รายงานวิจัยฉบับสมบูรณ์

การค้นหาคำสำคัญโดยใช้ความหมาย

(Semantic-Based Keyword Extraction)

โดย

รองศาสตราจารย์ ดร. โกสินทร์ จำนงไทย

31 พฤษภาคม พ.ศ. 2551

สัญญาเลขที่ RMU4880007

รายงานวิจัยฉบับสมบูรณ์

โครงการการค้นหาคำสำคัญโดยใช้ความหมาย

(Semantic-Based Keyword Extraction)

รองศาสตราจารย์ ดร. โกสินทร์ จำนงไทย

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

Project Code: **RMU4880007**
(รหัสโครงการ)

Project Title: **โครงการการค้นหาคำสำคัญโดยใช้ความหมาย**
(ชื่อโครงการ**)** **(Semantic-Based Keyword Extraction)**

Investigator: รองศาสตราจารย์ ดร. โกสินทร์ จำนงไทย
(ชื่อนักวิจัย) ภาควิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

E-mail Address: **kosin.cha@kmutt.ac.th, kosin.cha@gmail.com**

Project Period: **3 ปี** **(พฤษภาคม 49 – พฤษภาคม 51)**
(ระยะเวลาโครงการ)

In several keyword extraction systems based on semantic determination, the extracted keywords are selected from many candidates having similar meanings as ones in the knowledge base. To use only term similarity is limited because two candidates can be related without being similar. Keywords may not be considered only synonym but also antonym, hypernym and hyponym. Using term relatedness for considering candidates from a document can disclose more keywords than term similarity. In this research, we use three features as synonym (its similarity), hypernym (its generality) and hyponym (its particularity) formed as term relatedness for extracting keywords. Moreover, in previous researches, they used term relatedness level to extract a keyword. This judgment is limited to a single part-of-speech such as a pair of nouns or a pair of verbs. Since, each term has several senses of meaning; two related terms may be ignored if they are considered in different senses. Therefore, we need to know exactly which sense of term is considered. In this research, we use sentence-level relatedness determination instead of term relatedness to extract keywords. By using sentence-based relatedness, each sense of term can be known by its function in the sentence. Therefore, this research proposes a sentence relatedness measurement for determining keywords that can provide additional keywords. We experiment the proposed measurement to 100 technical papers from IEICE and ACM transactions in domain of computer science and engineering. The performance of the keyword extraction system is significantly improved.

Keywords— keyword extraction, synonym, hypernym, hyponym

สารบัญ

รายการ	หน้า
รายการรูปภาพ	**v**
รายการตาราง	**vii**
บทที่ 1 บทนำ	**1**
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	**7**
บทที่ 3 ระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมาย	**18**
บทที่ 4 ขั้นตอนการวิจัยและผลการวิจัย	**33**
บทที่ 5 การนำงานวิจัยไปใช้ประโยชน์	**38**
บรรณานุกรม	**40**
ภาคผนวก	**43**
[1] Chey, C., Kumhom, P. and Chamnongthai, K. (2006) Khmer Printed Character Recognition by using Wavelet Descriptors. **International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems**, 14(3), 337-350.	**44**
[2] Kongkachandra, R., Chamnongthai, K., and Kimpan, C., (2006), Intelligent Keyword Extraction for Digital Library, **Information Technology and Libraries**, Vol. 25, No. 2, (to be appeared).	**59**
[3] Abduction-Based Knowledge Revision for Key phrase Extraction System, **Computational Intelligence**, (to be submitted)	**76**

รายการรูปภาพ

รายการ	หน้า
รูปที่ 1.1 แสดงผลลัพธ์ของเครื่องมือค้นหา Google เมื่อใส่คำสอบถาม “Triangle” , “Bermuda” และ “Bermuda Triangle”	2
รูปที่ 1.2 กระบวนการในการ กำหนดคำสำคัญ (Keyword Assignment)	3
รูปที่ 1.3 แสดงถึงกระบวนการในการสกัดคำสำคัญ (Keyword Extraction)	5
รูปที่ 2.1 แสดงความสัมพันธ์ลักษณะต่าง ๆ ของ “car" กับคำอื่น ๆ	8
รูปที่ 2.2 ตัวอย่างการแสดงผลของเวิร์ดเนตจากการค้นหาคำว่า “match”	10
รูปที่ 2.3 แสดงตัวอย่าง Conceptual Graphs	11
รูปที่ 2.4 แสดงตัวอย่าง Conceptual Graphs	11
รูปที่ 2.5 แสดงระยะห่างระหว่าง “car” และ “boat” ในโครงข่ายเวิร์ดเนตเท่ากับ 6	12
รูปที่ 3.1 แผนภาพโดยรวมของระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมาย	18
รูปที่ 3.2 ขั้นตอนการรู้จำตัวอักษร (Optical Character Recognition)	20
รูปที่ 3.3 แผนผังรวมของระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมาย	22
รูปที่ 3.4 แสดงถึงการจัดเก็บข้อมูลภายในฐานความรู้ของคำสำคัญ	23
รูปที่ 3.5 การใช้กราฟเชิงความคิดแทนความหมายของประโยค “The dog scratches its ear with its paw”	24
รูปที่ 3.6 แสดงถึงการจัดเก็บข้อมูลภายในฐานความรู้คำสำคัญ	25
รูปที่ 3.7 หลักการของการดึงคำสำคัญเริ่มต้น (Initial Keyword Extraction	26
รูปที่ 3.8 วิธีการสำหรับคัดเลือกนามวลี	27
รูปที่ 3.9 วิธีการในการแปลงจากประโยคภาษาอังกฤษ เป็น กราฟความหมาย	28
รูปที่ 3.10 กราฟความหมายของคำสำคัญ “UNIX” และ คำคู่แข่ง “LINUX”	29
รูปที่ 3.11 การวัดความเกี่ยวข้องทางความหมาย	29
รูปที่ 3.12 การเก็บค่าข้อมูลในแต่ละโนดของกราฟความหมาย	30
รูปที่ 3.13 แสดงถึงขั้นตอนการดึงคำสำคัญที่มีความเกี่ยวข้องทางความหมาย	31
รูปที่ 3.14 แสดงถึงขั้นตอนการขยายฐานความรู้คำสำคัญ	31
รูปที่ 3.15 ตัวอย่างของการปรับปรุงฐานความรู้คำสำคัญ	32
รูปที่ 4.1 หลักในการประเมินประสิทธิภาพของระบบ	33
รูปที่ 5.1 ระบบผู้เชี่ยวชาญสำหรับวินิจฉัยโรค	38

(Interactive Medical Diagnosis Expert System)
รูปที่ 5.2 ระบบถาม-ตอบทางโทรศัพท์ **38**
(Intelligent Answering System for Tourist)
รูปที่ 5.3 ระบบอัจฉริยะสำหรับจัดประเภทบทความวิชาการ **39**
(Intelligent Technical Paper Classification)

รายการตาราง

รายการ	หน้า
ตารางที่ 4.1 การกำหนดค่าต่างๆสำหรับงานวิจัย	**34**
ตารางที่ 4.2 ประสิทธิภาพของระบบที่นำเสนอเมื่อเทียบกับงานวิจัยเดิมที่มีอยู่	**35**
ตารางที่ 4.3 ตัวอย่างของผลที่ได้ของการวิจัยเมื่อเทียบกับงานวิจัยเดิมที่ใช้คลังข้อมูล	**36**
ตารางที่ 4.4 ตัวอย่างของผลที่ได้ของการวิจัยเมื่อเทียบกับงานวิจัยเดิมที่ไม่ใช้คลังข้อมูล	**36**
ตารางที่ 4.5 ประสิทธิภาพของการนำเอาฐานความรู้ของคำสำคัญมาใช้	**37**

บทที่ 1 บทนำ

1.1 ความสำคัญ

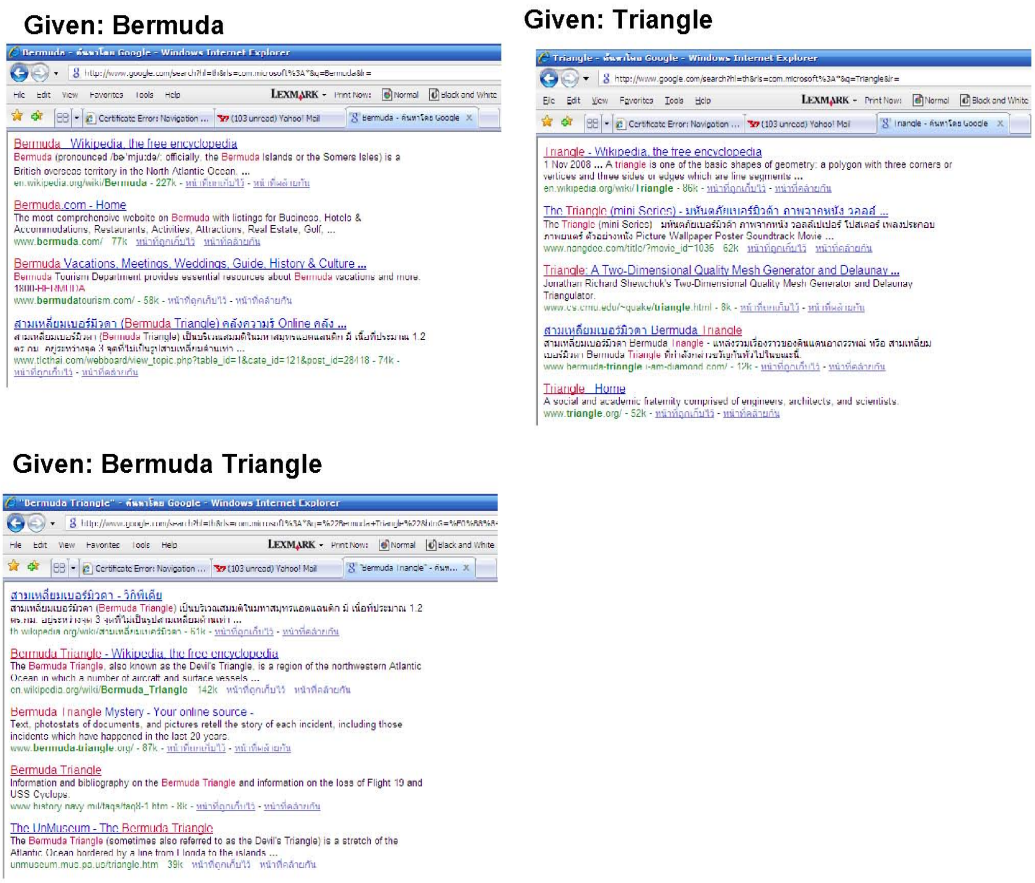
เมื่อ 20 กว่าปีที่แล้วถ้าเราต้องสืบค้นข้อมูลเกี่ยวกับเรื่องใดๆ เราจำเป็นต้องเดินทางไปยังห้องสมุดเพื่อค้นหาข้อมูลจากหนังสือบทความ นิตยสาร หรือหนังสือพิมพ์ต่างๆ อีกทั้งยังไม่สามารถการันตีได้ว่าเราจะเจอข้อมูลที่ต้องการจากการสืบค้นนั้นหรือไม่ ต่อมาอีก10ปี คอมพิวเตอร์ได้ถูกพัฒนาเพื่อใช้ในงานด้านการประมวลผลสารสนเทศมากขึ้น ตัวอย่างเช่น เครื่องมือค้นหา (Search Engine) ที่นิยมใช้กันในปัจจุบัน ได้แก่ Google, Yahoo, AltaVista หรือ InfoSeek ได้ช่วยผู้ใช้สามารถสืบค้นข้อมูลได้สะดวกสบาย และรวดเร็วขึ้น

อย่างไรก็ตาม แม้ผู้ใช้จะสามารถสืบค้นข้อมูลได้รวดเร็วขึ้น แต่ก็ยังเจอกับปัญหาของข้อมูลที่ได้จากการสืบค้นมีมากเกินไป ผู้ใช้ต้องมาคัดแยกสารสนเทศที่ต้องการออกจากข้อมูลที่สืบค้นมาได้อีกรอบหนึ่งซึ่งอาจจะเสียเวลามากเช่นกัน นอกจากนี้ การค้นหาข้อมูลผู้ใช้จำเป็นต้องใส่ คำสอบถาม (query) ที่เหมาะสมเพื่อให้ตรงกับเนื้อหาของเอกสารที่ต้องการ รูปที่1.1 แสดงผลลัพธ์ของเครื่องมือค้นหา Google เมื่อใส่คำสอบถาม ดังนี้ “Triangle” , “Bermuda” และ “Bermuda Triangle” ในตัวอย่างถ้าผู้ใช้เจตนามราจะค้นหาสารสนเทศที่เกี่ยวกับ “Bermuda Triangle” แต่ใส่คำสอบถามเพียง “Bermuda” หรือ “Triangle” ก็จะได้ข้อมูลอื่นๆที่ไม่ต้องการออกมาด้วย แต่ถ้าใส่ คำสอบถามเป็น “Bermuda Triangle” ก็จะได้สารสนเทศที่ตรงกับความต้องการ ปรากฏการณ์เช่นนี้สามารถบอกโดยนัยได้ว่า ถ้าเราใส่คำสอบถามที่เหมาะสม ก็จะได้รับข้อมูลที่ต้องการมากขึ้น เนื่องจากการทำงานโดยทั่วไปของเครื่องมือค้นหา คือ เปรียบเทียบ คำสอบถามกับคำสำคัญ (Keyword) ที่อยู่ในเอกสารซึ่งถ้าตรงกันก็จะแสดงเอกสารนั้นขึ้นมาให้กับผู้ใช้

เอกสารบางประเภท เช่น บทความวิชาการ หนังสือเรียน โดยส่วนใหญ่ผู้แต่งจะกำหนดคำสำคัญเอาไว้ในเอกสาร ทำให้เครื่องมือค้นหาสามารถเปรียบเทียบ คำสอบถาม กับ คำสำคัญเหล่านี้ได้ทันที แล้วได้ผลลัพธ์ ตามที่ต้องการ

ความสำคัญของ คำสำคัญ (Keyword) สามารถสรุปได้ดังนี้

1) คำสำคัญ สามารถใช้เป็นบทสรุปของเนื้อหาเอกสาร โดยรวมได้เราสามารถเดาเรื่องราวของเอกสารได้ด้วยการดูที่คำสำคัญ ซึ่งช่วยให้เข้าใจเอกสารนั้นได้เร็วขึ้น
2) คำสำคัญ สามารถใช้เป็น เครื่องบอกทาง ในการค้นหาข้อมูลที่เกี่ยวข้องอื่นๆได้
3) คำสำคัญ สามารถใช้ในการจัดแยกประเภทของเอกสารได้



รูปที่1.1 แสดงผลลัพธ์ของเครื่องมือค้นหา Google เมื่อใส่คำสอบถาม

“Triangle” , “Bermuda” และ “Bermuda Triangle”

1.2 ที่มาของปัญหา

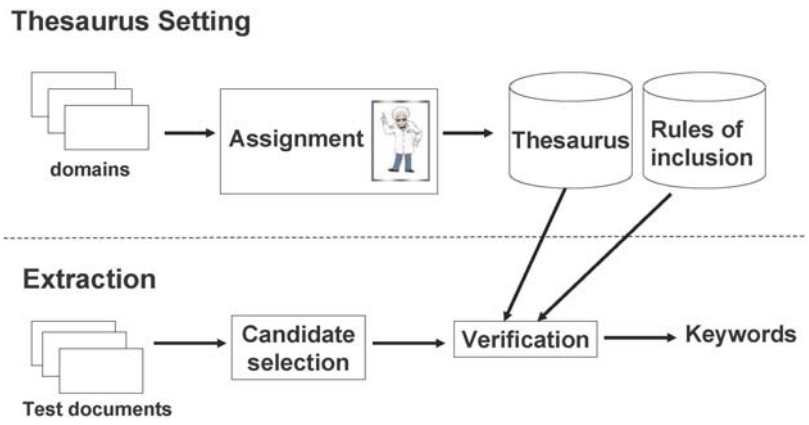
ถึงแม้ว่า คำสำคัญ จะเป็นที่ยอมรับว่าเป็นตัวแทนที่ดีของเอกสาร [1-11] แต่เอกสารส่วนใหญ่ผู้เขียนไม่ได้กำหนด คำสำคัญ เอาไว้ จะมีก็เพียงเอกสารบางชนิด เช่น บทความหรือหนังสือวิชาการเท่านั้น ดังนั้น การค้นหาคำสำคัญโดยอัตโนมัติ จึงเป็นสิ่งที่จำเป็นสำหรับการใช้งานด้านต่างๆ เช่น การค้นคืนเอกสาร การสรุปใจความสำคัญ การจัดประเภทเอกสาร และ การแปลเอกสาร เป็นต้น

งานวิจัย [12] ได้กล่าวว่า เทคนิคที่ใช้ในการสร้างระบบค้นหาคำสำคัญแบบอัตโนมัติสามารถแบ่งออกได้เป็น 2 กลุ่มคือ

1. การกำหนดคำสำคัญ (Keyword Assignment)
2. การสกัดคำสำคัญ (Keyword Extraction)

1.2.1 การกำหนดคำสำคัญ (Keyword Assignment)

เป็นกระบวนการในการดึงเอาคำสำคัญออกจากเอกสารแบบอัตโนมัติด้วยการจับคู่ ระหว่างคำที่พิจารณา (Candidate) กับ คำสำคัญที่กำหนดเอาไว้ ซึ่งเรียกว่า คำศัพท์ควบคุม (Controlled Vocabulary) หรือ อรรถาภิธาน (Thesaurus) ซึ่งถูกกำหนดขึ้นมาจากการที่ผู้เชี่ยวชาญ อ่านเอกสารทั้งหมด ทำความเข้าใจ และ กำหนดคำศัพท์ควบคุมเหล่านี้ขึ้น นอกจากนี้ จะกำหนดกฎที่ใช้ในการผนวกคำศัพท์ควบคุมเข้ากับคำอื่นๆ (Rules of Inclusion) ด้วย ขั้นตอนนี้แสดงในรูปที่ 1.2 ในส่วนเตรียมคำศัพท์ควบคุม (Thesaurus Setting) จากนั้นเมื่อต้องการค้นหาคำสำคัญแบบอัตโนมัติ เอกสารที่ต้องการทดสอบจะถูกนำมาพิจารณาหา คำที่พิจารณา (Candidate) จากนั้นก็ทำการเปรียบเทียบกับ คำศัพท์ควบคุมและกฎของการผนวกคำศัพท์ คำที่พิจารณา (Candidate) ใดเป็นไปตามกฎที่ใช้ ก็จะนับว่าเป็น คำสำคัญ (Keyword) ของเอกสารนั้น ขั้นตอนนี้คือ ส่วนดึงคำสำคัญ (Extraction) ในรูปที่ 1.2



รูปที่ 1.2 กระบวนการในการ กำหนดคำสำคัญ (Keyword Assignment)

ตัวอย่างเช่น ผู้เชี่ยวชาญ กำหนดคำว่า “childbirth” ขึ้นมาเป็นคำศัพท์ควบคุม จากนั้นก็จะใช้กฎในการผนวกคำศัพท์ เชื่อมโยงไปยังคำอื่นที่เกี่ยวข้อง ได้แก่ “birth”, “labor”, “labour”, “delivery”, “forceps”, “baby” และ “born” เป็นต้น

เนื่องจาก เทคนิคการกำหนดคำสำคัญ ใช้ผู้เชี่ยวชาญในการกำหนดคำศัพท์ควบคุม เราจึงถือได้ว่า คำสำคัญ ที่ได้เทคนิคนี้ เป็นตัวแทนทางความหมายที่ดีของเอกสาร แต่อย่างไร ก็ตามวิธีนี้มีข้อจำกัด ดังนี้

- วิธีนี้ต้องอาศัยผู้เชี่ยวชาญในการกำหนดคำศัพท์ควบคุม ซึ่งหาได้ยากและต้องเสียค่าใช้จ่ายมาก
- วิธีนี้ ต้องใช้เวลาและความทุ่มเทอย่างมาก
- วิธีนี้ ใช้กฎในการผนวกคำศัพท์ที่เป็นแบบ Heuristic ซึ่งขึ้นอยู่กับผู้เชี่ยวชาญเฉพาะคน

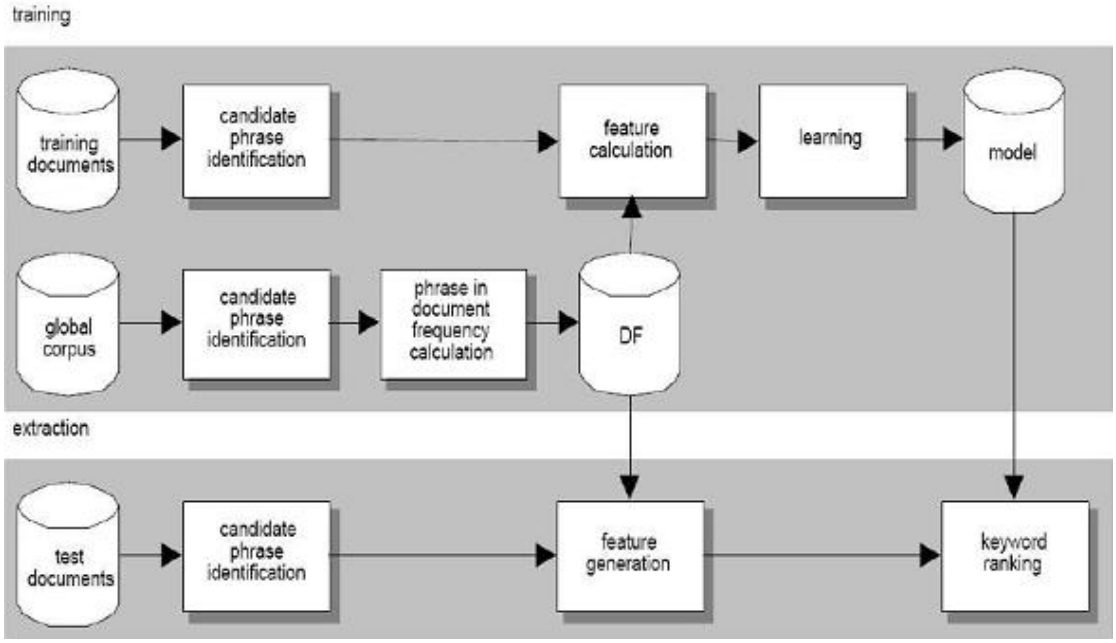
ได้มีงานวิจัยหลายงานที่พัฒนาขึ้นมาโดยใช้เทคนิคนี้ อาทิเช่น E.J. Schuegraf และ F. Van Bomme [14] เสนอวิธีการโดยใช้ผู้เชี่ยวชาญหลายคนมาร่วมกันกำหนดกฎในการผนวกคำศัพท์ ซึ่งงานวิจัยนี้จะพิจารณาว่า คำใดที่มักเกิดขึ้นร่วมกัน (Co-occurrence) กับคำศัพท์ควบคุมบ่อยๆ เป็นคำที่สามารถนำมาผนวกเข้ากับคำศัพท์ควบคุมได้ งานวิจัยนี้สามารถลดเวลาในการค้นหาความสำคัญได้ แต่ยังคงต้องใช้ผู้เชี่ยวชาญอยู่ งานวิจัยอีกงานโดย C.H. Lueng และ W.K. Kan [15] เสนอที่จะไม่ใช้ผู้เชี่ยวชาญ แต่ใช้การฝึกฝนจากกลุ่มตัวอย่างที่ถูกต้อง (Positive example) และไม่ถูกต้อง (Negative example) ในการหาความสัมพันธ์ระหว่าง คำศัพท์ควบคุม กับ คำอื่นๆที่จะนำมาผนวกเข้ากับคำศัพท์ควบคุม ซึ่งความสัมพันธ์เหล่านี้คำนวณได้มากจากผลคูณของ TF x OSDF x CSIDF โดยที่ TF: Term Frequency, OSDF: Document Frequency in Original Set และ CSIDF: Inverse Document Frequency in Combined Set)

1.2.2 การสกัดคำสำคัญ (Keyword Extraction)

เทคนิค นี้เป็นอีกทางเลือกหนึ่งในการค้นหาคำสำคัญจากเอกสารโดยไม่ต้องใช้ผู้เชี่ยวชาญ วิธีนี้จะอาศัยหลักการของการฝึกฝนเพื่อให้รู้จำรูปแบบของคำสำคัญ เพื่อนำรูปแบบที่ได้นี้ ไปค้นหาคำสำคัญในเอกสารอื่นๆ ต่อไป ดังนั้น แทนที่จะใช้ผู้เชี่ยวชาญมาเป็นคนกำหนดรูปแบบของคำสำคัญ เราใช้ ตัวอย่างของเอกสาร (Training document) และ คลังข้อมูล (Corpus) จำนวนมากมาฝึกสอนเพื่อให้ระบบสามารถสร้างแบบจำลองของคำสำคัญ (Keyword Model) ได้ รูปที่ 1.3 แสดงถึงกระบวนการในการสกัดคำสำคัญ (Keyword Extraction) ซึ่งประกอบไปด้วย 2 ส่วนคือ ส่วนของการฝึกฝนเพื่อสร้างแบบจำลอง และ ส่วนของการสกัดคำสำคัญ

ในส่วนของการฝึกฝน เอกสารเพื่อฝึกฝน (Training document) และ คลังข้อมูล (Corpus) จะถูกนำมา พิจารณาเพื่อเลือกคำที่เราจะนำมาพิจารณาว่าเป็นคำสำคัญ (Candidate phrase identification) จากนั้น คำที่ถูกเลือกเหล่านี้จะถูกนำมาวิเคราะห์ว่ามีลักษณะเด่นใดบ้าง (Feature Extraction) จากนั้นนำเข้าสู่การเรียนรู้ เพื่อสร้างแบบจำลองของคำสำคัญ และเก็บไว้ในฐานข้อมูลเพื่อใช้ในส่วนของการสกัดคำสำคัญ ต่อไป

ส่วนของการสกัดคำสำคัญ จะใช้ เอกสารที่ต้องการทดสอบมา พิจารณาเลือกคำที่เราจะนำมาพิจารณาว่าเป็นคำสำคัญ จากนั้น คำที่ถูกเลือกเหล่านี้จะถูกนำมาวิเคราะห์ว่ามีลักษณะเด่นใดบ้าง เมื่อได้แล้วนำมาเปรียบเทียบกับ แบบจำลองของคำสำคัญที่เก็บไว้ คำที่พิจารณาใดมีใกล้เคียงกับแบบจำลอง **N** ตัวแรก จะถูกกำหนดว่าเป็น คำสำคัญของเอกสาร



รูปที่ 1.3 แสดงถึงกระบวนการในการสกัดคำสำคัญ (Keyword Extraction)

ถึงแม้ว่าวิธีการนี้จะเป็นที่ยอมรับและใช้กันมากในงานวิจัยทางด้านนี้ เนื่องจากสามารถทำให้เป็นระบบอัตโนมัติได้ง่ายกว่าเทคนิคแรก แต่ก็ยังมีข้อจำกัด ดังนี้

- ในส่วนของการฝึกฝน เพื่อให้ได้แบบจำลองของคำสำคัญที่ดี ต้องการจำนวนเอกสารจำนวนมาก
- ในการเปรียบเทียบความเหมือนกันระหว่างคำที่พิจารณากับแบบจำลองคำสำคัญ ส่วนใหญ่จะเป็นการเปรียบเทียบในแนวตื้น (Shallow Comparison) เช่น มีรูปแบบการเขียนเหมือนกันหรือไม่ ซึ่งในความเป็นจริงแล้ว คำทีเหมือนกันอาจจะมีรูปแบบการเขียนที่ต่างกัน แต่มีความหมายเหมือนกัน

จากข้อจำกัดนี้ ทำให้ คำสำคัญ ที่ได้ ไม่สามารถนำมาใช้เป็นตัวแทนของเอกสารได้ดี เท่ากับ เทคนิคของการกำหนดคำสำคัญ (Keyword Assignment)

1.3 เป้าหมายของการวิจัย

จากข้อจำกัดของ การค้นหาคำสำคัญในปัจจุบัน ที่กล่าวมาข้างต้น งานวิจัยนี้ จึงนำเสนอวิธีการในการปรับปรุงวิธีการในการค้นหาคำสำคัญ โดยมุ่งเน้นจะให้มีข้อดีใน 2 เรื่อง คือ

1. เพื่อให้ได้ คำสำคัญ ที่เป็นตัวแทนของเอกสารได้ดี โดยตั้งสมมุติฐานว่า คำสำคัญที่เป็นตัวแทนเอกสารที่ดี จะมีความหมายสอดคล้องกับเนื้อหาโดยรวมของเอกสาร ดังนั้น งานวิจัยนี้เสนอการดึงคำสำคัญโดยพิจารณาความหมายของคำที่นำมาเปรียบเทียบกัน ระหว่างคำที่พิจารณา กับ แม่แบบของคำสำคัญที่กำหนด
2. เพื่อให้ใช้งานได้สะดวก โดยลดความต้องการเอกสารจำนวนมากที่นำมาใช้ในการฝึกฝนเพื่อให้ได้แบบจำลองคำสำคัญ งานวิจัยนี้เสนอ การดึงคำสำคัญ ที่ใช้ แม่แบบทางความหมายของคำสำคัญ ที่ได้มาจากเอกสารที่กำลังพิจารณาเท่านั้น ซึ่ง แม่แบบทางความหมายของคำสำคัญ สามารถหาได้จากเอกสารนั้นๆ โดยพิจารณาจาก ชื่อเรื่องของเอกสารนั้น โดยตั้งสมมุติฐานว่า ผู้เขียนเอกสาร มักจะตั้งชื่อเอกสารโดยใช้ คำที่มีความหมายเกี่ยวข้องกับเนื้อหาเอกสาร

1.4 ประโยชน์ที่ได้รับจากงานวิจัย

ผลที่ได้รับจากงานวิจัยนี้ คือ วิธีการในการค้นหาความสำคัญแบบอัตโนมัติ โดยพิจารณาดูจากความหมายของคำ ว่าสอดคล้องกับแม่แบบคำสำคัญ ซึ่งมีประโยชน์ต่อการประยุกต์ใช้งานต่างๆ ในหัวข้อดังต่อไปนี้

1. รูปแบบจำลองของคำสำคัญเชิงความหมาย ที่ได้จากงานวิจัย สามารถนำไปประยุกต์ใช้กับงานที่มุ่งเน้นการเปรียบเทียบในเชิงความหมาย
2. ขั้นตอนวิธีในการเปรียบเทียบเชิงความหมาย ซึ่งสามารถประยุกต์ใช้ได้กับงานอื่นๆ ที่เป็นการประมวลผลสารสนเทศแบบข้อความ (Text-based Information Processing)
3. วิธีการในการ ปรับปรุงฐานความรู้ของระบบค้นหาคำสำคัญ ให้ทันสมัยอยู่ตลอดเวลา ซึ่งสามารถไปประยุกต์ใช้กับงานอื่นๆ ที่เกี่ยวกับ การเรียนรู้เครื่องจักร (Machine Learning) และ การทำเหมืองข้อมูล (Data Mining)

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีที่ใช้ในงานวิจัย

รายงานวิจัยฉบับนี้ นำเสนอการใช้ความหมายของคำในประโยคมาช่วยในการค้นหาคำสำคัญของเอกสาร ซึ่งจำเป็นต้องใช้ทฤษฎีที่เกี่ยวข้องดังนี้

2.1.1 Semantic Similarity และ Semantic Relatedness

2.1.2 WordNet

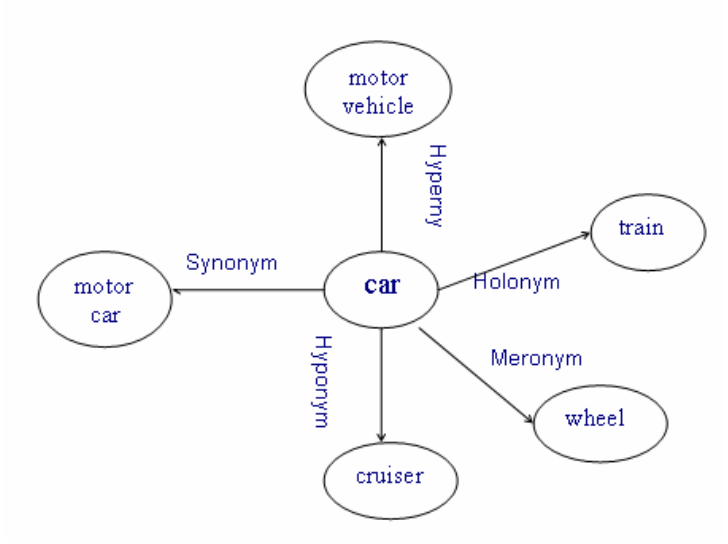
2.1.3 Knowledge Representation

2.1.1 Semantic Similarity และ Semantic Relatedness

Semantic Similarity คือการเปรียบเทียบคำโดยพิจารณาว่าเหมือนกันในเชิงความหมายหรือไม่ เช่น “car” กับ “bicycle” จะมองว่ามีความหมายเหมือนกัน คือเป็นพาหนะที่มีล้อเหมือนกัน ในการเปรียบเทียบจะได้ผลเป็นมีความหมายเหมือนกันหากผลการเปรียบเทียบที่ได้อยู่ภายในค่า Threshold ที่กำหนด แต่หากเกินค่า Threshold จะถือว่าไม่มีความหมายเหมือนกัน

Semantic Relatedness คือการเปรียบเทียบคำในเชิงความหมาย โดยจะพิจารณาว่ามีความหมายที่มีความสัมพันธ์กันหรือไม่ ซึ่งการพิจารณาความสัมพันธ์เชิงความหมายนั้นเป็นแนวคิดที่เป็นทั่วไปมากว่าการพิจารณาความหมายเหมือนกัน คือในขณะที่คำที่ไม่มีความเหมือนกันในเชิงความหมาย แต่อาจจะเป็นคำที่มีความสัมพันธ์กันในเชิงความหมาย เช่น “car” กับ “wheel” ไม่มีความหมายที่เหมือนกันกัน แต่มีความสัมพันธ์กันในเชิงความหมายในลักษะที่ “car” มี “wheel” เป็นส่วนประกอบ (Meronym) ซึ่งความสัมพันธ์กันในเชิงความหมายนั้นสามารถแบ่งลักษณะความสัมพันธ์ได้ดังนี้

- Synonym มีความหมายเหมือนกัน
- Antonym มีความหมายที่ตรงกันข้ามกัน
- Hypernym มีความหมายที่เป็นแบบทั่วไปกว่า
- Hyponym มีความหมายที่เป็นแบบเฉพาะเจาะจงกว่า
- Holonym มีความสัมพันธ์ทางความหมายในลักษณะที่เป็นส่วนประกอบหรือสมาชิกของ
- Meronym มีความสัมพันธ์ทางความหมายในลักษณะที่มีเป็นส่วนประกอบหรือมีเป็นสมาชิก



รูปที่ 2.1 แสดงความสัมพันธ์ลักษณะต่าง ๆ ของ “car" กับคำอื่น ๆ

ซึ่งการเปรียบเทียบเพื่อหาความสัมพันธ์ในเชิงความหมายนี้ใช้ในการสรุปใจความเอกสาร (Text Summarization) และระบบค้นคืนสารสนเทศ (Information Retrieval) ซึ่งจะทำให้ระบบงานค้นคืนสารสนเทศสามารถค้นคืนข้อมูลหรือเอกสารที่มีความสัมพันธ์กันได้ครอบคลุมความต้องการมากขึ้น

ในการเปรียบเพื่อหาความสัมพันธ์ในเชิงความหมายจะพิจารณาจาก

- โครงข่ายของเวิร์ดเนต (WordNet) ซึ่งคำสองคำที่มีระยะห่างน้อยจะถือว่ามีค่าความสัมพันธ์กันมาก โดยหาความสัมพันธ์เชิงความหมายของคำจากการคำนวณจากเส้นทางระหว่างคำ (Synset) ที่สั้นที่สุดในเวิร์ดเนต งานวิจัยในกลุ่มนี้มีข้อดีคือง่าย แต่มีข้อจำกัดตรงที่คำที่นำมาเปรียบเทียบกันนั้นต้องมีอยู่ใน ฐานข้อมูล WordNet นอกจากนี้ คำที่เขียนเหมือนกันอาจอยู่ได้หลายที่ใน WordNet ขึ้นอยู่กับความหมาย เช่น คำว่า “mouse” อาจหมายถึง “หนูที่เป็นสัตว์” หรือ “อุปกรณ์คอมพิวเตอร์” ทำให้ค่าที่ได้อาจจะผิดพลาดได้
- พิจารณาความสัมพันธ์จากค่าข้อมูล (Information Content หรือ Corpus Based) หรือข้อความที่เนื้อหาของคำทั้งสองสถิตอยู่ โดยวิธีนี้จะใช้การคำนวณทางสถิติ เช่น การกระจายของคำ, จำนวนครั้งที่เกิดขึ้น, ตำแหน่งของคำ ด้วยวิธีนี้สามารถแก้ปัญหาของคำที่ไม่มีใน WordNet ของกลุ่มแรกได้ แต่อย่างไรก็ตามวิธีการนี้จำเป็นต้องมีการสร้างคลังข้อมูลตัวอักษร (Corpus) และผลการคำนวณที่ได้ก็จะขึ้นอยู่กับขนาดและข้อมูลที่มีในคลังข้อมูลนั้นๆ
- พิจารณาจากการเปรียบเทียบคำจำกัดความหรือคำนิยามหรือคำแปลของคำทั้งคู่ (Gloss based) โดยดูคำนวณจากคำที่ใช้ร่วมกัน (Shared word) ในการอธิบายความหมายของคำทั้งคู่

ซึ่งในการหาค่าสัมพันธ์ในเชิงความหมายโดยพิจารณาจากข้อความที่เป็นเนื้อหาหรือการคำแปลที่มีการใช้คำร่วมกัน ต่อมีได้มีการปรับปรุงโดยในการเปรียบเทียบกันระหว่างเนื้อหาหรือคำแปลนั้นนอกจากจะพิจารณาความเหมือนของคำที่ใช้ร่วมกันแล้วยังมีการพิจารณาความสัมพันธ์ในเชิงความหมายที่เป็น Synonym, Hypernym และ Hyponym ด้วย โดยใช้เวิร์ดเนตเป็นฐานความรู้ (Knowledge-based) ในการพิจารณาเปรียบเทียบหาความสัมพันธ์ในระดับต่าง ๆ

2.1.2 เวิร์ดเนต (WordNet)

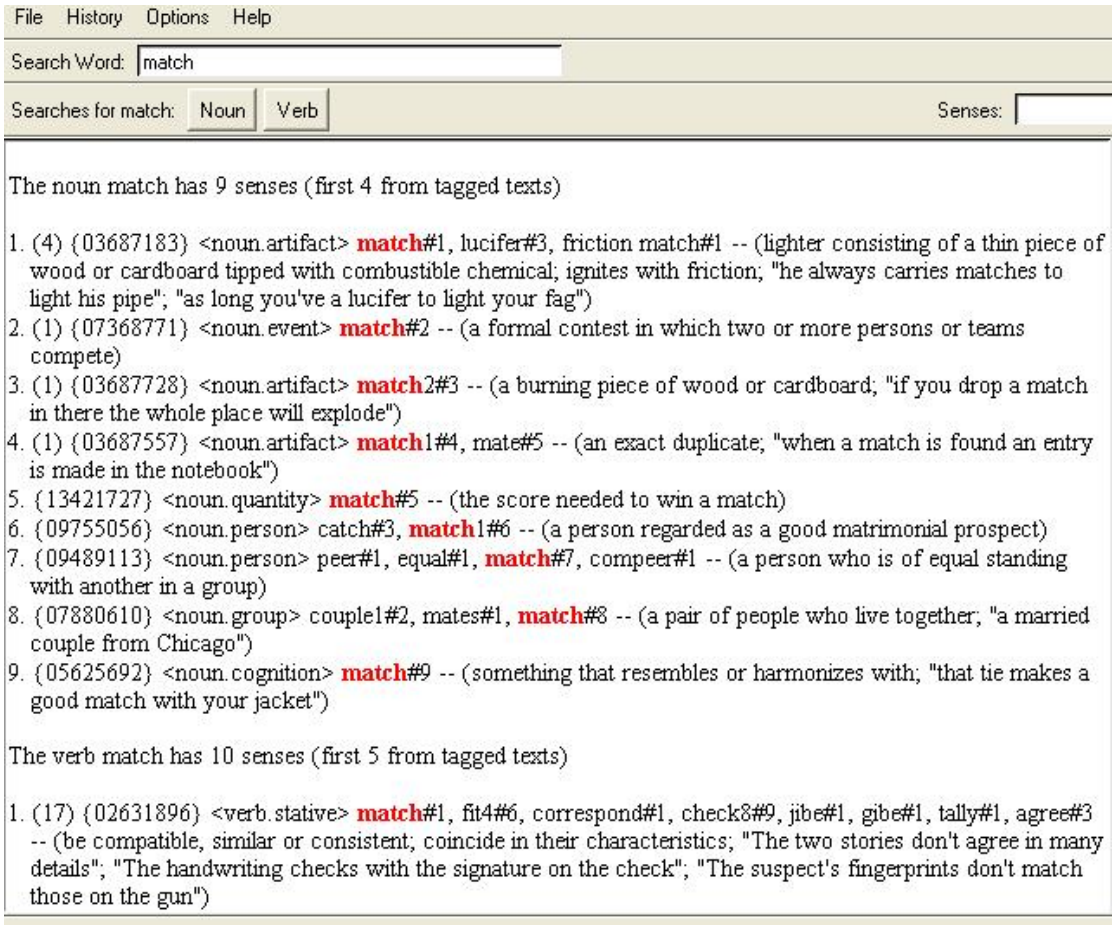
เวิร์ดเนต (Cognitive Science Laboratory at Princeton University) เป็นฐานข้อมูลพจนานุกรมภาษาอังกฤษผู้ใช้สามารถdownload มาใช้หรือใช้งาน ในลักษณะออนไลน์ได้ โดยเวิร์ดเนตได้รวมคำในภาษาอังกฤษที่มีความหมาย(Sense) เหมือนกันเข้าเป็นกลุ่มเรียกกว่า Synset เช่น “good” “right” และ “ripe” เป็น Synset เดียวกัน มีความหมายเหมือนกัน คือ “most suitable or right for a particular purpose”

ในระหว่าง Synset จะสัมพันธ์กันโดยมีความสัมพันธ์เชิงความหมายหลายรูปแบบคำ เช่น มีความหมายเหมือนกัน (Synonym) หรือมีความหมายตรงกันข้าม (Antonym) ซึ่งความสัมพันธ์จะขึ้นอยู่กับชนิดของคำโดยจะแยกความแตกต่างกันระหว่างคำที่เป็น Noun, Verb, Adjective และ Adverb ตามหลักไวยากรณ์ Synset ที่มีความสัมพันธ์กันเชิงความหมายจะมีหมายเลขที่เชื่อมต่อกัน ทุก Synset ที่เชื่อมต่อกันโดยตัวเลขของความเกี่ยวข้องเชิงความหมาย

คำ 1 คำในเวิร์ดเนตอาจมีได้หลายความหมาย นั่นคือคำ 1 คำ อาจปรากฏอยู่ในหลาย Synset โดยที่แต่ละ Synset มีหมายเลขกำกับที่แตกต่างกัน ในขณะเดียวกันอาจจะมีคำหลายคำที่มีความหมายเดียวกัน จากรูปที่ 2.1 จะเห็นได้ว่า “match” ที่เป็นคำนามมี 9 ความหมาย และ “match” ที่เป็นคำกริยามี 10 ความหมาย ในขณะที่บางความหมายมีคำที่มีหมายเหมือนกัน (Synonym) ด้วย

เช่น ว่า “match” ที่เป็นคำนามความหมายที่ 1 มี lucifer, friction, match เป็นคำที่มีความหมายเดียวกัน

เนื่องจากเวิร์ดเนตเป็นฐานข้อมูลพจนานุกรมซึ่งได้รวบรวมความหมายต่าง ๆ ของทั้งที่เป็น Noun, Verb, Adjective และ Adverb และสามารถเชื่อมโยงความสัมพันธ์เชิงความหมายระหว่างคำในแบบต่าง ๆ ได้ครอบคลุม รวมทั้งเวิร์ดเนตถูกพัฒนาและมีการเพิ่มคำศัพท์มาอย่างต่อเนื่องจนถึงปัจจุบัน ซึ่งประกอบไปด้วยข้อมูลของคำทั้งหมดประมาณ 90,000 คำ งานวิจัยนี้จึงเลือกใช้ เวิร์ดเนตเป็นฐานความรู้ในการเปรียบเทียบเพื่อคำนวณหาค่าความสัมพันธ์



รูปที่ 2.2 ตัวอย่างการแสดงผลของเวิร์ดเนตจากการค้นหาคำว่า “match”

2.1.3 การแทนความรู้ (Knowledge Representation)

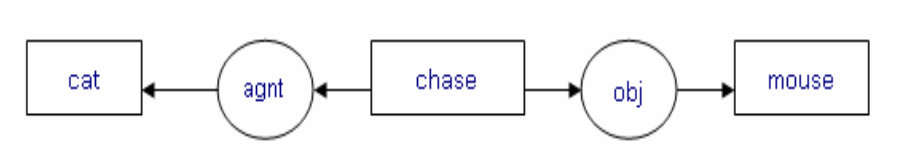
รายงานวิจัยฉบับนี้ใช้กราฟเชิงความคิด (Conceptual Graph) ในการแทนความรู้ กราฟเชิงความคิด (Chein et Marie-Laure Mugnier, M.,1992) คิดค้นโดย John F. Sowa เพื่อนำมาใช้เป็นรูปแบบการแทนความรู้ และนำมาประยุกต์ใช้กับการแสดงความหมายทางภาษาเพื่อดึงความรู้ (Knowledge Acquisition) ในการวิจัยทางด้านปัญญาประดิษฐ์(Artificial Intelligence) จุดเด่นที่สำคัญของกรางเชิงความคิดคือสามารถอธิบายภาษาธรรมชาติ (Natural Language) ออกมาในรูปแบบของสัญลักษณ์เชิงตรรกศาสตร์และเป็นตัวกลางในการสื่อความหมายระหว่างภาษาของคอมพิวเตอร์และภาษาของมนุษย์ ทำให้การสื่อแทนความหมายโดยใช้กราฟเชิงความคิดนั้นเข้าใจได้ง่ายและพัฒนากลไกในการหาเหตุผลเชิงตรรกะได้ ในการแทนความรู้โดยใช้กราฟเชิงความคิดประกอบด้วย

1. Concept Nodes แทนด้วย สัญลักษณ์รูปสี่เหลี่ยม ซึ่งจะแสดง Entities, Actions และ Attributes ซึ่งมี 2 Attributes

- Type จะเป็นตัวบอก class ของ element ที่แทนด้วย concept
- Referent จะเป็นตัวบอกความเฉพาะเจาะจงของ class ที่ถูกอ้างโดย node

2. Conceptual Relation Nodes แทนด้วย สัญลักษณ์รูปวงกลม จะแสดงความสัมพันธ์ระหว่าง Concept node โดย Attributes ของ Relation Nodes จะมี 2 Attributes คือ

- Valence จะเป็นตัวบอกจำนวนของ concept ที่อยู่ข้าง ๆ relation
- Type แสดงความเกี่ยวข้องของตัวมันเอง เช่น subject, attribute, และ object



รูปที่ 2.3 แสดงตัวอย่าง Conceptual Graphs

ในรูปที่ 2.2 แสดง conceptual graph ของประโยค “Tom is chasing a brown mouse” ประกอบด้วย 3 concepts และ 2 relations ซึ่งสามารถเขียนในรูปแบบการแสดงผลที่ไม่เป็นกราฟิกได้ดังนี้

[cat: Tom]<-(Agnt)<-[chase]->(Ptnt)->[mouse]->(Attr)->[brown]

รูปที่ 2.4 แสดงตัวอย่าง Conceptual Graphs

ทิศทางของลูกศรในรูปด้านบนแสดงถึงประธาน (Subject) และกรรม (Object) ของความเกี่ยวข้อง (Relation) concept [cat: Tom] เป็นconcept ที่เฉพาะเจาะจงของ Type cat ในขณะที่ [chase] และ [mouse] เป็น generic concept ความเกี่ยวข้อง (Attr) บอก attribute ของ mouseว่ามีสี brown

หลักการอีกส่วนหนึ่งที่สำคัญในการใช้กราฟเชิงความคิดในการแทนความรู้ คือการแสดงความเกี่ยวข้องระหว่างกลุ่มขององค์ประกอบแบบลำดับขั้น (Type hierarchy) หลักการดังกล่าวทำให้สามารถนำเอาคุณสมบัติการสืบทอด (Inheritance) มาประยุกต์ใช้ กล่าวคือองค์ประกอบที่มีลักษณะเฉพาะเรียกว่าเป็น Subtype ขององค์ประกอบที่มีลักษณะทั่วไปซึ่งเรียกว่า Supertype สิ่งที่

เป็น Subtype ขององค์ประกอบหนึ่ง ๆ จะสืบทอดคุณสมบัติขององค์ประกอบนั้น ๆ นอกจากนี้ หลักการของกราฟเชิงความคิดได้ถูกนำไปใช้ในระบบงานค้นคืนสารสนเทศ (Montes-y-Gomez, Lopezs-Lopez, and Gelbukh, A., 2000) (Montes-y-Gomez, Lopezs-Lopez, Gelbukh, and Baeza-Yates, 2001) โดยอาศัยหลักการเปรียบเทียบกันระหว่างกราฟ (Graph Matching)

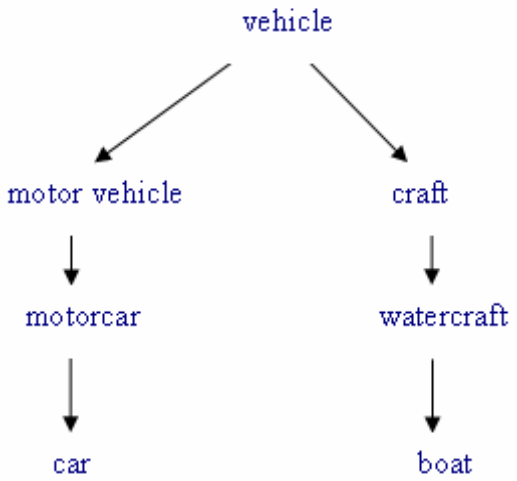
เนื่องจากกราฟเชิงความคิดสามารถแสดงส่วนประกอบ (concept) ในประโยค รวมทั้งแสดงความสัมพันธ์ของส่วนประกอบในประโยคนั้นด้วย ในงานวิจัยนี้จึงใช้กราฟเชิงความคิดในการแทนความหมายของประโยค

2.2 งานวิจัยอื่น ๆ ที่เกี่ยวข้อง

2.2.1 การเปรียบเทียบหาค่าความเหมือนเชิงความหมาย

Resnik (1995) และ Jiang and Conrath (1997) เสนอแนวคิดว่าคำจะมีความหมายเหมือนกันนั้นสามารถพิจารณาได้จากความเหมือนกันของข้อความที่เนื้อหาของคำทั้งสอง ซึ่งวิธีการนี้จำเป็นต้องมีสร้างฐานความรู้คลังข้อมูลตัวอักษร (Corpus) และผลการคำนวณที่ได้ก็จะขึ้นอยู่กับขนาดและข้อมูลที่มีใน คลังข้อมูลตัวอักษร (Corpus) ด้วย

Leacock and Chodorow (1998) เสนอการคำนวณหาค่าความเหมือนกันทางความหมายของคำโดยคำนวณจากระยะห่างระหว่างคำ (Synset) ทั้งสองที่สั้นที่สุดโดยอาศัยโครงข่ายในเวิร์ดเนตโดยพิจารณาความสัมพันธ์แบบ “is-a”



รูปที่ 2.5 แสดงระยะห่างระหว่าง “car” และ “boat” ในโครงข่ายเวิร์ดเนตเท่ากับ 6

2.2.2 การเปรียบเทียบหาค่าความสัมพันธ์เชิงความหมาย

Hist and St-Ong (1998) คำนวณค่าความสัมพันธ์เชิงความหมายโดยอาศัย โครงข่ายในเวิร์ดเนต จะพิจารณาจำนวนครั้งของการเปลี่ยนทิศทางของเส้นทาง มีสูตรในการคำนวณ คือ

$$Rel\ (c_1, c2) = C - path\ length - k * d \quad (2.1)$$

โดย*ที่* d จำนวนครั้งของการเปลี่ยนทิศทางใน path,
C *และ* k เป็นค่าคงที่
ซึ่งเส้นทางความสัมพันธ์ที่พิจารณานั้นประกอบด้วย Synonym, Hyponym, Hypernym และ Holonym โดยที่จะให้ความสำคัญในแต่ละรูปแบบความสัมพันธ์ที่เท่ากัน

Banerjee and Pederson (2003) เสนอวิธีการคำนวนหาความสัมพันธ์ในเชิงความหมายของคำซึ่งได้ประยุกต์ใช้หลักการของ Lesk (1986) ซึ่งจะนับจำนวนซึ่งสถิตในคำแปล (gloss) ของคำทั้งสองที่เหมือนกัน (overlap) โดยใช้ความคำแปลและโครงข่ายในเวิร์ดเนตเป็นฐานความรู้ แต่จะขยาย set ของคำแปลที่จะเปรียบเทียบโดยการนำเอาคำผลการเปรียบเทียบหาคำที่สถิตอยู่เหมือนกันในคำแปลของคำที่เป็น Hyponym และ Hypernym ของคำที่ต้องการเปรียบเทียบทั้งสองมาร่วมคำนวณค่าสัมพันธ์ด้วย ดังนี้

$$\begin{aligned} Relatedness(A,B) = {} & score(gloss(A), gloss(B)) + score(hype(A), hype(B)) + \\ & score(hypo(A), hypo(B)) + score(hype(A), gloss(B)) + \\ & score(gloss(A), hype(B)) \end{aligned} \quad (2.2)$$

นอกจากนี้ยังมีการให้น้ำหนักตามความยาวของวลีที่ปรากฏด้วย เช่น ปรากฏคำว่า “bank” เหมือนกันจะได้คะแนนเป็น 1 ในขณะที่ปรากฏคำว่า “bank account” เหมือนกันคะแนนที่ได้จะเป็น 2^2 เท่ากับ 4 เป็นต้น

Strube and Ponzetto (2006) เสนอวิธีการ WikiRelate โดยใช้หลักการเดียวกันกับ Lesk (1986) โดยใช้คำแปลหรือคำนิยามของคำใน Wikipedia ซึ่งเป็นพจนานุกรมภาษาอังกฤษแบบออนไลน์ในการคำนวณหาค่าความสัมพันธ์ในเชิงความหมาย โดยคำแปลที่นำมาคำนวณค่าความสัมพันธ์นั้นจะใช้ข้อความจากย่อหน้าแรกใน Wikipedia ในการเปรียบเทียบคำที่สถิตอยู่

เหมือนกันนั้นมีการให้น้ำหนัก ตามความยาวของวลีที่ปรากฏเช่นเดียวกับ Banerjee and Pederson (2003) โดยมีสูตรในการคำนวณหาความสัมพันธ์ดังนี้

$$Related_{gloss}(t_1,t_2) = \tanh(overlap(t_1,t_2) / length(t_1) + length(t_2)) \quad (2.3)$$

ซึ่งเมื่อเปรียบเทียบผลที่ได้กับการคำนวณหาค่าความสัมพันธ์กันโดยใช้นิยามของคำจากเวิร์ดเนต ปรากฏว่าให้ผลที่ดีกว่าการใช้คำนิยามจากเวิร์ดเนต

Zhang, Sun, Wang, and Bai, (2005) นำเสนอการวัดความสัมพันธ์ของประโยคกับหัวข้อเอกสารโดยการเปรียบเทียบหาความสัมพันธ์ในเชิงความหมายของประโยค ค่าความสัมพันธ์จะคำนวณจากผลรวมของค่าความสัมพันธ์ของคำในประโยค A กับประโยค B รวมกับผลรวมของค่าความสัมพันธ์ของคำในประโยค B กับประโยค A หารด้วยจำนวนคำทั้งหมดในประโยคทั้งสอง โดยค่าความสัมพันธ์ของคำในประโยค A กับประโยค B คือระยะห่างระหว่างคำในประโยค A กับคำในประโยค B ที่สั้นที่สุด ซึ่งในการคำนวณหาระยะห่างระหว่างคำที่สั้นที่สุดนั้นจะพิจารณาความสัมพันธ์จากฐานข้อมูลในเวิร์ดเนตระดับ Synonym Hyponym Hypernym รวมทั้ง Antonym และ derivation ด้วย โดยจะให้น้ำในความสัมพันธ์ระดับ Antonym และ derivation ในระดับต่ำสุด ถึงแม้ว่าจะพิจารณาความสัมพันธ์เชิงความหมายในระดับประโยคแต่ก็ไม่ได้พิจารณาความสัมพันธ์กันของคำในประโยคนั้น ๆ แต่อย่างใด

Yang and Powers (2005) ได้เสนอวิธีการในการวัดค่าความสัมพันธ์ทางความหมายระหว่างคำจากระยะห่างโหนดโดยอาศัยโครงข่ายในเวิร์ดเนต โดยความสัมพันธ์ที่พิจารณาได้แก่ Synonym, Antonym, Hypernym, Hyponym, Antonym, Holonym และ Meronym คือนอกเหนือจากพิจารณาความสัมพันธ์ “is-a” แล้ว ยังพิจารณาความสัมพันธ์ “part-of”, อีกด้วย มีการให้น้ำหนักความสัมพันธ์ในแบบ Synonym และ Antonym มากกว่าความสัมพันธ์ในแบบอื่นๆ ซึ่งผลการทดลองกับชุดคู่ลำดับของคำมาตรฐาน 28 คู่ โดยการพิจารณาของมนุษย์ปรากฏว่าให้ผลที่ดีกว่าเมื่อเปรียบเทียบกับงานวิจัยที่ผ่านมา

ตารางที่ 2.1

เปรียบเทียบงานวิจัยที่เกี่ยวข้อง

ผู้เขียน	เสนอ	ข้อดี	ข้อเสีย
Hist and St-Ong (1998)	คำนวณค่าความสัมพันธ์เชิงความหมายโดยอาศัยโครงข่ายในเวิร์ดเนต จะพิจารณาจำนวนครั้งของการเปลี่ยนทิศทางของเส้นทาง	ทำได้ง่าย	พิจารณาแค่ความสัมพันธ์ในแบบ Synonym และ Hyponym/Hypernym เท่านั้น
Lesk M. (1986)	นับจำนวนซึ่งปรากฏในคำแปล (gloss) ของคำทั้งสองที่เหมือนกัน (overlap)	ทำได้ง่าย	พิจารณาเฉพาะคำที่เหมือนกันใน gloss เท่านั้น รวมทั้งมีข้อจำกัดในเรื่องคำแปล (gloss) ที่มีอยู่ใน dictionary อาจให้ข้อมูลที่ไม่เพียงพอ
Banerjee and Pederson (2003)	นับจำนวนซึ่งปรากฏในคำแปล (gloss) ของคำทั้งสองที่เหมือนกัน (overlap) โดยมีการให้น้ำหนักตามความยาวของวลีที่ปรากฏด้วย เช่น ปรากฏคำว่า “bank account” เหมือนกัน คะแนนที่ได้จะเป็น 2 2	ไม่พิจารณาเฉพาะ gloss ของ คำนั้น ๆ เท่านั้น แต่ยังพิจารณา gloss ของคำที่เป็น Hyponym, Hypernym, Meronym Holonym and Troponym Synsets ของคำกำหนดอีกด้วย	ไม่ได้พิจารณาคำอื่น ๆ ที่อยู่ในประโยคหรือข้อความที่อยู่รอบ ๆ

ตารางที่ 2.1 (ต่อ)

ผู้เขียน	เสนอ	ข้อดี	ข้อเสีย
Strube and Ponzetto (2006)	ใช้ความหมายหรือคำนิยามของคำ ใน Wikipedia ซึ่งเป็นพจนานุกรมภาษาอังกฤษแบบออนไลน์ในการคำนวณหาค่าความสัมพันธ์ในเชิงความหมาย ซึ่งอาศัยหลักการของ Lesk และ Banerjee, Pederson เช่นกัน โดยมีการ normalize ค่าที่คำนวณได้อีกที	เนื่องจาก Wikipedia เป็น on-line encyclopedia database ที่มีขนาดใหญ่ ทำให้สามารถคำนวณค่าความสัมพันธ์จาก คำแปล (gloss) หรือ full text page ใน Wikipedia ได้	เปรียบเทียบคำที่เหมือนกันเท่านั้น
Yang and Powers (2005)	เสนอการพิจารณาความสัมพันธ์แบบ Synonym, Antonym, Hypernym, Hyponym, Antonym, Holonym และ Meronym คือนอกเหนือจากพิจารณาความสัมพันธ์ “is-a” แล้ว ยังพิจารณาความสัมพันธ์ “part-of”, อีกด้วย มีการให้น้ำหนักความสัมพันธ์	ครอบคลุมความสัมพันธ์ในเชิงความหมายมากขึ้น	เป็นการพิจารณาความสัมพันธ์ในระดับคำเท่านั้น

ตารางที่ 2.1 (ต่อ)

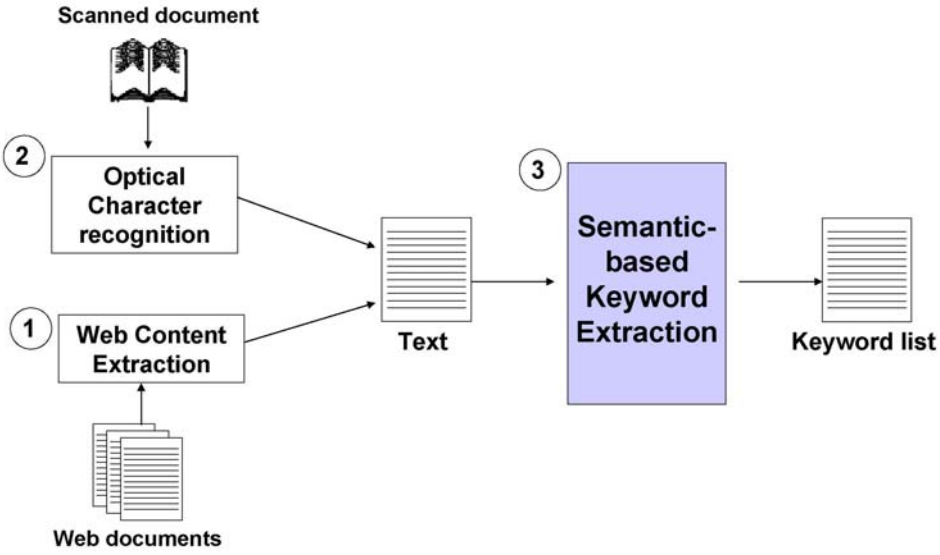
ผู้เขียน	เสนอ	ข้อดี	ข้อเสีย
Zhang, Sun, Wang and Bai (2005)	เสนอการหาประโยคจะสัมพันธ์กับหัวข้อเอกสารโดยการเปรียบเทียบเชิงความหมายของประโยค คำนวณค่าความสัมพันธ์จากผลรวมของค่าความสัมพันธ์ของคำในประโยค A กับประโยค B รวมกัน แล้วหารด้วยจำนวนคำทั้งหมดในประโยคทั้งสอง	เป็นการพิจารณาความสัมพันธ์ในระดับประโยค	ในการเปรียบเทียบระหว่างคำภายในประโยค พิจารณาแค่ความสัมพันธ์ในแบบ Synonym และ Hyponym/Hypernym เท่านั้น และไม่ได้พิจารณาความสัมพันธ์ระหว่างคำในประโยค

จากงานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบหาค่าความสัมพันธ์ในเชิงความหมายนั้นยังมีข้อจำกัด ซึ่งสรุปได้ดังนี้

1. ระดับของการเปรียบเทียบ งานวิจัยส่วนใหญ่ใช้การเปรียบเทียบในระดับคำ เพื่อเปรียบเทียบความเหมือนกันระหว่างคำ แต่เนื่องจากว่าคำหนึ่งคำอาจแทนความหมายได้หลายความหมาย อาจได้ผลลัพธ์ที่ไม่ถูกต้อง
2. ความสัมพันธ์ของคำ งานวิจัยโดยทั่วไปจะเปรียบเฉพาะคำที่มีความหมายเหมือนกัน (Synonym) ในความสัมพันธ์ “is-a” ซึ่งในการใช้งานจริงๆแล้ว ยังมีความสัมพันธ์อื่นๆ มาใช้ในการเปรียบเทียบได้ด้วย เช่น ความสัมพันธ์ “part-of”, ความสัมพันธ์ “member-is-a-kind-of”, ความสัมพันธ์ “member-is-a-part-of” เป็นต้น

บทที่ 3 ระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมาย

ในบทนี้จะกล่าวถึงโครงสร้างของระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมายที่ได้นำเสนอ ซึ่งนำเอาหลักการของการเปรียบเทียบเชิงความหมายมาประยุกต์ใช้ในการค้นหาคำสำคัญ รูปที่ 3.1 แสดงแผนภาพโดยรวมของระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมาย โดยระบบที่นำเสนอนี้สามารถค้นหาคำสำคัญจากเอกสารได้ 2 ลักษณะคือ เอกสารที่อยู่ในรูปของข้อความบนเว็บ และเอกสารที่เป็น Hard copy ที่ได้ผ่านการสแกนจากเครื่องตรวจกราด



รูปที่ 3.1 แผนภาพโดยรวมของระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมาย

เมื่อระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมายได้รับเอกสารสำหรับค้นหาคำสำคัญจากผู้ใช้ ก็จะทำการเตรียมเอกสารเหล่านั้นให้อยู่ในรูปของเอกสารข้อความ โดยผ่านกระบวนการ (1) การเลือกเฉพาะเนื้อหา (Web Content Extraction) ถ้าเอกสารที่เข้ามาเป็นข้อความบนเว็บ หรือ กระบวนการ (2) การรู้จำตัวอักษร (Optical Character Recognition) ถ้าเอกสารที่เข้ามาเป็น Hard copy ที่ได้ผ่านการสแกนจากเครื่องตรวจกราด เมื่อข้อมูลเข้าถูกแปลงเป็นเอกสารข้อความเรียบร้อยแล้ว ก็จะถูกส่งเข้ายังกระบวนการ (3) การค้นหาคำสำคัญโดยใช้ความหมาย (Semantic-based Keyword Extraction) ซึ่งผลลัพธ์ที่ได้จะเป็นรายการของคำสำคัญของเอกสารนั้น โดยคำสำคัญเหล่านี้คาดหมายว่าจะมีความหมายสอดคล้องกับเนื้อหาโดยรวมของเอกสาร

ในส่วนต่อไปจะกล่าวถึงรายละเอียดของกระบวนการต่างๆของระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมาย

(1) การเลือกเฉพาะเนื้อหา (Web Content Extraction)

ในปัจจุบัน เอกสารที่สำคัญต่างๆสามารถถูกเรียกสืบค้นได้โดยผ่านทางเครือข่ายโดยโปรแกรมเว็บบราวเซอร์ (Web Browser) ซึ่งเอกสารเหล่านั้นส่วนใหญ่จะอยู่ในรูปของ HTML (Hyper Text Mark-up Language) ซึ่งประกอบด้วย แถบป้าย (tag) มากมาย ดังแสดงเป็นตัวอย่างตามตาราง 3.1

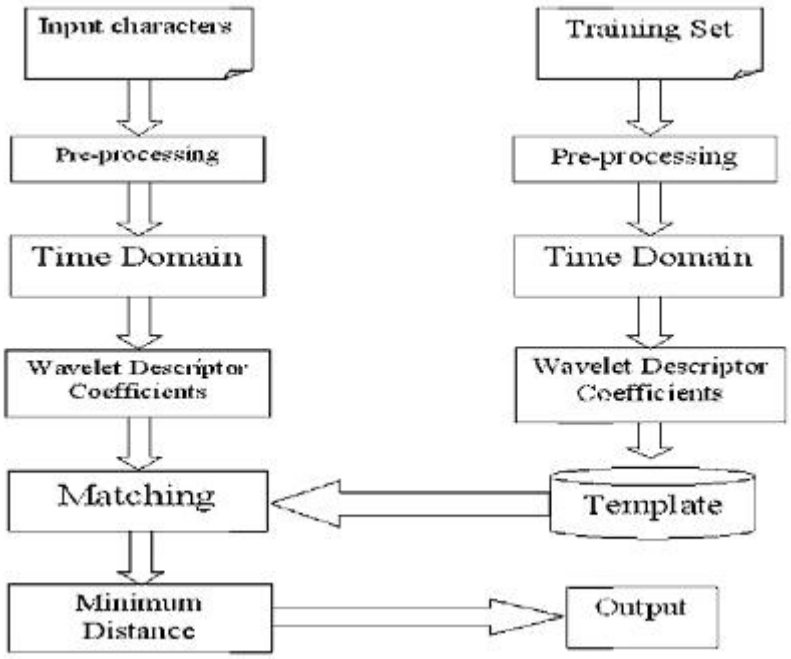
ตารางที่ 3.1 ตัวอย่างของแถบป้ายที่ใช้ในแฟ้มข้อมูลประเภท HTML

Tag	Description	DTD
<!--...-->	Defines a comment	STF
<!DOCTYPE>	Defines the document type	STF
<abbr>	Defines an abbreviation	STF
<acronym>	Defines an acronym	STF
<applet>	Defines an applet	TF
<area>	Defines an area inside an image map	STF
<basefont>	Defines a base font	TF
<bdo>	Defines the direction of text display	STF
<blockquote>	Defines a long quotation	STF
<body>	Defines the body element	STF
 	Inserts a single line break	STF
<caption>	Defines a table caption	STF
	Defines centered text	TF
<colgroup>	Defines groups of table columns	STF
	Defines a section in a document	STF
<dl>	Defines a definition list	STF
<dt>	Defines a definition term	STF
<font>	Defines text font, size, and color	TF
<h1> to <h6>	Defines header 1 to header 6	STF
<head>	Defines information about the document	STF
<hr>	Defines a horizontal rule	STF
<map>	Defines an image map	STF
<menu>	Defines a menu list	TF
<object>	Defines an embedded object	STF
<p>	Defines a paragraph	STF
<q>	Defines a short quotation	STF
<script>	Defines a script	STF
<select>	Defines a selectable list	STF
<small>	Defines small text	STF
	Defines a section in a document	STF
<style>	Defines a style definition	STF
<table>	Defines a table	STF
<tbody>	Defines a table body	STF
<td>	Defines a table cell	STF
<textarea>	Defines a text area	STF
<tfoot>	Defines a table footer	STF
<th>	Defines a table header	STF
<thead>	Defines a table header	STF
<title>	Defines the document title	STF
<tr>	Defines a table row	STF

ดังนั้น กระบวนการนี้จะเป็นการเลือกเอาเฉพาะส่วนของเนื้อความออกมาจากเอกสาร กระบวนการนี้จะอ่านเอกสารจำนวน 2 รอบ โดยรอบที่ 1 จะอ่านเอกสารเพื่อข้ามส่วนของ ตาราง เมนู รูปภาพ และ การกำหนดลักษณะพิเศษ จากนั้นบันทึกข้อมูลเก็บไว้ ในรอบที่ 2 จะเป็นการอ่านเพื่อตัดแถบป้ายออก ให้เหลือเฉพาะ ส่วนของเนื้อความ และจัดเก็บเป็นแฟ้มข้อมูลชนิดที่เป็นข้อความล้วน (Text file)

(2) การรู้จำตัวอักษร (Optical Character Recognition)

ในกรณีที่เอกสารที่รับเข้ามาอยู่ในรูปของ Hard Copy กระบวนการนี้จะทำหน้าที่ในการแปลงเอกสารที่อยู่ในรูปของภาพให้เป็นเอกสารข้อความ โดยจะใช้หลักการของรู้จักตัวอักษรบนโดเมนความถี่ กระบวนการนี้ได้ประยุกต์ใช้วิธีการของ (Chey, C. et al. 2006) ที่เลือกใช้ Wavelet descriptor มาเป็นเทคนิคสำคัญในการสร้างแม่แบบ (template) ของแต่ละตัวอักษร รายละเอียดของขั้นตอนมีดังนี้



รูปที่ 3.2 ขั้นตอนการรู้จำตัวอักษร (Optical Character Recognition)

รูปที่ 3.2 แสดงถึงแผนภาพของระบบรู้จำตัวอักษร ประยุกต์มาจากการรู้จำภาษาเขมรโดยใช้ Wavelet Descriptor (Percival, D. B. and Walden, A. T.: 2000) ทางด้านขวามือของรูป ชุดของตัวอักษรสำหรับการฝึกฝน (Training Set) ถูกนำมาใช้ในการสร้างแม่แบบ (Template) ซึ่ง

จัดเก็บไว้ในรูปแบบของค่าเฉลี่ยของสัมประสิทธิ์ของ Wavelet Descriptor ซึ่งคำนวณมาจากข้อมูลประเภทที่สามารถคำนวณได้ภายในช่วงเวลา (Temporal Data) ของแต่ละตัวอักษรของชุดของการฝึกฝน เมื่อแม่แบบของตัวอักษรแต่ละตัวได้ถูกสร้างขึ้น ก็จะถูกนำมาเปรียบเทียบกับ สัมประสิทธิ์ของ Wavelet Descriptor ของภาพตัวอักษรที่ได้รับการอินพุตเข้ามา ถ้าแม่แบบของตัวอักษรตัวใดมีค่าระยะห่างของความแตกต่าง (Minimum Distance) น้อยที่สุด ก็จะเป็นคำตอบของการรู้จำตัวอักษรนั้นๆ รายละเอียดของระบบรู้จำตัวอักษร มีดังนี้

(1) การเตรียมข้อมูล (Pre-processing)

ในขั้นตอนนี้ ภาพตัวอักษรที่ได้มาจากการสแกนจะถูกนำมาเตรียมด้วยการแปลงเป็นรูปขาวดำ โดยมีการกำหนดระดับเงื่อนไข (Threshold) ค่าความเข้มของแต่ละจุดของภาพ (Pixel) จะถูกนำมาเปรียบเทียบกับระดับเงื่อนไขนี้ ถ้าค่าความเข้มของแต่ละจุดของภาพ สูงกว่า ระดับเงื่อนไขจะถูกกำหนดให้เป็นสีดำ และในทางตรงกันข้ามจะเป็นสีขาว เมื่อได้รูปภาพตัวอักษรเป็นสีขาว-ดำแล้ว รูปภาพเหล่านี้จะถูกแยกออกเป็นตัวอักษรแต่ละตัว โดยการหาโครงร่างของแต่ละตัวอักษร เรียกว่า Skeletonization เมื่อแยกตัวอักษรออกเป็นแต่ละตัวได้แล้ว ก็จะถูกนำไปหารูปร่างของตัวอักษรด้วยการทำให้บางลง (Thinning) รูปภาพตัวอักษรแต่ละตัวจะถูกส่งไปยังขั้นตอนการดึงเอาลักษณะเด่นของตัวอักษร ต่อไป

(2) การดึงเอาลักษณะเด่นของตัวอักษร (Feature Extraction)

ขั้นตอนนี้ถือว่าสำคัญที่สุดในการรู้จำตัวอักษร เพราะลักษณะเด่นของตัวอักษรจะนำไปสู่การรู้จำตัวอักษรที่มีความถูกต้องสูง งานวิจัยนี้จะเริ่มต้นด้วย

- การแปลงโครงร่างของตัวอักษรที่ได้จากขั้นตอนการเตรียมข้อมูลให้อยู่ในรูปของ Temporal Domain ด้วยสมการที่ (3-1)

$$r(t) = \sqrt{(X(t) - cg_x)^2 + (Y(t) - cg_y)^2} \qquad (3\text{-}1)$$

เมื่อ r(t) คือ โครงร่างของตัวอักษรแต่ละตัวที่อยู่ในรูปของ Temporal domain
X(t), Y(t) คือ โครงร่างของตัวอักษรแต่ละตัวที่อยู่ในรูปของ Spatial domain
cg_x, cf_y คือ ตำแหน่งศูนย์กลางของตัวอักษร

- นำเอาโครงร่างของตัวอักษรทั้งหมดที่อยู่ในรูปของ Temporal Domain มาเข้าฟังก์ชั่น Wavelet Descriptor (Chey, C. et al. 2006) เพื่อหาลักษณะเด่นของแต่ละตัวอักษรซึ่งอยู่ในรูปของสัมประสิทธิ์เวฟเล็ท ที่สามารถหาลักษณะเด่นได้ทั้งแบบ global และ local และเก็บเป็นแม่แบบ (template) ของแต่ละตัวอักษร

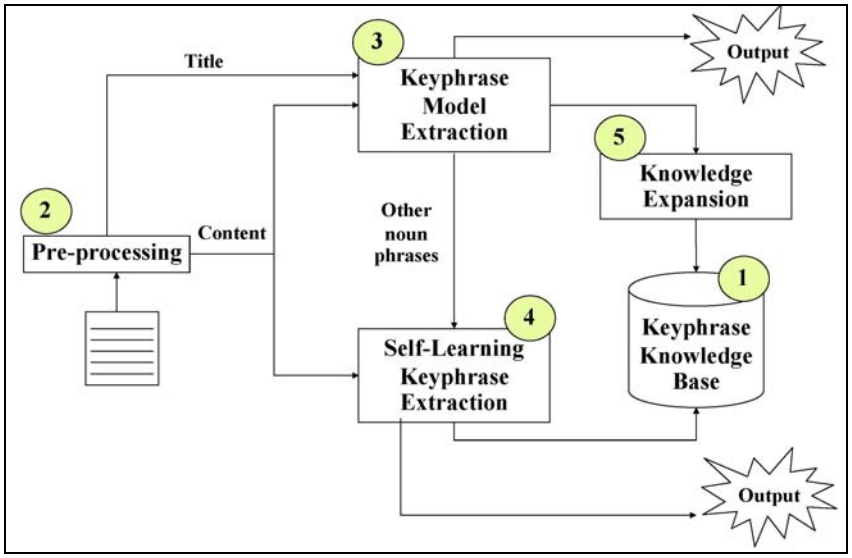
(3) การเปรียบเทียบ (Matching Process)

ขั้นตอนนี้จะเป็นการรู้จำตัวอักษรที่ใช้ในการทดสอบ ซึ่งผ่านการเตรียมข้อมูล และ ดึงลักษณะเด่นของตัวอักษร เหมือนกับตัวอักษรที่ใช้ฝึกฝน จะถูกนำมาเปรียบเทียบ กับ

แม่แบบที่ผ่านการฝึกฝนมาก่อนหน้านี้ การเปรียบเทียบใช้หลักการของ Minimum Euclidean Distance แม่แบบของตัวอักษรที่ให้ค่าระยะทางน้อยที่สุด คือคำตอบของตัวอักษรนั้นด้วยวิธีการรู้จำตัวอักษรที่ใช้ในงานวิจัยนี้ สามารถให้ความถูกต้องได้สูงสุดถึง 92.99%

(3) การค้นหาคำสำคัญโดยใช้ความหมาย (Semantic-based Keyword Extraction)

ขั้นตอนนี้ถือว่าเป็นหัวใจของงานวิจัยนี้ ซึ่งเสนอการค้นหาคำสำคัญจากตัวเอกสารเพียงอย่างเดียว ไม่มีการใช้คลังข้อมูลสำหรับฝึกฝน (Kongkachandra, R., et al.: 2006) ดังนั้นความหมายของเอกสารที่งานวิจัยนี้กำหนด จะมาจากประโยคต่างๆที่มี คำสำคัญ ปรากฏอยู่ คำสำคัญจะถูกดึงออกมาจากเอกสารด้วยการเปรียบเทียบระหว่างความหมายของ คำที่พิจารณา (Candidate) กับ คำที่เป็นตัวแทนของเอกสารซึ่งจะได้มาจากการพิจารณาจากหัวเรื่อง ดังนั้น ในงานวิจัยนี้จะมีข้อจำกัดตรงที่เอกสารที่นำมาใช้สำหรับค้นหาคำสำคัญจะต้องมีส่วนของหัวเรื่องที่สื่อความหมายของเอกสารทั้งหมดได้ รูปที่ 3.3 แสดงถึงแผนผังรวมของการค้นหาคำสำคัญโดยใช้ความหมาย



รูปที่ 3.3 แผนผังรวมของระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมาย

ส่วนประกอบต่างของระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมาย ที่แสดงในรูปที่ 3.3 มีดังนี้

1. ฐานความรู้ของคำสำคัญ (Keyword Knowledge Base)

ฐานความรู้ของคำสำคัญ ถูกออกแบบขึ้นมาเพื่อใช้สำหรับเก็บคำสำคัญ ความหมาย และกลุ่มของเอกสาร (Domain) ที่คำสำคัญนั้นสังกัดอยู่ ฐานความรู้ของคำสำคัญที่ใช้ในงานวิจัยนี้เป็น

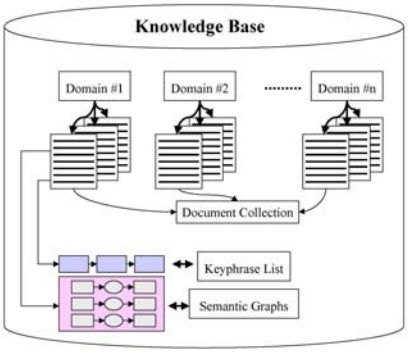
แบบที่ไม่เฉพาะเจาะจงกับกลุ่มของเอกสารกลุ่มใดกลุ่มหนึ่ง และสามารถเพิ่มเติมได้แบบอัตโนมัติ แต่อย่างไรก็ตาม ตอนที่เริ่มต้นสร้างฐานความรู้ของคำสำคัญนั้น ระบบจำเป็นต้องให้ผู้ใช้เป็นผู้กำหนดสิ่งต่างๆ เหล่านี้

(1) จำนวนกลุ่มของเอกสาร (Number of Domains)

(2) คำศัพท์ควบคุม (Controlled Vocabulary) ของแต่ละกลุ่ม ประมาณ 8-10 คำศัพท์ต่อกลุ่ม

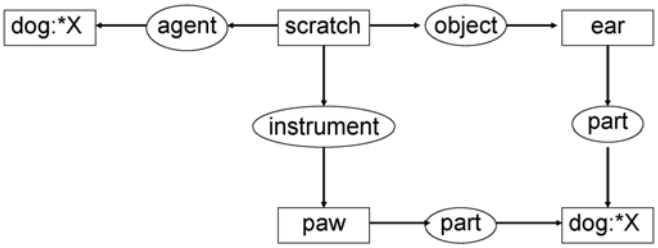
(3) โครงข่ายทางความหมายของกลุ่มของเอกสาร (Domain Hierarchy)

เมื่อระบบอัตโนมัติค้นหาคำสำคัญเชิงความหมายทำงานในแต่ละรอบ ฐานความรู้ของคำสำคัญจะถูกอัพเดทด้วยคำสำคัญ 2 ประเภทด้วยกัน ชนิดแรกคือ คำสำคัญเริ่มต้น (Initial keyword) ที่ได้จากการพิจารณาที่หัวเรื่องของเอกสาร ซึ่งได้มาจากส่วนของการดึงคำสำคัญเริ่มต้น (Initial Keyword Extraction) และ ชนิดที่สองคือ คำสำคัญที่มีความหมายเกี่ยวข้อง (Relative keyword) ที่ได้จากการพิจารณาที่เนื้อความของเอกสาร ซึ่งได้มาจากส่วนของการดึงคำสำคัญที่เกี่ยวข้องเชิงความหมาย (Semantic-Relatedness Keyword Extraction)



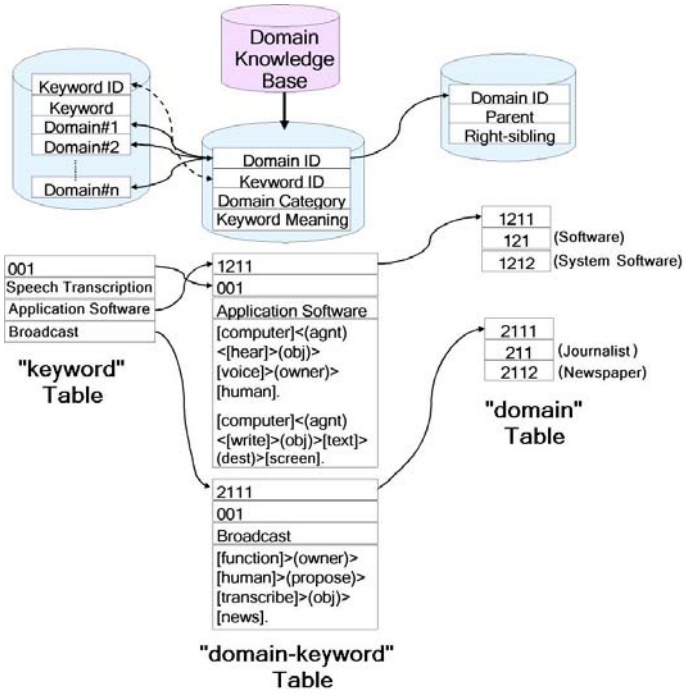
รูปที่ 3.4 แสดงถึงการจัดเก็บข้อมูลภายในฐานความรู้ของคำสำคัญ

รูปที่ 3.4 แสดงถึงการจัดเก็บข้อมูลภายในฐานความรู้ของคำสำคัญ จะเห็นว่าภายในฐานความรู้ของคำสำคัญจะแบ่งออกเป็น กลุ่มของเอกสาร ซึ่งภายในแต่ละกลุ่มจะรวบรวมเอกสารที่เกี่ยวข้องกับกลุ่มนั้นๆเข้าด้วยกัน และภายในแต่ละเอกสารจะเก็บ รายการคำสำคัญ ที่ได้จากการทำงานของระบบอัตโนมัติค้นหาคำสำคัญก่อนหน้านี้ พร้อมกันนี้ แต่ละคำสำคัญจะเก็บความหมายเอาไว้ด้วย ในงานวิจัยนี้ ความหมายของคำสำคัญ จะถูกเก็บในรูปแบบของ กราฟเชิงความคิด (Conceptual Graph) ซึ่งถูกกำหนดขึ้นเมื่อปี ค.ศ. 1984 โดยนักคณิตศาสตร์ชื่อจอห์น โซวะ (Sowa, J.F.: 1984) กราฟเชิงความคิด เป็นกราฟชนิด Bipartite ที่ประกอบไปด้วยโนด 2 ประเภทคือ Conceptual Entity และ Conceptual Relation รูปที่ 3.5 แสดงถึงการใช้กราฟเชิงความคิดแทนความหมายของประโยค “The dog scratches its ear with its paw”

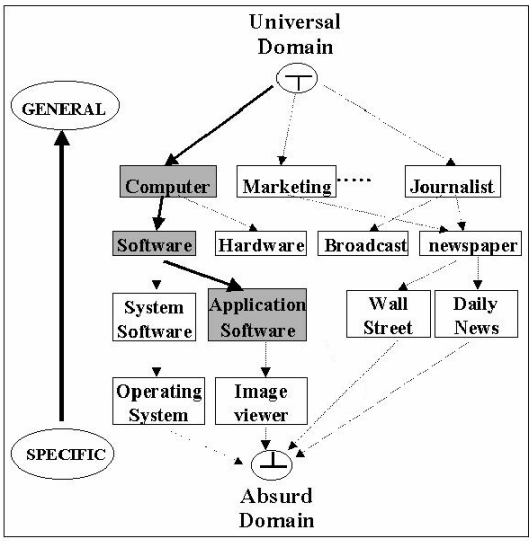


รูปที่ 3.5 การใช้กราฟเชิงความคิดแทนความหมายของ
ประโยค "The dog scratches its ear with its paw"

ข้อมูลที่อยู่ในกรอบสี่เหลี่ยมเรียกว่า Concept Node ซึ่งจะเก็บคำ โดยที่คำเหล่านี้อาจจะเป็น คำนาม คำกริยา หรือ คำคุณศัพท์ ส่วนข้อมูลที่อยู่ในวงกลมเรียกว่า Conceptual Relation Node ซึ่งจะเป็นความสัมพันธ์ระหว่าง Concept Node ที่เชื่อมกัน ความหมายของกราฟนี้ คือ "สุนัข (dog) เป็นประธาน (agent) ของการเกา (scratch) ซึ่งกรรม (object) ของการเกา (scratch) คือหู (ear) เครื่องมือ (instrument) ที่ใช้ในการเกา (scratch) คือเล็บ (paw) และ หูก็เป็นส่วนหนึ่ง (part) ของสุนัข"



(ก)



(ข)

รูปที่ 3.6 แสดงถึงการจัดเก็บข้อมูลภายในฐานความรู้คำสำคัญ

รูปที่ 3.4 แสดงถึงโครงร่างโดยคร่าวของฐานความรู้ของคำสำคัญ รายละเอียดในการจัดเก็บภายในฐานความรู้ของคำสำคัญ ดังแสดงในรูปที่ 3.6 ซึ่งประกอบไปด้วย ตารางความสัมพันธ์ 3 ตารางคือ ตารางคำสำคัญ (Keyword Table) ตารางกลุ่มของเอกสาร-คำสำคัญ (Domain-Keyword Table) และ ตารางกลุ่มของเอกสาร (Domain Table) ซึ่งสอดคล้องกับโครงข่ายของกลุ่มเอกสารในรูปที่ 3.6 (ข) ตัวอย่างคือ ในตารางคำสำคัญทางซ้ายมือสุดของรูปที่ 3.6 ก. คำสำคัญคือ “Speech Technology” (มีหมายเลข 001) ซึ่งสามารถปรากฏอยู่ในกลุ่มเอกสารได้ 2 ด้านคือ “Application Software” และ “Broadcast” ซึ่งมีลิงค์เชื่อมไปที่ ตารางกลุ่มของเอกสาร-คำสำคัญที่อยู่ตรงกลาง เนื่องจากงานวิจัยนี้เสนอการค้นหาคำสำคัญได้ในหลายกลุ่มของเอกสาร ดังนั้นเราจำเป็นต้องจัดระเบียบความสัมพันธ์ของกลุ่มต่างๆเอาไว้ ซึ่งเป็นไปตาม Semantic Web ที่ชื่อว่า WordNet (Teich, E. and Fankhauser, P.: 2004) และ (Yang, D., and Powers, D.,: 2005) ดังแสดงเป็นบางส่วนในรูปที่ 3.6 (ข)

2. ส่วนเตรียมข้อมูล (Preprocessing)

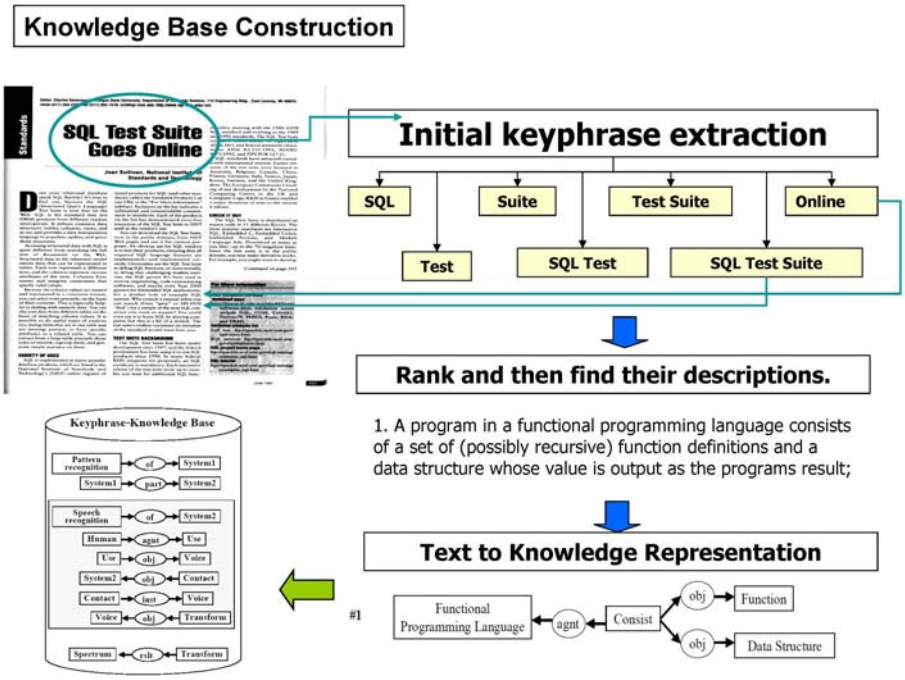
จากที่กล่าวไว้ว่า ฐานความรู้ของคำสำคัญ จะถูกอัพเดทด้วย คำสำคัญ 2 ประเภท ซึ่งได้มาจากการพิจารณาเอกสารคนละส่วน นั่นคือ ส่วนของหัวเรื่อง และ ส่วนของเนื้อความ ดังนั้น ในขั้นตอนของการเตรียมข้อมูล จะดำเนินการดังนี้

(1) แยกส่วนของหัวเรื่องออกจากเนื้อความ โดยใช้กฎที่ระบุว่า ตำแหน่งของประโยคในบรรทัดแรกของเอกสารจะเป็นหัวเรื่องของเอกสารนั้น

(2) ในส่วนของเนื้อความ ระบบจะแบ่งเอกสารออกเป็น หนึ่งประโยค/หนึ่งบรรทัด ซึ่งเกณฑ์ที่ใช้ในการตัดประโยคคือพิจารณาที่ ตัวแบ่งประโยค (Punctuation) เช่น จุลภาค เครื่องหมายหยุด เครื่องหมายคำถาม เป็นต้น

3. ส่วนของการดึงคำสำคัญเริ่มต้น (Initial Keyword Extraction)

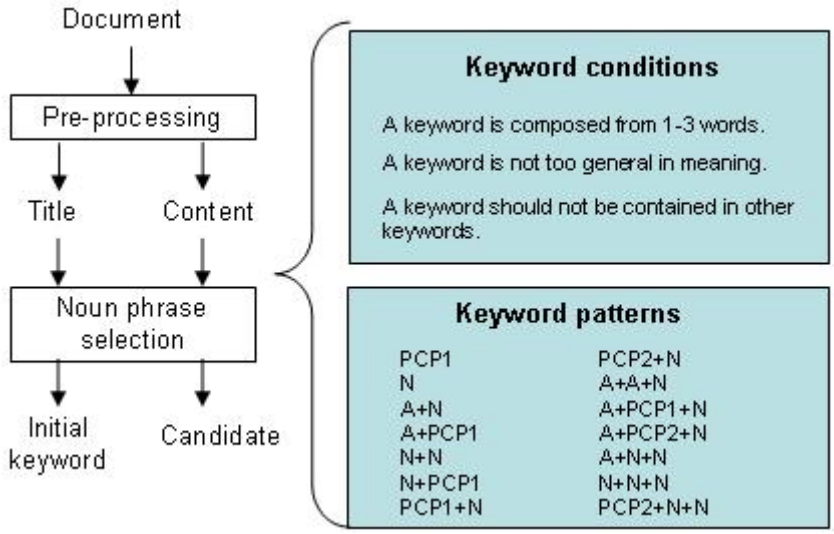
งานวิจัยนี้เสนอการดึงความสำคัญจากเอกสาร โดยการเปรียบเทียบกับความหมายของเอกสาร แต่เนื่องจากงานวิจัยนี้เสนอที่จะพิจารณาเพียงเอกสารนั้นๆในการดึงคำสำคัญออกมา ดังนั้นเราจำเป็นต้องเตรียมข้อมูลที่เป็นความหมายของเอกสาร M. Montes-y Gme และคณะ (M. Montes-y Gmez et al., 1999) ได้วิจัยไว้ว่า หัวเรื่องของเอกสาร ถือเป็นแหล่งอธิบายความหมายที่ดีของเอกสาร ดังนั้น หัวเรื่องของเอกสารจึงถูกใช้เป็นแหล่งข้อมูลที่จะใช้ในการดึงคำสำคัญเริ่มต้น โดยมีหลักการในการดำเนินการ ดังแสดงในรูป 3.7



รูปที่ 3.7 หลักการของการดึงคำสำคัญเริ่มต้น (Initial Keyword Extraction)

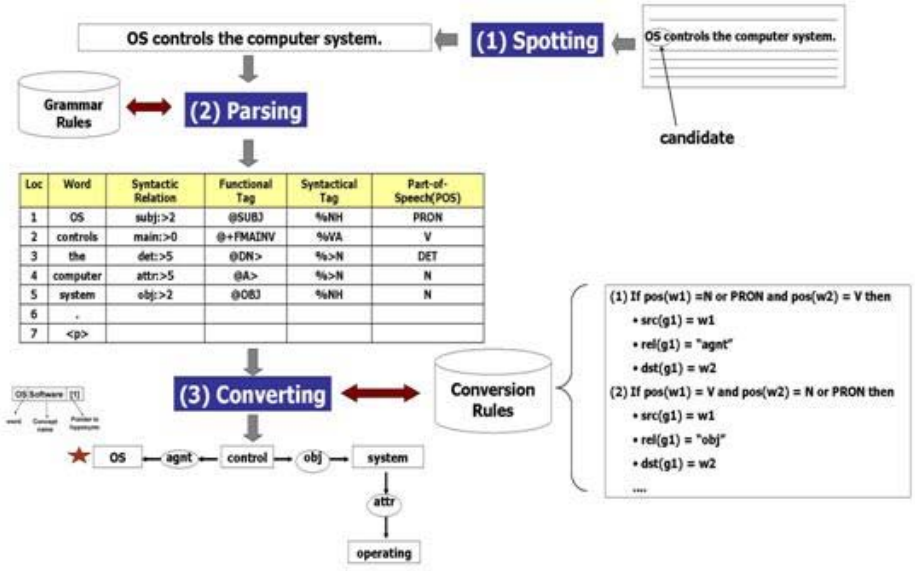
(1) หัวเรื่องของเอกสารจะถูกส่งไปยังขั้นตอนคัดเลือกคำคู่แข่ง (Candidate Selecting) เพื่อค้นหาคำสำคัญเริ่มต้น ในขั้นตอนนี้ หัวเรื่องของเอกสารจะถูกแจกแจงไวยากรณ์ เพื่อให้ทราบหน้าที่ของคำ (Part-of-speech) จากนั้นจะคัดเลือกเฉพาะคำที่ไม่อยู่ในรายการของคำหยุด (Stop-word list) และมีบทบาทเป็น นามวลี เท่านั้น โดยเปรียบเทียบกับรูปแบบของนามวลี ดังแสดงในรูปที่ 3.8 และรายการของคำหยุดที่กำหนดไว้ อย่างไรก็ตาม จากหัวเรื่องเดียวกัน เราสามารถดึงคำสำคัญเริ่มต้น ได้หลายคำ ซึ่งอาจจะมีส่วนที่ซ้อนทับกันอยู่ เช่น

“SQL”, “Test”, “Suite”, “Test Suite”, “SQL Test” และ “SQL Test Suite” (ดังตัวอย่างในรูป 3.7) งานวิจัยนี้ จะเลือกนามวลีที่ประกอบด้วยคำเดี่ยวตั้งแต่ 2 คำขึ้นไป ดังนั้น “Test Suite”, “SQL Test” และ “SQL Test Suite” จะถูกเลือกเป็นคำสำคัญเริ่มต้น สำหรับ “Online” นั้นก็จะถูกเลือกด้วย เนื่องจากไม่ซ้อนทับกับคำใด



รูปที่ 3.8 วิธีการสำหรับคัดเลือกนามวลี

จากข้อ (1) เราจะได้คำสำคัญเริ่มต้นออกมาจากเอกสาร ในขั้นตอนนี้ จะดึงเอาความหมายของแต่ละคำสำคัญออกมา แล้วจัดเก็บใน ฐานความรู้คำสำคัญ เพื่อใช้งานในระยะต่อไป ความหมายของคำสำคัญ ที่ระบุในงานวิจัยนี้ คือ ประโยคต่างๆ ที่อยู่ในเอกสาร ที่มีคำสำคัญปรากฏอยู่ ดังนั้น ส่วนของเนื้อความเอกสารจะถูกส่งมาเพื่อเข้าสู่ขั้นตอนการสร้างฐานความรู้ (Knowledge Generating) โดยใช้ คำสำคัญ ที่ได้จากข้อที่ (1) เป็นตัวชี้ตำแหน่งของประโยคเหล่านั้น จากนั้น ประโยคที่ดึงออกมาจะถูกนำมาแจกแจงไวยากรณ์ (Parsing) และแปลงให้อยู่ในรูปของกราฟเชิงความคิดโดยกฎในการแปลงที่กำหนด (Conversion Rules) ซึ่งกราฟเชิงความคิดที่ได้นี้ ต่อไปเรียกว่า **กราฟความหมาย (Semantic Graph)** วิธีการในการแปลงจากประโยคภาษาอังกฤษ เป็น กราฟความหมาย ดังแสดงในรูปที่ 3.9



รูปที่ 3.9 วิธีการในการแปลงจากประโยคภาษาอังกฤษ เป็น กราฟความหมาย

(2) คำสำคัญที่ได้ พร้อมกับกราฟความหมายที่ได้จาก (2) และ (3) จะถูก **ส่วนของการขยายฐานความรู้คำสำคัญ (Knowledge Expansion)** เพื่อเพิ่มเติมเข้าไปในฐานความรู้ของคำสำคัญ

4. ส่วนของการดึงคำสำคัญที่เกี่ยวข้องเชิงความหมาย (Semantic-Relatedness Keyword Extraction)

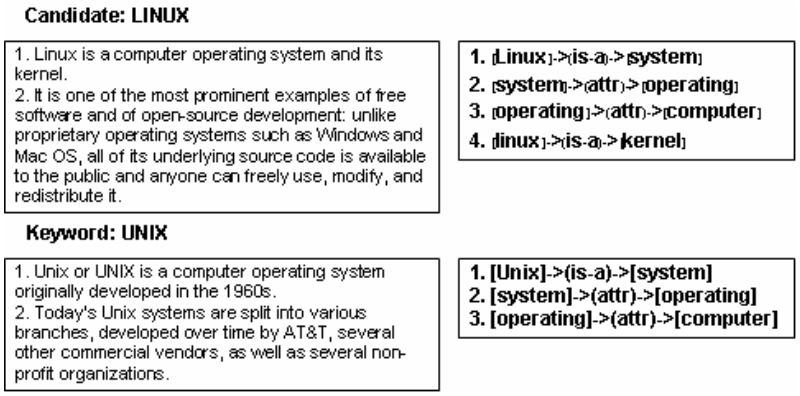
ฐานความรู้ของคำสำคัญที่ได้จากส่วนของการดึงคำสำคัญเริ่มต้น มีจำนวนไม่มากพอที่จะใช้ในการประยุกต์ใช้งาน ดังนั้น ขั้นตอนในส่วนนี้เป็นส่วนที่ดึงคำสำคัญเพิ่มเติม โดยพิจารณาจากคำคู่แข่งที่มีความหมายเกี่ยวข้องกับคำสำคัญเริ่มต้นที่อยู่ในฐานความรู้สำคัญ ขั้นตอนนี้ระบบจะเรียนรู้ด้วยตนเองด้วยการเปรียบเทียบความหมายของคำคู่แข่งแต่ละคำกับคำสำคัญเริ่มต้น คำคู่แข่งใดที่มีคะแนนที่ได้จากการวัดความเหมือนกันทางความหมายกับคำสำคัญเริ่มต้น แล้วมีค่าสูงกว่า ค่าเงื่อนไข (Threshold) ที่กำหนด เราจะถือว่าคำนั้นเป็นคำสำคัญด้วย เรียกว่า **คำสำคัญที่เกี่ยวข้องเชิงความหมาย** (Semantic-Related Keyword)

ขั้นตอนการทำงานใน 2 ขั้นตอนแรก ของส่วนนี้จะคล้ายกับของส่วนของการดึงความสำคัญเริ่มต้น มีที่แตกต่างกันคือในขั้นตอนที่ (1) ดังนี้

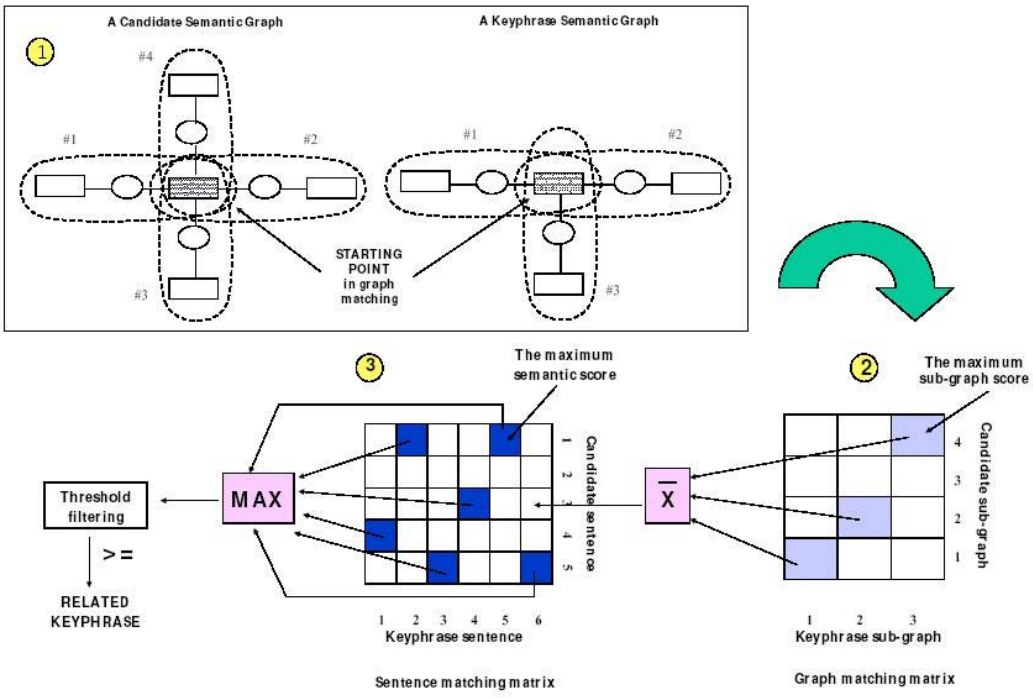
(1) เปลี่ยนจากหัวเรื่องของเอกสารเป็นเนื้อความของเอกสาร จะถูกส่งไปยังขั้นตอนคัดเลือกคำคู่แข่ง (Candidate Selecting) เพื่อค้นหานามวลี ซึ่งนามวลีใดมีการซ้อนทับกันแล้ว นามวลีที่มีจำนวนคำเดี่ยวมากกว่า 2 คำขึ้นไปจะถูกเลือกไว้พิจารณา

(2) ทำในลักษณะเดียวกันกับส่วนของการดึงคำสำคัญเริ่มต้น

(3) คำคู่แข่งพร้อมความหมายทั้งหมดที่ได้จากขั้นตอนข้างบนจะถูกส่งเข้าไปในขั้นตอนการวัดความเหมือนกันทางความหมาย (Semantic Similarity Scoring) คำคู่แข่งใดมีคะแนนสูงกว่าค่าเงื่อนไข (Threshold) ที่กำหนด เราจะถือว่าคำนั้นเป็นคำสำคัญด้วย เรียกว่า **คำสำคัญที่เกี่ยวข้องเชิงความหมาย** (Semantic-Related Keyword) วิธีการในการวัดความเหมือนกันทางความหมาย แสดงในรูปที่ 3.10 และ 3.11



รูปที่ 3.10 กราฟความหมายของคำสำคัญ “UNIX” และ คำคู่แข่ง “LINUX”

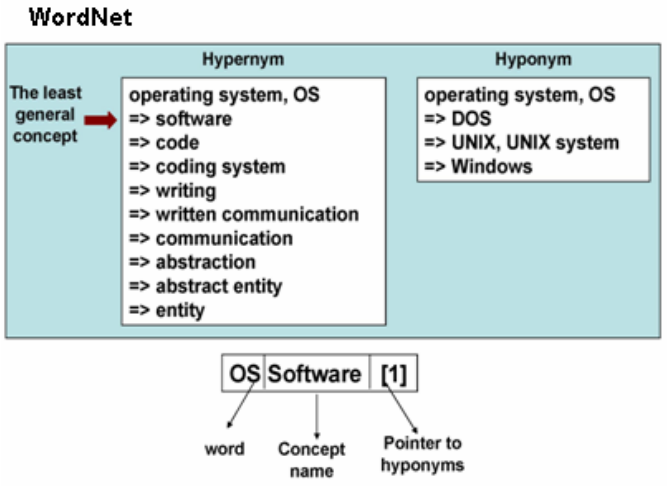


รูปที่ 3.11 การวัดความเกี่ยวข้องทางความหมาย

ในรูปที่ 3.10 คำสำคัญ คือ “UNIX” และมีประโยคที่มีคำว่า “UNIX” ปรากฏอยู่ 2 ประโยค ซึ่งเมื่อผ่านขั้นตอนการแปลงเป็นกราฟความหมายแล้ว จะได้กราฟความหมายทั้งหมด 3 กราฟ ดังแสดงทางขวามือของรูป ส่วนคำคู่แข่ง ที่นำมาพิจารณา คือคำว่า “LINUX” ซึ่งมีประโยคที่คำนี้

ปรากฏอยู่ 2 ประโยค และเมื่อผ่านขั้นตอนการแปลงเป็นกราฟความหมายแล้ว จะได้กราฟความหมายทั้งหมด 4 กราฟ

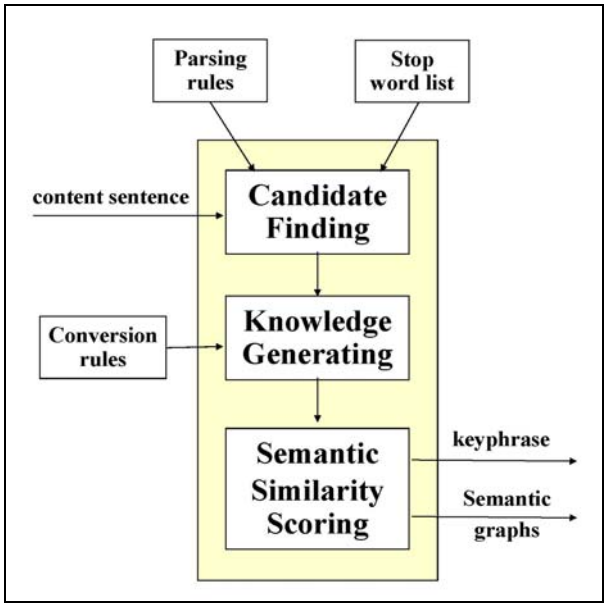
กราฟความหมายของคำสำคัญ 3 กราฟ และ กราฟความหมายของคำคู่แข่ง4 กราฟ จะถูกนำมาเปรียบเทียบดังแสดงในรูปที่ 3.11 การเปรียบเทียบเริ่มที่ โนด ที่มีจำนวนลิงค์มากที่สุด ซึ่งเรียกว่า Core Node ของทั้งสองฝ่าย การเปรียบเทียบจะทำทีละกราฟย่อย ซึ่งประกอบไปด้วย 2 Concept node และ 1 Conceptual Relation node ซึ่งข้อมูลในโนดประกอบด้วย คำ (word) ชื่อกลุ่มของคำ (concept name) และ ลิงค์ไปยังกลุ่มของคำที่เฉพาะเจาะจงกว่า (Pointer to hyponyms) ตัวอย่างของการเก็บข้อมูลในแต่ละโนด ดังแสดงในรูปที่ 3.12 โนดที่แสดงเป็นตัวอย่างเก็บคำว่า “OS” ซึ่งมี ชื่อกลุ่มของคำเป็น “Software” และ ลิงค์ไปยังกลุ่มของคำที่เฉพาะเจาะจงกว่า ซึ่งได้แก่ “DOS”, “UNIX” และ “Windows”



รูปที่ 3.12 การเก็บค่าข้อมูลในแต่ละโนดของกราฟความหมาย

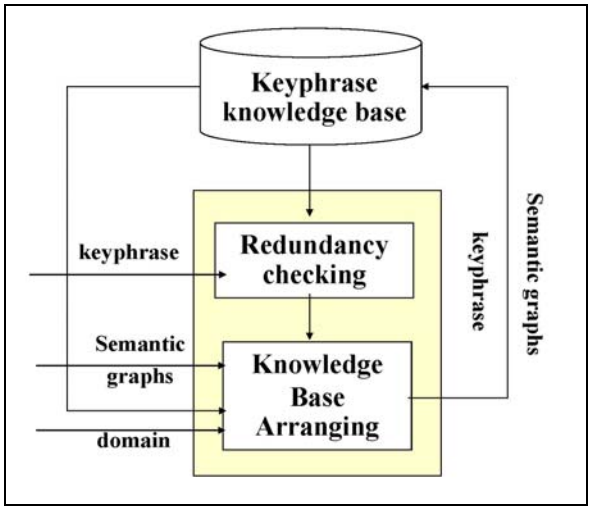
ในการเปรียบเทียบแต่ละกราฟย่อยจะได้ กราฟย่อยที่มีคะแนนสูงสุด ซึ่งจะเก็บไว้ใน เมตริกซ์ของการเปรียบเทียบกราฟ (Graph Matching Matrix) จากนั้น คะแนนของกราฟย่อยทั้งหมดของแต่ละประโยค จะถูกนำมาหาค่าเฉลี่ย และเก็บไว้ใน เมตริกซ์ของการเปรียบเทียบประโยค (Sentence Matching Matrix) คำคู่แข่งใดที่มีคะแนนของการเปรียบเทียบประโยคที่มากที่สุด จะถูกนำมาเปรียบเทียบกับค่าเงื่อนไข ซึ่งคำที่มีค่าสูงกว่าค่าเงื่อนไข ก็จะถูกยอมรับว่าเป็นคำสำคัญที่เกี่ยวข้องทางความหมาย

(4) คำสำคัญที่ได้ พร้อมกับกราฟความหมายที่ได้จาก (3) จะถูกส่งไปยัง **ส่วนของการขยายฐานความรู้คำสำคัญ (Knowledge Expansion)** เพื่อเพิ่มเติมเข้าไปในฐานความรู้ของคำสำคัญ รูปที่ 3.13 แสดงถึงขั้นตอนการดึงคำสำคัญที่มีความเกี่ยวข้องทางความหมาย



รูปที่ 3.13 แสดงถึงขั้นตอนการดึงคำสำคัญที่มีความเกี่ยวข้องทางความหมาย

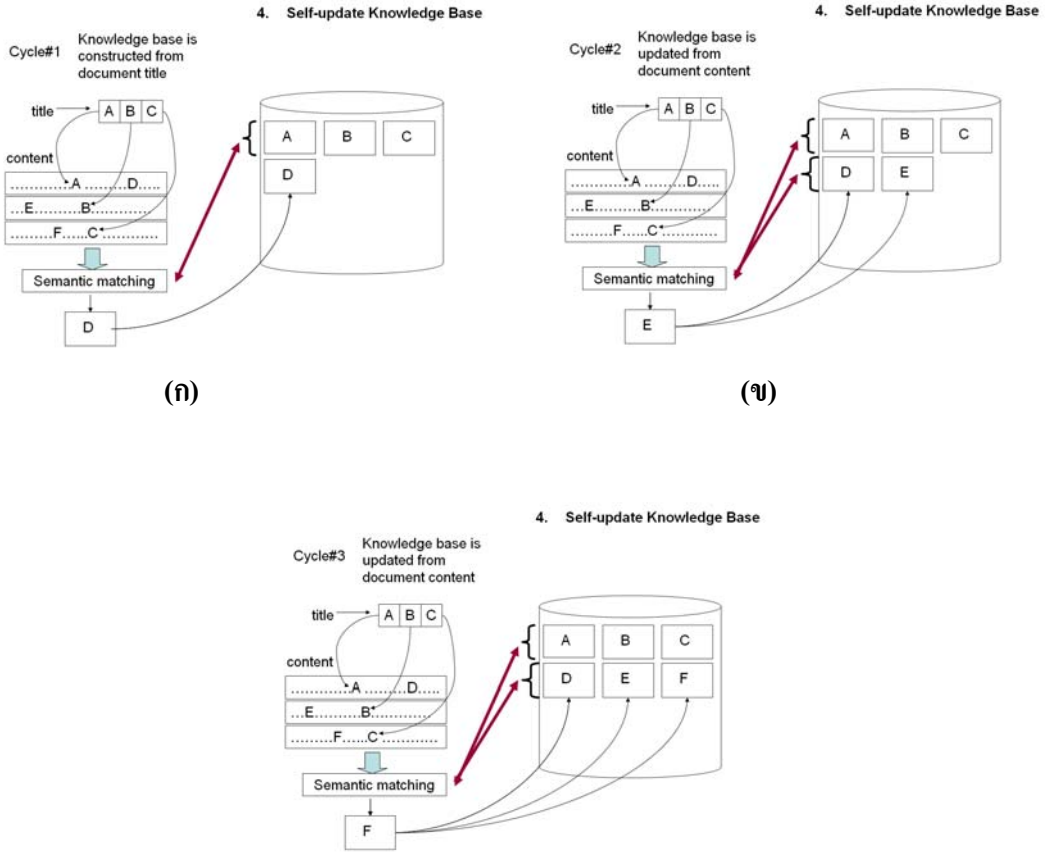
5. ส่วนของการขยายฐานความรู้คำสำคัญ (Knowledge Expansion)



รูปที่ 3.14 แสดงถึงขั้นตอนการขยายฐานความรู้คำสำคัญ

รูปที่ 3.14 แสดงถึงขั้นตอนการขายฐานความรู้คำสำคัญซึ่งเป็นจุดเด่นของงานวิจัยนี้ นั่นคือ การใช้งานฐานความรู้คำสำคัญที่ถูกสร้างและปรับปรุงให้ทันสมัยอย่างอัตโนมัติโดยไม่ใช้คลังข้อมูลขนาดใหญ่ ภายในขั้นตอนนี้ คำสำคัญ และความหมายที่ได้มาจากส่วนของการดึงคำสำคัญเริ่มต้น และส่วนของการดึงคำสำคัญที่เกี่ยวข้องเชิงความหมาย จะถูกส่งเข้าไปตรวจสอบใน Redundancy checking เพื่อพิจารณาว่าคำสำคัญเคยถูกกำหนดไว้ในฐานความรู้ของคำสำคัญหรือไม่ ถ้ายังไม่เคยถูกกำหนดมาก่อน ทั้งคำสำคัญและกราฟความหมาย จะถูกเพิ่มเข้าไปในฐานความรู้ของคำสำคัญ

โดย Knowledge Base Arranging ซึ่งจะปรับปรุงตะรางทั้ง 3 ตารางของฐานความรู้ แต่ถ้าเคยถูกกำหนดไว้แล้ว ระบบจะเพิ่มเฉพาะส่วนของความหมายเข้าไป โดยจะปรับปรุงเฉพาะ ตารางคำสำคัญ (Keyword Table) เท่านั้น



รูปที่ 3.15 ตัวอย่างของการปรับปรุงฐานความรู้คำสำคัญ

ตัวอย่างการปรับปรุงฐานความรู้ดังแสดงในรูป 3.15 (ก) - (ค) โดยเริ่มต้นฐานความรู้จะถูกสร้างจากคำสำคัญเริ่มต้นที่ได้จากส่วนของหัวเรื่อง ในตัวอย่างคือ A, B และ C ดังแสดงในรูป 3.15 (ก) จากนั้นระบบเข้าสู่ส่วนของการดึงคำสำคัญที่เกี่ยวข้องทางความหมายด้วยการพิจารณาส่วนของเนื้อความเอกสาร ซึ่งจะได้ D, E และ F ดังแสดงในรูปที่ 3.15 (ก)-(ค) ตามลำดับ

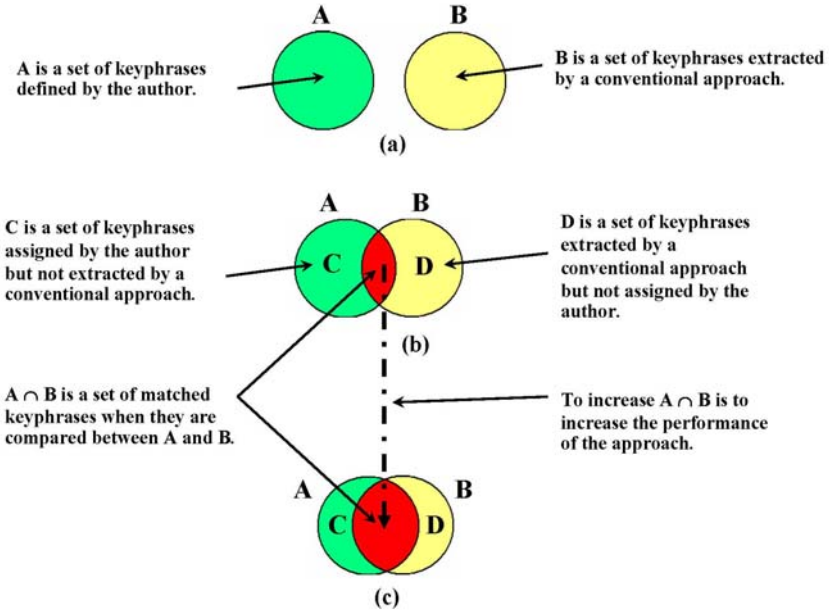
บทที่ 4 ขั้นตอนการวิจัยและผลการวิจัย

งานวิจัยนี้ ได้นำเสนอหัวข้อในการวิจัยไว้ 3 ส่วนด้วยกันคือ

1. งานวิจัยนี้สามารถค้นหาคำสำคัญได้ โดยไม่จำเป็นใช้คลังข้อมูลอื่นๆ ในการค้นหาคำสำคัญ นอกจากเอกสารนั้นๆ และ ฐานความรู้ของคำสำคัญที่ได้สร้างขึ้น
2. งานวิจัยนี้สามารถค้นหาคำสำคัญที่มีความหมายสอดคล้องกับความหมายของเอกสารได้ดีกว่าวิธีอื่น
3. งานวิจัยนี้เสนอการนำเอาฐานความรู้มาเพิ่มประสิทธิภาพในการค้นหาคำสำคัญ

ดังนั้น เพื่อพิสูจน์ประสิทธิภาพของงานวิจัย ในส่วนต่อไปนี้ จะกล่าวถึง วิธีการประเมินประสิทธิภาพของผลงานวิจัยที่ได้, การตั้งสมมุติฐานและออกแบบการวิจัย, ผลการทำวิจัย และวิเคราะห์ผลการวิจัย

4.1 วิธีการประเมินประสิทธิภาพ



รูปที่ 4.1 หลักในการประเมินประสิทธิภาพของระบบ

รูปที่ 4.1 ประกอบไปด้วย วงกลม A และ B ซึ่งแสดงถึงเซตของคำสำคัญที่กำหนดโดยผู้เขียนและคำสำคัญที่ได้จากระบบอัตโนมัติค้นหาคำสำคัญ ตามลำดับ ในทางอุดมคติ ระบบอัตโนมัติค้นหาคำสำคัญจะต้องผลิตเซตของคำสำคัญ B ที่มีขนาดเท่ากับ เซตของคำสำคัญ A แต่ในทางปฏิบัติระบบ

อัตโนมัติสำหรับดึงคำสำคัญที่มีอยู่ ผลิตได้เฉพาะส่วนที่เป็น A ∩ B ยังมีส่วนที่เป็นข้อผิดพลาดอยู่ 2 แบบด้วยกันคือ

1. ข้อผิดพลาดที่คำสำคัญที่อยู่เซตของคำสำคัญที่กำหนดโดยผู้เขียน แต่ไม่ถูกดึงออกมาด้วยระบบอัตโนมัติค้นหาคำสำคัญ ได้แก่ เซตของ C
2. ข้อผิดพลาดที่คำสำคัญที่ดึงออกมาด้วยระบบอัตโนมัติค้นหาคำสำคัญ แต่ไม่ได้ถูกกำหนดไว้โดยผู้เขียน ได้แก่ เซตของ D

ในการเพิ่มประสิทธิภาพของระบบอัตโนมัติค้นหาคำสำคัญ สามารถทำได้ด้วยการลดจำนวนข้อผิดพลาดทั้งสอง นั่นคือ เพิ่มขนาดของ A ∩ B สำหรับงานวิจัยนี้ จะเพิ่มประสิทธิภาพของระบบด้วยการลดขนาดของ เซต C และใช้ F-measure ดังแสดงในสมการ (6.3) เป็นตัววัดประสิทธิภาพของระบบ โดยใช้คำสำคัญที่ผู้เขียนกำหนดไว้ในแต่ละเอกสารเป็นมาตรฐานในการเปรียบเทียบ

$$\Pr ecision = \frac{A \cap B}{B} \qquad (6.1)$$

$$\mathrm{Re} call = \frac{A \cap B}{A} \qquad (6.2)$$

$$F - measure = \frac{2 \times (\Pr ecision \times \mathrm{Re} call)}{(\Pr ecision + \mathrm{Re} call)} \qquad (6.3)$$

4.2 การตั้งสมมุติฐานและออกแบบการวิจัย

ตารางที่ 4.1 การกำหนดค่าต่างๆสำหรับงานวิจัย

รายการที่กำหนด	ข้อมูลที่ใช้
ระบบอัตโนมัติค้นหาคำสำคัญที่มีอยู่	1. EXTRACTOR 2. KEA 3. Matsuo's System
จำนวนของกลุ่มเอกสาร	5 กลุ่ม: ธุรกิจ , คอมพิวเตอร์, การศึกษา, การแพทย์ และ จิตวิทยา
จำนวนเอกสารที่ใช้ทดสอบ	100 เอกสาร
จำนวนคำสำคัญที่ต้องการ	5, 10 and 15 คำสำคัญ
เครื่องมือทางภาษา	Machinese Syntax
ระดับเงื่อนไขในการพิจารณาคำสำคัญ	0.5, 0.8 และ 1.0

4.3 ผลการทำวิจัย

ในงานวิจัยนี้ ได้ออกแบบการทดลองเพื่อการประเมินประสิทธิภาพของระบบใน 3 แนวทางด้วยกัน คือ

1. งานวิจัยนี้สามารถค้นหาคำสำคัญได้ โดยไม่จำเป็นใช้คลังข้อมูลอื่นๆ ในการค้นหาคำสำคัญ นอกจากเอกสารนั้นๆ และ ฐานความรู้ของคำสำคัญที่ได้สร้างขึ้น โดยการเปรียบเทียบผลการวิจัยกับ งานวิจัยเดิมที่ใช้ คลังข้อมูล ได้แก่ EXTRACTOR [] และ KEA []
2. งานวิจัยนี้สามารถค้นหาคำสำคัญที่มีความหมายสอดคล้องกับความหมายของเอกสารได้ดีกว่าวิธีอื่น โดยการเปรียบเทียบผลการวิจัยกับ งานวิจัยเดิมที่ใช้ คลังความรู้และไม่ใช้คลังความรู้ ได้แก่ EXTRACTOR, KEA และ ระบบค้นหาคำสำคัญของ Matsuo []
3. งานวิจัยนี้เสนอการนำเอาฐานความรู้มาเพิ่มประสิทธิภาพในการค้นหาคำสำคัญ โดยการเปรียบเทียบประสิทธิภาพก่อนและใช้ฐานความรู้

4.3.1 เปรียบเทียบผลการวิจัยกับ งานวิจัยเดิมที่ใช้ คลังข้อมูล

ตารางที่ 4.2 แสดงให้เห็นถึงการเปรียบเทียบผลการวิจัยเมื่อนำเอาการพิจารณาทางความหมายเข้ามาใช้ เราได้ทดลองด้วยการนำเอา ระบบอัตโนมัติค้นหาคำสำคัญเดิม มา เพิ่มส่วนของการพิจารณาความหมายเข้าไป ขั้นตอนการทำการทดลองคือ นำเอาคำคู่แข่งที่ถูกปฏิเสธโดย EXTRACTOR, KEA และ Database Mapping มาพิจารณาอีกครั้งโดยใช้ความหมาย ผลการทดลองแสดงให้เห็นได้ว่า เมื่อเพิ่มการพิจารณาทางความหมายเข้าไปทำให้ระบบเดิมที่มีอยู่ทำงานได้ดีขึ้นอย่างมีนัยสำคัญ

ตารางที่ 4.2 ประสิทธิภาพของระบบที่นำเสนอเมื่อเทียบกับงานวิจัยเดิมที่มีอยู่

Methods	%Precision	%Recall	F-Measure
1. EXTRACTOR	0.53	0.70	0.60
2. EXTRACTOR + proposed function	0.56	0.80	0.66
3. KEA	0.53	0.72	0.61
4. KEA + proposed function	0.58	0.83	0.68
5. Database Mapping	0.52	0.67	0.58
6. Database Mapping + proposed function	0.61	0.74	0.67

4.3.2 เปรียบเทียบผลการวิจัยที่ใช้ความหมายกับงานวิจัยเดิมที่ไม่ใช้ความหมาย

จากการทดลองในข้อ 4.3.1 พิสูจน์ให้เห็นได้ว่า การพิจารณาความหมายช่วยเพิ่มประสิทธิภาพของการทำงานของระบบได้ ด้วยการเพิ่มฟังก์ชั่นนี้เข้าไปในระบบเหล่านั้น แต่อย่างไรก็ตาม ผลการทดลองที่ได้ขึ้นอยู่กับประสิทธิภาพของระบบเดิมด้วย ดังนั้น การทดลองในส่วนนี้ จะเป็นการพิสูจน์ว่า ระบบอัตโนมัติค้นหาคำสำคัญที่ใช้การพิจารณาทางความหมาย มีประสิทธิภาพที่ดีกว่าระบบเดิมที่อยู่

ตาราง 4.3 และ 4.4 แสดงผลการทดลองที่ได้จากการเปรียบเทียบกับ EXTRACTOR (Turney, P. D.: 1997,2000,2001,2003) และ KEA (Witten, I.H., et al.: 1999) ที่เป็นตัวแทนของระบบอัตโนมัติค้นหาคำสำคัญ ที่ใช้คลังข้อมูล และเปรียบกับ ระบบค้นหาคำสำคัญของ Matsuo (Matsuo, Y., and Ishizuka, M.: 2004) ที่เป็นตัวแทนระบบที่ไม่ใช้คลังข้อมูล ในตารางที่ 4.3 และ 4.4 คำที่เป็นตัวเอียงและหนา คือคำที่เหมือนกับคำสำคัญที่ผู้เขียนกำหนดไว้ ซึ่งจะเห็นว่าระบบที่นำเสนอให้ คำสำคัญที่ตรงกับผู้เขียนมากกว่า

ตารางที่ 4.3 ตัวอย่างของผลที่ได้ของการวิจัยเมื่อเทียบกับงานวิจัยเดิม ที่ใช้คลังข้อมูล

EXTRACTOR	KEA	KEA2	The proposed system
exhibit	Baton Based	Design Patterns	***interface design***
instruments	***interactive exhibits***	input device	***worldbeat system***
user feedback	***interface design***	user interface	***worldbeat exhibit***
user interface design	user interface	user ***interface design***	User interface
WorldBeat system	***WorldBeat system***	virtual realities	***music***

ตารางที่ 4.4 ตัวอย่างของผลที่ได้ของการวิจัยเมื่อเทียบกับงานวิจัยเดิม ที่ไม่ใช้คลังข้อมูล

Matsuo's System	The proposed system
digital computer	***digital computer***
storage capacity	***storage capacity***
imitation game	***imitation game***
machine	***discrete-state machine***
human mind	**intelligence**
universality	compute machinery
logic	**computation**
property	Object
mimic	sense
discrete-state machine	man

4.3.3 เปรียบเทียบประสิทธิภาพของการนำเอาฐานความรู้ของคำสำคัญมาใช้

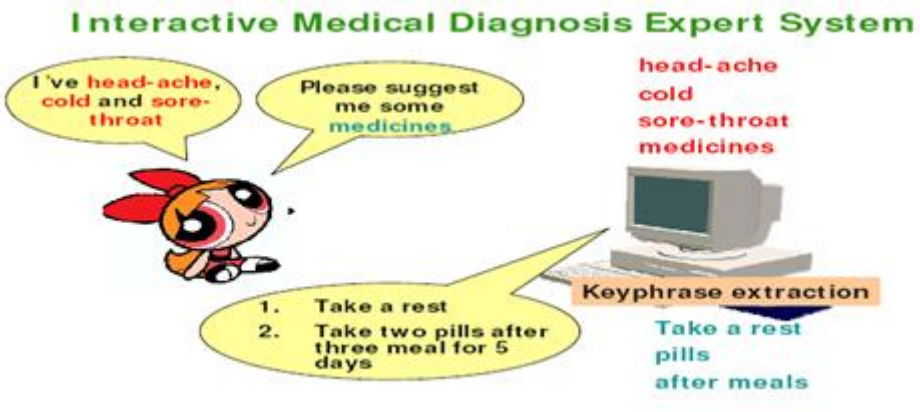
จากข้อเสนอของการวิจัยที่บอกว่า ฐานความรู้ที่สร้างและปรับปรุงตัวเองได้โดยอัตโนมัติ ช่วยเพิ่มประสิทธิภาพของการค้นหาคำสำคัญได้ ตารางที่ 3.5 แสดงให้เห็นว่า ในช่วงที่ 1-3 ของการทำงานของระบบอัตโนมัติค้นหาคำสำคัญ เปอร์เซ็นต์ของคำสำคัญที่ตรงกับคำสำคัญที่ผู้เขียนกำหนดมีมากขึ้นอย่างมีนัยสำคัญ แต่อย่างไรก็ตาม เนื่องจากงานวิจัยนี้เสนอว่าใช้เพียงเอกสารนั้นๆ ในการพิจารณาความหมาย ซึ่งส่วนใหญ่เอกสารที่นำมาจะเป็นเอกสารวิชาการที่มีข้อมูลไม่มากนัก ทำให้เมื่อทดลองไปอีก 2 รอบ ประสิทธิภาพของระบบยังคงเท่าเดิม

ตารางที่ 4.5 ประสิทธิภาพของการนำเอาฐานความรู้ของคำสำคัญมาใช้

# of extraction cycle	%of matched keyphrases
1	54.90
2	66.98
3	74.12
4	**75.55**
5	**75.55**

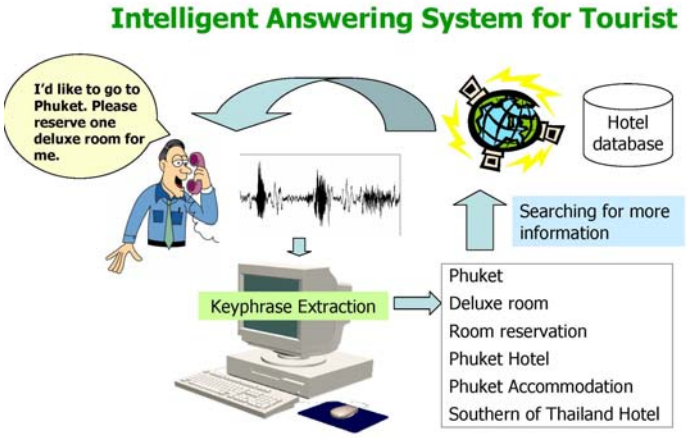
บทที่ 5 การนำงานวิจัยไปใช้ประโยชน์

รูปที่ 5.1 – 5.3 แสดงถึงตัวอย่างการประยุกต์ใช้ การค้นหาคำสำคัญโดยใช้ความหมาย กับ การประมวลผลสารสนเทศ ต่างๆ ได้แก่ ระบบถาม-ตอบ (Question/Answering System) การจัดประเภทเอกสาร (Text Classification) และ การสรุปใจความสำคัญของเอกสาร (Text Summarization)



รูปที่ 5.1 ระบบผู้เชี่ยวชาญสำหรับวินิจฉัยโรค

(Interactive Medical Diagnosis Expert System)



รูปที่ 5.2 ระบบถาม-ตอบทางโทรศัพท์ (Intelligent Answering System for Tourist)

จากรูปที่ 5.1 และ 5.2 เราสามารถประยุกต์ใช้ระบบอัตโนมัติค้นหาคำสำคัญ กับระบบสนทนาอัตโนมัติได้ ในรูปที่ 5.1 เป็นระบบผู้เชี่ยวชาญสำหรับวินิจฉัยโรคได้ และ รูปที่ 5.2 เป็นระบบถาม-

ตอบสำหรับการท่องเที่ยว ทั้งสองระบบผู้ใช้จะป้อนข้อมูลเข้ามาในลักษณะที่เป็นภาษาธรรมชาติ ซึ่งมีรูปแบบของการถามหลากหลาย ถ้าต้องให้ระบบทั้งสอง มาวิเคราะห์คำทุกคำในประโยค ก็จะทำให้ยุ่งยาก ดังนั้น ถ้ามีระบบอัตโนมัติค้นหาคำสำคัญ ดึงเฉพาะคำที่สำคัญ ออกมา ก็ทำให้ระบบทำงานได้รวดเร็วขึ้น

รูปที่ 5.3 ระบบอัจฉริยะสำหรับจัดประเภทบทความวิชาการ

(Intelligent Technical Paper Classification)

สำหรับรูปที่ 5.3 จะเป็นการประยุกต์ใช้ ระบบอัตโนมัติค้นหาคำสำคัญกับการแยกประเภทเอกสาร ตัวอย่างการประยุกต์ใช้คือ ระบบสำหรับจัดประเภทของบทความให้ตรงกับ Profile ของผู้เชี่ยวชาญ ระบบที่ใช้อยู่ปัจจุบันจะอาศัยมนุษย์ในการอ่านเพื่อทำความเข้าใจ และเลือกบทความให้ตรงกับความถนัดของผู้เชี่ยวชาญแต่ละท่าน แต่ถ้าระบบมีการจัดเก็บความถนัดของผู้เชี่ยวชาญแต่ละท่านด้วย คำสำคัญ ซึ่งอาจจะเป็นชื่อสาขางานวิจัยที่สนใจ จากนั้นระบบอัตโนมัติค้นหาคำสำคัญ ก็จะอ่านบทความและดึงเอาเฉพาะคำสำคัญออกมา จากนั้น คำสำคัญเหล่านี้ จะถูกนำไปเปรียบเทียบกับความถนัดที่ผู้เชี่ยวชาญให้ไว้ ก็จะสามารถจัดประเภทบทความได้ใกล้เคียงกับความต้องการของผู้เชี่ยวชาญได้

บรรณานุกรม

- Banerjee, S., Pederson T. (2003) “Extened gloss overlop as a measure of semantic relatedness.” In Proc. Of IJCAI-03:805-810.
- Breiman, L., (1996), “Bagging Predictors”, Machine Learning, Vol. 24, No. 2, pp. 123-140.
- Budanitsky, A. and Graeme H., (2001), “Semantic Distance in Wordnet: An Experimental, Application-Oriented Evaluation of Five Measures”, The NAACL 2001 Workshop on WordNet and Other Lexical Resources [Electronic], Pittsburgh, USA, Available: http://citeseer.ist.psu.edu/budanitsky01semantic.html [2005, December 10].
- Chein et Marie-Laure Mugnier, M. (1992) “Conceptual Graphs fundamental notions.” Intelligence Artificial, Vol.6,:365-406
- Chen, K.H. and Wu, C.T., (1999) “Automatically Controlled-Vocabulary Indexing for Text Retrieval”, The 12th Research on Computational Linguistics Conference, Hsinchu, Taiwan, pp. 171-185.
- Chey, C., Kumhom, P. and Chamnongthai, K. (2006) Khmer Printed Character Recognition by using Wavelet Descriptors. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 14(3), 337-350.
- Cognitive Science Laboratory at Princton University, “WordNet 2.1 Browser”, Available: http://wordnet.princton.edu
- Connexor Oy Inc., (2005) Machinese Syntax [Online], Available: http://www.connexor.com/demo/syntax/
- Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C. and Nevill-Manning, C.G., (1999) “Domain-specific keyphrase extraction”, The 16th International Joint Conference on Artificial Intelligence (IJCAI-99), July 31 - August 6, Stockholm, Sweden, pp. 668--673.
- Gutwin, C., Paynter, G.W., Witten, I.H., NevillManning, C.G. and Frank, E, (1998) Improving Browsing in Digital Libraries with Keyphrase Indexes, Technical Report, Department of Computer Science, University of Saskatchewan, Canada.
- Hist, G., St-Ong, D. (1998) “Lexical Chains as Representations.” In Fellbaum:305-332
- Jiang, J. J., and Conrath, W. D., (1997) “Semantic Similarity Based on Corpus Statistic and Lexical Taxonomy” In Proceedings of International Conference on Research in Computational Linguistics, Taiwan.
- Jones, S. and Paynter, G.W., (2003), “An Evaluation of Document Keyphrase Sets”, Journal of Digital Information [Electronic], Vol. 4, No. 1, Available: http://jodi.ecs.soton.ac.uk/Articles/v04/i01/Jones/ [2005, September 9].
- Kongkachandra, R., Chamnongthai, K., and Kimpan, C., (2006), Intelligent Keyphrase Extraction for Digital Library, **Information Technology and Libraries**, Vol. 25, No. 2, (to be appeared).
- Leacock, C., Chodorow, (1998) M. “Combining Local Context and WordNetSimilarity for Word Sense Identification.” In Fellbaum:265-283.
- Lesk, M., (1986) “Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to tell a Pine Cone from a ice Cream Cone.“ In Proceedings

of the 5th Annual Conference on Systems Documentation, Toronto, Ontario, Canada,:24-26.

- Matsuo, Y., and Ishizuka, M., (2004) “Keyphrase Extraction from a Single Document Using Word Co-Occurrence Statistical Information”, International Journal on Artificial Intelligence Tools, Vol. 13, No. 1, pp. 157-169.
- Montejo-Ráez and Steinberger, R., (2004) “Why Keywording Matters”, High Energy Physics Libraries Webzine [Electronic], Issue 10, Available: http://library.cern.ch/HEPLW/10/papers/2/ [2005, September 1].
- Montejo-Ráez, A., (2002) “Towards Conceptual Indexing Using Automatic Assignment of Descriptors”, Workshop in Personalization Techniques in Electronic Publishing on the Web: Trends and Perspectives, May 28, Málaga, Spain.
- Montes-y-Gmez, M., Lpez-Lpez, A. and Gelbukh, A.F., (1999) “Document Title Patterns in Information Retrieval”, Lectures Notes in Computer Science, Vol. 1692, pp. 364–367.
- Montes-y-Gomez, M., and Lopezs-Lopez A., (2000) “Comparison of Conceptual Graphs.” Proc. MICAI-2000, 1st Mexican International Conference on Artificial Intelligence, Acapulco, Mexico, April
- Montes-y-Gomez, M., Lopezs-Lopez, A., and Gelbukh, A. (2000) “Information Retrieval with Conceptual Graph Matching.” Proc. DEXA-2000, 11 th International Conference and Workshop on Database and Expert Systems Application, Greenwich England
- Montes-y-Gomez, M., Lopezs-Lopez, A., Gelbukh, A., and Baeza-Yates, R. (2001). Flexible Comparison of Conceptual Graphs, Proc DEXA-2001, 12th International Conference and workshop on Database and Expert System Applications, Lecture Notes in Computer Science, N2113, Spriger-Verlag, 2001, pp 102-111.
- Percival, D. B. and Walden, A. T. (2000). Wavelet Methods for Time Series Analysis. Cambridge University Press.
- Resnik, P., (1995) “Using Information Content to Evaluate Semantic Similarity in a Taxonomy.” Proceeding of The 14th International Joint Conference on Artificial Intelligence. Vol. 1:448-453.
- Sowa, J.F. (1984) Conceptual Structures: Information Processing, Mind And Machine, Addison-Wesley, Reading, MA, USA.
- Teich, E. and Fankhauser, P. (2004) “WordNet for Lexical Cohesion Analysis”, The Second Global WordNet Conference, January 21-23, Masaryk University, Brno, Czech Republic, pp. 326-331.
- Turney, P. D. (1997) Extraction of Keyphrases from Text: Evaluation of Four Algorithms, Technical Report NRC 41550, National Research council.
- Turney, P. D. (2000) “Learning Algorithms for Keyphrase Extraction”, Information Retrieval, Vol. 2, No. 4, pp. 303-336.
- Turney, P. (2001) “Mining the Web for Synonyms: PMI-IR Versus LSA on TOEFL”, The Twelfth European Conference on Machine Learning (ECML-2001), Freiburg, Germany, pp. 491-502.
- Turney, P. (2003) “Coherent Keyphrase Extraction via Web Mining”, The Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, pp. 434-439.

- University of Ottawa (2000) DIPETT on the Web [Online], Available: http://www.site.uottawa.ca/tanka/dipett-on-the-Web/frames.html [September 9].
- Witten, I.H., Paynter, G.W., Frank, E, Gutwin, C., and NevillManning, C.G. (1999) "Kea: Practical Automatic Keyphrase Extraction", The Fourth ACM Conference on Digital Libraries (DL'99), August 11-14, University of California, Berkeley, USA, pp. 254-256.
- Yang, D., and Powers, D., (2005) "Measuring Semantic Similarity in the Taxonomy of WordNet." Flinders University of South Australia.
- Zhang, H., Sun, J., Wang, B., and Bai, Shou. (2005) "Computation on Sentence Semantic Distance for Novelty Detection" Inst. of Computation Tech., The Chinese Academy of Science , Beijing, China.

-ภาคผนวก-

ผลงานที่ได้รับการตีพิมพ์

- Chey, C., Kumhom, P. and Chamnongthai, K. (2006) Khmer Printed Character Recognition by using Wavelet Descriptors. **International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems**, 14(3), 337-350.

- Kongkachandra, R., Chamnongthai, K., and Kimpan, C., (2006), Intelligent Keyphrase Extraction for Digital Library, **Information Technology and Libraries**, Vol. 25, No. 2, (to be appeared).

- Abductive Reasoning for Keyword Recovering in Semantic-based Keyword Extraction, **Informatica**, (submitted)

International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 14, No. 3 (2006) 337–350

World Scientific
www.worldscientific.com

KHMER PRINTED CHARACTER RECOGNITION BY USING WAVELET DESCRIPTORS

C. CHEY, P. KUMHOM, and K. CHAMNONGTHAI
King Mongkut's University of Technology Thonburi
91 Prachautit Rd., Bangmod, Tungkru, Bangkok 10140, THAILAND

Received 25 October 2005
Revised 3 May 2006

In Khmer printed characters, same character has various shapes according to the fonts and some characters are very similar in shape. In this paper we try to solve these problems, and propose a method of Khmer printed character recognition by using Wavelet Descriptors. In the recognition, firstly the Khmer printed character images are converted to skeleton forms, then skeletons of Khmer character are converted to temporal domain. The templates are obtained by wavelet coefficients from the character training set. To match the input characters with templates, the character recognition method using deformable wavelet descriptor is adapted by using fixed template and Euclidean distance classifier for matching. The smallest distance is the recognition result of the proposed method. As a result, the deformation can be skipped because it might get low recognition rate of similar characters. The experiment consists of two parts. The first part is to evaluate the overall recognition rate of input characters with three different sizes (22-point, 18-point and 12-point) from 10 different fonts of Khmer printed character. Twenty styles of characters are used as the training set. The results show 92.85, 91.66, and 89.27 percent for 22-point, 18-point, and 12-point respectively. The second part is to specifically evaluate the system, testing with one document that has 21 pages of Khmer printed character with different resolutions from a scanner and facsimile (fax). The document is initially printed with 300 dpi (dots per inch), then scanned with three different resolutions, 600 dpi, 300 dpi and 150 dpi. The document that received from fax machine is scanned by 300dpi. The results show 92.99, 88.61, and 80.05 percent recognition rate for 300, 150 dpi resolutions, and input from fax respectively.

Keywords: Optical Character Recognition; Wavelet Descriptor; Deformable Template Matching;Template Matching.

1. Introduction

Optical Character recognition (OCR) becomes more and more important in the modern world. This technology enables the use of a pen device as a keyboard and, thus, reduces the need for a bulky keyboard as standard equipment in palm (short note) computing devices. On-line hand writing character recognition is also important as a means of data entry for those that do not wish to type on the keyboard, for whatever their reason may be. For postal system, OCR can fasten the process of address classification by finding the zip code and separating the mail into different distribution piles for each destination. Also hard copies and many old books are needed to represent in electronic form that need OCR machine or OCR software package for converting. It also plays important role in many applications such as the

information retrieval system, knowledge-base-updating system, and e-library. However, an OCR can not be applied into all languages. There are many researches that proposed the methods for optical character recognition (OCR) system in different languages. These methods can be categorized into two groups, including statistical classification [5, 6], and template matching [1, 2, 3, 4].

In the first group, Kijsirikul, Sinthupinyo and Supanwansa [5] proposed Thai printed recognition using combination inductive logic programming with back propagation neural network (BNN). They present a new approach that combines two learning algorithms that are Inductive Logic Programming (ILP) and Back Propagation Neural Network (BNN). The results show high recognition rate with time consuming. Thammano and Ruxpakawon [6] proposed an approach to recognize the Thai printed characters using the hybrid of global feature, local feature, fuzzy membership function and neural network. The methods in this group give high recognition rate, but they require much more computation time.

For the second group in template matching approach, in general, a pre-processed image of an input character is matched with the templates representing some kind characters' features. Then, the input image is recognized as the character whose template match the image the most. There are two kinds of templates, fixed template and deformable template. The following two researches use the fixed templates. Liao and Lu [1] proposed a method for Chinese character recognition by using moment functions as the features of the template for recognizing Chinese characters, especially for characters that very similar in shapes. Euclidean distance classifier is used to classify all Chinese characters. The results show those features can represent all Chinese characters reasonably well, especially for Legendre moment. The methods mentioned above work well with the fixed fonts, they do not work well for the OCR system with various forms of inputs such as printed characters with many fonts and handwritten characters. Chiang, Liao, Lu and Pawlak [2] proposed a method for Chinese character recognition, especially for the characters closed in shapes, using Gegenbauer Moments as the features of the Chinese characters. The results show that Gegenbauer Moment-based features extracted from each Chinese character can represent all Chinese characters reasonably well. The Gegenbauer moment with smaller values of parameter would provide more recognition power to the recognition system. The methods mentioned above work well with the fixed fonts, they do not work well for the OCR system with various forms of inputs such as printed characters with many fonts and handwritten characters. The deformable template is more appropriate for such inputs. For the deformable template,Phaopanus, Arunrungrusmi and Chamnongthai [4] proposed deformable wavelet descriptors for the Thai handwritten character recognition. In the recognition, contour of Thai characters are utilized to determine template in term of temporal domain. Then, range of deformation is obtained by standard deviation of Wavelet descriptor coefficients from characters training set. To fit the templates with input characters, all templates are deformed to obtain the best fit by determining deformation range. The best fit of each character template is represented

Type	Members
Consonant	
Sub consonant	
Vowels	
Dependant vowels	
Other Symbols	
Numbers	

Fig. 1. Khmer characters

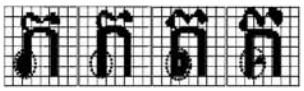


Fig. 2. a sample of four Similar Characters

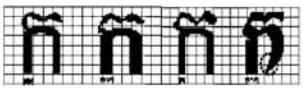


Fig. 3. a sample of same character from four different fonts

by score, and the character template with highest score is the recognition result. The recognition rate is high. However, the character that is closed in shapes is misclassified in some cases and takes much computation time. This paper proposes a method for Khmer printed character recognition by using wavelet descriptor in order to increase the accuracy rate and decrease the computation time. The authors try to solve the problems of different characters in similar shapes, and same characters in varied-shape fonts in order to increase the accuracy and simultaneously decrease the computation time.

2. Problem Analysis and Basic Idea

Khmer language that is daily used in Cambodia is one of the old languages and has a set of unique characters as shown in Fig. 1. Nowadays, there are around 30 fonts used in the Khmer language. Some different characters are similar in shapes as shown in Fig. 2, and same characters from different fonts vary according to font types as shown in Fig. 3.

Although languages have common features, each language possesses unique features. Like other languages, there are many printed fonts for the Khmer character. From those characters and fonts, some different characters are similar in shape as shown in Fig. 2. On the other hand, some characters have varied-shape fonts as shown in Fig. 3. These are causes of recognition errors.

The flesh of similar characters are very close in shapes as shown in Fig. 2, and

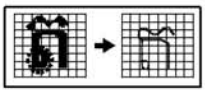


Fig. 4. character /tor/ in skeleton form

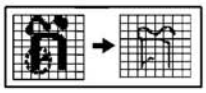


Fig. 5. character /kor/ in skeleton form

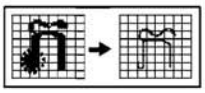


Fig. 6. character /phor/ in skeleton form

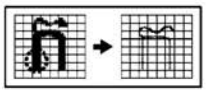


Fig. 7. character /gar/ in skeleton form

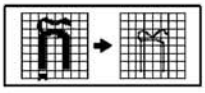


Fig. 8. character /gar/ in skeleton form

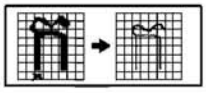


Fig. 9. character /gar/ in skeleton form

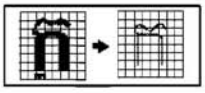


Fig. 10. character /gar/ from different fonts represented in skeleton form

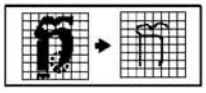


Fig. 11. characters /gar/ from different fonts represented in skeleton form

the flesh of same characters from different fonts are different according to the fonts. However, the skeletons of the similar characters are differentiated as shown in Fig. 4, Fig. 5, Fig. 6 and Fig. 7, and the skeletons of the same characters from different fonts are mostly same as shown in Fig. 8, Fig. 9, Fig. 10 and Fig. 11.

Since same characters from different fonts are varied, it makes the features of same characters varied. Some characters are very similar in shapes, it causes confusion with other characters. When we analyze them into skeleton form, we found that different characters in similar fonts are obviously able to differentiate as samples shown in Fig. 4-6, and same characters in shape-varied fonts are simultaneously recognized as the same result as shown in Fig. 7-11. The skeletonization technique is used to normalize the shapes of the characters. Wavelet descriptor uses the basic functions with local support and multiscale dilation, thus it can model local features as well as global ones. These make system flexible for the template, and robust against changes. Normally, information exists in the low frequency ranges[7]. At the same time, most of the methods would cut the signals in the high frequency range because they are usually noises. Therefore, Wavelet descriptor coefficient at the low frequency range is used as the feature extraction of all characters. The proposed method uses fixed template and Euclidean distance classifier as the matching scores instead in order to reduce the computation time. The reason is that, in most cases, the relative Euclidean distance should represent the relative correlation. In other words, if we compare the Euclidean distance of two templates from an input, the template with shorter distance would give a higher correlation score.

3. Proposed Method

In Fig. 12, a common setup of proposed method is illustrated. The first step in the process is to digitize the analog document using an optical scanner or digital camera. The extracted characters may then be preprocessed to facilitate the extraction of

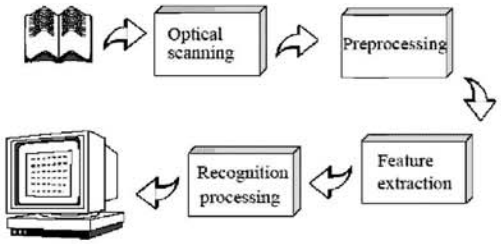


Fig. 12. OCR system

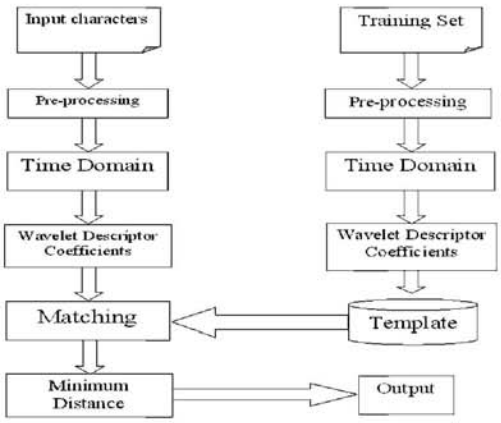


Fig. 13. block diagram of proposed method

features in the next step. The identity of each symbol is found by comparing the extracted features with descriptions of the character classes obtained through a previous learning phase. Finally the recognition process is used to recognize all characters.

3.1. *System*

Fig. 13 shows the system block diagram of Khmer character recognition. The set of characters called training set is used to construct the templates storing in terms of the means values of wavelet descriptor coefficients from their temporal data. During the running time, an input image is pre-processed and converted to the temporal domain, then, to wavelet coefficients. It is matched with all the templates using the Euclidean distances classifier. They find distance between input and every template then compare all distances. The smallest distance is classified to be the output of proposed method. The proposed method is separated into three parts,including Preprocessing, Feature extraction, and Classification process.

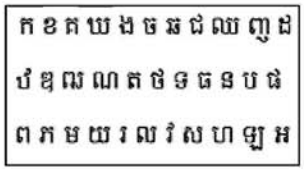


Fig. 14. character images before Thresholding

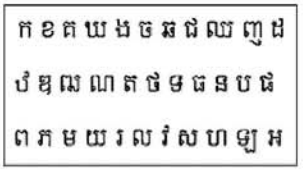


Fig. 15. binarized character images after Thresholding

3.2. *Pre-Processing*

In the pre-processing, we extract gray scale from character images obtained from scanner, then convert it into binary images by using thresholding. All the intensity values above the threshold intensity are converted to one high intensity value, representing black. All intensity values below the threshold are converted to another low intensity value representing white. The binary character images, then, are transformed to skeletonization form, applying the thinning algorithm to normalize the shape of the characters.

3.2.1. *Binarization*

In this paper, we assumed that the document does not have high noise content, in which case the image can be binarized directly. Fig. 14 shows a set of 33 Khmer consonants, representing in gray scale image. Fig. 15 is the character mages, representing in binary images that converted from gray scale image from Fig. 14.

3.2.2. *Thinning*

The skeleton image is usually smaller than its original image. With thinning algorithm, it not only increases the speed of recognition, but also improves the performance of recognition. Skeletonization is the process of peeling off of a pattern as many pixels as possible without affecting the general shape of the pattern. In other words, after pixels have been peeled off, the pattern should still be recognized. The obtained skeleton must have the following properties:

- as thin as possible
- connected
- centered

When these properties are satisfied, the algorithm must stop. A number of thinning algorithms have been proposed and are being used. The most common used algorithm is the classical Hilditch algorithm [8]. A simple algorithm is used for thinning images [9, 10]. The algorithm makes two passes: one pass on the actual image and another pass over intermediate images, which are constructed from the original image during the first pass. The result after the two passes decides if a pixel is to be removed to get the skeleton of the object of interest.

The thinning algorithm works as follows. Fig. 16 shows an input image on

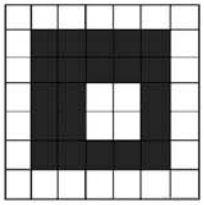


Fig. 16. a sample of character to be thinned

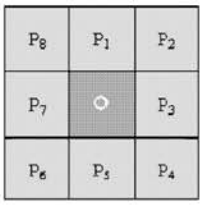


Fig. 17. mask for thinning

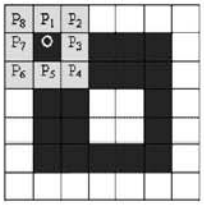


Fig. 18. using thinning mask

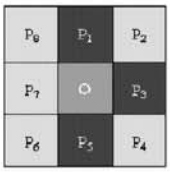


Fig. 19. transition thinning masks(a)

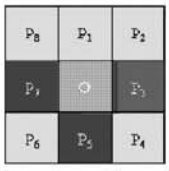


Fig. 20. transition thinning masks(b)

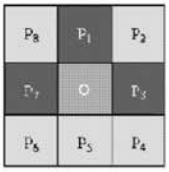


Fig. 21. transition thinning masks(c)

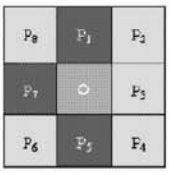


Fig. 22. transition thinning masks(d)

which the thinning algorithm is to be applied. Each box represents a pixel. Fig. 17 shows a 3x3 mask that is used to move over the input image during the first pass and then on the intermediate image during the second pass. The center of the mask is positioned on the pixel of interest during the scanning process and the numbered boxes represent the corresponding neighbor pixels. The pixel of interest, shown as "o", is marked for deletion when all the following conditions are satisfied. These conditions are checked only for a "dark pixel".

- If the number of neighboring 'dark pixels' of "o" is between 2 and 6.
- If the number of transitions from 'bright pixel' to 'dark pixel' in the order shown is one.
- If the logical function P1.P3.P5 = 0 and also P3.P5.P7 = 0, as shown by the darkened portions of the masks in the Fig. 19 and Fig. 20. These conditions are repeated for each pixel in the image as it is scanned. The intermediate result is an image, which consists of pixels that are marked for deletion and other pixels that are left as in the original input image. The second pass is performed on this intermediate image and now the conditions of deletion on the pixel are as follows:
- If the number of neighboring 'dark pixels' of "o" is between 2 and 6.

Fig. 23. original image

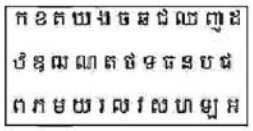


Fig. 24. original image after Thresholding

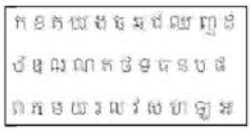


Fig. 25. thinned image

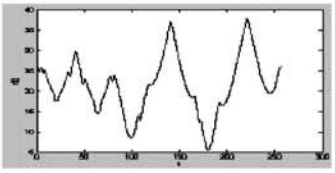


Fig. 26. temporal domain of /gar/ that similar with /phor/

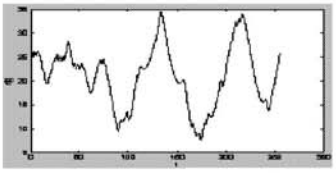


Fig. 27. temporal domain of /phor/ that similar with /gar/

- If the number of transitions from 'bright pixel' to "dark pixel" in the order shown is one.

- If the logical function P1.P3.P7 = 0 and also P1.P5.P7 = 0, as shown by the darkened portions of the masks in fig. 21 and fig. 22. As mentioned above, the pixel is deleted only when all the conditions are satisfied. Some results using this thinning algorithm are shown in fig. 25. It is evident from these results that input characters have been reduced to one-pixel width.

3.3. *Feature Extraction*

Feature extraction is the most important part in character recognition because features are main keys for recognizing unknown characters. The objective of feature extraction is to capture the essential characteristics of the characters, and it is generally accepted that this is one of the most difficult problems of pattern recognition. In our method, the characters after spatial domain are converted to wavelet descriptor coefficients, are called feature extractions.

3.3.1. *Temporal Domain*

First, we convert all of skeleton characters into temporal domain as reference pattern of each character. Then, the training character sets of Khmer printed characters are performed in the same way. In temporal domain, we want to extract only the wanted objects and remove the unwanted objects. So, the temporal algorithm does not only increase the speed of recognition, but also improves the performance of recognition. equation (1) gives the definition of the temporal domain for representing all skeleton characters. Representing all skeleton characters in temporal domain is giving by

$$r(t) = \sqrt{(X(t) - cg_x)^2 + (Y(t) - cg_y)^2} \tag{1}$$

where $r(t)$ represents the skeleton characters in temporal domain,
$[(X(t), Y(t))]$ is skeleton characters coordinate in spatial domain,and
$[cg_x, cg_y]$ is center of mass of the character.

3.3.2. *Wavelet Descriptors*

Wavelet based approaches become increasingly popular in pattern recognition, and recently have been applied for character recognition [3, 7, 4]. After all skeleton char-

acters are converted into temporal domain, the skeleton characters of training set in temporal domain are presented by wavelet descriptor to obtain wavelet coefficients. Wavelet descriptor uses a set of basic functions with local support and multiscale dilation, thus, it can model local as well as global features. Wavelet descriptor [7, 11] offers natural multi-resolution representation of the signal. Wavelet decomposes 1-D signal into multi-resolution scales. The wavelet equations are given as seen in appendix A.

3.4. *Matching Process*

Recognition techniques based on matching represent each class by a prototype pattern vector. An unknown pattern is assigned to the class to which it is closest in term of a predefined metric. In this paper, the minimum distance classifier, which, as its name implies, computes the (Euclidean) distance between unknown and each of prototype vectors called template, is applied. It chooses the smallest distance to make a decision. In matching process, firstly, the templates are obtained from wavelet coefficients of the characters training set. Then, wavelet coefficients of all input characters are matched with every template. To match the input characters and template the Euclidean distance classifier is used. We calculate the distance between input character and the templates. From matching, then, the template that gives the minimum distance is the output to be classified. The minimum classifier approach is given as seen in appendix B [11].

4. Experimental Results

In order to test the efficiency of the proposed algorithm, we test all Khmer characters. Each input character is binarized with '1' denoting the object and '0' the background. The input characters are segmented into isolated characters. Twenty font styles (LIMON S1, LIMON S2, LIMON S3, LIMON S4, LIMON S5, LIMON S6, LIMON S7, ABC-TEXT-11, ABC-TEXT-12, ABC-TEXT-13, ABC-TEXT-14, ABC-TEXT-15, ABC-TEXT-16, ABC-TEXT-17, ABC-TEXT-19, ABC-TEXT-20, Limon S2D, SOVAN 1, TAPROM and EKREACH) are used as training set. For testing, the first part is to evaluate the overall recognition rate of input characters with three different sizes (22-point, 18-point and 12-point) from 10 different fonts (ABC-TEXT-01, ABC-TEXT-02, ABC-TEXT-03, ABC-TEXT-04, ABC-TEXT-05, ABC-TEXT-60, ABC-TEXT-07, ABC-TEXT-08, ABC-TEXT-09 and ABC-TEXT-10) of Khmer printed character. The results show 92.85 percent recognition rate for the 22-point, 91.66 percent recognition rate for the 18-point, and 89.27 percent recognition rate for the 12-point. The second part is to specifically evaluate the system, testing with one document that has 21 pages from font ABC-TEXT-18 of Khmer printed character with different resolutions from scanner and fax. The document is printed with 300 dpi (dots per inch) then scanned with three different resolution, 600 dpi, 300 dpi and 150 dpi. For the document that received from fax machine is scanned with 300 dpi resolution. The results show 92.99 percent

Fig. 28. misclassified characters caused by skeleton

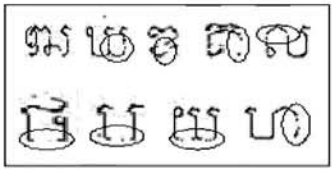


Fig. 29. misclassified characters caused by noise

recognition rate for 600 dpi resolution, 88.61 percent recognition rate for 300 dpi resolution, 80.05 percent recognition rate for 150 dpi resolution and 71.68 percent recognition rate for the input from fax machine. The testing characters are shown in appendix-A. All the experimental results are shown in Table 1 and Table 2.

5. Discussion

The first experiment, the proposed algorithm can receive high recognition rate. In this case, we also see that the system has misclassified some characters in some cases. These misclassifications are probably caused by preprocessing errors. During converting binary character image to skeleton characters form, some skeleton characters shape could not be obtained as the original one. It loses some parts of the characters as shown in Fig. 28.

The proposed algorithm also received 92.99 % recognition rate for the input data scanned with 600 dpi resolution, 88.61% recognition rate for the input data scanned with 300 dpi resolution, 80.05% recognition rate for the input data scanned with 150 dpi resolution and 71.68% recognition rate for the input data received from the Fax machine that scanned with 300 dpi resolution. In these cases, we also see that the system has misclassified some characters in some cases. These are probably caused by noise that makes some characters not fully connected or maybe make it looks like other characters. Especially, in case of the input characters from Fax machine the system has trouble to recognize most of the characters. From experiments, we realized that the system still has troubles identifying the characters that have low resolution, such as the input characters from Fax machine. Fig. 29 shows that some input characters are misclassified with others because some characters are not fully connected making them looked like other characters. In the case of similar

Table 1: Recognition results of input Images from 10 different fonts scanned by 300 dpi

Font size	Recognition rate
22-point	92.85%
18-point	91.66%
12-point	89.27%

Table 2: Recognition results of input images from ABC-TEXT-18 Font

Resolution	**Recognition Rate**
600 dpi	92.99%
300 dpi	88.61%
150 dpi	80.05%
Fax (300 dpi)	71.69%

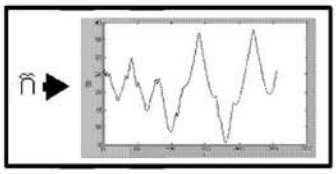


Fig. 30. character/gar/in temporal domain.

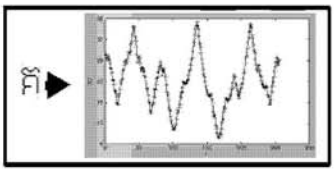


Fig. 31. character /kor/in temporal domain.

characters, comparing fixed template and deformable template, fixed template gets error rates lower than deformable wavelet descriptor template. In the deformable wavelet descriptor template, a set of input characters is used for obtaining the template which includes the mean values and the standard deviation of the wavelet coefficients. Then, an input is compared with the templates of all characters to get the best match. Before comparing, a template is allowed to deform within the range of mean value, plus and minus standard deviation as shown in fig. 32. The resulting deformed template is compared with the input to get the matching score using correlation. From Fig. 33 it can be seen that any characters that similar in shapes will fall into the same range of one template that causes the confusion in the decision process. From Table 3, also, we can see that the proposed method is more faster than previous work [4].

The possible improvements suggested for this work is by improving the efficiency of the skeleton representation of the characters, improving and increasing the efficiency of the binary character images, and using the post-processing to help correcting the misclassified characters. For increasing the efficiency of the recognition rate, also, the samples of the characters training set have to be increased. Scanned documents often contain noises that arises due to printer, scanner, fax, print quality, etc. Therefore, it is necessary to filter these noises before we process

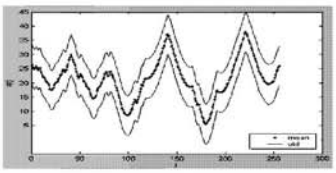


Fig. 32. character/gar/in temporal domain

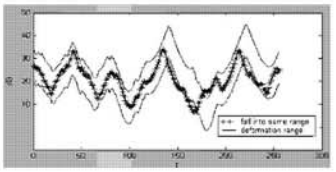


Fig. 33. character /kor/in temporal domain

Table 3: Computation time of the proposed method and previous work

Methods	Computation Time
Proposed Method	3 seconds/character
Deformable Wavelet Descriptor	8 Seconds/character

the image. The commonly used approach is low-pass filter. Although we received high accuracy for the input data from printing image but for the input from Fax machine the system received only 71. 68%. Therefore, this system is suitable for input images from printing that is not suitable for input images from Fax machine and some input images that has low resolution.

6. Conclusion

This paper has presented Khmer printed characters recognition by using wavelet descriptor. The binary Khmer printed characters are converted to skeleton characters form. Then, the skeleton characters are represented in temporal domain. The template was stored in term of the mean value of the wavelet coefficients of the characters from 20 different fonts. The input was matched with templates, using Euclidean distance classifier, to get the distances that the minimum distance is the output of the proposed method. The method first performs the skeleton character images into 1- dimension (1-D), called temporal domain. Then, we decomposed this 1-D signal into multi-resolution wavelet coefficients. After that, to cut the noise signal out from the real signal, only low frequency of wavelet descriptor coefficients is selected as features. Then the Euclidean classifier is used as the classification process. The experience achieved 92.99% recognition rate for the input data scanned with 600 dpi resolution, 88.61% recognition rate for the input data scanned with 300 dpi resolution, 80.05% recognition rate for the input data scanned with 150 dpi resolution and 71.68% recognition rate for the input data received from the fax machine that are scanned with 300 dpi resolution. The percentage of accuracy may be considered quite high regarding these input data, that are low resolution and high noise. From these results, we also see that the system has troubles classifying noise characters input images. If the poorly input data were removed from the test samples, the accuracy would improve significantly. Also, these accuracy rates can receive better results, using the linguistic to help after the classification process. It means that post processing is needed to improve the recognition rate.

Acknowledgements

I would like to express my deepest gratitude and sincere appreciation to Prof. Hang Chan Thon, Dean of Faculty of Science, Prof. Ing Heng, Head of Continuing Education Center, Royal University of Phnom Penh (RUPP), Cambodia and Dr. Luise Ahrens, RUPP's supervisor, who always support and give the suggestions. This research project is partly supported by TRF Research Scholar No. RMV4880007.

Appendix A

The wavelet equations are given as follow:

$$r(t) = r_a^M(t) + \sum_{m-M-m_0}^{M} r_d^M(t) \tag{A.1}$$

$$r_a^M(t) = \sum_n a_n^M \tilde{\phi}_n^M(t), r_d^m(t) = \sum_n c_n^M \tilde{\psi}_n^M(t) \tag{A.2}$$

Where a_n^M is the approximation wavelet coefficient, $c_n^M$is the detail wavelet coefficient ,
$\tilde{\phi}_n^M(t)$is scaling function, $\tilde{\psi}_n^M(t)$is dilation function,
$r_d^M(t)$is called wavelet detail and $r_a^M(t)$is called wavelet smooth.

Appendix B

The minimum classifier approach is given as follow: Suppose that we define the prototype of each pattern class to be the mean vector of the patterns of that class

$$m_j = \frac{1}{N_j} \sum_{x \varepsilon w_j} x_j \tag{B.1}$$

Where j=1, 2,3.....W ,
N_j is the number of pattern vectors from class w_j and the summation is taken over these vectors. As before, W is the number of pattern classes. One way to determine the class membership of an unknown pattern vector x is to assign it to the class of its closest prototype, as noted previously. Using the Euclidean distance classifier to determine closeness reduces the problem of computing the distance:

$$D_j(x) = \|x - m_j\| \tag{B.2}$$

Where j=1, 2,3.....W and
$\|a\| = (a^T a)^{\frac{1}{2}}$ is the Euclidean norm. Then x is assigned to class w_i if $D_i(x)$ is the smallest distance.

References

1. S. Liao and L.Qin, "A study of moment functions and its use in chinese character recognition," in *Proceedings of the Fourth International Conference (Document Analysis and Recognition)*, 1997, pp. 572–575.
2. L. Qin S. Liao, A. Chiang and M. Pawlak, "Chinese character recognition via gegenbauer moments," in *16th International Conference (Pattern Recognition)*, 2002, pp. 485–488.
3. S. Dinggang and H.S. Horace, "Discriminative wavelet shape descriptors for recognition of 2-d patterns," *Journal of the Pattern Recognition Society (Pattern Recognition)*, pp. 151–165, 1999.

4. S. Arunrungrusmi O. Phaopanus and K. Chamnongthai, "Handwriting thai characters recognition using deformable wavelet descriptor," in *IConIT'2001*, 2001, pp. 78–85.
5. S. Sinthupinyo B. Kijsirikul and A. Supanwansa, "Thai printed character recognition by combining inductive logic programming with backpropagation neural network," in *IEEE Asia-Pacific Conference (Circuits and Systems)*, 1998, pp. 539–542.
6. A. Thammano and P. Ruxpakwong, "Printed thai character recognition using the hybrid approach," *IEICE TRANS. Fundamentals (Special Section on Papers Selected from ITC-CSCC 2001)*, vol. E85-A, no. 6, pp. 1236–1241, 2002.
7. C. Guangyi, "Applications of wavelet transforms in pattern recognition and de-noising," in *Master of Computer Science Thesis, Computer Science, Faculty of Science, Concordia University*, 1999, pp. 4–5.
8. S. Anbumanim and K. Bhadri, "Optical character recognition of printed tamil character," in *Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg*, 2000, pp. 4–6.
9. S.W. Lee L. Lam and C.Y. Suen, "Thinning methodologies- a comprehensive survey," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869–885, 1992.
10. T.Y. Zhang and C.Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Comm. ACM*, vol. 27, no. 3, pp. 236–239, 1984.
11. R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, pp. 372–375, 541–543, 698–700, Addison Wesley, New York, 1992.

YAHOO! MAIL Classic

Your article Thursday, March 16, 2006 12:42 AM

From: "Webb, John P" <jwebb@wsu.edu>
To: krachada@yahoo.com
Cc: "Webb, John P" <jwebb@wsu.edu>

Rachada,

By this e-mail, I notify you officially that your article, INTELLIGENT KEYPHRASE EXTRACTION FOR DIGITAL LIBRARY, has been accepted for publication in Volume 25, number 2, June 2006, of **Information Technology and Libraries**, the scholarly journal of the Library and Information Technology Association of the American Library Association.

Sincerey,
John Webb, Editor, **ITAL**

John Webb
Assistant Director for Systems and Planning
Washington State University Libraries
Pullman, WA 99164-5610
509-335-9133 FAX 509-335-6721
jwebb@wsu.edu

INTELLIGENT KEYPHRASE EXTRACTION FOR DIGITAL LIBRARY

Rachada Kongkachandra[1], Kosin Chamnongthai[1] and Chom Kimpan[2]

[1]Faculty of Engineering, King Mongkut's University of Technology Thonburi,
[2]The Graduate School, Rangsit University

Abstract

Keyphrases play an important role in the digital library. Because of the compact size and meaningful representative of a document, users are able to operate digital library services such as searching, categorizing and summarizing large electronic documents within short computation time while using less storage space. This paper presents an intelligent keyphrase extraction for digital libraries including two of its main advantages. Firstly, our proposed keyphrase extraction can extract keyphrases from a document by considering its meaning. By using this approach, some keyphrases that are ignored by the conventional keyphrase extraction are rescued if they contain meanings related to the document meaning. This is comparable to human judgment. Another advantage of our proposed keyphrase extraction is that it dynamically creates and maintains keyphrase knowledge base in self-learning behavior. Our proposed approach initially creates a keyphrase knowledge base from possible noun phrases derived from the document title. Using these keyphrases, the keyphrase knowledge base is constructed, regardless whether it is empty or expanded. The sentences including the initial keyphrases are interpreted and converted into semantic graphs, which in turn are added into the knowledge base as the meaning of the corresponding keyphrases. Since it is possible that other noun phrases in the document exists as keyphrases, additional noun phrases are used in fed back to be revaluated in the self-learning unit. The semantic graphs of all sentences, including the remaining noun phrases are compared to all the graphs within the knowledge base. If the matching score is higher than the threshold, the corresponding noun phrases will be considered as keyphrases and the semantic graphs are added to the knowledge base. The experimental results were performed with 100 documents in business and computer fields and reveal an acceptable performance as compared the conventional systems.

1 Introduction

At the present time, digital library services such as summarizing, cataloging, storing, and retrieving information can be processed systematically and trustworthily by powerful computer technologies. Many digital libraries in several organizations have been extensively implemented through various approaches in order to decrease their computation time and storage spaces. The 'keyphrase-based digital library' is an acceptable approach that can serve users with these objectives.

1.1 How Keyphrases Extraction Important for Digital Library?

To satisfy the objectives of saving time and space in the processing digital library services, a keyphrase-based approach has been introduced. Since keyphrases are accepted as a good representative of the entire document, storage a large-size

document by the list of small words is possible. Keyphrases are used as an index to access the desired documents within digital libraries. Previously, keyphrases were manually extracted and provided in the digital libraries by human judgment. Through manual keyphrase extraction the concrete extraction rules were not provided. In addition, with manual approach, it was difficult to maintain the keyphrase knowledge as modern as possible. In order to systemically extract keyphrases and conveniently update the keyphrase knowledge base, an automatic keyphrase extraction was required.

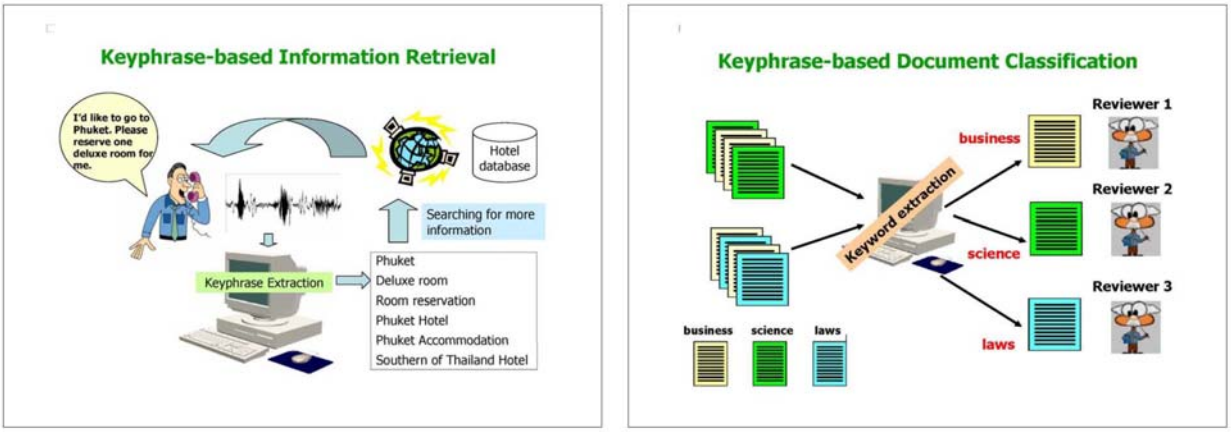


Figure 1 Examples of Keyphrase-Based Digital Library Services

Figure 1 presents the examples of the digital library services embedded with keyphrase extraction. The practical digital library applications including automatic keyphrase extraction are mentioned in [1-4]. People in [1] and [2] applied keyphrases in information retrieval to reduce the searching time. The keyphrase-based information retrieval compares a user-query with only the listed keyphrase instead of the entire document. The result of keyphrase extraction revealed the keyphrases of every document and consequently increased the amount of documents being used for searched. Therefore, the number of retrieval documents that were related to user-query was also increased. In the text summarization field [3] a keyphrase extraction was utilized in the pre-process. A keyphrase extraction was applied to produce a list of keyphrases related to each document. These keyphrases along with their sentences were then integrated to summarize the document. Such summary is compact because it contains only sentences in which the keyphrases are embedded. In machine translation applications [4], the keyphrase extraction can improve the machine translator in interpreting not only the grammatical sentences but also ungrammatical ones. Due to the translating time, it can shorten the translating by only using the keyphrases of each sentence.

It is clear that keyphrase extraction can help in improving various aspects of many applications. However, the most popular keyphrase extraction systems based on corpus, EXTRACTOR [5-6] and KEA [7], are time-consuming in preparation and expensive in maintenance. In the past decade, researchers have proposed keyphrases extraction methods that do not require corpuses, presented in [8] and [9]. In [8], a keyphrase is chosen from a noun phrase that is composed of one to three words and has high frequency of a noun phrases head noun. The top *K* keyphrase candidates for a document are then post-processed by removing single letter keyphrases and wholly contained sub-phrases. The method in [8] produces an average document score of 9.1

keyphrases per document and the authors in [8] claim that the keyphrase extraction system can work as efficiently as EXTRACTOR does without using training corpus. Subsequently, Matsuo et al. in [9] could extract keyphrases from a single document as well. They firstly extracted frequent terms and then extract a set of co-occurrences between each term and the frequent terms. If the probability distribution of co-occurrence between the term '*a*' and the frequent terms was biased to a particular subset of frequent terms, then term '*a*' is likely to be a keyphrase. For each set of fifteen extracted keyphrases, 50 percent of them were matched with the authors' keyphrases. Moreover, this keyphrase extractor was able to extract the less frequent keyphrases.

1.2 Impact of Semantic-based Keyphrase Extraction

Although the keyphrase extractors in [6]-[9] can extract keyphrases with acceptable performance, it is not guaranteed that all extracted keyphrases are related to the meanings of the given document. Since theses keyphrases are extracted based only on appearances, the meaning of keyphrases and their relations to the document are not analyzed. Numerous amounts of research about keyphrase extraction based on semantics are continuously being studied [10-13]. D. Maynard and S. Ananiadou in [11], incorporate an additional type of information i.e. the environment of the candidate term, to the C-value statistical measure as described in [10]. The *n*-grams of the environment information are extracted and calculated for weights by using corpus. However, this approach is limited because the contextual information is considered by using only their appearances. D.Maynard and S.Ananiadou in [12] solved this problem by using meanings of the contextual information. The semantic information is embedded within the context of all types of words. Not only is the frequency of occurrence computed, but also the semantic similarity. In [13], the researchers combined all the information of statistical, syntactical and semantic as criterion for considering the candidate keyphrases. Although, the extracted keyphrases were based on semantic information, the keyphrases were initially analyzed from the candidates based on their appearances. Therefore, it is not guaranteed that all extracted keyphrases are related to the meanings of the given document.

In this paper, we propose a semantic-based keyphrase extraction system to extract keyphrases that are related to the document meaning. We use the document title as the initial source to filter the initial keyphrases. The system is then self-learning and the remaining keyphrases are compared by their semantics to the initial keyphrases. In addition, the proposed keyphrase extraction system can automatically construct and expand knowledge bases containing keyphrases and their semantics for further usages. The remainder of paper is organized as follows. In section two, we briefly describe the basic idea and configuration of the semantic-based keyphrase extraction system. The knowledge base generation process and essential information will be discussed in section three. In addition, the technique used for scoring the semantic similarity is described in section four. The evaluation tests and results in keyphrase extraction application are demonstrated in section five. Finally the discussion and conclusions are given in sections six and seven, respectively.

2 Semantic-based Keyphrase Extraction System

2.1 Basic Idea.

In aspect of extracting keyphrases that are relevant to the document meaning, only statistic information likes their frequencies are not sufficient. The semantic information in the document, which can be linked to the meaning of whole document content, is required. M. Montes-y Gmez in [14] mentioned that the meaning of the document is mostly expressed in the document title. Therefore, we employ the title of the document as a resource for determining document's keyphrases.

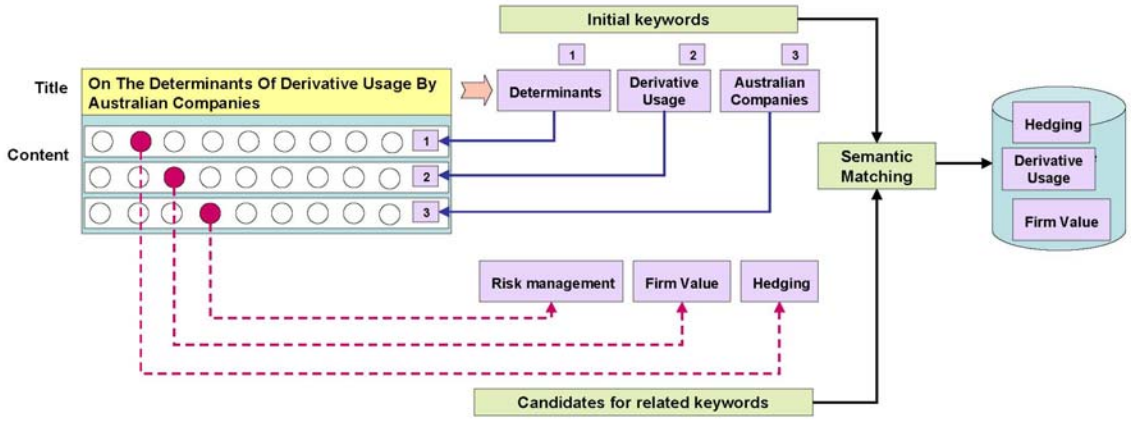


Figure 1 Basic Idea of the proposed system

Figure 1 demonstrates our proposed idea that is to extract the related keyphrases based on semantic-based keyphrase model constructed from the title sentence. All possible noun phrases within the title sentence, called initial keyphrases represented as box in Fig.1, are used as key indicators for all the sentences in the document content. Within each sentence that the initial keyphrases are embedded, all remaining noun phrases represented as shaded circle in Fig.1 having the chance to become the related keyphrases, if they have the similar meanings to the initial keyphrases.

2.2 System Configuration

Figure 2 illustrates an overview of the proposed semantic-based keyphrase extraction system that attempts to extract keyphrases from a document without using corpus in creating keyphrase models. In our proposed system, the initial keyphrases are automatically extracted from the document title and then reused for self-learning and other related keyphrases. All components of the self-learning keyphrase extraction system are described in the following section with respect to their contributions in the system.

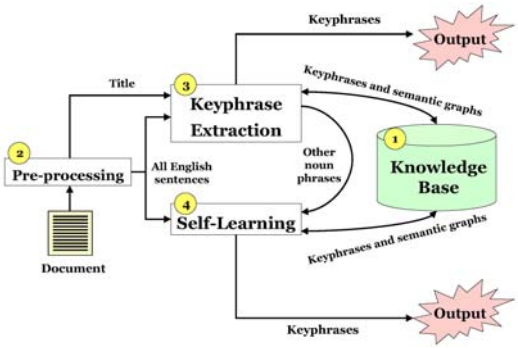
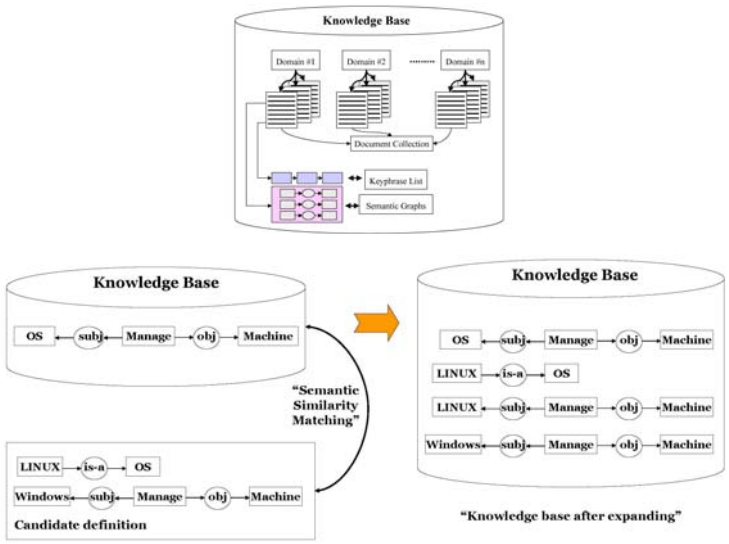


Figure 2 Semantic-based Keyphrase Extraction System



(a) (b)

Figure 3 The example of keyphrase knowledge used in the proposed system

2.2.1 Knowledge Base

The knowledge base is designed as the inventory of keyphrase and their meanings. We will subsequently refer a keyphrase and its meanings to "keyphrase knowledge". An example is demonstrated in Fig. 3(a) Figure 3(b) that illustrates how the keyphrase knowledge is updated.

2.2.2 Pre-processing Unit

The pre-processing unit is employed to segment the documents into two parts as title and content from various sources for keyphrase extraction and self-learning units. The title of the document is utilized in the keyphrase extraction unit for pulling out the initial keyphrases. In addition, the keyphrase extraction unit also uses the document's contents for selecting all keyphrase-embedded sentences for succeeding semantic graph conversion. Besides this, the content is used in the self-learning unit for finding candidates and generating semantic graphs.

2.2.3 Keyphrase Extraction Unit

The objective of the keyphrase extraction unit is to extract keyphrases and construct or expand the knowledge base. All candidates that are all possible noun phrase matches with the title are filtered as initial keyphrases.

Subsequently, all related sentences describing each keyphrase are collected and then represented as semantic graphs. In this paper, we select all sentences in which keyphrases are embedded as the meanings of the keyphrases. These meanings are then converted from the natural sentences into semantic graphs by considering the specified conversion rules. The details will be mentioned in the next section. Finally, these initial keyphrases are stored in the knowledge base as references.

2.2.4 Self-learning Unit

The self-learning unit is employed to extract the related keyphrases by reconsidering other noun phrases in the content of the document. In the same way as the keyphrase extraction unit, all noun phrases satisfied the previous conditions are selected as candidates. Next, all candidates are then used as index, searching over the document for all sentences that they are included in. These sentences are then transformed into semantic graphs by the same conversion rules as in the keyphrase extraction unit. Finally, all candidates are essential to be proved for their semantic similarity compared to the title. The top candidates with high similarity scores are selected as related keyphrases.

3 Knowledge Base Generation

3.1 Document Title Extraction

We present an automatic keyphrase system that uses electronic document as input. The majority of available documents are created in portable document format (pdf) because to their decreasing sizes. To extract a title sentence from a document, we firstly converted these documents into the well-known document format XML by using PDF2XML command within Acrobat Reader products. The phrase within the first line including <H1> and </H1> tags of the XML files is determined as the title of sentence. In addition to the advantage of using XML, we can exclude the figures and tables by removing all lines within <Figure>… </Figure> and <Table>… </Table>, respectively.

3.2 Initial Keyphrase Extraction

After we extract the document title, it is syntactical analyzed by Machinese Syntax based on Functional Dependency Grammar [15] for syntactic relation and morphology information. From the research of [16], [17], [18] and [19] that admit noun phrase as a basic pattern to represent keyphrases, we extract noun phrases and then determine them as the keyphrase if they are satisfied the following conditions:

1. A keyphrase should be composed from the appropriate lengths of word. Keyphrase is generally contained words from one to five.
2. A keyphrase is not too general in meaning. It should be excluded from the stop word list.
3. A keyphrase should not be contained in other keyphrases. The more specific keyphrases should be selected.

3.3 Semantic Representation

In a document the candidate/keyphrase meaning within one or more sentences may be included in the individual candidate/keyphrase. To represent an English sentence in machine understandable form, a knowledge representation is required. A conceptual graph [20] is selected to represent the meaning of a keyphrase, this is necessary because of the direct map between a graph and a natural sentence represents the meaning. In this paper, we added some semantics information to the original conceptual graph. Therefore, we will refer to the conceptual graph as "semantic graph" to indicate that a conceptual graph represents a meaning.

A semantic graph consists of three nodes, i.e. two concept nodes and one relation node. A concept node consists of three data, i.e. a words name, a general concept name and a link to hyponym information. We gain two sets of information i.e. hypernyms and hyponyms, the well-known lexical database named “Wordnet” [21]. The hypernyms reveals the list of general concepts of word “OS” ordered from the least general to the most one. The hyponyms are list of the specific concepts of word “OS”. Figure 4 shows how to define the concept node into the semantic graph.

In representing the relation node, we use a part of the output that is gained from the functional dependency grammar parser, “Machines Syntax”. The parser outputs the syntax, morphology and syntactic relation. The syntactic relation shows the relational link from a word to others based on their functions, which is similar to the conceptual relation in semantic graph.

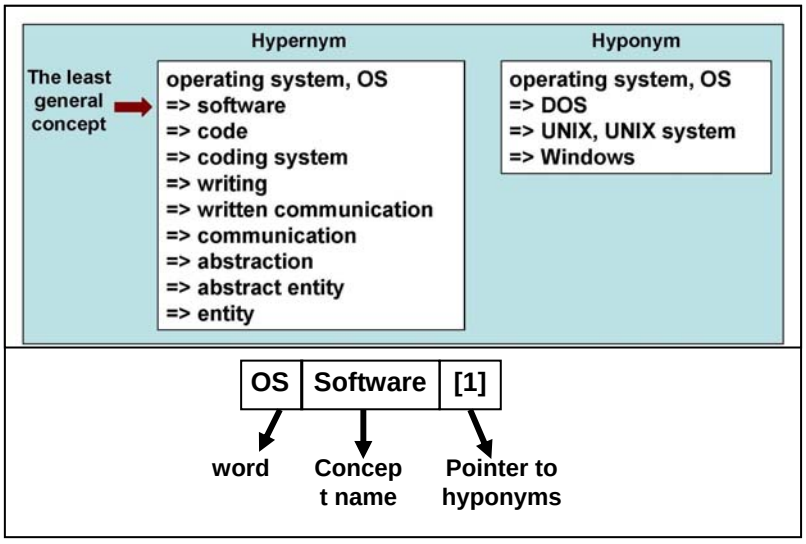


Figure 4 The concept node representation

3.4 Semantic Interpretation

The semantic interpretation is the process to convert natural sentences into specified knowledge representation, referred to as the semantic graph in this paper. The details are presented in Figure 5 as follows.

Input:
1. An English sentence.
2. A Lexical database.
3. A English Parser

Output: The semantic graphs.

Process:

1. Parsing a sentence for syntax, morphology and syntactical relations. An example of the result of parsing is shown in the table in Figure 5.
2. Selecting the word that has the syntactic relation as 'main' as the starting point.
3. Representing its concept node based on the idea in Section 3.2.
4. Finding all links of other words that have the linkages connected to it.
5. Following the link to the connected word that has the morphological tag as 'Noun', 'Verb' and 'Adjective'.
6. Representing the relation node by using the syntactic relation name of the connected word.
7. Repeat step 3 until there are no links left.

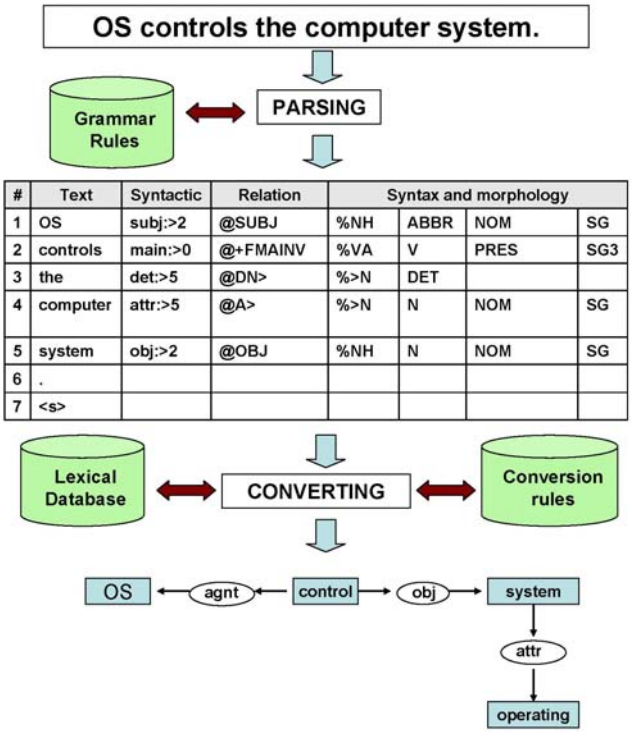


Figure 5 Semantic Interpretation Process

4 Semantic Similarity Matching

Determining the semantic similarity between two semantic graphs is depended on two features i.e. their structures and values. Two semantic graphs will simultaneously be

accepted if they are identical both in structure and content. However, many authors usually use different sentence structures and words while describing the same things. Therefore, using only this determination policy is too limited and as a consequence many semantic graphs are rejected.

In this paper, we propose using the determination policy for scoring the similarity between two different semantic graphs. The policy is described as follows:

1. If a candidate and a keyphrase is the same and only if there is at least one sentence of candidate's meanings that is equaled to the meanings of the keyphrase.
2. Since one sentence can derive one or more semantic graphs, all semantic graphs are used to determine the sentence similarity between the candidate and the keyphrase.
3. In each semantic graph, all elements i.e. two concept nodes and one relation node are matched. Its average score is computed and kept as semantic graph score.

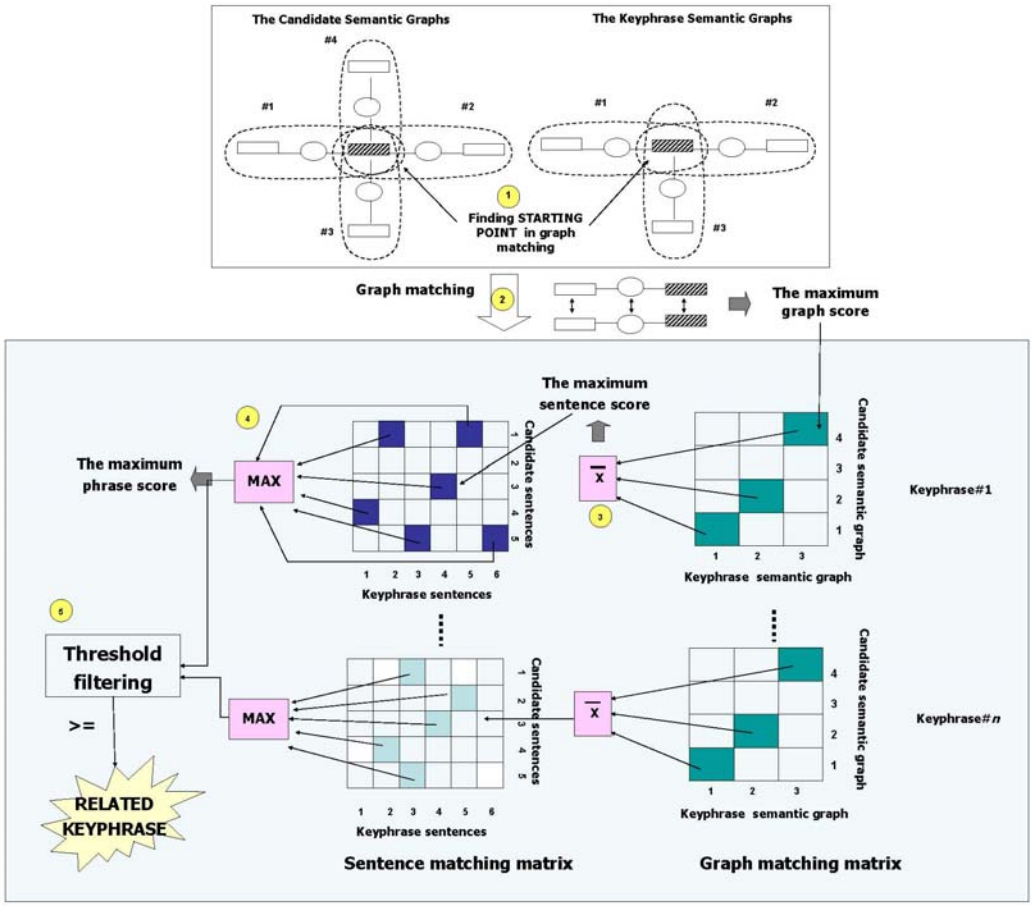


Figure 6 The process of semantic similarity verification

Figure 6 demonstrates the determination policy for matching the meanings of a candidate and a keyphrase.

(1) Starting Point Selection

In Figure 6 two semantic graphs are described with different semantic graph structures. The candidate has four semantic graphs and the keyphrase has three semantic graphs. When the number of semantic graphs is not identical, the number of matching is based on the lesser one. The matching process starts from the both core nodes. The core node is the conceptual nodes having the most arcs linked to them as shown as shaded rectangles in Figure 6(a).

(2) Graph Matching

All semantic graphs connected to the core node of candidate are compared with ones of keyphrase. The details of matching are zoomed in Figure 6(b). From the example, the graph-matching matrix is created in four rows and three columns, according to the number of sub-graphs of keyphrase and candidate, respectively.

Each element in the graph matching matrix is charted as the graph score computed by the (1), which is determined accordingly to the third determination policy.

$$graph\_score(i,j) = \frac{\sum_{n=1}^{3} hit(n))}{3} \qquad (1)$$

Where *hit(n)* is the number of matched nodes between candidate and keyphrase nodes in each semantic graph.

(3) Sentence Matching

According to the second policy, in which a sentence composes of one or more semantic graphs, all semantic graphs are necessary for sentence matching. Therefore, the sentence-matching matrix is built and the average score of all semantic graphs is calculated by (2) and filled in the matrix, as shown in the sentence-matching matrix.

$$sentence\_score(i,j) = \frac{\sum_{i=1}^{k} (\max_{j=1}^{c} (graph\_score(i,j)))}{k} \qquad (2)$$

(4) Phrase Scoring

In general, the important phrases are mentioned several times in a document. It implies that the meanings of a candidate or keyphrase may be composed from many sentences. In Figure 6, the candidate has five sentences and the keyphrase has six sentences describing them.

$$phrase\_score(i,j) = \max_{i=1}^{k} \max_{j=1}^{c} (sentence\_score(i,j)) \qquad (3)$$

With respect to the first determination policy, we used the maximum score computed by (3), as the representative for threshold filtering. The candidate that has the phrase-score over the specified threshold is accepted as the related keyphrase. The related

keyphrase and its semantic graphs are subsequently added into the keyphrase knowledge for later usages.

5 Evaluations and Results

In this section we show the experimental results of semantic-based keyphrase extraction system with self-learning knowledge base. For evaluating the performance of the proposed function, the keyphrase extraction system is set as shown in Table1.

We compare our experimental results with three conventional keyphrase extraction systems i.e. EXTRACTOR [5-6], KEA [7] and Matsuo's system [9]. Conventional keyphrase extraction systems in [5-7] are representatives of corpus-based keyphrase extraction systems. Matsuo's system in [9] is a representative of keyphrase extraction systems in non-corpus usage group. A set of 100 technical papers from the disciplines of business, computers, education, medicine and psychology were used for the test documents. The six documents from [22] were used as the standard for testing our proposed system with EXTRACTOR and KEA. One technical paper in [23] was used for the evaluation of the Matsuo's system. The remaining ninety-three papers were used for proving the powerful knowledge base.

Table 1: Experiment Settings

Configuration	**Practical Setting**
Conventional keyphrase extractor	1. EXTRACTOR 2. KEA 3. Matsuo's System
Number of domains	5 domains: business, computer, education, medicine and psychology
Number of documents	100 documents
Number of desired keyphrases	5, 10 and 15 keyphrases
Linguistic tools	Machinese Syntax for part-of-speech tagging and parsing
Threshold of keyphrase acceptance level	0.5, 0.8 and 1.0

To evaluate the performance of our proposed functions, we set the experiments use a percentage of the precision and recall. They are obtained from the following equations:

$$\Pr ecision = \frac{R}{C} \quad (4)$$

$$\mathrm{Re} call = \frac{R}{K} \quad (5)$$

Where R is the relevant keyphrases extracted by the extraction system.
C is all the candidates in the specified domain.
K is the total number of real keyphrases.

5.1 The comparison performance between our proposed keyphrase extraction system and corpus-based keyphrase extraction systems

Table 2 illustrates an example result compared to EXTRACTOR and KEA. We compared their results with the same document entitled "WorldBeat: Designing a

Baton-Based Interface for an Interactive Music Exhibit" [22]. The bold and italic phrases were those which were completely matched to the list of keyphrases as defined by the author. From the five extracted keyphrases, our proposed system can extract additional keyphrases as "worldbeat exhibit" and "music". This increases the performance of our proposed keyphrase extraction system over others. All 100 trial documents were then tested and the percentage of precision and recall are illustrated as in Table 3, respectively.

Table 2: An example of the proposed system compared to the corpus-based keyphrase extraction system

EXTRACTOR	KEA	KEA2	The proposed system
exhibit	Baton Based	Design Patterns	***interface design***
instruments	***interactive exhibits***	input device	***worldbeat system***
user feedback	***interface design***	user interface	***worldbeat exhibit***
user interface design	user interface	user ***interface design***	User interface
WorldBeat system	***WorldBeat system***	virtual realities	***music***

Table 3: The performance of the proposed keyphrase extraction system compared to corpus-based systems

Keyphrase Approaches	Performance	
	% of Precision	% of Recall
EXTRACTOR	75.23	55.95
KEA	76.05	57.86
Our proposed	**80.21**	**60.23**

5.2 The comparison performance between our proposed keyphrase extraction system and non-corpus usage keyphrase extraction systems

Since the Matsuo's system is not publicly provided on the Internet as EXTRACTOR and KEA, we evaluated the proposed system by using the same document, entitled "Computing Machinery and Intelligence" [23]. Table 4 demonstrates the extracted keyphrases from Matsuo's approach and ours. Our proposed keyphrase extraction system extracted additional keyphrases as "intelligence" and "computation" as shown as shaded rows in Table 4.

Table 4: Example results of the proposed keyphrase extraction system compared to non-corpus systems

Matsuo's System [9]	The proposed system
digital computer	***digital computer***
storage capacity	***storage capacity***
imitation game	***imitation game***
machine	***discrete-state machine***
human mind	***intelligence***
universality	compute machinery
logic	***computation***
property	Object
mimic	sense
discrete-state machine	man

5.3 The improvement of keyphrase extraction system by including self-learning knowledge base

In this paper, we also proposed the use of knowledge base for storing keyphrase list and their meanings. With the powerful of self-learning approach, our system can automatic create or expand the knowledge base. The knowledge base is beneficial to the next cycle of extraction. Table 5 shows the obtained results when we extract five keyphrases from the same set of test documents up to five cycles. From Table 5, the percentage of matched keyphrases are continually increased and then saturated at the fourth cycle of extraction. This event implies that from the empty knowledge base, our proposed system needs some periods for learning the system. From our experiment the number of learning cycles was never more than four cycles.

Table 5: The performance of the proposed system in various cycles of extraction

# of extraction cycle	%of matched keyphrases
1	54.90
2	66.98
3	74.12
4	75.55
5	75.55

5.4 The effect of desired keyphrase numbers to our proposed keyphrase extraction system

Since the number of author-defined keyphrases is varied from documents to documents, the appropriate numbers of keyphrases should be approximated in order to increase the system performance. In our experiments, we found that the different numbers of desired keyphrase yields the different performance as shown in Table 6. Table 7 demonstrates the pair wised comparative tests for the effects of desired keyphrase numbers to the performance of our proposed approach. The null hypothesis is "there is no difference when various keyphrases are desired". With the confidence level as 95%, the computed t scores are in the rejected regions ($< -t_{11,0.05}$ and $> t_{11,0.05}$). Therefore, the hypothesis is rejected.

Table 6: The performance of the proposed system in various numbers of desired keyphrases

Number of desired keyphrases	Performance	
	% of Precision	% of Recall
5	75.55	52.81
10	77.48	54.23
15	80.21	58.57

Table 7: The pair wise comparison for the effect of various numbers of desired keyphrases

Hypothesis test for the Effect of Number of Keyphrases and Recall

	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval		t	df	Sig. (2-tailed)
				Lower	Upper			
5-keyphrase v.s. 10-keyphrase	0.01833	0.22148	0.06393	-0.12239	0.15905	0.287	11	0.780
5-keyphrase v.s. 15-keyphrase	-0.04917	0.32411	0.09356	-0.25509	0.15676	-0.526	11	0.610
10-keyphrase v.s. 15-keyphrase	-0.06750	0.13532	0.03906	-0.15348	0.01848	-1.728	11	0.112

Hypothesis test for the Effect of Number of Keyphrases and Precision

	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval		t	df	Sig. (2-tailed)
				Lower	Upper			
5-keyphrase v.s. 10-keyphrase	0.01833	0.22148	0.06393	-0.12239	0.15905	0.287	11	0.780
5-keyphrase v.s. 15-keyphrase	-0.04917	0.32411	0.09356	-0.25509	0.15676	-0.526	11	0.610
10-keyphrase v.s. 15-keyphrase	-0.06750	0.13532	0.03906	-0.15348	0.01848	-1.728	11	0.112

6 Discussion

The experiments of the previous section demonstrate that the self-learning keyphrase extraction system presented in this paper is flexible because it has the ability to extract the related keyphrases without using training corpus. Based on the obtained results, the following observations are discussed.

6.1 The overall performance in term of extraction precision

Evaluating the proposed system with the conventional keyphrase extraction systems in corpus-based group, our system obtained the higher performance, a percentage of related keyphrases increased. Since those systems can extract the keyphrases based on their frequency of occurrences some keyphrases that were not exactly related to the document were extracted. The examples are ‘instruments’ and ‘input device’ are shown in the second and fourth column of Table 2. Evaluating the proposed system with the conventional keyphrase extraction system using no corpus, our system performed comparably. From the results in Table 4, four of them, represented in italic, were identical. However, the Matsuo’s system had only one keyphrase that matched exactly to the author-defined keyphrase, while our proposed system obtained three keyphrases that were identical to the author-defined keyphrases. Even the conventional system can extract the keyphrase that had less frequency of occurrences by considering the frequency of its co-occurrence terms. The system is also limited if the keyphrase had less frequency of co-occurrences. For example, the keyphrase ‘intelligence’, that is defined by the author but it is mentioned only once in the document, could be not extracted by the conventional system because its frequency

was less than three. Because the title of this paper is ‘Computing Machinery and Intelligence’ and it contains ‘intelligence’, our proposed system could extract it.

6.2 The effect of knowledge base for the next cycle of extraction

Table 5 illustrates the improvement of the system performance. However, the percentages of matched keyphrases are saturated at the fourth cycle of extraction. The possible cause for the saturated is the regulation used in extracting keyphrases.

1. A keyphrase should be composed from the appropriate lengths of word. (1-5 words/phrase)
2. A keyphrase is not too general in meaning. It should be excluded from the common word list.
3. A keyphrase should not be contained in other keyphrases.

By each rule, all possible candidates are filtered out to limited numbers of noun phrases. If the documents used for extracting are not increased, all possible noun phrases could not be increased as well. Therefore, the keyphrases could not be extracted anymore. The saturated state can therefore occur in some cycle of extraction.

6.3 Limitations

The self-learning keyphrase extraction system was able to yield more keyphrases which were matched to the author-defined keyphrases than that of the conventional systems, however, 100 percent of author-defined keyphrases could not be recalled. The limitations of the proposed system are depended on three factors:

1. The initial keyphrases;
2. The related keyphrases;
3. The content in the knowledge base in the previous cycle.

Since the initial keyphrases were extracted by considering all noun phrases from the title of the document, the number of noun phrases contained in the title was affected. The small number of noun phrases yielded a small number of initial keyphrases as well. In addition, it was essential that each noun phrase occurred at least once; otherwise, the proposed system is not able find any meaning for further matching. In the process of extracting related keyphrases, the sentences with the same meanings are possibly converted into different semantic graphs. The similar score was obtained from these semantic graphs but was not as high as expected. The related keyphrases located in the mentioned sentences were therefore not extracted.

In the succeeding extraction cycles, all semantic graphs of candidates were compared to the ones of all keyphrases inside the knowledge base that was constructed or expanded from the previous cycle. In the expanded knowledge base sometimes the contradicted knowledge was also stored. This kind knowledge can decrease the performance of the system.

7. Conclusion

This paper presents a self-learning keyphrase extraction system for a non-corpus environment. With the powerful self-learning approach, the initial and related keyphrases that are identical to the author-defined keyphrases in sense of meaning, can be extracted. In addition, the knowledge base constructed and expanded by our system can improve the performance of keyphrase extraction. The proposed keyphrase extraction system can produce keyphrases in high performance with 65.06%, 78.10% and 85.90% of matched keyphrases compared to the author-defined keyphrases for 5, 10, and 15 of desired keyphrases, respectively.

8 Acknowledgement

This paper is based upon work supported by the Thailand Research Fund under grant No. RMU4880007 of TRF Research Scholar. The authors also wish to thank Connexor Co. Ltd for the use of the academic license of Machinese Syntax which was used in our experiments. Finally, we would like to thank Dr. Pinit Kumhom at KMUTT for his comments during the system development.

References

[1] Gutwin, C., Paynter, G.W., Witten, I.H., NevillManning, C.G. and Frank, E, 1998, "Improving Browsing in Digital Libraries with Keyphrase Indexes", **Technical Report, Department of Computer Science**, University of Saskatchewan, Canada.

[2] Jones, S., 1999. "Phrasier: a System for Interactive Document Retrieval Using Keyphrases", **Proceedings of SIGIR99**, pp. 160-167.

[3] Kupiec, J., Pedersen, J., and Chen, F., 1995, "A Trainable Document Summarizer," **Proceedings of the18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**, pp. 68–73.

[4] Zajac, R. and Casper, M., 1997, "The Temple Web Translator," **1997 AAAI Spring Symposium on Natural Language Processing for the World Wide Web**, pp. 68–73.

[5] Turney, P. D., 1997, "Extraction of Keyphrases from Text: Evaluation of Four Algorithms," **Technical Report NRC 41550**, National Research council.

[6] Turney, P. D., 2000, "Learning Algorithms for Keyphrase Extraction," **Information Retrieval**, Vol. 2, No. 4, pp. 303-336.

[7] Witten, I.H., Paynter, G.W., Frank, E, Gutwin, C., and NevillManning, C.G., 1999, "Kea: Practical Automatic Keyphrase Extraction," **Proceedings of the Fourth ACM Conference on Digital Libraries (DL'99)**, pp. 254-256. (Poster presentation.)

[8] Barker, K. and Cornacchia, N., 2000, "Using Noun Phrase Heads to Extract Document Keyphrases," **Proceedings of the 13th Biennial Conference of the Canadian 12 Society on Computational Studies of Intelligence: Advances in Artificial Intelligence**, pp.40-52.

[9] Matsuo, Y., and Ishizuka, M., 2004, "Keyphrase Extraction from a Single Document Using Word Co-Occurrence Statistical Information", **International Journal on Artificial Intelligence Tools**, Vol. 13, No. 1, pp. 157-169.

[10] Frantzi, K.T. and Ananiadou, S., 1996, "A Hybrid Approach to Automatic Term Recognition", **Proc. Int. Conf. on Natural Language Processing and Industrial Applications**, Moncton , Canada , pp. 93–98.

[11] Maynard, D., and Ananiadou, S., 1999, "Identifying Contextual Information for Multi-Word Term Extraction", **The 5th International Congress on Terminology and Knowledge Engineering (TKE 99)**, pp. 212-21.

[12] Maynard, D., and Ananiadou, S., 1998, “Acquiring Contextual Information for Term Disambiguation”, **Proceedings of Computerm '98 Workshop on Computational Terminology (COLING/ACL '98)**, Montreal, Canada, pp. 86-91.

[13] Maynard, D. and Ananiadou, S., 2000, “TRUCKS: A Model for Automatic Multi-Word Term Recognition”, **Journal of Natural Language Processing**, Vol. 8, No. 1, pp. 101–125.

[14] Montes-y-Gmez, M., Lpez-Lpez, A. and Gelbukh, A.F., 1999, “Document Title Patterns in Information Retrieval,” **Lectures Notes in Computer Science**, Vol. 1692, pp. 364–367.

[15] Voutilainen, A. and Heikkila, J., 1993, “An English Constraint Grammar (ENGCG): A Surface-Syntactic Parser of English,” **The Fourteenth International Conference on English Language Research on Computerized Corpora**, pp. 189–199.

[16] Bourigault, D., 1992, “Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases”, **Proc. of the 14th International Conference on Computational Linguistics (COLING '92)**, Nantes, Frances, pp. 977-981.

[17] Ananiadou, S., 1994, “A Methodology for Automatic Term Recognition”, **Proc. of the 15th International. Conference on Computational Linguistics (COLING ’94)**”, Kyoto, Japan, pp.1034–1038.

[18] Lauriston, A., “Automatic Recognition of Complex Terms: Problems and the TERMINO Solution”, *Terminology*, Vol. 1, No. 1, 1994, pp.147-170.

[19] Justeson, J. and Katz, S., 1995, “Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text”, **Natural Language Engineering**, Vol. 1, pp. 9-27.

[20] Sowa, J.F., 1984, “Conceptual Structures: Information Processing, **Mind And Machine**”, Addison-Wesley, Reading, MA.

[21] Budanitsky, A. and Graeme H., 2001, “Semantic Distance in Wordnet: An Experimental, Application-Oriented Evaluation of Five Measures”, **Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources**, Pittsburgh, USA. (http://citeseer.ist.psu.edu/budanitsky01semantic.html)

[22] Jones, S. and Paynter, G. W., 2003, “An Evaluation of Document Keyphrase Sets,” **Journal of Digital Information**, Vol. 4, No. 1. (http://jodi.ecs.soton.ac.uk/Articles/v04/i01/Jones/)

[23] Turing, A.M., 1950, “Computing Machinery and Intelligence”, **Mind**, Vol. 49, pp. 433-460.

Informatica

User: **rdkpansy** • Role: Author **Change Role** | **Sign-Out** **Home**|**My Profile**|**About**|**Help**

New submissions

ID Type	Authors Title	Submitted	Round	Decision Decision date	Revision due Revision uploaded
INFOR0804-012 Article	Rachada Kongkachandra, Kosin Chamnongthai Abductive Reasoning for Keyword Recovering in Semantic-based Keyphrase Extraction	2008-04-19		- -	- -

Start new submission

<< Back to Author menu

Abductive Reasoning for Keyword Recovering in Semantic-based Keyword Extraction

Rachada Kongkachandra* and Kosin Chamnongthai†
*Dept. of Computer Science, Faculty of Science and Technology,
Thammasat University, Pathumthani, 12121, Thailand
e-mail: *krachada@yahoo.com*
†Dept. of Electronics and Telecommunication Engineering
Faculty of Engineering, King Mongkut's University of Technology Thonburi,
Bangkok,Thailand 10140 e-mail: *kosin.cha@gmail.com*

***Abstract*—This paper proposes semantic based keyphrase recovery for domain-independent keyphrase extraction. In this method, we add a keyphrase recovery function as a post-process of the conventional keyphrase extractors in order to reconsider the failed keyphrases by semantic matching based on sentence meaning. We also add the Domain Identification Function to determine the related domain of the keyphrases by using keyphrases extracted from the conventional systems in order to make the system as domain-independent. The semantic matching is performed to compare the similar meanings between ones of failed keyphrases and ones in the knowledge base. Therefore, the failed keyphrases that are matched by semantic matching are recused as keyphrases. The experiments with the summary sentences in 60 articles of IEICE Transactions on Information and Systems and glossaries from four resources are performed in initializing Domain Knowledge Base. Other summary sentences in 100 articles of IEICE Transactions on Information and Systems and in 15 chapters in a Computer Information System textbook are experimented in recovering the failed keyphrases. The results reveal that the proposed method increases the average performance of conventional EXTRACTOR and KEA approximately by 33.16 and 41.30% of precision, and 36.10 and 39.17% of recall, respectively.**

***Index Terms*—Keyword Extraction, Semantically Matching, Abductive Inference, Non-existing Keyword, New-defined Keyword.**

I. Introduction

Automatic keyphrase extraction plays important role for automatically spotting the keyphrases from the documents in order to assist people to search the required documents. Since the extracted keyphrases act as the representatives of document content, the contribution of keyphrase is also to help human for quickly understanding the contents of document. However, keyphrase composes of several number of keyphrases, keyphrase extraction is considered to cover the keyphrase extraction. The automatic keyphrase extraction system can be utilized in several applications such as information retrieval, text summarization, machine translation, speech understanding and so on. The extracted keyphrases affect the performance of these applications, and the high performance keyphrase extraction is required in term of speed, relativeness, accuracy, and robustness. However, since some keyphrases have no fixed pattern, and some are keyphrases that do not exist as the words in the document, it is difficult to extract the keyphrases by directly matching with database. There are several approaches working on developing the efficient keyphrase extraction systems. We categorize the existing researches into three groups based on their algorithms i.e using statistics, machine learning, and semantically matching. In the statistical-based approaches such as [1],[2], [3], and [4], they firstly extract the possible words sequences with no stop words and punctuation as candidates. The researchers in ([1],[2]) use all kinds of phrases as the utilized candidates while people in [3] and [4] are interested only two-words and ***n*** noun phrases, respectively. All filtered candidates are then verified their goodness with different methods varying from simply counting the candidates' occurrences ([3],[4]) to Naïve Bayes algorithm ([1],[2]), respectively. Candidates with high frequency are selected as keyphrases. The advantage of these approaches are simplicity. However, their accuracies are decreased because some actual keyphrases with less frequency are ignored and they need a large amount of data for training. In machine learning-based extraction, there are two keyphrase extraction systems i.e. EXTRACTOR and GenEX developed by P. Turney in [5] and [6]. They employ the similar way as [1],[2], but different number of features, to find the possible candidates. To determine keyphrases in the midst of candidates, each candidate is then matched to the keyphrases generated by using C4.5 decision-tree induction algorithm in training process. GenEX is the enhanced system of EXTRACTOR. It uses genetic algorithm to firstly tune up the 12 parameters used in candidate filtering process. From this approach, the accuracy of the extracted keyphrases is increased because they are determined from more details parameters. In addition, they can be applied in several domains of interest. However, they are complicate in term of training corpus requirements. The corpus used in these systems needs human expert to tag each phrase as keyphrase and non-keyphrase.

In semantically matching, the hybrid keyphrase extraction system between statistical and semantical approaches as TRUCKS has been developed. ([7],[8], [9] and [10]), attempt to find the keyphrases focusing on relativeness criteria. This approach has advantage in term of relativeness. Unfortunately, it suffers in simplicity qualification

because it needs to compute several scores. Although these current keyphrase extraction systems give the good performances, they still face the problems of rejecting actual keyphrases. A boosting algorithm proposed by Jordi Vivaldi et.al in [11], is used to solve this problem. They use AdaBoost Algorithm to find a highly accurate classification rule by combining multiple classifiers such as semantic content extractor, context analyzer, Greek and Latin forms analysis, and collocational analysis. This approach can improve the accuracy of the linguistic-based keyphrase extraction system, but has several disadvantages. It uses specific format of document, SGML, as input, domain-specific in medical domain, and corpus requirement for training.

In this paper, we focus to improve the conventional keyphrase extraction systems in terms of accuracy, and domain-independent. To improve the accuracy, we add the post-process for recovering the failed keyphrases by semantical matching. The semantical matching employed in this paper is based on sentence meaning so that not only keyphrases that do not exist as words in the sentence but also the ones that are ambiguous are rescued. For domain-independent, we also add the Domain Knowledge Base Initialization Function in order to create the initial knowledge base, and Domain Identification Process that utilizes keyphrases extracted from conventional extractors to determine the related domain and automatically update the keyphrases in the knowledge base.

II. Problem Analysis and Basic Ideas

A. Problem Analysis and Basic Concept

Normally, the conventional keyphrase extraction systems output keyphrases and non-keyphrases as shown in Fig.1. They are categorized into two groups according to the causes of their rejection i.e linguistics and statistics. In the first group, there are some actual keyphrases containing ambiguous words (the words that can be counted as noun, adjective, and verb simultaneously) and recognized as adjective or verb rather than noun according to grammar. Though they are keyphrases, the system ignores them, because the keyphrase is assumed as noun. Not only this, the order of words in each keyphrase is also important, some of them are rejected because they have different sequences compared to the author defined keyphrases. In another group, non-keyphrases are ignored because they occur in less frequencies and locate in insignificant location. In this paper, we give the first priority of noun consideration for all ambiguous words in order to expand the possibility of being keyphrases.

In another hand, there are some actual keyphrases that do not frequently occur, do not place in the significant position such as the first sentence in each paragraph, and do not order as the same as author definitions are determined as non-keyphrases as shown in Fig.1. However, these keyphrases have meaning related with the document content. To solve this problem, we propose semantic matching to find the relevant meaning and rescue them as keyphrases.

III. Abduction for Keyphrase Extraction

Abduction, originated by Charles Pierce, is a particular kind of hypothetical reasoning. In the simple case, it has the form:

From Q and $Q \leftarrow P$

infers P as a possible "explanation" of Q.

Abduction is the reasoning backward from the result to the cause. Since Q is a fact in the knowledge base and the condition that if P is true then P is true is preserved, there is a reason to suspect that P is true.

There are several applications using abduction as the reasoning technique such as medical diagnosis, language understanding and speech recognition. The new knowledge takes for granted to be added to the knowledge base if it is not inconsistent with ones in the knowledge base.

In this paper, we apply the concept of abduction as keyphrase recovering technique for keyphrase extraction system as presented in Fig.2. We assume that the well-formed knowledge base contains domains and their relevant keyphrases with the condition that if we mention to the specified domain then the keyphrases related to it are valid. The shaded box in the Fig. 2(a) and (b) presents the knowledge base of computer domain. Unfortunately, non-keyphrases as "CD-ROM" in Fig. 2(b) is not included in the knowledge base so it is then rejected. From the definition of abduction, if we assume the non-keyphrase as valid keyphrase and backward reasoning in the knowledge base, the non-keyphrase that is consistent with the existing knowledge base is rescued. To prove the consistency of the non-keyphrase, we use the integrity of meaning as condition. The non-keyphrase which has meaning consistent with others in the background knowledge is recovered.

IV. Keyphrase Extraction System with Semantic-based Keyphrase Recovery Function

Figure 3 illustrates the configuration of keyphrase extraction system with Semantic-based Keyphrase Recovery function. They are separated into two modes including training and recovering modes. The training mode is for initializing the main knowledge base while the recovering mode is for recovering the false rejected keyphrases. From Fig. 3, the main knowledge base is called "Domain Knowledge Base", as located on the top left of figure. The contents of the domain knowledge base are keyphrases, their definitions, and their relevant domains. In the following subsections, the structure of the domain knowledge base is firstly presented, and then the configuration details of training and recovering modes, respectively.

A. Domain Knowledge Base

The domain knowledge base acts as the human brain. It comprises from the links of domain nodes. It is organized in the hierarchical format ranking from the most general to the most specific nodes based on their meaning. The logical view of domain knowledge base is illustrated in Fig. 4(a). The internal structure of each domain is

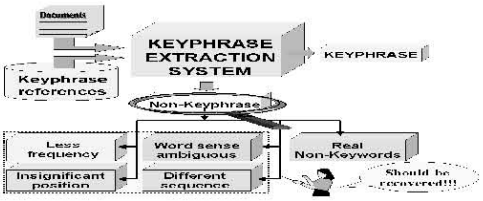


Fig. 1: Analysis of Conventional Keyphrase Extraction Systems Problem

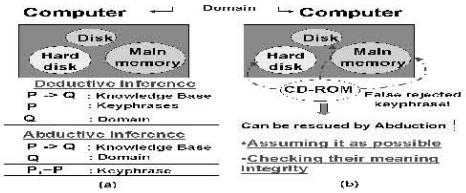


Fig. 2: Abduction for Keyphrase Extraction

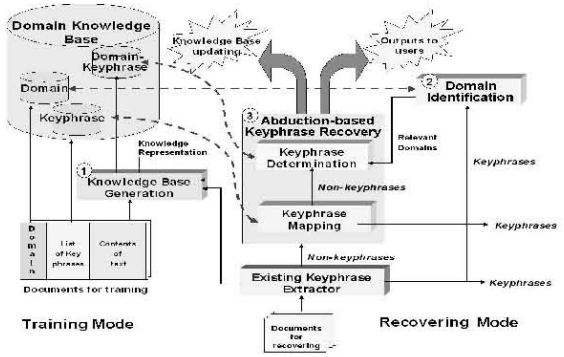


Fig. 3: Configuration of Semantic-Based Keyphrase Recovering Function

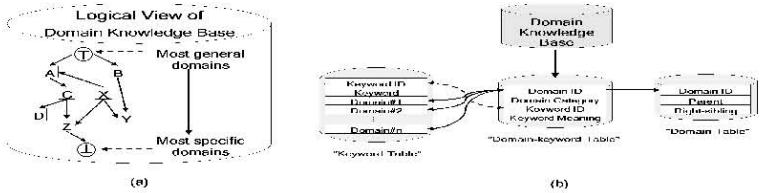


Fig. 4: (a) A logical view of Domain Knowledge Base and (b) An internal structure of Domain Knowledge Base

shown in the Fig. 4(b). It includes three tables named as keyphrase, domain-keyphrase and domain.

B. Training Mode

The objective of training mode is to automatically initialize the domain knowledge base. Since our approach is proposed for domain independent keyphrase recovery, it is necessary to build the large enough domain knowledge base. It is very difficult and expensive to create it by hand. To automatically initialized the domain knowledge base, it uses several documents for training.

The input used in the training mode is the text corpus of the formal definitions of keyphrases gathered from many sources such as on-line glossaries, encyclopedias and also the original documents for recovering. The outputs of this process are list of keyphrases, their domains, and their definitions represented in the form of knowledge representation.

The keyphrase names and domain names ,predefined by user, from the input text are stored in the domain knowledge base for the next uses. For the keyphrases definitions, they are created from the Knowledge Base Generation and then converted into the knowledge representation format. In this paper, the conceptual graph originated by [12] are employed. The conceptual graph-based definitions are then kept in the domain knowledge base.

Within the training mode, the domain knowledge base is automated created without human interference. Therefore, the domain independent knowledge base can be created by using the various domain documents in training mode.

C. Recovering Mode

After the domain knowledge base is set up, we can recover the non-keyphrases that are refused from the conventional keyphrase extraction system by the process in the recovering mode. There are three main parts as shown in box no. (1),(2) and (3) of Fig. 3.

*1) **Knowledge Base Generation**:* This process is used both in training and recovering modes. In training mode, the definition part of document is converted into the knowledge representation format. It accepts the English sentences, tags each sentence for its part-of-speech, parses it by applying the defined syntactic rules and finally interprets them and creates conceptual graphs. In recovering mode, the Keyphrase Determination Process needs more information as surrounding words, phrases or sentences for interpreting the meaning of those candidates The sentences embedded with the non-keyphrases are used as

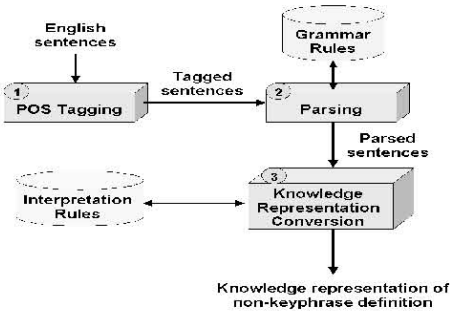


Fig. 5: The Configuration of the Knowledge Base Generation Process

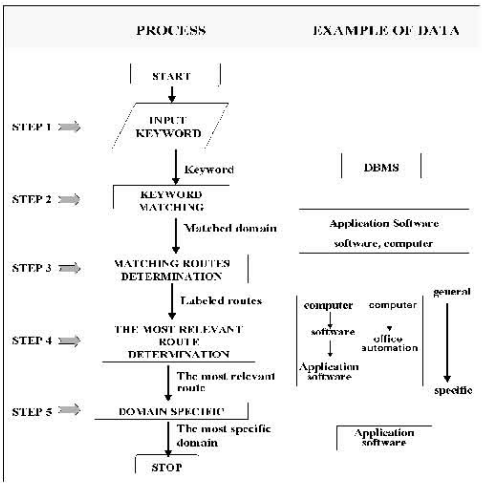


Fig. 6: Domain Identification Process

the context information. These sentences are automatically transformed into conceptual graph-based knowledge representations with the same process as done in the training mode.

*2) **Domain Identification**:* A candidate that is determined as keyphrase in one domain may be refused in the other domains. To recover the non-keyphrases, it is essential to firstly know its domain in order to limit the search spaces. The Domain Identification Process serves for this requirement by receiving the keyphrases produced from the conventional keyphrase extraction system as input. These keyphrases are then used as index to search in the Domain Knowledge Base for the relevant domain names. The Domain Identification Process is shown in Fig. 7.

*3) **Abduction-based Keyphrase Recovery**:* The Abduction-based Keyphrase Recovery is the main part of our proposal for recovering the false rejection keyphrases. Keyphrase Mapping Process firstly verifies the non-keyphrases by employing the Domain Knowledge Base. The unmatched non-keyphrases are then sent to the Keyphrase Determination for abductively recovering.

- **Keyphrase Mapping**
  The non-keyphrases produced by the conventional keyphrase extraction system along with its relevant domains are firstly checked by mapping its appearance with the keyphrase names in the Keyphrase table in the Domain Knowledge Base. After mapping, the matched candidates are output to the user as the correct keyphrases while the unmatched ones are then feeded to the next process, called Keyphrase Determination. Figure 9 shows the details of Keyphrase Mapping process.
- **Keyphrase Determination**
  The Keyphrase Determination uses the semantic matching based on abductive reasoning to rescue the remaining non-keyphrases. The input for this process are list of the remaining non-keyphrases, their relevant domains and the knowledge representation of their context information. The non-keyphrase are previously assumed as valid keyphrases and then evaluated their semantic(meaning) with the existing keyphrases in the Domain Knowledge Base. The ones that the similar scores reach the specified threshold are rescued and then updated in the Domain Knowledge Base for the next cycle of extraction. Figure 11 and Fig. 10 explain the process of this process.

In Keyphrase Mapping, the non-keyphrase that has only one appearance-match between it and contents in the considered domain is accepted as recovered keyphrase. Unfortunately, the meaning matching of the non-keyphrase and the conceptual graph in the domain knowledge base can not be accepted by only one match. These matched meaning are accumulated and then computed to find the "similarity score". The similarity score derived from (1) is used as indicator to promote the non-keyphrase as recovered keyphrase.

$$sscore = \max_i \{ \frac{(X_i) \times (W_i)}{(N_i)} \} \tag{1}$$

where $sscore$ is the similarity score of each non-keyphrase

(X_i) is the number of matched meaning

(N_i) is the number of all meaning in the considered domain

(W_i) is the weight of the considered domain.

The weight of the considered domain is calculated by (2).

$$W_i = \frac{1}{2^k} \tag{2}$$

where W_i is the weight of each non-keyphrase

k is the number of levels (distance) starting from the specified domain to the considered domain.

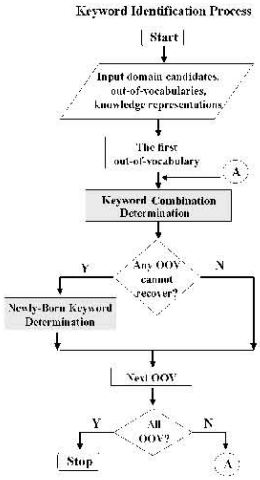


Fig. 7: The Abduction-based Keyphrase Recovery Process

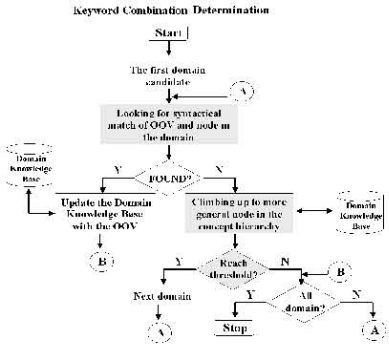


Fig. 8: The Keyphrase Mapping Process

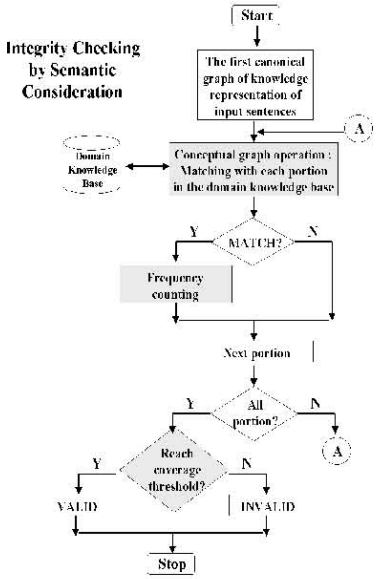


Fig. 9: The Integrity Checking in Keyword Identification Process

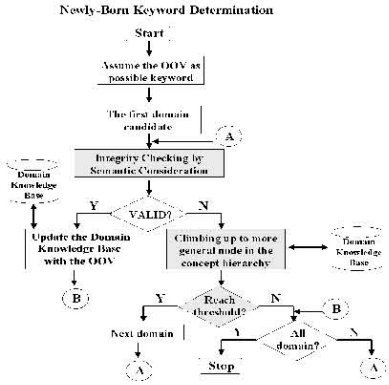


Fig. 10: The Keyphrase Determination Process

TABLE I: Data using in the experiments

Mode	No. of Domains	No. of Glossaries	No of Chapters	No of Articles
Training	10	4	15	60
Recovering	10	-	15	160

The acceptance of the non-keyphrases are determined by the following criteria as shown in (3)

$$Accept(sscore) = \begin{cases} 1 & \text{if } sscore \geq T \\ 0 & \text{if } sscore < T \end{cases} \quad (3)$$

where T is the specified threshold

Those non-keyphrases are accepted either in Keyphrase Mapping process or Keyphrase Determination process are automatically updated in the Domain Knowledge Base.

V. Experiments and Results

As we propose to enhance the keyphrase extraction system by appending the "Semantic-based Keyphrase Recovery Function" as a post-processing, the following steps are used to evaluate our approach.

A. Data Preparation

Since there are two modes in our proposed function, the two groups of data are also used. In training mode, we use three types of training text. The one is glossaries of all related keyphrases in the computer and telecommunication domains. These glossaries are from four locations, three of them from the websites owned by CNET Networks,

TABLE II: Performance comparisons between the conventional systems and our proposed Semantic-Based Keyphrase Recovery Function

Conventional Keyphrase Extraction Systems	Performance of Keyphrase Extraction					
	By Conventional Systems		Improved by Our Proposed Function		Total	
	Precision	Recall	Precision	Recall	Precision	Recall
1. EXTRACTOR	7.49	8.53	33.16	36.10	40.65	44.63
2. KEA	5.94	5.48	41.30	39.17	47.24	44.65

Inc. [13], Tech Target Company [14], and University of Chicago [15]. The remaining one is from the glossary chapter of [16]. The second type of training text is from the summary section of IEICE transactions on Information and Systems. We used 60 summaries from 10 domains. And the last type is all the original documents used for recovering.

In recovering mode, the data that we are used in our experiments are in text format collected from two resources. The first resource is from the summary section of 15 chapters in a computer textbook [16]. Each summary section consists of 6-8 paragraphs with 3-5 sentences for each paragraph. The second resource is from the summary section in 100 articles from the on-line of IEICE transactions. Each article section includes approximately 15-20 sentences. The configuration of data using in the experiments is shown in Table I.

The KEA developed by [1] and EXTRACTOR developed by [5] are baseline used as the representative of statistic-based and machine learning-based keyphrase extraction systems.

Because we propose to improve the performance of the conventional system by rescuing the non-keyphrases, the number of candidates (C) are the number of all non-keyphrases rejected by the baseline systems. While the total number of real keyphrases are counted from the list of keyphrases defined by the authors. By using our testing documents, Table II illustrates that with the additional functions, the performances of the conventional extraction systems can be improved.

VI. Conclusion

This paper proposes the post-processing function to recover the false reject keyphrases of the conventional keyphrase extraction systems. With this proposed function, the false reject keyphrases can be recovered in several domains of interest. The relevant domains are automatically identified by the Domain Identification process. We evaluate our proposed function with three conventional system i.e. EXTRACTOR, KEA and our keyphrase database-mapping based extractor. They are approximately increased 33.16 and 41.30% of precision in EXTRACTOR and KEA, and 36.10 and 39.17% of recall in EXTRACTOR and KEA, respectively.

Acknowledgments

This paper is based upon work supported by the Thailand Research Fund under grant No. RMU4880007 of TRF Research Scholar. The authors also wish to thank Connexor Co. Ltd for the use of the academic license of Machinese Syntax which was used in our experiments.

References

[1] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-specific keyphrase extraction," in *The Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999, pp. 668–673.

[2] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, "Kea:practical automatic keyphrase extraction," in *Proceedings of the Fourth ACM Conference on Digital Libraries*, 1999.

[3] H. Nakagawa, "Automatic term recognition based on statistics of compound nouns," *Terminology*, vol. 6, no. 2, pp. 195–210, 2000.

[4] K. Barker and N. Cornacchia, "Using noun phrase heads to extract document keyphrases," in *Canadian Conference on AI*, 2000, pp. 40–52.

[5] P. D. Turney, "Learning to extract keyphrases from text," ERB 1057, National Research council, Institute for Information Technology, 1999.

[6] P. D. Turney, "Learning algorithms for keyphrase extraction," *Information Retrieval*, vol. 2, no. 4, pp. 303–336, 2000.

[7] D. Maynard and S. Ananiadou, "Acquiring contextual information for term disambiguation," in *Proc. of Computerm '98 Workshop on Computational Terminology (COLING/ACL '98)*, Montreal, Canada, 1998, pp. 86–91.

[8] D. Maynard and S. Ananiadou, "Term extraction using a similarity-based approach," in *Recent Advances in Computational Terminology*, Amsterdam, 1999.

[9] D. Maynard and S. Ananiadou, "Identifying terms by their family and friends," in *Proc. of COLING 2000*, Saarbrucken, Germany, 2000, pp. 530–536.

[10] D. Maynard and S. Ananiadou, "Trucks: a model for automatic term recognition," *Journal of Natural Language Processing*, 2000.

[11] J. Vivaldi, L. Marquez, and H. Rodriguez, "Improving term extraction by system combination using boosting," in *European Conference on Machine Learning*, 2001, pp. 515–526.

[12] John F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA., 1984.

[13] CNET Networks, "http://www.cnet.com/resources/info/glossary," 1995-2002.

[14] Tech Target Company, "http://whatis.techtarget.com," 2002.

[15] UNIVERSITY OF CHICAGO CAMPUS COMPUTER STORES, "http://ccs.uchicago.edu/technotes/misc/glossary," 2002.

[16] Steven C. Lawlor, *Computer Information Systems*, Harcourt Brace Jovanovich, Inc., 2 edition, 1992.