

โปรแกรมค้นหาแบบจัดโครงสร้างผลการค้นหา  
A Hierarchical Network Search Engine

โดย

นาย ไชยวุฒิ อติพันธ์

รหัส 4067012

อาจารย์ที่ปรึกษา

ดร. ภัทรชัย ลลิตโรจน์วงศ์



\*H001966\*

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาคเรียนที่ 2 ปีการศึกษา 2545  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วัน เดือน ปี.....	29 ต.ค. 2545
เลขทะเบียน.....	01966
เลขเรียกหนังสือ.....	ฉท ๖๑13๖ ๕๔๕
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

ชื่อหัวข้อ	โปรแกรมค้นหาแบบจัดโครงสร้างผลการค้นหา
นักศึกษา	นายไชยวุฒิ อติพันธ์
อาจารย์ที่ปรึกษา	ดร. กัทรัชชัช ลลิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

### บทคัดย่อ

โครงการพัฒนาระบบงานฉบับนี้เป็นการพัฒนารูปแบบการแสดงผลการค้นหาของโปรแกรมค้นหาโดยคัดแยกเอกสารที่ได้จากการค้นหาออกเป็นหมวดหมู่ตามความหมายของเอกสาร แล้วนำหมวดหมู่ที่ได้นั้นมาแสดงผลต่อผู้ใช้ในรูปแบบลำดับชั้น ทั้งนี้เพื่อให้ผู้ใช้โปรแกรมค้นหาใช้งานและติดตามผลการค้นหาได้ง่ายขึ้น

<b>Title</b>	A Hierarchical Network Search Engine
<b>Student</b>	Mr. Chaiwut Atipan
<b>Advisor</b>	Dr. Pattarachai Lalijrojwong
<b>Level of Study</b>	Master of Science in Information Technology
<b>Major</b>	Information Science
<b>Academic Year</b>	2002

### ABSTRACT

In this project, a Web search software system has been developed. It categorizes the returned Web documents based on the idea of the ontological categorization. By this approach, the search results are organized in a semantic hierarchy that can assist user to find target information in a progressive and interactive manner.

## กิตติกรรมประกาศ

ผมขอขอบคุณทุกๆท่านที่เกี่ยวข้องกับการพัฒนาโครงการนี้ ขอขอบคุณ ดร. ภัทรชัย สถิตโรจน์วงศ์ อาจารย์ที่ปรึกษาที่ให้คำชี้แนะในการพัฒนาโครงการ ขอขอบคุณคุณแม่ที่เป็นกำลังใจให้การพัฒนาระบบงานนี้สำเร็จลุล่วงไปได้ด้วยดี



## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ .....	III
สารบัญ .....	IV
สารบัญรูป .....	VI
บทที่	
1. บทนำ .....	1
1.1 ปัญหาและความเป็นมา .....	1
1.2 วัตถุประสงค์ในการศึกษา .....	1
1.3 ขอบเขตการศึกษา .....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ .....	2
1.5 เครื่องมือที่ใช้ในการพัฒนาโครงการ .....	2
2. ทฤษฎีที่เกี่ยวข้อง .....	3
2.1 นิยามคำศัพท์ .....	3
2.2 การทำงานของโปรแกรมค้นหาข้อมูลอินเทอร์เน็ต .....	4
2.3 แนวความคิดการพัฒนาประสิทธิภาพของการแสดงผลการค้นหา .....	7
2.4 แนวความคิดที่นำมาใช้ในโครงการ .....	8
3. เครื่องมือที่ใช้ในการพัฒนาโครงการ .....	11
3.1 WordNet .....	11
3.2 .Net Framework .....	13
3.3 Google Web Application Program Interface .....	17
4. การออกแบบและพัฒนาโปรแกรม .....	18
4.1 ขั้นตอนการพัฒนาโปรแกรม .....	18
4.2 แนวคิดและหลักการที่ใช้ .....	18
4.3 หลักการทำงานของระบบ .....	19

4.4 สถาปัตยกรรมของระบบ .....	20
5. สรุปผลที่ได้จากการพัฒนาโครงการ .....	24
5.1 สรุปผลการใช้งาน .....	24
5.2 ข้อจำกัดของโปรแกรม .....	29
5.3 แนวทางการพัฒนาต่อ .....	29
บรรณานุกรม .....	30
ประวัติผู้เขียน .....	32



## สารบัญรูป

รูปที่	หน้า
2.1 สถาปัตยกรรมพื้นฐานของ Web Crawler และ indexer .....	4
2.2 สถาปัตยกรรมระบบของ Hang Cui .....	10
3.1 ตัวอย่างผลที่ได้จากการค้นหาของคำว่า Car ใน WordNet 1.7.1 .....	12
3.2 ตัวอย่าง Hypernym ของคำว่า Car ใน WordNet 1.7.1 .....	12
3.3 กำนามพื้นฐาน 25 คำของ WordNet .....	12
3.4 ส่วนประกอบหลักๆของ .Net Framework .....	12
4.1 องค์ประกอบของระบบ .....	20
4.2 สถาปัตยกรรมของระบบ .....	21
4.3 การลดความลึกให้แก่โครงสร้างผลการค้นหา .....	21
5.1 หน้าจอแสดงผลของโปรแกรม .....	24
5.2 หน้าจอแสดงผลการค้นหาคีย์เวิร์ด “AMD” .....	26
5.3 ตัวอย่างรายละเอียด โครงสร้างผลการค้นหาคีย์เวิร์ด “AMD” .....	27
5.4 ตัวอย่างการเรียกดูเรื่องการจัดการความร้อนของคีย์เวิร์ด “AMD” .....	27
5.5 ตัวอย่างการเรียกดูเรื่องการรับสมัครงานของคีย์เวิร์ด “AMD” .....	28
5.6 การจัดโครงสร้างให้แก่คำว่า “processor” ของคีย์เวิร์ด “AMD” .....	28

# บทที่ 1

## บทนำ

### 1.1 ปัญหาและความเป็นมา

บริการค้นหาข้อมูลในอินเทอร์เน็ตเป็นเครื่องมือที่สำคัญสำหรับอินเทอร์เน็ตในปัจจุบัน แต่ด้วยปริมาณเว็บไซต์ที่เพิ่มจำนวนมากขึ้นอยู่ตลอดเวลา ทำให้การค้นหาข้อมูลที่ต้องการเป็นเรื่องที่ยากลำบากมากขึ้น ซึ่งในปัจจุบันนี้มีโปรแกรมค้นหาข้อมูลอินเทอร์เน็ต (Web Search Engine) ให้ผู้ที่ต้องการค้นหาข้อมูลที่ต้องการอยู่มากพอสมควร ถึงแม้ว่าโปรแกรมค้นหาเหล่านี้จะช่วยผู้ใช้ในการค้นหาข้อมูลได้ก็ตาม แต่รูปแบบการแสดงผลการค้นหาที่ยังไม่ตรงกับความต้องการในการค้นหาของผู้ใช้ยังคงเป็นสิ่งที่ต้องพัฒนาต่อไป

ตัวอย่างปัญหาเรื่องรูปแบบการแสดงผลการค้นหาที่พบในปัจจุบัน

1. ผู้ใช้มือใหม่ต้องใช้ความพยายามไม่น้อยในการติดตามผลการค้นหา เนื่องจากผลการค้นหามักจะมีจำนวนมากจนไม่สามารถติดตามได้ทั้งหมด ซึ่งอาจจะเกิดจากการที่ผู้ใช้ใส่คีย์เวิร์ด (keyword) แบบกว้างๆ ไม่ใส่ให้เฉพาะเจาะจงอันเนื่องมาจากความไม่รู้หรือไม่ชอบพิมพ์ข้อความยาวๆ
2. อัลกอริทึมการจัดลำดับผลการค้นหาที่ยังไม่ตรงใจผู้ใช้หลายคน เนื่องจากผลการค้นหานั้นมีการกระจัดกระจายและผสมปนเปกันอยู่
3. ข้อมูลที่ได้จากการค้นหาจำนวนมากไปไม่ถึงผู้ใช้ เนื่องจากผู้ใช้ส่วนมากไม่สนใจที่จะดูผลการค้นหาหน้าอื่นๆนอกจากหน้าแรกๆ

จากปัญหาที่พบสามารถสรุปได้ว่าเทคโนโลยีที่ใช้ในโปรแกรมค้นหาในปัจจุบันยังไม่สามารถตอบสนองความต้องการของผู้ใช้ได้ถึงขีดสุด การเพิ่มความสามารถในการติดตามผลการค้นหา (browsability) จึงยังเป็นสิ่งที่ต้องการการพัฒนาให้ดียิ่งขึ้นไปอีก

### 1.2 วัตถุประสงค์ในการศึกษา

วัตถุประสงค์ของการพัฒนาโครงการนี้คือ

1. เพื่อเรียนรู้วิธีการสกัดความหมายจากรายละเอียดเล็กๆน้อยๆของเอกสาร
2. เรียนรู้เทคโนโลยี Web-based application และสามารถนำมาใช้ได้

### 3. นำหลักการงานที่ศึกษามาปฏิบัติจริงให้เห็นเป็นรูปเป็นร่าง

#### 1.3 ขอบเขตการศึกษา

1. ศึกษากระบวนการสกัดความหมายของเอกสารจากรายละเอียดที่ปรากฏอยู่ในเอกสารที่ได้จากโปรแกรมค้นหา แล้วนำไปนำเสนอให้อยู่ในรูปแบบลำดับชั้นตามโครงสร้างของฐานข้อมูลคำศัพท์ภาษาอังกฤษออนไลน์ WordNet 1.7.1
2. โปรแกรมค้นหาข้อมูลอินเทอร์เน็ตที่ใช้คือ Google
3. พิจารณาเฉพาะเอกสารเว็บที่เป็นภาษาอังกฤษ
4. พัฒนาโครงการโดยใช้ C# ใน ASP.Net

#### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

เมื่อพัฒนาโครงการนี้แล้ว คาดว่าจะช่วยให้ได้รับประโยชน์ดังนี้

1. รู้จักกับเทคโนโลยีการสกัดความหมายของเอกสารอิเล็กทรอนิกส์มากขึ้น
2. รู้จักโปรแกรมค้นหาข้อมูลในอินเทอร์เน็ตมากขึ้น
3. รู้จักการพัฒนาโปรแกรมประยุกต์บนเว็บมากขึ้น
4. รู้จักการรวบรวมข้อมูลจากอินเทอร์เน็ตมาใช้ในการพัฒนางานที่ต้องการมากขึ้น

#### 1.5 เครื่องมือที่ใช้ในการพัฒนาโครงการ

ซอฟต์แวร์: ระบบปฏิบัติการ Windows XP Professional Edition  
MS .Net Framework Version 1 Service Pack 2  
WordNet 1.7.1  
Google Web API beta2

ฮาร์ดแวร์: CPU AMD Duron 900 MHz  
SDRAM 256 MB PC-133

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

โครงการนี้เป็นการพัฒนารูปแบบการแสดงผลการค้นหาที่ได้จากโปรแกรมค้นหาข้อมูลอินเทอร์เน็ต ทฤษฎีที่ใช้ในการพัฒนาโครงการนี้จึงเป็นทฤษฎีที่เกี่ยวข้องกับโปรแกรมค้นหาข้อมูลอินเทอร์เน็ต ซึ่งประกอบด้วยนิยามคำศัพท์ที่เกี่ยวข้อง การทำงานของโปรแกรมค้นหาข้อมูลอินเทอร์เน็ต และตัวอย่างแนวความคิดการพัฒนาประสิทธิภาพการแสดงผลการค้นหา รายละเอียดของทฤษฎีที่เกี่ยวข้องมีดังนี้

#### 2.1 นิยามคำศัพท์

##### Google Web Application Program Interface (Google Web API)

Google Web API เป็นการเรียกใช้บริการค้นหาข้อมูลด้วย Google ผ่านช่องทางที่ Google กำหนดเอาไว้ ซึ่งจำเป็นต้องลงทะเบียนขอใช้บริการกับทาง Google ก่อน โดยสามารถดูรายละเอียดได้ที่ <http://www.google.com/api> ซึ่งในขณะที่พัฒนาโครงการนี้บริการ Google API ยังคงเป็นรุ่นทดลอง (beta version) อยู่

##### Hypernym

เป็นความสัมพันธ์ภายใน WordNet แบบ X Is-a-kind-of Y ตัวอย่างเช่นคำว่า Car ก็จะมี Hypernym เป็น motor vehicle และ automotive vehicle เป็นต้น

##### Inverted Index

Inverted Index เป็นดัชนีประเภทหนึ่งที่เป็นตัวชี้กลับไปยังตัวดัชนีอีกทีหนึ่ง นั่นคือ Inverted Index จะเป็นดัชนีของ index อีกทีหนึ่ง โดย inverted index จะเก็บข้อมูลสถิติของคำที่ปรากฏอยู่ใน index ซึ่งประกอบด้วยความถี่ในการปรากฏและตำแหน่งที่ตั้งของคำ

##### Snippet

เป็นคำอธิบายโดยย่อของเอกสารที่โปรแกรมค้นหาส่งกลับมา

### Stop-word

Stop-word เป็นคำที่ไม่ค่อยมีความหมายสำหรับโปรแกรมค้นหาข้อมูล ซึ่งจะไม่ถูกนำมาพิจารณาในการสร้างดัชนี เช่นพวก article, conjunction และ preposition เป็นต้น

### Stemming

เป็นการตัด Suffix ของคำภาษาอังกฤษให้เหลือเพียงรากศัพท์เท่านั้น ทั้งนี้ก็เพื่อลดขนาดของ inverted index file และช่วยในการสกัดความหมายเอกสารที่จะทำกันต่อไป

### Synonym

คือคำที่มีความหมายเหมือนกันแต่เขียนต่างกัน เป็นความสัมพันธ์แบบหนึ่งใน WordNet

### WordNet

WordNet เป็นฐานข้อมูลคำศัพท์ภาษาอังกฤษออนไลน์ที่พัฒนาโดยคณะนักวิจัยของมหาวิทยาลัย Princeton ซึ่งมีผู้นิยมใช้กันอย่างแพร่หลาย WordNet แบ่งข้อมูลคำศัพท์ออกเป็นหมวดหมู่ตามความหมายของคำศัพท์ ต่างจากฐานข้อมูลคำศัพท์ทั่วไปที่แบ่งคำศัพท์ออกเป็นหมวดหมู่ตามอักขระ

ในการพัฒนานี้จะใช้ WordNet 1.7.1 ซึ่งเป็นเวอร์ชันล่าสุดที่มีในขณะเริ่มต้นพัฒนาโครงการ

## 2.2 การทำงานของโปรแกรมค้นหาข้อมูลอินเทอร์เน็ต (Cui 2001; Sullivan 2002; Kirsanov 2002)

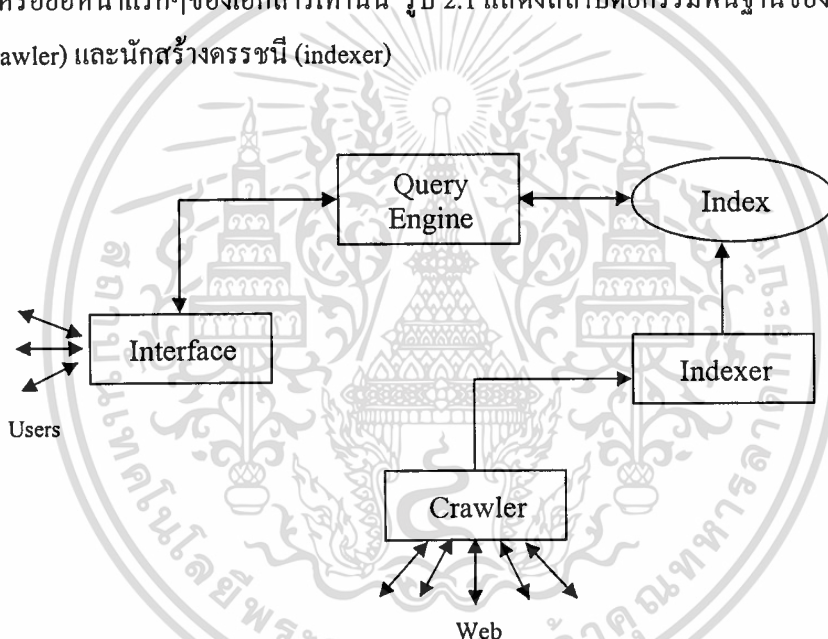
### 2.2.1 โปรแกรมค้นหาข้อมูล

โปรแกรมค้นหาข้อมูลจากอินเทอร์เน็ต(Search Engine) ในปัจจุบันสามารถแบ่งได้เป็น 2 ประเภทคือ Web search engine จริงๆ และ Web directories ทั้ง 2 ประเภทนี้แตกต่างกันที่ Web directories นั้นใช้มนุษย์ในการจัดเอกสารเว็บ โดยจัดแบ่งเอกสารเว็บออกเป็นหมวดหมู่และค้นหาแบบลำดับชั้นจากหัวข้อทั่วไปไปสู่หัวข้อที่เฉพาะเจาะจงมากยิ่งขึ้น ในขณะที่ Web search engine ใช้คอมพิวเตอร์ในการจัดแบ่งเอกสารเว็บแบบอัตโนมัติโดยการสำรวจและเข้าไปเยี่ยมชมเว็บต่างๆ ที่พบ เก็บรวบรวมข้อมูลของแต่ละเว็บ แล้วนำข้อมูลที่พบอย่างเช่น Web address, titles, headers, words, snippets, abstracts, size หรือแม้กระทั่งข้อความทั้งหมดที่ปรากฏมาสร้างดัชนีและตอบสนองต่อคำค้นหาของผู้ใช้โดยส่งผลลัพธ์เป็นเอกสารที่ตรงกับคำค้นหากลับไปยังผู้ใช้โดยจัดลำดับตามลำดับความน่าจะเป็นที่เอกสารนั้นเกี่ยวข้องกับคำค้นหาด้วย

## 2.2.2 การท่องไปตามเว็บต่างๆและสร้างครรรชนี

robot (หรือ spider) เป็นโปรแกรมที่ทำหน้าที่ท่องไปยังเว็บต่างๆ โดยอัตโนมัติ การทำงานจะเริ่มจากสืบค้นเอกสารอันหนึ่งซึ่งมักจะเป็นเอกสารที่โด่งดังและมีลิงค์จำนวนมาก แล้วเลือก URL ในลิงค์มา 1 อันเพื่อเข้าไปสืบค้นทำต่อไปเรื่อยๆจนไปเจอเอกสารที่ไม่มีลิงค์ที่จะไปแล้ว จึงจะถอยกลับไป 1 ชั้นแล้วเลือก URL อื่นเพื่อสำรวจต่อไป

ในการสำรวจเอกสารเว็บแต่ละครั้งนั้น robot จะเก็บรวบรวมข้อมูลของเอกสารเหล่านั้นแล้วแปลเอกสาร HTML ให้อยู่ในรูปแบบที่จะนำมาสร้างครรรชนี จากนั้นก็สร้างครรรชนีและเก็บข้อมูลที่ได้ลงฐานข้อมูลเพื่อการค้นหาอีกทีหนึ่ง ทั้งนี้ robot บางตัวจะแปลเอกสาร HTML ทั้งหมดแล้วนำคำทุกคำที่ปรากฏในเอกสารมาสร้างครรรชนี ในขณะที่ตัวอื่นๆจะทำครรรชนีไว้เฉพาะแท็ก TITLE, META หรือย่อหน้าแรกของเอกสารเท่านั้น รูป 2.1 แสดงสถาปัตยกรรมพื้นฐานของนักท่องเว็บ (Web Crawler) และนักสร้างครรรชนี (indexer)



รูปที่ 2.1 สถาปัตยกรรมพื้นฐานของ Web Crawler และ indexer

การทำงานของ robot ก่อให้เกิดปัญหาอย่างหนึ่งต่อระบบเครือข่ายคือ robot จะเป็นตัวเพิ่มการจราจรภายในเครือข่าย การที่ robot เข้าไปสำรวจเว็บหนึ่งๆอาจส่งผลกระทบต่อความเร็วในการเข้าถึงเว็บของผู้ใช้ปกติ เพื่อที่จะแก้ปัญหานี้ Web robot จึงมีกลไกให้ผู้ดูแลเว็บไซต์สามารถจำกัดการทำงานของ robot ได้ 2 วิธีด้วยกัน วิธีแรกคือกำหนดข้อจำกัดการทำงานไว้ในไฟล์ robots.txt ณ เว็บไซต์นั้นๆ ส่วนอีกวิธีคือกำหนดข้อจำกัดไว้ที่แท็ก META สำหรับ robot ว่าไม่อนุญาตให้นำไปสร้างครรรชนี ดังนี้ <META name="robots" content="noindex">

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.3 การจัดลำดับผลการค้นหา

การจัดลำดับผลการค้นหาถือว่าเป็นส่วนที่สำคัญส่วนหนึ่งของโปรแกรมค้นหาข้อมูลอินเทอร์เน็ตในการเพิ่มคุณภาพของผลการค้นหา การค้นหาข้อมูลอินเทอร์เน็ตในปัจจุบันนี้ผู้ใช้อาจจะได้ผลการค้นหาของการค้นหาครั้งหนึ่งเป็นจำนวนหลายพันรายการด้วยกัน แต่อย่างไรก็ตามผู้ใช้ส่วนมากมักจะสนใจเพียงแค่ผลการค้นหาลำดับต้นๆเท่านั้น นั่นคือการนำเอกสารที่ตรงกับความต้องการของผู้ใช้มาๆอยู่ลำดับต้นๆเป็นสิ่งจำเป็น ทั้งนี้โปรแกรมค้นหาแต่ละตัวมีวิธีการจัดอันดับแตกต่างกันไป แต่โดยพื้นฐานแล้ววิธีการจัดลำดับจะใช้เทคนิค Term Frequency and Location Ranking

Term Frequency and Location Ranking เป็นการจัดอันดับให้เอกสารเว็บโดยประเมินอันดับของเว็บจากตำแหน่งและความถี่ที่คีย์เวิร์ดนั้นปรากฏในเอกสาร ยิ่งพบอยู่ในตำแหน่งต้นๆของเอกสารและมีความถี่ในการปรากฏมากก็ยิ่งจัดไว้เป็นอันดับต้นๆ แต่วิธีนี้มีข้อเสียที่เจ้าของเว็บอาจจะใส่ข้อความบางอย่างซ้ำๆเป็นจำนวนมากไว้ในเอกสารแบบล่องหนเพื่อให้โปรแกรมค้นหานำไปไว้ในอันดับต้นๆ (เช่นกำหนดให้พื้นหลังและข้อความเป็นสีเดียวกัน ซึ่งผู้ท่องเว็บมองไม่เห็นแต่โปรแกรมค้นหาสามารถมองเห็น) แต่อย่างไรก็ตาม โปรแกรมค้นหาตัวหลักๆจะตรวจจับเทคนิคนี้และถ้าหากพบก็จะทำการลดอันดับของเว็บนั้นลงหรือลบออกจากรายการเลย

นอกเหนือจากการใช้ตำแหน่งและความถี่แล้ว โปรแกรมค้นหาแต่ละตัวยังนำปัจจัยอื่นๆมาใช้ในการเพิ่มอันดับของเอกสาร ตัวอย่างเช่น

1. META Tag Boosts Ranking เป็นการนำคำค้นที่ได้จากผู้ใช้ไปเปรียบเทียบกับ Keyword และ Description ที่ปรากฏในแท็ก META ของเอกสารเว็บ แต่มีข้อเสียคือไม่สามารถรับประกันได้ว่า Keyword และ Description ที่ปรากฏในเอกสารเว็บนั้นตรงกับเนื้อหาภายในเอกสารนั้นจริงๆ
2. Reviewed Status Boosts Ranking พิจารณาว่าเอกสารเว็บที่กำลังพิจารณาอยู่นั้นมีอยู่ในเว็บใดเรททอร์หรือไม่ ถ้ามีก็จะเพิ่มอันดับให้
3. Link Popularity Boosts Ranking พิจารณาว่าเอกสารเว็บที่กำลังพิจารณาอยู่นั้นถูกอ้างอิงถึงโดยเว็บอื่นๆมากแค่ไหน ยิ่งถูกอ้างอิงถึงมากก็ยิ่งเพิ่มอันดับให้มากตามไปด้วย
4. Direct Hits Boosts Ranking พิจารณาจากพฤติกรรมของผู้ใช้ว่าผู้ใช้เหล่านั้นเลือกที่จะเข้าเยี่ยมชมผลการค้นหาอันไหน แล้วเก็บรวบรวมข้อมูลไปเรื่อยๆจนได้ข้อมูลมากพอที่จะเอามาปรับอันดับการค้นหาให้เหมาะสมยิ่งขึ้น แต่ก็เกิดปัญหาขึ้นคือเจ้าของเว็บอาจจะเล่นไม่ซื่อโดยทำการค้นหาและเลือกผลการค้นหาที่เป็นเว็บของตัวเองหลายๆครั้งเพื่อเพิ่มอันดับให้แก่เว็บของตนเอง และอีกปัญหาหนึ่งคือผู้ใช้ส่วนมากสนใจแต่ผลการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค้นหาอันดับต้นๆ เพราะฉะนั้นมันจึงไม่ยุติธรรมต่อผลการค้นหาที่ไม่ได้อยู่อันดับต้นๆ เนื่องจากผลการค้นหาอันดับต้นๆย่อมถูกเลือกบ่อยกว่าอันดับที่อยู่ข้างหลังอยู่แล้ว

## 2.3 แนวความคิดการพัฒนาประสิทธิภาพของการแสดงผลการค้นหา

โปรแกรมค้นหาข้อมูลอินเทอร์เน็ตต้องทำงานหนักเพื่อแข่งกับการเจริญเติบโตอย่างรวดเร็วของข้อมูลในอินเทอร์เน็ต โดยการเพิ่มประสิทธิภาพการสร้างสรรค์ดัชนีข้อมูล เพิ่มประสิทธิภาพการค้นหา รวมทั้งการจัดอันดับเอกสารเว็บ ในขณะที่รูปแบบการแสดงผลการค้นหากลับไม่มีการพัฒนาไปมากนัก ยังคงอยู่ในรูปแบบเดิมๆ ที่ใช้กันมาตั้งแต่เกิด โปรแกรมค้นหาขึ้นมา ซึ่งเป็นการแสดงผลการค้นหาตามหลักการที่ว่า ผลการค้นหาทั้งหมดควรที่จะถูกแสดงให้ผู้ใช้เห็นและให้ความอิสระแก่ผู้ใช้ในการเลือกว่าจะเลือกผลการค้นหาอันไหนจากผลการค้นหาทั้งหมดที่มีให้ รูปแบบการแสดงผลการค้นหาแบบดั้งเดิมนี้มีจุดด้อยหลายอย่างที่เป็นที่สนใจสำหรับนักวิจัยหลายๆท่านในการพัฒนารูปแบบของผลการค้นหาให้ดียิ่งขึ้น

จากงานวิจัยเกี่ยวกับการพัฒนารูปแบบของผลการค้นหา (Cui 2001) สามารถแบ่งกลุ่มงานวิจัยได้เป็น 3 กลุ่ม คือ การจัดกลุ่มเอกสารเว็บ (Web Document Clustering), การแยกประเภทเอกสารเว็บ (Web Document Classification) และการแสดงผลการค้นหาเว็บ (Web Search Result Visualization) โดยมีรายละเอียดดังนี้

### 2.3.1 การจัดกลุ่มเอกสารเว็บ

การจัดกลุ่มเอกสารเว็บเป็นการจัดแบ่งเอกสารเว็บออกเป็นหมวดหมู่โดยใช้การประมวลผลข้อความ (text processing) ในการจัดแบ่งเอกสารเว็บแล้วเก็บข้อมูลที่ได้ลงฐานข้อมูลระบบเพื่อให้ผู้ใช้ค้นหาจากฐานข้อมูลนั้นอีกต่อหนึ่ง อย่างเช่นงานวิจัยของ Zamir (1997) ซึ่งเป็นการทำคัลสเตอร์โดยอาศัยการหาคำศัพท์ร่วม 1 คำที่ปรากฏใน Snippet ที่ได้มาจากโปรแกรมค้นหา หรือจะเป็นการทำคัลสเตอร์ตามคำศัพท์พร้อมทั้งลิงค์ที่ปรากฏในเอกสารเว็บ (Weiss 1996; Modha and Spangler 2000) ข้อดีของการทำคัลสเตอร์คือทำให้ผู้ใช้สามารถเลือกดูเฉพาะผลการค้นหาที่มีหัวข้อตรงกับที่สนใจได้ ไม่ต้องเสียเวลาอ่านรายละเอียดผลการค้นหาแต่ละอันว่าอันไหนที่ตรงกับความต้องการ อย่างไรก็ตาม วิธีการนี้มีจุดด้อยคือแต่ละคัลสเตอร์ไม่มีความเกี่ยวข้องกันทางความหมายเลย

### 2.3.2 การแยกประเภทเอกสารเว็บ

การแยกประเภทเอกสารเว็บเป็นการจัดแบ่งเอกสารเว็บออกเป็นหมวดหมู่เช่นเดียวกันกับการทำคลัสเตอร์ แต่หมวดหมู่ที่ใช้เป็นหมวดหมู่ที่กำหนดเอาไว้ก่อนแล้ว ซึ่งหมวดหมู่เหล่านี้สามารถสื่อความหมายได้ดีกว่าการทำคลัสเตอร์ อย่างไรก็ตามวิธีนี้มีจุดด้อยคือจำเป็นต้องใช้ Machine Learning นั่นคือต้องมีการฝึกฝนโปรแกรมให้เรียนรู้งานก่อนที่จะทำงานได้จริงๆซึ่งจะต้องใช้เวลามากพอสมควรและความแม่นยำที่ได้ยังไม่เป็นที่น่าพอใจ ตัวอย่างงานวิจัยที่เกี่ยวข้องคืองานวิจัยของ Koller (1997) และ Klas (2000) เป็นต้น

### 2.3.3 การแสดงผลการค้นหาเว็บ

การแสดงผลการค้นหาเว็บเป็นงานวิจัยที่เน้นการจัดผลการค้นหาที่ได้จากการค้นหาของผู้ใช้ให้อยู่ในรูปแบบที่ผู้ใช้เข้าใจและเรียกดูได้ง่าย ไม่เน้นการประมวลผลเพื่อดึงเมตาดาตาที่อยู่ในเอกสารมาเก็บในฐานข้อมูลอย่าง 2 วิธีแรก ตัวอย่างงานวิจัยที่น่าสนใจได้แก่ xFIND (Andrews 2001)

## 2.4 แนวความคิดที่นำมาใช้ในโครงการ

โครงการนี้นำวิธีการพัฒนาประสิทธิภาพผลการค้นหาจากงานวิจัยของ Hang Cui (2001) ซึ่งเป็นการจัดโครงสร้างผลการค้นหาที่ได้มาจากโปรแกรมค้นหาให้อยู่ในรูปแบบลำดับชั้นตามความหมายที่สกัดได้จากเอกสาร แล้วแสดงผลการค้นหาที่จัดโครงสร้างแล้วนั้นให้แก่ผู้ใช้ ทั้งนี้มีวัตถุประสงค์เพื่อให้ผู้ใช้สามารถเข้าใจและเรียกดูผลการค้นหาได้ดียิ่งขึ้น โดยศึกษาเฉพาะสถาปัตยกรรมที่ Hang Cui ออกแบบไว้แล้วนำมาปรับเปลี่ยนตามที่เหมาะสม ส่วนการ implement นั้นใช้วิธีทำตามความถนัดของผู้พัฒนา

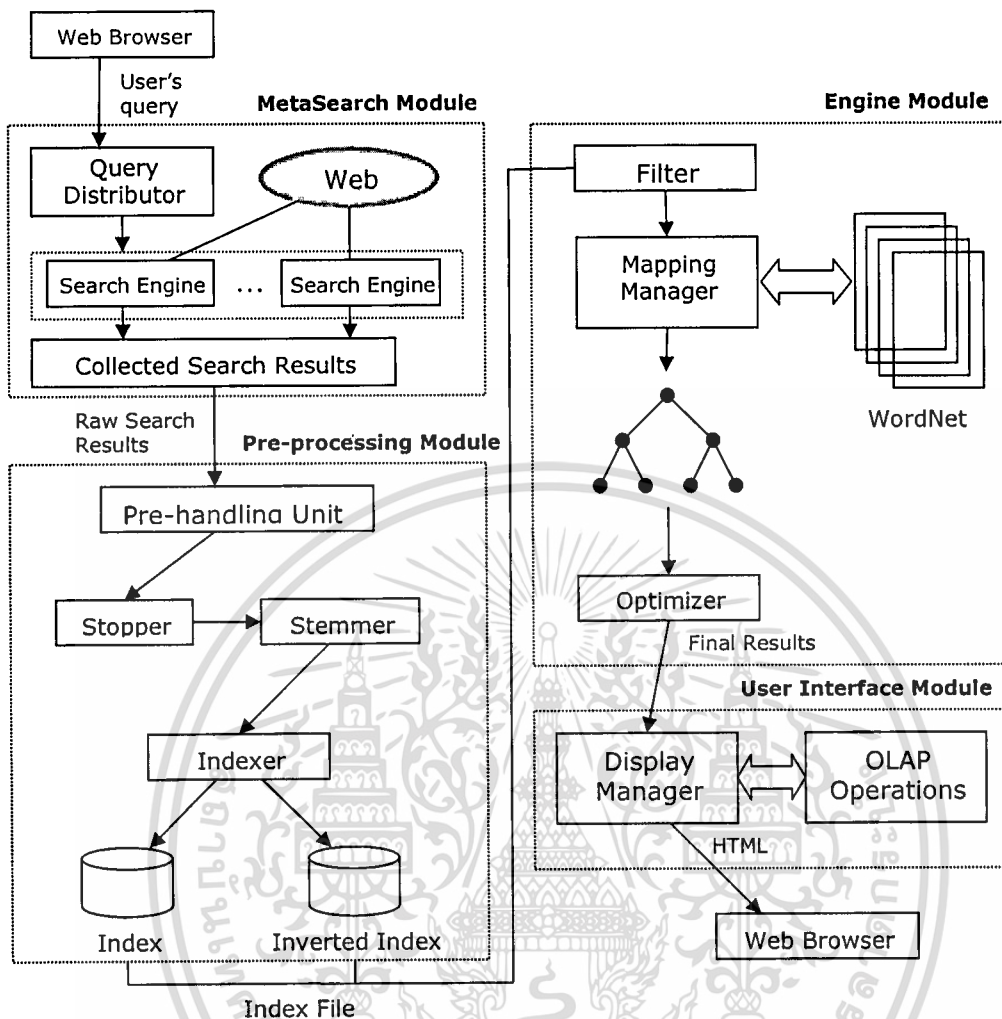
แนวความคิดนี้เป็นการจัดกลุ่มเอกสารตามความหมายที่สกัดได้จากเอกสารต่างจากการจัดกลุ่มเอกสารเว็บในหัวข้อ 2.3.1 ที่ใช้การประมวลผลข้อความซึ่งไม่มีความเกี่ยวข้องกันทางความหมาย ในขณะที่แนวความคิดนี้จะใช้ความหมายเป็นตัวแบ่งกลุ่มเอกสาร และแนวความคิดนี้ใช้โครงสร้างแบบลำดับชั้นของคำใน WordNet จึงมีความคล้ายคลึงกับการแยกประเภทเอกสารเว็บในหัวข้อ 2.3.2 แตกต่างกันที่แนวความคิดนี้ไม่ใช่ Machine learning จึงไม่ต้องเสียเวลาการฝึกฝนโปรแกรม

นอกจากนี้แล้วแนวความคิดนี้ได้นำการจัดการแสดงผลการค้นหาบนเว็บมาใช้ด้วย นั่นคือมีการจัดโครงสร้างผลการค้นหาให้อยู่ในรูปแบบต้นไม้ออกแบบโดยมีเอกสารเว็บอยู่ที่ leaf ส่วนโหนดอื่นๆจะแสดงความหมายโดยรวมของโหนดลูกของมัน ดังนั้นเส้นทางจาก root ไปยัง leaf จึง

เป็นเส้นทางที่ให้ผู้เริ่มเดินจากแนวความคิดกว้างๆ ไปสู่ความเฉพาะเจาะจงที่ผู้ใช้คาดหวังมากขึ้น ซึ่งคาดว่าผู้ใช้สามารถค้นดูผลการค้นหาที่ต้องการได้ง่ายและรวดเร็วยิ่งขึ้น

Hang Cui ได้ออกแบบระบบโดยแบ่งระบบออกเป็น 4 โมดูลด้วยกันคือ MetaSearch Module, Pre-processing Module, Engine Module และ User Interface Module ดังแสดงในรูป 2.2 โดยสรุปแล้ว Metasearch Module มีหน้าที่รับคำค้นหาจากผู้ใช้เพื่อส่งไปยังโปรแกรมค้นหาหลายตัวที่จัดเตรียมไว้ก่อนแล้ว แล้วรับผลการค้นหาซึ่งประกอบด้วย Title, Snippet และ URL ส่งต่อไปให้ Pre-processing Module ประมวลผลตัดผลการค้นหาที่ซ้ำซ้อนออก จากนั้นก็จัดการตัดแต่งข้อมูลใน Title และ Snippet เพื่อนำไปสร้างบรรณานุกรม แล้วนำบรรณานุกรมที่ได้ส่งต่อไปให้ Engine Module กลับกรองบรรณานุกรมพิจารณาเฉพาะคำนามที่ปรากฏบ่อยที่สุดในแต่ละผลการค้นหา นำคำนามนั้นมาหาโครงสร้างใน WordNet แล้วนำไปสร้างเป็นโครงสร้างต้นไม้ สุดท้ายก็ส่งโครงสร้างที่ได้ให้ User Interface Module สร้างภาพโครงสร้างผลการค้นหาแก่ผู้ใช้





รูปที่ 2.2 สถาปัตยกรรมระบบของ Hang Cui

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### เครื่องมือที่ใช้ในการพัฒนาโครงการ

เครื่องมือที่ใช้ในการพัฒนาโครงการนี้มี 3 อย่างด้วยกันคือ ฐานข้อมูลคำศัพท์ภาษาอังกฤษออนไลน์ WordNet เครื่องมือพัฒนาโปรแกรม .Net Framework และ Google Web API

#### 3.1 WordNet

WordNet (Miller 1995) เป็นฐานข้อมูลคำศัพท์ภาษาอังกฤษออนไลน์ที่พัฒนาโดยคณะนักวิจัยของมหาวิทยาลัย Princeton เพื่อให้คอมพิวเตอร์อ่าน ต่างจากพจนานุกรมทั่วไปที่ออกแบบมาให้มนุษย์อ่าน ทั้งนี้ข้อแตกต่างที่ชัดเจนระหว่าง WordNet กับพจนานุกรมทั่วไปคือ พจนานุกรมทั่วไปจะแบ่งคำศัพท์ตามตัวอักษร แต่ WordNet แบ่งคำศัพท์ภาษาอังกฤษออกเป็น 4 หมวดหมู่คือ คำนาม กริยา คำคุณศัพท์ และกริยาวิเศษณ์ แต่ละหมวดหมู่ก็จะแบ่งคำตาม Synonym อีกทีหนึ่ง ซึ่งคุณสมบัติที่สำคัญของ WordNet คือ WordNet จัดเก็บคำศัพท์ตามลักษณะความหมายของคำแทนที่จะเป็นรูปของคำศัพท์ เนื่องจากจุดมุ่งหมายในการพัฒนา WordNet ขึ้นมาคือเพื่อใช้ในการทำ computational linguistic การวิเคราะห์ข้อความ และด้านอื่นๆที่เกี่ยวข้อง นั่นเอง

หน่วยที่เล็กที่สุดของคำใน WordNet คือคู่ของ (Word, Sense) โดย Word คือรูปแบบพื้นฐานของคำศัพท์ และ Sense คือหมายเลขของความหมายใน WordNet ทั้งนี้ก็เนื่องจากคำศัพท์คำหนึ่งสามารถมีความหมายได้หลายความหมายดังตัวอย่างในรูปที่ 3.1 ซึ่งเป็นรูปแสดงให้เห็นว่าคำว่า Car ใน WordNet มีความหมายแบ่งได้เป็น 5 ความหมายด้วยกัน (5 senses) การระบุคำใน WordNet จึงใช้เป็นคู่ของ (Word, Sense) ซึ่งเรียกว่า Synonym set หรือ Synset

ความสัมพันธ์ใน WordNet มีอยู่หลายชนิดด้วยกัน แต่ที่นำมาใช้ในการพัฒนาโครงการนี้มี 2 อย่างคือ Synonym กับ Hypernym ซึ่ง Synonym คือความสัมพันธ์ของคำศัพท์ที่มีความหมายเหมือนกันอย่างน้อย 1 ความหมาย (ใน WordNet จะแบ่ง synonym โดยใช้หมายเลข Sense ซึ่งเป็นตัวแทนของความหมาย) ตัวอย่างเช่น car กับ automobile ต่างก็เป็นยานยนต์เหมือนกัน ดังนั้น car กับ automobile เป็น synonym กัน ส่วนความสัมพันธ์แบบ Hypernym คือความสัมพันธ์ประเภทเป็น "ชนิดหนึ่งของ" (is-a-kind-of) นั่นคือ X จะเป็น hypernym ของ Y ก็ต่อเมื่อ Y เป็นชนิดหนึ่งของ X ตัวอย่างเช่นคำว่า car ใน WordNet 1.7.1 มี hypernym เป็นลำดับชั้นดังรูป 3.2

The noun car has 5 senses (first 3 from tagged texts)

1. car, auto, automobile, machine, motorcar -- (4-wheeled motor vehicle; usually propelled by an internal combustion engine; "he needs a car to get to work")
2. car, railcar, railway car, railroad car -- (a wheeled vehicle adapted to the rails of railroad; "three cars had jumped the rails")
3. cable car, car -- (a conveyance for passengers or freight on a cable railway; "they took a cable car to the top of the mountain")
4. car, gondola -- (car suspended from an airship and carrying personnel and cargo and power plant)
5. car, elevator car -- (where passengers ride up and down; "the car was on the top floor")

รูปที่ 3.1 ตัวอย่างผลที่ได้จากการค้นหาของคำว่า Car ใน WordNet 1.7.1

Car  
 → motor\_vehicle, automotive\_vehicle  
   → wheeled\_vehicle  
     → vehicle  
       → conveyance, transport  
         → instrumentality, instrumentation  
           → artifact, artefact  
             → object, physical\_object  
               → entity, physical\_thing

รูปที่ 3.2 ตัวอย่าง Hypernym ของคำว่า Car ใน WordNet 1.7.1

นอกจากนี้แล้ว WordNet ยังมีความสัมพันธ์ของคำศัพท์ที่สำคัญอีกอย่างคือ Morphological ซึ่งเป็นรูปแบบของคำศัพท์ต่างๆ คำศัพท์ภาษาอังกฤษหนึ่งคำสามารถมีรูปไปได้หลายรูป เช่น รูปเอกพจน์ รูปพหูพจน์ รูปอดีตกาล เป็นต้น WordNet มีกลไกในการแปลงรูปของคำเหล่านี้เป็นรูปแบบพื้นฐานหรือรูปเอกพจน์ปัจจุบันกาลเพื่อให้คำศัพท์ที่รับมาจากผู้ใช้ตรงกับรูปแบบที่เก็บเอาไว้ในฐานข้อมูล

คำศัพท์ใน WordNet แบ่งออกเป็น 4 หมวดหมู่ด้วยกัน แต่หมวดหมู่ที่เกี่ยวข้องในการพัฒนาโครงการนี้มีเพียงหมวดเดียวคือ คำนาม

คำนามใน WordNet เก็บอยู่ในรูปของลำดับชั้นตามความหมายของคำศัพท์ โดยมีคำนามเบื้องต้นซึ่งอยู่ 25 คำด้วยกันดังรูปที่ 3.3 คำนามทั้ง 25 คำนี้ประกอบด้วยลำดับชั้นย่อยๆ จำนวนไม่เกิน 12 ลำดับชั้น ลำดับชั้นแรกจะเป็นลำดับชั้นที่สื่อความหมายของคำศัพท์แบบกว้างๆ และจะมีความเฉพาะเจาะจงมากยิ่งขึ้นไปอีกเมื่อลำดับชั้นยิ่งลึกลงไปเรื่อยๆ

{act, action, activity}	{natural object}
{animal, fauna}	{natural phenomenon}
{artifact}	{person, human being}
{attribute, property}	{plant, flora}
{body, corpus}	{possession}
{cognition, knowledge}	{process}
{communication}	{quantity, amount}
{event, happening}	{relation}
{feeling, emotion}	{shape}
{food}	{state, condition}
{group, collection}	{substance}
{location, place}	{time}
{motive}	

รูปที่ 3.3 คำนามพื้นฐาน 25 คำของ WordNet

## 3.2 .Net Framework (Scnk 2002)

### 3.2.1 .Net Framework คืออะไร?

.Net Framework คือ โครงสร้างการพัฒนาโปรแกรมคอมพิวเตอร์ ที่ถูกออกแบบมาเพื่ออำนวยความสะดวกในการพัฒนาโปรแกรมสมัยใหม่ที่ใช้งานผ่านระบบเครือข่าย (อินเทอร์เน็ต, อินทราเน็ต และอุปกรณ์เคลื่อนที่ เป็นต้น) ทั้งนี้ Bill Gates และ Steve Ballmer ได้บรรยายสรุปวิสัยทัศน์ ที่เกี่ยวกับ .Net เอาไว้ 3 ข้อหลักๆ ได้แก่

1. การพัฒนาโปรแกรมในรูปแบบของเว็บเซอร์วิสจะเป็นหัวใจหลักในการขับเคลื่อนโปรแกรมต่างๆ ที่ใช้งานบนอินเทอร์เน็ต ซึ่งเว็บเซอร์วิสจะช่วยให้การติดต่อสื่อสารระหว่างโปรแกรมประยุกต์บนอินเทอร์เน็ตนั้นง่ายและเป็นระบบมากยิ่งขึ้น

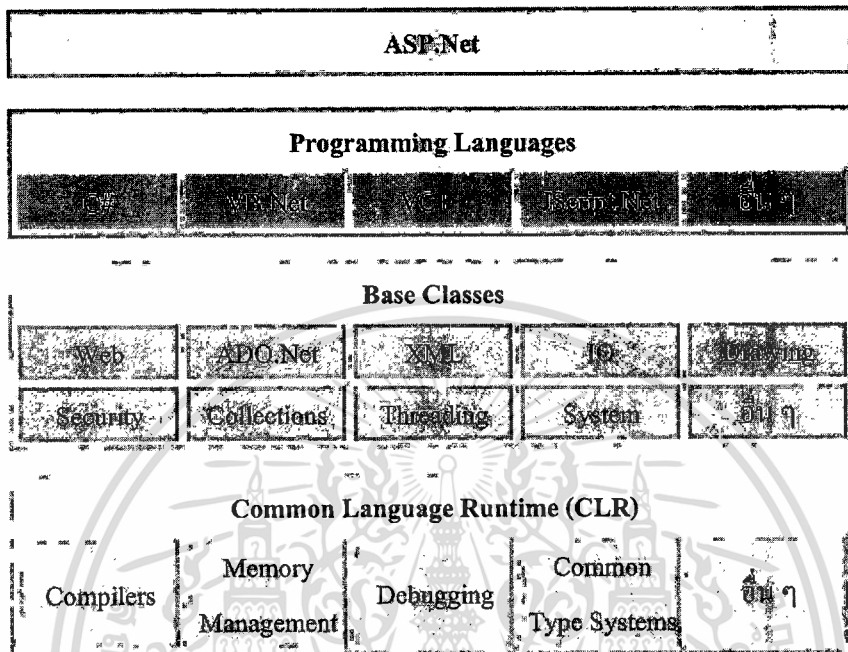
2. เว็บเซอร์วิสขั้นพื้นฐาน เช่นการตรวจสอบผู้ใช้ที่ล็อกอินเข้าสู่ระบบ จะถูกพัฒนาให้เป็นมาตรฐานและสามารถนำไปใช้ได้ทั่วไปบนอินเทอร์เน็ต
3. พีซีทั้งแบบตั้งโต๊ะและโน้ตบุ๊ก รวมทั้งอุปกรณ์เคลื่อนที่ที่ต่อเชื่อมกับอินเทอร์เน็ตได้ เช่น PDA และ โทรศัพท์มือถือ จะมีบทบาทและประโยชน์มากขึ้นไปอีกเมื่อสามารถติดต่อใช้งานโปรแกรมต่างๆ บนอินเทอร์เน็ตได้

ยกตัวอย่างง่าย ๆ จากภาพยนตร์เรื่อง The 6th Day ตอนที่ พระเอก อาร์โนลด์ เฮนัวไปง ประทับไปที่หน้าจอในรถแท็กซี่เว็บเซอร์วิสก็เกิดขึ้นในทันที เริ่มจากระบบตรวจสอบลายนิ้วมือซึ่งให้บริการตรวจสอบว่าลายนิ้วมือของผู้โดยสารนั้นเป็นใคร (User Authentication Web Service) พอทราบแล้วว่าเป็นใคร ระบบก็จะส่งข้อมูลไปยังบริษัทที่ให้บริการรถแท็กซี่ ซึ่งจะทำการคิดคำนวณค่าบริการ และส่งคำร้อง (request) ไปยังธนาคารที่ผู้โดยสารมีบัญชีอยู่เพื่อทำการหักค่าใช้จ่ายจากบัญชีของผู้โดยสาร ไปเข้าบัญชีของบริษัทรถแท็กซี่ ส่วนที่พนักงานขับรถจะได้จากการบริการก็จะถูกบันทึกไว้ในระบบข้อมูลพนักงานคนนั้น ๆ จะเห็นได้ว่าจากสถานการณ์ที่ได้ยกตัวอย่างไปนี่จำเป็นที่จะต้องใช้ข้อมูลและการประมวลผลจากเครื่องคอมพิวเตอร์รวมทั้งโปรแกรมต่าง ๆ อยู่หลายแห่งด้วยกัน ซึ่งแต่ละโปรแกรมนั้นก็อาจจะทำงานอยู่บนระบบที่แตกต่างกันไปเช่น Windows, Linux, Mainframe ฯลฯ ภาษาคอมพิวเตอร์ที่ใช้พัฒนาโปรแกรมเหล่านั้น ก็อาจจะแตกต่างกันออกไป ระบบฐานข้อมูลก็ไม่เหมือนกัน การที่จะทำให้ระบบหลายๆระบบทำงานต่อเชื่อมกันได้อย่างราบรื่นนั้นจึงไม่ใช่เรื่องง่ายเลย

ไมโครซอฟต์จึงได้พัฒนารูปแบบการพัฒนาระบบคอมพิวเตอร์ขึ้นมา ซึ่งเรียกว่า .Net Framework (จริงๆแล้วไมโครซอฟต์ไม่ได้เป็นผู้คิดค้นเรื่องพวกนี้ขึ้นมาแต่เพียงผู้เดียว สิ่งต่างๆ เหล่านี้มีผู้คิดค้นพัฒนาจากหลายบริษัท หลายหน่วยงาน ตัวอย่างเช่น Sun Microsystems, IBM ฯลฯ หากแต่ว่าไมโครซอฟต์นำแนวคิดเหล่านั้นมาออกแบบให้อยู่ในรูปแบบที่สามารถต่อเชื่อมกันได้ง่ายขึ้น เป็นระบบมากขึ้น) เพราะฉะนั้น ถ้าจะให้ให้นิยามคำว่า .Net คงไม่สามารถชี้เฉพาะได้ว่า .Net คืออะไร เพราะจริงๆ แล้ว .Net ประกอบไปด้วยส่วนประกอบต่าง ๆ หลายส่วนด้วยกัน ส่วนประกอบเหล่านี้ถูกออกแบบมาเพื่อให้ทำงานได้เข้าหากันได้ดีขึ้น

### 3.2.2 ส่วนประกอบของ .Net Framework

ส่วนประกอบหลัก ๆ ของ .Net Framework แบ่งเป็นชั้น ๆ ได้ดังรูป 3.4



รูปที่ 3.4 ส่วนประกอบหลักๆของ .Net Framework

1. **Common Language Runtime (CLR)** เป็นส่วนพื้นฐานที่ติดต่อกับระบบปฏิบัติการ Windows ทำหน้าที่เป็น run-time environment ให้กับโปรแกรมที่เขียนขึ้นสำหรับใช้บน .Net ทั้งนี้ CLR มีส่วนของคอมไพเลอร์ทั้งที่เป็นแบบปกติ (คอมไพเลอร์ก่อนที่จะนำโปรแกรมไปใช้) และแบบ Just-In-Time (คอมไพเลอร์เมื่อจะใช้โปรแกรมนั้น ๆ) มีส่วนของ Memory Management ที่เอาไว้สำหรับจัดสรรหน่วยความจำของเครื่องให้กับ โปรแกรม รวมไปถึงการเรียกคืนหน่วยความจำที่ไม่ได้ใช้อีกต่อไป (Garbage Collection) ส่วนของ Common Type Systems (CTS) ทำให้ภาษาต่างๆ ที่เขียนขึ้นบน .Net สามารถทำงานร่วมกันได้ เพราะขนาดและรูปแบบของข้อมูลที่เก็บไว้นั้นเป็นรูปแบบเดียวกัน

2. **Base Class** เป็นคลาสไลบรารีพื้นฐานที่โปรแกรมต่างๆ ไม่ว่าจะเขียนด้วยภาษาใดบน .Net ก็สามารถใช้ร่วมกันได้ เช่น การติดต่อระบบฐานข้อมูล (ADO.Net), การติดต่อกับระบบไฟล์ของเซิร์ฟเวอร์ (IO) เป็นต้น

**3. Programming Language** เป็นเซตของภาษาคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อการเขียนโปรแกรมบน .Net Framework ไม่ใคร่ซอฟต์แวร์นั้นเน้นไปที่ 3-4 ภาษาหลัก ๆ ได้แก่ VB.Net ซึ่งเป็นตัวที่พัฒนาต่อมาจาก VB, C# ซึ่งเป็นภาษาใหม่ที่มีไวยากรณ์ (syntax) ใกล้เคียงกับจาวาและ C++, Visual C++, และ JScript.Net ส่วนภาษาอื่น ๆ นั้นมีบริษัทหรือหน่วยงานอื่น ๆ เป็นผู้พัฒนาขึ้น ซึ่งคาดว่าจะมีเป็นสิบ ๆ ภาษา สำหรับ .Net Framework นั้นไม่ว่าจะเขียนโปรแกรมด้วยภาษาใดก็ตาม ตัวคอมไพเลอร์ใน CLR ก็จะคอมไพล์โปรแกรมนั้นให้อยู่ในรูปของ Intermediate Language (IL) ซึ่งจะถูกนำไปแปลเป็นภาษาเครื่อง (Native Code) อีกทีเมื่อตอนที่นำไปใช้

**4. ASP.Net** เป็นภาษาสคริปต์ที่พัฒนาต่อมาจาก ASP ตัวเก่า เพื่อให้โปรแกรมเมอร์สามารถพัฒนาเว็บแอปพลิเคชันให้ใช้ .Net ได้สะดวกยิ่งขึ้น ASP.Net นี้ ถึงแม้จะอ้างอิงมาจาก ASP ตัวเก่า แต่ก็มีไวยากรณ์หลายส่วนที่เปลี่ยนแปลงไป อย่างไรก็ตาม ผู้ที่เคยเขียน ASP มาก่อนไม่น่าจะมีปัญหาในการอ่านและเขียน ASP.Net มากนัก

สำหรับผู้ที่ยังไม่เคยกับจาวาของชันมาก่อน จะเห็นได้ว่า .Net Framework กับ J2EE ของชันนั้นคล้ายกันมากที่สุด สรุปโดยคร่าว ๆ ได้ดังนี้ CLR = JVM (Java Virtual Machine), IL = Java Bytecode, .Net base classes = Java Class Library, .Net Programming Languages = Java Language, ASP.Net = Java Server Page (JSP) ข้อแตกต่างในด้านสถาปัตยกรรมหลัก ๆ ก็คือ IL ของ .Net นั้นต้องทำงานบน Windows เท่านั้น (ไม่ใคร่ซอฟต์แวร์บอกว่าทำงานบนระบบปฏิบัติการไหนก็ได้ แต่อย่างที่ทราบกันดี คงยากที่จะได้เห็นในอนาคตอันใกล้ หรือไม่ก็คงไม่ดีเท่ากับทำงานบน Windows) แต่ Java Bytecode นั้นสามารถทำงานได้บนทุกปฏิบัติการที่มี JVM ส่วนในด้านประสิทธิภาพการทำงานนั้น ไม่ใคร่ซอฟต์แวร์ได้ทำการเปรียบเทียบโดยพัฒนาโปรแกรมร้านขายสัตว์เลี้ยงด้วย .Net โปรแกรมร้านขายสัตว์เลี้ยงนี้เป็นแอปพลิเคชันอ้างอิงที่พัฒนาบน J2EE ของชัน ไม่ใคร่ซอฟต์แวร์แสดงให้เห็นว่า .Net นั้นทำให้โค้ดสั้นลงหลายเท่าตัว ทำให้โปรแกรมทำงานเร็วขึ้นหลายเท่า เป็นต้น

### 3.2.3 ข้อดีของ .Net Framework

1. ทำให้พัฒนาโปรแกรมได้เร็วขึ้น เนื่องจาก .Net มีคลาสพื้นฐาน (base classes) ส่วนมากไว้ให้ใช้เรียบร้อยแล้ว โปรแกรมที่ถูกพัฒนาขึ้นใหม่สามารถนำไปคอมไพล์เพื่อให้โปรแกรมอื่น ๆ ได้ใช้อีก (reusable)
2. โปรแกรมมีความเชื่อถือได้ (reliable) มากขึ้น เนื่องจากการเขียนโปรแกรมบางรูปแบบบน .Net นั้นไม่สามารถกระทำได้ด้วยภาษาที่มีให้บน .Net ตัวอย่างเช่น การใช้ Pointer ในภาษา

C ที่ไม่มีใน .Net ทำให้ลดโอกาสที่โปรแกรมจะทำอะไรผิดพลาดจนทำให้ระบบไม่สามารถทำงานต่อไปได้

3. ปลอดภัยมากขึ้น เพราะว่า .Net Framework มีความสามารถในการควบคุมการทำงานของโปรแกรมว่า อะไรทำได้ อะไรทำไม่ได้

4. การนำโปรแกรมที่พัฒนาขึ้นไปใช้บนเซิร์ฟเวอร์จริงง่ายขึ้น .Net Framework อนุญาตให้โปรแกรมที่พัฒนาขึ้นนั้นระบุข้อมูลต่าง ๆ ไว้กับโค้ดเลย ทำให้ไม่ต้องนำไปลงทะเบียน (register) เหมือนที่การใช้คอมพิวเตอร์ต้องทำ

### 3.3 Google Web Application Program Interface

Google Web API (Google Web API 2002) เป็นการเรียกใช้บริการค้นหาข้อมูลด้วย Google ผ่านช่องทางที่ Google กำหนดเอาไว้ ซึ่งจำเป็นต้องลงทะเบียนขอใช้บริการกับทาง Google เพื่อที่จะรับ Google Web API license key ซึ่งจำเป็นต้องใช้ในการติดต่อกับ Google ผ่านช่องทางนี้ โดยมีข้อจำกัดในการใช้งานอยู่บ้างแต่ก็ไม่ได้เป็นอุปสรรคใดๆต่อการพัฒนาโครงการนี้ และในขณะที่พัฒนาโครงการนี้บริการ Google Web API ยังคงเป็นรุ่นทดลอง (beta version) อยู่

## บทที่ 4

### การออกแบบและพัฒนาโปรแกรม

#### 4.1 ขั้นตอนการพัฒนาโปรแกรม

ขั้นตอนการพัฒนาโปรแกรมประกอบด้วย

ขั้นที่ 1 ศึกษารวบรวมข้อมูลเกี่ยวกับการทำงานของโปรแกรมค้นหาข้อมูลอินเทอร์เน็ต

ขั้นที่ 2 ศึกษาถึงจุดด้อยของโปรแกรมค้นหาข้อมูลในอินเทอร์เน็ตที่ปรากฏให้เห็นในปัจจุบัน โดยศึกษาจากงานวิจัยที่เกี่ยวข้องกับการพัฒนาประสิทธิภาพโปรแกรมค้นหาข้อมูลที่เผยแพร่กันในอินเทอร์เน็ต

ขั้นที่ 3 เลือกวิธีการพัฒนาประสิทธิภาพโปรแกรมค้นหาข้อมูลอินเทอร์เน็ตจากงานวิจัยที่ได้ศึกษามา 1 วิธี

ขั้นที่ 4 ศึกษารายละเอียดของวิธีการทำงานของหลักการที่เลือกเอาไว้

ขั้นที่ 5 รวบรวมข้อมูลเกี่ยวกับองค์ประกอบที่จำเป็นต้องใช้ในการพัฒนาโปรแกรมค้นหาข้อมูลตามหลักการที่ได้เลือกเอาไว้

ขั้นที่ 6 นำองค์ประกอบที่หาได้จากอินเทอร์เน็ตและพัฒนาขึ้นเองมารวมกัน แล้วดัดแปลงแก้ไขให้เหมาะสมกับงาน

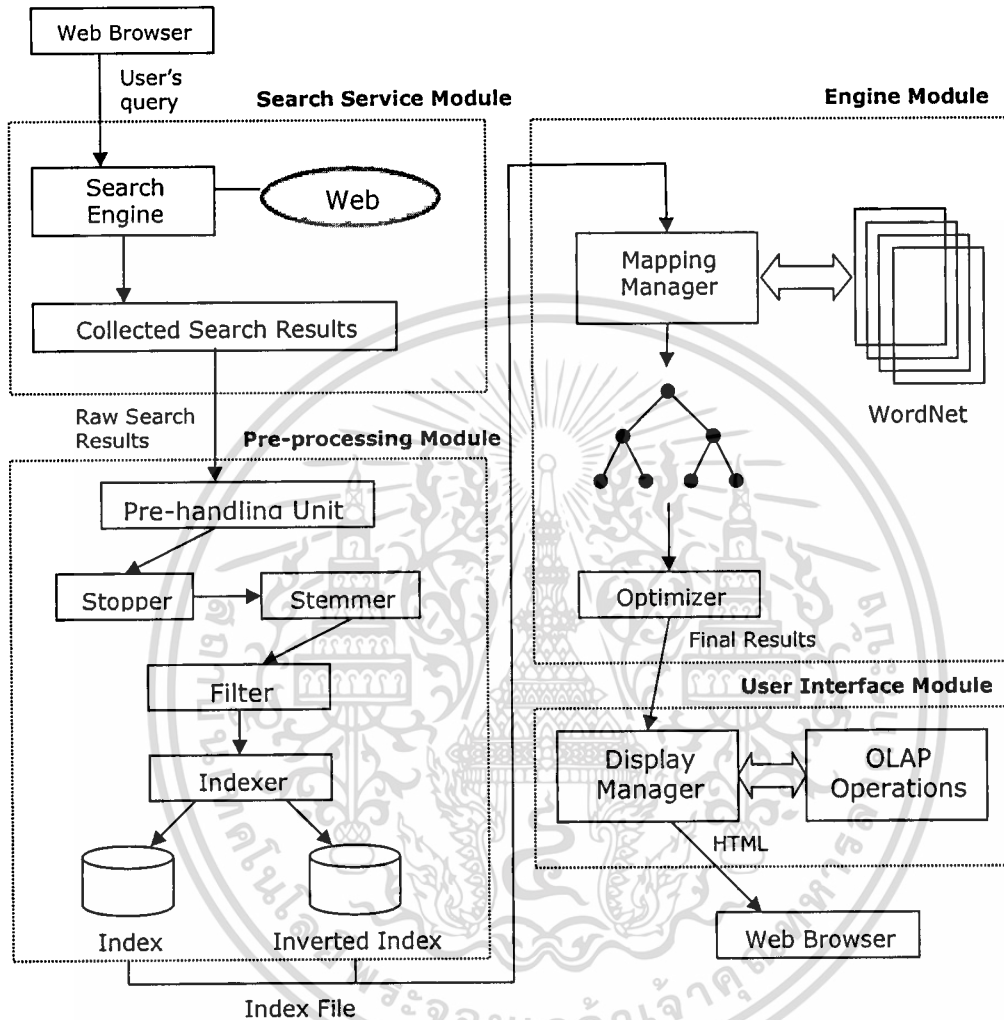
ขั้นที่ 7 ทดสอบและประเมินผลที่ได้จากการทำงานของโปรแกรมค้นหาที่พัฒนาขึ้น เพื่อนำไปใช้ในการแก้ไขจุดบกพร่องต่อไป

#### 4.2 แนวคิดและหลักการที่ใช้

หลักการของการจัดโครงสร้างผลการค้นหาแบบลำดับขั้นนี้มาจากงานวิจัยของ Hang Cui (2001) ซึ่งเป็นการจัดโครงสร้างผลการค้นหาที่ได้มาจากโปรแกรมค้นหาให้อยู่ในรูปแบบลำดับขั้นตามความหมายที่สกัดได้จากเอกสารแล้วแสดงผลการค้นหาที่จัดโครงสร้างแล้วนั้นให้แก่ผู้ใช้ ทั้งนี้มีวัตถุประสงค์เพื่อให้ผู้ใช้สามารถเข้าใจและเรียกดูผลการค้นหาได้ดียิ่งขึ้น โดยได้มีการเปลี่ยนแปลงสถาปัตยกรรมบางส่วนตามที่ผู้พัฒนาเห็นสมควร

เหตุผลที่เลือกพัฒนารูปแบบการแสดงผลการค้นหาก็เนื่องมาจากความคิดเห็นโดยส่วนตัวของผู้พัฒนาโครงการเองที่เห็นว่ารูปแบบการแสดงผลการค้นหาข้อมูลของโปรแกรมค้นหาข้อมูลอินเทอร์เน็ตในปัจจุบันยังไม่เป็นที่น่าพอใจ บ่อยครั้งที่ผู้ใช้โปรแกรมค้นหาข้อมูลจะต้องเสียเวลา

การติดต่อกับโปรแกรมค้นหาข้อมูลนั้นง่ายขึ้น ซึ่งจะช่วยลดเวลาที่ต้องใช้ในการพัฒนาอีกทางหนึ่งด้วย



รูปที่ 4.2 สถาปัตยกรรมของระบบ

#### 4.4.2 Pre-processing Module

Pre-processing Module มีหน้าที่หลักคือรับผลการค้นหาจัดเตรียมให้พร้อมที่จะนำไปจัดโครงสร้าง ซึ่งโมดูลนี้ประกอบด้วยงานย่อยๆดังนี้

1. ลบเครื่องหมายวรรคตอน ตัวเลข และแท็ก HTML จาก Title และ Snippet ของผลการค้นหาออกไปให้หมด

2. ลบ stop-word ทั้งหมดที่ปรากฏใน Title และ Snippet ออก ซึ่ง stop-word คือคำที่มักจะปรากฏในเอกสารอยู่บ่อยๆแต่ไม่ค่อยสื่อความหมาย ถ้าหากปล่อยให้เอาไว้จะส่งผลกระทบต่อการจัดโครงสร้างผลการค้นหาในภายหลังได้
3. นำ Title และ Snippet ที่ลบ stop-word ออกแล้วมาผ่านกระบวนการ stemming โดยให้ WordNet ทำให้ ซึ่งวัตถุประสงค์ของการทำ stemming คือเพื่อเปลี่ยนคำภาษาอังกฤษที่มี suffix ต่างกันแต่มีความหมายเหมือนกันให้กลับไปอยู่ในรูปแบบพื้นฐานให้เหมือนกันทั้งหมด
4. ลบคำศัพท์ภาษาอังกฤษที่ไม่ใช่ค่านามออกไปทั้งหมด โดยส่งคำศัพท์ภาษาอังกฤษแต่ละคำให้ WordNet ตรวจสอบว่ามีอยู่ในฐานข้อมูลหมวดค่านามของ WordNet หรือไม่ ถ้ามีก็จะถือว่าคำนั้นเป็นค่านาม
5. นำ Title, Snippet และ URL ที่ผ่านการตัดแต่งแล้วมาเก็บเอาไว้เป็นแฟ้มดรรชนี
6. นำข้อมูลในแฟ้มดรรชนีมาสร้าง inverted index file เก็บข้อมูลคำศัพท์ภาษาอังกฤษที่ปรากฏบ่อยที่สุดในแต่ละรายการผลการค้นหา
7. ส่ง inverted index file ที่ได้ไปยัง Engine Module

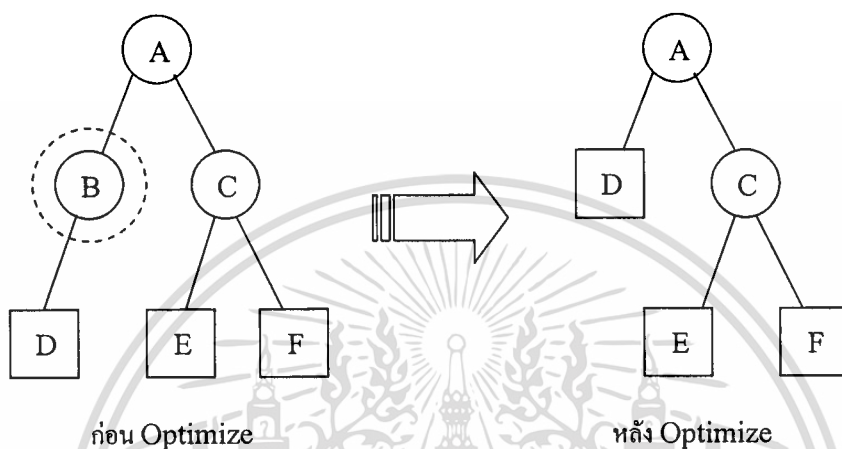
#### 4.4.3 Engine Module

Engine Module มีหน้าที่หลักคือการสร้างโครงสร้างแบบลำดับชั้นตาม WordNet ให้แก่ผลการค้นหาและปรับปรุงโครงสร้างที่ได้นั้นให้เหมาะสม ซึ่งมีขั้นตอนการทำงานเป็นดังนี้

1. เริ่มจากรับ inverted index file จาก Pre-processing Module แล้วอ่านไฟล์เพื่อดูผลการค้นหาแต่ละรายการนั้นปรากฏค่านามตัวใดบ่อยที่สุด
2. Mapping Manager นำค่านามที่ปรากฏบ่อยที่สุดสำหรับแต่ละรายการนั้นส่งไปยัง WordNet หา Hypernym ของคำแต่ละคำ
3. นำ Hypernym ที่ได้ไปสร้างเป็นเส้นทางในโครงสร้างแบบลำดับชั้น โดยให้ค่านามของ Hypernym นั้นอยู่ที่ leaf node
4. ทำซ้ำจนครบทุกรายการผลค้นหา ก็จะได้ผลการค้นหาที่จัดโครงสร้างใหม่แล้ว
5. Optimizer นำผลการค้นหาที่จัดโครงสร้างเสร็จแล้วมาตกแต่ง ลดความลึกของผลการค้นหาโดยการยุบโหนดที่มีโหนดลูกเพียงโหนดเดียวทิ้ง ดังตัวอย่างในรูปที่ 4.3

#### 4.4.4 User Interface Module

User Interface Module มีหน้าที่นำเสนอผลการค้นหาที่จัดโครงสร้างเป็นแบบลำดับชั้นให้อยู่ในรูปของเอกสาร HTML เพื่อที่จะแสดงผลการค้นหาให้ผู้ใช้งานได้ อินเทอร์เน็ตที่แสดงให้ผู้ใช้เห็นจะมีลักษณะเป็นแผนผังโฟลเดอร์ (folder tree) คล้ายๆกับการแสดงเพิ่มข้อมูลใน Windows Explorer ของระบบปฏิบัติการ Microsoft Windows



รูปที่ 4.3 การลดความลึกให้แก่โครงสร้างผลการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลที่ได้จากการพัฒนาโปรแกรม

จากการพัฒนาโปรแกรมตามที่ได้ออกแบบไว้ สามารถสรุปผลได้ดังนี้

#### 5.1 สรุปผลการใช้งาน

หน้าตาของ โปรแกรมเมื่อใช้ Internet Explorer บน Windows XP เป็นดังรูปที่ 5.1



รูปที่ 5.1 หน้าจอแสดงผลของโปรแกรม

เมื่อใส่คีย์เวิร์ดที่ต้องการค้นหาว่า “AMD” โปรแกรมก็จะประมวลผลค้นหาข้อมูลซึ่งถ้าหากมีข้อมูลในฐานข้อมูลอยู่แล้วเนื่องจากเคยค้นหาโดยใช้คีย์เวิร์คนี้มาแล้ว โปรแกรมก็จะอ่านข้อมูลที่เก็บไว้แล้วมาแสดงให้ผู้ใช้งานเห็นผ่านเว็บเบราว์เซอร์โดยใช้เวลาประมวลผลเล็กน้อย แต่ถ้าหากไม่มีข้อมูลของคำค้นหานั้นอยู่ในฐานข้อมูล โปรแกรมจะเชื่อมต่อกับ Google เพื่อค้นหาข้อมูล แล้วเก็บไว้ในฐานข้อมูล จากนั้นก็นำข้อมูลมาประมวลผลจัดโครงสร้างให้แก่ผลการค้นหา ซึ่งจำเป็นต้องใช้เวลาพอสมควร โดยในการทดสอบนั้นต้องใช้เวลาประมาณ 10 วินาทีเมื่อใช้เครื่องคอมพิวเตอร์ของผู้พัฒนาเอง ผลที่ได้จากการโปรแกรมเป็นดังรูปที่ 5.2

main - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail News RSS Local intranet

Address http://localhost/Project/main.aspx Go

## Hierarchical Organization Search Result based on Ontology Approach (Using WordNet 1.7.1)

Enter search keyword:   <== Click Here, Don't use Enter...

Spell Suggestion  Show Snippet

Result : Found in cache 100 record(s).  
Filtered: 84 result(s)

Results

- [-] abstraction
- [-] entity, physical\_thing
- [-] group, grouping
- [-] act, human\_action, human\_activity
- [-] phenomenon
- [-] state
- [-] cognition, knowledge, noesis
- [-] resource

Result : Found in cache 100 record(s).  
Filtered: 84 result(s)

Results

- [-] abstraction
- [-] entity, physical\_thing
- [-] group, grouping
- [-] act, human\_action, human\_activity
- [-] phenomenon
- [-] state
- [-] cognition, knowledge, noesis
- [-] resource

Done Local intranet

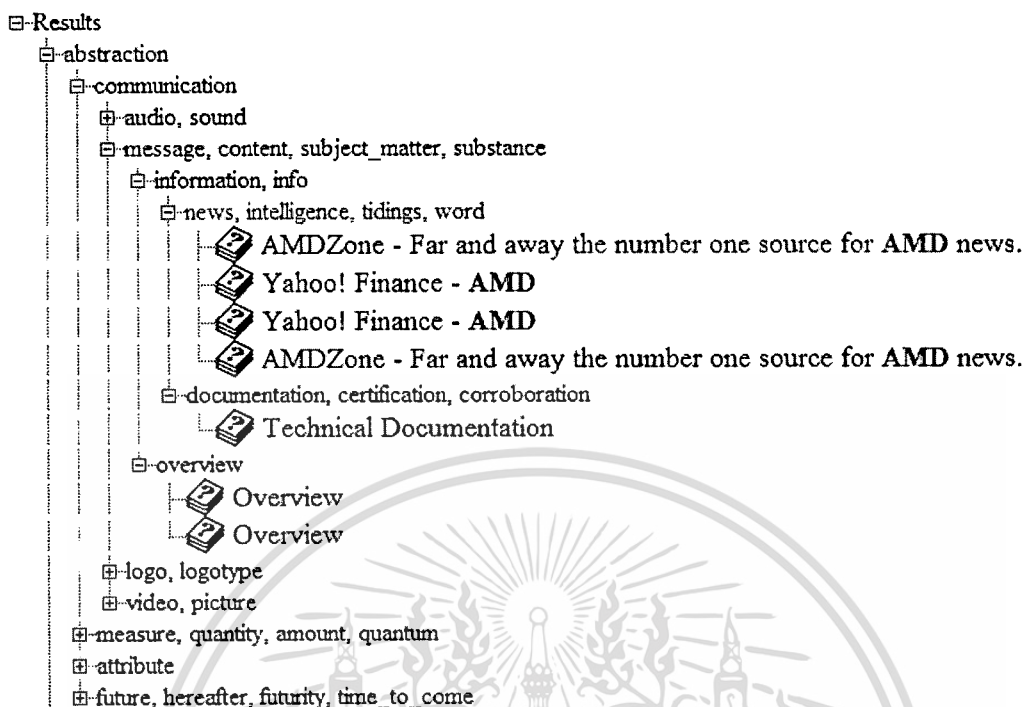
รูปที่ 5.2 หน้าจอแสดงผลการค้นหาด้วยเวิร์ด "AMD"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

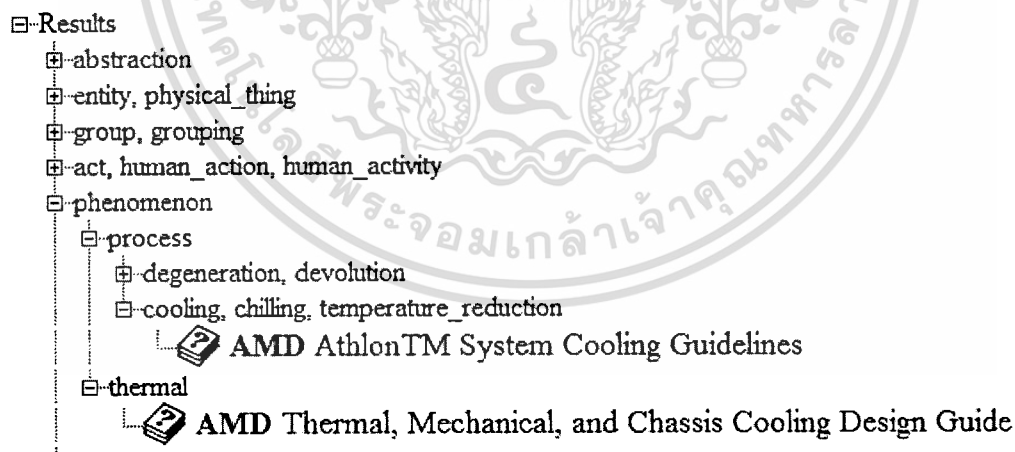
จากผลการค้นหาที่ได้ผู้ใช้สามารถเลือกดูโหนดที่สนใจได้โดยการใช้เมาส์คลิกที่โหนดที่ต้องการ แล้วโหนดจะถูกแสดงออกมา ดังรูปที่ 5.3 ซึ่งจากรูปจะเห็นว่าผลการค้นหาได้ถูกจัดกลุ่มเอาไว้ซึ่งสามารถช่วยให้ผู้ใช้เรียกดูผลการค้นหาทั้งหมดและเลือกเฉพาะที่ต้องการได้ง่ายกว่า การแสดงผลการค้นหาทั่วไป ตัวอย่างเช่นผู้ใช้ต้องการดูการจัดการเรื่องความร้อนก็สามารถเรียกดูได้ที่ Results => phenomenon => process => cooling, chilling, temperature\_reduction หรือที่ Results => phenomenon => thermal ดังรูปที่ 5.4 หรือถ้าหากต้องการดูเรื่องการรับสมัครงานก็สามารถดูได้ที่ Results => act, human\_action, human\_activity => activity => occupation, business, job, line\_of\_work, line ดังรูปที่ 5.5 เมื่อเปรียบเทียบกับผลการค้นหาปกติของ Google เอกสารเว็บเรื่องการจัดการเรื่องความร้อนจะอยู่ลำดับที่ 47 กับ 97 ตามลำดับ ซึ่งผู้ใช้ส่วนมากไม่นิยมดูผลการค้นหาที่ไม่ใช่หน้าแรกๆ ทำให้ผู้ใช้ส่วนมากพลาดเอกสารเว็บนี้ไปในขณะที่โปรแกรมนี้ทำให้ผู้ใช้หาเอกสารที่ต้องการได้โดยเข้าไปตามเส้นทางเกี่ยวกับการจัดการความร้อนก็สามารถค้นหาเอกสารนี้ได้แล้ว

อย่างไรก็ตามโปรแกรมนี้มีจุดด้อยเช่นกัน นั่นคือผลการค้นหาส่วนหนึ่งยังคงถูกจัดกลุ่มอย่างไม่เหมาะสมอยู่อย่างเช่น AMD นั่นเป็นหน่วยประมวลผล (Processor) ซึ่งเป็นสิ่งของ แต่กลับถูกตีความว่าเป็นกลุ่มของบุคคลในองค์กร (processor เหมือนกัน) ดังรูปที่ 5.5 นอกจากนี้ยังมีปัญหาอีกอย่างคือผู้ใช้โปรแกรมจะต้องมีความรู้ทางภาษาอังกฤษอยู่บ้างและเข้าใจโครงสร้างของคำนามใน WordNet จึงสามารถติดตามผลการค้นหาได้อย่างมีประสิทธิภาพ

จากการทดสอบการทำงานของโปรแกรมที่พัฒนาขึ้นนี้ พบว่าโปรแกรมสามารถจัดกลุ่มเอกสารตามความหมายที่ปรากฏในเอกสารได้ดีในระดับหนึ่ง ซึ่งสามารถช่วยให้ผู้ใช้เข้าถึงเอกสารที่ต้องการได้อย่างครอบคลุมและง่ายกว่าโปรแกรมค้นหาทั่วไปที่แสดงผลการค้นหาเป็นรายการยาวๆ ทั้งนี้ก็ขึ้นอยู่กับตัวผู้ใช้เองด้วยซึ่งผู้ใช้จำเป็นต้องมีความรู้ภาษาอังกฤษอยู่บ้าง ควรรู้จักการมองความหมายกว้างๆสู่ความเฉพาะเจาะจงมากขึ้น และควรจะมีสมาธิในโครงสร้างคำนามของ WordNet จึงสามารถที่จะเรียกดูผลการค้นหาที่ต้องการได้อย่างมีประสิทธิภาพ นอกจากนี้แล้วเนื่องจากกลไกการจัดกลุ่มเอกสารเว็บนี้ใช้ข้อมูลเพียงส่วนหนึ่งของเอกสารเท่านั้นจึงทำให้การจัดกลุ่มเอกสารเว็บด้วยวิธีการนี้มีความผิดพลาด จัดกลุ่มไม่ค่อยเหมาะสมอยู่บ้าง ซึ่งนี่ก็เป็นทางเลือกหนึ่งของการจัดการผลการค้นหาให้ผู้ใช้เรียกดูได้สะดวกขึ้น โดยเป็นการจัดโครงสร้างอย่างมีความหมาย ใช้พลังการประมวลผลน้อยกว่า ใช้เวลาน้อยกว่าและไม่ยุ่งยากเรื่องการฝึกฝน โปรแกรมเมื่อเทียบกับการแยกประเภทเอกสารที่ใช้กลไกการเรียนรู้ แต่ก็ต้องแลกกับความแม่นยำในการแยกเอกสารที่สูญเสียไปในระดับหนึ่ง



รูปที่ 5.3 ตัวอย่างรายละเอียด โครงสร้างผลการค้นหาทีวีเวิร์ด “AMD”



รูปที่ 5.4 ตัวอย่างการเรียกดูเรื่องการจัดการความร้อนของทีวีเวิร์ด “AMD”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [-] Results
  - [+] abstraction
  - [+] entity, physical\_thing
  - [+] group, grouping
  - [+] act, human\_action, human\_activity
    - [+] activity
      - [+] support
      - [+] occupation, business, job, line\_of\_work, line
        - [-] Job Opportunities
      - [+] photography, picture\_taking
      - [+] search, hunt, hunting
    - [+] order
    - [+] group\_action

รูปที่ 5.5 ตัวอย่างการเรียกดูเรื่องการรับสมัครงานของคีย์เวิร์ด “AMD”

- [-] Results
  - [+] abstraction
  - [+] entity, physical\_thing
  - [+] group, grouping
    - [+] social\_group
      - [+] organization, organisation
        - [+] processor
          - [-] Advanced Micro Devices, AMD - Processor Homepage
          - [-] AMD Athlon™ XP Desktop Processor
          - [-] Processor Pricing
          - [-] AMD Athlon™ Processor System Power Supplies
          - [-] AMD Athlon™ Processor Recommended Thermal Solutions-heatsinks
          - [-] Processor Roadmap
          - [-] Advanced Micro Devices, AMD - Processor Homepage
          - [-] AMD Duron™ Processors
          - [-] 2000
          - [-] AMD Athlon™ Processor Recommended Motherboards
          - [-] AMD Athlon™ 64 & AMD Opteron™ Processor News
          - [-] Advanced Micro Devices, AMD - Processor Homepage
          - [-] AMD K6-2
          - [-] Read Reviews and Compare Prices on Processors at DealTime.com.
          - [-] AMD Athlon™ Processor System Power Supplies
        - [+] association

รูปที่ 5.6 การจัดโครงสร้างให้แก่คำว่า “processor” ของคีย์เวิร์ด “AMD”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 ข้อจำกัดของโปรแกรม

โปรแกรมที่พัฒนาขึ้นมาใหม่เหมาะสำหรับผู้ใช้ที่ต้องการทราบข้อมูลหลายๆอย่างในสิ่งๆหนึ่งที่ ต้องการโดยใช้คำค้นหาแบบไม่เฉพาะเจาะจง อย่างเช่นต้องการค้นหาข้อมูลเกี่ยวกับประเทศไทย (ใช้คำค้นหาว่า “Thailand”) ว่าในอินเทอร์เน็ตมีข้อมูลเกี่ยวกับประเทศไทยในเรื่องอะไรบ้าง หรือ ต้องการทราบเรื่องราวเกี่ยวกับเทศกาลในประเทศไทย (ใช้คำค้นหาว่า “Thailand Festival”) ว่าประเทศไทยมีเทศกาลประเภทใดบ้าง ช่วงใด ที่ไหน เป็นต้น แต่สำหรับผู้ใช้ที่ต้องการค้นหาข้อมูลแบบ เฉพาะเจาะจงมากๆ อย่างเช่นต้องการทราบว่าโรงหนัง EGV มี URL ว่าอะไร จะไม่เหมาะกับการ ค้นหาโดยใช้โปรแกรมค้นหาอันนี้ การใช้โปรแกรมค้นหาที่แสดงผลการค้นหาแบบปกติจะสามารถให้ผลลัพธ์ที่ผู้ใช้ต้องการได้ง่ายกว่า

## 5.3 แนวทางการพัฒนาต่อ

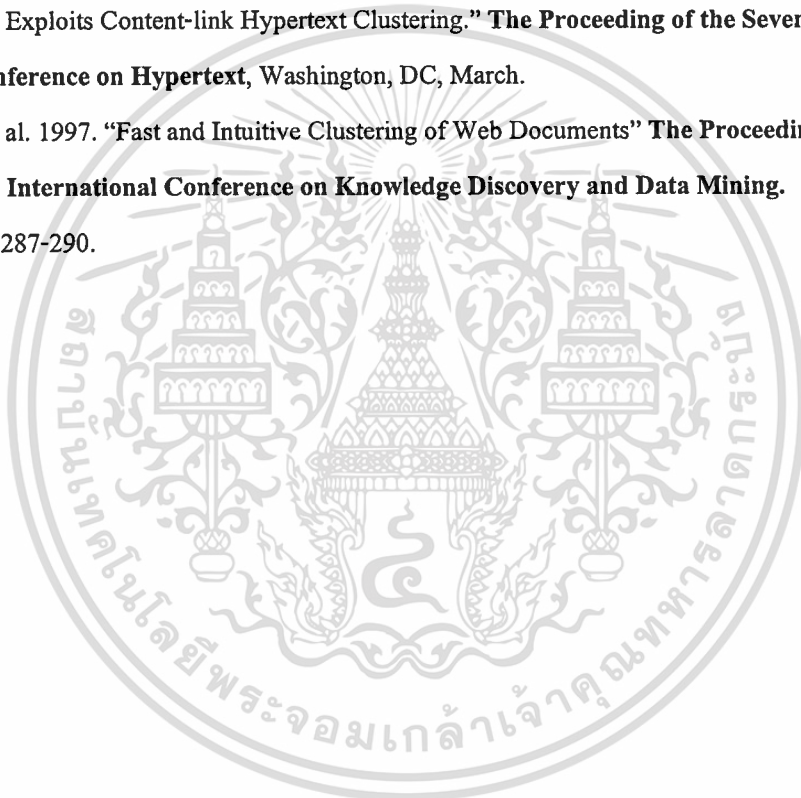
โปรแกรมค้นหาที่พัฒนาขึ้นมาใหม่สามารถพัฒนาต่อไปได้อีกดังนี้

1. เพิ่มส่วนการปรับแต่งรูปแบบโครงสร้างผลการค้นหาตามความต้องการของผู้ใช้แต่ละคน และมีส่วนของการเก็บข้อมูลผู้ใช้ (User Profile)
2. นำ NLP (Natural Language Processing) มาใช้ในการประมวลผลในส่วนของการคัดแยก คำนามออกมาจากข้อมูลผลการค้นหา เพื่อให้แน่ใจได้ว่าคำคำนั้นเป็นคำนามจริงๆ
3. เพิ่ม โปรแกรมค้นหาข้อมูลอินเทอร์เน็ตในส่วนของ Search Service Module เพื่อให้การค้นหา มีความครอบคลุมมากยิ่งขึ้น

## บรรณานุกรม

- Andrews, K. et al. 2001. "Search Result Visualization with xFIND.", **The Proceedings of the 2nd International Workshop on User Interface to Data Intensive Systems**. Zurich, Switzerland, June, pp. 50-58.
- Cui, H. and Zaiane, O.R. 2001. "Hierarchical Structural Approach to Improving the Browsability of Web Search Engine Results." **The Proceedings of the 12th International Workshop on Database and Expert Systems Applications**. Munich, Germany, September 3-7. pp. 956-960.
- Cui, H. 2001. **J-walker, a Metasearch Engine, Ontology Approach**. [Online]. Available: <http://www.cs.ualberta.ca/~cui/>.
- Google. Google Web APIs, 2002. <http://www.google.com/apis>.
- Kirsanov, D. 1997. **HTML Unleashed PRE: Strategies for Indexing and Search Engines**. [Online]. Available: <http://www.webreference.com/dlab/books/html-pre/43-0.html>.
- Klas, C.P. and Fuhr, N. 2000. "A New Effective Approach for Categorizing Web Documents" **22nd Annual Colloquium on Information Retrieval Research**. Cambridge, England.
- Koller, D. and Sahami, M. 1997. "Hierarchically Classifying Document Using Very Few Words." **The Proceedings of the 14th International Conference on Machine Learning**, Nashville, TN, July, pp. 170-178.
- Miller, G. 1990. "Nouns in WordNet: A Lexical Inheritance System", **International Journal of Lexicography**. 3 (4), pp. 245-264.
- Miller, G. 1995. "WordNet: a Lexical Database for English", **Communications of the ACM** 38 (11), November, pp. 39-41.
- Modha, D. and Spangler, W. 2000. "Clustering Hypertext with Applications to Web Searching" **The Proceedings of ACM Hypertext Conference**, San Antonio, Texas, May 30 - June 3.
- Scnk, 2002. **.Net Framework คืออะไร ?** [Online]. เข้าถึงได้ที่: [http://www.thaisharp.net/Show.aspx?art\\_id=9](http://www.thaisharp.net/Show.aspx?art_id=9)

- Search Engine World. 2002. **Keyword Surveillance Tool: Stop Word List.** [Online]. Available:  
<http://www.searchengineworld.com/spy/stopwords.htm>.
- Slesinsky, B. 1997. **Roll Your Own Search Engine.** [Online]. Available:  
<http://hotwired.lycos.com/webmonkey/code/97/16/index2a.html>.
- Sullivan, D. 2002. **How Search Engine Work.** [Online]. Available:  
<http://searchenginewatch.com/webmasters/work.html>.
- Weiss, R. Velez, B. and Sheldon, M. 1996. "Hypersuit: A Hierarchical Network Search Engine that Exploits Content-link Hypertext Clustering." **The Proceeding of the Seventh ACM Conference on Hypertext**, Washington, DC, March.
- Zamir, O. et al. 1997. "Fast and Intuitive Clustering of Web Documents" **The Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining.**  
 pp. 287-290.



## ประวัติผู้เขียน

ชื่อ-สกุล	นาย ไชยวุฒิ อติพันธ์
วันเดือนปีเกิด	4 กุมภาพันธ์ พ.ศ. 2522
ภูมิลำเนา	864/1-2 ถ.จารุเสถียร 2 ต.ปะลฐู อ.สุโหงปาตี จ.นราธิวาส
ประวัติการศึกษา	ระดับประถมศึกษา โรงเรียนบ้านโคกตา ระหว่างปี พ.ศ. 2528-2533 ระดับมัธยมศึกษาตอนต้น โรงเรียนสุโหงโกลก ระหว่างปี พ.ศ. 2534-2536 ระดับมัธยมศึกษาตอนปลาย โรงเรียนคณะราษฎรบำรุงฯ จังหวัดยะลา ระหว่างปี พ.ศ. 2537-2539 ระดับปริญญาตรี มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ระหว่างปี พ.ศ. 2540-2543

