```
not(has_sep(A)).

research.project_page(A) :- not(has_depart(A)), link-to(B,C,A,D), has_univers(C),
has_cluster(C), has_human(C).

research.project_page(A) :- has_sequenc(A), has_fri(A), not(has_univers(A)).

research.project_page(A) :- has_pattern(A), has_layer(A).

research.project_page(A) :- has_transform(A), not(has_univers(A)), has_principl(A).

research.project_page(A) :- has_sun(A), has_homepag(A), has_wisc(A).
```

รูปที่ 2.26 กฎ research.project_page เมื่อใช้เว็บเพจมหาวิทยาลัยเท็กซัสเป็นข้อมูลทดสอบ

```
research.project_page(A) :- has_relat(A), link-to(B,C,A,D), has_anchor_group(B).

research.project_page(A) :- not(has_interest(A)), has_research(A), has_includ(A),
has_public(A), not(has_professor(A)).

research.project_page(A) :- not(has_home(A)), has_show(A), not(has_oct(A)).

research.project_page(A) :- link-to(B,C,A,D), has_anchor_laboratori(B).

research.project_page(A) :- not(has_comput(A)), has_public(A), link-to(B,C,A,D),
has_engin(C).

research.project_page(A) :- not(has_home(A)), link-to(B,C,A,D),
has_neighborhood__imag(B).

research.project_page(A) :- has_extens(A), not(has_univers(A)), not(has_class(A)).

research.project_page(A) :- not(has_home(A)), link-to(B,C,A,D),
has_neighborhood__access(B).

research.project_page(A) :- has_approach(A), not(has_interest(A)), has_optim(A).

research.project_page(A) :- has_group(A), not(has_page(A)), has_adapt(A).

research.project_page(A) :- has_robert(A), has_put(A).

research.project_page(A) :- has_peopl(A), has_trace(A).
```

รูปที่ 2.27 กฎ research.project_page เมื่อใช้เว็บเพจมหาวิทยาลัยวอชิงตันเป็นข้อมูลทดสอบ

```
research.project_page(A) :- link-to(B,C,A,D), link-to(E,A,F,G), has_anchor_group(B).

research.project_page(A) :- has_project(A), has_peopl(A), has_intern(A).

research.project_page(A) :- not(has_home(A)), has_perform(A), has_specif(A),
not(has_student(A)).

research.project_page(A) :- not(has_scienc(A)), has_research(A), not(has_interest(A)),
link-to(B,A,C,D), has_librari(C).

research.project_page(A) :- not(has_home(A)), has_user(A), has_includ(A),
not(has_web(A)).

research.project_page(A) :- has_relat(A), not(has_home(A)), has_proceed(A).

research.project_page(A) :- not(has_comput(A)), has_share(A).

research.project_page(A) :- has_robert(A), has_laboratori(A), not(has_inform(A)).

research.project_page(A) :- not(has_date(A)).

research.project_page(A) :- has_move(A), has_browser(A).
```

รูปที่ 2.28 กฎ research.project_page เมื่อใช้เว็บเพจมหาวิทยาลัยวิสคอนซินเป็นข้อมูลทดสอบ

ตารางที่ 2.34 ประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายและตัวแยกแยะแบบไอแอลพี

ประเภท	ตัวแยกแยะ	ความแม่นยำ (%)	ความระลึก (%)	มาตรการเอฟ (%)
นักเรียน	Naive Bayes	81.64	87.80	84.61
	ILP-Foil	79.44	74.20	76.73
อาจารย์	Naive Bayes	77.48	52.60	62.76
	ILP-Foil	60.50	69.75	64.54
รายวิชา	Naive Bayes	84.89	91.55	88.09
	ILP-Foil	75.31	88.61	81.42
โครงการ	Naive Bayes	51.62	63.87	57.10
	ILP-Foil	31.33	53.71	39.58

จากผลการทดลองโดยรวมแล้วตัวแยกแยะแบบเบย์อย่างง่ายจะมีประสิทธิภาพสูงกว่าตัวแยกแยะแบบไอแอลพี โดยมีเพียงประเภทอาจารย์เท่านั้น ที่ตัวแยกแยะแบบไอแอลพีมีประสิทธิภาพสูงกว่าตัวแยกแยะแบบเบย์อย่างง่ายเล็กน้อย ส่วนอีก 3 ประเภทที่เหลือ คือ ประเภทนักเรียน ประเภทรายวิชา และประเภทโครงการ ตัวแยกแยะแบบไอแอลพีมีประสิทธิภาพต่ำกว่าตัวแยกแยะแบบเบย์อย่างง่ายค่อนข้างมาก

จากผลการทดลองสรุปได้ว่า ในการจำแนกประเภทเว็บเพจ ตัวแยกแยะแบบไอแอลพีมีประสิทธิภาพต่ำกว่าตัวแยกแยะแบบเบย์อย่างง่าย ซึ่งน่าจะเป็นเพราะตัวแยกแยะแบบไอแอลพีจะสร้างกฎสำหรับเว็บเพจแต่ละประเภทจากข้อมูลฝึก (Training Data) ซึ่งกฎที่สร้างขึ้นสามารถอธิบายข้อมูลฝึกได้เป็นอย่างดี แต่เนื่องจากเว็บเพจของแต่ละมหาวิทยาลัยจะมีลักษณะเฉพาะที่แตกต่างกัน กฎที่สร้างขึ้นจึงอธิบายข้อมูลทดสอบ (Test Data) ได้ไม่ดีเท่าที่ควร เนื่องจากเว็บเพจเป็นเอกสารที่ไม่มีกฎเกณฑ์ที่ตายตัว เช่น ต้องมีคำใดปรากฏอยู่บ้าง

2.7 ประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายที่มีการถ่วงน้ำหนักคำสำคัญที่ได้จากกฎที่ได้จากตัวแยกแยะแบบไอแอลพี

เราสันนิษฐานว่าคำที่ปรากฏในกฎที่ได้จากตัวแยกแยะแบบไอแอลพีน่าจะเป็นคำสำคัญที่มีนัยสำคัญในการจำแนกประเภทเว็บเพจ จึงได้ทดลองนำคำจากเพรดิเคต has_word มากำหนดเป็นคำสำคัญและปรับเพิ่มน้ำหนักเป็น 5 เท่า แล้วลองทดสอบประสิทธิภาพ ปรากฏว่าได้ผลดังนี้

ตารางที่ 2.35 ประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายที่ใช้คำจากกฎที่ได้จากไอแอลพีเป็นคำสำคัญ

ประเภท	ตัวแยกแยะ	ความแม่นยำ (%)	ความระลึก (%)	มาตรการเอฟ (%)
นักเรียน	Naive Bayes (normal)	81.64	87.80	84.61
	Naive Bayes (ILP-word)	87.49	86.02	86.75
อาจารย์	Naive Bayes (normal)	77.48	52.60	62.76
	Naive Bayes (ILP-word)	72.57	57.93	64.43
รายวิชา	Naive Bayes (normal)	84.89	91.55	88.09
	Naive Bayes (ILP-word)	82.08	93.15	87.27
โครงการ	Naive Bayes (normal)	51.62	63.87	57.10
	Naive Bayes (ILP-word)	45.94	63.76	53.40

จากผลการทดลอง โดยรวมแล้วตัวแยกแยะแบบเบย์อย่างง่ายที่ใช้คำที่ปรากฏในกฎไอแอลพีเป็นคำสำคัญมีประสิทธิภาพใกล้เคียงกับตัวแยกแยะแบบเบย์อย่างง่ายปกติ ไม่ได้มีประสิทธิภาพดีขึ้นหรือแย่ลงอย่างมีนัยสำคัญ โดยในประเภทนักเรียนและประเภทอาจารย์ ตัวแยกแยะแบบเบย์อย่างง่ายที่ใช้คำสำคัญจากกฎไอแอลพีจะมีประสิทธิภาพสูงกว่าตัวแยกแยะแบบเบย์อย่างง่ายปกติเล็กน้อย ส่วนในประเภทรายวิชาและประเภทโครงการตัวแยกแยะแบบเบย์อย่างง่ายปกติจะมีประสิทธิภาพสูงกว่าตัวแยกแยะแบบเบย์อย่างง่ายที่ใช้คำสำคัญจากกฎไอแอลพีเล็กน้อย

จากผลการทดลองจึงสรุปได้ว่า สมมติฐานว่าคำที่ปรากฏในกฎที่ได้จากไอแอลพีเป็นคำสำคัญที่มีนัยสำคัญสูงในการจำแนกประเภทเว็บเพจ ไม่เป็นจริง

2.8 ประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายที่มีการถ่วงน้ำหนักคำสำคัญที่ได้จากผู้เชี่ยวชาญ

เมื่อใช้คำที่ได้จากกฎไอแอลพีแล้วไม่สามารถปรับปรุงประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายให้ดีขึ้นได้ เราจึงทดลองใช้คนวิเคราะห์ชุดข้อมูล แล้วดึงเอาคำสำคัญออกมา แล้วทดลองเพิ่มน้ำหนักคำสำคัญที่ดึงออกมาได้ โดยเพิ่มน้ำหนัก 5 เท่า ปรากฏว่าได้ผลการทดลองดังนี้

ตารางที่ 2.36 ประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายที่มีการถ่วงน้ำหนักคำสำคัญที่ได้จากผู้เชี่ยวชาญ

ประเภท	ตัวแยกแยะ	ความแม่นยำ (%)	ความระลึก (%)	มาตรการเอฟ (%)
นักเรียน	Naive Bayes (normal)	81.64	87.80	84.61
	Naive Bayes (weighted)	80.34	90.13	84.95
อาจารย์	Naive Bayes (normal)	77.48	52.60	62.76
	Naive Bayes (weighted)	74.46	58.87	65.75
รายวิชา	Naive Bayes (normal)	84.89	91.55	88.09
	Naive Bayes (weighted)	85.56	92.55	88.92
โครงการ	Naive Bayes (normal)	51.62	63.87	57.10
	Naive Bayes (weighted)	74.89	53.47	62.39

จากผลการทดลองพบว่า ตัวแยกแยะแบบเบย์อย่างง่ายที่มีการถ่วงน้ำหนักคำสำคัญที่ได้จากผู้เชี่ยวชาญมีประสิทธิภาพสูงกว่าตัวแยกแยะแบบเบย์ปกติเล็กน้อย โดยมีประสิทธิภาพสูงขึ้นทุกประเภท

สรุปจากผลการทดลองได้ว่า การปรับเพิ่มน้ำหนักคำสำคัญ ช่วยให้ประสิทธิภาพในการจำแนกประเภทของตัวแยกแยะแบบเบย์อย่างง่ายสูงขึ้น

ตารางที่ 2.37 ประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายที่ใช้คำสำคัญจากไอแอลพี และที่ใช้คำสำคัญจากผู้เชี่ยวชาญ

ประเภท	ตัวแยกแยะ	ความแม่นยำ (%)	ความระลึก (%)	มาตรการเอฟ (%)
นักเรียน	Naive Bayes (ILP-word)	87.49	86.02	86.75

	Naive Bayes (weighted)	80.34	90.13	84.95
อาจารย์	Naive Bayes (ILP-word)	72.57	57.93	64.43
	Naive Bayes (weighted)	74.46	58.87	65.75
รายวิชา	Naive Bayes (ILP-word)	82.08	93.15	87.27
	Naive Bayes (weighted)	85.56	92.55	88.92
โครงการ	Naive Bayes (ILP-word)	45.94	63.76	53.40
	Naive Bayes (weighted)	74.89	53.47	62.39

ตารางนี้เปรียบเทียบประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายที่ใช้คำสำคัญจากกฎไอแอลพี และที่ใช้คำสำคัญจากผู้เชี่ยวชาญ จะได้ว่าตัวแยกแยะที่ใช้คำสำคัญจากผู้เชี่ยวชาญมีประสิทธิภาพสูงกว่า โดยมีเพียงประเภทนักเรียนเท่านั้น ที่ตัวแยกแยะที่ใช้คำสำคัญจากผู้เชี่ยวชาญมีประสิทธิภาพต่ำกว่าตัวแยกแยะที่ใช้คำสำคัญจากกฎไอแอลพีเล็กน้อย ส่วนอีก 3 ประเภท จะมีประสิทธิภาพสูงกว่า โดยเฉพาะประเภทโครงการมีประสิทธิภาพสูงกว่ามาก

ต่อมาเราทดลองปรับค่าถ่วงน้ำหนักคำสำคัญที่ได้จากผู้เชี่ยวชาญเป็น 10 เท่า และ 20 เท่า เพื่อทดสอบว่าค่าถ่วงน้ำหนักมีผลต่อประสิทธิภาพอย่างไร ปรากฏว่าได้ผลการทดลองดังนี้

ตารางที่ 2.38 ประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายเมื่อเราเปลี่ยนแปลงค่าถ่วงน้ำหนัก

ประเภท	ตัวแยกแยะ	ความแม่นยำ (%)	ความระลึก (%)	มาตรการเอฟ (%)
นักเรียน	Bayes (weighted 5x)	80.34	90.13	84.95
	Bayes (weighted 10x)	77.85	93.30	84.88
	Bayes (weighted 20x)	76.39	93.87	84.23
อาจารย์	Bayes (weighted 5x)	74.46	58.87	65.75
	Bayes (weighted 10x)	73.20	59.03	65.36
	Bayes (weighted 20x)	72.98	59.79	65.73
รายวิชา	Bayes (weighted 5x)	85.56	92.55	88.92
	Bayes (weighted 10x)	86.88	91.58	89.17
	Bayes (weighted 20x)	86.04	91.50	88.69
โครงการ	Bayes (weighted 5x)	74.89	53.47	62.39
	Bayes (weighted 10x)	86.01	48.67	62.16
	Bayes (weighted 20x)	86.01	44.45	58.61

จากผลการทดลองประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายที่มีการถ่วงน้ำหนักคำสำคัญที่ได้จากผู้เชี่ยวชาญ เมื่อเปลี่ยนแปลงค่าถ่วงน้ำหนักจาก 5 เท่า เป็น 10 เท่า และ 20 เท่า ประสิทธิภาพจะใกล้เคียงกันมาก แทบจะไม่แตกต่างกันเลย สรุปได้ว่าการปรับเพิ่มค่าถ่วงน้ำหนักไม่ได้ส่งผลต่อประสิทธิภาพในการจำแนกมากนัก

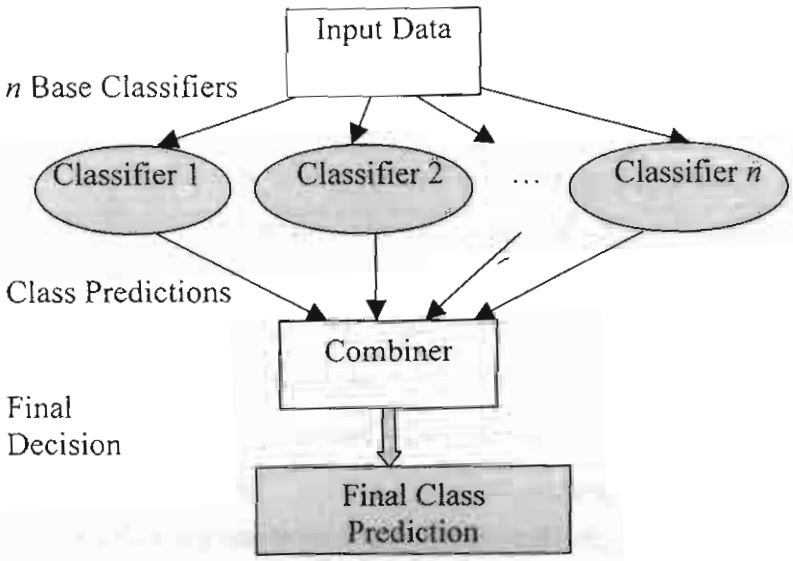
ตารางที่ 2.39 ประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายที่ใช้คำสำคัญจากไอแอลพี และที่ใช้คำสำคัญจากผู้เชี่ยวชาญซึ่งใช้ค่าถ่วงน้ำหนัก 5 เท่า, 10 เท่า และ 20 เท่า

ประเภท	ตัวแยกแยะ	ความแม่นยำ (%)	ความระลึก (%)	มาตรการเอฟ (%)
นักเรียน	Bayes (ILP-word)	87.49	86.02	86.75
	Bayes (weighted 5x)	80.34	90.13	84.95
	Bayes (weighted 10x)	77.85	93.30	84.88
	Bayes (weighted 20x)	76.39	93.87	84.23
อาจารย์	Bayes (ILP-word)	72.57	57.93	64.43
	Bayes (weighted 5x)	74.46	58.87	65.75
	Bayes (weighted 10x)	73.20	59.03	65.36
	Bayes (weighted 20x)	72.98	59.79	65.73
รายวิชา	Bayes (ILP-word)	82.08	93.15	87.27
	Bayes (weighted 5x)	85.56	92.55	88.92
	Bayes (weighted 10x)	86.88	91.58	89.17
	Bayes (weighted 20x)	86.04	91.50	88.69
โครงการ	Bayes (ILP-word)	45.94	63.76	53.40
	Bayes (weighted 5x)	74.89	53.47	62.39
	Bayes (weighted 10x)	86.01	48.67	62.16
	Bayes (weighted 20x)	86.01	44.45	58.61

ตารางนี้เปรียบเทียบประสิทธิภาพของตัวแยกแยะแบบเบย์อย่างง่ายที่ใช้คำสำคัญจากกฎไอแอลพี และที่ใช้คำสำคัญจากผู้เชี่ยวชาญซึ่งใช้ค่าถ่วงน้ำหนัก 5 เท่า, 10 เท่า และ 20 เท่า จะได้ว่าตัวแยกแยะที่ใช้คำสำคัญจากผู้เชี่ยวชาญมีประสิทธิภาพสูงกว่า โดยมีเพียงประเภทนักเรียนเท่านั้น ที่ตัวแยกแยะที่ใช้คำสำคัญจากผู้เชี่ยวชาญมีประสิทธิภาพต่ำกว่าตัวแยกแยะที่ใช้คำสำคัญจากกฎไอแอลพีเล็กน้อย ส่วนอีก 3 ประเภท จะมีประสิทธิภาพสูงกว่า โดยเฉพาะประเภทโครงการมีประสิทธิภาพสูงกว่ามาก

2.9 ประสิทธิภาพของตัวแยกแยะแบบเบย์ที่ทำงานร่วมกัน

จากการทดลองที่ผ่านมาพบว่าตัวแยกแยะแบบเบย์สมีความเสถียร ซึ่งยากต่อการปรับปรุงประสิทธิภาพโดยใช้หลักการของตัวแยกแยะที่ทำงานร่วมกัน แต่ตัวแยกแยะแบบเบย์สมีความเร็ว และความถูกต้อง และหลักการของตัวแยกแยะที่ทำงานร่วมกันโดยทั่วไปจะปรับปรุงประสิทธิภาพของตัวแยกแยะได้ด้วยการทำงานร่วมกันของตัวแยกแยะเดี่ยวที่มีความแตกต่างกัน เว็บเพจมีลักษณะเด่นที่สามารถนำมาสร้างตัวแยกแยะเดี่ยวที่มีความแตกต่างกันได้ ดังนั้นงานวิจัยนี้จึงทำการศึกษาการจำแนกเว็บเพจโดยใช้ลักษณะเด่น บนตัวแยกแยะแบบเบย์สอย่างง่ายที่ทำงานร่วมกัน



รูปที่ 2.29 ตัวแยกแยะแบบเบย์สอย่างง่ายที่ทำงานร่วมกัน

งานวิจัยด้านตัวแยกแยะที่ทำงานร่วมกัน เริ่มต้นในยุค 90 โดย Hensen และ Salamon (Hensen and Salamon, 1990) จากการสังเกตธรรมชาติของตัวแยกแยะแบบโครงข่ายประสาทเทียม ที่มีคุณสมบัติของความไม่เสถียร (Unstable) ซึ่งคุณสมบัตินี้เป็นผลดีต่อกระบวนการของตัวแยกแยะที่ทำงานร่วมกัน งานวิจัยด้านนี้จึงถูกศึกษาต่อมา เพื่อใช้ในด้านการรู้จำรูปแบบ (Pattern Recognition) โดยใช้ตัวแยกแยะเป็นการเรียนรู้แบบมีครู ถึงแม้ว่าปัจจุบันนี้จะมีการพัฒนาตัวแยกแยะที่มีประสิทธิภาพ อย่างเช่น ซัพพอร์ตเวกเตอร์แมชชีน และแม้ว่าการใช้ตัวแยกแยะที่ทำงานร่วมกันจะถูกมองในบางงานวิจัยว่าเป็นการใช้ตัวแยกแยะจำนวนมากอย่างฟุ่มเฟือย แต่งานวิจัยทางด้านนี้ยังได้รับการศึกษาอย่างต่อเนื่องและก้าวหน้าอย่างรวดเร็ว จนถึงปัจจุบันยังพบว่าตัวแยกแยะที่มีประสิทธิภาพเช่น ซัพพอร์ตเวกเตอร์แมชชีน ก็ได้รับประโยชน์จากงานวิจัยด้านตัวแยกแยะที่ทำงานร่วมกัน

การค้นพบที่สำคัญในหัวข้อของการผสมตัวแยกแยะมากกว่าหนึ่งตัว (Combining Classifiers) คือ สถาปัตยกรรมของตัวแยกแยะที่ทำงานร่วมกัน (Ensemble Architecture) ให้ความถูกต้องสูงกว่าการทำงานของตัวแยกแยะที่เป็นองค์ประกอบเพียงตัวเดียว การใช้ตัวแยกแยะที่ทำงานร่วมกันในการเรียนรู้ของเครื่องจักร ถูกมองว่าเป็นงานวิจัยที่ค่อนข้างใหม่ เกิดขึ้นเพียงแค่ช่วงสิบปีที่ผ่านมา แต่โดยความจริงแล้ว ความคิดนี้ปรากฏเป็นคำกล่าวในทฤษฎี Condorcent Jury มาก่อนหน้านี้แล้ว โดย Marquis Condocent เสนอทฤษฎีนี้ในปี 1784 และถูกอ้างอิงต่อมาโดย Nizan และ Paroush ในปี 1998 และ Christian List ในปี 2003 (Nizan and Paroush, 1998; List, 2003) มีข้อความโดยสรุปว่า "เมื่อความน่าจะเป็นที่แต่ละตัวแยกแยะ สามารถจำแนกได้ความถูกต้องมากกว่า 0.5 แสดงว่าผลการจำแนกของตัวแยกแยะส่วนใหญ่ ที่ตอบได้ถูกต้องนั้นมีมากกว่าครึ่ง และเมื่อตัว

แยกแยะมีจำนวนมากขึ้นเรื่อย ๆ ย่อมทำให้ผลการจำแนกส่วนใหญ่มีความน่าจะเป็นสูงขึ้นยิ่งเข้าใกล้ 1 เมื่อทุกความน่าจะเป็นที่ตัวแยกแยะจะจำแนกข้อมูลได้ถูกต้องมีมากกว่า 0.5"

ใน ค.ศ. 1990 สมมุติฐานที่ว่า "ตัวแยกแยะที่ทำงานร่วมกันส่วนใหญ่ให้ความแม่นยำมากกว่าตัวแยกแยะเดี่ยวที่เป็นองค์ประกอบ ก็ต่อเมื่อตัวแยกแยะที่เป็นองค์ประกอบเหล่านั้นมีความแม่นยำ (Accuracy) และมีความแตกต่างกัน (Diverse) อีกด้วย" ถูกแนะนำโดย Hensen และ Selamon (Hensen และ Salamon, 1990) ซึ่งพวกเขาพิสูจน์ว่าถ้าแต่ละตัวแยกแยะเดี่ยวที่เป็นองค์ประกอบเหล่านั้นไม่ขึ้นอยู่แก่กัน (Independence) และถ้าอัตราการผิดพลาด (Error Rates) ของพวกเขาน้อยกว่า 50% ของทั้งหมด แล้วอัตราการผิดพลาดของตัวแยกแยะที่ทำงานร่วมกันจะมีความผิดพลาดน้อยกว่า เมื่อเทียบกับความผิดพลาดที่เกิดจากตัวแยกแยะเดี่ยว และสองปัจจัยหลักที่สำคัญมากคือ ความแม่นยำของตัวแยกแยะ และความแตกต่างกันของตัวแยกแยะ ซึ่งถูกทดลองโดย Dietterich (Dietterich, 1997) สรุปได้ว่า "ถ้าอัตราการผิดพลาดของทุกตัวแยกแยะน้อยกว่า 50% (ซึ่งอัตราการผิดพลาดเหล่านั้นไม่เกี่ยวข้องกัน) แล้วความน่าจะเป็นที่ได้จากการลงคะแนนด้วยเสียงส่วนใหญ่ จะให้ผลออกมาเป็นค่าคงที่ซึ่งมากกว่าครึ่งหนึ่งของตัวแยกแยะที่ผิดพลาด"

เนื่องจากตัวแยกแยะทั่วไปจะมีประสิทธิภาพลดลงเมื่อจำแนกข้อมูลที่ยากขึ้น ดังนั้นจึงมีการคิดวิธีต่าง ๆ เพื่อเพิ่มประสิทธิภาพให้กับตัวแยกแยะข้อมูล ตัวแยกแยะที่ทำงานร่วมกันเป็นทางเลือกหนึ่งภายใต้แนวคิดที่ว่าตัวแยกแยะมากกว่าหนึ่งตัวทำงานร่วมกันจะให้ผลการจำแนกดีกว่าการจำแนกโดยตัวแยกแยะเดี่ยวที่เป็นสมาชิกของระบบ ซึ่งการเพิ่มประสิทธิภาพให้กับตัวแยกแยะที่ทำงานร่วมกัน ทำได้หลายแนวทาง เช่น การเลือกตัวแยกแยะที่เหมาะสม การเลือกวิธีลงคะแนนตัดสินคำตอบ และคัดเลือกตัวแทนเอกสารที่เหมาะสม เป็นต้น

2.9.1 ประเภทของวิธีการด้านตัวแยกแยะที่ทำงานร่วมกัน (Categorization Scheme of Classifier Ensemble) ตัวแยกแยะที่ทำงานร่วมกันถูกนำไปใช้และศึกษาวิจัยเพื่อปรับปรุงประสิทธิภาพแล้วหลายวิธี ขึ้นอยู่กับความเหมาะสมและข้อจำกัดของงาน ในที่นี้แบ่งเป็น 4 ประเภทตามงานวิจัยของ Shixin Yu (Shixin, 2003) เป็นหลัก ดังนี้

ก. ตัวแยกแยะที่ทำงานร่วมกันโดยจัดการข้อมูลฝึก (Classifier Ensemble Manipulation Training Sample)

ข. ตัวแยกแยะที่ทำงานร่วมกันกับลักษณะเด่นของชุดข้อมูลนำเข้าย่อยที่แตกต่างกัน (Classifier Ensemble with Difference Input Feature Subsets) คือพิจารณาลักษณะเด่นของชุดข้อมูลฝึกที่แตกต่างกัน เพื่อสร้างแต่ละตัวแยกแยะ

ค. ตัวแยกแยะที่ทำงานร่วมกันโดยมีชนิดของตัวแยกแยะที่แตกต่างกัน (Heterogeneous Classifier Ensemble) คือตัวแยกแยะแต่ละตัวมีขั้นตอนการฝึกและการจำแนกต่างกัน

ง. ตัวแยกแยะที่ทำงานร่วมกันโดยมีชนิดของตัวแยกแยะที่เหมือนกัน (Homogeneous Classifier Ensemble) คือแต่ละตัวแยกแยะมีวิธีการฝึกและการจำแนกเหมือนกัน

ตัวแยกแยะที่ทำงานร่วมกันโดยการจัดการข้อมูลฝึก ลักษณะการทำงานของตัวแยกแยะที่ทำงานร่วมกันประเภทนี้ต้องมี ขั้นตอนวิธีการเรียนรู้ (Learning Algorithms) ที่มีการทำงานตามของชุดคำสั่งโปรแกรมคอมพิวเตอร์ (Run) หลาย ๆ ครั้ง โดยที่แต่ละครั้งกระทำกับข้อมูลสำหรับฝึกที่แตกต่างกัน การทำงาน

ลักษณะนี้เหมาะกับการเรียนรู้ของขั้นตอนวิธีที่ไม่เสถียร (Unstable Learning Algorithm) หมายถึงขั้นตอนวิธีการเรียนรู้ที่ถูกทำให้เปลี่ยนแปลงการเรียนรู้ หรือผลลัพธ์ได้ เพียงแค่มีการเปลี่ยนแปลงเล็กน้อยในข้อมูลฝึก

วิธีการที่ประสบความสำเร็จและเป็นที่รู้จักมากที่สุด จนถือได้ว่าเป็นตัวแทนของการพัฒนาในด้านตัวแยกแยะที่ทำงานร่วมกันจนถึงปัจจุบันนี้ ก็คือวิธีการที่เรียกว่า แบ็กกิ้ง (Bagging) และ บูสต์ติง (Boosting)

วิธีการจำแนกข้อมูลด้วยการใช้แบ็กกิ้ง (Breiman, 1996) อาศัยชุดของการฝึกที่แตกต่างกันซึ่งถูกสร้างโดยการสุ่มและการแทนที่จากชุดข้อมูลดั้งเดิมสำหรับการฝึก (Bootstraps Replacement) และนำแต่ละชุดของการการฝึก ทำการฝึกเพื่อให้ได้ตัวแยกแยะต่าง ๆ สำหรับการจำแนกข้อมูล แต่ละตัวแยกแยะให้ผลลัพธ์ของแต่ละตัว ต่อมาจะนำผลลัพธ์เหล่านี้ไปผสมกันโดยวิธีการลงคะแนน อธิบายเป็นขั้นตอนวิธีดังนี้

กำหนดให้ L^T เป็นค่าเริ่มต้นข้อมูลฝึกตามขั้นตอนวิธีการเรียนรู้

N เป็นจำนวนครั้งของการทำงานตามของชุดคำสั่งโปรแกรมคอมพิวเตอร์

(1) วนซ้ำครั้งที่ i โดย $i = 1$ ถึง n

เพื่อหา $\overline{T}$ โดยที่ $\overline{T}$ หาได้จากวิธีการสุ่มข้อมูลฝึกจาก T โดยวิธี Bootstrapping

$h_i = L(\overline{T})$ คือ ขั้นตอนวิธีการเรียนรู้จากตัวอย่าง $\overline{T}$ เพื่อให้ได้ตัวแยกแยะประเภท h_i

(2) จากขั้นตอนแรกทำให้ได้ตัวแยกแยะ n ตัว เพื่อจำแนกข้อมูลทดสอบ x และแต่ละตัวแยกแยะให้ผลการจำแนกเป็นประเภท y นำทั้งหมดผ่านการลงคะแนน ประเภทใดมีผลการจำแนกสูงสุดก็ตอบการจำแนกประเภทตามนั้น ดังสมการ

$$h_f(x) = \arg\max_{y \in Y} \sum_i^N h_i(x) = y \qquad (14)$$

วิธีการจำแนกข้อมูลด้วยการใช้บูสต์ติง (Freund และ Schapire, 1996) จะทำการปรับน้ำหนักให้กับข้อมูลฝึก และสร้างตัวแยกแยะต่าง ๆ ขึ้นมาจากข้อมูลที่ถูกปรับน้ำหนักนั้น สุดท้ายอาศัยเทคนิคการลงคะแนนต่าง ๆ วิธีบูสต์ติง ถูกพิสูจน์ว่า แต่ละขั้นตอนของวิธีการเรียนรู้ที่อ่อนแอ (Weak Learning Algorithm) อาจจะถูกปรับปรุง (Boosted) ให้มีความเสถียรขึ้นได้

วิธีการจำแนกข้อมูลโดยการใช้บูสต์ติง ที่มีประสิทธิภาพเป็นที่รู้จักโดยทั่วไป เช่น เอด้าบูสต์ส (AdaBoots) สามารถอธิบายกระบวนการของวิธีการจำแนกข้อมูลโดยการใช้ เอด้าบูสต์ส ได้ดังนี้

(1) วนซ้ำครั้งที่ b, $(1, 2, 3, \ldots, b)$ เพื่อทำสิ่งต่อไปนี้

(1.1) สร้างตัวแยกแยะ $C^b(X^*)$ บนชุดข้อมูลฝึกที่ถูกให้น้ำหนัก

$$X^* = w_1^b x_1, w_2^b x_2, \ldots, w_n^b x_n \qquad (15)$$

ให้น้ำหนัก w_i^b โดย $i = 1, 2, 3, \ldots, n$ เป็นแต่ละข้อมูลฝึก (ซึ่ง $w_i^b = 1$ เมื่อ $b = 1$)

(1.2) นำชุดข้อมูลฝึกที่ถูกให้น้ำหนักในแต่ละรอบ b ให้กับตัวแยกแยะเพื่อเรียนรู้แล้วทดสอบว่าจำแนกชุดข้อมูลฝึกดังกล่าวได้ถูกต้องหรือไม่

โดย $\xi_i^b = 0$ เมื่อ x_i ถูกจำแนกได้อย่างถูกต้อง หรือเป็น 1 เมื่อเป็นกรณีอื่น

$$err_b = \frac{1}{n} \sum_{i=1}^{n} (w_i^b \xi_i^b) \qquad (16)$$

$$C_b = \frac{1}{2}\log(w_i^b \xi_i^b) \quad (17)$$

กำหนดน้ำหนักใหม่ให้กับรอบต่อไปจากสมการ

$$w_i^{b+1} = w_i^{b+1} \exp(-C_b \xi_i^b) \text{ โดยที่ } i = 1, 2, \ldots, n \quad (18)$$

และถูก Renormalization โดย $\sum_{i=1}^{n}(w_i^{b+1}) = n$ (19)

(2) จากขั้นตอนแรกในทุก ๆ รอบที่วนจะได้ชุดข้อมูลฝึกที่แตกต่างกันตามน้ำหนักที่ถูกกำหนดให้แต่ละชุดข้อมูลฝึก ทำให้ได้ตัวแยกแยะ Cb (X*) คือตัวแยกแยะ C ซึ่งเกิดจากการกระทำรอบที่ b ที่ได้จากชุดข้อมูลฝึก X^* ต่าง ๆ ซึ่งแต่ละตัวแยกแยะจะมีน้ำหนักของการทำนายเป็น C_b

(3) แต่ละตัวแยกแยะเรียนรู้ได้เป็นสมมติฐานของตัวแยกแยะนั้น หรือ h_t โดย t คือตัวแยกแยะใด ๆ จากนั้นนำผลการทำนายที่ได้ทั้งหมดมาผ่านกระบวนการลงคะแนนโดยใช้น้ำหนัก (Weight Voting) ซึ่งน้ำหนักของการลงคะแนนมาจากค่า C_b ดังสมการ

$$H_{final}(d) = \text{sgn}\{\sum_b C_b h_t(d)\} \quad (20)$$

การเพิ่มประสิทธิภาพให้กับตัวแยกแยะที่ทำงานร่วมกัน ทำได้หลายแนวทางดังตัวอย่างที่กล่าวไว้ตอนต้น แต่ปัจจัยที่สำคัญเช่นกันก็คือ วิธีการตัดสินเลือกผลลัพธ์สุดท้ายจากผลลัพธ์ทั้งหมดที่ได้จากทุกตัวแยกแยะในระบบ วิธีการซึ่งเป็นที่นิยมคืออาศัยเทคนิคการลงคะแนน ดังที่จะได้อธิบายในหัวข้อต่อไป

การเพิ่มประสิทธิภาพให้กับตัวแยกแยะที่ทำงานร่วมกัน ทำได้หลายแนวทางดังตัวอย่างที่กล่าวไว้ตอนต้น แต่ปัจจัยที่สำคัญเช่นกันก็คือ วิธีการตัดสินเลือกผลลัพธ์สุดท้ายจากผลลัพธ์ทั้งหมดที่ได้จากทุกตัวแยกแยะในระบบ วิธีการซึ่งเป็นที่นิยมคืออาศัยเทคนิคการลงคะแนน ดังที่จะได้อธิบายในหัวข้อต่อไป

2.9.2 เทคนิคการลงคะแนน (Voting Techniques) (Van Erp *et al.*, 2002) เทคนิคในการลงคะแนนเป็นวิธีการตัดสินผลการจำแนกข้อมูลจากทุก ๆ ตัวแยกแยะ เพื่อให้ได้ผลลัพธ์เดียวเป็นประเภทข้อมูลที่ถูกจำแนกได้จากระบบตัวแยกแยะที่ทำงานร่วมกัน ปัจจุบันนี้มีเทคนิคการลงคะแนนเกิดขึ้นมากมาย และมีชื่อเรียกวิธีการลงคะแนนที่แตกต่างกัน การจะนำเทคนิคใดไปใช้งานก็ขึ้นกับความเหมาะสมของข้อมูลหรือวิธีการทำงาน ในที่นี้จะยกตัวอย่างประเภทของการลงคะแนนตามงานวิจัยของ Merijn และคณะ เป็นหลักตาม 5 วิธีดังนี้

ก. วิธีการลงคะแนนแบบไม่ถูกให้น้ำหนัก (Unweighed Voting Methods) เป็นวิธีการลงคะแนนซึ่งให้น้ำหนักกับผลการจำแนกข้อมูลจากทุก ๆ ตัวแยกแยะเท่า ๆ กัน ตัวแยกแยะแต่ละตัวจะให้ผลการจำแนกเพียงค่าเดียวและถือเป็นหนึ่งคะแนน ตัวอย่างการลงคะแนนด้วยวิธีนี้คือ การลงคะแนนแบบจำนวนมาก (Plurality), การลงคะแนนแบบเสียงส่วนใหญ่ (Majority), การลงคะแนนแบบแก้ไข (Amendment Vote), การลงคะแนนแบบรันออฟ (Runoff vote) และ การลงคะแนนแบบคอนดอลเซท เคาท์ (Condorcet Count)

1) การลงคะแนนแบบจำนวนมาก เป็นรูปแบบง่ายที่สุดของการลงคะแนน ตัวแยกแยะแต่ละตัวจะให้ผลการจำแนกประเภทออกมา และถือเป็นหนึ่งคะแนน ประเภทใดมีคะแนนเสียงสูงที่สุด จะตอบผลการจำแนกตามนั้น โดยคะแนนเสียงที่สูงกว่านั้นอาจไม่ใช่เสียงส่วนใหญ่ก็ได้

2) การลงคะแนนแบบเสียงส่วนใหญ่ แตกต่างกับ การลงคะแนนแบบจำนวนมาก คือผลการจำแนกตอบตามคะแนนเสียงส่วนใหญ่ (ส่วนมากเกินครึ่ง)

3) การลงคะแนนแบบแก้ไข โดยนำคะแนนเสียงจากการทำนายผลในสองประเภทผ่านการลงคะแนนแบบเสียงส่วนใหญ่ก่อน หากประเภทใดมีคะแนนเสียงสูงกว่าจะถูกเลือกให้มาผ่านการลง

คะแนนแบบเสียงส่วนใหญ่ ร่วมกับคะแนนเสียงจากการจำแนกประเภทอื่น ๆ ถัดไป วิธีนี้มีข้อดีเมื่อมีประเภทใหม่ถูกเพิ่มเข้าไป แต่วิธีนี้ขาดความเป็นกลาง

4) การลงคะแนนแบบรันออฟ เป็นกระบวนการที่มี 2 ขั้นตอน ขั้นตอนแรกหาสองประเภทที่ได้คะแนนเสียงสูงที่สุดจากทุกประเภทที่ได้คะแนนเสียงจากทุกตัวแยกแยะ แล้วต่อมา ขั้นตอนที่สองใช้การลงคะแนนแบบเสียงส่วนใหญ่ ระหว่างสองประเภทที่ถูกตัดสินในขั้นตอนแรก โดยให้ทุกตัวแยกแยะประเภทจากเพียงแค่สองประเภทนี้อีกครั้งวิธีนี้ถูกพบว่าสามารถลดข้อผิดพลาดในการลงคะแนนแบบจำนวนมาก และการลงคะแนนแบบเสียงส่วนใหญ่ได้

5) การลงคะแนนแบบคอนดอลเซท เคาท์ โดยนำคะแนนเสียงที่ได้จากผลการจำแนกทุกตัวแยกแยะ ผ่านการเปรียบเทียบประเภทใด ๆ ทีละคู่ จนทุกประเภทถูกเปรียบเทียบกันหมด ผลการจำแนกสุดท้ายตอบตามประเภทที่ได้รับคะแนนเสียงสูงสุดจากทุกคู่ประเภทและสูงสุดของทั้งหมด วิธีนี้มีความซับซ้อนมากที่สุดเมื่อเทียบกับวิธีอื่น ๆ ใน วิธีการลงคะแนนแบบไม่ถูกให้น้ำหนัก แต่ก็สามารถลดข้อผิดพลาดที่เกิดขึ้นได้

ข. วิธีการลงคะแนนแบบถูกให้น้ำหนัก (Weighed Voting) แต่ละผลการจำแนกที่ได้จากทุกตัวแยกแยะ จะได้รับน้ำหนักที่ถูกทำให้เป็นมาตรฐานกลางแล้ว เพื่อให้เป็นสัดส่วนสอดคล้องกับจำนวนตัวแยกแยะที่เป็นองค์ประกอบ ผลการจำแนกสุดท้ายเชื่อถือตามผลการจำแนกที่ได้รับน้ำหนักสูงสุด

ค. วิธีการลงคะแนนแบบถูกให้น้ำหนักตามการลงคะแนนแบบเสียงส่วนใหญ่ (Weighted Majority Voting) คล้ายกับวิธีแรก แต่แตกต่างกันที่การกำหนดค่าน้ำหนัก โดยกำหนดค่าน้ำหนักเริ่มต้นเป็นหนึ่งในทุกตัวแยกแยะ เมื่อตัวแยกแยะให้ผลการจำแนกผิด น้ำหนักจะถูกทำให้ลดลงโดยคูณด้วยค่าสัมประสิทธิ์

ง. วิธีการลงคะแนนแบบค่าความเชื่อมั่น (Confidence Voting Methods) ตัวแยกแยะสามารถแสดงระดับของผลการจำแนกประเภทต่าง ๆ ได้ ประเภทที่มีค่าความเชื่อมั่น (Confidence) สูงกว่าหมายถึง ได้รับระดับคะแนนความถูกต้อง หรือความชอบจากตัวแยกแยะประเภทมากกว่า ตัวอย่างของคะแนนความเชื่อมั่นเช่น ความน่าจะเป็น (Probabilities) และระยะ (Distances) เป็นต้น สิ่งที่จำเป็นสำหรับวิธีการลงคะแนนเหล่านี้คือ ตัวแยกแยะต้องผลิตค่าความเชื่อมั่นในอัตราที่เหมาะสม และปัญหาคือ ข้อจำกัดของค่าความเชื่อมั่นควรเป็นจำนวนและสัดส่วนเท่าไร ตัวอย่างการลงคะแนนด้วยวิธีนี้คือ แพนดิโมเนียม (Pandemonium), หลักการบวก (Sum Rule) และ หลักการคูณ (Product Rule) เป็นต้น

1) แพนดิโมเนียม นั้นตัวแยกแยะทุกตัวจะมีคะแนนเป็นหนึ่งคะแนน สำหรับผลการจำแนกที่ได้จากตัวแยกแยะนั้น ๆ และทุกตัวแยกแยะต้องมีค่าความเชื่อมั่นให้กับผลการจำแนกด้วย ประเภทข้อมูลใดได้รับการลงคะแนนด้วยค่าความเชื่อมั่นสูงสุดของการลงคะแนนทั้งหมดจะถูกตัดสินผลการจำแนกตามประเภทนั้น วิธีนี้เป็นที่รู้จักครั้งแรกในตัวอย่างของการจำแนก โดยผู้เชี่ยวชาญ หรือตัวแทน (Separate Experts/ Agents) ในศาสตร์ของวิทยาการคอมพิวเตอร์ ซึ่งเป็นวิธีที่ง่าย เพื่อแสดงความแตกต่างของอันดับประเภทที่ถูกจำแนก และเลือกค่าความเชื่อมั่นสูงสุด นอกจากนี้ยังไม่จำกัดจำนวน หรือค่าของความเชื่อมั่นสำหรับตัวแยกแยะ ถือเป็นข้อจำกัดทำให้ยากต่อการเพิ่มระดับความถูกต้อง ตัวอย่างเช่น (ตัวแยกแยะ c_i, ให้ผลการจำแนก y_k, ที่มีความเชื่อมั่น

ใด ๆ) แล้วจะเลือกผลการจำแนกตามค่าสูงที่สุด ถ้า (c_1, y_1, 0.9), (c_2, y_2, 0.7) ผลของการลงคะแนนโดยใช้น้ำหนัก ก็จะเป็น y1 เนื่องจากค่าความเชื่อมั่นเป็น 0.9 ซึ่งมากกว่า 0.7

2) หลักการบวก โดยนำค่าความเชื่อมั่นของประเภทข้อมูลเดียวกันมาบวกกัน ประเภทใดมีผลรวมความเชื่อมั่นสูงจะจำแนกข้อมูลตามประเภทนั้น

3) หลักการคูณ ต่างกับหลักการบวก ที่ใช้ผลคูณของค่าความเชื่อมั่น ซึ่งวิธีนี้เป็นการทำลายโอกาสของประเภทที่มีค่าความเชื่อมั่นต่ำ

จ. วิธีการลงคะแนนแบบถูกจัดอันดับ (Ranked Voting Methods) ผลของการจำแนกทุกประเภทที่ได้รับจากตัวแยกแยะ จะถูกจัดอันดับตามค่าความเชื่อมั่น ค่าความถูกต้อง หรือค่าน้ำหนัก เป็นต้น ตัวอย่างการลงคะแนนด้วยวิธีนี้คือ บอร์ดาเคาท์ (Borda Count) ซึ่งมีหลักการคือตัวแยกแยะใด ๆ จะจัดอันดับผลการจำแนกประเภทอย่างสมบูรณ์และใช้อันดับของประเภทใด ๆ จากทุกตัวแยกแยะผ่านการรวมกัน หากประเภทใดให้ผลรวมของค่าการจัดอันดับสูงสุด จะจำแนกข้อมูลตามประเภทนั้น เช่น (ตัวแยกแยะ c_i, อันดับการจำแนก y_i, อันดับการจำแนก y1,) ถ้า (c1, 2 , 5), (c2, 3 , 1) แล้ว y1 มีคะแนน 2+3 , y2 มีคะแนน 5+1 ดังนั้นผลการจำแนกจะได้ประเภท y_2 เพราะมีคะแนนสูงกว่า เป็นต้น

ตารางที่ 2.40 จำนวนลักษณะเด่นในเอกสารประเภทต่าง ๆ ของชุดข้อมูล WebKB

ชื่อประเภท	จำนวนเอกสาร	จำนวนลักษณะเด่นเมื่อผ่านขั้นตอนเตรียมข้อมูลแล้ว							จำนวนเอกสารที่มีใน Content, Title+Meta, Header, Anchor
		Content	Meta	Title	Title +Meta	Header	Anchor	Title+Meta+ Header+Anchor	
1. Course	834	800	6	735	735	773	773	800	686
2. Faculty	1,020	987	2	786	786	874	901	983	649
3. Project	481	469	1	429	429	416	461	469	375
4. Student	1,485	1,432	9	1,191	1,194	1,219	1,370	1428	995
รวม	3,820	3,688	18	3,141	3,144	3,282	3,505	3,689	2,705
จำนวน Distinct Words ของชุดเอกสารที่มีใน Content, Title+Meta, Header, และ Anchor		3,505	-	-	345	1,164	2,509	2,990	-

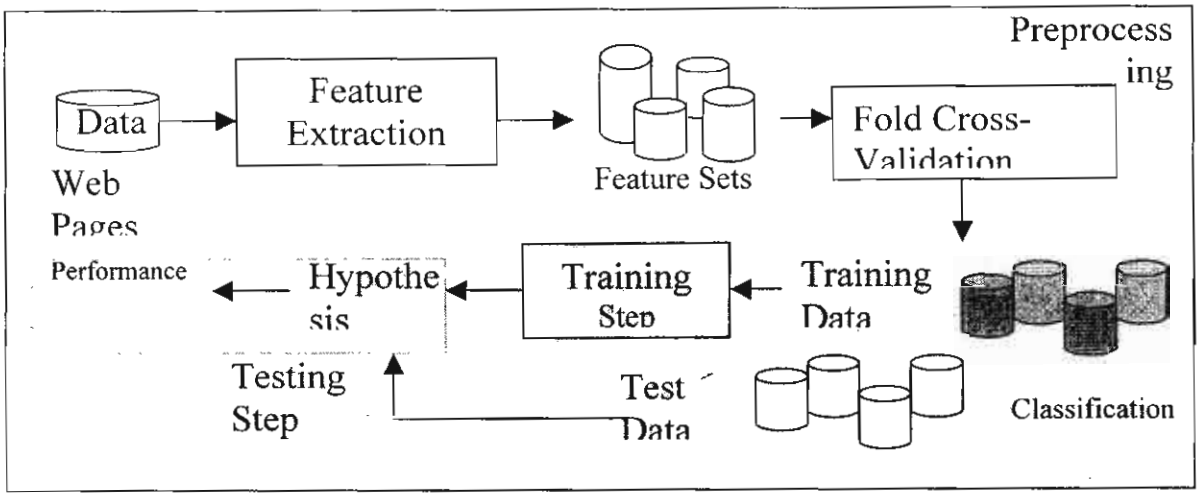
ตารางที่ 2 41 จำนวนลักษณะเด่นในเอกสารประเภทต่าง ๆ ของชุดข้อมูล WebPage

ชื่อประเภท	จำนวนเอกสาร	จำนวนลักษณะเด่นเมื่อผ่านขั้นตอนเตรียมข้อมูลแล้ว							จำนวนเอกสารที่มีใน Content, Title+Meta, Header, Anchor
		Content	Meta	Title	Title +Meta	Header	Anchor	Title+Meta+ Header+Anchor	
1. Commercial Banks	1,000	973	323	954	978	89	941	993	88
2 Building Societies	1,000	999	277	929	935	133	868	974	132
3. Insurance Agencies	1,000	999	378	953	966	170	906	998	145
4. Java	1,000	1,000	396	963	976	585	957	1,000	528
5 C / C++	1,000	999	386	905	924	658	953	962	648
6. Visual Basic	1,000	1,000	412	887	889	240	929	971	213
7. Astronomy	1,000	999	248	924	944	573	964	996	532
8. Biology	1,000	1,000	434	852	904	473	967	999	426
9. Soccer	1,000	1,000	464	927	944	221	825	978	199
10. Motor Sport	1,000	1,000	384	937	948	393	928	995	332

ตารางที่ 2.41 (ต่อ)

ชื่อประเภท	จำนวนเอกสาร	จำนวนลักษณะเด่นเมื่อผ่านขั้นตอนเตรียมข้อมูลแล้ว							จำนวนเอกสารที่มีใน Content, Title+Meta, Header, Anchor
		Content	Meta	Title	Title +Meta	Header	Anchor	Title+Meta+ Header+Anchor	
11. Sport	1,000	1,000	410	945	963	346	954	999	327
รวม	10,000	10,969	4,112	10,176	10,371	3,881	10,192	10,860	3,507
จำนวน Distinct Words ของชุดเอกสารที่มีใน Content, Title+Meta, Header, และ Anchor		17,758	-	-	1,649	2,262	7,637	8,376	-

ระบบการจำแนกเว็บเพจ แบ่งเป็น 2 ขั้นตอนหลัก คือขั้นตอนการเตรียมข้อมูล (Preprocessing Step) และขั้นตอนการจำแนกข้อมูล (Classification Step) ในขั้นตอนการเตรียมข้อมูลประกอบด้วยสองส่วน คือ ขั้นตอนการสกัดลักษณะเด่น (Feature Extraction Module) และส่วนการแบ่งข้อมูลโดยวิธีโฟลด-ครอส เฟลอิเดทฌัน สำหรับขั้นตอนการจำแนกเว็บเพจ ประกอบด้วยสองขั้นตอนย่อย คือ ขั้นตอนการฝึก (Training Step) และขั้นตอนการทดสอบ (Testing Step) ดังรูปที่ 2.30



รูปที่ 2.30 ระบบการจำแนกเว็บเพจ

ระบบเริ่มต้นทำงานจากการสกัดลักษณะเด่นของเว็บเพจ โดยจะคัดแยกส่วนของคำที่ต้องการออกจากเอกสาร เช่น คัดแยกเฉพาะส่วนของคำที่ปรากฏที่ชื่อเรื่อง (Title) คำที่ปรากฏที่หัวเรื่อง (Header) คำบรรยายลิงค์ (Anchor Description) เป็นต้น คำที่คัดแยกเรียบร้อยแล้วถูกเรียกว่า ชุดลักษณะเด่น (Feature Sets) ชุดลักษณะเด่นเหล่านี้จะถูกทำให้อยู่ในรูปแบบจำลองเวกเตอร์สเปซ ซึ่งมีสมาชิกที่ได้จากการคำนวณน้ำหนัก แบบทีเอฟไอดีเอฟ และทุกคำถูกทำให้เป็นตัวพิมพ์เล็กทั้งหมด ซึ่งคำที่ใช้เป็นคำที่ไม่ซ้ำกัน (Distinct Words) ที่ถูกกำจัดคำหยุด (Stop Words) ถูกทำให้อยู่ในรูปของรากศัพท์ (Stemming) โดย Porter Stemmer Algorithm (Porter, 1980) และถูกกำจัดคำที่ปรากฏน้อยกว่า i เอกสารออกแล้ว (i = 4 สำหรับชุดลักษณะเด่นเนื้อหาทั้งหมด สำหรับชุดลักษณะเด่นอื่นให้ i = 2) ในขั้นตอนการเตรียมข้อมูลนี้รวมไปถึง การกำจัดเอกสารว่าง หรือเอกสารที่ไม่มีคำใดถูกใช้ในการสร้างเวกเตอร์ออกด้วย

2.9.3 ลักษณะเด่นของเอกสาร

ขั้นตอนการสร้างชุดลักษณะเด่น อยู่ในขั้นตอนการสกัดลักษณะเด่น ซึ่งทำหน้าที่สกัดชุดลักษณะเด่น 5 ชุดดังนี้

(1) เนื้อหาทั้งหมด (Content Base) เป็นการเลือกใช้เนื้อหาทั้งหมดของหน้าเว็บเพจ หลังจากกรองกลุ่มตัวอักษรที่เป็นภาษาเอชทีเอ็มแอล (HTML Language) เรียบร้อยแล้ว

(2) ชื่อเรื่องและเมต้า เป็นการใช้กลุ่มตัวอักษรชื่อเรื่อง และเมต้า เนื่องจากการชื่อเรื่อง และเมต้า มักปรากฏในขอบเขตของแท็ก <head> และ </head> จึงนำมาใช้ร่วมกันเพื่อเพิ่มคำสำคัญของเอกสาร

ก. ชื่อเรื่อง (Title Tag) เตรียมโดยการคัดแยกเฉพาะกลุ่มตัวอักษรที่อยู่ในไวยากรณ์เอชทีเอ็มแอลชื่อเรื่อง <title> และ </title>

ข. เมต้า (Meta Tag) ที่เป็นการกำหนดคำสำคัญประจำเว็บเพจ เตรียมโดยการคัดแยกกลุ่มตัวอักษรที่อยู่ในไวยากรณ์เอชทีเอ็มแอลสองแบบคือ <meta name= "keywords"

content=" "> และ <meta name= "description" content=" "> ซึ่งใช้กลุ่มตัวอักษรใน content=" " เป็นลักษณะเด่นของเอกสารร่วมกับชื่อเรื่อง

(3) หัวเรื่อง (Header Tag) เตรียมโดยการคัดแยกเฉพาะกลุ่มตัวอักษรที่อยู่ในไวยากรณ์เอชทีเอ็มแอลหัวเรื่อง <Hi> และ </Hi> เมื่อ i = 1 ถึง 6 เป็นการเลือกเฉพาะหัวเรื่องในหน้าเว็บเพจเป็นลักษณะเด่นของเอกสาร

(4) คำบรรยายลิงค์ (Anchor Description) เตรียมโดยการคัดแยกเฉพาะกลุ่มตัวอักษรที่อยู่ในไวยากรณ์เอชทีเอ็มแอลคำบรรยายลิงค์สองแบบคือ แบบแรกใช้กลุ่มตัวอักษรใน
 และ แบบที่สองคือ <img src=" " alt=" "> และ ซึ่งเป็นการใช้กลุ่มตัวอักษรใน alt=" " เป็นลักษณะเด่นของเอกสาร

(5) ชื่อเรื่อง เมต้า หัวเรื่อง และคำบรรยายลิงค์ เป็นการใช้กลุ่มตัวอักษรดังกล่าวร่วมกันเป็นลักษณะเด่นของเอกสาร

```
<HTML>
<TITLE>Kaydon Bearings</TITLE>
<META name= "keyword" content="Slewing Rings, Worm Drive, Construction Equipment, Man lifts" >
<BODY>
<B>Kaydon bearings</B> are the designer's for difficult and sensitive applications in material
handing, construction equipment, robotic and other key industrial positions. Carefully engineered
solutions are available
<H1>Other Kaydon Divisions at yours service:/</H1><BR>
<A HREF= "/keydon/cooper.default.htm">Cooper Split Roller Bearings</A>|
</BODY>
</HTML>
```

รูปที่ 2.31 ตัวอย่างเว็บเพจที่มีลักษณะเด่นทุกแบบ

จากรูปที่ 2.31 เมื่อสกัดลักษณะเด่นแบบเนื้อหาทั้งหมดแล้วจะได้ข้อความดังนี้ "Kaydon Bearings Kaydon bearings are the designer's for difficult and sensitive applications in material handing, construction equipment, robotic and other key industrial positions. Carefully engineered solutions are available
Other Kaydon Divisions at yours service:/ Cooper Split Roller Bearings|"

เมื่อสกัดลักษณะเด่นแบบชื่อเรื่องและเมต้า แล้วจะได้ข้อความดังนี้ "Kaydon Bearings Slewing Rings, Worm Drive, Construction Equipment, Man lifts"

เมื่อสกัดลักษณะเด่นแบบหัวเรื่อง แล้วจะได้ข้อความดังนี้ "Other Kaydon Divisions at yours service:/"

เมื่อสกัดลักษณะเด่นแบบคำบรรยายลิงค์ แล้วจะได้ข้อความดังนี้ "Cooper Split Roller Bearings"

การใช้ลักษณะเด่นดังกล่าว เนื่องจากลักษณะเด่นเหล่านั้น สามารถให้คำสำคัญที่เหมาะสมสำหรับเป็นตัวแทนเอกสารที่ดีได้ มีจำนวนคำสำคัญที่ไม่น้อยเกินไป และข้อมูลของแต่ละชุดลักษณะเด่นมีความแตกต่างกันเพียงพอ คือไม่เกิดขึ้นซ้ำ ๆ กัน เช่นข้อมูลในลักษณะเด่นชื่อเรื่อง และเมต้า จะไม่เป็นข้อมูลชุดเดียวกันหรือคล้ายคลึงกันอย่างมากกับ ข้อมูลของชุดลักษณะเด่นหัวเรื่อง และชุดลักษณะเด่นคำบรรยายลิงค์ เพราะความแตกต่างกันเป็นคุณสมบัติที่ต้องมีในตัวแยกแยะเดี่ยวอันเป็นองค์ประกอบของตัวแยกแยะที่ทำงานร่วมกัน

ชุดข้อมูลเอกสารที่เตรียมได้ จะถูกจัดเก็บอยู่ในรูปของโครงสร้างแฟ้มเอกสาร และแฟ้มเอกสารที่ใช้ในการทดลองจะใช้แต่เอกสารที่มีในทั้ง 5 ชุดลักษณะเด่น เพื่อให้สะดวกต่อการประมวลผลในขั้นตอนต่อไป และเพื่อการเปรียบเทียบประสิทธิผลที่เหมาะสมกับทุกลักษณะเด่น

2.9.4 การจำแนกเว็บเพจ

หลังจากขั้นตอนการสกัดลักษณะเด่นของเอกสารเสร็จสิ้นลง ชุดข้อมูลเอกสารจะถูกนำมาแบ่งเป็นส่วน ๆ เพื่อใช้ในการฝึก และการทดสอบประสิทธิภาพ โดยแบ่งข้อมูลสำหรับฝึก และทดสอบ ในอัตราส่วน 80:20 ด้วยวิธี 5-โฟลด ครอส แฝลอิเดทฌัน เมื่อเรียบร้อยแล้วจึงนำมาใช้ในขั้นตอนการจำแนกเว็บเพจ โดยนำชุดข้อมูลฝึกผ่านขั้นตอนวิธีการจำแนกประเภทแบบต่าง ๆ ได้แก่ การจำแนกเว็บเพจด้วยตัวแยกแยะเดี่ยว และการจำแนกเว็บเพจด้วยตัวแยกแยะที่ทำงานร่วมกัน สำหรับขั้นตอนการทดสอบ ใช้ชุดข้อมูลทดสอบผ่านสมมุติฐานหรือตัวแยกแยะแบบเบย์สอย่างง่าย จากนั้นจึงวัดประสิทธิผล ในที่นี้ใช้ค่าความถูกต้อง ค่าความแม่นยำ ค่าความระลึก และ F1-measure เพื่อเปรียบเทียบประสิทธิผลของตัวแยกแยะเดี่ยวที่เกิดจากแหล่งตัวแทนเอกสารแตกต่างกัน และตัวแยกแยะที่ทำงานร่วมกันโดยใช้การลงคะแนนต่างกัน

ตารางที่ 2.42 ขั้นตอนวิธีการเรียนรู้ของตัวแยกแยะที่ทำงานร่วมกัน

Algorithm : learn_ensemble

Input : training set $x_{content}^{train}$, $x_{meta+title}^{train}$, x_{h}^{train} and x_{a}^{train}

integer s (number of base classifiers)

1. for j = 1 to s // for each feature in training set x

1.1. h[j] = naiveBayesLearner(x_j^{train}) //Train naive Bayes with x_j^{train}

2. end for s

Output: S base classifier is $h_0,\ldots,h_{s-1}$

ตารางที่ 2.43 ขั้นตอนวิธีการจำแนกเว็บเพจด้วยตัวแยกแยะที่ทำงานร่วมกัน

Algorithm : classify_ensemble

Input : test set $x_{content}^{test}$, $x_{meta+title}^{test}$, x_{h}^{test} and x_{a}^{test}

1. for j = 0 to s-1 // for each feature in test set x

/*Classifier, j, classify Web page d and keep its prediction in y[j] */

1.1 if the voting technique is majority vote

then y[j] = naiveBayesClassify(h[j], $x_{j_d}^{test}$)

else //voting technique is Borda Count

for k = 0 to l // l is number of class

rank_value[k][j] = naiveBayesClassify(h[j], $x_{j_d}^{test}$)

rank [k] = sum(rank_value[k][j])

end for k

ตารางที่ 2.43 (ต่อ)

Algorithm : classify_ensemble

end for s

/* Finalize the prediction */

2. if the voting technique is majority vote

then class = maximum(y[j])

else //voting technique is Borda Count

class = maximum(rank [k])

Output: Class of Web page d

2.9.5 ผลการวิจัยและวิจารณ์การจำแนกเว็บเพจด้วยตัวแยกแยะเดี่ยว

ตัวแยกแยะเดี่ยวสร้างจากชุดลักษณะเด่นของเว็บเพจที่แตกต่างกัน ชุดลักษณะเด่นที่สามารถนำมาเป็นตัวแทนเอกสารที่ดีได้ควรมีจำนวนข้อมูลที่เพียงพอต่อการฝึก และให้ผลการจำแนกที่ดีได้ ชุดลักษณะเด่นของเว็บเพจบางอย่างพบเห็นได้บ่อย เนื่องจากผู้สร้างเว็บเพจมักใช้อยู่เสมอ เช่น ชุดลักษณะเด่นชนิดคำบรรยายลิงค์ ที่เกือบทุกเว็บเพจจะมีการลิงค์เสมอ ชุดลักษณะเด่นชนิดเมต้ากับชื่อเรื่อง ก็มีปรากฏอยู่เสมอเนื่องจากผู้สร้างเว็บเพจใส่ไว้เป็นข้อมูลสำหรับเสิร์จเอนจินจะอ้างถึงเว็บเพจได้ สำหรับชุดลักษณะเด่นชนิดหัวเรื่องถือได้ว่าให้คำสำคัญที่เหมาะสมที่จะเป็นตัวแทนเอกสารที่ดีได้เช่นกัน แต่สำหรับชุดลักษณะเด่นที่เรียกว่าข้อมูลกำกับฟิลด์ (Tagged Field) ชนิดอื่น เช่น ตัวหนา ตัวเอียง ขีดเส้นใต้ เป็นต้น มักปรากฏในเอกสารจำนวนน้อย และมีคำในเอกสารที่น้อยหรือมีคำสำคัญที่ไม่ค่อยเหมาะที่จะเป็นตัวแทนเอกสารที่ดีได้ การทดลองนี้จึงพิจารณาชุด

ลักษณะเด่น 5 ชนิด คือ ชุดลักษณะเด่นชนิดเนื้อหาทั้งหมด ชุดลักษณะเด่นชนิดชื่อเรื่องและเมต้า ชุดลักษณะเด่นชนิดหัวเรื่อง และ ชุดลักษณะเด่นชนิดคำบรรยายลิงค์ ซึ่งชุดลักษณะเด่นแต่ละชนิด ให้ประสิทธิภาพที่แตกต่างกัน แสดงได้ดังตารางที่ 2.43 ซึ่งเป็นการจำแนกเว็บเพจบนชุดข้อมูล WebKB และตารางที่ 2.44 เป็นชุดข้อมูล WebPage

จากตารางที่ 2.43 และ 2.44 $S_{Content}$ หมายถึงชุดลักษณะเด่นชนิดเนื้อหาทั้งหมด $S_{Title+Meta}$ หมายถึงชุดลักษณะเด่นชนิดชื่อเรื่องและเมต้า S_H หมายถึงชุดลักษณะเด่นชนิดหัวเรื่อง S_A หมายถึงชุดลักษณะเด่นชนิดคำบรรยายลิงค์ และ $S_{Title+Meta+H+A}$ หมายถึงชุดลักษณะเด่นชนิดชื่อเรื่อง เมต้า หัวเรื่อง และคำบรรยายลิงค์

จากตารางที่ 2.44 และ 2.45 แสดงให้เห็นว่าการใช้เนื้อหาทั้งหมดเป็นชุดลักษณะเด่น ($S_{Content}$) ให้ผลการจำแนกเว็บเพจตามค่า F1-measure เป็น 81.1949% ในตารางที่ 2.44 และ 90.1844% ในตารางที่ 2.45 ซึ่งสูงกว่าการใช้ ชุดลักษณะเด่นชนิดอื่น เนื่องจากสามารถแทนเอกสารได้เนื้อหาที่ครอบคลุมมากที่สุด แต่การใช้เนื้อหาทั้งหมดเป็นชุดลักษณะเด่น ส่งผลให้คำที่ไม่เหมาะสมปะปนอยู่ด้วยซึ่งอาจทำให้ลดประสิทธิภาพในการจำแนกเว็บเพจลงได้ ดังตารางที่ 2.44 พบว่าชุดลักษณะเด่นชนิดชื่อเรื่อง เมต้า หัวเรื่อง และคำบรรยายลิงค์ ($S_{Title+Meta+H+A}$) ให้ผลการจำแนกดีกว่า $S_{Content}$ คือ 91.5762% และ 90.1844%

ตารางที่ 2.44 ผลการจำแนกเว็บเพจของตัวแยกแยะเดี่ยว บนชุดข้อมูล WebKB

Performance	SContent	STitle+Meta	SH	SA	STitle+Meta+H+A
Accuracy (%)	82.1852	73.3704	78.4815	74.0000	77.0741
Precision (%)	81.5786	76.6627	78.1869	73.2657	75.4841
Recall (%)	81.1945	72.2007	77.5427	73.2686	76.3692
F1-measure (%)	81.1949	73.6988	77.6813	72.9537	75.7053

ตารางที่ 2.45 ผลการจำแนกเว็บเพจของตัวแยกแยะเดี่ยว บนชุดข้อมูล WebPage

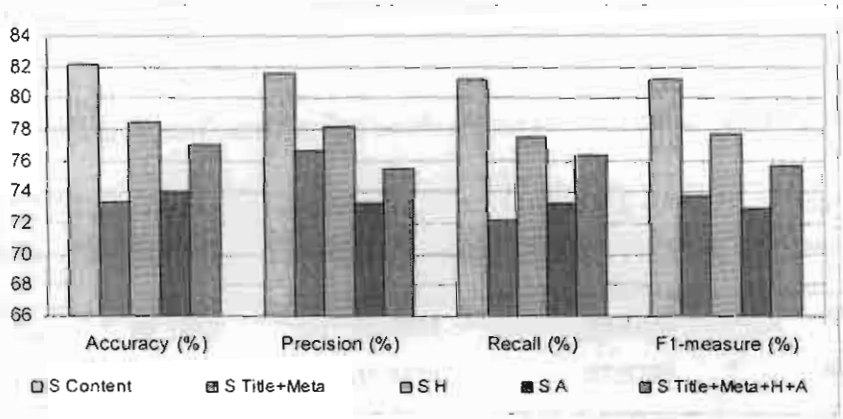
Performance	SContent	STitle+Meta	SH	SA	STitle+Meta+H+A
Accuracy (%)	92.3554	84.5134	77.3484	90.4372	93.8223
Precision (%)	90.0152	84.0737	73.5346	88.5244	91.3186
Recall (%)	90.9053	84.1084	74.4192	89.6310	92.6130
F1-measure (%)	90.1844	83.7148	73.5981	88.5406	91.5762

การใช้คำบรรยายลิงค์เป็นชุดลักษณะเด่นให้ F1-measure ต่ำกว่าการใช้เนื้อหาทั้งหมด ดังในตารางที่ 2.44 เป็น 72.9537% กับ 81.1949% ในตารางที่ 2.45 เป็น 88.5406% กับ 90.1844% เพราะคำบรรยายลิงค์ไม่สามารถแทนเอกสารที่ไม่มีลิงค์หรือมีคำบรรยายลิงค์เพียงเล็กน้อยได้ ทำให้ไม่ครอบคลุมเนื้อหา แต่สำหรับ

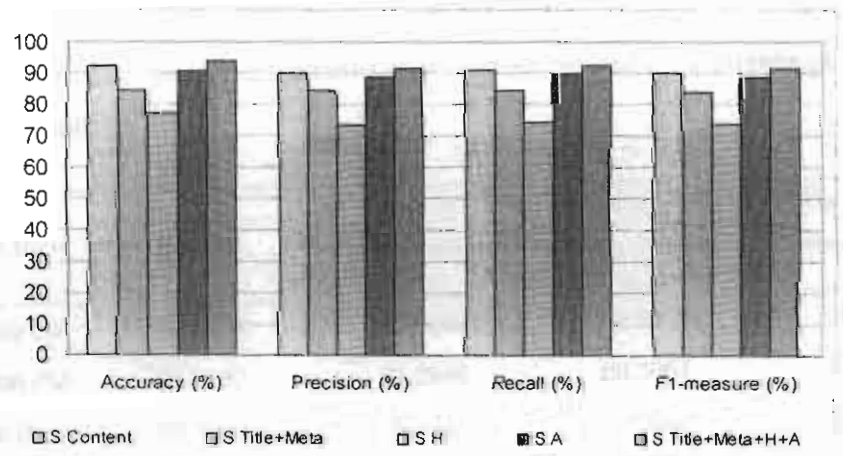
เว็บเพจที่มีลิงค์มากอาจทำให้คำสำคัญที่เหมาะสมสำหรับเป็นตัวแทนเอกสารถูกบดบังด้วยคำอื่น ๆ ที่ปรากฏเนื่องจากมีคำบรรยายลิงค์เป็นจำนวนมากที่ไม่เกี่ยวข้องกับเนื้อหาของเอกสารเลย คำเหล่านี้ทำให้ประสิทธิภาพในการจำแนกลดลง

ชุดลักษณะเด่นชนิดชื่อเรื่องและเมต้า กับ ชุดลักษณะเด่นชนิดหัวเรื่อง ถึงแม้จะให้ผลการจำแนกไม่ดีเท่าการใช้เนื้อหาทั้งหมดเป็นชุดลักษณะเด่น เพราะให้คำสำคัญจำนวนน้อยกว่า แต่พบว่าในตารางที่ 2.44 ชุดลักษณะเด่นชนิดหัวเรื่อง ให้ F1-measure เป็น 77.6813% ซึ่งสูงกว่าการใช้ชุดลักษณะเด่นชนิดคำบรรยายลิงค์ (72.9537%) และชุดลักษณะเด่นชนิดคำบรรยายลิงค์ยังให้ค่าความแม่นยำเป็น 73.2657% ซึ่งน้อยกว่าชุดลักษณะเด่นชนิดชื่อเรื่องและเมต้า (76.6627%) เนื่องจากสามารถลดคำที่ไม่จำเป็นได้จำนวนมาก

รายละเอียดที่ได้ดังตารางที่ 2.44 และ 2.45 ได้สรุปไว้เป็นกราฟภาพที่ 32 และ 33



รูปที่ 32 กราฟแสดงผลการจำแนกเว็บเพจของตัวแยกแยะเดี่ยว บนชุดข้อมูล WebKB



รูปที่ 33 กราฟแสดงผลการจำแนกเว็บเพจของตัวแยกแยะเดี่ยว บนชุดข้อมูล WebPage

เนื่องจากการใช้เนื้อหาทั้งหมดเป็นแหล่งตัวแทนเอกสาร เป็นวิธีที่นิยมใช้ในงานด้านการจำแนกเว็บเพจ และจากการทดลองพบว่าให้ผลการจำแนกเว็บเพจสูงกว่าแหล่งตัวแทนเอกสารอื่น ดังนั้นการทดลองต่อไปจึงใช้แหล่งตัวแทนเอกสารเนื้อหาทั้งหมดเป็นเกณฑ์ในการเปรียบเทียบ (Baseline) กับวิธีการจำแนกแบบอื่น ๆ ต่อไป

2.9.5 ผลการวิจัยและวิจารณ์การจำแนกเว็บเพจด้วยตัวแยกแยะที่ทำงานร่วมกัน

ตัวแยกแยะที่ทำงานร่วมกันนี้ สร้างจากตัวแยกแยะเดี่ยวหลาย ๆ ตัว ซึ่งตัวแยกแยะเดี่ยวแต่ละตัวสร้างจากชุดลักษณะเด่นที่ต่างชนิดกัน 5 ชนิด ดังนั้นตัวแยกแยะที่ทำงานร่วมกันนี้จึงประกอบด้วยตัวแยกแยะเดี่ยว 5 ตัว คือ $S_{Content}$ $S_{Title+Meta}$ S_H และ S_A ดังที่ได้อธิบายไว้แล้วในหัวข้อวิธีการ สำหรับในหัวข้อนี้จะแสดงถึงผลการจำแนกเว็บเพจโดยตัวแยกแยะที่ทำงานร่วมกัน และผลการจำแนกที่ใช้วิธีการตัดสินผลการจำแนกสุดท้ายที่แตกต่างกัน 2 วิธีคือ การลงคะแนนเสียงส่วนมาก ($E_{M\_V\ (Content,\ Title+Meta,\ H,\ A)}$) และ บอร์ดาเคาท์ ($E_{B\_C\ (Content,\ Title+Meta,\ H,\ A)}$) นอกจากนั้นยังแสดงให้เห็นถึงบอร์ดา เคาท์ ที่ถูกให้น้ำหนักกับอันดับที่สูงที่สุด โดยการคูณด้วยค่าสัมประสิทธิ์ ($\alpha = 2$) ($E_{B\_C\_Weight\ (Content,\ Title+Meta,\ H,\ A)}$) ซึ่งผลการทดลองบนชุดข้อมูล WebKB และ WebPage ได้แสดงในตารางที่ 2.46และ 2.47 ตามลำดับ

ตารางที่ 2.46 ประสิทธิภาพของตัวแยกแยะเดี่ยว และตัวแยกแยะที่ทำงานร่วมกัน บนชุดข้อมูล WebKB

Performance	$S_{Content}$	E_{M_V} (Content, Title+Meta, H, A)	E_{B_C} (Content, Title+Meta, H, A)	$E_{B\_C\_Weight}$ (Content, Title+Meta, H, A)
Accuracy (%)	82.1852	85.2963	86.2222	86.2222
Precision (%)	81.5786	85.3574	87.8798	87.2153
Recall (%)	81.1945	84.7564	86.3742	86.2814
F1-measure (%)	81.1949	84.9382	86.7959	86.5911

ตารางที่ 2.47 ประสิทธิภาพของตัวแยกแยะเดี่ยว และตัวแยกแยะที่ทำงานร่วมกัน บนชุดข้อมูล WebPage

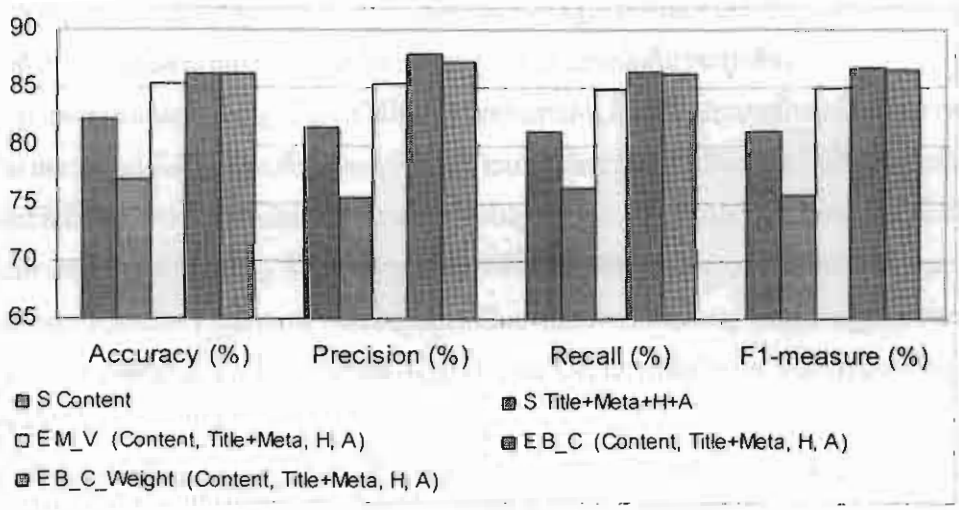
Performance	$S_{Content}$	E_{M_V} (Content, Title+Meta, H, A)	E_{B_C} (Content, Title+Meta, H, A)	$E_{B\_C\_Weight}$ (Content, Title+Meta, H, A)
Accuracy (%)	92.3554	93.7659	92.5529	95.0917
Precision (%)	90.0152	91.2946	89.3867	92.2931
Recall (%)	90.9053	92.0344	89.9943	93.0925
F1-measure (%)	90.1844	91.4155	89.4381	92.5254

จากตารางที่ 2.46 และ 2.47 $S_{Content}$ หมายถึงตัวแยกแยะเดี่ยวที่ใช้ชุดลักษณะเด่นชนิดเนื้อหาทั้งหมด $E_{M\_V\ (Content,\ Title+Meta,\ H,\ A)}$ หมายถึง ตัวแยกแยะที่ทำงานร่วมกันที่ใช้การลงคะแนนแบบเสียงส่วนมาก $E_{B\_C\ (Content,\ Title+Meta,\ H,\ A)}$ หมายถึงตัวแยกแยะที่ทำงานร่วมกันที่ใช้การลงคะแนนแบบบอร์ดา เคาท์ และ

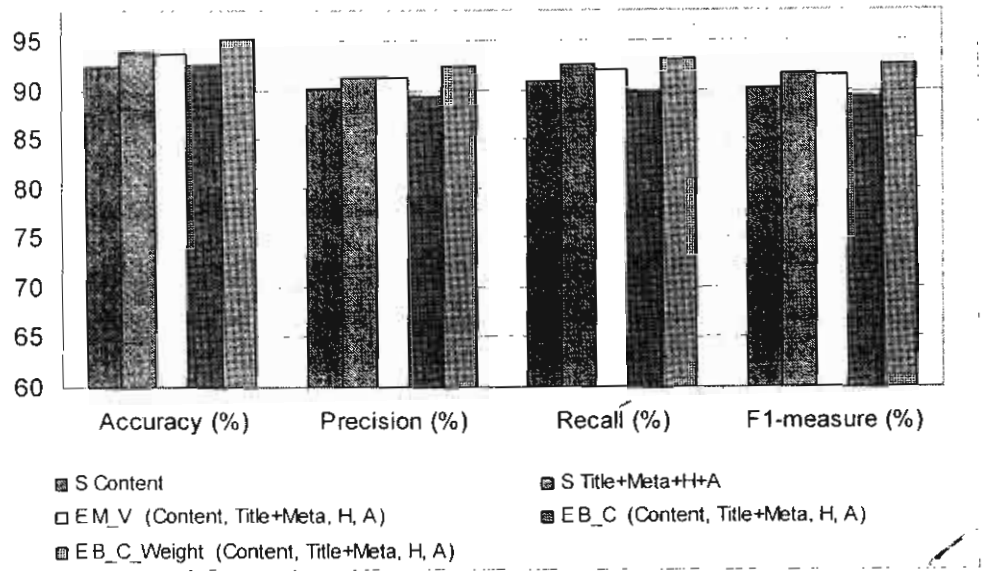
$E_{B_C_Weight\ (Content,\ Title+Meta,\ H,\ A)}$หมายถึง ตัวแยกแยะที่ทำงานร่วมกันที่ใช้การลงคะแนนตัดสินแบบบอร์ดา เคาท์ ที่ให้น้ำหนักจาก $\alpha = 2$

จากผลการทดลองในตารางที่ 2.46 และตารางที่ 2.47 พบว่าตัวแยกแยะที่ทำงานร่วมกันทั้งสามแบบ (E_{M_V}, E_{B_C} และ $E_{B\_C\_Weight}$) ให้ค่าความถูกต้องเป็น 85.2963% 86.2222% และ 86.2222% ในตารางที่ 2.45 และเป็น 93.7659% 92.5529% และ 95.0917% ในตารางที่ 2.46 ซึ่งสูงกว่า $S_{Content}$ ดังในตารางที่ 2.46 และ 2.47 เป็น 82.1852% และ 92.3554% ตามลำดับ เนื่องจากตัวแยกแยะที่เป็นองค์ประกอบของตัวแยกแยะที่ทำงานร่วมกัน ให้ผลที่ผิดพลาดในส่วนที่แตกต่างกัน ทำให้ใช้ผลที่ถูกต้องของแต่ละตัวแยกแยะเพื่อชดเชยส่วนที่ผิดพลาดซึ่งกันและกันได้

สำหรับ E_{B_C} ที่ให้ความถูกต้องสูงกว่า E_{M_V} เนื่องจาก E_{B_C} ให้โอกาสกับทุกประเภท โดยพิจารณาการให้อันดับผลการจำแนกทุกประเภทข้อมูลที่ทำโดยแต่ละตัวแยกแยะ และใช้ค่าอันดับเหล่านั้นเพื่อตัดสินผลการจำแนกสุดท้าย ดังในตารางที่ 2.46 เป็น 86.2222% กับ 85.2963% และสำหรับ $E_{B\_C\_Weight}$ ให้ผลการจำแนกสูงกว่าวิธีอื่นในทั้งสองชุดข้อมูล (86.2222% และ 95.0917%) เนื่องจากให้น้ำหนักกับอันดับสูงที่สุด เพราะอันดับสูงที่สุด มักจะเป็นประเภทที่ถูกต้องเสมอ จึงเป็นการให้โอกาสกับคำตอบที่คาดว่าจะถูกต้องที่สุด แต่ก็ไม่ปิดโอกาสประเภทอื่นที่มีอันดับรองลงมาเช่นกัน รายละเอียดที่ได้ดังตารางที่ 16 และ18 ได้สรุปไว้เป็นกราฟรูปที่ 32



รูปที่ 32 กราฟแสดงผลการจำแนกเว็บเพจของตัวแยกแยะเดี่ยว เปรียบเทียบกับตัวแยกแยะที่ทำงานร่วมกัน บนชุดข้อมูล WebKB



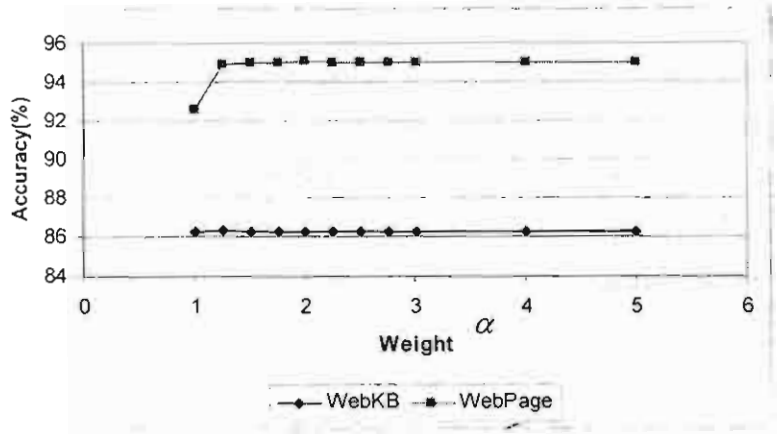
รูปที่ 33 กราฟแสดงผลการจำแนกเว็บเพจของตัวแยกแยะเดี่ยว เปรียบเทียบกับตัวแยกแยะที่ทำงานร่วมกัน บนชุดข้อมูล WebPage

จากรูปที่ 32 และ 33 พบว่า E_{M_V} ให้ผลความถูกต้องสูงกว่าตัวแยกแยะเดี่ยว ในทั้งสองชุดข้อมูล เนื่องจากถ้าตัวแยกแยะส่วนใหญ่ลงคะแนนให้กับประเภทใด ประเภทนั้นก็น่าจะถูกต้อง

การลงคะแนนแบบ E_{B_C} เป็นการให้โอกาสกับทุกประเภท ถึงแม้ว่าประเภทนั้นจะมีความน่าจะเป็นต่ำกว่าก็ตาม เพราะเป็นไปได้ที่คำตอบที่ถูกต้องอาจไม่ใช่ประเภทที่มีความน่าจะเป็นสูงสุด วิธีนี้จึงเหมาะกับระบบที่ตัวแยกแยะที่เป็นองค์ประกอบส่วนใหญ่ ให้ความถูกต้องไม่สูงมากนัก แสดงว่าประเภทที่มีความน่าจะเป็นสูงที่สุดอาจไม่คำตอบที่ถูก ดังนั้น E_{B_C} จึงให้ผลการจำแนกที่ดีขึ้นในชุดข้อมูล WebKB แต่ให้ผลการจำแนกลดลงในชุดข้อมูล WebPage เนื่องจากมีการสร้างตัวแยกแยะที่เป็นองค์ประกอบส่วนใหญ่ ที่ให้ความถูกต้องค่อนข้างสูงอยู่แล้ว เมื่อให้โอกาสกับประเภทอื่นที่มีความน่าจะเป็นต่ำกว่า จึงเป็นการเปิดโอกาสให้ได้ประเภทที่ไม่ถูกต้องเป็นคำตอบ

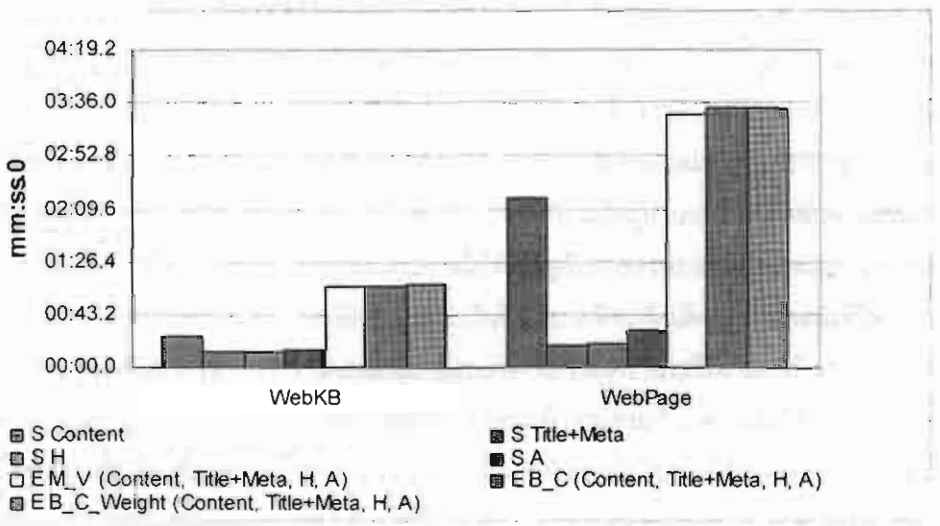
การลงคะแนนแบบ $E_{B\_C\_Weight}$ เป็นการเพิ่มโอกาสให้กับประเภทที่มีความน่าจะเป็นสูงที่สุด สำหรับชุดข้อมูล WebKB ที่ตัวแยกแยะที่เป็นองค์ประกอบส่วนใหญ่ ให้ความถูกต้องน้อย แสดงว่าประเภทที่มีความน่าจะเป็นสูงที่สุดอาจไม่ใช้คำตอบที่ถูกต้องก็ได้ หากเพิ่มโอกาสให้กับประเภทนั้นก็จะยิ่งเป็นการเปิดโอกาสให้ได้ประเภทที่ไม่ถูกต้องเป็นคำตอบ แต่วิธีนี้เป็นผลดีกับชุดข้อมูล WebPage เพราะตัวแยกแยะที่เป็นองค์ประกอบส่วนใหญ่ ให้ความถูกต้องสูงอยู่แล้ว เมื่อยิ่งเพิ่มโอกาสกับประเภทที่มีความน่าเป็นสูงที่สุด ก็ยิ่งเป็นการเปิดโอกาสให้ได้ประเภทที่ถูกต้องมากขึ้น

จากตารางที่ 2.46 และ 2.47 แสดงให้เห็นว่าบอร์ดา เคาท์ ที่ถูกให้น้ำหนักกับอันดับที่สูงที่สุด โดยการคูณด้วยค่าสัมประสิทธิ์ ($\alpha = 2$) ให้ผลการจำแนกที่ดีกว่าตัวแยกแยะเดี่ยว และสำหรับค่าสัมประสิทธิ์อื่น ได้แสดงไว้เป็นกราฟรูปที่ 34



รูปที่ 34 กราฟแสดงผลความถูกต้องของวิธี บอร์ดา เคาท์ ที่ถูกให้น้ำหนักกับอันดับที่สูงที่สุด โดยการคูณด้วยค่าสัมประสิทธิ์

ตัวแยกแยะที่ทำงานร่วมกันให้ผลการจำแนกดีกว่าตัวแยกแยะเดี่ยว แต่อย่างไรก็ตามตัวแยกแยะที่ทำงานร่วมกันก็ใช้เวลาในการประมวลผลนานกว่าตัวแยกแยะเดี่ยว เวลาในการประมวลผล(นาที) ของตัวแยกแยะที่ทำงานร่วมกันได้แสดงไว้เป็นกราฟรูปที่ 35



รูปที่ 35 กราฟแสดงเวลาในการประมวลผล

2.9.5 สรุป

การจำแนกเว็บเพจโดยใช้ ชุดลักษณะเด่นชนิดเนื้อหาทั้งหมด เป็นวิธีดั้งเดิมที่สะดวกและให้ผลการจำแนกที่ดี สำหรับตัวแยกแยะที่ทำงานร่วมกันช่วยเพิ่มความถูกต้องให้กับระบบได้ เมื่อตัวแยกแยะที่เป็นองค์ประกอบให้ส่วนที่ผิดพลาดแตกต่างกันเพื่อนำส่วนที่ถูกต้องมาชดเชยส่วนที่ผิดพลาดซึ่งกันและกันได้ ดังนั้นแต่ละตัวแยกแยะควรมีความแตกต่างกัน วิธีการสร้างตัวแยกแยะที่แตกต่างกันสำหรับงานวิจัยนี้ใช้ชุดลักษณะเด่น

ต่างชนิดกัน ชุดลักษณะเด่นแต่ละชนิดมีข้อดีและข้อเสียที่ต่างกัน การเลือกใช้ชุดลักษณะเด่นชนิดเนื้อหาทั้งหมด ชุดลักษณะเด่นชนิดชื่อเรื่องและเมต้า ชุดลักษณะเด่นชนิดหัวเรื่อง และการใช้ชุดลักษณะเด่นชนิดคำบรรยายลิงค์ เนื่องจากให้ความถูกต้องที่ไม่ต่ำจนเกินไป เป็นชุดลักษณะเด่นที่มีในเว็บเพจส่วนใหญ่ และเป็นชุดลักษณะเด่นที่ให้คำสำคัญที่ดีพอที่จะเป็นตัวแทนเอกสารที่ดีได้ และข้อสำคัญที่สุดคือ ชุดลักษณะเด่นแต่ละชนิดนำมาสร้างตัวแยกแยะที่มีความแตกต่างกันได้ และจากการทดลองแสดงให้เห็นถึงการใช้งานร่วมกันของชุดลักษณะเด่นแต่ละชนิด ในตัวแยกแยะที่ทำงานร่วมกัน แล้วให้ผลการจำแนกสูงกว่าตัวแยกแยะเดี่ยวที่เป็นองค์ประกอบของระบบตัวแยกแยะที่ทำงานร่วมกัน

ตัวแยกแยะแบบเบย์สมีความเสถียร ซึ่งยากต่อการปรับปรุงประสิทธิภาพโดยใช้หลักการของตัวแยกแยะที่ทำงานร่วมกัน เช่นวิธีการบูสต์ติงโดยตรง ไม่สามารถปรับปรุงประสิทธิภาพตัวแยกแยะแบบเบย์สได้ งานวิจัยนี้แสดงให้เห็นถึงวิธีการที่สามารถปรับปรุงประสิทธิภาพตัวแยกแยะแบบเบย์ส จากหลักการของตัวแยกแยะที่ทำงานร่วมกันได้ โดยใช้ลักษณะเด่นของเว็บเพจ เพื่อนำมาสร้างตัวแยกแยะเดี่ยวที่มีความแตกต่างกันได้ เพราะหลักการของตัวแยกแยะที่ทำงานร่วมกันโดยทั่วไปจะปรับปรุงประสิทธิภาพของตัวแยกแยะได้ด้วยการทำงานร่วมกันของตัวแยกแยะเดี่ยวที่มีความแตกต่างกัน

วิธีการดังกล่าวเหมาะสมกับชุดข้อมูลทดลองที่สามารถสกัดลักษณะเด่นได้ครบทุกชนิด ซึ่งถ้าลักษณะเด่นชนิดใดมีจำนวนน้อย หรือไม่ปรากฏในเว็บเพจใดก็จะไม่เหมาะสมที่จะนำวิธีการนี้ไปใช้จำแนกเอกสารดังกล่าว นอกจากนี้ถึงแม้ว่าผลการทดลองจะแสดงให้เห็นว่าตัวแยกแยะที่ทำงานร่วมกันให้ผลดีกว่าตัวแยกแยะเดี่ยว แต่ตัวแยกแยะที่ทำงานร่วมกันมีความซับซ้อนกว่า ใช้เวลาในการประมวลผลนานกว่าการจำแนกเว็บเพจด้วยตัวแยกแยะเดี่ยว และวิธีการนี้ยังต้องใช้เวลาในขั้นตอนการเตรียมข้อมูล และการสกัดลักษณะเด่นก่อนการประมวลผล รวมถึงวิธีการใช้ตัวแยกแยะที่ทำงานร่วมกันนี้ยังขึ้นอยู่กับการลงคะแนนตัดสินผลการจำแนกประเภทที่ต้องเลือกให้เหมาะกับตัวแยกแยะอีกด้วย และเป็นไปได้ยากที่จะคัดเลือกตัวแยกแยะที่ให้ความแตกต่างกันอย่างเหมาะสมกับการใช้วิธีการจำแนกแบบเบย์สอย่างง่าย ซึ่งมีความเสถียร และเป็นการยากที่จะกำหนดจำนวนตัวแยกแยะที่เพียงพอ หรือเหมาะสมต่อระบบการทำงานร่วมกัน เพราะยิ่งใช้จำนวนตัวแยกแยะมากขึ้น หรือซับซ้อนขึ้นก็จะใช้เวลาในการประมวลผลนานขึ้นเช่นกัน ดังนั้นในการวิจัยนี้จึงไม่ได้เปลี่ยนแปลงระบบการทำงานของตัวแยกแยะแบบเบย์สอย่างง่ายให้ซับซ้อนมากขึ้น แต่ทำการจำแนกโดยใช้ตัวแยกแยะแบบเบย์สมากกว่า 1 ตัวเพื่อชดเชยข้อผิดพลาดซึ่งกันและกันเท่านั้น

3. **ผลลัพธ์ที่ได้จากโครงการ**

3.1 งานวิจัยในช่วงปีแรกได้รับการตอบรับให้นำเสนอในที่ประชุมวิชาการนานาชาติ The International Conference on Computational Intelligence ซึ่งจัดขึ้นเมื่อวันที่ 17-19 ธันวาคม 2547 ณ เมือง Istanbul ประเทศตุรกี

ชื่อบทความ Combining ILP with semi-supervised learning for Web page categorization

หน้า 322-325

3.2 งานวิจัยในช่วงปีที่สองได้รับการตอบรับให้นำเสนอในที่ประชุมวิชาการนานาชาติ The International Conference on Intelligent Computing ซึ่งจะจัดขึ้นในวันที่ 23-26 สิงหาคม 2548 ณ เมือง Hefei ประเทศจีน

ชื่อบทความ Ensemble classifiers using different feature sets for Web page categorization

3.3 ผลงานวิจัย**นี้ได้รับการคัดเลือกแล้ว** จากที่ประชุมวิชาการตามข้อ3.2 ให้ตีพิมพ์ในวารสารทางวิชาการนานาชาติ (International Journal) ได้แก่

International Journal of Information Acquisition, International Journal of Information Technology, **หรือ** International Journal of Pattern Recognition and Artificial Intelligence

ซึ่งทางที่ประชุมวิชาการจะแจ้งให้ทราบต่อไปภายหลังงานประชุมวิชาการในเดือนสิงหาคม 2548

อ้างอิง

Breiman, L. 1996. Bagging predictors. **Machine Learning**. 24(2): 123-140.

Craven, M., D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery. 2000. Learning to construct knowledge bases from the world wide web. **Artificial Intelligence**. 118(1-2): 69-114.

Dietterich, T. G. 1997. Machine learning research: Four Current Directions. **AI Magazine**. 18(4): 97-136.

Freund, Y. and R.E. Schapire. 1996. Experiments with a new boosting algorithm, pp. 148-156. **Proceedings of the Thirteenth International Conference on Machine Learning**. San Francisco, CA: Morgan Kaufmann.

Hansen, L. and P. Salamon. 1990. Neural network ensembles. **IEEE Trans. Pattern Analysis and Machine Intelligence**. 12: 993-1001.

List, C. 2003. **What is special about the proportion?** A research report on special majority voting and the classical Condorcet jury theorem. Public Economics 0304004, Economics Working Paper Archive at WUSTL.

Shixin, Y. 2003. **Feature selection and classifier ensembles: A Study on Hyperspectral Remote Sensing Data**. PhD thesis, University of Antwerp (CMI).

Sinka, M.P. and D.W. Corne. 2002. A large benchmark dataset for web document clustering. In Abraham, A., Ruiz-del-Solar, J., Koeppen, M. (eds.), **Soft Computing Systems: Design, Management and Applications, for Frontiers in Artificial Intelligence and Applications**. 8: 881-890.

Van Erp, M., L.G. Vuurpijl and L.R.B. Schomaker. 2002. An overview and comparison of voting methods for pattern recognition, pp. 195-200. **Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR.8)**. Niagara-on-the-Lake, Canada.

ภาคผนวก

Combining ILP with Semi-supervised Learning for Web Page Categorization

Nuanwan Soonthornphisaj and Boonserm Kijsirikul

Abstract –This paper presents a semi-supervised learning algorithm called Iterative-Cross Training (ICT) to solve the Web pages classification problems. We apply Inductive logic programming (ILP) as a strong learner in ICT. The objective of this research is to evaluate the potential of the strong learner in order to boost the performance of the weak learner of ICT. We compare the result with the supervised Naive Bayes, which is the well-known algorithm for the text classification problem. The performance of our learning algorithm is also compare with other semi-supervised learning algorithms which are Co-Training and EM. The experimental results show that ICT algorithm outperforms those algorithms and the performance of the weak learner can be enhanced by ILP system.

Keywords—Inductive Logic Programming, Semi-supervised Learning, Web Page Categorization.

I. Introduction

THE Web page categorization task is a challenging problem since the growth rate of the Web page is very high. Nevertheless, there is no common style of these Web pages in each category. Therefore we need human experts to categorize these Web pages into categories which is a time consuming and expensive task.

Many researches have been done to find the efficient algorithm which can classify these documents automatically. These algorithms use machine learning concept such as neural networks, naive Bayes and *k*-nearest neighbors [1], [2], [3] in order to find the general model of each Web's category based on a set of labeled documents. Unfortunately, the efficient algorithms still need a high portion of labeled documents to construct the model.

In order to relax the problem of the big amount of these labeled documents, the semi-supervised learning algorithms are developed. These algorithms need a small amount of labeled data, after that the algorithm will try to label the unlabeled documents and accumulate the newly labeled data during the learning process.

This paper aims to combine the Inductive Logic Programming (ILP) with the iterative cross-training algorithm (ICT). ICT is the semi-supervised learning algorithm which consists of two learners, the strong and the weak learner.

This work was supported by the Thailand Research Fund under Grant no. MRG4680156

Nuanwan Soonthornphisaj is with the Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand; e-mail fscinws@ku.ac.th

Boonserm Kijsirikul is with the Computer Engineering Department, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand; e-mail boonserm.k@chula.ac.th

The strategy of ICT is to find the strong learner that can induce the ability of the weak learner and use the weak learner with the knowledge supplied from the strong learner to do its job in the real world application.

Our assumption is that the ILP is the strong learner since it is normally supplied with the domain knowledge during the learning process, hence it has high abilty in making the decision of the Web's category. The reason that supports this idea comes from the experiments which were done on the Thai-non Thai Web page classification [4]. In that problem domain, ICT use a word-segmentation with the domain knowledge in the form of dictionary as a strong learner. We found that the word-segmentation classifier could enhance the performance of the simple naïve Bayes classifier of ICT.

This paper is organized as follows, section 2 of this paper gives the concept and the detail of ICT. Other algorithm used in comparison with ICT will be given in section 3. The experimental results and the conclusion part will be described in section 4 and 5.

II. Iterative Cross-Training

A. The Architure of ICT

In this section, we present the framework of the ICT, which consists of two learners. Each learner gets a small amount of labeled data. The strong learner (*classifier1*) starts the learning process from the labeled data and classify unlabeled data (*TrainingData2*). The weak learner (*classifier2*) uses these newly labeled data to learn and classify *TrainingData1*. The detail of ICT algoritm will be given in Table I.

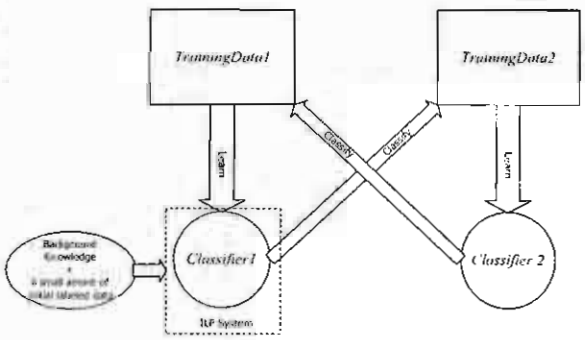


Figure. 1. The Architure of Iterative Cross-Training

TABLE I
THE LEARNING ALGORITHM OF ICT

Given:

- Two training sets *TrainingData1* for the strong learner and *TrainingData2* for the weak learner (*TrainingData1* and *TrainingData2* both contain U labeled examples).
- Use labeled data in *TrainingData1* to estimate the parameter set θ_s of the strong learner.
- Use labeled data in *TrainingData2* to estimate the parameter set θ_w of the weak learner.
- Loop until all data are labeled.
- Use the strong classifier with current θ_s to classify *TrainingData2* into categories.
  - Train the weak learner by the labeled examples in *TrainingData2* to estimate the parameter set θ_w of the classifier
  - Use the weak classifier with current θ_w to classify *TrainingData1* into categories.
  - Train the strong classifier by the labeled examples in *TrainingData1* to estimate the parameter set θ_s of the classifier.

B. The Strong Learner

The ILP system is embeded in the strong learner of ICT. Many ILP systems have been developed such as GOLEM [5], FOIL [6], PROGOL [7]. We choose the PROGOL system which uses a technique called inverse entailment to generate the single most specific hypothesis that, together with the background knowledge entails the observe data [8]. PROGOL uses the sequential covering algorithm to learn a set of rules from the hypothesis space. It employs A* search along the way to find a set of rules that represent the concept of the class. Many researchers point out that PROGOL is seen as a standard ILP learner and is often used as a benchmark when new ILP systems are introduced.

Our ILP system is supplied with two set of examples, i.e., a small amount of initial labeled data and unlabeled data. The ILP system makes use of background knowledge about the categories of the Web pages together with a set of initial labeled data to induce a set of rules (θ_s). Then the system classifies unlabeled examples using the rule set and feeds the newly labeled examples to the weak learner. The weak learner starts its process with the accumulated labeled data to estimated θ_w and classifies *Trainingdata1* into categories. The ILP system continues using its background knowledge and *Trainingdata1* as labeled examples to induce a new rule set. This process is repeated until the system is converged.

The feature sets used by the ILP system are in predicate forms. We extract three feature sets from each Web page as follows.

1) A title predicate, *has_title(p,word)*, is created using words appearing in the title of the Web page, *p*.
2) A heading predicate, *has_head(p,word)*, is created using words appearing in all headings of the Web page, *p*.
3) A hyperlink predicate, *has_link(p,word)*, is created using words appearing in all hyperlinks of the Web page, *p*.

For the background knowledge which is an important part of the ILP system, we supply the knowledge for each Web page category. This knowledge is also written in predicate form. Table II and III give example lists of background knowledge for the DrugUsage and WebKb dataset.

TABLE II
A SET OF BACKGROUND KNOWLEDGE FOR DRUGUSAGE DATASET

DrugUsage dataset		
class name	**adverse**	
adverse(adverse)	symtom(hematology)	symtom(acute)
adverse(reaction)	symtom(vascular)	symtom(liver)
adverse(interaction)	symtom(cardiovascul)	symtom(metabolism)
symtom(sleepy)	symtom(digestion)	symtom(hepatitis)
symtom(nervous)	symtom(allergy)	symtom(urinary)
class name	**overdose**	
overdose(overdosage)	effect(vomit)	contraindicate (hypoclycemia)
overdose (contraindicate)	contraindicate (contraindicate)	contraindicate(heart)
effect(fatal)	contraindicate (hypersensitivity)	contraindicate (hypertension)
effect(toxic)	contraindicate(peptic)	contraindicate (allergic)
effect(coma)	contraindicate(ulcer)	contraindicate (hypertrophy)
class name	**warning**	
warning(warn)	targetpeople(nurse)	targetpeople(dilivery)
warning(precaution)	targetpeople(pediatric	targetpeople(maternal)
targetpeople(pregnancy)	targetpeople(labor)	targetpeople(animal)
targetpeople(mother)		
class name	**patient information**	
patientinfo(patient)	physician(physician)	usage(room)
information(inform)	physician(doctor)	usage(shake)
information(product)	patientinfo(take)	usage(breath)
information (prescription)	usage(temperature)	usage(instruction)
class name	**clinical pharm**	
phamacology (phamacology)	clinical(clinic)	druganalysis(negative)
phamacology (phamacodynamic)	druganalysis(gram)	dilution(dilution)
phamacology (pharmacokinetic)	druganalysis(positive	dilution(technique)

C. The Weak Learner

For the weak learner, we employ the naive Bayes algorithm which makes its prediction based on the probability obtained from the Bayes theorem. Note that there are 2 assumptions concerning with naïve Bayes. These assumptions are (1) The presence of each word is conditionally independent of all other words in the document given the class label and (2) an assumption that the position of a word is unimportant, e.g. encountering the word "subject" at the beginning of a document is the same as encountering it at the end. The classification result (l^*) obtained from naïve Bayes can be found in Equation 1 and 2.

$$l^* = \underset{l_j}{argmax}\ Pr(l_j) \prod_{i=1}^{n} Pr(w_i \mid l_j, w_1, \ldots, w_{i-1}) \quad (1)$$

$$= \underset{l_j}{argmax}\ Pr(l_j) \prod_{i=1}^{n} Pr(w_i \mid l_j) \quad (2)$$

TABLE III
A SET OF BACKGROUND KNOWLEDGE FOR WEBKB DATASET

Class name			
course	**student**	**faculty**	**project**
subject(cs)	sport(soccer)	academic(faculty)	project_def (project)
subject(cse)	sport(hockey)	Academic (institute)	project_def (mission)
subject(ee)	sport(softball)	academic (university)	project_def (objective)
assignment(assign)	sport(golf)	academic (department)	project_def (propose)
assignment(solution)	sport(ski)	teach(course)	project_group (group)
assignment (homework)	hobby(travel)	teach(subject)	project_group (member)
assignment(problem)	hobby(movie)	teach(student)	project_group (researcher)
assignment(question)	hobby(game)	interest(research)	project_group (researcher)
assignment(quiz)	hobby(cook)	interest(paper)	project_group (people)
class(course)	hobby(cat)	interest (publication)	project roup (manager)
class(lecture)	hobby(dog)	interest(subject)	project_group (alumni)
class(lab)	relative(wife)	job(lecture)	place (laboratory)
semester(fall)	relative(friend)	job(teach)	place(lab)
semester(winter)	relative(father)	job(course)	
semester(spring)	relative(son)	activity(member)	
semester(authum)	relative (daughter)	activity(acm)	
material(handout)	relative(family)	activity(ieee)	
material(syllabus)	personal(resume	Activity (commitee)	
material(textbook)	personal(life)	activity (conference)	

The concept of naive Bayes can be found in [4]. Note that the parameter θ_i of the naive Bayes are the probabilities, $Pr(l_j)$ and $Pr(w_i|l_j)$ which are estimated from the training data. The prior probability, $Pr(l_j)$, is estimated from the ratio between the number of examples belonging to class l_j, and the number of all examples. The value of $Pr(w_i|l_j)$ is the conditional probability of seeing word w_i given class label l_j.

III. ALGORITHMS USED IN COMPARISON

A. ICT-NB Algorithm

For ICT-NB algorithm, we implement two classifiers of ICT using naive Bayes. The different between these two classifiers is the feature set. The first classifier is supplied with the words appearing on the heading of the Web page as the feature set, whereas the second classifier uses the words appearing on the content as the feature set. The learning mechanism is the same as the ICT algorithm.

B. Supervised Naive Bayes Algorithm

This algorithm is a single classifier which gets a high portion of labeled documents during the learning process. We test the performance of supervised naive Bayes using two feature sets which are heading and content feature.

C. Co-training Algorithm

The Co-Training algorithm was first introduced by Blum and Mitchell in 1998 [9]. The concept of the algorithm is based on the boosting technique.

That means, the algorithm learns from a small number of initial labeled data, and then it will incrementally classify unlabeled data into categories. The basic assumption of Co-Training is that the instance distribution is compatible with the target function. It requires that, for most examples, the target functions over each feature set predict the same label. For example, in the Web page domain, the class of the instance should be identifiable using either the text appearing on the hyperlink or the text appearing in the page content. The second assumption is that the features in one set of an instance are conditionally independent of the features in the second set, given the class of the instance.

D. Expectation-Maximization Algorithm

Another boosting style algorithm is Expectation-Maximization (EM). This algorithm was first introduced by Dempster et al. [10]. It is an iterative algorithm for maximum likelihood estimation in problems with incomplete data.

Given a model of data generation, and data with some missing values, EM iteratively uses the current model to estimate the missing values, and then uses the missing value estimates to improve the model. Using all of the available data, EM will locally maximize the likelihood of the parameters and give estimates for the missing values. Therefore, the class labels of the unlabeled data are treated as the missing values. EM has two steps, which are the E-step and M-step, respectively. The E-step calculates probabilistically weighted class labels for every document using the classifier. For the M-step, it estimates new classifier parameters using all documents. In Nigam, et al.'s work [11], they combined EM with a naive Bayes classifier to solve the text classification problem. The algorithm has shown to be able to significantly increase text classification accuracy when given limited amounts of labeled data and large amounts of unlabeled data.

IV. EXPERIMENTAL RESULT

The Web page datasets, we use for our experiments are the WebKb dataset [13] and the Drug-Usage dataset [12]. The performance evaluation is done using the standard precision (P), recall (R) and F_1-measure (F_1). These measurements are defined as follows.

$$P = \frac{\text{no. of correctly predicted examples in the target class}}{\text{no. of predicted examples in the target class}} \quad (3)$$

$$R = \frac{\text{no. of correctly predicted examples in the target class}}{\text{no. of all examples in the target class}} \quad (4)$$

$$F_1 = \frac{2PR}{P+R} \quad (5)$$

The experimental set up for each dataset is as follows.

A. The WebKb Dataset

For ICT, Co-Training and EM, we randomly selected 30% of all examples from each category to be initial labeled data. The unlabeled training data consisted of 30% of all examples and 40% of all examples were used as a test set. The experiments were conducted using 5-fold cross-validation. Table IV shows the results of all experiments conducted on the WebKb dataset. In the table, ICT-ILP stands for the performance of ICT which combines the ILP system in one of the classifiers. ICT-NB is ICT which combines two naive Bayes classifers, each of which learns from different feature sets. Co-Training stands for the Co-Training algorithm, S-Bayes stands for the supervised naive Bayes algorithm. Note that the *Classifier1* of ICT-ILP is the *Progol* system. For other algorithms, *Classifier1* means the heading-based classifier. The *Classifier2* is the content-based classifier for all of the algorithms.

Considering the two versions of ICT (ICT-ILP and ICT-NB) in Table IV, we found that ICT-ILP's performance measured by F_1 was increased from 78.25% to 80.90% on *Classifier1*. Moreover, *Classifier1* was able to boost the performance of *Classifier2*. The *Classifier2*'s performance was enhanced from 71.76% to 84.44%. Compared to the supervised naive Bayes algorithm, ICT-ILP outperformed S-Bayes on *Classifier1*. The reason that ICT-ILP got the highest performance came from the contribution of the strong learner (the Progol system).

B. DrugUsage Dataset

For ICT, Co-Training and EM, we selected 33% of all examples to be initial labeled data. The training set consisted of 33% and the remaining 34% was a test set. For the supervised naive Bayes classifier, we selected 66% of all examples to be labeled data. The test set consisted of 34% of all examples. All experiments were conducted using 3-fold cross validation.

For the performance of *Classifier1* (as shown in Table V), ICT-ILP got the highest F_1. ICT-NB's performance was increased from 69.17% to 89.90%. This means that the ILP system had contributed 30% of performance enhancement to ICT-NB. For *Classifier2*, ICT-ILP got 65.39% measured by F_1, which was higher than that of ICT-NB. The overall learning process of ICT-ILP took 1 hour to converge.

The learning process of ICT-ILP took more time than the ICT-NB, since the strong learner, Progol, needed time to do a general-to-specific search to get the optimum set of rules. In each iteration of ICT, the Progol took about 1 hour to generate the rules, therefore it took 3 hours for ICT-ILP to converge.

TABLE IV
THE PERFORMANCE OF CLASSIFIERS ON THE WEBKB DATASET

	Classifier1			Classifier2		
Algorithm	P	R	F_1	P	R	F_1
ICT-ILP	80.00	81.82	80.90	82.61	86.36	84.44
ICT-NB	71.85	94.39	78.25	67.23	86.73	71.76
Co-Training	73.95	84.69	75.64	79.69	60.72	66.14
S-Bayes	74.99	87.24	79.91	76.95	84.18	79.60
EM	68.62	91.64	75.98	76.78	74.28	70.70

TABLE V
THE PERFORMANCE OF CLASSIFIERS ON THE DRUGUSAGE DATASET.

	Classifier1			Classifier2		
Algorithm	P	R	F_1	P	R	F_1
ICT-ILP	82.37	98.32	89.9	56.03	88.2	65.39
ICT-NB	60.54	80.66	69.17	57.14	70.3	63.04
Co-Training	55.51	62.52	58.81	50.45	77.56	61.14
S-Bayes	75.74	92.67	83.35	68.81	87.12	76.89
EM	72.41	87.5	79.25	33.33	95.83	49.46

V. CONCLUSION

We have presented the enhancement version of ICT using the ILP system. We found that the induced rules have more efficiency in classify unlabeled examples. This evidence can be seen from all experimental results. The benefit of the ILP system can be seen clearly when all categories in the dataset are closely related. The reason is that most of the words in the closely related categories are likely to be equally distributed. Thus using the statistical approach like the naive Bayes learner might not perform well enough to distinguish the difference between categories. The representation of the rule sets, on the other hand, can point out the specific location in each Web page that can be used as a standard prototype of the categories.

REFERENCES

[1] R. A. Calvo and H. A. Ceccatto, "Intelligent document classification," *Intelligent Data Analysis*, vol. 4, no.5, 2000.

[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys* (CSUR), vol.34, no. 1, pp. 1-47, 2002.

[3] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proc. 22nd Annu. Int. SIGIR*, Berkley, 1999, pp. 42-49.

[4] B. Kijsirikul, P. Sasipongpairoege, N. Soonthornphisaj and S. Mcknavin, "Supervised and unsupervised learning algorithms for Thai Web page identification," In *Proc. Pacific Rim Int. Conf. on Artificial Intelligence*, Australia, 2000, pp. 690-700.

[5] S. Muggleton, S. and C. Feng, "Efficient induction of logic programs," In *Proc. 1st Conf. Algorithmic Learning Theory*. 1990.

[6] J.R. Quinlan, "Learning logical definitions from relations," *Machine Learning*, vol. 5, no. 3, pp.239-266, 1990.

[7] S. Muggleton, "Inverse entailment and progol," *New Generation Computing*, vol. 13, pp. 245-286, 1995.

[8] T.M.Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997, pp. 180-184.

[9] A. Blum and T.M. Mitchell, "Combining labeled and unlabeled data with co-training," In *Proc. 11th Annu. Conf. Computational Learning Theory*, 1998.

[10] A.P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society* Series B vol. 39, pp. 1-38, 1977.

[11] K. Nigam, A. McCallum, S. Thrun and T.M. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning*, vol. 9 no. 2, pp.103-134, 1999.

[12] DrugUsage. 2001. Data set. http://www.kindcu.siit.ac.th, Phatumtani, Thailand.

[13] WebKb. 2000. Data set. http://www.cs.cmu.edu/afs/cs.cmu.edu Carnegie Mellon University, U.S.A.

Ensemble Classifiers using Different Feature Sets for Web Page Categorization

Nuanwan Soonthornphisaj[1], and Boonserm Kijsirikul[2]

[1] Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand
fscinws@ku.ac.th

[2] Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand
Boonserm.k@chula.ac.th

Abstract. We propose a new approach to solve the Web pages categorization task. Our method consists of a set of learners which learn from different feature sets of Web page. The motivation of this work is to utilize html tag which is the unique characteristic of the Web document. These html tags provide high expressive feature that can be use to enhance the Web page categorization task. We supply these features to ensemble classifiers in order to learn and construct the models for the classifiers. The reason that makes ensemble classifiers yield better performance than the single classifier comes from the fact that each base classifier learns and constructs different models. These models lead to the different predictions on a Web page and are finalized by the voting technique. We set up experiments on WebKB and WebPage dataset using 5 fold cross validation technique. Each base classifier learns from different features which are content, meta+title, heading and link description. We compare the performance of ensemble to the single classifier and found that ensemble classifiers outperform single classifier on both datasets.

1 Introduction

Ensemble classifiers were used as a methodology in a broad area of research. Recently, the technique was implemented to solve a bioinformatics problem in order to predict the protein structure [1]. A work done by Didaci [2] employed ensemble to solve the intrusion detection on network. Grimaldi and Cunningham [3] used ensemble technique for the music recommendation system. All of these works show the success performance of ensemble. Therefore, it is interesting to see the potential of ensemble on Web page categorization problem.

Considering the related works done on text classification, we found that Adaboost technique was used in combination with support vector machine [4]. The

performance measured on Reuter and OHVMED dataset are 74.21% and 42.92%. Furthermore, the research done by Fuernkranz [5] which used the content and anchor text as a feature set got 86.86% as an average performance. Another research done by Diao et al [6] shows a promising result in Chinese news classification. They got 86.80% on the classification result. Therefore, we propose to use ensemble classifiers with variety of feature sets that can be found on the Web document.

The rest of the paper is organized as follows. In Section 2, we explain the concept of feature-based ensemble classifier. In Section 3, we briefly describe a baseline algorithm which is a naïve Bayes concept. The experimental results and conclusion part are shown in Section 4 and 5.

2 Ensemble Classifiers

This section gives details about the overview of ensemble technique, follows by the problem definition of the learning task and the feature-based ensemble framework.

2.1 Overview

One of the major advance in learning technique in the past decade was the development of ensemble or committee approaches that learn and retain multiple hypothesis and combine their decision during classification [10]. There are two well known techniques used in ensemble which are bagging and boosting. The technique called boosting [11] is an ensemble method that learns a series of weak classifiers each one focusing on correcting the errors made by the previous one. Unlike boosting, bagging technique main considering is to build a model from a subsampling training set. The concept of ensemble is to obtain a set of diversity hypotheses that can be combined in order to find an optimum classification result.

In our framework, each learner employs naïve Bayes concept which is a strong algorithm for text classification. The motivation is that we want to enhance the potential of naïve Bayes in text classification. Therefore, we integrate many naïve Bayes learners using different feature sets via ensemble technique.

2.2 Problem Definition

We start with the problem definition of supervised learning task. Below is the notation and definitions.

Y is a set of classes.
T is a set of training instances, each element is feature-classification pairs.
C is a classifier.
C^* is an ensemble.
C_i is the i^{th} classifier in ensemble C*
H is a set of hypotheses obtained from the learner

L is a learner which is a function from training set to classifiers.

For the supervised learning problem, a learner is given a set of training instances with some unknown function $y = f(x)$ in the form $\{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\}$. Each instance is a vector of the form $< x_{i,1}, x_{i,2}, \ldots, x_{i,k} >$ A learner L, is trained on a set of training examples T to produce a hypothesis, h, with the belief that it represents the true target function. The classifier employs the hypothesis to predict the class of example x. The main objective of the classification task is to learn and construct a classifier that minimizes the error in predictions of an unseen data.

The ensemble classifier C^*, collects all predictions from each base classifier C_i and make the final prediction $C^*(x)$ by the voting technique. There are two main approaches to finalize the prediction result of base classifiers, which are combining approach and selection approach. For the combining approach, when the base classifiers produce their class predictions, the final prediction result can be obtained via voting technique which considers those predictions. For the selection approach, the final result will rely on the selected classifier which has the most confident classification result. Therefore, only one classifier is selected to produce a final classification result of ensemble.

The simplest method for the combining approach is the voting technique, that collects all classification results from base classifiers and makes a voting result, y_i. The advantage of an ensemble is that they provide more accurate result than any of the base classifiers in the ensemble. But a good ensemble is one where the base classifiers in the ensemble are both accurate and make their errors on different parts of the input space [7].

In this work, we study three voting techniques which are majority vote, Borda count and weighted Borda count technique. The majority voting technique considers all predictions and make a final result using equation 1

$$y = \underset{h_i}{\operatorname{argmax}}\, h_i(x_i) \tag{1}$$

The Borda count method [12] needs a complete preference ranking (r_k) from all voters over all candidates. It computes the mean rank of each candidate over all voters. All predictions are reranked by the mean rank and the top ranked prediction class will be output as a finalized prediction.

$$y = \underset{r_k}{\operatorname{argmax}}\, r_k(x_i) \tag{2}$$

The weighted Borda count technique pays more concentration on the first rank prediction by putting more weight on the classification result of the first rank.

```
<HTML>
<TITLE> Prof. Adison Hoffman </TITLE>
<META> name = "keywords" content = "lecturer,
artificitial intelligence">
<body>
<B> Prof. Adison Hoffman</B> is currently an
lecturer at university of Heidelberg. He started
his work here since 1980. He is an expert in
Aritificial Intelligence.
<H1> Publications list </H1>
<A HREF="/list.html"> paper in Artificial
Intelligence </A>
</body>
```

Fig. 1. Feature sets based on html tags in a Web document

2.3 Feature-based Ensemble

Our proposed ensemble classifier consists of a set of learner L, which learns from different feature set of a Web page. These feature sets are words occurred in the html tag. (see Fig 1) The feature sets used in our work are as follows.

1. words appear in the anchor tag <A>
2. words appear in the heading tags <H1> <H2> <H3> and <H4>
3. words appear in meta and title tags <meta> and <title>
4. words appear in the content of the page <Body>

Therefore our feature-based ensemble has 4 base learners and 4 base classifiers. Each element in the feature vector is calculated using TFIDF [13]

3 Baseline Algorithm

This section presents a baseline algorithm used in our study which is a naïve Bayes concept.

For text classification problem, Naïve Bayes has shown its potential in many datasets. Its original idea comes from Bayes theorem which addresses the way to calculate the conditional probability. The light version of Bayes theorem called naïve Bayes works under the assumption that the likelihood of the feature's occurrence is mutually independent for each class.

Given a feature set of the Web page, naïve Bayes predicts the class of the Web page using the prior probability, $P(y_i)$ and the probability of Web page in class y_i having the observed $feature_j$, $P(feature_j|y_i)$ (see equation 3). The class that has the highest posterior probability will be assigned to the Web page.

$$P(y_i|feature_j) = P(y_i) \prod_{j=1}^{a} P(feature_j|y_i) \quad (3)$$

4 Experiments

4.1 Feature Extraction

The feature sets used in our experiments are words appear in the content of Web page, words appear on meta and title html-tag, words appear on <h_i> tag and words appear on <a> html-tag. The feature extraction process starts with the html tags removal then words that occur less than 3 times are excluded. After that the stopword deletion is done followed by the stemming process.

4.2 Experimental Setup

The purpose of the first experiment is to evaluate the performance of each feature set using single classifier. The single classifier uses naïve Bayes algorithm as a learning mechanism but learns from different features as follows.

(1)The single content-based classifier, which is a baseline algorithm (2) The single naive Bayes using tag <meta>+<title> classifier. (3)The single naive Bayes using tag <h_i> classifier. (4) The single naive Bayes using tag <a> classifier. (5) The single naive Bayes using tag <meta>+ <title>, <h_i>, <a> classifier. (6)The single naive Bayes using content and tag <meta>+<title>, <h_i>, <a>

The second experiment aims to investigate the potential of ensemble using variety of feature sets. We use three types of voting technique which are majority vote, Borda count and weighted Borda count.

4.3 Performance Measurement

We use metrics such as precision, recall and F_1-measure to evaluate the performance of learning algorithms. These metrics have been widely used for performance comparison.

$$\text{Precision} = \frac{\text{number of correctly predicted examples in the target class}}{\text{number of predicted examples in the target class}} \quad (4)$$

$$\text{Recall} = \frac{\text{number of correctly predicted example in the target class}}{\text{number of all examples in the target class}} \quad (5)$$

$$F_1\text{-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

4.4 Performance of Classifiers

The performance of single classifiers conducted on WebKB dataset are shown in Figure 2. Considering F_1-measure, we found that the single classifier using content got the highest performance (81.19%). The single classifier using heading feature got 77.68% followed by the single classifier using title+meta+h+a (75.70%) and the single classifier using title+meta (73.70%). The lowest performance belongs to the single classifier using word appearing on anchor text of the Web page which got 72.95%

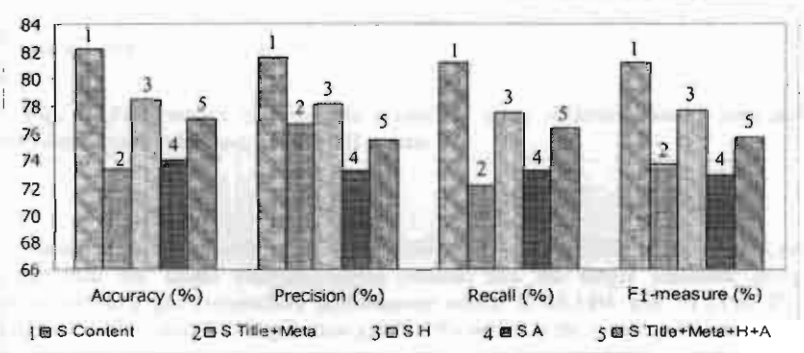


Fig. 2. Performance of single classifiers using different feature sets on WebKB dataset

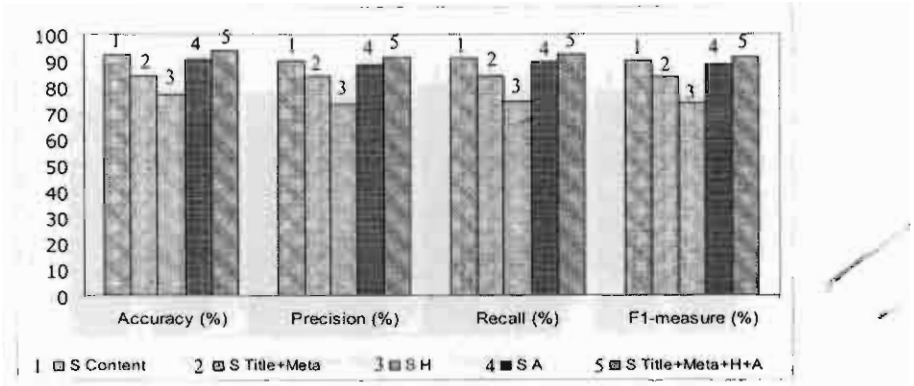


Fig. 3. Performance of single classifiers using different feature sets on WebPage data set

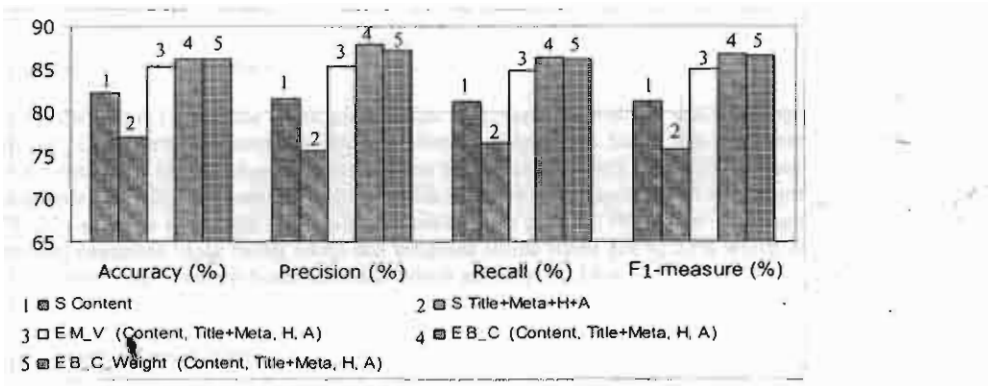


Fig. 4. Performance of ensemble classifiers using different feature sets and different voting techniques on WebKB dataset

Considering the experimental result done on WebPage dataset (Figure 3), we found that the single classifier using content and the single classifier using title+meta+h+a got competitive performance which is 90.18% and 91.57%. The single classifier using heading feature got 73.59% which is the lowest performance

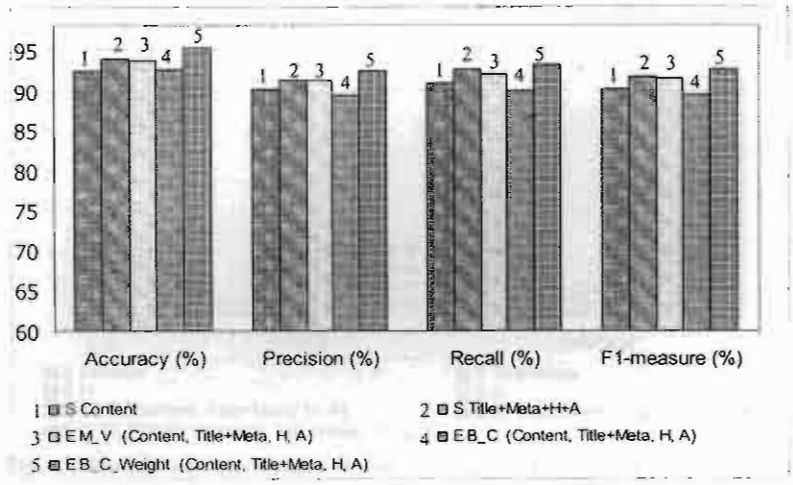


Fig. 5. The performance ensemble classifiers using different feature sets and different voting techniques on WebPage dataset

Figure 4 and 5 show the result of ensemble classifiers on WebKB and WebPage dataset. Considering experiments done on WebKB dataset, we found that ensemble using weighted Borda count got the highest performance which is 86.59%. Then ensemble using Borda count got 86.22% which is higher than majority vote which got 85.29% whereas the single content-base classifier got only 81.19%. For WebPage dataset, ensemble using Borda count and weighted Borda count got 92.52% which is higher than single content-based classifier which got only 90.18%

4.5 Computational Time

Since the ensemble classifiers consists of many classifiers, therefore it consume more computational time than single classifier. Figure 6 shows computational time used by all classifiers in our experiments on WebKB and Webpage dataset.

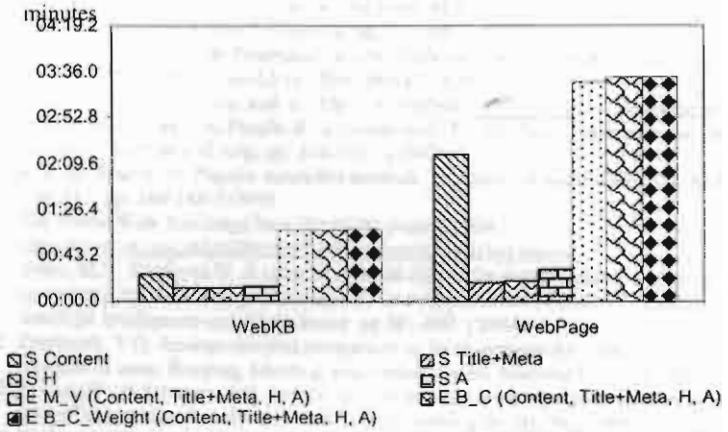


Fig. 6. Computational time measured on single and ensemble classifiers on two datasets

5 Conclusion

We presented a feature-based ensemble classifiers to solve the Web page categorization problem. Ensemble classifiers have an advantage that the system combines many learners those learn from many views of examples. These learners construct their models and are used by the classifiers. Therefore, constructing ensemble classifiers using different feature sets can enhance the performance since the system find the best solution based on the learner's experties to finalize the prediction result.

Acknowledgments. This work is supported by Thailand Research Fund under Grant no. MRG4680156

References

1 Chen, H., H.X. Zhou, X. Hu and I. Yoo. Classification comparison of prediction of solvent accessibility from protein sequences, Proceedings of the Second Conference on Asia-Pacific Bioinformatics Vol 29. pp. 33-338. (2004).

2 Didaci, L., G. Giacinto and F. Roli. Ensemble learning for intrusion detection in computer networks. Proceedings of AI*IA, Workshop on Apprendimento automatico: metodi e applicazioni. (2002)

3 Grimaldi, M. and P. Cunningham. Experimenting with music taste prediction by user profiling, Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval, pp. 173-180. (2004).

4 Cai, L. and T. Hofmann. Text categorization by boosting automatically extracted concepts, Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 182-189. (2003).
5 Fürnkranz, J. Hyperlink Ensembles: A case study in hypertext classification. Special Issue on Fusion of Multiple Classifiers. 3(4): 299-312. (2002).
6 Diao, L., K. Hu, Y. Lu and C. Shi. A method to Boost Naive Bayesian classifiers, Proceedings of the 6th Pacific-Asia Conference, Pacific Asia Conference on Knowledge Discovery and Data Mining, pp. 115-122. (2002).
7 Opitz, D., Maclin, R. Popular ensemble methods: an empirical study, Journal of AI Research, Vol. 11, , pp. 169-198. (1999)
8 The World Wide Knoledge base (WebKB) project, 2004, http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.
9 Sinka, M.P., Corne, D.W. A large benchmark dataset for web document clustering. Computing Systems: Design, Management and Applications. Vol. 87 of Frontiers in Artificial Intelligence and Applications, pp.881-890. (2002).
10 Dietterich, T.G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, 40(2), 139-157
11 Freund, Y., & Schapire, R.E. Experiments with a new boosting algorithm. Proceedings of the 13th International conference on Machine Learning (ICML-96). (1996)
12 Merijn van, E. and Louis. An overvies and comparison of voting methods for pattern recogniton, Proceeding of the 8th International Workshop on Frontier in Handwriting Recognition(IWFHR), Niagara-on-the-Lake, Canada (2002)
13 Frank, W. and Baez-Yates, R. Information Retrieval, prentice Hall.(1992)