ระบบการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering

Clustering System using Suffix Tree Clustering technique

โดย

ธนพล สุวรรณทัต

รหัสประจำตัว 46066703

อาจารย์ที่ปรึกษา

ผศ.ดร. วรพจน์ กรีสุระเดช

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2548

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	ระบบการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering
นักศึกษา	นายธนพล สุวรรณทัต
อาจารย์ที่ปรึกษา	ผศ.ดร. วรพจน์ กรีสุระเดช
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

ปัจจุบันข้อมูลบนอินเตอร์เน็ทมีอัตราการเพิ่มขึ้นเป็นจำนวนมหาศาล ส่งผลให้ระบบการสืบค้นข้อมูลบนอินเตอร์เน็ทแสดงผลการสืบค้นในลักษณะลำดับรายการที่มีจำนวนมหาศาลเช่นกัน ด้วยเหตุนี้ทำให้ผู้ใช้ไม่ได้รับความสะดวกในการเข้าถึงข้อมูลที่ตรงกับความต้องการ จึงมีแนวคิดในการจัดกลุ่มข้อมูลด้วยการใช้เทคนิค Suffix Tree Clustering โดยการนำ snippets ที่ได้จากระบบสืบค้นทางอินเตอร์เน็ทมาทำการจัดกลุ่ม โดยดูจากการใช้ phrase ร่วมกันของแต่ละ snippets ซึ่งต้องอาศัยหลักการทำงานของโครงสร้างข้อมูลแบบ suffix tree โครงการพัฒนาระบบการจัดกลุ่มด้วยเทคนิค suffix tree clustering ได้พัฒนาขึ้นโดยมีจุดประสงค์เพื่อจัดกลุ่มข้อมูล เพื่ออำนวยความสะดวกให้กับผู้ใช้ในด้านการเข้าถึงข้อมูลที่ต้องการได้รวดเร็วและง่ายขึ้น

Title	Clustering System using Suffix Tree Clustering technique
Student	Mr. Thanapon Suwannathat
Advisor	Asst.Prof.Dr. Worapoj Kreesuradej
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2005

ABSTRACT

Nowadays, there is the large amount of data and information on the Internet and it has continuously increased. This also gives rise to the number of search result. Therefore, sometimes it is not convenient for users accessing to the specific data and information that they want. This results in a new idea of clustering data and information by using a "Suffix Tree Clustering" technique. This technique will be undertaken by bringing "snippets" that are derived from search engines to be clustered. For clustering, it can be seen from the use of a shared phrase of each snippet. This method is driven by using a process of suffix tree data structure. A project of clustering data and information using Suffix Tree Clustering technique is developed for providing convenience to users in order to access the specific data and information that they want faster and easier.

กิตติกรรมประกาศ

ในการพัฒนาโครงงานเรื่องระบบการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering (Clustering System using Suffix Tree Clustering technique) นั้นได้รับความช่วยเหลือจากบุคคลหลาย ๆ ท่าน ทำให้การพัฒนาระบบนั้นสำเร็จลุล่วงไปได้ด้วยดี ข้าพเจ้าจึงขอขอบพระคุณมา ณ ที่นี้ด้วย

ข้าพเจ้าต้องขอขอบพระคุณ ผศ.ดร.วรพจน์ กรีสุระเดช อาจารย์ที่ปรึกษาวิชาโครงการพัฒนาระบบงานที่กรุณาให้คำแนะนำและเป็นที่ปรึกษา อันเป็นประโยชน์ต่อการพัฒนาระบบรวมทั้งเป็นผู้ตรวจสอบความถูกต้องของโครงการฉบับนี้

ขอขอบพระคุณ คุณพ่อ คุณแม่ และบุคคลในครอบครัวที่ได้ให้การสนับสนุนทางด้านกำลังใจในการเรียนจนทำให้โครงการพัฒนาระบบนี้สำเร็จด้วยดี รวมทั้งขอขอบคุณพี่ ๆ และเพื่อน ๆ IS 16.1 ทุก ๆ คนที่ให้ความช่วยเหลือในด้านต่างๆ ที่เกี่ยวกับโครงการไว้ ณ ที่นี้

ธนพล สุวรรณทัต

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

หน้า

ตารางที่

สารบัญรูป

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมา

ระบบสืบค้นข้อมูลบนอินเตอร์เน็ท (Search Engine) ปัจจุบันได้รับความนิยมเป็นอย่างสูง เนื่องจากอินเตอร์เน็ทเป็นแหล่งข้อมูลขนาดใหญ่เปรียบเสมือนห้องสมุดขนาดใหญ่ มีจำนวนข้อมูลในระดับหลายพันล้านเว็บเพจ และมีอัตราการเติบโตของจำนวนข้อมูลในอินเตอร์เน็ทมีมากกว่า 1,500,000 หน้า/วัน ถึงแม้ว่าการมีข้อมูลจำนวนมากมายมหาศาลจะเป็นสิ่งที่ดี แต่ก็ส่งผลให้ระบบการทำงานของการสืบค้นข้อมูลมีปัญหาด้านต่างๆมากมาย เช่น ปัญหาการแสดงผลการสืบค้นให้กับผู้ค้นหา หรือปัญหาการเข้าถึงข้อมูลที่ตรงความต้องการของผู้ค้นหา เพราะข้อมูลที่สืบค้นมาได้มีจำนวนมากเกินไป และข้อมูลจำนวนมากที่ได้มาก็ไม่ใช่ข้อมูลที่ตรงกับความต้องการทั้งหมด

ระบบสืบค้นบนอินเตอร์เน็ทจะมีการแสดงผลแบบเรียงลำดับรายการยาวๆของ snippets การมีข้อมูลเว็บเพจจำนวนมากๆ ทำให้รายการผลการสืบค้นในรูปของ snippets มีรายการยาวมากๆ ส่งผลให้ผู้ใช้ไม่สะดวกในการเข้าถึงข้อมูลที่ตรงตามความต้องการ คือ ผู้ค้นหามักจะไม่เข้าไปดูข้อมูลหรือผลการสืบค้นที่อยู่ลำดับหลังๆ จึงทำให้ผลการสืบค้นในลำดับหลังๆไม่ถูกนำไปใช้แม้จะเกี่ยวข้องกับคำที่ค้นหาและมีเนื้อหาที่ผู้ค้นหาต้องการก็ตาม การแสดงผลการสืบค้นข้อมูลในรูปลำดับของผลการสืบค้น โดยเรียงลำดับจากผลที่มีความสัมพันธ์กับคำที่ใช้ในการค้นหาจากมากที่สุดไปถึงน้อยที่สุด เมื่อมีข้อมูลจำนวนมากในอินเตอร์เน็ท ทำให้ผลการสืบค้นมีจำนวนมากตามไปด้วย ซึ่งผลการสืบค้นนี้จะมีความสัมพันธ์กับคำที่ใช้ในการสืบค้นในหลากหลายเนื้อหาที่แตกต่างกันไป เช่น สืบค้นด้วยคำว่า SEA GAME จะได้เอกสารที่มีความสัมพันธ์กับคำว่า SEA GAME ในหลากหลายเนื้อหาได้แก่ SEA GAME ซึ่งเป็นการแข่งขันกีฬาของภูมิภาคเอเชียตะวันออกเฉียงใต้ , SEA GAME ที่เป็นกีฬาทางทะเล เช่น เรือใบ กระดานโต้คลื่น, SEA GAME ซึ่งเป็นเกมส์คอมพิวเตอร์ที่มีทะเลเข้ามาเกี่ยวข้อง หรือแม้แต่ การตกปลาในทะเลก็มีเนื้อหาที่เกี่ยวข้องกับ SEA GAME ได้เช่นกัน เมื่อผล การสืบค้นที่มีเนื้อหาแตกต่างกันถูกนำมาแสดงผลรวมกัน ในรูปแบบของลำดับรายการทำให้เกิดความยุ่งยากแก่ผู้ค้นหาเพราะผู้ที่ค้นหาข้อมูลจะต้องมองหาเฉพาะข้อมูลที่มีเนื้อหาที่ต้องการจากลำดับรายการของผลการสืบค้นข้อมูลที่แสดงโดยระบบสืบค้นบนอินเตอร์เน็ท จะเห็นได้ว่าเป็นการค้นหาที่ซ้ำซ้อนและผู้ใช้ต้องเสียเวลาในการค้นหาข้อมูลที่ต้องการอีกครั้งหนึ่ง แทนที่จะช่วยประหยัดเวลาในการค้นหาอย่างที่ควรจะ

เป็น แนวทางหนึ่งที่สามารถนำมาแก้ปัญหานี้ได้คือ การจัดกลุ่มผลการสืบค้นข้อมูลให้อยู่ในรูปของกลุ่มผลการสืบค้นตามเนื้อหาที่ผลนั้นๆเกี่ยวข้อง จึงแสดงลำดับของกลุ่มผลการสืบค้น ซึ่งจะช่วยให้การแสดงผลการสืบค้นมีประสิทธิภาพมากยิ่งขึ้น อันจะช่วยประหยัดเวลาของผู้ที่ค้นหาข้อมูลโดยเลือกกลุ่มของเนื้อหาที่สนใจแทนที่จะเลือกดูผลการค้นหาทีละรายการ และยังเพิ่มโอกาสที่ผลการค้นหาในลำดับหลังๆจะถูกนำมาใช้ เพราะถูกนำมาแสดงในกลุ่มเดียวกับผลอื่นๆที่มีเนื้อหาด้านเดียวกัน (ภาณุพงศ์ ชวะวิทย์. 2547)

การจัดกลุ่มผลการสืบค้นข้อมูลบนอินเตอร์เน็ท (Web Search Results Clustering) นี้ มีลักษณะการทำงานเป็นแบบ Online เราจะใช้การจัดกลุ่มผลการสืบค้นตามหลักของ Suffix Tree Clustering (STC) โดยจะทำการจัดกลุ่มผลการสืบค้นซึ่งข้อมูลที่เป็นผลการสืบค้นในระบบงานนี้จะหมายถึง snippets ซึ่งเป็นคำอธิบายสั้น ๆ ของเอกสารแต่ละลำดับรายการที่ได้รับจากการสืบค้นทางอินเทอร์เน็ต ซึ่งจะระบุกลุ่มจากผลการสืบค้น โดยดูจากการใช้ phrase ร่วมกัน และสร้างกลุ่มของผลการสืบค้นที่มีความคล้ายคลึงกับ phrase นั้น เป็นลักษณะการจัดกลุ่มไปพร้อมกับการค้นหาป้ายชื่อของกลุ่ม ผลจะได้กลุ่มมาพร้อมกับป้ายชื่อ แล้วแสดงผลตามลำดับรายการความสำคัญของกลุ่ม โดยดูจากความยาวของ phrase และจำนวนสมาชิกภายในกลุ่ม เพื่อให้ผู้ค้นหาเลือกค้นหาตามป้ายชื่อของกลุ่มที่ต้องการ

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาลักษณะการทำงานและการแสดงผลของระบบสืบค้นข้อมูลบนอินเตอร์เน็ท เพื่อให้เกิดความเข้าใจและรู้ถึงจุดเด่น จุดด้อย ของระบบที่มีอยู่ในปัจจุบัน ซึ่งจะนำไปสู่การแก้ปัญหาและการพัฒนาให้ดียิ่งขึ้น
2. เพื่อศึกษากระบวนการทำงานของการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering และนำไปประยุกต์ใช้กับการจัดกลุ่มผลการสืบค้นบนอินเตอร์เน็ทได้อย่างมีประสิทธิภาพ
3. เพื่อค้นหาข้อดี-ข้อเสีย ของการจัดกลุ่มผลการสืบค้นของระบบสืบค้นข้อมูลบนอินเตอร์เน็ทตามหลัก Suffix Tree Clustering (STC)
4. เพื่อพัฒนาระบบการสืบค้นข้อมูลและการแสดงผลให้มีประสิทธิภาพมากขึ้นโดยการจัดกลุ่มผลการสืบค้น เพื่อที่จะทำให้ผู้ใช้เข้าถึงข้อมูลที่ตรงกับความต้องการได้ง่ายและรวดเร็วมากขึ้น

1.3 ขอบเขตของการศึกษา

1. ทำการจัดกลุ่มข้อมูลซึ่งเป็นผลการสืบค้นจากระบบสืบค้นบนอินเทอร์เน็ต (Search Engine) โดยใช้อัลกอริทึม Suffix Tree Clustering (STC)
2. ข้อมูลที่ได้รับจากระบบสืบค้นบนอินเทอร์เน็ตนั้น ในโครงการพัฒนาระบบงานนี้ จะใช้เพียงแค่ snippets เป็นข้อมูลหลักในการจำแนกและจัดกลุ่มข้อมูลเท่านั้น
3. โครงการพัฒนาระบบงานนี้ได้พัฒนาขึ้นเพื่อรองรับการจัดกลุ่มผลการสืบค้นที่เป็นภาษาอังกฤษเท่านั้น

1.4 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษาหลักการทำงานและการแสดงผล ของระบบสืบค้นบนอินเตอร์เน็ท (Search Engine)
2. ศึกษาหลักการและกระบวนการทำงานของอัลกอริทึม Suffix Tree Clustering (STC)
3. ศึกษาหลักการทำงานร่วมกันของระบบสืบค้นบนอินเตอร์เน็ทและการจัดกลุ่มข้อมูลด้วยอัลกอริทึม Suffix Tree Clustering (STC)
4. ศึกษาภาษาโปรแกรม JAVA
5. ออกแบบและพัฒนาระบบ
6. ทดสอบการใช้งานของระบบ
7. สรุปผลการศึกษาและการดำเนินงาน

1.5 ผลที่คาดว่าจะได้รับ

1. เข้าใจในหลักการทำงานและการแสดงผลของระบบสืบค้นบนอินเตอร์เน็ท (Search Engine)
2. เข้าใจหลักการและขั้นตอนการทำงานของอัลกอริทึม Suffix Tree Clustering (STC)
3. ทราบถึงข้อดีข้อเสียของการจัดกลุ่มข้อมูลด้วยอัลกอริทึม Suffix Tree Clustering (STC)
4. ทำให้สามารถเข้าถึงข้อมูลจากการสืบค้นข้อมูลบนอินเทอร์เน็ตได้ง่ายและรวดเร็วและตรงกับความต้องการมากขึ้น

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

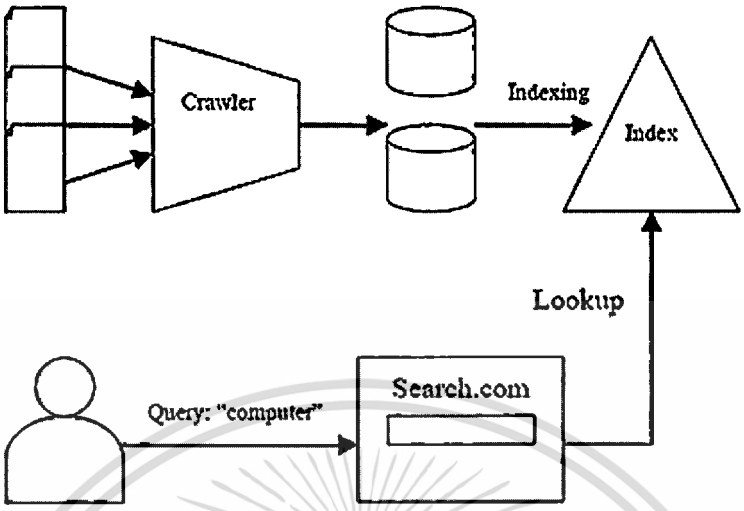
ในการพัฒนาระบบสาสนเทศการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering ได้มีการนำทฤษฎีต่าง ๆ มาใช้ดังต่อไปนี้

2.1 ระบบสืบค้นข้อมูลบนอินเตอร์เน็ท (Search Engines)

Crawler-Based Search Engine หรือ Search Engines ในปัจจุบันมีอยู่เป็นจำนวนมากเช่น Google, AltaVista, Lycos, Yahoo, HotBot สามารถแบ่งออกเป็น 3 ประเภท (วิริช ลีลาภัทร และ พรฤดี เนติโสภากุล . 2548 : 65-66) คือ

2.1.1 Indexing Search Engine หรือ Keyword Search

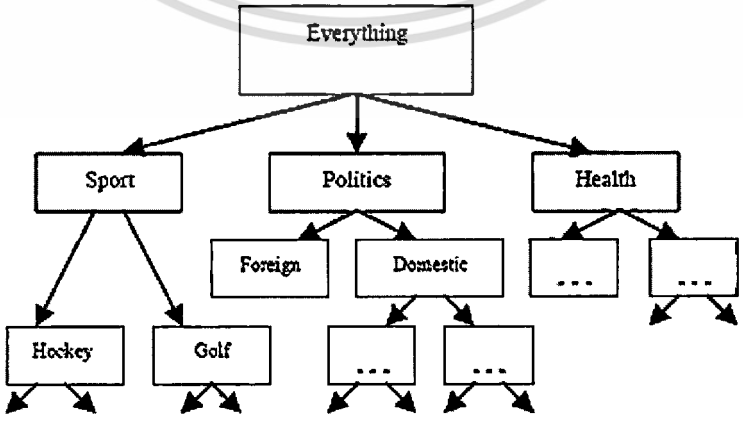
เป็นระบบซอร์ฟแวร์ที่มีโปรแกรมโรบอต (Robot) หรือบางครั้งเรียกว่า สไปเดอร์ (Spider) หรือ ครอว์เลอร์ (Crawler) ทำหน้าที่ไปยังเว็บไซต์ต่างๆในอินเทอร์เน็ตและอ่านเว็บเพจจากไซต์เหล่านั้น เพื่อนำมาสร้างดัชนีรายการของเว็บเพจโดยอัตโนมัติ การทำดัชนีรายการด้วย Search Engine จึงสามารถสร้างดัชนีรายการของเว็บเพจได้จำนวนมากกว่าของไดเรคทอรี เนื่องจากดัชนีรายการของไดเรคทอรีจะถูกจัดการโดยมนุษย์แทนที่จะใช้คอมพิวเตอร์ ถ้าหากเว็บเพจที่ถูกทำดัชนีแล้วเกิดการเปลี่ยนแปลง Search Engine จะทราบถึงการเปลี่ยนแปลงและจัดนำเว็บเพจที่มีการเปลี่ยนแปลงนั้นมาสร้างดัชนีใหม่ ซึ่งการสร้างดัชนีใหม่นี้อาจส่งผลต่อการจัดลำดับของเว็บเพจนั้น ตัวอย่างของ Search Engine ที่มีอยู่ในปัจจุบัน เช่น HotBot , AltaVista เป็นต้น โดย Indexing Search Engine จะมีสถาปัตยกรรมดังในรูปที่ 2.1



รูปที่ 2.1 สถาปัตยกรรมของ Indexing Search Engine

2.1.2 Subject Directory

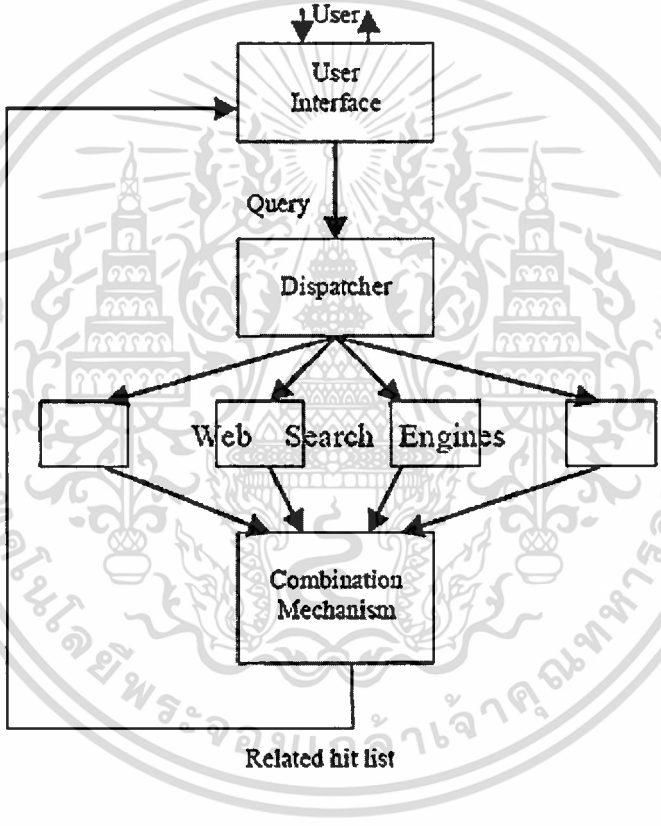
จะมีลักษณะเป็นรายการของเว็บไซต์หรือเว็บเพจ ซึ่งได้มีการจัดรวมไว้โดยการแบ่งเป็นหมวดหมู่ ตามลักษณะและหัวข้อของเนื้อหา เพื่อให้ผู้ค้นหาสามารถเลือกค้นได้ในแต่ละหมวดหมู่หัวข้อเนื้อหา (Subject Categories) จะถูกแบ่งเป็นหมวดหมู่ย่อย (Sub-Categories) ที่จัดเรียงลดหลั่นกันหลายระดับจนกระทั่งถึงรายการชื่อเว็บไซต์ที่นำเสนอเนื้อหาที่สอดคล้องกับหมวดหมู่ย่อยนั้น พร้อมกับมีตัวเชื่อมโยง (Link) เพื่อชี้ไปยังเว็บไซต์นั้น กระบวนการสร้างและจัดทำผ่านการดำเนินการโดยบุคคลหรือกลุ่มบุคคลที่ทำหน้าที่ในการจัดการหมวดหมู่หัวข้อเนื้อหา เลือกและจำแนกเว็บไซต์ลงในหมวดหมู่ โดย Subject Directory จะมีสถาปัตยกรรมดังในรูปที่ 2.2



รูปที่ 2.2 สถาปัตยกรรมของ Subject Directory

2.1.3 Multi-Engine Search Tool หรือ Meta-Search Engine

คือระบบค้นหาที่สามารถสืบค้นข้อมูลจาก Search Engine และหรือ Web Directories ได้มากกว่า 1 ตัวในเวลาเดียวกัน และแสดงผลการสืบค้นที่ได้รับจาก Search Engine เหล่านั้นในเวลาเดียวกัน โดยเสนอผลการสืบค้นในรูปแบบที่สะดวก ซึ่งในบางครั้งจะมีการปรับแต่งผลการสืบค้นที่ได้รับทั้งหมดให้อยู่ในรูปแบบเดียวกันและบูรณาการผลการสืบค้นเหล่านี้เข้าเป็นชุดเดียวกัน โดย Meta-Search Engine จะมีสถาปัตยกรรมดังในรูปที่ 2.3



รูปที่ 2.3 สถาปัตยกรรมของ Meta-Search Engine

แม้ว่า Search Engine แต่ละประเภทจะช่วยในการค้นหาข้อมูลบนอินเทอร์เน็ตได้เป็นอย่างดี แต่ปัญหาและข้อจำกัดของ Search Engine แต่ละประเภทก็ยังมีอยู่ (วิริช ลีลาภัทร และ พรฤดี เนติโสภากุล . 2548 : 65-66) เช่น

- การชี้ไปยังหน้าเอกสารที่ยังไม่มีการปรับปรุงข้อมูลหรือไม่มีข้อมูลตามที่ระบุไว้ (Dead Link)

- Subject หรือ Directory จะพบกับปัญหาเรื่องปริมาณสาระสนเทศที่เก็บรวบรวมไว้มีจำนวนไม่มากนัก รวมถึงระเวลาในการจัดเก็บข้อมูลเป็นเวลานานอันเป็นผลมาจากต้องใช้คนในการตรวจสอบและจัดเก็บ
- Indexing Search Engine หรือ Keyword Search จะให้จำนวนของผลลัพธ์หรือผลการสืบค้นมากเกินไป ขาดการประเมินและกลั่นกรองสาระของเว็บเพจที่ได้เก็บรวบรวมมา
- ระบบสืบค้นให้ลำดับรายการยาวๆ ทำให้ผู้สืบค้นไม่ได้รับความสะดวกในการเข้าถึงข้อมูลที่ตรงกับความต้องการ และเสียเวลาในการค้นหาข้อมูลที่ต้องการ เช่น ผู้สืบค้นต้องการสืบค้นคำว่า " Jaguar " ในความหมายที่สัมพันธ์กับ " เสือ " ผู้สืบค้นอาจจะต้องไปค้นหาในลำดับรายการที่ 10 , 11 , 32 และ 71 เป็นต้น (Zeng et.al. 2004)

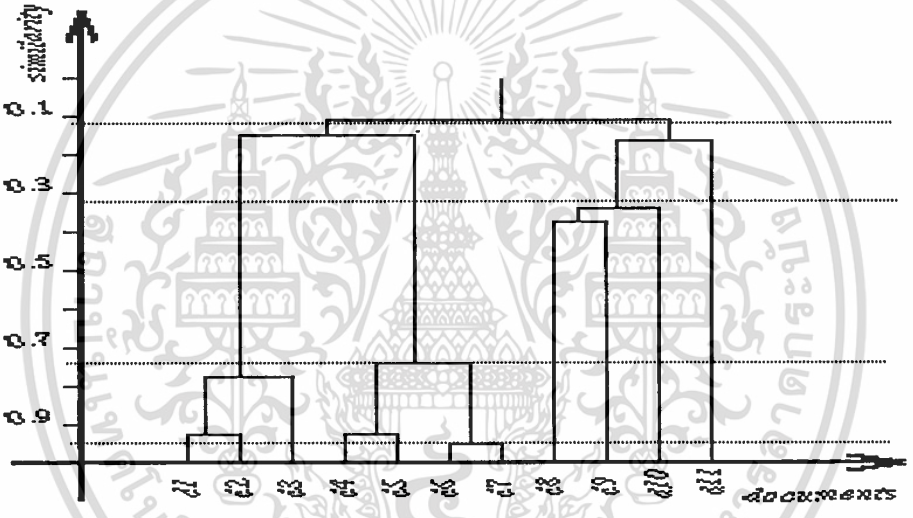
2.2 วิธีการจัดกลุ่มข้อมูล (Clustering Methods)

การจัดกลุ่ม (Clustering) มีวัตถุประสงค์หลักเพื่อแยกข้อมูลออกเป็นกลุ่มย่อยๆ หรือ คลัสเตอร์ตามลักษณะความเหมือนกันของข้อมูล โดยข้อมูลที่เหมือนกันจะถูกจัดให้อยู่ในกลุ่มเดียวกัน ซึ่งข้อมูลที่อยู่ภายในกลุ่มเดียวกันจะมีค่าความเหมือนกันมากกว่าข้อมูลที่อยู่ต่างกลุ่มกัน เทคนิคการจัดกลุ่มได้นำไปใช้กับวัตถุที่มีค่าของคุณสมบัติเป็นค่าเชิงตัวเลขอย่างแพร่หลาย ซึ่งสามารถจัดกลุ่มข้อมูลเชิงตัวเลขได้เป็นอย่างดี ปัจจุบันได้เริ่มนำเทคนิคการจัดกลุ่มมาใช้กับวัตถุที่มีค่าของคุณสมบัติเป็นค่าเชิงตัวอักษรเพิ่มมากขึ้น การจัดกลุ่มผลการสืบค้นของ Search Engine ในรูปของ Web Search Results Clustering ก็เป็นส่วนหนึ่งของการจัดกลุ่มด้วยค่าคุณสมบัติเป็นค่าเชิงตัวอักษร ซึ่งมีเทคนิคการจัดกลุ่ม 2 รูปแบบ (ภาณุพงศ์ ชวะวิทย์. 2547) คือ

2.2.1 Hierarchical Clustering

มีลักษณะการจัดกลุ่มเป็นลำดับชั้น โดยเริ่มจากข้อมูลทั้งหมดจะอยู่เพียงคลัสเตอร์เดียวในระดับบนสุด แล้วเมื่อผ่านขั้นตอนการจัดกลุ่มข้อมูลจะได้คลัสเตอร์ย่อยๆ จนกระทั่งได้คลัสเตอร์ที่มีข้อมูลเพียงชุดเดียวที่ระดับล่างสุด ในขั้นตอนสุดท้ายของการทำงานของอัลกอริทึมจะได้โครงสร้างต้นไม้ของกลุ่มข้อมูล ซึ่งแสดงถึงความสัมพันธ์ของกลุ่มข้อมูลว่ามีความสัมพันธ์กันอย่างไร โดยถ้าทำการตัดโครงสร้างต้นไม้ในระดับที่ต้องการข้อมูลก็จะแยกจากกันเป็นกลุ่มๆ Hierarchical Clustering มีการแสดงผลลัพธ์ด้วยแผนภาพโครงสร้างต้นไม้ หรือ ที่เรียกอีกอย่างหนึ่งว่า " dendrogram " ดังแสดงในรูปที่ 2.4 ซึ่งช่วยสร้างความเข้าใจได้ง่ายขึ้น โครงสร้างของ dendrogram จะประกอบด้วยชั้นของ node แสดงถึงการจัดกลุ่มในชั้นนั้นๆ ในแต่ละคลัสเตอร์เส้น

ที่เชื่อมระหว่าง node แสดงถึงการรวมกันของคลัสเตอร์เป็นคลัสเตอร์ใหม่อีกคลัสเตอร์หนึ่ง และถ้าเราตัดแผนภาพ dendrogram ตามขวางในแต่ละระดับชั้นเราจะได้ผลของการทำ clustering ในระดับชั้นนั้นๆ วิธีที่นิยมใช้กันมากที่สุดคือ วิธี agglomerative จะเริ่มจากข้อมูลที่อยู่ต่างคลัสเตอร์และในแต่ละขั้นตอนการทำงานจะรวมข้อมูลที่เหมือนกันหรือคล้ายคลึงกันให้อยู่ในคลัสเตอร์เดียวกัน ทำซ้ำไปเรื่อยๆจนกระทั่งจำนวนของคลัสเตอร์มีค่าน้อยที่สุด และวิธี divisive วิธีนี้จะเริ่มจากรวมข้อมูลทั้งหมดให้อยู่ในคลัสเตอร์เดียว แล้วทำการพิจารณาว่าคลัสเตอร์ใดควรจะถูกแยกออกมาและจะแยกมันออกมาด้วยวิธีใด ซึ่งเป็นสิ่งที่สำคัญที่สุดในกระบวนการทำงานของ divisive

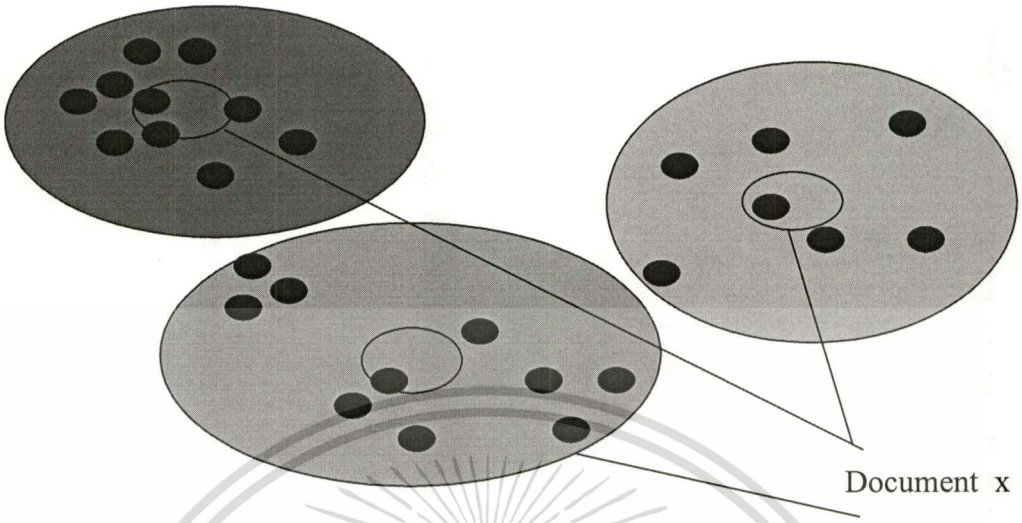


รูปที่ 2.4 แผนภาพ Hierarchical Clustering

จากรูปที่2.4 ถ้าจัดกลุ่มที่ค่า similarity = 0.1 เอกสารมี 2 กลุ่ม คือ{ d1 , d2 , d3 , d4 , d5 , d6 , d7 } , { d8 , d9 , d10 , d11 } และ ถ้าจัดกลุ่มที่ค่า similarity = 0.7 จะได้เอกสาร 6 กลุ่มคือ { d1 , d2 , d3 } , { d4 , d5 , d6 , d7 } , { d8 } , { d9 } , { d10 } , { d11 } เป็นต้น

2.2.2 Flat Clustering

เป็นลักษณะการจัดกลุ่มที่ตรงข้ามกับแบบ Hierarchical Clustering เพราะ Flat Clustering จะเป็นลักษณะค่อยๆจัดกลุ่มทีละเอกสารที่เข้ามาตามคุณลักษณะว่าเอกสารนี้ควรจะอยู่ในกลุ่มใด ถ้าไม่มีคุณลักษณะที่ตรงกับกลุ่มที่มีอยู่เดิมก็จะทำการสร้างกลุ่มใหม่ ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 แผนภาพ Flat Clustering

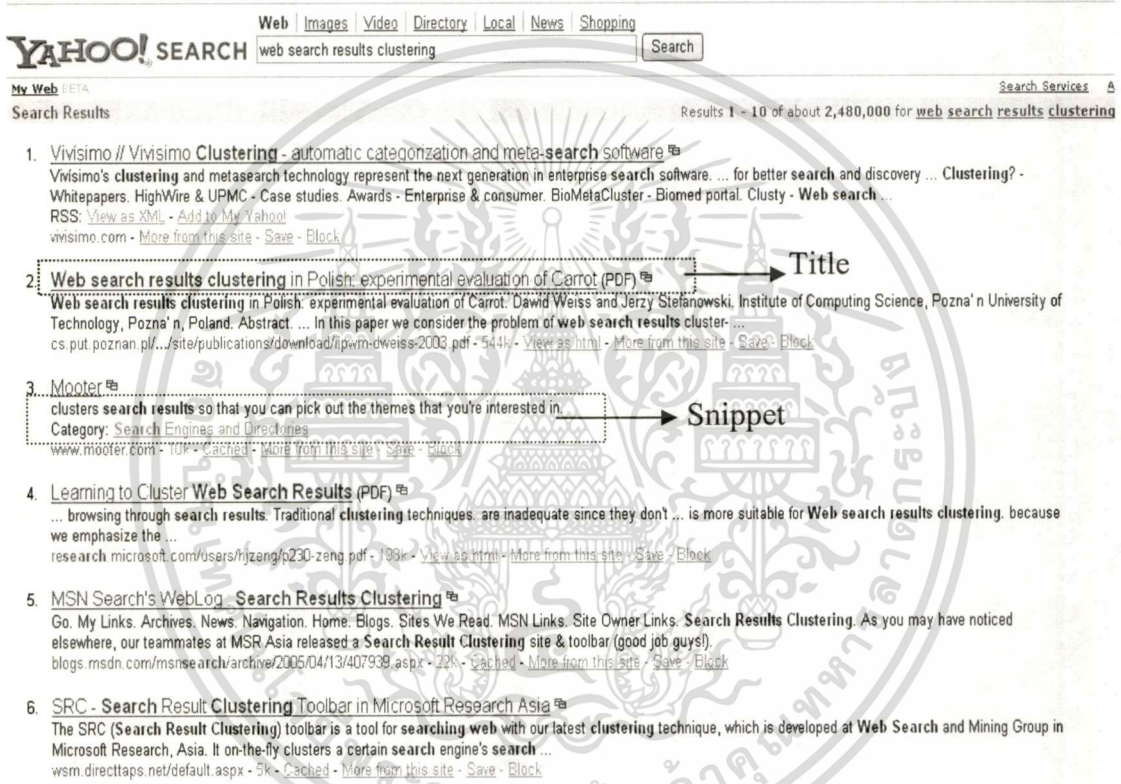
จากรูปที่ 2.5 เมื่อเอกสาร x ถูกนำเข้าระบบเพื่อจัดกลุ่ม จะมีการคำนวณค่า similarity ที่ใกล้เคียงกับแต่ละกลุ่ม คือ C1 , C2 , C3 ค่าความเหมือนในกลุ่มใดมากที่สุด ก็จะนำเอกสาร x เข้าไปอยู่เป็นสมาชิกในกลุ่มนั้น แต่ในกรณีที่ การคำนวณค่า similarity ของเอกสาร x กับกลุ่มเอกสารที่มีอยู่เดิมคือ C1 , C2 , C3 แล้วปรากฏว่าเอกสาร x ไม่เหมือนหรือคล้ายคลึงกับกลุ่มใดเลย เราจะต้องทำการสร้างกลุ่มขึ้นมาใหม่ โดยมีเอกสาร x เป็นสมาชิกภายในกลุ่ม

ปัจจุบันการจัดกลุ่มได้ถูกพัฒนาขึ้นมาด้วยหลายๆรูปแบบและวิธีในการจัดกลุ่ม เพื่อให้ผลการจัดกลุ่มมีความยืดหยุ่นและถูกต้องเหมาะสมมากที่สุด เช่น การนำกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม เข้ามามีส่วนร่วมในการพัฒนาการจัดแบ่งกลุ่มข้อมูล ซึ่งมีผลดีคือสามารถจัดแบ่งกลุ่มของข้อมูลที่มีการกระจายของกลุ่มข้อมูลจำนวนมาก แต่มีข้อมูลจำนวนน้อยๆได้ เช่น วิธี Self-Organizing Maps Clustering หรือการนำเอาคุณสมบัติขิง Fuzzyset มาใช้ในการจัดแบ่งกลุ่ม ซึ่ง Fuzzy set จะเพิ่มข้อมูลในส่วนของค่าความเป็นสมาชิก (degree of membership) เข้ามา เช่นวิธี fuzzy C-mean clustering

2.3 การจัดกลุ่มผลการสืบค้นข้อมูลบนอินเทอร์เน็ต (Web Search Results Clustering)

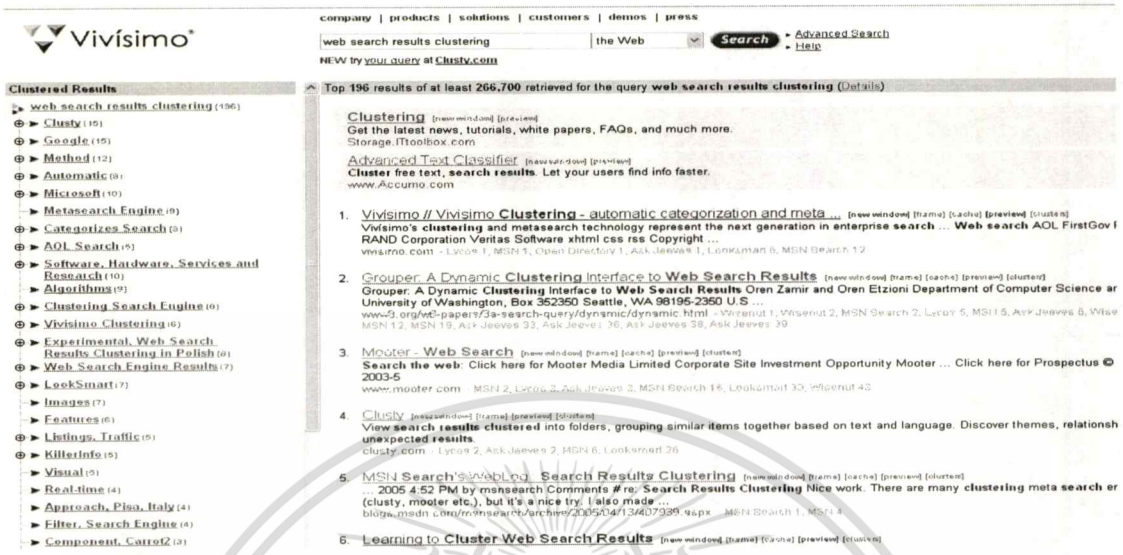
เนื่องจากผู้สืบค้นได้รับผลการสืบค้นจากระบบสืบค้นบนอินเทอร์เน็ต เป็นรายการยาวๆ ทำให้ผู้สืบค้นต้องเสียเวลาเพื่อค้นหาข้อมูลที่ตรงกับความต้องการอีกครั้งหนึ่ง หรือคำที่ค้นหานั้นมีหลายความหมาย เช่น ผู้สืบค้นต้องการค้นหาคำว่า “ Jaguar ” ในความหมายที่เกี่ยวข้องกับสัตว์ แต่ระบบสืบค้นทั่วๆไปอาจให้ผลการสืบค้นตามที่ผู้สืบค้นต้องการในลำดับที่ 10, 11, 32 หรือ 71

ของลำดับรายการค้นหาที่ได้มา จากตัวอย่างในรูปที่ 2.6 เป็นผลการสืบค้นคำว่า " Web Search Results Clustering " จะเรียงลำดับความสำคัญของเอกสาร นั่นคือผลที่ได้มามีจำนวนมากเกินความต้องการของผู้สืบค้น ส่งผลให้ผู้สืบค้นต้องทำการค้นหาด้วยตนเองอีกครั้งหนึ่งว่าเอกสารที่ตนเองต้องการนั้นอยู่ในลำดับที่เท่าไร โดยดูจาก title และ snippets ที่ระบบแสดงให้ จากตัวอย่างเป็นระบบสืบค้นของ yahoo.com ได้ผลการสืบค้นทั้งหมด 2,480,000 รายการ



รูปที่ 2.6 ตัวอย่างผลการสืบค้นของระบบสืบค้น yahoo.com

จากปัญหาดังกล่าวทำให้มีการนำผลการสืบค้นมาจัดกลุ่ม เพื่ออำนวยความสะดวกให้กับผู้สืบค้น ค้นหาเอกสารที่ต้องการตามกลุ่มต่างๆ และสามารถเข้าถึงข้อมูลที่ต้องการได้สะดวกรวดเร็วมากยิ่งขึ้น ดังแสดงในรูปที่ 2.7 เป็นการจัดกลุ่มผลการสืบค้น ของระบบสืบค้นชื่อ vivisimo.com ผู้ค้นหาสามารถเลือกค้นหาในกลุ่มที่ต้องการได้



รูปที่ 2.7 ตัวอย่างผลการสืบค้นของระบบสืบค้น vivisimo.com

ลักษณะการจัดกลุ่มผลการสืบค้นแบ่งเป็น 4 รูปแบบ คือ (Ferragina et.al. 2005)

2.3.1 Single Word and Flat Clustering

เป็นการใช้ Single Word เป็นตัวกำหนด feature ของเอกสาร(ใช้คำๆเดียวเป็นตัวแทนของเอกสาร) เพื่อนำไปจัดกลุ่มตามรูปแบบของ flat clustering

2.3.2 Sentence and Flat Clustering

เป็นการใช้ Sentence เป็นตัวกำหนด feature ของเอกสาร (ใช้ sentence เป็นตัวแทนเอกสาร) เพื่อนำไปจัดกลุ่มตามรูปแบบของ flat clustering

2.3.3 Single Word and Hierarchical Clustering

เป็นการใช้ Single Word เป็นตัวกำหนด feature ของเอกสารเพื่อนำไปจัดกลุ่มตามรูปแบบของ Hierarchical Clustering เช่น ระบบของ Frequent Item Hierarchical Clustering(FIHC) ใช้ความถี่ของคำ(vector model) ในการจัดกลุ่ม

2.3.4 Sentence and Hierarchical Clustering

ใช้ Sentence เป็นตัวกำหนด feature ของเอกสาร เพื่อจัดกลุ่มตามแบบของ Hierarchical Clustering เช่น ระบบ SNAKET นำเสนอแนวทางการพัฒนาการจัดกลุ่มผลการสืบค้นข้อมูลบนอินเตอร์เน็ท โดยชื่อว่า SHOC มีการนำ Suffix Array ซึ่งเป็นโครงสร้างข้อมูลรูปแบบหนึ่ง มาใช้ในการค้นหา phrase แล้วนำ phrase เป็นป้ายชื่อ(Label) มาจัดกลุ่มด้วยSingula Value Decomposition (SVD) ร่วมกับ Latent Semantic Indexing (LSI)

ปัจจุบันมีหลายๆระบบที่พัฒนา Web-Snippets Clustering ในลักษณะของ (meta)search engine ประกอบด้วย vivisimo.com, clusty.com, kartoon.com, mooter.com, copernic.com, iboogie.com, groxis.com, dogpile.com ในปี 2001 – 2003 searchenginewatch.com จัดงาน " best meta search engine award " ในส่วนของการจัดกลุ่มที่มีประสิทธิภาพรางวัลนี้ได้แก่ vivisimo.com และในเดือนมกราคม 2005 google.com เลือกให้ vivisimo.com เป็นระบบที่มีการจัดเตรียมผลการสืบค้นที่ดีที่สุด ดังนั้น google และ Microsoft ได้ให้ความสนใจในการจัดกลุ่มผลการสืบค้น เพราะในอนาคตมันจะเป็นเทคโนโลยีของการจัดลำดับความสำคัญของผลการสืบค้น (" clustering technology is the PAGERANK of the future ") จากรายงานดังกล่าวแสดงให้เห็นถึงการให้ความสนใจและความสำคัญของการจัดกลุ่มผลการสืบค้นจากปี 2001 - 2005 เรื่อยมา แต่เทคนิคของการจัดกลุ่มระบบสืบค้นในปัจจุบันยังมีข้อบกพร่องอยู่มากเช่น ความชัดเจนของป้ายชื่อกลุ่ม (label) ความซ้ำซ้อนของข้อมูลในแต่ละกลุ่ม (overlap) และ ความสามารถในการแสดงผลของข้อมูลทั้งหมดหลังจากการจัดกลุ่ม (coverage) เป็นต้น (Ferragina et.al. 2005)

2.4 Suffix Tree Clustering (STC)

Suffix Tree Clustering (STC) เป็นเทคนิคในการจัดกลุ่มข้อมูลที่มีลักษณะการจัดกลุ่มเป็นแบบ Flat Clustering โดยจะทำงานร่วมกับโครงสร้างข้อมูลแบบ Suffix tree และจะใช้เวลาในการประมวลผลแปรผันตามกับจำนวนเอกสารทั้งหมดใน collection จะทำการจัดกลุ่มโดยพิจารณาจากการใช้ phrase ร่วมกันของเอกสาร STC จะใช้ phrase เป็นตัวแทนของเอกสาร โดยที่ phrase ในที่นี้จะหมายถึง ลำดับของคำที่ต่อเนื่องกันตั้งแต่ 1 คำหรือมากกว่าที่ปรากฏในเอกสาร Suffix Tree Clustering จะมีขั้นตอนในการจัดกลุ่มข้อมูลดังต่อไปนี้ (Zamir and Etzioni . 1998)

การทำงานของ STC แบ่งเป็น 3 ขั้นตอนคือ (Zamir and Etzioni . 1998)

1. Document " Cleaning " หรือ การทำ "Preprocessing" เป็นขั้นตอนเตรียมข้อมูลก่อนที่จะนำไปทำการจัดกลุ่ม เพื่อลดขนาดและความซ้ำซ้อนของข้อมูล หรือลดสิ่งที่ไม่เป็นประโยชน์ต่อการทำงานของระบบ โดยการจัดการกับตัวอักษรและคำในแต่ละเอกสาร โดยจะมีกระบวนการทำงานตามขั้นตอนต่อไปนี้

- Sign คือ การกำจัดเครื่องหมาย คำ และ สัญลักษณ์ ต่าง ๆ ที่มีผลต่อคุณภาพในการจัดกลุ่มข้อมูล เช่น "=" , "(" , "{" , ">" เป็นต้น
- Stop word คือ การกำจัดคำที่ไม่สื่อความหมาย และไม่เป็นประโยชน์ในการนำไปจัดกลุ่มข้อมูล โดยทั่วไปประกอบด้วย คำนำหน้านามในภาษาอังกฤษ(articles) เช่น " the " , " an " , " a " เป็นต้น คำบุพบท(preposition) เช่น " for " , " of " เป็นต้น

คำสรรพนาม(pronouns) เช่น " you " , " we" เป็นต้น การทำ stop word จะช่วยให้จำนวนของ index ลดลงมากกว่า 40 % (Osinski and Stefanowski. 2003)

- Stemming คือ การลดรูปคำ แปลงคำ เช่น การตัด prefix, suffix , การลดรูปของคำจากรูปพหูพจน์ให้เป็นเอกพจน์ เช่น " connected " , " connecting " , จะถูกแปลงให้อยู่ในรูปของคำว่า " connect " สำหรับคำศัพท์ภาษาอังกฤษ ปัจจุบันที่นิยมใช้คือ Porter Algorithm และการทำ stemming จะช่วยลดความแตกต่างกันของ index

2. Identifying Base Clusters ในขั้นตอนนี้จะเป็นขั้นตอนในการคัดเลือก base cluster ซึ่งจะพิจารณาจากการใช้ phrase ร่วมกันของเอกสาร โดย base cluster จะเป็นกลุ่มของเอกสารที่มีการใช้ phrase ร่วมกัน ในการคัดเลือก base cluster จะมีการนำโครงสร้างข้อมูลแบบ Suffix Tree มาใช้เพื่อช่วยในการค้นหา base cluster ในขั้นตอน Identifying Base Clusters จะมีขั้นตอนในการทำงานดังต่อไปนี้

2.1 สร้างรายการลำดับของคำ - ในขั้นตอนนี้จะเป็นการนำข้อมูลในเอกสาร (snippet) มาสร้างรายการลำดับของคำที่ต่อเนื่องกัน (phrase) ก่อนที่จะนำไปสร้าง Suffix Tree โดยคาดว่าลำดับของคำที่ถูกสร้างขึ้นนั้นมีความน่าจะเป็นว่ามันจะปรากฏเป็น phrase ในภายหน้า ซึ่ง STC จะใช้ phrase เป็นตัวแทนของเอกสาร เทคนิคในการใช้สร้างรายการลำดับคำของเอกสารที่ใช้ก็คือ เทคนิค " n- gram "

n-gram เป็นการสร้างรายการลำดับของ n คำ ซึ่งใช้ลักษณะความน่าจะเป็นว่ามันจะปรากฏเป็นวลี (phrase) ในภายหน้า ดังตัวอย่างการแตกประโยคไปเป็นกลุ่มของคำที่อยู่ติดกันตามความยาวของ " n " (Chambers et.al. 2004)

" I don't know what to say "

n = 1 → 1-gram (unigram) : I , don't , know , what , to , say

n = 2 → 2-gram (bigram) : I don't , don't know , know what , what to , to say

n = 3 → 3-gram (trigram) : I don't know , don't know what , know what to , what to say

:

:

n-gram

จะเห็นได้ว่า การใช้เทคนิค n-gram ในการสร้างรายการลำดับของคำ จะทำให้จำนวนของคำใน phrase ซึ่งอาจจะถูกเลือกเป็นป้ายชื่อของกลุ่มนั้นมีความเป็นระเบียบ เพราะจะมีจำนวนของคำใน phrase ได้ไม่เกินจำนวนขนาดของ n-gram

2.2 สร้าง Suffix Tree – ในขั้นตอนนี้จะเป็นการนำข้อมูลในเอกสารทั้งหมดที่ผ่านการสร้างรายการลำดับของคำด้วยเทคนิค n-gram แล้ว มาทำการสร้าง Suffix Tree โดย Suffix Tree จะมีส่วนประกอบต่าง ๆ ดังต่อไปนี้

 - สัญลักษณ์ " สี่เหลี่ยม " แทน leaf ของ Suffix Tree โดยภายใน leaf จะประกอบไปด้วยตัวเลขคู่ลำดับ 2 หมายเลข หมายเลขแรกจะระบุถึง หมายเลขของเอกสารที่ phrase นั้น ๆ ปรากฏอยู่ และหมายเลขที่ 2 จะระบุถึงลำดับของรายการลำดับคำที่ปรากฏในเอกสารนั้น ๆ

—— - สัญลักษณ์ " เส้นตรง " แทน edge ของ Suffix Tree โดยแต่ละ edge จะมี label กำกับอยู่ ซึ่ง label จะเป็นคำ ๆ หนึ่งที่ประกอบอยู่ในรายการลำดับคำ (phrase) ของเอกสาร

 - สัญลักษณ์ " วงกลม " แทน node ของ Suffix Tree

Suffix Tree จะมีลักษณะพิเศษคือ edge ที่ออกไปจาก node เดียวกันจะต้องมี label ของ edge ไม่ซ้ำกัน ในการสร้าง Suffix Tree จะเป็นการนำคำทีละคำใน phrase แทน label ที่กำกับ edge ของ Suffix Tree มาสร้างเรียงลำดับกันไปโดย leaf จะอยู่ส่วนท้ายสุดของ Suffix Tree เพื่อระบุถึงหมายเลขเอกสารที่ phrase นั้นปรากฏอยู่และ ลำดับของ phrase ในเอกสาร

ตัวอย่างการสร้าง Suffix Tree จากเอกสาร 3 เอกสารดังต่อไปนี้

D1 : cat ate cheeses

D2 : mouse ate chesses too

D3 : cat ate mouse too

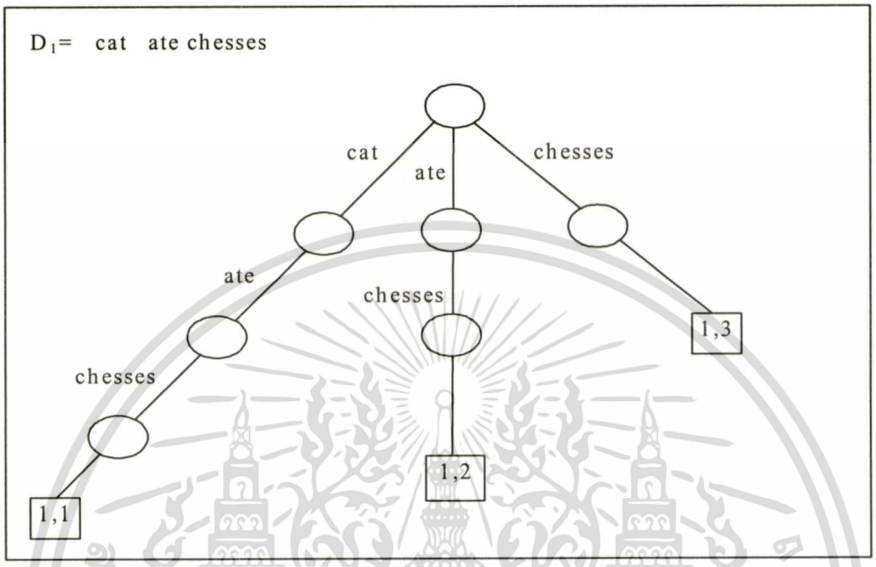
จากเอกสาร D1, D2, D3 เมื่อนำไปสร้างรายการลำดับของคำด้วยเทคนิค n-gram จะได้รายการลำดับของคำโดยใช้ n-gram <= 3 ดังต่อไปนี้

D1 : cat ate cheeses , ate cheeses , cheeses

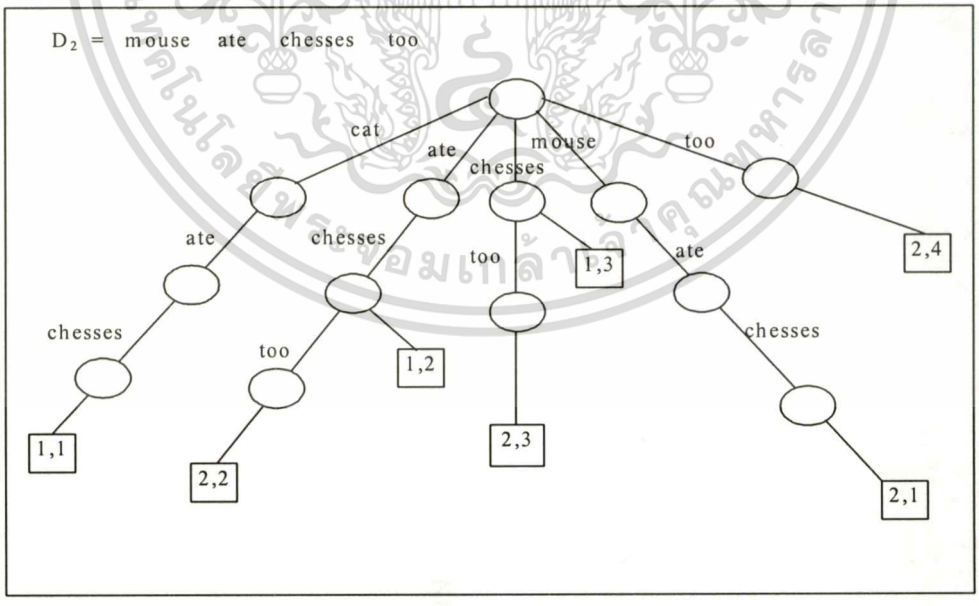
D2 : mouse ate cheese , ate cheeses too , cheeses too , too

D3 : cat ate mouse , ate mouse too , mouse too , too

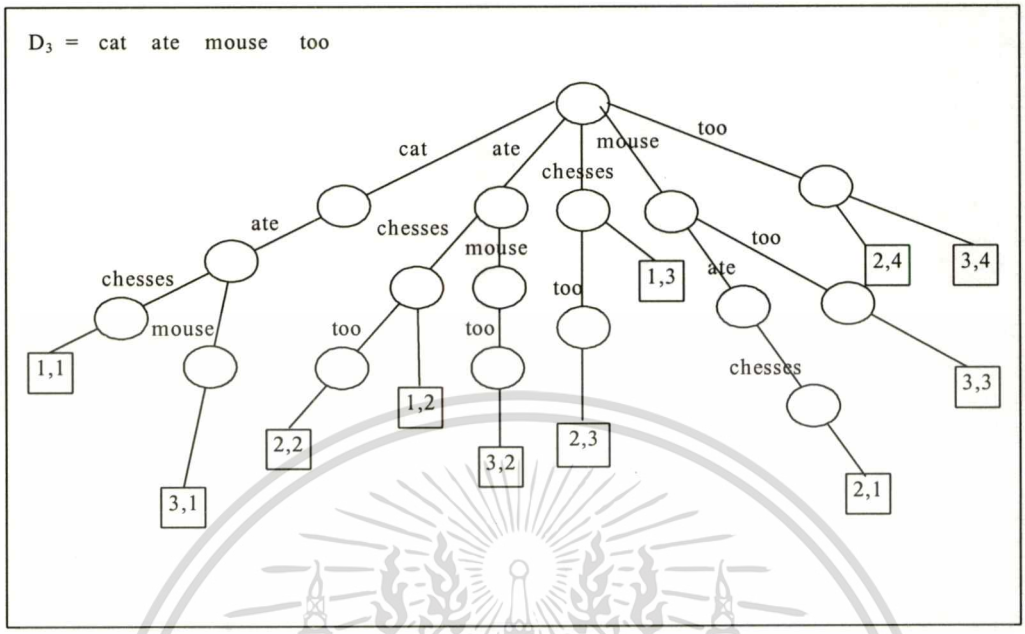
สามารถสร้าง Suffix Tree จากข้อมูลในเอกสารทั้ง 3 เอกสารได้ดังแสดงในรูปที่ 2.8, รูปที่ 2.9, รูปที่ 2.10



รูปที่ 2.8 แสดงตัวอย่างการสร้าง Suffix Tree ของเอกสารที่ 1

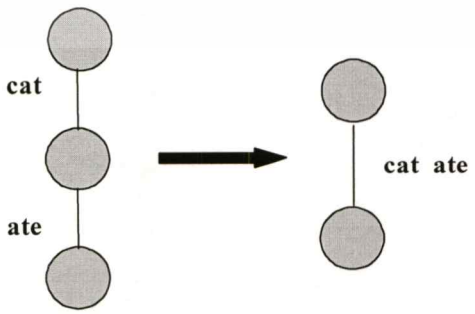


รูปที่ 2.9 แสดงตัวอย่างการสร้าง Suffix Tree ของเอกสารที่ 1 และ 2



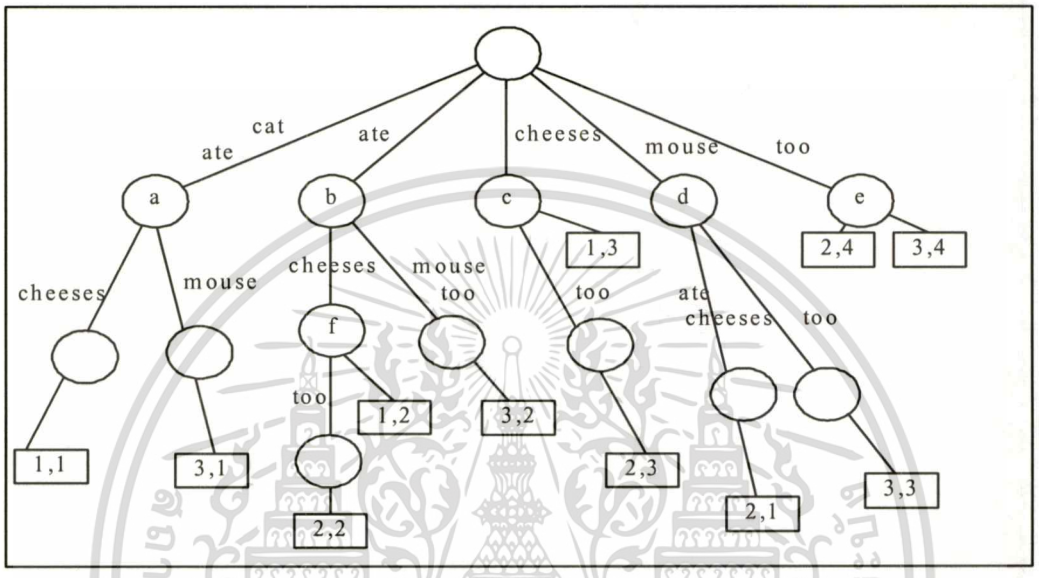
รูปที่ 2.10 แสดงตัวอย่างการสร้าง Suffix Tree ของเอกสารที่ 1,2 และ 3

2.3 ยุบรวม node ของ Suffix Tree - เมื่อสร้าง Suffix Tree จากข้อมูลของทุกเอกสารเรียบร้อยแล้ว ทำการยุบรวม node ของ Suffix Tree ที่เป็นไปได้เพื่อรวม label ของแต่ละ edge เพื่อให้ได้ phrase ที่มีจำนวนคำมากที่สุด เนื่องจาก STC จะให้ความสำคัญกับจำนวนคำใน phrase ยิ่งมีจำนวนคำมากจะยิ่งให้ความสำคัญมาก ซึ่งจะได้อธิบายต่อไปในขั้นตอนต่อไป และยังเป็นการลด จำนวนของ node ที่อาจจะถูกเลือกให้เป็น base cluster ลง ทำให้เวลาที่ใช้ในการค้นหาและระบุ base cluster นั้นน้อยลงอีกด้วย หลักในการยุบรวม node ของ Suffix Tree สามารถแสดงได้ดังรูปที่ 2.11



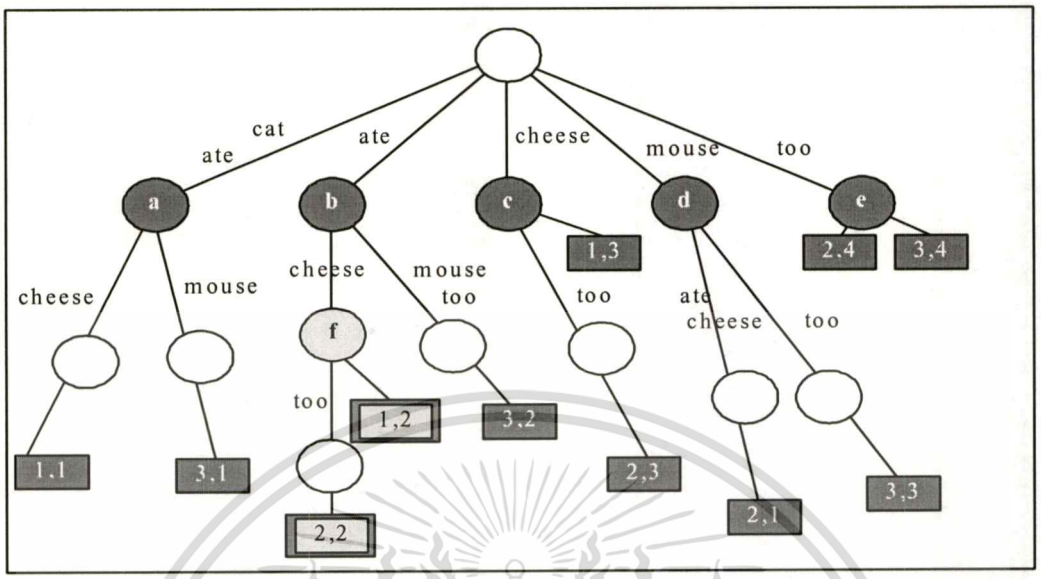
รูปที่ 2.11 การรวม node ของ Suffix Tree

จากรูปที่ 2.11 เมื่อทำการยุบรวม node ของ Suffix Tree แล้ว label ของ edge จะเป็นการนำเอา label ของ node ที่ถูกยุบรวมมาเชื่อมต่อกัน ให้เป็น phrase ที่มีจำนวนคำมากขึ้น เมื่อทำการยุบรวม node แล้วสามารถแสดง Suffix Tree ได้ดังรูปที่ 2.12



รูปที่ 2.12 Suffix Tree ที่ยุบรวม node แล้ว

2.4 ค้นหาและระบุ Base Clusters - ในขั้นตอนนี้จะทำการระบุ base clusters โดยจะพิจารณาจากโครงสร้างข้อมูล Suffix Tree และจะมีหลักเกณฑ์ในการเลือก base cluster คือ จำนวนของเอกสาร หรือ จำนวน leaf ที่ปรากฏภายใน sub - tree ของ node ใด ๆ มีตั้งแต่ 2 เอกสาร หรือ 2 leaf ขึ้นไป node นั้นจะถูกเลือกให้เป็น base cluster โดยที่ป้ายชื่อ (phrase) ของ base cluster คือ การนำ edge- label จากเส้นทางตั้งแต่ root จนถึง node นั้น ๆ มารวมกัน จะถูกกำหนดให้เป็น ป้ายชื่อของกลุ่ม (phrase) และ เอกสารที่ปรากฏจะถูกกำหนดให้เป็นเอกสารภายในกลุ่มของ base cluster นั้น ๆ เช่น จากรูปที่ 2.13 base cluster ของ node f จะมีป้ายชื่อของกลุ่มคือ ate + cheeses = ate cheese และ เอกสารในกลุ่มของ base cluster f คือ เอกสารที่ 1 และ เอกสารที่ 2



รูปที่ 2.13 แสดงการระบุ base cluster จากโครงสร้างข้อมูล Suffix Tree

จากรูปที่ 2.13 สามารถระบุ base cluster ได้ทั้งหมด 6 base cluster ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 แสดงผลการทำงานของขั้นตอน Identifying Base Clusters ของ STC

Node	*Phrase*	*Documents*	*S(B)*
a	cat ate	1,3	(2*2) = 4
b	ate	1,2,3	(3*0) = 0
c	cheese	1,2	(2*0) = 0
d	mouse	2,3	(2*0) = 0
e	too	2,3	(2*0) = 0
f	ate cheese	1,2	(2*2) = 4

2.5 คำนวณคะแนนของ Base Cluster - ในขั้นตอนนี้จะเป็นการคำนวณคะแนนของ base cluster (S (B)) เพื่อนำคะแนนนี้ไปใช้ในการคัดเลือกตัวแทนของกลุ่มและคำนวณคะแนนในการจัดลำดับความสำคัญของการแสดงผลในขั้นตอน Combining Base Clusters ต่อไป

สูตรที่ใช้ในการคำนวณค่า S(B)

$$s(B) = |B| * f(|P|)$$

$$f(|P|) = \begin{cases} 0, & if \quad |P| = 1 \\ |P|, & if \quad 2 \le |P| \le 6 \\ \alpha, & if \quad |P| > 6 \end{cases}$$

เมื่อ

B คือ Base Cluster

$|B|$ คือ จำนวนของเอกสารใน B

P คือ phrase ที่จะนำมาใช้เป็นป้ายชื่อของ Base Cluster

$|P|$ คือ จำนวนของคำที่อยู่ใน phrase

α คือ ค่าคงที่

จากสูตร S (B) จะเท่ากับจำนวนของเอกสารใน base cluster คูณกับค่าของ *f(|p|)* ซึ่งค่าของ *f(|p|)* จะมีค่าเป็น 0 เมื่อจำนวนคำใน phrase ซึ่งเป็นป้ายชื่อของ base cluster มีจำนวนคำเท่ากับ 1 คำ (single word) จะมีค่าเท่ากับ 2 ถึง 6 เมื่อจำนวนคำใน phrase มีจำนวนคำเท่ากับ 2 ถึง 6 คำ และจะมีค่าเท่ากับค่าคงที่ใด ๆ เมื่อมีจำนวนคำใน phrase มากกว่า 6 คำขึ้นไป การคำนวณคะแนนของ base cluster สามารถแสดงตัวอย่างการคำนวณคะแนนได้ดังต่อไปนี้

จากตารางที่ 2.1 base cluster "a" มีป้ายชื่อของกลุ่ม (phrase) คือ "cat ate" ที่มีจำนวนคำภายใน phrase เท่ากับ 2 คำ และ มีจำนวนของเอกสารที่ปรากฏใน base cluster 2 เอกสารคือ เอกสารที่ 1 และ เอกสารที่ 3 จากสูตร

$$s(B) = |B| * f(|P|)$$

$|B| = 2$

$f(|P|) = 2$

ดังนั้น S(B) = 2*2 = 4

จากการให้คะแนนของ base cluster จะเห็นว่า STC จะให้ความสำคัญของ phrase ที่มีจำนวนของคำที่ประกอบกันภายใน phrase หรือ ป้ายชื่อของกลุ่มมากกว่า 1 คำ มากกว่า phrase ที่มีจำนวนคำภายใน phrase เพียงคำเดียว (single word) สังเกตได้จากการคิดคะแนน S(B) คือ เมื่อ

จำนวนคำใน phrase มีเพียง 1 คำจะให้ค่า $f(|P|)$ เป็น 0 ส่งผลให้ S(B) มีค่าเป็น 0 ทันที และผลของการคำนวณคะแนนของ base cluster จะแสดงในตารางที่ 2.1

3. Combining Base Clusters จากผลการทำงานในขั้นตอน Identifying Base Cluster แสดงให้เห็นว่าเอกสารอาจมีการใช้ phrase ร่วมกันมากกว่า 1 phrase ส่งผลให้เอกสารสามารถปรากฏอยู่ได้ในหลาย ๆ base cluster ทำให้ base cluster บางกลุ่มมีความเหมือนกันสูงมาก ในขั้นตอนนี้จึงเป็นการรวม based cluster ที่มีความเหมือนกัน โดยจะพิจารณาจากการใช้เอกสารร่วมกันของ base cluster คือ base cluster ใดมีการใช้เอกสารร่วมกันมาก ก็จะมีความคล้ายกันมาก เมื่อรวม base cluster ที่คล้ายกันแล้วจะได้ merged cluster ซึ่งในขั้นตอน Combining Base Clusters มีหลักการทำงานดังนี้

คำนวณค่า similarity ของแต่ละ base cluster เพื่อที่จะค้นหาว่า base cluster ใดบ้างที่มีความคล้ายกัน ถ้า base cluster คู่ใดมีค่า similarity เป็น 1 ก็ให้ทำการรวม based cluster เข้าด้วยกัน แล้วเลือก based cluster ที่มีคะแนน S(B) สูงที่สุดเป็นตัวแสดงผล เป็นตัวแทนของ base cluster ทั้งหมดใน merged cluster โดยที่ป้ายชื่อของกลุ่มก็คือป้ายชื่อ (phrase) ของ base cluster ที่มีคะแนน S(B) สูงสุด และเอกสารที่ปรากฏใน merged cluster นั้นคือเอกสารทั้งหมดที่ปรากฏใน base cluster ทุก base cluster ที่เป็นสมาชิกของ merged cluster นั้น

สูตรในการคำนวณค่า similarity

$$similarity(B_1, B_2) = \begin{cases} 1, & if \quad (|B_1 \cap B_2| / |B_1|) > \partial \quad and \\ & \quad (|B_1 \cap B_2| / |B_2|) > \partial \\ 0, & otherwise \end{cases}$$

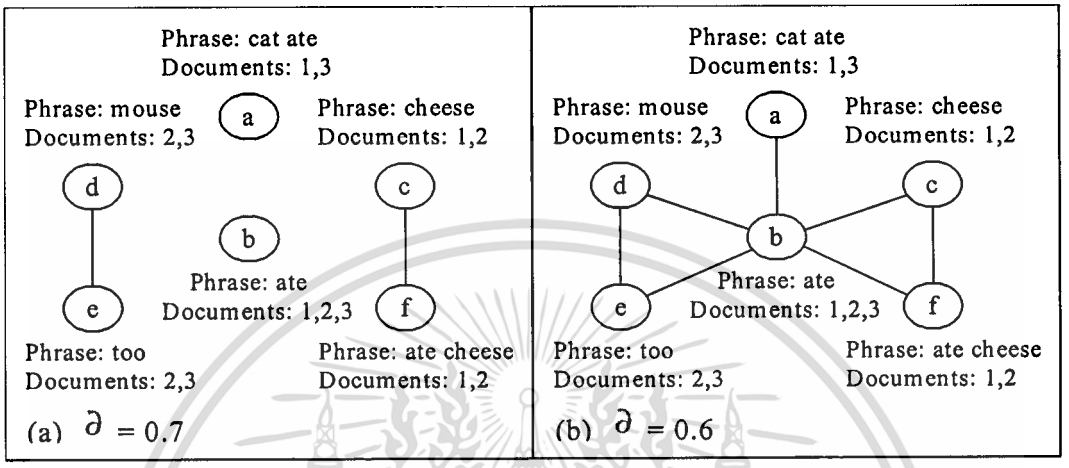
เมื่อ

B คือ Base Cluster

∂ คือ ค่า minimum support

ค่า similarity จะเท่ากับ 1 ก็ต่อเมื่อ จำนวนสมาชิกของ B_1 กับ B_2 ที่ซ้ำกัน (จำนวนของเอกสารที่ base cluster ทั้ง 2 ใช้ร่วมกัน) หารด้วยจำนวนสมาชิกของ B_1 และ B_2 (จำนวนของเอกสารที่ปรากฏใน base cluster 1 และ base cluster 2) มีค่ามากกว่าค่าความเหมือนกันขั้นต่ำ (minimum support หรือ ∂) ที่กำหนดทั้งสองกรณี ในกรณีอื่นๆจะถือว่าค่า similarity มีค่าเป็น 0 ซึ่งโดยปกติค่าของ minimum support จะมีค่าตั้งแต่ 0.5 - 0.8 แล้วแต่ผู้พัฒนาจะกำหนด

จากการคำนวณค่า similarity สามารถนำมาวาดเป็นกราฟของความเหมือนกันของ based cluster ดังแสดงในรูปที่ 2.14 (Zamir and Etzioni . 1998)



รูปที่ 2.14 กราฟเส้นการเชื่อมโยงตามค่า similarity ของแต่ละ base cluster

จากรูปที่ 2.14 base cluster ใดที่มีเส้นเชื่อมกันจะหมายความว่ามีค่าของ similarity = 1 ซึ่งมีความคล้ายกันจะถูกรวมเข้าไว้เป็นกลุ่ม ๆ เดียวกัน

จากรูปที่ 2.9 (a) ระบบกำหนดให้ค่า ∂ = 0.7 นั่นคือจำนวนสมาชิกต้องเหมือนกัน (จำนวนของเอกสารที่ base cluster ทั้ง 2 ใช้ร่วมกัน)70% ของสมาชิกภายในกลุ่มทั้งสอง จากเดิมจะได้ base cluster = 6 กลุ่ม คือ base cluster a,b,c,d,e,f เมื่อคิดค่า similarity แล้วทำการยุบรวมกัน จะเหลือ merged cluster = 4 กลุ่มคือ merged cluster {a},{b},{c,f},{d,e} ดังตารางที่2.2

จากรูปที่ 2.9 (b) ระบบกำหนดให้ค่า ∂ = 0.6 นั่นคือจำนวนสมาชิกต้องเหมือนกัน (จำนวนของเอกสารที่ base cluster ทั้ง 2 ใช้ร่วมกัน)60% ของสมาชิกภายในกลุ่มทั้งสอง จากเดิมจะได้ base cluster = 6 กลุ่ม คือ base cluster a,b,c,d,e,f เมื่อคิดค่า similarity แล้วทำการยุบรวมกัน จะเหลือ merged cluster = 1 กลุ่มคือ merged cluster {a,b,c,d,e,f} ดังตารางที่ 2.2

ตารางที่ 2.2 แสดงผลการทำงานของขั้นตอน Combining Base Clusters ของ STC

Figure	*Cluster Number*	*base cluster*	*Documents*	*S(C)*
(a)	1	a	1,3	4
	2	b	1,2,3	0
	3	d,e	2,3	0
	4	c,f	1,2	4
(b)	1	a,b,c,d,e,f,g	1,2,3	8

ในท้ายที่สุดแล้วการจัดกลุ่มข้อมูลซึ่งเป็นผลสืบค้นที่ได้รับจากอินเทอร์เน็ต อาจจะได้กลุ่มของข้อมูลเป็นจำนวนมาก ดังนั้นจึงต้องทำการจัดลำดับความสำคัญในการแสดงผล โดยการคำนวณคะแนนของ merged cluster (S(C)) เพื่อเป็นคะแนนในการจัดลำดับความสำคัญในการแสดงผลให้กับผู้ใช้ โดยสูตรที่ใช้ในการคำนวณคะแนนของ merged cluster (S(C)) คือ

$$S(c) = \sum_{b \in c} S_b$$

เมื่อ

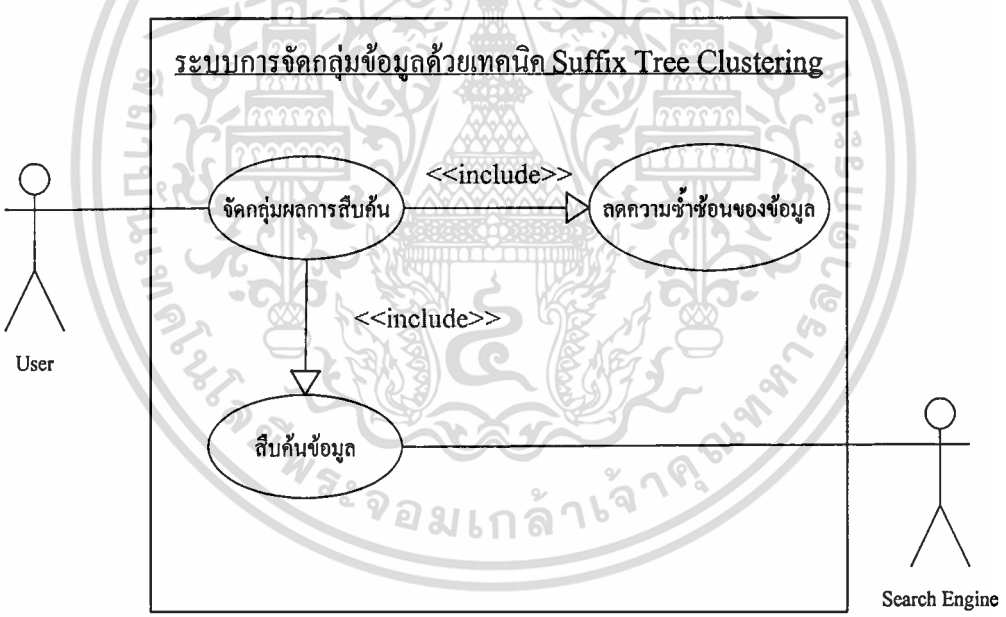
b คือ base cluster ที่ถูกรวมเข้ามาไว้ใน merged cluster

S(C) จะเป็นผลรวมของคะแนนของ base cluster หรือ S(B) ทั้งหมดที่อยู่ใน merged cluster เมื่อคำนวณค่าของ S(C) เรียบร้อยแล้วก็จะแสดงผล merged cluster โดยจะแสดงผลเรียงลำดับตามค่า S(C) จากมากไปหาน้อย และจำนวนของกลุ่มที่จะแสดงผลให้ผู้ใช้จะเป็นไปตามความสนใจเช่น 5 10 หรือ 20 กลุ่มเป็นต้น

บทที่ 3
การวิเคราะห์และออกแบบระบบงาน

ในการวิเคราะห์และออกแบบโครงการพัฒนาระบบงานนี้ จะได้นำเสนอในรูปแบบของ UML (Unified Modeling Language)โดยจะใช้ Use Case Diagram, Activity Diagram และ Class Diagram ในการนำเสนอ

3.1 Use Case Diagram ของระบบ



รูปที่ 3.1 Use Case Diagram ของระบบ

จากรูปที่ 3.1 แสดง Use Case Diagram ของระบบการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering โดยภายในระบบประกอบไปด้วย Use Case ต่าง ๆ ดังต่อไปนี้

- จัดกลุ่มผลการสืบค้น
- ลดความซ้ำซ้อนของข้อมูล
- สืบค้นข้อมูล

3.1.1 Actor Descriptions

ในส่วนนี้จะอธิบายถึงรายระเอียดของ Actor ของระบบ ซึ่งภายในระบบจะประกอบไปด้วย Actor ดังต่อไปนี้

- User
- Search Engine

รายละเอียดของ Actor User และ Actor Search Engine จะถูกแสดงในตารางที่ 3.1 และตารางที่ 3.2

ตารางที่ 3.1 รายละเอียดของ Actor : User

Actor	User (ผู้ใช้)
รายละเอียด	บุคคลทั่วไปที่ต้องการสืบค้นข้อมูลจาก Search Engine
ตัวอย่าง	สืบค้นข้อมูล

ตารางที่ 3.2 รายละเอียดของ Actor : Search Engine

Actor	Search Engine
รายละเอียด	Search Engine ที่ให้บริการในการสืบค้นข้อมูลผ่านเครือข่ายอินเทอร์เน็ต
ตัวอย่าง	สืบค้นข้อมูลและส่งผลลัพธ์ของการสืบค้นกลับไปยังระบบ

3.1.2 Use Case Descriptions

ส่วนนี้จะอธิบายถึงรายละเอียดต่าง ๆ ของ Use Case ภายในระบบ ดังแสดงในตารางที่ 3.3, ตารางที่ 3.4 และ ตารางที่ 3.5

ตารางที่ 3.3 รายละเอียดของ Use Case "จัดกลุ่มผลการสืบค้น"

Use Case	จัดกลุ่มผลการสืบค้น
Brief Description	ระบบจัดกลุ่มผลการสืบค้น ทำการจัดกลุ่มข้อมูลที่ได้จากการสืบค้นจาก Search Engine
Actor	User (ผู้ใช้)
Trigger	ผู้ใช้ต้องการสืบค้นข้อมูลจาก Search Engine พร้อมทั้งจัดกลุ่มข้อมูลที่ได้จากการสืบค้น

ตารางที่ 3.3 (ต่อ)

Pre-condition	ข้อมูล(ผลสืบค้น)ที่จะถูกนำมาจัดกลุ่ม ได้รับการลดความซ้ำซ้อน (Document Cleaning)เรียบร้อยแล้ว
Post-condition	ข้อมูล(ผลสืบค้น) ได้รับการจัดกลุ่ม
Primary scenario	1. ผู้ใช้ป้อนคำที่ต้องการสืบค้น 2. ระบบทำการสืบค้นข้อมูลจาก Search Engine 3. ลดความซ้ำซ้อนของข้อมูล (Document Cleaning) 4. แบ่งข้อมูล(snippet) ออกเป็นรายการลำดับ n คำ ตามหลัก n-gram 5. สร้าง Suffix Tree 6. ยุบรวม node ของ Suffix Tree 7. เลือก Base Cluster 8. คำนวณคะแนนของแต่ละ Base Cluster S(b) 9. คำนวณค่า similarity ของแต่ละ Base Cluster 10. รวม Base Cluster ที่มีความคล้ายกันเข้าด้วยกัน 11. คำนวณคะแนนของ Merged Cluster S(c) 12. จัดลำดับความสำคัญในการแสดงผลกลุ่มข้อมูล 13. แปลงชื่อของกลุ่มในรูปแบบ stem ให้อยู่ในรูปแบบปรกติ 14. แสดงผลการจัดกลุ่ม
Alternatives	2a. ไม่มีผลการสืบค้นข้อมูล : จบการทำงาน 7a. ไม่สามารถเลือก Base Cluster ได้ : จบการทำงาน

ตารางที่ 3.4 รายละเอียดของ Use Case "ลดความซ้ำซ้อนของข้อมูล"

Use Case	ลดความซ้ำซ้อนของข้อมูล
Brief Description	ทำการเตรียมข้อมูล(ผลการสืบค้น) ลดความซ้ำซ้อนของข้อมูลและกำจัดข้อมูลส่วนที่ไม่เป็นประโยชน์ ก่อนที่จะนำข้อมูลไปทำการจัดกลุ่มในขั้นตอนต่อไป
Actor	-
Pre-condition	-
Post-condition	ข้อมูลได้รับการลดความซ้ำซ้อนและกำจัดส่วนที่ไม่เป็นประโยชน์ออกไป

ตารางที่ 3.4 (ต่อ)

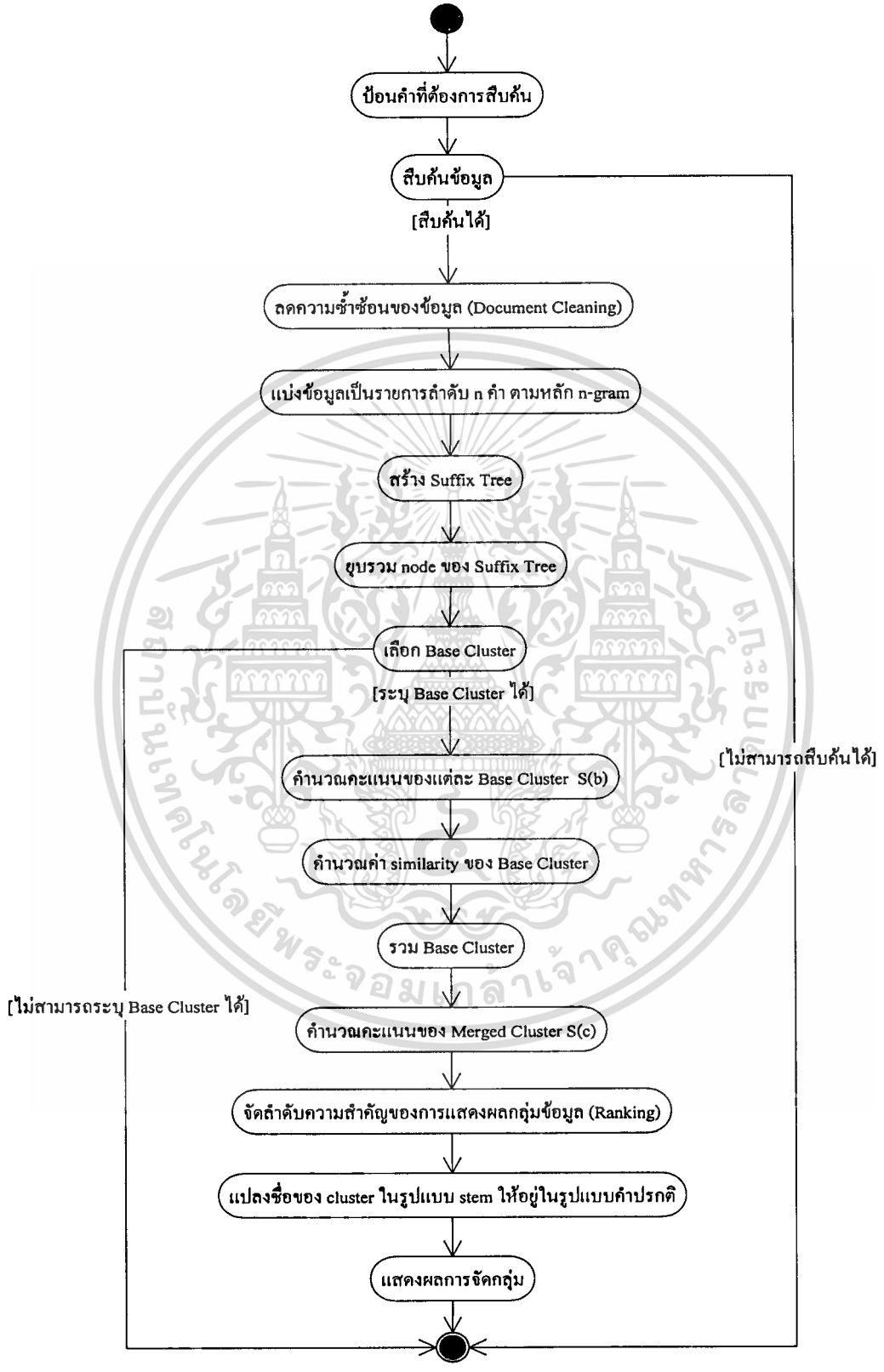
Primary scenario	1. กำจัดเครื่องหมาย,คำ, สัญลักษณ์และตัวอักษรที่ไม่เป็นประโยชน์ 2. กำจัด Stopwords 3. ลดรูปคำ (Stem)
Alternatives	-

ตารางที่ 3.5 รายละเอียดของ Use Case "สืบค้นข้อมูล"

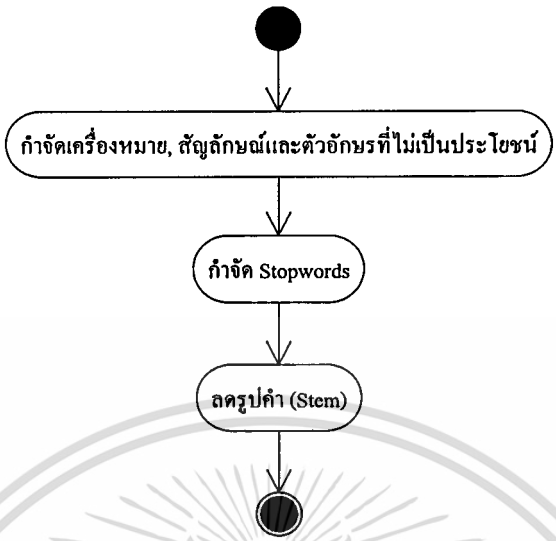
Use Case	สืบค้นข้อมูล
Brief Description	ทำการสืบค้นข้อมูลจากคำ (keyword) ที่ผู้ใช้ต้องการสืบค้นจาก Search Engine
Actor	Search Engine
Pre-condition	-
Post-condition	ได้รับผลการสืบค้นจาก Search Engine
Primary scenario	1. ระบบเชื่อมต่อกับ Search Engine 2. ระบบทำการสืบค้นข้อมูลจาก Search Engine 3. บันทึกผลการสืบค้น
Alternatives	1a. ไม่สามารถเชื่อมต่อได้ : จบโปรแกรม 2a. ไม่มีผลสืบค้น : จบโปรแกรม

3.2 Activity Diagram

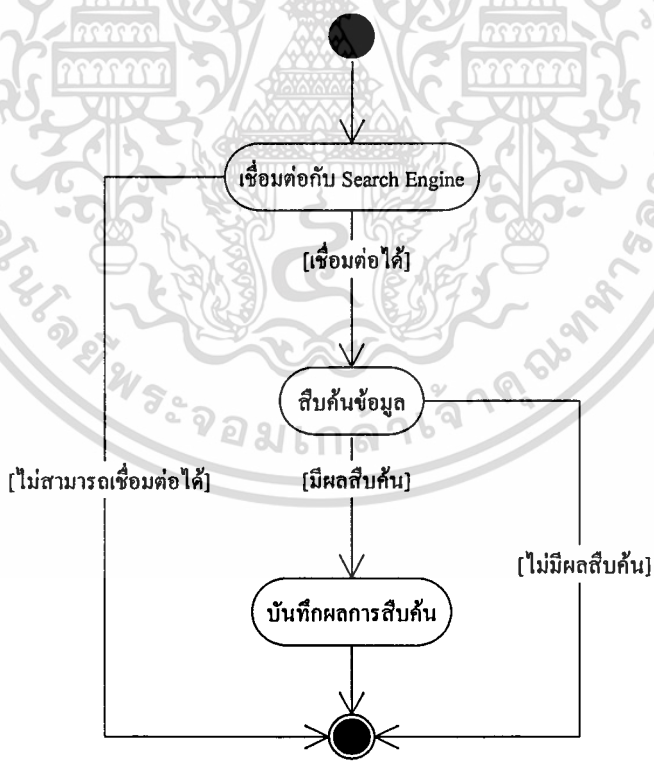
ในส่วนนี้จะเป็นการอธิบายถึงกระบวนการทำงานของ Use Case ต่าง ๆได้แก่ Use Case "จัดกลุ่มผลการสืบค้น" , "ลดความซ้ำซ้อนของข้อมูล" และ "สืบค้นข้อมูล" ด้วย Activity Diagram ดังแสดงในรูปที่ 3.2, รูปที่ 3.3 และ รูปที่ 3.4



รูปที่3.2 Activity Diagram ของ Use Case "จัดกลุ่มผลการสืบค้น"

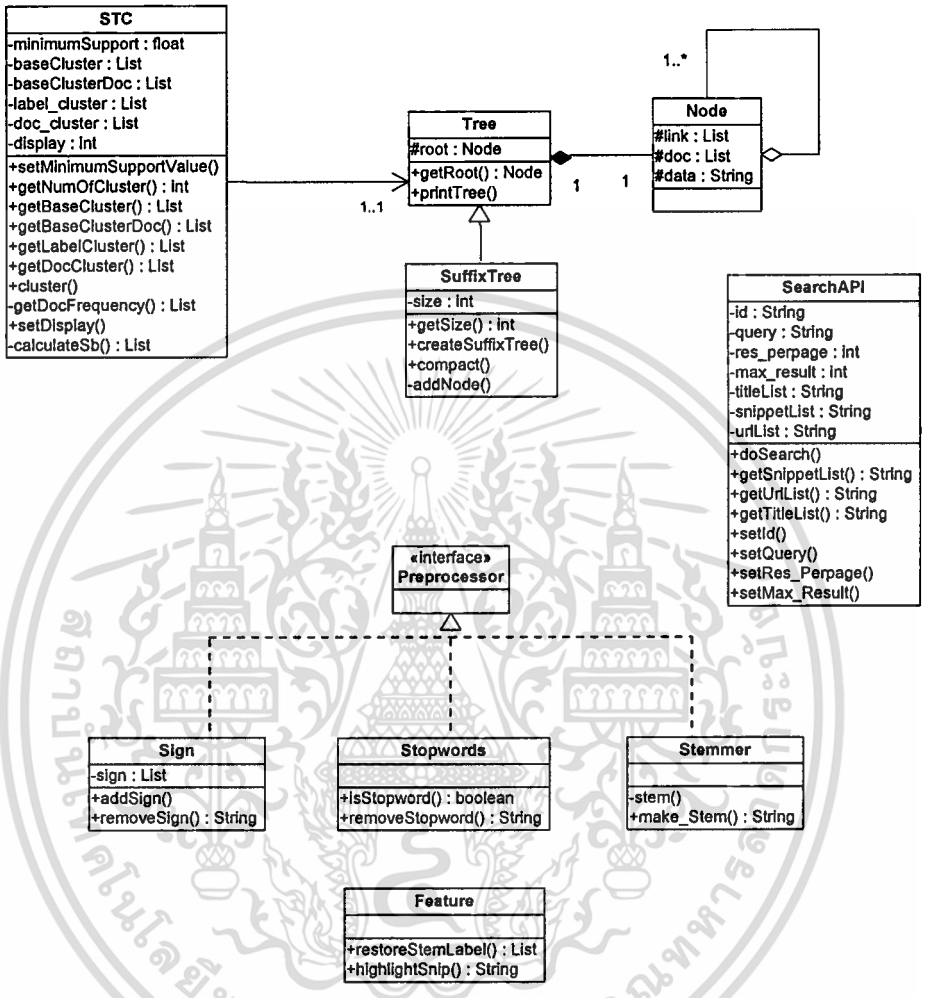


รูปที่3.3 Activity Diagram ของ Use Case "ลดความซ้ำซ้อนของข้อมูล"



รูปที่3.4 Activity Diagram ของ Use Case "สืบค้นข้อมูล"

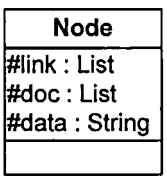
3.3 Class Diagram



รูปที่3.5 Class Diagram ของระบบ

จากรูปที่3.5 แสดงความสัมพันธ์ของ Class ต่างๆ ภายในระบบการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering โดยภายในแต่ละ Class จะประกอบไปด้วย Attribute และ Method ต่าง ๆ ดังต่อไปนี้

- **Class Node**



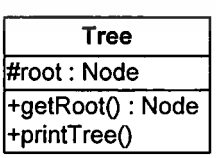
รูปที่3.6 Class Node

จากรูปที่3.6 แสดงรายละเอียดของ Class Node โดยภายใน Class Node จะประกอบไปด้วย Attribute และ Method ดังแสดงในตารางที่ 3.6

ตารางที่ 3.6 รายละเอียด Attribute ของ Class Node

ชื่อ	ชนิดข้อมูล	คำอธิบาย
link	java.util.*List* <<ArrayList>>	ใช้เก็บตำแหน่ง Address ของ node ที่มาเชื่อมต่อกับตัวของมันเอง
doc	java.util.*List* <<ArrayList>>	เก็บหมายเลขเอกสารที่ phrase นับตั้งแต่ root จนถึง node นั้น ๆ ปรากฏอยู่
data	java.lang.String	เก็บข้อมูลของ node (label ของ node)

- **Class Tree**



รูปที่ 3.7 Class Tree

จากรูปที่ 3.7 แสดงรายละเอียดของ Class Tree ซึ่งจะทำหน้าที่เป็น Class ต้นแบบ ของ Class Suffix Tree โดยภายใน Class Tree จะประกอบไปด้วย Attribute และ Method ดังแสดงใน ตารางที่ 3.7 และ ตารางที่ 3.8

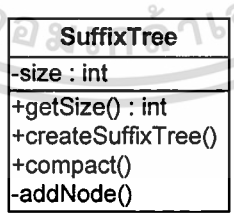
ตารางที่ 3.7 รายละเอียด Attribute ของ Class Tree

ชื่อ	ชนิดข้อมูล	คำอธิบาย
root	Node	node ที่เป็น node แรกของ Tree(root)

ตารางที่ 3.8 รายละเอียด Method ของ Class Tree

ชื่อ	ชนิดของข้อมูลที่ถูกส่งกลับ	คำอธิบาย
getRoot()	Node	ใช้สำหรับดึงค่าข้อมูลที่เป็น node แรกของ Tree (root)
printTree()	-	พิมพ์ node ทั้งหมดของ Tree ให้ปรากฏบนหน้าจอ System ของระบบโดยจะพิมพ์ออกมาโดยเป็นไปตามลำดับการท่อง Tree แบบ pre-order

- **Class Suffix Tree**



รูปที่ 3.8 Class Suffix Tree

จากรูปที่ 3.8 แสดงรายละเอียดของ Class Suffix Tree โดยภายใน Class Suffix Tree จะประกอบไปด้วย Attribute และ Method ดังแสดงในตารางที่ 3.9 และ ตารางที่ 3.10

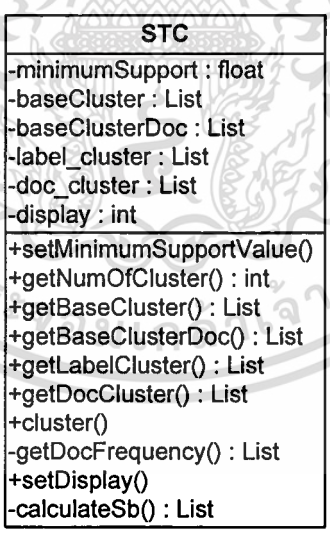
ตารางที่ 3.9 รายละเอียด Attribute ของ Class Suffix Tree

ชื่อ	ชนิดข้อมูล	คำอธิบาย
size	int <<จำนวนเต็ม>>	จำนวน node ทั้งหมดของ Suffix Tree

ตารางที่ 3.10 รายละเอียด Method ของ Class Suffix Tree

ชื่อ	ชนิดของข้อมูลที่ถูกส่งกลับ	คำอธิบาย
getSize()	int <<จำนวนเต็ม>>	ใช้ตรวจสอบจำนวน node ทั้งหมด ของ Suffix Tree
createSuffixTree()	-	สร้าง Suffix Tree
compact()	-	ใช้สำหรับยุบรวม node ของ Suffix Tree
addNode()	-	ใช้สำหรับเพิ่ม node เข้าไปใน Suffix Tree

- **Class STC**



รูปที่ 3.9 Class STC

จากรูปที่ 3.9 แสดงรายละเอียดของ Class STC ซึ่ง Class STC จะเป็น Class ที่กระทำกระบวนการต่าง ๆ ตามอัลกอริทึม Suffix Tree Clustering โดยภายใน Class STC จะประกอบไปด้วย Attribute และ Method ดังแสดงในตารางที่ 3.11 และ ตารางที่ 3.12

ตารางที่ 3.11 รายละเอียด Attribute ของ Class STC

ชื่อ	ชนิดข้อมูล	คำอธิบาย
minimumSupport	float <<เลขทศนิยม>>	ค่า minimum support ของการหาค่า similarity ในขั้นตอนการทำ Combining Base Clusters
baseCluster	java.util.*List* <<ArrayList>>	รายการของ base cluster ทั้งหมดที่เกิดขึ้นภายหลังจากการระบุ base cluster แล้ว (Identifying Base Cluster)
baseClusterDoc	java.util.*List* <<ArrayList>>	รายการของกลุ่มของเอกสารที่ปรากฏอยู่ในแต่ละ base cluster
label_cluster	java.util.*List* <<ArrayList>>	รายชื่อของกลุ่มที่ผ่านการจัดกลุ่มแล้ว
doc_cluster	java.util.*List* <<ArrayList>>	เอกสารที่ปรากฏอยู่ในแต่ละกลุ่มหลังจากที่ผ่านการจัดกลุ่มแล้ว
display	int <<จำนวนเต็ม>>	จำนวนกลุ่มของเอกสารที่ต้องการแสดงผล

ตารางที่ 3.12 รายละเอียด Method ของ Class STC

ชื่อ	ชนิดของข้อมูลที่ถูกส่งกลับ	คำอธิบาย
setMinimumSupport Value()	-	ตั้งค่า minimum support ซึ่งจะมีค่าได้ตั้งแต่ 0.5-0.8
getNumOfCluster()	int <<จำนวนเต็ม>>	ใช้เมื่อต้องการทราบถึงจำนวนของ กลุ่ม (cluster) ทั้งหมดที่เกิดขึ้นหลังจากการจัดกลุ่ม
getBaseCluster()	java.util.*List* <<ArrayList>>	ใช้ตรวจสอบรายการของ base cluster ทั้งหมดที่เกิดขึ้นภายหลังจากการเลือก base cluster แล้ว
getBaseClusterDoc()	java.util.*List* <<ArrayList>>	ใช้ตรวจสอบรายการของกลุ่มของเอกสารที่ปรากฏอยู่ในแต่ละ base cluster

ตารางที่ 3.12 (ต่อ)

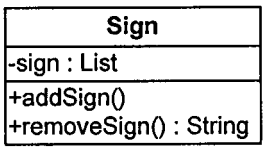
getLabelCluster()	java.util.*List* <<ArrayList>>	ใช้สำหรับอ่านค่ารายการของกลุ่ม (cluster)ที่เกิดขึ้นหลังจากการจัดกลุ่ม
getDocCluster()	java.util.*List* <<ArrayList>>	ใช้สำหรับอ่านรายการกลุ่มของเอกสารที่ปรากฏในแต่ละกลุ่ม(cluster) ภายหลังที่ผ่านการจัดกลุ่มแล้ว
cluster()	-	จัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering
getDocFrequency()	java.util.*List* <<ArrayList>>	ใช้สำหรับตรวจสอบจำนวนเอกสารที่ปรากฏใน subtree ของ node ใด ๆ
getSb()	java.util.*List* <<ArrayList>>	ใช้ตรวจสอบรายการของค่า S(B) ที่เกิดขึ้นในแต่ละ Base Cluster
setDisplay()	-	ใช้ตั้งค่าจำนวนกลุ่มทั้งหมดที่ต้องการแสดงผล

- **Interface Preprocessor**

รูปที่ 3.10 Interface Preprocessor

จากรูปที่ 3.10 แสดง Interface Preprocessor ซึ่งทำหน้าที่เป็น Marker Interface มีหน้าที่สำหรับระบุว่า Class ใด ๆ ที่ implements Interface นี้จะมีหน้าที่ในการทำ Preprocessing (Document Cleaning) และยังมีประโยชน์สำหรับการรองรับการเพิ่มขีดความสามารถของการทำ Preprocessing ในอนาคต

- **Class Sign**



รูปที่ 3.11 Class Sign

จากรูปที่ 3.11 แสดงรายละเอียดของ Class Sign ซึ่ง Class Sign เป็น Class ที่ทำหน้าที่ในการทำขั้นตอนหนึ่งของ Document Cleaning คือการกำจัดเครื่องหมาย, คำ, สัญลักษณ์, ตัวอักษร ที่ไม่เป็นประโยชน์ต่อการจัดกลุ่ม โดยภายใน Class Sign จะประกอบไปด้วย Attribute และ Method ดังแสดงในตารางที่ 3.13 และ ตาราง 3.14

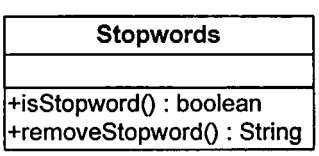
ตารางที่ 3.13 รายละเอียด Attribute ของ Class Sign

ชื่อ	ชนิดข้อมูล	คำอธิบาย
sign	java.util.*List* <<ArrayList>>	เก็บเครื่องหมาย, คำ ทั้งหมดที่ต้องการกำจัดออกของ Class Sign

ตารางที่ 3.14 รายละเอียด Method ของ Class Sign

ชื่อ	ชนิดของข้อมูลที่ถูกส่งกลับ	คำอธิบาย
addSign()	-	เพิ่มเครื่องหมาย, คำ ที่ต้องการกำจัดออกเข้าไปในรายการเครื่องหมายที่มีอยู่เดิม
removeSign()	java.lang.String	กำจัดเครื่องหมาย, คำ, ตัวอักษรที่ไม่มีความหมายและไม่เป็นประโยชน์

- **Class Stopwords**



รูปที่ 3.12 Class Stopwords

จากรูปที่ 3.12 แสดงรายละเอียดของ Class Stopwords ซึ่ง Class Stopwords เป็น Class ที่ทำหน้าที่ในการทำขั้นตอนหนึ่งของ Document Cleaning คือการกำจัด Stopwords โดยรายการของคำที่เป็น Stopwords จะแสดงในตารางที่ 3.15 และภายใน Class Stopwords จะประกอบไปด้วย Method ดังแสดงในตารางที่ 3.16

ตารางที่ 3.15 รายการคำที่เป็น Stopwords

A	a, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, all, allow, allows, almost, alone, along, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appreciate, appropriate, are, around, as, aside, ask, asking, associated, at, available, away, awfully
B	b, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, believe, below, beside, besides, best, better, between, beyond, both, brief, but, by
C	c, came, can, cannot, cant, cause, causes, certain, certainly, changes, clearly, co, com, come, comes, concerning, consequently, consider, considering, contain, containing, contains, corresponding, could, course, currently
D	d, definitely, described, despite, did, different, do, does, doing, done, down, downwards, during
E	e, each, edu, eg, eight, either, else, elsewhere, enough, entirely, especially, et, etc, even, ever, every, everybody, everyone, everything, everywhere, ex, exactly, example, except
F	f, far, few, fifth, first, five, followed, following, follows, for, former, formerly, forth, four, from, further, furthermore
G	g, get, gets, getting, given, gives, go, goes, going, gone, got, gotten, greetings
H	h, had, happens, hardly, has, have, having, he, hello, help, hence, her, here, hereafter, hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hopefully, how, howbeit, however
I	i, ie, if, ignored, immediate, in, inasmuch, inc, indeed, indicate, indicated, indicates, inner, insofar, instead, into, inward, is, it, its, itself

ตารางที่ 3.15 (ต่อ)

J	j, just
K	k, keep, keeps, kept, know, knows, known
H	h, had, happens, hardly, has, have, having, he, hello, help, hence, her, here, hereafter, hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hopefully, how, howbeit, however
I	i, ie, if, ignored, immediate, in, inasmuch, inc, indeed, indicate, indicated, indicates, inner, insofar, instead, into, inward, is, it, its, itself
J	j, just
K	k, keep, keeps, kept, know, knows, known
L	l, last, lately, later, latter, latterly, least, less, lest, let, like, liked, likely, little, ll, look, looking, looks, ltd
M	m, mainly, many, may, maybe, me, mean, meanwhile, merely, might, more, moreover, most, mostly, much, must, my, myself
N	n, name, namely, nd, near, nearly, necessary, need, needs, neither, never, nevertheless, new, next, nine, no, nobody, non, none, noone, nor, normally, not, nothing, novel, now, nowhere
O	o, obviously, of, off, often, oh, ok, okay, old, on, once, one, ones, only, onto, or, other, others, otherwise, ought, our, ours, ourselves, out, outside, over, overall, own
P	p, particular, particularly, per, perhaps, placed, please, plus, possible, presumably, probably, provides
Q	q, que, quite, qv
R	r, rather, rd, re, really, reasonably, regarding, regardless, regards, relatively, respectively, right
S	s, said, same, saw, say, saying, says, second, secondly, see, seeing, seem, seemed, seeming, seems, seen, self, selves, sensible, sent, serious, seriously, seven, several, shall, she, should, since, six, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, soon, sorry, specified, specify, specifying, still, sub, such, sup, sure

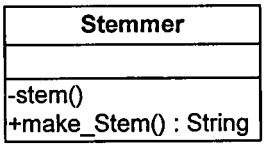
ตารางที่ 3.15 (ต่อ)

T	t, take, taken, tell, tends, th, than, thank, thanks, thanx, that, thats, the, their, theirs, them, themselves, then, thence, there, thereafter, thereby, therefore, therein, theres, thereupon, these, they, think, third, this, thorough, thoroughly, those, though, three, through, throughout, thru, thus, to, together, too, took, toward, towards, tried, tries, truly, try, trying, twice, two
U	u, un, under, unfortunately, unless, unlikely, until, unto, up, upon, us, use, used, useful, uses, using, usually, uucp
V	v, value, various, ve, very, via, viz, vs
W	w, want, wants, was, way, we, welcome, well, went, were, what, whatever, when, whence, whenever, where, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, whoever, whole, whom, whose, why, will, willing, wish, with, within, without, wonder, would, would
X	x
Y	y, yes, yet, you, your, yours, yourself, yourselves
Z	z, zero

ตารางที่ 3.16 รายละเอียด Method ของ Class Stopwords

ชื่อ	ชนิดของข้อมูลที่ถูกส่งกลับ	คำอธิบาย
isStopword()	boolean	ตรวจสอบคำว่าคำ ๆ นั้นเป็น stopwords หรือไม่
removeStopword()	java.lang.String	กำจัด stopwords ออกจากข้อมูลก่อนที่จะนำไปจัดกลุ่มต่อไป

- **Class Stemmer**



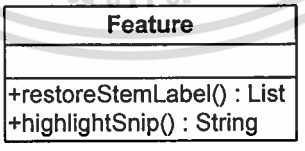
รูปที่ 3.13 Class Stemmer

จากรูปที่ 3.13 แสดงรายละเอียดของ Class Stemmer ซึ่ง Class Stemmer เป็น Class ที่ทำหน้าที่ในการทำขั้นตอนหนึ่งของ Document Cleaning คือการลดรูปคำตามหลักการทำงานของ Porter Algorithm โดยภายใน Class Stemmer จะประกอบไปด้วย Method ดังแสดงในตารางที่ 3.17

ตารางที่ 3.17 รายละเอียด Method ของ Class Stemmer

ชื่อ	ชนิดของข้อมูลที่ถูกส่งกลับ	คำอธิบาย
stem()	-	ดำเนินการทำงานตามหลัก Porter Algorithm
make_Stem()	java.lang.String	ลดรูปคำตามหลักของ Porter Algorithm

- **Class Feature**



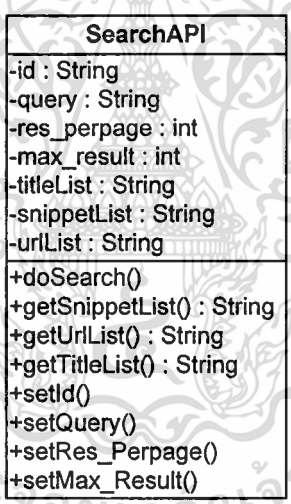
รูปที่ 3.14 Class Feature

จากรูปที่3.14 แสดงรายละเอียดของ Class Feature ซึ่ง Calss Feature จะเป็น Class ที่รวบรวมคุณสมบัติพิเศษต่าง ๆ ของระบบเอาไว้ โดยภายใน Class Feature จะประกอบไปด้วย Method ดังแสดงในตารางที่ 3.18

ตารางที่ 3.18 รายละเอียด Method ของ Class Feature

ชื่อ	ชนิดของข้อมูลที่ถูกส่งกลับ	คำอธิบาย
restoreStemLabel()	java.util.*List* <<ArrayList>>	แปลงชื่อของกลุ่มซึ่งเดิมเป็นคำที่เป็นถูกลดรูปอยู่ให้เป็นคำเต็ม ที่สามารถสื่อความหมายได้ดีกว่าคำที่ถูกลดรูป
highlightSnip()	java.lang.String	เน้นคำที่เป็นชื่อของกลุ่มในแต่ละเอกสารภายในกลุ่มให้เป็นตัวอักษรเข้มในขณะที่แสดงผลทาง web browser

- **Class SearchAPI**



รูปที่ 3.15 Class SearchAPI

จากรูปที่ 3.15 แสดงรายละเอียดของ Class SearchAPI ซึ่ง Calss SearchAPI จะเป็น Class ที่ทำหน้าที่หลักในการเชื่อมต่อและสืบค้นข้อมูลจาก Search Engine โดยภายใน Class SearchAPI จะประกอบไปด้วย Attribute และ Method ดังแสดงในตารางที่ 3.19 และ ตารางที่ 3.20

ตารางที่ 3.19 รายละเอียด Attribute ของ Class SearchAPI

ชื่อ	ชนิดข้อมูล	คำอธิบาย
id	java.lang.String	ค่า user id ของการเข้าใช้งาน search engine
query	java.lang.String	คำที่ต้องการสืบค้น
res_perpage	int <<จำนวนเต็ม>>	จำนวนผลสืบค้นที่ดึงมาจาก Search Engine ในแต่ละครั้ง (ไม่เกินครั้งละ 50 เอกสาร)
max_result	int <<จำนวนเต็ม>>	จำนวนผลสืบค้นทั้งหมดที่ต้องการดึงมาจาก Search Engine
titleList	java.lang.String	รายการ title ทั้งมดของผลสืบค้น
snippetList	java.lang.String	รายการ snippet ทั้งหมดของผลสืบค้น
urlList	java.lang.String	รายการ url ทั้งหมดของผลสืบค้น

ตารางที่ 3.20 รายละเอียด Method ของ Class SearchAPI

ชื่อ	ชนิดของข้อมูลที่ถูกส่งกลับ	คำอธิบาย
doSearch()	-	สืบค้นข้อมูลจาก Search Engine
getSnippetList()	java.lang.String	ดึงค่ารายการของ snippet ทั้งหมด
getUrlList()	java.lang.String	ดึงค่ารายการของ url ทั้งหมด
getTitleList()	java.lang.String	ดึงค่าของ title ทั้งหมด
setId()	-	ตั้งค่า user id ในการเข้าใช้ search engine
setQuery()	-	ตั้งค่าคำที่ต้องการสืบค้น
setRes_Perpage()	-	ตั้งค่าจำนวนผลสืบค้นที่ต้องการดึงมาจาก Search Engine ในแต่ละครั้ง
setMax_Result()	-	ตั้งค่าจำนวนผลสืบค้นทั้งหมดที่ต้องการดึงมาจาก Search Engine

บทที่ 4
การพัฒนาระบบ

4.1 เครื่องมือที่ใช้ในการพัฒนาระบบ

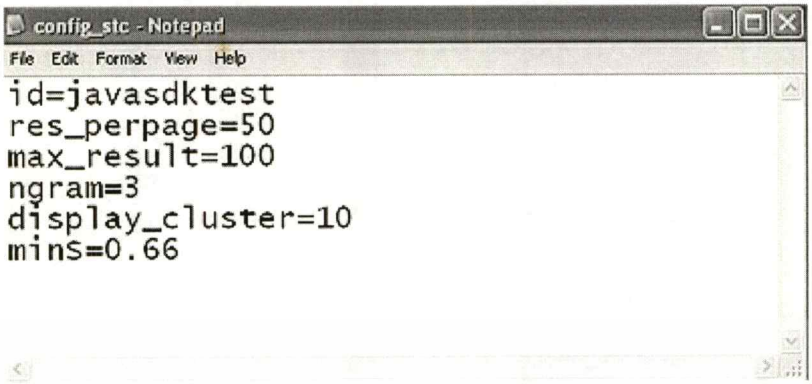
- ภาษาที่ใช้ในการพัฒนา : Java (J2SE 5.0, J2EE 1.4), JSP
- เครื่องมือที่ช่วยในการเขียน Code Program : Netbean IDE Version 4.1
- เว็บเซิร์ฟเวอร์ : Apache Tomcat Version 5.5
- ระบบปฏิบัติการ : Window XP
- โปรแกรมเว็บเบราว์เซอร์ : Internet Explorer 6.0
- Java Library : yahoo_search _sdk-1.1.0

4.2 การเชื่อมต่อกับ Search Engine

ในโครงการพัฒนาระบบงานนี้ได้มีการเชื่อมต่อไปยัง Search Engine เพื่อที่จะสืบค้นข้อมูลและนำผลสืบค้นข้อมูล (snippets) มาทำการจัดกลุ่ม โดย Search Engine ที่ใช้ในโครงงานพัฒนาระบบนี้คือ Yahoo Search Engine ซึ่งเป็นบริการ web service ที่พัฒนาขึ้นมาเพื่อให้นักพัฒนาสามารถใช้งาน Yahoo Search Engine ได้ ก่อนที่จะใช้งาน Yahoo Search Engine จะต้องทำการลงทะเบียนขอ Application ID และ download Java Library : yahoo_search _sdk-1.1.0 จาก http://developer.yahoo.net/search/index.html จึงสามารถใช้งาน Yahoo Search Engine ได้

4.3 การตั้งค่าการทำงานของระบบ

ในการใช้งานระบบการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering นั้น ระบบจะสามารถตั้งค่าการทำงานต่าง ๆ ของระบบได้จากการเปลี่ยนค่าข้อมูลในไฟล์ config_stc.properties ดังแสดงในรูปที่ 4.1 ซึ่งไฟล์นี้จะถูกเก็บอยู่ใน working directory ของเว็บเซิร์ฟเวอร์ โดยภายในไฟล์ config_stc.properties นั้นจะประกอบไปด้วยตัวแปรต่าง ๆ ที่สามารถตั้งค่าได้ดังนี้

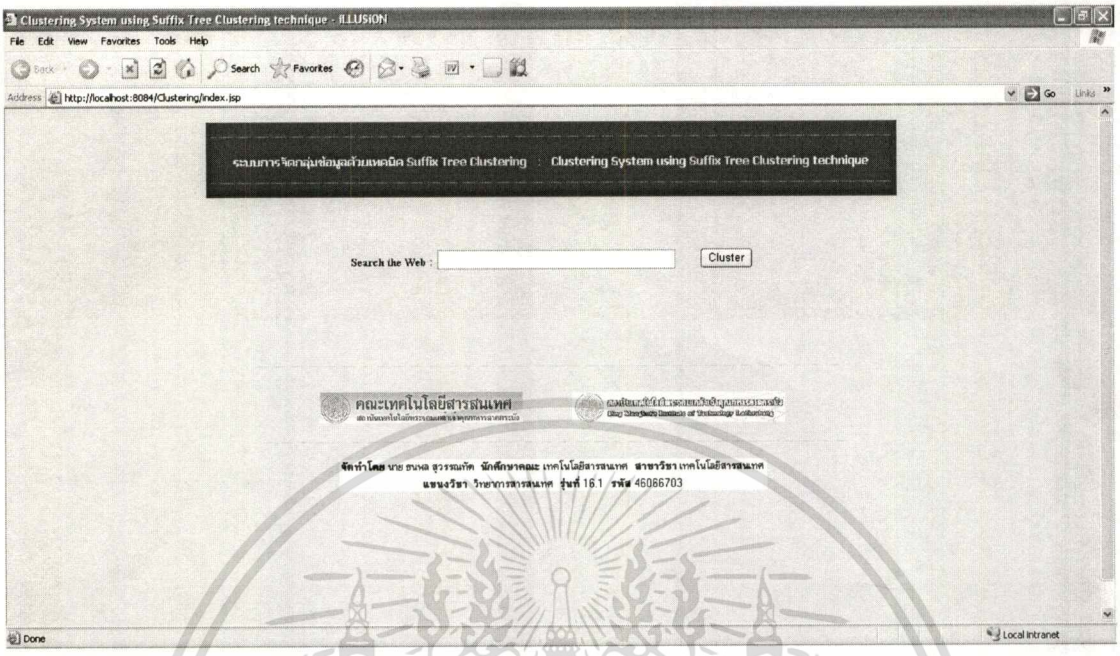


รูปที่ 4.1 ไฟล์ config_stc.properties

- id - ใช้สำหรับตั้งค่า Application ID ที่ใช้ในการเชื่อมต่อไปยัง Search Engine
- res_perpage - ใช้สำหรับตั้งค่าจำนวนผลสืบค้นที่ดึงมาจาก Search Engine ในแต่ละครั้ง โดยจะสามารถดึงได้ครั้งละมากที่สุดไม่เกิน 50 เอกสาร
- max_result - ใช้สำหรับตั้งค่าของจำนวนผลสืบค้นทั้งหมดที่ต้องการดึงมาจาก Search Engine
- ngram - ใช้ระบุค่าของ n-gram ที่ใช้ในระบบ
- display_cluster - ใช้ระบุจำนวนกลุ่มของเอกสารที่ต้องการแสดงผล
- minS - ใช้กำหนดค่าของ minimum support ที่ใช้ในการหา simlarity ในขั้นตอนการทำ Combining Base Clusters

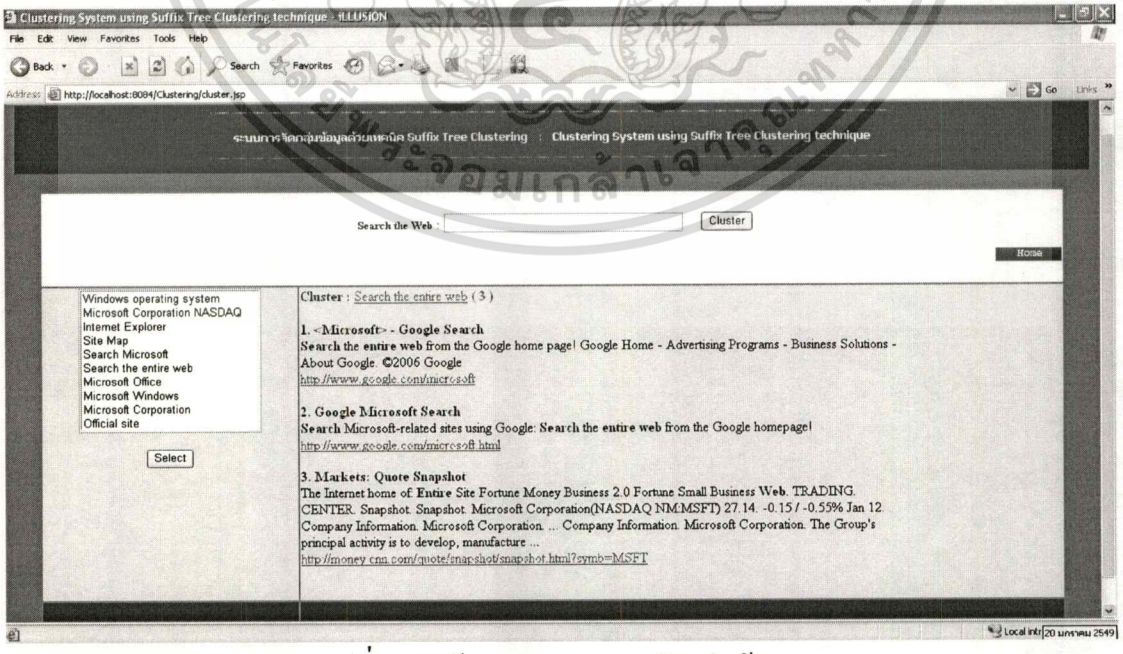
4.4 การทำงานของระบบ

ระบบการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering มีการทำงานและหน้าจอต่าง ๆ ดังต่อไปนี้



รูปที่ 4.2 หน้าจอหลักของระบบ

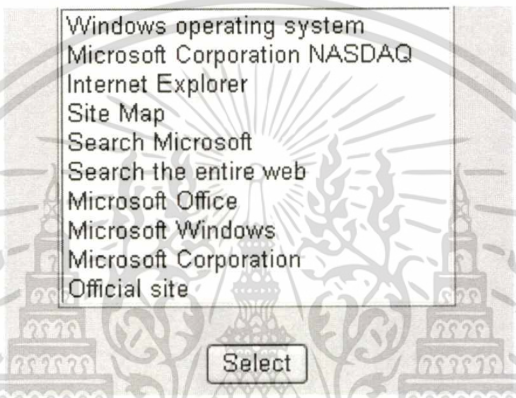
จากรูปที่ 4.2 แสดงหน้าจอหลักของระบบ ผู้ใช้จะต้องทำการป้อนข้อมูลที่เป็นคำที่ต้องการสืบค้น เพื่อที่ระบบจะได้ทำการสืบค้นข้อมูลที่ต้องการและจะทำการจัดกลุ่มผลสืบค้นต่อไป



รูปที่ 4.3 หน้าจอแสดงผลการจัดกลุ่มข้อมูล

จากรูปที่ 4.3 แสดงหน้าจอการแสดงผลการจัดกลุ่มข้อมูลที่ได้จากผลการสืบค้นจากอินเทอร์เน็ต โดยภายในหน้าจอจะประกอบไปด้วยส่วนประกอบที่สำคัญ 3 ส่วนคือ

1. ส่วนแสดงผลของการจัดกลุ่ม ดังแสดงในรูปที่ 4.4 จะทำหน้าที่ในการแสดงผลของการจัดกลุ่มข้อมูลเป็นชื่อของกลุ่ม โดยผู้ใช้สามารถเลือกดูเอกสารที่มีอยู่ภายในกลุ่มได้โดยการเลือกชื่อกลุ่มและกดปุ่ม “select” ระบบจะทำการแสดงเอกสารทั้งหมดที่อยู่ภายในกลุ่มที่ถูกเลือก



รูปที่ 4.4 List box สำหรับเลือกกลุ่มข้อมูล

2. ส่วนนี้ทำหน้าที่รับคำที่ต้องการสืบค้น (keyword) จากผู้ใช้ ดังรูปที่ 4.5

รูปที่ 4.5 ส่วนของการสืบค้นข้อมูล

3. ส่วนนี้จะทำหน้าที่ในการแสดงผลเอกสารทั้งหมดที่อยู่ภายในกลุ่มที่ถูกเลือก ดังแสดงในรูปที่ 4.6

Cluster : Windows operating system (5)

1. Microsoft Windows Update
update site for the Microsoft **Windows operating system**, keeping users' computers updated with the latest security patches and features. Includes **Windows** updates, Internet Explorer updates, and other software.
http://windowsupdate.microsoft.com/

2. Microsoft Windows Family
official site for the Microsoft family of **Windows operating systems**. Learn more about **Windows** XP, including how-tos and tips.
http://www.windows.com/

3. Microsoft Windows Update
update site for the Microsoft **Windows operating system**, keeping users' computers updated with the latest security patches and features. Includes **Windows** updates, Internet Explorer updates, and other software.
http://www.windowsupdate.com/

4. Tech Blogs on ZDNet | blogs.ZDNet.com
... Microsoft has issued a correction to the statements that were given to me during our last conference call ... at Macworld Expo. Wait for Microsoft WMF patch, no thanks! ...
http://blogs.zdnet.com/

5. Microsoft Connect
Microsoft.com Home. Search Microsoft.com for: Connect Home. Make Your Input Count. Welcome to Microsoft Connect, the new product development collaboration site at Microsoft.
http://connect.microsoft.com/

รูปที่ 4.6 ส่วนของการแสดงผลเอกสารในกลุ่มที่ถูกเลือก

บทที่ 5
สรุปผล และข้อเสนอแนะ

จากการศึกษาการจัดกลุ่มข้อมูลด้วยอัลกอริทึม Suffix Tree Clustering เพื่อที่จะสร้างโครงงานพัฒนาระบบนี้ สามารถสรุปผลการดำเนินงานและสรุปผลการทดลองรวมทั้งข้อเสนอแนะ ดังต่อไปนี้

5.1 สรุปผลการดำเนินงานและการทดลอง

1. จากการพัฒนาระบบการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering ระบบสามารถจำแนกข้อมูลออกเป็นกลุ่มได้ ทำให้สามารถเข้าถึงข้อมูลที่ต้องการคือผลการสืบค้นจากอินเตอร์เน็ตได้สะดวกและรวดเร็วมากขึ้น
2. เอกสารที่อยู่ภายในกลุ่มแต่ละกลุ่มมีการ overlap กัน คือเอกสารหนึ่ง ๆ สามารถปรากฏได้ในหลาย ๆ กลุ่ม
3. อัลกอริทึม Suffix Tree Clustering ใช้เพียงพื้นฐานเดียวในการคัดเลือกป้ายชื่อกลุ่ม (Cluster Label) คือยึดค่าความถี่ของการใช้วลีพื้นฐาน (common phrase) ที่แต่ละเอกสารใช้ร่วมกัน บางครั้งวลีพื้นฐานที่แต่ละเอกสารใช้ร่วมกันมากๆนั้น อาจจะไม่มีอำนาจจำแนกได้ดีพอ
4. อัตราการได้รับผลการสืบค้นของผู้สืบค้น(coverage) ระบบอาจไม่สามารถแสดงผลการจัดกลุ่มให้กับผู้สืบค้นได้ครบทุกเอกสาร เพราะการจัดกลุ่มโดยใช้ เทคนิค Suffix Tree Clustering จะได้รับผลการจัดกลุ่มจำนวนมาก ทำให้ระบบจะต้องเลือกแสดงผลเฉพาะกลุ่มที่มีคะแนนในระดับสูงอยู่ในระดับต้นๆ ทำให้บางเอกสารที่ไม่ได้ประกอบอยู่ในกลุ่มดังกล่าวไม่ได้ถูกแสดงผลให้กับผู้สืบค้น
5. ผลการสืบค้นจาก Search Engine บางรายการไม่สมบูรณ์ คือจะไม่มีคำอธิบาย snippet ทำให้ผลการสืบค้นรายการนั้น ๆ ไม่สามารถนำไปจัดกลุ่มได้

5.2 ข้อเสนอแนะ

จากการที่ได้ทดลองใช้งานโปรแกรมและศึกษาการทำงานของระบบการจัดกลุ่มข้อมูลด้วยเทคนิค Suffix Tree Clustering พบว่าระบบยังมีข้อจำกัดในบางเรื่อง ซึ่งอาจเสนอแนะแนวทางในการพัฒนาและข้อจำกัดต่าง ๆ เพื่อนำไปใช้พัฒนาต่อไปให้มีประสิทธิภาพดีขึ้นดังต่อไปนี้

1. ระบบงานนี้เหมาะสำหรับใช้จัดกลุ่มกับข้อมูลผลสืบค้นที่เป็นภาษาอังกฤษเท่านั้น เพราะขั้นตอนในการลดความซ้ำซ้อนของเอกสาร(Document Cleaning) นั้นใช้ Porter Algorithm ซึ่งจะใช้งานได้อย่างมีประสิทธิภาพเฉพาะกับภาษาอังกฤษเท่านั้น
2. ความถูกต้องและประสิทธิภาพของการจัดกลุ่มข้อมูลนั้นขึ้นอยู่กับมาตรฐานของข้อมูล(input)ที่นำมาจัดกลุ่มด้วย ดังนั้นควรที่จะพัฒนาในส่วนของ Document Cleaning ให้มีประสิทธิภาพมากขึ้น
3. เนื่องจากขั้นตอนในการดึงข้อมูลที่เป็นผลสืบค้นจาก Search Engine เพื่อนำมาทำการจัดกลุ่มนั้น Yahoo Search Engine มีข้อจำกัดในการดึงผลสืบค้นได้มากที่สุดครั้งละ 50 เอกสารเท่านั้น ถ้าต้องการผลสืบค้นจำนวนมากจะต้องทำการดึงผลสืบค้นหลายครั้ง ซึ่งจะทำให้ใช้เวลาในการดึงผลสืบค้นทั้งหมดมาก ทำให้ใช้เวลาโดยรวมในการประมวลผลมาก ดังนั้นหากต้องการให้เวลาในการประมวลผลโดยรวมรวดเร็วขึ้นจึงควรเชื่อมต่อกับ Search Engine ที่มีประสิทธิภาพมากกว่านี้

บรรณานุกรม

ภาณุพงศ์ ชวะวิทย์. 2547. "ระบบจัดกลุ่มผลการค้นหาข้อมูลของเสิร์ชเอ็นจิ้นในเครือข่ายใยแมงมุม." วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

วิริช ลีลาภัทร และ พรฤดี เนติโสภากุล. 2548. "สถาปัตยกรรมเว็บเสิร์ชเอ็นจินที่สามารถปรับตัวตามผู้ค้นหา." เทคโนโลยีสารสนเทศ. 1(1): 65-71.

Chambers, N. et.al. 2004. **Approaches for Automatically Tagging Affect.** [Online]. Available: http://www.ihmc.us/~nchambers/publish/aaai2004-affect.pdf.

Ferragina, P. and Gulli, A. 2005. **A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering.** [Online]. Available: http://www.di.unipi.it/~gulli/papers/i12_ferragina_gulli.pdf

Osinki, S. and Stefanowski, J. 2003. "An Algorithm For Clustering Of Web Search Results." **Degree of Master Thesis**, Poznan University of Technology.

Porter, M. 1980. **The Porter Stemming Algorithm** . [Online]. Available : http://www.tartarus.org/~martin/PorterStemmer/index-old.html

Zamir, O. and Etzioni O,. 1998. **Web Document Clustering: A Feasibility Demonstration.** [Online]. Available: ttp://citeseer.ist.psu.edu/cache/papers/cs/18/http:zSzzSzzhadum.cs.washington.eduzSzzamirzSzsigir98.pdf/zamir98web.pdf.

Zeng, H. et.al. 2004. **Learning to Cluster Web Search Results.** [Online]. Available: http://research.microsoft.com/users/hjzeng/p230-zeng.pdf.

ประวัติผู้เขียนโครงงาน

ชื่อผู้จัดทำโครงงาน นายธนพล สุวรรณทัต

วันเดือนปีเกิด 24 มีนาคม 2524

สถานที่เกิด จังหวัด กรุงเทพมหานคร

ประวัติการศึกษา

ประถมศึกษา โรงเรียน จันทรวิทยา

มัธยมศึกษา โรงเรียน สวนกุหลาบวิทยาลัย

อุดมศึกษา สาขา วิทยาการคอมพิวเตอร์
ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะ วิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีที่สำเร็จการศึกษา 2545