แบบจำลองการแก้ปัญหาความกำกวมของคำจากคลังข้อความ โดยใช้เทคนิคคำบริบท

Word Sense Disambiguation Model from Corpus Using Context Word Technique

กาญจนา ทองกลิ่น

Kanjana Thongklin

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Prince of Songkla University
2551

ชื่อวิทยานิพนธ์ แบบจำลองการแก้ปัญหาความกำกวมของคำจากคลังข้อความโดยใช้เทคนิคคำบริบท

ผู้เขียน นางสาวกาญจนา ทองกลิ่น

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

…….....
(ดร.วิภาดา เวทย์ประสิทธิ์)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

…….....
(ดร.ศิริรัตน์ วณิชโยบล)

คณะกรรมการสอบ

……...ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.กฤษณะ ชินสาร)

…….....กรรมการ
(ดร.ภราดร ภักดีวานิช)

…….....กรรมการ
(ดร.ศิริรัตน์ วณิชโยบล)

…….....กรรมการ
(ดร.วิภาดา เวทย์ประสิทธิ์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

……
(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)
คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ แบบจำลองการแก้ปัญหาความกำกวมของคำจากคลังข้อความโดยใช้คำบริบท
ผู้เขียน นางสาวกาญจนา ทองกลิ่น
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2550

บทคัดย่อ

การแก้ปัญหาความกำกวมของคำเป็นหนึ่งในงานด้านการประมวลผลภาษาธรรมชาติ วิทยานิพนธ์นี้ได้เสนอแนวคิดใหม่ในการแก้ปัญหาความกำกวมของคำโดยใช้หน้าต่างคำบริบท (Context Window) โดยสร้างแบบจำลองการแก้ปัญหาความกำกวมของคำและการเลือกแอทริบิวต์โดยใช้อัตราส่วนเกนและโครงข่ายประสาทเทียมแบบเรเดียลเบซิสฟังก์ชัน หรือ Word Sense Disambiguation and Attribute Selection (WSD_AS) Using Gain Ratio and RBF Neural Network และพัฒนาโปรแกรมในการแก้ปัญหาความกำกวมของคำตามแบบจำลองดังกล่าว โดยใช้ Visual Basic.Net ทำงานร่วมกับโปรแกรม SenseTools และ NSP ในการสร้างข้อมูลให้อยู่ในรูปแบบ arff และใช้โปรแกรม WEKA แบบ Command Line Interface ในการจำแนกความหมาย ขั้นตอนการทำงานของแบบจำลองการแก้ปัญหาความกำกวมของคำประกอบด้วย 4 ขั้นตอนคือ 1) เตรียมคลังข้อความโดยตัดคำที่เป็น Stoplist ออก 2) สร้างแอทริบิวต์โดยใช้คำบริบททั้งทางซ้ายและขวา 3) เลือกแอทริบิวต์โดยใช้เทคนิค GainRatioAttributeEval และ4) จำแนกความหมายโดยใช้อัลกอริทึม RBF Neural Network ผลการทดลองจากคลังข้อความมาตรฐาน Senseval-2 จากคำกำกวมต่าง ๆโดยเปรียบเทียบกับงานวิจัยอื่น ๆแสดงให้เห็นว่าแบบจำลองที่นำเสนอให้ค่าความถูกต้องสูงที่สุด

Thesis Title	Word Sense Disambiguation Model form Corpus Using Context Word Technique
Author	Miss Kanjana Thongklin
Major Program	Computer Science
Academic Year	2007

ABSTRACT

Word sense disambiguation is one of natural language processing tasks. This thesis proposes new idea for the word sense disambiguation by using context window. The model of Word Sense Disambiguation and Attribute Selection (WSD_AS) Using Gain Ratio and RBF Neural Network has been constructed and developed for the word sense disambiguation. Visual Basic.Net, Sense Tools, and NSP are used for programming in order to arrange the data into the format of arff. Command Line Interface of WEKA is used to classify the word sense. The model of word sense disambiguation composes of 4 steps; step 1) preparing data storage by eliminating stoplist words, step 2) creating attribute using both left-hand and right-hand sides, step 3) selecting attribute by the technique of GainRatioAttributeEval, and step 4) classifying the word sense by using algorithm RBF Neural Network. The experimental result with the Senseval-2 corpus of various ambiguous words when comparing with other studies indicates that the presented model gives highest accuracy.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จลุล่วงได้ด้วยดี ด้วยความช่วยเหลือและสนับสนุนจากบุคคลหลายฝ่าย ซึ่งผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูงคือ

ดร.วิภาดา เวทย์ประสิทธิ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่กรุณาให้คำปรึกษาแนะนำในการทำวิทยานิพนธ์ และช่วยเหลือในการแก้ปัญหาต่างๆ พร้อมทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัย

ดร.ศิริรัตน์ วณิชโยบล อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่กรุณาให้ข้อเสนอแนะต่าง ๆ รวมทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัย

ผศ.ดร.กฤษณะ ชินสาร และ ดร.ภราดร ภักดีวานิช กรรมการสอบวิทยานิพนธ์ที่กรุณาช่วยตรวจทานวิทยานิพนธ์ให้มีความสมบูรณ์

อาจารย์ภาควิชาวิทยากรคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ทุกท่าน ที่ให้ความรู้เพิ่มเติมทางด้านวิชาการ ซึ่งสามารถนำมาใช้ให้เกิดประโยชน์ในวิทยานิพนธ์ได้

เจ้าหน้าที่ภาควิชาวิทยาการคอมพิวเตอร์ และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่านที่ให้ความช่วยเหลือในด้านเอกสาร และการเบิกจ่ายวัสดุต่าง ๆ ที่ใช้ในงานวิจัย

เพื่อน ๆ และพี่ ๆ น้องๆ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ ที่เป็นกำลังใจ และช่วยเหลือในอุปสรรคต่างๆ และให้คำปรึกษาเพิ่มเติมในการทำวิทยานิพนธ์

คุณพ่อ คุณแม่ และน้อง ที่เป็นกำลังใจและให้การสนับสนุนในการทำวิทยานิพนธ์ของผู้วิจัยเสมอมา

ผู้วิจัยขอขอบคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้

กาญจนา ทองกลิ่น

สารบัญ

หน้า

สารบัญ (6)

รายการตาราง..... (8)

รายการภาพประกอบ..... (10)

บทที่ 1 บทนำ

1.1 การตรวจเอกสาร

1.1.1 คำบริบท..... 2

1.1.2 การเลือกแอทริบิวต์..... 3

1.1.3 โครงข่ายประสาทเทียม..... 5

1.1.4 ต้นไม้ตัดสินใจ..... 6

1.1.5 คลังข้อความ..... 8

1.2 วัตถุประสงค์ของโครงการ..... 9

1.3 ขอบเขตของการดำเนินงาน..... 9

1.4 ขั้นตอนและระยะเวลาการดำเนินงาน

1.4.1 ขั้นตอนการดำเนินงาน..... 9

1.4.2 ระยะเวลาดำเนินการ..... 10

1.4.3 แผนการดำเนินการวิจัย..... 11

1.5 สถานที่และเครื่องมือที่ใช้

1.5.1 สถานที่..... 11

1.5.2 เครื่องมือที่ใช้..... 12

1.6 ประโยชน์ที่คาดว่าจะได้รับ..... 12

บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับการแก้ปัญหาความกำกวม

2.1 คำกำกวม..... 13

2.2 คลังข้อความ Senseval-2..... 14

2.3 การตัดคำที่เป็น Stoplist..... 18

2.4 โปรแกรม NSP..... 20

2.5 โปรแกรม SenseTools..... 20

2.6 คำบริบท..... 21

2.7 การเลือกแอทริบิวต์

สารบัญ (ต่อ)

หน้า

2.7.1 Information Gain Attribute Evaluation…… 21

2.7.2 Gain Ratio Attribute Evaluation……..……..…….. 22

2.8 การจำแนก

2.8.1 Neural Network……..……..……..…… 23

2.8.2 Decision Tree……..……..…… 25

2.9 Cross Validation……..……..……... 28

บทที่ 3 แบบจำลองการแก้ปัญหาความกำกวมของคำจากคลังข้อความโดยใช้คำบริบท..... 30

บทที่ 4 โปรแกรมการแก้ปัญหาความกำกวมของคำจากคลังข้อความโดยใช้เทคนิคคำบริบท

4.1 ผังการทำงานของโปรแกรม..... 44

4.2 ส่วนประกอบของโปรแกรม..... 48

4.3 ผลการทำงานของโปรแกรม..... 50

4.4 เครื่องมือที่ใช้ในการพัฒนาโปรแกรม..... 53

บทที่ 5 ผลการทดลองและวิจารณ์

5.1 ตัวอย่างคำกำกวม art……... 58

5.2 ตัวอย่างคำกำกวม dyke……... 73

5.3 คำกำกวม bum และ church..... 86

5.4 เปรียบเทียบผลการทดลองและวิจารณ์ผลการทดลอง..... 91

บทที่ 6 บทสรุปและข้อเสนอแนะ

6.1 สรุปผลการวิจัย..... 97

6.2 ปัญหาและอุปสรรค..... 99

6.3 ข้อเสนอแนะ..... 99

บรรณานุกรม..... 100

ภาคผนวก

ก การใช้งาน Command Line Interface ใน WEKA…….. 105

ข ผลงานวิจัยตีพิมพ์ NCCIT'07 ……..... 114

ค ผลงานวิจัยตีพิมพ์ RIVF'08 ……... 121

ประวัติผู้เขียน..... 128

รายการตาราง

ตาราง		หน้า
1.1	ระยะเวลาดำเนินการวิจัย.....	11
2.1	ตัวอย่างคำกำกวม.....	13
2.2	ss_type.....	15
2.3	Lexicographer File.....	15
2.4	ตัวอย่างความหมายของคำกำกวม art.....	18
3.1	จำแนกความหมายโดยแบ่งกลุ่มเป็น 2 ความหมายและแบ่งกลุ่มตามจำนวนความหมายทั้งหมด.....	41
5.1	ผลการทดลองเปรียบเทียบการใช้การตัดคำและไม่ตัดคำในแต่ละอัลกอริทึม อัลกอริทึม IBk กำหนดค่า k ให้มีค่าเท่ากับ 1.....	56
5.2	ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางซ้ายที่ขนาดหน้าต่าง 1 2 3 4 และ 5.....	64
5.3	ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางขวาที่ขนาดหน้าต่าง 1 2 3 4 และ 5.....	65
5.4	ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททั้งทางซ้ายและขวา ที่ขนาดหน้าต่าง 1 2 3 4 และ 5.....	66
5.5	ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางขวาทางซ้าย และทางซ้ายและขวา เมื่อขนาดหน้าต่างเท่ากับ 4.....	67
5.6	ตารางแสดงค่าความถูกต้องการจำแนกความหมายเมื่อกรองแอทริบิวต์ให้มีจำนวนต่างๆ และไม่กรองแอทริบิวต์.....	69
5.7	ตารางแสดงค่าความถูกต้องของการจำแนกความหมายเมื่อเปรียบเทียบการกรองแบบ InfoGainAttributeEval และ GainRatioAttributeEval.....	70
5.8	ตารางแสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3 และ RBFNetwork	71
5.9	ตารางแสดงค่าความถูกต้อง เปรียบเทียบการแบ่งกลุ่มแบบ 2 ความหมายและความหมายทั้งหมด กรองแบบ GainRatioAttributeEval และใช้ลกอริทึม RBFNetwork.....	72
5.10	ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางซ้ายที่ขนาดหน้าต่าง 1 2 3 4 และ 5.....	79
5.11	ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางขวาที่ขนาดหน้าต่าง 1 2 3 4 และ 5.....	80

รายการตาราง (ต่อ)

ตาราง		หน้า
5.12	ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททั้งทางซ้ายและขวา ที่ขนาดหน้าต่าง 1 2 3 4 และ 5……	81
5.13	ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางขวา ทางซ้าย และทางซ้ายและขวา เมื่อขนาดหน้าต่างเท่ากับ 4……..	82
5.14	ตารางแสดงค่าความถูกต้องการจำแนกความหมายเมื่อกรองแอทริบิวต์ให้มีจำนวนต่างๆ และไม่กรองแอทริบิวต์……..	84
5.15	ตารางแสดงค่าความถูกต้องของการจำแนกความหมายเมื่อเปรียบเทียบการกรองแบบ InfoGainAttributeEval และ GainRatioAttributeEval……	85
5.16	ตารางแสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3 และ BFNetwork	85
5.17	แสดงค่าความถูกต้องของคำกำกวมทั้งหมดเมื่อใช้ขนาดหน้าต่างแบบต่างๆ	92
5.18	แสดงคำกำกวมทั้งหมดโดยใช้เทคนิคคำบริบทเมื่อจำนวนแอทริบิวต์เท่ากับ 40……..	93
5.19	ผลการทดลองเมื่อเปรียบเทียบการแก้ปัญหาความกำกวมโดยใช้คำบริบทกับวิธีอื่นๆโดยใช้คลังข้อความ Senseval-2……	96

รายการภาพประกอบ

ภาพประกอบ		หน้า
1.1	โครงสร้างของ Feedforward Network.....	6
1.2	โครงสร้างของ Recurent Network.....	6
1.3	ตัวอย่างต้นไม้ตัดสินใจ.....	7
2.1	ตัวอย่างรูปแบบของคลังข้อความ Senseval-2.....	14
2.2	Stoplist.....	18
2.3	โครงสร้างของโครงข่ายประสาทเทียม.....	23
2.4	การสอนโครงข่ายประสาทเทียม.....	24
2.5	ตัวอย่างข้อมูลอากาศแยกตามแอทริบิวต์.....	27
2.6	ต้นไม้ตัดสินใจของข้อมูลอากาศ.....	28
2.7	K-folds Cross Validation.....	29
3.1	รายละเอียดแบบจำลองการแก้ปัญหาความกำกวมของคำโดยใช้คำบริบท.....	30
3.2	คลังข้อความภาษาอังกฤษของ Senseval-2.....	32
3.3	ไฟล์นามสกุล .xml.....	33
3.4	ไฟล์นามสกุล .count.....	34
3.5	ตัวอย่าง TAG_FILE	35
3.6	ตัวอย่าง INSTANCE_FILE	35
3.7	รูปแบบของคำบริบท 3 แบบ.....	36
3.8	ตัวอย่างคำหลังจากตัดประโยคให้มีคำบริบท 3 แบบ.....	36
3.9	ตัวอย่างคำที่สร้างเป็น 1-gram (output.txt : ไฟล์เอาท์พุดของ count.pl).....	37
3.10	Regular Expression (regex.txt: ไฟล์เอาท์พุทของ nsp2regex.pl).....	38
3.11	รูปแบบของข้อมูลที่แปลงเป็น Feature Vector (art.n.xml.arff: ไฟล์เอาท์พุทของ xml2arff.pl).....	39
3.12	การเปลี่ยนสัญลักษณ์% ให้เป็น ~ (art.arff: ไฟล์เอาท์พุทของ tilde.pl).....	40
3.13	Confusion Matrix.....	42
3.14	การคำนวณค่า Accuracy และ Confusion Matrix ที่จำแนกถูก 100%.....	43
4.1	ผังการทำงานของโปรแกรม.....	45
4.2	ผังการทำงานของโปรแกรม Step1: Data preprocessing.....	46
4.3	ผังการทำงานของโปรแกรม Step2: Create Attribute Using Context Window.....	46

รายการภาพประกอบ (ต่อ)

ภาพประกอบ		หน้า
4.4	ผังการทำงานของโปรแกรม Step3: Attribute Selection……	47
4.5	ผังการทำงานของโปรแกรม Step4: Word Sense Disambiguation.....	47
4.6	หน้าจอหลักของโปรแกรมการแก้ปัญหาความกำกวมของคำ……	48
4.7	หน้าจอการทำงานของโปรแกรมการแก้ปัญหาความกำกวมของคำ……	49
4.8	ตัวอย่างการเลือกคำกำกวมที่ต้องการ……	50
4.9	ตัวอย่างขั้นตอนที่ 1 การเตรียมข้อมูล……	50
4.10	ตัวอย่างขั้นตอนที่ 2 การสร้างแอทริบิวต์……....	51
4.11	ตัวอย่างขั้นตอนที่ 3 การเลือกแอทริบิวต์……....	51
4.12	ตัวอย่างขั้นตอนที่ 4 การจำแนกความหมาย……	51
4.13	ตัวอย่างการแสดงผลลัพธ์ของการจำแนกความหมาย……..	52
4.14	การตรวจสอบความหมายของคำกำกวม……	52
5.1	แสดงค่าความถูกต้องเมื่อใช้อัลกอริทึม IBk เมื่อค่า k มีค่าต่างกัน……....	55
5.2	แสดงค่าความถูกต้องเมื่อใช้อัลกอริทึม IBk เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบปกติและแบบตัดคำ……	56
5.3	แสดงค่าความถูกต้องเมื่อใช้อัลกอริทึม ID3 เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบปกติและแบบตัดคำ……	57
5.4	แสดงค่าความถูกต้องเมื่อใช้อัลกอริทึม NaiveBayes เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบปกติและแบบตัดคำ ……...	57
5.5	คำบริบททางซ้าย…….	58
5.6	คำบริบททางขวา……	58
5.7	คำบริบททั้งทางซ้ายและขวา……...	59
5.8	สร้างแอทริบิวต์โดยใช้คำบริบททางซ้ายและขวา……..	59
5.9	กรองแอทริบิวต์ให้มีจำนวน 30 แอทริบิวต์……	62
5.10	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่าง ๆ โดยใช้คำบริบททางซ้าย……..	64
5.11	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่าง ๆ โดยใช้คำบริบททางขวา……..	65
5.12	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่าง ๆ โดยใช้คำบริบททั้งทางซ้ายและขวา……	66

รายการภาพประกอบ (ต่อ)

ภาพประกอบ		หน้า
5.13	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อใช้คำบริบททางขวา ทางซ้าย และทั้งทางซ้ายและขวา……..	68
5.14	แสดงค่าความถูกต้อง เปรียบเทียบการกรองแอทริบิวต์จำนวนต่าง ๆและไม่กรองแอทริบิวต์ (ค่าสุดท้าย) โดยใช้อัลกอริทึม RBFNetwork กรองแบบ GainRatioAttributeEval……..	69
5.15	แสดงค่าความถูกต้อง เปรียบเทียบการกรองแบบ InfoGainAttributeEval และ GainRatioAttributeEval.....	70
5.16	แสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3 และ RBFNetwork……....	71
5.17	กราฟแสดงค่าความถูกต้องเปรียบเทียบการแบ่งกลุ่มแบบ 2 ความหมายและความหมายทั้งหมด กรองแบบ GainRatioAttributeEval ใช้อัลกอริทึม RBFNetwork.....	72
5.18	คำบริบททางซ้าย…….	73
5.19	คำบริบททางขวา……	73
5.20	คำบริบททั้งทางซ้ายและขวา……..	74
5.21	สร้างแอทริบิวต์โดยใช้คำบริบททางซ้ายและขวา……..	74
5.22	กรองแอทริบิวต์ให้มีจำนวน 30 แอทริบิวต์……	77
5.23	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่าง ๆ. โดยใช้คำบริบททางซ้าย……..	79
5.24	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่าง ๆ โดยใช้คำบริบททางขวา …….	80
5.25	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่าง ๆ โดยใช้คำบริบททั้งทางซ้ายและขวา……	81
5.26	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อใช้คำบริบททางขวา ทางซ้าย และทั้งทางซ้ายและขวาเมื่อขนาดหน้าต่างความกว้างเท่ากับ 4 ……..	83
5.27	แสดงค่าความถูกต้อง เปรียบเทียบการกรองแอทริบิวต์จำนวนต่าง ๆและไม่กรองแอทริบิวต์ (ค่าสุดท้าย) โดยใช้อัลกอริทึม RBFNetwork กรองแบบ GainRatioAttributeEval……..	84
5.28	แสดงค่าความถูกต้อง เปรียบเทียบการกรองแบบ InfoGainAttributeEval และ GainRatioAttributeEval……	85

รายการภาพประกอบ (ต่อ)

ภาพประกอบ		หน้า
5.29	แสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3 และ RBFNetwork ……..	86
5.30	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่าง ๆ โดยใช้คำบริบททั้งทางซ้ายและขวา……	87
5.31	กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อใช้คำบริบททางขวา ทางซ้าย และทั้งทางซ้ายและขวาเมื่อขนาดหน้าต่างความกว้างเท่ากับ 4……....	88
5.32	แสดงค่าความถูกต้อง เปรียบเทียบการกรองแอทริบิวต์จำนวนต่าง ๆและไม่กรองแอทริบิวต์ (ค่าสุดท้าย) โดยใช้อัลกอริทึม RBFNetwork กรองแบบ GainRatioAttributeEval……	89
5.33	แสดงค่าความถูกต้อง เปรียบเทียบการกรองแบบ InfoGainAttributeEval และ GainRatioAttributeEval……..	90
5.34	แสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3 และ RBFNetwork……....	90
5.35	กราฟแสดงค่าความถูกต้องเปรียบเทียบการแบ่งกลุ่มแบบ 2 ความหมายและความหมายทั้งหมด กรองแบบ GainRatioAttributeEval ใช้อัลกอริทึม RBFNetwork……	91

บทที่ 1

บทนำ

งานทางด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing) ในปัจจุบันเช่น การแปลภาษา การค้นคืนเอกสาร นับเป็นงานวิจัยที่ได้รับความสนใจเป็นอย่างยิ่ง การพัฒนาให้งานเหล่านั้นมีความสมบูรณ์และถูกต้องมากขึ้นจะต้องแก้ปัญหาหลายอย่าง ความกำกวมของคำเป็นปัญหาหนึ่งที่สำคัญเพราะคำที่มีความหมายหลายความหมายหรือที่เรียกว่าคำกำกวม (Ambiguous Word) สามารถวิเคราะห์ได้หลายความหมาย เมื่อระบบเข้าใจความหมายผิดจะเป็นอุปสรรคและทำให้เกิดความผิดพลาดในงานได้ ตัวอย่างเช่น “I walked to the bank.” คำว่า “bank” เป็นคำกำกวมอาจหมายถึง ธนาคาร หรือ ตลิ่ง เป็นต้น การแก้ปัญหาความกำกวมของคำ (Word Sense Disambiguation) จึงเป็นงานที่ได้รับความสนใจเป็นอย่างมากและแต่ละงานวิจัยได้พัฒนาโดยใช้เทคนิคที่แตกต่างกัน โดยพิจารณาคำบริบท การเลือกแอทริบิวต์ เทคนิคการเรียนรู้ของเครื่องและคลังข้อความ

คำบริบท (Context) เป็นหลักการที่สำคัญในการหาความหมายที่ถูกต้องของคำ ในการเข้าใจความหมายของคำหนึ่งคำจำเป็นจะต้องรู้ว่าคำบริบทของคำนั้นเป็นอย่างไร มีความหมายไปในทิศทางใดจึงทำให้สามารถหาความหมายของคำนั้นได้ ตัวอย่างเช่น “I walked to the bank, the water looked inviting.” จากคำบริบทของคำกำกวม “bank” คือ “water” ทำให้ทราบว่าคำกำกวมนี้ หมายถึง ตลิ่ง ไม่ใช่ ธนาคาร คำบริบทมีความสำคัญกับงานทางด้านการประมวลผลภาษาธรรมชาติหลายด้านเช่น การค้นคืนเอกสาร การแปลภาษา การสรุปความ การจัดประเภทเอกสาร งานวิจัยที่ใช้คำบริบทในการแก้ปัญหาความกำกวมเช่นระบุขนาดหน้าต่างในการทดลองซึ่งขนาดหน้าต่างและลักษณะของคำบริบทที่เลือกมาใช้ในการทดลองเลือกตามความเหมาะสม เช่น การใช้คำบริบททางซ้าย การใช้บริบททางขวา หรือการใช้บริบททางซ้ายและขวา การใช้ขนาดหน้าต่างที่แตกต่างกัน

การเลือกแอทริบิวต์ (Attribute Selection) หรือการกรองแอทริบิวต์เป็นเทคนิคหนึ่งในงานด้านการทำเหมืองข้อมูลเนื่องจากปัญหาของจำนวนแอทริบิวต์มีปริมาณมากเกินไป ทำให้งานมีประสิทธิภาพน้อยลงเช่น เมื่อใช้ในการจำแนกประเภทกับงานด้านการทำเหมืองข้อมูล หากจำนวนแอทริบิวต์มากเกินไปและมีแอทริบิวต์ที่ไม่เกี่ยวข้องเป็นจำนวนมากอาจทำให้ผลลัพธ์ในการจำแนกประเภทนั้นไม่ตรงกับความต้องการ ในการลดจำนวนแอทริบิวต์ที่ไม่เกี่ยวข้องออกทำให้เหลือเฉพาะแอทริบิวต์ที่มีความเกี่ยวข้องกันเท่านั้น ข้อดีของการลดจำนวนแอทริบิวต์คือใช้ตัวอย่างที่มีความเกี่ยวข้องเท่านั้นมาสอน ในการประมวลผลภาษาธรรมชาติเนื่องจากต้องใช้คำมาสร้างเป็นแอทริบิวต์ดังนั้นจึงทำให้มีแอทริบิวต์เป็นจำนวนมาก การลดจำนวนแอทริบิวต์ที่ไม่จำเป็นออกทำให้ประสิทธิภาพในการทำงานดีขึ้น

เทคนิคการเรียนรู้ของเครื่อง (Machine Learning) เป็นเทคนิคที่ใช้ในงานประยุกต์หลากหลาย ในงานวิจัยด้านการแก้ปัญหาความกำกวมของคำใช้เทคนิคการเรียนรู้ของเครื่องเป็นแนวทางในการนำมาใช้ในการจำแนกความหมาย (Classification) เพื่อเพิ่มความถูกต้องและความแม่นยำในการจำแนกความหมาย โดยใช้เทคนิคการเรียนรู้แบบใช้ตัวอย่างสอน (Supervised Learning) เช่น โครงข่ายประสาทเทียม (Neural Networks) และการใช้ต้นไม้ตัดสินใจ (Decision Tree)

คลังข้อความภาษา (Corpus) หมายถึงข้อมูลภาษาที่ได้มีการเก็บบันทึกไว้ในระบบคอมพิวเตอร์ โดยจะให้ข้อมูลในการใช้ภาษาที่เกิดขึ้นจริง การใช้คลังข้อความในงานประยุกต์ต่างๆ เช่น การเรียนการสอนภาษา การวิจัยทางภาษาศาสตร์ การแปลภาษาและการสอนภาษา การประมวลผลภาษาธรรมชาติ ในงานวิจัยด้านการแก้ปัญหาความกำกวมของคำใช้คลังข้อความ Senseval-2 เป็นคลังข้อความมาตรฐานในการวัดประสิทธิภาพของการแก้ปัญหาความกำกวมของคำ ประกอบด้วยคำกำกวมหลายคำ แต่ละคำมีตัวอย่างประโยคจำนวนมากและมีการกำหนดความหมายของคำกำกวมในแต่ละประโยคไว้เพื่อใช้ในการจำแนกความหมายได้ถูกต้อง

วิทยานิพนธ์นี้ใช้เทคนิคของคำบริบทและกรองแอทริบิวต์ในการสร้างแบบจำลองแก้ปัญหาความกำกวมของคำ โดยพิจารณาขนาดหน้าต่างคำบริบทและการกรองแอทริบิวต์ที่ไม่เกี่ยวข้องออกและทดสอบกับคลังข้อความ Senseval-2 (Pedersen, 2001: Online) ซึ่งเป็นคลังข้อความมาตรฐานในการวิเคราะห์ประสิทธิภาพของการแก้ปัญหาความกำกวมมาใช้ในการแก้ปัญหาความกำกวมของคำ

1.1 การตรวจเอกสาร

หลักการและเทคนิคที่ใช้ในการแก้ปัญหาความกำกวมของคำคือ คำบริบท (Context) การเลือกแอทริบิวต์ (Attribute Selection) โครงข่ายประสาทเทียม (Neural Network) ต้นไม้ตัดสินใจ (Decision Tree) และคลังข้อความ Senseval-2

1.1.1 คำบริบท (Context)

คำบริบท (Context) คือคำแวดล้อมที่อยู่รอบๆคำที่มีความหมายกำกวมโดยคำบริบทแต่ละคำจะเป็นคำที่กล่าวถึงสิ่งต่างๆในลักษณะที่มีความสัมพันธ์กันทางความหมายในประโยคนั้นๆ ในการนำคำบริบทมาใช้ในการแก้ปัญหาความกำกวมของคำจะทำให้ทราบว่าความหมายของคำที่กำกวมนั้นควรมีความหมายไปในทางใดและสามารถกำหนดความหมายของคำกำกวมได้อย่างถูกต้อง ซึ่งสมารถนำไปประยุกต์ใช้กับงานด้านภาษาธรรมชาติ

หลากหลายเช่น การแปลภาษา (Vickrey *et al*., 2005) งานวิจัยที่ได้นำคำบริบทมาใช้ในการแก้ปัญหาความกำกวมของคำจะมีรูปแบบการแทนคำบริบทที่ต่างกันออกไปเช่น การใช้คำบริบทร่วมกับลักษณะอื่นๆอาจเป็น ตำแหน่งของคำบริบท ตำแหน่งของคำกำกวม หรือหน้าที่ของคำ (Part-of-speech: POS) เช่น ประโยค “I went to the bank.” คำกำกวม bank พิจารณาร่วมกับตำแหน่งคำที่ 5 และชนิดของคำคือ คำนาม เมื่อนำไปเป็นแอทริบิวต์ จะได้แอทริบิวต์ที่เพิ่มขึ้นอีก 2 แอทริบิวต์คือ แอทริบิวต์ตำแหน่ง แอทริบิวต์ชนิดของคำ เป็นต้น งานวิจัยที่ใช้หลักการนี้คือ การใช้บริบททั้งทางซ้ายและขวาขนาดหน้าต่าง ± 2 สำหรับแบบจำลองการใช้ค่าเอนโทรปีสูงสุด (Maximum Entropy Model) (Chao and Dyer, 2002) และใช้บริบททั้งทางซ้ายและขวาขนาดหน้าต่าง ± 1 ± 2 ± 3 สำหรับการเลือกลักษณะที่เหมาะสมที่สุดให้กับคำกำกวม (Feature Selection for Maximum Entropy-based Model) (Suarez and Palomar, 2002) การใช้บริบททั้งทางซ้ายและขวาขนาดหน้าต่าง ± 50 สำหรับการใช้ค่าน้ำหนักให้กับตัวจำแนก (Weighted Combination of Classifiers) (Anh *et al*., 2005) ข้อดีของวิธีนี้คือจะได้ลักษณะในการจำแนกความหมายเพิ่มขึ้นและดีขึ้น และอีกรูปแบบคือการใช้คำบริบทร่วมกับการใช้คำที่มีความสัมพันธ์กัน คล้ายกัน เช่น คำว่า money อาจใช้คำอื่นที่มีความหมายใกล้เคียงกันคือ exchange coin หรือใช้การปรากฏร่วมของคำ (Co-occurrence) เช่น คำว่า money และ payment งานวิจัยที่ใช้หลักการนี้คือ การใช้ความคล้ายกันของคำสองคำ (Yoon *et al*., 2006) การคำนวณความคล้ายกันของคำในบริบท (Casado *et al*., 2005) และใช้การปรากฏร่วมของคำบริบท (Oh and Choi, 2002) และสำหรับการใช้คลาส (Class-based Collocations Model) (O’Hara *et al*., 2004) เป็นต้น ข้อดีของวิธีนี้คือค่าความถูกต้องจะเพิ่มขึ้นเนื่องจากการใช้ลักษณะที่หลากหลายสามารถรองรับคำที่อาจไม่ใช่คำเดียวกันแต่มีความคล้ายกันได้ แต่ทั้งสองวิธีนี้อาจมีข้อเสียคือเมื่อใช้คำบริบทร่วมกับลักษณะต่างๆ ลักษณะที่เพิ่มขึ้นนั้นเมื่อนำไปเป็นแอทริบิวต์ทำให้จำนวนแอทริบิวต์เพิ่มขึ้นอาจมีบางแอทริบิวต์ที่ไม่เกี่ยวข้องซึ่งมีผลต่อประสิทธิภาพในการจำแนกความหมาย

1.1.2 การเลือกแอทริบิวต์ (Attribute Selection)

Attribute Selection หรือ Feature Selection เป็นการลดแอทริบิวต์ที่ซ้ำซ้อนหรือไม่เกี่ยวข้องออกให้เหลือเฉพาะแอทริบิวต์ที่มีความเกี่ยวข้องกันหรือมีความสัมพันธ์กัน ข้อดีของการลดแอทริบิวต์คือทำให้ประมวลผลรวดเร็วและได้ค่าความถูกต้องสูงขึ้น งานวิจัยหลายด้านนำเทคนิคนี้ไปประยุกต์ใช้ เช่น การเปรียบเทียบการกรองแอทริบิวต์แต่ละวิธีโดยใช้ฐานข้อมูล UCI ซึ่งประกอบด้วยฐานข้อมูลย่อย 15 ฐานข้อมูล (Hall and Holmes, 2003) การเลือกแอทริบิวต์ของการวิเคราะห์ความสัมพันธ์ร่วมของตัวแปรทางเคมี (Okada, 2005) และการวิเคราะห์ตัวแปรของผลิตภัณฑ์เพื่อเป็นปัจจัยทางเศรษฐกิจ (Flores *et al*., 2008) เป็นต้น การเลือกแอทริบิวต์สามารถแบ่งได้เป็น 2 ประเภทคือ การเลือกแอทริบิวต์โดยการประเมินค่าให้กับ

แอทริบิวแต่ละแอทริบิวต์ และการเลือกแอทริบิวต์โดยการประเมินค่าให้กับสับเซตของแอทริบิวต์โดยมีรายละเอียดดังนี้

1) Single-attribute Evaluators เป็นการประเมินค่าให้กับแอทริบิวต์แต่ละแอทริบิวต์โดยใช้ Ranker Search ค่าของแอทริบิวต์ที่ได้จะเรียงลำดับจากมากไปหาน้อย ข้อดีของวิธีนี้คือสามารถระบุจำนวนแอทริบิวต์ที่ต้องการได้ โดยตัดจำนวนแอทริบิวต์ที่เหลือทิ้งไป ตัวอย่างเทคนิคของการเลือกแอทริบิวต์ประเภทนี้ เช่น

- GainRatioAttributeEval จะประเมินค่าของแอทริบิวต์โดยวัด Gain Ratio ให้กับคลาสนั้นๆ วิธีนี้เป็นวิธีที่ง่ายและรวดเร็วมาก งานวิจัยที่ใช้หลักการนี้เช่น การระบุเสียงพูดใช้ฐานข้อมูลเสียง 2001 NIST SRE (Ganchev *et al*., 2006)

- InfoGainAttributeEval จะใช้การประเมินค่าของแอทริบิวต์โดยวัด Information Gain วิธีนี้เป็นวิธีที่ง่ายและรวดเร็ว งานวิจัยที่ใช้หลักการนี้เช่น งานวิจัยด้านการทำเหมืองข้อมูลโดยฐานขอมูล UCI ประกอบด้วยฐานข้อมูลย่อย 9 ฐานข้อมูล (Huang *et al*., 2004)

- ChiSquaredAttributeEval จะประเมินค่าแอทริบิวต์โดยคำนวณค่า Chi-Square ทางสถิติ งานวิจัยที่ใช้หลักการนี้เช่น งานวิจัยด้านชีวสารสนเทศ (Koh and Wong, 2007)

- ReliefFAttributeEval ใช้การประเมินค่าความแตกต่างของแอทริบิวต์กับตัวอย่างใกล้เคียง (K Nearest Neighbours) ในคลาสเดียวกันหรือต่างคลาสกันจำนวน K ตัว ถ่าค่าความแตกต่างเป็น 1 แสดงว่ามีคววามแตกต่างกันมาก ถ้าค่าที่ได้เป็น 0 จะมีความเหมือนกันมาก งานวิจัยที่ใช้หลักการนี้เช่น งานวิจัยด้านการทำเหมืองข้อมูล (Huang *et al*., 2004 ; Symeonidis *et al*., 2007)

2) Attribute Subset Evaluators เป็นการประเมินค่าสับเซตของแอทริบิวต์โดยการประเมินค่าให้กับแอทริบิวต์แล้วนำแอทริบิวต์อื่น (Attribute Subset) เข้ามาประเมินร่วมกัน วิธีการนี้ไม่สามารถเรียงลำดับของแอทริบิวต์ได้ (Ranker) เนื่องจากไม่ได้ประเมินแอทริบิวต์เพียงแอทริบิวต์เดียว และไม่สามารถเลือกจำนวนแอทริบิวต์ที่ต้องการได้เหมือนกับ Single-attribute Evaluators ตัวอย่างเทคนิคของการเลือกแอทริบิวต์ประเภทนี้ เช่น

- WrapperSubsetEval ใช้การประเมินเซตของแอทริบิวต์โดยใช้ K-folds Cross-Validation กับตัวจำแนกประเภท (Classifier) ดังนั้นวิธีการนี้ทำให้ใช้เวลาในการทำงานจนกว่าจะครบจำนวน folds งานวิจัยที่ใช้หลักการนี้เช่น งานวิจัยด้านการทำเหมืองข้อมูลโดยใช้ฐานข้อมูลจาก MLC++ Machine Learning Laboratory ประกอบด้วย 8 ชุดข้อมูล เช่นข้อมูลบัตรเครดิต (Dong and Kothari, 2003)

- ConsistencySubsetEval ประเมินเซตของแอทริบิวต์โดยใช้ระดับความเข้ากันได้ของเซตย่อยของแอทริบิวต์ งานวิจัยที่ใช้หลักการนี้เช่น งานวิจัยด้านการรักษาผู้ป่วย (Borges and Nievola, 2005)

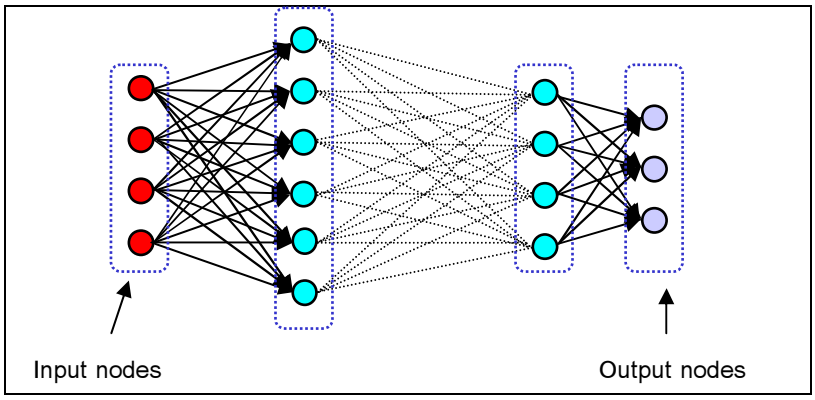
- CfsSubsetEval ใช้การประเมินความสัมพันธ์ร่วมระหว่างเซตของแอทริบิวต์กับคลาสเช่น งานวิจัยที่ใช้การทำเหมืองข้อมูลด้านการจัดการห่วงโซ่อุปทาน (Symeonidis *et al*., 2007) การจำแนกประเภทในงานด้านชีวสารสนเทศ (Kouskoumvekaki *et al*., 2008)

1.1.3 โครงข่ายประสาทเทียม (Neural Network)

โครงข่ายประสาทเทียม (Neural Network) เป็นการเลียนแบบการทำงานของสมองมนุษย์ ที่ประกอบไปด้วยเซลล์พิเศษมากมายที่เรียกว่าเซลล์ประสาท (Neuron) การเลียนแบบการทำงานของสมองมนุษย์ของเครื่องคอมพิวเตอร์เริ่มจากการกำหนดให้แต่ละโหนด (Node) เปรียบเสมือนเป็นเซลล์ประสาท และสร้างการเชื่อมต่อให้กับโหนดเหล่านั้นให้เป็นโครงข่าย (Network) แต่ละโครงข่ายจะประกอบไปด้วยโหนดที่ถูกจัดแบ่งเป็นชั้นๆ เรียกว่าเลเยอร์ (Layer) แต่ละเลเยอร์จะมีหน้าที่การทำงานแตกต่างกัน พื้นฐานที่สำคัญของโครงข่ายประสาทเทียมประกอบไปด้วย 3 ชั้น หรือ 3 เลเยอร์ ได้แก่ ชั้นข้อมูลเข้า (Input Layer) ที่ถูกเชื่อมต่อกับชั้นซ่อน (Hidden Layer) ซึ่งเชื่อมต่อกับชั้นผลลัพธ์ (Output Layer) Input Unit จะทำหน้าที่แทนส่วนของข้อมูลดิบ ที่จะถูกป้อนเข้าสู่เครือข่าย Hidden Unit จะถูกกำหนดโดยการทำงานของ Input Unit และค่าน้ำหนักบนความสัมพันธ์ระหว่าง Input Unit และ Hidden Unit และการทำงานของ Output Unit จะขึ้นอยู่กับการทำงานของ Hidden Unit และค่าน้ำหนักระหว่าง Hidden Unit และ Output Unit การประยุกต์ใช้ข่ายงานระบบประสาทจึงเป็นแนวทางซึ่งมีผู้นำมาประยุกต์ใช้งานหลายประเภท งานวิจัยที่ใช้โครงข่ายประสาทเทียมมาใช้ในด้านการแก้ปัญหาความกำกวมของคำเช่น การแก้ปัญหาความกำกวมของคำในภาษาจีนโดยใช้คำบริบทและใช้เทคนิค Back Propagation Neural Network (Liu *et al*., 2005) การแก้ปัญหาความกำกวมของคำโดยใช้การจัดกลุ่มของคลาส (Legrand and Pulido, 2004) และงานทางด้านอื่นได้แก่ งานการจดจำรูปแบบ ลายมือ ลายเซนต์ ตัวอักษร (Sae-Tang and Methaste, 2002) งานทำนาย เช่น พยากรณ์อากาศ (Wettayaprasit and Nanakorn, 2006; Wettayaprasit *et al*., 2007) พยากรณ์หุ้น โครงข่ายประสาทเทียมสามารถแบ่งได้เป็นประเภทดังนี้

1.1.3.1 Feedforward Network เป็นโครงข่ายประสาทเทียมที่มีการเรียนรู้แบบมีผู้สอน การประมวลผลข้อมูลไปข้างหน้าอย่างเดียวเชื่อมโยงจากชั้นที่ติดกันโดยไม่ย้อนกลับ จากโหนดอินพุท (Input Node) ไปยังโหนดเอาท์พุท (Output Node) (Haykin, 2008) โครงสร้างของ Feedforward Network ดังภาพประกอบ 1.1 เช่น โครงข่ายประสาทเทียมแบบเพอร์เซปตรอนหลายชั้น (Multilayer Perceptron Neural Network: MLP) ประกอบด้วย

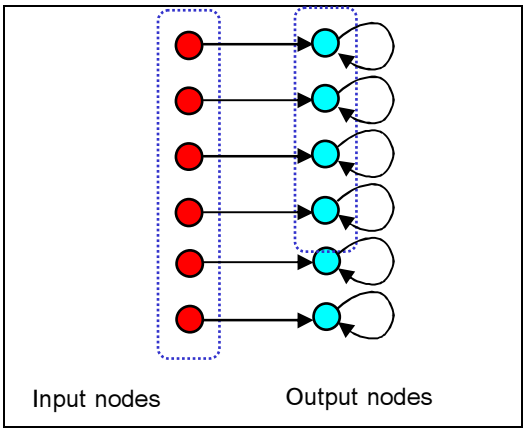
ชั้นข้อมูลเข้า (Input Layer) ชั้นซ่อน (Hidden Layer) และชั้นผลลัพธ์ (Output Layer) ซึ่งจำนวนของ Hidden Layer มีหนึ่งเลเยอร์หรือมากกว่าก็ได้ และโครงข่ายประสาทเทียมแบบเรเดียลเบซิสฟังก์ชัน (Radial Basis Function Neural Network: RBF) ประกอบด้วย ชั้นข้อมูลเข้า (Input Layer) ชั้นซ่อน (Hidden Layer) และชั้นผลลัพธ์ (Output Layer) แต่จะมีจำนวน Hidden Layer เพียงหนึ่งเลเยอร์เท่านั้น



ภาพประกอบ 1.1 โครงสร้างของ Feedforward Network

(ที่มา: ศุภชัย ตั้งบุญญะศิริ, 2551: ระบบออนไลน์)

1.1.3.2 Recurrent Network เป็นโครงข่ายประสาทเทียมที่มีการเรียนรู้แบบมีผู้สอน การประมวลผลข้อมูลอาจมีการย้อนกลับจากชั้นหนึ่งไปยังชั้นก่อนหน้าจนกระทั่งได้คำตอบ ข้อเสียของโครงข่ายประสาทเทียมแบบย้อนกลับคือ ใช้เวลาในการประมวลผลนาน ดังภาพประกอบ 1.2



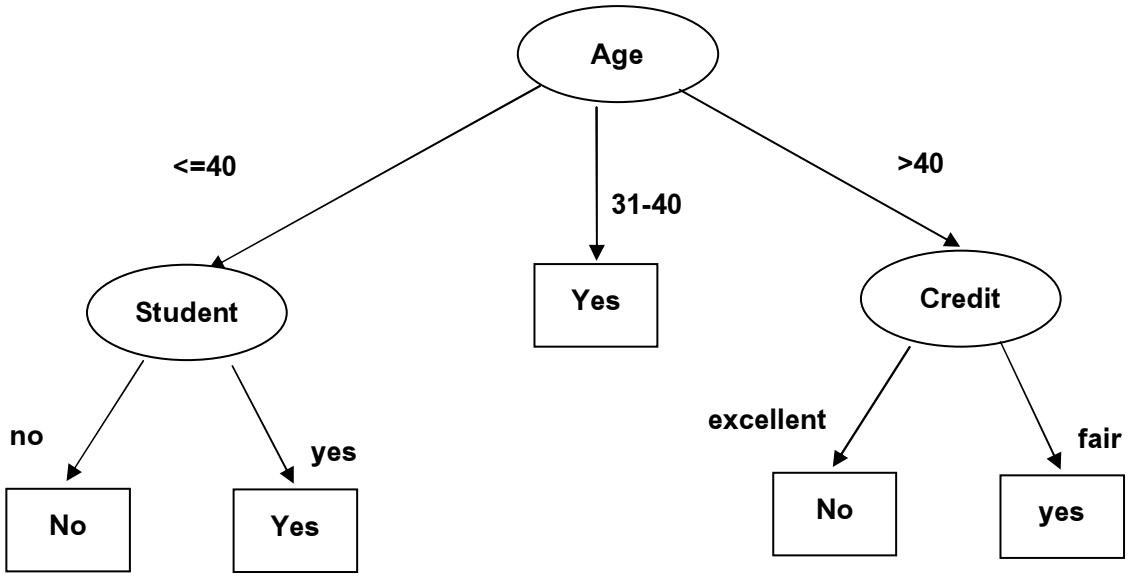
ภาพประกอบ 1.2 โครงสร้างของ Recurrent Network

(ที่มา: ศุภชัย ตั้งบุญญะศิริ, 2551: ระบบออนไลน์)

1.1.4 ต้นไม้ตัดสินใจ (Decision Tree)

เป็นวิธีการเรียนรู้โดยการจำแนกประเภท (Classification) ข้อมูลในกลุ่มตัวอย่างออกเป็นกลุ่มย่อย (Class) ต่างๆ แบบมีผู้สอน (Supervised Learning) โดยที่ผลลัพธ์

ของข้อมูลในกลุ่มย่อยแต่ละกลุ่มเป็นอย่างเดียวกัน ต้นไม้ที่ได้จากการเรียนรู้ทำให้ทราบว่าแอทริบิวต์ใดของข้อมูลเป็นตัวกำหนดผลลัพธ์ และแอทริบิวต์ของข้อมูลแต่ละแอทริบิวต์มีความสำคัญมากน้อยต่างกันอย่างไร ต้นไม้ตัดสินใจประกอบด้วยโหนด (Node) และกิ่ง (Link) ซึ่งจะต่อกับโหนด ที่ปลายสุดของโหนดเรียกว่าลีฟโหนด (Leaf Node) โดยโหนดจะแทนแอทริบิวต์ กิ่งจะแทนผลการทดสอบและลีฟโหนดจะแทนคลาส (Class) การเลือกคุณสมบัติที่ใช้เป็นรากหรือโหนดในต้นไม้โดยการคำนวณค่าเกนของแอทริบิวต์แล้วเลือกแอทริบิวต์ที่มีค่าเกนสูงสุดมาเป็นราก ตัวอย่างอัลกอริทึมของต้นไม้ตัดสินใจเช่น อัลกอริทึม CART ใช้ค่าดัชนีจีนี (Gini Index) สามารถใช้กับแอทริบิวต์ที่เป็นค่าต่อเนื่องหรือไม่ก็ได้เหมาะกับแอทริบิวต์ที่มีค่าแตกต่างกันมาก อัลกอริทึม ID3 ใช้ค่าเกนสารสนเทศ (Information Gain) โดยใช้กับค่าแอทริบิวต์แบบกลุ่ม (Categorical) ถ้าค่าแอทริบิวต์เป็นค่าต่อเนื่องให้แปลงเป็นค่าไม่ต่อเนื่องก่อน และ C4.5 จะใช้ค่ามาตรฐานเกน (Gain Criteria) ซึ่งได้จากค่าเกนสารสนเทศหารด้วยค่าสารสนเทศ (Entropy) ของแต่ละแอทริบิวต์ งานวิจัยที่ใช้ต้นไม้ตัดสินใจมาใช้ในด้านการแก้ปัญหาความกำกวมของคำเช่น การแก้ปัญหาความกำกวมโดยใช้ต้นไม้ตัดสินใจ J48 (O'Hara *et al*., 2004) เป็นต้น ตัวอย่างรูปแบบของต้นไม้ตัดสินใจแสดงดังภาพประกอบ 1.3 ซึ่งเป็นต้นไม้ตัดสินใจของการซื้อคอมพิวเตอร์ (กรุง สินอภิรมย์สราญ, 2551: ระบบออนไลน์)



ภาพประกอบ 1.3 ตัวอย่างต้นไม้ตัดสินใจ

1.1.5 คลังข้อความ

คลังข้อความภาษา (Corpus) หมายถึงข้อมูลภาษาที่ได้มีการเก็บบันทึกไว้ในระบบคอมพิวเตอร์ โดยจะให้ข้อมูลในการใช้ภาษาที่เกิดขึ้นจริง จึงเป็นแหล่งข้อมูลที่สำคัญต่อการศึกษาวิจัยต่างๆเกี่ยวกับภาษา การใช้คลังข้อความในงานประยุกต์ต่างๆ เช่น การเรียนการสอนภาษา การวิจัยทางภาษาศาสตร์ การแปลภาษาและการสอนภาษา การประมวลผลภาษาธรรมชาติ การทำพจนานุกรมและการประมวลศัพท์ (วิโรจน์ อรุณมานะกุล, 2550: ระบบออนไลน์) ประเภทของคลังข้อความภาษาแบ่งตามจำนวนภาษาได้ดังนี้

1) คลังข้อความภาษาเดียว (Monolingual Corpus) เป็นคลังข้อความของภาษาใดภาษาหนึ่งซึ่งใช้ประโยชน์ในงานการประมวลผลภาษาธรรมชาติในด้าน การตัดคำ การเรียนการสอนภาษาเช่น คลังข้อความ Senseval-2

Senseval-2 (Pedersen, 2001: Online) เป็นคลังข้อความมาตรฐานในการวัดประสิทธิภาพของการแก้ปัญหาความกำกวมของคำ ประกอบ ด้วยคลังข้อความย่อยหลายภาษาเช่น ภาษาอิตาลี ภาษาญี่ปุ่น และภาษาอังกฤษ เป็นต้น คลังข้อความ Senseval-2 เป็นคลังข้อความของข้อมูลสอนและทดสอบ ประกอบด้วยคำกำกวมหลายคำ แต่ละคำมีตัวอย่างประโยคจำนวนมากและมีการกำหนดความหมายของคำกำกวมในแต่ละประโยคไว้เพื่อใช้ในการจำแนกความหมายได้ถูกต้อง งานวิจัยด้านการแก้ปัญหาความกำกวมของคำส่วนใหญ่ได้นำคลังข้อความ Senseval-2 มาใช้ในการวัดประสิทธิภาพของแบบจำลองการแก้ปัญหาความกำกวมของคำ (Mihalcea, 2004; Suarez and Palomar, 2002; Pham *et al.,* 2005; Ciaramita *et al*., 2003)

2) คลังข้อความสองภาษา (Bilingual Corpus) สามารถแบ่งได้เป็น 2 ลักษณะคือ

- คลังข้อความเทียบบท (Parallel Corpus) เป็นคลังข้อความของทั้งภาษาต้นฉบับและภาษาแปล ซึ่งมีการจับคู่ข้อมูลระหว่างคลังข้อความทั้งสอง เช่น จับคู่ระหว่างประโยคของภาษาต้นฉบับกับภาษาแปล ทำให้รู้ว่าประโยคแบบนี้แปลออกมาเป็นอีกภาษาหนึ่งอย่างไร ประโยชน์ของคลังข้อความเทียบบทในการแปลคือ ทำให้เห็นตัวอย่างในการแปล สามารถค้นดูคำแปลที่เคยใช้กัน และสามารถเข้าใจธรรมชาติของการแปลได้ (วิโรจน์ อรุณมานะกุล, 2550: ระบบออนไลน์)

- คลังข้อความเทียบภาษา (Comparable Corpus) เป็นคลังข้อความของสองภาษาโดยข้อมูลภาษาทั้งสองนั้นเป็นภาษาต้นฉบับทั้งคู่ แต่สามารถเปรียบเทียบกันได้ (วิโรจน์ อรุณมานะกุล, 2550: ระบบออนไลน์)

คลังข้อความภาษาไทยนั้นปัจจุบันมีผู้ได้สร้างขึ้นแล้ว เช่น คลังข้อความออร์คิด (Nectec, 1997: Online) สร้างโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และ

คอมพิวเตอร์แห่งชาติ (NECTEC) และคลังข้อความ Thai Concordance (Aroonmanakun, 1999: Online) จากจุฬาลงกรณ์มหาวิทยาลัยแต่เป็นข้อมูลที่จัดเก็บตามสะดวก ไม่ครอบคลุมการใช้ภาษาในลักษณะต่างๆอย่างเคร่งครัด ระบบคลังข้อความทั้งสองเป็นคลังข้อความแบบคลังข้อความภาษาเดียว

1.2 วัตถุประสงค์ของโครงการ

1.2.1 สร้างแบบจำลองการแก้ปัญหาความกำกวมของคำจากคลังข้อความโดยใช้เทคนิคคำบริบท

1.2.2 พัฒนาโปรแกรมการแก้ปัญหาความกำกวมของคำจากคลังข้อความโดยใช้เทคนิคคำบริบท

1.3 ขอบเขตการดำเนินงาน

1.3.1 ออกแบบและสร้างแบบจำลองในการแก้ปัญหาความกำกวมของคำ

1.3.2 พัฒนาโปรแกรมการแก้ปัญหาความกำกวมของคำ

1.3.3 ใช้คลังข้อความ Senseval-2 ซึ่งเป็นคลังข้อความมาตรฐานในการวัดประสิทธิภาพของการแก้ปัญหาความกำกวมมาใช้ในการแก้ปัญหาความกำกวมของคำ

1.4 ขั้นตอนและระยะเวลาการดำเนินการ

1.4.1 ขั้นตอนการดำเนินงาน

1) ศึกษางานวิจัยและเอกสารที่เกี่ยวข้องกับระบบการแก้ปัญหาความกำกวมของคำ

2) เตรียมข้อความ โดยข้อความที่นำมาทดสอบเป็นคลังข้อความภาษาอังกฤษของ Senseval-2 ซึ่งเป็นคลังข้อความมาตรฐานที่ใช้ในการวัดประสิทธิภาพของโปรแกรมด้านการแก้ปัญหาความกำกวมของคำ

3) สร้างแอทริบิวต์โดยใช้คำบริบท

4) เลือกแอทริบิวต์เพื่อลดจำนวนแอทริบิวต์ที่ไม่สำคัญออก

5) การจำแนกความหมายของคำ (Classification) นำข้อมูลที่ได้มาจำแนกความหมายของคำเพื่อหาความหมายที่ถูกต้องของคำในบริบท

6) ศึกษาเทคโนโลยีและเครื่องมือสนับสนุน

7) วิเคราะห์และออกแบบโปรแกรมการแก้ปัญหาความกำกวมของคำ

8) พัฒนาโปรแกรมการแก้ปัญหาความกำกวมของคำ

9) ทดสอบและติดตั้งโปรแกรมการแก้ปัญหาความกำกวมของคำ

10) จัดทำเอกสารประกอบโปรแกรมการแก้ปัญหาความกำกวมของคำและเขียนผลงานวิจัย

11) จัดทำเอกสารวิทยานิพนธ์

1.4.2 ระยะเวลาการดำเนินงาน

มิถุนายน 2549 – มีนาคม 2551

1.4.3 แผนการดำเนินการวิจัย

ตารางที่ 1.1 แสดงระยะเวลาการดำเนินงานวิจัย

กิจกรรม/ขั้นตอนการดำเนินงาน	เดือน																							
	2549							2550												2551				
	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4
1.ศึกษาและทำความเข้าใจการแก้ปัญหาความกำกวมและเทคนิคการแก้ปัญหาความกำกวม																							
2.ศึกษางานวิจัยและเอกสารที่เกี่ยวข้อง																							
3.ศึกษาเทคโนโลยีและเครื่องมือสนับสนุน																							
4.วิเคราะห์และออกแบบระบบ																							
5.พัฒนาระบบ																							
6.ทดสอบและติดตั้งระบบ																							
7.จัดทำเอกสารประกอบระบบและเขียนผลงานวิจัย																							
8.จัดทำเอกสารวิทยานิพนธ์																							

1.5 สถานที่และเครื่องมือที่ใช้

1.5.1 สถานที่

ห้องปฏิบัติการคอมพิวเตอร์ CS 207ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

1.5.2 เครื่องมือที่ใช้

1) ด้านฮาร์ดแวร์

คอมพิวเตอร์ส่วนบุคคล หน่วยความจำ 1 กิกะไบต์ ฮาร์ดดิสก์ ความจุ 40 กิกะไบต์ สำหรับพัฒนาและเป็นเครื่องทดสอบ

2) ด้านซอฟต์แวร์

2.1) ระบบปฏิบัติการ Microsoft Windows XP

2.2) Perl

2.3) Ngram Statistic Package (NSP)

2.4) SenseTools

2.5) WEKA

2.6) Visual Basic.Net

2.7) OMtoSVAL2

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้แบบจำลองการแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคคำบริบท

1.6.2 ได้โปรแกรมจากแบบจำลองการแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคคำบริบท

บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับปัญหาความกำกวมของคำ

ทฤษฎีต่าง ๆที่เกี่ยวข้องกับความกำกวมของคำประกอบด้วย คำกำกวม คลังข้อความ Senseval-2 การตัดคำที่เป็น Stoplist โปรแกรม NSP โปรแกรม SenseTools คำบริบท การเลือกแอทริบิวต์ (Attribute Selection) การจำแนกความหมายของคำ (Classification) และการใช้การตรวจสอบไขว้ (Cross Validation)

2.1 คำกำกวม

คำกำกวม (Ambiguous Word) คือคำที่มีความหมายหลายความหมายในบริบทที่แตกต่างกัน ความกำกวมของคำเป็นสิ่งที่ทำให้เกิดความผิดพลาดได้ในงานประยุกต์ด้านเทคโนโลยีของภาษา เช่น การแปลภาษา (Machine Translation) (Carpuat and Wu, 2005) และการสืบค้นเอกสาร (Information Retrieval) (Stokoe *et al.,* 2003) ตัวอย่างคำที่มีความหมายกำกวม (Riloff, 2006: Online) ดังตารางที่ 2.1

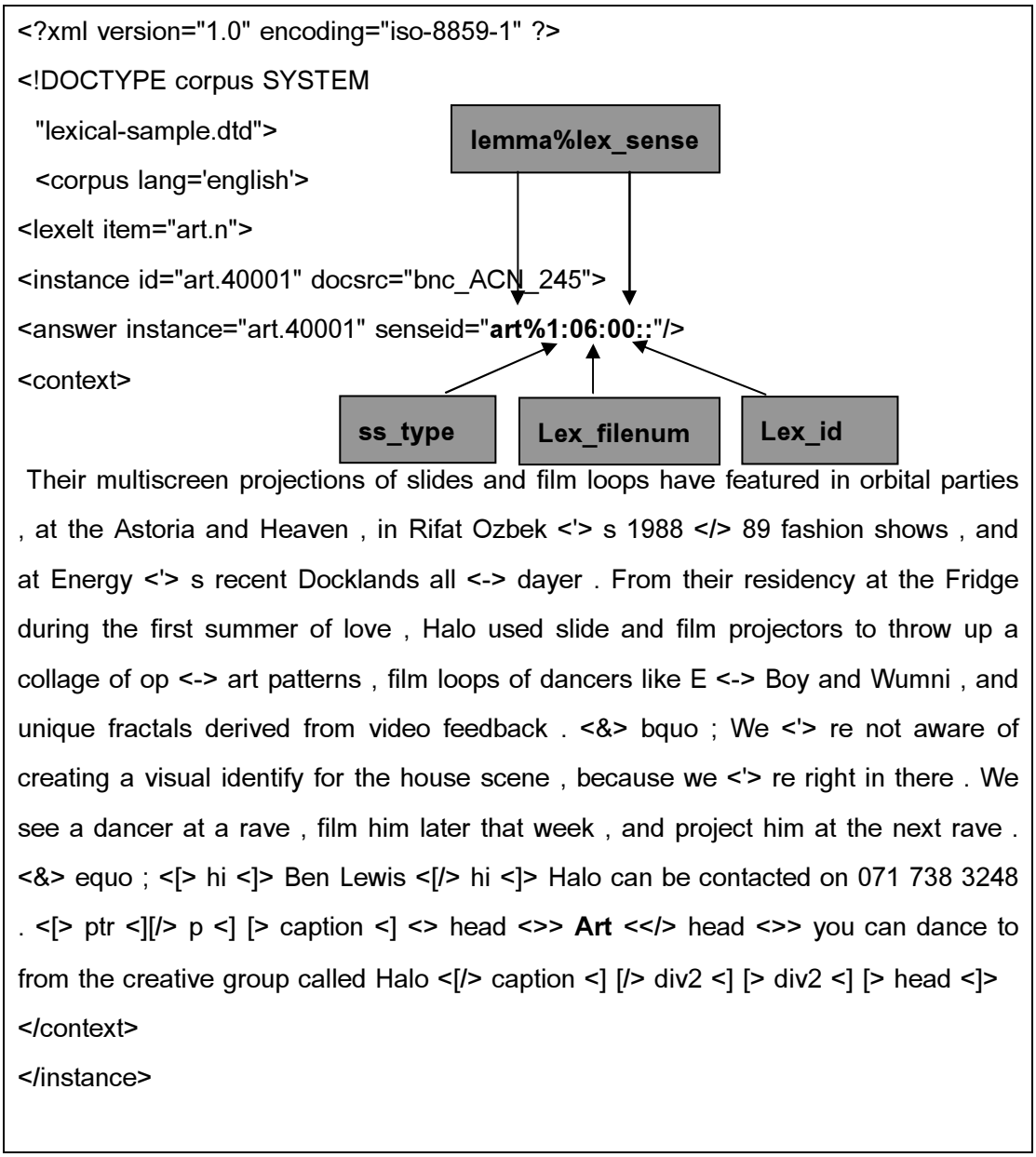
ตารางที่ 2.1 ตัวอย่างคำกำกวม

ประโยคที่มีคำกำกวม	ประโยคขยาย	ความหมาย
I walked to the bank.	กรณีที่ 1 The water looked inviting. กรณีที่ 2 I needed to deposit a check.	ตลิ่ง ธนาคาร
He did not want to run again.	กรณีที่ 1 His ankle was still sore. กรณีที่ 2 He was tired of politics.	วิ่ง ทำงาน
The table looked good.	กรณีที่ 1 The legs were beautifully carved. กรณีที่ 2 The numbers substantiated their claims.	โต๊ะ ตาราง

จากตัวอย่างคำกำกวมดังตารางที่ 2.1 ประโยคในคอลัมน์ที่ 1 เป็นประโยคที่มีคำกำกวมเป็นคำที่ขีดเส้นใต้คือ bank run และ table มีความหมายได้มากกว่าหนึ่งความหมาย ส่วนประโยคจากคอลัมน์ที่ 2 แบ่งเป็นสองกรณี เป็นประโยคที่ขยายเพื่อให้สามารถทราบว่าคำกำกวมนั้นมีความหมายอย่างไร คอลัมน์ที่ 3 จะแสดงความหมายของทั้งสองกรณีของคำกำกวมแต่ละคำ

2.2 คลังข้อความ Senseval-2

คลังข้อความ Senseval-2 (Pedersen, 2001: Online) เป็นคลังข้อความมาตรฐานที่ใช้ในการทดสอบประสิทธิภาพของโปรแกรมในการแก้ปัญหาความกำกวมของคำ คลังข้อความ Senseval-2 มีลักษณะเป็นรูปแบบของ XML ดังภาพประกอบ 2.1



ภาพประกอบ 2.1 ตัวอย่างรูปแบบของคลังข้อความ Sneseval-2

ความหมายของคำกำกวมแต่ละความหมายกำหนดโดย WordNet (George, 1998: Online) ซึ่งเป็นสัญลักษณ์ของความหมายมีรูปแบบดังนี้

lemma % lex_sense

- **lemma** คือ คำกำกวม
- **lex_sense** คือ สัญลักษณ์ของความหมายประกอบด้วยสัญลักษณ์ย่อย คั่นด้วยเครื่องหมาย : แต่ละความหมายมีรูปแบบดังนี้

ss_type:lex_filenum:lex_id มีส่วนประกอบดังนี้

- ss_type เป็นตัวเลขจำนวนเต็ม 1 หลักแทนด้วยชนิดของคำดังตารางที่ 2.2 ดังนี้

ตารางที่ 2.2 ss_type

รหัส	ประเภทของคำ	ความหมาย
1	NOUN	คำนาม
2	VERB	คำกริยา
3	ADJECTIVE	คำคุณศัพท์
4	ADVERB	คำกริยาวิเศษณ์

- lex_filenum เป็นตัวเลขจำนวนเต็ม 2 หลักแทนด้วยลักษณะของความหมายซึ่งเป็นตัวเลขที่บ่งบอกทิศทางของความหมาย (Sense) ตามไฟล์พจนานุกรม (Lexicographer File) ดังตารางที่ 2.3

- lex_id เป็นตัวเลขจำนวนเต็ม 2 หลักหมายถึงส่วนเพิ่มเติมของคำศัพท์ (lemma) ค่าปกติจะกำหนดเป็น 00

ตารางที่ 2.3 Lexicographer File

File Number	Name	Contents
00	adj.all	all adjective clusters
01	adj.pert	relational adjectives (pertainyms)
02	adv.all	all adverbs
03	noun.Tops	unique beginner for nouns
04	noun.act	nouns denoting acts or actions
05	noun.animal	nouns denoting animals

ตารางที่ 2.3 Lexicographer File (ต่อ)

File Number	Name	Contents
06	noun.artifact	nouns denoting man-made objects
07	noun.attribute	nouns denoting attributes of people and objects
08	noun.body	nouns denoting body parts
09	noun.cognition	nouns denoting cognitive processes and contents
10	noun.communication	nouns denoting communicative processes and contents
11	noun.event	nouns denoting natural events
12	noun.feeling	nouns denoting feelings and emotions
13	noun.food	nouns denoting foods and drinks
14	noun.group	nouns denoting groupings of people or objects
15	noun.location	nouns denoting spatial position
16	noun.motive	nouns denoting goals
17	noun.object	nouns denoting natural objects (not man-made)
18	noun.person	nouns denoting people
19	noun.phenomenon	nouns denoting natural phenomena
20	noun.plant	nouns denoting plants
21	noun.possession	nouns denoting possession and transfer of possession
22	noun.process	nouns denoting natural processes
23	noun.quantity	nouns denoting quantities and units of measure
24	noun.relation	nouns denoting relations between people or things or ideas
25	noun.shape	nouns denoting two and three dimensional shapes

ตารางที่ 2.3 Lexicographer File (ต่อ)

File Number	Name	Contents
26	noun.state	nouns denoting stable states of affairs
27	noun.substance	nouns denoting substances
28	noun.time	nouns denoting time and temporal relations
29	verb.body	verbs of grooming, dressing and bodily care
30	verb.change	verbs of size, temperature change, intensifying,etc.
31	verb.cognition	verbs of thinking, judging, analyzing, doubting
32	verb.communication	verbs of telling, asking, ordering, singing
33	verb.competition	verbs of fighting, athletic activities
34	verb.consumption	verbs of eating and drinking
35	verb.contact	verbs of touching, hitting, tying, digging
36	verb.creation	verbs of sewing, baking, painting, performing
37	verb.emotion	verbs of feeling
38	verb.motion	verbs of walking, flying, swimming
39	verb.perception	verbs of seeing, hearing, feeling
40	verb.possession	verbs of buying, selling, owning
41	verb.social	verbs of political and social activities and events
42	verb.stative	verbs of being, having, spatial relations
43	verb.weather	verbs of raining, snowing, thawing, thundering
44	adj.ppl	participial adjectives

ตัวอย่างความหมายของคำกำกวม art ความหมายแรกคือ art%1:06:00 เมื่อแปลตามลักษณะของสัญลักษณ์รหัส 06 คือ noun.artifact จะได้ความหมายที่เกี่ยวกับผลิตภัณฑ์ศิลปะ ความหมายที่ 2 และ 3 แสดงรายละเอียดของความหมายดังตารางที่ 2.4

ตารางที่ 2.4 ตัวอย่างความหมายของคำกำกวม art

รหัส	lemma%	lex_sense			ความหมาย
		ss_type	lex_filenum	lex_id	
art%1:06:00::	art%	1	06	00	ผลิตภัณฑ์ศิลปะ
art%1:04:00::	art%	1	04	00	การสร้างงานศิลปะ
art%1:09:00::	art%	1	09	00	ทักษะ

2.3 การตัดคำที่เป็น Stoplist

ในการสืบค้นเอกสาร (Information Retrieval) จะมีคำบางคำในข้อความซึ่งเป็นคำที่มีความเกี่ยวข้องกับเอกสารน้อย และเอกสารมีขนาดใหญ่ขึ้น ทำให้ประสิทธิภาพในการค้นคืนเอกสารต่ำลง คำเหล่านั้นเรียกว่า Stoplist (Frakes and Yates, 1992) คำที่เป็น Stoplist ดังภาพประกอบ 2.2 คำเหล่านี้เมื่อตัดออกจะทำให้ประสิทธิภาพในการสืบค้นเอกสารดีขึ้นและมีความสำคัญต่อการแก้ปัญหาความกำกวมของคำ เนื่องจากคำเหล่านี้ไม่ได้นำมาวิเคราะห์หาความหมายของคำ วิทยานิพนธ์นี้ได้นำเทคนิคการตัดคำที่เป็น Stoplist ออกจากคลังข้อความเพื่อให้เพิ่มความถูกต้องในการแก้ปัญหาความกำกวมและเพิ่มความรวดเร็วในการทดลอง เนื่องจากมีการตัดคำที่ไม่สำคัญทิ้งไป

a	about	above	across	after	again
against	all	almost	alone	along	already
also	although	always	among	an	and
another	any	anybody	anyone	anything	anywhere
are	area	areas	around	as	ask
asked	asking	asks	at	away	b
back	backed	backing	backs	be	because
became	become	becomes	been	before	began
behind	being	beings	best	better	between
big	both	but	by	c	came
can	cannot	case	cases	certain	certainly
clear	clearly	come	could	d	did
differ	different	differently	do	does	done

ภาพประกอบ 2.2 Stoplist

down	downed	downing	downs	during	each
early	either	end ended	ending	ends	enough
even	evenly	ever	every	eberybody	everyone
everything	everywhere	f	face	faces	fact
facts	far	felt	few	find	finds
first	for	four	from	full	fully
further	furthered	furthering	furthers	g	gave
general	generally	get	gets	give	given
gives	go	going	good	goods	got
great	greater	greatest	group	grouped	grouping
groups	h	had	has	have	having
he	her	herself	here	high	higher
highest	him	himself	his	how	however
i	if	important	in	interest	interested
interesting	interests	into	is	it	its
itself	j	just	k	keepkeeps	kind
knew	know	nown	knows	l	large
largely	last	later	latest	least	less
let	lets	like	likely	long	longer
longest	m	made	make	making	man
many	may	me	member	members	men
might	more	most	mostly	mr	mrs
much	must	myself	n	necessary	need
needing	needs	never	new	newest	next
no	non	nobody	none	nothing	now
number	numbered	numbering	numbers	of	off
often	old	oldest	on	once	one
open	opened	opening	opens	order	ordered
ordering	orders	others	our	out	over
part	parted	parting	parts	perhaps	places
point	pointing	points	possible	present	presenting
presents	problem	problems	puts	q	quite
r	really	right	room	rooms	s
said	same	saw	say	second	seconds
see	seem	seeming	seems	sees	several

ภาพประกอบ 2.2 Stoplist (ต่อ)

she	should	show	showed	shows	side
sides	since	smaller	smallest	so	some
someone	something	somewhere	state	still	such
sure	t	taken	than	that	the
them	then	there	therefore	they	thing
things	think	this	those	though	thought
three	through	thus	to	together	too
took	toward	turned	turning	turns	two
under	until	up	upon	use	uses
used	v	w	want	wanted	wanting
was	way	ways	we	wells	went
were	what	where	whether	which	while
whole	whose	why	will	within	without
work	worked	works	would	x	y
years	yet	you	young	youngest	your
yours	z				

ภาพประกอบ 2.2 Stoplist (ต่อ)

2.4 โปรแกรม NSP (Ngram Statistic Package)

Ngram Statistic Package หรือ NSP เป็นโปรแกรมที่ช่วยในการวิเคราะห์สร้าง N-gram และนับจำนวนความถี่ของคำทั้งหมดในข้อความ พัฒนาขึ้นโดยใช้ภาษา perl (Pedersen, 2006: Online)

N-gram หมายถึง ลำดับของคำ จำนวน N ตัว แต่ละคำจะถูกสร้างเป็น N-gram คั่นด้วยเครื่องหมาย <> เช่น big<> เป็น 1-gram ของคำ “big” และ stock<>falling<> เป็น 2-gram ของคำ “stock” และ “falling” เป็นต้น ตัวอย่างของประโยค **“I went to the bank.”** เมื่อนำมาสร้าง 1-gram จะได้ **I<> went<> to<> the<> bank<>**

โปรแกรม NSP ประกอบด้วยโปรแกรมย่อยที่ใช้ในงานวิจัยคือ count.pl ใช้สำหรับสร้างจำนวน N-gram ให้กับคำ

2.5 โปรแกรม SenseTools

SenseTools เวอร์ชัน 0.3 (Pedersen, 2003: Online) เป็นโปรแกรมที่ทำหน้าที่แปลงข้อความให้อยู่ในรูปแบบ Feature Vectors ซึ่งเป็นรูปแบบของ arff ที่ใช้ใน WEKA (Ian and Frank, 2005a) โดยทำงานต่อจากโปรแกรม count.pl ของ NSP โปรแกรมย่อยของ

SenseTools ที่ใช้ในงานวิจัยนี้คือ nsp2regex.pl ใช้สำหรับสร้าง Regular Expressions xml2arff.pl ใช้สำหรับสร้าง Feature Vectors รูปแบบ arff และ tilde.pl ใช้สำหรับเปลี่ยนสัญลักษณ์ % เป็น ~ ทั้งสามโปรแกรมย่อยนี้จะทำงานตามลำดับเพื่อสร้างข้อความให้มีรูปแบบ arff ที่สมบูรณ์ ข้อความที่เหมาะสมที่นำมาใช้ในการสร้างให้เป็นรูปแบบ arff ด้วย SenseTools นี้จะต้องเป็นรูปแบบของคลังข้อความ Senseval-2

2.6 คำบริบท (Context)

คำบริบท (Context) ของคำที่มีความหมายกำกวม (Ambiguous Word) เป็นคำแวดล้อมอยู่รอบๆคำกำกวมทางซ้ายและขวา คำบริบทแต่ละคำจะเป็นคำที่กล่าวถึงสิ่งต่างๆที่ในลักษณะที่มีความสัมพันธ์กันทางความหมายในประโยคนั้นๆ ในการนำคำบริบทมาใช้ในการแก้ปัญหาความกำกวมของคำจะทำให้ทราบความหมายที่ถูกต้องของคำที่มีความหมายกำกวมได้ จากตัวอย่างประโยคที่มีคำว่า art เป็นคำกำกวมดังนี้

“There’s always one to be heard somewhere during the summer; in the piazza in front of the **art** gallery and Town Hall or in a park.”

คำทั้งหมดที่อยู่ทางซ้ายมือของ “art” จะหมายถึงคำบริบททางซ้าย และคำทั้งหมดที่อยู่ทางขวามือของ art จะหมายถึงคำบริบททางขวา

2.7 การเลือกแอทริบิวต์ (Attribute Selection)

การเลือกแอทริบิวต์ (Attribute Selection) หรือการกรองแอทริบิวต์เป็นการลดจำนวนแอทริบิวต์ที่ไม่เกี่ยวข้องออก โดยจะถูกตัดออกไปเหลือเฉพาะแอทริบิวต์ที่มีความสัมพันธ์กันเท่านั้น ข้อดีของการลดจำนวนแอทริบิวต์คือใช้ตัวอย่างที่มีความสำคัญมาสอนทำให้ผลการจำแนกความหมายได้ค่าความถูกต้องสูงขึ้น เทคนิคการกรองแอทริบิวต์ในวิทยานิพนธ์นี้มี 2 วิธีดังต่อไปนี้

2.7.1 Information Gain Attribute Evaluation

เป็นการลดจำนวนแอทริบิวต์ที่ใช้การประเมินค่าของแอทริบิวต์โดยวัด Information Gain (Ganchev *et al*., 2006; Ian and Frank, 2005b) ซึ่งเป็นตัววัดความสัมพันธ์ของแอทริบิวต์ให้กับคลาสนั้นๆ การหาค่า IG (Information Gain) สามารถคำนวณได้ดังสมการที่ (2.1)

$$IG = H(Y) - H(Y \mid X) \qquad (2.1)$$

กำหนดให้ Y คือ คลาส และ X คือ แอทริบิวต์
$H(Y)$ คือ ค่าเอนโทรปีของ Y
$H(Y \mid X)$ คือ ค่าเอนโทรปีของ Y เมื่อมีเงื่อนไข X

การหาค่า $H(Y)$ แสดงได้ดังสมการที่ (2.2) และการหาค่า $H(Y \mid X)$ แสดงได้ดังสมการที่ (2.3)

$$H(Y) = -\sum_{y \in Y} p(y) \log_2 (p(y)) \tag{2.2}$$

$$H(Y \mid X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y \mid x) \log_2 (p(y \mid x)) \tag{2.3}$$

โดยที่ $p(y)$ คือ ความน่าจะเป็นของ y
$p(x)$ คือ ความน่าจะเป็นของ x
$p(y \mid x)$ คือ ความน่าจะเป็นของ y เมื่อรู้ x

2.7.2 Gain Ratio Attribute Evaluation

เป็นการลดจำนวนแอทริบิวต์ที่ใช้การประเมินค่าของแอทริบิวต์โดยวัด Gain Ratio (Ganchev *et al*., 2006; Ian and Frank, 2005b) ซึ่งวัดความสัมพันธ์ของแอทริบิวต์อีกประเภทหนึ่งแต่จะมีการปรับสเกลตามค่าของข้อมูลในแอทริบิวต์ที่สนใจให้กับคลาสนั้นๆ การหา GR (Gain Ratio) คำนวณได้ดังสมการที่ (2.4)

$$GR = \frac{IG}{H(X)} \tag{2.4}$$

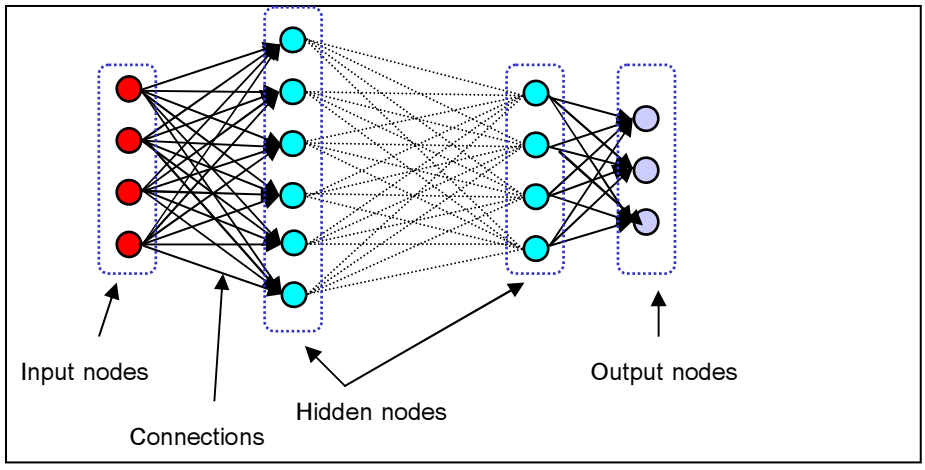
การกรองทั้งสองแบบนี้เป็นการกรองที่ต้องใช้การค้นหาแบบจัดลำดับ (Ranker Search Method) แอทริบิวต์จะถูกเรียงลำดับความสำคัญโดยตัดแอทริบิวต์ที่ไม่ต้องการออก ข้อแตกต่างระหว่าง IG และ GR (Ganchev *et al*., 2006) คือ GR ได้จากการหารด้วยค่าเอนโทรปีทำให้ค่าที่ได้อยู่ระหว่าง [0,1] ถ้าค่า GR เท่ากับ 0 หมายถึงไม่มีความสัมพันธ์ระหว่าง Y และ X ถ้าค่า GR มีค่าเท่ากับ 1 แสดงว่ามีความสัมพันธ์ระหว่าง Y และ X มากที่สุด ค่าที่ได้จาก GR จึงเป็นค่าที่น้อยเมื่อเทียบกับ IG ที่มีค่ามากกว่า

2.8 การจำแนก (Classification)

การจำแนกเป็นการเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยจะเรียนรู้จากลักษณะของตัวอย่างแล้วนำไปทำนายข้อมูลอื่นที่ไม่รู้ประเภท เทคนิคของการจำแนกความหมายของคำที่นำมาใช้ในวิทยานิพนธ์นี้มี 2 เทคนิคคือ Neural Network โดยใช้อัลกอริทึม RBFNetwork และ Decision Tree ใช้อัลกอริทึม ID3

2.8.1 Neural Network

โครงข่ายประสาทเทียม (Neural Networks) เป็นการเลียนแบบการทำงานของสมองมนุษย์ ที่ประกอบไปด้วยเซลล์พิเศษจำนวนมากที่เรียกว่าเซลล์ประสาท (Neuron) การเลียนแบบการทำงานของสมองมนุษย์ของเครื่องคอมพิวเตอร์เริ่มจากการกำหนดให้แต่ละโหนด (Node) เปรียบเสมือนเป็นเซลล์ประสาท และสร้างการเชื่อมต่อให้กับโหนดเหล่านั้นให้เป็นโครงข่าย (Network) แต่ละโครงข่ายจะประกอบไปด้วยโหนดที่ถูกจัดแบ่งเป็นชั้นๆ เรียกว่าเลเยอร์ (Layer) แต่ละเลเยอร์จะมีหน้าที่การทำงานแตกต่างกันดังภาพประกอบ 2.3 และการสอนโครงข่ายประสาทเทียมแสดงดังภาพประกอบ 2.4 (กรุง สินอภิรมย์สราญ, 2551: ระบบออนไลน์)



ภาพประกอบ 2.3 โครงสร้างของโครงข่ายประสาทเทียม
(ที่มา: ศุภชัย ตั้งบุญญะศิริ, 2551: ระบบออนไลน์)

1) เริ่มกำหนดค่าถ่วงน้ำหนักอย่างสุ่ม
2) วนซ้ำกับชุดของตัวอย่างทีละรอบ (Epoch)
    2.1 สำหรับตัวอย่าง 1 ตัวอย่าง
        2.1.1 คำนวณผลลัพธ์ที่ได้จากการใช้เครือข่ายโดยคำนวณแต่ละหนึ่งหน่วยสมอง (Neuron) คำนวณค่าผลรวมถ่วงน้ำหนักของข้อมูลเข้าแล้วจึงส่งผลลัพธ์ที่คำนวณได้เข้าฟังก์ชันกระตุ้น (Activation Function)
        2.1.2 ถ้าค่าที่คำนวณผิดจากผลลัพธ์ที่ต้องการ ให้ปรับค่าถ่วงน้ำหนักให้เหมาะสม
3) ทำซ้ำจนกระทั่งครบตามเงื่อนไขการหยุด เช่น การเรียนรู้ลู่เข้า (การทำนายไม่ดีขึ้น) หรือทำจนครบจำนวนการทำซ้ำสูงสุดที่กำหนด (Max epochs)

ภาพประกอบ 2.4 การสอนโครงข่ายประสาทเทียม

การใช้โครงข่ายประสาทเทียมในการจำแนกประเภทมีข้อดีคือความถูกต้องในการทำนายตัวอย่างที่พบใหม่มักสูงกว่าวิธีอื่น ตัวแบบที่ได้จะไม่เปลี่ยนไปมากเมื่อข้อมูลที่ใช้มีความผิดปกติอยู่ ผลลัพธ์ที่ต้องการสามารถเป็นค่าต่อเนื่องและค่าไม่ต่อเนื่องก็ได้และการหาคลาส คำนวณได้เร็วหลังผ่านขั้นการเรียนรู้แล้ว ส่วนข้อจำกัดของโครงข่ายประสาทเทียมคือ มักใช้เวลาในการเรียนรู้นาน มีพารามิเตอร์มากและการเลือกพารามิเตอร์ที่เหมาะสมเป็นเรื่องยาก (กรุง สินอภิรมย์สราญ, 2551: ระบบออนไลน์) ในวิทยานิพนธ์นี้ใช้โครงข่ายประสาทเทียมแบบ Radial Basis Function Neural Network ซึ่งมีรายละเอียดดังนี้

2.8.1.1 RBFNetwork (Radial Basis Function Neural Network) เป็นโครงข่ายประสาทเทียมประกอบด้วย 3 Layer คือ Input Layer Hidden Layer และ Output Layer (Ian and Frank, 2005b) โดย Hidden Unit มีรูปแบบการประมวลผลโดยใช้ฟังก์ชัน กระตุ้นแบบเรเดียล (Radial Activated Function) (Nikolaev, 2008 : Online) ซึ่งมี 3 แบบดังนี้

1) Multiquadratics: $\varphi(x) = (x^2 + c^2)^{1/2}$ เมื่อ $c>0$
2) Inverse multiquadratics: $\varphi(x) = 1 / (x^2 + c^2)^{1/2}$ เมื่อ $c > 0$
3) Gaussian: $\varphi(x) = \exp(- x^2 / 2\sigma^2)$ เมื่อ $\sigma >0$

โดยทั่วไปจะใช้ Gaussian function เป็น Radial Activated Function เอาท์พุทของฟังก์ชันกระตุ้นแบบเรเดียล ผลลัพธ์อยู่ในช่วง (0, 1) ดังสมการที่ (2.5)

$$F(x) = \sum_{i=1}^{n} w_i \exp(-\| x - x_i \|^2 / 2\sigma_i^2) \quad (2.5)$$

เมื่อ w_i คือ เป็นน้ำหนักของเอาท์พุดระหว่างHidden Unit และ Output Unit

n คือ จำนวน basis function

x_i คือ ศูนย์กลางของ basis function

x คือ input

2.8.2 Decision Tree

ต้นไม้ตัดสินใจเป็นการจำแนกโดยแทนความรู้ในรูปแบบของต้นไม้โดยการเรียนรู้จะใส่ข้อมูลเข้าไปและสร้างเป็นโมเดลอยู่ในรูปต้นไม้ตัดสินใจโดยที่กิ่ง (Link) ต่อกับโหนด (Node) ที่ปลายสุดของโหนดเรียกว่าลีฟโหนด (Leaf Node) แต่ละโหนดจะแทนแอทริบิวต์และกิ่งจะแทนผลในการทดสอบและลีฟโหนดจะแทนคลาสที่กำหนด ลักษณะการเรียนรู้ของต้นไม้ตัดสินใจ (กรุง สินอภิรมย์สราญ, 2551: ระบบออนไลน์) มีดังนี้

1) ผลการเรียนรู้แสดงอยู่ในรูปที่เข้าใจง่าย ทำให้ง่ายต่อการวิเคราะห์แอทริบิวต์ที่มีผลต่อการจำแนกกลุ่มต่างๆ

2) แต่ละเส้นทางจากโหนดรากถึงใบสามารถแสดงให้อยู่ในรูปกฎ IF-THEN ได้

3) มีความทนทานต่อข้อมูลเข้าที่มีสิ่งรบกวน

4) การเรียนรู้มีความรวดเร็วเมื่อเทียบกับอัลกอริทึมสำหรับจำแนกประเภทชนิดอื่น

5) เป็น Top-down Recursive divide-and-conquer คือสร้างต้นไม้จากบนลงล่างด้วยวิธีการแบ่งปัญหาออกเป็นปัญหาย่อย โดยเริ่มต้นเลือกแอทริบิวต์ที่ดีที่สุดมาสร้างเป็นโหนดรากจากข้อมูลสอน ถ้าแอทริบิวต์เป็นค่าต่อเนื่องต้องแปลงแอทริบิวต์ให้เป็นค่าไม่ต่อเนื่องก่อนแล้วจึงวนสร้างโหนดลูกและต้นไม้ย่อยของแต่ละกิ่ง

6) ข้อมูลผ่านการแบ่งแยกที่โหนดรากตามค่าแอทริบิวต์ของโหนดราก

7) หาแอทริบิวต์ที่ดีที่สุดของข้อมูลที่ผ่านการแบ่งแยกมาสร้างเป็นโหนดลูกของโหนดรากนั้นต่อไป

8) เงื่อนไขในการหยุดแบ่งคือ ตัวอย่างทุกตัวค่ามีคลาสเหมือนกันหมดหรือไม่มีแอทริบิวต์เหลือในการแบ่ง

9) แอทริบิวต์ถูกเลือกจากลำดับของตัวชี้วัดเช่น Information Gain และ Gain Ratio เป็นต้น

ในวิทยานิพนธ์นี้ใช้ต้นไม้ตัดสินใจ แบบ ID3 ซึ่งมีรายละเอียดดังนี้

2.8.2.1 ID3 เป็นขั้นตอนวิธีในการสร้างต้นไม้การตัดสินใจจากตัวอย่างแบบ divide-and-conquer เพื่อใช้ในการจำแนกข้อมูลในอนาคต (Ian and Frank, 2005b) ในการกำหนดแอทริบิวต์ใดให้เป็นโหนดรากของต้นไม้จะต้องคำนวณค่าเอนโทรปี (Entropy) และค่าเกน (Information Gain) เมื่อแอทริบิวต์ใดมีค่า Information Gain สูงสุดจะถูกเลือกเป็นโหนดราก และคำนวณค่า Information Gain ของแอทริบิวต์ที่เหลือไปเรื่อยๆจนได้ต้นไม้ตัดสินใจที่สมบูรณ์ การคำนวณค่า Information Gain และเอนโทรปีสามารถคำนวณได้จากสมการที่ (2.6) และ (2.7) ตามลำดับ

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2(p_i) \tag{2.6}$$

โดยที่ S คือ เซตของตัวอย่าง

c คือ จำนวนค่าของแอทริบิวต์

p_i คือ ความน่าจะเป็นของค่าแอทริบิวต์ที่ i

ดังนั้นค่า Information Gain ของความสัมพันธ์ระหว่างแอทริบิวต์ A กับเซตของตัวอย่าง S คำนวณได้จาก

$$Entropy(S) = \sum_{v \in Values(A)}^{n} \frac{|S_v|}{|S|} Entropy(S_v) \tag{2.7}$$

โดยที่ $Values(A)$ คือ เซตของค่าที่เป็นไปได้ของแอทริบิวต์ A

S_v คือ สับเซตของ S ของแอทริบิวต์ A มีค่า v

การเลือกแอทริบิวต์โดยใช้ Information Gain จากตัวอย่างข้อมูลอากาศ (Ian and Frank, 2005b) ดังภาพประกอบ 2.5 ประกอบด้วย 4 แอทริบิวต์คือ outlook temperature humidity และ windy เมื่อคำนวณค่า Information Gain แต่ละแอทริบิวต์จะได้ดังนี้

Gain (outlook) = 0.247

Gain (temperature) = 0.029

Gain (humidity) = 0.152

Gain (windy) = 0.048

หลังจากได้ค่า Information Gain แล้ว จะเห็นว่า แอทริบิวต์ outlook มีค่า Information Gain สูงสุดดังนั้นจึงเลือก outlook เป็นโหนดรากของต้นไม้

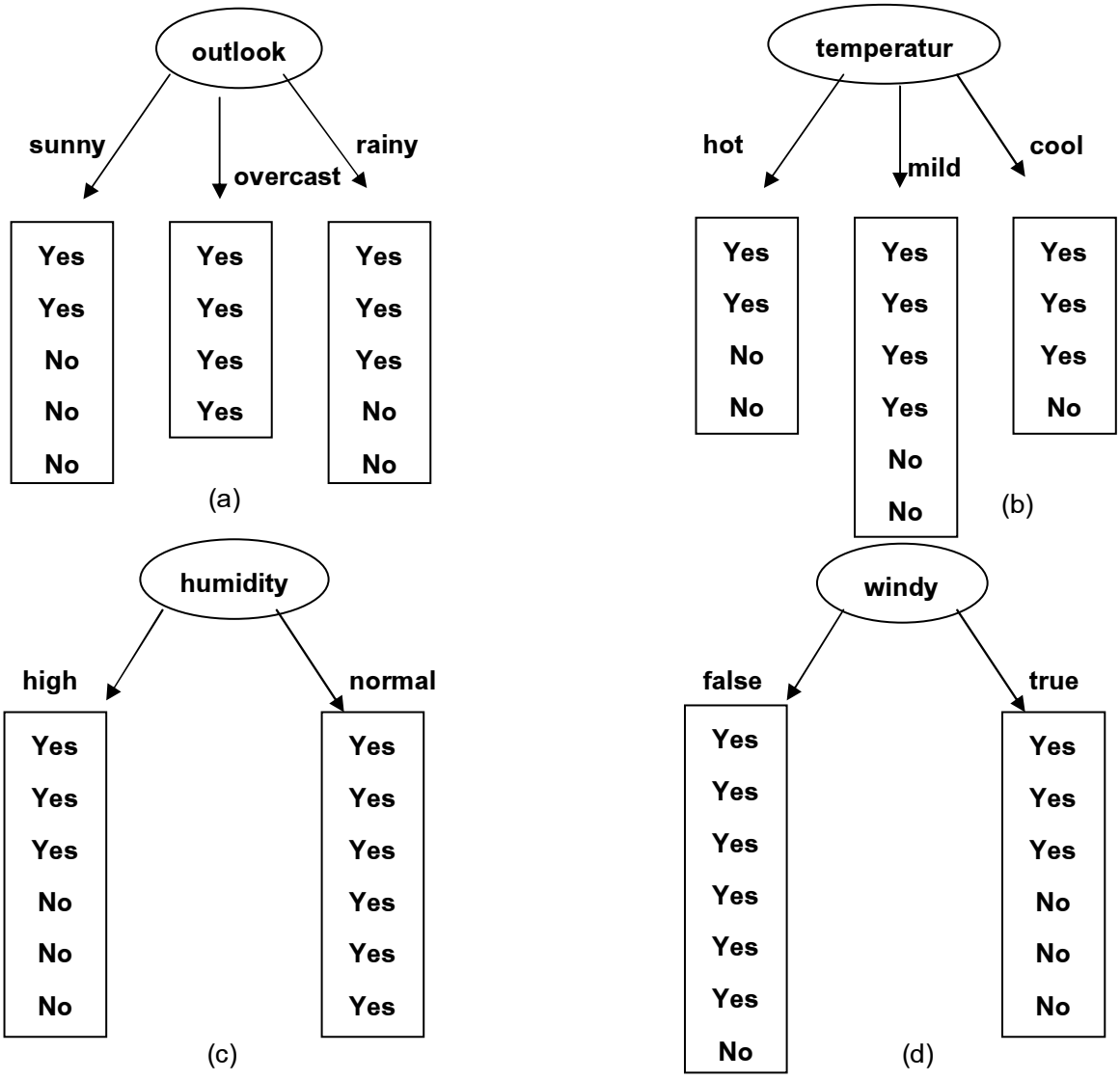
เมื่อได้โหนดรากแล้วทำการคำนวณค่า Information Gain ของแอทริบิวต์ที่เหลือจะได้ Information Gain ดังนี้

Gain (temperature) = 0.571

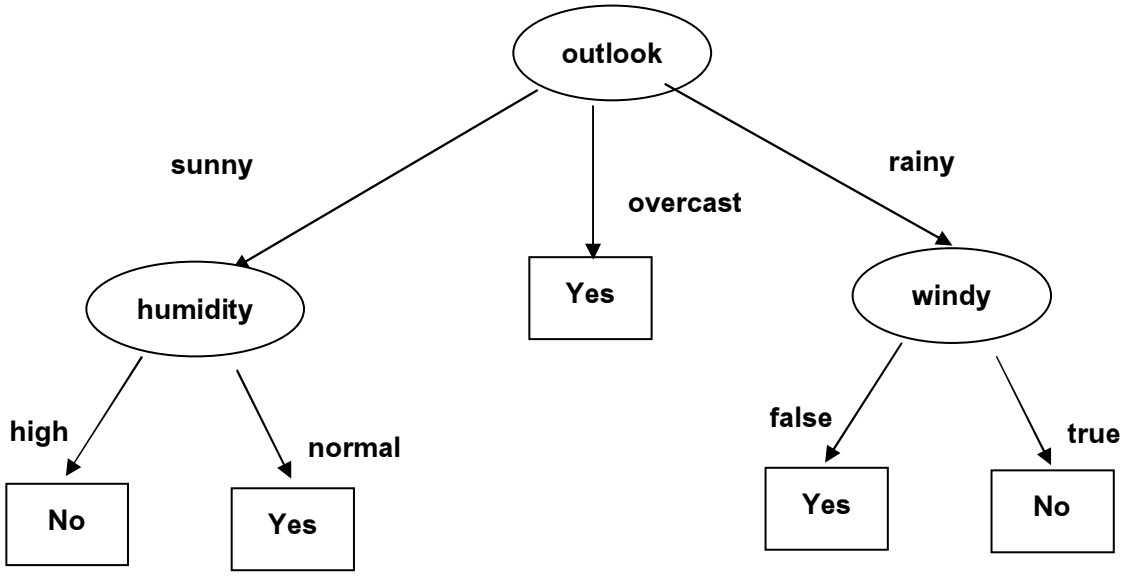
Gain (humidity) = 0.971

Gain (windy) = 0.020

จะเห็นได้ว่า humidity เป็นแอทริบิวต์ที่มีค่า Information Gain มากที่สุดจึง เลือกแอทริบิวต์นี้มาเป็นโหนด ทำการคำนวณไปเรื่อย ๆจนกระทั่งได้ต้นไม้ที่สมบูรณ์ดังภาพประกอบ 2.6



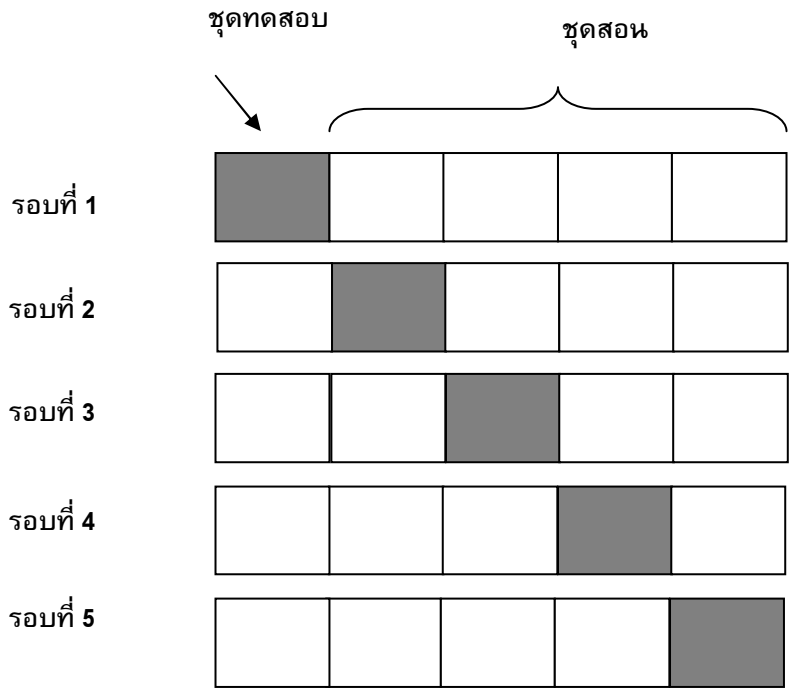
ภาพประกอบ 2.5 ตัวอย่างข้อมูลอากาศแยกตามแอทริบิวต์



ภาพประกอบ 2.6 ต้นไม้ตัดสินใจของข้อมูลอากาศ

2.9 Cross Validation

เป็นวิธีการในคาดการณ์ความผิดพลาดของโมเดล โดยการทำงานจะเป็นลักษณะ K-folds Cross Validation เป็นการแบ่งข้อมูลออกเป็น K ชุดๆเท่ากัน ในการทำงานเป็นชุดสอน (Train Set) และชุดทดสอบ (Test Set) โดยทำงานทั้งหมด K ครั้ง การทำงานรอบแรกข้อมูลชุดที่ 1 จะเป็นชุดทดสอบ ข้อมูลชุดที่เหลือ (K-1) จะเป็นชุดสอน และในรอบต่อไปข้อมูลชุดที่ 2 จะเป็นชุดทดสอบข้อมูลชุดที่เหลือจะเป็นชุดสอน จบครบทั้งหมด K รอบ ข้อดีของวิธีนี้คือ ข้อมูลทุกตัวจะมีโอกาสเป็นทั้งชุดสอนและชุดทดสอบและในการสอนแต่ละครั้งจะมีข้อมูลจากทุกคลาส การเลือกจำนวน Folds จะพิจารณาจากจำนวนตัวอย่าง หากจำนวนตัวอย่างมีจำนวนมากสามารถเลือกจำนวน Folds ที่เหมาะสมได้ดี ตัวอย่างการทำงาน 5-folds Cross Validation แสดงดังภาพประกอบ 2.7 แต่หากจำนวนตัวอย่างน้อยการใช้จำนวน Folds ไม่ควรจะมากเกินไปหรือทำงานแบบ Leave-one-out Cross Validation คือการทำ K-folds Cross validation เมื่อกำหนดให้ K มีค่าเท่ากับจำนวนข้อมูลทั้งหมด ดังนั้นในกรณีที่เรามีข้อมูล 10 ตัวอย่าง จะต้องทำทำงานแบบ 10-folds Cross Validation โดยที่ในแต่ละชุดจะมีตัวอย่างข้อมูล 1 ตัวอย่าง



ภาพประกอบ 2.7 K-folds Cross Validation

บทที่ 3

แบบจำลองการแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคคำบริบท

วิทยานิพนธ์นี้ได้ใช้คำบริบทมาสร้างแบบจำลองการแก้ปัญหาความกำกวมของคำและการเลือกแอทริบิวต์โดยใช้อัตราส่วนเกนและโครงข่ายประสาทเทียมแบบเรเดียลเบซิสฟังก์ชัน หรือ Word Sense Disambiguation and Attribute Selection (WSD_AS) Using Gain Ratio and RBF Neural Network มีหลักการทำงาน 4 ขั้นตอนคือ ขั้นตอนที่ 1 เตรียมคลังข้อความ ขั้นตอนที่ 2 สร้างแอทริบิวต์โดยใช้คำบริบท ขั้นตอนที่ 3 เลือกแอทริบิวต์ และขั้นตอนที่ 4 จำแนกความหมายของคำกำกวม รายละเอียดของแบบจำลองการแก้ปัญหาความกำกวมของคำ 4 ขั้นตอนดังภาพประกอบ 3.1

ขั้นตอนที่ 1 : เตรียมคลังข้อความ
1.1 เตรียมคลังข้อความ Senseval-2 ซึ่งอยู่ในรูปแบบของ XML
1.2 เลือกคำกำกวมที่ต้องการ
1.3 ตัดคำที่เป็น Stoplist ออกจากคลังข้อความ
ขั้นตอนที่ 2 : สร้างแอทริบิวต์โดยใช้คำบริบท
2.1 กำหนดขนาดหน้าต่างของคำบริบทเท่ากับ n
2.2 สร้างแอทริบิวต์โดยใช้คำบริบทซึ่งมี 3 กรณีคือ 2.2.1 ใช้บริบททางซ้ายเท่านั้น 2.2.2 ใช้บริบททางขวาเท่านั้น 2.2.3 ใช้บริบททั้งทางซ้ายและขวา
2.3 ใช้ โปรแกรม โปรแกรม NSP และ SenseTools แปลงข้อความให้อยู่ในรูปแบบ Feature Vectors (0 หรือ 1) ในรูปแบบไฟล์นามสกุล arff 2.3.1 สร้างจำนวน N-gram โดยใช้โปรแกรม NSP (ชื่อไฟล์ count.pl) คำสั่งที่ใช้ : count.pl --ngram n OUTPUT_FILE INPUT_FILE จะได้ : count.pl --ngram 1 output.txt window.xml 2.3.2 สร้าง Regular Expressions โดยใช้โปรแกรม SenseTools (ชื่อไฟล์ nsp2regex.pl) คำสั่งที่ใช้ : nsp2regex.pl INPUT_FILE >> REGEX_FILE จะได้ : nsp2regex.pl output.txt >> regex.txt

ภาพประกอบ 3.1 รายละเอียดแบบจำลองการแก้ปัญหาความกำกวมของคำโดยใช้คำบริบท

2.3.3 สร้าง Feature Vectors รูปแบบ arff โดยใช้โปรแกรม SenseTools (ชื่อไฟล์ xml2arff.pl) คำสั่งที่ใช้ : xml2arff.pl --training TRAIN_FILE --test TEST_FILE REGEX_FILE จะได้ : xml2arff.pl -- training art.n.xml -- test art.n.xml regex.txt
2.3.4 เปลี่ยนสัญลักษณ์ % เป็น ~ โดยใช้โปรแกรม SenseTools (ชื่อไฟล์ tilde.pl) คำสั่งที่ใช้ : tilde.pl SOURCE > > OUTPUT จะได้ : tilde.pl art.n.xml >> art.arff
ขั้นตอนที่ 3 : เลือกแอทริบิวต์
3.1 เลือกชนิดตัวกรองแทริบิวซึ่งมี 2 กรณีคือ 3.1.1 InfoGainAttributeEval 3.1.2 GainRatioAttributeEval
3.2 เลือกจำนวนแอททริบิที่ต้องการกรองเช่น 40 50 หรือ 60 แอทริบิวต์ (Optional)
ขั้นตอนที่ 4 : จำแนกความหมาย
4.1 เลือกวิธีการจำแนกความหมาย 4.1.1 แบ่งกลุ่มเป็น 2 ความหมายเท่านั้น 4.1.2 แบ่งกลุ่มตามจำนวนความหมายทั้งหมด
4.2 เลือกอัลกอริทึมที่ใช้จำแนกความหมาย 4.2.1 RBFNetwork 4.2.2 ID3
4.3 คำนวณค่าความถูกต้องในการจำแนกความหมายของคำกำกวม

ภาพประกอบ 3.1 รายละเอียดแบบจำลองการแก้ปัญหาความกำกวมของคำ
โดยใช้คำบริบท (ต่อ)

3.1 ขั้นตอนที่ 1: เตรียมคลังข้อความ

คลังข้อความ Senseval-2 เป็นคลังข้อความมาตรฐานในการวัดประสิทธิภาพของการแก้ปัญหาความกำกวมของคำ ในคลังข้อความ Senseval-2 ประกอบด้วยคลังข้อความย่อยหลายภาษาเช่น ภาษาอิตาลี ภาษาญี่ปุ่น และภาษาอังกฤษ เป็นต้น ในวิทยานิพนธ์นี้ได้เลือกใช้คลังข้อความที่เป็นภาษาอังกฤษโดยมีรายละเอียดดังนี้

ขั้นตอนที่ 1.1 เตรียมคลังข้อความ Senseval-2 ในรูปแบบของ XML ที่เป็นคลังข้อความภาษาอังกฤษคือ eng-lex-sample ประกอบด้วยคำที่มีความหมายกำกวมเช่น คำว่า art อาจหมายถึง ทักษะ ผลิตภัณฑ์ที่เป็นงานศิลปะ หรือ การสร้างงานศิลปะ เป็นต้น ความหมาย

ของคำกำกวมแต่ละความหมายแสดงเป็นสัญลักษณ์ของความหมายที่กำหนดโดย WordNet ตัวอย่างของคลังข้อความ Senseval-2 ดังภาพประกอบ 3.2 โดยคำกำกวมหนึ่งคำประกอบด้วยสองไฟล์คือ ไฟล์ที่มีนามสกุลเป็น .xml และ .count ลักษณะของไฟล์นามสกุล .xml เป็นไฟล์ที่มีรูปแบบเป็น XML ใช้ในการสอนและทดสอบสำหรับวิทยานิพนธ์นี้ และลักษณะของไฟล์นามสกุล .count เป็นไฟล์ที่มีรูปแบบเป็นข้อความปกติ ใช้ในการสร้างคำบริบทสำหรับวิทยานิพนธ์นี้ ตัวอย่างไฟล์นามสกุล .xml และไฟล์นามสกุล .count ของคำกำกวมคำว่า art ดังภาพประกอบ 3.3 และภาพประกอบ 3.4 ตามลำดับ

Senseval-2 ภาษาอังกฤษ

<?xml version="1.0" encoding="iso-8859-1" ?>

<!DOCTYPE corpus SYSTEM "lexical-sample.dtd">

<corpus lang=**'english'**>

<lexelt item="art.n">

<instance id="art.40002" docsrc="bnc_A70_2636">

ความหมายของคำกำกวม art คือ 1:06:00::

<answer instance="art.40002" senseid="**art%1:06:00::"/>**

เริ่มต้นประโยค

<context>

Leeds is well-equipped for sports, with 21 golf courses and 22 sports and leisure centres, but if all this action leaves you feeling in need of a rest, you can always take yourself off to the theatre. Leeds has four to choose from. Most famous is the Leeds City Varieties, one of the oldest music halls in the country and home of BBC TV's [hi]The Good Old Days [/hi].There's also the Grand Theatre, which hosts touring companies and is the permanent home of Opera North. [/p] [p]

One of Yorkshire's famous sayings is &bquo;Where there's muck, there's brass&equo;. And, while there may not be a lot of muck any more, there is still plenty of brass. [/p] [p] For, when it comes down to it, there's nothing to beat a brass band. There's always one to be heard somewhere during the summer — in the

ภาพประกอบ 3.2 คลังข้อความภาษาอังกฤษของ Senseval-2

แสดงคำกำกวม art

piazza in front of the **<head>art</head>**gallery and Town Hall or in a park.

</context>

</instance>

</lexelt>

</corpus>

ภาพประกอบ 3.2 คลังข้อความภาษาอังกฤษของ Senseval-2 (ต่อ)

รูปแบบไฟล์ XML

<? xml version="1.0" encoding="iso-8859-1" ?>

<!DOCTYPE corpus SYSTEM "lexical-sample.dtd">

<corpus lang='english'>

<lexelt item="art.n">

<instance id="art.40002" docsrc="bnc_A70_2636">

<answer instance="art.40002" senseid="art%1:06:00::"/>

<context>

Leeds is well <-> equipped for sports , with 21 golf courses and 22 sports and leisure centres , but if all this action leaves you feeling in need of a rest , you can always take yourself off to the theatre . Leeds has four to choose from . Most famous is the Leeds City Varieties , one of the oldest music halls in the country and home of BBC TV <'> s <[> hi <]> The Good Old Days <[/> hi <]> . There <'> s also the Grand Theatre , which hosts touring companies and is the permanent home of Opera North . <[/> p <] [> p <]> One of Yorkshire <'> s famous sayings is <&> bquo ; Where there <'> s muck , there <'> s brass <&> equo ; . There <'> s always one to be heard somewhere during the summer <&> mdash ; in the piazza in front of the **<<> head <>> art <</> head <>>** gallery and Town Hall or in a park .

</context>

</instance>

</lexelt>

</corpus>

ภาพประกอบ 3.3 ไฟล์นามสกุล .xml

รูปแบบทั่วไปของประโยค

Leeds is well <-> equipped for sports , with 21 golf courses and 22 sports and leisure centres , but if all this action leaves you feeling in need of a rest , you can always take yourself off to the theatre . Leeds has four to choose from . Most famous is the Leeds City Varieties , one of the oldest music halls in the country and home of BBC TV <'> s <[> hi <]> The Good Old Days <[/> hi <]> . There <'> s also the Grand Theatre , which hosts touring companies and is the permanent home of Opera North . <[/> p <] [> p <]> One of Yorkshire <'> s famous sayings is <&> bquo ; Where there <'> s muck , there <'> s brass <&> equo ; . And , while there may not be a lot of muck any more , there is still plenty of brass . <[/> p <] [> p <]> For , when it comes down to it , there <'> s nothing to beat a brass band . There <'> s always one to be heard somewhere during the summer <&> mdash ; in the piazza in front of the **<<> head <>> art <</> head <>>** gallery and Town Hall or in a park .

ภาพประกอบ 3.4 ไฟล์นามสกุล .count

ในกรณีที่ข้อความที่ต้องการทดสอบไม่อยู่ในรูปแบบของ Senseval-2 จะต้องแปลงข้อความดังกล่าวให้อยู่ในรูปแบบ Senseval-2 ก่อนโดยใช้โปรแกรม OMtoSVAL2 (Pedersen and Purandare, 2002) โดยมีคำสั่งการทำงานดังนี้

คำสั่งที่ใช้ : omwe2sval.pl TAG_FILE INSTANCE_FILE

โดยที่ TAG_FILE เป็นไฟล์ของความหมายของคำกำกวมแต่ละคำดังภาพประกอบ 3.5 ในคอลัมน์แรกคือดัชนีของคำ (Index) คอลัมน์ที่สองคือ ความหมายของคำ และ INSTANCE_FILE คือไฟล์ของประโยคที่มีคำกำกวม ดังภาพประกอบ 3.6 โดยไฟล์ INSTANCE_FILE จะต้องกำหนดชนิดของคำแต่ละคำไว้หลังจากคำเหล่านั้นซึ่งคั่นด้วยเครื่องหมาย / เช่น are/VBP หมายถึงคำว่า art มีชนิดของคำเป็นคำกริยา (VBP) และต้องระบุคำกำกวมและตำแหน่งของคำกำกวมในประโยคหลังเครื่องหมาย ? ด้วยเช่น acts ? 11 หมายถึงคำกำกวม “acts” อยู่ในตำแหน่งคำที่ 11 ของประโยค

```
act.n.tb.138    act%1:10:02::
```

ภาพประกอบ 3.5 ตัวอย่าง TAG_FILE

```
act.n.tb.138 acts ? 11 Under/IN current/JJ law/NN ,/, such/JJ
             suspects/NNS are/VBP immune/JJ from/IN prosecution/NN for/IN
             acts/NNS committed/VBN while/IN not/RB British/JJ citizens/NNS ./.
```

ภาพประกอบ 3.6 ตัวอย่าง INSTANCE_FILE

ขั้นตอนที่ 1.2 เลือกคำกำกวมที่ต้องการจำแนกความหมาย

ขั้นตอนที่ 1.3 ตัดคำที่เป็น Stoplist (จากภาพประกอบ 2.2) ออกจากไฟล์ .count เนื่องจากคำที่เป็น Stoplist คือคำที่ทำให้ไฟล์มีขนาดใหญ่และเป็นคำฟุ่มเฟือยเพราะคำเหล่านี้ไม่ได้นำมาวิเคราะห์หาความหมายของคำ และมีผลทำให้ประสิทธิภาพในการแก้ปัญหาความกำกวมลดลง ในขั้นตอนนี้จะต้องตัดตัวอักษรที่เป็นสัญลักษณ์ต่างๆที่ไม่มีความหมายออก เช่น สัญลักษณ์ [ ] < > ( ) $ % # @ * ^ . , ; & = \ ' - / ! ? : ` \ " " , ตัวอย่างประโยค เช่น :

"There ' s alway one to be heard somewhere during the summer; in the piazza in front of the **art** gallery and Town Hall or in a park ."

คำว่า art เป็นตัวหนาเป็นคำที่มีความหมายกำกวมส่วนคำที่เป็น Stoplist จะขีดเส้นใต้ เมื่อตัดคำที่เป็น Stoplist และสัญลักษณ์ต่างๆออกแล้วจะได้ประโยคดังนี้

"heard summer piazza front **art** gallery Town Hall park"

3.2 ขั้นตอนที่ 2: สร้างแอทริบิวต์โดยใช้คำบริบท

ขั้นตอนที่ 2.1 เลือกขนาดหน้าต่างของคำบริบท ขนาดหน้าต่างที่ใช้ในการทดลองคือ n เมื่อ n=1 หมายถึงเพิ่มคำอีก 1 คำมาพิจารณาเป็นค่า แอทริบิวต์

ขั้นตอนที่ 2.2 สร้างแอทริบิวต์โดยใช้คำบริบท กำหนดให้ **W** เป็นคำที่มีความหมายกำกวมและ n คือ ขนาดหน้าต่าง ดังนั้น W_{+1}, W_{+2}, W_{+3},..., W_{+n} เป็นบริบททางขวา

n ตัว และ $W_{-n}...W_{-3}$, W_{-2}, W_{-1} เป็นบริบททางซ้าย n ตัว ตัวอย่างเช่น ถ้าขนาดหน้าต่างเป็น n=1 ในการสร้างแอทริบิวต์โดยใช้คำบริบท 3 กรณีแสดงดังภาพประกอบ 3.7

กรณีที่ 1 ใช้คำบริบททางซ้ายอย่างเดียวจะได้คำบริบทรวมทั้งคำกำกวมเป็น W_{-1} **W**

กรณีที่ 2 ใช้คำบริบททางขวาอย่างเดียวจะได้คำบริบทรวมทั้งคำกำกวมเป็น **W** W_{+1}

กรณีที่ 3 ใช้คำบริบททั้งทางซ้ายและขวา จะได้คำบริบทรวมทั้งคำกำกวมเป็น W_{-1} **W** W_{+1}

ในทำนองเดียวกันถ้าขนาดหน้าต่างเป็น 2 (n=2) คำบริบทที่ได้ต้องเพิ่มคำทางซ้ายหรือขวา 2 คำจะได้คำบริบทรวมทั้งคำกำกวมกรณีที่ 1 คือ W_{-2}, W_{-1} **W** กรณีที่ 2 คือ **W** W_{+1} W_{+2} และกรณีที่ 3 คือ W_{-2} W_{-1} **W** W_{+1} W_{+2} ตามลำดับ จากประโยค “There heard summer piazza front **art** gallery Town Hall park.” เมื่อตัดประโยคให้มีคำบริบททั้งสามแบบโดยกำหนดให้ n=3 กรณีที่1 จะได้ “summer piazza front **art** ” โดยจะตัด “heard” ทิ้งไป กรณีที่ 2 จะได้ “**art** gallery Town Hall” และกรณีที่ 3 จะได้ “summer piazza front **art** gallery Town Hall ” ดังภาพประกอบ 3.8

กรณีที่ 1 :	W_{-3}	W_{-2}	W_{-1}	**W**			
กรณีที่ 2 :				**W**	W_{+1}	W_{+2}	W_{+3}
กรณีที่ 3 :	W_{-3}	W_{-2}	W_{-1}	**W**	W_{+1}	W_{+2}	W_{+3}

ภาพประกอบ 3.7 แสดงรูปแบบของคำบริบท 3 แบบ

ตัวอย่างกรณีที่ 1 :	W_{-3}	W_{-2}	W_{-1}	**W**			
	summer	piazza	front	**art**			
ตัวอย่างกรณีที่ 2 :				**W**	W_{+1}	W_{+2}	W_{+3}
				art	gallery	Town	Hall
ตัวอย่างกรณีที่ 3 :	W_{-3}	W_{-2}	W_{-1}	**W**	W_{+1}	W_{+2}	W_{+3}
	summer	piazza	front	**art**	gallery	Town	Hall

ภาพประกอบ 3.8 ตัวอย่างคำหลังจากตัดประโยคให้มีคำบริบท 3 แบบ

ขั้นตอนที่ 2.3 ใช้โปรแกรม NSP และ SenseTools แปลงข้อความดังกล่าวให้อยู่ในรูปแบบ Feature Vectors ให้มีค่าเป็น 0 หรือ1 ไฟล์ที่ได้เป็นนามสกุล arff ขั้นตอนการแปลงข้อความด้วย SenseTools และ NSP มี 4 ขั้นตอนย่อยดังภาพประกอบ 3.1 (ขั้นตอน 2.3)

2.3.1 สร้างคำให้เป็นจำนวน N-gram โดยใช้ไฟล์ count.pl โดยไฟล์อินพุทเป็นไฟล์ที่ได้จากการตัดประโยค (จากขั้นตอน 2.2) ตัวอย่างเมื่อเลือกใช้คำบริบททั้งทางซ้ายและขวา (จากกรณีที่ 3 ในดังภาพประกอบ 3.8) คือ “summer piazza front **art** gallery Town Hall” สมมติให้เป็นไฟล์ชื่อ window.xml ดังนั้นคำสั่งที่ใช้ในการทำงานจะได้

คำสั่งที่ใช้ : count.pl --ngram n OUTPUT_FILE INPUT_FILE
จะได้ : count.pl --ngram 1 output.txt window.xml

ผลลัพธ์ที่ได้จากขั้นตอนนี้ (output.txt) แสดงดังภาพประกอบ 3.9 คำแต่ละคำถูกสร้างเป็น N-gram และแสดงจำนวนความถี่ของแต่ละคำ เช่น “Town<>1” หมายถึง “Town” มีความถี่เท่ากับ 1 เป็นต้น

```
7
Town<>1
gallery<>1
front<>1
piazza<>1
summer<>1
art<>1
Hall<>1
```

ภาพประกอบ 3.9 ตัวอย่างคำที่สร้างเป็น 1-Gram (output.txt : ไฟล์เอาท์พุทของ count.pl)

2.3.2 สร้าง Regular Expressions ของคำแต่ละคำเพื่อใช้ในการสร้างแอทริบิวต์โดยใช้ไฟล์ nsp2regex.pl โดยไฟล์อินพุทในขั้นตอนนี้คือ output.txt จากภาพประกอบ 3.9 ดังนั้นคำสั่งที่ใช้ในการทำงานจะได้

คำสั่งที่ใช้ : nsp2regex.pl INPUT_FILE >> REGEX_FILE
จะได้ : nsp2regex.pl output.txt >> regex.txt

ผลลัพธ์ของ Regular Expressions แต่ละคำคือไฟล์ regex.txt ดังภาพประกอบ 3.10

```
/\s(<[^>]*>)*Town(<[^>]*>)*\s/ @name = Town
/\s(<[^>]*>)*gallery(<[^>]*>)*\s/ @name = gallery
/\s(<[^>]*>)*front(<[^>]*>)*\s/ @name = front
/\s(<[^>]*>)*piazza(<[^>]*>)*\s/ @name = piazza
/\s(<[^>]*>)*summer(<[^>]*>)*\s/ @name = summer
/\s(<[^>]*>)*art(<[^>]*>)*\s/ @name = art
/\s(<[^>]*>)*Hall(<[^>]*>)*\s/ @name = Hall s/ @name = capacity
```

ภาพประกอบ 3.10 Regular Expressions (regex.txt: ไฟล์เอาท์พุทของ nsp2regex.pl)

2.3.3 สร้าง Feature Vectors ของตัวอย่าง (Instance) ให้อยู่ในรูปแบบเป็น 0 หรือ 1 โดยใช้ไฟล์ xml2arff.pl โดยไฟล์อินพุทประกอบด้วย 2 ไฟล์คือ ไฟล์นามสกุล .xml ของคำกำกวม และไฟล์ Regular Expressions คือ regex.txt จากขั้นตอน (2.3.2) ดังนั้นคำสั่งที่ใช้ในการทำงานจะได้

```
คำสั่งที่ใช้ : xml2arff.pl --training TRAIN_FILE --test TEST_FILE  REGEX_FILE
จะได้      : xml2arff.pl -- training art.n.xml -- test art.n.xml regex.txt
```

หลังจากขั้นตอนนี้จะได้ดังภาพประกอบ 3.11 ไฟล์เอาท์พุทที่ได้คือ art.n.xml.arff ส่วนแรกจะอธิบายถึง relation ของแอทริบิวต์ของคำ 7 คำ บรรทัดที่สองและบรรทัดอื่นๆที่ขึ้นต้นด้วย @attribute และลงท้ายด้วย {0,1} หมายถึงแอทริบิวต์ โดยเริ่มจากแอทริบิวต์ที่ 0 ในที่นี้คือแอทริบิวต์ “@attribute 'Town' {0,1} “ ตัวอย่างแอทริบิวต์เช่น @attribute 'front' {0,1} หมายถึงแอทริบิวต์ชื่อ front มีค่าที่เป็นไปได้สองค่าคือ 0 และ 1 ส่วนแอทริบิวต์สุดท้ายคือ @attribute 'senseclass' เป็นแอทริบิวต์ที่ประกอบด้วยคลาสที่เป็นไปได้ทั้งหมดในที่นี้จะแทนด้วยความหมายที่เป็นไปได้ทั้งหมดของคำกำกวม เช่น art มีทั้งหมด 3 ความหมายคือ art~1:06:00:: art~1:09:00:: และ art~1:04:00:: บรรทัดต่อไปคือ @data เป็นจุดเริ่มต้นของข้อมูลทั้งหมดที่ได้จากการแปลงให้อยู่ในรูปแบบ Feature Vector ซึ่งถ้าค่าของแอทริบิวต์มีค่าเป็น 1 หมายถึงแอทริบิวต์นั้นมีคำที่ต้องการอยู่ เช่น {0,0,0,0,0,1,0, art%1:04:00::} หมายถึง แอทริบิวต์ที่ 5 คือแอทริบิวต์ ‘art’ มีคำว่า “art” อยู่

```
@relation 'RELATION'
@attribute 'Town' {0,1}
@attribute 'gallery' {0,1}
@attribute 'front' {0,1}
@attribute 'piazza' {0,1}
@attribute 'summer' {0,1}
@attribute 'art' {0,1}
@attribute 'Hall' {0,1}
@attribute 'senseclass' {art%1:06:00::, art%1:09:00::, art%1:04:00::}
@data
{0,0,0,0 ,1,1,0,art%1:06:00::}
{1,1,1,1,1,1,1, art%1:06:00::}
{0,0,0,0,0,1,0, art%1:04:00::}
{0,0,0,0,0,1,0, art%1:09:00::}
```

ภาพประกอบ 3.11 รูปแบบข้อมูลที่แปลงเป็น Feature Vector
(art.n.xml.arff : ไฟล์เอาท์พุทของ xml2arff.pl)

2.3.4 เปลี่ยนสัญลักษณ์ % ให้เป็น ~โดยใช้ไฟล์ tilde.pl เพราะสัญลักษณ์ % ในโปรแกรม WEKA หมายถึงคอมเมนท์ (comment) ในขั้นตอนนี้ไฟล์อินพุทคือ art.n.xml.arff ดังนั้นคำสั่งที่ใช้ในการทำงานจะได้

```
คำสั่งที่ใช้ : tilde.pl SOURCE >> OUTPUT
จะได้       : tilde.pl art.n.xml  >> art.arff
```

หลังจากขั้นตอนนี้ผลลัพธ์ที่ได้เป็นรูปแบบ arff ที่สมบูรณ์ในไฟล์เอาท์พุทที่ได้คือ art.arff ดังภาพประกอบ 3.12

```
@relation 'RELATION'
@attribute 'Town' {0,1}
@attribute 'gallery' {0,1}
@attribute 'front' {0,1}
@attribute 'piazza' {0,1}
@attribute 'summer' {0,1}
@attribute 'art' {0,1}
@attribute 'Hall' {0,1}
@attribute 'senseclass' {art~1:06:00::, art~1:09:00::, art~1:04:00::}
@data
{0,0,0,0,1,1,0, art~1:06:00::}
{1,1,1,1,1,1,1, art~1:06:00::}
{0,0,0,0,0,1,0, art~1:04:00::}
{0,0,0,0,0,1,0,  art~1:09:00::}
```

ภาพประกอบ 3.12 การเปลี่ยนสัญลักษณ์ % ให้เป็น ~ (art.arff: ไฟล์เอาท์พุทของ tilde.pl)

3.3 ขั้นตอนที่ 3: เลือกแอทริบิวต์

ขั้นตอนการเลือกแอทริบิวต์เป็นขั้นตอนการเลือกแอทริบิวต์ที่มีความสัมพันธ์กัน โดยแอทริบิวต์ที่เหลือจะถูกตัดออกโดยมีรายละเอียดดังนี้

ขั้นตอนที่ 3.1 เลือกชนิดตัวกรองแอทริบิวต์ซึ่งมี 2 กรณีคือ InfoGainAttributeEval และ GainRatioAttributeEval เนื่องจากตัวกรอง 2 เทคนิคนี้เป็นเทคนิคที่ง่ายและรวดเร็ว

ขั้นตอนที่ 3.2 เลือกจำนวนแอทริบิวต์ที่ต้องการโดยการใส่จำนวน แอทริบิวต์ที่มีค่าน้อยกว่าจำนวนแอทริบิวต์ทั้งหมด (Attribute Selection) ในการทดลองนี้ใช้จำนวนแอทริบิวต์เป็น 40 50 60 100 150 200 300 และ 500 แอทริบิวต์ ขึ้นอยู่กับจำนวนแอทริบิวต์ที่มากที่สุดที่สามารถกรองได้ เช่น จำนวนแอทริบิวต์ที่ไม่ได้กรองมี 720 แอทริบิวต์ สามารถกรองได้มากสุด 500 แอทริบิวต์ หากจำนวนแอทริบิวต์มีจำนวน 452 จำนวนแอทริบิวต์มากสุดที่สามารถกรองได้คือ 300 แอทริบิวต์ แต่ถ้าเลือกแอทริบิวต์เท่ากับจำนวนแอทริบิวต์ที่มีอยู่จะหมายถึงไม่ทำการกรองแอทริบิวต์ (Non-Attribute Selection)

3.4 ขั้นตอนที่ 4: จำแนกความหมาย

การจำแนกความหมาย (Classification) ของคำเป็นขั้นตอนในการเลือกความหมายที่ถูกต้องของคำกำกวมจากความหมายทั้งหมดโดยใช้โปรแกรม WEKA

ขั้นตอนที่ 4.1 เลือกวิธีการจำแนกความหมายโดยแบ่งกลุ่มเป็น 2 ความหมายเท่านั้นหรือแบ่งกลุ่มตามจำนวนความหมายทั้งหมด การแบ่งกลุ่มเป็น 2 ความหมาย คือ ถ้าคำมี 3 ความหมายคือ X Y และ Z จะมีการพิจารณา 3 กรณีคือ กรณีที่ 1 สนใจเฉพาะความหมาย X จะได้ X = คลาส YES และ Y Z = คลาส NO กรณีที่ 2 สนใจเฉพาะความหมาย Y จะได้ Y = คลาส YES และ X Z = คลาส NO กรณีที่ 3 สนใจเฉพาะความหมาย Z จะได้ Z = คลาส YES และ X Y = คลาส NO ดังตารางที่ 3.1 คอลัมน์ที่ 1 ส่วนการแบ่งกลุ่มตามจำนวนความหมายทั้งหมด เป็นกรณีที่กำหนดให้มีจำนวนความหมายตามจริงโดยที่จำนวนความหมายของคำกำกวมมากกว่า 2 ความหมาย ตัวอย่างเช่น ถ้าคำมี 3 ความหมาย (X Y Z) กำหนดให้ความหมาย X เป็น คลาส X ความหมาย Y เป็น คลาส Y และความหมาย Z เป็น คลาส Z ดังตารางที่ 3.1 คอลัมน์ที่ 2

ตารางที่ 3.1 จำแนกความหมายโดยแบ่งกลุ่มเป็น 2 ความหมายและแบ่งกลุ่มตามจำนวนความหมายทั้งหมด

แบ่งกลุ่มเป็น 2 ความหมาย	**แบ่งกลุ่มตามจำนวนความหมายทั้งหมด**
กรณีที่ 1 สนใจความหมาย X	ความหมาย X = คลาส X
ความหมาย X = คลาส YES	ความหมาย Y = คลาส Y
ความหมาย Y = คลาส NO	ความหมาย Z = คลาส Z
ความหมาย Z = คลาส NO	
กรณีที่ 2 สนใจความหมาย Y	
ความหมาย X = คลาส NO	
ความหมาย Y = คลาส YES	
ความหมาย Z = คลาส NO	
กรณีที่ 3 สนใจความหมาย Z	
ความหมาย X = คลาส NO	
ความหมาย Y = คลาส NO	
ความหมาย Z = คลาส YES	

ขั้นตอนที่ 4.2 จำแนกความหมายของคำกำกวมโดยเลือกอัลกอริทึมในการจำแนกความหมาย 2 อัลกอริทึมคือ RBFNetwork และ ID3

ขั้นตอนที่ 4.3 คำนวณค่าความถูกต้องในการจำแนกความหมายของคำกำกวม โดยค่าความถูกต้องในการจำแนกนี้สามารถวิเคราะห์ได้จาก Confusion Matrix ดังตัวอย่างภาพประกอบ 3.13

Correctly Classified Instances	171	82.2115 %
Incorrectly Classified Instances	37	17.7885 %
	=== Confusion Matrix ===	
	a b c <-- classified as	
	98 0 0 | a = 1:06:00::	
	13 35 4 | b = 1:09:00::	
	18 2 38 | c = 1:04:00::	

ภาพประกอบ 3.13 Confusion Matrix

จากภาพประกอบ 3.13 แสดง Confusion Matrix ของการจำแนกความหมายจำแนกถูกต้อง 82.2115 % และจำแนกผิด 17.7885 % ในแต่ละความหมายมีรายละเอียดดังนี้

ความหมาย 1:06:00:: มีจำนวนตัวอย่างทั้งหมด 98+0+0 =98 ตัวอย่าง ผลของการจำแนกความหมาย จำแนกถูก ทั้งหมด 98 ตัวอย่าง

ความหมาย 1:09:00:: มีจำนวนตัวอย่างทั้งหมด 13+35+4= 52 ตัวอย่าง ผลของการจำแนกความหมาย จำแนกถูก ทั้งหมด 35 ตัวอย่าง จำแนกผิดเป็นความหมาย 1:06:00:: ทั้งหมด 13 ตัวอย่างและจำแนกผิดเป็นความหมาย 1:04:00:: ทั้งหมด 4 ตัวอย่าง

ความหมาย 1:04:00:: มีจำนวนตัวอย่างทั้งหมด 18+2+38=58 ตัวอย่าง ผลของการจำแนกความหมาย จำแนกถูก ทั้งหมด 35 ตัวอย่าง จำแนกผิดเป็นความหมาย 1:06:00:: ทั้งหมด 18 ตัวอย่างและจำแนกผิดเป็นความหมาย 1:09:00:: ทั้งหมด 2 ตัวอย่าง

ค่าความถูกต้องคิดเป็นเปอร์เซ็นต์และ Confusion Matrix เมื่อจำแนกถูก 100% แสดงได้ดังภาพประกอบ 3.14

	True	False
True	True Positive	False Positive
False	False Negative	True Negative

$$\text{accuracy}(\%) = \left(\frac{\text{number of True Positive + number of True Negative}}{\text{number of True Positive + False Positive + False Negative + True Negative}} \right) \times 100$$

```
=== Confusion Matrix ===
 a  b  c   <-- classified as
98  0  0 |  a = 1:06:00::
 0 52  0 |  b = 1:09:00::
 0  0 58 |  c = 1:04:00::
```

Accuracy ที่จำแนกถูก 100% จะได้ = $\left(\frac{98 + 52 + 58}{208} \right) \times 100$

ภาพประกอบ 3.14 Confusion Matrix ที่จำแนกถูกต้อง 100%

บทที่ 4

โปรแกรมการแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคคำบริบท

เพื่อให้ผู้ใช้สามารถใช้โปรแกรมการแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคคำบริบทได้ง่าย จึงได้ออกแบบส่วนการติดต่อกับผู้ใช้ให้อยู่ในรูปแบบที่ใช้งานง่ายด้วย Graphic User Interface ในการทำงานของโปรแกรมจะอธิบายด้วยผังการทำงานของโปรแกรม ส่วนประกอบของโปรแกรม ผลการทำงานของโปรแกรมและเครื่องมือที่ใช้ในการพัฒนาโปรแกรม

4.1 ผังการทำงานของโปรแกรม

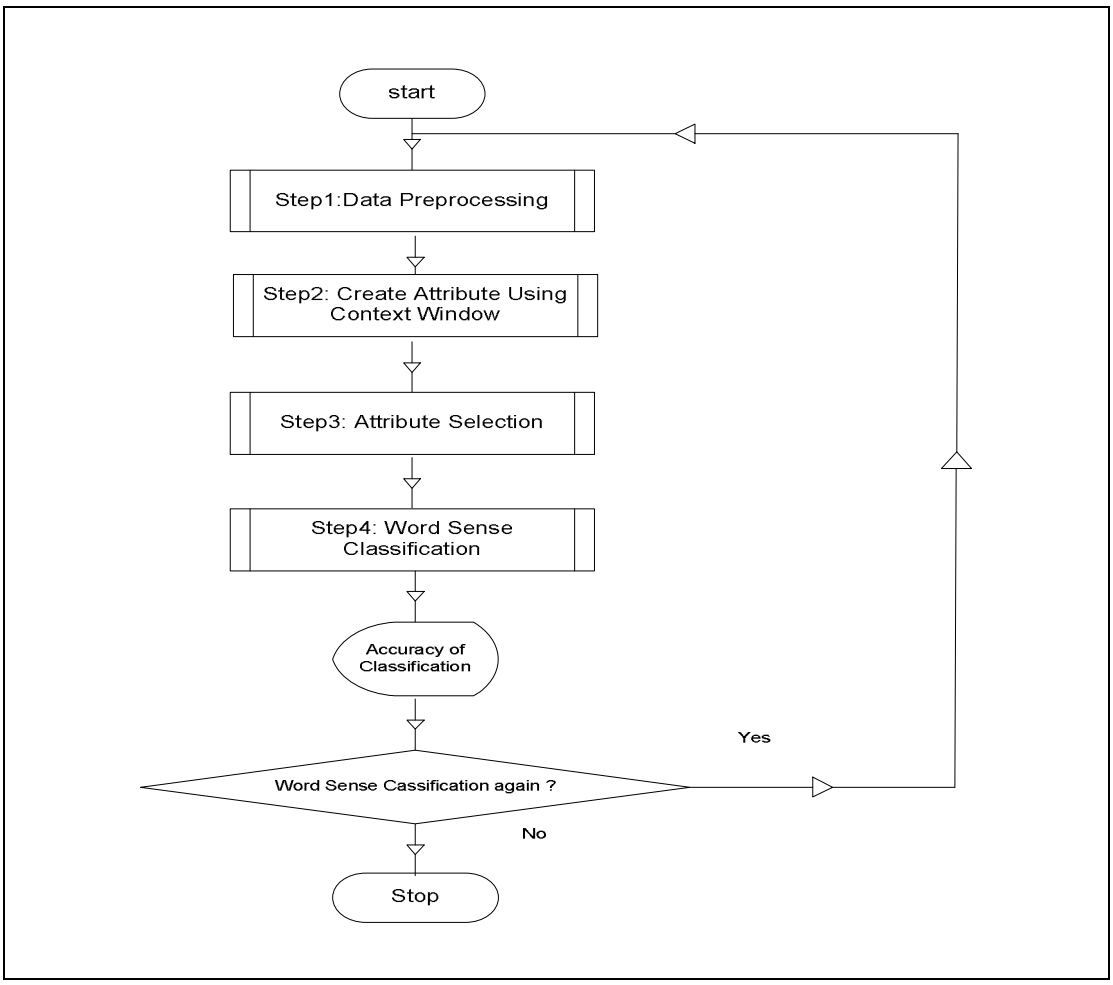
4.1.1 ผังงานโปรแกรมหลักของโปรแกรมการแก้ปัญหาควมกำกวมของคำแสดงดังภาพประกอบ 4.1 โดยขั้นตอนการทำงานทั้งหมดของโปรแกรมประกอบด้วย 4 ขั้นตอนคือ 1) เตรียมคลังข้อความ (Data Preprocessing) 2) สร้างแอทริบิวต์โดยใช้คำบริบท (Create Attribute Using Context Window) 3) การเลือกแอทริบิวต์ (Attribute Selection) และ4) การจำแนกความหมาย (Word Sense Classification) เมื่อทำงานทั้งหมด 4 ขั้นตอนจะแสดงค่าความถูกต้องในการจำแนกความหมาย หลังจากนั้นสามารถจำแนกความหมายของคำกำกวมอื่นๆที่ต้องการได้อีก

4.1.2 ผังงานโปรแกรมย่อยของ Step1: Data Preprocessing แสดงดังภาพประกอบ 4.2 ประกอบด้วยการทำงานในส่วนของการตัดคำที่เป็น Stoplist และสัญลักษณ์ต่างๆออก

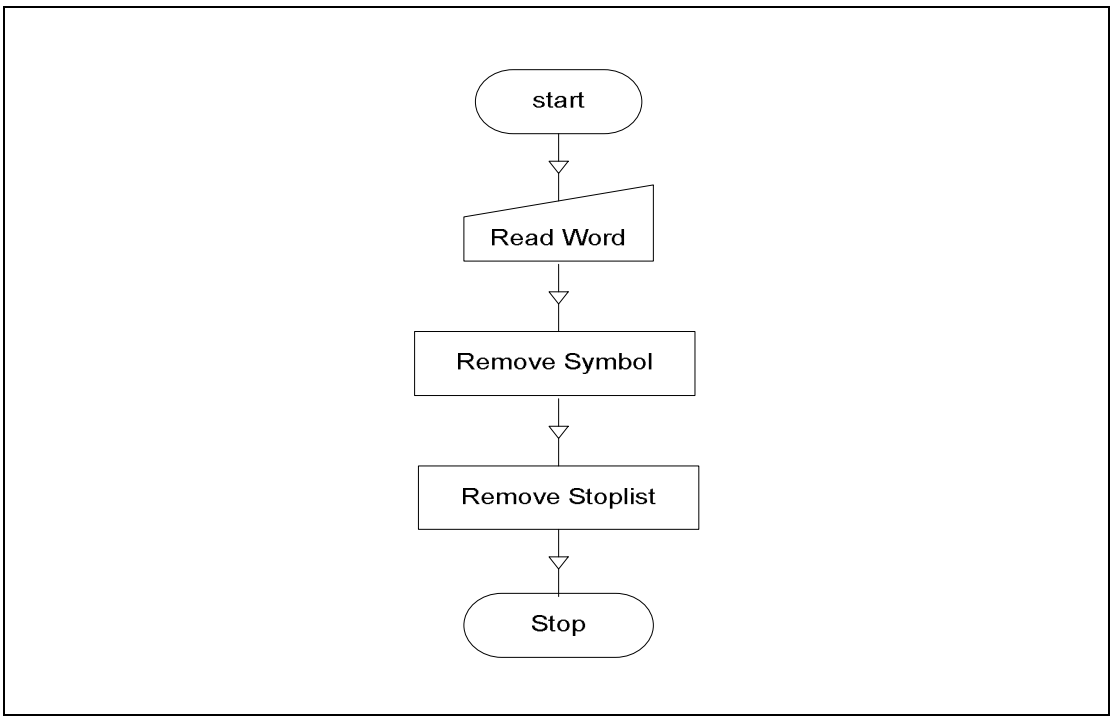
4.1.3 ผังงานโปรแกรมย่อยของ Step2: Create Attribute Using Context Window แสดงดังภาพประกอบ 4.3 ขั้นตอนนี้จะต้องระบุ ขนาดหน้าต่างและประเภทของขนาดหน้าต่างที่ต้องการ

4.1.4 ผังงานโปรแกรมย่อยของ Step3: Attribute Selection แสดงดังภาพประกอบ 4.4 ประกอบด้วยการเลือกเทคนิคการกรองที่ต้องการและระบุจำนวนแอทริบิวต์ที่ต้องการ

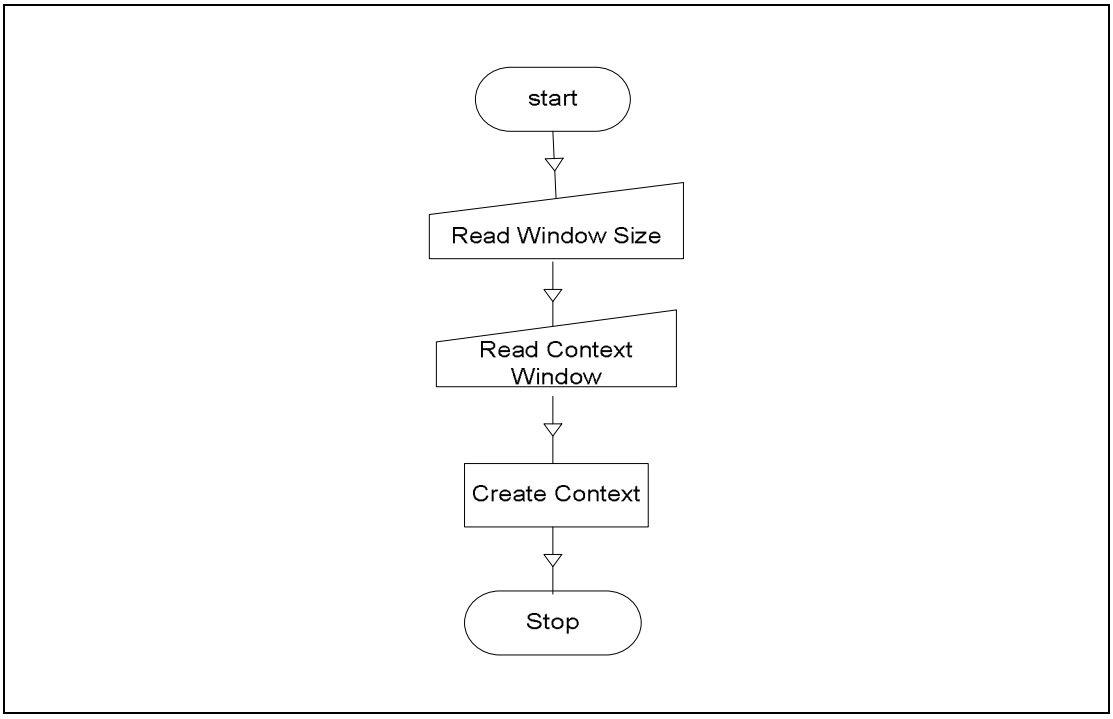
4.1.5 ผังงานโปรแกรมย่อยของ Step4: Word Sense Classification แสดงดังภาพประกอบ 4.5 ขั้นตอนนี้เป็นการจำแนกความหมายโดยต้องเลือกวิธีการจำแนกความหมายและอัลกอริทึมในการจำแนกความหมาย



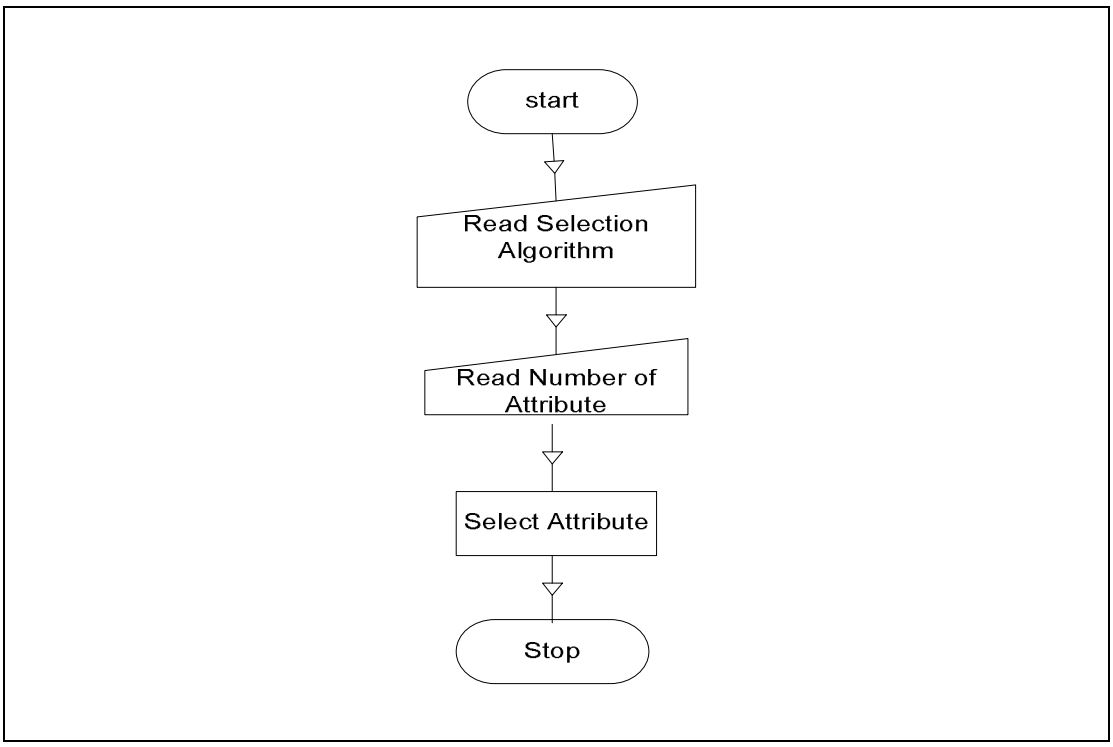
ภาพประกอบ 4.1 ผังการทำงานของโปรแกรม



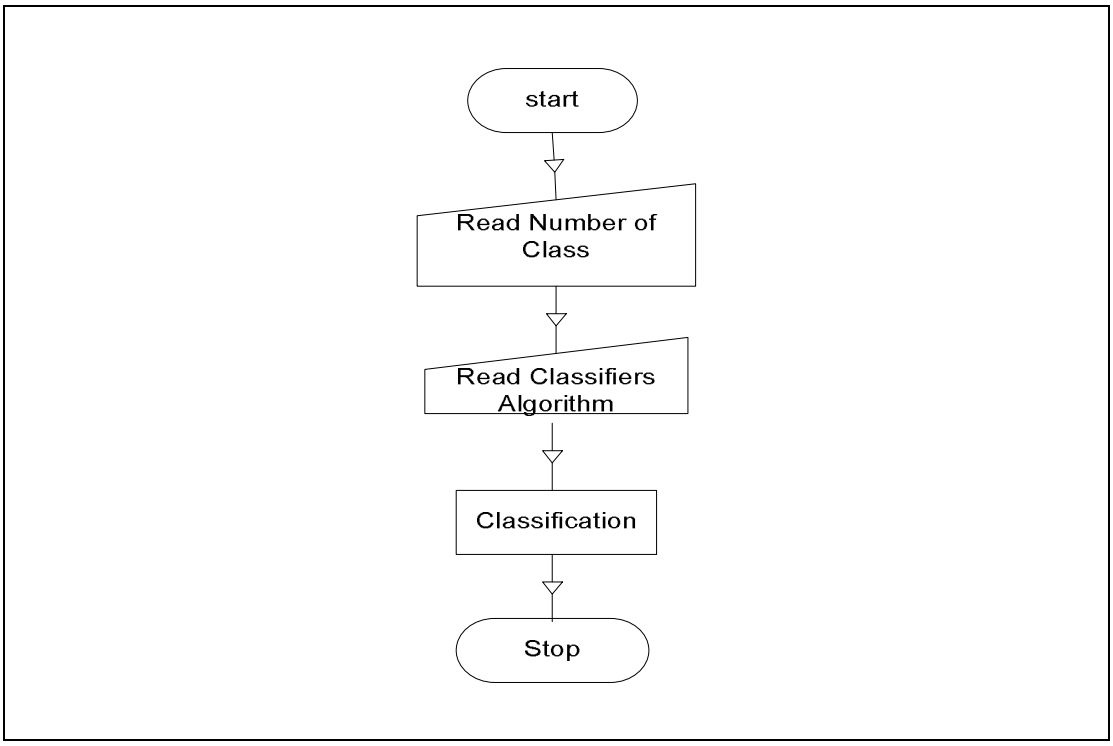
ภาพประกอบ 4.2 ผังการทำงานของโปรแกรม Step1: Data Preprocessing



ภาพประกอบ 4.3 ผังการทำงานของโปรแกรม Step2: Create Attribute Using Context Window



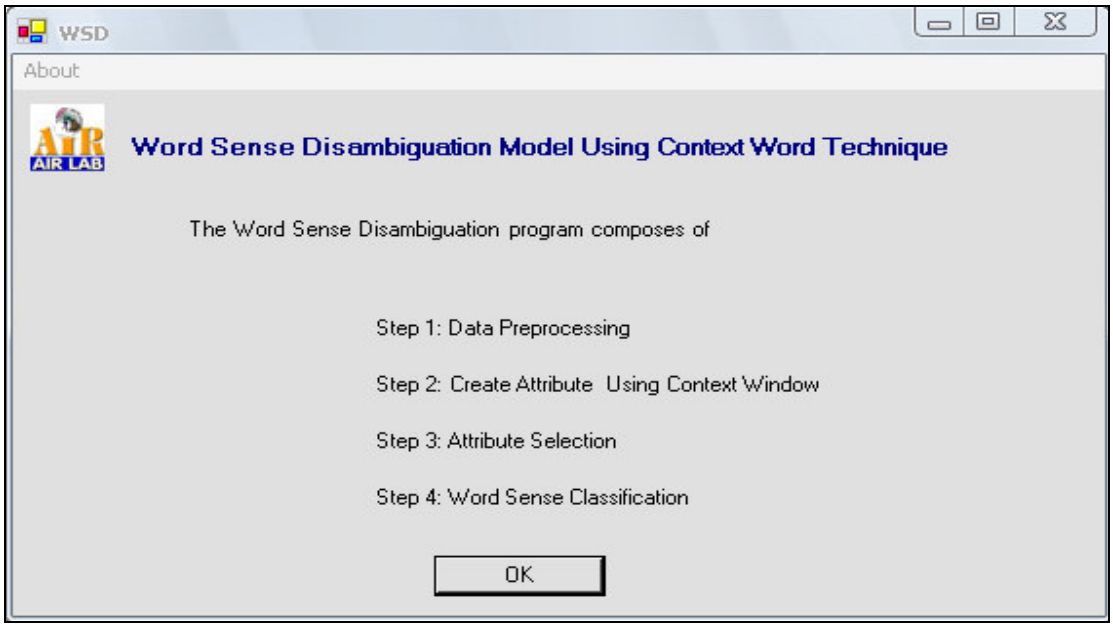
ภาพประกอบ 4.4 ผังการทำงานของโปรแกรม Step3: Attribute Selection



ภาพประกอบ 4.5 ผังการทำงานของโปรแกรม Step4: Word Sense Classification

4.2 ส่วนประกอบของโปรแกรม

เมื่อเปิดโปรแกรมการแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคคำบริบทมา จะปรากฏหน้าจอซึ่งแสดงขั้นตอนการทำงานโดยรวมของโปรแกรมทั้งหมดดังภาพประกอบ 4.6



ภาพประกอบ 4.6 หน้าจอหลักของโปรแกรมการแก้ปัญหาความกำกวมของคำ

เมื่อคลิกเลือกปุ่ม OK จะปรากฏหน้าจอของการแก้ปัญหาความกำกวมของคำทั้งหมด 4 ขั้นตอน ดังภาพประกอบ 4.7 โปรแกรมจะประกอบด้วย 3 ส่วนคือ

4.2.1 ส่วนการทำงานทั้งหมด 4 ขั้นตอนคือ

1) Preprocessing เป็นการเตรียมคลังข้อความให้อยู่ในรูปแบบที่พร้อมในการทำงานโดยเริ่มจากการอ่านคำกำกวมที่ต้องการจากไฟล์แล้วจึงทำการ ตัดสัญลักษณ์และคำที่เป็น Stoplist ออกจากไฟล์

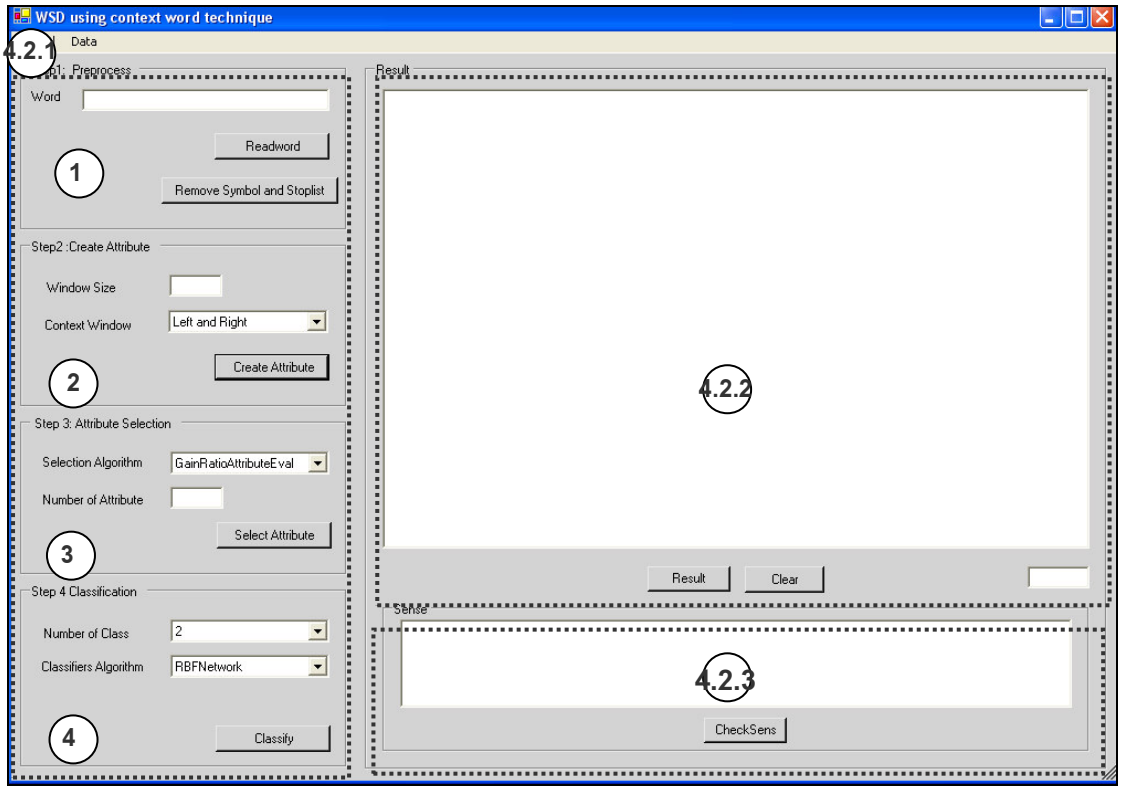
2) Create Attribute เป็นขั้นตอนในการสร้างแอทริบิวต์ ในช่อง Window Size เป็นช่องสำหรับใส่ขนาดหน้าต่างที่ต้องการเช่น 1 2 3 หรือ 4 ในช่อง Context Word จะมีให้เลือกคำบริบททางซ้าย (Left) คำบริบททางขวา (Right) และคำบริบททั้งทางซ้ายและขวา (Left and Right)

3) Attribute Selection เป็นการเลือกแอทริบิวต์ที่ไม่ต้องการออก โดยจะต้องเลือกเทคนิคในการกรอง (Selection Algorithm) ซึ่งมี 2 แบบคือ GainRatioAttributeEval และ InfoGainAttributeEval และใส่จำนวนแอทริบิวต์ที่ต้องการลงในช่อง Number of Attribute แอทริบิวต์นอกเหนือจากนี้จะถูกตัดออกไป

4) Classification เป็นการจำแนกความหมายของคำกำกวม โดยจะต้องเลือกจำนวนของคลาส (Number of Class) ซึ่งมี 2 แบบคือแบบ 2 คลาส (ความหมาย) หรือ จำนวนคลาสทั้งหมด (จำนวนความหมายทั้งหมด) และเลือกอัลกอริทึม (Clasiifiers Algorithm) ซึ่งมี 2 อัลกอริทึมคือ RBFNetwork และ ID3

4.2.2 ส่วนการแสดงผลของการจำแนกความหมายซึ่งแสดงผลเป็นเปอร์เซ็นต์ของค่าความถูกต้อง (Accuracy)

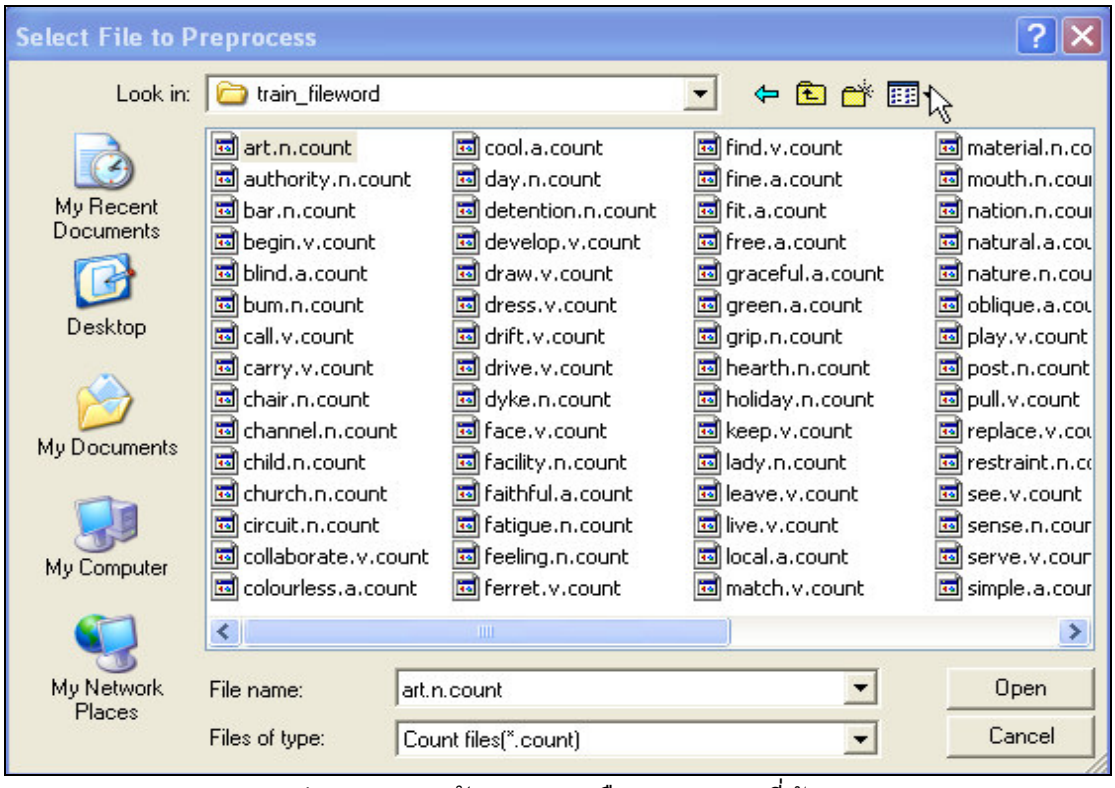
4.2.3 ส่วนการตรวจสอบความหมายของคำกำกวม ใช้ตรวจสอบจำนวนความหมายและความหมายของคำกำกวม



ภาพประกอบ 4.7 หน้าจอการทำงานของโปรแกรมการแก้ปัญหาความกำกวมของคำ

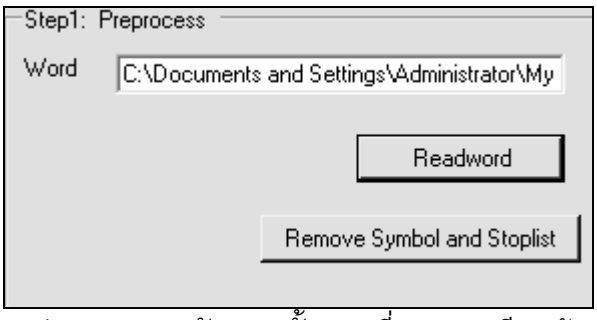
4.3 ผลการทำงานของโปรแกรม

ขั้นตอนที่ 1 Preprocess คลิกเลือกคำโดยปุ่ม Readword เพื่อเลือกคำที่ต้องการดังภาพประกอบ 4.8



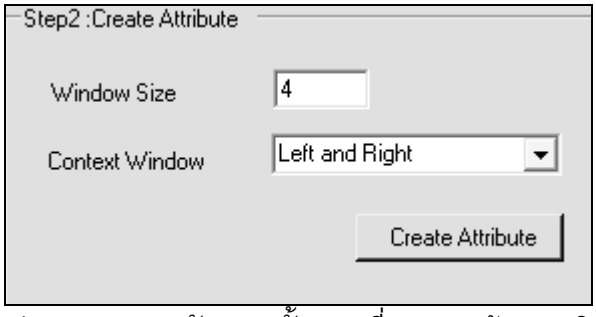
ภาพประกอบ 4.8 ตัวอย่างการเลือกคำกำกวมที่ต้องการ

หลังจากนั้นคลิกปุ่ม Remove Symbol and Stoplist เพื่อตัดสัญลักษณ์และคำที่เป็น Stoplist ออก ตัวอย่างขั้นตอนที่ 1 ดังภาพประกอบ 4.9



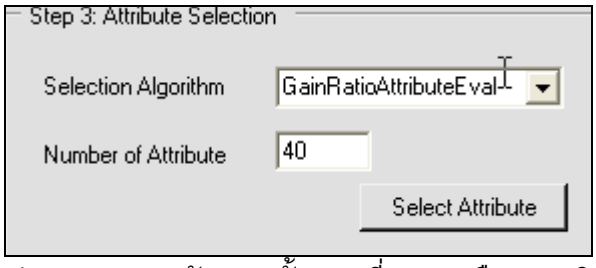
ภาพประกอบ 4.9 ตัวอย่างขั้นตอนที่ 1 การเตรียมข้อมูล

ขั้นตอนที่ 2 Create Attribute โดยระบุขนาดหน้าต่างที่ต้องการและเลือกรูปแบบของคำบริบทดังภาพประกอบ 4.10



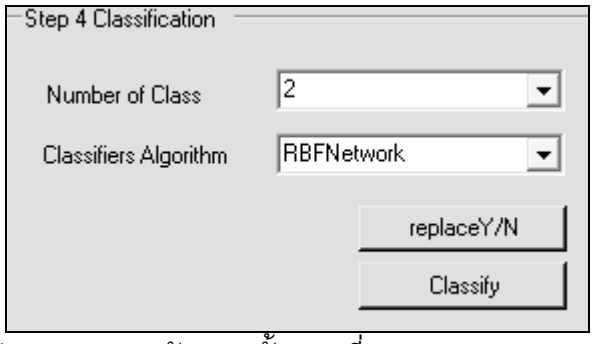
ภาพประกอบ 4.10 ตัวอย่างขั้นตอนที่ 2 การสร้างแอทริบิวต์

ขั้นตอนที่ 3 Attribute Selection โดยเลือกเทคนิคการกรองและระบุจำนวนแอทริบิวต์ที่ต้องการดังภาพประกอบ 4.11



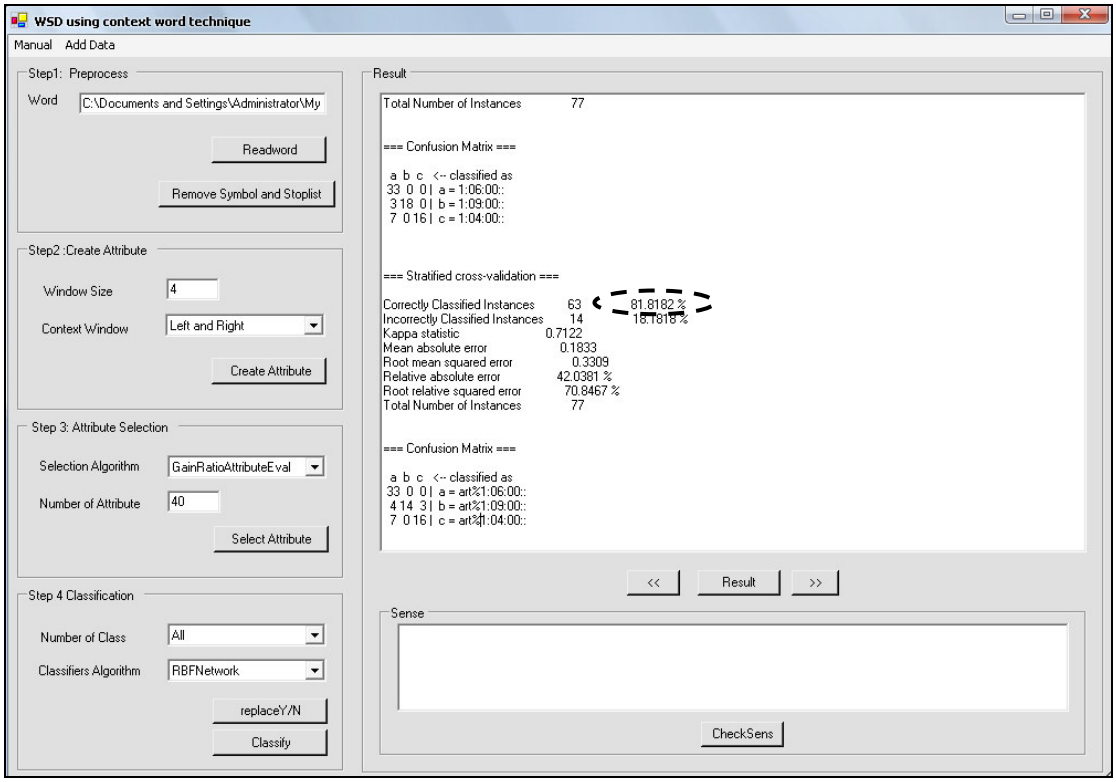
ภาพประกอบ 4.11 ตัวอย่างขั้นตอนที่ 3 การเลือกแอทริบิวต์

ขั้นตอนที่ 4 Word Sense Classification โดยเลือกเลือกประเภทของการจำแนกและอัลกอริทึมในการจำแนกดังภาพประกอบ 4.12



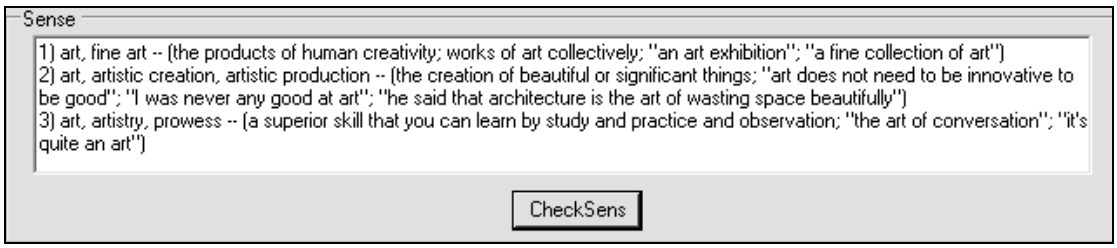
ภาพประกอบ 4.12 ตัวอย่างขั้นตอนที่ 4 การจำแนกความหมาย

เมื่อกำหนดค่าทั้ง 4 ขั้นตอนแล้วผลลัพธ์ค่าความถูกต้องของการจำแนกความหมายแสดงได้ดังภาพประกอบ 4.13



ภาพประกอบ 4.13 ตัวอย่างการแสดงผลลัพธ์ของการจำแนกความหมาย

เมื่อต้องการตรวจสอบความหมายของคำกำกวมคลิกปุ่ม Check Sense จะแสดงสัญลักษณ์ของความหมายทั้งหมดของคำกำกวมนั้น ดังภาพประกอบ 4.14



ภาพประกอบ 4.14 การตรวจสอบความหมายของคำกำกวม

4.4 เครื่องมือที่ใช้ในการพัฒนาโปรแกรม

4.4.1 ด้านฮาร์ดแวร์

1) คอมพิวเตอร์ส่วนบุคคล หน่วยความจำ 1 กิกะไบต์ ฮาร์ดดิสก์ ความจุ 40 กิกะไบต์ สำหรับพัฒนาและเป็นเครื่องทดสอบ

4.4.2 ด้านซอฟต์แวร์

1) ระบบปฏิบัติการ Microsoft Windows XP

2) Perl

3) Ngram Statistic Package (NSP)

4) SenseTools

5) WEKA (คู่มือการใช้งานโปรแกรม WEKA แบบ Command Line Interface แสดงดังภาคผนวก ก)

6) Visual Basic.Net

7) OMtoSVAL2

บทที่ 5

ผลการทดลองและวิจารณ์

วิทยานิพนธ์นี้ใช้ข้อมูลในการทดลองคือคลังข้อความ Senseval-2 ใช้แบบจำลองการแก้ปัญหาความกำกวมของคำจากคลังข้อความโดยใช้เทคนิคคำบริบท (Word Sense Disambiguation Model from Corpus Using Context Word Technique)

แบบจำลองการแก้ปัญหาความกำกวมของคำจากคลังข้อความโดยใช้เทคนิคคำบริบท

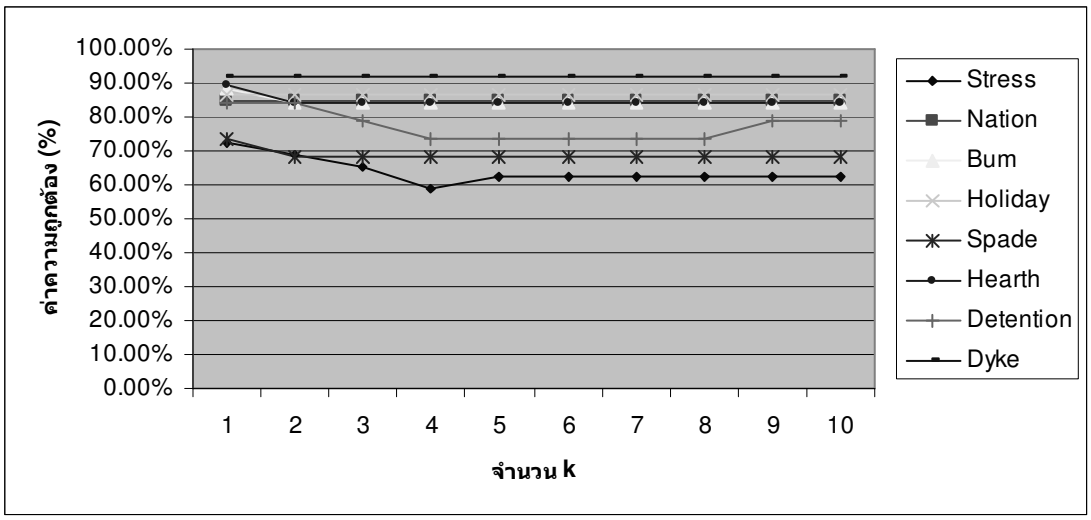
ขั้นตอนที่ 1: เตรียมคลังข้อความ

1.1 เตรียมคลังข้อความ Senseval-2 ซึ่งอยู่ในรูปแบบของ XMLโดยคำกำกวมหนึ่งคำประกอบด้วยสองไฟล์คือ ไฟล์ที่มีนามสกุลเป็น .xml และ .count

1.2 เลือกคำกำกวมที่ต้องการ

1.3 ตัดคำที่เป็น Stoplist ออกจากไฟล์นามสกุล .count เนื่องจากคำที่เป็น Stoplist คือคำที่ทำให้ไฟล์มีขนาดใหญ่และเป็นคำฟุ่มเฟือย ไม่ได้นำมาวิเคราะห์หาความหมายของคำ ทำให้ประสิทธิภาพในการแก้ปัญหาความกำกวมลดลง ในขั้นตอนนี้จะต้องตัดตัวอักษรที่เป็นสัญลักษณ์ต่างๆ ออกด้วย

คำกำกวมจากคลังข้อความ Senseval-2 ที่ใช้ในการทดสอบประสิทธิภาพในการจำแนกความหมายเมื่อตัดคำที่เป็น Stoplist และไม่ตัดคำที่เป็น Stoplist แสดงดังตารางที่ 5.1 โดยประกอบด้วยคำกำกวมดังต่อไปนี้ stress fatigue nation bum holiday spade hearth detention และ dyke ซึ่งแต่ละคำจะมีจำนวนตัวอย่างแตกต่างกัน โดยทดลองกับสามอัลกอริทึม คือ IBk ID3 และ NaiveBayes สำหรับการทดลองในอัลกอริทึม IBk นั้นได้ทำการทดลองเริ่มต้น โดยกำหนดค่า k ให้มีค่า 1 ถึง 10 จากภาพประกอบ 5.1 ผลการทดลองเมื่อเพิ่มค่า k ขึ้นค่าความถูกต้องจะมีค่าคงที่ตัวอย่างเช่น คำว่า “dyke” และมีบางคำที่มีความถูกต้องลดลงตัวอย่างเช่น คำว่า “stress” และ “detention” เป็นต้น และจากค่าความถูกต้องที่ได้ ค่า k เท่ากับ 1 จะมีความถูกต้องสูงสุด ดังนั้นจึงได้เลือกค่า k เท่ากับ 1 ในการทดลองขั้นต่อไปของอัลกอริทึม IBk

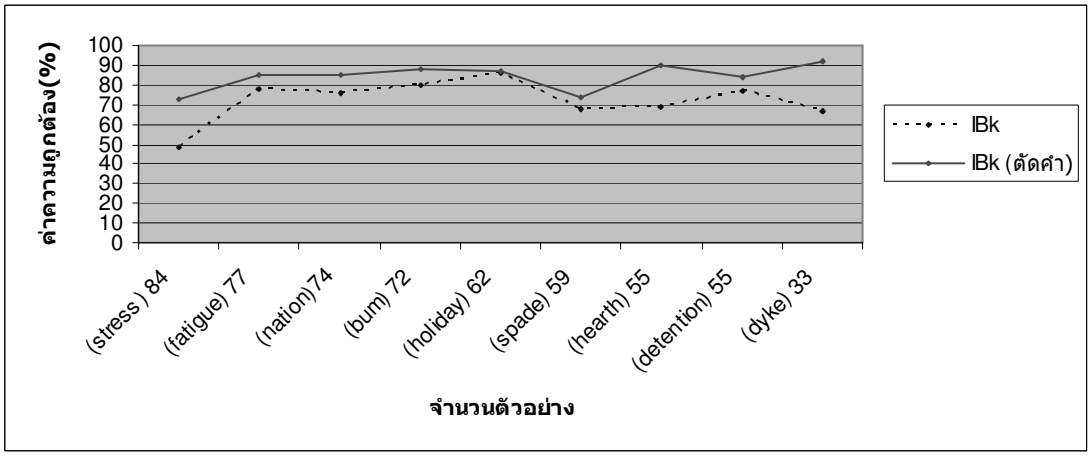


ภาพประกอบ 5.1 แสดงค่าความถูกต้องเมื่อใช้อัลกอริทึม IBk เมื่อค่า k มีค่าต่างกัน

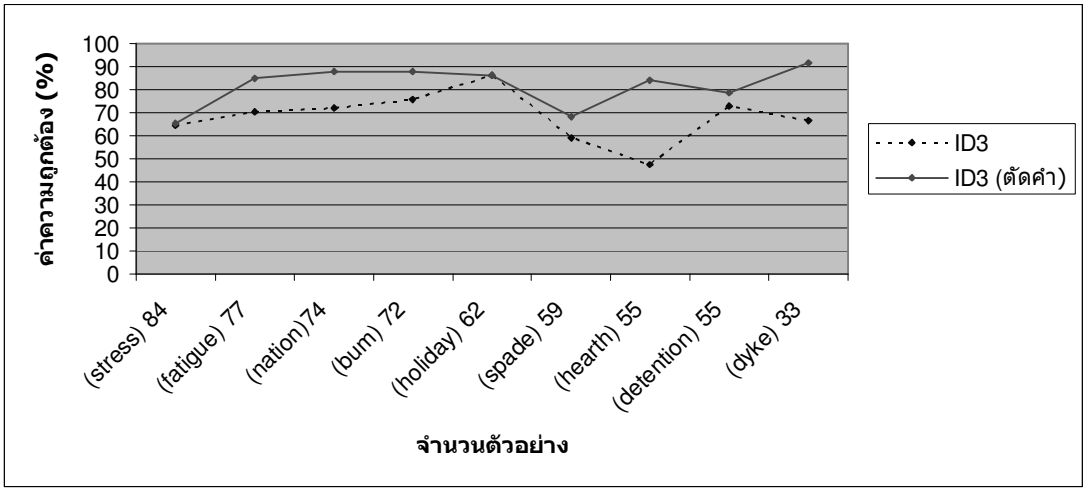
ผลการทดลองแสดงให้เห็นว่าการแก้ปัญหาความกำกวมโดยใช้การตัดคำที่เป็น Stoplist จากทั้งสามขั้นตอนวิธีคือ IBk ID3 และ NaiveBayes จะให้ค่าความถูกต้องสูงกว่าการทดลองโดยไม่มีการตัดคำที่เป็น Stoplist ตัวอย่างเช่น hearth จากตารางที่ 5.1 โดยขั้นตอนวิธี IBk ไม่ตัดคำมีความถูกต้อง 68.42% เมื่อตัดคำได้ค่าความถูกต้อง 89.47% ขั้นตอนวิธี ID3 ไม่ตัดคำมีความถูกต้อง 47.36% เมื่อตัดคำได้ค่าความถูกต้อง 84.21% ขั้นตอนวิธี NaiveBayes ไม่ตัดคำมีความถูกต้อง 68.42% เมื่อตัดคำได้ค่าความถูกต้อง 84.21% ผลการเปรียบเทียบแต่ละขั้นตอนวิธีการทดลองโดยไม่ตัดคำ Stoplist และตัดคำที่เป็น Stoplist จากขั้นตอนวิธี IBk ID3 และ NaiveBayes แสดงได้ดังภาพประกอบ 5.2 5.3 และ 5.4 ตามลำดับ

ตารางที่ 5.1 ผลการทดลองเปรียบเทียบการใช้การตัดคำและไม่ตัดคำในแต่ละอัลกอริทึม
อัลกอริทึม IBk กำหนดค่า k ให้มีค่าเท่ากับ 1

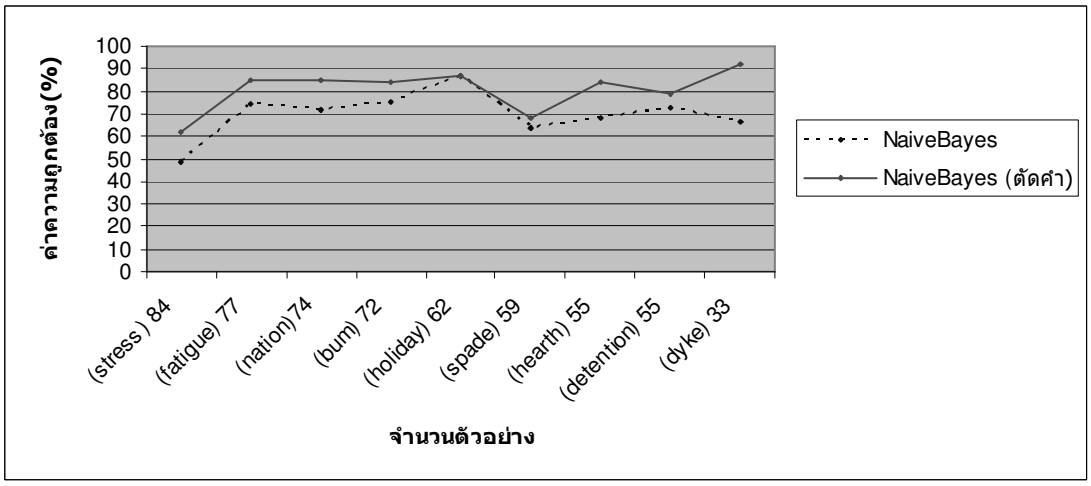
คำ	จำนวน	Classified Accuracy					
		IBk (k=1)		ID3		NaiveBayes	
	ตัวอย่าง	ไม่ตัด stoplist	ตัด stoplist	ไม่ตัด stoplist	ตัด stoplist	ไม่ตัด stoplist	ตัด stoplist
stress	81	48.38%	**72.41%**	64.51%	**65.51%**	48.38%	**62.06%**
fatigue	77	77.77%	**85.15%**	70.37%	**85.18%**	74.01%	**85.18%**
nation	74	76.00%	**84.61%**	72.00%	**88.00%**	72.00%	**84.61%**
bum	72	79.31%	**88.00%**	75.86%	**88.00%**	75.00%	**84.00%**
holiday	62	86.36%	**86.39%**	86.36%	**86.36%**	86.36%	**86.36%**
spade	59	68.18%	**73.68%**	59.09%	**68.42%**	63.63%	**68.42%**
hearth	55	68.42%	**89.47%**	47.36%	**84.21%**	68.42%	**84.21%**
detention	55	77.27%	**84.21%**	72.72%	**78.94%**	72.72%	**78.94%**
dyke	33	66.67%	**91.66%**	66.66%	**91.66%**	66.66%	**91.66%**



ภาพประกอบ 5.2 แสดงค่าความถูกต้องเมื่อใช้อัลกอริทึม IBk เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบปกติและแบบตัดคำ



ภาพประกอบ 5.3 แสดงค่าความถูกต้องเมื่อใช้อัลกอริทึม ID3 เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบปกติและแบบตัดคำ



ภาพประกอบ 5.4 แสดงค่าความถูกต้องเมื่อใช้อัลกอริทึม NaiveBayes เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบปกติและแบบตัดคำ

จากผลการทดลองดังกล่าวแสดงให้เห็นว่าเมื่อตัดคำที่เป็น Stoplist ออกจากคลังข้อความจะทำให้ค่าความถูกต้องในการจำแนกความหมายสูงกว่าเมื่อไม่ตัดคำที่เป็น Stoplist จึงได้ใช้การตัดคำที่เป็น Stoplist ออกจากไฟล์ของคำกำกวมทุกคำเพื่อเพิ่มค่าความถูกต้องในการจำแนกความหมายให้ดีขึ้น

คำกำกวมที่ใช้ในการทดลองการจำแนกความหมายคือ art authority bar bum chair replace hearth local detention child church child และ dyke ทดลองแบบ 10 Folds cross-validation สามารถแสดงรายละเอียดจากตัวอย่างคำกำกวม art และ dyke โดยคำสองคำนี้ได้ตัดคำที่เป็น stoplist แล้วจากขั้นตอนที่ 1 ดังนั้นในขั้นตอนต่อไปจะเป็นการทำงานของขั้นตอนที่ 2-4 ซึ่งมีรายละเอียดดังนี้

5.1 ตัวอย่างคำกำกวม art

คำกำกวม art ของคลังข้อความ Senseval-2 ประกอบด้วยความหมายกำกวม คือ 1:04:00:: (การสร้างงานศิลปะ) 1:06:00:: (ผลิตภัณฑ์ศิลปะ) 1:09:00:: (ทักษะ)

ขั้นตอนที่ 2: สร้างแอทริบิวต์โดยใช้คำบริบท

2.1 กำหนดขนาดหน้าต่างของคำบริบทตามต้องการ ในที่นี้สมมติเลือกขนาดหน้าต่างคำบริบทเป็น 4

2.2 ตัดประโยคจากไฟล์ .count ให้มีคำบริบทเพื่อสร้างแอทริบิวต์ซึ่งแบ่งเป็น 3 แบบดังนี้

แบบที่ 1 ใช้บริบททางซ้ายเท่านั้นดังภาพประกอบ 5.5

Art
heard summer piazza front **art**
charge extended critical capacity **art**
Paintings drawings sculpture period **art**
defined common experiences **art**
nationalism pompous conventional boring **art**
Goldsmiths graduates handling demands **art**

ภาพประกอบ 5.5 คำบริบททางซ้าย

แบบที่ 2 ใช้บริบททางขวาเท่านั้นดังภาพประกอบ 5.6

Art dance creative called Halo
art gallery Town Hall park
art aesthetic
art 350 display ranging Tudor
art whatever music poetry
art Western world
art world media

ภาพประกอบ 5.6 คำบริบททางขวา

แบบที่ 3 ใช้บริบททั้งทางซ้ายและขวาดังภาพประกอบ 5.7

Art dance creative called Halo
heard summer piazza front **art** gallery Town Hall park
charge extended critical capacity **art** aesthetic
Paintings drawings sculpture period **art** 350 display ranging Tudor
defined common experiences **art** whatever music poetry
nationalism pompous conventional boring **art** Western world
Goldsmiths graduates handling demands **art** world media

ภาพประกอบ 5.7 คำบริบททางซ้ายและขวา

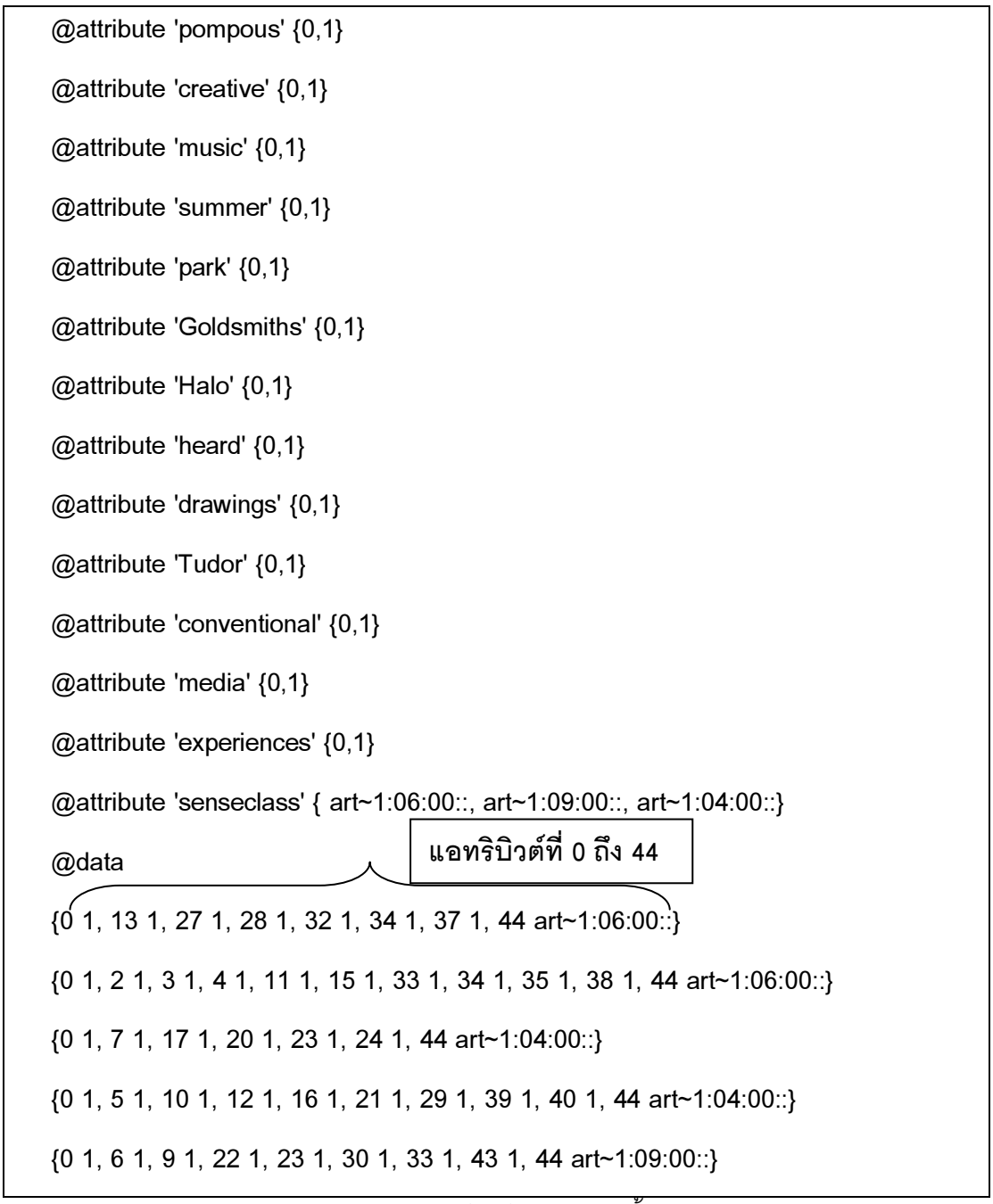
2.3 ใช้โปรแกรม SenseTools และ โปรแกรม NSP ในการแปลงข้อความดังกล่าวให้อยู่ในรูปแบบ Feature Vectors ดังภาพประกอบ 5.8 ซึ่งใช้คำบริบททั้งทางซ้ายและขวา (แบบที่3) คำบริบททั้งหมดถูกสร้างเป็นแอทริบิวต์แต่ละแอทริบิวต์มีค่าที่เป็นไปได้ 2 ค่าคือ 0 และ 1 เช่น **“**@attribute 'art' {0,1}**”** หมายถึงแอทริบิวต์ art มีค่าที่เป็นไปได้คือ 0 หรือ 1 แอทริบิวต์สุดท้ายคือแอทริบิวต์ senseclass บอกความหมายของคำกำกวมซึ่งมีค่าที่เป็นไปได้ 3 ค่า คือ art~1:04:00:: art~1:06:00:: และ art~1:09:00:: ในส่วนการเขียนตัวอย่าง (Instance) จึงแสดงเฉพาะแอทริบิวต์ที่มีค่า 1 แอทริบิวต์ที่มีค่า 0 จะไม่แสดงเพราะแอทริบิวต์มีจำนวนมาก เช่น “{0 1, 2 1, 3 1, 4 1, 11 1, 15 1, 33 1, 34 1, 35 1, 38 1, 44 art~1:06:00::}” หมายถึง แอทริบิวต์ที่ 0 คือ “art” มีค่าเป็น 1 แสดงว่า มีคำว่า art อยู่ แอทริบิวต์ที่ 2 คือ “gallery” มีค่าเป็น 1 แสดงว่า มีคำว่า gallery อยู่ แอทริบิวต์อื่นที่ไม่เขียนแสดงว่ามีค่าเป็น 0 หมายถึงไม่มีคำนั้นอยู่ และแอทริบิวต์สุดท้ายคือแอทริบิวต์ที่ 44 เป็นแอทริบิวต์ที่บอกคลาสหรือความหมายของคำ

```
@relation 'RELATION'

@attribute 'art' {0,1}

@attribute 'world' {0,1}

@attribute 'gallery' {0,1}

@attribute 'front' {0,1}

@attribute 'piazza' {0,1}

@attribute 'ranging' {0,1}
```

ภาพประกอบ 5.8 สร้างแอทริบิวต์โดยใช้คำบริบททั้งทางซ้ายและขวา

```
@attribute 'defined' {0,1}
@attribute 'charge' {0,1}
@attribute 'boring' {0,1}
@attribute 'poetry' {0,1}
@attribute 'sculpture' {0,1}
@attribute 'Hall' {0,1}
@attribute 'Paintings' {0,1}
@attribute 'called' {0,1}
@attribute 'Western' {0,1}
@attribute 'Town' {0,1}
@attribute 'period' {0,1}
@attribute 'critical' {0,1}
@attribute 'handling' {0,1}
@attribute 'nationalism' {0,1}
@attribute 'capacity' {0,1}
@attribute '350' {0,1}
@attribute 'common' {0,1}
@attribute 'aesthetic' {0,1}
@attribute 'extended' {0,1}
@attribute 'graduates' {0,1}
@attribute 'demands' {0,1}
@attribute 'dance' {0,1}
@attribute 'Art' {0,1}
@attribute 'display' {0,1}
@attribute 'whatever' {0,1}
```

ภาพประกอบ 5.8 สร้างแอทริบิวต์โดยใช้คำบริบททั้งทางซ้ายและขวา (ต่อ)



ภาพประกอบ 5.8 สร้างแอทริบิวต์โดยใช้คำบริบททั้งทางซ้ายและขวา (ต่อ)

ขั้นตอนที่ 3: เลือกแอทริบิวต์

เมื่อได้ข้อมูลในรูปแบบ arff แล้วกรองแอทริบิวต์ด้วยตัวกรอง 2 แบบคือ InfoGainAttributeEval และ GainRatioAttributeEval สมมติกรองให้มีจำนวนแอทริบิวต์เท่ากับ 30 จะเห็นว่าจำนวนแอทริบิวต์จาก 45 แอทริบิวต์ (แอทริบิวต์ที่ 0 ถึง 44) ดังภาพประกอบ 5.8 ลดลงเป็น 30 แอทริบิวต์ (แอทริบิวต์ที่ 0 ถึง 29) ดังภาพประกอบ 5.9

```
@relation 'RELATION'

@attribute capacity {0,1}
@attribute critical {0,1}
@attribute display {0,1}
@attribute 350 {0,1}
@attribute extended {0,1}
@attribute charge {0,1}
@attribute ranging {0,1}
@attribute period {0,1}
@attribute sculpture {0,1}
@attribute Paintings {0,1}
@attribute drawings {0,1}
@attribute Tudor {0,1}
@attribute summer {0,1}
@attribute music {0,1}
@attribute called {0,1}
@attribute Hall {0,1}
@attribute heard {0,1}
@attribute Town {0,1}
@attribute creative {0,1}
@attribute piazza {0,1}
@attribute gallery {0,1}
@attribute front {0,1}
@attribute poetry {0,1}
@attribute experiences {0,1}
@attribute defined {0,1}
@attribute park {0,1}
@attribute common {0,1}
@attribute dance {0,1}
@attribute Art {0,1}
```

ภาพประกอบ 5.9 กรองแอทริบิวต์ให้มีจำนวน 30 แอทริบิวต์

```
@attribute senseclass {art~1:06:00::,art~1:09:00::,art~1:04:00::}
@data                                          แอทริบิวต์ที่ 0 ถึง 29
{12 1,14 1,18 1,27 1,28 1, 29 art~1:06:00::}
{12 1,13 1,15 1,16 1,17 1,19 1,20 1,21 1,25 1, 29 art~1:06:00::}
{0 1,1 1,4 1,5 1,29 art~1:04:00::}
{2 1,3 1,6 1,7 1,8 1,9 1,10 1,11 1,29 art~1:04:00::}
{13 1,22 1,23 1,24 1,26 1, 29 art~1:09:00::}
```

ภาพประกอบ 5.9 กรองแอทริบิวต์ให้มีจำนวน 30 แอทริบิวต์ (ต่อ)

ขั้นตอนที่ 4: จำแนกความหมาย

การจำแนกความหมายของคำเป็นขั้นตอนในการเลือกความหมายที่ถูกต้องของคำกำกวม จำแนกความหมายของคำที่กำกวมโดยใช้ 10-folds Cross Validation โดยเลือกวิธีการจำแนกความหมาย 2 แบบ คือการจำแนกโดยใช้สองความหมายและใช้ความหมายทั้งหมด อัลกอริทึมในการจำแนกความหมาย 2 อัลกอริทึมคือ RBFNetwork และ ID3

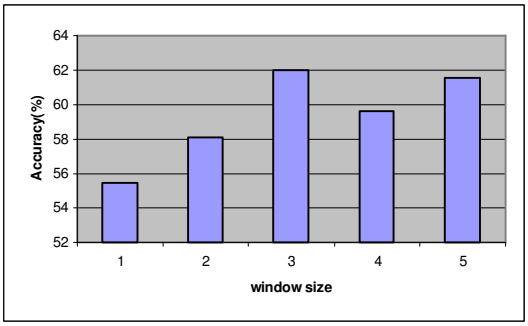
ผลการทดลองสามารถสรุปได้ 6 ประเด็นคือประเด็นการเลือกขนาดหน้าต่างคำบริบท ประเด็นการเลือกรูปแบบหน้าต่างคำบริบท ประเด็นจำนวนแอทริบิวต์สำหรับการกรองแอทริบิวต์ ประเด็นการเลือกเทคนิคการกรองแอทริบิวต์ ประเด็นการเลือกอัลกอริทึมการจำแนกความหมาย และประเด็นการเลือกจำนวนคลาส

1) ประเด็นการเลือกขนาดหน้าต่างคำบริบท

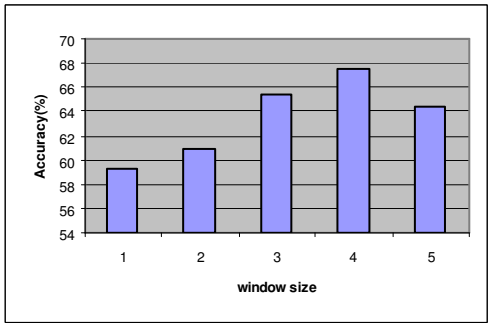
เมื่อความกว้างขนาดหน้าต่างเพิ่มขึ้นค่าความถูกต้องมีแนวโน้มเพิ่มขึ้นดังตัวอย่างตารางที่ 5.2 และภาพประกอบ 5.10(d) ของบริบททางซ้าย ใช้อัลกอริทึม RBFNetwork กรองแบบ GainRatioAttributeEval ที่ขนาดหน้าต่าง 1 2 3 4 และ 5 มีค่าความถูกต้องคือ 66.35% 67.14% 69.71% 70.51% และ 71.15% เป็นต้น การใช้บริบททางขวาแสดงดังตารางที่ 5.3 และภาพประกอบ 5.11 และบริบททั้งทางซ้ายและขวาแสดงดังตารางที่ 5.4 และภาพประกอบ 5.12 จะเห็นได้ว่าทั้งอัลกอริทึม ID3 และ RBFNetwork ของการกรองทั้งสองแบบคือ InfoGainAttributeEval และ GainRatioAttributeEval ของบริบททางซ้ายและบริบททั้งทางซ้ายและขวา มีแนวโน้มการทำงานในทำนองเดียวกัน อย่างไรก็ตามถ้าขนาดหน้าต่างมากเกินไปอาจส่งผลต่อค่าความถูกต้องที่ลดลงได้เช่นกัน

ตารางที่ 5.2 ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางซ้าย ที่ขนาดหน้าต่าง 1 2 3 4 และ 5

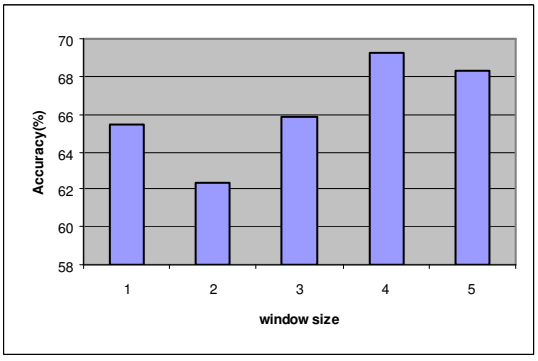
window size	Accuracy			
	ID3		RBFNetwork	
	InfoGainAttributeEval	GainRatioAttributeEval	InfoGainAttributeEval	GainRatioAttributeEval
1	55.45 %	59.24 %	65.4 %	66.35 %
2	58.09 %	60.95 %	62.38 %	67.14 %
3	62.01 %	65.38 %	65.86 %	69.71 %
4	59.61 %	67.5 %	69.23 %	70.51 %
5	61.53 %	64.42 %	68.26 %	71.15 %



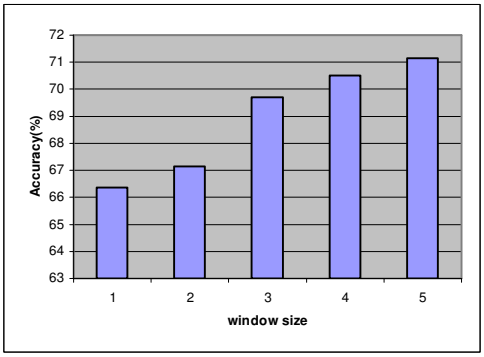
(a) ใช้ ID3 ในการจำแนกและกรองโดย InfoGainAttributeEval



(b) ใช้ ID3 ในการจำแนกและกรองโดย GainRatioAttributeEval



(c) ใช้ RBF ในการจำแนกและกรองโดย InfoGainAttributeEval

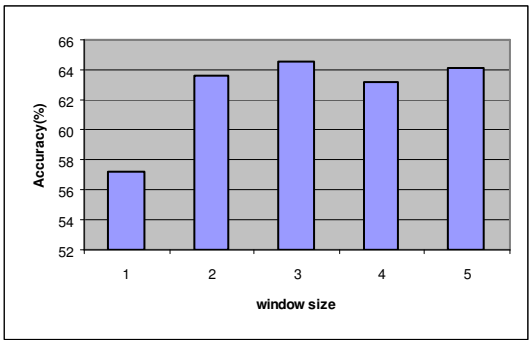


(d) ใช้ RBF ในการจำแนกและกรองโดย GainRatioAttributeEval

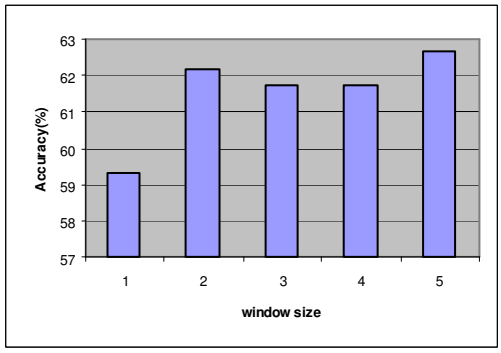
ภาพประกอบ 5.10 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่าง ๆ โดยใช้บริบททางซ้าย

ตารางที่ 5.3 ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางขวา ที่ขนาดหน้าต่าง 1 2 3 4 และ 5

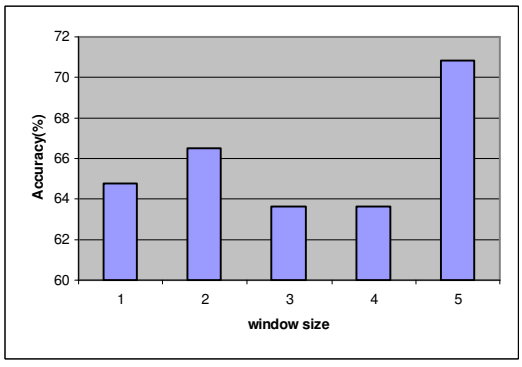
window size	Accuracy			
	ID3		RBFNetwork	
	InfoGainAttributeEval	GainRatioAttributeEval	InfoGainAttributeEval	GainRatioAttributeEval
1	57.14 %	59.33 %	64.76 %	63.15 %
2	63.63 %	62.2 %	66.5 %	64.59 %
3	64.59 %	61.72 %	63.63 %	66.02 %
4	63.15 %	61.72 %	63.63 %	65.07 %
5	64.11 %	62.67 %	70.83 %	67.46 %



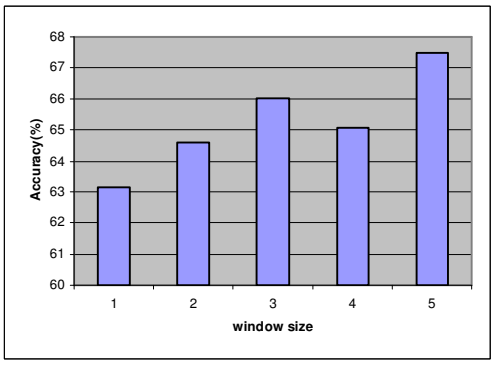
(a) ใช้ ID3 ในการจำแนกและกรองโดย InfoGainAttributeEval



(b) ใช้ ID3 ในการจำแนกและกรองโดย GainRatioAttributeEval



(c) ใช้ RBF ในการจำแนกและกรองโดย InfoGainAttributeEval

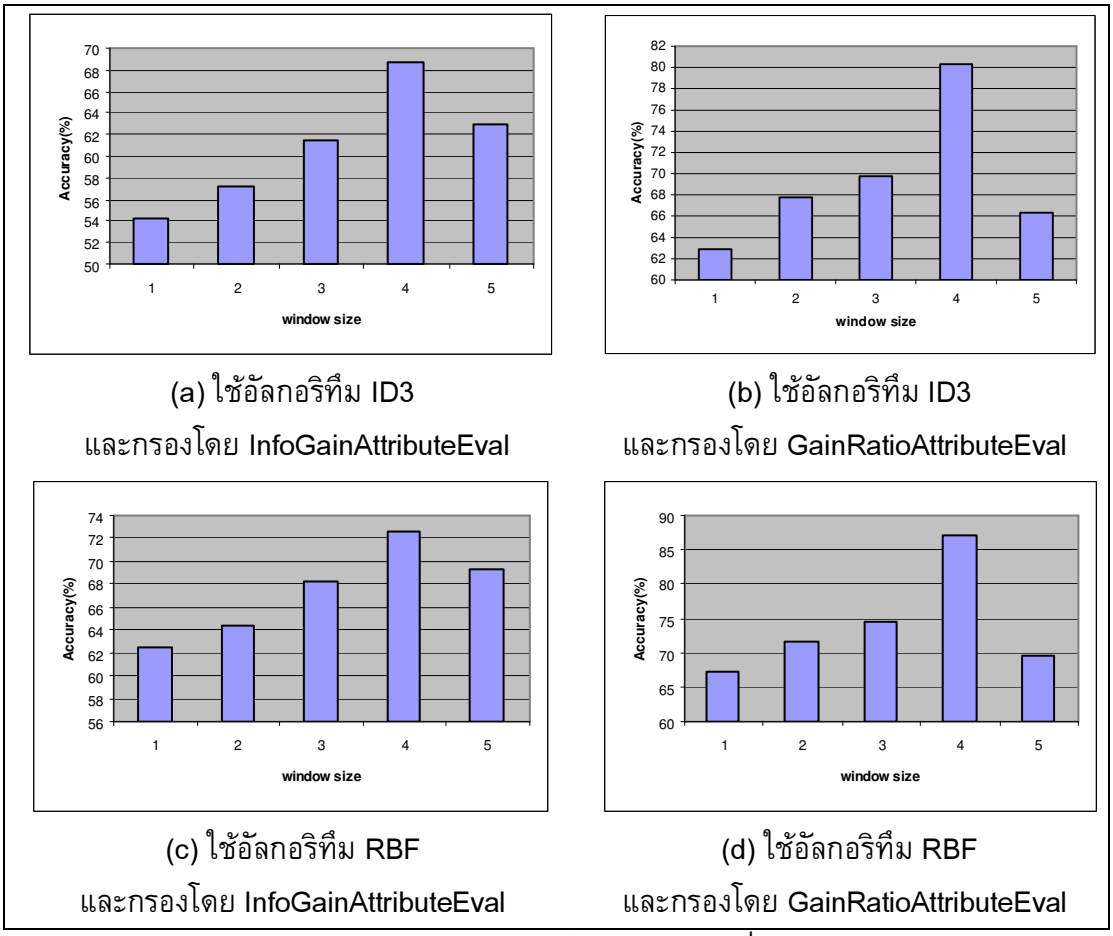


(d) ใช้ RBF ในการจำแนกและกรองโดย GainRatioAttributeEval

ภาพประกอบ 5.11 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่างๆ โดยใช้บริบททางขวา

ตารางที่ 5.4 ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททั้งทางซ้ายและขวา ที่ขนาดหน้าต่าง 1 2 3 4 และ 5

window size	Accuracy			
	ID3		RBFNetwork	
	InfoGainAttributeEval	GainRatioAttributeEval	InfoGainAttributeEval	GainRatioAttributeEval
1	54.28 %	62.85 %	62.38 %	67.14 %
2	57.21 %	67.78 %	64.42 %	71.63 %
3	61.53 %	69.71 %	68.28 %	74.51 %
4	68.75 %	80.28 %	72.59 %	87.01 %
5	62.98 %	66.34 %	69.23 %	69.71 %



(a) ใช้อัลกอริทึม ID3 และกรองโดย InfoGainAttributeEval

(b) ใช้อัลกอริทึม ID3 และกรองโดย GainRatioAttributeEval

(c) ใช้อัลกอริทึม RBF และกรองโดย InfoGainAttributeEval

(d) ใช้อัลกอริทึม RBF และกรองโดย GainRatioAttributeEval

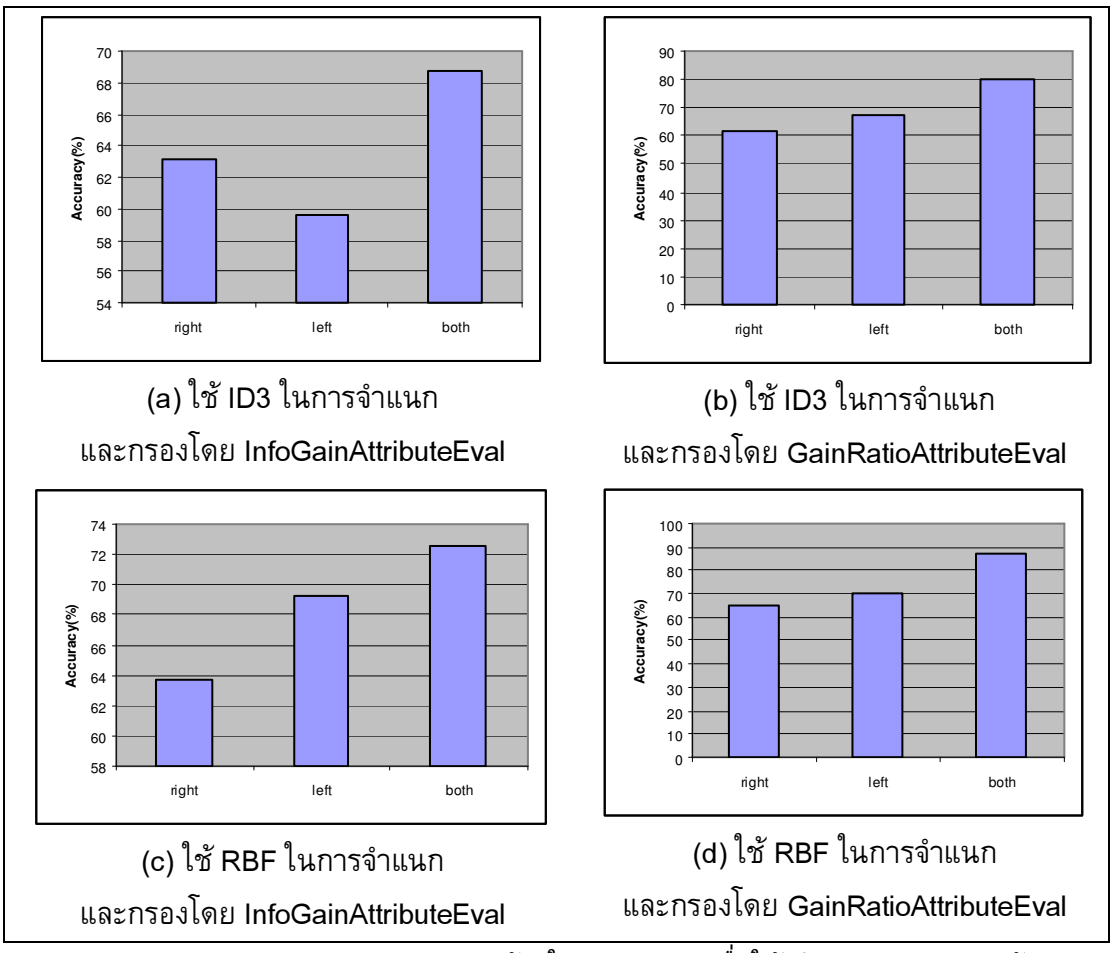
ภาพประกอบ 5.12 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่างๆ โดยใช้บริบททั้งทางซ้ายและขวา

2) ประเด็นการเลือกรูปแบบหน้าต่างคำบริบท

ถ้าเลือกขนาดหน้าต่างเท่ากับ 4 การใช้หน้าต่างคำบริบททั้งซ้ายและขวาจะได้ค่าความถูกต้องสูงกว่าเลือกหน้าต่างคำบริบททางด้านซ้ายหรือขวาเพียงอย่างเดียวดังตัวอย่างตารางที่ 5.5 และภาพประกอบ 5.13(d) ใช้อัลกอริทึม RBFNetwork กรองแบบ GainRatioAttributeEval ใช้บริบททางขวา มีค่าความถูกต้อง 65.07% ใช้คำบริบททางซ้ายมีค่าความถูกต้อง 70.51% และใช้คำบริบททั้งทางซ้ายและขวามีค่าความถูกต้องคือ 87.01% จะเห็นได้ว่าทั้งอัลกอริทึม ID3 และ RBFNetwork ของการกรองทั้งสองแบบคือ InfoGainAttributeEval และ GainRatioAttributeEval มีแนวโน้มการทำงานในทำนองเดียวกัน ข้อสังเกต ส่วนใหญ่ประเภทหน้าต่างคำบริบทแบบทางขวามีค่าความถูกต้องน้อยที่สุดเนื่องจากคำที่ใช้ขยายคำกำกวมส่วนใหญ่มักอยู่ทางซ้ายมือของคำกำกวมตามหลักการเขียนของภาษาอังกฤษ

ตารางที่ 5.5 ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางขวา ทางซ้าย และทางซ้ายและขวา เมื่อขนาดหน้าต่างเท่ากับ 4

window size	Accuracy			
	ID3		RBFNetwork	
	InfoGainAttributeEval	GainRatioAttributeEval	InfoGainAttributeEval	GainRatioAttributeEval
Right	63.15 %	61.72 %	63.63 %	65.07 %
Left	59.61 %	67.5 %	69.23 %	70.51 %
Both	68.75 %	80.28 %	72.59 %	87.01 %



(a) ใช้ ID3 ในการจำแนก และกรองโดย InfoGainAttributeEval

(b) ใช้ ID3 ในการจำแนก และกรองโดย GainRatioAttributeEval

(c) ใช้ RBF ในการจำแนก และกรองโดย InfoGainAttributeEval

(d) ใช้ RBF ในการจำแนก และกรองโดย GainRatioAttributeEval

ภาพประกอบ 5.13 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อใช้บริบททางขวา ทางซ้าย และทั้งทางซ้ายและขวา เมื่อขนาดหน้าต่างความกว้างเท่ากับ 4

3) ประเด็นจำนวนแอทริบิวต์สำหรับการกรองแอทริบิวต์

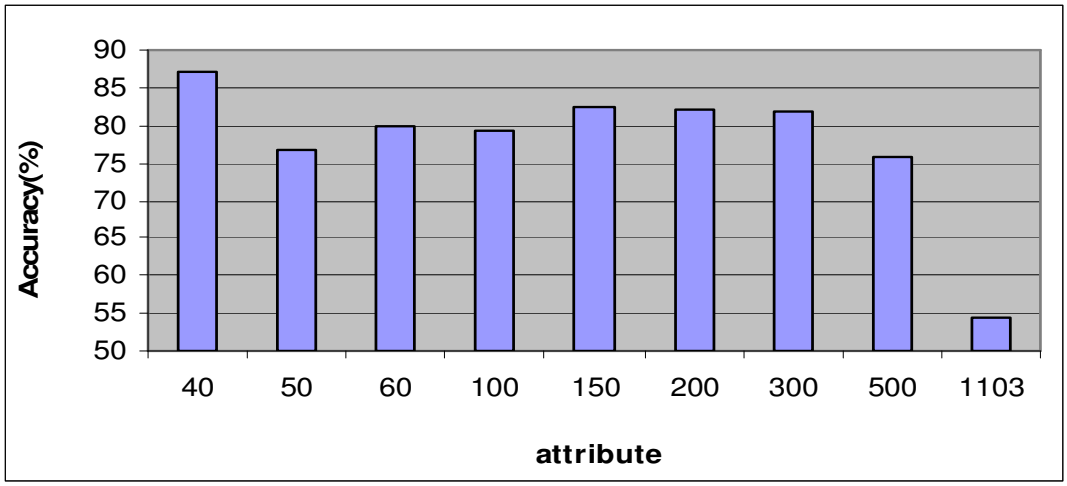
ในการทดลอง จำนวนแอทริบิวต์ที่ได้จากคำกำกวมแต่ละคำมีค่าไม่แน่นอนขึ้นอยู่กับขนาดของหน้าต่างที่เลือกใช้ ประเภทของหน้าต่างคำ บริบทที่เลือกใช้และจำนวนประโยคหรือจำนวนตัวอย่างข้อมูล การกรองแอทริบิวต์แบบไม่มีการกรองหมายถึงให้ใช้จำนวนแอทริบิวต์ที่ได้มาจากการทดลองในขั้นตอนที่ 2 ในการสร้างแอทริบิวต์ คำว่า art มีจำนวนแอทริบิวต์ทั้งหมด 1103 แอทริบิวต์ เมื่อเลือกการกรองแอทริบิวต์เท่ากับ 40 แอทริบิวต์หมายถึงต้องการจำนวนแอทริบิวต์ที่ใช้เพียง 40 แอทริบิวต์เท่านั้น ในการทดลองผู้ใช้สามารถเลือกจำนวนแอทริบิวต์ตามที่ต้องการได้เช่น 50 100 150 และ 500 เป็นต้น

ผลการทดลองสามารถสรุปได้ว่าเมื่อใช้การกรองแอทริบิวต์จะให้ค่าความถูกต้องสูงกว่าเมื่อไม่กรองแอทริบิวต์เช่น ตารางที่ 5.6 และภาพประกอบ 5.14 เมื่อกรองแอทริบิวต์แบบ GainRatioAttributeEval และใช้อัลกอริทึม RBF ใน

การจำแนกโดยกรองให้มีจำนวนแอทริบิวต์ 40 แอทริบิวต์ ค่าความถูกต้องในการจำแนกคือ 87.01% มากว่าเมื่อไม่กรองแอทริบิวต์จำนวน 1103 ให้ค่าความถูกต้องเพียง 54.32%

ตารางที่ 5.6 ตารางแสดงค่าความถูกต้องของการจำแนกความหมายเมื่อกรองแอทริบิวต์ให้มีจำนวนต่างๆ และไม่กรองแอทริบิวต์

จำนวนแอทริบิวต์	Accuracy
40	87.01 %
50	76.92 %
60	79.8 %
100	79.32 %
150	82.4 %
200	82.21 %
300	81.73 %
500	75.96 %
1103	54.32 %



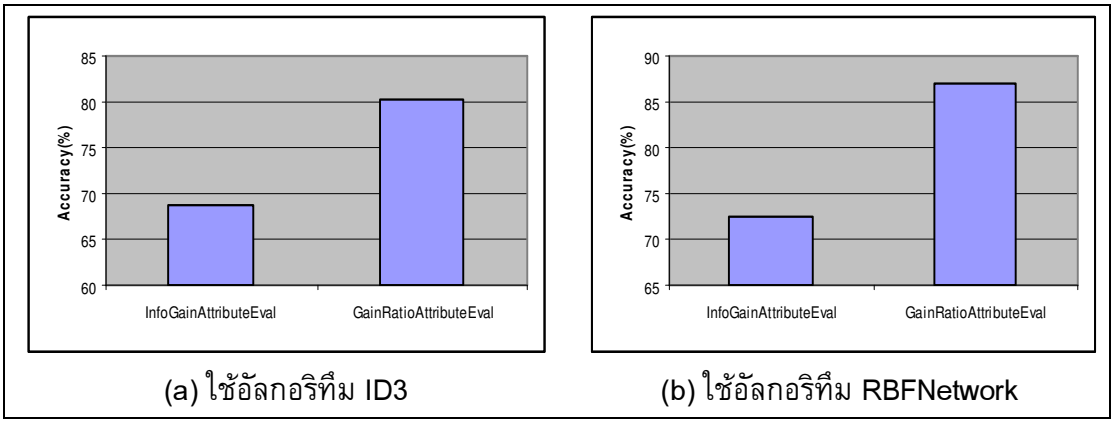
ภาพประกอบ 5.14 แสดงค่าความถูกต้อง เปรียบเทียบการกรองแอทริบิวต์จำนวนต่างๆ และไม่กรองแอทริบิวต์ (ค่าสุดท้าย) โดยใช้อัลกอริทึม RBFNetwork กรองแบบ GainRatioAttributeEval

4) ประเด็นการเลือกเทคนิคการกรองแอทริบิวต์

ในการทดลองทั้งสองแบบคือ GainRatioAttributeEval และ InfoGainAttributeEval จากผลการทดลอง เมื่อกรองแอทริบิวต์เท่ากับ 40 แอทริบิวต์ อัลกอริทึม GainRatioAttributeEval ให้ค่าความถูกต้องสูงกว่า อัลกอริทึม InfoGainAttributeEval ดังตารางที่ 5.7 และภาพประกอบ 5.15(a) สำหรับอัลกอริทึม ID3 เมื่อกรองด้วย GainRatioAttributeEval ให้ค่าความถูกต้อง 80.28% ซึ่งมากกว่าการกรองด้วย InfoGainAttributeEval คือ 68.75% สำหรับอัลกอริทึม RBFNetwork ผลการทดลองเป็นไปในทำนองเดียวกันกล่าวคือ GainRatioAttributeEval ให้ค่าความถูกต้อง 87.01% มากว่าการกรองด้วย InfoGainAttributeEval ให้ค่าความถูกต้อง 72.59% เป็นต้น

ตารางที่ 5.7 ตารางแสดงค่าความถูกต้องของการจำแนกความหมายเมื่อเปรียบเทียบการกรองแบบ InfoGainAttributeEval และ GainRatioAttributeEval

Algorithm	Accuracy	
	ID3	RBFNetwork
InfoGainAttributeEval	68.75 %	72.59 %
GainRatioAttributeEval	80.28 %	87.01 %



(a) ใช้อัลกอริทึม ID3 (b) ใช้อัลกอริทึม RBFNetwork

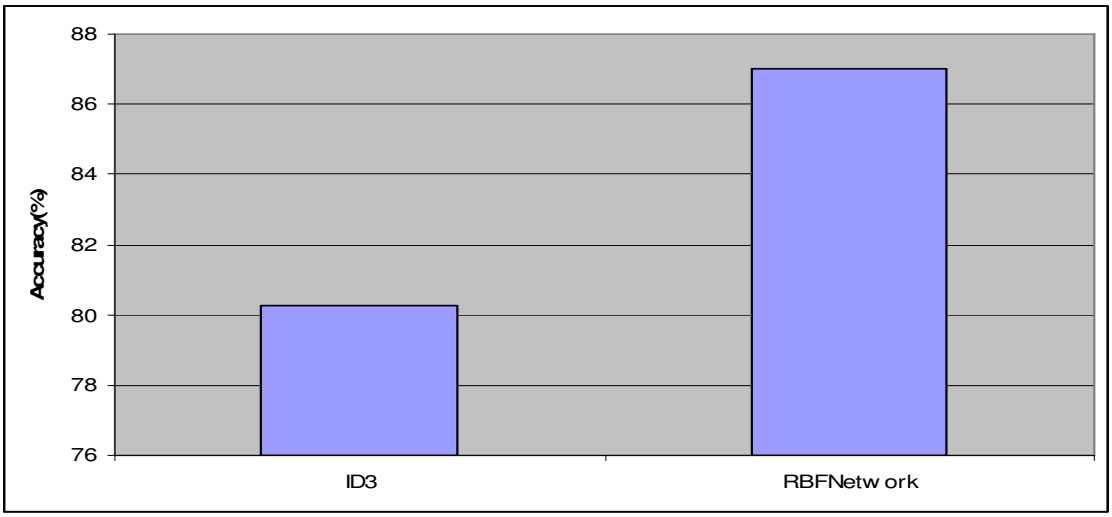
ภาพประกอบ 5.15 แสดงค่าความถูกต้อง เปรียบเทียบการกรองแบบInfoGainAttributeEval และ GainRatioAttributeEval

5) ประเด็นการเลือกอัลกอริทึมการจำแนกความหมาย

การจำแนกความหมายโดยใช้อัลกอริทึม RBFNetwork ให้ค่าความถูกต้องสูงกว่าอัลกอริทึม ID3 กำหนดค่าการกรองแอทริบิวต์เท่ากับ 40 แอทริบิวต์และเลือกการกรองแบบ GainRatioAttributeEval ดังตารางที่ 5.8 และภาพประกอบ 5.16 เมื่อใช้อัลกอริทึม RBFNetwork ให้ค่าความถูกต้อง 87.01% มากว่าเมื่อใช้อัลกอริทึม ID3 ให้ค่าความถูกต้อง 80.28% เป็นต้น

ตารางที่ 5.8 ตารางแสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3และ RBFNetwork

Algorithm	Accuracy
ID3	80.28 %
RBFNetwork	87.01 %



ภาพประกอบ 5.16 แสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3 และ RBFNetwork

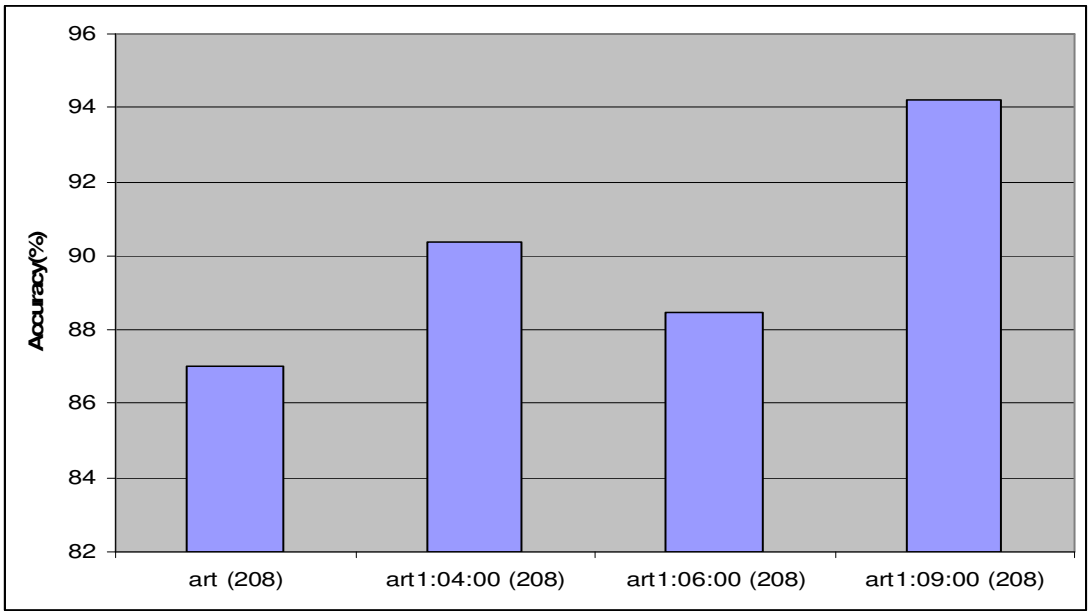
6) ประเด็นการเลือกจำนวนคลาส

ในการทดลองมีการจำแนกความหมาย 2 แบบ คือจำแนกโดยแบ่งกลุ่มเป็น 2 ความหมายและใช้ความหมายทั้งหมดของคำกำกวมที่มีอยู่ จากตารางที่ 5.9 และภาพประกอบ 5.17 คำกำกวม art มีความหมายทั้งหมด 3 ความหมาย คือ 1:04:00:: 1:06:00:: และ 1:09:00:: จากการทดลองโดยใช้อัลกอริทึม RBFNetwork ที่มีขนาดหน้าต่าง 4 ใช้การกรองแบบ GainRatioAttributeEval ที่มีจำนวนแอทริบิวต์เท่ากับ 40 แอทริบิวต์ ผลการทดลองพบว่า ค่าความถูกต้องโดยแบ่งกลุ่มเป็น 2 ความหมายมีค่าสูงกว่าค่าความถูกต้องโดยการใช้ความหมายทั้งหมดของคำกำวมที่มีอยู่ ตัวอย่างเช่นค่าความถูกต้องของคำกำกวม art เมื่อจำแนกโดยแบ่งกลุ่มเป็น 2 ความหมายมีดังนี้ ความหมายที่หนึ่ง (1:09:00::) ให้

ค่าความถูกต้อง 94.23% ความหมายที่สอง (1:04:00::) ให้ค่าความถูกต้อง 90.38% และความหมายที่สาม (1:06:00::) ให้ค่าความถูกต้อง 88.46% ซึ่งสูงกว่าใช้ความหมายทั้งหมดของคำกำกวมที่มีอยู่ที่ให้ค่าความถูกต้องเพียง 87.01% เท่านั้น

ตารางที่ 5.9 ตารางแสดงค่าความถูกต้องเปรียบเทียบการแบ่งกลุ่มแบบ 2 ความหมาย และความหมายทั้งหมด กรองแบบ GainRatioAttributeEval และใช้อัลกอริทึม RBFNetwork

Sense	Accuracy
art (208)	87.01 %
art1:04:00:: (208)	90.38 %
art1:06:00:: (208)	88.46 %
art1:09:00:: (208)	94.23 %



ภาพประกอบ 5.17 กราฟแสดงค่าความถูกต้องเปรียบเทียบการแบ่งกลุ่มแบบ 2 ความหมาย และความหมายทั้งหมด กรองแบบ GainRatioAttributeEval และใช้อัลกอริทึม RBFNetwork

5.2 ตัวอย่างคำกำกวม dyke

คำกำกวม dyke ของคลังข้อความ Senseval-2 ประกอบด้วยความหมายกำกวม คือ 1:06:00:: (กำแพงกั้นน้ำ) 1:18:00:: (เลสเบี้ยน)

ขั้นตอนที่ 2: สร้างแอทริบิวต์โดยใช้คำบริบท

2.1 กำหนดขนาดหน้าต่างของคำบริบทตามต้องการ ในที่นี้สมมติเลือกขนาดหน้าต่างเป็น 4

2.2 ตัดประโยคจากไฟล์ .count ให้มีคำบริบทเพื่อสร้างแอทริบิวต์ ซึ่งแบ่งเป็น 3 แบบดังนี้

แบบที่ 1 ใช้บริบททางซ้ายเท่านั้นดังภาพประกอบ 5.18

onto Marken joined mainland **dyke**
Similarly assuming magnetizations microdiorite **dykes**
haymeadows damp pastures intersected **dykes**
masochists self proclaimed fierce **dykes**
miles Flat fields interspersed **dykes**
gross darling tell lucky **dyke**
swarm east west trending **dykes**

ภาพประกอบ 5.18 คำบริบททางซ้าย

แบบที่ 2 ใช้บริบททางขวาเท่านั้นดังภาพประกอบ 5.19

dyke
dykes acquired initial stages brittle
dykes patrolled dragonflies summer submerged
dykes screamed love lady
dykes gleaming June sunshine spread
dyke putting roses cheek
dykes North Barra South Uist

ภาพประกอบ 5.19 คำบริบททางขวา

แบบที่ 3 ใช้บริบททั้งทางซ้ายและขวาดังภาพประกอบ 5.20

onto Marken joined mainland **dyke**
Similarly assuming magnetizations microdiorite **dykes** acquired initial stages brittle
haymeadows damp pastures intersected **dykes** patrolled dragonflies summer
submerged
masochists self proclaimed fierce **dykes** screamed love lady
miles Flat fields interspersed **dykes** gleaming June sunshine spread
gross darling tell lucky **dyke** putting roses cheek
swarm east west trending **dykes** North Barra South Uist

ภาพประกอบ 5.20 คำบริบททางซ้ายและขวา

2.3 ใช้โปรแกรม SenseTools และ โปรแกรม NSP ในการแปลงข้อความดังกล่าวให้อยู่ในรูปแบบ Feature Vectors ตัวอย่างดังภาพประกอบ 5.21 ซึ่งใช้บริบททั้งทางซ้ายและขวา (แบบที่3) คำบริบททั้งหมดถูกสร้างเป็นแอทริบิวต์แต่ละแอทริบิวต์มีค่าที่เป็นไปได้ 2 ค่าคือ 0 และ 1 เช่น “@attribute 'dyke' {0,1}” หมายถึงแอทริบิวต์ dyke มีค่าที่เป็นไปได้คือ 0 และ 1 แอทริบิวต์สุดท้ายคือแอทริบิวต์ senseclass บอกความหมายของคำกำกวมซึ่งมีค่าที่เป็นไปได้ 2 ค่า คือ dyke~1:06:00:: และ dyke~1:18:00

```
@relation 'RELATION'
@attribute 'dykes' {0,1}
@attribute 'dyke' {0,1}
@attribute 'joined' {0,1}
@attribute 'lucky' {0,1}
@attribute 'patrolled' {0,1}
@attribute 'submerged' {0,1}
@attribute 'pastures' {0,1}
@attribute 'damp' {0,1}
@attribute 'stages' {0,1}
@attribute 'sunshine' {0,1}
```

ภาพประกอบ 5.21 สร้างแอทริบิวต์โดยใช้คำบริบททั้งทางซ้ายและขวา

```
@attribute 'spread' {0,1}
@attribute 'interspersed' {0,1}
@attribute 'brittle' {0,1}
@attribute 'swarm' {0,1}
@attribute 'lady' {0,1}
@attribute 'fierce' {0,1}
@attribute 'proclaimed' {0,1}
@attribute 'west' {0,1}
@attribute 'east' {0,1}
@attribute 'June' {0,1}
@attribute 'masochists' {0,1}
@attribute 'gross' {0,1}
@attribute 'Uist' {0,1}
@attribute 'microdiorite' {0,1}
@attribute 'darling' {0,1}
@attribute 'Marken' {0,1}
@attribute 'intersected' {0,1}
@attribute 'fields' {0,1}
@attribute 'Barra' {0,1}
@attribute 'love' {0,1}
@attribute 'onto' {0,1}
@attribute 'acquired' {0,1}
@attribute 'miles' {0,1}
@attribute 'screamed' {0,1}
@attribute 'Flat' {0,1}
```

ภาพประกอบ 5.21 สร้างแอทริบิวต์โดยใช้คำบริบททั้งทางซ้ายและขวา (ต่อ)

```
@attribute 'North' {0,1}
@attribute 'putting' {0,1}
@attribute 'tell' {0,1}
@attribute 'Similarly' {0,1}
@attribute 'roses' {0,1}
@attribute 'summer' {0,1}
@attribute 'self' {0,1}
@attribute 'magnetizations' {0,1}
@attribute 'mainland' {0,1}
@attribute 'South' {0,1}
@attribute 'trending' {0,1}
@attribute 'cheek' {0,1}
@attribute 'gleaming' {0,1}
@attribute 'haymeadows' {0,1}
@attribute 'assuming' {0,1}
@attribute 'initial' {0,1}
@attribute 'dragonflies' {0,1}
@attribute 'senseclass' {dyke~1:06:00::, dyke~1:18:00::}
@data
{0 1, 8 1, 12 1, 23 1, 31 1, 38 1, 42 1, 44 1, 49 1, 50 1, 52 dyke~1:06:00::}
{0 1, 4 1, 5 1, 6 1, 7 1, 26 1, 35 1, 40 1, 48 1, 51 1, 52 dyke~1:06:00::}
{0 1, 14 1, 15 1, 16 1, 20 1, 29 1, 33 1, 41 1, 52 dyke~1:18:00::}
{0 1, 9 1, 10 1, 11 1, 19 1, 27 1, 32 1, 34 1, 47 1, 52 dyke~1:06:00::}
{1 1, 2 1, 3 1, 21 1, 24 1, 36 1, 37 1, 39 1, 52 dyke~1:18:00::}
{0 1, 1 1, 13 1, 17 1, 18 1, 22 1, 28 1, 35 1, 44 1, 45 1, 52 dyke~1:06:00::}
```

แอทริบิวต์ที่ 0 ถึง 52

ภาพประกอบ 5.21 สร้างแอทริบิวต์โดยใช้คำบริบททั้งทางซ้ายและขวา (ต่อ)

ขั้นตอนที่ 3: เลือกแอทริบิวต์

เมื่อได้ข้อมูลในรูปแบบ arff แล้วกรองแอทริบิวต์ด้วยตัวกรอง 2 แบบคือ InfoGainAttributeEval และ GainRatioAttributeEval สมมติกรองให้มีจำนวนแอทริบิวต์เท่ากับ 30 จะเห็นว่าจำนวนแอทริบิวต์จาก 53 แอทริบิวต์ (แอทริบิวต์ที่ 0 ถึง 52) ดังภาพประกอบ 5.21 ลดลงเป็น 30 แอทริบิวต์ (แอทริบิวต์ที่ 0 ถึง 29) ดังภาพประกอบ 5.22

```
@relation 'RELATION'

@attribute lady {0,1}
@attribute screamed {0,1}
@attribute roses {0,1}
@attribute putting {0,1}
@attribute tell {0,1}
@attribute masochists {0,1}
@attribute darling {0,1}
@attribute gross {0,1}
@attribute fierce {0,1}
@attribute love {0,1}
@attribute proclaimed {0,1}
@attribute lucky {0,1}
@attribute self {0,1}
@attribute North {0,1}
@attribute South {0,1}
@attribute June {0,1}
@attribute pastures {0,1}
@attribute east {0,1}
@attribute submerged {0,1}
@attribute Uist {0,1}
@attribute patrolled {0,1}
@attribute interspersed {0,1}
@attribute spread {0,1}
@attribute brittle {0,1}
```

ภาพประกอบ 5.22 กรองแอทริบิวต์ให้มีจำนวนแอทริบิวต์ 30 แอทริบิวต์

```
@attribute swarm {0,1}
@attribute damp {0,1}
@attribute west {0,1}
@attribute sunshine {0,1}
@attribute stages {0,1}
@attribute senseclass {dyke~1:06:00::,dyke~1:18:00::}
@data
{14 1,23 1,28 1, 29 dyke~1:06:00::}
{13 1,16 1,18 1,20 1,25 1, 29 dyke~1:06:00::}
{0 1,1 1,5 1,8 1,9 1,10 1,12 1,29 dyke~1:18:00::}
{15 1,21 1,22 1,27 1, 29 dyke~1:06:00::}
{2 1,3 1,4 1,6 1,7 1,11 1,29 dyke~1:18:00::}
{13 1,14 1,17 1,19 1,24 1,26 1, 29 dyke~1:06:00::}
```

แอทริบิวต์ที่ 0 ถึง 29

ภาพประกอบ 5.22 กรองแอทริบิวต์ให้มีจำนวนแอทริบิวต์ 30 แอทริบิวต์ (ต่อ)

ขั้นตอนที่ 4: จำแนกความหมาย

จำแนกความหมายของคำที่กำกวมโดยใช้ 10 folds-cross validation โดยเลือกวิธีการจำแนกความหมาย 2 แบบ คือการจำแนกโดยใช้สองความหมายและใช้ความหมายทั้งหมด อัลกอริทึมในการจำแนกความหมาย 2 อัลกอริทึมคือ RBFNetwork และ ID3

ผลการทดลองสามารถสรุปได้ 5 ประเด็นคือประเด็นการเลือกขนาดหน้าต่างคำบริบท ประเด็นการเลือกรูปแบบหน้าต่างคำบริบท ประเด็นจำนวนแอทริบิวต์สำหรับการกรองแอทริบิวต์ ประเด็นการเลือกเทคนิคการกรองแอทริบิวต์ และประเด็นการเลือกอัลกอริทึมการจำแนกความหมาย เนื่องจากคำว่า dyke มี 2 ความหมายจึงไม่แสดงประเด็นการเลือกจำนวนคลาส

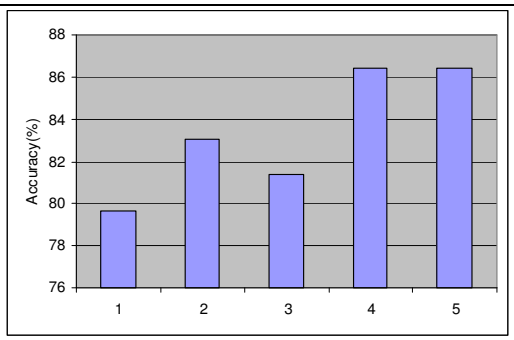
1) ประเด็นการเลือกขนาดหน้าต่างคำบริบท

เมื่อความกว้างขนาดหน้าต่างเพิ่มขึ้นค่าความถูกต้องมีแนวโน้มเพิ่มขึ้นดังตัวอย่างตารางที่ 5.10 และภาพประกอบ 5.23 (b) ของบริบททางซ้าย ใช้อัลกอริทึม ID3 กรองแบบ GainRatioAttributeEval ที่ขนาดหน้าต่าง 1 2 3 4 และ 5 มีค่าความถูกต้องคือ 79.16% 83.1% 83.1% 86.4% และ 86.4% เป็นต้น การใช้บริบททางขวาแสดงดังตารางที่ 5.11 และภาพประกอบ 5.24 และการใช้บริบททั้งทางซ้ายและขวาแสดงดังตารางที่ 5.12 และภาพประกอบ 5.25 จะเห็นได้ว่าทั้งอัลกอริทึม ID3 และ RBFNetwork ของการ

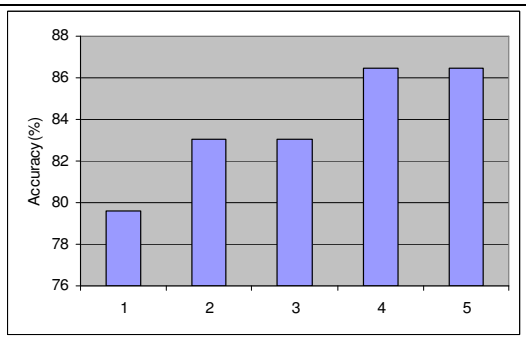
กรองทั้งสองแบบคือ InfoGainAttributeEval และ GainRatioAttributeEval ของบริบททางขวาและบริบททั้งทางซ้ายและขวา มีแนวโน้มการทำงานในทำนองเดียวกัน อย่างไรก็ตามถ้าขนาดหน้าต่างมากเกินไปอาจส่งผลต่อค่าความถูกต้องที่ลดลงได้เช่นกัน

ตารางที่ 5.10 ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางซ้าย ที่ขนาดหน้าต่าง 1 2 3 4 และ 5

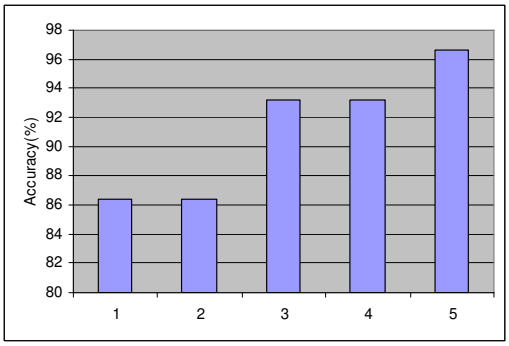
window size	Accuracy			
	ID3		RBFNetwork	
	InfoGainAttributeEval	GainRatioAttributeEval	InfoGainAttributeEval	GainRatioAttributeEval
1	79.7 %	79.6 %	86.4%	86.4%
2	83.1 %	83.1 %	86.4 %	86.4 %
3	81.4 %	83.1 %	93.2 %	91.5 %
4	86.4%	86.4 %	93.2 %	89.8 %
5	86.4 %	86.4 %	96.6 %	89.8 %



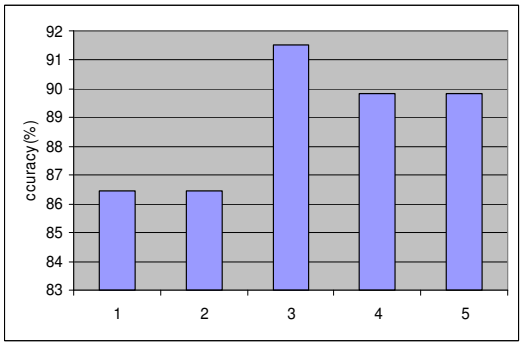
(a) ใช้ ID3 ในการจำแนกและกรองโดย InfoGainAttributeEval



(b) ใช้ ID3 ในการจำแนกและกรองโดย GainRatioAttributeEval



(c) ใช้ RBF ในการจำแนกและกรองโดย InfoGainAttributeEval

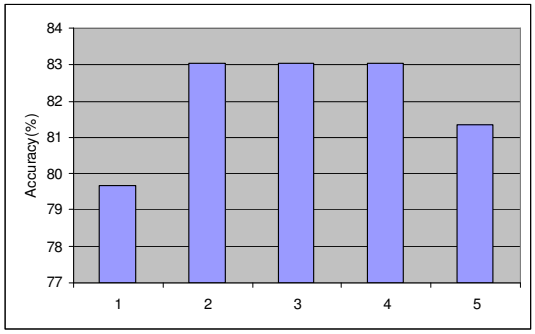


(d) ใช้ RBF ในการจำแนกและกรองโดย GainRatioAttributeEval

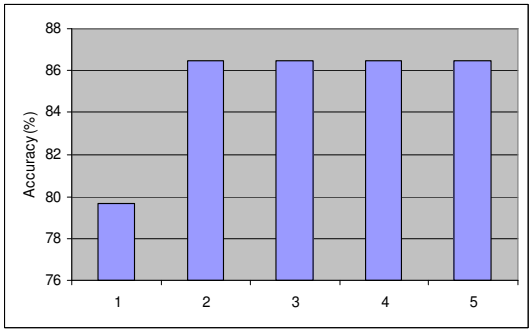
ภาพประกอบ 5.23 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่างๆ โดยใช้บริบททางซ้าย

ตารางที่ 5.11 ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางขวา ที่ขนาดหน้าต่าง 1 2 3 4 และ 5

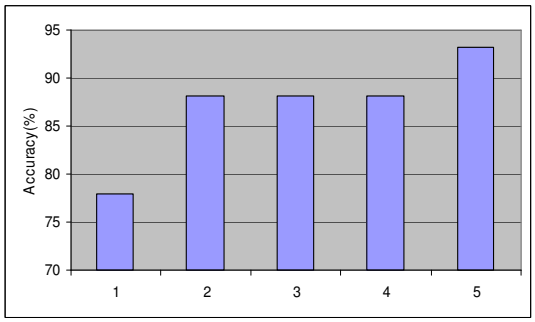
window size	Accuracy			
	ID3		RBFNetwork	
	InfoGainAttributeEval	GainRatioAttributeEval	InfoGainAttributeEval	GainRatioAttributeEval
1	79.7 %	79.7 %	77.9 %	77.9 %
2	83.1 %	86.4 %	88.1 %	84.7 %
3	83.1 %	86.4 %	88.1 %	88.1 %
4	83.1 %	86.4 %	88.1 %	89.83 %
5	81.4 %	86.4 %	93.2 %	91.52 %



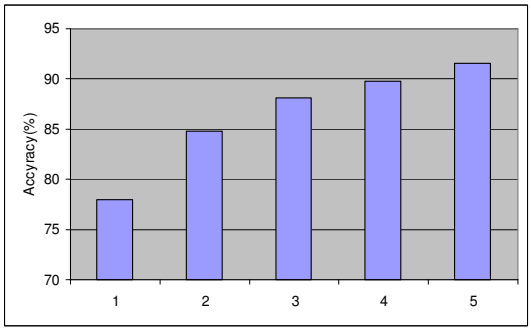
(a) ใช้ ID3 ในการจำแนกและกรองโดย InfoGainAttributeEval



(b) ใช้ ID3 ในการจำแนกและกรองโดย GainRatioAttributeEval



(c) ใช้ RBF ในการจำแนกและกรองโดย InfoGainAttributeEval

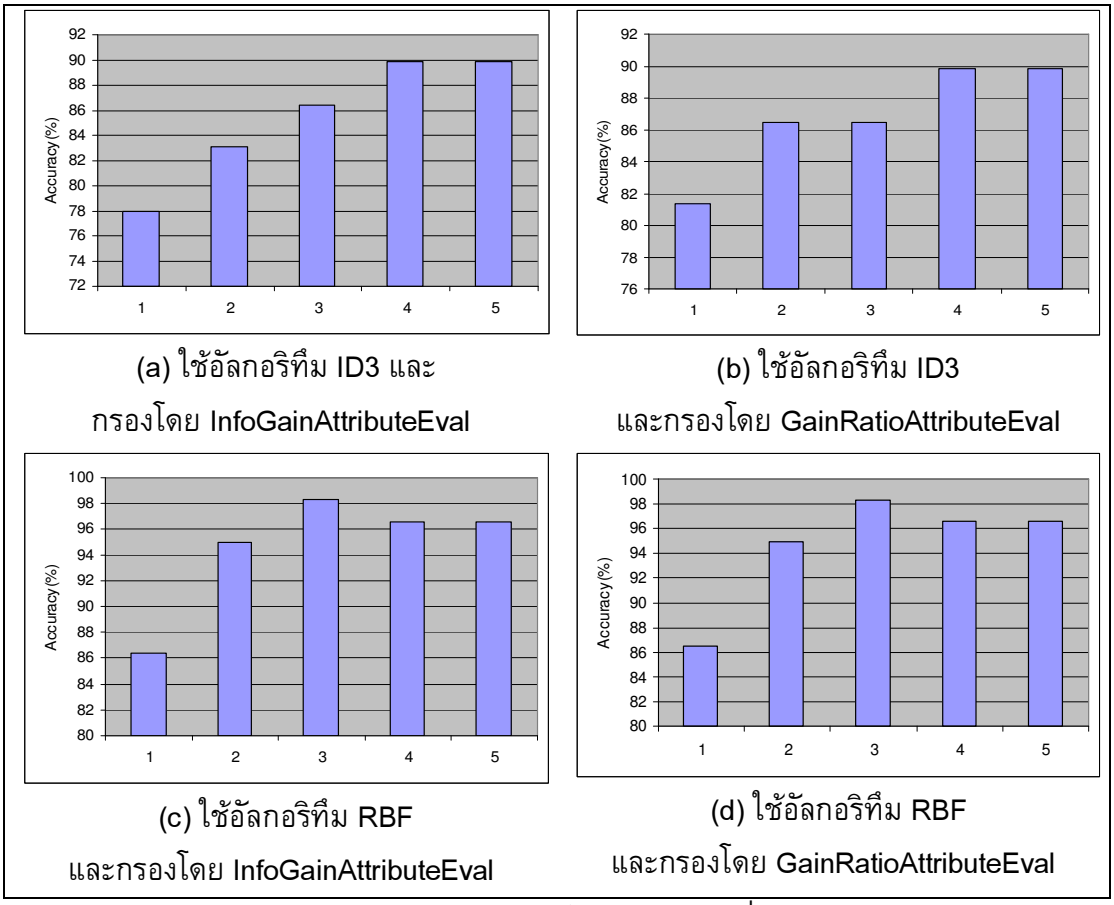


(d) ใช้ RBF ในการจำแนกและกรองโดย GainRatioAttributeEval

ภาพประกอบ 5.24 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่างๆ โดยใช้บริบททางขวา

ตารางที่ 5.12 ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททั้งทางซ้ายและขวา ที่ขนาดหน้าต่าง 1 2 3 4 และ 5

window size	Accuracy			
	ID3		RBFNetwork	
	InfoGainAttributeEval	GainRatioAttributeEval	InfoGainAttributeEval	GainRatioAttributeEval
1	77.96 %	81.4 %	86.4 %	86.4 %
2	83.05 %	86.44 %	93.2 %	94.91 %
3	86.44 %	86.44 %	98.3 %	98.3 %
4	89.83 %	89.93 %	93.2 %	96.6 %
5	89.83 %	89.93 %	96.6 %	96.6 %



(a) ใช้อัลกอริทึม ID3 และกรองโดย InfoGainAttributeEval

(b) ใช้อัลกอริทึม ID3 และกรองโดย GainRatioAttributeEval

(c) ใช้อัลกอริทึม RBF และกรองโดย InfoGainAttributeEval

(d) ใช้อัลกอริทึม RBF และกรองโดย GainRatioAttributeEval

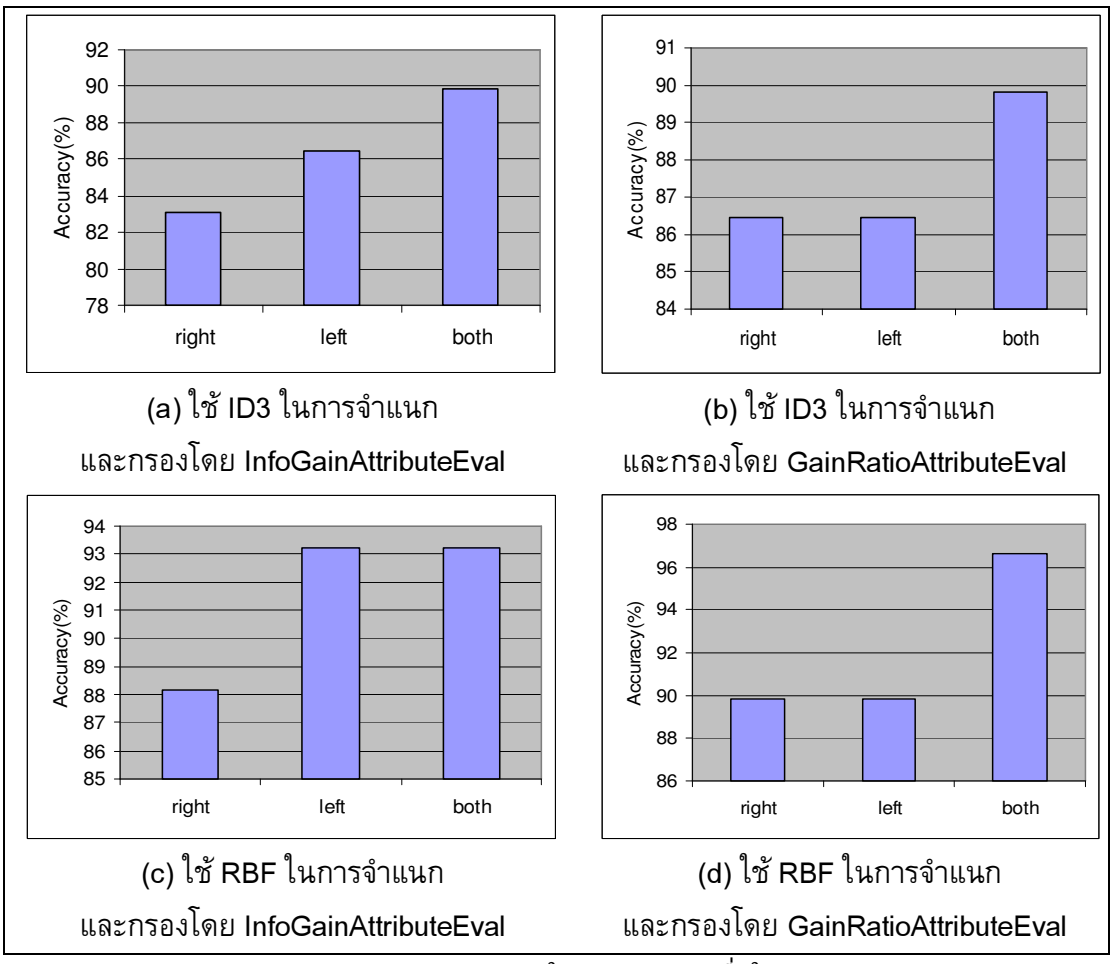
ภาพประกอบ 5.25 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่างๆ โดยใช้บริบททั้งทางซ้ายและขวา

2) ประเด็นการเลือกรูปแบบหน้าต่างคำบริบท

ถ้าเลือกขนาดหน้าต่างเท่ากับ 4 การใช้หน้าต่างคำบริบททั้งซ้ายและขวาจะได้ค่าความถูกต้องสูงกว่าเลือกหน้าต่างคำบริบททางด้านซ้ายหรือขวาเพียงอย่างเดียวดังตัวอย่างตารางที่ 5.13 และภาพประกอบ 5.26 (d) ใช้อัลกอริทึม RBFNetwork กรองแบบ GainRatioAttributeEval เมื่อใช้บริบททางขวามีค่าความถูกต้อง 89.8% ใช้คำบริบททางซ้ายมีค่าความถูกต้อง 89.8% และใช้คำบริบททั้งทางซ้ายและขวามีค่าความถูกต้องคือ 96.6% ข้อสังเกต ส่วนใหญ่ประเภทหน้าต่างคำบริบทแบบทางขวามีค่าความถูกต้องน้อยที่สุดเนื่องจากคำที่ใช้ขยายคำกำกวมส่วนใหญ่มักอยู่ทางซ้ายมือของคำกำกวมตามหลักการเขียนของภาษาอังกฤษ

ตารางที่ 5.13 ตารางแสดงค่าความถูกต้องการจำแนกความหมายของการใช้บริบททางขวา ทางซ้าย และทั้งทางซ้ายและขวา เมื่อขนาดหน้าต่างเท่ากับ 4

window size	Accuracy			
	ID3		RBFNetwork	
	InfoGainAttributeEval	GainRatioAttributeEval	InfoGainAttributeEval	GainRatioAttributeEval
Right	83.05 %	86.4 %	88.13 %	89.8 %
Left	86.44 %	86.4 %	93.2 %	89.8 %
Both	89.83 %	89.83 %	93.2 %	96.6 %



(a) ใช้ ID3 ในการจำแนก และกรองโดย InfoGainAttributeEval

(b) ใช้ ID3 ในการจำแนก และกรองโดย GainRatioAttributeEval

(c) ใช้ RBF ในการจำแนก และกรองโดย InfoGainAttributeEval

(d) ใช้ RBF ในการจำแนก และกรองโดย GainRatioAttributeEval

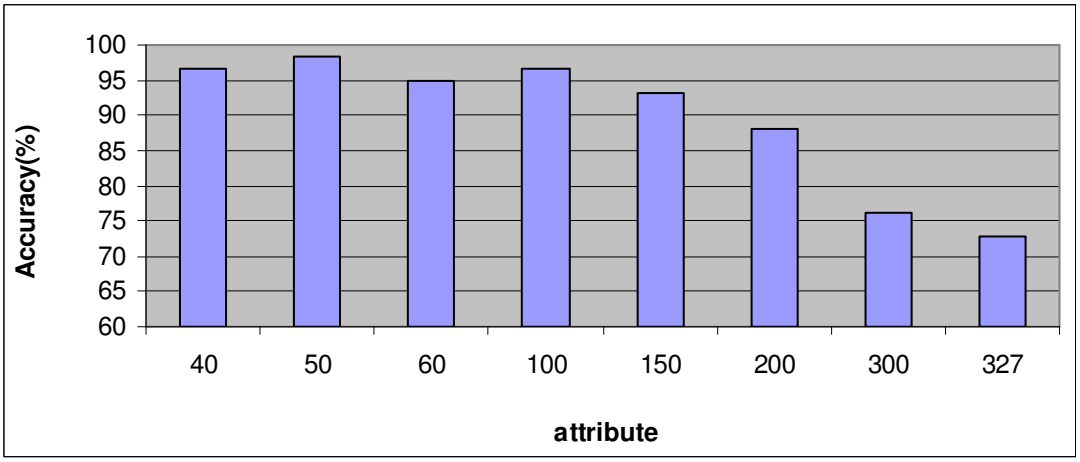
ภาพประกอบ 5.26 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อใช้บริบททางขวา ทางซ้าย และทั้งทางซ้ายและขวา เมื่อขนาดหน้าต่างความกว้างเท่ากับ 4

3) ประเด็นจำนวนแอทริบิวต์สำหรับการกรองแอทริบิวต์

ผลการทดลองสามารถสรุปได้ว่า เมื่อใช้การกรองแอทริบิวต์จะให้ค่าความถูกต้องสูงกว่าเมื่อไม่กรองแอทริบิวต์เช่น ตารางที่ 5.14 และภาพประกอบ 5.27 เมื่อกรองแอทริบิวต์แบบ GainRatioAttributeEval และใช้อัลกอริทึม RBF ในการจำแนก โดยกรองให้มีจำนวนแอทริบิวต์ 40 แอทริบิวต์ ค่าความถูกต้องในการจำแนกคือ 96.61% มากว่าเมื่อไม่กรองแอทริบิวต์จำนวน 327 ให้ค่าความถูกต้องเพียง 72.88%

ตารางที่ 5.14 ตารางแสดงค่าความถูกต้องของการจำแนกความหมายเมื่อกรองแอทริบิวต์ให้มีจำนวนต่างๆ และไม่กรองแอทริบิวต์

จำนวนแอทริบิวต์	Accuracy
40	96.61 %
50	98.3 %
60	94.91 %
100	96.61 %
150	93.22 %
200	88.13 %
300	76.27 %
327	72.88 %



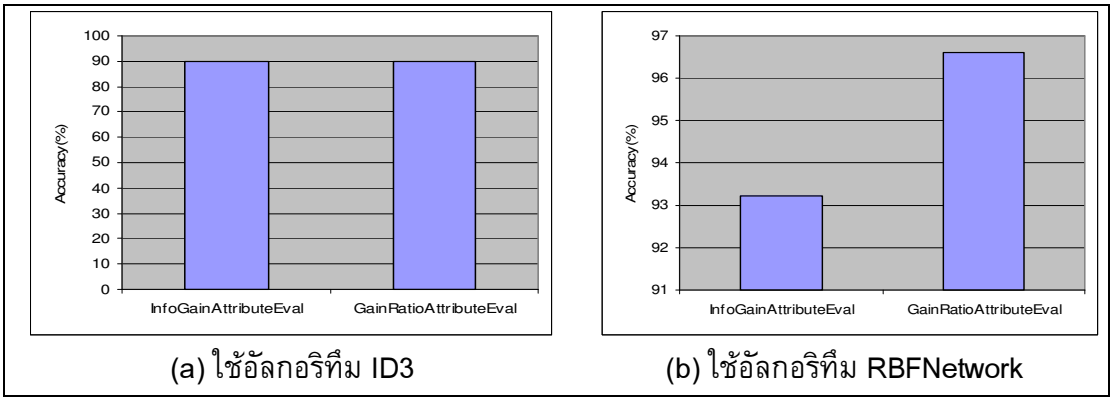
ภาพประกอบ 5.27 แสดงค่าความถูกต้อง เปรียบเทียบการกรองแอทริบิวต์จำนวนต่างๆ และไม่กรองแอทริบิวต์ (ค่าสุดท้าย) โดยใช้อัลกอริทึม RBFNetwork GainRatioAttributeEval

4) ประเด็นการเลือกเทคนิคการกรองแอทริบิวต์

ในการทดลองทั้งสองแบบคือ GainRatioAttributeEval และ InfoGainAttributeEval จากผลการทดลอง เมื่อกรองแอทริบิวต์เท่ากับ 40 แอทริบิวต์ อัลกอริทึม GainRatioAttributeEval ให้ค่าความถูกต้องสูงกว่าหรือเท่ากับอัลกอริทึม InfoGainAttributeEval ดังตารางที่ 5.15 และภาพประกอบ 5.28 (a) สำหรับอัลกอริทึม ID3 เมื่อกรองด้วย GainRatioAttributeEval ให้ค่าความถูกต้อง 89.83% ซึ่งเท่ากับการกรองด้วย InfoGainAttributeEval คือ 89.83% สำหรับอัลกอริทึม RBFNetwork เทคนิค GainRatioAttributeEval ให้ค่าความถูกต้อง 96.61% มากว่าการกรองด้วย InfoGainAttributeEval ให้ค่าความถูกต้อง 93.22% เป็นต้น

ตารางที่ 5.15 ตารางแสดงค่าความถูกต้องของการจำแนกความหมายเมื่อเปรียบเทียบการกรองแบบInfoGainAttributeEval และ GainRatioAttributeEval

Algorithm	Accuracy	
	ID3	RBFNetwork
InfoGainAttributeEval	89.83 %	93.22 %
GainRatioAttributeEval	89.83 %	96.61 %



(a) ใช้อัลกอริทึม ID3 (b) ใช้อัลกอริทึม RBFNetwork

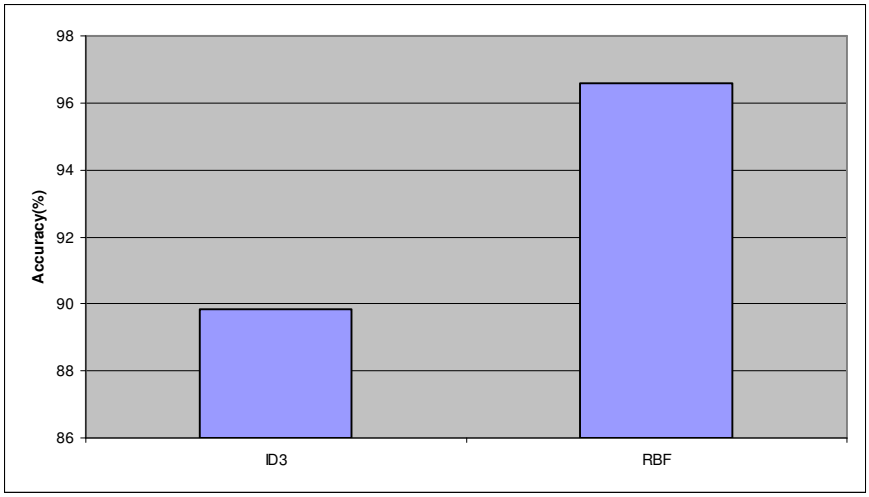
ภาพประกอบ 5.28 แสดงค่าความถูกต้อง เปรียบเทียบการกรองแบบ InfoGainAttributeEval และ GainRatioAttributeEval

5) ประเด็นการเลือกอัลกอริทึมการจำแนกความหมาย

การจำแนกความหมายโดยใช้อัลกอริทึม RBFNetwork ให้ค่าความถูกต้องสูงกว่าอัลกอริทึม ID3 กำหนดค่าการกรองแอทริบิวต์เท่ากับ 40 แอทริบิวต์และเลือกการกรองแบบ GainRatioAttributeEval ดังตารางที่ 5.16 และภาพประกอบ 5.29 เมื่อใช้อัลกอริทึม RBFNetwork ให้ค่าความถูกต้อง 96.61% มากว่าเมื่อใช้อัลกอริทึม ID3 ให้ค่าความถูกต้อง 89.83% เป็นต้น

ตารางที่ 5.16 แสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3และ RBFNetwork

Algorithm	Accuracy
ID3	89.83 %
RBFNetwork	96.61 %



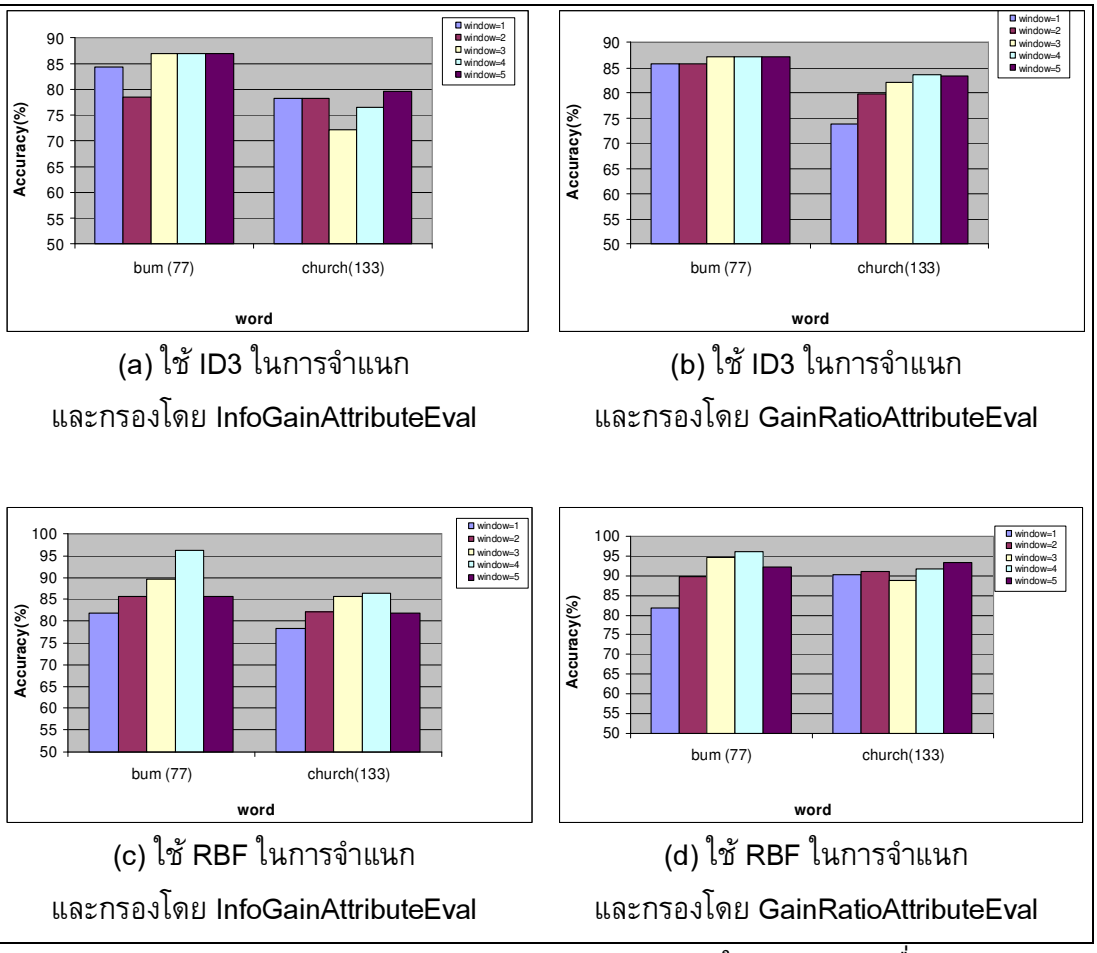
ภาพประกอบ 5.29 แสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3 และ RBFNetwork

5.3 คำกำกวม bum และ church

ผลการทดลองคำกำกวมอื่นคือ bum และ church ทดลองแบบ 10 Folds cross-validation สามารถสรุปรายละเอียดได้ 6 ประเด็นดังนี้

1) ประเด็นการเลือกขนาดหน้าต่างคำบริบท

เมื่อความกว้างขนาดหน้าต่างเพิ่มขึ้นค่าความถูกต้องมีแนวโน้มเพิ่มขึ้นดังตัวอย่างภาพประกอบ 5.30(b) ของบริบททั้งทางซ้ายและขวา ใช้อัลกอริทึม ID3 กรองแบบ GainRatioAttributeEval ของคำกำกวม church ที่ขนาดหน้าต่าง 1 2 3 4 และ 5 มีค่าความถูกต้องคือ 73.7% 79.69% 81.95 % 83.45% และ 83.33% เป็นต้น จะเห็นได้ว่าทั้งอัลกอริทึม ID3 และ RBFNetwork ของการกรองทั้งสองแบบคือ InfoGainAttributeEval และ GainRatioAttributeEval มีแนวโน้มการทำงานในทำนองเดียวกัน อย่างไรก็ตามถ้าขนาดหน้าต่างมากเกินไปอาจส่งผลต่อค่าความถูกต้องที่ลดลงได้เช่นกัน



(a) ใช้ ID3 ในการจำแนก และกรองโดย InfoGainAttributeEval

(b) ใช้ ID3 ในการจำแนก และกรองโดย GainRatioAttributeEval

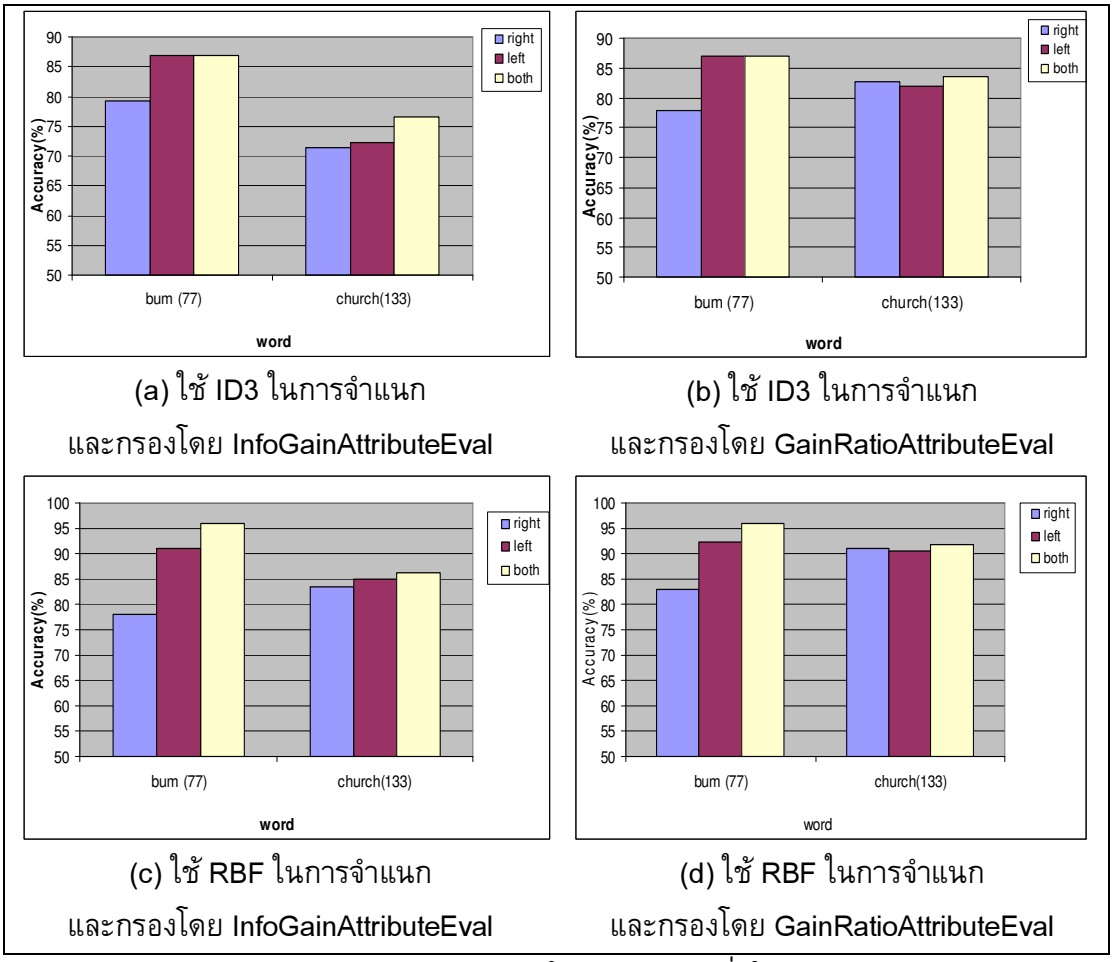
(c) ใช้ RBF ในการจำแนก และกรองโดย InfoGainAttributeEval

(d) ใช้ RBF ในการจำแนก และกรองโดย GainRatioAttributeEval

ภาพประกอบ 5.30 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อขนาดหน้าต่างมีขนาดต่าง ๆโดยใช้บริบททั้งทางซ้ายและขวา

2) ประเด็นการเลือกรูปแบบหน้าต่างคำบริบท

ถ้าเลือกขนาดหน้าต่างเท่ากับ 4 การใช้หน้าต่างคำบริบททั้งซ้ายและขวาจะได้ค่าความถูกต้องสูงกว่าเลือกหน้าต่างคำบริบททางด้านซ้ายหรือขวาเพียงอย่างเดียวตัวอย่างเช่น ภาพประกอบ 5.31(d) ใช้อัลกอริทึม RBFNetwork กรองแบบ GainRatioAttributeEval ของคำกำกวม bum เมื่อใช้บริบททางขวามีค่าความถูกต้อง 83.11% ใช้คำบริบททางซ้ายมีค่าความถูกต้อง 92.2% และใช้คำบริบททั้งทางซ้ายและขวามีค่าความถูกต้องคือ 96.1% ข้อสังเกต ส่วนใหญ่ประเภทหน้าต่างคำบริบทแบบทางขวามีค่าความถูกต้องน้อยที่สุดเนื่องจากคำที่ใช้ขยายคำกำกวมส่วนใหญ่มักอยู่ทางซ้ายมือของคำกำกวมตามหลักการเขียนของภาษาอังกฤษ



(a) ใช้ ID3 ในการจำแนก และกรองโดย InfoGainAttributeEval

(b) ใช้ ID3 ในการจำแนก และกรองโดย GainRatioAttributeEval

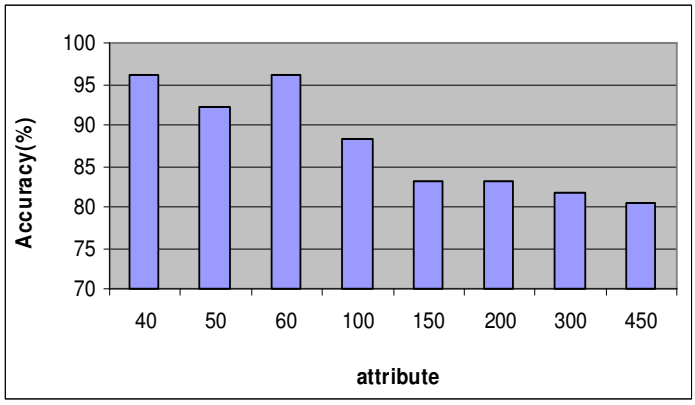
(c) ใช้ RBF ในการจำแนก และกรองโดย InfoGainAttributeEval

(d) ใช้ RBF ในการจำแนก และกรองโดย GainRatioAttributeEval

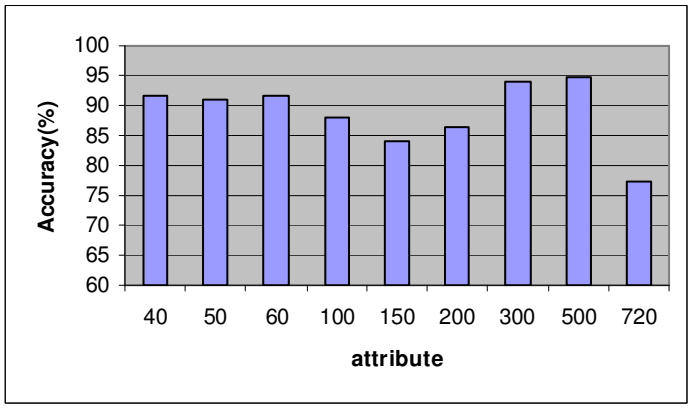
ภาพประกอบ 5.31 กราฟแสดงค่าความถูกต้องในการจำแนกเมื่อใช้บริบททางขวา ทางซ้าย และทั้งทางซ้ายและขวา เมื่อขนาดหน้าต่างความกว้างเท่ากับ 4

3) ประเด็นจำนวนแอทริบิวต์สำหรับการกรองแอทริบิวต์

ผลการทดลองสามารถสรุปได้ว่าเมื่อใช้การกรองแอทริบิวต์จะให้ค่าความถูกต้องสูงกว่าเมื่อไม่กรองแอทริบิวต์เช่น ภาพประกอบ 5.32 (b) ของคำกำกวม church เมื่อกรองแอทริบิวต์แบบ GainRatioAttributeEval และใช้อัลกอริทึม RBFNetwork ในการจำแนก โดยกรองให้มีจำนวนแอทริบิวต์ 40 แอทริบิวต์ ค่าความถูกต้องในการจำแนกคือ 94.69% มากว่าเมื่อไม่กรองแอทริบิวต์จำนวน 720 ให้ค่าความถูกต้องเพียง 77.27% จะเห็นได้ว่าคำกำกวม bum มีแนวโน้วการทำงานในทำนองเดียวกัน



(a) คำกำกวม bum

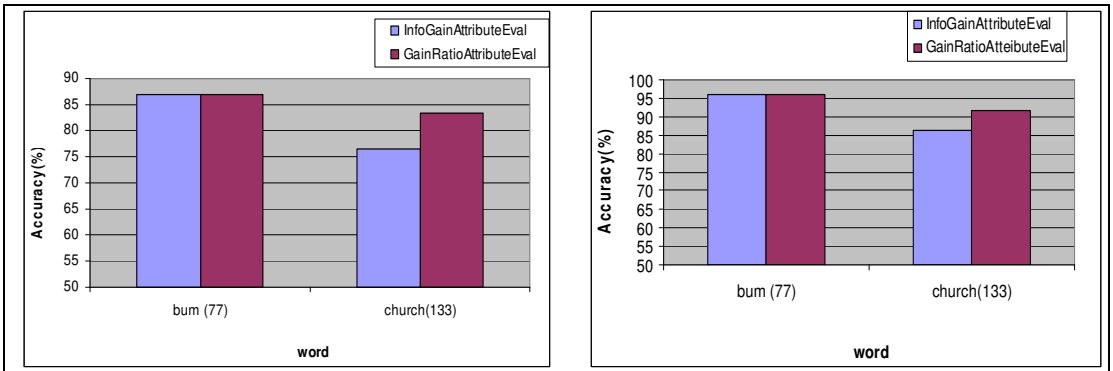


(b) คำกำกวม church

ภาพประกอบ 5.32 แสดงค่าความถูกต้อง เปรียบเทียบการกรองแอทริบิวต์จำนวนต่างๆ และไม่กรองแอทริบิวต์ (ค่าสุดท้าย) โดยใช้อัลกอริทึม RBFNetwork GainRatioAttributeEval

4) ประเด็นการเลือกเทคนิคการกรองแอทริบิวต์

ในการทดลองทั้งสองแบบคือ GainRatioAttributeEval และ InfoGainAttributeEval จากผลการทดลอง เมื่อกรองแอทริบิวต์เท่ากับ 40 แอทริบิวต์ อัลกอริทึม GainRatioAttributeEval ให้ค่าความถูกต้องสูงกว่าหรือเท่ากับอัลกอริทึม InfoGainAttributeEval ดังภาพประกอบ 5.33(a) ของคำกำกวม church สำหรับอัลกอริทึม ID3 เมื่อกรองด้วย GainRatioAttributeEval ให้ค่าความถูกต้อง 83.45% ซึ่งสูงกว่าการกรองด้วย InfoGainAttributeEval คือ 76.51% สำหรับอัลกอริทึม RBFNetwork ผลการทดลองเป็นไปในทำนองเดียวกันกล่าวคือ GainRatioAttributeEval ให้ค่าความถูกต้อง 91.1% มากว่าการกรองด้วย InfoGainAttributeEval ให้ค่าความถูกต้อง 86.4%เป็นต้น ส่วนคำกำกวม bum ทั้งเทคนิค InfoGainAttributeEval และ GainRatioAttributeEval ให้ค่าความถูกต้องเท่ากันทั้งอัลกอริทึม ID3 และ RBFNetwork

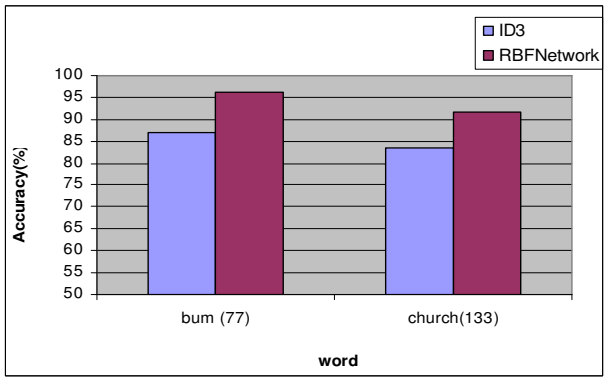


(a) ใช้อัลกอริทึม ID3 (b) ใช้อัลกอริทึม RBFNetwork

ภาพประกอบ 5.33 แสดงค่าความถูกต้อง เปรียบเทียบการกรองแบบ InfoGainAttributeEval และ GainRatioAttributeEval

5) ประเด็นการเลือกอัลกอริทึมการจำแนกความหมาย

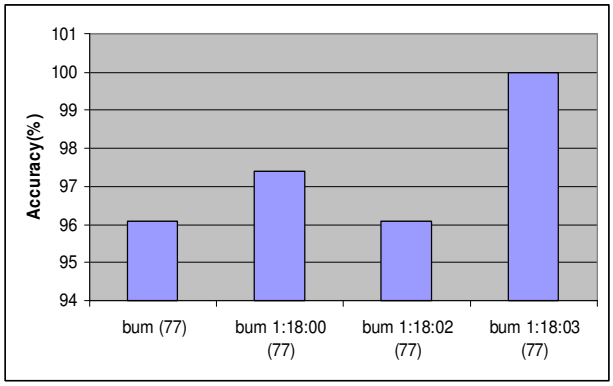
การจำแนกความหมายโดยใช้อัลกอริทึม RBFNetwork ให้ค่าความถูกต้องสูงกว่าอัลกอริทึม ID3 กำหนดค่าการกรองแอทริบิวต์เท่ากับ 40 แอทริบิวต์และเลือกการกรองแบบ GainRatioAttributeEval ดังภาพประกอบ 5.34 ของคำกำกวม bum เมื่อใช้อัลกอริทึม RBFNetwork ให้ค่าความถูกต้อง 96.61% มากว่าเมื่อใช้อัลกอริทึม ID3 ให้ค่าความถูกต้อง 87.01% เป็นต้น จะเห็นได้ว่าคำกำกวม bum มีแนวโน้มการทำงานในทำนองเดียวกัน



ภาพประกอบ 5.34 แสดงค่าความถูกต้อง เปรียบเทียบอัลกอริทึม ID3 และ RBFNetwork

6) ประเด็นการเลือกจำนวนคลาส

ผลการทดลองพบว่า ค่าความถูกต้องโดยแบ่งกลุ่มเป็น 2 ความหมายมีค่าสูงกว่าค่าความถูกต้องโดยการใช้ความหมายทั้งหมดของคำกำกวมที่มีอยู่ ตัวอย่างดังภาพประกอบ 5.35 ค่าความถูกต้องของคำกำกวม bum เมื่อจำแนกโดยแบ่งกลุ่มเป็น 2 ความหมายมีดังนี้ ความหมายที่หนึ่ง (1:18:03) ให้ค่าความถูกต้อง 100% ความหมายที่สอง (1:18:00) ให้ค่าความถูกต้อง 97.4% และความหมายที่สาม (1:18:02) ให้ค่าความถูกต้อง 96.1% ซึ่งสูงกว่าใช้ความหมายทั้งหมดของคำกำกวมที่มีอยู่ที่ให้ค่าความถูกต้องเพียง 96.1% เท่านั้น



ภาพประกอบ 5.35 กราฟแสดงค่าความถูกต้องเปรียบเทียบการแบ่งกลุ่มแบบ 2 ความหมายและความหมายทั้งหมด กรองแบบ GainRatioAttributeEval และใช้อัลกอริทึม RBFNetwork

5.4 เปรียบเทียบผลการทดลองและวิจารณ์ผลการทดลอง

ผลการทดลองคำกำกวมที่ใช้ในการจำแนกความหมายของคำอื่นๆคือ art authority bar bum chair replace hearth local detention child church child dyke cool fit colorless faithful begin find และ keep ทดลองแบบ 10 Folds cross-validation เมื่อเปรียบเทียบผลการทดลองโดยใช้คำบริบททั้งทางซ้ายและขวา ทางซ้าย และทางขวาของคำกำกวมทั้งหมดซึ่งประกอบด้วยคำนาม (Noun) คำกริยา (Verb) และคำคุณศัพท์ (Adjective) ผลการทดลองจากตารางที่ 5.17 แสดงให้เห็นว่าเมื่อใช้คำบริบททั้งทางซ้ายและขวาให้ค่าความถูกต้องกว่าเมื่อใช้บริบททางซ้ายอย่างเดียวหรือขวาอย่างเดียวทั้งคำนาม คำกริยา และคำคุณศัพท์ และผลการทดลองเมื่อเลือกใช้คำบริบททั้งทางซ้ายและขวา ขนาดหน้าต่างเท่ากับ 4 กรองโดย GainRatioAttributeEval โดยใช้อัลกอริทึม RBFNetwork สามารถสรุปรายละเอียดของผลการทดลองได้ดังตารางที่ 5.18

จากผลการทดลองจากตารางที่ 5.17 และ 5.18 สามารถสรุปได้ว่าแบบจำลองการแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคคำบริบทสามารถนำไปใช้ได้ทั้งคำนาม คำกริยา และคำคุณศัพท์

ตารางที่ 5.17 แสดงค่าความถูกต้องของคำกำกวมทั้งหมดเมื่อใช้ขนาดหน้าต่างแบบต่างๆ

Ambiguity Word	Part-Of-Speech	Both	Left	Right
art	noun	**94.23%**	90.38%	90.86%
authority	noun	**94.41%**	90.02%	90.23%
bum	noun	**100%**	96.80%	97.40%
bar	noun	**100%**	61.31%	63.37%
hearth	noun	**96.96%**	95.31%	76.56%
stress	noun	**100%**	95.58%	94.11%
detention	noun	**96.82%**	95.23%	85.71%
dyke	noun	**96.61%**	89.83%	89.83%
church	noun	**91.66%**	90.73%	88%
child	noun	**90.67%**	86.32%	83.76%
replace	verb	**97.67%**	74.41%	87.20%
begin	verb	95.62%	**95.98%**	95.07%
find	verb	**83.09%**	71.83%	70.58%
keep	verb	**84.45%**	84.37%	82.29%
colorless	adjective	**97.01%**	97.01%	82.08%
cool	adjective	**94.62%**	94.56%	91.30%
fit	adjective	**98.21%**	94.64%	91.07%
faithful	adjective	**97.87%**	79.97%	78.72%
local	adjective	87.01%	**88.01%**	80.51%

ตารางที่ 5.18 แสดงคำกำกวมทั้งหมดโดยใช้เทคนิคคำบริบทเมื่อจำนวนแอทริบิวต์เท่ากับ 40

Ambiguity Word	Part-Of-Speech	Number of class	Number of instance	Accuracy
art: meaning 1(1:09:00::)	**noun**	**2**	**208**	**94.23%**
art: meaning 2 (1:04:00::)	noun	2	208	90.38%
art: meaning 3 (1:06:00::)	noun	2	208	88.46%
art (3 meanings)	noun	3	208	87.01%
bum: meaning 1 (1:18:03::)	**noun**	**2**	**77**	**100%**
bum: meaning 2 (1:08:00::)	noun	2	77	94.40%
bum: meaning 3 (1:18:02::)	noun	2	77	96.10%
bum (3 meanings)	noun	3	77	96.10%
authority: meaning 1(1:07:02::)	**noun**	**2**	**179**	**94.41%**
authority: meaning 2 (1:14:00::)	**noun**	2	179	91.62%
authority: meaning 3 (1:18:01::)	noun	2	179	89.94%
authority: meaning 4 (1:07:00::)	noun	2	179	89.94%
authority (4 meanings)	noun	4	179	69.83%
bar: meaning 1(1:14:00::)	**noun**	**2**	**248**	**100%**
bar: meaning 2(1:06:06::)	**noun**	**2**	**248**	**100%**
bar: meaning 3(1:06:05::)	noun	2	248	98.79%
bar: meaning 4(1:10:00::)	noun	2	248	98.38%
bar: meaning 5(1:06:00::)	noun	2	248	95.16%
bar: meaning 6(1:06:04::)	noun	2	248	80.64%
bar (6 meanings)	noun	6	248	59.27%
chair: meaning 1(1:18:00::)	**noun**	**2**	**139**	**100%**
chair: meaning 2 (1:06:00::)	**noun**	2	139	98.56%
chair: meaning 3 (1:04:00::)	noun	2	139	98.56%
chair (3 meanings)	noun	3	139	95.68%
hearth: meaning 1(1:06:00::)	**noun**	**2**	**66**	**96.96%**
hearth: meaning 2 (1:15:00::)	**noun**	**2**	**66**	**96.96%**
hearth: meaning 3 (1:06:01::)	noun	2	66	92.42%
hearth: (3 meanings)	noun	3	66	81.81%

ตารางที่ 5.18 แสดงคำกำกวมทั้งหมดโดยใช้เทคนิคคำบริบทเมื่อจำนวนแอทริบิวต์เท่ากับ 40 (ต่อ)

Ambiguity Word	Part-Of-Speech	Number of class	Number of instance	Accuracy
stress: meaning 1(1:26:03::)	**noun**	**2**	**81**	**100%**
stress: meaning 2(1:26:02::)	noun	2	81	93.82%
stress: meaning 3(1:26:01::)	noun	2	81	88.88%
stress: (3 meanings)	noun	3	81	72.83%
Detention	noun	2	63	96.82%
Dyke	noun	2	59	96.61%
Church	noun	2	133	91.66%
Child	noun	2	118	90.67%
begin: meaning 1(2:42:00)	**verb**	**2**	**548**	**95.63%**
begin: meaning 2(2:42:03)	verb	2	548	95.43%
begin: meaning 3(2:30:01)	verb	2	548	83.02%
begin: meaning 4(2:42:04)	verb	2	548	76.27%
begin: meaning 5(2:30:00)	verb	2	548	66.44%
begin: (5 meanings)	verb	5	548	62.59%
find: meaning 1(2:32:00)	**verb**	**2**	**71**	**83.09%**
find: meaning 2(2:40:00)	verb	2	71	80.28%
find: meaning 3(2:39:02)	verb	2	71	80.28%
find: meaning 4(2:31:10)	verb	2	71	76.05%
find: (4 meanings)	verb	4	71	59.15%
keep: meaning 1 (2:41:00)	**verb**	**2**	**96**	**84.45%**
keep: meaning 2 (2:41:01)	verb	2	96	76.04%
keep: meaning 3 (2:42:00)	verb	2	96	71.87%
keep: meaning 4 (2:42:07)	verb	2	96	69.62%
keep: (4 meanings)	verb	4	96	68.75%
replace: meaning 1(2:40:00::)	**verb**	**2**	**86**	**97.67%**
replace: meaning 2 (2:30:00::)	verb	2	86	91.86%
replace: meaning 3 (2:41:00::)	verb	2	86	87.20%
replace (3 meanings)	verb	3	86	69.76%
local	adjective	2	382	87.01%
colorless	adjective	2	67	97.01%
cool	adjective	2	92	94.62

ตารางที่ 5.18 แสดงคำกำกวมทั้งหมดโดยใช้เทคนิคคำบริบทเมื่อจำนวนแอทริบิวต์เท่ากับ 40 (ต่อ)

Ambiguity Word	Part-Of-Speech	Number of class	Number of instance	Accuracy
fit	adjective	2	56	98.21%
faithful: meaning 1 (3:00:00)	adjective	2	47	97.87%
faithful: meaning 2 (3:00:01)	adjective	2	47	85.10%
faithful: meaning 3 (5:00:00)	adjective	2	47	80.85%
faithful: (3 meanings)	adjective	4	47	80.85%

ตารางที่ 5.19 แสดงผลการเปรียบเทียบการแก้ปัญหาความกำกวมของคำโดยใช้คำบริบทกับวิธีการอื่นๆ พบว่าวิธีการการแก้ปัญหาความกำกวมของคำโดยใช้คำบริบท (WSD_AS) เมื่อไม่ตัดคำที่เป็น Stoplist และตัดคำที่เป็น Stoplist ให้ค่าความถูกต้องสูงกว่าวิธี Bootstrapping และ Maximum Entropy เช่นตัวอย่างคำว่า art เมื่อใช้การแก้ปัญหาความกำกวมของคำโดยใช้คำบริบทโดยไม่ตัด Stoplist ให้ค่าความถูกต้อง 85.79% และตัด Stoplist ให้ค่าความถูกต้อง 94.23% ซึ่งสูงกว่าใช้วิธี Bootstrapping แบบ Self-Training และ Co-Training ที่ให้ค่าความถูกต้องเพียง 59.61% และ 59.61% ตามลำดับ

ตาราง 5.19 ผลการทดลองเมื่อเปรียบเทียบการแก้ปัญหาความกำกวมโดยใช้คำบริบทกับวิธีอื่น ๆ โดยใช้คลังข้อความ Senseval-2

Ambiguity Word	**WSD_AS**		**Bootstrapping** (Mihalcea, 2004)		**Maximum Entropy** (Palomar and Suarez, 2002)
	Accuracy ตัด Stoplist	Accuracy ไม่ตัด Stoplist	Self-Training	Co-Training	
art	**94.23 %**	85.79 %	59.61 %	59.61 %	65.2 %
church	**96.24 %**	80 %	72.22 %	69.44 %	67.9 %
child	**90.67 %**	86.32 %	68.33 %	68.33 %	90.5 %
authority	**94.41 %**	87.5 %	58.75 %	62.50 %	-
bar	**100 %**	63.37 %	35.48 %	34.67 %	-
bum	**100 %**	93.42 %	58.13 %	46.51 %	-
chair	**100 %**	93.43 %	80.95 %	80.95 %	-
hearth	**96.96 %**	76.56 %	55.17 %	65.51 %	-
stress	**100 %**	88.23 %	52.63 %	57.89 %	-
detention	**96.82 %**	84.12 %	91.66 %	91.66 %	-
dyke	**96.61 %**	87.93 %	42.30 %	50 %	-

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้เสนอแนวคิดใหม่ในการแก้ปัญหาความกำกวมของคำโดยใช้หน้าต่างคำบริบท (Context Window) โดยสร้างแบบจำลองการแก้ปัญหาความกำกวมของคำและการเลือกแอทริบิวต์โดยใช้อัตราส่วนเกนและโครงข่ายประสาทเทียมแบบเรเดียลเบซิสฟังก์ชัน Word Sense Disambiguation and Attribute Selection (WSD_AS) Using Gain Ratio and RBF Neural Network งานวิจัยนี้ได้ใช้เลือกคำบริบทแบบทั้งทางซ้ายและขวา ใช้อัลกอริทึม RBFNetwork ในการจำแนกความหมาย ใช้เทคนิคการกรองแบบ GainRatioAttributeEval ใช้คลังข้อความมาตรฐาน Senseval-2 ในการทดสอบประสิทธิภาพของแบบจำลอง

ผู้ทำการวิจัยได้พัฒนาโปรแกรมจากแบบจำลองที่นำเสนอเพื่อแก้ปัญหาความกำกวมของคำจากคลังข้อความ Senseval-2 ที่เป็นภาอังกฤษคือ eng-lex-sample ซึ่งผู้ใช้สามารถใช้งานง่ายด้วย Graphic User Interface โปรแกรมการแก้ปัญหาความกำกวมพัฒนาโดยใช้ Visual Basic.Net ทำงานร่วมกับโปรแกรม SenseTools และ NSP ในการสร้างข้อมูลให้อยู่ในรูปแบบ arff และใช้โปรแกรม WEKA แบบ Command Line Interface ในการจำแนกความหมาย ผลลัพธ์ที่ได้ให้ค่าความถูกต้องสูงกว่าวิธีอื่นที่ใช้คลังข้อความ Senseval-2 เช่นเดียวกัน

ผลการทดลองของงานวิจัยนี้ในการตัดคำที่เป็น Stoplist เรื่อง “การแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคการตัดคำสำหรับคลังข้อความ Senseval-2” ได้รับการตีพิมพ์ใน The Third National Conference On Computing and Information Technology วันที่ 25-25 พฤษภาคม 2550 ดังภาคผนวก ข และผลการทดลองเกี่ยวกับการใช้หน้าต่างคำบริบทและการเลือกแอทริบิวต์เรื่อง “Word Sense Disambiguation and Attribute Selection Using Gain Ratio and RBF Neural Network” ได้รับการตอบรับการตีพิมพ์ใน 2008 IEEE International Conference On Research, Innovation and Vision for the Future in Computing & Communications Technologies ประเทศเวียดนาม วันที่ 13-17 กรกฎาคม 2551 ดังภาคผนวก ค

ผลลัพธ์ของการแก้ปัญหาความกำกวมโดยใช้เทคนิคคำบริบทสามารถสรุปตามแบบจำลองการแก้ปัญหาความกำกวม 4 ขั้นตอนดังนี้ 1) การเตรียมคลังข้อความ ผลการทดลอง

จากขั้นตอนนี้จะสรุปในประเด็นการตัดคำที่เป็น Stoplist ออก 2) การสร้างแอทริบิวต์ ผลการทดลองจากขั้นตอนนี้จะสรุปในประเด็นการเลือกขนาดหน้าต่างและประเด็นการเลือกประเภทของคำบริบท 3) การเลือกแอทริบิวต์ ผลการทดลองจากขั้นตอนนี้จะสรุปในประเด็นจำนวนแอทริบิวต์สำหรับการกรองแอทริบิวต์ และประเด็นการเลือกเทคนิคในการกรองแอทริบิวต์ และ 4) การจำแนกความหมาย ผลการทดลองจากขั้นตอนนี้จะสรุปในประเด็นการเลือกวิธีการจำแนกความหมายและประเด็นการเลือกอัลกอริทึม โดยมีรายละเอียดดังนี้

6.1.1 ประเด็นการตัดคำที่เป็น Stoplist การตัดคำที่เป็น Stoplist ออกจากไฟล์ทำให้การจำแนกความหมายให้ค่าความถูกต้องสูงกว่าการไม่ตัดคำที่เป็น Stoplist เนื่องจากคำ Stoplist เป็นคำฟุ่มเฟือยไม่ได้ใช้ประโยชน์ในการหาความหมาย จึงทำให้ประสิทธิภาพในการแก้ปัญหาความกำกวมลดลง

6.1.2 ประเด็นการเลือกขนาดหน้าต่าง จากผลการทดลองจะเห็นได้ว่า การใช้ขนาดหน้าต่างของคำบริบทที่มีความกว้างที่เหมาะสมมีผลต่อค่าความถูกต้องในการจำแนกความหมาย ความกว้างที่น้อยหรือมากเกินไปจะทำให้ประสิทธิภาพในการจำแนกความหมายไม่ดี จากผลการทดลองงความกว้างของขนาดหน้าต่างที่เหมาะสมคือ 4

6.1.3 ประเด็นการเลือกประเภทของคำบริบท ในการหาความหมายถ้าเราใช้คำบริบทมาช่วยในการแปลความหมายที่แตกต่างกันจะทำให้ผลการจำแนกความหมายแตกต่างกัน ประเภทของคำบริบทมี 3 แบบ คือ ใช้คำบริบททางซ้าย ใช้คำบริบททางขวา และใช้คำบริบททั้งทางซ้ายและขวา จากผลการทดลองจะเห็นได้ว่า การใช้คำบริบททั้งทางซ้ายและขวาให้ค่าความถูกต้องสูงกว่าการใช้บริบททางซ้ายอย่างเดียวหรือขวาเพียงอย่างเดียว

6.1.4 ประเด็นจำนวนแอทริบิวต์สำหรับการกรองแอทริบิวต์ การกรองแอทริบิวต์ให้มีจำนวนลดลงมีผลต่อการจำแนกข้อมูล จากผลการทดลองจะเห็นได้ว่า การกรองแอทริบิวต์ให้มีค่าลดลงจะทำให้ค่าความถูกต้องของข้อมูลสูงกว่าแบบไม่กรองแอทริบิวต์ใดๆ คือใช้ทุกแอทริบิวต์ทั้งหมดที่มี

6.1.5 ประเด็นการเลือกเทคนิคในการกรองแอทริบิวต์ เทคนิคในการกรองแอทริบิวต์ 2 เทคนิค คือ GainRatioAttributeEval และ InfoGainAttributeEval จากผลการทดลองจะเห็นได้ว่า การกรองแอทริบิวต์โดยใช้ GainRatioAttributeEval ให้ค่าความถูกต้องสูงกว่า InfoGainAttributeEval

6.1.6 ประเด็นการเลือกวิธีการจำแนกความหมาย วิธีการในการจำแนกความหมาย 2 แบบคือ การจำแนกความหมายโดยแบ่งกลุ่มเป็น 2 ความหมายและการใช้ความหมายทั้งหมดที่มีอยู่กรณีที่มีความหมาย 2 ความหมายขึ้นไป จากผลการทดลองจะเห็นได้ว่า การจำแนกความหมายโดยแบ่งกลุ่มเป็น 2 ความหมายให้ค่าความถูกต้องสูงกว่าการใช้ความหมายทั้งหมด

6.1.7 ประเด็นการเลือกอัลกอริทึม อัลกอริทึมในการจำแนกข้อมูลมี 2 อัลกอริทึมคือ RBFNetwork และ ID3 จากผลการทดลองจะเห็นได้ว่า อัลกอริทึมในการจำแนกข้อมูลแบบ RBFNetwork ให้ค่าความถูกต้องสูงกว่าอัลกอริทึม ID3

6.2 ปัญหาและอุปสรรค

6.2.1 เนื่องจากในการทำงานกับข้อมูลที่มีแอทริบิวต์จำนวนมากด้วยโปรแกรม WEKA ต้องใช้หน่วยความจำสูง 1 กิกะไบท์ จึงจะสามารถทำงานได้อย่างมีประสิทธิภาพ หากหน่วยความจำน้อยเกินไปจะทำให้ทำงานช้าหรือไม่สามารถทำงานได้

6.2.2 เนื่องจากคลังข้อความที่นำมาทดสอบอยู่ในรูปแบบที่ไม่สามารถนำมาใช้งานได้ทันทีต้องใช้เวลาในการแปลงให้อยู่ในรูปแบบที่พร้อมใช้งานก่อน ซึ่งต่างจากข้อมูลในงานด้านอื่นๆที่สามารถนำมาใช้งานได้ทันที

6.3 ข้อเสนอแนะ

6.3.1 การนำแบบจำลองการแก้ปัญหาความกำกวมโดยใช้เทคนิคคำบริบทนี้ไปใช้กับภาษาอื่นเช่น ภาษาจีน ภาษาญี่ปุ่น อาจได้ผลลัพธ์ที่แตกต่างกันขึ้นอยู่กับลักษณะของการวางตำแหน่งของคำบริบทในภาษานั้นๆ เช่น ในภาษาอังกฤษคำขยายคำกำกวมส่วนใหญ่จะอยู่ด้านซ้าย ส่วนคำภาษาอื่นอาจจะมีโครงสร้างที่ต่างกัน

6.3.2 การนำแบบจำลองการแก้ปัญหาความกำกวมโดยใช้เทคนิคคำบริบทนี้ไปใช้กับภาษาไทย มีส่วนที่เปลี่ยนแปลงไปจากเดิมคือในส่วนการตัดคำที่เป็น Stoplist หากไม่มี Stoplist ที่เป็นภาษาไทยที่ถูกจัดเก็บรวบรวมไว้เป็นมาตรฐาน สามารถแปลความหมายจาก Stoplist ที่เป็นภาษาอังกฤษได้

บรรณานุกรม

กรุง สินอภิรมย์สราญ. 2551. Data Mining, Data Warehouse and Visualization. [ออนไลน์] เข้าถึงได้จาก http://pioneer.netserv.chula.ac.th/~skrung/csc662/ (วันที่สืบค้น17 มีนาคม 2551)

วิโรจน์ อรุณมานะกุล. 2550. ภาษาศาสตร์คลังข้อมูล. [ออนไลน์] เข้าถึงได้จาก http://pioneer.chula.ac.th/~awirote/2209673/index.html (วันที่สืบค้น16 พฤศจิกายน 2550)

ศุภชัย ตั้งบุญญะศิริ และ กฤติกา วงศาวณิช. 2551. [ออนไลน์] เข้าถึงได้จาก http://202.28.94.55/web/320417/2548/work1/g26/Files/Report_Neural%20Network.doc(วันที่สืบค้น17 มีนาคม 2551)

Aroonmanakun, A. 1999. Concordance. http://www.arts.chula.ac.th/~ling/ThaiConc/ (accessed 17/3/08).

Anh, C. L., Huynh, V., and Shimazu, A. 2005. An evidential reasoning approach to weighted combination of classifiers for word sense disambiguation. MLDM. pp 516-525.

Borges, H.B., and Nievola, J.C. 2005. Attribute selection methods comparison for Classification of diffuse large B-Cell lymphoma. Proceedings of the Fourth International Con-ference on Machine Learning and Applications, pp 201-206.

Carpuat, M., and Wu, D. 2005. Word sense disambiguation vs. statistic machine translation. Proceedings of the 43rd Annual Meeting of the ACL, pp 387-394.

Casado, M.R., Alfonseca, E., and Castells, P. 2005. Using context-window overlapping in synonym discovery and ontology extension. International Conference on Recent Advances in Natural Language Processing.

Chao, G., and Dyer, M.G. 2002. Maximum entropy models for word sense disambiguation. Proceedings of the 19th international conference on Computational linguistics, pp 1-7.

Ciaramita, M., Johnson, M., and Hofmann, T. 2003. Hierarchical semantic classification: Word sense disambiguation with world knowledge. Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI) (2003).

Dong, M., and Kothari, R. 2003. Feature subset selection using a new definition of classifiability q. Pattern Recognition Letters, pp 1215-1225.

Flores, M.J., Gamez, J.A., and Mateo, J.L. 2008. Mining the ESROM: A study of breeding value classification in Manchego sheep by means of attribute selection and construction, pp 167-177.

Frakes, W. B., and Yates, R.B. 1992. Information retrieval data structure & algorithm New Jersey: Prentice Hall.

Ganchev, T., Zervas, P., Fakotakis, N., and Kokkinakis, G. 2006. Benchmarking feature selection techniques on the speaker verification task. 5th Interna-tional. Symposium on Communication systems, networks and digital signal processing, pp 314-318.

George, A. Miller. 1998. WordNet: a lexical database for the English language. http://wordnet.princeton.edu/license (accessed 17/03/08).

Hall, M.A., and Holmes, G. 2003. Benchmarking attribute selection techniques for discrete class data mining. IEEE Transaction on knowledge and data engineering, pp 1437-1447.

Haykin, S. 2008. Feedforward neural networks: Introduction. http://media.wiley.com /product_data/excerpt/19/04713491/0471349119.pdf (accessed 17/03/08).

Huang, Y., McCullagh, PJ., and Black, ND. 2004. Feature selection via supervised model construction. Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), pp 411-414.

Ian, W.H., and Frank, E. 2005a. WEKA (Waikato environment for knowledge analysis). http://www.cs.waikato.ac.nz/ml/weka/ (accessed 17/03/08).

Ian, W.H., and Frank, E. 2005b. Data mining: Practical machine learning tools and technique, 2nd Edition Morgan Kaufman: San Francisco.

Koh, C.H., and Wong, L. 2007. Recognition of polyadenylation sites from Arabidopsis genomic sequences. Proceedings of 18th International Conference on Genome Informatics (GIW), pp 73—82.

Kouskoumvekaki, I., Yang, Z., and Jonsdottir, S. O. 2008. Identification of biomarkers for genotyping Aspergilli using non-linear methods for clustering and classification. BMC Bioinformatics 2008.

Legrand, S., and Pulido, J.R.G., P. 2004. A Hybrid approach to word sense disambiguation: Neural clustering with class labeling. 8th European Conference on Principles and Prac-tice of Knowledge Discovery in Databases (PKDD).

Liu, T., Lu, Z., and Li, S. 2005. Chinese word sense disambiguation based on neural networks. Journal of Harbin Institute of Technology, pp 408-417.

Mihalcea, R. 2004. Co-training and self-training for word sense disambiguation. 8th Conference on Computational Natural Language Learning (CoNLL-2004), pp 238-241.

Nectec. 1997. Orchid. http://www.links.nectec.or.th/orchid/ (accessed 17/03/08).

Nikolaev, N. 2008. Radial-Basis Function Networks. http://homepages.gold.ac.uk/nikolaev/311rbf.htm (accessed 17/03/08).

Oh, J., and Choi, K. 2002. Word sense disambiguation using static and dynamic sense vectors. 19th International Conference on Computational Linguistics, pp1-7.

O'Hara, T., Bruce, R., Donner, J., and Wiebe, J. 2004. Class-based collocations for word-sense disambiguation. Proceeding. Senseval 3 Workshop on Evaluation of Systems for the Semantic Analysis of Text.

Okada, T. 2005. Attribute selection in chemical graph mining using correlations among Linear fragments. International Conference on Recent Advances in Natural Language Processing (RANLP 2005).

Pedersen, T. 2001. SENSEVAL-2: Second International workshop on evaluating word sense disambiguation systems. http://www.senseval.org. (accessed 17/03/08).

Pedersen, T. and Purandare, A. 2002. OMtoSVAL2 Package. http://www.d.umn.edu/~tpederse/Code/Readme.OMtoSVAL2-v0.1.txt. (17 March 2008).

Pedersen, T. 2003. SenseTools. http://www.d.umn.edu/~tpederse/sensetools.html. (accessed 17/03/08).

Pedersen, T. 2006. NSP (Ngram statistics package). http://search.cpan.org/dist/Text-NSP/. (accessed 17/03/08).

Pham, T.P., Ng., H.W., and Lee, W.S. 2005. Word sense disambiguation with semi-supervised learning. Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005), pp 1093-1098.

Riloff, E. 2006. Natural language processing: semantics, the basics. http://www.cs.utah.edu/classes/cs5340/slides/semantics-basics.pdf (accessed 17/03/08).

Sae-Tang, S., and Mathaste, I. 2002. Thai online handwritten character recognition using windowing backpropagation neural networks. In Proceeding of the IASTED International Conference Applied Informatics 2002.

Stokoe, C., Oakes, M. P., and Tait, J. 2003. Word sense disambiguation in Infor-mation retrieval revisited. Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in informaion retrieval, pp 159-166.

Suarez, A., and palomar, M. 2002. Best feature selection for maximum entropy-based word sense disambiguation, Proceedings of the 6th International Conference on Applica-tions of Natural Language to Information Systems, pp 213-217.

Symeonidis, A.L., Nikolaidou, V., and Mitkas, P.A. 2007. Exploiting data mining techniques for Improving the efficiency of a supply chain management agent. Proceedings of the 2006 IEEE/WIC/ACM International conference on Web Intelligence and Intelligent Agent Technology, pp 23-26.

Vickrey, D., Biewald, L., Teyssier, M., Koller, D. 2005. Word-sense disambiguation for machine translation. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp 771-778.

Wettayaprasit, W., and Nanakorn, P. 2006. Feature extraction and Interval filtering Technique for time-series forecasting using neural networks. Proceeding 2006 IEEE International Conferences on Cybernetics and Intelligent Systems (CIS).

Wettayaprasit, W., Laosen, N., and Chevakidagarn, S. 2007. Data filtering technique for neural networks forecasting. Proceeding the 3rd WSEAS International Symposium on Data Mining and Intelligent Information Processing (DATAMIN'07).

Yoon, Y., Seon, C.N., Lee, S., and Seo, J. 2006. Unsupervised word sense dis-ambiguation for Korean through the acyclic weighted digraph using corpus and dictionary. Information Processing and Management: an International Journal, pp 836-847.

ภาคผนวก

ภาคผนวก ก

การใช้งาน Command Line Interface ใน WEKA-3-4

โปรแกรม WEKA (Waikato Environment for Knowledge Analysis) เป็นโปรแกรมที่ใช้ในงานด้านการเรียนรู้ของเครื่อง นิยมนำมาใช้ในในการวิเคราะห์ข้อมูลในการทำเหมืองข้อมูลเช่นการจำแนก (Classification) และการจัดกลุ่ม (Clustering) ข้อมูล ส่วนประกอบของWEKA มีทั้งแบบ User Interface และ Command Line Interface ในงานวิจัยนี้ได้ใช้แบบ Command Line User Interface ซึ่งสามารถเรียกใช้จากโปรแกรมที่พัฒนาได้

ก.1 โครงสร้างของ WEKA

โครงสร้างของ WEKA ประกอบด้วยแพคเกจในการทำงานหลายแพคเกจดังตัวอย่างในภาพประกอบ ก.1 – ก.7

Package weka.filters

Interface Summary	
StreamableFilter	Interface for filters can work with a stream of instances.
SupervisedFilter	Interface for filters that make use of a class attribute.
UnsupervisedFilter	Interface for filters that do not need a class attribute.

Class Summary	
AllFilter	A simple instance filter that passes all instances directly through.
Filter	An abstract class for instance filters: objects that take instances as input, carry out some transformation on the instance and then output the instance.
NullFilter	A simple instance filter that allows no instances to pass through.

ภาพประกอบ ก.1 แพคเกจ Filters

Package weka.associations

Class Summary	
Apriori	Class implementing an Apriori-type algorithm.
AprioriItemSet	Class for storing a set of items.
Associator	Abstract scheme for learning associations.
ItemSet	Class for storing a set of items.
LabeledItemSet	Class for storing a set of items together with a class label.
PredictiveApriori	Class implementing the predictive apriori algorithm to mine association rules.
PriorEstimation	Class implementing the prior estimattion of the predictive apriori algorithm for mining association rules.
RuleGeneration	Class implementing the rule generation procedure of the predictive apriori algorithm.
RuleItem	Class for storing an (class) association rule.
Tertius	Class implementing a Tertius-type algorithm.

ภาพประกอบ ก.2 แพคเกจ Associations

Package weka.classifiers

Interface Summary

IterativeClassifier	Interface for classifiers that can induce models of growing complexity one step at a time.
Sourcable	Interface for classifiers that can be converted to Java source.
UpdateableClassifier	Interface to incremental classification models that can learn using one instance at a time.

Class Summary

BVDecompose	Class for performing a Bias-Variance decomposition on any classifier using the method specified in:
BVDecomposeSegCVSub	This class performs Bias-Variance decomposion on any classifier using the sub-sampled cross-validation procedure as specified in:
CheckClassifier	Class for examining the capabilities and finding problems with classifiers.
CheckClassifier.PostProcessor	a class for postprocessing the test-data
Classifier	Abstract classifier.
CostMatrix	Class for storing and manipulating a misclassification cost matrix.
Evaluation	Class for evaluating machine learning models.
IteratedSingleClassifierEnhancer	Abstract utility class for handling settings common to meta classifiers that build an ensemble from a single base learner.
MultipleClassifiersCombiner	Abstract utility class for handling settings common to meta classifiers that build an ensemble from multiple classifiers.
RandomizableClassifier	Abstract utility class for handling settings common to randomizable classifiers.
RandomizableIteratedSingleClassifierEnhancer	Abstract utility class for handling settings common to randomizable meta classifiers that build an ensemble from a single base learner.
RandomizableMultipleClassifiersCombiner	Abstract utility class for handling settings common to randomizable meta classifiers that build an ensemble from multiple classifiers based on a given random number seed.
RandomizableSingleClassifierEnhancer	Abstract utility class for handling settings common to randomizable meta classifiers that build an ensemble from a single base learner.
SingleClassifierEnhancer	Abstract utility class for handling settings common to meta classifiers that use a single base learner.

ภาพประกอบ ก.3 แพคเกจ Classifiers

Package weka.attributeSelection

Interface Summary

AttributeTransformer	Abstract attribute transformer.
ErrorBasedMeritEvaluator	Interface for evaluators that calculate the "merit" of attributes/subsets as the error of a learning scheme
RankedOutputSearch	Interface for search methods capable of producing a ranked list of attributes.
StartSetHandler	Interface for search methods capable of doing something sensible given a starting set of attributes.

Class Summary

ASEvaluation	Abstract attribute selection evaluation class
ASSearch	Abstract attribute selection search class.
AttributeEvaluator	Abstract attribute evaluator.
AttributeSelection	Attribute selection class.
BestFirst	Class for performing a best first search.
CfsSubsetEval	CFS attribute subset evaluator.
ChiSquaredAttributeEval	Class for Evaluating attributes individually by measuring the chi-squared statistic with respect to the class.
ClassifierSubsetEval	Classifier subset evaluator.
ConsistencySubsetEval	Consistency attribute subset evaluator.
ExhaustiveSearch	Class for performing an exhaustive search.
GainRatioAttributeEval	Class for Evaluating attributes individually by measuring gain ratio with respect to the class.
GeneticSearch	Class for performing a genetic based search.
GreedyStepwise	Class for performing a hill climbing search (either forwards or backwards).
HoldOutSubsetEvaluator	Abstract attribute subset evaluator capable of evaluating subsets with respect to a data set that is distinct from that used to initialize/ train the subset evaluator.
InfoGainAttributeEval	Class for Evaluating attributes individually by measuring information gain with respect to the class.
OneRAttributeEval	Class for Evaluating attributes individually by using the OneR classifier.
PrincipalComponents	Class for performing principal components analysis/transformation.
RaceSearch	Class for performing a racing search.
RandomSearch	Class for performing a random search.
Ranker	Class for ranking the attributes evaluated by a AttributeEvaluator Valid options are:
RankSearch	Class for evaluating a attribute ranking (given by a specified evaluator) using a specified subset evaluator.
ReliefFAttributeEval	Class for Evaluating attributes individually using ReliefF.
SubsetEvaluator	Abstract attribute subset evaluator.
SVMAttributeEval	Class for Evaluating attributes individually by using the SVM classifier.
SymmetricalUncertAttributeEval	Class for Evaluating attributes individually by measuring symmetrical uncertainty with respect to the class.
UnsupervisedAttributeEvaluator	Abstract unsupervised attribute evaluator.
UnsupervisedSubsetEvaluator	Abstract unsupervised attribute subset evaluator.
WrapperSubsetEval	Wrapper attribute subset evaluator.

ภาพประกอบ ก.4 แพคเกจ AttributeSelection

Package weka.experiment

Interface Summary

Compute	Interface to something that can accept remote connections and execute a task.
RemoteExperimentListener	Interface for classes that want to listen for updates on RemoteExperiment progress
ResultListener	Interface for objects able to listen for results obtained by a ResultProducer
ResultProducer	This interface defines the methods required for an object that produces results for different randomizations of a dataset.
SplitEvaluator	Interface to objects able to generate a fixed set of results for a particular split of a dataset.
Task	Interface to something that can be remotely executed as a task.

Class Summary

AveragingResultProducer	AveragingResultProducer takes the results from a ResultProducer and submits the average to the result listener.
ClassifierSplitEvaluator	A SplitEvaluator that produces results for a classification scheme on a nominal class attribute.
CostSensitiveClassifierSplitEvaluator	A SplitEvaluator that produces results for a classification scheme on a nominal class attribute, including weighted misclassification costs.
CrossValidationResultProducer	Generates for each run, carries out an n-fold cross-validation, using the set SplitEvaluator to generate some results.
CSVResultListener	CSVResultListener outputs the received results in csv format to a Writer
DatabaseResultListener	DatabaseResultListener takes the results from a ResultProducer and submits them to a central database.
DatabaseResultProducer	DatabaseResultProducer examines a database and extracts out the results produced by the specified ResultProducer and submits them to the specified ResultListener.
DatabaseUtils	DatabaseUtils provides utility functions for accessing the experiment database.
Experiment	Holds all the necessary configuration information for a standard type experiment.
InstanceQuery	Convert the results of a database query into instances.
InstancesResultListener	InstancesResultListener outputs the received results in arff format to a Writer.
LearningRateResultProducer	LearningRateResultProducer takes the results from a ResultProducer and submits the average to the result listener.
OutputZipper	OutputZipper writes output to either gzipped files or to a multi entry zip file.
PairedCorrectedTTester	Behaves the same as PairedTTester, only it uses the corrected resampled t-test statistic.
PairedStats	A class for storing stats on a paired comparison (t-test and correlation)
PairedStatsCorrected	A class for storing stats on a paired comparison.
PairedTTester	Calculates T-Test statistics on data stored in a set of instances.
PropertyNode	Stores information on a property of an object: the class of the object with the property; the property descriptor, and the current value.
RandomSplitResultProducer	Generates a single train/test split and calls the appropriate SplitEvaluator to generate some results.
RegressionSplitEvaluator	A SplitEvaluator that produces results for a classification scheme on a numeric class attribute.
RemoteEngine	A general purpose server for executing Task objects sent via RMI.
RemoteExperiment	Holds all the necessary configuration information for a distributed experiment.
RemoteExperimentEvent	Class encapsulating information on progress of a remote experiment
RemoteExperimentSubTask	Class to encapsulate an experiment as a task that can be executed on a remote host.
Stats	A class to store simple statistics
TaskStatusInfo	A class holding information for tasks being executed on RemoteEngines.

ภาพประกอบ ก.5 แพคเกจ Experiment

Package weka.gui

Interface Summary

CustomPanelSupplier	An interface for objects that are capable of supplying their own custom GUI components.
Logger	Interface for objects that display log (permanent historical) and status (transient) messages.
TaskLogger	Interface for objects that display log and display information on running tasks.

Class Summary

AttributeListPanel	Creates a panel that displays the attributes contained in a set of instances, letting the user select a single attribute for inspection.
AttributeSelectionPanel	Creates a panel that displays the attributes contained in a set of instances, letting the user toggle whether each attribute is selected or not (eg: so that unselected attributes can be removed before classification).
AttributeSummaryPanel	This panel displays summary statistics about an attribute: name, type number/% of missing/unique values, number of distinct values.
AttributeVisualizationPanel	Creates a panel that shows a visualization of an attribute in a dataset.
ComponentHelper	A helper class for some common tasks with Dialogs, Icons, etc.
CostMatrixEditor	Class for editing CostMatrix objects.
DatabaseConnectionDialog	A dialog to enter URL, username and password for a database connection.
ExtensionFileFilter	Provides a file filter for FileChoosers that accepts or rejects files based on their extension.
FileEditor	A PropertyEditor for File objects that lets the user select a file.
GenericArrayEditor	A PropertyEditor for arrays of objects that themselves have property editors.
GenericObjectEditor	A PropertyEditor for objects.
GenericPropertiesCreator	This class can generate the properties object that is normally loaded from the GenericObjectEditor.props file (= PROPERTY_FILE).
GUIChooser	The main class for the Weka GUIChooser.
HierarchyPropertyParser	This class implements a parser to read properties that have a hierarchy(i.e.
InstancesSummaryPanel	This panel just displays relation name, number of instances, and number of attributes.
JTableHelper	A helper class for JTable, e.g.
ListSelectorDialog	A dialog to present the user with a list of items, that the user can make a selection from, or cancel the selection.
Loader	This class is for loading resources from a JAR archive.
LogPanel	This panel allows log and status messages to be posted.
LookAndFeel	A little helper class for setting the Look and Feel of the user interface.
PropertyDialog	Support for PropertyEditors with custom editors: puts the editor into a separate frame.
PropertyPanel	Support for drawing a property value in a component.
PropertySelectorDialog	Allows the user to select any (supported) property of an object, including properties that any of it's property values may have.
PropertySheetPanel	Displays a property sheet where (supported) properties of the target object may be edited.
ResultHistoryPanel	A component that accepts named stringbuffers and displays the name in a list box.
ResultHistoryPanel.RKeyAdapter	Extension of KeyAdapter that implements Serializable.
ResultHistoryPanel.RMouseAdapter	Extension of MouseAdapter that implements Serializable.
SaveBuffer	This class handles the saving of StringBuffers to files.
SelectedTagEditor	A PropertyEditor that uses tags, where the tags are obtained from a weka.core.SelectedTag object.
SetInstancesPanel	A panel that displays an instance summary for a set of instances and lets the user open a set of instances from either a file or URL.
SimpleCLI	Creates a very simple command line for invoking the main method of classes.
SimpleDateFormatEditor	Class for editing SimpleDateFormat strings.
SplashWindow	A Splash window.
SysErrLog	This Logger just sends messages to System.err.
TableMap	In a chain of data manipulators some behaviour is common.
TableSorter	A sorter for TableModels.
ViewerDialog	A downsized version of the ArffViewer, displaying only one Instances-Object.
WekaTaskMonitor	This panel records the number of weka tasks running and displays a simple bird animation while their are active tasks

ภาพประกอบ ก.6 แพคเกจ Gui

Package weka.clusterers

Interface Summary

NumberOfClustersRequestable	Interface to a clusterer that can generate a requested number of clusters

Class Summary

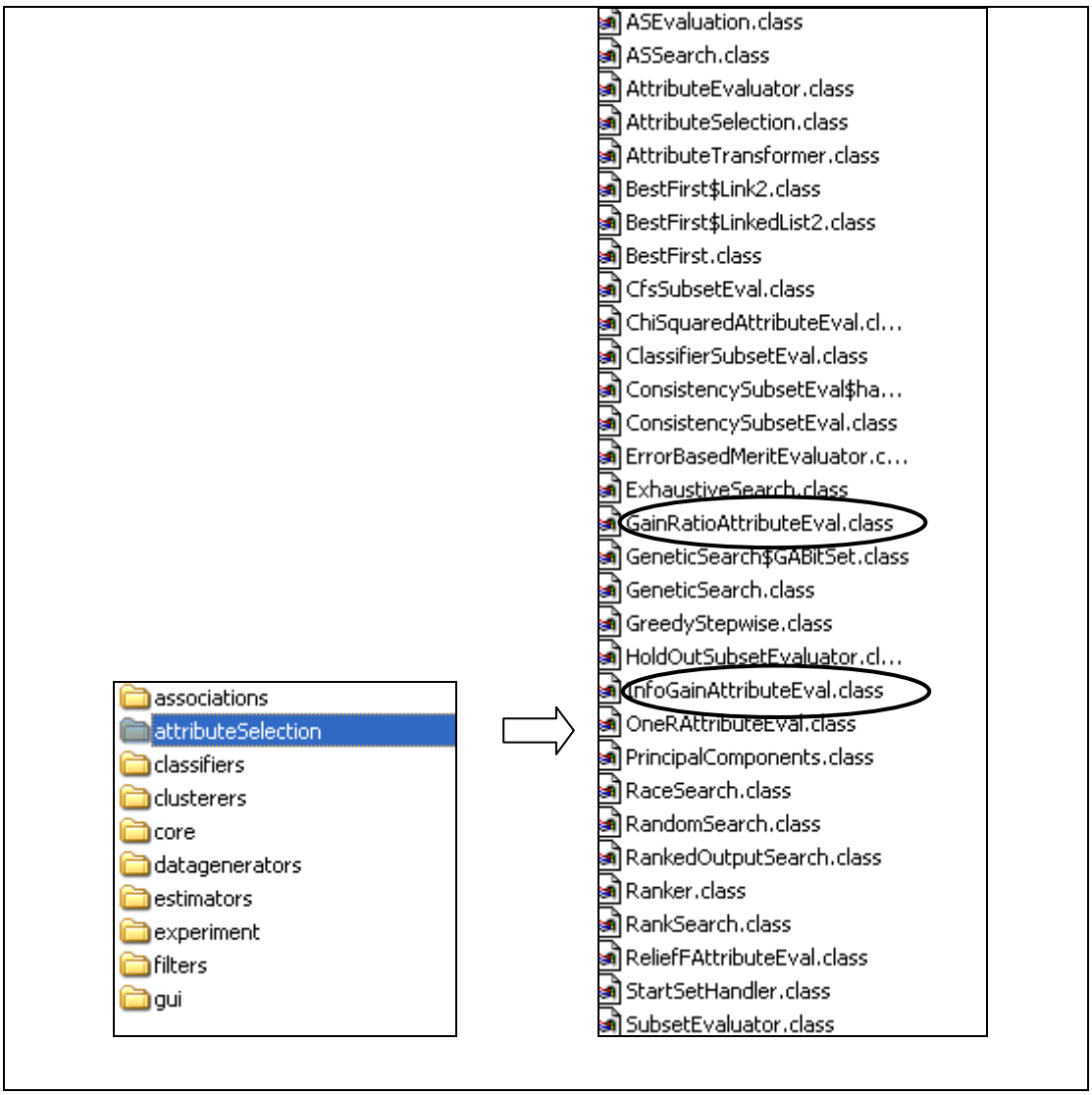
Clusterer	Abstract clusterer.
ClusterEvaluation	Class for evaluating clustering models.
Cobweb	Class implementing the Cobweb and Classit clustering algorithms.
DensityBasedClusterer	Abstract clustering model that produces (for each test instance) an estimate of the membership in each cluster (ie.
EM	Simple EM (expectation maximisation) class.
FarthestFirst	Implements the "Farthest First Traversal Algorithm" by Hochbaum and Shmoys 1985: A best possible heuristic for the k-center problem, Mathematics of Operations Research, 10(2):180-184, as cited by Sanjoy Dasgupta "performance guarantees for hierarchical clustering", colt 2002, sydney works as a fast simple approximate clusterer modelled after SimpleKMeans, might be a useful initializer for it Valid options are:
MakeDensityBasedClusterer	Class for wrapping a Clusterer to make it return a distribution and density.
SimpleKMeans	Simple k means clustering class.

ภาพประกอบ ก.7 แพคเกจ Clusterers

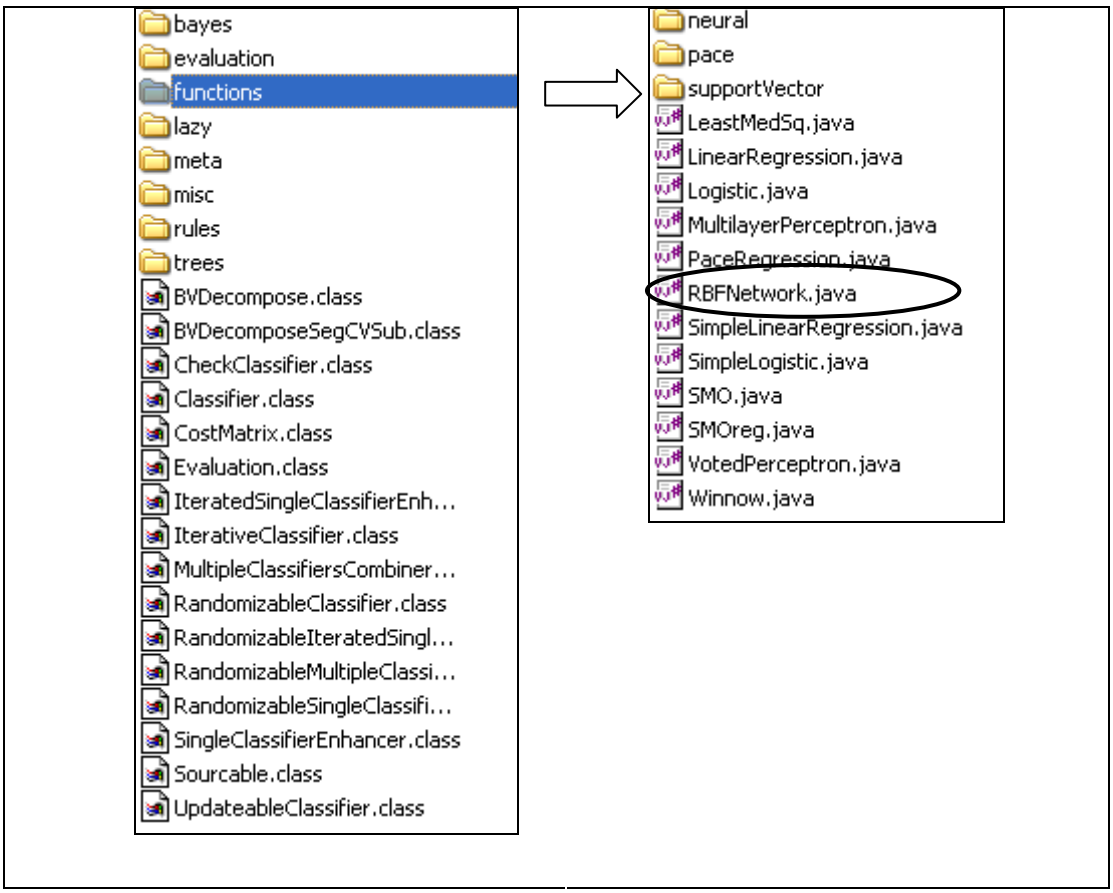
ในงานวิจัยนี้ได้ใช้เลือกใช้แพคเกจ Attribute Selection และ Classifiers ในการทำงานและได้แสดงตัวอย่างการเรียกใช้ Attribute Selection และ Classifiers คือ Attribute Selection ได้เลือกใช้ GainRatioAttributeEval และ InfoGainAttributeEval ส่วน Classifiers ได้เลือกใช้ RBFNetwork และ ID3 โดยมีตัวอย่างการใช้งานดังนี้ ดังนี้

1) Attribute Selection เป็นการกรองแอททริบิวที่ไม่จำเป็นออกโดยคลาสต่างๆที่สามารถเรียกใช้ได้ใน AttributeSelection มีหลายคลาส เช่น GainRatioAttributeEval InfoGainAttributeEval ChiSquareAttributeEval และ CfsSubsetEval เป็นต้นซึ่งแต่ละอัลกอริทึมจะมีวิธีการคำนวณค่าความสำคัญของแอททริบิวที่แตกต่างกัน แสดงดังภาพประกอบ ก.8

2) Classifiers เป็นการจำแนกประเภทข้อมูลโดยคลาสต่างๆที่สามารถเรียกใช้ได้ใน Classifiers มีหลายคลาสเช่น คลาสที่เกี่ยวกับต้นไม้ตัดสินใจ (Tree) กฎ (Rules) และฟังก์ชัน (Function)เป็นต้น ซึ่งแต่ละประเภทจะมีคลาสย่อยอยู่หลายวิธีเช่นในฟังก์ชัน จะประกอบด้วย RBFNetwork SMO MultilayerPerceptron เป็นต้น แสดงดังภาพประกอบ ก.9



ภาพประกอบ ก.8 Class ต่างๆใน Attribute Selection



ภาพประกอบ ก.9 Class ต่างๆใน Classifiers

ก.2 การเขียนคำสั่งใน WEKA

การใช้งานแบบคำสั่ง (Command Line) ใน WEKA มีวิธีการใช้แตกต่างจากการทำงานจาก User Interface โดยจะประกอบด้วยพารามิเตอร์ที่ใช้ในการทำงามากมาย เช่น

1) Attribute Selection จะมีพารามิเตอร์ที่จำเป็นในการใช้งานดังนี้

-S คือ เรียกใช้ class การจัดลำดับของแอทริบิวแบบ Ranker
จาก weka.attributeSelection.Ranker

-N คือ จำนวนแอทริบิวที่ต้องการกรอง

-E คือ เรียกใช้ class การกรองแบบ GainRatioAttributeEval
จาก weka.attributeSelection.GainRatioAttributeEval

- i คือ ไฟล์ Input

- o คือ ไฟล์ Output

ตัวอย่างการใช้คำสั่งสำหรับการกรองแอททริบิวมีดังนี้

```
java weka.filters.supervised.attribute.AttributeSelection -S "weka.attributeSelection.Ranker -N 39" -
E "weka.attributeSelection.GainRatioAttributeEval" -i art.arff -o artATTR40.arff
```

ในที่นี้จะกรองโดยใช้เทคนิค GainRatioAttributeEval ให้มีจำนวนแอททริบิวที่ต้องการ 40 แอททริบิว (พารามิเตอร์จำนวนแอททริบิวต้องใส่ให้น้อยกว่าแอททริบิวที่ต้องการจริงอยู่ 1) โดยไฟล์ Input คือ art.arff และไฟล์ Output คือ artATTR40.arff

2) Classifiers จะมีพารามิเตอร์ที่จำเป็นในการใช้งานดังนี้

-x คือ จำนวน Fold ในการแบ่งข้อมูลแบบ Cross Validation

-t คือ ไฟล์ Input

ตัวอย่างการใช้คำสั่งสำหรับการจำแนกประเภทมีดังนี้

```
Java weka.classifiers.functions.RBFNetwork -x 10 -t art_w4.arff >> art_w4_RBF.txt
```

ในที่นี้จะจำแนกประเภทของข้อมูลโดยใช้การแบ่งข้อมูลเป็น 10-Fold Cross Validation และไฟล์ที่เป็น Input คือ art_w4.arff

ภาคผนวก ข

ผลงานตีพิมพ์

เรื่อง การแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคการตัดคำสำหรับคลังข้อความ Senseval-2

Conference The Third National Conference On Computing and Information Technology (NCCIT' 07)

สถานที่ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ประเทศไทย

วันที่ 25-25 พฤษภาคม 2550

การแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคการตัดคำสำหรับคลังข้อความ Senseval-2
Word Sense Disambiguation Using Stoplist Removing for Senseval-2 Corpus

กาญจนา ทองกลิ่น ศิริรัตน์ วณิชโยบล และ วิภาดา เวทย์ประสิทธิ์
ห้องวิจัยปัญญาประดิษฐ์, ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ อ.หาดใหญ่ จ.สงขลา
E-mail: kanjana_Thongklin@hotmail.com, sirirut.v@psu.ac.th , wwettayaprasit@yahoo.com

บทคัดย่อ

การแก้ปัญหาความกำกวมของคำเป็นหนึ่งในงานด้านการประมวลผลภาษาธรรมชาติในการแปลภาษาและการสืบค้นเอกสาร บทความนี้ได้เสนอขั้นตอนการแก้ปัญหาความกำกวมของคำและเปรียบเทียบประสิทธิภาพขั้นตอนวิธีการเรียนรู้ 3 แบบคือ การจำแนกแบบ Nearest Neighbor (IBk) ต้นไม้การตัดสินใจ(ID3) และขั้นตอนวิธีเบย์ (NaiveBayes) ผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธี IBk และการตัดคำให้ค่าความถูกต้องสูงสุด

คำสำคัญ: การแก้ปัญหาความกำกวม, คลังข้อความ

Abstract

Word Sense Disambiguation is one of the works in natural language processing for language interpretation and information retrieval. This paper presents the steps of the solution of word ambiguity and comparing the efficiency of three learning algorithms that are Nearest Neighbor (IBk) Classification, Decision Tree (ID3), and Bayes Method (NaiveBayes). The experimental result indicated that Nearest Neighbor (IBk) Classification with stoplist removing gives maximum accuracy.

Keyword: Word Sense Disambiguation, Corpus

1. บทนำ

ภาษาธรรมชาติที่ใช้ในการสื่อสารมีคำบางคำที่มีความหมายหลายความหมายในบริบทที่แตกต่างกัน คำที่มีหลายความหมายนั้นเรียกว่าเกิดความกำกวม (Ambiguity) ความกำกวมของคำเป็นสิ่งที่ทำให้เกิดความผิดพลาดได้ในงานประยุกต์ด้านเทคโนโลยีของภาษา เช่น การแปลภาษา (Machine Translation) [1, 2] และการสืบค้นเอกสาร (Information Retrieval) [3]

การแก้ปัญหาความกำกวมของคำ (Word Sense Disambiguation) เป็นกระบวนการในการแทนความหมายที่ถูกต้องของคำในบริบท การแก้ปัญหาความกำกวมที่ได้นำเทคนิคการทำเหมืองข้อมูลมาใช้มี 2 แบบคือ การแก้ปัญหาความกำกวมโดยใช้ตัวอย่างสอน (Supervise Word Sense Disambiguation) [4] และการแก้ปัญหาความกำกวมโดยไม่ใช้ตัวอย่างสอน (Unsupervise Word Sense Disambiguation) [5, 6] เป็นต้น

สำหรับบทความนี้เสนอขั้นตอนวิธีการแก้ปัญหาความกำกวมของคำและเปรียบเทียบประสิทธิภาพขั้นตอนวิธีระหว่างการจำแนกแบบ Nearest Neighbor โดยใช้ขั้นตอนวิธี IBk, ต้นไม้การตัดสินใจโดยใช้ขั้นตอนวิธี ID3 และขั้นตอนวิธีเบย์โดยใช้ NaïveBayes ในส่วนที่ 2 ของบทความนี้ได้กล่าวถึงโปรแกรม คลังข้อความ Senseval-2 และเทคนิคการจำแนกโดยการทำเหมืองข้อมูล ส่วนที่ 3 นำเสนอขั้นตอนวิธีการแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคการตัดคำ ผลการทดลองอยู่ในส่วนที่ 4 และ ส่วนที่ 5 คือ บทสรุป

2. คลังข้อความ โปรแกรม และเทคนิคการจำแนก

2.1 คลังข้อความ Senseval-2

คลังข้อความ Senseval-2 [8] เป็นคลังข้อความมาตรฐานที่ใช้ในการทดสอบประสิทธิภาพของโปรแกรมในการแก้ปัญหาความกำกวมของคำ คลังข้อความ Senseval-2 มีลักษณะเป็นรูปแบบของ XML เริ่มด้วย <corpus></corpus> มีแอทริบิว lang เป็นส่วนที่บอกภาษาของข้อความดังภาพที่ 1 ซึ่งเป็นตัวอย่างของคลังข้อความ Senseval ภาษาอังกฤษมีแอทริบิว lang='english' แต่ละวรรคตอนจะประกอบด้วย

<instance></instance> <answer></answer> และ <context></context> ซึ่งใน <context></context> ประกอบด้วยประโยคและคำที่มีความหมายกำกวมจะอยู่ระหว่าง <head></head> ส่วน <answer></answer> จะมีแอทริบิวสำคัญคือ senseid บอกความหมายของคำในบริบทนั้น เช่น senseid = "art_gallery%1:06:00::"และsenseid = "art% 1:04:00:: " หมายถึง คำศัพท์คำว่า “art” ทั้งสองนี้มีความหมายต่างกัน

2.2 โปรแกรมที่ใช้ในการเตรียมข้อมูล

ในขั้นตอนการเตรียมข้อมูลสำหรับคลังข้อความที่อยู่ในรูปแบบ XML ข้อความต้องแปลงนั้นให้อยู่ในรูปแบบ arff โดยใช้โปรแกรมที่เกี่ยวข้องคือ Ngram Statistic Package (NSP) ในการสร้าง N-gram ให้กับข้อความ ในบทความนี้ได้ทดลองโดยการสร้าง 2-gram (เนื่องจากต้องการเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีทั้งสามคือ IBk, ID3 และ NaiveBayes เท่านั้น) และ SenseTools จะถูกใช้ในการแปลงข้อความให้อยู่ในรูปแบบ arff

2.2.1 Ngram Statistic Package

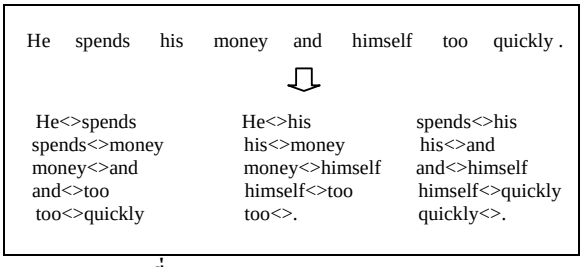
Ngram Satistic Package (NSP) [9] เป็นโปรแกรมที่ ช่วยในการวิเคราะห์สร้าง Ngram ให้กับข้อความ ตัวอย่างการสร้าง 2-gram ด้วย NSP จากประโยค “He spends his money and himself too quickly.” ดังภาพที่ 2 เมื่อสร้างเป็น 2-gram ด้วย NSP จะพิจารณาคำแรกกับคำถัดไปอีก 2 คำ โดยจับคู่คำหลักคือ He กับคำถัดไปลำดับที่ 1 คือ spends จะได้ He<>spends และพิจารณาคำหลักกับคำถัดไปลำดับที่ 2 คือ his จะได้ He<>his เมื่อพิจารณา 2 คำเสร็จให้คำถัดไปเป็นคำหลักในที่นี้คือ spends จะได้เป็น spends<>his และ Spends<>money แล้วพิจารณาแบบเดิมไปเรื่อยๆจนหมดทุกคำจะได้ 2-gram ของประโยค 1 ประโยค

2.1.2 SenseTools

SenseTools เวอร์ชัน 0.3 [10] เป็นโปรแกรมที่ทำหน้าที่แปลงข้อความให้อยู่ในรูปแบบของ arff ซึ่งเป็นรูปแบบที่ WEKA ใช้ ซึ่งทำงานร่วมกับ NSP

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE corpus SYSTEM
  "lexical-sample.dtd">
  <corpus lang='english'>
    <lexelt item="art.n">
<instance id="art.40002" docsrc="bnc_A70_2636">
<answer instance="art.40002" senseid="art_gallery%1:06:00::"/>
<context>
Leeds is well-equipped for sports, with 21 golf courses and 22
sports and leisure centres, but if all this action leaves you
feeling in need of a rest, you can always take yourself off to
the theatre.Leeds has four to choose from. Most famous is the
Leeds City Varieties, one of the oldest music halls in the
country and home of BBC TV's [hi]The Good Old Days [/hi].
There's also the Grand Theatre, which hosts touring companies
and is the permanent home of Opera North. [/p] [p]
One of Yorkshire's famous sayings is &bquo;Where there's muck,
there's brass&equo;. And, while there may not be a lot of muck
any more, there is still plenty of brass. [/p] [p]
For, when it comes down to it, there's nothing to beat a brass band.
There's always one to be heard somewhere during the summer
&mdash; in the piazza in front of the <head>art</head>gallery
and Town Hall or in a park.
</context>
</instance>
<instance id="art.40004" docsrc="bnc_A6U_637">
<answer instance="art.40004" senseid="art%1:04:00::"/>
<context>
When things are on the up and the lodestar of a transformatory
politics shines bright, so too does &bquo;the avant-garde project
of overcoming the separation of art and life&equo; (p. 171). [/p] [p]
In this perspective it seems that Callinicos can only mean
relatively little with his disclaimers about good art.
The individual
&bquo;good&equo; work might get thrown up, however unpropitious the
circumstance. But it can only be a quirk; and the force of its
&bquo;goodness&equo; is strictly limited and circumscribed.
Only once, in a fleeting reference to Matisse is there a sense of
the boot being on the other foot, of art offering a sense of
liberation from social ideology. But even this is done in the
name of a supposed &bquo;immediate sensuous charge&equo; rather
than any more extended critical capacity of <head>art</head>or
the aesthetic.
</context>
</instance>
</lexelt>
</corpus>
```

ภาพที่ 1 : แสดงลักษณะของ Senseval-2



ภาพที่ 2 : แสดงการสร้าง 2-gram ด้วย NSP

2.3. การตัดคำ

ในการสืบค้นเอกสาร (Information Retrieval) [7] จะมีคำบางคำในข้อความซึ่งเป็นคำที่มีความเกี่ยวข้องกับเอกสารน้อย ทำให้ประสิทธิภาพในการค้นคืนเอกสารต่ำลง และเอกสารมีขนาดใหญ่ขึ้น คำพวกนั้นเรียกว่า stoplist ตัวอย่างเช่น to, of, and, but, could, the และ is เป็นต้น คำเหล่านี้เมื่อตัดออกจะทำให้ประสิทธิภาพในการสืบค้นเอกสาร

ดีขึ้นและมีความสำคัญต่อการแก้ปัญหาความกำกวมด้วย เนื่องจากคำเหล่านี้ไม่ได้นำมาวิเคราะห์หาความหมายของคำ บทความนี้ได้นำเทคนิคการตัดคำที่เป็น stoplist ออกจากคลังข้อความเพื่อให้เพิ่มความถูกต้องในการแก้ปัญหาความกำกวม และเพิ่มความรวดเร็วในการทดลองเนื่องจากมีการตัดคำที่ไม่สำคัญทิ้งไป

2.4. เทคนิคการจำแนกโดยการทำเหมืองข้อมูล

เทคนิคการจำแนกของการทำเหมืองข้อมูลที่เลือกใช้ในบทความนี้มี 3 แบบคือ IBk, ID3 และ NaiveBayes โดยมีรายละเอียดดังนี้ดังนี้

ขั้นตอนวิธี IBk [11] เป็นขั้นตอนวิธีอย่างง่ายของการจำแนกแบบ K-Nearest Neighbor ซึ่งเป็นเทคนิคการจำแนกประเภทหนึ่งมีลักษณะเดียวกันกับการจัดแบ่งคลาส ในการใช้งาน K-NN นั้นต้องระบุค่าตัวเลขจำนวนเต็มบวกให้กับ K เช่น 1-NN, 2-NN, 3-NN,…, K-NN โดยที่ K เป็นตัวบอกจำนวนกรณีที่จะต้องค้นหาในการทำนายกรณีใหม่ เช่น 4-NN หมายถึงขั้นตอนวิธีนี้จะหาค่า 4 กรณีที่มีลักษณะใกล้เคียงกับกรณีใหม่มากที่สุดและกำหนดเงื่อนไขใหม่ๆให้กับคลาสที่ใกล้เคียงมากที่สุด

ขั้นตอนวิธี ID3 [12] เป็นขั้นตอนวิธีของการจำแนกโดยใช้ต้นไม้การตัดสินใจ ขั้นตอนวิธีนี้จะใช้ตัวอย่างในการสร้างต้นไม้ซึ่งนำมาใช้ในการจำแนกข้อมูลที่ไม่รู้ โครงสร้างของต้นไม้ในแต่ละโหนดจะเป็นแอทริบิว แต่ละกิ่งจะเป็นผลในการทดสอบและลีฟโหนดแสดงคลาสที่กำหนดไว้

ขั้นตอนวิธี NaiveBayes [11] เป็นขั้นตอนวิธีของการจำแนกโดยใช้หลักการของทฤษฎีเบย์ในการคำนวณหาความน่าจะเป็นซึ่งถูกใช้ในการทำนายผลเมื่อทำการวิเคราะห์กรณีใหม่ การทำนายผลทำได้โดยการรวมผลของตัวแปรอิสระที่มีต่อตัวแปรตามโดยจะวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรอิสระแต่ละตัวกับตัวแปรตามเพื่อใช้ในการสร้างเงื่อนไขความน่าจะเป็นสำหรับแต่ละความสัมพันธ์

3. แบบจำลองการแก้ปัญหาความกำกวมของคำโดยใช้เทคนิคการตัดคำ

จุดประสงค์ของการทดลองนี้คือการเปรียบเทียบประสิทธิภาพของขั้นตอนวิธี 3 ขั้นตอนวิธีคือ IBk, ID3 และ NaiveBayes ในการแก้ปัญหาความกำกวมโดยใช้เทคนิคการตัดคำ มี 3 ขั้นตอน

ขั้นตอนที่ 1 เตรียมคลังข้อความ ในที่นี้จะใช้ eng-lex-sample เป็นข้อความภาษาอังกฤษของคลังข้อความ Senseval-2 อยู่ในรูปแบบของ XML

ขั้นตอนที่ 2 ใช้ SenseTools และ NSP ในการแปลงข้อความ XML ให้อยู่ในรูปแบบ arff เพื่อนำไปทดลองใน WEKA [13]

ขั้นตอนที่ 3 เมื่อได้ข้อมูลในรูปแบบ arff นำไปทดลองกับ WEKA โดยใช้สามขั้นตอนวิธี คือ IBk, ID3 และ NaiveBayes โดยขั้นตอนวิธีโดยใช้ IBk ทำการทดลองโดยให้ค่า k เป็น 1,2,…,10

ในการทดลองแบ่งออกเป็น 2 แบบ คือ

แบบที่ 1 ขั้นตอนการแก้ปัญหาความกำกวมของคำโดยการไม่ตัดคำ ทำการทดลองในขั้นตอนที่ 1, 2 และ 3 จากภาพที่ 3 เพื่อดูประสิทธิภาพของขั้นตอนวิธีแต่ละวิธีในการแก้ปัญหาความกำกวมเมื่อทดลองโดยไม่มีการตัดคำ

แบบที่ 2 ขั้นตอนการแก้ปัญหาความกำกวมของคำโดยการตัดคำ ทำการทดลองในขั้นตอนที่ 1, 2, 3 และ 4 จากภาพที่ 4 โดยในขั้นตอนนี้ทำการทดลองเพิ่มการตัดคำเพื่อดูประสิทธิภาพของขั้นตอนวิธีแต่ละวิธีในการแก้ปัญหาความกำกวมโดยเพิ่มการตัดคำที่เป็น stoplist ออก

ภาพที่ 3 : แสดงขั้นตอนการแก้ปัญหาความกำกวมของคำโดยการไม่ตัดคำ

ภาพที่ 4 : แสดงขั้นตอนการแก้ปัญหาความกำกวมของคำโดยการตัดคำ

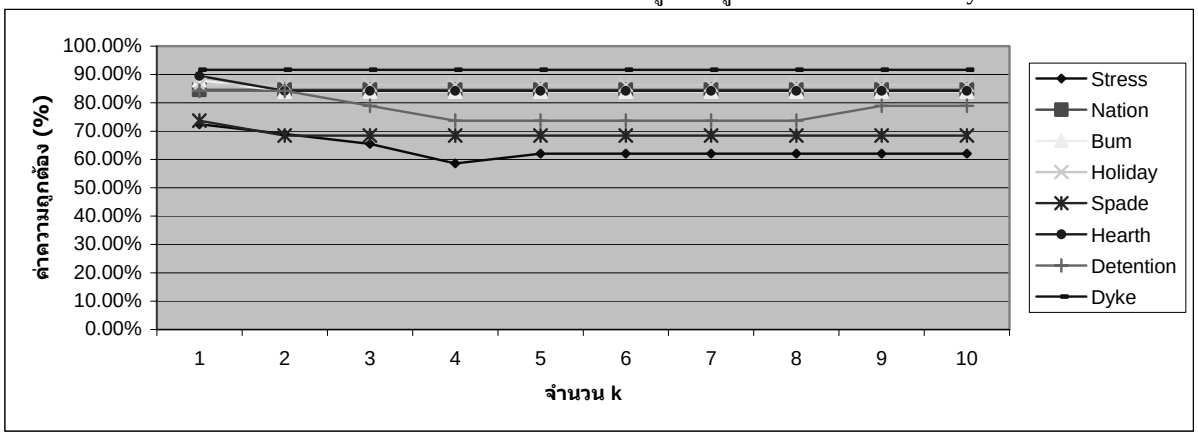
4. ผลการทดลอง

ข้อมูลที่ใช้ในการทดลองคือคลังข้อความ Senseval-2 คำกำกวมจากคลังข้อความ Senseval-2 ที่ใช้ในการทดลองแสดงดังตารางที่ 1 โดยประกอบด้วยคำกำกวมดังต่อไปนี้ stress, fatigue, nation, bum, holiday, spade, hearth, detention และ dyke ซึ่งแต่ละคำจะมีจำนวนตัวอย่างแตกต่างกัน ทำงานใน WEKA [13] แบ่งข้อมูลเป็น Train Set และ Test Set โดยทดลองกับสามขั้นตอนวิธีคือ IBk, ID3 และ NaiveBayes สำหรับการทดลองในขั้นตอน IBk นั้นได้ทำการทดลองเริ่มต้นโดยกำหนดค่า k ให้มีค่า 1 ถึง 10 จากภาพที่ 5 ผลการทดลองเมื่อเพิ่มค่า k ขึ้นค่าความถูกต้องจะมีค่าคงที่ตัวอย่างเช่นคำว่า “dyke” และมีบางคำที่มีความถูกต้องลดลงตัวอย่างเช่นคำว่า “stress” และ “detention” เป็นต้น และจากค่าความถูกต้องที่ได้ ค่า k เท่ากับ 1 จะมีความถูกต้องสูงสุด ดังนั้นจึงได้เลือกค่า k เท่ากับ 1 ในการทดลองขั้นต่อไปของขั้นตอนวิธี IBk

ผลการทดลองแสดงให้เห็นว่าการแก้ปัญหาความกำกวมโดยใช้การตัดคำ (แบบที่ 2) จากทั้งสามขั้นตอนวิธีคือ IBk, ID3 และ NaiveBayes จะให้ค่าความถูกต้องสูงกว่าการทดลองโดยไม่มีการตัดคำที่เป็น stoplist (แบบที่1) ตัวอย่างเช่น hearth จากตารางที่ 1 โดยขั้นตอนวิธี IBk ไม่ตัดคำมีความถูกต้อง 68.42% เมื่อตัดคำได้ค่าความถูกต้อง 89.47 % ขั้นตอนวิธี ID3 ไม่ตัดคำมีความถูกต้อง 47.36% เมื่อตัดคำได้ค่าความถูกต้อง 84.21% ขั้นตอนวิธี NaiveBayes ไม่ตัดคำมีความถูกต้อง 68.42% เมื่อตัดคำได้ค่าความถูกต้อง 84.21% ผลการเปรียบเทียบแต่ละขั้นตอนวิธีการทดลองแบบ 1 และแบบ 2 จากขั้นตอนวิธี IBk , ID3 และ NaiveBayes แสดงได้ดังภาพที่ 6, 7 และ 8 ตามลำดับ ภาพที่ 9 แสดงให้เห็นว่าเมื่อทดลองโดยใช้การแก้ปัญหาความกำกวมโดยใช้การตัดคำโดยส่วนใหญ่ขั้นตอนวิธี IBk ให้ค่าให้ค่าความถูกต้องสูงกว่าใช้ขั้นตอนวิธี ID3 และ NaiveBayes

5. บทสรุป

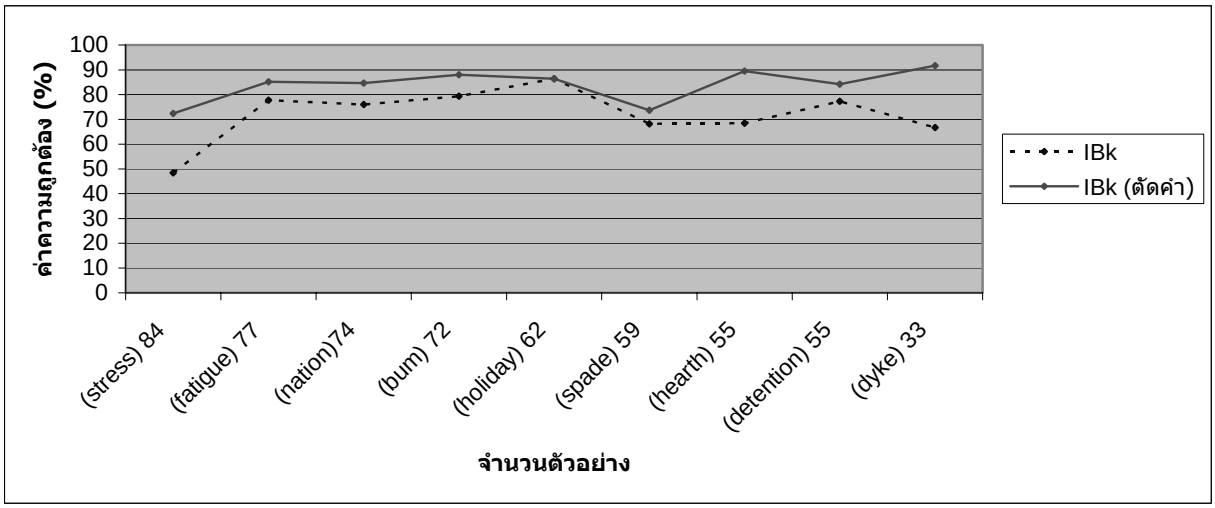
บทความนี้นำเสนอขั้นตอนการแก้ปัญหาความกำกวมของคำโดยใช้การเปรียบเทียบประสิทธิภาพของขั้นตอนวิธี 3 วิธีคือ IBk, ID3 และ NaiveBayes สามารถสรุปได้ว่า การแก้ปัญหาความกำกวมโดยใช้การตัดคำให้ค่าความถูกต้องสูงกว่าการแก้ปัญหาความกำกวมโดยไม่ตัดคำ และขั้นตอนวิธีที่ให้ค่าความถูกต้องสูงสุดคือ ขั้นตอนวิธี IBk ซึ่งให้ค่าความถูกต้องสูงกว่า ID3 และ NaiveBayes



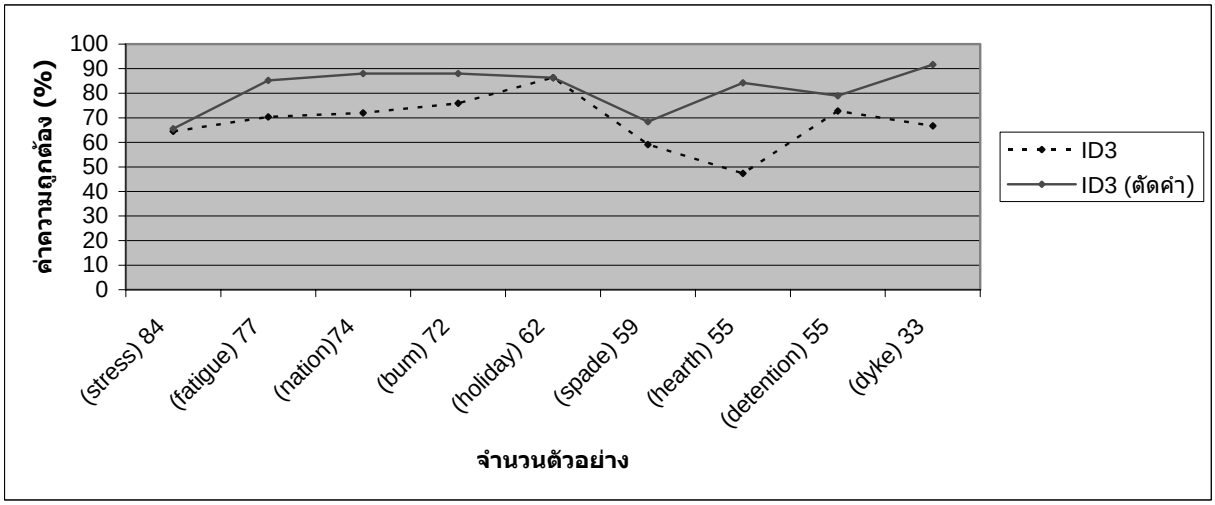
ภาพที่ 5: แสดงค่าความถูกต้องเมื่อใช้ขั้นตอนวิธี IBk เมื่อค่า k มีค่าต่างกัน

ตารางที่ 1: ผลการทดลองโดยเปรียบเทียบการใช้การตัดคำและไม่ตัดคำในแต่ละขั้นตอนวิธี ในขั้นตอนวิธี IBk กำหนดค่า k ให้มีค่าเท่ากับ 1

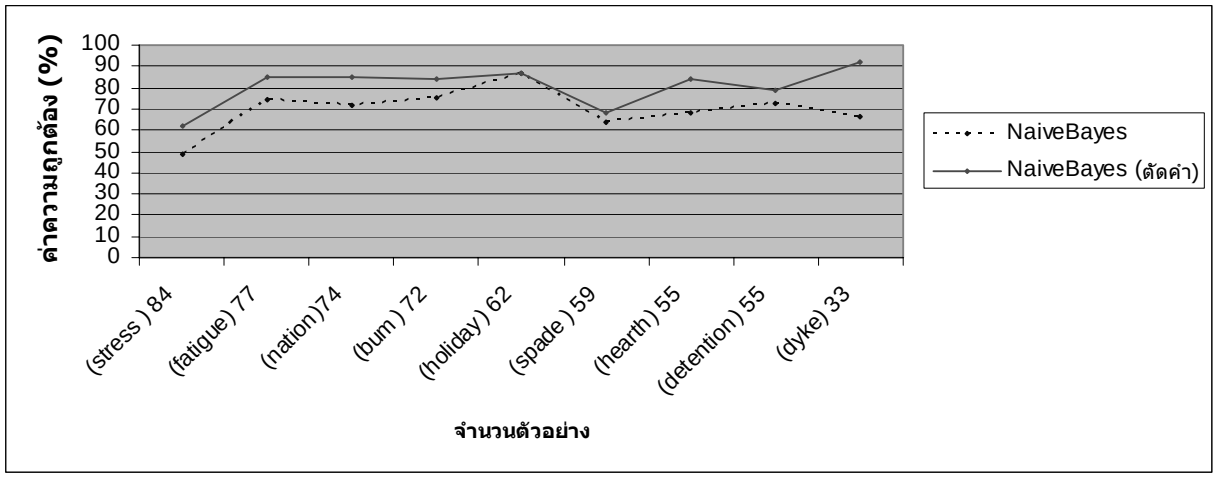
คำ	จำนวนตัวอย่าง	Classified Accuracy					
		IBk (k=1)		ID3		NaiveBayes	
		แบบ1	แบบ2	แบบ1	แบบ2	แบบ1	แบบ2
stress	84	48.38%	**72.41%**	64.51%	**65.51%**	48.38%	**62.06%**
fatigue	77	77.77%	**85.15%**	70.37%	**85.18%**	74.01%	**85.18%**
nation	74	76.00%	**84.61%**	72.00%	**88.00%**	72.00%	**84.61%**
bum	72	79.31%	**88.00%**	75.86%	**88.00%**	75.00%	**84.00%**
holiday	62	86.36%	**86.39%**	86.36%	**86.36%**	86.36%	**86.36%**
spade	59	68.18%	**73.68%**	59.09%	**68.42%**	63.63%	**68.42%**
hearth	55	68.42%	**89.47%**	47.36%	**84.21%**	68.42%	**84.21%**
detention	55	77.27%	**84.21%**	72.72%	**78.94%**	72.72%	**78.94%**
dyke	33	66.67%	**91.66%**	66.66%	**91.66%**	66.66%	**91.66%**



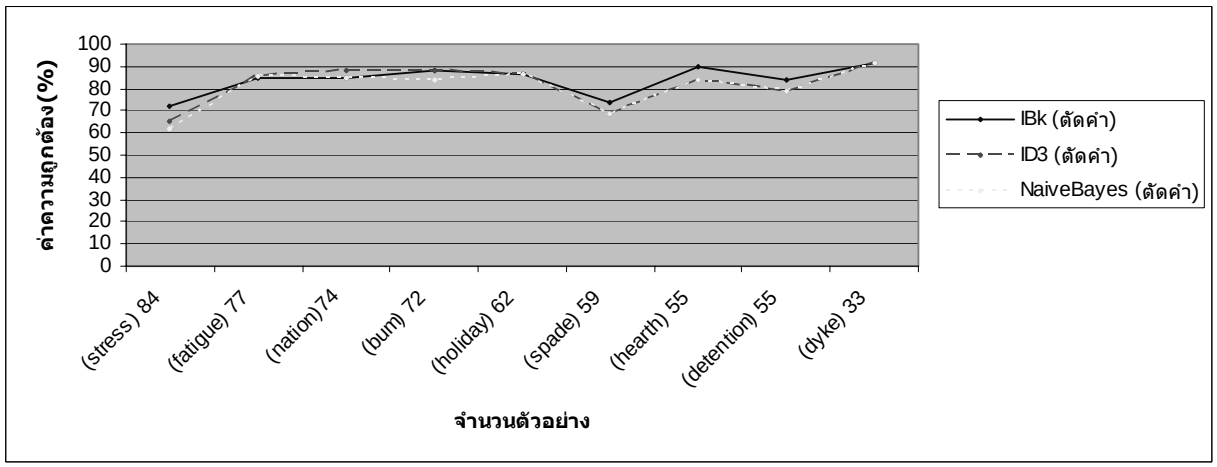
ภาพที่ 6: แสดงค่าความถูกต้องเมื่อใช้ขั้นตอนวิธี IBk เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบปกติและแบบตัดคำ



ภาพที่ 7: แสดงค่าความถูกต้องเมื่อใช้ขั้นตอนวิธี ID3 เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบปกติและแบบตัดคำ



ภาพที่ 8: แสดงค่าความถูกต้องเมื่อใช้ขั้นตอนวิธี NaiveBayes เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบปกติและแบบตัดคำ



ภาพที่ 9: แสดงค่าความถูกต้องเมื่อใช้ขั้นตอนวิธี IBk, ID3, และ NaiveBayes เมื่อทดลองโดยการแก้ปัญหาความกำกวมแบบตัดคำ

6. เอกสารอ้างอิง

[1] L. Spacia ,"A Hybrid Model for Word Sense Disambiguation in English-Portuguese Machine Translation," *Computational Linguistics UK* , pp. 71-78, 2005.

[2] D. Dinh,"Building a training corpus for word sense disambiguation in English – to – Vietnamese Machine Translation" , *International Conference On Computational Linguistics*, pp. 1-7, 2002.

[3] S.Wermter, and C. Hung, "Selforganizing classification on the Reuters news corpus" , *The 19th lntemnrionnl Conference on Computnthml Linguistics*, 2002.

[4] J. Preiss, "Probabilistic WSD in SENSEVAL-3," *Annual Meeting of the Association for Computational Linguistics*, pp. 213-216, 2004.

[5] J. Chang, J.Chen, H.Sheng, and S.Ker, "Combining Machine Readable Lexical Resources And Bilingual Corpora for Broad Word Sense Disambiguation" , In *Proceedings of Conference of the America Machine Translation Association 96*, pp. 115-124, 1996.

[6] K.Linden, and K. Lagus,"Word Sense Disambiguation in Document Space", *IEEE Int. Conference on Systems, Man and Cybernetics*, pp. 98-107, 2002.

[7] William B.Frakes, Ricardov Baeza-Yates "Information Retrieval Data structure & Algorithm" ,1992.

[8] Senseval [online].available: http://www.senseval.org/

[9] NSP [online].available: http://search.cpan.org/dist/Text-NSP

[9] SenseTools [online]. available: http://www.d.umn.edu/~tpederse/sensetools.html

[11] Sripatum University [online].available:alumni.spu.ac.th/mallika/msit7/ webboard/fileupload

[12] Mahanakorn University of Technology [online]. available: www.mut.ac.th/~b3121065/datamining.doc

[13] WEKA (Waikato Environment for Knowledge Analysis) [online].available: http://www.cs.waikato.ac.nz/ml/weka

ภาคผนวก ค

ผลงานตีพิมพ์

เรื่อง Word Sense Disambiguation and Attribute Selection Using Gain Ratio and RBF Neural Network

Conference 2008 IEEE International Conference On Research, Innovation and Vision for the Future in Computing & Communications Technologies (RIVF' 08)

สถานที่ ประเทศเวียดนาม

วันที่ 13-17 กรกฎาคม 2551

Word Sense Disambiguation and Attribute Selection using Gain Ratio and RBF Neural Network

Kanjana Thongklin
Faculty of Technology and Environment
Prince of Songkla University
Phuket, Thailand
kanjana_thongklin@hotmail.com

Sirirut Vanichayobon
iSTAR Research Laboratory
Computer Science Department
Prince of Songkla University
Songkhla, Thailand
sirirut.v@psu.ac.th

Wiphada Wettayaprasit
Artificial Intelligence Research Laboratory
Computer Science Department
Prince of Songkla University
Songkhla, Thailand
wwettayaprasit@yahoo.com

***Abstract*—Word sense disambiguation is one of natural language processing tasks. This study proposes new idea for word sense disambiguation by using context window of left-hand side type, right-hand side type and both left-hand and right-hand sides and the attribute selection. The techniques used for this study are GainRatioAttributeEval and InfoGainAttributeEval. RBF Neural Network and ID3 algorithm are used to classify the sense of words. The result of the study from Senseval-2 corpus indicates that the context window of both left-hand side and right-hand side give highest accuracy. The attribute selection by the GainRatioAttributeEval technique gives higher accuracy than InfoGainAttributeEval. The RBF Neural Network algorithm gives higher accuracy than the ID3 algorithm.**

Keywords-natural language processing; word sense disambiguation; RBF Neural Network; ID3; attribute selection

I. INTRODUCTION

In the communication by natural language, there are some of same words used that have different meanings when these same words are used in different contexts. The same words with different meanings are the causes of ambiguity. The consequence of ambiguity of words brings to the error of machine translation [1, 2] and information retrieval [3]. Word sense disambiguation is a process to represent the right meaning for the ambiguous words.

Section 2 of this study is the principle of context window, attribute selection, and classification. Section 3 proposes (WSD_AS) model. Section 4 is the result of the study by using Senseval-2. Section 5 is the conclusion.

II. CONTEXT WINDOW ATTRIBUTE SELECTION AND CLASSIFICATION

A. Context Window

The contexts of ambiguous words are words around those ambiguous words of both left-hand side and right-hand side. The context of each word will mention on things that have relations with the meaning in that sentence. In using context to problem solving for the ambiguous words, this will give the right meaning of those ambiguous words. For example, the sentence with ambiguous word "art" is shown as follows:

"There's always one to be heard somewhere during the summer; in the piazza in front of the ***art*** *gallery and Town Hall or in a park."*

The context on the left-hand side means all words on the left-hand side of "**art**". The context on the right-hand side means all words on the right-hand side of "**art**". When the window size is +n, this means moving to the right-hand side n positions from the ambiguous word. And when the window size is –n, this means moving to the left hand-side n positions from the ambiguous word. The previous studies that used the context to problem solving for the ambiguous words are such as using both left-hand side and right-hand side with the window size of ± 2 for maximum entropy model [4], and class-based collocations model [5], using both left-hand side and right-hand side with the window size of ± 1 ± 2 ± 3 for feature selection for maximum entropy-based model [6].

B. Attribute Selection

Attribute selection is a method of reducing for non-related number of attributes. The number of attributes will be reduced to have only related attributes. The advantage of reducing the number of attributes is using significant samples to be trained. The result of this selection gives higher accuracy. The studies of data mining that used the technique of attribute selection are such as using UCI repository of machine learning database composed of 9 databases by the selection of Information Gain Ratio Attribute Evaluation and Relief Attribute Evaluation [7] and using Speaker Recognition Evaluation (SRE) database by selection Information Gain Attribute Evaluation and Gain Ratio Attribute Evaluation [8]. The examples of attributes selection for the study are as follows.

1) Information Gain Attribute Evaluation:

This selection is a reduction for the number of attributes that will be used to evaluate the value of attribute by measuring Information Gain values [8, 9]. Information Gain (IG) can be calculated by (1).

$$IG = H(Y) - H(Y \mid X) \quad (1)$$

where Y is class and X is input attribute

$H(Y)$ is entropy of Y

$H(Y \mid X)$ is conditional entropy of Y given X

The calculation of $H(Y)$ is shown by (2) and the calculation of $H(Y \mid X)$ is shown by (3).

$$H(Y) = -\sum_{y \in Y} p(y) \log_2 (p(y)) \quad (2)$$

$$H(Y \mid X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y \mid x) \log_2 (p(y \mid x)) \quad (3)$$

where $p(y)$ is the probability of y , $p(x)$ is the probability of x , and $p(y \mid x)$ is the conditional probability of y given x

2) Gain Ratio Attribute Evaluation:

This selection is a reduction for the number of attributes that will be used to evaluate the value of attribute by measuring Gain Ratio [8, 9]. There will be an adjustment on the scale due to the value of data in the interested attribute with that class. Gain Ratio (GR) can be calculated by (4).

$$GR = \frac{IG}{H(X)} \quad (4)$$

Both of these two attribute selections are used in Ranker Search Method [9]. Attributes will be sorted according to their significant values. The unused attributes will be eliminated. The difference between IG and GR [8, 9] is that GR received from the dividing by entropy value that the output value is in the range [0, 1]. If GR is equal to 1, this means that there is highest relationship between Y and X . The value received from GR will be smaller when compared with the value of IG .

C. *Classification*

Classification is a technique of data mining. This study will use two techniques of ID3 and RBFNetwork. ID3 is an algorithm using decision tree [10]. ID3 algorithm will be used to create the decision tree while each node represents an attribute, each branch represents the test result, and leaf node represents the class. RBFNetwork or RBF Neural Network [9, 10] is composed of three layers that are input layer, hidden layer, and output layer. RBF Neural Network uses Radial Basis Activation Function which is Gaussian Function ($\phi(X)$) in (5) for *j*= 1,…,L, where X is the input feature, L is the number of hidden units, and μ_j is the average of j^{th} Gaussian Function.

$$\phi_j(X) = \exp\left[-(X - \mu_j)^T \sum_j^{-1} (X - \mu_j) \right] \quad (5)$$

III. WORD SENSE DISAMBIGUATION ATTRIBUTE SELECTION MODEL USING GAIN RATIO AND RBF NEURAL NETWORK

There are 4 steps of word sense disambiguation and attribute selection (WSD_AS) model as shows in Fig 1. Step 1 is data preprocessing. Step 2 is creating attribute using context window. Step 3 is attribute selection. And step 4 is word sense classification.

Step 1: Data Preprocessing
1.1 Preprocess data for Senseval-2 in the format of XML.
1.2 Use SenseTools program to split ambiguous words that is in the format of XML. Each ambiguous word will be kept in each separated file.
1.3 Eliminate stoplist words from the data.
Step 2: Create Attribute Using Context Window
2.1 Specify the window size of context = n.
2.2 Construct attribute by using context window with 3 cases.
2.1.1 Use left hand side of the context window.
2.1.2 Use right hand side of the context window.
2.1.3 Use both left hand and right hand sides of the context window.
2.3 Use SenseTools program and NSP program to convert the sentence into the form of feature vectors (0 or 1) with arff format file.
2.3.1 Create N-Gram Using NSP Program (count.pl)
Command : count.pl –ngram n OUTPUT_FILE INPUT_FILE
2.3.2 Create Regular Expressions Using SenseTools Program (nsp2regex.pl)
Command : nsp2regex.pl INPUT_FILE REGEX_FILE
2.3.3 Create Feature Vectors Using SenseTools Program (xml2arff.pl)
Command : xml2arff.pl –training TRAIN_FILE – test TEST_FILE REGEX_FILE
2.3.4 Change symbol "%" to be symbol "~" Using SenseTools Program (tilde.pl)
Command : tilde.pl SOURCE >> OUTPUT
Step 3: Attribute Selection
3.1 Select attribute selectors.
3.1.1 InfoGainAttributeEval. See (1)
3.1.2 GainRatioAttributeEval. See (2)
3.2 Select the number of attributes needed for selection such as 40, 50 or 60.
Step 4: Word Sense Classification
4.1 Choose number of class for classification.
4.1.1 Set the class number into two classes only (YES or NO).
4.1.2 Choose the class number more than two classes.
4.2 Choose classification algorithm.
4.2.1 RBFNetwork
4.2.2 ID3
4..3 Calculate the classification accuracy.

Figure 1. Show the proposed WSD_AS Model .

A. *Step1:Data Preprocessing*

Senseval-2 [11] is standard data in measuring the efficiency of word ambiguity problem solving. The data composes of sub-data with various languages such as Italian, Japanese, English, and etc. This study selected English language as shows in Fig 2.

1.1) The preprocessing data for senseval-2 is to set in the format of XML using English Language (eng-lex-sample). For example, from Fig 2., ambiguous word "**art**" has 3 meanings: skill, art work, or art creation.

1.2) Use SenseTools program [12] to split ambiguous words that is in the format of XML from Senseval-2 data to be the file of each ambiguous word.

1.3) Eliminate stoplist words [13, 14] and punctuation such as ; , . ( ) , and etc from file due to those stoplist words are redundancy and will enlarge the size of the file. Since these words will not be analyzed to find the meaning of the

ambiguous word. Sample of stoplist words is shown in Fig 3. The elimination of stoplist word can be below. Suppose that the sentence is shown as follows :

"There's always one to be heard somewhere during the summer; in the piazza in front of the **art** gallery and Town Hall or in a park."

The word "**art**" (bold) has ambiguous meaning. The stoplist words are those underlined. When the stoplist words and punctuation are eliminated, the result received is shown as follows:

"heard summer piazza front **art** gallery Town Hall park"

B. Step 2: Attribute Construction by Context Window

2.1) The experiment will select the window size of the context. The window size used for the experiment is n. When n = 1, this means that there is adding one more word to be considered as attribute value.

2.2) The experiment will construct attribute by using context. Let w be an ambiguous word and n be the window size. Then $w_{+1}, w_{+2}, w_{+3}, \ldots, w_{+n}$ are contexts on the right-hand side and $w_{-n}, \ldots, w_{-3}, w_{-2}, w_{-1}$ are contexts on the left-hand side. For example, when the window size is one (n = 1), the attribute construction can be selected 3 cases as shows in Fig 4.

Case 1: Move the context to the left-hand side with one more position: (w_{-1} **w**).

Case 2: Move the context to the right-hand side with one more position: (**w** w_{+1}).

Case 3: Move the context to both right-hand side and left-hand side with one more position: (w_{-1} **w** w_{+1}).

Similarly, if the size of the window is 2 (n=2), the context received will have to add two more words on the left-hand side or the right-hand side, therefore case 1 will be w_{-2} w_{-1} **w**, case 2 will be **w** w_{+1} w_{+2} , and case 3 will be w_{-2} w_{-1}**w** w_{+1} w_{-2}, respectively. For example, in Fig 5., from the following sentence "heard summer piazza front **art** gallery Town Hall park." if the context window is 3 (n = 3), case 1 will be "summer piazza front **art**", case 2 will be "**art** gallery Town Hall", case 3 will be "summer piazza front **art** gallery Town Hall", respectively.

2.3) Use SenseTools program and NSP program [15] to convert the sentence into the form of feature vectors (0 or 1). The family name of the file received is arff format. There are 4 sub-steps of converting the sentence with SenseTools program and NSP program as shows in Fig 1.

2.3.1 The count.pl is a file for constructing words in N-gram format. The propose of this sub-step is to count the frequency of the each word in the context window. In the experiment, we will used 1-Gram. For example, from Fig 5., with case 3, context window selection (both left-hand side and right-hand side), the output of this sub-step 1-Gram is shown as follow: "**Town<>1**" means the word "**Town**" has the frequency equal to 1, etc.

2.3.2 The nsp2regex.pl is a file for constructing Regular Expressions of each word.

2.3.3 The xml2arff.pl is a file for identifying the instance word in the form of feature vectors (0 or 1). If the instance word is existing, then the feature vector value is equal to 1, otherwise it will equal to 0 (not existing).

2.3.4 The tilde.pl is a file for changing symbol "%" to be ready to use for WEKA [16] as shows in Fig 6. The first part will describe the relation of each attribute.

English language of Senseval-2

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE corpus SYSTEM
  "lexical-sample.dtd">
<corpus lang='english'>
<lexelt item="art.n">
<instance id="art.40004" docsrc="bnc_A6U_637">
```

The sense of ambiguous word "art" is 1:04:00

```
<answer instance="art.40004" senseid="art%1:04:00::"/>
```

Start the sentence

```
<context>
When things are on the up and the lodestar of a transformatory politics shines bright , so too does <&> bquo ; the avant <-> garde project of overcoming the separation of art and life <&> equo ; <(> p . 171 <)> . <[/> p <] [> p <|> In this perspective it seems that Callinicos can only mean relatively little with his disclaimers about good art . The individual <&> bquo ; good <&> equo ; work might get thrown up , however unpropitious the circumstance . But it can only be a quirk ; and the force of its <&> bquo ; goodness <&> equo ; is strictly limited and circumscribed . Only once , in a fleeting reference to Matisse is there a sense of the boot being on the other foot , of art offering a sense of liberation from social ideology . But even this is done in the name of a supposed <&> bquo ; immediate sensuous charge <&> equo ; rather than
any more extended critical
```

Show ambiguous word "art"

```
capacity of <<> head <>> art <</> head <>> or the aesthetic .
</context>
</instance>
</lexelt>
</corpus>
```

Figure 2. Ambiguous word "art" in Senseval-2 corpus.

a	and	above	always	after	there	before
both	of	one	in	is	he	must
be	the	we	somewhere	you	to	or

Figure 3. Sample of stoplist words.

LHS	:	W_{-3}	W_{-2}	W_{-1}	**W**		
RHS	:				**W**	W_{+1}	W_{+2}	W_{+3}
LHS & RHS:		W_{-3}	W_{-2}	W_{-1}	**W**	W_{+1}	W_{+2}	W_{+3}

Figure 4. Show selected attribute with window size = 1.

LHS	:	W_{-3} summer	W_{-2} piazza	W_{-1} front	**W art**		
RHS	:				**W art**	W_{+1} gallery	W_{+2} Town	W_{+3} Hall
LHS & RHS:		W_{-3} summer	W_{-2} piazza	W_{-1} front	**W art**	W_{+1} gallery	W_{+2} Town	W_{+3} Hall

Figure 5. Show example of selected attribute with window size = 3.

C. *Step 3: Attribute Selection*

3.1) Select the type of attribute selections, which are InfoGainAttributeEval and GainRatioAttributeEval.

3.2) Select the numbers of attribute such as 40, 50, 60, 100, 150, and 200.

D. *Step 4: Word Sense Classification*

4.1) Select the method of classification by separating into only 2 classes of meanings or by using the whole classes of meanings when the number of class is greater than two. For

```
@relation 'RELATION'
@attribute 'Town' {0,1}
@attribute 'gallery' {0,1}
@attribute 'front' {0,1}
@attribute 'piazza' {0,1}
@attribute 'summer' {0,1}
@attribute 'art' {0,1}
@attribute 'Hall' {0,1}
@attribute 'senseclass' {art~1:06:00, art~1:09:00, art~1:04:00}
@data
{0,0,0,0 ,1,1,0, art~1:06:00
{1,1,1,1,1,1,1, art~1:06:00
{0,0,0,0,0,1,0 , art~1:04:00}
{0,0,0,0,0, 1,0, art~1:09:00}
```

Figure 6. Example of completed arff file using SenseTools program (tilde.pl)

separating into only 2 classes of meanings, this means that if a word has 3 meanings that are X, Y, and Z, there will be 3 cases of consideration. Case 1 will pay attention only the meaning of X that will receive X = YES and Y, Z = NO. Case 2 will pay attention only the meaning of Y that will receive Y = YES and X, Z = NO. Case 3 will pay attention only the meaning of Z that will receive Z = YES and X, Y = NO. For using the whole classes of meanings, this case will specify the numbers of truly meaning when the numbers of meaning of ambiguous word are more than 2 meanings. For example, if a word has 3 meanings that are X, Y, and Z, let the first meaning be class X, the second meaning be class Y, and the third meaning be class Z.

4.2) This step is to choose the classification algorithm, which is RBFNetwork or ID3. The software used is WEKA [16].

4.3) The last step is to calculate the accuracy of ambiguous words. The accuracy for classification is calculated from the proportion of correct classified instances and number of all instances times 100 as equation (6).

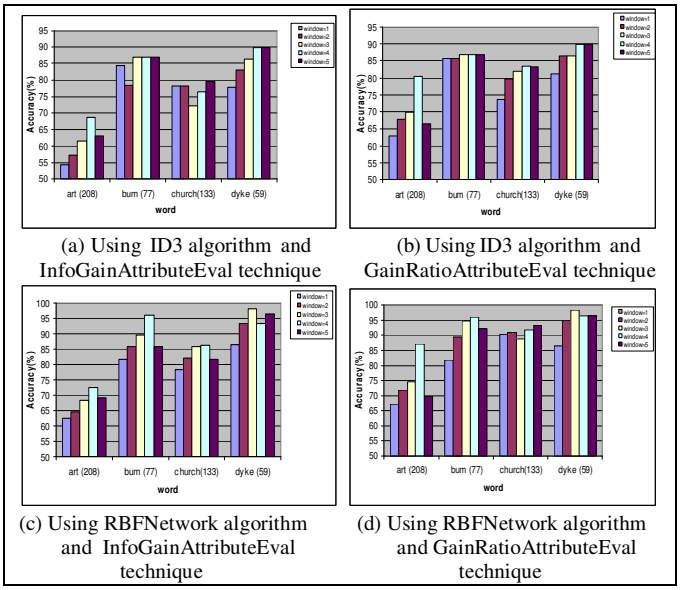
$$\text{accuracy}(\%) = \frac{\text{number of correct classified instances}}{\text{number of all instances}} \times 100 \qquad (6)$$

IV. EXPERIMENT

Senseval-2 corpus is used for this experiment. There are 8 ambiguous words which are art, bar, bum, chair, hearth, stress, dyke, and church. 10-folds Cross Validation is used for this study by dividing data into training set and testing set. The data used will be divided into 10 equivalently. Then 9 parts will be used for training set and 1 part will be used for testing set. Data will be rotated 10 times for different testing set and different training set. The experimental results can be concluded into 6 issues that are issues of A) Selecting the Context Window Size, B) Selecting the Context Window Format, C) The number of Attribute Selection, D) Choosing Algorithm Selection Technique, E) Choosing Classification Algorithm, and F) Choosing the Number of Class.

A. *Issue of Selecting the Context Window Size*

If the width of the window size is increasing, then the accuracy will be increasing as shows in Fig 7(b). For example, the word "**church**" using ID3 algorithm with GainRatioAttributeEval at the window sizes ±1, ±2, ±3, ±4, and ±5 received the accuracy as follows: 73.68%, 79.69%, 81.95%, 83.45%, and 83.33%, respectively. The study indicates that ID3 algorithm and RBFNetwork algorithm (both InfoGainAttributeEval and GainRatioAttributeEval) of other ambiguous words such as art, bum, and dyke have the same result as shows in Fig 7.



(a) Using ID3 algorithm and InfoGainAttributeEval technique

(b) Using ID3 algorithm and GainRatioAttributeEval technique

(c) Using RBFNetwork algorithm and InfoGainAttributeEval technique

(d) Using RBFNetwork algorithm and GainRatioAttributeEval technique

Figure 7. Shows accuracy values when the window sizes are equal to ±1, ±2, ±3, ±4 and ±5 of context on left-hand side and right-hand side.

B. *Issue of Selecting the Context Window Format*

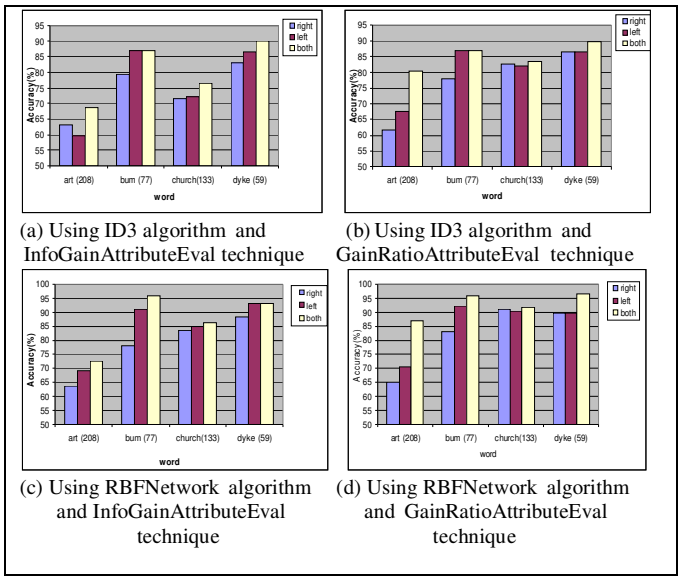
The experimental result shows that context windows on both left-hand side and right-hand side will give higher accuracy than selecting context windows on the left-hand side only or right-hand side only as shows in Fig 8(d). For the ambiguous word "**dyke**", the right-hand side accuracy is 89.83%, left-hand side accuracy is 89.83%, and both right-hand side and left- hand side accuracy is 96.61%, respectively.

C. *Issue of the number of Attribute Selection*

The selection of attribute without selection (non-attribute selection) means using the numbers of attributes received from attribute construction (in step 2). For example, "**art**" has the numbers of 1103 attributes, "**dyke**" has the numbers of 327 attributes, "**church**" has the numbers of 720 attributes, and "**bum**" has the numbers of 450 attributes. When select the attributes equal to 40 attributes, this means that the study will used only 40 attributes. In this experiment, user can choose the numbers of attributes such as 50, 100, 150, 500, and etc.

The experimental result shows that using the attribute selection algorithm will give higher accuracy than using non-

attribute selection as shows in Fig 9(a). For example, the ambiguous word "**art**", with RBFNetwork algorithm and GainRatioAttributeEval, the 40 attributes accuracy is 87.01%, while the non-attribute selection (1103 attributes) accuracy is very low at 54.32%. The experimental results for other words such as dyke, church, and bum are in the same patterns when the smaller selection attribute gave higher accuracy than the non-attribute selection.



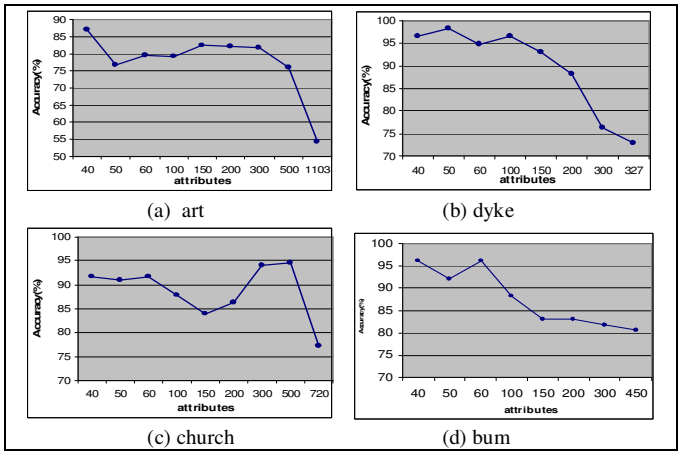
(a) Using ID3 algorithm and InfoGainAttributeEval technique

(b) Using ID3 algorithm and GainRatioAttributeEval technique

(c) Using RBFNetwork algorithm and InfoGainAttributeEval technique

(d) Using RBFNetwork algorithm and GainRatioAttributeEval technique

Figure 8. Shows accuracy values of using right-hand context, left-hand context, and both left and right hands context.



(a) art

(b) dyke

(c) church

(d) bum

Figure 9. Shows the accuracy when compare the selection for the different numbers of attributes and selection of attributes (last value) by RBFNetwork algorithm and GainRatioAttributeEval technique.

D. *Issue of Choosing Attribute Selection Technique*

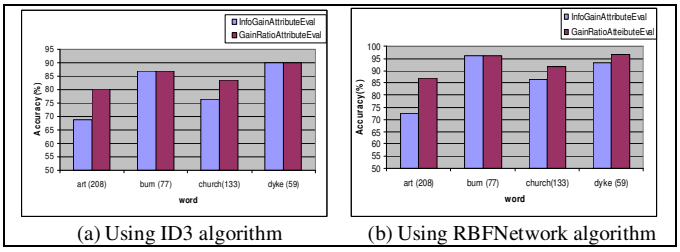
The experimental result shows that GainRatioAttributeEval gives higher accuracy than InfoGainAttributeEval as shows in Fig 10(a). The accuracy of ambiguous word "**church**" with the GainRatioAttributeEval is 83.45%, which is higher than InfoGainAttributeEval at 76.51%. Experimental result from RBFNetwork algorithm has the same pattern. The accuracy of GainRatioAttributeEval is 91.61% that is higher than 86.36% of InfoGainAttributeEval.

E. *Issue of Choosing Classification Algorithm*

The experimental result shows that the RBFNetwork classification algorithm gives higher accuracy than the ID3 classification algorithm. For example, in Fig 11., the ambiguous word "**dyke**" with RBFNetwork algorithm gives higher accuracy at 96.61% when ID3 algorithm is only 89.83%.

F. *Issue of Choosing the Number of Class*

There are two types of the number of class selection. First is the two-class pattern. Second is more than one-class pattern. From Fig 12., the ambiguous word "**art**" has 3 different meanings that are 1:04:00, 1:06:00, and 1:09:00. Ambiguous word "**bum**" has 3 different meanings that are 1:18:00, 1:18:02, and 1:08:03. The experiment uses RBFNetwork algorithm of the window size equal to ±4 and the selection technique GainRatioAttributeEval that the number of attributes is 40. The experimental result indicates that accuracy value by separating classifications into 2 classes of meaning has higher value than using the whole classes of meanings of ambiguous words that exist. The accuracy value of ambiguous word "**art**" when classify by separating into 2 classes of meaning are as follows. The first meaning (1:09:00) is 94.23%.The second meaning (1:04:00) is 90.38%. The third meaning (1:06:00) is 88.46%. The three meaning is 87.01%.



(a) Using ID3 algorithm

(b) Using RBFNetwork algorithm

Figure 10. Shows accuracy value when compare InfoGainAttributeEval with GainRatioAttributeEval technique.

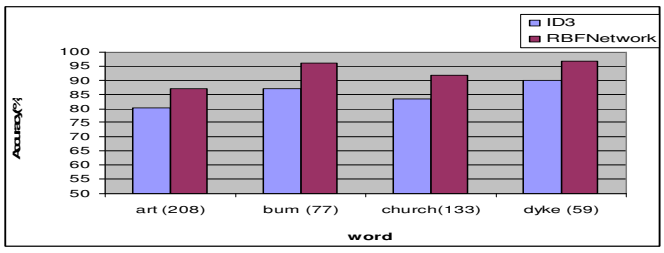


Figure 11. Show accuracy value when compare ID3 algorithm with RBFNetwork algorithm.

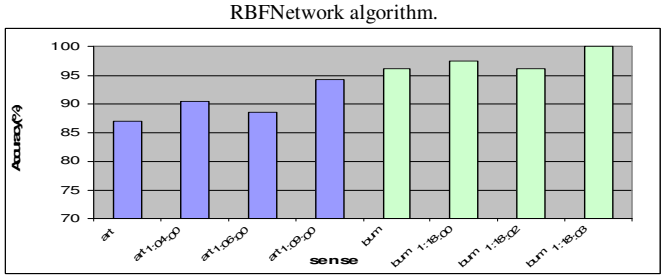


Figure 12.The accuracy of the classification by separating into 2 classes of meaning and the whole classes of meaning.

The detail for the experimental result of 8 ambiguous words is shown in Table I.

TABLE I. THE EXPERIMENTAL RESULT OF EACH AMBIGUOUS WORD BY USING RBFNETWORK ALGORITHM FOR BOTH LEFT-HAND AND RIGHT-HAND CONTEXT, AND GAINRATIOATTRIBUTEEVAL TECHNIQUE

Ambiguity Word	Number of class	Number of instance	Accuracy
art: meaning 1(1:09:00)	**2**	**208**	**94.23%**
art: meaning 2 (1:04:00)	2	208	90.38%
art: meaning 3 (1:06:00	2	208	88.46%
art (3 meanings)	3	208	87.01%
bum: meaning 1 (1:18:03)	**2**	**77**	**100%**
bum: meaning 2 (1:08:00)	2	77	94.40%
bum: meaning 3 (1:18:02)	2	77	96.10%
bum (3 meanings)	3	77	96.10%
bar: meaning 1(1:14:00)	**2**	**248**	**100%**
bar: meaning 2(1:06:06)	**2**	**248**	**100%**
bar: meaning 3(1:06:05)	2	248	98.79%
bar: meaning 4(1:10:00)	2	248	98.38%
bar: meaning 5(1:06:00)	2	248	95.16%
bar: meaning 6(1:06:04)	2	248	80.64%
bar (6 meanings)	6	248	59.27%
chair: meaning 1(1:18:00)	**2**	**139**	**100%**
chair: meaning 2 (1:06:00)	2	139	98.56%
chair: meaning 3 (1:04:00)	2	139	98.56%
chair (3 meanings)	3	139	95.68%
hearth: meaning 1(1:06:00)	**2**	**66**	**96.96%**
hearth: meaning 2 (1:15:00)	**2**	**66**	**96.96%**
hearth: meaning 3 (1:06:01)	2	66	92.42%
hearth: (3 meanings)	3	66	81.81%
stress: meaning 1(1:26:03)	**2**	**81**	**100%**
stress: meaning 2(1:26:02)	2	81	93.82%
stress: meaning 3(1:26:01)	2	81	88.88%
stress: (3 meanings)	3	81	72.83%
dyke	2	59	96.61%
church	2	133	91.66%

V. CONCLUSION

This paper presents steps for problem solving of ambiguous words by using context window and attributes selection. The result for the problem-solving of ambiguous words can be concluded that 1) using the proper window size of context effects on the accuracy of the classification, 2) using both left-hand side and right-hand side context window give higher accuracy than using only left-hand side or right-hand side context window, 3) attribute selection technique gives higher accuracy than the non-attribute selection technique 4) GainRatioAttributeEval selection gives higher accuracy than attribute selection of InfoGainAttributeEval selection, 5) the classification of meaning by separating into 2 classes of meaning gives higher accuracy than using whole classes of meanings, and 6) RBFNetwork algorithm gives higher accuracy than ID3 algorithm. In conclusion, this paper presents technique to enhance the prediction of ambiguous words with high accuracy value by using RBFNetwork algorithm, GainRatioAttributeEval attribute selection, and both left-hand side and right-hand side context window.

ACKNOWLEDGMENT

The author would like to express sincere thanks to Dr. Prawat Wettayaprasit for his advising on English and also to the Artificial Intelligence Research Laboratory at the Department of Computer Science, Prince of Songkla University, Thailand for this study.

REFERENCES

[1] L. Spacia. "A hybrid model for word sense disambiguation in English-Portuguese machine translation," *Int. Conf. on Computational linguistics*, 2005, pp. 71 -78.

[2] D. Dinh. "Building a training corpus for word sense disambiguation in English-to-Vietnamese machine translation," *Int. Conf. on Computational linguistics*, 2002.

[3] S. Wermter and C. Hung. "Selforganizing classification on the Reuters news corpus," *The 19th Int. Conf. on Computational linguistics*, 2002.

[4] T. O'Hara, R. Bruce, J. Donner and J. Wiebe. "Class- based collocations for word-sense disambiguation," in *Proc. Senseval 3 Workshop on evaluation of systems for the semantic analysis of text*, July 2004.

[5] G. Ckao and M.G. Dyer. "Maximum entropy models for word sense Disambiguation," in *Proc. The 19th int. conf. on Computational linguistics*, 2002.

[6] A. Suarez and M. Palomar. "Best feature selection for maximum entropy-based word sense disambiguation," in *Proc. The 6th Int. Conf. on applications of natural language to information systems-revised*, 2002, pp. 213-217.

[7] Y. Huang, P.J. McCullagh and N.D. Black. "Feature selection via supervised model construction," *The 4th IEEE Int. Conf. on Data Mining* , 2004, pp. 411-414.

[8] T. Ganchev, P. Zervas, N. Fakotakis and G. Kokkinakis. "Benchmarking feature selection techniques on the speaker verification task," *The 5th Int. Symposium on Communication systems, networks and digital signal processing*, July 2006, pp. 314-318.

[9] I.H. Witten and E. Frank. *Data Mining Pactical Machine Learning Tools and Technique*. San Francisco: Morgan Kaufman, 2005.

[10] A.G. Bor. "Introduction of the Radial Basis Function (RBFNetwork)." Internet:http://axiom.anu.edu.au/~daa/courses/GSAC6017/rbf.pdf, [Mar. 2, 2008].

[11] "Senseval-2." Internet: http://www.senseval.org, [Mar. 2, 2008].

[12] "SenseTools." Internet: http://www.d.umn.edu/~tpederse/sensetools.. html, [Mar. 2, 2008].

[13] B. Frakes and R. Baeza-Yates. *Information Retrieval Data structure &Algo- rithm*. New Jersey: Prentice Hall, 1992, pp. 113-115.

[14] K. Thongklin, S. Vanichayobon and W. Wettayaprasit. "Word sense disambiguation using stoplist removing for Senseval-2 corpus," in *Proc. The 3rd national conference on computing and information technology*, Bangkok, Thailand, May 2007, pp. 316-321.

[15] "NSP (Ngram Statistic Package)." Internet: http://search.cpan.org/dist/ Text-NSP, [Mar. 2, 2008].

[16] I.H. Witten and E. Frank. "WEKA (Waikato Environment for Knowledge Analysis)." Internet: http://www.cs.waikato.ac.nz/ml/weka, [Mar. 2, 2008].

ประวัติผู้เขียน

ชื่อ สกุล นางสาวกาญจนา ทองกลิ่น

รหัสประจำตัวนักศึกษา 4822005

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วท.บ. (วิทยาการคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2547

การตีพิมพ์เผยแพร่ผลงาน

1. Thongklin, K., Vanichayobon, S., and Wettayaprasit, W. 2007. Word Sense Disambiguation using Stoplist removing for Senseval-2 Corpus. In Proceedings the 3rd National Conference on Computing and Information Technology (NCCIT'07). Bangkok, Thailand, pp. 316-321.
2. Thongklin, K., Vanichayobon, S., and Wettayaprasit, W. 2008. Word Sense Disambiguation and Attribute Selection Using Gain Ratio and RBF Neural Network. In 2008 IEEE International Conference on Research, Innovation and Vision for the future in Computing and Communications Technologies (RIVF'08). Vietnam.