สถาปัตยกรรมเว็บเสิร์ชเอนจิ้น ที่สามารถปรับตัวตามผู้ค้นหา

วิริศ ลีลาภัทร*, พรฤดี เนติโสภากุล**

บทคัดย่อ

ปัญหาของการค้นหาข้อมูลบนเครือข่ายอินเทอร์เน็ตโดยใช้เว็บเสิร์ชเอนจิ้นคือผลลัพธ์การค้นหาที่ได้ยังขาดความแม่นยำและความครอบคลุมของผลลัพธ์ซึ่งแนวทางในการปรับปรุงหรือแก้ไขปัญหาข้างต้นที่กล่าวมามีอยู่หลายทางด้วยกัน เช่น การนำผลข้อมูลป้อนกลับ (Feedback) จากผู้ค้นหางานในการค้นหาครั้งต่อไปหรือการใช้ Semantic Network ในการค้นหา สำหรับในบทความนี้จะกล่าวถึงแนวคิด การออกแบบสถาปัตยกรรมของเว็บเสิร์ชเอนจิ้นที่สามารถปรับตัวตามผู้ค้นหารวมถึงส่วนประกอบต่างๆและการทำงานของแต่ละส่วน

คำสำคัญ:

Search Engine, User Feedback, User Adaptive

1. บทนำ

ในปัจจุบันเครื่องมือที่ถูกใช้ในการค้นหาข้อมูลที่มีอยู่อย่างมากมายมหาศาลและนับวันจะเพิ่มปริมาณมากขึ้นเรื่อย ๆ บนเครือข่ายอินเทอร์เน็ตที่ได้รับความนิยมมากที่สุดในขณะนี้คือเสิร์ชเอนจิ้น ถึงแม้ว่าเสิร์ชเอนจิ้นจะได้รับความนิยมมากที่สุดในเวลานี้ แต่เสิร์ชเอนจิ้นก็ยังมีประเด็นที่เป็นข้อเสียหลัก ๆ ที่พบก็คือ ประเด็นแรกเรื่องของผลลัพธ์ที่ได้จากการเสิร์ชแต่ละครั้งมีปริมาณที่มากเกินไปอีกประเด็นหนึ่งที่ตามมาก็คือความครอบคลุมและความแม่นยำของผลลัพธ์ที่ได้ยังไม่ดีพอ ทั้งนี้สาเหตุมาจากคำที่ผู้ค้นหาใช้ในการค้นหาอาจจะอยู่ในหลายเรื่อง หลายเนื้อหา ทำให้บางครั้งก็ไม่ตรงกับความต้องการของผู้ค้นหาโดยแนวทางในการแก้ปัญหาดังกล่าวได้แก่ การสร้าง Semantic Network ของเว็บและการเก็บข้อมูลป้อนกลับของผู้ค้นหา (User Feedback) [14] โดยแนวทางการเก็บข้อมูลป้อนกลับของผู้ค้นหา (User Feedback) เป็นแนวทางหนึ่งที่น่าสนใจที่จะนำมาใช้ในการปรับปรุงการทำงานของเสิร์ชเอนจิ้น http://www.anirach.com ให้มีประสิทธิภาพดียิ่งขึ้นโดยข้อมูลของผู้ค้นหาจะมีประโยชน์ในการใช้งานของผู้ค้นหาในครั้งถัดไปคือเสิร์ชเอนจิ้นนั้นสามารถรู้ได้ว่าผู้ค้นหาคนนั้นต้องการค้นหาข้อมูลอะไร

บทความนี้นำเสนอเกี่ยวกับแนวคิดที่ใช้ในการปรับปรุงเว็บเสิร์ชเอนจิ้นโดยการนำเอาข้อมูลของผู้ค้นหามาประยุกต์ใช้งานร่วมกันเว็บเสิร์ชเอนจิ้น นอกจากนี้ยังจะกล่าวถึงการออกแบบสถาปัตยกรรม และอธิบายส่วนประกอบต่าง ๆ ของเว็บเสิร์ชเอนจิ้นที่สามารถปรับตัวตามการใช้งานของผู้ค้นหาได้

สำหรับในหัวข้อต่อไปจะกล่าวถึงประเภทของเสิร์ชเอนจิ้นว่ามีกี่ประเภท เสิร์ชเอนจิ้นแต่ละประเภทมีลักษณะการทำงานอย่างไร งานวิจัยต่าง ๆ ที่เกี่ยวข้องกับการปรับปรุงเสิร์ชเอนจิ้น แนวคิดและการออกแบบของสถาปัตยกรรมของเว็บเสิร์ชเอนจิ้นที่สามารถปรับตัวตามผู้ค้นหา รวมถึงอธิบายการทำงานโดยรวมของเสิร์ชเอนจิ้นที่สร้างขึ้นมาและอธิบายรายละเอียดการทำงานของส่วนประกอบต่าง ๆ ตามลำดับ

2. ประเภทของเสิร์ชเอนจิ้นและงานวิจัยที่เกี่ยวข้องกับการปรับปรุงเสิร์ชเอนจิ้น

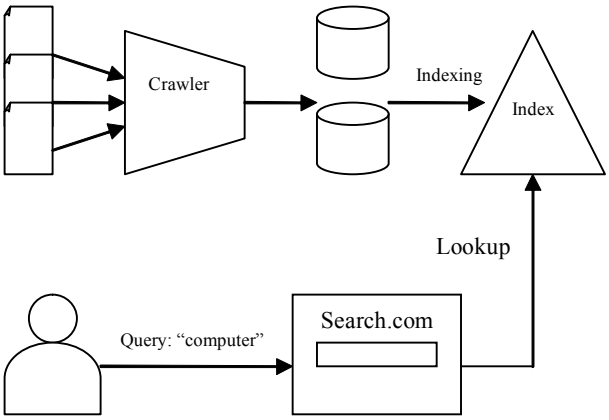
บริการค้นหาเว็บเพจที่มีอยู่ในปัจจุบัน เช่น Google, AltaVista, Lycos, Yahoo, Hotbot สามารถเรียกโดยรวมว่าเป็นเสิร์ชเอนจิ้นซึ่งแบ่งได้เป็น 3 ประเภท [6], [15], [16] คือ

2.1 Indexing Search Engine (keyword search)

เสิร์จเอ็นจิน เป็นระบบซอฟต์แวร์ที่มีโปรแกรม โรบอต (Robot) ซึ่งบางครั้งเรียกว่า สไปเดอร์ (Spider) หรือ ครอว์เลอร์ (Crawler) ทำหน้าที่เดินทางไปยังเว็บไซต์ต่าง ๆ ในอินเทอร์เน็ตและอ่านเว็บเพจจากไซต์เหล่านั้นเพื่อนำมาสร้างดัชนีรายการของเว็บเพจโดยอัตโนมัติ การทำดัชนีรายการเว็บเพจด้วยเสิร์จเอ็นจินจึงสามารถสร้างดัชนีของเว็บเพจได้เป็นจำนวนที่มากกว่าของไดเรคทอรี เนื่องจากดัชนีของไดเรคทอรีจะถูกจัดการโดยมนุษย์แทนที่จะใช้คอมพิวเตอร์ ถ้าหากเว็บเพจที่ถูกทำดัชนีแล้วเกิดมีการเปลี่ยนแปลง เสิร์จเอ็นจินจะทราบถึงการเปลี่ยนแปลงและจัดนำเว็บเพจที่มีการเปลี่ยนแปลงนั้นมาสร้างดัชนีใหม่ ซึ่งการสร้างดัชนีใหม่นี้อาจส่งผลต่อการจัดลำดับของเว็บเพจนั้น ตัวอย่างของเสิร์จเอ็นจินที่มีอยู่ในปัจจุบัน เช่น HotBot, AltaVista เป็นต้น

* *คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง*

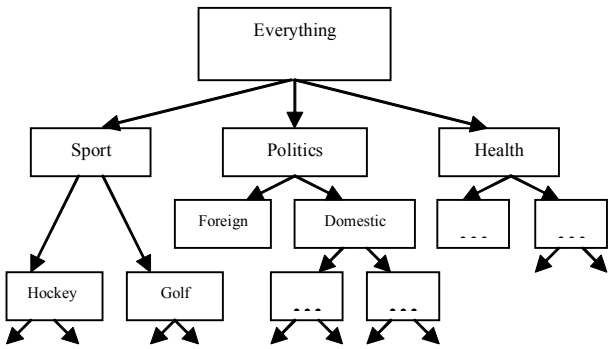
** *คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง*



ภาพที่ 1 สถาปัตยกรรมของ Indexing Search Engine (keyword search) [8]

2.2 Subject Directories

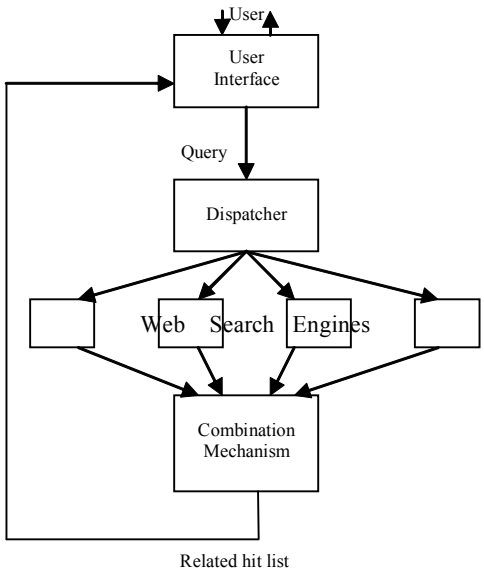
จะมีลักษณะเป็นรายการของเว็บไซต์ หรือ เว็บเพจ ซึ่งได้มีการจัดรวบรวมไว้โดยการแบ่งเป็นหมวดหมู่ตามลักษณะและหัวข้อเนื้อหา เพื่อให้ผู้ค้นหาสามารถเลือกค้นได้ ในแต่ละหมวดหมู่หัวข้อเนื้อหา (Subject Categories) จะถูกแบ่งเป็นหมวดหมู่ย่อย (Sub-Categories) ที่จัดเรียงลดหลั่นกันหลายระดับจนกระทั่งถึงรายการชื่อเว็บไซต์ที่นำเสนอเนื้อหาที่สอดคล้องกับหมวดหมู่ย่อยนั้นพร้อมกับมีตัวเชื่อมโยง (Link) เพื่อชี้ไปยังเว็บไซต์นั้น กระบวนการสร้างและจัดทำผ่านการดำเนินการโดยบุคคลหรือกลุ่มบุคคลที่ทำหน้าที่ในการจัดการหมวดหมู่หัวข้อเนื้อหา เลือกแนะจำแนกเว็บไซต์ลงในหมวดหมู่



ภาพที่ 2 แสดง Topic hierarchy taxonomies ของ Subject Directories [8]

2.3 Multi-Engine Search Tool หรือ Meta-Search Engine

Meta Search Engine คือ Search Engine ที่สามารถสืบค้นข้อมูลจาก Search Engine และหรือ Web Directories ได้มากกว่า 1 ตัวในเวลาเดียวกัน และแสดงผลการสืบค้นที่ได้รับจาก Search Engine เหล่านั้นในเวลาเดียวกันโดยเสนอผลการสืบค้นในรูปแบบที่สะดวก ซึ่งในบางครั้งจะมีการปรับแต่งผลการสืบค้นที่ได้รับทั้งหมดให้อยู่ในรูปแบบเดียวกัน และบูรณาการผลการสืบค้นเหล่านี้เข้าเป็นชุดเดียวกัน



ภาพที่ 3 แสดงสถาปัตยกรรมของ Meta-Search Engine [1]

ถึงแม้ว่าเสิร์ชเอนจิ้นแต่ละประเภทจะใช้ในการค้นหาข้อมูลบนเครือข่ายได้เป็นอย่างดีแต่ปัญหาหรือข้อจำกัดที่เกิดขึ้นกับเสิร์ชเอนจิ้นแต่ละประเภทก็ยังมีอยู่ อันได้แก่ ปัญหาเรื่องการชี้ไปยังหน้าเอกสารที่ไม่มีการปรับปรุงข้อมูลหรือไม่มีข้อมูลตามที่ระบุไว้ (Dead Link) ปริมาณสารสนเทศที่รวบรวมไว้มีไม่มากนักรวมถึงระยะเวลาในการจัดเก็บข้อมูลที่ใช้เวลานานอันเป็นผลมาจากต้องใช้คนในการตรวจสอบและจัดเก็บซึ่งปัญหาดังกล่าวเกิดขึ้นกับเสิร์ชเอนจิ้นแบบ Subject Directories ปัญหาที่พบในเสิร์ชเอนจิ้นแบบ Indexing Search Engine คือเรื่องของผลลัพธ์ที่ได้มากเกินไป ขาดการประเมินและกลั่นกรองสาระในเว็บเพจที่ไปเก็บรวบรวมมา ส่วนเสิร์ชเอนจิ้นประเภท Meta-Search Engine ปัญหาที่พบก็คือผลลัพธ์ที่ได้จากการค้นหามักจะซ้ำกันอันเป็นผลมาจากตัวเสิร์ชเอนจิ้นต้องไปดึงข้อมูลมาจากหลาย ๆ ที่ [17]

จากปัญหาของเสิร์ชเอนจิ้นที่กล่าวมาแล้วข้างต้น ได้มีงานวิจัยหลาย ๆ เรื่องที่ทำการวิจัยเพื่อปรับปรุงและแก้ไขปัญหาดังกล่าว สำหรับในบทความนี้จะขอยกตัวอย่างงานวิจัย

พอสังเขป ได้แก่ A Fast Regular Expression Indexing Engine [10]

บทความนี้นำเสนอเป็นการแก้ไขปัญหาเกี่ยวกับเรื่องการลดระยะเวลาที่ใช้ในการจับคู่ของ Regular Expression กับฐานข้อมูลแบบตัวหนังสือที่มีขนาดใหญ่ให้น้อยลง โดยมีแนวคิดเรื่องการสร้าง Pre-Built Index ที่เรียกว่า Multigram-Index เพื่อแยกกลุ่มข้อมูลแบบตัวอักษรซึ่งประกอบด้วยการจับคู่ของกลุ่มคำกับ Regular Expression โดย Pre-Build Index ที่สร้างขึ้นจะเพิ่มความเร็วให้กับการจับคู่ Regular Expression กับข้อมูลบนฐานข้อมูลแบบตัวหนังสือขนาดใหญ่ โดยให้ผลลัพธ์ที่เร็วขึ้นถึง 2 เท่า นอกจากนี้แล้ว Multigram-Index ยังช่วยลดขนาดของดรรชนีโดยที่ไม่ทำให้ประสิทธิภาพลดลง Information Extracting and Gathering for Search Engine: The Taylor Approach [11]

บทความนี้เป็นการนำเสนอเทคนิคที่เรียกว่าExtraction/Gathering เพื่อนำมาใช้กับเสิร์ชเอนจิ้นประเภท Domain-Specific เพื่อที่ต้องการจะส่งผลลัพธ์การค้นหาด้วยความถูกต้องในระดับกรานูลาริตี้กลับยังผู้ค้นหา โดย Extraction จะทำการเลือกเนื้อหาเอกสารทั้งหมดที่เกี่ยวข้อง ส่วน Gathering จะทำการสร้างหน้าคำตอบที่บรรจุเนื้อหาเหล่านั้น การทำงานของเทคนิคดังกล่าวคือการกระจายกลุ่มเอกสารที่มีการเก็บรวบรวมไว้ สร้างตัวบอกตำแหน่งที่ใช้ในการค้นหาของเอกสารที่ป้อนเข้ามาและไวยากรณ์ของเอกสารเพื่อใช้ในการกระจายกลุ่มเอกสาร จากนั้นก็จะถูกอินเด็กซ์และจะคืนลำดับรายการของเนื้อหาเอกสารโดยที่จะใช้ชนิดของคิวรี่และโครงสร้างเอกสารเพื่อใช้หาความถูกต้องในระดับกรานูลาริตี้ของคำตอบ N-Sums: A Framework for Web-based Search Engine [12]

บทความนี้มีวัตถุประสงค์เพื่อออกแบบ สร้าง และบำรุงรักษาเสิร์ชเอนจิ้นประเภท Specialized Domain โดยมีแนวคิดของการจดจำกลไกลการค้นหาที่แตกต่างและหลากหลายโดยนำมาผสมกันเพื่อให้ได้ผลลัพธ์ที่ดีในการค้นหาโดยมีการออกแบบพัฒนาคิวรี่ให้เหมาะสมและวิธีการในจัดลำดับคะแนนของผลลัพธ์ที่ได้เพื่อลดการค้นหาผลลัพธ์อีกครั้งในรอบที่สองและนำผลไปแสดงที่หน้าผลลัพธ์ แนวคิดนี้เป็นการนำเอากลยุทธ์ในการค้นหา อันได้แก่ General Purpose Search Engine, Specialized Domain Search Engine, Direct Search of Static/Dynamic Web Page และ Local Database (s) of Search Information มารวมเข้าด้วยกัน ซึ่งทำให้ประสิทธิภาพดีกว่าเสิร์ชเอนจิ้น แบบ General Purpose WebReader: a Mechanism for Automating the Search and Collecting Information from the World Wide Web [13]

บทความนี้เป็นการนำเสนอการสร้างมิดเดิลแวร์ขึ้นมาเพื่อแก้ปัญหาเรื่องขอบเขตการค้นหาข้อมูลที่กว้างขวางเกินความจำเป็นและความแม่นยำของผลลัพธ์ที่ต่ำลงในเสิร์ชเอนจิ้นที่เป็น Keyword-Based ซึ่งส่งผลต่อประสิทธิภาพในการหาข้อมูลจากเวป มิดเดิลแวร์ที่สร้างขึ้นมาจะทำงานอยู่ระหว่าง browser กับผู้ค้นหาเพื่อทำการค้นหาและเก็บรวบรวมข้อมูลจากเวปโดยอัตโนมัติ โดยอาศัยหลักการของ Meta-Data ที่กำหนดใน XML และการจัดการของ XSL ข้อได้เปรียบที่สำคัญคือการนำเอาแหล่งของคีย์เวิร์ดมาเป็นตัวแปรเงื่อนไขสำหรับผลลัพธ์คิวรี่ของ WebSQL มิดเดิลแวร์นี้ทำงานได้ทั้งกับเวปเพจและเอกสารที่เชื่อมโยงไปยังเวปเพจด้วย

3. การปรับปรุงเว็บเสิร์ชเอ็นจิ้น

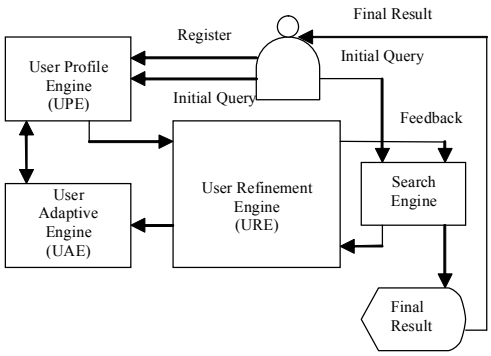
3.1 แนวคิด

เนื่องจากว่าข้อมูลที่มีอยู่บนเครือข่ายอินเทอร์เน็ตมีจำนวนมาก ในขณะที่ผู้ค้นหามีเวลาจำกัดในการค้นหาข้อมูลที่ต้องการประกอบกับผู้ค้นหาแต่ละคนไม่ได้มีความสนใจในทุกหัวข้อ สนใจเฉพาะในบางหัวข้อเท่านั้น จึงเป็นแนวคิดในการออกแบบเสิร์ชเอนจิ้นที่ใช้ในการค้นหาข้อมูล โดยช่วยให้ผู้ค้นหาประหยัดเวลาในการค้นหาเฉพาะหัวข้อที่ผู้ค้นหาสนใจ และเสิร์ชเอนจิ้นยังสามารถปรับตัวตามการใช้งานของผู้ค้นหาได้โดยมีการเพิ่มแก้ไข และลบข้อมูลการค้นหาของผู้ค้นหาได้

3.2 สถาปัตยกรรม

สถาปัตยกรรมที่ถูกออกแบบขึ้นมามีส่วนประกอบหลักๆ ด้วยกัน 3 ส่วน คือ User Refinement Engine (URE), User Profile Engine (UPE) และ User Adaptive Engine (UAE) ดังที่แสดงในรูปที่ 4 การทำงานโดยรวมของเสิร์ชเอนจิ้นจะเริ่มจากผู้ค้นหาจะต้องทำการลงทะเบียนกับเสิร์ชเอนจิ้นก่อน จากนั้นเมื่อผู้ค้นหาต้องการค้นหา ผู้ค้นหาก็จะทำการป้อน Initial Query เข้ามาโดย Initial Query ของผู้ค้นหาจะถูกส่งไปจัดเก็บไว้ที่ UPE และส่งเข้าเสิร์ชเอนจิ้นเพื่อทำการค้นหา ผลลัพธ์ของการค้นหาในครั้งแรก (Initial Result) ที่ได้จะถูกส่งเข้า URE ผลลัพธ์ที่ได้จาก URE จะอยู่ในรูปของ

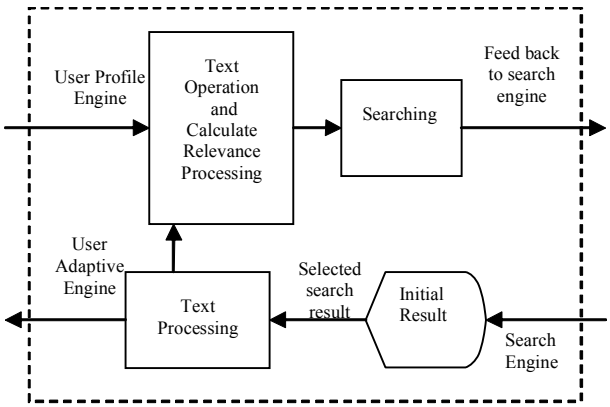
คำศัพท์ซึ่งจะถูกส่งไปยัง UAE เพื่อทำการตรวจสอบคำศัพท์ จากนั้นก็ส่งไปจัดเก็บไว้ที่ UPE นอกจากนี้แล้วผลลัพธ์ที่ได้จาก URE ก็ถูกส่งกลับไปที่เสิร์ชเอนจิ้นเพื่อทำการค้นหาใหม่อีกครั้ง ซึ่งก็จะได้ผลลัพธ์ขั้นสุดท้ายออกมา



ภาพที่ 4 สถาปัตยกรรมของเสิร์ชเอนจิ้นที่สามารถปรับตัวตามผู้ค้นหา

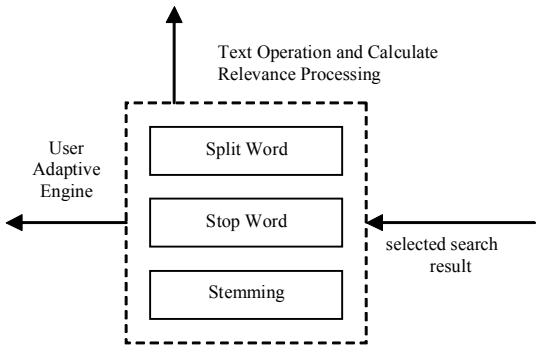
รายละเอียดและการทำงานของแต่ส่วนประกอบทั้ง 3 มีดังต่อไปนี้

3.2.1 User Refinement Engine (URE)



ภาพที่ 5 แสดงรายละเอียดของ URE

URE ประกอบไปด้วย 4 ส่วนด้วยกัน ดังที่แสดงในภาพที่ 5 ส่วนแรกคือ Initial Result จะเป็นส่วนที่นำผลลัพธ์จากการเสิร์ชครั้งแรกมาแสดงผลโดยผู้ค้นหาสามารถจะทำการเลือกเฉพาะลิงค์ที่ตรงตามความต้องการได้ เมื่อผู้ค้นหาทำการเลือกเสร็จแล้ว ลิงค์ที่ถูกเลือกจะถูกส่งมายังส่วนที่สอง คือ Text Processing ดังที่แสดงในภาพที่ 6 ซึ่งประกอบด้วยการทำงาน 3 ขั้นตอน ได้แก่



ภาพที่ 6 แสดงรายละเอียดของ Text Processing

- Split Word จะทำหน้าที่ในการแยกคำจาก Tag Title และ Tag Description ของลิงค์ที่ผู้ค้นหาเลือก

- Stop Word [3] จะทำหน้าที่ในการกรองคำที่มีความสำคัญน้อยที่สุดออก โดยปกติคำเหล่านี้จะพบบ่อยในเอกสาร อย่างเช่น a, an, the, that, this, of เป็นต้น

- Stemming [4], [5] จะทำหน้าที่ในการลดรูปของคำศัพท์ (โดยการตัด Prefixes และ Suffixes ของคำให้อยู่ในรูปของรากศัพท์) อย่างเช่น

CONNECTED -> CONNECT

CONNECTING -> CONNECT

CONNECTION -> CONNECT

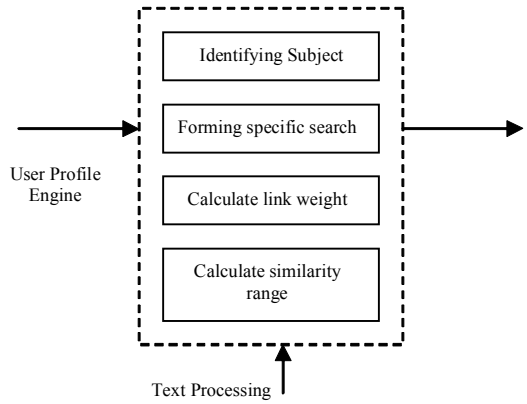
CONNECTIONS -> CONNECT

เป็นต้น

เมื่อผ่านขั้นตอนทั้ง 3 แล้ว ผลลัพธ์จะถูกส่งไป 2 ทาง คือส่งไปยัง UAE และ Text Operation and Calculate Relevance Processing ต่อไป สำหรับในส่วนที่สาม คือ Text Operation and Calculate Relevance Processing ดังที่แสดงในภาพที่ 7 จะประกอบด้วยการทำงาน 4 ขั้นตอน ได้แก่

- Identifying Subject จะทำหน้าที่ในการระบุว่าคิวรี่ที่ผู้ค้นหาป้อนเข้ามาเกี่ยวกับเรื่องอะไรโดยมีการดึงข้อมูลมาจาก UPR

- Forming specific search จะทำหน้าที่ในการค้นหาแบบเจาะจง ตามที่กำหนด โดยใช้คำศัพท์ที่ผู้ค้นหาป้อนเข้ามาและคำศัพท์ที่เก็บอยู่ใน UPR ที่ปรากฏร่วมกันเพื่อหาหัวเรื่อง (Subject) ที่เฉพาะเจาะจงมากขึ้น และหน้าที่อีกอย่างหนึ่งคือการสร้างรูปแบบ Regular Expression เพื่อใช้ในการค้นหาหัวเรื่องแบบเฉพาะด้วย



ภาพที่ 7 แสดงรายละเอียดของText Operation and Calculate relevance processing

- Calculate link weight จะทำหน้าที่ในการคำนวณหา Link Weight โดยดูว่ามี Link ไหนบ้างที่สัมพันธ์กัน ซึ่งจะทำให้ได้ ชุดของลิงค์ใหม่ขึ้นมา ในการคำนวณ Link Weight จะใช้ Hits Algorithm [2], [9] โดยคำนวณจาก Authorities และ Hubs

Authorities are referenced to by lots of good hubs:

$$a_p = \sum_{q:q\to p} h_q \qquad (1)$$

$q : q \to p$ is all base pages q that have a link to p

Hubs point to lots of good authorities:

$$h_p = \sum_{q:p\to q} a_q \qquad (2)$$

$q : p \to q$ is all base q such that p has a link to q

- Calculate Similarity Range จะทำหน้าที่ในการคำนวณหาขอบเขตที่จะค้นหาในคราวถัดไป Pseudocode ในการคำนวณหา Similarity Range [9]

Given a page, P, let R (the root set) be t (e.g. 200) pages that point to P.

Grow a base set S from R

Run HITS on S

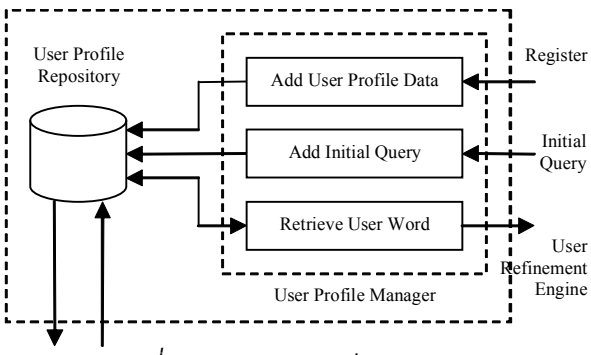
Return the best authorities in S as the best similar-pages for P.

Finds authorities in the “link neighbor-hood” of P

ส่วนที่สี่ คือ Searching จะทำหน้าที่ในการส่งผลลัพธ์ที่ได้จาก Text Operation and Calculate Relevance Processing กลับไปให้เสิร์ชเอนจิ้นทำการค้นหาอีกครั้ง เพื่อให้ได้ Final Result กลับไปยังผู้ค้นหา

3.2.2 User Profile Engine (UPE)

UPE (ในภาพที่ 8) จะมีส่วนประกอบ 2 ส่วน คือ User Profile Manager (UPM) และ User Profile Repository (UPR) โดย UPM จะทำหน้าที่จัดการข้อมูลเบื้องต้นของผู้ค้นหา ได้แก่ การจัดเก็บ ชื่อผู้ค้นหา วันที่ล่าสุดที่ใช้งาน คีย์เวิร์ดที่ใช้ในการค้นหาและช่วงระยะเวลาในการใช้งานของผู้ค้นหา นอกจากนี้เมื่อผู้ค้นหาเข้ามาทำการเสิร์ชในคราวถัดไป UPM จะทำหน้าที่สืบค้นคำศัพท์ของผู้ค้นหานั้น ๆ ที่ถูกเก็บไว้ใน UPR เพื่อทำการป้อนกลับเข้าสู่ URE สำหรับในส่วนของ UPR จะเป็นศูนย์กลางที่ใช้เก็บข้อมูลของผู้ค้นหาหลาย ๆ คนที่เข้ามาใช้งาน โดย UPR จะมีโครงสร้างการจัดเก็บข้อมูล 2 ส่วนหลัก ๆ ส่วนแรกคือการจัดเก็บข้อมูลเบื้องต้นของผู้ค้นหาที่ได้มาจากการใช้งานครั้งแรกและส่วนที่สองคือการจัดเก็บคำศัพท์ของผู้ค้นหาที่ได้จาก UAE หน้าที่หลักอีกประการหนึ่งของ UPE คือจะมีส่วนที่เรียกว่า User Session เพื่อให้ผู้ค้นหาสามารถทำการปรับปรุง Session ในการค้นหาที่เคยใช้ในอดีตได้หรือทำการเริ่ม Session ใหม่ในการค้นหาได้



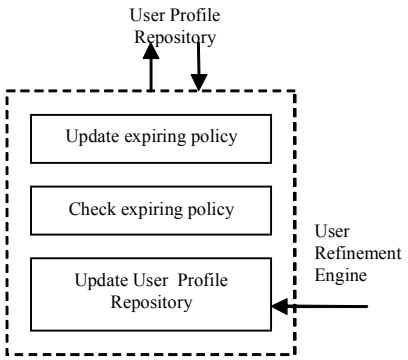
ภาพที่ 8 แสดงรายละเอียดของ UPE

3.2.3 User Adaptive Engine (UAE)

UAE (ในภาพที่ 9) ถือได้ว่าเป็นส่วนประกอบหลักที่สำคัญของเสิร์ชเอนจิ้นที่สร้างขึ้นมา หน้าที่แรกของ UAE คือ จัดการข้อมูลของผู้ค้นหาแต่ละคนที่เข้ามาใช้งานโดยจะตรวจสอบว่าข้อมูลใดของผู้ค้นหาแต่ละคนที่เปลี่ยนแปลงไปจากเดิมบ้างโดยนำข้อมูลใหม่ที่ได้จาก URE ไปเปรียบเทียบกับข้อมูลเดิมที่ถูกเก็บไว้ใน UPR หากข้อมูลใหม่ที่ได้ต่างจากข้อมูลเดิม

ที่มีอยู่ UAE ก็จะทำการปรับปรุงข้อมูลของผู้ค้นหาใหม่ ข้อมูลที่ UAE จะทำการตรวจสอบและเปรียบเทียบได้แก่ คำศัพท์ โดยที่ UAE จะตรวจสอบว่าคำศัพท์นั้นมีอยู่แล้วหรือไม่ ถ้าเป็นคำศัพท์ใหม่ก็จะทำการจัดเก็บลงใน UPR หน้าที่ที่สองของ UAE คือการกำหนด Expiring Policy เพื่อตรวจสอบการหมดอายุของคำศัพท์ ถ้าคำศัพท์ตัวนั้นไม่ได้ถูกใช้ภายในระยะเวลาที่กำหนด เมื่อ UAE ตรวจพบว่าคำศัพท์นั้นหมดอายุก็จะทำการลบทิ้งโดยอัตโนมัติซึ่งจะมีผลดีในแง่ของการช่วยลดภาระในการจัดเก็บข้อมูลของ UPR อีกด้วย หน้าที่ที่สาม ของ UAE คือทำการสร้างคำศัพท์ใหม่ให้กับผู้ค้นหา (Rebuild User Vocabulary) และสุดท้ายหน้าที่ที่สี่ของ UAE คือการคำนวณหาค่าน้ำหนักของคำศัพท์ (Weighting Vocabulary, WV) โดย กำหนดให้ a เป็นคำศัพท์ใด ๆ

WV= (จำนวนครั้งที่ a ถูกใช้งาน / จำนวนครั้งของคำศัพท์ทั้งหมดที่ถูกใช้งาน)



ภาพที่ 9 แสดงรายละเอียดของ UAE

ค่าที่ได้จาก WV จะเป็นตัวแสดงให้เห็นถึงการใช้งานของคำศัพท์นั้น ๆ ว่าถูกใช้งานบ่อยมากแค่ไหน โดยค่า WV จะใช้เป็นเงื่อนไขหนึ่งใน Expiring Policy ในการลบคำศัพท์คือหากคำศัพท์นั้นมีค่า WV น้อยกว่าที่กำหนดและครบระยะเวลาตามที่ Expiring Policy กำหนด คำศัพท์นั้นก็จะถูกทำการลบทิ้งโดยอัตโนมัติ

สำหรับ User Adaptive Web Search Engine Architecture ถูกออกแบบมาเพื่อให้เว็บเสิร์ชเอนจิ้นนั้นสามารถปรับตัวเข้ากับผู้ค้นหาได้ โดยที่ผู้ค้นหาจะใช้เวลาค้นหาน้อยลงในการค้นหาครั้งถัด ๆ ไปและได้เว็บเพจที่ตรงตามความต้องการมากขึ้น

4. บทสรุป

สำหรับในบทความนี้ได้นำเสนอแนวคิดที่สร้างขึ้นมาใหม่ที่เรียกว่า User Adaptive Web Search Engine Architecture เพื่อใช้ในการจัดเก็บข้อมูลต่าง ๆ ของผู้ค้นหา โดยที่เสิร์ชเอนจิ้นสามารถนำข้อมูลของผู้ค้นหามาปรับให้เข้ากับการค้นหาข้อมูลของตัวเสิร์ชเอนจิ้นเองได้อย่างเหมาะสม ซึ่งจะมีส่วนประกอบหลัก 3 ส่วนอันได้แก่ User Refinement Engine (URE) ทำหน้าที่นำผลลัพธ์ที่ได้จากการค้นหาครั้งแรกมาปรับปรุงแล้วนำผลลัพธ์ที่ได้จากการปรับปรุงไปใช้ในการค้นหาครั้งถัดไป ส่วน User Profile Engine (UPE) ทำหน้าที่ในการจัดการข้อมูลและคำศัพท์ของผู้ค้นหา และ User Adaptive Engine (UAE) คอยตรวจสอบและปรับเปลี่ยนคำศัพท์ที่ใช้ในการค้นหาของผู้ค้นหาตามพฤติกรรมของผู้ค้นหาแต่ละราย

สิ่งที่จะทำต่อไปในอนาคตคือการสร้างเว็บเสิร์ชเอนจิ้นให้สามารถใช้งานได้จริงตามสถาปัตยกรรมที่ได้ออกแบบไว้ จากนั้นทำการทดลองและนำผลลัพธ์ที่ได้จากการทดลองไปเปรียบเทียบประสิทธิภาพกับเสิร์ชเอนจิ้นตัวอื่น ๆ ที่มีอยู่ในปัจจุบัน

อนึ่งงานที่สามารถพัฒนาต่อในอนาคตคือการพัฒนาเก็บคำศัพท์และค้นหาคำศัพท์ให้เป็นในลักษณะของ Semantic Network สำหรับผู้ค้นหาเฉพาะรายซึ่งในสถาปัตยกรรมนี้สามารถทำได้โดยการปรับปรุงในส่วนของ User Profile Repository (UPR)

เอกสารอ้างอิง

[1] Lawrence, Steve. and Lee, Giles C., “Inquirus, the NECI Meta search engine”. Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, Elsevier Science, pp. 95-105, 1998.

[2] Ramakrishnan, Raghu. and Gehrke, Johannes. Database Management Systems. New York: McGraw-HILL, 2003.

[3] Baeza–Yates, Ricardo. and Ribeiro-Neto, Berthier. Modern Information Retrieval. New York: ACM Press, 1999.

[4] Porter, Martin. “The Porter Stemming Algorithm.” [Online].Available: http://www.tartarus.org/~matin/PorterStemmer/voc.txt February 12, 2003.

[5] Singh, Shanker Brijesh., “Search Algorithms.”, DRTC Workshop on Digital Libraries: Theory and Practice, Documentation Research and Training Center, Bangalore, March 2003.

[6] Sullivan, Danny. “How Search Engines Work?” [Online]. Available: http://searchenginewatch.com/webmasters/article.php/2168031. October 14, 2002.

[7] software-x.com. “ranking.” [Online]. Available: http://www.software-x.com/software/ranking.html March 8, 2004.

[8] Suel, Torsten. “Web Protocols and Web Search Engines” [Online]. Available: http://cis.poly.edu/cs912/lectures.html. February 12, 2003.

[9] Mooney, Raymond J. “Advances and Link Analysis.” [Online]. Available: http://www.cs.utexas.edu/users/mooney/ir-course/. March 8, 2004.

[10] Cho, Junghoo. and Rajagopalan, Sridhar., “A Fast Regular Expression Index Engine.”, In Proceedings of 2002 International Conference on Data Engineering (ICDE), February 2002. pp. 419-430.

[11] Paradis, Franois. “Information Extraction and Gathering for Search Engines: The Taylor Approach.”, in RIAO (Recherche d’Informations Assiste par Ordinateur), Paris, France, April 12-14, 2000.

[12] Rattana, A. and Davison, A. “N-Sums : A Framework for Web-based Search Engine.”, IIWAS 2001 : The 3rd Int. Conf. on info. Integration and Web-based Applications and Services, Linz, Austria, September 10th-12th, 2001, pp.175-184.

[13] Choi Yuk Chan, Jessica. and Li, Qing. “WebReader: a Mechanism for Automating the Search and Collecting Information from the World Wide Web.”, WISE 2000, Proceedings of the First International Conference on Web Information Systems Engineering. June 19-21, 2000, pp.47-56.

[14] Sheth, Amit., Bertram, Clemens., Avant, David., Hammond, Brian., Kochut, Krysztof. and Yashodhan, Warke. “Industry Report : Managing Semantic Content for the Web.”, IEEE INTERNET COMPUTING. July-August, 2002, p80-87.

[15] วรางฤทธิ์, สุทธิพล. “การออกแบบและพัฒนาระบบเสิร์จเอ็นจิ้นขนาดใหญ่”. วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์. 2544.

[16] ลอยฟ้า, สมาน. “Meta Search Engine: เครื่องมือช่วยค้นข้อมูลบนเว็บ.” [Online]. Available: http://www.thaimisc.com/freewebboard/php/vreply.php?user=sata1&topic=972. กันยายน 26, 2545.

[17] Srinakharinwirot Online Teaching/Learning. “การสืบค้นสารสนเทศบนอินเทอร์เน็ต.” [Online]. Available: http://sot.swu.ac.th/ittools/lesson04/s1t3.htm. กุมภาพันธ์ 22, 2547.