สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การศึกษาทดลองมาตรวัดความเหมือนข้อความแบบผสม

An Experiment Study on Hybrid Text Similarity Metrics

นางสาวประกายแก้ว แสงใหญ่
นางสาวรุจิมาศ เตชุปกรณ์

สารบัญ
เลขทะเบียน 117237
วัน,เดือน,ปี 19 ก.ค. 2554

b 122.1099 ป
i

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AN EXPERIMENT STUDY ON HYBRID TEXT SIMILARITY METRICS

MISS PRAKAYKAEW SAENGYAI

MISS RUJIMAS TAYCHUPAKORN

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

IN COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงานพิเศษ	การศึกษาทดลองมาตรวัดความเหมือนข้อความแบบผสม		
	An Experiment Study on Hybrid Text Similarity		
ชื่อนักศึกษา	นางสาว ประกายแก้ว	แสงใหญ่	50050164
	นางสาว รุจิมาศ	เตชุปกรณ์	50050193
ปริญญา	วิทยาศาสตรบัณฑิต		
สาขาวิชา	วิทยาการคอมพิวเตอร์		
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.วีระ บุญจริง		

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2553

คณะกรรมการสอบ	ลายมือชื่อ
ผู้ช่วยศาสตราจารย์ ดร.ศรัณย์ อินทโกสุม ประธานกรรมการ	
อาจารย์วีระชัย ตันยะสิทธิ์ กรรมการ	
รองศาสตราจารย์ ดร.วีระ บุญจริง กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การศึกษาทดลองมาตรวัดความเหมือนข้อความแบบผสม		
ชื่อนักศึกษา	นางสาว ประกายแก้ว	แสงใหญ่	50050164
	นางสาว รุจิมาศ	เตชุปกรณ์	50050193
ปริญญา	วิทยาศาสตรบัณฑิต		
สาขาวิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2553		
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.วีระ บุญจริง		

บทคัดย่อ

มาตรวัดความเหมือนของข้อความ คือ มาตรวัดที่ให้ค่าที่ใช้แสดงความเหมือนระหว่างข้อความ โดยในปัญหาพิเศษนี้จะศึกษามาตรวัดความเหมือนข้อความแบบผสม ซึ่งเป็นมาตรวัดที่เกิดจากการนำมาตรวัดความเหมือนตามคำใช้ร่วมกับมาตรวัดความเหมือนตามสายอักขระ ซึ่งเดิมมาตรวัดตามคำจะไม่สามารถวัดความเหมือนของข้อความในกรณีที่มีข้อความผิดพลาด ส่วนมาตรวัดแบบสายอักขระถ้ามีข้อความที่สลับตำแหน่งกันก็ไม่สามารถวัดความเหมือนของข้อความได้เช่นกัน จึงศึกษาทดลองใช้มาตรวัดความเหมือนแบบผสมกับชุดข้อมูลต่างๆ ว่าจะวัดความเหมือนของข้อความได้แม่นยำขึ้นกว่ามาตรวัดเดิมหรือไม่ โดยนำมาตรวัดตามสายอักขระต่างๆมาใช้ร่วมกับมาตรวัดตามคำซึ่งในที่นี้เลือก tf-idf มาศึกษาทดลองกับชุดข้อมูลที่มีลักษณะแตกต่างกันเพื่อดูว่ามาตรวัดแบบผสมใดใช้กับชุดข้อมูลนั้นๆแล้วได้ประสิทธิภาพความแม่นยำในการค้นหาข้อมูลได้ดีที่สุด ซึ่งผลการทดลองพบว่า tf-idf ทำงานร่วมกับ ChapmanLengthDeviation แล้วให้ความแม่นยำสูงสุด

คำสำคัญ : มาตรวัดความเหมือนของข้อความ, tf-idf, ความเหมือนข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	An Experiment Study on Hybrid Text Similarity		
Students	Miss Prakaykaew	Saengyai	50050164
	Miss Rujimas	Taychupakorn	50050193
Degree	Bachelor of Science		
Major Program	Computer Science		
Academic Year	2010		
Advisor	Associate Professor Dr.Veera Boonjing		

ABSTRACT

String similarity metrics give similarity values among texts. This special problem is an experimental study on hybrid text similarity metrics. The hybrid metrics combine token-based metrics with character-based metrics. This study employs tf-idf metric as a token-based metric. It combines this metric with different character-based metrics and tests new obtained metrics with different data sets. The results show that tf-idf with ChapmanlengthDeviation gives the best accuracy.

Keywords : String Similarity Metric, tf-idf, String Similarity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่อง การศึกษาทดลองมาตรวัดความเหมือนข้อความแบบผสมสำเร็จลงได้ด้วยความช่วยเหลือและสนับสนุนในทุกๆด้านดังมีบุคคลต่อไปนี้

1. ขอขอบพระคุณบิดา มารดา ญาติพี่น้อง ที่ให้กำลังใจและดูแลคณะผู้จัดทำเสมอมา

2. ขอขอบพระคุณรองศาสตราจารย์ ดร. วีระ บุญจริง และอาจารย์ทุกท่าน ที่คอยสั่งสอนชี้แนะแนวทางในการทำปัญหาพิเศษและความรู้ทางการเรียนมาตลอด 4 ปี

3. ขอขอบพระคุณคุณสุนิสา ตรีอำนรรค ที่ช่วยให้คำปรึกษาและแนะนำข้อมูลในการทำปัญหาพิเศษ

4. ขอบคุณพี่ เพื่อน น้องภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ช่วยให้คำปรึกษาและกำลังใจต่างๆเป็นอย่างดี

นางสาวประกายแก้ว แสงใหญ่

นางสาวรุจิมาศ เตชุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	i
บทคัดย่อภาษาอังกฤษ	ii
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญรูป	vi
สารบัญตาราง	vii
บทที่ 1 บทนำ	**1**
1.1 ที่มาและความสำคัญของปัญหาพิเศษ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปัญหาพิเศษ	2
1.4 ส่วนประกอบของวิทยานิพนธ์	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	**3**
2.1 ทฤษฎีการวัดความเหมือนของข้อความ	3
2.1.1 มาตรวัดความเหมือนตามสายอักขระ	3
2.1.1.1 เลเวนสเทน	3
2.1.1.2 จาโร	4
2.1.1.3 สมิธวอเตอร์แมน	7
2.1.1.4 มองค์อัลคัน	9
2.1.2 มาตรวัดความเหมือนตามคำ	9
2.1.3 มาตรวัดความเหมือนแบบผสม	10
2.1.4 การประเมินผล	11
2.2 งานวิจัยที่เกี่ยวข้อง	12
บทที่ 3 การศึกษาทดลองมาตรวัดความเหมือนข้อความแบบผสม	**13**
3.1 การวัดความเหมือนของข้อความแบบผสม	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.2 ชุดข้อมูล	15
ตารางฐานข้อมูล	15
บทที่ 4 การทดลอง	**28**
4.1 เครื่องมือที่ใช้ในการทดลอง	28
4.1.1 ฮาร์ดแวร์	28
4.1.2 ซอฟต์แวร์	28
4.2 ชุดข้อมูล	28
4.3 การทดลอง	29
4.4 วิธีวัดประสิทธิภาพผลการทดลอง	29
4.5 ผลการทดลอง	31
4.5.1 ผลการทดลองมาตรวัดความเหมือนตามอักขระที่ใช้ร่วมกับมาตรวัดความเหมือนข้อความ tf-idf ที่ชุดข้อมูล Census	31
4.5.2 ผลการทดลองมาตรวัดความเหมือนตามอักขระที่ใช้ร่วมกับมาตรวัดความเหมือนข้อความ tf-idf ที่ชุดข้อมูล Restaurant	32
4.5.3 ผลการทดลองมาตรวัดความเหมือนตามอักขระที่ใช้ร่วมกับมาตรวัดความเหมือนข้อความ tf-idf ที่ชุดข้อมูล Business	33
บทที่ 5 บทสรุปและข้อเสนอแนะ	**34**
5.1 บทสรุป	34
5.2 ข้อเสนอแนะ	34
เอกสารอ้างอิง	**35**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 สมการเลเวนสเทน	3
2.2 ตัวอย่างการคิดสมการเลเวนสเทนที่ตำแหน่ง 1, 2	4
2.3 สมการจาโร	4
2.4 สมการจาโรวิงเคอร์	5
2.5 ตัวอย่างการคิดความเหมือนของข้อความ MARTHA และ MARHTA	5
2.6 ตัวอย่างการคิดความเหมือนของข้อความ JONES และ JOHNSON	6
2.7 สมการสมิธวอเตอร์แมน	7
2.8 ตัวอย่างการคิดสมิธวอเตอร์แมน	8
2.9 สมการมองค์อัลคัน	9
2.10 สมการ tf-idf	9
2.11 สมการ Soft tf-idf	10
2.12 สมการของค่าความแม่นยำ	11
2.13 สมการของค่าเฉลี่ยความแม่นยำ	11
3.1 การวัดค่าความเหมือนของข้อความโดยการให้คะแนนค่าน้ำหนักด้วย tf-idf	13
3.2 ขั้นตอนการวัดความเหมือนข้อความแบบผสม	14
3.3 อัลกอริทึมที่ใช้ในการตัดข้อความเป็นคำ	16
3.4 คำสั่ง SQL ที่ใช้ในการเก็บค่า tf	17
3.5 คำสั่ง SQL ที่ใช้ในการเก็บค่า idf	18
3.6 คำสั่ง SQL ที่ใช้ในการหาค่ารากที่สองของผลรวมค่า tf^2 และ idf^2	19
3.7 คำสั่ง SQL ที่ใช้ในการหาค่าน้ำหนักของคำ	20
3.8 คำสั่ง SQL ที่ใช้ในการเปรียบเทียบข้อความที่เหมือนกันระหว่าง 2 ข้อความ	20
4.1 สมการของค่าเฉลี่ยความแม่นยำ	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 ตารางที่ใช้จัดเก็บข้อความ โดยจะเก็บข้อความไว้ 2 ตาราง ตาราง a,b	15
3.2 ตารางจัดเก็บคำของข้อความ ซึ่งจัดเก็บคำได้ 2 ตาราง token_a , token_b	15
3.3 ตารางที่ใช้จัดเก็บค่า tf คือ ตาราง tf_a,tf_b	17
3.4 ตารางที่ใช้จัดเก็บค่า idf คือตาราง idf_a, idf_b	18
3.5 ตารางที่ใช้เก็บค่าตัวหาร คือ ตาราง len_a	18
3.5 ตารางที่ใช้เก็บค่าตัวหาร คือ ตาราง len_b	19
3.6 ตารางที่ใช้เก็บค่าน้ำหนักของคำ คือ ตาราง weight_a, weight _b	19
3.7 ตารางที่ใช้เปรียบเทียบข้อความที่เหมือนกันของค่า tf-idf	20
3.8 ตารางตัวอย่างที่ใช้เก็บข้อความ	21
3.9 ตารางตัวอย่างที่ใช้เก็บคำ	22
3.10 ตารางตัวอย่างที่ใช้เก็บค่า tf	23
3.11 ตารางตัวอย่างที่ใช้เก็บค่า idf	24
3.12 ตารางตัวอย่างที่ใช้เก็บค่าที่เป็นส่วนหารในการคิดค่าความเหมือนของข้อความ	25
3.13 ตารางตัวอย่างที่ใช้เก็บค่าน้ำหนัก	26
3.14 ตารางตัวอย่างที่ใช้เก็บค่าน้ำหนักความเหมือนของคำ	27
4.1 ชุดข้อมูลที่ใช้ในทดลอง	29
4.2 ผลการทดลองเปรียบเทียบค่าเฉลี่ยความแม่นยำที่ค่าความเหมือน 90 % ในชุดข้อมูล Census	31
4.3 ผลการทดลองเปรียบเทียบค่าเฉลี่ยความแม่นยำที่ค่าความเหมือน 90 % ในชุดข้อมูล Restaurant	32
4.4 ผลการทดลองเปรียบเทียบค่าเฉลี่ยความแม่นยำที่ค่าความเหมือน 90 % ในชุดข้อมูล Business	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหาพิเศษ

ปัจจุบันการเก็บข้อมูลลงในฐานข้อมูลมีความสำคัญมากขึ้น การค้นหาข้อมูลที่เป็นข้อความในฐานข้อมูลจึงเป็นสิ่งจำเป็นในการค้นหาข้อมูลเพื่อให้ได้ผลลัพธ์ที่ต้องการ บางครั้งอาจเกิดการพิมพ์ผิดพลาดหรือสลับตำแหน่งของข้อความที่ต้องการค้นหา ทำให้ได้รับผลลัพธ์ที่ไม่ตรงกับข้อมูลที่ต้องการและเป็นการสูญเสียเวลาในการค้นหา จึงมีการคิดค้นมาตรวัดความเหมือนของข้อความต่างๆเข้ามาช่วยในการค้นหาข้อมูลให้ใกล้เคียงหรือตรงกับข้อความที่ต้องการหามากที่สุด

มาตรวัดความเหมือนของข้อความ (String Similarity Metric) เป็นมาตรวัดที่ช่วยในการวัดหาค่าความเหมือนของข้อความ โดยที่งานวิจัยต่างๆที่ผ่านมาได้นำเสนอมาตรวัดแบบต่างๆ ทั้งแบบมาตรวัดตามคำและมาตรวัดตามสายอักขระ ซึ่งหากใช้มาตรวัดแบบใดแบบหนึ่งอาจยังค้นหาข้อมูลไม่ตรงกับที่ต้องการ เนื่องจากมาตรวัดตามคำไม่เหมาะกับข้อความที่พิมพ์ผิด ส่วนมาตรวัดความเหมือนตามอักขระไม่เหมาะกับข้อความที่มีการสลับตำแหน่งเช่นกัน

ในปัญหาพิเศษนี้จึงต้องการศึกษาว่า หากนำมาตรวัดทั้งสองแบบที่เรียกว่า มาตรวัดแบบผสม มาใช้ร่วมกัน จะให้ประสิทธิภาพดีกว่าหรือไม่อย่างไร โดยในปัญหาพิเศษนี้มีการใช้มาตรวัดความเหมือนตามคำแบบ tf-idf ซึ่งเป็นมาตรวัดที่สามารถวัดระหว่างสองข้อความได้ แม้จะมีการจัดเรียงตำแหน่งของคำแตกต่างกัน แต่ไม่สามารถวัดสองข้อความได้ในกรณีที่มีการพิมพ์ผิดของคำในข้อความนั้นได้ จึงต้องการหามาตรวัดความเหมือนตามสายอักขระแบบใดที่สามารถใช้ร่วมกับมาตรวัดแบบ tf-idf แล้วค้นหาข้อมูลได้แม่นยำที่สุด

1.2 วัตถุประสงค์

ปัญหาพิเศษนี้จะมีการศึกษามาตรวัดความเหมือนแบบผสม โดยนำมาตรวัดตามคำแบบ tf-idf มาทดลองใช้ร่วมกับมาตรวัดความเหมือนตามอักขระแบบต่างๆ กับชุดข้อมูลเพื่อดูว่ามาตรวัดใดเหมาะสมกับชุดข้อมูลแต่ละชุดและให้ความแม่นยำในการค้นหามากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปัญหาพิเศษ

ปัญหาพิเศษนี้จะทำการทดลองมาตรวัดความเหมือนแบบผสม โดยมีขอบเขตที่ต้องการให้มาตรวัดความเหมือนแบบผสม ที่เกิดจากจากมาตรวัดตามคำและมาตรวัดความเหมือนตามอักขระมาใช้ร่วมกันแล้วได้ความถูกต้องสำหรับการค้นหาให้มากที่สุดในชุดข้อมูลแบบต่างๆ โดยทดลองกับมาตรวัดแบบ tf-idf กับมาตรวัดแบบอักขระกับแต่ละชุดข้อมูล แล้วหาค่าเฉลี่ยความแม่นยำ เพื่อดูว่ามาตรวัดแบบตามสายอักขระใดเหมาะที่จะใช้กับมาตรวัดตามคำแบบ tf-idf

1.4 ส่วนประกอบของวิทยานิพนธ์

1. ศึกษาข้อมูลเกี่ยวกับมาตรวัดความเหมือนข้อความในแต่ละแบบ
2. กำหนดขอบเขต วัตถุประสงค์ และความต้องการของปัญหา
3. วิเคราะห์และออกแบบการทดลอง
4. ทดลองการทำงานของมาตรวัดความเหมือนข้อความแบบผสม
5. หาค่าเฉลี่ยความแม่นยำของมาตรวัดแบบผสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้เป็นการกล่าวถึงมาตรวัดที่ใช้ในการวัดความเหมือนของข้อความที่เกี่ยวข้อง และทฤษฎีต่างๆที่ใช้ในปัญหาพิเศษ ในส่วนที่ 2.1.1 กล่าวถึงมาตรวัดตามสายอักขระ 2.1.2 กล่าวถึงมาตรวัดตามคำ 2.1.3 กล่าวถึงมาตรวัดแบบผสม 2.1.4 กล่าวถึงการประเมินผลเพื่อหาค่าเฉลี่ยความแม่นยำ และสุดท้าย 2.2 งานวิจัยที่เกี่ยวข้องกับปัญหาพิเศษนี้

2.1 ทฤษฎีการวัดความเหมือนของข้อความ

2.1.1 มาตรวัดความเหมือนตามสายอักขระ

มาตรวัดความเหมือนตามสายอักขระออกแบบเพื่อจัดการกับข้อผิดพลาดของการพิมพ์ ซึ่งมีมาตรวัดต่างๆดังนี้

2.1.1.1 เลเวนสเทน (Levenshtein) เป็นมาตรวัดที่ให้ค่าคะแนนจากจำนวนการกระทำ (การเพิ่ม, ลบ, แทนที่) ที่น้อยที่สุดกับตัวอักษรทีละตัว เพื่อเปลี่ยนข้อความหนึ่งเป็นอีกข้อความหนึ่ง โดยวิธีที่ง่ายที่สุดคือกำหนดค่าแต่ละการกระทำเท่ากับ 1 โดยสามารถหาได้จากสมการตามรูปที่ 2.1

$$
D(i,j) = \min \begin{cases} 0 & \text{// ตำแหน่งเริ่มต้น} \\ D(i-1,j-1) + W(i,j) & \text{// แทนที่} \\ D(i-1,j)+1 & \text{// เพิ่ม} \\ D(i,j-1)+1 & \text{// ลบ} \end{cases}
$$

รูปที่ 2.1 สมการเลเวนสเทน

การคำนวณค่าความเหมือนของข้อความ S_1 = kitten และ S_2 = sitting ซึ่งหาได้จากสมการตามรูปที่ 2.2

กำหนดให้ i คือ ตำแหน่งอักขระของข้อความที่ 1

j คือ ตำแหน่งอักขระของข้อความที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ W(i,j) ในสมการแทนที่จะมีค่าเป็น 1 เมื่อ $i \neq j$

W(i,j) ในสมการแทนที่จะมีค่าเป็น 0 เมื่อ $i = j$

		k	i	t	t	e	n
	0	1	2	3	4	5	6
s	1	1	2	3	4	5	6
i	2	2	1	2	3	4	5
t	3	3	2	1	2	3	4
t	4	4	3	2	1	2	3
i	5	5	4	3	2	2	3
n	6	6	5	4	3	3	2
g	7	7	6	5	4	4	3

$$D(1,2) = \min \begin{cases} D(0,1) + W(1,2) = 1 + 1 = 2 \\ D(0,2) + 1 = 2 + 1 = 3 \\ D(1,1) + 1 = 1 + 1 = 2 \end{cases}$$

รูปที่ 2.2 ตัวอย่างการคิดสมการเลเวนสเทนที่ตำแหน่ง 1, 2

2.1.1.2 จาโร (Jaro) เป็นมาตรวัดที่ให้ค่าคะแนนจากจำนวนตัวอักษรร่วมของทั้งสองข้อความ และจำนวนตัวอักษรร่วมของทั้งสองข้อความที่ไม่เหมือนกัน โดยอักษรร่วม คือ อักษรที่เหมือนกันและอยู่ในตำแหน่งใกล้เคียงกัน หาได้จากสมการตามรูปที่ 2.3

$$d_j = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right)$$

รูปที่ 2.3 สมการจาโร

หาระยะห่างความเหมือนของอักขระระหว่าง 2 ข้อความซึ่งอักขระที่เหมือนกันต้องห่างกันไม่เกิน

จากสูตร $\left\lfloor \frac{\max(|s_1|,|s_2|)}{2} \right\rfloor - 1.$

ถ้าอักขระใน 2 ข้อความไม่มีการสลับตำแหน่ง จะกำหนดให้ t = 0

โดยที่ s_1, s_2 คือ ข้อความ s_1 และ s_2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

m คือ จำนวนอักษรของทั้งสองข้อความที่เหมือนกัน

t คือ $\left\lfloor \frac{\text{จำนวนรวมของการเปลี่ยนตำแหน่งตัวอักษร}}{2} \right\rfloor$

ต่อมามีการเสนอจาโรวิงเคอร์ (Jaro-Winkler) เป็นการปรับปรุงจาโรเนื่องจากพบว่ามักเกิดความคลาดเคลื่อนกับตัวอักษรตัวที่สี่เป็นต้นไป มาตรวัดจาโรวิงเคอร์จึงนำความยาวตัวอักษรร่วมที่ยาวที่สุดในตอนต้นของสายอักขระ ซึ่งความยาวไม่เกินสี่ตัวอักษรมาใช้เพื่อปรับปรุงความคลาดเคลื่อนนั้น หาได้จากสมการตามรูปที่ 2.4

$$d_w = d_j + (\ell p(1 - d_j))$$

รูปที่ 2.4 สมการจาโรวิงเคอร์

โดยที่ d_j คือ ระยะทางของวิธีจาโร ระหว่างข้อความ s_1 และ s_2

l คือ ความยาวของคำนำหน้าข้อความสูงสุดไม่เกิน 4 ตัวอักษรแรกที่เหมือนกัน

p เป็นค่าคงที่ที่ใช้ปรับเพิ่มขึ้นสำหรับคำนำหน้า ค่ามาตรฐานสำหรับค่าคงที่ในการทำงาน Winkler คือ p = 0.1

การคำนวณค่าความเหมือนของข้อความ s_1 เป็น MARTHA และ s_2 เป็น MARHTA หาได้จากรูปที่ 2.5

	M	A	R	T	H	A
M	1	0	0	0	0	0
A	0	1	0	0	0	0
R	0	0	1	0	0	0
H	0	0	0	0	1	0
T	0	0	0	1	0	0
A	0	0	0	0	0	1

รูปที่ 2.5 ตัวอย่างการคิดความเหมือนของข้อความ MARTHA และ MARHTA

เริ่มจากการหาระยะห่างความเหมือนกันของอักขระระหว่าง 2 ข้อความ

จากสูตร $\left\lfloor \frac{\max(|s_1|,|s_2|)}{2} \right\rfloor - 1$ $= \frac{5}{2} - 1 = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

m = 6 (มีอักษรของทั้งสองข้อความที่มีเหมือนกัน คือ มี M, A, R, T, H, A)

$s_1 = 6$ $s_2 = 6$

L = 3 (มาจากอักษรนำหน้าซึ่งมี 3 ตัวอักษร คือ M, A, R)

t = $\frac{2}{2}$ = 1 (เนื่องจาก มีการสลับตำแหน่งจาก T/H กับ H/T ทั้งหมดเป็น 2 คู่)

คำนวณตามสูตรของจาโร จะได้

$$d_j = \frac{1}{3}\left(\frac{6}{6} + \frac{6}{6} + \frac{6-1}{6}\right) = 0.944$$

นำค่า d_j มาคำนวณในสูตรของจาโรวิงเคอร์ ได้ดังนี้

$$d_w = 0.944 + (3 * 0.1(1 - 0.944)) = 0.961$$

การคำนวณค่าความเหมือนของข้อความ s_1 เป็น JONES และ s_2 เป็น JOHNSON หาได้จากรูปที่ 2.6

	J	O	N	E	S
J	1	0	0	0	0
O	0	1	0	0	0
H	0	0	0	0	0
N	0	0	1	0	0
S	0	0	0	0	1
O	0	1	0	0	0
N	0	0	1	0	0

รูปที่ 2.6 ตัวอย่างการคิดความเหมือนของข้อความ JONES และ JOHNSON

m = 4 (มีอักษรของทั้งสองข้อความที่มีเหมือนกัน คือ มี J, O, N, S)

$s_1 = 5$ $s_2 = 7$

L = 2 (มาจากอักษรนำหน้าซึ่งมี 2 ตัวอักษร คือ J, O)

t = 0 (เนื่องจากไม่มีการสลับตำแหน่งของตัวอักษร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวณตามสูตรของจาโร จะได้

$$Dj = \frac{1}{3} \times \left(\frac{4}{5} + \frac{4}{7} + \frac{4-0}{4}\right) = 0.790$$

นำค่า d_j มาคำนวณในสูตรของจาโรวิงเคอร์ ได้ดังนี้

$$d_w = 0.790 + (2 * 0.1(1 - 0.790)) = 0.832$$

2.1.1.3 สมิธวอเตอร์แมน (Smith-Waterman) เป็นมาตรวัดที่พัฒนามาจากเลเวนสเทน โดยพิจารณาจากการจัดเรียงตำแหน่งของตัวอักษรในตำแหน่งที่ใกล้เคียงกันของข้อความ มีการเพิ่มช่องว่าง (Gap) ให้มีค่าการกระทำน้อยกว่าตัวกระทำการเพิ่ม ลบ และแทนที่ ทำให้ตัวอักษรตอนกลางถูกให้ความสำคัญมากกว่าตำแหน่งตอนต้นและท้าย หาได้จากสมการตามรูปที่ 2.7

$$D(i,j) = \max \begin{cases} 0 & \text{// ตำแหน่งเริ่มต้น} \\ D(i-1,j-1) + w(i,j) & \text{// แทนที่} \\ D(i-1,j) + w(i,-) & \text{// ลบ} \\ D(i,j-1) + w(-,j) & \text{// เพิ่ม} \end{cases}$$

รูปที่ 2.7 สมการสมิธวอเตอร์แมน

โดยที่ *a* , *b* คือ ข้อความ

i , j คือ ตำแหน่งของตัวอักษร

D(i,j) คือ ค่าสูงสุดของการกระทำการ

w(i,j) คือ ค่าคะแนนความเหมือนของอักขระ

การคำนวณค่าความเหมือนของข้อความ a เป็น ACACACTA และ b เป็น AGCACACA หาได้จากรูปที่ 2.8

ในที่นี้กำหนดให้ $w(a,b) = 2$ เมื่อ $i = j$

$w(a,b) = -1$ เมื่อ $i \neq j$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	-	A	C	A	C	A	C	T	A
-	0	0	0	0	0	0	0	0	0
A	0	2							
G	0	1							
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

$$D(1,1) = \text{Max}\begin{cases} D(0,0) + W(1,1) & = 0 + 2 & = & 2 \\ D(0,1) + W(a,-) & = 0 + (-1) & = & -1 \\ D(1,0) + W(-,b) & = 0 + (-1) & = & -1 \end{cases}$$

$$D(2,1) = \text{Max}\begin{cases} D(1,0) + W(2,1) & = 0 + (-1) & = & -1 \\ D(1,1) + W(a,-) & = 2 + (-1) & = & 1 \\ D(2,0) + W(-,b) & = 0 + (-1) & = & -1 \end{cases}$$

คำนวณจนได้ครบทั้งเมทริกซ์ดังนี้

	–	A	C	A	C	A	C	T	A
–	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

$$D(1,1) = \text{Max}\begin{cases} D(0,0) + W(1,1) & = 0 + 2 & = & 2 \\ D(0,1) + W(a,-) & = 0 + (-1) & = & -1 \\ D(1,0) + W(-,b) & = 0 + (-1) & = & -1 \end{cases}$$

รูปที่ 2.8 ตัวอย่างการคิดสมิธวอเตอร์แมน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ที่ได้จะเป็น

ข้อความที่ 1 = A-CACACTA

ข้อความที่ 2 = AGCACAC-A

2.1.1.4 มองค์อัลคัน (Monge Elkan) เป็นมาตรวัดความเหมือนของข้อความ โดยนำข้อความย่อยมาวนซ้ำเปรียบเทียบกันด้วยสมิธวอเตอร์แมนโกโท เพื่อให้ได้คะแนนค่าความเหมือนของข้อความย่อยที่เหมือนกันมากที่สุด หาได้จากสมการตามรูปที่ 2.9

$$sim(s,t) = \frac{1}{K}\sum_{i=1}^{K}\max_{j=1}^{L} sim'(A_i, B_j)$$

รูปที่ 2.9 สมการมองค์อัลคัน

โดยที่ ข้อความ s แตกเป็นข้อความย่อย ($A_1, A_2, \ldots, A_K$)

ข้อความ t แตกเป็นข้อความย่อย ($B_1, B_2, \ldots, B_L$)

sim' (A_i, B_i) เปรียบเทียบแต่ละข้อความย่อยด้วยสมิธวอเตอร์แมนโกโท

2.1.2 มาตรวัดความเหมือนตามคำ

โคเฮ็น (Cohen) เสนอวัดค่าความเหมือนระหว่างสองข้อความด้วยโคไซน์ (Cosine similarity) จากการให้คะแนนค่าน้ำหนักจากความถี่ของคำที่ปรากฏอยู่ เมื่อได้คะแนนค่าน้ำหนักของคำแล้วนำมาเปรียบเทียบความเหมือนกันด้วยโคไซน์ ถ้าทำมุมน้อยแสดงว่าสองข้อความมีความเหมือนกันมาก

การวัดค่าความเหมือนของข้อความโดยการให้คะแนนค่าน้ำหนักด้วย tf-idf สามารถหาได้จากสมการตามรูปที่ 2.10

$$TFIDF(S,T) = \sum V(w,S) \cdot V(w,T)$$

$$V'(w,S) = log(TF_{w,S} + 1) \cdot log(IDF_w)$$

$$V(w,S) = V'(w,S) / \sqrt{\sum V'(w,S)^2}$$

รูปที่ 2.10 สมการ tf-idf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่	S,T	คือ	ข้อความ S และ T
	V(w,S), V'(w,S)	คือ	คะแนนค่าน้ำหนักของคำ w ในข้อความ S
ให้	tf	คือ	จำนวนความถี่ของคำที่พบในเอกสาร
	idf	คือ	อัตราส่วนกลับของจำนวนคำทั้งหมดในเอกสารกับจำนวนคำที่พบคำนั้นปรากฏอยู่

2.1.3 มาตรวัดความเหมือนแบบผสม

ในงานวิจัยได้นำมาตรวัดความเหมือนแบบผสมระหว่างมาตรวัดความเหมือนแบบตามสายอักขระและมาตรวัดตามคำ โดยใช้ชื่อว่า Soft tf-idf ซึ่งได้แนวคิดจากการวนซ้ำเปรียบเทียบข้อความย่อยของมาตรวัดมองค์อัลคัน

กำหนดให้มาตรวัดที่ใช้ในการเปรียบเทียบข้อความย่อยเป็นฟังก์ชั่นลำดับสอง Soft tf-idf ใช้ฟังก์ชันลำดับสองนี้วนซ้ำเพื่อเปรียบเทียบคำในทั้งสองข้อความและหาค่าความเหมือนที่สูงสุด จากนั้นนำมาถ่วงน้ำหนักร่วมกับค่าของคำใน tf-idf โดยมาตรวัด Soft tf-idf นี้ใช้จาโรวิงเคอร์เป็นฟังก์ชันอันดับสอง

มาตรวัด Soft tf-idf หาได้จากสมการตามรูปที่ 2.11

$$SoftTFIDF(S,T) = \sum_{w \in CLOSE(\theta,S,T)} V(w,S) \cdot V(w,T) \cdot D(w,T)$$

รูปที่ 2.11 สมการ Soft tf-idf

โดยที่	s, t	คือ ข้อความ s กับข้อความ t
	D (w,T)	$= MAX_{v \in T}$ dist(w, v)
	dist(w, v)	คือ ฟังก์ชันลำดับสองที่ใช้หาค่าความเหมือนด้วยจาโรวิงเคอร์
	V (w, s)	คือ คะแนนค่าน้ำหนักของคำ w ในข้อความ s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การประเมินผล

ในงานของมาตรวัดความเหมือนข้อความระหว่างสองแหล่งข้อมูลจะใช้วิธีดังนี้

1.) ค่าความแม่นยำ (P) เป็นเปอร์เซ็นต์ความถูกต้องของการวัดความเหมือนความเหมือนข้อความ หาได้จากอัตราส่วนของข้อความที่วัดความเหมือนได้ถูกต้องกับจำนวนข้อความทั้งหมดที่วัดได้ หาได้จากสมการตามรูปที่ 2.12

$$P = \frac{\text{จำนวนข้อความที่วัดถูกต้อง}}{\text{จำนวนข้อความทั้งหมดที่วัดได้}}$$

รูปที่ 2.12 สมการของค่าความแม่นยำ

2.) ค่าเฉลี่ยความแม่นยำ (Average Precision) เพื่อใช้วัดประเมินประสิทธิภาพของมาตรวัด ซึ่งหาได้จากสมการตามรูปที่ 2.13

$$\text{Avg. Precision} = \frac{1}{m} \sum_{r=1}^{N} c(i) \frac{s(i)}{i}$$

รูปที่ 2.13 สมการของค่าเฉลี่ยความแม่นยำ

โดยที่ *c(i)* คือ จำนวนที่วัดถูกต้องก่อนข้อความที่ *i*

s(i) จะมีค่าระหว่าง 0 -1 ซึ่ง

ถ้าเข้าใกล้ 1 แสดงว่าข้อความที่ *i* มีการวัดข้อความได้ถูกต้องมาก

ถ้าเข้าใกล้ 0 แสดงว่าข้อความที่ *i* มีการวัดข้อความได้ถูกต้องน้อย

N คือ จำนวนข้อความทั้งหมดที่วัดว่าเหมือนกัน

m คือ จำนวนข้อความทั้งหมดที่วัดถูกต้อง

r คือ ลำดับ (rank)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่ผ่านมานำเสนอมาตรวัดความเหมือนข้อความ tf-idf แบบที่ทนทานต่อข้อผิดพลาด โดยแก้ปัญหาข้อความที่ต้องการค้นหาในฐานข้อมูลข้อความมีการพิมพ์ผิดหรือสลับตำแหน่งเกิดขึ้น ในขณะที่มาตรวัดอื่นๆยังให้ค่าความแม่นยำของมาตรวัดที่ไม่สูงมากนัก ซึ่งมาตรวัดที่งานวิจัยนำเสนอได้ทดลองแล้วพบว่านำสมิธวอเตอร์แมนโกโทมาใช้ร่วมกับ tf-idf ทดลองกับชุดข้อมูลต่างๆ แล้วให้ค่าความแม่นยำที่สูงที่สุด ในระดับเกณฑ์ค่าความเหมือนที่กำหนดไว้คือ 0.9 ยกเว้นชุดข้อมูล Animal ที่ใช้มาตรวัดความเหมือนข้อความ tf-idf แบบที่ทนทานต่อข้อผิดพลาดแล้วให้ค่าความแม่นยำน้อยกว่ามาตรวัดแบบจาโรวิงเคอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การศึกษาทดลองมาตรวัดความเหมือนข้อความแบบผสม

มาตรวัดความเหมือนข้อความโดย แบ่งออกเป็น 2 ประเภทใหญ่ๆ คือ มาตรวัดความเหมือนแบบตามสายอักขระ และมาตรวัดความเหมือนแบบตามคำ ซึ่งจากผลการทดลองก่อนหน้า พบว่า มาตรวัดความเหมือนตามสายอักขระ สามารถวัดความเหมือนของข้อความได้ในกรณีที่ข้อความนั้นพิมพ์ผิด แต่ไม่สามารถวัดความเหมือนของข้อความได้ในกรณีที่มีการสลับตำแหน่งของคำ และมาตรวัดความเหมือนแบบตามคำ ก็สามารถวัดความเหมือนของข้อความได้ในกรณีที่มีการสลับตำแหน่งของคำ แต่ไม่สามารถวัดความเหมือนของข้อความได้เมื่อมีคำที่ผิด ดังนั้นในการทดลองนี้จะทำการศึกษามาตรวัดความเหมือนของข้อความโดยใช้มาตรวัดตามคำแบบ tf-idf ทดสอบกับมาตรวัดตามสายอักขระอื่น ๆ เพื่อหามาตรวัดที่ให้ค่าเฉลี่ยความแม่นยำที่ดีที่สุดในการวัดความเหมือนของข้อความ (An Experiment Study on Hybrid Text Similarity)

3.1 การวัดความเหมือนของข้อความแบบผสม

ปัญหาพิเศษนี้ต้องการค้นหามาตรวัดความเหมือนแบบตามสายอักขระที่ทำงานร่วมกับมาตรวัดความเหมือนแบบข้อความ tf-idf ที่เพิ่มประสิทธิภาพในการค้นหาข้อความได้อย่างแม่นยำมากยิ่งขึ้น หาได้จากสมการตามรูปที่ 3.1

$$\text{Error tolerance tf-idf}(S, T) = \sum V(w,S) \cdot V(w,T)$$

$$V'(w,S) = \log(\text{simmetric' } tf_{w,S} + 1) \cdot \log(\text{simmetric' } idf_w)$$

$$V(w,S) = V'(w,S) / \sum V'(w,S)^2$$

โดยที่ S,T คือ ข้อความ S และ T

V(w,S), V'(w,S) คือ คะแนนค่าน้ำหนักของคำ w ในข้อความ S

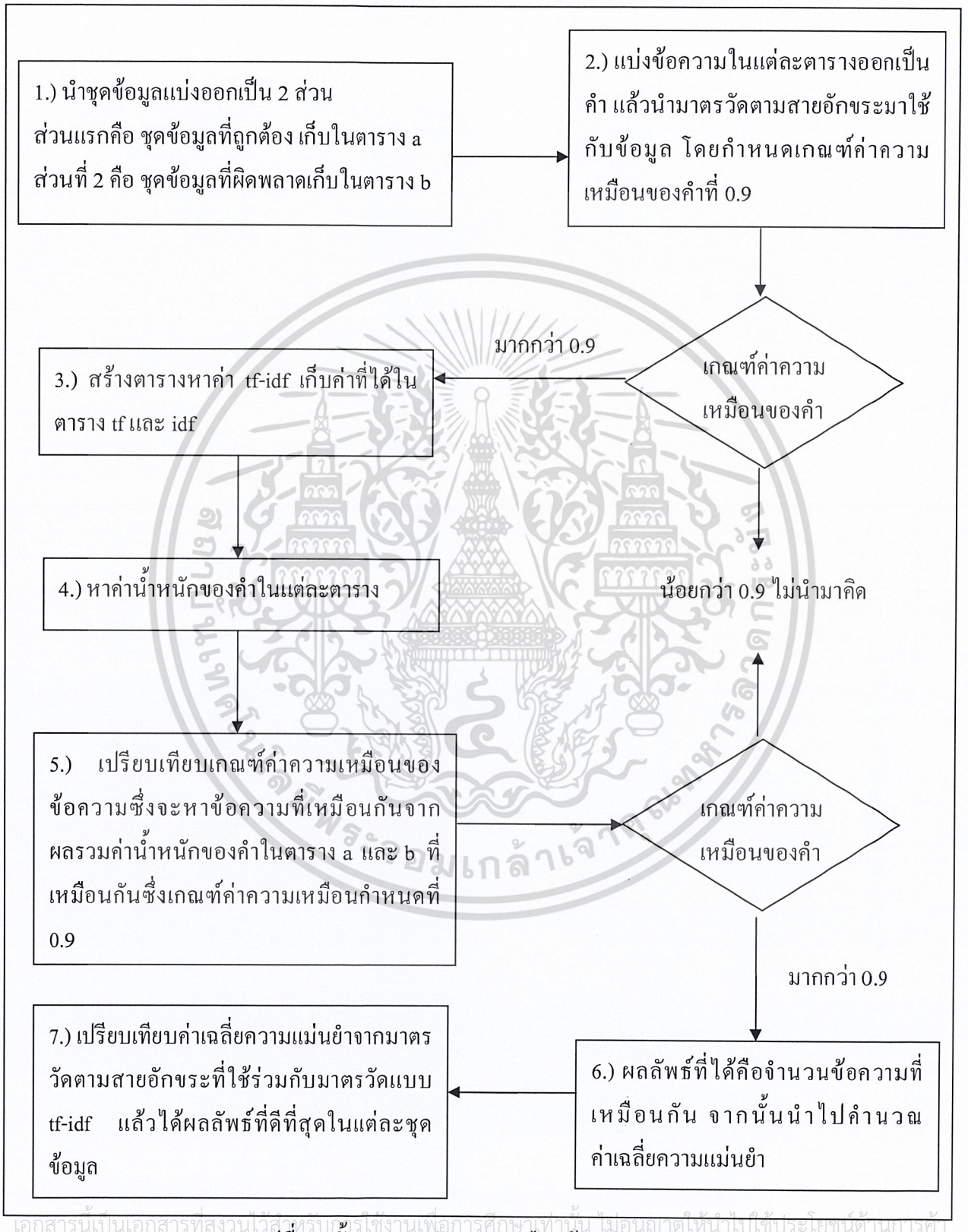
ให้ Simmetric'tf คือ จำนวนความถี่ของคำซึ่งเปรียบเทียบจากมาตรวัดความเหมือนตามสายอักขระที่พบในเอกสาร

Simmetric'idf คือ อัตราส่วนกลับของจำนวนคำทั้งหมดกับจำนวนคำที่มีคำ w ปรากฏซึ่งเทียบกับมาตรวัดความเหมือนตามสายอักขระ

รูปที่ 3.1 การวัดค่าความเหมือนของข้อความโดยให้คะแนนค่าน้ำหนักด้วยมาตรวัดแบบ if-idf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการศึกษาทดลองมาตรวัดความเหมือนของข้อความแบบผสมมีการดำเนินงานตามรูปที่ 3.2 ดังนี้



รูปที่ 3.2 ขั้นตอนการวัดความเหมือนข้อความแบบผสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ชุดข้อมูล

ฐานข้อมูลที่ใช้ในการทดลองค้นหามาตรวัดความเหมือนของข้อความจะนำชุดข้อมูลมาสร้างเป็นฐานข้อมูล

- **ตารางฐานข้อมูล**

ในส่วนของฐานข้อมูลเราจะแบ่งชุดข้อมูลออกเป็น 2 ตารางเพื่อทำการทดลองซึ่งจะมีตารางเพื่อใช้เก็บข้อมูลต่างๆดังนี้

ตารางที่ 3.1 ตารางจัดเก็บข้อความ โดยจะเก็บข้อความไว้ 2 ตาราง ตาราง a,b

ตาราง a

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของข้อความ
2	string	Nvarchar(255)	ข้อความ

ตาราง b

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของข้อความ
2	string	Nvarchar(255)	ข้อความ

ตารางที่ 3.2 ตารางจัดเก็บคำของข้อความ ซึ่งจัดเก็บคำได้ 2 ตาราง token_a , token_b

ตาราง token_a

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของข้อความ
2	token_id	Int	รหัสของคำ
3	token	Nvarchar(255)	ข้อความที่ตัดเป็นคำ

ตาราง token_b

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของข้อความ
2	token_id	Int	รหัสของคำ
3	token	Nvarchar(255)	ข้อความที่ตัดเป็นคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการจัดเก็บคำของแต่ละข้อความ จะเขียนโปรแกรมขึ้นมาตัดคำด้วยภาษา PHP เมื่อเจอช่องว่างก็จะตัดคำออกมา แล้วเก็บใส่ในตาราง token อัลกอริทึมที่ใช้ตัดคำ สามารถดูได้จากรูปที่ 3.3

```
<? $objConnect = odbc_connect("dataset_census","","") or die("Error Connect to Database");

if(!$objConnect)

{ exit("Error". odbc_error() ."! -". odbc_errormsg() . ".");}

$SQL = "SELECT record_id,string FROM a order by 1";

$result = odbc_exec($objConnect,$SQL);          $count=1;

while($row = odbc_fetch_array($result)){

 $newStr = ereg_replace('[[:space:]]+', ' ', trim($row['attr0']));

 $explode = explode(" " , $newStr);

foreach($explode as $i=>$values) {

$test = htmlspecialchars($explode[$i], ENT_QUOTES);

      $SQL = "insert into token2(record_id,token_id,token) VALUES ('$row[record_id]', '$count', '$test')";

                  if (!odbc_exec($objConnect,$SQL))

                  {   die($SQL);}

                  else

                {echo "DEBUG: $SQL<br><span style='color:#00ff00;'>ป้อนสำเร็จ!</span><br>";}

              $count++;

        }

}

odbc_close($objConnect); ?>
```

รูปที่ 3.3 อัลกอริทึมที่ใช้ในการตัดข้อความเป็นคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ตารางที่ใช้จัดเก็บค่า tf คือ ตาราง tf_a,tf_b

ตาราง tf_a

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของแถว
2	token	Int	ข้อความที่ตัดเป็นคำ
3	tf	Float	ค่าความถี่ของคำ

ตาราง tf_b

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของแถว
2	token	Int	ข้อความที่ตัดเป็นคำ
3	tf	Float	ค่าความถี่ของคำ

จากตารางที่ 3.3 จะเขียนคำสั่ง SQL เก็บค่า tf ในแต่ละตาราง จากตาราง token_a, token_b โดยที่จะหามาตรวัดตามสายอักขระของข้อความที่มีค่าความเหมือนมากกว่า θ (ในที่นี้กำหนดเป็น 0.9) ตามรูปที่ 3.4

```
insert into tf_a
select a.record_id , a.token_id , log(count(*)+1) as tf
from token_a as a , token_a as b
where simmetric(a.token,b.token) > 0.9 and a.record_id =
b.record_id
group by 1,2
order by 1,2

insert into tf_b
select a.record_id , a.token_id , log(count(*)+1) as tf
from token_b as a , token_b as b
where simmetric(a.token,b.token) > 0.9 and a.record_id =
b.record_id
group by 1,2
order by 1,2
```

รูปที่ 3.4 คำสั่ง SQL ที่ใช้ในเก็บค่า tf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

117237

ตารางที่ 3.4 ตารางที่ใช้จัดเก็บค่า idf คือตาราง idf_a,idf_b

ตาราง idf_a

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	token	Nvarchar(255)	คำ
2	idf	Float	ค่าของจำนวนคำที่พบ

ตาราง idf_b

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	token	Nvarchar(255)	คำ
2	idf	Float	ค่าของจำนวนคำที่พบ

จากตารางที่ 3.4 จะเขียนคำสั่งจัดเก็บค่า idf ในแต่ละตารางโดยจะหาค่า log จากจำนวนแถวทั้งหมดของ ตาราง a และ b หารด้วยจำนวนของคำนั้นที่พบ ซึ่งคำในและมาตรวัดตามสายอักขระต่าง ๆ(simmetric)ต้องมีค่ามากกว่าค่าความเหมือนที่กำหนด ตามรูปที่ 3.5

```
insert into idf_a
select     a.token , log(number_record_a/count(b.record_id)) as idf
from       (SELECT DISTINCT token
                    FROM  token_a) AS a CROSS JOIN
                    token_a AS b
where      (simmetric(a.token, b.token) > 0.9)
group by 1 ;

insert into idf_b
select     a.token , log(number_record_b/count(b.record_id)) as idf
from       (SELECT DISTINCT token
                    FROM  token_b) AS a CROSS JOIN
                    token_a AS b
where      (simmetric(a.token, b.token) > 0.9)
group by 1 ;
```

รูปที่ 3.5 คำสั่ง SQL ที่ใช้ในการหาค่าของ idf

ตารางที่ 3.5 ตารางที่ใช้เก็บค่าตัวหาร คือ ตาราง len_a, len_b

ตาราง len_a

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของแถว
2	len	Float	ค่าของตัวหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง len_b

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของแถว
2	len	Float	ค่าของตัวหาร

จากตารางที่ 3.5 นำค่าจากตาราง tf, token, idf หาค่ารากที่สองของผลรวมค่า tf^2 และ idf^2 ตามรูปที่ 3.6

```
insert into len_a
select record_id , sqrt(sum(I.idf*I.idf*T.tf*T.tf))
from (SELECT      a.record_id, b.token, a.tf
FROM          tf_a as a inner JOIN
token_a as b ON a.record_id = b.record_id AND
a.token = b.token_id) as T , idf_a as  I
where T.token = I.token
group by T.record_id;

insert into len_b
select record_id , sqrt(sum(I.idf*I.idf*T.tf*T.tf))
from (SELECT      a.record_id, b.token, a.tf
FROM          tf_a as a inner JOIN
token_a as b ON a.record_id = b.record_id AND
a.token = b.token_id) as T , idf_a as  I
where T.token = I.token
group by T.record_id;
```

รูปที่ 3.6 คำสั่ง SQL ที่ใช้ในการหาค่ารากที่สองของผลรวมค่า tf^2 และ idf^2

ตารางที่ 3.6 ตารางที่ใช้เก็บค่าน้ำหนักของคำ คือ ตาราง weight_a, weight _b

ตาราง weight_a

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของข้อความ
2	len	Float	ค่าของตัวหาร

ตาราง weight_b

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_id	Int	รหัสของข้อความ
2	len	Float	ค่าของตัวหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 3.6 จะเก็บค่าน้ำหนักของคำ ซึ่งในตาราง weight จะหาได้จาก ตาราง token ,tf และ len ด้วยคำสั่ง T.tf * I.idf / L.len ตามรูปที่ 3.7

```
insert into weight_a
select T.record_id , T.token , T.tf * I.idf / L.len
from
(SELECT      a.record_id, b.token, a.tf
FROM          TF_A as a inner JOIN
Token_A as b ON a.record_id = b.record_id AND
a.token = b.token_id) as T ,
IDF1 as  I , len_A as L
where T.token = I.token and T.record_id = L.record_id;

insert into weight_b
select T.record_id , T.token , T.tf * I.idf / L.len
from
(SELECT      a.record_id, b.token, a.tf
FROM          TF_A as a inner JOIN
Token_A as b ON a.record_id = b.record_id AND
a.token = b.token_id) as T ,
IDF1 as  I , len_A as L
where T.token = I.token and T.record_id = L.record_id;
```

รูปที่ 3.7 คำสั่ง SQL ที่ใช้ในการหาค่าน้ำหนักของคำ

ตารางที่ 3.7 ตารางที่ใช้เปรียบเทียบข้อความที่เหมือนกันของค่า tf-idf

ตาราง tf-idf

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
1	record_ida	Int	รหัสของข้อความตาราง a
2	record_idb	Int	รหัสของข้อความตาราง b
3	weight	Float	ค่าของตัวหาร

จากตารางที่ 3.7 เป็นตารางที่ใช้เปรียบเทียบข้อความที่เหมือนกันระหว่าง 2 ข้อความ โดยหาได้จากตารางที่ 3.6 ตามรูปที่ 3.8

```
Insert tf-idf
select weight_a.record_id , weight_b.record_id
 SUM(weight_a.weight* weight_a.weight)
from weight_a , weight_b
where weight_a.token = weight_b.token
group by weight_a.record_id , weight_b.record_id
having SUM(weight_a.weight* weight_b.weight) >= 0.9
order by 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 คำสั่ง SQL ที่ใช้ในการเปรียบเทียบข้อความที่เหมือนกันระหว่าง 2 ข้อความ

3.3 ตัวอย่างวิธีการวัดความเหมือนของข้อความ tf-idf

1.) นำชุดข้อมูลมาสร้างฐานข้อมูล โดยแบ่งชุดข้อมูลออกเป็น 2 ส่วนเก็บในตารางที่ 3.8 ดังนี้

ตาราง a เป็นข้อมูลที่ถูกต้อง

ตาราง b เป็นข้อมูลที่สังเคราะห์ขึ้นเพื่อให้เกิดคำผิดพลาด และใช้ในการทดลอง

ตารางที่ 3.8 ตารางตัวอย่างที่ใช้เก็บข้อความ

ตาราง a

record_id	string
1	DANNY BASSWOOD
2	DANNY WEBSTER
3	DANNY BANK
4	DANNY STARKEY
5	WEBSTER DANNY

ตาราง b

record_id	string
1	WEBSTER DENNY

2.) จากตารางที่ 3.8 จะทำการตัดข้อความออกเป็นคำ ๆ โดยตัดคำจากช่องว่าง แล้วเก็บค่าในตาราง token รวมทั้งกำหนดรหัสข้อความและรหัสคำ ตามตารางที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 ตารางตัวอย่างที่ใช้เก็บคำ

ตาราง token_a

record_id	token_id	token
1	1	DANNY
1	2	BASSWOOD
2	1	DANNY
2	2	WEBSTER
3	1	DANNY
3	2	BANK
4	1	DANNY
4	3	STARKEY
5	1	WEBSTER
5	2	DANNY

ตาราง token_b

record_id	token_id	token
1	1	WEBSTER
1	2	DENNY

3.) จากตารางที่ 3.9 เมื่อตัดข้อความออกเป็นคำ ต่อมาให้หาค่า tf โดยหาจาก log(count*+1) จากตาราง token ซึ่ง count* จะนับจำนวนคำที่พบในข้อความ แล้วหาค่า log จากนั้นเก็บค่าความถี่ของคำในตาราง tf ตามตารางที่ 3.10 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 ตารางตัวอย่างที่ใช้เก็บค่า tf

ตาราง tf_a

record_id	token_id	token	tf
1	1	DANNY	LOG(1+1) = 0.301
1	2	BASSWOOD	LOG(1+1) = 0.301
2	1	DANNY	LOG(1+1) = 0.301
2	2	WEBSTER	LOG(1+1) = 0.301
3	1	DANNY	LOG(1+1) = 0.301
3	2	BANK	LOG(1+1) = 0.301
4	1	DANNY	LOG(1+1) = 0.301
4	3	STARKEY	LOG(1+1) = 0.301
5	1	WEBSTER	LOG(1+1) = 0.301
5	2	DANNY	LOG(1+1) = 0.301

ตาราง tf_b

record_id	token_id	token	tf
1	1	WEBSTER	LOG(1+1) = 0.301
1	2	DENNY	LOG(1+1) = 0.301

4.) ตาราง idf หาได้จากตาราง token ของคำแต่ละคำ โดยหาค่า idf ได้จาก log (number_record_a/count (b.record_id)) ซึ่ง number_record_a คือจำนวนแถวทั้งหมดของตาราง a หารด้วยจำนวนแถวที่พบคำนั้น จากนั้นก็หาค่า log ของ idf ตามตารางที่ 3.11 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 ตารางตัวอย่างที่ใช้เก็บค่า idf

ตาราง idf_a

token	idf
DANNY	LOG (5 / 5) = 0
BASSWOOD	LOG (5 / 1) = 0.6990
WEBSTER	LOG (5 / 2) = 0.3980
BANK	LOG (5 / 1) = 0.6990
4848	LOG (5 / 1) = 0.6990
STARKEY	LOG (5 / 1) = 0.6990

ตาราง idf_b

token	idf
WEBSTER	LOG (5 / 2) = 0.3980
DENNY	LOG (5 / 5) = 0

5.) สร้างตาราง len หาค่าได้จากตาราง tf, idf หาค่ารากที่สองของผลรวมค่า tf^2 และ idf^2 เก็บในตารางที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.12 ตารางตัวอย่างที่ใช้เก็บค่าที่เป็นส่วนหารในการคิดค่าความเหมือนของข้อความ

ตาราง len_a

record_id	len SQRT(SUM (tf * tf *idf * idf))
1	$SQRT(0.301)^2*(0.301)^2+(0)^2*(0.6990)^2 = (0.301)^2$
2	$SQRT(0.301)^2*(0.301)^2+(0)^2*(0.3980)^2 = (0.301)^2$
3	$SQRT(0.301)^2*(0.301)^2+(0)^2*(0.6990)^2 = (0.301)^2$
4	$SQRT(0.301)^2*(0.301)^2*(0.301)^2+(0)^2*(0.6990)^2*(0.6990)^2 = 0.0273$
5	$SQRT(0.301)^2*(0.301)^2+(0)^2*(0.6990)^2 = (0.301)^2$

ตาราง len_b

record_id	len SQRT(SUM (tf **idf * idf))
1	$SQRT(0.301)^2*(0.301)^2+(0.3980)^2*(0)^2 = (0.301)^2$

6.) ตาราง weight จะหาค่าน้ำหนักของคำ จากตาราง token, tf ,idf, len แล้วจัดเก็บในตาราง ซึ่งค่าของ weight หาได้จากสมการ (tf*idf)/len ตามตารางที่ 3.13 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.13 ตารางตัวอย่างที่ใช้เก็บค่าน้ำหนัก

ตาราง weight_a

record_id	token_id	weight(tf*idf)/len
1	1	$(0.301*0.6990) /(0.301)^2$
1	2	$(0.301*0.7782)/ (0.301)^2$
2	1	$(0.301*0.6990)/ (0.301)^2$
2	2	$(0.301*0.4771)/ (0.301)^2$
3	1	$(0.301*0.6990)/ (0.301)^2$
3	2	$(0.301*0.7782)/ (0.301)^2$
4	1	(0.301*0.6990)/ 0.0273
4	2	(0.301*0.7782)/ 0.0273
4	3	(0.301*0.7782)/ 0.0273
5	1	$(0.301*0.4771)/ (0.301)^2$
5	2	$(0.301*0.6990)/ (0.301)^2$

ตาราง weight_b

record_id	token_id	weight(tf*idf)/len
1	1	$(0.301*0.4771) /(0.301)^2$
1	2	$(0.301*0.6990)/ (0.301)^2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.) วิธีหาค่าความเหมือนของข้อความสองชุดข้อมูลจากตาราง weight_a และ weight_b หาได้จากผลรวมค่าน้ำหนักของคำในข้อความชุดที่ a และชุดข้อมูล b ที่เหมือนกัน จากผลการทดลองพบว่า ผลลัพธ์คือข้อความที่ 1 และ 5 ของชุดข้อมูล a เหมือนกับข้อความที่ 1 ของชุดข้อมูล b ตามตารางที่ 3.14 ดังนี้

ตารางที่ 3.14 ตาราง tf-idf ที่ใช้เก็บค่าน้ำหนักความเหมือนของคำ

weight_a. record_id	weight_b record_id	SUM(weight_a.weight * weight_b.weight)	ค่าความเหมือน
1	1	(0 * 0)	0
2	1	(1 * 1) + (0 * 0)	1
3	1	(0 * 0)	0
4	1	(0 * 0)	0
5	1	(0 * 0) + (1 * 1)	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการการทดลอง

ปัญหาพิเศษนี้ได้ศึกษาการทดลองมาตรวัดแบบผสมที่ทำงานร่วมกับ tf-idf ในแต่ละชุดของข้อมูลเพื่อทดสอบประสิทธิภาพว่ามาตรวัดไหนทำงานดีที่สุดในแต่ละชุดข้อมูล

4.1 เครื่องมือที่ใช้ในการทดลอง

เครื่องมือที่ใช้ในการทดลอง มีคุณสมบัติดังต่อไปนี้

4.1.1 ฮาร์ดแวร์

หน่วยประมวลผล (CPU)	: Intel Core 2 Duo 2.1 GHz
หน่วยความจำหลัก (RAM)	: 4.00 GB
หน่วยความจำสำรอง (Hard Disk)	: 320 GB

4.1.2 ซอฟต์แวร์

ระบบปฏิบัติการ : Window 7

โปรแกรม Microsoft SQL Server 2005

โปรแกรม Appserv

โปรแกรม Editplus

โปรแกรม SimMetric จาก http://sourceforge.net/projects/simmetrics

4.2 ชุดข้อมูล

ชุดข้อมูลที่ใช้ทดลองศึกษามาตรวัดความเหมือนข้อความแบบผสม ซึ่งข้อมูลชุดนี้นำมาใช้กับการทดลองมาตรวัดความเหมือนโดยเฉพาะ ซึ่งชุดข้อมูลที่ใช้ทดลองมีดังนี้

ชุดข้อมูลที่ใช้ทดลองประกอบด้วย 3 ชุดข้อมูล ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ชุดข้อมูล Census เป็นข้อมูลสำมะโนประชากรที่เกิดจากการสังเคราะห์ทำให้ข้อความในชุดข้อมูลนี้ประกอบด้วยข้อความที่มีคำที่พบข้อผิดพลาดจำนวนมาก [1]

2. ชุดข้อมูล Restaurant เป็นข้อมูลแนะนำร้านอาหารของ Foldor และZagat

3. ชุดข้อมูล Business เป็นข้อมูลรายชื่อเว็บไซต์จาก hooverweb และ iontech

ตารางที่ 4.1 แสดงจำนวนข้อความ และจำนวนคำที่ใช้ในการทดลอง สามารถดาวน์โหลดได้จาก http://www.cs.utexas.edu/users/ml/riddle/data.html

ตารางที่ 4.1 ชุดข้อมูลที่ใช้ในทดลอง

ชุดข้อมูล	จำนวนข้อความ	จำนวนคำ
Census	841	5,765
Restaurant	863	10,846
Business	2,139	10,526

4.3 การทดลอง

การทดลองเพื่อหามาตรวัดความเหมือนแบบผสม โดยใช้ tf-idf กับมาตรวัดความเหมือนอักขระที่มีอยู่กับข้อมูลแต่ละลักษณะข้อมูล ได้ทำทดลองดัง ต่อไปนี้

- **หามาตรวัดความเหมือนตามสายอักขระร่วมกับมาตรวัดความเหมือนข้อความ tf-idf ที่เหมาะสมสำหรับชุดข้อมูลแต่ละลักษณะของข้อมูล**

ปัญหาพิเศษทดลองวัดความเหมือนของข้อความในแต่ละชุดข้อมูล ด้วยมาตรวัดความเหมือนข้อความ tf-idf ซึ่งหาว่ามาตรวัดตามสายอักขระแบบใดทำงานร่วมกับมาตรวัดความเหมือนข้อความ tf-idf แล้วให้ประสิทธิภาพดีที่สุด ในแต่ละลักษณะของข้อมูลและเพื่อหาว่ามาตราวัดตามสายอักขระแบบใดเหมาะกับชุดข้อมูลในลักษณะใด ซึ่งในแต่ละชุดข้อมูลได้กำหนดค่าความเหมือนไว้ที่ 0.9

4.4 วิธีวัดประสิทธิภาพผลการทดลอง

เมื่อทดลองเปรียบเทียบข้อความด้วยมาตรวัดความเหมือนข้อความที่ให้ค่าความเหมือนมากกว่าเกณฑ์ค่าความเหมือนที่กำหนด ซึ่งจะได้คู่ของข้อความที่เหมือนกัน จากนั้นวัดประสิทธิภาพของผลการทดลองด้วยค่าเฉลี่ยความแม่นยำ มีวิธีทำงานดังต่อไปนี้

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามมิให้ทำซ้ำ ดัดแปลง หรือจำหน่ายเพื่อการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นได้รับอนุญาตเป็นลายลักษณ์อักษรจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีทุกครั้งที่มีการนำไปใช้

1. ข้อความที่วัดความเหมือนได้ถูกต้องจนถึงผิดพลาดมีค่าอยู่ระหว่าง 0 -1

2. หาจำนวนของคู่ข้อความที่วัดได้ถูกต้องแล้วนำมาเปรียบเทียบกับคู่ข้อความที่ได้จากการคิวรี่

3. หาค่าเฉลี่ยคู่ข้อความในแต่ละบรรทัด ซึ่งจะนำผลคูณของคู่ข้อความที่วัดได้ถูกต้องคูณด้วยคู่ข้อความที่วัดได้แล้วหารด้วยจำนวนข้อความในแต่ละบรรทัด

จากนั้นหาค่าเฉลี่ยความแม่นยำของข้อความที่วัดได้ทั้งหมด จากการหาผลรวมแล้วหารด้วยจำนวนข้อความทั้งหมด

$$\text{Avg. Precision} = \frac{1}{m} \sum_{r=1}^{N} \frac{c(i)\, s(i)}{i}$$

รูปที่ 4.1 สมการค่าเฉลี่ยความแม่นยำ

โดยที่ *c(i)* คือ จำนวนที่วัดถูกต้องก่อนข้อความที่ *i*

s(i) จะมีค่าระหว่าง 0 -1 ซึ่ง

ถ้าเข้าใกล้ 1 แสดงว่าข้อความที่ *i* มีการวัดข้อความได้ถูกต้องมาก

ถ้าเข้าใกล้ 0 แสดงว่าข้อความที่ *i* มีการวัดข้อความได้ถูกต้องน้อย

N คือ จำนวนข้อความทั้งหมดที่วัดว่าเหมือนกัน

m คือ จำนวนข้อความทั้งหมดที่วัดถูกต้อง

r คือ ช่วง (rank)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ผลการทดลอง

4.5.1 ผลการทดลองมาตรวัดความเหมือนตามอักขระที่ใช้ร่วมกับมาตรวัดความเหมือนข้อความ tf-idf ที่ชุดข้อมูล Census

จากการศึกษาทดลองมาตรวัดความเหมือนของข้อความ tf-idf ซึ่งใช้มาตรวัดความเหมือนตามสายอักขระ ได้แก่ Levenstein, MongeElkan, JaroWinkler, SmithWatermanGotoh ฯลฯ ผลการทดลองแสดงไว้ดังตารางที่ 4.2 พบว่ามาตรวัดความเหมือนตามสาย SmithWatermanGotoh ให้ประสิทธิภาพการทำงานกับชุดข้อมูลที่มีข้อความเกิดข้อผิดพลาดเป็นจำนวนมากได้ดีที่สุด โดยให้ค่าเฉลี่ยความแม่นยำมากกว่ามาตรวัดความเหมือนแบบสายอักขระอื่นๆ

ตารางที่ 4.2 ผลการทดลองเปรียบเทียบค่าเฉลี่ยความแม่นยำ ที่ค่าความเหมือน 90 % ในชุดข้อมูล Census

มาตรวัดลำดับสอง	ค่าเฉลี่ยความแม่นยำ
BlockDistance	0.00000
ChapmanLengthDeviation	0.06240
ChapmanMeanLength	0.00000
CosineSimilarity	0.00000
DiceSimilarity	0.00000
EuclideanDistance	0.00000
Jaro	0.00000
JaroWinkler	0.00000
Levenstein	0.00000
MatchingCoefficient	0.00000
MongeElkan	0.51765
NeedlemanWunch	0.00000
OverlapCoefficient	0.00000
QGramsDistance	0.00000
Smith Waterman	0.57783
Smith Watermangotoh	0.58293

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 ผลการทดลองมาตรวัดความเหมือนตามอักขระที่ใช้ร่วมกับมาตรวัดความเหมือนข้อความ tf-idf ที่ชุดข้อมูล Restaurant

จากการศึกษาทดลองมาตรวัดความเหมือนของข้อความ tf-idf ซึ่งใช้มาตรวัดความเหมือนตามสายอักขระ ได้แก่ Levenstein, MongeElkan, JaroWinkler, SmithWatermanGotoh ฯลฯ ผลการทดลองแสดงไว้ดังตารางที่ 4.3 พบว่ามาตรวัดความเหมือนตามสายอักขระแบบ ChapmanLengthDeviation ให้ประสิทธิภาพการทำงานได้ดีที่สุด โดยให้ค่าเฉลี่ยความแม่นยำมากกว่ามาตรวัดความเหมือนแบบสายอักขระอื่นๆ

ตารางที่ 4.3 ผลการทดลองเปรียบเทียบค่าเฉลี่ยความแม่นยำ ที่ค่าความเหมือน 90 % ในชุดข้อมูล Restaurant

มาตรวัดลำดับสอง	ค่าเฉลี่ยความแม่นยำ
BlockDistance	0.42000
ChapmanLengthDeviation	0.84571
ChapmanMeanLength	0.00000
CosineSimilarity	0.42000
DiceSimilarity	0.42000
EuclideanDistance	0.42000
JaccardSimilarity	0.42000
Jaro	0.56665
JaroWinkler	0.62094
Levenstein	0.50122
MatchingCoefficient	0.42000
MongeElkan	0.72894
NeedlemanWunch	0.50122
OverlapCoefficient	0.42000
QGramsDistance	0.42000
SmithWaterman	0.72894
SmithWatermanGotoh	0.72894

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.3 ผลการทดลองมาตรวัดความเหมือนตามอักขระที่ใช้ร่วมกับมาตรวัดความเหมือนข้อความ tf-idf ที่ชุดข้อมูล Business

นำมาตรวัดความเหมือนข้อความ ที่ใช้ tf-idf แล้วให้ประสิทธิภาพดีที่สุด โดยใช้มาตรวัดความเหมือนตามสายอักขระ ได้แก่ Levenstein, MongeElkan, JaroWinkler, SmithWatermanGotoh ฯลฯ ผลการทดลองแสดงไว้ดังตารางที่ 4.4 พบว่ามาตรวัดความเหมือนตามสายอักขระแบบ ChapmanLengthDeviation ให้ประสิทธิภาพการทำงานได้ดีที่สุด โดยให้ค่าเฉลี่ยความแม่นยำมากกว่ามาตรวัดความเหมือนแบบสายอักขระอื่นๆ

ตารางที่ 4.4 ผลการทดลองเปรียบเทียบค่าเฉลี่ยความแม่นยำ ที่ค่าความเหมือน 90 % ในชุดข้อมูล Business

มาตรวัดลำดับสอง	ค่าเฉลี่ยความแม่นยำ
BlockDistance	0.42000
ChapmanLengthDeviation	0.84571
ChapmanMeanLength	0.00000
CosineSimilarity	0.42000
DiceSimilarity	0.42000
EuclideanDistance	0.42000
JaccardSimilarity	0.42000
Jaro	0.56666
JaroWinkler	0.62095
Levenstein	0.50122
MatchingCoefficient	0.42000
MongeElkan	0.72894
NeedlemanWunch	0.50122
OverlapCoefficient	0.42000
QGramsDistance	0.42000
SmithWaterman	0.72894
SmithWatermanGotoh	0.72894

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

งานวิจัยนี้เป็นการศึกษามาตรวัดความเหมือนข้อความแบบผสม ซึ่งในการทดลองจะใช้มาตรวัดความเหมือนข้อความ if-idf ผสมกับมาตรวัดแบบอักขระแบบต่าง ๆ เพื่อหาว่ามาตรวัดแบบใดเหมาะสมกับชุดข้อมูลในแต่ละลักษณะ จากการทดลองพบว่า

ชุดข้อมูล Census เป็นข้อมูลสำมะโนประชากรที่เกิดจากการสังเคราะห์ซึ่งจะมีข้อความที่มีคำที่พบข้อผิดพลาดจำนวนมาก เมื่อทดลองใช้มาตรวัดความเหมือนข้อความ if-idf ร่วมกับ SmithWatermanGotoh ให้ความถูกต้องแม่นยำมากกว่ามาตรวัดความเหมือนข้อความตามสายอักขระอื่น ๆ จึงสรุปได้ว่าชุดข้อมูล Census ใช้มาตรวัดความเหมือนข้อความ if-idf ร่วมกับมาตรวัดความ SmithWatermanGotoh ให้ประสิทธิภาพการทำงานดีที่สุด

ชุดข้อมูล Restaurant เป็นข้อมูลแนะนำร้านอาหาร ส่วนชุดข้อมูล Business เป็นข้อมูลรายชื่อเว็บไซต์จาก hooverweb และ iontech พบว่าทั้งสองชุดข้อมูลมีอักษรย่อในข้อความเป็นจำนวนมาก

เมื่อทดลองใช้มาตรวัดความเหมือนข้อความ if-idf ร่วมกับ ChapmanLengthDeviation ให้ความถูกต้องแม่นยำมากกว่ามาตรวัดความเหมือนข้อความตามสายอักขระอื่น ๆ สรุปได้ว่าชุดข้อมูล Restaurant และ Business ใช้มาตรวัดความเหมือนข้อความ if-idf ร่วมกับมาตรวัดตามสายอักขระ ChapmanLengthDeviation ให้ประสิทธิภาพการทำงานดีที่สุด

5.2 ข้อเสนอแนะ

1.จากการทดลองได้ศึกษามาตรวัดความเหมือนแบบผสม ในการทดลองได้ใช้มาตรวัดข้อความแบบ if-idf เป็นเกณฑ์หลัก ซึ่งอาจมีมาตรวัดข้อความความเหมือนแบบอื่น ๆ ที่ทำงานได้มีประสิทธิภาพมากกว่า if-idf

2. การทดลองหามาตรวัดความเหมือนแบบผสม ขึ้นอยู่กับลักษณะชุดข้อมูลที่ใช้ทดลอง เพราะฉะนั้นถ้าชุดข้อมูลเปลี่ยน ทำให้มาตรวัดความเหมือนตามสายอักขระเปลี่ยนด้วย

3. การทดลองได้ปรับค่าความเหมือนของข้อความไว้ที่ 90% นั่นหมายความว่า ยอมให้มีข้อผิดพลาดระหว่างข้อความได้แค่ 10 % ซึ่งการกำหนดเกณฑ์ค่าความเหมือนเป็นค่าประมาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. WilliamW. Cohen, Pradeep Ravikumar , Stephen E. Fienberg . 2003 . A Comparison of String Distance Metrics for Name-Matching Tasks

2. WilliamW. Cohen, Pradeep Ravikumar, Stephen E. Fienberg . 2003 . A Comparison of String Metrics for Matching Names and Records

3. [Online].Available : http://sourceforge.net/projects/secondstring/

4. [Online].Available : http://staffwww.dcs.shef.ac.uk/people/S.Chapman/stringmetrics.html

5. [Online].Available : http://anastasiosyal.com/archive/2009/01/11/18.aspx

6. [Online].Available : http://alias-i.com/lingpipe/docs/api/com/aliasi/spell/JaroWinklerDistance.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้