บทที่ 2

กรรมวิธีการรู้จำรูปแบบ

กระบวนการการรู้จำตัวอักษรภาษาไทยโดยใช้เทคนิคแบบฟัซซีโลจิก และวิธีซินแทกติกมีเป้าหมายเพื่อปรับปรุงอัตราการรู้จำให้สูงขึ้นกว่างานวิจัยเดิมที่ผ่านมาโดยมีขั้นตอนการทำงานของกระบวนรู้จำดังรูปที่ 2.1 คือ

รูปที่ 2.1 ผังการทำงานของการรู้จำตัวอักษรภาษาไทยโดย
ใช้เทคนิคแบบฟัซซีโลจิกและวิธีซินแทกติก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ลักษณะการทำงาน

เมื่อรับภาพตัวอักษรภาษาไทยจากอุปกรณ์รับรู้ทางแสงอันได้แก่ เครื่องสแกนเนอร์ ซึ่งจะเก็บภาพตัวอักษรภาษาไทยที่ได้ลงไปบนแฟ้มข้อมูลภาพซึ่งเป็นลักษณะจุดภาพ นำภาพมาเข้ากระบวนการกำจัดสัญญาณรบกวนเพื่อให้ข้อมูลจุดภาพที่สแกนเข้ามาสมบูรณ์ขึ้น และนำมาทำตัวอักษรให้บางเพื่อให้เหลือความกว้างของเส้นเพียง 1 จุดภาพภาพที่ได้หลังจากขั้นตอนจะมีลักษณะโครงร่างเหมือนตัวอักษรภาษาไทย จากนั้นข้อมูลจุดภาพของตัวอักษรจะถูกนำไปเข้ารหัสเพื่อหาทิศทางของจุดภาพแล้วจึงแปลงให้เป็นเวกเตอร์ ซึ่งเวกเตอร์เหล่านี้จะถูกนำมาเชื่อมโยงกันเป็นรูป

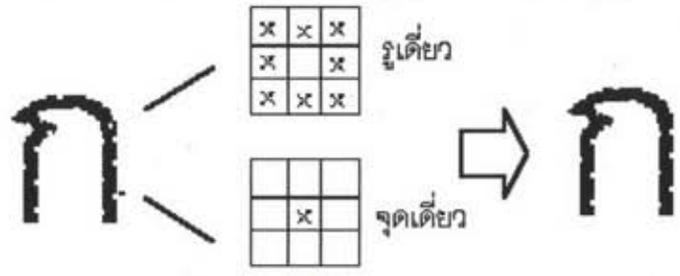
ต้นไม้เราเรียกว่า ต้นไม้เวกเตอร์ แต่เพื่อความรวดเร็วในการเปรียบเทียบก็จะเปลี่ยนเวกเตอร์ให้เหลือเพียงทิศทางของเวกเตอร์เพียงอย่างเดียว ที่เรียกว่า หน่วยสร้างพื้นฐาน (primitive) โดยยังคงนำมาเชื่อมโยงเป็นรูปต้นไม้เหมือนกับต้นไม้เวกเตอร์ ซึ่งเราเรียกว่า ต้นไม้หน่วยสร้างพื้นฐาน (primitive tree) แต่มีลักษณะการจัดเรียงลำดับ node ของต้นไม้เป็นแบบ postfix เพื่อนำไปใช้ในการหาค่าระยะระหว่างต้นไม้ที่จะในไปใช้ในการเปรียบเทียบกับตัวอักษรต้นแบบ โดยการรู้จำก็เพียงแต่เลือกเอาเฉพาะต้นไม้ของตัวอักษรต้นแบบที่ได้มีการเลือกตัวอักษรเก็บไว้ในแฟ้มต้นแบบไว้แล้ว โดยเปรียบเทียบเฉพาะตัวอักษรต้นแบบที่มีหัวอยู่ในเซตเดียวกันมาเปรียบเทียบกัน โดยถ้าผลการเปรียบเทียบกับต้นไม้ของตัวอักษรต้นแบบตัวใดมีค่าความแตกต่างน้อยที่สุด และต่ำกว่าค่าที่ยอมรับได้ แสดงว่าตัวอักษรที่นำมารู้จำจะมีความคล้ายกับตัวอักษรต้นแบบนั้นมากที่สุด

แต่ถ้าในกรณีผลการเปรียบเทียบไม่เป็นที่น่าพอใจ ต้นไม้ เวกเตอร์จะถูกนำไปปรับปรุงคุณภาพของเวกเตอร์ และถ้าสามารถปรับปรุงเวกเตอร์ได้ ต้นไม้เวกเตอร์จะถูกนำไปแปลงให้เป็นต้นไม้หน่วยสร้างพื้นฐาน และนำกลับไปเปรียบเทียบกับตัวอักษรต้นแบบ แต่ถ้าไม่สามารถปรับปรุงเวกเตอร์ได้ ก็จะนำเทคนิคฟัซซีโลจิกเข้ามาเปรียบเทียบโดยหาค่าเฉลี่ยของค่าความเป็นสมาชิก (membership value) โดยการวัดความเหมือน (similarity measure) ซึ่งถ้าค่าเฉลี่ยของค่าความเป็นสมาชิกระหว่างตัวอักษรที่ต้องการรู้จำกับตัวอักษรต้นแบบคู่ใดมีค่าเฉลี่ยมากที่สุดแสดงว่าตัวอักษรที่นำมารู้จำมีความคล้ายกับตัวอักษรต้นแบบนั้นมากที่สุด

Preprocessing

ข้อมูลภาพของตัวอักษรภาษาไทยที่รับเข้ามาจากเครื่องสแกนเนอร์จะเก็บลงเป็นแฟ้มข้อมูลภาพในนามสกุลแฟ้มของ BMP และ PCX ซึ่งนำมาใช้ในงานวิทยานิพนธ์ครั้งนี้ โดยการเก็บภาพจะเก็บภาพเป็นแบบ Binary หรือ ขาวดำ ซึ่งการเก็บข้อมูลเป็นแบบที่ใช้ 1 บิตข้อมูลต่อ 1 จุดภาพ กล่าวคือ 1 จุดภาพจะมีเพียง 2 ค่าเท่านั้น คือ 1 แสดงเป็นจุดภาพของตัวอักษร (ดำ) และ 0 แสดงว่าไม่ใช่จุดภาพ (ขาว) โดยขบวนการรับภาพของตัวอักษรเข้ามาที่มีลักษณะเป็นจุดภาพ อาจจะเกิดสัญญาณรบกวนปะปนเข้ามากับข้อมูลภาพได้ ดังนั้นจะต้องมาปรับปรุงคุณภาพข้อมูลของตัวอักษรก่อนดังนี้

การกำจัดสัญญาณรบกวน [4] สัญญาณรบกวนที่มีลักษณะเป็นจุดเดียว หรือ รูเดียว จะสามารถกำจัดได้โดยใช้วิธีการที่นิยมกันมากคือใช้วินโดว์ขนาด 3x3 จุดภาพตรวจสอบกับทุก ๆจุดภาพดังรูปที่ 2.2



ก. ก่อนการกำจัดสัญญาณรบกวน ข. หลังการกำจัดสัญญาณรบกวน

รูปที่ 2.2 การกำจัดสัญญาณรบกวน

การทำตัวอักษรให้บาง[5],[6] มีเป้าหมายที่จะทำให้ข้อมูลของตัวอักษรมีความกว้างของเส้นเพียง 1 จุดภาพ เพื่อเป็นโครงร่างของภาพ (skeleton) ซึ่งวิธีการทำตัวอักษรให้บางที่ได้ใช้ในงานวิทยานิพนธ์[4] ที่ผ่านมาได้ใช้วิธีการทำให้บางแบบดั้งเดิม (Classical Thinning) ของ Pavlidis นั้นไม่สามารถทำให้ปรากฏรอยหยักกับหัวของตัวอักษรภาษาไทยบางตัวได้ (ข,ม) และ ยังทำให้เส้นด้านบนของ สระอื หายไปหนึ่งเส้นทำให้เกิดการรู้จำผิดพลาดเป็นตัว สระอี หรือ ไม่รู้จำ ซึ่งในงานวิทยานิพนธ์นี้ได้นำวิธีทำให้บางโดยใช้ SPTA (SAFE POINT THINNING ALGORITHM)[5] เข้ามาร่วมประยุกต์ใช้เพื่อจะปรับปรุงการทำให้บางวิธีเดิมให้ปรากฏรายละเอียดของภาพตัวอักษรชัดเจนมากขึ้นโดยอัลกอริทึมของการทำให้บางแบบดั้งเดิม(Classical Thinning) ซึ่งแสดงไว้ใน ภาคผนวก ข ต่อไปนี้จะอ้างถึงจุดต่าง ๆ ที่ถูกกำหนดในการทำให้บางแบบดั้งเดิม ซึ่งแทนค่าตัวเลขที่มีความหมายดังต่อไปนี้

0 แทน จุดว่างเปล่าบนภาพ (blank pixel)

1 แทน จุดภาพ (image pixel)

2 แทน จุดที่เป็นโครงร่างของภาพ (skeletal pixel)

3 แทน จุดที่สามารถกำจัดทิ้ง (deletable pixel)

ในการตรวจสอบแต่ละจุดภาพโดยวิธีการทำให้บางแบบดั้งเดิม ก็จะทำโดยการใช้วินโดว์ขนาด 3 x 3 จุดภาพดังในรูปที่ 2.3 สำหรับส่วนรูปที่ 2.4 เป็นรูปแบบที่นำไปใช้เพื่อการตรวจสอบหาจุดที่เป็นโครงร่างของภาพโดยในแต่ละวินโดว์ที่แทนด้วยกลุ่มตัวอักษร A หรือ B นั้น ในกลุ่มของตัวอักษร A หรือ B ต้องมีจุดที่มีค่าไม่เป็น 0 อย่างน้อย 1 จุดจึงจะถือว่าจุด P นั้นเป็นโครงร่างของ

ภาพ ส่วนขอบของภาพนั้นจะแบ่งได้ออกเป็น 4 อย่าง คือ ขอบบน , ขอบล่าง , ขอบซ้าย , และขอบขวา จากรูปที่2.4 จุดที่เป็นขอบของภาพในแต่ละขอบจะพิจารณาเป็นดังนี้

จุด X เป็น "ขอบบน" ของภาพเมื่อจุดภาพข้างเคียง X_2 มีค่าเป็น 0
จุด X เป็น "ขอบล่าง" ของภาพเมื่อจุดภาพข้างเคียง X_6 มีค่าเป็น 0
จุด X เป็น "ขอบซ้าย" ของภาพเมื่อจุดภาพข้างเคียง X_4 มีค่าเป็น 0
จุด X เป็น "ขอบขวา" ของภาพเมื่อจุดภาพข้างเคียง X_0 มีค่าเป็น 0

X_3	X_2	X_1
X_4	P	X_0
X_5	X_6	X_7

รูปที่2.3 วินโดว์ขนาด 3x3

A	A	A
0	P	0
B	B	B

A	A	A
A	P	0
A	0	2

รูปที่ 2.4 รูปแบบในการตรวจสอบหาจุดที่เป็น .
โครงร่างของภาพของ Classical Thinning

จากการสังเกตและศึกษารูปแบบในการตรวจสอบของ SPTA (SAFE POINT THINNING ALGORITHM) มีวินโดว์ 3x3 ที่ใช้ในการตรวจสอบอยู่หนึ่งรูปแบบที่การทำให้บางแบบดั้งเดิม (Classical Thinning) ไม่มีคือรูปแบบดังรูปที่ 2.5

X	X	X
0	P	0
Y	Y	Y

รูปที่ 2.5 รูปแบบใน SPTA ไม่เหมือนใน Classical Thinning

รูปแบบในรูปที่ 2.5 ที่ตัวอักษร X หรือ Y เราสามารถแทนจุด (X หรือ Y) ด้วยค่า 0 หรือ 1 ซึ่งเราจะเรียกการแทนค่าจุดในลักษณะนี้ว่า don't care โดยรูปแบบในรูปที่ 2.5 เป็นรูปแบบ ทั่วไป ขณะที่ตัวอักษร X เป็น 0 และ ตัวอักษร Y ต้องมีค่าเป็น 1 อย่างน้อย 1 ตัว พร้อมกับหมุนวินโดว์ในรูปที่ 2.5 ไป 90 องศาจะได้ลักษณะรูปแบบที่แยกรายละเอียดออกมา 2 รูปแบบ ดังรูปที่ 2.6 ก และ รูปที่ 2.6 ข ในทางกลับกันให้ตัวอักษร Y เป็น 0 และ ตัวอักษร X ต้องมีค่าเป็น 1 อย่างน้อย 1 ตัว พร้อมกับหมุนวินโดว์ในรูปที่ 2.5 ไป 90 องศา ก็จะได้ลักษณะรูปแบบออกมา 2 รูปแบบ ดังรูปที่ 2.6 ค และ รูปที่ 2.6 ง

0	0	0
0	P	0
Y	Y	Y

ก

0	0	Y
0	P	Y
0	0	Y

ข

X	X	X
0	P	0
0	0	0

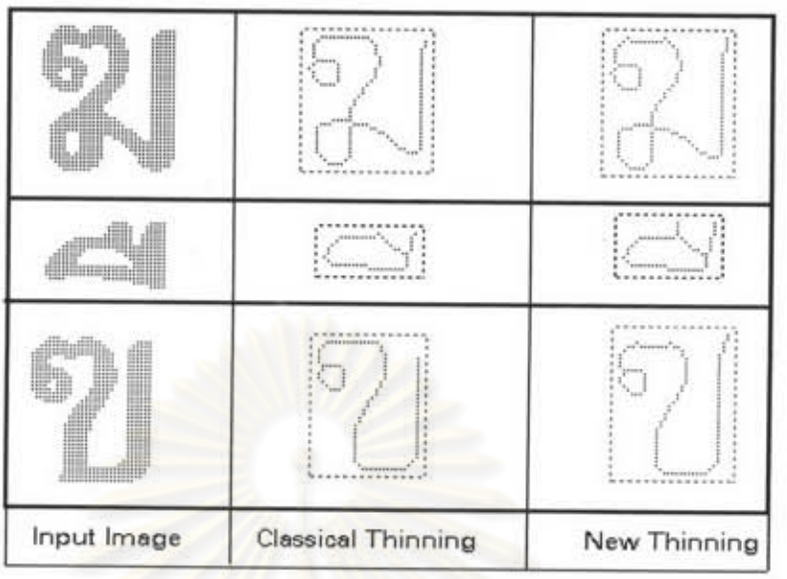
ค

X	0	0
X	P	0
X	0	0

ง

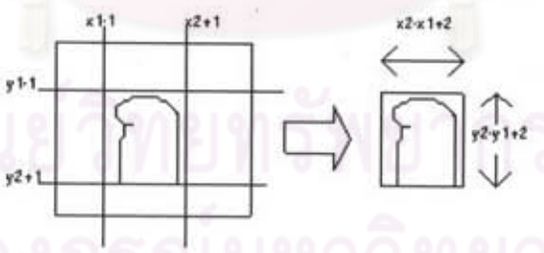
รูปที่ 2.6 รูปแบบที่เป็นไปได้ทั้งหมดของรูปที่ 2.5

ได้มีการทดสอบโดยการนำรูปที่ 2.6 ในแต่ละรูปแบบไปทำการเพิ่มเข้าไปในส่วนรูปแบบในการตรวจสอบหาจุดที่เป็นโครงร่างแล้ว รูปที่ 2.6 ก. จะได้รับการคัดเลือกเพิ่มเข้าในส่วนรูปแบบในการตรวจสอบหาจุดที่เป็นโครงร่างของภาพในการทำให้บางแบบดั้งเดิม (Classical Thinning) ซึ่งจะได้ผลของภาพในการทำให้บาง (Thinning) เป็นที่น่าพอใจดังรูปที่ 2.7 คือทำให้ปรากฏรอยหยักเช่น ตัวอักษร ข,ฆ และปรากฏขีดของเส้นตรงเช่น สระอือ เป็นต้น



รูปที่ 2.7 แสดงเปรียบเทียบการทำให้บางโดยใช้ Classical Thinning กับ Thinning ที่นำเอา SPTA Thinning มาประยุกต์ใช้ร่วม

<u>การปรับกรอบของตัวอักษร</u> [4]เป็นการลดขอบภาพที่ว่างเป็นจำนวนมากให้พอดีกับขนาดของภาพโดยขอบนอกสุดของภาพทั้ง 4 ด้านจะทำให้ว่างไว้ด้านละ 1 จุดภาพซึ่งได้ข้อมูลภาพใหม่ที่มีกรอบภาพดังรูปที่ 2.8

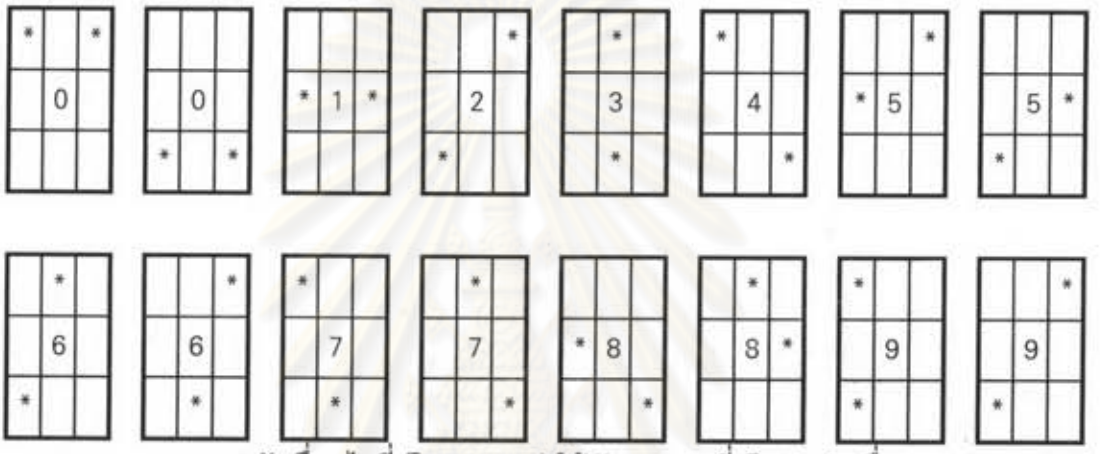


รูปที่ 2.8 แสดงการปรับกรอบภาพ

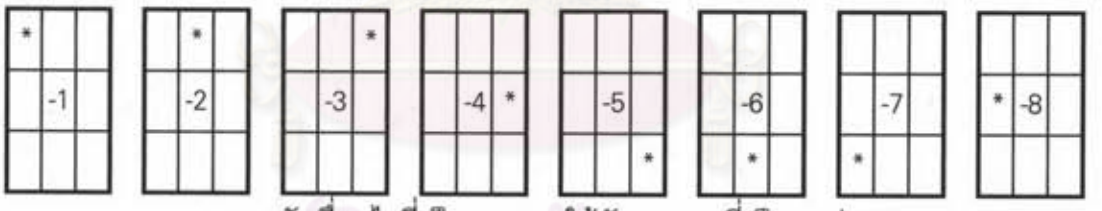
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Pattern Representation

หลังจากการเข้าขบวนการ Preprocessing แล้วเราจะได้โครงร่างของภาพตัวอักษรภาษาไทย จากนั้นนำมาแทนค่าด้วยรหัสเงื่อนไขเพื่อหาทิศทางของจุดภาพที่เรียงต่อเนื่องกันไป โดยใช้ตารางหน้าต่างขนาด 3x3 จุดภาพ ตรวจสอบกับทุก ๆ จุดภาพและทำการเปรียบเทียบกับจุดภาพรอบข้างทั้ง 8 จุดภาพในตารางหน้าต่าง โดยมีเงื่อนไขการแทนค่าด้วยรหัสเงื่อนไขดังรูปที่ 2.9



ก. รหัสเงื่อนไขที่เป็นเลขบวก ให้กับจุดภาพที่เป็นจุดต่อเนื่อง



ข . รหัสเงื่อนไขที่เป็นเลขลบ ให้กับจุดภาพที่เป็นจุดปลาย

รูปที่ 2.9 รหัสเงื่อนไข

รหัสเงื่อนไขของแต่ละจุดภาพที่แสดงในรูปที่ 2.9 ก. ที่เป็นจำนวนบวกที่มีค่า 1 ถึง 8 แสดงถึงมุมที่กระทำในทิศทางต่าง ๆ กัน เช่น รหัส 1 ถึง 4 เรียกว่ารหัสเงื่อนไขหลัก ทำมุมเป็นทวีคูณของมุม 45 องศา รหัส 5 ถึง 8 เรียกว่ารหัสเงื่อนไขรอง ทำมุมเป็นทวีคูณของมุม 30 องศา ส่วนรหัสเงื่อนไข 0 และ 9 เป็นจุดหักเหของเส้นโครงร่าง ส่วนตัวเลขลบแสดงถึงเป็นจุดปลายของภาพดังรูปที่ 2.9 ข. , ตัวเลข -9 แสดงถึงจุดทางแยกที่มีจุดรอบข้างมากกว่า 2 จุดขึ้นไป และ จุดที่ไม่ใช่จุดภาพ หรือจุดภาพที่มีจุดภาพรอบข้างนอกเหนือจากที่ระบุไว้ให้แทนด้วยรหัสเงื่อนไข -128 แสดงได้ดังรูปที่ 2.9

รูปที่ 2.10 การเข้ารหัสเงื่อนไข

การแปลงจุดภาพที่เข้ารหัสให้เป็นเวกเตอร์[4]

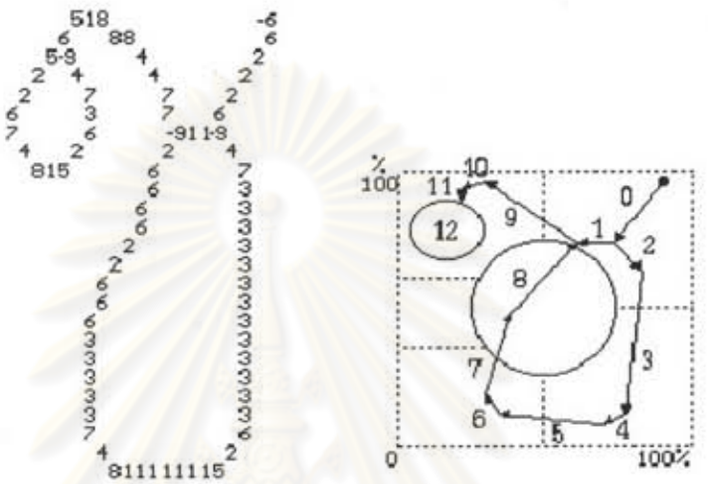
จากจุดภาพที่มีรหัสเงื่อนไขเหมือนกัน หรือเป็นรหัสเงื่อนไขที่ต่อเนื่องกัน ดังตาราง 2.1 มีทิศทางไปทางเดียวกันมารวมกันเป็น 1 เวกเตอร์ แล้วทำการเก็บข้อมูลของตำแหน่งต้นทางและปลายทางไว้ เราเรียกเวกเตอร์ชนิดนี้ว่าเป็นเวกเตอร์เส้นตรง

รหัสเงื่อนไข	รหัสต่อเนื่อง	รหัสไม่ต่อเนื่อง
1	1,5,8	-
2	2,5,6	9,0
3	3,6,7	-
4	4,7,8	9,0
5	1,2,5,6	0,8,9
6	2,3,5,6	0,7,9
7	3,4,7,8	0,6,9
8	2,4,7,8	0,5,9
9	2,4,5,6,7,8	ทุกรหัส
0	2,4,5,6,7,8	ทุกรหัส

ตารางที่ 2.1 รหัสต่อเนื่อง และรหัสไม่ต่อเนื่อง ของรหัสเงื่อนไข

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ส่วนเวกเตอร์อีกชนิดหนึ่งที่ได้มีการนำมาใช้คือเวกเตอร์วงกลม เป็นการพิจารณาแทนจุดภาพที่มีตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายวนมาบรรจบกันที่ตำแหน่งเดียวกัน ซึ่งเป็นได้ทั้งวงกลมอิสระ หรือวงกลมที่มีเวกเตอร์อื่นมาต่อเชื่อมด้วย ซึ่งจะกำหนดให้มีจุดต่อเชื่อมเพียง 1 จุดให้สามารถเชื่อมต่อกับเวกเตอร์เส้นตรงได้ 1 เวกเตอร์

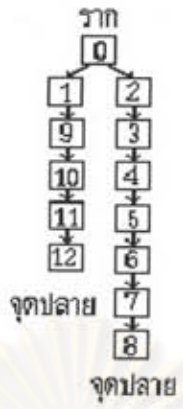


ก. ข้อมูลภาพที่เข้ารหัสเงื่อนไข ข. เวกเตอร์ของตัวอักษร

รูปที่ 2.11 แสดงการแปลงรหัสภาพเป็นเวกเตอร์

การเชื่อมเวกเตอร์เป็นรูปต้นไม้

เวกเตอร์ที่ได้จะถูกนำมาเชื่อมโยงเข้าด้วยกันเป็นรูปต้นไม้ เพื่อต่อเชื่อมเวกเตอร์ต่าง ๆ ให้เหมือนกับรูปของโครงร่างตัวอักษรตามเดิม ซึ่งแบบของรูปต้นไม้ที่นำมาใช้กับงานวิทยานิพนธ์นี้คือ ต้นไม้แบบแตกสองลำดับที่ 3 (Binary-Tree of Order 3) โดยข้อมูลของชุดของเวกเตอร์ประกอบด้วย ต้นไม้ของเวกเตอร์ , ตารางตำแหน่งราก , ตารางตำแหน่งจุดปลายของตัวอักษร , ตารางตำแหน่งจุดปลายที่เป็นวงกลม และจำนวนจุดทางแยกของตัวอักษร ซึ่งทั้งหมดจะนำไปใช้ประโยชน์ในการรู้จำ [4]



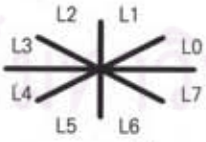
รูปที่ 2.12 แสดงต้นไม้ของเวกเตอร์ของรูปที่ 2.11

การเปลี่ยนเวกเตอร์เป็นหน่วยสร้างพื้นฐาน

เพื่อสามารถนำไปเปรียบเทียบกับตัวอักษรต้นแบบได้อย่างรวดเร็วโดยการ แปลงเวกเตอร์ให้อยู่ในรูปของหน่วยสร้างพื้นฐาน ซึ่งลักษณะของหน่วยสร้างพื้นฐาน ที่ได้กำหนดไว้มี

1.1 หน่วยสร้างพื้นฐานเส้นตรง มี 8 รูปแบบ(freeman chain code) เพื่อใช้แทนเวกเตอร์เส้นตรงที่ทำมุมกับทิศทั้ง 8 โดยที่แต่ละหน่วยสร้างพื้นฐาน ทำมุม 45 องศา ดังแสดงในรูปที่ 2.13 เช่น หน่วยสร้างพื้นฐานเส้นตรงหมายเลข 0 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 0 ถึง 45 องศา ,หน่วยสร้างพื้นฐานเส้นตรงหมายเลข 1 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 45 ถึง 90 องศา เป็นต้น

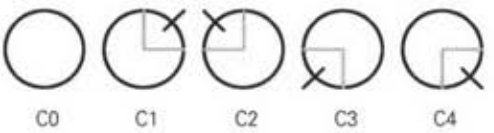
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.13 หน่วยสร้างพื้นฐานเส้นตรง

1.2 หน่วยสร้างพื้นฐานวงกลม มี 5 รูปแบบ โดยหน่วยสร้างพื้นฐานแรกเป็นวงกลมที่ไม่มีจุดต่อเชื่อมกับเวกเตอร์อื่น ใช้แทนเวกเตอร์วงกลมที่ไม่มีจุดต่อเชื่อมกับเวกเตอร์อื่นเลย

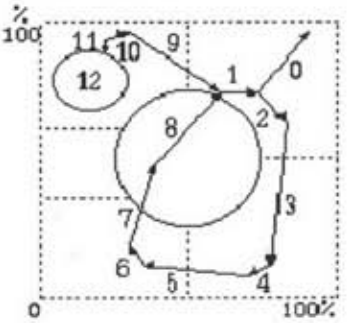
ส่วนหน่วยสร้างพื้นฐานที่เหลือเป็นวงกลมที่มีจุดต่อเชื่อมในพื้นที่ 1 ใน 4 ของวงกลม แทนเวกเตอร์วงกลมที่มีจุดต่อเชื่อม ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 หน่วยสร้างพื้นฐานวงกลม

หลังจากเปลี่ยนเวกเตอร์เป็นหน่วยสร้างพื้นฐาน จะนำหน่วยสร้างพื้นฐานที่ได้มาเชื่อมโยงเข้าด้วยกันเป็นต้นไม้หน่วยสร้างพื้นฐานที่มีลักษณะคล้ายกับต้นไม้ของเวกเตอร์ แต่มีการจัดเรียงลำดับของ node ในต้นไม้เสียใหม่ ให้มีการจัดเรียงลำดับแบบ Postfix [4] เพื่อสำหรับนำไปใช้ในการเปรียบเทียบกับตัวอักษรต้นแบบ ดังรูปที่ 2.15 แสดงการแปลงเวกเตอร์ที่แสดงในรูปที่ 2.11 โดยข้อมูลของต้นไม้หน่วยสร้างพื้นฐาน จะประกอบไปด้วย ตัว หน่วยสร้างพื้นฐาน , ค่ามุมเบี่ยงเบน , เขตของส่วนหัวและส่วนท้ายของหน่วยสร้างพื้นฐาน , หมายเลข node ตามลำดับ Postfix , ตัวชี้ไปยัง node บน 1 ตัวชี้ และ ตัวชี้ไปยัง node ล่าง 3 ตัวชี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ก. เวกเตอร์ของตัวอักษร
ที่จัดเรียงแบบ postfix

รูป ข ต้นไม้เวกเตอร์ รูป ค ต้นไม้หน่วยสร้างพื้นฐาน

รูปที่ 2.15 แสดงการแปลงต้นไม้เวกเตอร์ให้เป็นต้นไม้หน่วยสร้างพื้นฐาน
โดยมีการเรียงลำดับ node ตามลำดับ postfix

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Classification

การเปรียบเทียบตัวอักษรโดยวิธีซินแทกติก

การเปรียบเทียบนี้เป็นการเปรียบเทียบระหว่างตัวอักษรที่ต้องการรู้จำกับตัวอักษรต้นแบบ ซึ่งได้มีการเปรียบเทียบดังนี้คือ

1. การเปรียบเทียบโดยการวิเคราะห์โครงร่างของตัวอักษร เป็นการเปรียบเทียบเพื่อหาค่าความแตกต่างระหว่างต้นไม้ของหน่วยสร้างพื้นฐาน ของตัวอักษรต้นแบบ(กลุ่มที่มีตัวอักษรคล้ายคลึงกัน)กับตัวอักษรที่ต้องการรู้จำ โดยใช้วิธีการคำนวณหาระยะห่างระหว่างต้นไม้ (Tree-to-Tree Distance) ซึ่งผลเปรียบเทียบที่ได้เราเรียกว่า ค่าระยะ [4]

2. การเปรียบเทียบโดยการวิเคราะห์ทาง feature เพื่อสนับสนุนการเปรียบเทียบโดยการวิเคราะห์โครงร่างของตัวอักษรแบบหาระยะระหว่างต้นไม้ของตัวอักษร ซึ่งการเปรียบเทียบโดยวิธีการนี้เป็นการนำเอาลักษณะเด่นของตัวอักษรมาทำการเปรียบเทียบ ซึ่งผลการเปรียบเทียบเราเรียกว่า ค่าความต่าง [4]

ลักษณะเด่นที่นำมาเปรียบเทียบได้แก่

2.1 การเปรียบเทียบตำแหน่งของจุดปลาย

2.2 การเปรียบเทียบส่วนหัวของตัวอักษร

2.3 การเปรียบเทียบกรณีพิเศษเช่น การเปรียบเทียบประเภทหัวของตัวอักษรเป็นต้น

3. การวิเคราะห์ผลการเปรียบเทียบ จากวิธีการเปรียบเทียบตัวอักษรที่ต้องการรู้จำกับตัวอักษรต้นแบบที่ผ่านมา ผลที่ได้คือค่าระยะและค่าความต่าง เพื่อใช้ในการพิจารณาเลือกตัวอักษรต้นแบบที่มีความคล้ายกับตัวอักษรที่ต้องการรู้จำมากที่สุดได้ โดย

3.1 พิจารณาค่าความต่างของตัวอักษรต้นแบบที่น้อยที่สุด เป็นอันดับแรก

3.2 พิจารณาค่าระยะของตัวอักษรต้นแบบที่มีค่าน้อยที่สุดเป็น อันดับต่อไป

3.3 ค่าความต่างที่ยอมรับได้จะต้องมีค่าไม่เกิน 6 ซึ่งได้จากตัวอักษรที่มีจำนวนจุดปลายที่อยู่ในระดับตัวอักษรเดียวกันต่างกันไม่เกิน 3 จุด หรือตัวอักษรที่มีจำนวนจุดปลายที่อยู่ต่างระดับตัวอักษรไม่เกิน 1 จุด

3.4 ค่าระยะที่ยอมรับได้จะต้องมีค่าไม่เกิน 20 ได้จากค่าสูงสุดของค่าระยะของตัวอักษรที่นำมาทดสอบกับตัวอักษรต้นแบบ

การปรับปรุงเวกเตอร์

จากการเปรียบเทียบที่ได้กล่าวมา ถ้าผลเปรียบเทียบยังไม่อยู่ในเกณฑ์ที่ยอมรับได้ คือ มีค่าระยะมากกว่า 20 และมีค่าความต่างมากกว่า 6 ข้อมูลภาพของตัวอักษรที่รับเข้ามาโอกาสผิดไปจากต้นแบบที่เป็นไปได้ จึงต้องมีการปรับปรุงเวกเตอร์เพื่อพยายามค้นหาสิ่งที่คาดว่าจะเป็นสิ่งผิดปกติ ในเวกเตอร์ของตัวอักษรที่ต้องการรู้จำ เพื่อทำการแก้ไขให้ได้ใกล้เคียงกับลักษณะของตัวอักษรที่เป็นต้นฉบับมากที่สุด ซึ่งการปรับปรุงเวกเตอร์แบ่งเป็น 2 ประเภทคือ

1. การเชื่อมเวกเตอร์ที่อยู่ใกล้กัน [4]

โดยจะพิจารณาเวกเตอร์เส้นตรงที่มีจุดปลายของตัวอักษร 2 เวกเตอร์ที่อยู่ใกล้กันมากที่สุด และมีระยะห่างของจุดปลายไม่เกิน 7% ของขนาดของกรอบเวกเตอร์ ซึ่งมีโอกาสที่จะเชื่อมเป็นเวกเตอร์เส้นเดียวกันได้

2. การตัดเวกเตอร์ส่วนเกินออก

เวกเตอร์ส่วนเกินก็คือเวกเตอร์เส้นตรงที่เป็นจุดปลายของตัวอักษรและมีความยาวของเวกเตอร์ไม่เกิน 15% ของขนาดของกรอบเวกเตอร์ การตัดจะตรวจหาเวกเตอร์ที่สั้นที่สุดที่ต่อเชื่อมกับจุดทางแยก และจะตัดเวกเตอร์นี้ออกก่อนเวกเตอร์ที่สั้นที่สุดที่ต่อกับเวกเตอร์อื่นเป็นเส้นตรง ทั้งนี้เพราะเวกเตอร์ที่ต่อกับจุดทางแยกมีโอกาศเป็นเส้นกิ่งขนาดเล็กที่เป็นส่วนเกินของตัวอักษร การตัดเวกเตอร์ส่วนเกินจะตัดออกครั้งละ 1 จุดเท่านั้น

การเปรียบเทียบโดยใช้เทคนิคฟัซซีโลจิก[7]-[9]

การนำเทคนิคฟัซซีโลจิกมาใช้เพื่อเปรียบเทียบหาตัวอักษรในกรณีการเปรียบเทียบโดยวิธีชินแทกติกไม่สามารถรู้จำตัวอักษร หรือรู้จำตัวอักษรผิด ซึ่งเทคนิคนี้ช่วยแก้ปัญหาเรื่องความไม่แม่นยำในการตัดสินใจดังภาคผนวก ก.

การเปรียบเทียบโดยใช้เทคนิคฟัซซีโลจิกได้แบ่งออกเป็น 2 ส่วน คือ

1. ขั้นตอนการตรวจสอบ

มีจุดประสงค์ เพื่อตัดสินว่าควรใช้เทคนิคฟัซซีโลจิกวัดค่าความเหมือนของตัวอักษรหรือไม่ โดยมีการกำหนดขีดเริ่มเปลี่ยน (Threshold) ซึ่งจะพิจารณาเฉพาะตัวอักษรที่ต้องการรู้จำกับตัวอักษรต้นแบบที่มีค่าระยะ และค่าความต่างดังที่กล่าวมาข้างต้นที่น้อยที่สุดจากการเปรียบเทียบโดยวิธีการที่ผ่านมา

ขีดเริ่มเปลี่ยน(Threshold) ที่ใช้พิจารณาได้แก่

- ความแตกต่างของจำนวนส่วนย่อย(Session)
- ความแตกต่างของจำนวนวงกลม
- มีเส้นระดับบน (Topline) ที่แตกต่างกว่าเส้นระดับ 100% หรือไม่
- มีเส้นระดับล่าง(Baseline) ที่แตกต่างกว่าเส้นระดับ 0% หรือไม่

ในกรณีปกติทั่วไป ความแตกต่างเท่ากับ 0 แต่ถ้าเกิดกรณีใดกรณีหนึ่งความแตกต่างไม่เท่ากับ 0 ย่อมแสดงว่าตัวอักษรที่ต้องการรู้จำผิดไปจากตัวอักษรต้นแบบตัวแรกที่มีค่าระยะ และ ค่าความต่างน้อยที่สุด ดังนั้นจะใช้เทคนิคฟัซซีเข้ามาในการเปรียบเทียบ

2. การวัดและเปรียบเทียบค่าความเหมือน (Similarity)

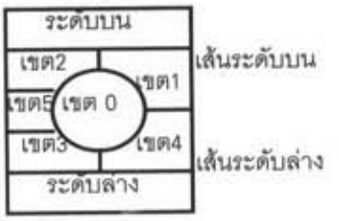
ในการวัดค่าความเหมือนจะกระทำกับเอกลักษณ์ของข้อมูลของตัวอักษรภาพที่ต้องการรู้จำกับข้อมูลของตัวอักษรภาพต้นแบบที่มีอยู่ แล้วจึงทำการหาปริมาณของความเป็นสมาชิก (membership)ของความเหมือนที่วัดได้ ซึ่งค่าความเป็นสมาชิกต้องอยู่ระหว่าง 0 ถึง 1 : [0,1]

$$\mu_{\text{-unique-}} : x \longrightarrow [0,1]$$

2.1 การวัดความเหมือน (Similarity measure)

2.1.1 ค่าความเหมือนของเขตหัวตัวอักษร

เขตหัวตัวอักษรกำหนดไว้ทั้งหมด 8 เขต คือเขต 0-7 ดังรูปที่ 2.16



รูปที่ 2.16 เขตหัวตัวอักษร

โดยที่ ระดับบนคือ เขต 6 และ ระดับล่างคือ เขต 7

หาความแตกต่างของเขตหัวตัวอักษรที่ต้องการรู้จำกับเขตหัวตัวอักษรต้นแบบแล้วนำมาหารด้วยจำนวนเขตทั้งหมดเพื่อเป็นการเฉลี่ยของความแตกต่างของเขตดังสมการ (1)

$$d_0 = \| \text{Zone-}_{\text{"ตัวอักษรที่รับเข้ามา"}} - \text{Zone-}_{\text{"ตัวอักษรต้นแบบ"}} \| / 8 \quad (1)$$

2.1.2 ค่าความเหมือนโดยจำนวนวงกลม

จำนวนวงกลมที่เกิดขึ้นได้มากที่สุดในตัวอักษรไทยมีแค่ 2 วงกลมเท่านั้น ดังนั้นการหาค่าเฉลี่ยของความแตกต่างของจำนวนวงกลมของตัวอักษรที่ต้องการรู้จำกับตัวอักษรต้นแบบจะหารด้วย 2 ดังสมการ (2)

$$d_1 = \| \text{จำนวนวงกลมของตัวอักษรที่ต้องการรู้จำ} - \text{จำนวนวงกลมของต้นแบบ} \| / 2 \quad (2)$$

2.1.3 ความเหมือนโดยจำนวนจุดปลายของตัวอักษร

จุดปลายคือ จุดที่มีจุดอื่นรอบตัวเพียงจุดเดียวในวินโดว์ขนาด 3x3 ดังรูปที่ 2.9ข. จำนวนจุดปลายที่เกิดขึ้นมากที่สุดในตัวอักษรภาษาไทยมีจำนวน 6 จุดเท่านั้น ดังนั้นการหาค่าเฉลี่ยของความแตกต่างของจำนวนจุดปลายของตัวอักษรที่ต้องการรู้จำกับตัวอักษรต้นแบบจะหารด้วย 6 ดังสมการ (3)

$$d_2 = \| \text{จำนวนจุดปลายของตัวอักษรที่ต้องการรู้จำ} - \text{จำนวนจุดปลายของต้นแบบ} \| / 6 \quad (3)$$

2.1.4 ค่าความเหมือนโดยระดับบน หรือ ระดับล่าง

เป็นการหาค่าเฉลี่ยของความแตกต่างที่เกิดขึ้นของระดับบน หรือ ระดับล่าง โดยเป็นการหาความแตกต่างของระดับบน หรือ ระดับล่างที่เกิดขึ้นของตัวอักษรที่ต้องการรู้จำ กับ ระดับบน หรือ ระดับล่างของตัวอักษรต้นแบบ ตามลำดับดังสมการ (4) และ สมการ(5)

d_3 = ll ค่าระดับบนของตัวอักษรที่ต้องการรู้จำ - ค่าระดับบนของต้นแบบ ll / 100 (4)

d_4 = ll ค่าระดับล่างของตัวอักษรที่ต้องการรู้จำ - ค่าระดับล่างของต้นแบบ ll / 100 (5)

โดยค่าแตกต่างของระดับบน และระดับล่างถูกกำหนดไว้เท่ากับ 100 เปอร์เซ็นต์ของขนาดภาพตัวอักษร

2.1.5 ค่าความเหมือนโดยจำนวน node ของต้นไม้

เป็นการหาค่าเฉลี่ยของความแตกต่างของจำนวน node ของต้นไม้ที่ต้องการรู้จำกับจำนวน node ของต้นไม้ต้นแบบ โดยจะทำการหารด้วยจำนวน node ที่มากที่สุด(maxnode) โดยเปรียบเทียบจากจำนวน node ของต้นไม้ที่ต้องการรู้จำกับจำนวน node ของต้นไม้ต้นแบบ ดังสมการ (6)

d_5 = ll จำนวน node ของตัวอักษรที่ต้องการรู้จำ - จำนวน node ของต้นแบบ ll / maxnode (6)

นำ d_i โดยที่ i = 0 ถึง 5 มาหาค่าความเป็นสมาชิกของความเหมือน μ_i โดยนำไปใส่ในฟังก์ชั่นความเป็นสมาชิก (membership function)

$$\mu_i = \exp(-k * d_i) \quad (7)$$

กำหนด k = 0.6 เพราะกราฟเอกโปเนเชียลให้ความชันที่เหมาะสม

2.2 ขั้นตอนการเปรียบเทียบ

จากที่ผ่านมาจะได้ฟัซซีเซต(C)ของค่าความเป็นสมาชิก (μ_i) ของความเหมือนของตัวอักษรที่ต้องการรู้จำกับตัวอักษรต้นแบบ C = { μ_0 , μ_1 , μ_2 , μ_3 , μ_4 , μ_5 }

ให้ μ(C) คือปริมาณเฉลี่ยของความเป็นสมาชิกของ C ซึ่งมีความเป็นไปได้ที่ตัวอักษรที่ต้องการรู้จำ (i) กับตัวอักษรต้นแบบ (R) เป็นตัวอักษรเดียวกัน โดย μ(C) จะพิจารณาค่าของ μ_i ทั้งหมดใน C

$$\mu(c) = \sum_{i=0}^{5} \mu_i$$

ให้ $\mu(C_k)$ คือปริมาณเฉลี่ยของความเป็นสมาชิกของตัวอักษรที่ต้องการรู้จำ i และตัวอักษรต้นแบบ R_k โดยที่ k คือ จำนวนอักษรต้นแบบทั้งหมดที่ถูกจับคู่โดยการเปรียบเทียบโดยวิธีซินแทกติกที่ผ่านมา โดย $\mu(C_m)$ คือปริมาณความเป็นสมาชิกเฉลี่ยที่มากที่สุดของตัวอักษรที่ต้องการรู้จำ i กับอักษต้นแบบ k ตัว

$$\mu(C_m) = \underset{k}{Max}\ \mu(C_k)$$

เราจะสรุปได้ว่า ตัวอักษรที่ต้องการรู้จำ i ถูกรู้จำเป็นตัวอักษรต้นแบบ R_m ด้วยค่าความเป็นสมาชิกเฉลี่ยของความเหมือน $\mu(C_m)$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย