่อนข้างมาก ยิ่งไปกว่านั้นถ้ามีปริมาณการร้องขอเป็นจำนวนมากพร้อม ๆ กันหรือเป็นแปลงจากชื่อเซิร์ฟเวอร์เดิมซ้ำ ๆ กันก็อาจจะทำให้ดีเอ็นเอสเซิร์ฟเวอร์ทำงานหนักมากขึ้นและเป็นเหตุให้ระบบเครือข่ายหยุดให้บริการได้ ดังนั้นวิธีการนี้จึงช่วยลดการติดต่อกับดีเอ็นเอสเซิร์ฟเวอร์และทำให้ตัวเก็บเว็บเพจทำงานได้อย่างรวดเร็วมากยิ่งขึ้น

โครงสร้างข้อมูลที่ใช้จะต้องมีคุณสมบัติในการเข้าถึงข้อมูลที่เก็บอย่างรวดเร็ว เช่น แฮชเทเบิ้ล (Hash table) เป็นต้น ข้อมูลอาจจะถูกจัดเก็บไว้ในดิสก์หรือไว้ในหน่วยความจำก็ได้ ซึ่งขึ้นอยู่กับวัตถุประสงค์ในการออกแบบว่าต้องการเน้นปริมาณข้อมูล หรือต้องการความเร็วในการเข้าถึงข้อมูลมากกว่ากัน ในการเข้าถึงข้อมูลถ้าถูกสร้างด้วยแฮชเทเบิ้ลจะส่งคีย์เพื่อเป็นคำค้นเข้าไปและแคชจะไปค้นหาและตอบกลับมาเป็นค่าที่ต้องการ เช่น ถ้าส่งคำค้น www.ku.ac.th และจะได้ข้อมูลตอบกลับมาเป็น 158.108.216.5 เป็นต้น

ตัวอย่างงานวิจัยที่ใช้ดีเอ็นเอสแคช คือ เว็บคราวเลอร์โพลีเทคนิค (Shkapenyuk and Suel, 2002) ซึ่งเป็นเว็บคราวเลอร์ประสิทธิภาพสูงได้ใช้แคชไลบรารี่ชื่อว่าเอดีเอ็นเอส (ADNS) ซึ่งเป็นโอเพ่นซอร์สซอฟต์แวร์ที่สามารถดาวน์โหลดมาใช้และปรับแต่งได้ง่าย Heydon and Najork (1999) ได้วัดผลและรายงานประสิทธิภาพของการใช้ดีเอ็นเอสแคช จากการทดสอบพบว่าช่วยลดเวลาในการร้องขอจากดีเอ็นเอสเซิร์ฟเวอร์ถึง 25% จึงแสดงถึงความจำเป็นในการใช้ดีเอ็นเอสแคชได้เป็นอย่างดี

2.3 คิวการดาวน์โหลดต่อเซิร์ฟเวอร์ (Per-Server Queue)

คุณสมบัติข้อหนึ่งของเว็บคราวเลอร์จะต้องมีความสุภาพ (Politeness) คือ การไม่เข้าไปทำให้ระบบเครือข่ายที่เว็บคราวเลอร์ไปเก็บเว็บเพจนั้นเกิดภาระในการให้บริการแก่เว็บคราวเลอร์ของเรามากจนเกินไปจนไม่สามารถให้บริการแบบปกติกับการร้องขอแบบอื่น ๆ ได้ ดังนั้นคิวการดาวน์โหลดต่อเซิร์ฟเวอร์จึงเป็นวิธีการที่ทำหน้าที่จัดลำดับยูอาร์แอลใหม่ โดยพิจารณาจากชื่อของเว็บเซิร์ฟเวอร์ที่ปรากฏอยู่ในยูอาร์แอลนั้นโดยจะจัดให้สลับกันและพยายามไม่ให้อยู่ในช่วงเวลาเดียวกัน ทั้งนี้เพื่อเป็นการป้องกันไม่ให้ตัวเก็บเว็บเพจเข้ามาเก็บเว็บเพจเป็นจำนวนมากในช่วงเวลาขณะนั้น จนทำให้เว็บเซิร์ฟเวอร์หยุดการให้บริการเนื่องจากประมวลผลและส่งข้อมูลไม่ทัน (Denial of Service: DoS) หรือคล้ายกับว่ากำลังถูกโจมตีจากเว็บคราวเลอร์นั่นเอง

วิธีการจัดลำดับยูอาร์แอลที่น่าสนใจคือเทคนิคการสลับเฟส (Phase swapping) ซึ่งนำเสนอโดย Koht-Arsa and Sanguanpong (2002) โดยนำเอาชื่อเซิร์ฟเวอร์มามอดดูโลกับเฟส (เฟสจะเพิ่มขึ้นทีละ 1 ตามช่วงเวลาที่เปลี่ยนไปและจะกลับมาเริ่มต้นใหม่เป็นค่าแรกอีกครั้งเมื่อเพิ่มจนถึงค่าสูงสุดที่ตั้งไว้) ก็จะได้ค่าตัวเลขค่าหนึ่งที่มีความสัมพันธ์กับตัวเก็บเว็บเพจที่จะรับผิดชอบในการเก็บเว็บเพจนั้นต่อไป

ข้อกำหนดในการป้องกันเว็บคราวเลอร์ (Robot exclusion protocol) คือข้อตกลงที่กำหนดมาเพื่อป้องกันไม่ให้เว็บคราวเลอร์เข้าไปเก็บเว็บเพจที่ผู้ดูแลข้อมูลของเว็บเซิร์ฟเวอร์นั้นไม่ได้อนุญาต ในทางกลับกันเพื่อใช้กำหนดว่าเว็บคราวเลอร์สามารถเก็บไฟล์ในไดเร็คทอรี่ใดได้เช่นเดียวกัน ข้อกำหนดดังกล่าวถูกนำเสนอในครั้งแรกโดย Koster (1993) รายละเอียดของข้อกำหนดจะเก็บอยู่ในไฟล์ข้อความชื่อ robots.txt (1996) ซึ่งจะอยู่ในไดเร็กทอรี่ที่เป็นรากของเว็บไซต์ เช่น http://www.ku.ac.th/robots.txt เมื่อเว็บคราวเลอร์เข้ามาในเว็บไซต์แล้วจะต้องเปิดไฟล์ดังกล่าวและทำตามข้อกำหนดที่ได้กำหนดไว้

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /images/
Disallow: /tmp/
Disallow: /private/
```

ภาพที่ 3 ตัวอย่างข้อความในไฟล์ robots.txt

ข้อกำหนดในการเข้าถึงคือ User-agent เป็น “*” เป็นค่าโดยปริยายซึ่งจะยอมให้ทุกเว็บคราวเลอร์เข้ามาได้ แต่ถ้าระบุชื่อลงไปจะหมายความว่าเว็บคราวเลอร์ที่มีชื่อยูสเซอร์เอเจ้นท์ (User-agent) ตรงกันกับที่ระบุเท่านั้นจึงจะเก็บเว็บเพจได้ สำหรับ Disallow จะเป็นข้อมูลที่ไม่ให้เว็บคราวเลอร์เข้าไปใช้งานไฟล์ใดได้บ้าง รายละเอียดของไฟล์ดังกล่าวแสดงดังภาพที่ 3

2.4 การจัดเก็บเอกสาร (Document Repository)

เมื่อเว็บคราวเลอร์เก็บเว็บเพจมาแล้ว จะทำการบันทึกเว็บเพจเหล่านั้นลงในพื้นที่สำหรับเก็บเว็บเพจเพื่อนำข้อมูลเว็บเพจนั้นมาใช้ประโยชน์ในลำดับถัดไปของระบบสืบค้นข้อมูล พื้นที่จัดเก็บดังกล่าวมักจะใช้เป็นหน่วยความจำถาวร เช่น ในฮาร์ดดิสก์เพื่อให้รองรับกับปริมาณเว็บเพจที่มีเป็นจำนวนมาก วิธีลดความสิ้นเปลืองวิธีหนึ่งคือการบีบอัดข้อมูลเว็บเพจให้มีขนาดที่เล็กลงเพื่อประหยัดพื้นที่ในการจัดเก็บ โดยทั่วไปจะใช้ไลบรารี่ที่เป็นมาตรฐานในการบีบอัด เช่น ซีลิบ (Zlib, 2005) เป็นต้น โดยมีอัตราการบีบอัดเว็บเพจถ้ามีขนาด 10 กิโลไบต์จะถูกบีบอัดลงมาเหลือประมาณ 2-4 กิโลไบต์โดยประมาณ

งานวิจัยที่เกี่ยวข้องโดยส่วนใหญ่มักจะทำการแฮชเนื้อหาของเว็บเพจและจัดเก็บควบคู่ไปด้วยเพื่อนำมาใช้ในการตรวจสอบเนื้อหาซึ่งจะกล่าวถึงในหัวข้อถัดไป แต่สำหรับในวิทยานิพนธ์นี้เพื่อให้การจัดเก็บเว็บเพจได้ง่ายและหลีกเลี่ยงชื่อไฟล์ที่ยาวเกินไป จะจัดเก็บโดยสร้างไดเร็กทอรี่แยกตามเว็บไซต์ที่เป็นของเว็บเพจเหล่านั้น ยกตัวอย่างเช่น http://www.ku.ac.th/help.html จะถูกแปลงเป็นไดเร็กทอรี่ www-ku-ac-th และชื่อไฟล์ของเว็บเพจในไดเร็กทอรี่นั้น เกิดจากการแปลงยูอาร์แอลทั้งหมดโดยใช้เอ็มดีไฟว์ (MD5, 2005) ซึ่งเป็นแฮชฟังก์ชั่นอัลกอริทึมในการแปลงข้อมูลให้เป็นรหัสแฮช 32 บิต ใช้คลาสมาตรฐานของจาวาในการแปลง ดังนั้นผลลัพธ์สุดท้ายจะได้ชื่อไฟล์ที่มีความยาวคงที่ ซึ่งในที่นี้รหัสแฮชของยูอาร์แอลข้างต้นคือ e512d612a558dcbc4f6e3c1d3ab6683b และจะใช้ทั้งไดเร็กทอรี่และไฟล์ดังกล่าวนี้ในการอ้างอิงต่อไป

2.5 การตรวจสอบเนื้อหาของเว็บเพจที่เก็บมาแล้ว (Content Seen Test)

ด้วยปริมาณอันมหาศาลของเว็บเพจบนอินเทอร์เน็ต จึงมีโอกาสที่จะเกิดกรณีที่เว็บเพจหนึ่งอาจจะมีเนื้อหาเหมือนกับเว็บเพจอื่นก็ได้ ดังนั้นถ้าเว็บคราวเลอร์เก็บเว็บเพจที่มีเนื้อหาซ้ำซ้อนก็จะทำให้สูญเสียทรัพยากรคอมพิวเตอร์โดยไม่จำเป็น เพื่อป้องกันไม่ให้เว็บคราวเลอร์เก็บเว็บเพจที่

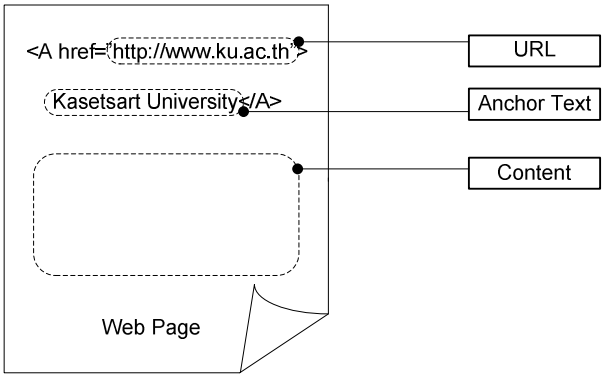
มีเนื้อหาซ้ำซ้อน งานวิจัยส่วนใหญ่จึงมีเทคนิคในการตรวจสอบเนื้อหาของเว็บเพจอย่างง่าย เช่น นำเว็บเพจมาผ่านการแปลงโดยใช้อัลกอริทึมเช่น เอ็มดีไฟว์แล้วนำมาตรวจสอบ หากพบว่าข้อมูลที่ได้เหมือนกันกับเว็บเพจที่เก็บมาแล้วแสดงว่าเป็นเนื้อหาเดียวกัน ดังนั้นเว็บคราวเลอร์ก็จะทิ้งเว็บเพจนั้น ในทางตรงกันข้ามหากข้อมูลแตกต่างกันเว็บคราวเลอร์จะเก็บบันทึกเว็บเพจนั้นตามวิธีการเก็บเอกสารเว็บเพจดังที่กล่าวมาแล้วต่อไป

3. ตัวคัดแยกยูอาร์แอล (URL Parser)

ทำหน้าที่ในการคัดแยกยูอาร์แอลออกมาจากเว็บเพจ (Link extraction) จากนั้นจะทำการแปลงยูอาร์แอลให้เป็นรูปแบบมาตรฐาน (URL normalize) คัดกรองเฉพาะรูปแบบยูอาร์แอลที่ต้องการ (URL filter) และสุดท้ายจะนำยูอาร์แอลนั้นไปตรวจสอบว่าเคยเก็บเว็บเพจจากยูอาร์แอลนี้แล้วหรือไม่ (URL seen) เพื่อป้องการการเก็บเว็บเพจนั้นซ้ำซ้อนซึ่งรายละเอียดจะกล่าวถึงในหัวข้อถัดไป

3.1 การแยกยูอาร์แอล (Link Extraction)

จะแยกยูอาร์แอลออกมาจากเว็บเพจ โดยอาศัยแท็กของภาษาเอชทีเอ็มแอล (HTML) ที่เขียนกำกับอยู่ สำหรับการแยกแยะยูอาร์แอลและเนื้อหาของเว็บเพจออกมาเป็นส่วนประกอบต่าง ๆ ได้นั้นแสดงดังภาพที่ 4 และแท็กเอชทีเอ็มแอลที่สำคัญที่ใช้คัดแยกยูอาร์แอลนั้นแสดงดังตารางที่ 2 ได้ดังต่อไปนี้



ภาพที่ 4 ส่วนประกอบภายในเว็บเพจ

ตารางที่ 2 แท็กเอชทีเอ็มแอลที่กำกับในการแยกลิงค์

แท็ก	ตัวอย่างการใช้แท็ก
A	<A href=”http://www.ku.ac.th”>Kasetsart University</A>
IMG	<IMG src=”http://www.ku.ac.th”>
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”>

3.2 รูปแบบยูอาร์แอลมาตรฐาน (URL Normalize)

ในกรณียูอาร์แอลมีชื่อต่างกันแต่อ้างถึงเว็บเพจเดียวกันจะต้องนำมาแปลงให้อยู่ในรูปแบบของยูอาร์แอลมาตรฐาน (Normalize) เพื่อป้องกันการเก็บเว็บเพจเดียวกันซ้ำซ้อน รายละเอียดการแปลงยูอาร์แอลให้อยู่ในรูปแบบมาตรฐานแสดงตามขั้นตอนต่อไปนี้

3.2.1 แปลงโปรโตคอลและชื่อโฮสต์เป็นอักษรตัวเล็ก เช่น HTTP://www.KU.ac.th จะถูกแปลงเป็น http://www.ku.ac.th

3.2.2 ทำการลดรูปยูอาร์แอลที่มีการอ้างอิงอยู่ภายในเอกสารเดียวกัน ให้มีรูปแบบที่สั้นลง เช่น http://www.ku.ac.th/faq.html#what จะถูกลดรูปให้เป็น http://www.ku.ac.th/faq.html

3.2.3 เข้ารหัสยูอาร์แอลในกรณีที่เป็นอักขระพิเศษ เช่น http://www.ku.ac.th/~user1 จะถูกเข้ารหัสเป็น http://www.ku.ac.th/%7Euser1

3.2.4 ในบางยูอาร์แอลจะเพิ่มอักขระ “/” ต่อท้ายยูอาร์แอลเข้าไปด้วย ยกตัวอย่างเช่น http://www.ku.ac.th จะถูกแปลงเป็น http://www.ku.ac.th/

3.2.5 ในบางกรณีชื่อไฟล์ที่มีชื่อเป็น index.html หรือ index.htm จะถูกลบออกไป เพราะเนื่องจากสมมติฐานว่า ชื่อไฟล์ดังกล่าวเป็นชื่อไฟล์โดยปริยายของเว็บเซิร์ฟเวอร์อยู่แล้ว เช่น http://www.ku.ac.th/index.html ถูกลบออกไปเป็น http://www.ku.ac.th

3.2.6 กรณีที่พบอักษร ".../" ที่หมายถึงการอ้างอิงไปยังพาเร้นท์ไดเร็กทอรี่ของไฟล์ที่ปรากฏอยู่ในยูอาร์แอลนั้นและจะถูกลบออกไป เช่น http://www.ku.ac.th/pub/.../index.html จะถูกเปลี่ยนเป็น http://www.ku.ac.th/pub/index.html

3.2.7 ลบหมายเลขพอร์ตออกไปในกรณีที่เป็น 80 เท่านั้น เช่น http://www.ku.ac.th:80 ถูกเปลี่ยนเป็น http://www.ku.ac.th เนื่องจากพอร์ตดังกล่าวเป็นเลขมาตรฐานของเว็บเซิร์ฟเวอร์อยู่แล้ว

3.3 ตัวกรองยูอาร์แอล (URL Filter)

ตัวกรองยูอาร์แอลมีวัตถุประสงค์เพื่อเลือกยูอาร์แอลที่มีรูปแบบที่ต้องการก่อนที่จะส่งเข้าไปในยูอาร์แอลคิวต่อไป การเลือกรูปแบบยูอาร์แอลที่ดีและตรงตามรูปแบบมาตรฐานจะทำให้เว็บคราวเลอร์ทำงานได้อย่างราบรื่น และยังเป็นวิธีการหนึ่งในการป้องกันปัญหาการเกิดกับดักเว็บคราวเลอร์อีกด้วย

กับดักเว็บคราวเลอร์ (Crawler trap) เกิดจากยูอาร์แอลหรือกลุ่มของยูอาร์แอลที่ทำให้เว็บคราวเลอร์ทำงานไม่จบสิ้นจนไม่สามารถไปเก็บเว็บเพจอื่นได้ ในบางครั้งปัญหาดังกล่าวอาจเกิดขึ้นโดยไม่ได้ตั้งใจของผู้สร้างหรือเจ้าของเว็บเพจนั้นก็เป็นได้ ยกตัวอย่างเช่น การสร้างซิมโบลิคลิงค์ (Symbolic link) ในกลุ่มของเว็บเพจที่อยู่ในสาระบบไฟล์เดียวกันที่มีลิงค์เชื่อมระหว่างไฟล์ด้วยกันจนมีลักษณะของวงกลม (Cycle) ดังนั้นเมื่อเว็บคราวเลอร์เข้ามาเก็บเว็บเพจที่ในกลุ่มนี้ก็จะติดกับดักเช่นกัน หรือในบางครั้งอาจเกิดจากความตั้งใจของผู้สร้างเว็บเพจเอง เช่น เขียนโปรแกรมเพื่อให้สร้างเอกสารที่เชื่อมต่อกันอย่างต่อเนื่องกันไม่รู้จบเป็นต้น

งานวิจัยของ Pant *et al*. (2002) จะเลือกเฉพาะยูอาร์แอลที่ชี้ไปหาเว็บเพจคงที่ (Static page) นั่นคือไม่มีการส่งพารามิเตอร์โดยใช้เครื่องหมาย "?" หรือ "&" กำกับ นอกจากนั้นยังจำกัดความยาวของยูอาร์แอลให้มีความยาวขนาดไม่เกิน 255 ตัวอักษร โดยเลือกเฉพาะยูอาร์แอลที่มีโปรโตคอลและนามสกุลของไฟล์เป็นที่รู้จักกันเป็นอย่างดีเท่านั้น

3.4 การตรวจสอบยูอาร์แอลที่เก็บมาแล้ว (URL seen)

เป็นการตรวจสอบยูอาร์แอลว่าเว็บคราวเลอร์เคยเก็บเว็บเพจจากยูอาร์แอลนี้หรือไม่ หากเคยเก็บมาแล้วจะไม่นำยูอาร์แอลดังกล่าวใส่ลงในยูอาร์แอลคิว เพื่อป้องกันไม่ให้เว็บคราวเลอร์เก็บเว็บเพจซ้ำซ้อนซึ่งเป็นเหตุให้เว็บคราวเลอร์ไม่สามารถเก็บเว็บเพจอื่นได้ วิธีที่ง่ายที่สุดที่ใช้ในการตรวจสอบ คือ การแปลงยูอาร์แอล (ที่เปลี่ยนชื่อเซิร์ฟเวอร์ของยูอาร์แอลนั้นให้เป็นหมายเลขไอพีแอดเดรสแล้ว) ให้อยู่ในรูปแบบของเลขฐาน 16 โดยใช้อัลกอริทึมเอ็มดีไฟว์และนำมาเทียบกับเลขที่ได้จากฐานข้อมูลยูอาร์แอลที่เก็บรวบรวมมาแล้ว (URL seen database)

อย่างไรก็ตามเนื่องจากปริมาณยูอาร์แอลมีเป็นจำนวนมาก ดังนั้นประสิทธิภาพในด้านความเร็วในการตรวจสอบจึงเป็นตัวแปรสำคัญที่งานวิจัยส่วนใหญ่คำนึงถึง ซึ่งจะกล่าวในรายละเอียดดังนี้ อินเทอร์เน็ตอาร์ไคว์ (Burner, 1997) แก้ปัญหานี้โดยเทคนิคเรียกว่าบลูมฟิลเตอร์ (Bloom filter) โดยใช้บิตแมป (Bit map) ขนาดใหญ่โดยเริ่มต้นใส่ค่าเริ่มต้นเป็น 0 ในทุกช่อง เมื่อต้องการตรวจสอบยูอาร์แอลจะนำมายูอาร์แอลมาแฮชด้วยขนาดของบิตแมป ถ้าตกอยู่ในช่องใดและช่องนั้นมีค่า 1 แสดงว่ามียูอาร์แอลซ้ำกัน วิธีดังกล่าวจะให้ประสิทธิภาพที่ดีแต่ต้องใช้หน่วยความจำเป็นจำนวนมากในการเก็บบิตแมป เมอร์เคเตอร์ (Heydon and Najork, 1999) ใช้หน่วยความจำแคชเก็บค่าแฮชของยูอาร์แอลที่ใช้บ่อย ความบ่อยขึ้นอยู่กับความถี่ของชื่อเซิร์ฟเวอร์ของยูอาร์แอลนั้น ๆ ในขณะที่ยูอาร์แอลส่วนที่เหลือจะเก็บไว้ในดิสก์ เมื่อต้องการตรวจสอบความซ้ำซ้อนจะมาตรวจในแคชก่อน เมื่อไม่พบจึงไปตรวจสอบในดิสก์อีกทีหนึ่ง ซึ่งทำให้ประสิทธิภาพดีขึ้นกว่าการตรวจสอบยูอาร์แอลในดิสก์เพียงอย่างเดียวเป็นอย่างมาก

ในวิทยานิพนธ์นี้ไม่ได้มุ่งเน้นในการสร้างเว็บคราวเลอร์สมรรถนะสูง ดังนั้นจะสร้างตัวจดจำยูอาร์แอลโดยใช้ฐานข้อมูลเบิร์กเลย์ (Oracle Berkeley Database, 1994) ซึ่งเป็นฐานข้อมูลที่เป็นไฟล์บนดิสก์โดยใช้โครงสร้างข้อมูลแบบตารางแฮชซึ่งมีความเร็วและประสิทธิภาพในการค่อนข้างดีพอสมควร

ในแบบจำลองพื้นฐานในภาพที่ 1 และรายละเอียดในทางเทคนิคที่แสดงในภาพที่ 2 นั้นเป็นการนำเสนอเว็บคราวเลอร์ที่ทำงานแบบเดี่ยว (Sequential processing) หมายความว่าเว็บคราวเลอร์จะทำงานในคอมพิวเตอร์เครื่องเดียวเท่านั้น อย่างไรก็ตามงานยังมีงานวิจัยที่เกี่ยวกับเว็บคราวเลอร์ประสิทธิภาพสูงอีกกลุ่มหนึ่งได้นำเสนอการออกแบบเว็บคราวเลอร์ให้ทำงานในแบบขนาน

(Parallel processing) หมายความว่าเว็บคราวเลอร์จะทำงานได้ทีละหลายเครื่องพร้อมกัน เนื่องจากจำนวนเว็บเพจที่มีมหาศาลอยู่ในอินเทอร์เน็ต จากรายงานของเว็บไซต์ชื่อเวิร์ลไวด์เว็บไซส์ (World Wide Web Size, 2009) พบว่าในปี 2009 จำนวนเว็บเพจได้เกิน 20.39 พันล้านเพจแล้ว และยังมีแนวโน้มที่จะเพิ่มสูงขึ้นอีกเท่าตัวในอนาคต ดังนั้นการใช้เว็บคราวเลอร์แบบขนานจึงเป็นประโยชน์ในการเก็บเว็บเพจในระยะเวลาที่สั้นลง

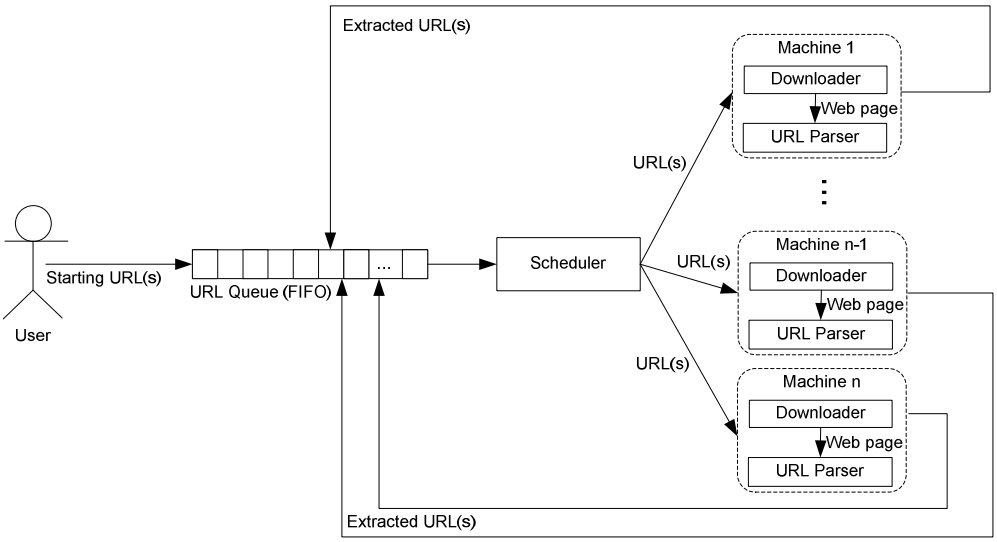
แบบจำลองเว็บคราวเลอร์แบบขนาน

แบบจำลองของเว็บคราวเลอร์แบบขนานสามารถแบ่งได้เป็น 2 ประเภทได้แก่ แบบมีเครื่องศูนย์กลาง และแบบไร้เครื่องศูนย์กลางซึ่งแสดงอยู่ในภาพที่ 5 และ 6 ตามลำดับ

1. แบบมีเครื่องศูนย์กลาง (Centralized Machine)

จากแบบจำลองนี้จะมีคอมพิวเตอร์อยู่เครื่องหนึ่งทำหน้าที่ในการเลือกยูอาร์แอลออกมาจากยูอาร์แอลคิวแล้วส่งไปให้คอมพิวเตอร์เครื่องใดเครื่องหนึ่งให้ทำหน้าที่ในการเก็บเว็บเพจต่อไป จะเรียกคอมพิวเตอร์นั้นว่าเป็น “ตัวจัดลำดับงาน” (Scheduler) การแบ่งงานจะแบ่งได้อย่างเหมาะสมและสมดุลย์เพียงใดขึ้นอยู่วิธีการของการจัดลำดับงาน (Scheduling algorithm) ที่อยู่ในตัวจัดลำดับงานนั้น อย่างไรก็ตามงานวิจัยส่วนใหญ่มักจะนิยมใช้เทคนิคของแฮชฟังก์ชั่นโดยนำยูอาร์แอลมามอดูโลกับจำนวนของเครื่องที่ทำหน้าที่เก็บเว็บเพจและผลลัพธ์ที่ได้จะเป็นหมายเลขของเครื่องที่รับผิดชอบนั่นเอง

การมีเครื่องศูนย์กลางมีข้อดี คือ ออกแบบง่ายไม่ซับซ้อนเนื่องจากคอมพิวเตอร์จะสื่อสารผ่านทางเครื่องศูนย์กลางเท่านั้นทำให้การสื่อสารระหว่างเครือข่ายมีน้อยตามไปด้วย ส่วนข้อเสียคือความทนทานต่อความผิดพลาดของระบบต่ำกล่าวคือถ้าเครื่องที่เป็นศูนย์กลางทำงานผิดพลาดหรือเสียหายจะส่งผลต่อการทำงานทั้งหมดของระบบ



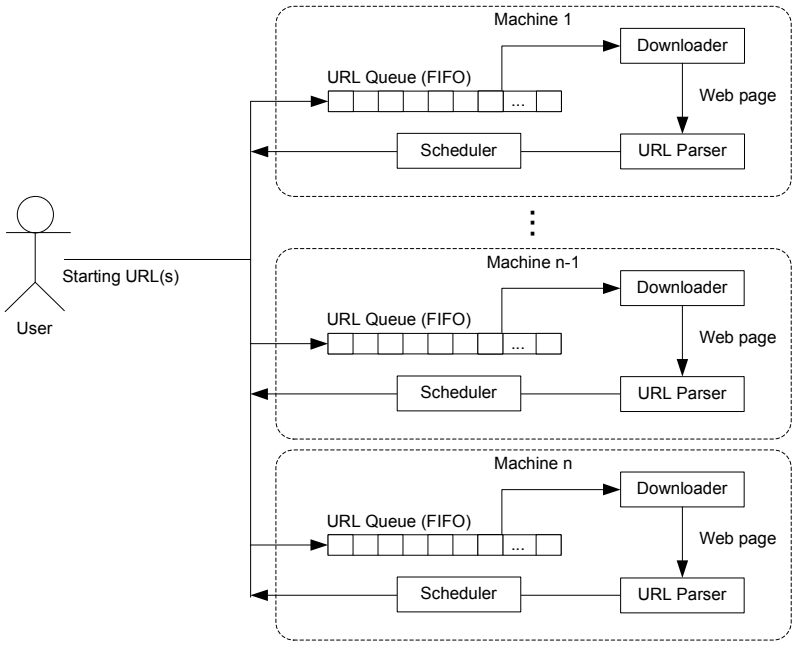
ภาพที่ 5 เว็บคราวเลอร์แบบขนานแบบมีเครื่องศูนย์กลาง

ตัวอย่างของระบบเว็บคราวเลอร์ที่ใช้อยู่บนพื้นฐานของแบบจำลองนี้ได้แก่ กูเกิ้ล (Brin and Page, 1998) ซึ่งจะมีเครื่องศูนย์กลางทำหน้าที่เป็นยูอาร์แอลคิวเซิร์ฟเวอร์และเป็นทั้งตัวจัดลำดับงาน ส่วนเครื่องที่ทำหน้าที่เก็บเว็บเพจจะมีทั้งสิ้น 3 เครื่อง โพลีเทคนิค (Burner, 1997) จะมีเครื่องศูนย์กลางในการจัดลำดับงานและส่งให้เครื่องที่ทำหน้าที่เก็บเว็บเพจจำนวน 8 เครื่อง ส่วนเคยู สไปเดอร์ (Koht-Arsa and Sanguanpong, 2002) จะมีเครื่องทำหน้าที่เก็บเว็บเพจอยู่ทั้งสิ้น 4 เครื่อง

2. แบบไร้เครื่องศูนย์กลาง (Decentralized Machine)

แบบจำลองนี้คอมพิวเตอร์ทุกเครื่องจะมีตัวจัดลำดับงานและยูอาร์แอลคิวเป็นของตนเอง โดยตัวจัดลำดับงานในแต่ละเครื่องจะตรวจสอบว่ายูอาร์แอลนั้นเป็นของเครื่องที่ตนเองรับผิดชอบหรือไม่ ถ้าเป็นจะส่งยูอาร์แอลนั้นลงในยูอาร์แอลคิวในเครื่องตนเอง แต่ถ้าเป็นของเครื่องอื่นจะส่งยูอาร์แอลนั้นให้เครื่องอื่นที่รับผิดชอบทำการเก็บเว็บเพจต่อไป

แบบไร้เครื่องศูนย์กลางมีข้อดี คือ ระบบมีความทนทานต่อความผิดพลาดสูงเนื่องจากทุกเครื่องรับผิดชอบเท่าเทียมกัน ดังนั้นเมื่อเครื่องใดเกิดขัดข้องเครื่องที่ยังเหลือก็สามารถทำงานทดแทนกันได้ ส่วนข้อเสียคือความซับซ้อนสูง และมีการสื่อสารระหว่างเครื่องคอมพิวเตอร์และเครือข่ายมากทำให้เกิดความไม่คุ้มค่าถ้าหากเก็บเว็บเพจจำนวนเพียงจำนวนเล็กน้อย



ภาพที่ 6 เว็บคราวเลอร์แบบขนานแบบไร้เครื่องศูนย์กลาง

ตัวอย่างของระบบเว็บคราวเลอร์ที่อยู่บนพื้นฐานของแบบจำลองนี้ได้แก่ เมอร์เคเตอร์ (Heydon and Najork, 1999) ใช้แฮชฟังก์ชั่นในการคำนวณโหลดเพื่อกระจายภาระงานให้กับตัวเก็บเว็บเพจ ซึ่งจะคล้ายกันกับยูบีไอ (Shkapenyuk and Suel, 2002) ใช้วิธีแฮชที่เรียกว่าวิธีแฮชที่สอดคล้องกัน (Consistent hashing) ซึ่งจะคอยตรวจสอบจำนวนของเครื่องคอมพิวเตอร์อยู่ตลอดเวลา เพื่อประโยชน์ในการเพิ่มเครื่องในอนาคตหรือเกิดเครื่องใดเครื่องหนึ่งเสียหาย

ตารางที่ 3 สรุปวิธีการออกแบบเว็บคราวเลอร์ประสิทธิภาพสูง

	ยูอาร์แอลคิว	ดีเอ็นเอส	ภาษาโปรแกรม	เธรดและ I/O	การกระจายงาน/ขนาด/อัตรา
กูเกิ้ล	-	แคชในแต่ละเครื่อง	ซีพลัสพลัสและไพธอน	300 (ASYN)	4 เครื่อง 24 ล้านเพจ 47 เพจ/วินาที
เมอร์เคเตอร์	ตารางแฮชในหน่วยความจำและเรียงข้อมูลในดิสก์	พัฒนาเอง	จาวา (ปรับปรุงความสามารถเองในบางส่วน)	100 (SYN)	4 เครื่อง 891 ล้านเพจ 600 เพจ/วินาที
อินเทอร์เน็ตอาร์ไคว์	บลูมฟิลเตอร์	-	-	64 (ASYN)	- 100 ล้านเพจ 10/วินาที
ยูบีไอ	-	-	จาวา	4 (SYN)	16 เครื่อง - 52 เพจ/วินาที
โพลีเทคนิค	เร็ดแบล็กทรีในหน่วยความจำและจัดเรียงข้อมูลในดิสก์	เอดีเอ็นเอส	ซีพลัสพลัสและไพธอน	1000 (ASYN)	3 เครื่อง 120 ล้านเพจ 140 เพจ/วินาที
เคยูสไปเดอร์	เอวีแอลทรี	-	ซีพลัสพลัส	300 (SYN)	4 เครื่อง 4 แสนเพจ 618 เพจ/วินาที

ที่มา: A Survey of High-Performance Web Crawler (Boswell, 2003)

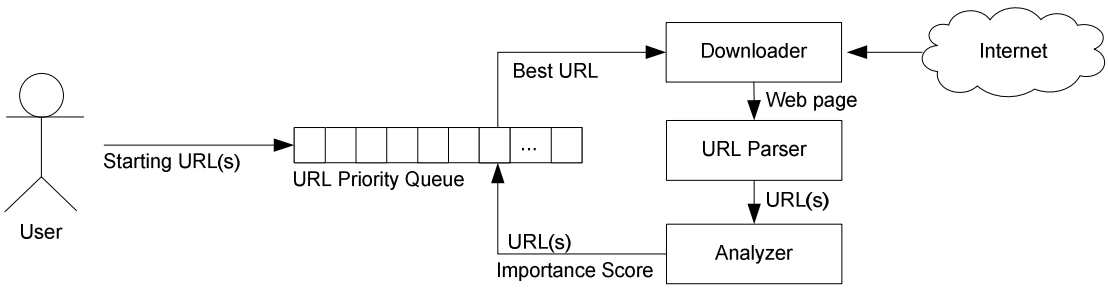
ถึงแม้ว่าเว็บคราวเลอร์ประสิทธิภาพสูงจะมีความสามารถในการเก็บรวบรวมเว็บเพจได้เป็นจำนวนมาก แต่ถ้าพิจารณาข้อมูลเว็บเพจที่เก็บมาได้จะพบว่ามีเว็บเพจเป็นจำนวนมากที่มีข้อมูลที่ไม่เป็นประโยชน์ในการทำดัชนีสำหรับระบบสืบค้นข้อมูล (Chakrabarti *et al.*, 1999) ในหัวข้อถัดไปจะกล่าวถึงเว็บคราวเลอร์แบบเจาะจงเนื้อหา ซึ่งจัดเป็นเว็บคราวเลอร์ที่ถูกออกแบบมาให้มีความชาญฉลาดที่สามารถเลือกเก็บเว็บเพจในหัวเรื่องที่ผู้ใช้ต้องการได้ ดังนั้นจึงช่วยแก้ปัญหาที่กล่าวมาในข้างต้นได้เป็นอย่างดี

การเก็บเว็บเพจแบบเฉพาะเจาะจงหัวเรื่อง

กล่าวได้ว่าประสิทธิภาพในการสืบคืนข้อมูลของเสิร์ชเอ็นจินนั้นขึ้นอยู่กับผลลัพธ์ที่ได้จากการสืบค้นว่ามีความครอบคลุมกับความต้องการของผู้ใช้มากเพียงไร ดังนั้นเสิร์ชเอ็นจินที่มีชื่อเสียงจึงพยายามที่จะเก็บรวบรวมเว็บเพจให้มากที่สุดเท่าที่จะทำได้ อย่างไรก็ตามหากจะประมาณการถึงจำนวนเว็บเพจที่มีอยู่ในอินเทอรเน็ตและระยะเวลาที่ตองใชในการรวบรวมเว็บเพจเหลานั้น ล้วนเป็นสิ่งที่คาดการณ์ได้ยากยิ่ง อีกทั้งเว็บเพจจำนวนมากที่เก็บรวบรวมมาไดกลับไมมีประโยชนในการทำดัชนีเนื่องจากเปนเว็บเพจที่มีเนื้อความที่ไม่เกี่ยวข้องหรือกว้างเกินไป (Chakrabarti *et al.*, 1999) จากปญหาเหล่านี้กอใหเกิดแนวคิดในการเก็บรวบรวมเว็บเพจใหตรงกับหัวเรื่องที่ตองการโดยรวบรวมเฉพาะเว็บเพจที่กลาวถึงในหัวเรื่องนั้น ดังนั้นเว็บเพจที่เก็บรวบรวมมาไดจึงนาจะมีประโยชนตอการทำดัชนีหรือการสืบคนมากยิ่งขึ้น ซึ่งจะเรียกการเก็บรวบรวมเว็บเพจในลักษณะนี้เรียกวา “การเก็บเว็บเพจแบบเฉพาะเจาะจงหัวเรื่อง”

เว็บคราวเลอรที่ใช้จะตองมีความสามารถในการเลือกเก็บเฉพาะเว็บเพจที่มีเนื้อความเกี่ยวข้องกับหัวเรื่องที่กำหนดใหเทานั้น ดังนั้นจึงเรียกเว็บคราวเลอรประเภทนี้เรียกวา “เว็บคราวเลอรแบบเฉพาะเจาะจงหัวเรื่อง” (Topic-specific web crawler)

ในหัวขอนี้กลาวถึงรูปแบบการทำงานของการเก็บเว็บเพจแบบเฉพาะเจาะจงหัวเรื่องดวยเว็บคราวเลอร รวมถึงอธิบายสวนประกอบและทฤษฎีที่นำมาประยุกตใชเพื่อทำใหเว็บคราวเลอรมีความสามารถในการเลือกเก็บเว็บเพจตามคำสำคัญที่ผู้ใช้ได้กำหนด



ภาพที่ 7 แบบจำลองของเว็บคราวเลอร์แบบเจาะจงหัวเรื่อง

เว็บคราวเลอรแบบเฉพาะเจาะจงหัวเรื่องจะมีการทำงานคล้ายกันกับเว็บคราวเลอรแบบธรรมดาทั่วไป แตเนื่องจากเว็บคราวเลอรตองเก็บเฉพาะเว็บเพจที่มีเนื้อหาเกี่ยวของกับหัวเรื่องที่กำหนด ผู้ตั้งค่าระบบจึงไมเพียงแตตองกำหนดยูอารแอลเริ่มตนใหกับเว็บคราวเลอรเทานั้น แต่ตองกำหนดคำสำคัญของหัวเรื่อง (Topic keyword) ที่เกี่ยวของกับหัวเรื่องนั้นดวย ทั้งนี้การกำหนดหัวเรื่องที่แม่นยำเป็นสิ่งที่จำเปนมากสำหรับเว็บคราวเลอร เพราะเว็บคราวเลอรจะอาศัยเทคนิคการเปรียบเทียบคำสำคัญกับเนื้อความภายในเว็บเพจ และคาดเดาวาเว็บเพจที่พบนั้นมีความเป็นไปไดมากนอยเพียงใดที่จะเกี่ยวของหรือคลายคลึงกับหัวเรื่องที่กำหนดให้และจะมีผลตอยูอารแอลใหมที่พบภายในเว็บเพจนั้นดวย หากเนื้อความของเว็บเพจนั้นมีความเกี่ยวของสูงจึงมีความเปนไปไดมากที่ยูอารแอลใหมจะเปนเว็บเพจที่มีความเกี่ยวของเชนเดียวกัน แตในทางกลับกันเนื้อความไมเกี่ยวข้องกับหัวเรื่อง ยูอารแอลใหมภายในเว็บเพจจะมีความเกี่ยวข้องน้อยลงไปด้วยตามลำดับ

จากแบบจำลองในภาพที่ 7 จะมีสวนที่เพิ่มเติมขึ้นมาไดแก ตัววิเคราะห์เว็บเพจ (Analyzer) ทำหน้าที่วิเคราะหวาเว็บเพจที่เก็บมานั้นมีความเกี่ยวของกับหัวเรื่องที่ตองการมากนอยเพียงใด ผลลัพธจากการวิเคราะห์จะเป็นคาตัวเลขหนึ่งคาที่เรียกวา “คาความสำคัญ” (Important score) โดยอาจเปนคาจำนวนเต็มหรือทศนิยมซึ่งแล้วแต่วิธีการในการคำนวณค่า นอกจากนั้นยูอารแอลคิวที่ใชจะเป็นแบบเขาออกตามคาความสำคัญ (Priority queue) เมื่อตัววิเคราะหสงยูอารแอลที่พบใหมมาพรอมกับค่าความสำคัญ ยูอารแอลคิวนี้จะเรียงลำดับการเขาออกของยูอารแอลตามค่าความสำคัญ เนื่องจากคาความสำคัญบงบอกถึงความเกี่ยวของกับหัวเรื่อง ดังนั้นยูอารแอลที่ถูกคาดหมายวามีความเกี่ยวของมากที่สุดควรถูกเก็บมากอนยูอารแอลอื่น ๆ ภายในคิว

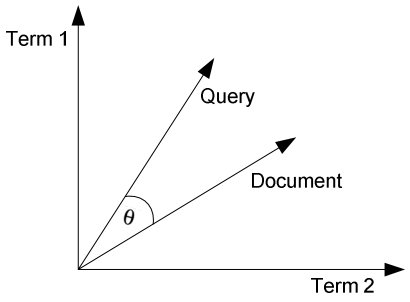
ในหัวข้อต่อไปจะกล่าวถึงในรายละเอียดการวิเคราะหเว็บเพจ ได้แก่การนำเสนอความรู้ ซึ่งเป็นวิธีในการวิเคราะห์เว็บเพจแบบต่าง ๆ และตัวอย่างงานวิจัยที่ได้นำเสนอวิธีการต่าง ๆ ในการให้คะแนนความสำคัญกับยูอาร์แอล

1. การนำเสนอความรู้ (Knowledge Representation)

ในหัวข้อนี้จะนำเสนอวิธีการที่ใช้ในการวิเคราะห์เว็บเพจว่ามีความเกี่ยวข้องกับหัวเรื่องที่ผู้ใช้ต้องการมากเพียงไร ซึ่งประกอบด้วยวิธีการการวิเคราะห์ในเชิงเนื้อหาได้แก่ แบบจำลองเว็คเตอร์สเป็ซ ทีเอฟไอดีเอฟและวิธีวิเคราะห์เชิงโครงสร้างลิงค์ได้แก่ ฮิตส์ เป็นต้น

1.1 แบบจำลองเว็คเตอร์สเปซ (Vector Space Model: VSM)

แบบจำลองเว็คเตอร์สเปซ (Baeza-Yates and Ribeiro-Neto, 1999) (Salton *et al.*, 1975) เป็นวิธีที่นิยมใช้ในการหาความคล้ายคลึงระหว่างเอกสาร เมื่อนำเอกสารมาเปรียบเทียบกัน หรือระหว่างเอกสารกับคำสำคัญ (keyword) ที่ผู้ใช้ส่งเข้ามา วิธีการคือจะแปลงเอกสารดังกล่าวให้อยู่ในรูปของเว็คเตอร์ ขั้นตอนการแปลงเอกสารให้อยู่ในรูปของเว็คเตอร์คือการสกัดคำออกมาจากเอกสาร โดยจะละทิ้งคำที่เป็นคำหยุด (Stop word) ซึ่งถือว่าเป็นคำที่ไม่มีความหมายในการบ่งบอกลักษณะของเอกสารนั้นออกไป เช่น “and” หรือ “the” เป็นต้น จากนั้นทำการเปลี่ยนคำที่ได้ดังกล่าวให้อยู่ในรูปของคำรากศัพท์ (Term) เช่นคำว่า “visited” ถูกแปลงเป็น “visit” เป็นต้น คำที่ได้มาแล้วอาจจะปรากฏได้มากกว่า 1 ที่ในเอกสารนั้นซึ่งจะเรียกว่าความถี่ของคำ (Term frequency) ดังนั้นเว็คเตอร์ที่ได้ในแต่ละตำแหน่งจะเก็บความถี่ของคำแต่ละคำ และขนาดของเว็คเตอร์จะเท่ากับจำนวนของคำที่อยู่ในเว็คเตอร์นั้น



ภาพที่ 8 ความคล้ายคลึงโคซายน์ระหว่างเว็คเตอร์ของเอกสารคู่หนึ่ง

ในการเปรียบเทียบความคล้ายคลึงระหว่างเว็คเตอร์ของเอกสารคู่หนึ่ง สามารถคำนวณได้จากค่าความคล้ายคลึงโคซายน์ (Cosine similarity: θ) ระหว่างเว็คเตอร์คู่นั้นดังแนวคิดตามภาพที่ 8 และดังสมการที่ 1

$$sim(p,q) = \frac{\sum_{i=1}^{t}(w_{i,p} \times w_{i,q})}{\sqrt{(\sum_{i=1}^{t} w_{i,p})^2 \times (\sum_{i=1}^{t} w_{i,q})^2}} \qquad (1)$$

เมื่อ $w_{i,p}$ และ $w_{i,q}$ คือความถี่ของคำที่ i ในเว็คเตอร์ของเอกสาร p และ q ตามลำดับ โดย t คือขนาดหรือเรียกว่ามิติของเว็คเตอร์นั้น ซึ่งค่าความคล้ายคลึงที่ได้จะอยู่ระหว่างค่า 0 ถึง 1 ถ้ามีค่ามากจะหมายความว่าเอกสารคู่ดังกล่าวมีความคล้ายคลึงกันมาก ส่วนถ้าค่าที่ได้เป็น 0 แสดงว่าเอกสารคู่นั้นไม่มีความคล้ายคลึงกันเลย

1.2 ทีเอฟไอดีเอฟ (TF-IDF)

วิธีการนับความถี่ของคำที่ปรากฏในเอกสารดังที่กล่าวมาแล้วข้างต้น นั้นยังไม่อาจบอกได้ว่าคำนั้นมีความสำคัญต่อเอกสารอย่างแท้จริง เช่น คำว่า “University” ปรากฏอยู่ในเอกสารหนึ่งเป็นจำนวนมาก ดังนั้นจึงน่าจะถือว่าคำนี้เป็นคำสำคัญที่ระบุคุณลักษณะของเอกสารได้ อย่างไรก็ตามถ้ามีเอกสารทั้งหมดจำนวน 5 เอกสารและมีคำดังกล่าวปรากฏในทุกเอกสาร คำดังกล่าวอาจจะไม่เป็นคำที่สำคัญก็ได้ เมื่อเปรียบเทียบกับคำว่า “Kasetsart” ซึ่งปรากฏเป็นจำนวนมากในเอกสารหนึ่ง แต่ปรากฏเพียงเอกสารเดียวจากเอกสารทั้งหมดเหล่านั้น ดังนั้นจึงหมายความว่า “Kasetsart” ไม่ได้เป็นคำทั่วไปหากแต่เป็นคำเฉพาะเจาะจงซึ่งสามารถใช้ระบุลักษณะของเอกสารนั้นได้จึงน่าจะมีความสำคัญของคำมากกว่า “University” ดังนั้นจากข้อสังเกตุดังกล่าวจึงเป็นที่มาของสมการในการหาความถี่ของคำทีเอฟไอดีเอฟ (Salton and Buckley, 1988) ดังนี้

$$w_{i,p} = tf_{i,p} \times \log_2 (\frac{N}{df_i}) \qquad (2)$$

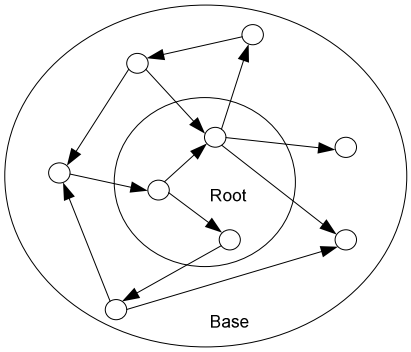
เมื่อ $tf_{i,p}$ คือความถี่ของคำที่ i ในเอกสาร p (เพื่อให้ค่าดังกล่าวอยู่ในขอบเขตที่คำนวณได้จึงมักจะนำ $\frac{tf_{i,p}}{n}$ เมื่อ n คือจำนวนคำทั้งหมดในเอกสาร p) สำหรับ N คือจำนวนเอกสารทั้งหมด ส่วนค่า df_i คือจำนวนเอกสารที่มีคำที่ i ปรากฏอยู่ และฟังก์ชั่น $\log_2$ ทำหน้าแปลงตัวเลขที่คำนวณได้ให้อยู่ในรูปของเลขฐานเดียวกันเพื่อให้สามารถวัดความแตกต่างได้

1.3 ฮิตส์ (Hypertext-Induced Topic Search: HITS)

เป็นเทคนิคในการจัดลำดับเว็บเพจจากผลลัพธ์ที่ได้ของระบบสืบค้นข้อมูลเช่นเดียวกับเพจแรงค์ (Brin and Page, 1999) ซึ่งถูกนำเสนอโดย Kleinberg (1998) จุดประสงค์คือต้องการแก้ปัญหาผลการค้นหาคำในหัวข้อที่มีความหลากหลาย (Broad topic) เช่น คำว่า “Harvard” ซึ่งจะ

ให้ผลลัพธ์จากการสืบค้นมากเกินกว่าผู้ใช้จะตรวจสอบได้ ดังนั้นงานวิจัยฮิตส์จึงเสนอมาตรวัด 2 ประเภทคือ ฮับ (Hub) และออทอริตี้ (Authority) โดยให้นิยามของฮับและออทอริตี้ไว้ดังนี้ เว็บเพจที่เป็นฮับคือเว็บเพจที่มีเส้นทางไปยังเว็บเพจที่น่าเชื่อถือเป็นจำนวนมาก และเว็บเพจที่เป็นออทอริตี้คือเว็บเพจที่มีความน่าเชื่อถือและถูกอ้างอิงจากเว็บเพจที่เป็นฮับจำนวนมาก

ในการคำนวณมาตรวัดฮับและออทอริตี้จำเป็นต้องสร้างเว็บกราฟจากคำค้นของผู้ใช้ โดยเริ่มต้นเมื่อผู้ใช้พิมพ์คำค้น (Query) ให้กับระบบสืบค้นข้อมูล ระบบสืบค้นข้อมูลจะให้ผลลัพธ์เป็นกลุ่มของลิงค์ที่มีความเกี่ยวข้องกับคำค้นนั้น ซึ่งในที่นี้จะเรียกว่า รูทเซ็ต (Root set) จากนั้นจึงเก็บเว็บเพจและนำแต่ละเว็บเพจที่อยู่ในรูทเซตมาขยายออกไปทั้งลิงค์ชี้เข้าและลิงค์ชี้ออกซึ่งจะเรียกว่า เบสเซ็ต (Base set) ดังภาพที่ 9 เมื่อได้กลุ่มเว็บเพจมาจำนวนหนึ่งแล้ว



ภาพที่ 9 รูทเซ็ตและเบสเซ็ต

กำหนดให้กลุ่มของเว็บเพจที่เก็บมาแล้วเป็นเสมือนกราฟแบบมีทิศทาง $G = (V, E)$ โดยกำหนดให้ V คือกลุ่มของเว็บเพจในเว็บกราฟ G และ E คือกลุ่มของลิงค์ที่เชื่อมต่อระหว่างเว็บเพจ และกำหนดให้ u คือเว็บเพจต้นทางและ v คือเว็บเพจปลายทาง ดังนั้นจะสามารถเขียนเมทริกซ์ความสัมพันธ์ M เป็นตัวแทนโครงสร้างของเว็บเพจในเว็บกราฟดังสมการที่ 3

$$M_{u,v} = \begin{cases} 1 & \text{If } (u,v) \in E \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

กำหนดให้ $\vec{A}_i$ และ $\vec{H}_i$ เป็นเว็คเตอร์ขนาด $N \times 1$ เมื่อ $N = |V|$ แสดงค่าออทอริตี้และค่าฮับในการคำนวณรอบที่ i ใดๆตามลำดับ โดยกำหนดให้ค่าออทอริตี้และค่าฮับเป็นเว็คเตอร์เริ่มต้น $\vec{A}_0 = \vec{H}_0 = [1]_{N\times 1}$ ดังนั้นสามารถเขียนเป็นสมการคำนวณได้ดังนี้

$$\vec{A}_{i+1} = M^T \times \vec{H}_i \quad (4)$$

$$\vec{H}_{i+1} = M \times \vec{A}_i \quad (5)$$

การคำนวณจะเป็นแบบวนซ้ำ (Iterative) โดยการคำนวณแต่ละรอบจะต้องปรับให้เว็คเตอร์ทั้ง $\vec{A}_i$ และ $\vec{H}_i$ ให้ผลรวมของสมาชิกที่อยู่ในแต่ละเว็คเตอร์ทั้งสองมีค่าเท่ากับ 1 เสมอและจะต้องตรวจสอบว่าการคำนวณลู่เข้าสู่คำตอบ (converge) หรือไม่ กล่าวคือถ้าผลรวมของสมาชิกของเว็คเตอร์ในการคำนวณรอบที่ $i+1$ กับ i มีความแตกต่างกันน้อยกว่าค่าที่ยอมรับได้ δ ค่าหนึ่งจึงหยุดการคำนวณ

1.4 ตัวจำแนกแบบเบส์อย่างง่าย (Naïve Bayes Classifier)

ตัวจำแนกแบบเบส์อย่างง่าย (Naïve Bayes classifier) (Mitchell, 1997) คือแบบจำลองที่ใช้สำหรับจำแนกข้อมูลที่ได้รับความนิยมอย่างแพร่หลายในงานที่เกี่ยวกับการเรียนรู้ของเครื่องจักรและการทำเหมืองข้อมูลซึ่งอยู่บนพื้นฐานจากกฎของเบส์ (Bayes theorem) โดยมีสมมติฐานที่กำหนดให้การเกิดของเหตุการณ์ต่าง ๆ ที่ใช้ในการจำแนกนั้นเป็นอิสระต่อกัน (conditionally independence)

1.4.1 กฎของเบส์ (Bayes' Theorem)

กำหนดให้ $P(H)$ คือความน่าจะเป็นที่จะเกิดเหตุการณ์ H และ $P(H \mid E)$ คือความน่าจะเป็นที่จะเกิดเหตุการณ์ H เมื่อเกิดเหตุการณ์ E ดังนั้นจากกฎของเบส์เราสามารถทำนายเหตุการณ์ที่จะเกิดขึ้นได้ดังสมการที่ 6

$$P(H \mid E) = \frac{P(E \mid H) \times P(H)}{P(E)} \quad (6)$$

ยกตัวอย่างการทำนายว่าฝนจะตกเมื่อมีเหตุการณ์มีเมฆดำ กำหนดให้ H คือเหตุการณ์ที่ฝนจะตกและ E คือเหตุการณ์มีเมฆดำ ดังนั้นเราจะสามารถทำนายการเกิดฝนตกได้ดังต่อไปนี้

P(ฝนตก | เมฆดำ) = [*P*(เมฆดำ | ฝนตก) x *P*(ฝนตก)] / *P*(เมฆดำ)
กำหนดให้

P(เมฆดำ | ฝนตก) คือความน่าจะเป็นที่มีเมฆดำเมื่อฝนตก ซึ่งในกรณีนี้เราจะพิจารณาการเกิดเมฆดำเมื่อมีเหตุการณ์ฝนตกเท่านั้น

P(ฝนตก) คือความน่าจะเป็นที่ฝนจะตก ความน่าจะเป็นนี้สามารถเก็บรวบรวมโดยใช้หลักการณ์ทางสถิติ เช่น การบันทึกวันที่มีฝนตกภายใน 1 ปี

P(เมฆดำ) คือความน่าจะเป็นที่มีเมฆดำ ความน่าจะเป็นนี้สามารถเก็บรวบรวมโดยใช้หลักการณ์ทางสถิติ แต่อย่างไรก็ตามการเกิดของเหตุการณ์ต่างๆที่ใช้ในการจำแนกถ้าไม่ใช่เหตุการณ์หลักที่เราพิจารณามักจะไม่ถูกบันทึกไว้และในบางกรณีเหตุการณ์เหล่านี้ยากต่อการบันทึก เช่น *P*(เมฆดำ) เป็นต้น

จากตัวอย่างที่กล่าวมานั้นเราสามารถทำนายเหตุการณ์โดยสังเกตการเกิดของเหตุการณ์บางอย่าง ซึ่งเหตุการณ์ที่นำมาใช้ในการทำนายนั้นต้องสอดคล้องกับเหตุการที่จะทำนาย เช่น ถ้าหากเราต้องการทำนายการตกของฝน เราก็จะไม่ใช้การเกิดแผ่นดินไหวมาพิจารณา เพราะการเกิดแผ่นดินไหวไม่ได้สอดคล้องกับการเกิดฝนตกเป็นต้น

1.4.2 แบบจำลองความน่าจะเป็นแบบเบส์อย่างง่าย (Naïve Bayes model)

ในการจำแนกข้อมูลนั้นอาจมีการเกิดของเหตุการณ์ต่าง ๆ ที่เกี่ยวข้องมากกว่า 1 ชนิดและเมื่อนำมาประยุกต์ใช้กับกฎของเบส์ดังที่กล่าวมา จะทำให้เกิดการคำนวณที่ซับซ้อนมากขึ้น เนื่องจากการขึ้นต่อกันของเหตุการณ์ เช่น เหตุการณ์ที่สองจะเกิดขึ้นเมื่อเกิดเหตุการณ์ที่หนึ่งไปแล้ว เป็นต้น ดังนั้นโมเดลความน่าจะเป็นแบบเบส์อย่างง่ายจึงตั้งให้สมมติฐานของแต่ละเหตุการณ์ต่าง ๆ ไม่ขึ้นต่อกันซึ่งเป็นที่มาของคำว่า “Naive” นั่นเอง พิจารณากฎของเบส์จากสมการที่ 7 แสดงการคำนวณเมื่อมี *n* เหตุการณ์โดยนำความน่าจะเป็นของแต่ละเหตุการณ์มาคูณกันได้ดังต่อไปนี้

$$P(H \mid E_1, E_2, ..., E_n) = \prod_i^n \frac{P(E_i \mid H) \times P(H)}{P(E_i)} \qquad (7)$$

ในการประยุกต์ใช้โมเดลความน่าจะเป็นแบบเบส์อย่างง่ายมาเป็นตัวจำแนกข้อมูลนั้น เราต้องนิยามคลาสของข้อมูลที่ต้องการจำแนก เช่น ถ้าต้องการจำแนกว่าควรจะไปเล่นเทนนิสหรือไม่ $\{yes, no\} \in H$ นั่นคือเราต้องคำนวณหาความน่าจะเป็นของแต่ละคลาสทั้ง *yes* และ *no* เสียก่อน ถ้าค่าความน่าจะเป็นของคลาสใดมีค่ามากกว่า เราจะจำแนกผลลัพธ์ให้จัดอยู่ในคลาสนั้น ซึ่งเขียนได้ดังสมการที่ 8 ดังนี้

$$output_{NB} = \arg\max_{h_j \in H} P(h_j) \prod_{i}^{n} P(E_i \mid h_j) \tag{8}$$

เมื่อ $output_{NB}$ คือผลลัพธ์ของคลาสที่ได้จำแนกแล้วโดย $\arg\max$ ทำหน้าที่เลือกคลาสผู้ชนะโดยพิจารณาคลาสที่มีความน่าจะเป็นมากที่สุด h_j คือคลาสที่ j ใดๆที่เรากำลังคำนวณความน่าจะเป็นอยู่ สำหรับ E_i คือเหตุการณ์ที่ i ใด ๆ ที่เรานำมาพิจารณาเพื่อจำแนก เนื่องจากเราต้องการเพียงคลาสที่ชนะไม่ได้ต้องการค่าความน่าจะเป็น ดังนั้นสามารถตัดตัวหาร $P(E_i)$ ออกไปได้เพื่อทำให้การคำนวณง่ายขึ้นเพราะตัวหารตัวเดียวกันของทั้งสองคลาสจึงไม่มีผลต่อคำตอบที่ได้ สมมติว่าเรามีเหตุการณ์ที่บันทึกไว้ดังตารางที่ 4 ต่อไปนี้

ตารางที่ 4 เหตุการณ์ตัวอย่างที่บันทึกไว้สำหรับสร้างแบบจำลองความน่าจะเป็นแบบเบส์อย่างง่าย

Outlook	Windy	Play
Rainy	True	No
Rainy	False	Yes
Rainy	True	No
Rainy	False	No
Sunny	True	No
Sunny	False	Yes
Sunny	True	Yes
Sunny	True	No
Sunny	False	No
Sunny	False	Yes

จากตารางที่ 4 เราจะสร้างแบบจำลองความน่าจะเป็นแบบเบส์อย่างง่ายได้ดังตารางที่ 5 ดังต่อไปนี้

ตารางที่ 5 แบบจำลองความน่าจะเป็นแบบเบส์อย่างง่าย

ความน่าจะเป็น	ค่า	ความน่าจะเป็น	ค่า
$P(play = yes)$	4/10	$P(play = no)$	6/10
$P(outlook = rainy \mid play = yes)$	1/4	$P(outlook = rainy \mid play = no)$	3/6
$P(outlook = sunny \mid play = yes)$	3/4	$P(outlook = sunny \mid play = no)$	3/6
$P(windy = true \mid play = yes)$	1/4	$P(windy = true \mid play = no)$	4/6
$P(windy = false \mid play = yes)$	3/4	$P(windy = false \mid play = no)$	2/6

เมื่อเราได้แบบจำลองความน่าจะเป็นมาเรียบร้อยแล้ว ถ้าต้องการจำแนกข้อมูลว่าควรจะไปเล่นเทนนิสหรือไม่ เมื่อมีเงื่อนไขของเหตุการณ์ $outlook = sunny$ และ $windy = false$ เราจะสามารถทำนายได้ดังต่อไปนี้

$$\begin{aligned} P(h \in yes \mid E_i) &= P(yes) \times P(outlook = sunny \mid play = yes) \times P(windy = false \mid play = yes) \\ &= 0.4 \times 0.75 \times 0.75 \\ &= 0.225 \end{aligned}$$

$$\begin{aligned} P(h \in no \mid E_i) &= P(no) \times P(outlook = sunny \mid play = no) \times P(windy = false \mid play = no) \\ &= 0.6 \times 0.5 \times 0.33 \\ &= 0.099 \end{aligned}$$

เพราะฉะนั้น $output_{NB} = yes$ จึงสรุปว่าด้วยเหตุการณ์ที่กำหนดมาในข้างต้นจัดอยู่ในคลาสของ yes ดังนั้นจึงทำนายว่าควรจะไปเล่นเทนนิสได้นั่นเอง

2. อัลกอริทึมที่ใช้ในเว็บคราวเลอร์แบบเจาะจงหัวเรื่อง (Web Crawling Algorithms)

หัวข้อนี้จะนำเสนอบางส่วนของงานวิจัยที่เกี่ยวข้องกับเว็บคราวเลอร์แบบเจาะจงหัวเรื่อง ซึ่งมีการคำนวณค่าความสำคัญแก่ยูอาร์แอลโดยอาศัยทฤษฎีที่กล่าวมาในหัวข้อข้างต้นมาใช้ในการคำนวณ ประกอบไปด้วยอัลกอริทึมแบบดั้งเดิม ฟิชเสิร์ชและชาร์คเสิร์ช โฟกัสคราวเลอร์ คอนเท็กซ์โฟกัสคราวเลอร์ โฟกัสคราวเลอร์แบบปรับตัวเองได้และเว็บคราวเลอร์แบบเรียนรู้ได้ตามลำดับ

2.1 อัลกอริทึมแบบดั้งเดิม (Naïve Best First Crawler)

นำเสนอโดย Cho *et al.* (1998) และ Hersovici *et al.* (1998) ซึ่งมีแนวคิดที่ว่าถ้าความคล้ายคลึงระหว่างคำสำคัญที่ผู้ใช้ส่งมากับเนื้อหาของเว็บเพจมีมาก ดังนั้นยูอาร์แอลออกจากเว็บเพจนั้นก็น่าที่จะนำพาเว็บคราวเลอร์ไปสู่เว็บเพจอื่น ๆ ที่มีเนื้อหาที่เกี่ยวข้องกันอย่างมากตามไปด้วย จากแนวคิดดังกล่าวจึงให้คะแนนความสำคัญกับยูอาร์แอลแต่ละตัวที่ชี้ออกจากเว็บเพจนั้นตามค่าคะแนนความคล้ายคลึงที่คำนวณได้ โดยใช้สมการความคล้ายคลึงที่ 1 ในการคำนวณ ข้อจำกัดของอัลกอริทึมนี้คือยูอาร์แอลแต่ละตัวในเว็บเพจใด ๆ จะได้รับคะแนนที่เท่าเทียมกัน จึงเป็นไปได้ว่าในบางยูอาร์แอลอาจไม่พบเว็บเพจที่เกี่ยวข้อง ดังนั้นเว็บคราวเลอร์จึงมีโอกาสสูงมากที่จะออกนอกเส้นทางจากกลุ่มเว็บเพจที่มีความเกี่ยวข้อง

2.2 ฟิชเสิร์ชและชาร์คเสิร์ช (Fish Search and Shark Search)

ฟิชเสิร์ชนำเสนอในปี 1994 โดย Bra *et al.* (1994) แนวความคิดหลักจะคล้ายกับอัลกอริทึมแบบดั้งเดิม แต่จะเพิ่มค่าความลึกในการค้นหาที่ยอมรับได้มากที่สุด (Depth) เมื่อเว็บคราวเลอร์ไม่พบเว็บเพจที่เกี่ยวข้องและมีระยะทางที่เกินค่าความลึกดังกล่าวจะตัดเส้นทางนั้นทิ้งเพื่อป้องกันไม่ให้เว็บคราวเลอร์หลุดไปสู่เส้นทางที่เป็นเว็บเพจที่ไม่เกี่ยวข้อง

ชาร์คเสิร์ชนำเสนอโดย Hersovici *et al.* (1998) ซึ่งพัฒนามาจากฟิชเสิร์ช และยังค้นพบว่าแองเคอร์เท็กซ์และคำที่อยู่บริเวณรอบแองเคอร์เท็กซ์นั้นเป็นส่วนประกอบที่สำคัญอย่างยิ่งที่จะใช้ในการคาดเดายูอาร์แอลว่าเป็นเว็บเพจที่เกี่ยวข้องหรือไม่ นอกจากนั้นแล้วยังมีสมมติฐานเพิ่มเติมว่าถ้าเว็บเพจที่เป็นที่ชี้เข้ามาหามีความเกี่ยวข้อง ดังนั้นเว็บเพจที่ถูกชี้มานั้นก็น่าที่จะมีความเกี่ยวข้องตามไปด้วย แนวคิดทั้งหมดเขียนเป็นสมการให้คะแนนแก่ยูอาร์แอลได้ดังสมการที่ 9

$$score(url) = \gamma.inherited(url) + (1-\gamma)neighborhood(url) \qquad (9)$$

จากสมการที่ 9 กำหนดให้ $\gamma < 1$ ซึ่งเป็นค่าถ่วงน้ำหนักของทั้งสองฟังก์ชั่น โดยฟังก์ชั่น $inherited(url)$ ทำหน้าที่หาค่าคะแนนความคล้ายคลึงของคำที่ปรากฏในเว็บเพจเมื่อเทียบกับคำสำคัญที่ผู้ใช้ส่งเข้ามา ส่วนฟังก์ชั่น $neighborhood(url)$ ทำหน้าที่หาค่าคะแนนความคล้ายคลึงของคำที่ปรากฏในแองเคอร์เท็กซ์เมื่อเทียบกับคำสำคัญที่ผู้ใช้ส่งเข้ามาเช่นเดียวกัน

$$inherited(url) = \begin{cases} \delta.sim(q,p) & \text{if } sim(q,p) > 0 \\ \delta.inherited(p) & \text{otherwise} \end{cases} \quad (10)$$

จากสมการที่ 10 คือรายละเอียดของฟังก์ชั่น $inherited(url)$ โดยกำหนดให้ $\delta < 1$ คือค่าเสื่อมสภาพ (Decay factor) ระหว่าง $sim(q, p)$ ซึ่งคือค่าความคล้ายคลึงระหว่างคำที่ปรากฏในเว็บเพจ p เมื่อเทียบกับคำสำคัญที่ผู้ใช้ส่งเข้ามาคือ q กับค่าคะแนนที่ได้จากเว็บเพจที่ชี้เข้ามาหา $inherited(p)$ นั่นเอง

$$neighborhood(url) = \beta.anchor(url) + (1-\beta).context(url) \quad (11)$$

จากสมการที่ 11 คือรายละเอียดของฟังก์ชั่น $neighborhood(url)$ โดยกำหนดให้ $\beta < 1$ คือค่าถ่วงน้ำหนักระหว่าง $anchor(url)$ ซึ่งเป็นฟังก์ชั่นสำหรับหาค่าความคล้ายคลึงระหว่างคำที่ปรากฏในแองค์เคอร์เท็กซ์เมื่อเทียบกับคำสำคัญที่ผู้ใช้ส่งเข้ามากับฟังก์ชั่นที่หาค่าความคล้ายคลึงของกลุ่มคำที่ปรากฏอยู่รอบแองค์เคอร์เท็กซ์นั้นซึ่งจะเรียกว่า $context(url)$ เมื่อเทียบกับคำสำคัญที่ผู้ใช้ส่งเข้ามาเช่นกัน

2.3 โฟกัสคราวเลอร์ (Focused Crawler)

โฟกัสคราวเลอร์ถูกพัฒนาขึ้นโดย Chakrabarti *et al.* (1999) โดยมีตัวจำแนกเอกสารเว็บ (Hypertext classifier) ทำหน้าที่จำแนกเว็บเพจที่เก็บมาแล้วและผลลัพธ์ที่ได้จะถูกแบ่งออกมาเป็น 2 กลุ่ม คือ กลุ่มเว็บเพจที่มีความเกี่ยวข้องและกลุ่มเว็บเพจที่ไม่มีความเกี่ยวข้องกับหัวเรื่องที่ผู้ใช้กำหนด โดยกำหนดสมมติฐานว่ายูอาร์แอลที่พบในเว็บเพจที่มีความเกี่ยวข้องก็น่าที่จะชี้ไปยังเว็บเพจอื่นๆ ที่มีความเกี่ยวข้องกับหัวเรื่องนั้นตามไปด้วย และเว็บเพจที่มีหัวเรื่องเดียวกันมักจะอยู่ในเส้นทางที่ใกล้กันในเว็บกราฟ (Topical locality) (Davison, 2000) ดังนั้นยูอาร์แอลที่มีคุณสมบัติดังกล่าวจะได้รับคะแนนความสำคัญที่สูงกว่ายูอาร์แอลที่จัดว่าอยู่ในกลุ่มที่ไม่เกี่ยวข้องกับหัวเรื่อง

ตัวจำแนกที่ใช้ในอัลกอริทึมนี้คือตัวจำแนกแบบเบส์ (Bayesian classifier) จะทำการจำแนกเว็บเพจ p ที่เก็บมาแล้วโดยนำกลุ่มคำที่ปรากฏอยู่ในเว็บเพจดังกล่าว มาจำแนกกับกลุ่มคำที่ปรากฏในหมวดหมู่ย่อย (Category) c (ที่ผู้ใช้ทำเครื่องหมายว่าเป็นหมวดหมู่ที่ต้องการ $c \in good$) ซึ่งมีลักษณะเป็นโครงสร้างต้นไม้ที่สร้างจากดีมอส (DMOZ, 1998) ขั้นตอนการจำแนกคือการหาค่า

ความน่าจะเป็น $\Pr(c \mid p)$ เมื่อเว็บเพจ p จะมีความเกี่ยวข้องกับหมวดหมู่ย่อย c ดังนั้นคะแนนความสำคัญของยูอาร์แอลที่ได้จากเว็บเพจ p เขียนเป็นสมการได้ดังนี้

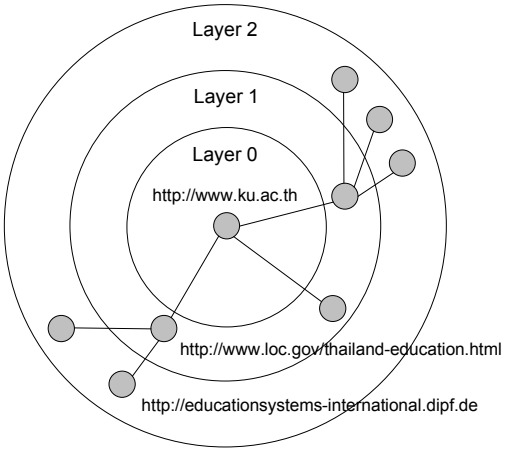
$$R(p) = \sum_{c \in good} \Pr(c \mid p) \tag{12}$$

โฟกัสคราวเลอร์ให้คะแนนความสำคัญแก่ยูอาร์แอล 2 แบบ ได้แก่แบบเบา (Soft focused) คือจะให้คะแนนความสำคัญกับทุกยูอาร์แอลที่ได้จากเว็บเพจนั้น ซึ่งคล้ายกับอัลกอริทึมแบบดั้งเดิม และแบบหนัก (Hard focused) จะละทิ้งยูอาร์แอลที่มีความเกี่ยวข้องที่น้อยกล่าวคือจะเลือกเฉพาะยูอาร์แอลที่ได้จากเว็บเพจ p ใด ๆ ที่จำแนกกับหมวดหมู่ย่อยที่เป็นลีฟ (leaf) c' โหนดในโครงสร้างต้นไม้นั้นแล้วพบว่าพาเร้นท์ (parent) ของ c' เป็นหัวเรื่องที่ผู้ใช้สนใจเท่านั้น

วิธีการหนึ่งที่น่าสนใจของโฟกัสคราวเลอร์คือตัวกลั่นกรอง (Distiller) วิธีการนี้พัฒนามาจากฮิตส์ (ดังสมการที่ 6, 7) (Kleinberg, 1998) ซึ่งจะทำการค้นหากลุ่มเว็บเพจที่มีค่าฮับสูงจากกลุ่มเว็บเพจที่เกี่ยวข้องกับหัวเรื่องนั้น ดังนั้นยูอาร์แอลจากกลุ่มเว็บเพจที่มีค่าฮับสูงจะถูกเว็บคราวเลอร์นำไปประมวลผลก่อน จากผลการทดลองพบว่าวิธีการนี้ทำให้เว็บคราวเลอร์ยังคงอยู่ในเส้นทางของเว็บเพจที่เกี่ยวข้องได้ อย่างไรก็ตามการคำนวณฮิตส์ต้องมีเว็บกราฟที่มีขนาดใหญ่พอสมควร จึงไม่สามารถคำนวณได้ในขณะที่เว็บคราวเลอร์ทำงานตลอดทุกรอบการทำงานได้ ดังนั้นตัวกลั่นกรองจะถูกตั้งเวลาในการคำนวณให้ตื่นขึ้นมาคำนวณ (Watchdog) ตามเวลาต้องการ

2.4 คอนเท็กซ์โฟกัสคราวเลอร์ (Context Focused Crawler)

พัฒนาขึ้นโดย Diligenti *et al.* (2000) โดยใช้หลักการของตัวจำแนกแบบเบส์เช่นเดียวกัน สิ่งที่แตกต่างกันคือตัวจำแนกจะถูกสอนเพื่อใช้ในการประมาณการระยะทางว่าเว็บเพจที่สนใจมีโอกาสที่จะเป็นเว็บเพจที่เกี่ยวข้องมากเท่าใด เช่น ต้องการหาบทความเกี่ยวกับ “Thailand University” สามารถไปที่โฮมเพจมหาวิทยาลัยที่เกี่ยวข้อง แต่ถ้าโฮมเพจดังกล่าวไม่มีคำสำคัญที่เกี่ยวกับหัวเรื่องที่ต้องการ จะทำให้โฟกัสคราวเลอร์มีโอกาสสูงมากที่จะไม่พบเว็บเพจที่ต้องการ อย่างไรก็ตามถ้ามีการประมาณระยะทางมาก่อนว่าเว็บเพจที่เกี่ยวข้องมีความเป็นไปได้ว่าอยู่ห่างจากโฮมเพจของหน่วยงานราชการเป็นระยะทางอีกประมาณ 2 ลิงค์ ก็น่าที่จะทำให้เว็บคราวเลอร์มีโอกาสที่จะเก็บเว็บเพจที่เกี่ยวข้องได้เพิ่มขึ้น



ภาพที่ 10 ตัวอย่างคอนเท็กซ์กราฟ

การประมาณระยะทางนั้นจะต้องสร้างคอนเท็กซ์กราฟ (Context graph) ที่มีจำนวนเลเยอร์อยู่ L ระดับเริ่มต้นตั้งแต่กลุ่มเว็บเพจเป้าหมาย (Seed page) โดยจะนับเป็นเลเยอร์ลำดับที่ 0 ในคอนเท็กซ์กราฟและเว็บเพจที่ชี้เข้ามาหากลุ่มเว็บเพจเริ่มต้นดังกล่าวจะนับเป็นเลเยอร์ลำดับที่ 1 และกลุ่มของเว็บเพจที่ชี้เข้ามาหาเลเยอร์ลำดับที่ 1 จะนับเป็นเลเยอร์ลำดับที่สองตามลำดับ ในทางปฏิบัติแล้วสามารถหาลิงค์ชี้เข้าของเว็บเพจได้โดยใช้เสิร์ชเอ็นจินดังภาพที่ 10

เว็บเพจที่อยู่ในเลเยอร์เดียวกันจะถูกนำมารวมกัน (Merge context graph) จากนั้นจะสกัดเอาคำของเว็บเพจในแต่ละเลเยอร์ออกมาทำการลบคำหยุดและเปลี่ยนให้เป็นคำรากศัพท์แล้วนำคำมาใส่ในเว็คเตอร์ตามกรรมวิธีการคำนวณของทีเอฟไอดีเอฟ เมื่อได้กลุ่มคำที่ต้องการแล้วจะใช้ตัวจำแนกแบบเบส์คำนวณคำในแต่ละเว็บเพจของแต่ละเลเยอร์ $\Pr(t \mid c_l)$ คือความน่าจะเป็นที่จะปรากฏคำ t ถูกจำแนกอยู่ในคลาส c_l ที่เลเยอร์ลำดับที่ l สำหรับความน่าจะเป็นก่อนจะเป็น $\Pr(c_l) = 1/L$ เมื่อ L คือจำนวนของเลเยอร์ทั้งหมด ส่วนความน่าจะเป็นหลังเมื่อต้องการหาคะแนนความสำคัญของยูอาร์แอลที่ออกจากเว็บเพจ p ใด ๆ จะคำนวณโดยหาความน่าจะเป็นที่ p ถูกจำแนกอยู่ในคลาส c_l นั่นคือ $\Pr(c_l \mid p)$ ตามกฎความน่าจะเป็นของเบส์ การคำนวณจะกระทำในแต่ละเลเยอร์ซึ่งเลเยอร์ที่มีความน่าจะเป็นมากที่สุดคือผู้ชนะ อย่างไรก็ตามค่าความน่าจะเป็นของผู้ชนะจะต้องมากกว่าค่าที่ยอมรับได้ (Threshold) ถ้าน้อยกว่าค่าดังกล่าวจะถูกจำแนกให้อยู่ในกลุ่มของ “เว็บเพจอื่น” (other)

2.5 โฟกัสคราวเลอร์แบบปรับตัวเองได้ (Arbitrary Predicate)

เนื่องจากข้อจำกัดของโฟกัสคราวเลอร์ คือผู้ตั้งค่าระบบจะต้องรู้คำสำคัญของเรื่องที่ตนเองต้องการถูกต้องแน่ชัดจึงจะได้เว็บเพจที่ต้องการ แต่โดยทั่วไปแล้วผู้ใช้อาจไม่ทราบคำสำคัญที่ตนต้องการอย่างแน่ชัด จึงทำให้โฟกัสคราวเลอร์นั้นไม่สามารถหาเว็บเพจที่ต้องการได้อย่างมีประสิทธิภาพ นอกจากนั้นแล้วโฟกัสคราวเลอร์ยังละเลยข้อมูลที่มีความสำคัญเพื่อใช้ในการคาดเดายูอาร์แอล เช่น คำสำคัญที่ปรากฏอยู่ในยูอาร์แอลเป็นต้น ดังนั้น Aggarwal *et al.* (2001) จึงนำเสนอวิธีการออกแบบเว็บคราวเลอร์ให้มีความสามารถในการเรียนรู้เพื่อปรับตัวเองให้มุ่งไปสู่การเก็บเว็บเพจที่มีเนื้อหาตรงต่อความต้องการของผู้ใช้ได้

เว็บคราวเลอร์แบบชาญฉลาดถูกออกแบบมาให้รับข้อมูลเข้าที่มีความหลากหลาย เช่น คำสำคัญของหัวเรื่อง ข้อมูลบางส่วนของเว็บเพจ หรือคำที่มีความเกี่ยวข้องกับหัวเรื่องอยู่บ้าง โดยเว็บคราวเลอร์นี้จะมีความสามารถในการปรับตัวเองเพื่อค้นหาเว็บเพจที่เกี่ยวข้องให้ได้มากยิ่งขึ้น โดยไม่ต้องมีการสอน (Training) เว็บคราวเลอร์ก่อน ในแต่ละรอบของการทำงาน (Crawling cycle) เว็บคราวเลอร์จะปรับปรุงฐานความรู้ (Knowledge base) ให้เรียนรู้เพิ่มขึ้น เพื่อใช้ฐานความรู้นี้เป็นตัวให้คะแนนความสำคัญกับยูอาร์แอลแต่ละตัวในยูอาร์แอลคิว คล้ายกับมนุษย์ที่มีการเรียนรู้ในเรื่องที่ตนเองสนใจเพิ่มขึ้นตามประสบการณ์ที่ผ่านมา สำหรับฐานความรู้นั้นมีทั้งสิ้น 4 ชนิดซึ่งประกอบไปด้วย เนื้อหาจากเว็บเพจต้นทาง (Content based) คำสำคัญที่ปรากฏอยู่ในชื่อเซิร์ฟเวอร์และพาธของยูอาร์แอล (URL token based) รูปแบบการเชื่อมโยงของลิงค์ (Link based) และการเชื่อมโยงจากเว็บเพจที่เป็นไซบลิ้ง (Sibling based) ซึ่งจะได้กล่าวถึงในรายละเอียดต่อไป

กำหนดให้ C คือเหตุการณ์ของเว็บเพจที่เก็บรวบรวมมาแล้วและมีเนื้อหาที่เกี่ยวข้อง ส่วน $P(C)$ คือความน่าจะเป็นที่เหตุการณ์ C จะเป็นเว็บเพจที่มีเนื้อหาที่เกี่ยวข้อง ซึ่งสามารถคำนวณได้จากสัดส่วนของจำนวนเว็บเพจที่มีความเกี่ยวข้องทั้งหมดเทียบกับจำนวนเว็บเพจทั้งหมดในฐานความรู้ และกำหนดให้ E คือเหตุการณ์ต่าง ๆ ที่พบจากเว็บเพจที่กำลังเก็บรวบรวมอยู่ เช่น คำสำคัญที่พบจากเว็บเพจต้นทาง จำนวนของลิงค์ที่มีความเกี่ยวข้อง เป็นต้น ดังนั้นความน่าจะเป็นที่ C จะเป็นเว็บเพจที่เกี่ยวข้องเมื่อเกิดเหตุการณ์ E จะเป็น $P(C \mid E) = (C \cap E) / P(E)$ โดยคำนวณค่าอัตราความน่าสนใจ (Interest ratio) ได้ดังต่อไปนี้ $P(C \mid E) / P(C) = (C \cap E) / P(C) \times P(E)$ หรือเขียนให้สั้นลงจะได้ฟังก์ชั่นของอัตราความน่าสนใจกล่าวคือ $I(C, E) = P(C \mid E) / P(C)$ กำหนดให้ $E_1, ..., E_k$ คือ

เหตุการณ์ต่าง ๆ มีจำนวนทั้งสิ้น k เหตุการณ์และ $\varepsilon = E_1 \cap E_2 ... E_k$ ดังนั้นจะได้ฟังก์ชั่นของอัตราความน่าสนใจของเหตุการณ์ต่าง ๆ นั่นคือ $I(C,\varepsilon) = \prod_{i=1}^{k} I(C,E_i)$

เนื้อหาของเว็บเพจต้นทาง (Content based) จะนำคำสำคัญของหัวเรื่องที่ได้จากเว็บเพจนั้นมาคำนวณหาค่าอัตราความน่าสนใจดังสมการที่ 13 ได้ดังต่อไปนี้

$$I_{content}(C) = \prod_{i \in M ; \sigma(C,Q_i) \geq t} P(C \cap Q_i) / P(C) \times P(Q_i) \qquad (13)$$

เมื่อ $I_{content}(C)$ คืออัตราความน่าสนใจของเนื้อหาในเว็บเพจนั้น C คือกลุ่มของเว็บเพจที่ได้เก็บรวบรวมมาแล้วทั้งหมดและเป็นเว็บเพจที่เกี่ยวข้องในฐานความรู้ $P(C)$ คือความน่าจะเป็นของเว็บเพจ C ที่มีความเกี่ยวข้องกับหัวเรื่องนั้นซึ่งหาได้จากฐานความรู้ M คือกลุ่มของคำที่ปรากฏอยู่ในเว็บเพจต้นทางและเป็นคำที่ใช้บ่งชี้ถึงเว็บเพจปลายทาง และ $\sigma(C,Q_i)$ คือฟังก์ชั่นวัดค่าความสำคัญ (Significant function) ของคำ ซึ่งจะต้องมากกว่าค่าที่ยอมรับได้ t (จากการทดลองค่า $t = 2$) ดังนั้นคำที่ไม่มีความสำคัญจะถูกละทิ้งไป

คำสำคัญที่ปรากฏอยู่ในชื่อเซิร์ฟเวอร์และพาธของยูอาร์แอล (URL token based) กล่าวคือถ้าคำที่ปรากฏอยู่ในส่วนประกอบของยูอาร์แอลเป็นคำที่ทำให้อัตราความน่าสนใจสูง ก็จะทำให้ได้เว็บเพจถัดไปมีความเกี่ยวข้องสูงตามไปด้วย โดยกำหนดให้ M คือเซตของคำสำคัญที่ปรากฏอยู่ในส่วนประกอบของยูอาร์แอลนั้น ได้แก่ ชื่อเซิร์ฟเวอร์และพาธของยูอาร์แอล เป็นต้น ดังนั้นสมการในการคำนวณอัตราความน่าสนใจจะใช้สมการที่ 13 เช่นเดียวกัน

การเชื่อมโยงของลิงค์ (Link based) อยู่บนพื้นฐานของสมมติฐานที่ว่าเว็บเพจที่มีเนื้อหาเดียวกันมักจะมีลิงค์เชื่อมโยงกัน (Topical locality) เช่นเดียวกับสมมติฐานของโฟกัสคราวเลอร์ โดยเขียนออกมาเป็นสมการที่ 14 ได้ดังต่อไปนี้

$$I_{link}(C) = \left(\frac{N_{pp}}{N_l \times P(C) \times P(C)}\right) \times \left(\frac{N_{np}}{N_l \times (1 - P(C)) \times P(C)}\right) \qquad (14)$$

เมื่อ $I_{link}(C)$ คืออัตราความน่าสนใจของลิงค์ที่อยู่ในเว็บเพจนั้น N_{pp} คือจำนวนของลิงค์ที่เว็บเพจต้นทางเป็นเว็บเพจที่เกี่ยวข้องและชี้ไปยังเว็บเพจปลายทางที่เกี่ยวข้องในฐานความรู้ ส่วน

N_{np} คือจำนวนของลิงค์ที่เว็บเพจต้นทางเป็นเว็บเพจที่ไม่เกี่ยวข้องแต่ชี้ไปยังเว็บเพจปลายทางที่เกี่ยวข้องในฐานความรู้ N_l คือจำนวนลิงค์ทั้งหมดในฐานความรู้ โดยกำหนดให้ $P(C)\times P(C)$ ถ้าเว็บเพจต้นทางและชี้ไปยังเว็บเพจปลายทางที่มีความเกี่ยวข้องทั้งคู่ ส่วน $1-P(C)\times P(C)$ ถ้าเว็บเพจต้นทางไม่มีความเกี่ยวข้องและชี้ไปยังเว็บเพจปลายทางมีความเกี่ยวข้องกับหัวเรื่องนั้น

การเชื่อมโยงจากเว็บเพจที่เป็นไซบลิ้ง (Sibling based) คือ ถ้าเว็บเพจต้นทางใด ๆ มีลิงค์ชี้ไปยังเว็บเพจอื่นที่มีเนื้อหาเกี่ยวข้องกัน ดังนั้นลิงค์อื่น ๆ ที่ออกจากเว็บเพจต้นทางเดียวกันนั้นก็น่าที่จะชี้ไปยังเว็บเพจที่มีเนื้อหาเกี่ยวข้องกันตามไปด้วย เราจะเรียกเว็บเพจอื่น ๆ ซึ่งเป็นลูกของเว็บเพจต้นทางนั้นว่า “ไซบลิ้ง” ดังนั้นอัตราความน่าสนใจของเว็บเพจที่เป็นไซบลิ้งสามารถคำนวณหาได้จากสมการที่ 15 ดังต่อไปนี้

$$I_{sibling}(C)=\frac{N_p}{N_s\times P(C)} \tag{15}$$

เมื่อ $I_{sibling}(C)$ คืออัตราความน่าสนใจของเว็บเพจที่เป็นไซบลิ้ง N_p คือจำนวนของไซบลิ้งเว็บเพจและเป็นเว็บเพจที่เกี่ยวข้องส่วน N_s คือจำนวนเว็บเพจที่เป็นไซบลิ้งทั้งหมดในฐานความรู้

จากนั้นทำการรวมสมการทั้งหมดเข้าด้วยกันด้วยค่าถ่วงน้ำหนักค่าหนึ่ง และใช้ log เป็นฟังก์ชั่นทางคณิตศาสตร์สำหรับปรับค่าตัวเลขทั้งหมดให้เป็นทศนิยมในรูปแบบของลอการิทึม ข้อมูลที่ได้จากการทำงานแต่ละรอบจะถูกเพิ่มเติมลงไปในฐานความรู้เพื่อปรับให้การงานในรอบถัดไปของเว็บคราวเลอร์สามารถเก็บเว็บเพจที่เกี่ยวข้องได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

2.6 เว็บคราวเลอร์แบบเรียนรู้ได้ (Learnable Web Crawler)

พัฒนาในปี 2002 โดย Rungsawang and Angkawattanawit (2002) โดยมุ่งประเด็นไปในการทำงานของเว็บคราวเลอร์เมื่อผู้ใช้สั่งให้ทำงานในครั้งถัดไป เนื่องจากการเก็บเว็บเพจ เพียงครั้งเดียวไม่อาจครอบคลุมทั่วถึงกับปริมาณเว็บเพจทั้งหมดในอินเทอร์เน็ตได้ โดยจะนำฐานความรู้ที่เว็บคราวเลอร์เรียนรู้มาจากการทำงานครั้งก่อนหน้ามาใช้ประโยชน์ในการทำงานครั้งถัดไป ดังนั้นการทำงานของเว็บคราวเลอร์ในรอบถัดไปจะมีประสิทธิภาพในการเก็บเว็บเพจที่เกี่ยวข้องได้สูงขึ้น

ฐานความรู้ที่ใช้ประกอบไปด้วยยูอาร์แอลเริ่มต้น คำสำคัญของหัวเรื่อง และการคาดเดายูอาร์แอล โดยในรอบถัดไปการหายูอาร์แอลเริ่มต้นจะหามาจากเว็บเพจที่มีค่าฮับสูงซึ่งน่าจะเป็นกลุ่มเว็บเพจที่ชี้ไปยังเว็บเพจที่เกี่ยวข้องได้จำนวนมาก ดังนั้นกลุ่มของยูอาร์แอลเริ่มต้นที่ได้จะเป็นลำดับของเว็บเพจที่มีค่าฮับสูงไล่เรียงลงมายังค่าที่ต่ำกว่าตามลำดับ

กำหนดให้ p คือเว็บเพจและกำหนดให้ $R(p)$ คือฟังก์ชั่นสำหรับใช้ในการตัดสินว่าเว็บเพจ p มีความเกี่ยวข้องกับหัวเรื่องที่ผู้ตั้งค่าระบบต้องการหรือไม่ โดยฟังก์ชั่นดังกล่าวจะนำเนื้อหาของเว็บเพจไปเปรียบเทียบความคล้ายคลึงกับกลุ่มของยูอาร์แอลหรือแองเคอร์เท็กซ์ที่ถูกจัดแบ่งหมวดหมู่ (Category) ไว้แล้วในดีมอส (DMOZ, 1998) ดังนั้นผลการตัดสินของฟังก์ชั่น $R(p)$ จะมีสองค่าคือ p จะมีความเกี่ยวข้อง ถ้าเนื้อหาของเว็บเพจมีความคล้ายคลึงกับยูอาร์แอลหรือแองเคอร์เท็กซ์ในหมวดหมู่ดังกล่าว และ p ไม่มีความเกี่ยวข้อง ถ้าไม่สามารถหาความคล้ายคลึงระหว่างเนื้อหาของเว็บเพจ p กับยูอาร์แอลหรือแองค์เคอร์เท็กซ์ตั้งแต่รูท (root) จากหมวดหมู่ของหัวเรื่องที่กำหนดจนถึงยูอาร์แอลสุดท้ายที่เป็นลีฟ (leaf) ในหมวดหมู่นั้นได้

ในการหาคำสำคัญของหัวเรื่องเมื่อเว็บเพจ p ใด ๆ ถูก $R(p)$ ตัดสินว่ามีความเกี่ยวข้องจะนำคำสำคัญที่อยู่ภายในแท็กไตเติ้ล (Title) ของเว็บเพจนั้นเพิ่มลงไปในฐานความรู้ ส่วนคำสำคัญในแองค์เคอร์เท็กซ์จะถูกเพิ่มลงไปในฐานความรู้ด้วยเช่นเดียวกัน

ส่วนวิธีในการคาดเดายูอาร์แอล เมื่อเว็บเพจใด ๆ ถูกตัดสินโดย $R(p)$ ว่าเว็บเพจ p เป็นเว็บเพจที่ไม่มีความเกี่ยวข้องกับหัวเรื่อง ดังนั้นจะต้องนำยูอาร์แอลของเว็บเพจนั้นมาคำนวณโดยใช้สมการความคล้ายคลึงระหว่างหัวเรื่องและเนื้อหาของแองเคอร์เท็กซ์ แล้วนำค่าคะแนนดังกล่าวมาเป็นค่าคะแนนความสำคัญสำหรับยูอาร์แอลต่อไป

3. การวัดผล (Evaluation Method)

จากการศึกษาค้นคว้าพบว่าแนวทางวัดความเกี่ยวข้องของเว็บเพจมี 2 วิธีได้แก่ การวัดด้วยวิธีการทางคอมพิวเตอร์ และการวัดด้วยผู้เชี่ยวชาญ

3.1 การวัดด้วยเทคนิคทางคอมพิวเตอร์

3.1.1 การใช้ตัวจำแนก (Classifier) เป็นการนำเอาตัวจำแนกมาช่วยในการหาคำตอบว่าเว็บเพจที่เก็บมาได้มีความเกี่ยวข้องหรือไม่ อย่างไรก็ตามผลลัพธ์และความแม่นยำที่ได้จากตัวจำแนกจะขึ้นอยู่กับตัวอย่างที่นำมาสอนตัวจำแนกนั้น คือ ตัวอย่างที่ถูก (Positive sample) และตัวอย่างที่ผิด (Negative sample) เป็นต้น ตัวอย่างงานวิจัยที่ใช้วิธีนี้คือ Menzer *et al.* (2001) ซึ่งผลที่ได้จากการวัดมักจะเป็นค่าสัดส่วนระหว่างเว็บเพจที่มีหัวเรื่องที่กำหนดเทียบกับเว็บเพจทั้งหมด

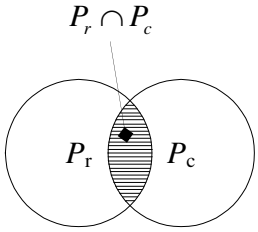
3.1.2 ใช้ระบบสืบค้นข้อมูล (Search Engine) เป็นการเอาระบบสืบค้นข้อมูลมาช่วยในการวัดผล โดย Menzer *et al.* (2001) ได้นำเอาระบบสืบค้นข้อมูลชื่อว่า SMART (Sallton, 1997) มาใช้ โดยนำเว็บเพจที่ได้มาใส่ในระบบสืบค้นข้อมูล การวัดผลเว็บเพจหนึ่งทำได้โดยการคำนวณอันดับที่ปรากฏในผลลัพธ์เปรียบเทียบกับเว็บเพจที่เก็บมาได้ในช่วงเวลานั้น อย่างไรก็ตามความแม่นยำของการวัดผลขึ้นอยู่กับระบบสืบค้นข้อมูลเป็นหลัก

3.2 การวัดด้วยผู้เชี่ยวชาญ

การวัดโดยใช้ผู้เชี่ยวชาญเป็นวิธีการวัดผลที่ให้คำตอบได้แม่นยำที่สุด แต่การวัดแบบนี้มักเสียค่าใช้จ่ายสูงและใช้เวลาในการวัดมาก รวมไปถึงผู้เชี่ยวชาญต้องมีความสมัครใจในการวัดด้วย Rennie and McCallum (1999) ใช้เว็บคราวเลอร์เก็บรวบรวมเอกสารงานวิจัยจากอินเทอร์เน็ต โดยให้อาสาสมัครช่วยกันตรวจสอบว่าข้อมูลที่เก็บมาเป็นงานวิจัยหรือไม่ จะเห็นว่าการใช้ผู้เชี่ยวชาญทำให้ได้ข้อมูลที่แม่นยำ อย่างไรก็ตามหากเปลี่ยนเป็นเนื้อหาอื่นทำให้ต้องใช้ผู้เชี่ยวชาญในสาขาอื่น จึงเป็นการสิ้นเปลืองและทำได้ยากขึ้น วิธีการหนึ่งคือการสร้างระบบหัวเรื่องโดยมนุษย์ เช่น ดีมอส ซึ่งไดเร็กทอรี่ที่ถูกสร้างมาโดยคนที่มีความเชี่ยวชาญในแต่ละสาขา ดังนั้นการนำเว็บเพจมาทำการตรวจสอบกับดีมอสจึงเป็นวิธีที่ง่ายในการวัดผลว่าเว็บเพจที่เก็บมาได้มีความเกี่ยวข้องกับหัวเรื่องที่กำหนดไว้หรือไม่

4. มาตรวัด (Evaluation Matrices)

ในการวัดผลประสิทธิภาพของเว็บคราวเลอร์แบบเฉพาะเจาะจงหัวเรื่องนั้นงานวิจัยส่วนใหญ่นิยมที่ใช้มาตรวัดอยู่ 2 ชนิดได้แก่ อัตราการเก็บเกี่ยว (Harvest rate) และความครอบคลุม (Coverage) จากภาพที่ 12 กำหนดให้ P_r คือจำนวนของเว็บที่เกี่ยวข้อง และ P_c คือจำนวนเว็บเพจที่เว็บคราวเลอร์เก็บมาแล้วตามลำดับ ดังนั้นสามารถอธิบายความหมายของแต่ละมาตรวัดได้ดังนี้



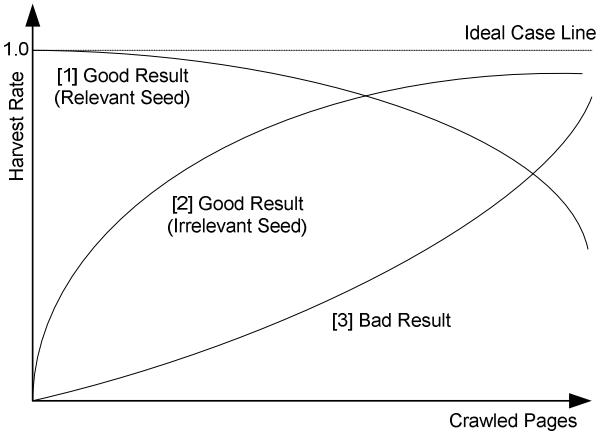
ภาพที่ 11 กลุ่มของเว็บเพจที่เกี่ยวข้อง Pr และกลุ่มของเว็บเพจที่เก็บมาแล้ว Pc

4.1 อัตราการเก็บเกี่ยว (Harvest rate) คือสัดส่วนของเว็บเพจที่เว็บคราวเลอร์เก็บมาแล้วและเป็นเว็บเพจที่เกี่ยวข้องเมื่อเทียบกับจำนวนเว็บเพจทั้งหมดที่เว็บคราวเลอร์เก็บมาทั้งหมด งานวิจัยส่วนใหญ่มักใช้มาตรวัดนี้เป็นพื้นฐานในการวัดเพื่อแสดงให้เห็นว่าเว็บคราวเลอร์ยังเก็บเว็บเพจที่เกี่ยวข้องได้ตลอดเวลาหรือไม่โดยแสดงเป็นสมการได้ดังนี้

$$\text{harvest rate} = \frac{P_r \cap P_c}{P_c} \tag{16}$$

กราฟของอัตราการเก็บเกี่ยวประกอบด้วยแกนนอน (X) คือจำนวนของเว็บเพจที่เว็บคราวเลอร์เก็บมาแล้วตามช่วงเวลาซึ่งเป็นจำนวนเต็มบวกที่มีค่าเท่าไรก็ได้ ส่วนแกนตั้ง (Y) คือค่าอัตราการเก็บเกี่ยวของเว็บเพจที่เก็บมาได้โดยคำนวณจากสมการที่ 16 ซึ่งมีค่าสูงสุดคือ 1.0

จากภาพที่ 12 จะเห็นว่าเว็บคราวเลอร์แบบเฉพาะเจาะจงหัวเรื่องที่มีประสิทธิภาพที่สมบูรณ์แบบที่สุดนั้น จะต้องเป็นไปตามเส้นในอุดมคติ นั่นคือทุกเว็บเพจที่เว็บคราวเลอร์เก็บมาได้จะเป็นเว็บเพจที่เกี่ยวข้องกับหัวเรื่องเสมอ แต่ในความเป็นจริงแล้วยังมีการทำงานที่ผิดพลาดอยู่บ้างและเมื่อเก็บเว็บเพจไปแล้วเป็นจำนวนมากแนวโน้มในการค้นพบเว็บเพจที่มีความเกี่ยวข้องจะน้อยลงตามลำดับดังกราฟเส้นที่ [1]



ภาพที่ 12 ลักษณะของกราฟอัตราการเก็บเกี่ยว

ดังนั้นจึงสามารถสรุปได้ว่าลักษณะของกราฟที่ดีควรจะมีแนวโน้มของกราฟใกล้เคียงกับเส้นที่ [1] และ [2] กล่าวคือในช่วงเริ่มต้นในการเก็บควรจะเก็บเว็บเพจที่เกี่ยวข้องในปริมาณที่มากเป็นเหมือนดังเส้นที่ [1] เมื่อเริ่มจากกลุ่มยูอาร์แอลเริ่มต้นที่มีความเกี่ยวข้องกับหัวเรื่อง และเส้นที่ [2] เมื่อเริ่มจากกลุ่มของยูอาร์แอลที่ไม่มีความเกี่ยวข้อง เมื่อเปรียบเทียบกับกราฟในเส้นที่ [3] ซึ่งเป็นลักษณะของกราฟที่ไม่ดีจะมีเหตุผลในการเปรียบเทียบอยู่ 2 ประการคือ

ประการแรก คือ บางครั้งเว็บคราวเลอร์ถูกจำกัดด้วยจำนวนเว็บเพจที่ต้องการจัดเก็บเนื่องจากเหตุปัจจัยหลายประการเช่น พื้นที่ดิสก์ไม่เพียงพอ หรือกำหนดเวลาในการจัดเก็บ เป็นต้น ในขณะที่เว็บเพจที่เก็บมาได้จำนวนมากในช่วงเริ่มต้นกลับไม่มีความเกี่ยวข้องกับหัวเรื่องที่ต้องการ

ประการที่สอง คือ สัดส่วนของเว็บเพจที่เว็บคราวเลอร์เก็บได้ แต่ส่วนใหญ่เป็นเว็บเพจที่ไม่มีความเกี่ยวข้องกับหัวเรื่องที่กำหนด ดังนั้นย่อมหมายถึงการเสียทรัพยากรคอมพิวเตอร์ไปโดยไม่ได้รับผลตอบแทนที่คุ้มค่า เช่น เว็บคราวเลอร์ต้องเสียการทำงานของซีพียู การใช้หน่วยความจำและพื้นที่ของดิสก์สำหรับในการจัดเก็บ

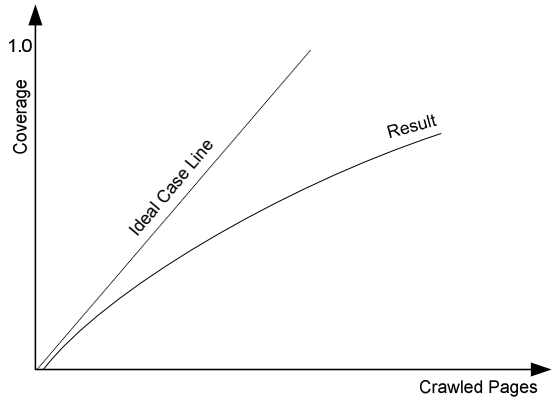
4.2 ความครอบคลุม (Coverage) คือ สัดส่วนของเว็บเพจที่เว็บคราวเลอร์เก็บมาแล้วและเป็นเว็บเพจที่เกี่ยวข้องเมื่อเทียบกับจำนวนของเว็บเพจที่เกี่ยวข้องที่อยู่ในฐานข้อมูลเว็บเพจ (หรือฐานข้อมูลเว็บกราฟ) มีงานวิจัยเพียงจำนวนหนึ่งเท่านั้นที่ใช้มาตรวัดแบบนี้ เนื่องจากไม่ทราบจำนวนที่แท้จริงของเว็บเพจที่เกี่ยวข้องทั้งหมดที่อยู่บนอินเทอร์เน็ตได้ ดังนั้นการวัดผลจึงต้องวัดใน

ฐานข้อมูลเว็บเพจที่เก็บมาแล้วเท่านั้น มาตรวัดนี้แสดงถึงประสิทธิภาพว่าเว็บคราวเลอร์เก็บเว็บเพจที่เกี่ยวข้องได้ครบหรือไม่ ซึ่งเขียนเป็นสมการที่ 17 ได้ดังนี้

$$\text{coverage} = \frac{P_r \cap P_c}{P_r} \tag{17}$$

จากภาพที่ 13 จะเห็นว่าเส้นในอุดมคติเป็นเส้นทแยงมุม โดยความชันของเส้นขึ้นอยู่กับจำนวนเว็บเพจที่เกี่ยวข้องเมื่อเทียบกับจำนวนเว็บเพจทั้งหมดในฐานข้อมูลเว็บเพจ เช่น สมมติว่ามีเว็บเพจทั้งหมดในฐานข้อมูล 100 เว็บเพจแบ่งเป็นเว็บเพจที่เกี่ยวข้อง 50 เว็บเพจ ดังนั้นความชันของเส้นในอุดมคติจะมีค่าความครอบคลุมเป็น 1.0 (100%) เมื่อเว็บคราวเลอร์เก็บเว็บเพจครบ 50 เว็บเพจและเป็นเว็บเพจที่เกี่ยวข้องทั้งหมดเป็นต้น

ดังนั้นวิธีการอ่านและวิเคราะห์กราฟความครอบคลุม จะดูว่าเส้นกราฟที่ได้ใกล้เคียงกับเส้นในอุดมคติมากเพียงไร หากว่ากราฟที่ได้ใกล้กับเส้นในอุดมคติมากแสดงว่าเว็บคราวเลอร์พบเว็บเพจที่มีความเกี่ยวข้องในฐานข้อมูลในปริมาณที่มาก



ภาพที่ 13 ลักษณะของกราฟความครอบคลุม

การค้นหาเว็บเพจแบบเฉพาะเจาะจงภาษา

ปัจจุบันการสร้างระบบสืบค้นข้อมูลที่สามารถค้นคืนข้อมูลโดยให้ผลลัพธ์ครอบคลุมกับผู้ใช้ที่เกี่ยวข้องกับประเทศหรือท้องถิ่นนั้นกำลังได้รับความนิยมเพิ่มมากขึ้น ดังนั้นนักวิจัยในหลายประเทศจึงมีความพยายามที่จะเก็บรวบรวมเว็บเพจที่เขียนขึ้นโดยใช้ภาษาของประเทศตนเองส่วนหนึ่งจากอินเตอร์เน็ตเพื่อนำมาสร้างเป็นระบบสืบค้นข้อมูล ห้องสมุดอิเล็กทรอนิกส์ รวมไปถึงระบบการจัดเก็บข้อมูลเว็บเพจขนาดใหญ่มาก (Very large-scale web archive) เช่น ระบบฐานข้อมูลห้องสมุดแห่งชาติของประเทศสวีเดน (Kulturarw3, 1997) ระบบสืบค้นข้อมูลภาษาไทยสรรสาร (Sansarn, 2001) เป็นต้น ซึ่งจะเรียกการเก็บรวบรวมเว็บเพจในลักษณะนี้เรียกวา “การเก็บเว็บเพจแบบเฉพาะเจาะจงภาษา” และเว็บคราวเลอร์ที่ใช้ในการเก็บรวบรวมเว็บเพจดังกล่าวเรียกว่า “เว็บคราวเลอร์แบบเฉพาะเจาะจงภาษา” (Language-specific web crawler)

แบบจำลองของเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาจะคล้ายกับแบบจำลองของเว็บคราวเลอร์แบบเฉพาะเจาะจงหัวเรื่องดังภาพที่ 7 สิ่งที่แตกต่างคือโดยตัววิเคราะห์เว็บเพจจะแบ่งออกเป็นส่วนในการวิเคราะห์ภาษาในของเว็บเพจซึ่งจะกล่าวถึงในหัวข้อการตรวจสอบภาษาในเว็บเพจ และวิธีการในการให้คะแนนความสำคัญแก่ยูอาร์แอลเพื่อค้นหาเว็บเพจที่เป็นภาษาที่ต้องการ ซึ่งจะกล่าวถึงในหัวข้องานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้อง

Koht-arsa and Sanguanpong (2001, 2002) ได้เสนอวิธีการสร้างเว็บคราวเลอร์ที่มีสมรรถนะสูงเรียกว่าเคสไปเดอร์ (Kspider) โดยทดสอบประสิทธิภาพสมรรถนะของเว็บคราวเลอร์ในการค้นหาเว็บเพจที่อยู่ภายใต้อินเตอร์เน็ตโดเมน ".th" นอกจากนั้นแล้วยังนำข้อมูลของเว็บเพจที่เก็บมาแล้วมาวิเคราะห์ทางสถิติเพื่อแสดงเป็นข้อมูลลักษณะของเว็บเพจภาษาไทย

Tongchim *et al*. (2006) ได้เสนอวิธีการสร้างเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาโดยใช้ตัวตรวจสอบภาษาที่เป็นโอเพ่นซอร์สซอร์ฟแวร์ซึ่งเรียกว่าลิบส์ (LIBS) ที่พัฒนาขึ้นบนพื้นฐานของอัลกอริทึมสตริงเคอร์เนลซึ่งนำเสนอโดย Kruengkrai *et al*. (2005) ทำหน้าที่ตรวจสอบภาษาบนเว็บเพจ เว็บคราวเลอร์ดังกล่าวถูกออกแบบมาให้ทำงานร่วมกันหลายตัวเชื่อมต่อด้วยระบบเครือข่ายความเร็วสูง โดยจะใช้แฮชฟังก์ชั่นและอัลกอริทึมในการหาเว็บคราวเลอร์ที่ตั้งอยู่ใกล้ที่สุดของยูอาร์แอลนั้น ในการแบ่งกระจายภาระให้เว็บคราวเลอร์แต่ละตัว อย่างไรก็ตามวิธีการดังกล่าวยังค้นหาและเก็บรวบรวมเว็บเพจแบบเจาะจงอินเทอร์เน็ตโดเมน ".th" ของประเทศไทยและ ".jp" ของประเทศญี่ปุ่นเท่านั้น ซึ่งจากงานวิจัยของ Somboonviwat *et al*. (2005, 2006) กล่าวว่าปริมาณเว็บเพจภาษาไทยเป็นจำนวนมากกว่าครึ่งหนึ่งของเว็บเพจภาษาไทยทั้งหมด จะอยู่นอกเหนือจากอินเทอร์เน็ตโดเมน "th" ดังนั้นการค้นหาแบบจำกัดอินเทอร์เน็ตโดเมนดังกล่าว จึงไม่อาจครอบคลุมต่อจำนวนเว็บเพจภาษาไทยที่มีอยู่ทั้งหมดได้

Somboonviwat *et al*. (2005, 2006) ได้พัฒนาวิธีการค้นหาเว็บเพจภาษาไทยโดยใช้การตรวจสอบภาษาบนเว็บเพจด้วยวิธีเข้ารหัสอักษรและเท็กซ์แคท (Textcat) ซึ่งถูกพัฒนาโดย Cavnar and Trenkle (1994) ควบคู่กันไปด้วย และได้ศึกษาลักษณะของเว็บเพจภาษาไทยในเว็บกราฟที่เก็บตัวอย่างมาวิเคราะห์พบว่าเว็บเพจต้นทางที่เป็นเว็บเพจภาษาไทยมักชี้ไปยังเว็บเพจปลายทางที่เป็นเว็บเพจภาษาไทย (Language locality) อีกทั้งยังพบเว็บเพจภาษาไทยจำนวนมากภายในเซิร์ฟเวอร์หรือโดเมนเดียวกัน อีกทั้งยังพบว่าจะมีโอกาสพบเจอเว็บเพจภาษาไทยเป็นจำนวนมากจากเว็บเพจภาษาอื่นภายในระยะทางที่ห่างจากเว็บเพจภาษาไทยต้นทาง (distance) ค่าหนึ่ง อย่างไรก็ตามข้อสังเกตของงานวิจัยนี้คือไม่ได้ใช้ค่าคะแนนของภาษาที่ปรากฏอยู่ในแองเคอร์เท็กซ์ซึ่งเป็นข้อมูลที่สำคัญมากในการคาดเดาเว็บเพจภาษาไทยที่จะเก็บต่อไปได้

อุปกรณ์และวิธีการ

อุปกรณ์

1. ซอฟท์แวร์
    1. ระบบปฏิบัติการเซ็นต์โอเอสลีนุกซ์ (CentOS) เวอร์ชั่น 5.2
    2. จาวาเวอร์ชั่น 6.0
    3. ระบบจัดการฐานข้อมูล MySQL เวอร์ชั่น 5.0
    4. ระบบฐานข้อมูลเบิร์กเลย์ (Berkeley DB)
    5. โปรแกรมอีคลิปส์เวอร์ชั่น 3.3 (Eclipse 3.3)

2. ฮาร์ดแวร์
    1. เครื่องคอมพิวเตอร์ความเร็ว 2.4 GHz
    2. หน่วยความจำหลัก 1 GHz
    3. ฮาร์ดดิสก์ความจุ 80 GB
    4. การ์ดเครือข่าย 10/100 Mpbs 1 ตัว

วิธีการ

1. การตรวจสอบภาษาไทยในเว็บเพจ

การที่เราจะบอกว่าเว็บเพจนี้เป็นเว็บเพจภาษาไทยที่ต้องการหรือไม่ จะต้องมีกระบวนการสำหรับตรวจสอบภาษาไทยในเว็บเพจนั้น กระบวนการดังกล่าวนี้ถือได้ว่ามีความสำคัญสำหรับเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาไทยเป็นอย่างมาก เพราะหากการตรวจสอบภาษาไทยไม่แม่นยำและมีความผิดพลาดสูงจะส่งผลให้เว็บเพจที่เก็บรวบรวมมาจะมีเว็บเพจภาษาอื่นปะปนมาเป็นจำนวนมากอีกด้วย ในหัวข้อนี้จะกล่าวถึงวิธีการตรวจสอบภาษาไทยในเว็บเพจอย่างละเอียดซึ่งแบ่งออกเป็น 2 ตอนได้แก่ ขั้นตอนการเตรียมเนื้อหาก่อนการตรวจสอบและวิธีในการตรวจสอบภาษา ซึ่งจะแบ่งแยกย่อยออกเป็นอีก 4 วิธี คือ วิธีการตรวจสอบจากการเข้ารหัสตัวอักษร วิธีเท็กซ์แคท วิธี

สตริงเคอร์เนล และวิธีการใช้เล็กซ์โต สำหรับหัวข้อสุดท้ายเราจะกำหนดนิยามของเว็บเพจภาษาไทยเพื่อที่จะใช้ในการทดลองของวิทยานิพนธ์เล่มนี้โดยเราจะเลือกวิธีใช้วิธีการเล็กซ์โตเป็นหลักซึ่งจะได้กล่าวลงในรายละเอียดต่อไป

1.1 ขั้นตอนการเตรียมเนื้อหาก่อนการตรวจสอบ (Document Pre-Processing)

เราจะอ่านเว็บเพจจากแหล่งที่เก็บ (Document repository) โดยจะอ่านทีละไบต์พร้อมกับทำการเปลี่ยนจากไบต์ให้เป็นอักษร ตามการเข้ารหัสตัวอักษรที่ปรากฏอยู่ในเมต้าแอทริบิวต์ (Meta Charset) ของเว็บนั้น สำหรับในบางเว็บเพจอาจไม่พบแอทริบิวต์ดังกล่าว ดังนั้นเราจะใช้โปรแกรมสำหรับตรวจสอบการเข้ารหัสอักษร (CPDetector, 2008) ซึ่งใช้สำหรับการคาดเดาการเข้ารหัสอักษรจากเว็บเพจนั้น จากนั้นเราจะส่งเนื้อหาเว็บเพจมาให้ตัวคัดแยกเอชทีเอ็มแอล (Jericho, 2004) เพื่อแยกส่วนประกอบของเว็บเพจออกเป็นแต่ละส่วน เช่น ส่วนที่เป็นไตเติ้ล (Title) เนื้อหาหลัก (Content) ยูอาร์แอลและแองเคอร์เท็กซ์ เป็นต้น จากนั้นจะนำเนื้อหาของเว็บเพจที่คัดแยกมาแล้วมาทำความสะอาดโดยลบอักษรพิเศษออกไปและจะส่งผ่านเนื้อหาที่สมบูรณ์แล้วให้กับขั้นตอนการตรวจสอบเนื้อหาต่อไป

1.2 ขั้นตอนการตรวจสอบภาษาจากเนื้อหาของเว็บเพจ (Language Identification)

คือขั้นตอนการนำเนื้อหาที่ได้จากหัวข้อที่แล้วมาตรวจสอบเพื่อระบุว่าเว็บเพจนี้เป็นเว็บเพจที่เขียนขึ้นโดยใช้ภาษาใด ในที่นี้เราได้นำเสนอวิธีการอยู่ 4 วิธีดังต่อไปนี้ การตรวจสอบภาษาจากการเข้ารหัสเว็บเพจ วิธีการนี้จะตรวจสอบที่การเข้ารหัสอักษรของเว็บเพจเท่านั้นดังนั้นจะไม่ต้องผ่านขั้นตอนการเตรียมเนื้อหาจากหัวข้อที่แล้วก่อนก็ได้ วิธีตรวจสอบภาษาโดยใช้เท็กซ์แคท วิธีตรวจสอบภาษาโดยใช้สตริงเคอร์เนล และวิธีตรวจสอบภาษาไทยโดยใช้เล็กซ์โต ซึ่งรายละเอียดแต่ละวิธีการจะกล่าวถึงในหัวข้อถัดไป

1.2.1 ตรวจสอบภาษาจากการเข้ารหัสเว็บเพจ

วิธีการตรวจสอบภาษาไทยที่ง่ายที่สุดคือการตรวจสอบการเข้ารหัสตัวอักษร (Character set) ของเว็บเพจที่อยู่ภายใต้เมต้าแท็ก (Meta tag) จากมาตรฐานขององค์กรระหว่างประเทศที่พัฒนาด้านเทคโนโลยีเว็บ (W3C: http://www.w3.org/) ได้กำหนดมาตรฐานการเข้ารหัสอักษรที่สัมพันธ์กับเว็บเพจนั้นๆ เช่น Windows-874 TIS-620 และ ISO-8859-11 เป็นการเข้ารหัสอักษรของเว็บเพจภาษาไทยเป็นต้น อย่างไรก็ตามมีการเข้ารหัสอักษรบางชนิดซึ่งจะไม่สามารถระบุว่าเป็นเว็บเพจภาษาไทยได้ นั่นคือ UTF-8 เป็นต้น ดังนั้นวิธีการตรวจสอบภาษาดังกล่าวยังให้ผลที่ไม่ครอบคลุมต่อเว็บเพจภาษาไทยทั้งหมดมากนัก

1.2.2 วิธีตรวจสอบภาษาโดยใช้เท็กซ์แคท

วิธีการตรวจสอบภาษาโดยใช้เท็กซ์แคท (Text categorization) นำเสนอโดย Cavnar and Trenkle (1994) โดยมีวัตถุประสงค์เพื่อใช้ในการจัดการเอกสารกลุ่มข่าว (News group) อย่างไรก็ตามเราสามารถนำแบบจำลองนี้มาประยุกต์ในการตรวจสอบภาษาได้ เท็กซ์แคทนั้นถูกสร้างขึ้นอยู่บนพื้นฐานของแบบจำลองเอ็นแกรม (N-gram) ซึ่งเป็นวิธีในการแบ่งคำที่ปรากฏอยู่ในเอกสารด้วยอักขระจำนวน n ตัว เช่นถ้าเรามีคำว่า “CRAWL” และกำหนดให้ช่องว่างแทนด้วย “_” ดังนั้นเราสามารถแบ่งคำตามวิธีเอ็นแกรมได้ดังนี้

หนึ่งแกรม ($n = 1$) C, R, A, W, L, _
สองแกรม ($n = 2$) CR, RA, AW, WL, L_
สามแกรม ($n = 3$) CRA, RAW, AWL, WL_, L__

เริ่มต้นเราจะสร้างโพรไฟล์ (Profile) ต้นแบบของเอกสารภาษาต่าง ๆ ซึ่งโพรไฟล์คือชุดข้อมูลที่ถูกสกัดออกมาจากเอกสารตัวอย่างหลายเอกสารและหลาย ๆ ภาษา โดยแต่ละภาษาจะถูกแบ่งคำออกมาโดยใช้วิธีของเอ็นแกรมดังที่กล่าวมาข้างต้น จากนั้นจะเลือกแกรมที่มีความถี่สูงสุดจำนวน m อันดับแรกออกมา และบันทึกเก็บไว้เป็นไฟล์แยกเป็นไฟล์ละหนึ่งภาษา เช่น thai.lm, english.lm หมายถึงโพรไฟล์ของภาษาไทยและภาษาอังกฤษตามลำดับ สำหรับใช้เป็นชุดข้อมูลอ้างอิงในการเปรียบเทียบกับเอกสารอื่น ๆ ต่อไป

เมื่อต้องการตรวจสอบภาษาไทยของเว็บเพจโดยใช้เท็กซ์แคท เราจะนำเนื้อหาของเว็บเพจนั้นมาแบ่งออกมาเป็นคำโดยใช้วิธีเอ็นแกรม ซึ่งจำนวนของ n ที่จะแบ่งจะต้องเท่ากันกับโพรไฟล์ ต้นแบบที่ได้กล่าวมา จากนั้นเลือกแกรมที่มีความถี่สูงสุดจำนวน m อันดับแรกออกมาเปรียบเทียบระยะทางกันกับโพรไฟล์ของภาษาต่าง ๆ โดยใช้ฟังก์ชั่นระยะทาง (Distance function) ถ้าระยะทางระหว่างเนื้อหาของเว็บเพจนั้นเทียบกับโพรไฟล์ของภาษาใดแล้วน้อยที่สุด จะถือว่าเว็บเพจดังกล่าวเป็นเว็บเพจของภาษานั้นนั่นเอง

1.2.3 วิธีตรวจสอบภาษาโดยใช้สตริงเคอร์เนล

วิธีการตรวจสอบภาษาแบบสตริงเคอร์เนล (String Kernel) นำเสนอโดย Kruengkrai *et al.* (2005) โดยจะแสดงตัวอักษรที่ปรากฏในเอกสารในรูปของสตริงของไบต์ที่ติดกัน ซึ่งจะแตกต่างจากวิธีของเอ็นแกรมที่แสดงในรูปสตริงของตัวอักขระ ดังนั้นด้วยวิธีการนำเสนอแบบสตริงเคอร์เนลนี้ทำให้เอกสารเว็บเพจของแต่ละภาษาสามารถตรวจสอบภาษาได้โดยอิสระต่อการเข้ารหัสตัวอักษรนั่นเอง

ในการตรวจสอบภาษาของเอกสารจะมีวิธีการคล้ายกับเอ็นแกรม กล่าวคือการนำซับสตริงของเอกสารที่ต้องการทดสอบ มาเปรียบเทียบกับซับสตริงของเอกสารเว็บเพจตัวอย่างที่เป็นภาษาต่าง ๆ เรียกว่าการคำนวณสตริงเคอร์เนล โดยเปลี่ยนซับสตริงที่ปรากฏอยู่ในเอกสารนั้นให้เป็นฟีเจอร์เวคเตอร์ (Feature vector) ที่มีความยาวตั้งแต่หนึ่งถึง r คล้ายกับวิธีการของแบบจำลองเอ็นแกรม (เมื่อกำหนดให้ $n = r$) ดังนั้นซับสตริงที่เป็นไปได้ทั้งหมดจะถูกแยกออกจากเอกสารนั้นเพื่อสร้างตัวแทนโดยฟีเจอร์เวคเตอร์ แล้วนับจำนวนซับสตริงที่ซ้ำกันระหว่างสองซับสตริงเพื่อดูว่ามีความเหมือนกันมากน้อยเพียงใด หากซับสตริงของเอกสารที่ต้องการทดสอบมีความเหมือนกันที่ใกล้เคียงกับซับสตริงของเอกสารเว็บเพจตัวอย่างของภาษาใด จะถือว่าเป็นเอกสารที่เขียนโดยใช้ภาษานั้นนั่นเอง

1.2.4 ตรวจสอบภาษาไทยโดยใช้เล็กซ์โต

เล็กซ์โต (Lexeme Tokenizer: LexTo) คือโปรแกรมที่ทำหน้าที่สำหรับตัดคำภาษาไทย ซึ่งถูกพัฒนาโดยหน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) เพื่อใช้ในระบบสืบค้นข้อมูลภาษาไทยสรร

สาร (Sansarn, 2001) วิธีการตัดคำนั้นจะใช้การอ้างอิงคำในพจนานุกรมภาษาไทยเล็กซิตรอน (lexitron.txt) เป็นหลัก ซึ่งประกอบไปด้วยคำที่เป็นภาษาไทยจำนวนทั้งสิ้น 35933 คำซึ่งเราสามารถเพิ่มคำใหม่ ๆ เข้าไปอีกก็ได้ โปรแกรมเล็กซ์โตเป็นส่วนหนึ่งของกลุ่มโปรแกรมวิเคราะห์ภาษาไทย (ThaiAnalyzer package) ซึ่งเป็นกลุ่มโปรแกรมที่มีหน้าที่สำหรับจัดการภาษาไทยอันประกอบไปด้วยฟังก์ชั่นที่หลากหลาย อาทิเช่น ฟังก์ชั่นการกรองคำ (Word filter) ฟังก์ชั่นการลบคำหยุด (Word stopper) เป็นต้น เพื่อให้ได้ผลลัพธ์เป็นคำไทยที่สมบูรณ์ที่เหมาะสมในการสร้างเป็นดัชนีสำหรับสืบค้นข้อมูลโดยใช้โปรแกรมอาพาเช่ลูซีน (Apache Lucene, 2001) อีกทีหนึ่ง

ยกตัวอย่างการตัดคำภาษาไทยของเล็กซ์โตเมื่อมีข้อความจากเว็บเพจหนึ่งคือ “กทม. ร่วมสนับสนุนการท่องเที่ยว Amazing Thailand 2009” ผลลัพธ์การตัดคำของเล็กซ์โตจะเป็นดังต่อไปนี้

ตารางที่ 6 ผลลัพธ์การตัดคำโดยใช้เล็กซ์โต

คำที่	คำที่ได้	ชนิดของคำ
1	กทม	Known
2	ร่วม	Known
3	สนับสนุน	Known
4	การ	Known
5	ท่องเที่ยว	Known
6	Amazing	Unknown
7	Thailand	Unknown
8	2009	Unknown

จากตารางที่ 6 พบว่าถ้ามีคำใดปรากฏอยู่ในพจนานุกรมภาษาไทย เล็กซ์โตจะกำหนดให้เป็นคำที่รู้จัก (known) ในทางกลับกันหากคำที่ไม่ปรากฏในพจนานุกรมภาษาไทยจะถูกกำหนดให้เป็นคำที่ไม่รู้จัก (unknown) ดังนั้นผลลัพธ์ของการตัดคำจะขึ้นอยู่กับความครอบคลุมของคำที่อยู่ในพจนานุกรมภาษาไทยเป็นหลัก แต่อย่างไรก็ตามจากผลการทดลองที่ผ่านมาพบว่าวิธีนี้ให้ความแม่นยำอยู่ในระดับที่ดี (ดูผลการทดสอบในตารางที่ 8 ประกอบ)

เราสามารถประยุกต์ใช้เล็กซ์โตเพื่อหาอัตราส่วนความเป็นภาษาไทยของเว็บเพจ p ใดๆได้โดยกำหนดให้ t คือจำนวนของคำที่เล็กซ์โตรู้จักและ N คือจำนวนของคำทั้งหมดที่ได้จากการตัดคำของเล็กซ์โต ดังนั้นการหาอัตราส่วนความเป็นภาษาไทยของเว็บเพจ p แสดงได้ดังสมการที่ 18

$$\Gamma(p) = \frac{t}{N} \tag{18}$$

จากผลลัพธ์ของการตัดคำของเล็กซ์โตในตารางที่ 6 เราสามารถคำนวณหาค่าอัตราส่วนความเป็นภาษาไทยของเว็บเพจนั้นได้คือ $5/8 = 0.625$ หรือ 62.5% นั่นเอง

1.3 นิยามของเว็บเพจภาษาไทย

วิทยานิพนธ์นี้จะเลือกใช้วิธีการตรวจสอบภาษาไทยแบบเล็กซ์โต ดังที่กล่าวมาแล้วในหัวข้อที่ 1.2.4 เนื่องจากให้ผลลัพธ์ในการตรวจสอบภาษาที่ดีกว่าวิธีการอื่น และกำหนดนิยามความเป็นเว็บเพจภาษาไทยสำหรับวิทยานิพนธ์นี้นั่นคือ เว็บเพจ p ใด ๆ จะเป็นเว็บเพจภาษาไทยก็ต่อเมื่อมีอัตราส่วนของความเป็นภาษาไทย $\Gamma(p)$ (จากสมการที่ 18) มากกว่าค่าของ Z ที่กำหนดไว้ สำหรับวิทยานิพนธ์นี้เราจะกำหนดค่า Z ไว้ที่ 0.1 (ภาพผนวกที่1: ผลการปรับค่าที่ใช้ตัดสินว่าเป็นเว็บเพจภาษาไทย) ซึ่งเขียนเป็นสมการสำหรับตัดสินเว็บเพจ p ว่าเป็นเว็บเพจภาษาไทยหรือไม่ $R(p)$ ได้ดังต่อไปนี้

$$R(p) = \begin{cases} \text{Thai} & \text{if } \Gamma(p) \geq Z \\ \text{Other} & \text{otherwise} \end{cases} \tag{19}$$

จากผลการตรวจสอบภาษาโดยใช้เล็กซ์โตจากตัวอย่างในหัวข้อที่ผ่านมามีค่าเป็น 0.625 ดังนั้น $\Gamma(0.625) \geq 0.1$ เป็นความจริงดังนั้นเราจะถือว่า $R(p)$ ให้คำตอบเป็นเว็บเพจภาษาไทย

2. ชุดข้อมูลทดสอบ

เป็นชุดข้อมูลที่ถูกสร้างขึ้นมาเพื่อวัตถุประสงค์ในการทดสอบอยู่ 2 ข้อ กล่าวคือข้อแรกเพื่อใช้ในการทดสอบประสิทธิภาพของตัวตรวจสอบภาษาแต่ละประเภทเราจะเรียกชุดข้อมูลแรกนี้ว่า “ชุดข้อมูลทดสอบตัวตรวจสอบภาษา” (Language detector dataset) และข้อที่สองคือเพื่อใช้ในการ

ทดสอบประสิทธิภาพของอัลกอริทึมเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาไทยที่ได้นำเสนอขึ้นมา ซึ่งถือเป็นเนื้อหาหลักของวิทยานิพนธ์เล่มนี้ โดยเราจะเรียกชุดข้อมูลนี้ว่า “ชุดข้อมูลเว็บกราฟ” (Web graph dataset) รายละเอียดของชุดข้อมูลแต่ละชนิดจะได้กล่าวถึงในหัวข้อต่อไป

2.1 ชุดข้อมูลสำหรับตัวตรวจสอบภาษา (Language detector dataset)

ชุดข้อมูลสำหรับตัวตรวจสอบภาษาประกอบไปด้วยกลุ่มของเว็บเพจหลายภาษาที่ถูกเก็บรวบรวมมาเพื่อนำมาใช้ในการวัดประสิทธิภาพของวิธีการตรวจสอบภาษาแบบต่าง ๆ ดังที่ได้นำเสนอไปแล้วในหัวข้อการตรวจสอบภาษาไทยในเว็บเพจ

2.1.1 การสร้างชุดข้อมูล

การสร้างชุดข้อมูลนี้เริ่มต้นเราจะใช้กูเกิ้ลเสิร์ชเอ็นจิ้น (Google, 2007) และกำหนดยูอาร์แอลให้ค้นหาเว็บเพจในหัวข้อ “University” พร้อมกับกำหนดค่าภาษาลงไปด้วย เช่น ตารางที่ 7 ในที่นี้เราจะใช้ยูอาร์แอลของภาษาทั้งสิ้น 43 ภาษา และกูเกิ้ลจะแสดงรายการผลลัพธ์ที่ได้โดยจะเป็นยูอาร์แอลของเว็บเพจภาษาที่ต่างกันขึ้นอยู่กับค่าของภาษาที่กำหนดลงไปในตอนแรก จากนั้นเราจะเลือกเอายูอาร์แอล 10 อันดับแรกของผลลัพธ์ที่ออกมาเพื่อนำมาสร้างเป็นชุดข้อมูลทดสอบ อย่างไรก็ตามเนื่องจากเราเน้นที่การตรวจสอบภาษาไทยเป็นหลัก ดังนั้นยูอาร์แอลที่เป็นภาษาไทยเราจะเพิ่มจำนวนให้มากกว่าภาษาอื่นในที่นี้จะกำหนดไว้ประมาณ 60 ยูอาร์แอล

ตารางที่ 7 ตัวอย่างยูอาร์แอลภาษาภาษานานาชาติที่กำหนดให้กูเกิ้ลค้นหา

ลำดับที่	ยูอาร์แอล	ภาษา
1	http://www.google.com/search?lr=lang_th&q=university	ไทย
2	http://www.google.com/search?lr=lang_zh-CN&q=university	จีน
3	http://www.google.com/search?lr=lang_ja&q=university	ญี่ปุ่น
4	http://www.google.com/search?lr=lang_fr&q=university	ฝรั่งเศส
…	…	…
43	http://www.google.com/search?lr=lang_ko&q=university	เกาหลี

ชุดข้อมูลสำหรับตัวตรวจสอบภาษาจะเกิดจากการเก็บเว็บเพจจากยูอาร์แอลภาษาต่างๆ ซึ่งในบางยูอาร์แอลอาจจะไม่สามารถเก็บเว็บเพจจากยูอาร์แอลนั้นได้ ในที่นี้เมื่อเก็บเว็บเพจครบทุกยูอาร์แอลแล้ว เราจะได้ผลลัพธ์เป็นจำนวนเว็บเพจทั้งสิ้น 552 เว็บเพจ แบ่งเป็นเว็บเพจภาษาไทย 58 เว็บเพจและเว็บเพจภาษาอื่น 494 เว็บเพจตามลำดับ เว็บเพจทั้งหมดจะไม่มีเฟรมแท็ก (Frame tag) และเป็นเว็บเพจที่ไม่มีรีไดเร็กแท็ก (Meta redirect tag) เพื่อให้เป็นข้อมูลที่สมบูรณ์เหมาะแก่การตรวจสอบภาษาจากเว็บเพจนั้นนั่นเอง

2.1.2 การทดสอบประสิทธิภาพของตัวตรวจสอบภาษา

เราจะทดสอบประสิทธิภาพของวิธีตรวจสอบภาษาทั้งหมด 3 วิธีได้แก่วิธีเท็กซ์แคท วิธีสตริงเคอร์เนล และวิธีเล็กซ์โตตามลำดับ โดยใช้ชุดข้อมูลทดสอบตัวตรวจสอบภาษาดังที่กล่าวไปในหัวข้อที่แล้วซึ่งแต่ละวิธีการจะมีการตั้งค่าก่อนการทดสอบที่แตกต่างกันดังต่อไปนี้

ก. วิธีเท็กซ์แคท: เราจะใช้โปรแกรมที่พัฒนาตามวิธีเท็กซ์แคทที่มีชื่อว่า textcat1.0.1 (TextCat) เมื่อใช้งานเราจะต้องกำหนดโพรไฟล์ของภาษาตัวอย่างให้กับโปรแกรมก่อน โดยการกำหนดที่ไฟล์ textcat.conf ซึ่งเป็นไฟล์ที่บอกว่าจะใช้โพรไฟล์ของภาษาอะไรบ้างเป็นต้น ในที่นี้เรากำหนดโพรไฟล์ไว้ 21 ภาษา สำหรับการใช้งานเราจะป้อนเนื้อหาให้กับเท็กซ์แคท จากนั้นเท็กซ์แคทจะคำนวณและให้ผลลัพธ์เป็นชื่อของภาษาออกมา เช่น “thai” หมายถึงเนื้อหานี้เป็นภาษาไทยเป็นต้น เนื่องจากเราต้องการผลลัพธ์เพียง 2 คำตอบคือภาษาไทยหรือภาษาอื่น ดังนั้นจึงสามารถใช้โพรไฟล์ที่มีจำนวนภาษาน้อยกว่าภาษาในชุดข้อมูลทดสอบได้ แต่ต้องมีโพรไฟล์ภาษาไทยรวมอยู่ด้วย

ข. วิธีสตริงเคอร์เนล: เราจะใช้โปรแกรมที่พัฒนาตามกรรมวิธีของสตริงเคอร์เนลเรียกว่าลิบส์ (LIBS) ซึ่งถูกคิดค้นและพัฒนาขึ้นโดย Tongchim *et al*. (2006) โปรแกรมนี้จะทำงานเป็นในลักษณะไคลเอนท์-เซิร์ฟเวอร์ กล่าวคือเริ่มต้นเราจะต้องสั่งให้โปรแกรมนี้ทำงานในแบบเซิร์ฟเวอร์ก่อน จากนั้นเราจะต้องเขียนโปรแกรมเชื่อมต่อเข้าไปยังชื่อเซิร์ฟเวอร์และพอร์ตที่กำหนดไว้ ในที่นี้เซิร์ฟเวอร์ทำงานที่พอร์ต 51717 ส่วนไคลเอนท์จะเขียนขึ้นโดยใช้ภาษาจาวาทำการเชื่อมต่อเข้าไปยังช่องทางดังกล่าวพร้อมกับส่งเนื้อหาเว็บเพจเข้าไป โปรแกรมลิบส์จะคำนวณและให้ผลลัพธ์ออกมาเป็นข้อความที่มีชื่อของภาษาออกมาด้วย ซึ่งคล้ายกันกับเท็กซ์แคท เช่นให้คำว่า “thai” ออกมาดังนั้นจะหมายถึงเนื้อหาของเว็บเพจนี้เป็นภาษาไทยเป็นต้น

ค. วิธีเล็กซ์โต: วิธีการใช้งานจะทำการประกาศคลาสเล็กซ์โตและกำหนดพจนานุกรมภาษาไทย (lexitron.txt) ให้กับโปรแกรม จากนั้นเราจะส่งเนื้อหาของเว็บเพจ p เข้าไปยังเล็กซ์โตจากนั้นเล็กซ์โตจะให้ผลลัพธ์เป็นกลุ่มของคำ เช่น เป็นคำที่รู้จักและคำที่ไม่รู้จักหรืออื่น ๆ ดังแสดงตามตัวอย่างในตารางที่ 6 เมื่อได้ผลลัพธ์ดังนี้เราจะทำการคำนวณหา $\Gamma(p)$ และตรวจสอบภาษาไทย $R(p)$ ดังที่ได้กล่าวมาแล้วในหัวข้อวิธีการตรวจสอบภาษาไทยโดยใช้เล็กซ์โตและนิยามของเว็บเพจภาษาไทยตามลำดับ

2.1.3 ผลการทดสอบประสิทธิภาพของตัวตรวจสอบภาษา

จากผลการทดสอบตัวตรวจสอบภาษาทั้ง 3 วิธีที่ได้กล่าวมาแล้ว กับชุดข้อมูลทดสอบตัวตรวจสอบภาษาซึ่งประกอบด้วยเว็บเพจทั้งสิ้น 552 เว็บเพจแบ่งเป็นภาษาไทย 58 และภาษาอื่น 494 เว็บเพจตามลำดับให้ผลลัพธ์ออกมาแล้วดังแสดงในตารางที่ 8

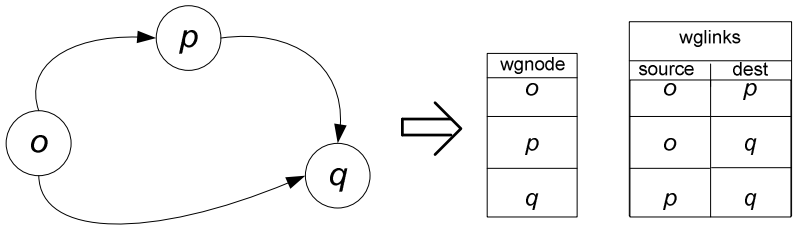
ตารางที่ 8 เปรียบเทียบประสิทธิภาพของตัวตรวจสอบภาษา

วิธีการ		ปริมาณภาษาที่พบในชุดข้อมูลทดสอบ		ความถูกต้อง (%)	เวลาที่ใช้	สาเหตุที่ผิดพลาด
		ภาษาไทย	ภาษาอื่น			
เล็กซ์โต	ภาษาไทย	48	0	82.76	16 วินาที	ปริมาณเนื้อหาที่เป็นภาษาไทยน้อยเกินกว่าที่จะตรวจสอบได้
(LexTo)	ภาษาอื่น	10	494		0.02 วินาที/หน้า	
เท็กซ์แคท	ภาษาไทย	29	0	50.0	306 วินาที	การเข้ารหัสเว็บเพจไม่เป็นชนิดเดียวกับการเข้ารหัสอักษรของโพรไฟล์เท็กซ์แคท
(TextCat)	ภาษาอื่น	29	494		0.5 วินาที/หน้า	
ลิบส์	ภาษาไทย	44	0	75.8	2953 วินาที	ปริมาณเนื้อหาที่เป็นภาษาไทยน้อยเกินกว่าที่จะตรวจสอบได้
(LIBS)	ภาษาอื่น	14	494		5.34 วินาที/หน้า	
รวม		58	494			

จากตารางที่ 8 จะเห็นว่าวิธีการเล็กซ์โตให้ผลลัพธ์ที่ดีที่สุดจากทั้ง 3 วิธีที่นำเสนอทั้งในแง่ของความถูกต้อง 82.76% หรือ (48/58) * 100 และในด้านของเวลาซึ่งใช้เวลาในการตรวจสอบแต่ละเว็บเพจเพียง 0.02 วินาที/หน้าเท่านั้น ดังนั้นในทุกการทดลองเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาไทยในวิทยานิพนธ์เล่มนี้ทั้งหมดจะตรวจสอบภาษาไทยโดยใช้วิธีของเล็กซ์โตและใช้นิยามของเว็บเพจภาษาไทยที่ได้กล่าวไว้ในหัวข้อที่ 1.3 เท่านั้น

2.2 ชุดข้อมูลเว็บกราฟ (Web graph dataset)

ใช้สำหรับทดสอบประสิทธิอัลกอริทึมและเพื่อศึกษาคุณสมบัติบางประการของเว็บเพจภาษาไทย ในการสร้างเว็บกราฟจะเก็บรวบรวมกลุ่มของเว็บเพจมาส่วนหนึ่งและทำการเชื่อมโยงเว็บเพจดังกล่าวเข้าด้วยกันโดยลิงค์ที่ออกจากเว็บเพจนั้นลักษณะเดียวกันกับอินเทอร์เน็ต หากพิจารณาเว็บเพจให้เป็นโหนดและลิงค์คือเส้นเชื่อมเว็บเพจเข้าด้วยกันก็จะมีลักษณะคล้ายกันกับกราฟ ดังนั้นจึงเรียกว่าเป็นชุดข้อมูลเว็บกราฟนั่นเอง ในวิทยานิพนธ์นี้จะใช้ฐานข้อมูลซึ่งประกอบไปด้วย 2 ตารางได้แก่ wgnode สำหรับเก็บรายละเอียดของเว็บเพจ และ wglinks สำหรับเก็บลิงค์ที่เชื่อมระหว่างเว็บเพจนั้น แนวคิดดังกล่าวแสดงเป็นแบบจำลองดังภาพที่ 14



ภาพที่ 14 การสร้างกราฟโดยใช้ตารางจากระบบฐานข้อมูล

2.2.1 การสร้างชุดข้อมูลเว็บกราฟ

เนื่องจากเราต้องการสัดส่วนของเว็บเพจที่เป็นภาษาไทยและภาษาอังกฤษที่ใกล้เคียงกัน ดังนั้นเราจะเริ่มต้นการเก็บเว็บเพจจากกลุ่มของยูอาร์แอลเริ่มต้นโดยใช้กูเกิ้ลหาลิงค์ที่ชี้เข้ามายังเว็บเพจประเทศไทยเขียนด้วยภาษาอังกฤษนั่นคือ http://www.nationmultimedia.com/ จำนวน 1 ระดับโดยใช้คำสั่ง link: http://www.nationmultimedia.com และเลือกเฉพาะยูอาร์แอลที่เป็นเว็บเพจที่เขียนด้วยภาษาอังกฤษซึ่งได้แก่ยูอาร์แอล 10 อันดับดังต่อไปนี้

ตารางที่ 9 ยูอาร์แอลเริ่มต้นสำหรับสร้างชุดข้อมูลเว็บกราฟ

ลำดับที่	ยูอาร์แอลเริ่มต้น
1	http://www.thaiwebsites.com/portal.asp
2	http://www.2bangkok.com/2bangkok/Skytrain/BTSMain4.shtml
3	http://thailand.yinyangandtaichichuan.org/informationlinks.html
4	http://www.cwanswers.com/8921/thailand/
5	http://thailandjumpedtheshark.blogspot.com/2009/05/pornpimol-pimping-for-abhsit.html
6	http://www.kas.de/proj/home/links/6/2/index.html
7	http://www.prachatai.com/english/node/1208/
8	http://news.carrentals.co.uk/bangkok-high-speed-airport-link-may-be-delayed-further-3427147.html
9	http://myprops.org/content/Man-found-dead-mouse-in-malt-loaf/
10	http://www.statbrain.com/nationmultimedia.com/

เราสร้างกราฟโดยใช้อัลกอริทึมค้นหาในแนวกว้าง (Breadth First Search: BFS) นอกจากนั้นแล้ว เราได้กำหนดรายละเอียดเพิ่มเติมตามงานวิจัยของ Pant *et al*. (2002) เข้าไปอีก ได้แก่กำหนดขนาดของยูอาร์แอลคิวไว้ที่ 12000 ยูอาร์แอล เพื่อเป็นการป้องกันกับดักเว็บคราวเลอร์ เราจะจำกัดความยาวของยูอาร์แอลไม่เกิน 200 ตัวอักษร และยังเลือกเก็บเว็บเพจเฉพาะนามสกุลที่เป็นที่รู้จักกันดี เช่น .html .htm .asp .aspx หรือ .php เป็นต้น

เว็บกราฟที่สร้างเสร็จแล้วและจะถูกใช้ในการทดลองจะประกอบด้วยกลุ่มเว็บเพจเป็นจำนวนทั้งสิ้น 103734 เว็บเพจโดยแบ่งเป็นเว็บเพจภาษาไทยจำนวน 36134 (34.83%) และเว็บเพจภาษาอื่นเป็นจำนวน 67600 (65.16%) และระยะทางที่มากที่สุดของลิงค์ที่ห่างจากกลุ่มยูอาร์แอลเริ่มต้นคือ 15 ซึ่งจากงานวิจัยของ Somboonviwat *et al.* (2005) ได้อธิบายว่าระยะทางที่ห่างจากเว็บเพจภาษาไทยต้นทางที่มากที่สุดที่ยังพบเว็บเพจภาษาไทยอยู่ที่ 4 ดังนั้นจึงพอคาดเดาได้ว่าจากปริมาณของเว็บเพจที่อยู่ในเว็บกราฟของเราน่าจะให้ผลในการทดลองที่น่าเชื่อถือได้

2.2.2 ลักษณะของชุดข้อมูลเว็บกราฟและความสัมพันธ์กับเว็บเพจภาษาไทย

จากการวิเคราะห์ชุดข้อมูลเว็บกราฟเมื่อพิจารณาความสัมพันธ์ของเว็บเพจภาษาไทยต่ออินเทอร์เน็ตโดเมนจะพบว่ามากกว่าครึ่งหนึ่ง (56.36%) ของเว็บเพจภาษาไทยอยู่ภายใต้อินเทอร์เน็ตโดเมน ".com" และเมื่อพิจารณาโดเมน ".th" จะพบว่ามีเพียง 15.85% คือเว็บเพจภาษาไทยและ 4.68% คือเว็บเพจภาษาอื่น เช่น เว็บเพจที่เกี่ยวข้องกับประเทศไทยแต่เขียนด้วยภาษาอังกฤษ เป็นต้น ปรากฏอยู่ในโดเมนดังกล่าวตามลำดับ ดังนั้นจึงสอดคล้องกับสมมติฐานที่ว่าวิธีการค้นหาเว็บเพจภาษาไทยด้วยการจำกัดการค้นหาเพียงเว็บเพจในโดเมน ".th" จะทำให้ได้เว็บเพจภาษาไทยที่ไม่ครอบคลุมต่อปริมาณเว็บเพจภาษาไทยทั้งหมดดังแสดงในตารางที่ 10

ตารางที่ 10 ความสัมพันธ์ของเว็บเพจภาษาไทยต่ออินเทอร์เน็ตโดเมนที่พบในชุดข้อมูลเว็บกราฟ

อินเทอร์เน็ตโดเมน	ภาษาไทย		ภาษาอื่น	
.com	20366	(56.36%)	58429	(86.43%)
.net	9317	(25.78%)	583	(0.86%)
.th	5727	(15.85%)	3164	(4.68%)
.org	491	(1.36%)	3763	(5.57%)
.info	27	(0.07%)	26	(0.04%)
others	206	(0.57%)	1635	(2.42%)
รวม	36134		67600	

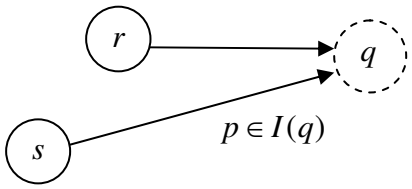
เมื่อพิจารณาความสัมพันธ์ของเว็บเพจภาษาไทยกับการเข้ารหัสอักษรในเว็บเพจซึ่งถูกสกัดมาจากเมต้าแท็กและผลลัพธ์จากโปรแกรมสำหรับตรวจสอบการเข้ารหัสอักษรดังที่กล่าวมาแล้วในหัวข้อที่ 1.1 ดังนั้นจากตารางที่ 11 จะพบว่าเกือบจะถึงครึ่งหนึ่งของเว็บเพจภาษาไทย (33.33%) ถูกเข้ารหัสอักษรด้วย UTF-8 ดังนั้นการตัดสินเว็บเพจว่าเป็นภาษาไทยโดยพิจารณาเพียงเข้ารหัสอักษร Windows-874/TIS-620 หรือ ISO-8859-11 ตามข้อกำหนดของ W3C จึงไม่ครอบคลุมต่อปริมาณเว็บเพจภาษาไทยทั้งหมดที่เข้าถูกเข้ารหัสตัวอักษรแบบอื่นที่นอกเหนือจากข้อกำหนดนี้

ตารางที่ 11 ความสัมพันธ์ของเว็บเพจภาษาไทยต่อการเข้ารหัสอักษรที่พบในชุดข้อมูลเว็บกราฟ

รหัสอักษร	ภาษาไทย		ภาษาอื่น	
TIS-620	15128	(41.87%)	1203	(1.78%)
UTF-8	12043	(33.33%)	23674	(35.02%)
Windows-874	8551	(23.66%)	1984	(2.93%)
ISO-8859-1	289	(0.80%)	38486	(56.93%)
Windows-1252	93	(0.26%)	507	(0.75%)
US-ASCII	10	(0.03%)	444	(0.66%)
ISO-8859-11	3	(0.01%)	3	(0.00%)
Others	17	(0.05%)	1299	(1.92%)
รวม	36134		67600	

3. ฟีเจอร์ของเว็บเพจภาษาไทย

พิจารณาเว็บกราฟดังภาพที่ 15 กำหนดให้ q คือเว็บเพจปลายทางหรือเว็บเพจที่ยังไม่ถูกเก็บรวบรวมมาซึ่งแสดงด้วยเส้นประ $I(q)$ คือเซตของเว็บเพจต้นทางใดๆที่ถูกเก็บรวบรวมมาแล้วและชี้เข้ามายังเว็บเพจ q ซึ่งได้แก่เว็บเพจ $\{r,s\}$ ดังนั้นถ้าหาก $p \in I(q)$ ดังนั้น p ก็อาจจะเป็นเว็บเพจ r หรือเว็บเพจ s เป็นต้น เราจะใช้สัญลักษณ์เหล่านี้ในการหาฟีเจอร์ทั้ง 4 ชนิดได้แก่ ภาษาไทยจากเว็บเพจต้นทาง ภาษาไทยจากแองค์เคอร์เท็กซ์ของเว็บเพจต้นทางในส่วนที่เว็บเพจต้นทางชี้ไปยังเว็บเพจปลายทาง อินเทอร์เน็ตโดเมนของเว็บเพจปลายทาง และเส้นทางของเว็บเพจภาษาไทยต้นทางจนถึงเว็บเพจปลายทางซึ่งจะกล่าวในหัวข้อถัดไปตามลำดับ



ภาพที่ 15 ตัวอย่างของเว็บกราฟปลายทาง q ที่ถูกชี้โดยเว็บเพจต้นทาง r และ s

3.1 ภาษาไทยจากเว็บเพจต้นทาง (Parent language: a_1)

ฟีเจอร์นี้เกิดจากสมมติฐานว่า เมื่อเราพิจารณาเว็บเพจใด ๆ ในอินเทอร์เน็ตมักจะพบว่าผู้พัฒนามักจะสร้างลิงค์เพื่อเชื่อมโยงเนื้อหาของเว็บเพจที่เขียนด้วยภาษาเดียวกันเข้าไว้ด้วยกัน เพื่อให้ผู้อ่านเว็บเพจในภาษานั้นอ่านเนื้อหาและเข้าใจได้อย่างราบรื่นและต่อเนื่องโดยไม่ต้องแปลให้เป็นภาษาอื่น ในกรณีของเว็บเพจภาษาไทยก็เช่นเดียวกันถ้าหากเว็บเพจที่เก็บรวบรวมมาเป็นเว็บเพจภาษาไทยดังนั้นจึงมีความเป็นไปได้สูงที่ยูอาร์แอลที่ชี้ออกจากเว็บเพจนั้นจะเป็นยูอาร์แอลของเว็บเพจภาษาไทยด้วยเช่นเดียวกัน ดังนั้นจากเว็บกราฟตัวอย่างดังภาพที่ 15 เราจะพบว่าความเป็นไปได้ที่เว็บเพจ q จะเป็นเว็บเพจภาษาไทยเกิดจากปริมาณของภาษาไทยที่พบในเว็บเพจ p ซึ่งสามารถเขียนแทนด้วยฟังก์ชั่น $a_1(q)$ ดังสมการที่ 20 ได้ดังต่อไปนี้

$$a_1(q) = \Gamma(p) \tag{20}$$

เมื่อ $a_1(q)$ คือค่าความเป็นไปได้ที่ยูอาร์แอลของเว็บเพจ q จะเป็นภาษาไทย ซึ่งจะมีค่าระหว่าง $0 \le a_1(q) \le 1$ และ $\Gamma(p)$ คืออัตราส่วนความเป็นภาษาไทยของเว็บเพจ p ที่คำนวณได้จากสมการที่ 18 ดังที่กล่าวมาแล้วและ $p \in I(q)$

จากผลการทดสอบสมมติฐานดังกล่าวนี้ในชุดข้อมูลเว็บกราฟ (Web graph dataset) ดังแสดงในตารางที่ 12 เมื่อพิจารณาเฉพาะคลาสที่เป็นเว็บเพจภาษาไทยพบว่าฟีเจอร์นี้ให้ความแม่นยำ (Precision) หรืออัตราการเก็บเกี่ยว (Harvest rate) ที่ 7353 / (7353+1082) = 0.87 และให้ผลลัพธ์ในด้านค่าความระลึก (Recall) หรือความครอบคลุม (Coverage) อยู่ที่ 7353 / 7897 = 0.93 ดังนั้นค่าเอฟ (F-Measure) ซึ่งเป็นความสัมพันธ์ระหว่างค่าความแม่นยำกับค่าความระลึก ดังแสดงในตารางที่ 12 จะเป็น 0.90 เช่นเดียวกันเมื่อพิจารณาคลาสที่เป็นเว็บเพจภาษาอื่นจะเห็นว่าให้ค่าเอฟอยู่ที่ 0.97 ดังนั้นจึงอาจกล่าวได้ว่าฟีเจอร์นี้เป็นฟีเจอร์ที่ดีมากในการแบ่งแยกความแตกต่าง (discriminate) ระหว่างคลาสของเว็บเพจภาษาไทยกับเว็บเพจภาษาอื่น อีกทั้งยังแสดงให้เห็นว่าฟีเจอร์นี้มีความสำคัญที่สุดในการค้นหาเว็บเพจภาษาไทยเมื่อเทียบกับฟีเจอร์อีก 3 ชนิดที่เหลือนั่นเอง

3.2 ภาษาไทยจากแองค์เคอร์เท็กซ์ (Anchor text Language: a_2)

แองเคอร์เท็กซ์ คือข้อความที่อธิบายเกี่ยวกับรายละเอียดของยูอาร์แอลหรือลิงค์นั้น โดยที่ผู้ที่อ่านเว็บเพจ จึงมักจะเห็นข้อความนี้และอ่านก่อนที่จะตัดสินใจตามลิงค์เพื่อไปยังหน้าอื่นอยู่เสมอ ดังนั้นจึงถือว่าแองค์เคอร์เท็กซ์เป็นข้อมูลที่สำคัญที่ใช้ในการคาดเดาถึงเนื้อหาของเว็บเพจปลายทางได้เป็นอย่างดี สำหรับฟีเจอร์นี้เกิดจากสมมติฐานที่ว่า หากผู้อ่านเว็บเพจต้องการที่จะอ่านเว็บเพจที่เป็นภาษาภาษาไทยในหน้าถัดไป ผู้อ่านเว็บเพจมักจะเลือกคลิกลิงค์จากแองเคอร์เท็กซ์ที่เป็นภาษาไทยมากกว่าแองเคอร์เท็กซ์ที่เป็นภาษาหน้าอื่นที่ไม่ใช่ภาษาไทยเป็นต้น และอีกสมมติฐานหนึ่งก็คือ ผู้สร้างเว็บเพจจะใช้แองเคอร์เท็กซ์ภาษาไทยมาอธิบายเว็บเพจปลายทางที่เป็นภาษาไทยเช่นเดียวกัน จากแนวความคิดของทั้งสองสมมติฐานดังกล่าว เมื่อพิจารณาเว็บกราฟตัวอย่างดังภาพที่ 15 เราจะพบว่าความเป็นไปได้ที่เว็บเพจ q จะเป็นเว็บเพจภาษาไทยเกิดจากปริมาณของภาษาไทยที่พบในแองค์เคอร์เท็กซ์ของยูอาร์แอล q จากเว็บเพจ p ซึ่งสามารถเขียนแทนด้วยฟังก์ชั่น $a_2(q)$ ดังสมการที่ 21 ได้ดังต่อไปนี้

$$a_2(q) = \Gamma(anchortext_{p,q}) \tag{21}$$

เมื่อ $a_2(q)$ คือค่าความเป็นไปได้ที่ยูอาร์แอลของเว็บเพจ q จะเป็นภาษาไทย ซึ่งจะมีค่าระหว่าง $0 \le a_2(q) \le 1$ และ $\Gamma(anchortext_{p,q})$ คืออัตราส่วนความเป็นภาษาไทยของแองเคอร์เท็กซ์ที่บ่งบอกถึงเว็บเพจ q ที่ปรากฏอยู่ในเว็บเพจ p

จากผลการทดสอบสมมติฐานดังกล่าวดังแสดงในตารางที่ 12 เมื่อพิจารณาเฉพาะคลาสที่เป็นเว็บเพจภาษาไทยพบว่าฟีเจอร์นี้ให้อัตราการเก็บเกี่ยวที่ 0.92 และให้ความครอบคลุม 0.56 ดังนั้นอาจกล่าวได้ว่าฟีเจอร์นี้ให้ความแม่นยำสูงซึ่งเหมาะสมอย่างยิ่งสำหรับใช้ในการคาดเดายูอาร์แอล q ว่าจะเป็นเว็บเพจภาษาไทยหรือไม่ อย่างไรก็ตามฟีเจอร์ดังกล่าวให้ค่าความครอบคลุมที่ค่อนข้างต่ำ ซึ่งอาจเป็นไปได้ว่ามีแองค์เคอร์เท็กซ์ภาษาอื่นปรากฏอยู่ในเว็บเพจในชุดข้อมูลเว็บกราฟเป็นจำนวนมาก ดังนั้นค่าเอฟจึงได้เท่ากับ 0.69 และเมื่อพิจารณาคลาสที่เป็นเว็บเพจภาษาอื่นจะเห็นว่าให้ค่าเอฟอยู่ที่ 0.94 กล่าวโดยสรุปแล้วถึงแม้ว่าฟีเจอร์นี้จะแบ่งแยกความแตกต่างได้ไม่ดีมากนักเมื่อเทียบกับฟีเจอร์แรก แต่ก็ให้ความแม่นยำหรืออัตราการเก็บเกี่ยวที่สูง ซึ่งเหมาะสมสำหรับนำมาใช้ค้นหาเว็บเพจภาษาไทยได้เป็นอย่างดี

3.3 อินเทอร์เน็ตโดเมนของเว็บเพจภาษาไทย (Thai domain name: a_3)

เมื่อแยกโครงสร้างของยูอาร์แอลออกมาจะพบว่ามีอินเทอร์เน็ตโดเมนท้องถิ่นระดับบนสุด (Country Code Top Level Domain: ccTLD) ซึ่งเป็นตัวกำหนดว่ายูอาร์แอลนั้นเป็นของเว็บเพจประจำประเทศใด ยกตัวอย่างเช่นเว็บเพจภาษาไทยส่วนหนึ่งจะมียูอาร์แอลที่กำกับด้วยโดเมน ".th" เป็นต้น ดังนั้นความเป็นไปได้ที่เว็บเพจ q จะเป็นเว็บเพจภาษาไทยจะเกิดจากอินเทอร์เน็ตโดเมนที่ปรากฏอยู่ในยูอาร์แอลของเว็บเพจตนเอง ซึ่งสามารถเขียนแทนด้วยฟังก์ชั่น $a_3(q)$ ดังสมการที่ 22 ได้ดังต่อไปนี้

$$a_3(q) = \begin{cases} 1 & \text{if } domain(url_{p,q}) \in \text{'.th'} \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

เมื่อ $a_3(q)$ คือค่าความเป็นไปได้ที่ยูอาร์แอลของเว็บเพจ q จะเป็นภาษาไทยโดยกำหนดให้ $url_{p,q}$ คือยูอาร์แอลของเว็บเพจ q และ $domain$ คือฟังก์ชั่นที่หาอินเทอร์เน็ตโดเมนของยูอาร์แอล ยกตัวอย่างเช่น $\text{domain('http://www.ku.ac.th')}$ จะได้ผลลัพธ์ออกมาเป็น ".th" เป็นต้น

จากผลการทดสอบสมมติฐานดังกล่าวดังแสดงในตารางที่ 12 เมื่อพิจารณาเฉพาะคลาสที่เป็นเว็บเพจภาษาไทยพบว่าฟีเจอร์นี้ให้อัตราการเก็บเกี่ยวหรือความแม่นยำที่ 0.73 และให้ความครอบคลุม 0.26 ซึ่งอาจกล่าวได้ว่ามีความแม่นยำ 73% ที่เว็บเพจ q จะเป็นเว็บเพจภาษาไทยจะเกิดจากอินเทอร์เน็ตโดเมน ".th" ที่ปรากฏอยู่ในยูอาร์แอลของเว็บเพจดังกล่าวนี้ เมื่อวิเคราะห์ค่าความครอบคลุมจะเห็นว่าอยู่ในระดับที่น้อย ซึ่งจากการพิจารณาข้อมูลในตารางที่ 10 จะพบว่าเว็บเพจภาษาไทยในชุดข้อมูลเว็บกราฟที่เป็น ".th" มีเพียง 15.85% เท่านั้น จึงทำให้ความสามารถในการค้นหาเว็บเพจภาษาไทยของฟีเจอร์นี้ไม่ครอบคลุมถึงปริมาณเว็บเพจภาษาไทยทั้งหมดได้ ดังนั้นค่าเอฟจึงได้เท่ากับ 0.39 และเมื่อพิจารณาคลาสที่เป็นเว็บเพจภาษาอื่นจะเห็นว่าให้ค่าเอฟอยู่ที่ 0.91

ถึงแม้ว่าจะมีงานวิจัยที่กล่าวว่าเว็บเพจภาษาไทยมากกว่าครึ่งหนึ่งอยู่นอกเหนือโดเมนดังกล่าว (Somboonviwat *et al.*, 2005) ซึ่งทำให้วิธีในการจำกัดการค้นหาแบบจำกัดเฉพาะโดเมนให้ค่าความครอบคลุมที่น้อย อย่างไรก็ตามถ้าหากเกิดเหตุการณ์ที่เว็บคราวเลอร์ตกไปอยู่ในกลุ่มของเว็บเพจภาษาอื่นจนทำให้ไม่สามารถค้นหาเว็บเพจภาษาไทยต่อไปอีกได้ วิธีนี้จะมีประโยชน์มากที่

จะทำให้เว็บคราวเลอร์กลับมายังเส้นทางที่เป็นเว็บเพจภาษาไทยได้อย่างรวดเร็วมากยิ่งขึ้นโดยพิจารณาเพียงอินเทอร์เน็ตโดเมนนั่นเอง

3.4 เส้นทางของเว็บเพจภาษาไทย (Thai web page path: a_4)

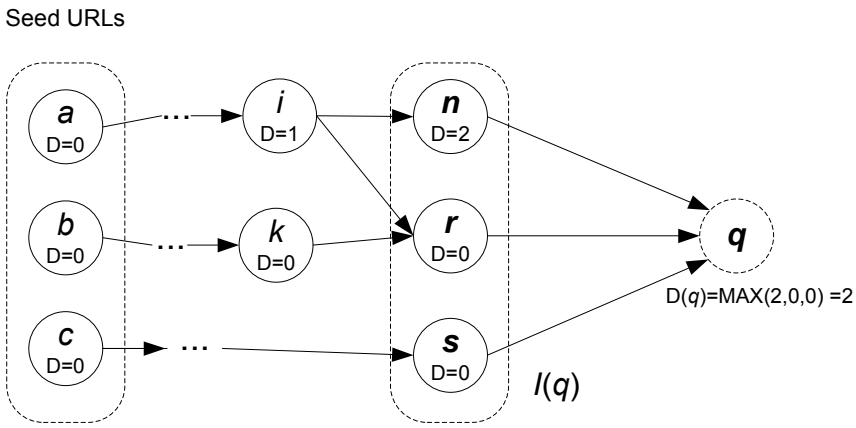
ฟีเจอร์นี้เกิดจากสมมติฐานถ้าหากพิจารณาเว็บเพจใด ๆ ในเว็บกราฟ ที่วางอยู่ในเส้นทางที่ประกอบไปด้วยกลุ่มของเว็บเพจภาษาไทยลิงค์เชื่อมต่อกันเป็นจำนวนมาก ดังนั้นเว็บเพจนี้ก็น่าที่จะเป็นเว็บเพจภาษาไทยมากกว่า เมื่อเทียบกับการที่เว็บเพจนี้วางอยู่ในเส้นทางที่ประกอบไปด้วยกลุ่มของเว็บเพจภาษาอื่นจำนวนมากลิงค์เชื่อมต่อกันอยู่

พิจารณาเว็บกราฟตัวอย่างในภาพที่ 16 ให้ c และ k เป็นเว็บเพจภาษาอื่น ส่วนเว็บเพจที่นอกเหนือจากนั้นจะกำหนดให้เป็นเว็บเพจภาษาไทยทั้งหมด เราจะให้เว็บคราวเลอร์เริ่มเดินทางจากกลุ่มเว็บเพจเริ่มต้นได้แก่ เว็บเพจ a , b และ c ตามลำดับโดยเรากำหนดระยะทางเริ่มต้นให้เป็นศูนย์ทั้งหมด จากภาพเมื่อเว็บคราวเลอร์เดินทางจากกลุ่มเว็บเพจเริ่มต้นผ่านไปยังเว็บเพจอื่น ๆ จะเห็นว่าค่าของเส้นทางของเว็บเพจใด ๆ จะเพิ่มขึ้นเมื่อเว็บเพจนั้นเป็นภาษาไทย และค่าของเส้นทางจะเป็น 0 เมื่อพบว่าเว็บเพจต้นทางเป็นภาษาอื่น ดังนั้นถ้าเว็บเพจใดประกอบด้วยลิงค์ชี้เข้ามากกว่า 1 ลิงค์ เราจะเลือกพิจารณาลิงค์ของเว็บเพจที่มีค่าของเส้นทางที่มากที่สุดก่อนเสมอและบวกด้วย 1 ถ้าเว็บเพจต้นทางเป็นภาษาไทย แต่ถ้าเว็บเพจต้นทางเป็นภาษาอื่น ดังนั้นเว็บเพจปลายทางจะมีค่าเป็น 0 ทันที

จากแนวคิดดังที่กล่าวมาแล้วข้างต้นจะพบว่าเส้นทางของเว็บเพจ q ใด ๆ จะเกิดจากค่าของเส้นทางที่ได้จากเว็บเพจต้นทาง p โดยที่เว็บเพจ $p \in I(q)$ ดังนั้นในที่นี้ p อาจจะเป็นเว็บเพจ n, r หรือ s ก็ได้ซึ่งสามารถเขียนแทนด้วยฟังก์ชั่น $D(q)$ ได้ดังต่อไปนี้

$$D(q) = \begin{cases} 1 + \underset{p \in I(q)}{MAX}(D(p)) & \text{if } R(p) = \text{Thai} \\ 0 & \text{otherwise} \end{cases} \qquad (23)$$

เมื่อ $D(q)$ คือค่าเส้นทางของยูอาร์แอลของเว็บเพจ q ซึ่งเป็นเลขจำนวนเต็มบวก $R(p)$ คือฟังก์ชั่นในการตรวจสอบเว็บเพจ p ว่าเป็นภาษาไทยตามนิยามของเว็บเพจภาษาไทยและ MAX คือฟังก์ชั่นที่ใช้ในการหาตัวเลขที่มากที่สุดจากตัวเลขชุดหนึ่ง



ภาพที่ 16 ฟีเจอร์เส้นทางของเว็บเพจภาษาไทย

จากนั้นเราจะปรับค่าของ $D(q)$ ซึ่งเป็นเลขจำนวนเต็มบวกให้อยู่ในรูปของทศนิยมซึ่งเขียนแทนด้วยฟังก์ชั่น $a_4(q)$ ได้ดังสมการที่ 24 ดังต่อไปนี้

$$a_4(q) = \log(1 + D(q)) \tag{24}$$

เมื่อ $a_4(q)$ คือความเป็นไปได้ที่ยูอาร์แอลของเว็บเพจ q จะเป็นภาษาไทย log คือฟังก์ชั่นทางคณิตศาสตร์ที่ใช้สำหรับแปลงเลขจำนวนเต็มบวกให้อยู่ในรูปของลอกอริทึม เนื่องจาก log(0) ไม่สามารถที่จะหาค่าได้ ดังนั้นเราจึงต้องปรับค่าของเส้นทางด้วย $1 + D(q)$ ก่อนนั่นเอง

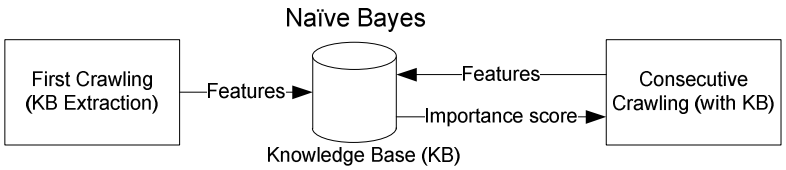
จากผลการทดสอบสมมติฐานเส้นทางของเว็บเพจภาษาไทย (a_4) ดังแสดงในตารางที่ 12 เมื่อพิจารณาเฉพาะคลาสที่เป็นเว็บเพจภาษาไทยพบว่าฟีเจอร์นี้ให้อัตราการเก็บเกี่ยวที่ 0.89 และให้ความครอบคลุม 0.85 และให้ค่าเอฟที่ 0.87 และเมื่อพิจารณาคลาสที่เป็นเว็บเพจภาษาอื่นจะเห็นว่าให้ค่าเอฟอยู่ที่ 0.97 จึงอาจกล่าวได้ว่าฟีเจอร์นี้ใช้ในการแบ่งแยกความแตกต่างระหว่างคลาสของเว็บเพจภาษาไทยกับเว็บเพจภาษาอื่นได้ดี และประโยชน์อย่างหนึ่งที่คาดว่าจะได้รับจากฟีเจอร์นี้คือความสามารถในการนำพาเว็บคราวเลอร์ให้เข้าไปอยู่ในเส้นทางของเว็บเพจภาษาไทยได้อย่างรวดเร็วและต่อเนื่อง ซึ่งเมื่อนำมาใช้ร่วมกับฟีเจอร์อื่น ๆ ก็น่าที่จะช่วยเสริมให้ประสิทธิภาพในการค้นหาเว็บเพจภาษาไทยได้ดียิ่งขึ้นตามไปด้วย

ตารางที่ 12 ผลการทดสอบการค้นหาเว็บเพจของฟีเจอร์แต่ละชนิดโดยใช้ยูอาร์แอลเริ่มต้นจากตารางที่ 9 และกำหนดให้เว็บคราวเลอร์เก็บเว็บเพจเป็นจำนวนทั้งสิ้น 40000 เว็บเพจ

ฟีเจอร์	ผลที่ได้	จำนวนเว็บเพจในชุดข้อมูลเว็บกราฟ		Precision (Harvest Rate)	Recall (Coverage)	F-Measure $F=\frac{2\times P\times R}{(P+R)}$
		ภาษาไทย	ภาษาอื่น	(P)	(R)	
ภาษาไทยจาก	ภาษาไทย	7353	1082	0.87	0.93	0.90
เว็บเพจต้นทาง (a_1)	ภาษาอื่น	544	31021	0.98	0.97	0.97
ภาษาไทยจากแองค์	ภาษาไทย	4388	390	0.92	0.56	0.69
เคอร์เท็กซ์ (a_2)	ภาษาอื่น	3509	31713	0.9	0.99	0.94
อินเทอร์เน็ตโดเมน	ภาษาไทย	2087	782	0.73	0.26	0.39
ของภาษาไทย (a_3)	ภาษาอื่น	5810	31321	0.84	0.98	0.91
เส้นทางของเว็บเพจ	ภาษาไทย	6735	862	0.89	0.85	0.87
ภาษาไทย (a_4)	ภาษาอื่น	1162	31241	0.96	0.97	0.97
รวม		7897	32103			

4. ขั้นตอนวิธีของเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาไทย

วิธีการเก็บรวบรวมเว็บเพจภาษาไทยที่เราได้นำเสนอในวิทยานิพนธ์เล่มนี้แบ่งออกเป็น 2 ขั้นตอนย่อยได้แก่ ขั้นตอนแรกคือการเก็บเว็บเพจเพื่อนำมาสร้างฐานความรู้ทำหน้าที่สกัดฟีเจอร์ทั้ง 4 ชนิดและเก็บลงสู่ฐานความรู้ และขั้นตอนที่สองคือการเก็บเว็บเพจในรอบถัดไปโดยใช้ฐานความรู้ที่สร้างขึ้น ขั้นตอนนี้จะใช้ฐานความรู้และแบบจำลองความน่าจะเป็นแบบเบส์อย่างง่ายในการคำนวณค่าคะแนนความสำคัญเมื่อกำหนดฟีเจอร์ทั้ง 4 ชนิดจากเว็บเพจที่เพิ่งเก็บรวบรวมมา คะแนนดังกล่าวจะถูกนำมากำหนดให้กับยูอาร์แอลแต่ละตัวที่ออกจากเว็บเพจนั้น รายละเอียดแต่ละขั้นตอนและอัลกอริทึมที่นำเสนอสรุปได้ดังภาพที่ 17 ซึ่งจะได้กล่าวถึงในหัวข้อถัดไป



ภาพที่ 17 ภาพรวมของขั้นตอนวิธีของเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาไทย

4.1 การเก็บเพจเพื่อนำมาสร้างฐานความรู้ (Knowledge based extraction)

เมื่อเว็บคราวเลอร์ไม่มีฐานความรู้อันใดเกี่ยวกับลักษณะของเว็บเพจภาษาไทยก็จะไม่มีความสามารถที่จะเก็บเว็บเพจภาษาไทยอย่างมีประสิทธิภาพตามที่ต้องการได้ เปรียบเสมือนกับเว็บคราวเลอร์ที่ไม่มีความรู้จึงคล้ายกับเว็บคราวเลอร์ที่ตาบอดเดินทางไปเก็บเว็บเพจภาษาไทยนั่นเอง ดังนั้นหัวใจสำคัญในขั้นตอนนี้คือการสร้างฐานความรู้ที่เกี่ยวกับลักษณะของเว็บเพจภาษาไทย เพื่อให้เว็บคราวเลอร์ใช้เรียนรู้ในการเก็บเว็บเพจภาษาไทยในครั้งถัดไปซึ่งจะเป็นอัลกอริทึมหลักของการเก็บเว็บเพจภาษาไทยนั่นเอง

การสร้างฐานความรู้นั้นเราจะทำการเก็บรวบรวมเว็บเพจขึ้นมาก่อนจำนวนหนึ่งเพื่อนำเว็บเพจเหล่านั้นมาสกัดเอาฟีเจอร์ทั้ง 4 ชนิดที่จะใช้ในการบ่งบอกถึงลักษณะของเว็บเพจภาษาไทย แล้วนำมาสร้างเป็นฐานความรู้เก็บไว้สำหรับใช้ในการเก็บเว็บเพจในครั้งถัดไป ซึ่งเขียนเป็นโค้ดเทียมได้ดังอัลกอริทึมที่ 1

อัลกอริทึมที่ 1 ขั้นตอนการเก็บเว็บเพจเพื่อนำมาสร้างฐานความรู้

```
 1: KB_Extraction(starting_urls, a1[], a2[], a3[], a4[]) {
 2:         forall(v in starting_urls) {
 3:                 node = Create_Node(v, a1[v], a2[v], a3[v], a4[v])
 4:                 Enqueue(FIFO_QUEUE, node)
 5:         }
 6:         while(FIFO_QUEUE is not empty and i < max_pages) {
 7:                 node = Dequeue(FIFO_QUEUE)
 8:                 p = Fetch_Page(node.url)
 9:                 if(R(p) == Thai) {
10:                         class = "thai"
11:                 } else {
12:                         class = "non"
13:                 }
14:                 Add_To_KB(node.a1,node.a2, node.a3, node.a4, class)
15:
16:                 extract features a1 and a4 from p
17:                 forall(q in Extract_URL(p)) {
18:                         discard q when q is visited
19:                         extract features a2 and a3 from q
20:                         new_node = Create_Node(q, a1, a2, a3, a4)
21:                         Enqueue(FIFO_QUEUE, new_node)
22:                 }
23:                 i = i + 1
24:         }
25: }
```

ก่อนที่จะเข้าสู่อัลกอริทึมที่ 1 เราจะเก็บเว็บเพจเริ่มต้นมากลุ่มหนึ่งจากนั้นจะคัดแยกยูอาร์แอลที่ชี้ออกจากเว็บเพจกลุ่มนั้นเรียกว่า starting_urls ซึ่งปรากฏอยู่ในอัลกอริทึมที่ 1 บรรทัดที่ 1 และกำหนดให้ $a_1[], a_2[], a_3[]$ และ $a_4[]$ คือฟีเจอร์เริ่มต้นทั้ง 4 ชนิดที่สกัดออกมาจากกลุ่มของเว็บเพจเริ่มต้นเหล่านี้และจะถูกกำหนดให้กับยูอาร์แอลแต่ละตัวในรูปแบบของคู่ลำดับอาร์เรย์ ยกตัวอย่างเช่น $a_1['\text{http://www.ku.ac.th}']$ หมายถึงค่าคะแนนฟีเจอร์ a_1 ของยูอาร์แอล http://www.ku.ac.th เป็นต้น

ฟังก์ชั่น Create_Node ในบรรทัดที่ 3 และ 20 เป็นฟังก์ชั่นที่ใช้ในการสร้างชุดข้อมูล Node ซึ่งเป็นชุดข้อมูลที่ถูกออกแบบมาเพื่อให้เราจัดการกับข้อมูลจำนวนมากได้ง่ายขึ้น โดยประกอบไปด้วยยูอาร์แอล ฟีเจอร์ทั้ง 4 ชนิดและคลาสที่เป็นคำตอบที่ได้จากการตรวจสอบภาษาในเว็บเพจ p

ตัวแปร max_pages ในบรรทัดที่ 6 คือตัวแปรที่ใช้สำหรับจำกัดจำนวนของเว็บเพจที่จะนำมาสร้างฐานความรู้ ฟังก์ชั่น Dequeue ในบรรทัดที่ 7 จะเป็นนำยูอาร์แอลออกจากยูอาร์แอลคิว โดยพิจารณาลำดับของยูอาร์แอลที่เข้ามาก่อนก็จะนำออกมาเก็บเว็บเพจจากยูอาร์แอลนั้นก่อน ซึ่งเป็นวิธีการเก็บเว็บเพจในแนวกว้างนั่นเอง สำหรับ $R(p)$ ในบรรทัดที่ 9 คือการตรวจสอบเว็บเพจว่าเป็นภาษาไทยหรือไม่ตามนิยามของเว็บเพจภาษาไทยที่ได้กล่าวไปแล้ว ส่วนฟังก์ชั่น Add_To_KB ในบรรทัดที่ 14 เป็นฟังก์ชั่นในการเพิ่มข้อมูลลงไปในฐานความรู้อันประกอบไปด้วยฟีเจอร์ทั้ง 4 ชนิดและคลาสที่เป็นคำตอบของฟีเจอร์นั้น ซึ่งลักษณะของฐานความรู้แสดงดังภาพที่ 18 สำหรับบรรทัดที่ 16 และ 19 เป็นการสกัดเอาฟีเจอร์ทั้งหมด 4 ชนิดจากเว็บเพจ p ที่ได้เก็บรวบรวมมาแล้ว และคำอธิบายในบรรทัดที่ 18 คือเว็บคราวเลอร์จะทิ้งยูอาร์แอล q เมื่อพบว่าเคยเก็บเว็บเพจจากยูอาร์แอลนี้ไปแล้ว

```
@relation all
@attribute parent real
@attribute anchor real
@attribute domain real
@attribute distance real
@attribute output {thai, non}
@data
0.00,0.00,0,0.00,thai
0.55,0.00,1,0.00,thai
0.55,0.80,1,0.00,non
0.55,1.00,0,0.16,thai
0.55,0.00,0,0.00,non
...
```

ภาพที่ 18 ตัวอย่างฐานความรู้ที่ได้จากการสร้างโดยใช้อัลกอริทึมที่ 1

ในทางปฏิบัติแล้วเราจะวัดความถูกต้องของฐานความรู้ในภาพที่ 18 โดยใช้เวก้า (Weka, 1994) ซึ่งเป็นชุดซอฟท์แวร์ที่สร้างขึ้นมาเพื่อใช้ทดสอบอัลกอริทึมที่เกี่ยวกับการทำเหมืองข้อมูล (Data mining) และการเรียนรู้ของเครื่องจักร (Machine learning) ซึ่งครอบคลุมไปถึงแบบจำลองความน่าจะเป็นแบบเบส์อย่างง่าย นอกจากนั้นแล้วเรายังสามารถประยุกต์ใช้เอพีไอของเวก้าในการคำนวณความน่าจะเป็นแบบเบส์อย่างง่ายกับฐานความรู้ดังสมการที่ 25 อีกด้วยซึ่งจะกล่าวถึงในหัวข้อถัดไป

4.2 การเก็บเว็บเพจในรอบถัดไปโดยใช้ฐานความรู้ (Consecutive Crawling)

เมื่อเว็บคราวเลอร์มีฐานความรู้ก็เปรียบเสมือนกับเว็บคราวเลอร์มีประสบการณ์เกี่ยวกับลักษณะของเว็บเพจภาษาไทยมาแล้ว ดังนั้นหากเว็บคราวเลอร์เก็บเว็บเพจในรอบถัดไปแล้วใช้ฐานความรู้เข้ามาเพื่อคำนวณหาความน่าจะเป็นที่ยูอาร์แอลถัดไปจะเป็นเว็บเพจภาษาไทย แล้วเลือกเก็บเว็บเพจจากยูอาร์แอลที่มีค่าความน่าจะเป็นที่สูงสุดก่อน ก็น่าที่จะทำให้ประสิทธิภาพในการเก็บรวบรวมเว็บเพจภาษาไทยนั้นดีขึ้นเป็นอย่างมาก

ในการเก็บเว็บเพจภาษาไทยในครั้งนี้เว็บคราวเลอร์จะสกัดเอาฟีเจอร์ทั้ง 4 ชนิดจากเว็บเพจที่เพิ่งจะเก็บรวบรวมมา แล้วมาทำการคำนวณร่วมกับฐานความรู้ที่ได้จากอัลกอริทึมที่ 1 ซึ่งจะได้ผลลัพธ์คือค่าคะแนนความสำคัญที่ใช้บ่งบอกถึงความน่าจะเป็นของยูอาร์แอลแต่ละตัวจากเว็บเพจนั้นที่จะเป็นยูอาร์แอลของเว็บเพจภาษาไทย และนำยูอาร์แอลแต่ละตัวใส่ลงไปในยูอาร์แอลคิว พร้อมจัดเรียงลำดับยูอาร์แอลในคิวใหม่จากคะแนนที่มากที่สุดไปหาน้อยที่สุด เพื่อให้เว็บคราวเลอร์เลือกเก็บเว็บเพจจากยูอาร์แอลที่มีคะแนนมากที่สุดก่อนต่อไป

4.2.1 การคำนวณความน่าจะเป็นแบบเบส์อย่างง่ายกับฐานความรู้

ดังที่กล่าวถึงวิธีการคำนวณไว้บางส่วนแล้วจากหัวข้อที่ 1.4 สมการที่ 8 ดังนั้นเราจะกำหนดสมมติฐานให้ฟีเจอร์ a_1, a_2, a_3 และ a_4 ให้เป็นเหตุการณ์ที่ไม่ขึ้นต่อกัน กำหนดให้ x คือคลาสของคำตอบซึ่งมีสองคลาสคือ Th หรือ $\neg Th$ และกำหนดให้ N คือจำนวนของฟีเจอร์ซึ่งในที่นี้มีอยู่ทั้งสิ้น 4 ชนิด พิจารณาการให้คะแนนกับยูอาร์แอลของเว็บเพจ q ใด ๆ ที่ยังไม่ถูกเก็บรวบรวมมาดังนั้นเราสามารถหาความน่าจะเป็นที่ q จะเป็น Th หรือ $\neg Th$ เมื่อพิจารณาฟีเจอร์ a_i ได้ดังสมการต่อไปนี้

$$P(q \in x \mid a_i) = \underset{x \in \{Th, \neg Th\}}{P(x)} \prod_{i=1}^{N} P(a_i \mid q \in x) \quad (25)$$

เมื่อ $P(x)$ คือความน่าจะเป็นของคลาส Th หรือ $\neg Th$ ซึ่งสามารถคำนวณได้จากฐานความรู้ ส่วน $P(a_i \mid q \in x)$ คือความน่าจะเป็นของฟีเจอร์ a_i ใด ๆ โดยที่ q เป็นคลาส Th หรือ $\neg Th$ ซึ่งคำนวณได้จากฐานความรู้เช่นเดียวกัน สุดท้ายเราจะทำการนอร์มอลไลซ์คะแนนของยูอาร์แอลใหม่ให้เป็นหน่วยเดียวกันโดยคิดในเฉพาะคลาสที่จะเป็นภาษาไทย Th เท่านั้นเรียกว่าค่าคะแนนความสำคัญของยูอาร์แอล q หรือ $U(q)$ แสดงได้ดังสมการต่อไปนี้

$$U(q) = \frac{P(q \in Th \mid a_i)}{P(q \in Th \mid a_i) + P(q \in \neg Th \mid a_i)} \quad (26)$$

4.2.2 อัลกอริทึมในการเก็บเว็บเพจในรอบถัดไปโดยใช้ฐานความรู้

หัวใจของเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาไทยคือวิธีในการคำนวณค่าคะแนนความสำคัญของยูอาร์แอลแต่ละตัวที่ออกจากเว็บเพจ p ดังนั้นเราจะใช้สมการที่ 26 ทำการคำนวณค่าคะแนนความสำคัญดังกล่าว มากำหนดให้กับยูอาร์แอล q ซึ่งจะเขียนเป็นอัลกอริทึมในการเก็บรวบรวมเว็บเพจภาษาไทยโดยใช้วิธีคำนวณความน่าจะเป็นแบบเบส์อย่างง่ายได้ดังอัลกอริทึมที่ 2

จากอัลกอริทึมที่ 2 ในหน้าถัดไป เราจะเริ่มต้นอธิบายเช่นเดียวกับอัลกอริทึมที่ 1 เริ่มจากการเก็บเว็บเพจเริ่มต้นมากลุ่มหนึ่ง จากนั้นจะคัดแยกยูอาร์แอลที่ชี้ออกจากเว็บเพจกลุ่มนั้นเรียกว่า starting_urls ซึ่งปรากฏอยู่ในบรรทัดที่ 1 และกำหนดให้ $a_1[], a_2[], a_3[]$ และ $a_4[]$ คือฟีเจอร์เริ่มต้นทั้ง 4 ชนิดตามลำดับ ที่ถูกสกัดหาค่าของแต่ละฟีเจอร์จากกลุ่มของเว็บเพจเริ่มต้นเหล่านี้ ค่าของฟีเจอร์ต่าง ๆ จะถูกกำหนดให้กับยูอาร์แอลแต่ละตัวในแบบคู่ลำดับอาร์เรย์ เช่น $a_1['\text{http://www.ku.ac.th}']$ หมายถึงค่าคะแนนฟีเจอร์ a_1 ของยูอาร์แอล http://www.ku.ac.th เป็นต้น ในส่วนของฟังก์ชั่น Calculate_URL_score ในบรรทัดที่ 3 และ 13 จะทำหน้าที่คำนวณหาค่าคะแนนความสำคัญของยูอาร์แอลแต่ละตัวโดยพิจารณาจากฟีเจอร์ทั้ง 4 ชนิดและใช้วิธีการคำนวณความน่าจะเป็นแบบเบส์อย่างง่ายดังสมการที่ 26

อัลกอริทึมที่ 2 การค้นหาเว็บเพจภาษาไทยโดยใช้ฐานความรู้

```
 1:  NaiveBayes_LanguageSpecificWebCrawling (starting_urls, a1[], a2[], a3[], a4[]) {
 2:        forall(v in starting_urls) {
 3:              score = Calculate_URL_score(a1[v], a2[v], a3[v], a4[v])
 4:              Enqueue(PRIORITY_QUEUE, v, score)
 5:        }
 6:        while(PRIORITY_QUEUE is not empty) {
 7:              url = Dequeue (PRIORITY_QUEUE)
 8:              p = Fetch_Page(url)
 9:              extract features a1 and a4 from p
10:              forall(q in Extract_URL(p)) {
11:                    discard q when q is visited
12:                    extract features a2 and a3 from q
13:                    score = Calculate_URL_score(a1, a2, a3, a4)
14:                    Enqueue(PRIORITY_QUEUE, q, score)
15:              }
16:              Reorder_Queue(PRIORITY_QUEUE)
17:        }
18:  }
```

บรรทัดที่ 3 ตัวแปร PRIORITY_QUEUE หมายถึงยูอารแอลคิวนี้จะเรียงลำดับการเขาออกของยูอารแอลตามค่าความสำคัญ เนื่องจากคาความสำคัญบงบอกถึงความเกี่ยวของกับเว็บเพจภาษาไทยดังนั้นจึงสามารถเรียงลำดับยูอาร์แอลได้ใหม่ตามค่าคะแนนความสำคัญได้ บรรทัดที่ 7 คือเว็บคราวเลอร์จะนำยูอาร์แอลออกมาจากยูอาร์แอลคิวตามค่าคะแนนความสำคัญที่สูงที่สุดก่อน สำหรับบรรทัดที่ 9 และ 12 เป็นการสกัดเอาฟีเจอร์ทั้งหมด 4 ชนิดจากเว็บเพจ p ที่ได้เก็บรวบรวมมาแล้ว และคำอธิบายในบรรทัดที่ 11 คือเว็บคราวเลอร์จะทิ้งยูอาร์แอล q เมื่อพบว่าเคยเก็บเว็บเพจจากยูอาร์แอลนี้ไปแล้ว สุดท้ายคือฟังก์ชั่น Reorder_Queue ทำหน้าที่ในการจัดเรียงลำดับคะแนนความสำคัญของยูอาร์แอลแต่ละตัวในยูอาร์แอลคิวใหม่จากมากที่สุดไปหาน้อยที่สุดตามลำดับ

5. การกำหนดค่าในการทดลอง

สำหรับการวัดประสิทธิภาพเราจะวัดวิธีการที่ได้นำเสนอเปรียบเทียบกับอีก 2 วิธีคือวิธีค้นหาเว็บเพจภาษาไทยแบบดั้งเดิมและวิธีการค้นหาในแนวกว้างซึ่งมีรายละเอียดในการกำหนดค่าก่อนการทดลองได้ดังต่อไปนี้

5.1 วิธีการที่นำเสนอ (Naïve Bayes-based Language-specific Web Crawling: NB-LSWC)

เริ่มต้นเราจะสร้างฐานความรู้ขึ้นมาก่อนโดยใช้อัลกอริทึมที่ 1 เพื่อทำการสกัดความรู้จากเว็บเพจที่อยู่บนชุดข้อมูลเว็บกราฟ (Web graph dataset) ยูอาร์แอลเริ่มต้นและค่าคะแนนของฟีเจอร์ทั้ง 4 ชนิดที่แสดงดังตารางที่ 13 จะเกิดจากการคัดแยกและคำนวณมาจากเว็บเพจที่เก็บมาแล้วในชุดข้อมูลเว็บกราฟ เหตุผลที่เลือกกลุ่มยูอาร์แอลเริ่มต้นกลุ่มนี้เพราะว่าเราต้องการให้ฐานความรู้มีเว็บเพจที่เป็นภาษาไทยและภาษาอื่นในอัตราส่วนอย่างละเท่า ๆ กัน ในขณะที่อัลกอริทึมสกัดความรู้กำลังประมวลผลที่เว็บเพจใดจะต้องทำเครื่องหมายลงบนเว็บเพจนั้นว่าถูกนำไปสร้างเป็นฐานความรู้แล้ว (Mark trained) และในการทดลองอื่นจะไม่สามารถใช้เว็บเพจที่ถูกทำเครื่องหมายนี้นำไปใช้ในการวัดผลได้

ตารางที่ 13 ยูอาร์แอลเริ่มต้นสำหรับสร้างฐานความรู้

ลำดับ	ยูอาร์แอลเริ่มต้น	a_1	a_2	a_3	a_4
1	http://www.nectec.or.th/	0	0	1	1
2	http://kanchanapisek.or.th/	0	0	1	1
3	http://www.kas.de/proj/home/links/6/2/index.html	0	0	0	1
4	http://myprops.org/channels/wesmirch-hollywood-gossip/	0	0	0	1
5	http://www.amnesty.or.th/main/default.asp	0	0	1	1

ฐานความรู้ที่สร้างเรียบร้อยแล้วจะประกอบด้วยเว็บเพจเป็นจำนวน 10000 เว็บเพจตามที่ได้กำหนดไว้แล้วในตัวแปร max_pages จากอัลกอริทึมที่ 1 โดยทั้ง 10000 เว็บเพจนี้เมื่อผ่านการตรวจสอบภาษาแล้วจะประกอบไปด้วยเว็บเพจในคลาสภาษาไทยจำนวน 6939 เว็บเพจ และคลาสที่เป็นภาษาอื่นจำนวน 3061 เว็บเพจตามลำดับ

เมื่อสกัดความรู้จากเว็บเพจครบเป็นจำนวนตามที่กำหนดแล้วจะทำการตรวจสอบความถูกต้องของฐานความรู้นั้นโดยใช้เวก้าโดยตั้งค่าโปรแกรมให้แบ่งข้อมูลเพื่อสอนและทดสอบ (Train-and-Test) กับแบบจำลองเป็นจำนวนทั้งสิ้น 10 ชุด (10-folds cross validation) โดยเลือกใช้แบบจำลองความน่าจะเป็นแบบเบส์อย่างง่ายในการทดสอบ

ผลการวัดความถูกต้องของฐานความรู้โดยใช้เวก้าจากตารางในภาคผนวกที่1 พบว่าให้ค่าเอฟ (F-measure) เมื่อถูกจำแนกเป็นเว็บเพจภาษาไทยเป็น 0.911 และเมื่อถูกจำแนกเป็นภาษาอื่นเป็น 0.764 ดังนั้นฐานความรู้นี้ให้ค่าความแม่นยำในระดับที่ดี จึงเหมาะสมที่จะนำมาใช้ในขั้นตอนการเก็บเว็บเพจครั้งถัดไปในอัลกอริทึมที่ 2 ได้เป็นอย่างดี เพราะฉะนั้นในขั้นตอนการทดลองจะใช้ฐานความรู้ที่สร้างเสร็จเรียบร้อยแล้ว นำไปวัดผลในหัวข้อของทุกการทดลองทั้งหมดตามลำดับ

สำหรับการวัดผลประสิทธิภาพเราจะตั้งค่าพารามิเตอร์ในอัลกอริทึมที่ 2 โดยกำหนดให้ยูอาร์แอลเริ่มต้น ค่าของฟีเจอร์ทั้ง 4 ชนิดและคะแนนของยูอาร์แอลเริ่มต้นเป็นดังตารางที่ 13 14 และ 15 ตามลำดับ ซึ่งจะกล่าวถึงในหัวข้อของผลการทดลองต่อไป

5.2 วิธีค้นหาเว็บเพจภาษาไทยแบบดั้งเดิม (Language-specific Web Crawling: LSWC)

เราได้พัฒนาระบบการค้นหาเว็บเพจภาษาไทยตามวิธีการที่ Somboonviwat *et al.* (2005) ซึ่งได้นำเสนอไว้ในงานวิจัยชื่อ “Finding Thai web page in foreign web spaces” เนื่องจากวิธีการของงานวิจัยดังกล่าวนำเสนอในรูปแบบของแนวความคิด ซึ่งไม่ได้ใช้รหัสเทียมในการนำเสนอ ดังนั้นเราจึงได้พยายามแปลงแนวความคิดดังกล่าวในอยู่ในรูปแบบของภาษาโปรแกรมเท่าที่จะสามารถถอดความหมายได้ จากรายละเอียดการตั้งค่าของวิธีการดังกล่าวจะกำหนดค่าระยะทางที่ห่างจากเว็บเพจภาษาไทยต้นทางที่มากที่สุด $T = 5$ และค่าจำนวนเว็บเพจที่เป็นภาษาอื่นที่มากที่สุดสำหรับตัดสินว่าเซิร์ฟเวอร์นั้นไม่ใช่เซิร์ฟเวอร์ของภาษาไทยคือ $S = 3$

5.3 วิธีการค้นหาในแนวกว้าง (Breadth-First Search: BFS)

ตามวิธีค้นหาในแนวกว้างเราจะเปลี่ยนจากยูอาร์แอลคิวจากเดิมที่เป็นแบบให้ความสำคัญ (Priority Queue) มาเป็นยูอาร์แอลคิวแบบเข้าก่อนออกก่อน (First-In First Out: FIFO) ดังนั้นเว็บคราวเลอร์จะเลือกเก็บเว็บเพจตามลำดับของยูอาร์แอลที่เข้ามาก่อนเท่านั้น ซึ่งเราจะใช้วิธีค้นหาในแนวกว้างเพื่อการวิเคราะห์ถึงประสิทธิภาพที่เป็นพื้นฐาน (Baseline) ของเว็บคราวเลอร์ได้

ผลและวิจารณ์

วิทยานิพนธ์นี้จะแบ่งการทดสอบประสิทธิภาพของวิธีการที่นำเสนอและเปรียบเทียบกับวิธีการอื่นภายใต้สภาพแวดล้อมอยู่ 2 แบบ ได้แก่แบบแรกคือทดสอบบนชุดข้อมูลเว็บกราฟเพื่อหาค่าอัตราการเก็บเกี่ยวและหาค่าความครอบคลุม ส่วนแบบที่สองคือทดสอบบนอินเทอร์เน็ตเพื่อหาค่าอัตราการเก็บเกี่ยวเพียงอย่างเดียว ส่วนค่าความครอบคลุมไม่สามารถหาได้เนื่องจากเราไม่ทราบปริมาณของเว็บเพจภาษาไทยทั้งหมดในอินเทอร์เน็ต นอกจากนั้นแล้วเพื่อแสดงให้เห็นว่าเว็บคราวเลอร์ที่เราได้นำเสนอนั้นสามารถทำงานภายใต้สภาพแวดล้อมจริงได้

1. การทดสอบบนชุดข้อมูลเว็บกราฟ (Test on Web graph dataset)

แบ่งการทดสอบเป็น 3 แบบได้แก่การทดสอบประสิทธิภาพเมื่อกำหนดยูอาร์เริ่มต้นเป็นภาษาไทยกลุ่มที่ 1 (Thai1) การทดสอบประสิทธิภาพเมื่อกำหนดยูอาร์เริ่มต้นเป็นภาษาไทยกลุ่มที่ 2 (Thai2) การทดสอบประสิทธิภาพเมื่อกำหนดยูอาร์เริ่มต้นเป็นภาษาอังกฤษ (English) ซึ่งจะกล่าวถึงในหัวข้อถัดไป

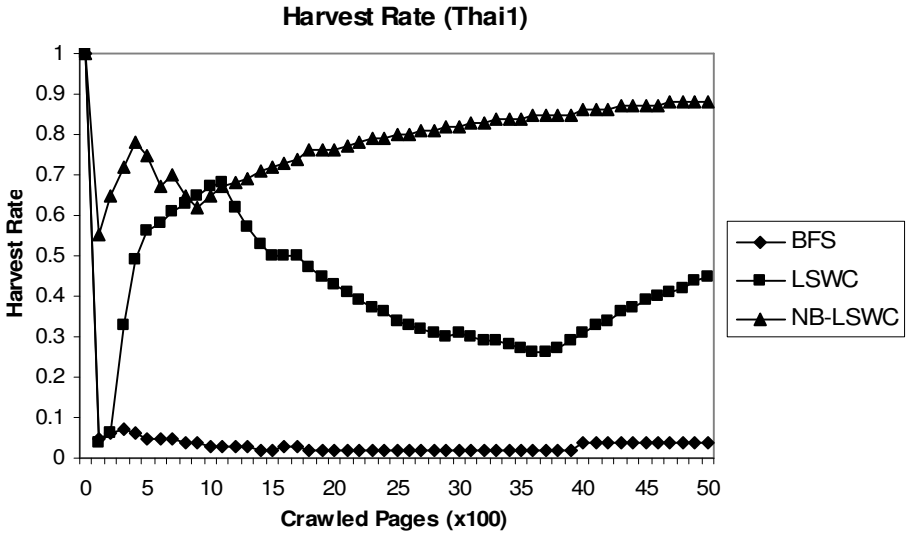
1.1 การทดสอบประสิทธิภาพเมื่อกำหนดยูอาร์เริ่มต้นเป็นภาษาไทยกลุ่มที่ 1 (Thai1)

เป็นการทดสอบประสิทธิภาพของแต่ละวิธีการเมื่อกำหนดให้กลุ่มของยูอาร์แอลเริ่มต้นซึ่งเป็นยูอาร์แอลของเว็บเพจภาษาไทยกลุ่มที่ 1 ซึ่งเป็นยูอาร์แอลที่เซิร์ฟเวอร์ของเว็บเพจนั้นประกอบด้วยเว็บเพจทั้งภาษาไทยและภาษาอื่นอยู่เป็นจำนวนมาก เช่น เว็บบล็อก (Web blog) เป็นต้น โดยเราจะคำนวณฟีเจอร์เริ่มต้นและคะแนนความสำคัญเริ่มต้นแล้วนำมากำหนดให้กับยูอาร์แอลต่างๆ ดังแสดงดังตารางที่ 14

ตารางที่ 14 ยูอาร์แอลเริ่มต้นที่เป็นเว็บเพจภาษาไทยกลุ่มที่ 1

ลำดับ	ยูอาร์แอล	a_1	a_2	a_3	a_4	*score*
1	http://n-natta.blogspot.com/2009/03/once-upon-midnight-dreary-while-i.html	0	0	0	0	0
2	http://rikker.blogspot.com/2009/03/thai-101-giveaway-by.html	0.03	0.33	0	0	0.01
3	http://nhrcthai.wordpress.com	0.01	0.5	0	0	0.01

ทำการวัดผลโดยใช้วิธีการที่นำเสนออัลกอริทึมที่ 2 (NB-LSWC) โดยผ่านการตั้งค่าพารามีเตอร์และฐานความรู้ดังที่กล่าวมาในหัวข้อที่แล้ว วัดผลเปรียบเทียบกับวิธีค้นหาเว็บเพจภาษาไทยแบบดั้งเดิม (LSWC) และวิธีการค้นหาในแนวกว้าง (BFS) ซึ่งผลลัพธ์จะเป็นกราฟอัตราการเก็บเกี่ยวและความครอบคลุมซึ่งแสดงในภาพที่ 19

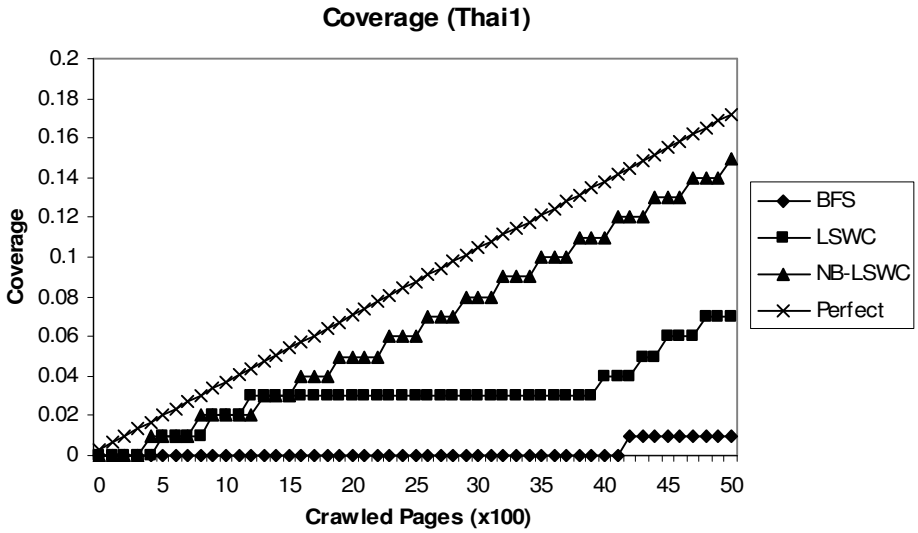


ภาพที่ 19 กราฟแสดงอัตราการเก็บเกี่ยวเมื่อกำหนดยูอาร์แอลเริ่มต้นเป็นเว็บเพจภาษาไทยกลุ่มที่ 1 และวัดผลบนชุดข้อมูลทดสอบ

จากผลการทดสอบในกราฟภาพที่ 19 เนื่องจากยูอาร์แอลเริ่มต้นเป็นเว็บเพจภาษาไทย ดังนั้นจะพบว่าเว็บคราวเลอร์ทุกวิธีการจะเริ่มค้นพบเว็บเพจภาษาไทยประมาณช่วงต้นของการเก็บเว็บเพจใกล้เคียงกัน อย่างไรก็ตามหากพิจารณาวิธีการค้นหาเว็บเพจภาษาไทยที่เราได้นำเสนอจะพบว่าเส้นของกราฟได้แกว่งอยู่ที่ประมาณ 0.55 ถึง 0.8 ในช่วงของการเก็บ 1000 เว็บเพจ และเริ่มคงที่ประมาณ 0.7 ถึง 0.85 เมื่อเก็บเว็บเพจตั้งแต่ 1000 เว็บเพจเป็นต้นไป สาเหตุที่ในช่วงเริ่มต้นเส้นกราฟยังมีการแกว่งนั้น เนื่องมาจากมีเว็บภาษาอังกฤษปะปนอยู่ในเซิร์ฟเวอร์ของยูอาร์แอลเริ่มต้นเป็นจำนวนมาก ทำให้ยังไม่สามารถค้นหาเว็บเพจภาษาไทยกลุ่มใหญ่จำนวนมากอย่างต่อเนื่องได้ ในขณะที่วิธีการแบบดั้งเดิมจะพบว่าในช่วง 100-300 เว็บเพจแรกเว็บคราวเลอร์หาเว็บภาษาไทยได้น้อยมากซึ่งอัตราการเก็บเกี่ยวอยู่ที่ 0.05 เท่านั้นและเส้นกราฟจะเริ่มไต่ขึ้นเมื่อเก็บเว็บเพจไปแล้วเป็นจำนวนถึง 300 เว็บเพจประสิทธิภาพจึงเริ่มดีขึ้น

เมื่อนำผลลัพธ์มาเปรียบเทียบกันจะพบว่าวิธีที่เรานำเสนอให้ประสิทธิภาพในด้านอัตราเก็บเกี่ยวได้ดีกว่าอย่างเห็นได้ชัด สาเหตุหนึ่งที่วิธีการแบบดั้งเดิมให้ผลลัพธ์ที่ไม่ดี เพราะว่ายูอาร์แอลเริ่มต้นที่กำหนดให้ส่วนหนึ่งเป็นยูอาร์แอลของเว็บเพจภาษาไทยที่อยู่ในเซิร์ฟเวอร์ที่เป็นของต่างประเทศ เช่น เว็บบล็อก (Web blog) หรือเว็บท่าสำหรับให้บริการด้านโฮสต์ (Web hosting) เป็นต้น ดังนั้นในเซิร์ฟเวอร์นั้นจะประกอบไปด้วยเว็บเพจภาษานานาชาติเป็นจำนวนมาก เมื่อวิธีการแบบดั้งเดิมวิเคราะห์เซิร์ฟเวอร์นั้นว่าเป็นภาษาไทยซึ่งผิดไปจากความเป็นจริง จึงเป็นเหตุให้ได้เว็บเพจภาษาอื่นปะปนมาด้วยเป็นจำนวนมากทำให้ประสิทธิภาพไม่ดีไปกว่าวิธีการที่เราได้นำเสนอในวิทยานิพนธ์นี้นั่นเอง

จากผลการทดสอบความครอบคลุมในภาพที่ 20 จากวิธีการที่เราได้นำเสนอพบว่าเส้นกราฟความครอบคลุมใกล้เคียงไปกับเส้นในอุดมคติ (Perfect) อย่างไรก็ตามยังมีความห่างจากเส้นจากเส้นในอุดมคติอยู่บ้าง แสดงให้เห็นว่ามีประสิทธิภาพในการค้นพบเว็บเพจภาษาไทยค่อนข้างดีพอสมควร เมื่อเปรียบเทียบกับวิธีการอื่นจะพบว่าวิธีการที่เราได้นำเสนอให้ผลความครอบคลุมต่อเว็บเพจภาษาไทยได้ดีกว่าวิธีการอื่น ดังนั้นจึงเป็นสิ่งที่ยืนยันได้ว่าถึงแม้จะเป็นยูอาร์แอลของเว็บเพจภาษาไทยที่อยู่ในเซิร์ฟเวอร์ต่างประเทศ ก็จะไม่ส่งผลต่อวิธีการค้นหาเว็บเพจภาษาไทยที่เรานำเสนอได้มากนัก



ภาพที่ 20 กราฟแสดงความครอบคลุมเมื่อกำหนดยูอาร์แอลเริ่มต้นเป็นเว็บเพจภาษาไทยกลุ่มที่ 1 และวัดผลบนชุดข้อมูลทดสอบ

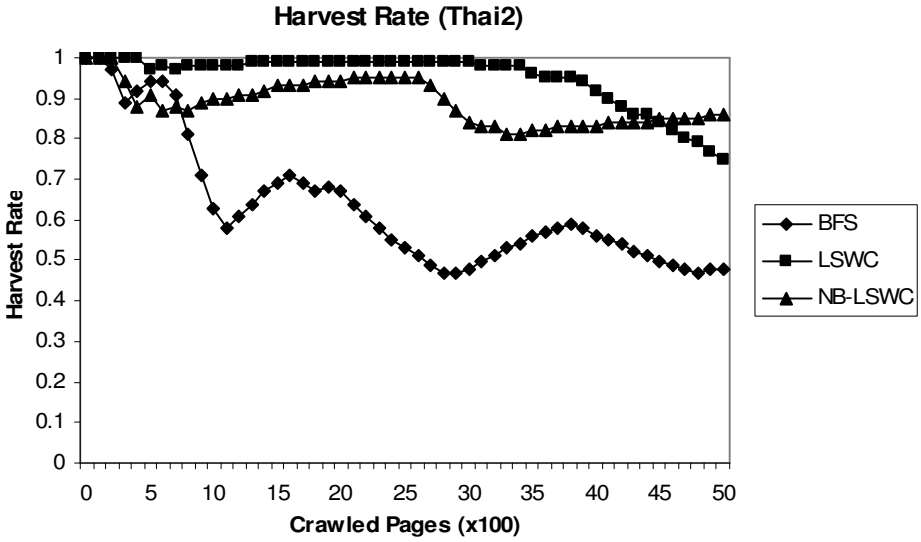
1.2 การทดสอบประสิทธิภาพเมื่อยูอาร์เริ่มต้นเป็นภาษาไทยกลุ่มที่ 2 (Thai2)

การทดลองนี้จะเป็นการทดสอบประสิทธิภาพของแต่ละวิธีการเมื่อกำหนดให้กลุ่มของยูอาร์แอลเริ่มต้นซึ่งเป็นยูอาร์แอลของเว็บเพจภาษาไทยกลุ่มที่ 2 กล่าวคือเป็นยูอาร์แอลของเซิร์ฟเวอร์ที่เป็นภาษาไทยและประกอบไปด้วยเว็บเพจภาษาไทยที่อยู่ในเซิร์ฟเวอร์นั้นเป็นจำนวนมาก ซึ่งรายละเอียดของยูอาร์แอลเริ่มต้น ค่าของฟีเจอร์ทั้ง 4 ชนิดและค่าคะแนนความสำคัญของยูอาร์แอลแสดงดังตารางที่ 15

ตารางที่ 15 ยูอาร์แอลเริ่มต้นที่เป็นเว็บเพจภาษาไทยกลุ่มที่ 2

ลำดับ	ยูอาร์แอล	a_1	a_2	a_3	a_4	*score*
1	http://news.mthai.com	0.55	1	0	0.69	0.88
2	http://dir.narak.com/news_and_media	0.43	1	0	0.69	0.83
3	http://www.thaibiznews.net	0	0	0	0	0
4	http://news.sanook.com/recommended/index3.php	0.50	0	0	0.69	0.26
5	http://www.suthichaiyoon.com	0	0	0	0	0

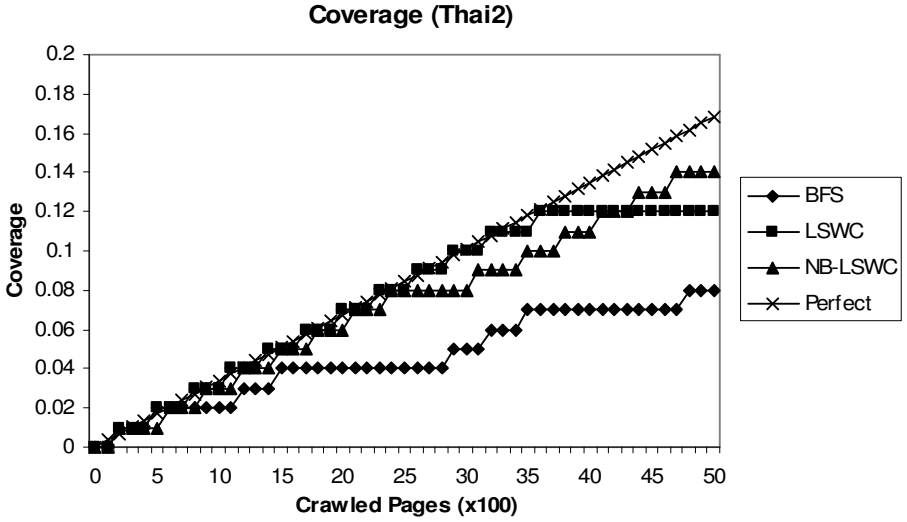
ทำการวัดผลโดยอ้างอิงวิธีการตั้งค่าและเปรียบเทียบผลลัพธ์กับวิธีการอื่นเช่นเดียวกับการทดลองที่ 1 สำหรับกราฟผลการทดสอบจะแสดงดังภาพที่ 21



ภาพที่ 21 กราฟแสดงอัตราการเก็บเกี่ยวเมื่อกำหนดยูอาร์แอลเริ่มต้นเป็นเว็บเพจภาษาไทยกลุ่มที่ 2 และวัดผลบนชุดข้อมูลทดสอบ

จากผลการทดสอบในภาพที่ 21 เว็บคราวเลอร์ทุกวิธีการจะเริ่มพบเว็บเพจภาษาไทยทันทีที่ได้เก็บเว็บเพจจากยูอาร์แอลเริ่มต้นที่กำหนดให้ ซึ่งดูได้จากค่าอัตราการเก็บเกี่ยวขึ้นสูงอยู่ในระดับ 1.0 ในช่วง 100-300 เว็บเพจแรกของการเริ่มต้นและลดต่ำลงตามลำดับ เมื่อพิจารณาดูลักษณะของกราฟจากวิธีการค้นหาแนวกว้าง (BFS) พบว่ายูอาร์แอลเริ่มต้นที่เรากำหนดให้เป็นยูอาร์แอลที่นำพาไปสู่กลุ่มของเว็บเพจภาษาไทยเป็นจำนวนมากซึ่งมีค่าเฉลี่ยในการพบเว็บเพจภาษาไทยมีสูงถึง 60-70% ถึงแม้ว่าเราจะไม่ใช้วิธีการจัดลำดับคะแนนในยูอาร์แอลคิวเลยก็ตามซึ่งเป็นเครื่องบ่งชี้ว่าเส้นทางนี้เป็นเส้นทางของเว็บเพจภาษาไทยอย่างแท้จริง เมื่อพิจารณาเปรียบเทียบวิธีการค้นหาเว็บเพจภาษาไทยแบบดั้งเดิมกับวิธีการที่เราได้นำเสนอจะพบว่าวิธีการแบบดั้งเดิมให้ประสิทธิภาพในการเก็บเว็บเพจภาษาไทยได้ดีกว่า ซึ่งคงอัตราการเก็บเกี่ยวไว้ได้ 1.0 ตั้งแต่ 0 จนไปถึง 3500 เว็บเพจ ในขณะที่วิธีการของเราให้ผลในการค้นหาที่ไม่ดีไปกว่าวิธีการแบบดั้งเดิม แต่ในภาพรวมยังอยู่ในระดับ 0.9 ซึ่งถือเป็นระดับที่ดีเช่นเดียวกัน สาเหตุหนึ่งที่ทำให้วิธีการแบบดั้งเดิมดีกว่าวิธีการที่เรานำเสนอนั้น เนื่องจากยูอาร์แอลเริ่มต้นกลุ่มนี้เป็นยูอาร์แอลที่มีปริมาณเว็บเพจภาษาไทยที่อยู่ใน

เซิร์ฟเวอร์ภาษาไทยจำนวนมาก ดังนั้นถ้าหากวิธีการดั้งเดิมตรวจสอบว่าเซิร์ฟเวอร์ของเว็บเพจเหล่านี้เป็นเซิร์ฟเวอร์ของเว็บเพจภาษาไทย เว็บเพจทั้งหมดภายใต้เซิร์ฟเวอร์นั้นก็จะถูกเก็บรวมมาทั้งหมด ดังนั้นความแม่นยำในการตรวจสอบเซิร์ฟเวอร์ภาษาไทยจึงเป็นปัจจัยหลักในการเก็บเว็บเพจที่ควรคำนึงถึงด้วย หากระบุเซิร์ฟเวอร์ได้ถูกต้องค่าของอัตราการเก็บเกี่ยวจะมีค่าสูงแต่ในทางตรงกันข้ามหากระบุเซิร์ฟเวอร์ไม่ถูกต้องก็จะได้ค่าของอัตราการเก็บเกี่ยวที่น้อยส่งผลให้เว็บครอว์เลอร์เก็บเว็บเพจภาษาไทยน้อยกว่าที่ควรจะเป็น



ภาพที่ 22 กราฟแสดงความครอบคลุมเมื่อกำหนดยูอาร์แอลเริ่มต้นเป็นเว็บเพจภาษาไทยกลุ่มที่ 2 และวัดผลบนชุดข้อมูลทดสอบ

จากผลการทดสอบความครอบคลุมในภาพที่ 22 พบว่าวิธีการแบบดั้งเดิมให้ผลลัพธ์ความครอบคลุมเทียบเท่ากับเส้นในอุดมคติตั้งแต่ 0 ถึง 3500 เว็บเพจแรก ซึ่งให้บ่งบอกถึงความครอบคลุมต่อเว็บเพจภาษาไทยที่ดีกว่าวิธีการที่เรานำเสนออย่างเห็นได้ชัด แสดงให้เห็นว่าการเก็บเว็บเพจภาษาไทยทั้งหมดที่อยู่ภายใต้เซิร์ฟเวอร์ที่เป็นภาษาไทย สามารถทำให้ทั้งอัตราการเก็บเกี่ยวและความครอบคลุมเพิ่มสูงขึ้นได้อย่างต่อเนื่อง อย่างไรก็ตามปัจจัยที่สำคัญอย่างหนึ่งคือการตั้งค่าพารามิเตอร์สำหรับตัดสินว่าเป็นเซิร์ฟเวอร์ภาษาไทยให้เหมาะสมนั้น ล้วนแต่เป็นสิ่งที่ควรคำนึงถึงด้วยเช่นเดียวกัน

1.3 การทดสอบประสิทธิภาพเมื่อยูอาร์เริ่มต้นเป็นภาษาอังกฤษ (English)

การทดลองนี้จะเป็นการทดสอบประสิทธิภาพของแต่ละวิธีการเมื่อกำหนดให้กลุ่มของยูอาร์แอลเริ่มต้นซึ่งเป็นยูอาร์แอลของเว็บเพจภาษาอังกฤษ เพื่อทดสอบประสิทธิภาพในการค้นหาเว็บเพจภาษาไทยในระดับที่มีความยากในการค้นหาเพิ่มขึ้นแต่ยังอยู่ในขอบเขตที่ทุกวิธีการยังสามารถค้นหาได้ ซึ่งรายละเอียดของยูอาร์แอลเริ่มต้น ค่าของฟีเจอร์ทั้ง 4 ชนิดและค่าคะแนนความสำคัญของยูอาร์แอลแสดงดังตารางที่ 16 อย่างไรก็ในบางยูอาร์แอลที่อยู่ในกลุ่มเริ่มต้นดังกล่าวมีช่องทางที่เปิดโอกาสให้เว็บคราวเลอร์ออกไปสู่เว็บเพจภาษาไทยได้ เพื่อให้มีระดับของความยากไม่มากจนเกินไป

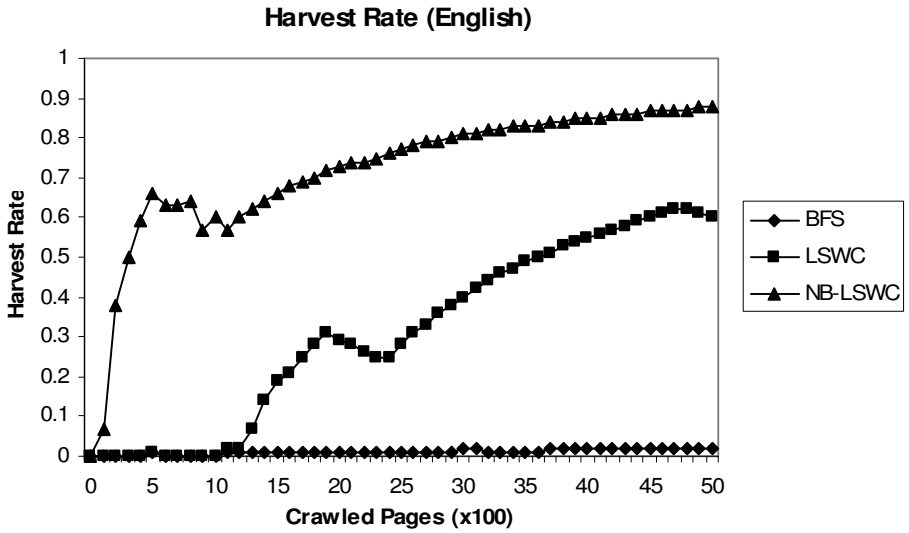
ตารางที่ 16 ยูอาร์แอลเริ่มต้นที่เป็นเว็บเพจภาษาอังกฤษ

ลำดับ	ยูอาร์แอล	a_1	a_2	a_3	a_4	*score*
1	http://www.my-thai-friend.com	0	0	0	0	0
2	http://www.carrentals.co.uk	0	0	0	0	0
3	http://www.geocities.com/amazingth/thailinks.html	0	0	0	0	0
4	http://www.asiamedia.ucla.edu	0	0	0	0	0
5	http://www.the-freight.com	0	0	0	0	0

ทำการวัดผลโดยอ้างอิงวิธีการตั้งค่าและเปรียบเทียบผลลัพธ์กับวิธีการอื่นเช่นเดียวกับการทดลองที่ผ่านมาสำหรับกราฟผลการทดสอบจะแสดงดังภาพที่ 23

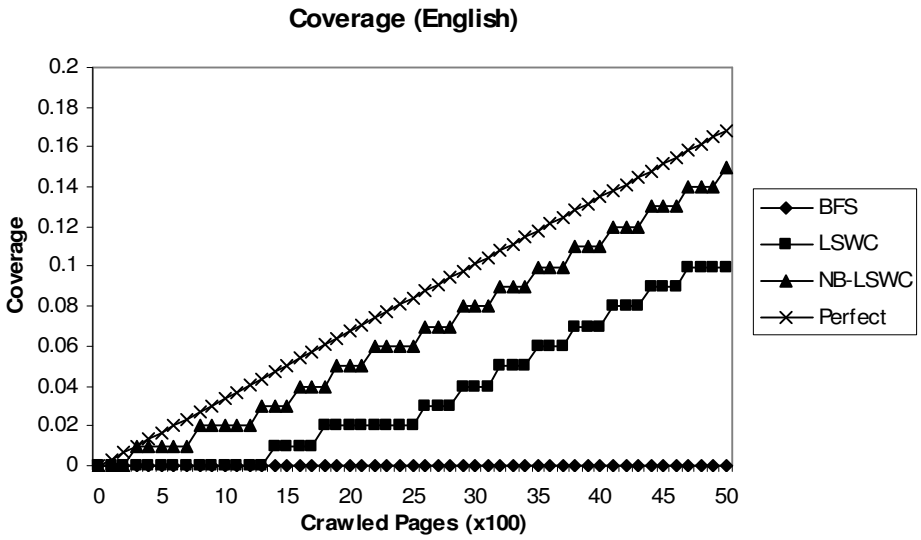
จากกราฟผลการทดลองในภาพที่ 23 ในหน้าถัดไป พิจารณากราฟของวิธีที่เรานำเสนอพบว่าเส้นกราฟจะเริ่มพบเว็บเพจภาษาไทยตั้งแต่ 100 เว็บเพจแรกเป็นต้นไป และมีแนวโน้มสูงขึ้นอย่างต่อเนื่องจนเริ่มคงที่ที่ระดับ 0.75 ถึง 0.8 เมื่อเก็บเว็บเพจตั้งแต่ 1000 เว็บเพจเป็นต้นไป พิจารณาเปรียบเทียบกับวิธีการค้นหาเว็บเพจภาษาไทยแบบดั้งเดิมพบว่าวิธีการที่เราได้นำเสนอให้ผลลัพธ์ที่ดีกว่าเป็นอย่างมาก สาเหตุส่วนหนึ่งเกิดจากการพบยูอาร์แอลที่มีอินเตอร์เน็ตโดเมน ".th" ได้เร็วกว่าวิธีการแบบดั้งเดิมจึงทำให้เข้าไปเจอกลุ่มเว็บเพจภาษาไทยกลุ่มใหญ่ได้เร็วมากยิ่งขึ้น ในทางกลับกันวิธีการแบบดั้งเดิมไม่ได้ใช้ฟีเจอร์นี้จึงทำให้การพบเจอเว็บเพจภาษาไทยกลุ่มแรกช้ากว่าวิธีการของเรา จึงอาจพบข้อสรุปได้บางอย่างว่าถึงแม้จะมีเว็บเพจจำนวนมากอยู่

นอกเหนือจากอินเทอร์เน็ตโดเมนดังกล่าว ทำให้เราไม่อาจจะใช้ฟีเจอร์นี้เป็นฟีเจอร์หลักในการค้นหาเว็บเพจภาษาไทยได้ แต่อย่างไรก็ตามถ้าหากเว็บคราวเลอร์ตกไปอยู่ในเส้นทางของเว็บเพจภาษาอื่นเป็นจำนวนมากทำให้ไม่สามารถกลับเข้ามาสู่เส้นทางที่เป็นเว็บเพจภาษาไทยได้ ดังนั้นฟีเจอร์นี้จะช่วยควบคุมให้เว็บคราวเลอร์กลับเข้ามาสู่เส้นทางที่เป็นเว็บเพจภาษาไทยได้เป็นอย่างดี



ภาพที่ 23 กราฟอัตราการเก็บเกี่ยวเมื่อกำหนดยูอาร์แอลเริ่มต้นเป็นเว็บเพจภาษาอังกฤษและวัดผลบนชุดข้อมูลทดสอบ

จากผลการทดสอบความครอบคลุมในภาพที่ 24 ในหน้าถัดไป ถึงแม้ยูอาร์แอลเริ่มต้นจะเป็นของเว็บเพจภาษาอังกฤษ แต่วิธีการที่เราได้นำเสนอจะพบว่าเส้นกราฟความครอบคลุมยังมีความห่างจากเส้นจากเส้นในอุดมคติอยู่เล็กน้อย แสดงให้เห็นว่ามีประสิทธิภาพในการค้นพบเว็บเพจภาษาไทยยังอยู่ในระดับที่ดีพอสมควร เมื่อเปรียบเทียบกับวิธีการอื่นจะพบว่าวิธีการที่เราได้นำเสนอให้ผลความครอบคลุมต่อเว็บเพจภาษาไทยได้ดีกว่าวิธีการอื่นอย่างเห็นได้ชัด ซึ่งปัจจัยหนึ่งที่ทำให้เราค้นหาเว็บเพจภาษาไทยได้อย่างรวดเร็วเกิดจากฟีเจอร์อินเทอร์เน็ตโดเมนของเว็บเพจภาษาไทยนั่นเอง



ภาพที่ 24 กราฟความครอบคลุมเมื่อกำหนดยูอาร์แอลเริ่มต้นเป็นเว็บเพจภาษาอังกฤษและวัดผลบนชุดข้อมูลทดสอบ

สรุปผลการทดสอบบนชุดข้อมูลเว็บกราฟจะเห็นว่าวิธีการที่เรานำเสนอในวิทยานิพนธ์นี้จะให้ผลลัพธ์ที่ดีกว่าวิธีการแบบดั้งเดิม เมื่อระดับในการค้นหามีความยากขึ้น เช่น กรณีที่เว็บคราวเลอร์ตกอยู่ในกลุ่มของเว็บเพจภาษาอื่นเป็นจำนวนมาก เป็นต้น ซึ่งถือเป็นข้อดีเป็นอย่างมาก เนื่องจากทำให้เว็บคราวเลอร์สามารถที่จะกลับมายังเส้นทางของเว็บเพจภาษาไทยได้ ทำให้ผลลัพธ์ของอัตราการเก็บเกี่ยวยังคงสูงได้อย่างต่อเนื่อง เมื่อพิจารณาวิธีการค้นหาเว็บเพจภาษาไทยแบบดั้งเดิมจะเห็นว่าวิธีการนี้มีข้อดีอย่างหนึ่งซึ่งถือเป็นจุดแข็งก็คือ อัตราการเก็บเกี่ยวจะสูงมากถ้าเก็บเว็บเพจทั้งหมดที่อยู่ภายใต้เซิร์ฟเวอร์ของเว็บเพจภาษาไทย ดังนั้นถ้าหากระบุเซิร์ฟเวอร์ใด ๆ ว่าเป็นเซิร์ฟเวอร์ของเว็บเพจภาษาไทยอย่างแม่นยำ จะทำให้ประสิทธิภาพในการเก็บรวบรวมเว็บเพจภาษาไทยนั้นสูงขึ้นได้

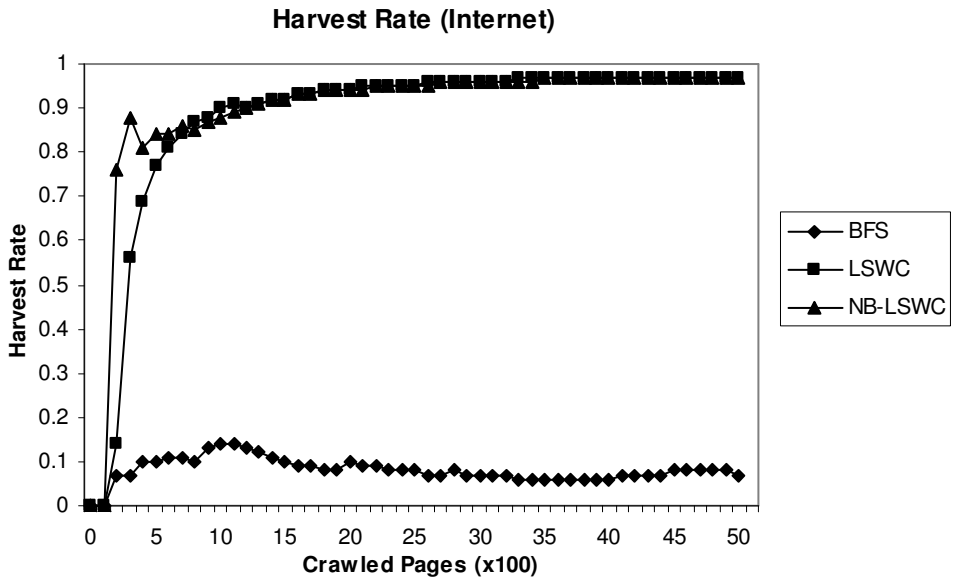
2. การทดสอบบนอินเทอร์เน็ต (Test on Internet)

เราจะทดสอบประสิทธิภาพของแต่ละวิธีการบนอินเทอร์เน็ตเพื่อศึกษาค่าอัตราการเก็บเกี่ยวเมื่อทดสอบภายใต้สภาพแวดล้อมจริง โดยกำหนดให้ยูอาร์แอลเริ่มต้นเป็นยูอาร์แอลภาษาอังกฤษที่มีลิงค์ชี้เข้าไปยังเว็บเพจภาษาไทย 1 ระดับ ซึ่งรายละเอียดของยูอาร์แอลเริ่มต้น ค่าของฟีเจอร์ทั้ง 4 ชนิดและค่าคะแนนความสำคัญของยูอาร์แอลแสดงดังตารางที่ 17

ตารางที่ 17 ยูอาร์แอลเริ่มต้นที่เป็นเว็บเพจภาษาอังกฤษสำหรับใช้ทดสอบบนอินเทอร์เน็ต

ลำดับ	ยูอาร์แอล	a_1	a_2	a_3	a_4	*score*
1	http://myprops.org/content/Man-found-dead-mouse-in-malt-loaf	0	0	0	0	0
2	http://www.thaiwebsites.com/portal.asp	0	0	0	0	0
3	http://news.carrentals.co.uk/bangkok-high-speed-airport-link-may-be-delayed-further-3427147.html	0	0	0	0	0
4	http://thailand.yinyangandtaichichuan.org/informationlinks.html	0	0	0	0	0
5	http://www.prachatai.com/english/node/1208/	0	0	0	0	0

จากผลการทดสอบประสิทธิภาพของอัตราการเก็บเกี่ยวดังกราฟในภาพที่ 25 พบว่าวิธีการที่เราได้นำเสนอให้ผลลัพธ์ที่ดีกว่าวิธีค้นหาเว็บเพจภาษาไทยแบบดั้งเดิมในช่วงของการเก็บเว็บเพจเริ่มต้นกล่าวคือ 100 ถึง 500 เว็บเพจแรกหลังจากนั้นทั้งสองวิธีให้ผลลัพธ์ใกล้เคียงกัน ซึ่งสาเหตุหนึ่งที่ทำให้เจอกลุ่มเว็บเพจภาษาไทยได้รวดเร็วเนื่องจากวิธีการของเราพบยูอาร์แอลที่มีอินเทอร์เน็ตโดเมน ".th" ก่อนวิธีการแบบดั้งเดิม ดังนั้นจึงทำให้พบเว็บเพจภาษาไทยกลุ่มใหญ่ได้อย่างรวดเร็วขึ้น



ภาพที่ 25 กราฟอัตราการเก็บเกี่ยวเมื่อกำหนดยูอาร์แอลเริ่มต้นเป็นกลุ่มเว็บเพจภาษาอังกฤษและวัดผลบนอินเทอร์เน็ต

สรุปและข้อเสนอแนะ

สรุป

การเก็บรวบรวมเว็บเพจแบบเฉพาะเจาะจงภาษา มีวัตถุประสงค์เพื่อนำเว็บเพจเหล่านี้มาสร้างเป็นระบบคลังข้อมูลเว็บเพจขนาดใหญ่ (Large-scale national web archive) สำหรับนำมาสร้างเป็นระบบสืบค้นข้อมูลให้กับผู้ใช้ในท้องถิ่นนั้น วิธีการเก็บรวบรวมเว็บเพจแบบเฉพาะเจาะจงภาษาที่มีประสิทธิภาพที่ดีนั้น จะต้องพยายามเก็บรวบรวมเว็บเพจที่เป็นภาษาเป้าหมายให้ได้เป็นจำนวนมากที่สุดภายใต้ระยะเวลาที่ได้คาดการณ์ไว้ ทั้งนี้ก็เพื่อให้สามารถใช้ทรัพยากรคอมพิวเตอร์ เช่น พื้นที่จัดเก็บข้อมูล หรือระบบเครือข่าย เป็นต้น ได้อย่างคุ้มค่าและใช้เพื่อให้เกิดประโยชน์สูงสุด

วิทยานิพนธ์นี้จึงนำเสนอวิธีการค้นหาเว็บเพจภาษาไทยโดยใช้ฐานความรู้ซึ่งแบ่งวิธีการออกเป็น 2 วิธีการได้แก่ วิธีแรก คือ การเก็บเพจเพื่อนำมาสร้างฐานความรู้ คือการสกัดเอาฟีเจอร์ที่สำคัญที่ใช้ในการบ่งบอกถึงเว็บเพจภาษาไทยซึ่งประกอบด้วยฟีเจอร์ทั้ง 4 ชนิดได้แก่ ภาษาไทยจากเว็บเพจต้นทาง ภาษาไทยจากแองค์เคอร์เท็กซ์ของเว็บเพจต้นทาง อินเทอร์เน็ตโดเมนของเว็บเพจปลายทาง และเส้นทางของเว็บเพจภาษาไทยจากเว็บเพจต้นทางตามลำดับ และวิธีที่สอง คือ การเก็บเว็บเพจในรอบถัดไปโดยใช้ฐานความรู้ จะใช้วิธีการคำนวณความน่าจะเป็นแบบเบส์อย่างง่ายคำนวณระหว่างฟีเจอร์ของเว็บเพจที่ได้เก็บรวบรวมมากับฐานความรู้ที่สร้างไว้แล้วในข้อแรก ผลลัพธ์ที่ได้คือค่าคะแนนความสำคัญซึ่งใช้บ่งบอกถึงความน่าจะเป็นที่ยูอาร์แอลที่ออกจากเว็บเพจนั้น จะเป็นยูอาร์แอลของเว็บเพจภาษาไทย และนำค่าคะแนนดังกล่าวมากำหนดให้กับยูอาร์แอลพร้อมกับนำยูอาร์แอลใส่ลงในยูอาร์แอลคิว จากนั้นจัดเรียงยูอาร์แอลในคิวใหม่จากค่าคะแนนความสำคัญจากมากไปหาน้อยที่สุดตามลำดับ เพื่อให้การเว็บคราวเลอร์เลือกเก็บเว็บเพจที่มีคะแนนสูงสุดก่อนนั่นเอง

จากการทดลองที่ผ่านมาพบว่าวิธีการที่นำเสนอนี้ให้ประสิทธิภาพที่เกิดจากการวัดทั้งอัตราการเก็บเกี่ยวและความครอบคลุมได้ดีกว่าวิธีการอื่น ดังนั้นจึงมุ่งหวังว่าวิธีการนี้จะสามารถนำไปใช้งานจริงในเก็บรวบรวมเว็บเพจภาษาไทยเพื่อให้บรรลุถึงเป้าประสงค์ของระบบสืบค้นข้อมูลดังที่กล่าวมาในข้างต้นได้

ข้อเสนอแนะ

ประสิทธิภาพในการเก็บเว็บเพจภาษาไทยของวิธีการที่นำเสนอนี้จะขึ้นอยู่กับความแม่นยำของฐานความรู้เป็นหลัก กล่าวคือถ้าหากฐานความรู้สร้างมาได้ไม่ครอบคลุมกับฟีเจอร์ทั้ง 4 ชนิด เช่น บางฟีเจอร์อาจมีปริมาณเว็บเพจที่ให้ค่าของฟีเจอร์นั้นน้อยเกินไป การกำหนดปริมาณเว็บเพจในฐานความรู้น้อยเกินไป หรืออาจจะมีปริมาณเว็บเพจที่เป็นคลาสของภาษาไทยหรือภาษาอื่นเป็นจำนวนที่ไม่สมดุลย์กันหรือห่างกันมากจนเกินไป จะทำให้ประสิทธิภาพในการค้นหาเว็บเพจภาษาไทยลดต่ำลงเป็นอย่างมาก ดังนั้นจึงต้องกำหนดจำนวนเว็บเพจในฐานความรู้ให้มากเพียงพอที่จะครอบคลุมฟีเจอร์ทั้ง 4 ชนิดนี้ให้ได้ รวมไปถึงต้องพยายามกำหนดให้จำนวนของเว็บเพจในฐานความรู้ที่เป็นคลาสของภาษาไทยและภาษาอื่นให้มีปริมาณที่เท่าเทียมสมดุลย์กัน และสิ่งที่สำคัญอีกอย่างหนึ่งคือในระหว่างที่สร้างฐานความรู้ ควรหมั่นวัดความแม่นยำของฐานความรู้ประกอบร่วมกันไปด้วย ทั้งนี้ก็เพื่อให้ฐานความรู้มีความแม่นยำสูง ซึ่งจะส่งผลให้วิธีการนี้มีประสิทธิภาพยิ่งขึ้น

สำหรับงานในอนาคตจากการทดลองที่ผ่านมา เราจะพบว่าวิธีค้นหาเว็บเพจภาษาไทยแบบดั้งเดิมได้ใช้ฟีเจอร์หนึ่งที่มีความสำคัญต่อการค้นหาเว็บเพจภาษาไทยนั่นคือ เซิร์ฟเวอร์ที่เป็นของเว็บเพจภาษาไทย ถ้าเราสามารถหาวิธีการในการค้นหาเซิร์ฟเวอร์ของเว็บเพจภาษาไทยที่แม่นยำแล้ว จะทำให้ประสิทธิภาพในการค้นหาเว็บเพจภาษาไทยเพิ่มขึ้นมากกว่าเดิมได้อีกมาก เพราะว่าหากเราพบเซิร์ฟเวอร์ดังกล่าวที่ต้องการแล้ว เราจะเก็บเว็บเพจที่อยู่ในเซิร์ฟเวอร์นั้นทั้งหมด ซึ่งทำให้เราไม่จำเป็นต้องคำนวณค่าคะแนนความสำคัญของยูอาร์แอลที่อยู่ภายใต้เซิร์ฟเวอร์นั้นอีกแล้ว ซึ่งจะส่งผลให้ความเร็วในการเก็บเว็บเพจเพิ่มสูงขึ้นโดยที่ไม่ส่งผลต่ออัตราการเก็บเกี่ยว เมื่อนำฟีเจอร์ที่เป็นเซิร์ฟเวอร์ของเว็บเพจภาษาไทยมารวมกับฟีเจอร์ทั้ง 4 ชนิดที่นำเสนอในวิทยานิพนธ์นี้ เราอาจจะสร้างเป็นเว็บคราวเลอร์แบบเฉพาะเจาะจงภาษาไทยแบบไฮบริด (Hybrid) ในอนาคตได้

เอกสารและสิ่งอ้างอิง

Aggarwal, C.C., Al-Garawi, F. and P.S. Yu. 2001. Intelligent crawling on the World Wide Web with arbitrary predicates. *In* **Proceedings of the 10th International World Wide Web Conference**, Hong Kong, May 2001.

Apache Lucene: A full-featured text search engine library. Available Source: http://lucene.apache.org/java/docs/index.html, 2001.

Asynchronous-capable DNS client library and utilities (ADNS). Available Source: http://www.chiark.greenend.org.uk/~ian/adns/, 2009.

Baeza-Yates, R. and B. Ribeiro-Neto. 1999. **Modern Information Retrieval.** Addison-Wesley.

Boldi, P., Codenotti, B., Santini, M. and S. Vigna. 2004. UbiCrawler: a scalable fully distributed web crawler. **Software —Practice & Experience** (vol.34 n.8): 711-726.

Boswell, D. 2003. **Distributed High-performance Web Crawlers: A Survey of the State of the Art.** Available Source: http://www.cs.ucsd.edu/~dboswell/PastWork/WebCrawlingSurvey.pdf.

Bra, P. D., Houben, G., Kornatzky, Y. and R. Post. 1994a. Information Retrieval in the World Wide Web: Making Client-based Searching Feasible. *In* **proceedings of the 1st International World Wide Web Conference.**

Bra, P. D., Houben, G., Kornatzky, Y. and R. Post. 1994b. Searching for Arbitrary Information in the WWW: The Fish Search for Mosaic. *In* **proceedings of the 2nd International World Wide Web Conference.**

Brin, S. and L. Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *In* **the proceedings of the 7th International World Wide Web Conference (WWW7),** Brisbane, April 1998. Available Source: http://dbpubs.stanford.edu:8090/pub/1998-8: 107-117.

Burner, M. 1997. Crawling towards eternity: Building an archive of the World Wide Web. *In* **Web Techniques Magazine,** Vol. 2, Nr. 5, May 1997.

Cavnar, W and J. Trenkle. 1994. N-gram-based text categorization. *In* **Symposium on Document Analysis and Information Retrieval,** University of Nevada Las Vegas**:** 161-176.

Chakrabarti, S., van den Berg, M. and B. Dom. 1999. Focused crawling: a new approach to topic-specific Web resource discovery. *In* **Proceeding of the 8th International conference on World Wide Web,** May 1999, Toronto, Canada: 1623-1640.

Cho, J., Garcia-Molina H. and L. Page. 1998. Efficient crawling through URL ordering. *In* **Proceedings of the 7th international conference on World Wide Web7,** April 1998, Brisbane, Australia: 161-172.

CPDetector - Character set detector. Available Source: http://cpdetector.sourceforge.net/usage.shtml, August 2009.

Davison, B. 2000. Topical Locality in the Web. *In* **Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval,** Athens, Greece: ACM, July 2000: 272-279.

Diligenti, M., Coetzee, F., Lawrence, S., Giles, C. L. and M. Gori. 2000. Focused crawling using context graphs. *In* **proceedings 26th International Conference on Very Large Databases (VLDB 2000),** Cairo, Egypt: 527-534.

DMOZ: Open Directory Project. Available Source: http://www.dmoz.org, August, 2009.

Google. Available Source: http://www.google.com/, August, 2009.

Gray, M. 1993. **Internet Growth and Statistics: Credits and Background.** Available Source: http://www.mit.edu/people/mkgray/net/background .html.

Hersovici, M., Jacovi, M., Maarek, Y.S., Pelleg, D., Shtalhaim, M., and S. Ur. 1998. The Shark-search Algorithm - An Application: Tailored Web Site Mapping. *In* **proceedings of the 7th World Wide Web Conference**, 1998, Brisbane. Available Source: http://www7.scu.edu.au/programme/fullpapers/1849/com1849.htm

Heydon, A. and M. Najork. 1999. Mercator: A scalable, extensible Web crawler. *In* **proceedings of the 8th International World Wide Web Conference,** (v.2 n.4): 219-229.

Kleinberg, J. 1998. Authoritative sources in a hyperlinked environment. *In* **Proceedings of the 9th annual ACM-SIAM symposium on Discrete algorithms,** January 25-27, 1998, San Francisco, California, United States: 668-677.

Koster, M. 1993. **Guidelines for Robot Writes.** Available Source: http://www.robotstxt.org/wc/guidelines.html.

Kruengkrai, C., Srichaivattana, P., Sornlertlamvanich, V. and H. Isahara. 2005. Language identification based on string kernels. *In* **proceedings of the 5th Symposium on Communication and Information Technologies**, 12-14 Oct. 2005 (2): 926- 929.

Kulturarw3 - National Library of Sweden. Available Source: http://www.kb.se/english, August, 2009.

LexTo – Thai Analyzer. Available Source: http://sansarn.com/look/download.html, August, 2009.

LIBS - Language Identifier on Byte String. Available Source: http://www.tcllab.org/canasai/software/libs/, August , 2009.

MD5: The MD5 Message-Digest Algorithm. Available Source: http://tools.ietf.org/html/rfc1321, 2005.

Menczer, F., Pant, G., Srinivasan, P. and M. E. Ruiz. 2001. Evaluating topic-driven web crawlers. *In* **Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval,** September 2001, New Orleans, Louisiana, United States: 241-249.

Mitchell, T. 1997. **Machine Learning.** McGraw-Hill: 177-178.

Oracle Berkeley Database. Available Source: http://www.oracle.com/technology/products/berkeley-db/index.html, August, 2009.

Page, L., Brin, S., Motwani, R. and T. Winogard. 1998. The pagerank citation ranking: Bringing order to the web. *In* **Technical report Stanford Digital Library Technologies Project.**

Pant, G., Srinivasan, P. and F. Menczer. 2002. Exploration versus Exploitation in Topic Driven Crawlers. *In* **Proceedings WWW-02 Workshop on Web Dynamics.**

Rennie, J. and A. McCallum. 1999. Efficient Web Spidering with Reinforcement Learning. *In* **proceedings of the 16th International Conference on Machine Learning.**

Rungsawang, A. and N. Angkawattanawit. 2002. Learnable Crawling: An efficient Approach to Topic-specific Web Resource Discovery. *In* **Proceedings 2002 International symposium on communication and Information Technology,** October, 2002**, Journal of Network and Computer Applications,** April 2005 (Volume 28, Issue 2): 97-114.

Sanguanpong, S. and K. Koht-arsa. 2001. In-memory URL Compression. *In* **Proceedings the 5th National Computer Science and Engineering Conference (NCSEC),** 7-9 November 2001.

Sanguanpong, S. and K. Koht-Arsa. 2002. High Performance Large Scale Web Spider Architecture. *In* **Proceedings 2002 International Symposium on Communication and Information Technology,** October 23 2002.

Salton, G. 1997. **The SMART Retrieval System – Experiments in Automatic Document Processing.** Prentice-Hall, Englewood Cliffs, NJ.

Salton, G. and C. Buckley. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *In* **proceeding of 1998 Information Processing and Management** 24(5): 513-23.

Salton, G., A. Wong and C. Yang. 1975. A vector Space Model for Automatic Indexing. *In* **Proceeding of 1975 ACM Conference on Communication** 18 (11): 613-620.

Sansarn – Thai Search Engine. Available Source: http://www.sansarn.com/, 2001.

Sedgewick, R. and P. Flajolet. 1995. **An Introduction to the Analysis of Algorithms.** Addison-Wesley.

Shkapenyuk, V. and T. Suel. 2002. Design and implementation of a high performance distributed web crawler. *In* **International Conference on Data Engineering (ICDE).**

Somboonviwat, K., Kitsuregawa, M. and T. Tamura. 2005. Finding Thai web page in foreign web spaces. *In* **proceedings of 21st International Conference on Data Engineering (ICDE'05),** 2005: 1254.

TextCat: The Java Text Categorizing Library (JTCL). Available Source: http://sourceforge.net/projects/textcat/, August, 2009.

Tamura, T, Somboonviwat, K. and M. Kitsuregawa. 2006. A Method for Language-Specific Web Crawling and Its Evaluation. **Journal, System and Computers in Japan**, 2 Feb 2006: 199-209.

Tongchim, S., Srichaivattana, P., Kruengkrai, C., Sornlertlamvanich, V. and H. Isahara. 2006. Collaborative Web Crawler over High-speed Research Network, *In* **proceedings of the First International Conference on Knowledge**. **Information and Creativity Support Systems**: 302-308, Ayutthaya, Thailand, August 1-4, 2006.

Weka 3: Data Mining Software in Java. Available Source: http://www.cs.waikato.ac.nz/ml/weka, 2009.

World Wide Web Size - The size of WWW. Available Source: http://www.worldwidewebsize.com/, August, 2009.

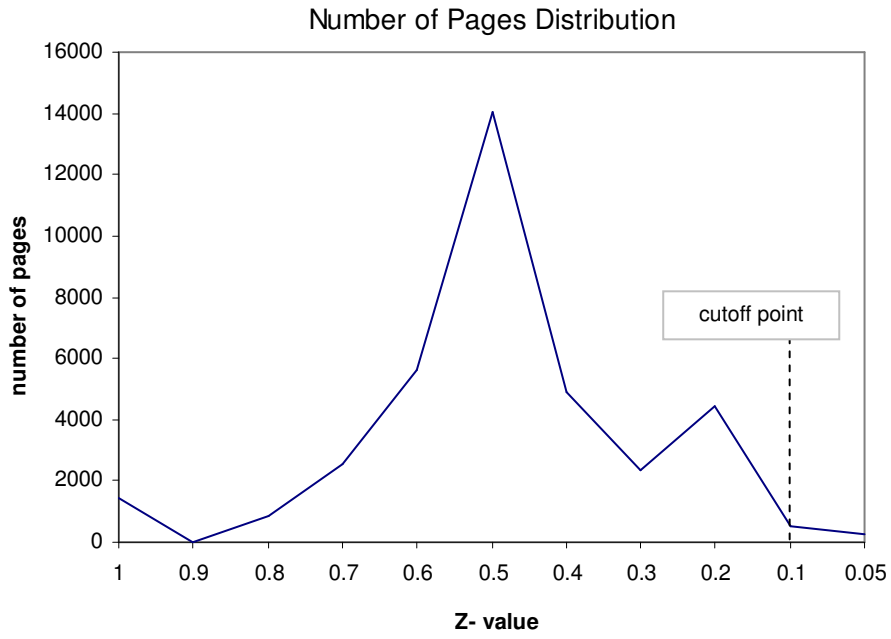
W3C. Available Source: http://www.w3.org/International/, August, 2009.

ZLIB: A Massively Spiffy Yet Delicately Unobtrusive Compression Library. Available Source: http://www.zlib.net, July 18, 2005.

ภาคผนวก

การปรับค่าที่ใช้ตัดสินว่าเป็นเว็บเพจภาษาไทย

ในหัวข้อนี้จะกล่าวถึงผลของการปรับค่าที่ใช้ตัดสินว่าเป็นเว็บเพจภาษาไทย (*Z*) ซึ่งใช้กับวิธีการเล็กซ์โต โดยทดสอบกับเว็บเพจในชุดข้อมูลเว็บกราฟ (Web graph dataset) ในหัวข้อที่ 2.2 ดังที่กล่าวมาแล้ว จากกราฟในภาพผนวกที่ก1 นำเสนอด้วยแกนนอน (x) มีความหมายคือค่า *Z* ที่เราปรับในแต่ละช่วงเริ่มตั้งแต่ 1 จนถึง 0.05 ตามลำดับ ส่วนแกนตั้ง (y) คือจำนวนของเว็บเพจที่มีความสัมพันธ์กับแกนนอนนั่นเอง



ภาพผนวกที่ 1 ผลการปรับค่าที่ใช้ตัดสินว่าเป็นเว็บเพจภาษาไทย

จากภาพผนวกที่1 จะเห็นว่าจำนวนเว็บเพจที่มากที่สุดจะอยู่ในช่วงที่เราปรับค่า Z เป็น 0.5 และลดน้อยลงไปจนถึง 0.1 ตามลำดับ หลังจากนั้นตั้งแต่ 0.1 จนถึง 0.05 จะพบจำนวนของเว็บเพจเพียงเล็กน้อยเท่านั้น และจำนวนของเว็บเพจใน 0.05 ค่อนข้างคงที่ ซึ่งไม่แตกต่างกันกับ 0.1 มากนัก ด้วยเหตุนี้เราจะเลือกใช้ค่าสำหรับตัดสินว่าเป็นภาษาไทย สำหรับเล็กซ์โตที่ Z = 0.1 ซึ่งใช้ในการทดลองทั้งหมดของวิทยานิพนธ์เล่มนี้

การวัดความถูกต้องของฐานความรู้โดยใช้เวก้า

ในหัวข้อนี้จะกล่าวถึงผลของการวัดความถูกต้องของฐานความรู้โดยใช้เวก้า โดยฐานความรู้ถูกสร้างโดยใช้อัลกอริทึมที่ 1 ของหัวข้อการกำหนดค่าในการทดลองข้อ 5.1 ดังที่กล่าวมาแล้ว ซึ่งประกอบด้วยจำนวนเว็บเพจในฐานความรู้ทั้งสิ้น 10000 เว็บเพจ แบ่งเป็นเว็บเพจภาษาไทยจำนวน 6939 เว็บเพจและเว็บเพจภาษาอื่น 3061 เว็บเพจตามลำดับ ผลลัพธ์จากการวัดจะได้ค่าความแม่นยำ (precision) ซึ่งเปรียบได้กับอัตราการเก็บเกี่ยว และค่าความระลึก (recall) ซึ่งเปรียบได้กับความครอบคลุม และกำหนดหน่วยวัดเพิ่มขึ้นอีก คือ ค่าเอฟ (F-measure) ซึ่งเป็นค่าความสัมพันธ์ระหว่างค่าความแม่นยำกับค่าความระลึก โดยเราจะใช้ค่านี้เป็นตัวแทนในการนำเสนอประสิทธิภาพจากค่าทั้งสองนั่นเอง

ตารางผนวกที่ 1 ผลการวัดความถูกต้องของฐานความรู้โดยใช้เวก้า

ผลลัพธ์ที่ได้	จำนวนเว็บเพจในฐานความรู้		Precision	Recall	F-Measure
	ภาษาไทย	ภาษาอื่น	(P)	(R)	$F=\frac{2\times P\times R}{(P+R)}$
ภาษาไทย	6623	974	0.87	0.95	0.911
ภาษาอื่น	316	2087	0.86	0.68	0.764
	6939	3061			

ประวัติการศึกษา และการทำงาน

ชื่อ –นามสกุล	นายเอกสิทธิ์ ศรีสุขะ
วัน เดือน ปี ที่เกิด	12 เมษายน 2522
สถานที่เกิด	จ. สระแก้ว
ประวัติการศึกษา	สาขาเทคโนโลยีสารสนเทศเพื่ออุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ (ปราจีนบุรี)
ตำแหน่งหน้าที่การงานปัจจุบัน	ผู้ชำนาญด้านซอฟท์แวร์
สถานที่ทำงานปัจจุบัน	บริษัท พันทวณิช จำกัด
ผลงานดีเด่นและรางวัลทางวิชาการ	-
ทุนการศึกษาที่ได้รับ	งานวิจัยนี้ได้รับทุนจาก สถาบันบัณฑิตวิทยาศาสตร์และเทคโนโลยีไทย (TGIST) สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ สัญญารับทุนเลขที่ TG-44-11-49-074M