การถอดชื่อบุคคลจากอักษรไทยเป็นอักษรโรมันโดยอาศัยความนิยมในการใช้เป็นฐาน

นายเอกพล ตั้งวีระพงษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

ROMANIZATION OF THAI PROPER NAMES BASED ON POPULARITY OF USAGE

Mr. Akegapon Tangverapong

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2008

หัวข้อวิทยานิพนธ์	การถอดชื่อบุคคลจากอักษรไทยเป็นอักษรโรมันโดยอาศัยความนิยมในการใช้เป็นฐาน
โดย	นายเอกพล ตั้งวีระพงษ์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร. อติวงศ์ สุชาโต
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ผู้ช่วยศาสตราจารย์ ดร.โปรดปราน บุณยพุกกณะ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ นงลักษณ์ โควาวิสารัช)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร. อติวงศ์ สุชาโต)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(ผู้ช่วยศาสตราจารย์ ดร.โปรดปราน บุณยพุกกณะ)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.สุดาพร ลักษณียนาวิน)

เอกพล ตั้งวีระพงษ์ : การถอดชื่อบุคคลจากอักษรไทยเป็นอักษรโรมันโดยอาศัยความนิยมในการใช้เป็นฐาน. (ROMANIZATION OF THAI PROPER NAMES BASED ON POPULARITY OF USAGE) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร. อติวงศ์ สุชาโต, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ผศ.ดร.โปรดปราน บุณยพุกกณะ, 68หน้า.

การขาดมาตรฐานในการถอดอักษรไทยเป็นอักษรโรมันในการเขียนชื่อบุคคลไทยอย่างเหมาะสมทำให้การค้นหาชื่อบุคคลเป็นเรื่องที่ท้าทาย การถอดชื่อของบุคคลอย่างถูกต้องจะเป็นส่วนสำคัญในการค้นหาเอกสารที่เป็นภาษาอังกฤษที่เกี่ยวข้องกับบุคคลนั้นจากชื่อของบุคคลที่สะกดด้วยตัวอักษรไทยเพียงอย่างเดียว แต่การถอดอักษรบนพื้นฐานจากการออกเสียงชื่อของบุคคลเหล่านั้นโดยตรงมักจะนำไปสู่ความผิดพลาดจากการสะกดชื่อด้วยอักษรโรมันคนละแบบกับที่เจ้าของใช้เนื่องจากการสะกดด้วยอักษรไทยกับอักษรโรมันไม่ได้สัมพันธ์กันแบบ 1 ต่อ 1 ทั้งยังมีความนิยมส่วนบุคคลเข้ามาเกี่ยวข้องอีกด้วย งานวิจัยนี้เสนอวิธีการถอดอักษรโดยพิจารณาความนิยมในการใช้เข้ามาเกี่ยวข้อง โดยการแบ่งชื่อบุคคลไทยเป็นสายลำดับของแกรมซึ่งเป็นหน่วยย่อยที่ลักษณะคล้ายพยางค์ที่มีการบังคับจากระบบการเขียนและการออกเสียงทั้งจากภาษาไทยและภาษาอังกฤษ รวบรวมนำมาสร้างเป็นพจนานุกรมแกรมสะสมจากชื่อบุคคลไทย 130,000 ชื่อ ใช้แบบจำลองทางสถิติเข้ามาช่วยในการฝึกฝนบนพื้นฐานของแกรม เมื่อเปรียบเทียบกับวิธีการที่ใช้เป็นฐานซึ่งให้ผลความถูกต้องของการถอดอักษร 18 % วิธีการนี้ให้ผลที่ดีกว่าโดยให้ความถูกต้องของการถอด 46% - 75 % ของชื่อบุคคลที่สะกดอักษรโรมันเมื่อจำนวนของตัวเลือกที่จะเป็นคำตอบมากขึ้นจาก 1 ถึง 15

ภาควิชา : .....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต : .....เอกพล ตั้งวีระพงษ์.....
สาขาวิชา : ....วิทยาศาสตร์คอมพิวเตอร์.ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก : .....อติวงศ์.....
ปีการศึกษา : 2551 ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์ร่วม.....

4971492721 : MAJOR COMPUTER SCIENCE

KEYWORDS: THAI ROMANIZATION/ PERSON NAME/ TRANSCRIPTION / ITIL

AKEGAPON TANGVERAPONG : ROMANIZATION OF THAI PROPER NAMES BASED ON POPULARITY OF USAGE. ADVISOR : ASST. PROF. ATIWONG SUCHATO, Ph.D., CO-ADVISOR : ASST. PROF. PROADPRAN P. PUNYABUKKANA, Ph.D., 68 pp.

The lack of standards for Romanization of Thai proper names makes searching activity a challenging task. This is particularly important when searching for people-related documents based on orthographic representation of their names using either solely Thai or English alphabets which is Roman based directly on the names' pronunciations often fails to deliver exact English spellings due to the non-1-to-1 mapping from Thai to English spelling and personal preferences. This paper proposes a Romanization approach where popularity of usages is taken into consideration. Thai names are parsed into sequences of grams, units of syllable-sized or larger governed by pronunciation and spelling constraints in both Thai and English writing systems. A Gram lexicon is constructed from a corpus of more than 130,000 names. Statistical models are trained accordingly based on the Gram lexicon. The proposed method significantly outperformed the current Romanization approach. Approximately 46% to 75% of the correct English spellings are covered when the number of proposed hypotheses increases from 1 to 15.

Department : .....Computer Engineering.. Student's Signature : ...เอกพล ตั้งวิริยะพงษ์.....

Field of Study : ...Computer Science..... Advisor's Signature : .....อติวงศ์.....

Academic Year : .2008..... Co-Advisor's Signature : .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์อย่างยิ่งของผศ.ดร.อติวงศ์ สุชาโต และ ผศ.ดร.โปรดปราน บุณยพุกกณะ อาจารย์ที่ปรึกษาทั้งสองท่านซึ่งได้ให้ความรู้ประสิทธิ์ประสาทวิชา แนะนำแนวทางการวิจัย ให้กำลังใจ และให้การสนับสนุนเป็นอย่างดี จนทำให้การวิจัยในครั้งนี้สำเร็จออกมาด้วยดี

ขอขอบพระคุณ รองศาสตราจารย์ นงลักษณ์ โควาวิสารัช และผู้ช่วยศาสตราจารย์ ดร. สุดาพร ลักษณียนาวิน กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบและแก้ไขวิทยานิพนธ์ฉบับนี้

ท้ายที่สุด ผู้วิจัยขอขอบคุณเพื่อน ๆ ทุก ๆ คน รวมทั้งครอบครัว เพื่อนร่วมงาน และผู้บังคับบัญชาในสายงาน ที่คอยติดตาม ให้กำลังใจและสนับสนุน รวมถึงท่านอื่น ๆ ที่มิได้กล่าวชื่อไว้ ณ ที่นี้ที่มีส่วนช่วยให้วิทยานิพนธ์สำเร็จได้ด้วยดี

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การถอดอักษรไทยเป็นโรมันในภาษาอังกฤษมีขึ้นเพื่อให้ชาวต่างประเทศหรือผู้ที่มีความรู้ในภาษาอังกฤษสามารถอ่านสายลำดับอักษรโรมัน แล้วออกเสียงได้คล้ายกับการออกเสียงที่คนไทยอ่านสายลำดับอักษรไทยที่ถอดไปเป็นสายลำดับอักษรโรมันนั้น นิยมใช้ในการสอนพูดภาษาไทยให้คนต่างประเทศ ใช้เขียนเนื้อเพลงในภาษาไทยให้คนต่างประเทศออกเสียงตามเนื้อร้องของไทยเพื่อความเพลิดเพลิน ดังนั้นการถอดอักษรด้วยจุดประสงค์ข้างต้นจึงค่อนข้างมีแบบแผนแน่นอนในแต่ละกลุ่มผู้ถอดเพื่อให้ผู้อ่านชาวต่างประเทศสามารถเรียนรู้และจดจำได้ง่าย แต่สำหรับการถอดอักษรในชื่อของคนไทยนั้นมีวัตถุประสงค์แตกต่างจากประเด็นข้างต้น เหตุเพราะการตั้งชื่อของบุคคลบางครั้งต้องการให้มีความเป็นเอกลักษณ์โดดเด่นเฉพาะตัวเพื่อให้แตกต่างจากบุคคลอื่น และเมื่อถอดชื่อของตนเองเป็นอักษรโรมันก็ยังต้องการให้มีลักษณะเฉพาะตัวนั้นยังคงอยู่ ประกอบกับการถอดจากอักษรไทยไปเป็นอักษรโรมันสามารถถอดได้หลายรูปแบบโดยที่แต่ละรูปแบบยังคงออกเสียงได้ใกล้เคียงกับชื่อในภาษาไทย ทั้งนี้ไม่ได้มีกฎแน่นอนในการถอดแม้จะมีหลักเกณฑ์การถอดอักษรไทยเป็นอักษรโรมันแบบถ่ายเสียง ตามประกาศราชบัณฑิตยสถาน ฉบับวันที่ 11 มกราคม พ.ศ.2542 ก็ตาม แต่ก็ยังมีหลักการถอดอื่นๆ ด้วย ดังนั้นการถอดชื่อเป็นอักษรโรมันของคนไทยนั้นจึงมีความหลากหลายตามความชอบและความนิยม และด้วยความนิยมที่แตกต่างกันในแต่ละกลุ่มบุคคลนี้เองทำให้เกิดความแตกต่างกันในการถอดชื่อบุคคลจากอักษรไทยเป็นโรมันแม้ว่าบุคคลนั้นจะมีชื่อภาษาไทยชื่อเดียวกันก็ตาม หรือในครอบครัวเดียวกันก็ยังปรากฏว่าถอดนามสกุลเป็นอักษรโรมันแตกต่างกันอีกด้วยทำให้เราไม่สามารถระบุได้ว่าชื่อที่สะกดด้วยอักษรโรมันในภาษาอังกฤษของแต่ละบุคคลสะกดอย่างไรได้อย่างชัดเจน

ในการถอดชื่อจากอักษรไทยเป็นอักษรโรมันจะถอดแบบถ่ายเสียงที่ละพยางค์ จากหลักภาษาไทยพยางค์ประกอบขึ้นด้วย พยัญชนะต้น สระ อาจจะมีวรรณยุกต์ และตัวสะกดร่วมอยู่ด้วย ในการถอดชื่อจากอักษรไทยเป็นอักษรโรมันจะมีหลักการถ่ายเสียงอยู่บ้าง เช่น พยัญชนะต้น "ก" อาจแปลงเป็นอักษร "k", "c", หรือ "g" ในอักษรโรมัน เมื่อรวมการถอดอักษรจากไทยเป็นโรมันของทุกอักษรในชื่อและนามสกุลก็จะได้จำนวนชื่อโรมันหลากหลายแบบมาก นอกจากนี้การถอดชื่อของบุคคลอาจไม่ถอดตามการเทียบเสียงแต่ถอดตามความนิยมเช่น "โรจ" ถอดเป็น"rote" จะเห็นได้ว่าตัวสะกด "จ" ไม่ได้ถอดเป็น "d" หรือ "t" แต่ถูกถอดเป็น "te" ซึ่งอาจออกเสียงได้เป็นอีก 1

พยางค์ตามความนิยม นอกจากนี้ยังมีการเขียนเพื่อคงรูปไว้จากภาษาบาลีและสันสกฤตซึ่งคำไทยนิยมยืมมาใช้ เช่น "มธุรยา" ถอดเป็น "madhuraya" สังเกตว่าพยางค์ "ธุ" เขียนเป็น "dhu" ซึ่งไม่ได้ออกเสียง ธุ

ด้วยการถอดชื่อเหมาะสมที่จะทำในหน่วยพยางค์จึงจำเป็นที่ต้องแบ่งชื่อของบุคคลออกเป็นพยางค์ก่อนการถอด แต่ด้วยในภาษาไทยมีการลดรูปสระเช่น เสียง "อะ" กึ่งเสียง การใช้พยัญชนะตัวเดียวเป็นทั้งพยัญชนะต้นและตัวสะกด และลักษณะอื่นๆ ทำให้การแบ่งชื่อในภาษาไทยออกเป็นพยางค์ไม่เหมาะสมที่จะนำมาถอดเป็นอักษรโรมัน จึงทำให้เกิดปัญหาการแบ่งชื่อบุคคลเป็นชิ้นส่วนที่เหมาะสมก่อนการนำไปถอดอักษร

การวิจัยที่ผ่านมาการถอดอักษรแบบถ่ายเสียงได้มีมาต่อเนื่องทั้งในภาษาต่างประเทศและในภาษาไทย ทั้งยังมีเครื่องมือที่ช่วยในการถอดอักษรไทยเป็นอักษรโรมันกับคำในบริบทั่วไป แต่ด้วยความที่ชื่อบุคคลเป็นชื่อเฉพาะจึงมีวิธีสะกดได้หลากหลาย ทั้งยังมีความนิยมในการสะกดให้มีความแตกต่างจากบุคคลอื่น ดังนั้นการถอดชื่อบุคคลเป็นอักษรโรมันจึงยิ่งมีความหลากหลายและมีความเฉพาะตัวมากขึ้นจนไม่สามารถหาหลักเกณฑ์ที่แน่นอนในการถอดชื่อบุคคลจากอักษรไทยเป็นอักษรโรมันได้ การใช้หลักทางสถิติเข้ามาช่วยน่าจะเป็นทางแก้ไขที่เหมาะสมในการแก้ไขปัญหาเรื่องความแตกต่างอันเกิดจากความนิยมในการถอดอักษรของชื่อบุคคลได้

ดังนั้นวิทยานิพนธ์นี้จึงศึกษาวิธีการถอดชื่อบุคคลจากอักษรไทยเป็นอักษรโรมันโดยอาศัยความนิยมในการใช้เป็นฐานเพื่อนำวิธีการดังกล่าวไปใช้เป็นคำค้นหาเอกสารที่มีชื่อคนไทยที่เขียนเป็นอักษรโรมัน

1.2 วัตถุประสงค์ของการวิจัย

เพื่อศึกษาและพัฒนาวิธีการถอดชื่อบุคคลที่เขียนด้วยอักษรไทยเป็นอักษรโรมันโดยอาศัยความนิยมในการใช้เป็นฐาน วิธีการนี้จะเป็นแนวทางในการพัฒนาโปรแกรมคอมพิวเตอร์ในการค้นหาชื่อคนไทยที่เขียนด้วยอักษรโรมันในเอกสารภาษาอังกฤษซึ่งมีระบบการเขียนที่ใช้อักษรโรมันเป็นฐาน

1.3 ขอบเขตของการวิจัย

1. ข้อมูลนำเข้าเป็นชื่อ และนามสกุลในภาษาไทยเท่านั้น มีวรรคตรงกลางระหว่างชื่อและนามสกุล โดยไม่รวมไปถึงชื่อของบุคคลต่างชาติแต่เขียนแบบไทย เช่น ชื่อของคนญี่ปุ่น หรือคนยุโรป
2. การแปลงชื่อเป็นอักษรโรมันจะไม่คำนึงถึงคำนำหน้าชื่อ

3. อักขระพิเศษที่นอกเหนือจากพยัญชนะไทย สระ และวรรณยุกต์จะไม่ถูกพิจารณา เช่น . ฯ ๎ ? ๋ เป็นต้น

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1. ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง
2. เตรียมฐานข้อมูลชื่อบุคคลที่สะกดด้วยอักษรไทย และอักษรโรมัน
3. สุ่มแยกชุดข้อมูลทดสอบออกจากชุดข้อมูลฝึก
4. แบ่งชื่อในชุดข้อมูลฝึกที่สะกดทั้งจากอักษรไทยและอักษรโรมันออกเป็นสายลำดับของแกรม ด้วยการสร้างโปรแกรมช่วยในการแบ่งแกรม
5. สร้างแบบจำลองภาษาจากชื่อภาษาอังกฤษในชุดข้อมูลฝึกที่แบ่งแกรมแล้ว
6. สร้างแบบจำลองการแปลวลีจากชื่อทั้งไทยและอักกฤษในชุดฝึกที่แบ่งแกรมแล้ว
7. ทดสอบผลความถูกต้องในการถอดชื่อบุคคล
8. สรุปและวิจารณ์ผลที่ได้
9. จัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. วิธีการถอดอักษรชื่อบุคคลจากอักษรไทยเป็นอักษรโรมันโดยอาศัยความนิยมในการใช้เป็นฐาน
2. เป็นแนวทางในการสร้างโปรแกรมคอมพิวเตอร์ในการถอดชื่อบุคคลไทยที่สะกดด้วยอักษรไทยเป็นอักษรโรมัน

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 5 บทดังต่อไปนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึง ความเป็นมาและความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องในงานวิจัยนี้ บทที่ 3 กล่าวถึงการดำเนินงานวิจัย บทที่ 4 เป็นการทดลองและผลที่ได้จากการทดลองตามชุดการทดลองต่างๆ และท้ายสุดคือบทที่ 5 กล่าวถึงสรุปผลการวิจัยและข้อเสนอแนะ

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตอบรับให้ตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “Romanization of Thai Proper Names Based On Popularity of Usages” โดย

Akegapon Tangverapong, Atiwong Suchato, Proadpran Punyabukkana นำเสนอในงานประชุมวิชาการ “The 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-09)” ณ โรงแรม เดอะ อิมพีเรียล ควีน พาร์ค กรุงเทพมหานคร ประเทศไทย ระหว่างวันที่ 27-30 เมษายน 2552 ตีพิมพ์ในวารสาร PAKDD 2009, LNAI 5476, pp. 580–587, 2009.Springer-Verlag Berlin Heidelberg, 2006

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 การใช้ตัวอักษรโรมันเพื่อการถ่ายเสียงหรือการเขียน (Romanization)

ในพจนานุกรมภาษาศาสตร์ ชื่อ “ A Dictionary of linguistics” Pie และ Gaynor [1] ได้ให้คำจำกัดความไว้ 2 ประการ คือ

1) Romanization คือ การเปลี่ยนระบบการเขียนที่ไม่ใช่อักษรโรมันให้อยู่ในรูปอักษรโรมัน เช่น การใช้อักษรโรมันเขียนแทนตัวอักษรญี่ปุ่น ซึ่งมีลักษณะเป็นอักษรภาพและเป็นอักษรแทนพยางค์ (ideograph and sysllabic)
2) Romanization คือ การถ่ายเสียงภาษาหนึ่ง ๆ มาเป็นอักษรโรมัน เช่น ถ่ายเสียงภาษาจีนด้วยอักษรโรมัน

เพื่อป้องกันความสับสนจะขอกล่าวถึงระบบใกล้เคียงที่เกี่ยวข้องกันอันได้แก่ ระบบการถ่ายเสียง (Transcription) ระบบการถอดถ่ายตัวอักษร (Transliteration) และ ระบบการใช้ตัวอักษรโรมันเพื่อการถ่ายเสียง หรือ การเขียน (Romanization)

1) ระบบการถ่ายเสียง [2] คือ นักสัทศาสตร์ได้ตั้งตัวอักษรขึ้นชุดหนึ่งเป็นสัทอักษรหรืออักษรแทนเสียง (Phonetic Alphabet) คือตัวอักษรซึ่งมีค่าเฉพาะในทางสัทศาสตร์ ตัวอักษรเหล่านั้นถูกกำหนดให้แสดงลักษณะของการออกเสียงมากกว่าจะแสดงเสียงที่ปรากฏในภาษาใดภาษาหนึ่ง ระบบการถ่ายเสียงเป็นระบบการบันทึกข้อมูลทางภาษาโดยใช้สัทอักษรเหล่านั้น เช่น เสียงกักเสียดแทรก เพดานแข็ง ไม่มีลักษณะพ่นลม อโฆษะ ถ่ายเสียงโดยตัวสัทอักษรซึ่งสมาคมสัทศาสตร์ระหว่างชาติ (The International Phonetic Association - IPA) ตั้งขึ้น คือ [/tɕ /] หรือถ่ายเสียงโดยใช้อักษรไทย คือ “จ” ระบบการถ่ายเสียงแบ่งออกเป็น 2 แบบใหญ่ๆ ได้แก่ การถ่ายเสียงอย่างแคบ (Narrow transcription) คือ บันทึกลักษณะความแตกต่างของเสียงทุกเสียงอย่างถูกต้องที่สุด และการถ่ายเสียงอย่างหยาบ (Broad transcription) คือ การบันทึกความแตกต่างของเสียงเฉพาะที่ทำให้สามารถแยกความหมายในภาษาได้เท่านั้น เช่น การบันทึกคำว่า แม้น จะบันทึกอย่างละเอียดว่า [mɛ́ːn] แต่เมื่อวิเคราะห์ดูแล้วพบว่าการที่ออกเสียงสระขึ้นจมูกหรือไม่ขึ้นจมูกก็ไม่มีความหมายแตกต่างกันแต่อย่างใดในภาษาไทย จึงบันทึกเสียงอย่างหยาบว่า [mɛ́ːn]

2) ระบบการถอดถ่ายตัวอักษร คือ ระบบการถอดถ่ายตัวอักษรภาษาหนึ่งด้วยอักษรของอีกภาษาหนึ่งแบบอักษรต่ออักษรโดยพยายามให้ได้เสียงใกล้เคียงกันที่สุด ตามที่ควรแล้วก็จะต้องอยู่ในลักษณะที่ได้เก็บเอามาครบถ้วนทุกตัวอักษรเท่าที่จะเป็นไปได้ เพื่อว่าจะได้เป็นร่องรอยให้ผู้อื่นถอดคืนไปได้โดยไม่ตกหล่นผิดเพี้ยน ซึ่งเป็นวิธีการที่ใช้ในการเขียนคำทับศัพท์ รูปที่ 1 แสดงตัวอย่างระบบการถอดถ่ายตัวอักษรอังกฤษเป็นอักษรไทยแบบอักษรต่ออักษร และรูปที่ 2 แสดงการระบบการถอดถ่ายตัวอักษรอังกฤษเป็นอักษรไทยแบบที่พยายามให้การถอดอักษรได้เสียงใกล้เคียงที่สุด

อักษรภาษาที่ 1	หน่วยเสียงภาษาที่ 1		หน่วยเสียงภาษาที่ 2	อักษรภาษาที่ 2
ภาษาอังกฤษ“b”	/b/	=	/b/	“บ”
“p”	/p/	=	/ph/	“พ”

รูปที่ 1 การถอดถ่ายตัวอักษรอังกฤษเป็นอักษรไทยแบบอักษรต่ออักษร

อักษรภาษาที่ 1	หน่วยเสียงภาษาที่ 1		หน่วยเสียงภาษาที่ 2	อักษรภาษาที่ 2
p →	/ph/	=	/ph/	“พ”
p ↘	/p/	=	/p/	“ป”

รูปที่ 2 การถอดถ่ายตัวอักษรอังกฤษเป็นอักษรไทยแบบพยายามให้ได้เสียงใกล้เคียงที่สุด

ราชบัณฑิตยสถาน (1968) ได้สร้างระบบมาตรฐานในการ ”ถ่ายเสียง” ตัวอักษรไทยให้เป็นตัวอักษรโรมัน 2 ระบบ คือ ระบบแม่นยำ (Precise system) และระบบทั่วไป (General System) ระบบแม่นยำมีอักษรโรมันหนึ่งตัวสำหรับเสียงพยัญชนะไทยหนึ่งๆ และมีการแสดงเสียงวรรณยุกต์ และแสดงตัวอักษรที่ไม่ออกเสียงในคำเพื่อแสดงที่มาของคำด้วย ระบบทั่วไป ถ่ายเฉพาะเสียงของคำเท่านั้น ไม่มีเครื่องหมายพิเศษใดๆ เพื่อแสดงเสียงวรรณยุกต์เลย เช่น อินทร์บุรี inburi อยุธยา ayutthaya และได้อธิบายเพิ่มเติมว่า ระบบการใช้ตัวอักษรโรมันเพื่อการ ”ถ่ายเสียง” ใช้ทั้งวิธีการ ”ถ่ายเสียง และการถอดถ่ายตัวอักษร”

ทั้งพจนานุกรมภาษาศาสตร์และราชบัณฑิต ให้ความหมายของระบบการใช้ตัวอักษรโรมันเพื่อการถ่ายเสียงหรือการเขียนที่คล้อยตามกัน แต่คำจำกัดความของพจนานุกรมภาษาศาสตร์ชัดเจนกว่า ในที่นี้จึงขอใช้คำจำกัดความของคำนี้ตามพจนานุกรมของ Pie และ Gaynor ดังนี้ ระบบการใช้ตัวอักษรเพื่อการถ่ายเสียงหรือการเขียน (Romanization) คือระบบการถ่ายเสียง หรือการถอดถ่ายตัวอักษรภาพที่ไม่ใช้อักษรโรมันให้อยู่ในรูปของอักษรโรมัน ผู้วิจัยขอเรียกว่าการถอดอักษร และในงานวิจัยนี้ไม่ทำการถอดอักษรที่ระดับพยางค์เนื่องจากมีข้อเสียบางประการดังที่จะได้

กล่าวในหัวข้อต่อไป และได้เสนอการแบ่งชื่อคนไทยออกเป็นหน่วยย่อยที่เหมาะแก่การถอดอักษรในหัวข้อที่ 3.1

เพื่อความเข้าใจในการถอดอักษรในหัวข้อต่อไปจะกล่าวถึง ลักษณะของอักษรในภาษาไทยที่ใช้ประกอบกันเป็นพยางค์ซึ่งเป็นหน่วยย่อยที่สุดที่สามารถทำการถอดอักษรได้ จากนั้นจะกล่าวถึงเสียงในภาษาไทยตามแบบของหลักสากลของอักษรที่ประกอบเป็นพยางค์เหล่านั้น

2.2 อักษรในภาษาไทย

ในภาษาไทยมีตัวอักษรที่ใช้แทนเสียง 3 ชนิด [3] ได้แก่ พยัญชนะ สระ และ วรรณยุกต์

2.2.1 พยัญชนะ

พยัญชนะไทยมีทั้งสิ้น 44 รูป ได้แก่ “ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ” มีลักษณะเป็นอักษรไตรยางศ์ได้แก่ อักษรกลาง อักษรสูง อักษรต่ำคู่ อักษรต่ำเดี่ยว โดยที่อักษรสูงและอักษรต่ำคู่เมื่อรวมกันจะผันวรรณยุกต์ได้ครบทั้ง 5 เสียง พยัญชนะทั้ง 44 รูปในภาษาไทยสามารถทำหน้าที่ได้ถึง 6 หน้าที่และออกเสียงต่างกันไปตามแต่ละหน้าที่ ได้แก่

1) เป็นพยัญชนะต้น ปัจจุบันยกเลิกไป 2 ตัว ได้แก่ ฃ ฅ
2) เป็นพยัญชนะท้ายพยางค์ (ตัวสะกด) เราอาจพบในบางคำที่เป็นทั้งพยัญชนะต้นและพยัญชนะสะกดเช่น อัตรา
3) เป็นอักษรควบ (ควบแท้และ ไม่แท้)
4) เป็นอักษรนำ-อักษรตาม การมีอักษรนำทำให้การแยกระหว่างอักษรควบ และอักษรนำได้ยากขึ้น เช่น ปลา กับ ปลาส
5) เป็นสระ ได้แก่ อ ร ว ย การที่พยัญชนะสามารถเป็นสระได้ทำให้เกิดความกำกวมขึ้นเพราะสามารถอ่านได้หลายรูปแบบ ยากต่อการหาขอบเขตคำ
6) เป็นตัวการันต์ ทั้งแบบการันต์ตัวเดียว และ 2 ตัว

โดยมากพยัญชนะไทยมักทำหน้าที่ได้มากกว่าหนึ่งหน้าที่ แต่ก็มีพยัญชนะไทยที่ทำหน้าที่เดียว คือ เป็นพยัญชนะต้น ได้แก่ “ฮ” หรือ พยัญชนะที่ทำได้ครบทั้ง 6 หน้าที่ได้แก่ “ร” นอกจากนี้พยัญชนะไทยตัวเดียวกันเมื่อทำหน้าที่แตกต่างกันในพยางค์ที่แตกต่างกัน เช่น “ข” ในชื่อ “ขจร” และ “ขันติ” เมื่อนำชื่อทั้งคู่ไปถอดอักษรก็จะได้อักษรอังกฤษจาก “ข” ที่ต่างกัน เช่น อาจจะได้ khajorn และ khanti ตามลำดับ จะสังเกตได้ว่า “ข” ในชื่อ “ขจร” ทำหน้าที่เป็นอักษรนำและออกกึ่งเสียงจึงมักถอดเป็นสายลำดับอักษรอังกฤษ “kha” แต่ “ข” ใน “ขันติ” เป็นพยัญชนะต้นจึงมักถอดเป็นสายลำดับอักษรอังกฤษ “kh”

2.2.2 สระ

รูปสระในภาษาไทย มี 32 รูป แบ่งได้เป็น 3 ประเภทดังนี้

1) รูปสระที่แทนเสียงสระเดี่ยว มี 18 รูป ได้แก่ –ะ –า ิ ี ึ ื ุ ู เ–ะ เ– แ–ะ แ– โ–ะ โ– เ–าะ –อ เ–อะ เ–อ
2) รูปสระที่แทนเสียงสระประสม มี 6 รูป ได้แก่ เ–ียะ เ–ีย เ–ือะ เ–ือ –ัวะ –ัว
3) รูปสระที่แทนเสียงสระเกิน มี 8 รูป ได้แก่ ฤ ฤๅ ฦ ฦๅ –ำ ใ– ไ– เ–า (สระเกิน คือสระที่มีเสียงพยัญชนะปนอยู่ เช่น –ำ ประสมจากสระ –ะ + ม

บางสระในภาษาไทยสามารถแสดงในรูปอื่นๆ ที่แตกต่างไปจากรูปเดิม สามารถแบ่งได้เป็น 4 ประเภท ได้แก่

1) รร (ร หัน) ใช้แทนเสียงสระอะ และเสียงพยัญชนะ “น” /a n/ เช่น กรรณ์
2) การใช้พยัญชนะประสมในรูปสระมี 3 ตัว ได้แก่ อ ว และ ย เช่น เสือ บัว เสีย
3) สระลดรูป เช่น ขืน (ลด ตัว อ) เพชร (ลดวิสรรชนีย์) นก (ลดรูปสระ โ-ะ)
4) สระเปลี่ยนรูป เช่น สังข์ (เปลี่ยน “-ะ” เป็น ”ั”), เป็ด (เปลี่ยนจาก “เ–ะ” เป็น “เ็- -” เมื่อมีพยัญชนะสะกด)

นอกจากนี้ยังมีกฎยกเว้นในการแสดงรูปสระเช่น สระ เ–อ เมื่อมีตัวสะกดจะเปลี่ยน “อ” เป็น “ิ” เช่น เดิม แต่ยกเว้นกรณีที่ตัวสะกดเป็น “ย” จะไม่เติม “ิ” เช่น เลย

2.2.3 วรรณยุกต์

วรรณยุกต์มี 4 รูป 5 เสียง ประโยชน์ของวรรณยุกต์ช่วยทำให้คำมีความหมายมากขึ้น แตกต่างจากภาษาของชาติอื่น ๆ เช่น ปา ป่า ป้า ป๊า ป๋า ในการถอดอักษรจะไม่พิจารณาวรรณยุกต์

2.3 เสียงในภาษาไทย

2.3.1 พยัญชนะ

เสียงของพยัญชนะไทยประกอบด้วย เสียงพยัญชนะต้นเดี่ยว 21 หน่วยเสียง [4] และเสียงพยัญชนะควบกล้ำ 12 หน่วยเสียง และพยัญชนะสะกด 8 หน่วยเสียง แม้ว่าพยัญชนะไทยมีลักษณะเป็นอักษรไตรยางค์คือ มีเสียงสูง เสียงกลาง และเสียงต่ำ การจำแนกนี้ทำให้เมื่อรวมกับเสียงสระจะมีเสียงวรรณยุกต์แตกต่างออกไป แต่เมื่อเราเทียบเสียงตามหลักสากลแล้วเราจะพบว่าอักษรเสียงสูง และเสียงต่ำจะเป็นเสียงพยัญชนะเดียวกัน อย่างเช่น “ข” อักษรสูง และ “ค” อักษรต่ำ จะแทนเสียง[k^h] และนิยมถอดเป็นอักษรอังกฤษ “kh” เหมือนกัน เสียงพยัญชนะต้นทั้ง 21 หน่วยเสียงแสดงได้ดังตารางที่ 1

ตารางที่ 1 เสียงและรูปพยัญชนะต้นเดี่ยวในภาษาไทย

หน่วยเสียง	**รูปพยัญชนะ**	**หน่วยเสียง**	**รูปพยัญชนะ**
/p/	ป	/m/	ม
/p^h/	พ ภ ผ	/n/	น ณ
/b/	บ	/ŋ/	ง
/t/	ต ฏ	/f/	ฟ ฝ
/t^h/	ท ธ ฒ ฑ ถ ฐ	/s/	ซ ศ ษ ส
/d/	ด ฎ ฑ	/h/	ฮ ห
/tɕ/	จ	/r/	ร ฤ
/$tɕ^h$/	ช ฌ ฉ	/l/	ล ฬ
/k/	ก	/w/	ว
/k^h/	ค ฆ ข	/j/	ย ญ
/?/	อ		

เสียงพยัญชนะสะกดในภาษาไทยมาตรฐานมี 8 หน่วยเสียงที่สามารถปรากฏในตำแหน่งท้ายพยางค์ได้ดังที่แสดงในตารางที่ 2 แต่ยังมีอักษรไทยบางตัวไม่สามารถปรากฏเป็นเสียงพยัญชนะสะกดได้ ได้แก่ ฝ ผ ฉ ฌ ห ฮ และ อ

ตารางที่ 2 เสียงและรูปพยัญชนะสะกดในภาษาไทย

หน่วยเสียง	รูปพยัญชนะสะกด
/p/	บ ป พ ภ ฟ
/t/	ด ฎ ต ฏ ท ธ ฒ ฑ ถ ฐ จ ช ซ ศ ษ ส
/k/	ก ค ฎ ข
/m/	ม
/n/	น ณ ร ล ฬ ญ
/ŋ/	ง
/w/	ว
/j/	ย

พยัญชนะควบกล้ำในภาษาไทยมาตรฐานมีหน่วยเสียง 12 หน่วยเสียง ดังแสดงในตารางที่ 3 โดยมีอักษร “ร” “ล” “ว” เป็นอักษรควบกล้ำกับพยัญชนะ “ก” “ค” “ต” “ท” “ป” และ “พ” นับเป็นพยัญชนะควบกล้ำแท้ เช่น “เกลือ” (และแบบที่ควบกล้ำไม่แท้ เช่น “ทราบ”)

ตารางที่ 3 เสียงและรูปพยัญชนะควบกล้ำในภาษาไทย

หน่วยเสียง	รูปพยัญชนะ	หน่วยเสียง	รูปพยัญชนะ
/pr/	ปร ปฤ	/ p^hr/	พร พฤ ภร ภฤ
/tr/	ตร ตฤ	/ t^hr/	ทร ทฤ
/kr/	กร กฤ	/k^hr/	คร ขร
/pl/	ปล	/p^hl/	พล ผล
/kl/	กล	/k^hl/	คล ขล
/kw/	กว	/k^hw/	คว ขว

2.3.2 สระ

ภาษาไทยมาตรฐานมีหน่วยเสียงสระทั้งหมด 21 หน่วยเสียง [4] หน่วยเสียงสระสั้น (Short Vowels) 9 หน่วยเสียง หน่วยเสียงสระยาว (Long Vowels) 9 หน่วยเสียง และหน่วยเสียงสระประสม (Diphthong Vowels) 3 หน่วยเสียง โดยหน่วยเสียงสระเดี่ยวทั้ง 18 หน่วยสามารถหาคู่เทียบเสียง (Minimal Pairs) ได้ และหน่วยเสียงสระประสมทั้ง 3 หน่วยเสียงไม่สามารถหาคู่เทียบเสียงได้ เสียงสระแสดงได้ดังตารางที่ 4

ตารางที่ 4 เสียงและรูปสระในภาษาไทย

	หน่วยเสียง	รูปสระ	หน่วยเสียง	รูปสระ
หน่วยเสียงสระเดี่ยว	/i/	◌ิ	/i:/	◌ี
	/ɯ/	◌ึ ฤ ฦ	/ɯ:/	◌ึ ฤๅ ฦๅ
	/u/	◌ุ	/u:/	◌ู
	/e/	เ–ะ เ็–	/e:/	เ-
	/ɤ/	เ–อะ เ◌ิ–	/ɤ:/	เ-อ เ–ย
	/o/	โ–ะ ––	/o:/	โ–
	/ɛ/	แ–ะ แ็–	/ɛ:/	แ–
	/ɔ/	เ–าะ	/ɔ:/	ออ

	หน่วยเสียง	รูปสระ	หน่วยเสียง	รูปสระ
	/a/	–ะ –รร– ั– ำ ไ- ใ- เ-า	/a:/	–า
หน่วยเสียงสระประสม	/ia/	เ–ียะ	/ia:/	เ–ีย
	/ɯa/	เ–ือะ	/ɯa:/	เ–ือ
	/ua/	–ัวะ	/ua:/	–ัว –ว–

การถอดอักษรต้องพิจารณาทั้งรูปและเสียงซึ่งในภาษาไทยมีทั้งคำพ้องรูป และคำพ้องเสียง เมื่อพิจารณาจะพบว่าความสัมพันธ์ของรูปอักษรไทยกับหน่วยเสียงในภาษาไทยไม่ได้มีลักษณะเป็น 1 ต่อ 1 แต่มีความสัมพันธ์เป็นแบบอักษร 1 รูปเป็นได้หลายหน่วยเสียง เช่น ในพยัญชนะ “ล” เมื่อเป็นพยัญชนะต้นจะมีเสียง /l/ เมื่อเป็นพยัญชนะสะกดมีเสียง /n/ และในทางกลับกัน หน่วยเสียง 1 หน่วยเสียงแทนด้วยอักษรได้หลายรูป เช่น เสียง /n/ ใช้แทนรูปพยัญชนะต้น “ณ” “น” และพยัญชนะสะกด “ร” “ล” “ฬ” “ณ” “น”

สำหรับสระในภาษาไทยซึ่งมีวิธีการเขียนทั้งแบบตรงรูป ลดรูป และเปลี่ยนรูป ทำให้มีความซับซ้อนมากกว่าพยัญชนะ เนื่องจากการเขียนในหลายๆ รูปให้เสียงสระเสียงเดียวกัน เช่น คำว่า กะ กัน กรรณ์ เสียงสระของทั้ง 3 คำนี้เป็นเสียงเดียวกัน คือ /a/ หรือ ในทางกลับกัน สระ 1 รูป อาจเป็นได้หลายเสียง พบมากในกรณีพยัญชนะที่ทำหน้าที่เป็นสระ เช่น คำว่า ตัว ตัด จะสังเกตได้ว่า –ั ในคำว่าตัวมีเสียง /ua:/ ส่วน –ั ในคำว่าตัดมีเสียง /a/ นอกจากนี้การที่สระสามารถลดรูปได้ทำให้มองเหมือนเป็นสระไม่ปรากฏรูปเหมือนกันแต่มีเสียงสระแตกต่างกัน เช่น ชื่อ อสมา (อะ-สะ-มา) อรชุน (ออ-ระ-ชุน) สังเกตได้ว่า ตัว “อ” ที่อยู่หน้าชื่อทั้งสองลดรูปสระทั้งคู่ แต่ออกเสียง /a/ และ /ɔ:/ ตามลำดับ การออกเสียงสระแตกต่างกันนอกจากเกิดขึ้นกับการลดรูปสระแล้วยังเกิดกับคำพ้องรูปด้วย เช่น ชื่อ ครคิต อาจออกเสียงได้เป็น คอน-คิด หรือ คอ-ระ-คิด ซึ่งมีความแต่กต่างกันทั้งเสียงและจำนวนพยางค์

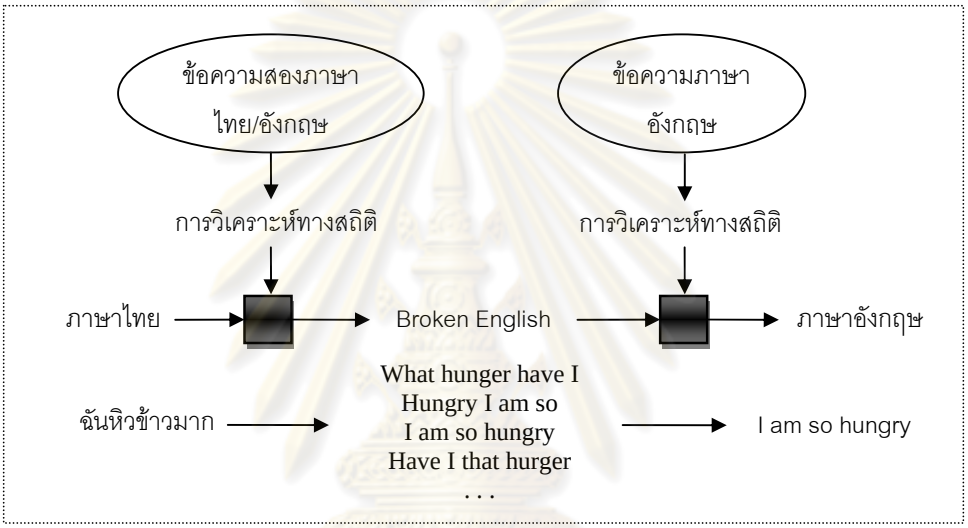
2.3.3 วรรณยุกต์

เสียงวรรณยุกต์แบ่งเป็น 2 ประเภท คือ เสียงวรรณยุกต์ระดับ (Level Tone) เป็นเสียงวรรณยุกต์ที่มีระดับความถี่ค่อนข้างคงที่ตลอดพยางค์มี 3 หน่วยเสียง ได้แก่ หน่วยเสียงสามัญ เอก ตรี และ เสียงวรรณยุกต์เปลี่ยนระดับ (Contour Tone) เป็นเสียงวรรณยุกต์ที่มีระดับความถี่ของการออกเสียงเปลี่ยนแปลงมากในช่วงพยางค์หนึ่งๆ เสียงวรรณยุกต์เปลี่ยนระดับมี 2 หน่วยเสียง คือ หน่วยเสียงวรรณยุกต์ โท และจัตวา

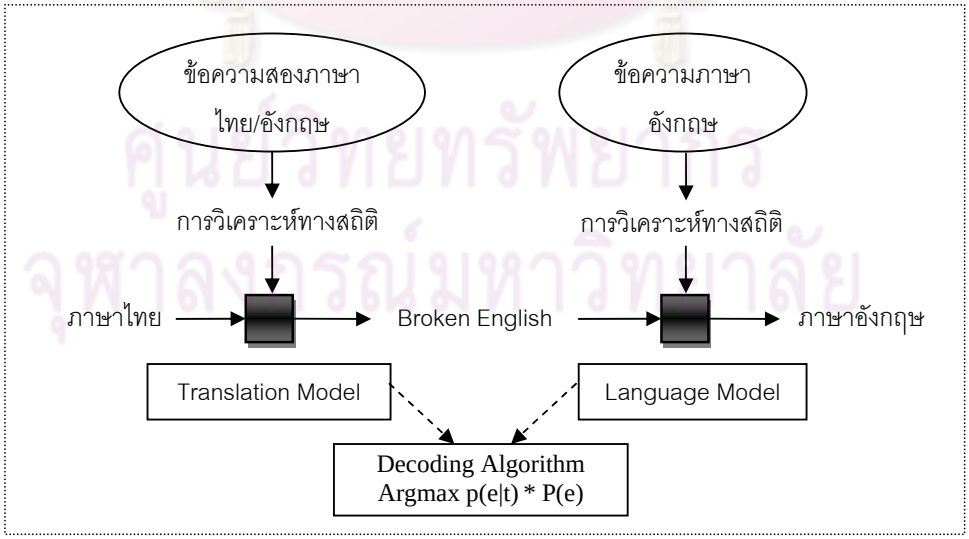
2.3 เครื่องแปลภาษาแบบอาศัยสถิติ (Statistical Machine Translation)

เครื่องแปลภาษาแบบใช้สถิติ (Statistical Machine Translation) [5][6] เป็นวิธีการทำให้คอมพิวเตอร์สามารถแปลภาษาได้ จากภาษาต้นทางเป็นภาษาปลายทางได้โดยวิธีการเรียนรู้จาก

ฐานข้อมูลขนาดใหญ่ วิธีการนี้คอมพิวเตอร์ไม่จำเป็นต้องเข้าใจลักษณะของภาษาอาทิเช่น ไวยากรณ์ หรือ ความหมายของทั้งต้นทางและภาษาปลายทาง ขั้นตอนการแปลประกอบด้วย 2 ขั้นตอนหลักดังที่แสดงในรูปที่ 3 ขั้นแรกเมื่อเราใส่ข้อความ “ฉันหิวข้าวมาก” เข้าไปที่กล่องสี่เหลี่ยมทางด้านซ้ายที่แสดงในรูปซึ่งเป็นเสมือนเครื่องสร้างข้อความภาษาอังกฤษจากข้อความภาษาไทยที่ใส่เข้าไป ข้อความภาษาอังกฤษที่ถูกสร้างขึ้นจากกล่องนี้ บางข้อความอาจไม่ถูกต้องตามไวยากรณ์หรือกระทั่งอาจแปลความหมายไม่ได้ ขั้นที่ 2 กล่องสี่เหลี่ยมทางขวาที่แสดงในรูปจะช่วยเลือกข้อความที่สมบูรณ์ที่สุดในบรรดาข้อความที่กล่องทางซ้ายสร้างขึ้น



รูปที่ 3 ภาพรวมการทำงานของเครื่องจักรแปลภาษา



รูปที่ 4 ภาพความสัมพันธ์ของแบบจำลองการแปล (Translation Model), แบบจำลองภาษา (Language Model) และวิธีการถอดรหัส (Decoding Algorithm)

จากรูปที่ 4 กล่องสี่เหลี่ยมทางซ้ายคือแบบจำลองการแปล (Translation Model) สร้างมาจากการสกัดข้อมูลจากฐานข้อมูลสองภาษาที่มีข้อความภาษาต้นทางและข้อความภาษาปลายทางคู่กัน แบบจำลองนี้จะให้คะแนนความสัมพันธ์กันของข้อความภาษาอังกฤษแต่ละข้อความกับข้อความภาษาไทย กล่องสี่เหลี่ยมทางขวามือคือแบบจำลองภาษา (Language Model) สร้างมาจากข้อความภาษาอังกฤษเพียงอย่างเดียว แบบจำลองนี้จะให้คะแนนความถูกต้องของข้อความภาษาอังกฤษ นอกจากแบบจำลองทั้ง 2 แล้วเครื่องแปลภาษาต้องมีวิธีการถอดรหัส (Decoding Algorithm) ข้อความภาษาต้นทางที่ใส่เข้ามาให้เป็นภาษาปลายทาง

ดังนั้นเครื่องแปลภาษาจะมีปัญหาทั่วไป 3 ข้อได้แก่

1) แบบจำลองภาษา (Language Model)
   - กำหนดข้อความภาษาอังกฤษ *e* หาค่าความน่าจะเป็น *P*(*e*) ได้จากสูตร
   - ข้อความภาษาอังกฤษที่ดี => *P*(*e*) สูง
   - ข้อความภาษาอังกฤษที่ไม่ดี => *P*(*e*) ต่ำ
2) แบบจำลองการแปลภาษา (Translation Model)
   - กำหนดคู่ของข้อความ <*t*,*e*> หาค่าความน่าจะเป็น *P*(*e*|*t*) ได้จากสูตร
   - <*t*,*e*> ที่แปลได้สัมพันธ์กันดี => *P*(*f*|*e*) สูง
   - <*t*,*e*> ที่แปลได้สัมพันธ์กันไม่ดี => *P*(*f*|*e*) ต่ำ
3) วิธีการถอดรหัส (Decoding Algorithm)
   - กำหนดแบบจำลองภาษา แบบจำลองการแปล และข้อความภาษาไทยใหม่ สามารถหาข้อความภาษาอังกฤษที่มีคะแนนของ *P*(*e*) * *P*(*e*|*t*)

2.4 แบบจำลองภาษา (Language Model)

แบบจำลองภาษา [7] คือ แบบจำลองที่จะสามารถบอกเราได้ว่าประโยคหรือสายลำดับของคำใดๆ มีความเป็นไปได้ที่จะเกิดขึ้นในภาษา หรือไม่ อาทิเช่น สายลำดับ "จะ ไป" มีโอกาสเกิดขึ้นได้แต่ สายลำดับ "ไป จะ" ไม่สามารถเกิดขึ้นได้ เป็นต้น แบบจำลองภาษาอาจจะบอกเป็นค่าความน่าจะเป็นที่จะเกิดประโยค W เรียกความน่าจะเป็นนี้ว่า $P(W)$ เช่นให้ค่า P(จะ ไป) = 0.8 และ P(ไป จะ) = 0.01 การสร้างแบบจำลองภาษาที่สามารถบอกค่าความน่าจะเป็นได้นี้สร้างมาจากแบบจำลองเอ็นแกรม (n-gram)

2.4.1 การนับคำในฐานข้อมูล

เมื่อกล่าวถึงความน่าจะเป็น เราจำเป็นที่จะต้องระบุถึงสิ่งที่เราจะนับและตำแหน่งที่เราจะพบมัน ในภาษาไทยเราไม่มีการเปลี่ยนรูปแบบคำ (Word Form) ออกมาเป็นเลมมา (Lemma)

เหมือนในภาษาอังกฤษ ในงานวิจัยนี้แบ่งชื่อบุคคลออกเป็นแกรม ดังนั้นประเภทของคำ (Type) และจำนวนของคำ (Tokens) ในงานวิจัยจะมีจำนวนเท่ากัน คือ นับตามจำนวนแกรม

2.4.2 เอ็นแกรมสามัญ (Simple N-Gram)

เอ็นแกรมทำงานโดยเมื่อมีประโยค $W = w_1 ... w_n$ (ในงานวิจัยนี้คือชื่อภาษาอังกฤษที่แบ่งแล้ว) เราสามารถคำนวนความน่าจะเป็นที่จะเกิดปะโยค W นี้ได้จาก $P(w_1, w_2 ..., w_{n-1}, w_n)$ ด้วยการใช้กฎลูกโซ่ของความน่าจะเป็นทำให้แยกคำนวณค่าได้โดย

$$P(w_1^n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2)...P(w_n \mid w_1^{n-1})$$
$$= \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

โดยที่ $P(w_1^n)$ คือ ความน่าจะเป็นของสายลำดับ w_1 ถึง w_n

แต่เนื่องจากความยากในการหา $P(w_k \mid w_1^{k-1})$ ทำให้เราประมาณค่าความน่าจะเป็นของประโยคไม่ได้ ดังนั้นแบบจำลองเอ็นแกรมจะคำนวณค่าความน่าจะเป็นของการเกิดคำใดๆ โดยพิจารณาจาก N-1 คำก่อนหน้า อาทิเช่น N = 2 ซึ่งเรียกว่า ไบแกรม (Bigram หรือ 2-gram) นั้น จะให้ค่าความน่าจะเป็นของ คำใดๆ โดยดูจากคำก่อนหน้าเพียงคำเดียว เรามักจะเขียนค่าความน่าจะเป็นนี้ในรูปของ $P(w_2 \mid w_1)$ ซึ่งหมายถึงความน่าจะเป็นที่จะพบคำ w_2 จะตามหลังคำ w_1 เมื่อรวมทั้งประโยคเราจะสามารถคำนวณความน่าจะเป็นได้โดย

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k \mid w_{k-1})$$

ตัวอย่างอาทิเช่น

กำหนดให้

ตารางที่ 5 ตัวอย่างค่าความน่าจะเป็นของไบแกรม

P(ไป|จะ) = 0.8	P(จะ|ไป) = 0.01	P(จะ|ผม) = 0.7
P(ตลาด|ไป) = 0.5	P(โรงเรียน|ไป) = 0.6	P(ผม|ไป) = 0.02

เมื่อมีประโยคยาวๆ W = (w1 ... w4) ค่าความน่าจะเป็นที่จะเกิดประโยคดังกล่าวก็สามารถคำนวณได้โดยการคูณต่อๆ กันไปดังนี้

P(ผม จะ ไป โรงเรียน) = P(จะ|ผม) * P(ไป|จะ) * P(โรงเรียน|ไป)
= (0.7)(0.8)(0.6) = 0.336

P(จะ ไป ผม จะ) = P(ไป|จะ) * P(ผม|ไป) * P(จะ|ผม)
= (0.8)(0.02)(0.7) = 0.0112

จะพบว่า "จะ ไป ผม จะ" มีโอกาสเกิดต่ำมากเมื่อเทียบกับ "ผม จะ ไป โรงเรียน" ในทำนองเดียวกัน เราสามารถสร้างแบบจำลองไตรแกรม (3-gram model) โดยที่กำหนดค่าความน่าจะเป็นของแต่ละคำ โดยพิจารณาจากคำก่อนหน้า 2 คำ

อาทิเช่น

P(ผม จะ ไป โรงเรียน) = P(จะ|ผม) * P(ไป|ผม จะ) * P(โรงเรียน|จะ ไป)

เพราะฉะนั้นสำหรับแบบจำลองเอ็นแกรมสิ่งที่จะต้องคำนวณเตรียมเอาไว้ก็คือ P(w2| w1) สำหรับไบแกรม หรือ P(w3| w1,w2) สำหรับไตรแกรม เราสามารถฝึกไบแกรมด้วยวิธีการนับค่าและปรับส่วนบรรทัดฐาน (normalizing) ได้ด้วยชุดข้อมูลฝึก

โดย $$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1})}$$

และสำหรับรูปเอ็นแกรม $$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}, w_n)}{C(w_{n-N+1}^{n-1})}$$

$C(w_{n-1} \mid w_n)$ คือจำนวนครั้งในชุดข้อมูลฝึกที่เกิด w_{n-1} คู่กับ w_n และ $C(w_{n-1})$ ก็คือจำนวนครั้งที่เกิด w_{n-1}

2.4.3 **การทำให้ราบเรียบ** (Smoothing)

ปัญหาสำคัญของแบบจำลองเอ็นแกรมมาตรฐานนั้น คือ ต้องได้รับการฝึกจากฐานข้อมูลซึ่งมีขนาดจำกัด บางประโยคสามารถเกิดขึ้นได้ในภาษาที่สมบูรณ์แบบแต่อาจจะไม่มีอยู่ในฐานข้อมูล ในความเป็นจริงแทบจะเป็นไปไม่ได้ที่ชุดข้อมูลฝึกจะมีคำเกิดขึ้นครบทุกคู่เพื่อใช้คำนวณค่าไบแกรมได้ และถ้าคำคู่ไหนไม่เกิดขึ้นในชุดข้อมูลฝึก เช่น C(ไป,โรงเรียน) = 0 ก็จะทำให้ P(โรงเรียน|ไป) = 0 และ P(ผม จะ ไป โรงเรียน) ก็จะเท่ากับ 0 ทำให้เกิดปัญหาขึ้นเนื่องจากไม่มีข้อมูลในชุดข้อมูลฝึกซึ่งไม่ได้หมายความว่าจะไม่มีโอกาสเกิดขึ้น ในการแก้ปัญหานี้เราเรียกว่าการปรับเรียบ (Smoothing) ซึ่งมีด้วยกันหลายวิธี

การทำให้ราบเรียบโดยการเพิ่ม 1 (Add-One Smoothing) คือ ในช่วงที่เราสร้างตารางนับไบแกรมนั้น ก่อนที่เราจะปรับค่าบรรทัดฐาน (normalization) ให้เป็นความน่าจะเป็น เราจะเพิ่มหน่วยนับขึ้นอีก 1 ให้กับทุกหน่วยซึ่งจะทำให้เราหาความน่าจะเป็นเกิดขึ้นกับทุกคู่ของไบแกรมได้ แม้ว่าวิธีการนี้จะให้ผลที่ไม่ดีนักและก็ไม่เป็นที่นิยมแต่ก็ช่วยให้พอเห็นมุมมองของการปรับเรียบ การคำนวณค่าความน่าจะเป็นของคำแต่ละคู่ในไบแกรมเมื่อปรับเรียบด้วยวิธีนี้จะกลายเป็น

$$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}, w_n) + 1}{C(w_{n-1}) + V}$$ เมื่อ V คือประเภทของคำ (type)

การปรับเรียบยังมีอีกหลายวิธีการที่นิยมกว่า เช่น การทำให้ราบเรียบโดยการลดแบบวิทเท็นเบล (Witten-Bell Discounting) มีแนวคิดในการใช้จำนวนของสิ่งที่เคยพบครั้งหนึ่งในการประมาณจำนวนของสิ่งที่ยังไม่เคยพบ หรือ การทำให้ราบเรียบด้วยวิธีการลดแบบกูดทูริง (Good-Turing Discounting) มีแนวคิดในการปรับเรียบค่าหน่วยนับที่เป็น 0 หรือมีจำนวนน้อยๆ จากการสังเกตุจำนวนหน่วยนับที่มีค่ามากกว่า

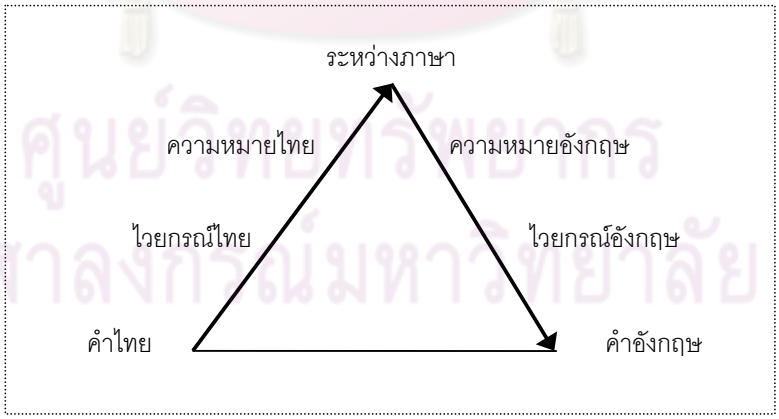
2.4.4 การทำให้ราบเรียบย้อน (Back-off Smoothing)

ด้วยวิธีการปรับเรียบสามารถช่วยเราแก้ปัญหาความถี่ 0 ในเอ็นแกรมได้แต่เรายังมีอีกวิธีการที่เข้ามาช่วยได้อีก ตัวอย่างเช่นในกรณีที่เราไม่มีตัวอย่างของบางไตรแกรม $w_{n-2}w_{n-1}w_n$ ในการคำนวณค่า $p(w_n \mid w_{n-1}w_{n-2})$ ดังนั้นเราจึงพยายามประมาณด้วยความน่าจะเป็นของไบแกรม $p(w_n \mid w_{n-1})$ และเช่นกันเมื่อเรายังไม่สามารถหาได้เราก็ประมาณด้วยยูนิแกรม (unigram) $p(w_n)$ ดังนั้นในการคำนวนความน่าจะเป็นของแบบจำลองไตรแกรมจะมีลักษณะ

$$P(w_i \mid w_{i-2}w_{i-1}) = \begin{cases} P(w_i \mid w_{i-2}w_{i-1}) & ,if C(w_{i-2}w_{i-1}w_i) > 0 \\ \alpha_1 P(w_i \mid w_{i-1}) & ,if C(w_{i-2}w_{i-1}w_i) = 0 and C(w_{i-1}w_i) > 0 \\ \alpha_2 P(w) & ,other \end{cases}$$

α_1 และ α_2 คือ ค่าน้ำหนัก (Back-off weight) ซึ่งขึ้นกับอัลกอริทึมที่เลือกใช้ในการทำให้ราบเรียบย้อน (Back-off Smoothing) อาทิเช่นวิธีกูดทูริง (Good-Turing)

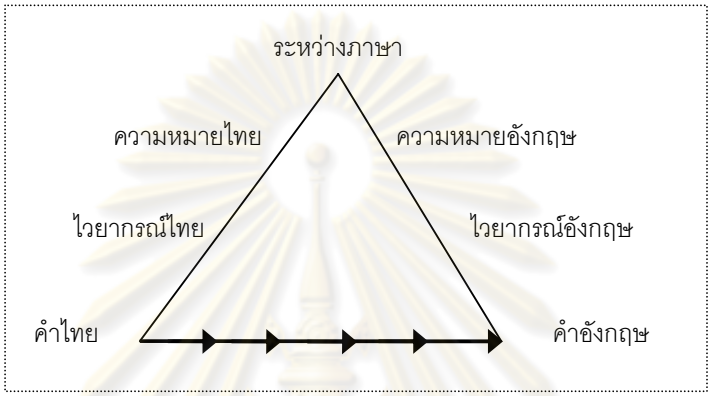
2.5 แบบจำลองการแปลวลี (Phrase Based Translation Model)



รูปที่ 5 ภาพวิธีการแปลภาษาของมนุษย์

รูปที่ 5 แสดงวิธีการแปลภาษาของมนุษย์ [5] เริ่มจากมีคำไทยนำมาเรียงต่อกันตามโครงสร้างไวยกรณ์ไทย แล้วสามารถแปลความหมายตามแบบของไทยออกมาได้ จากนั้นจึงเทียบ

ไปเป็นความหมายภาษาอังกฤษแล้วจึงสร้างโครงสร้างไวยากรณ์ภาษาอังกฤษแล้วแปลออกมาเป็นคำในภาษาอังกฤษ วิธีการแปลแบบนี้แตกต่างจากการแปลภาษาของเครื่องแปลภาษาที่ได้แสดงในรูปที่ 6 คือ แปลจากคำไทยไปเป็นคำอังกฤษตรงๆ โดยไม่สนใจไวยากรณ์และความหมายของทั้ง 2 ภาษาเนื่องจากถูกจำกัดโดยประสิทธิภาพของความสามารถทางสถิติ แต่ก็อย่างไรก็ตามยังได้มีการพัฒนาการแปลจากระดับของคำมาเป็นระดับของวลี

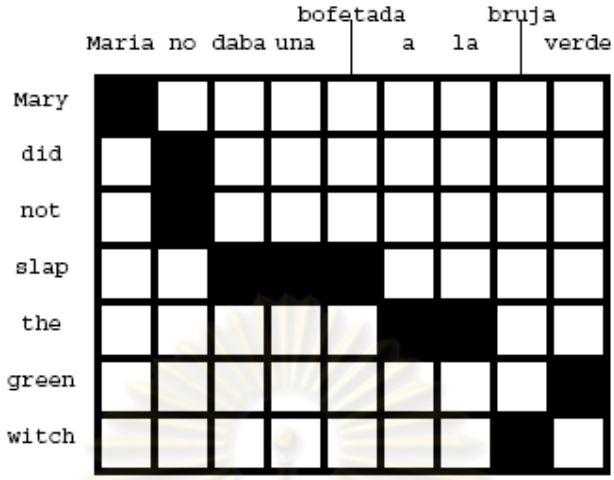


รูปที่ 6 ภาพวิธีการแปลภาษาของเครื่องแปลภาษา

แบบจำลองการแปล [5] (Translation Model) จะให้ค่าความน่าจะเป็นของการแปลข้อความ p(*e*|*t*) จาก

$$P(f \mid e) = \frac{count(f, e)}{count(e)}$$

สมการอุดมคตินี้ไม่สามารถหาค่าได้เนื่องจากเราไม่สามารถหาข้อความทั้งหมดทุกข้อความที่มีอยู่ได้ ดังนั้นเราจึงต้องแยกออกเป็นส่วนเล็กๆ และให้มีการจับคู่กันระหว่างแต่ละคำในข้อความ ให้การจับคู่ *a* เราจะสามารถหา $P(f \mid e) = \sum_{a} p(a, f \mid e)$ โดยเราสามารถจิตนาการของลักษณะการจับคู่ได้ดังรูปที่ 7 ช่องสีดำแสดงถึงการจับคู่กันระหว่างคำภาษาอังกฤษในแนวตั้งและคำภาษาฝรั่งเศสในแนวนอน



รูปที่ 7 รูปแสดงตารางการจับคู่ข้อความในภาษาฝรั่งเศสกับภาษาอังกฤษ (จาก Knight [5])

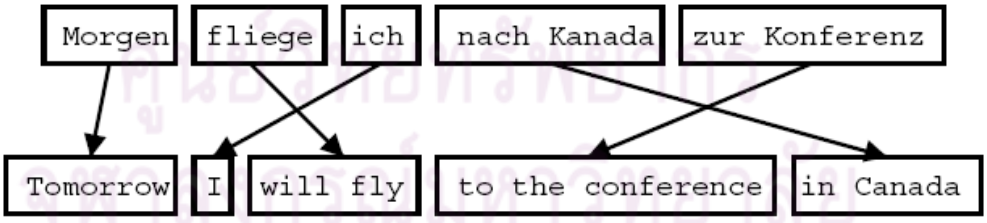
ณ ตอนนี้เราสามารถคำนวณสมการที่ 2 ได้โดยเราต้องการนิยามค่าของ p(a, f|e)

$$P(a, f \mid e) = \prod_{j=1}^{M} t(f_j \mid e_i)$$

โดยที่เราสามารถหาค่า $t(f_j \mid e_i)$ ได้จากในตารางการจับคู่โดย

$$t(f_j \mid e_i) = \frac{count(f_j, e_i)}{count(e_i)}$$

สำหรับการแปลแบบวลีนั้นเราจะไม่แบ่งการจับคู่ในระดับคำอย่างที่นำเสนอก่อนหน้า แต่จะมีลักษณะการจับคู่ของวลีกับวลีดังรูปที่ 8 วิธีการจับแบบนี้ช่วยแก้ปัญหาในหลายเรื่อง เช่น สามารถทำให้แปลได้ทั้งไปและกลับแบบจำนวนมากไปจำนวนมาก (many to many) ของภาษาต้นทางและภาษาปลายทาง การสลับที่กันของวลี และสามารถเรียนรู้วลียาวๆ ได้



รูปที่ 8 แสดงการจับคู่กันของวลีภาษาฝรั่งเศษกับวลีภาษาอังกฤษ (จาก Knight [5])

การจับคู่กันของวลีในตารางจับคู่จะมีลักษณะซับซ้อนกว่าการจับคู่ในระดับคำ รูปที่ 9 แสดงการจับคู่วลีของข้อความภาษาฝรั่งเศษ " Maria no daba una bofetada a la bruja verde" กับข้อความภาษาอังกฤษ “Mary did not slap the green witch” แบ่งเป็นรอบทั้งสิ้น 5 รอบ โดยการจับคู่ที่ได้เพิ่มขึ้นในแต่ละรอบ ได้แก่

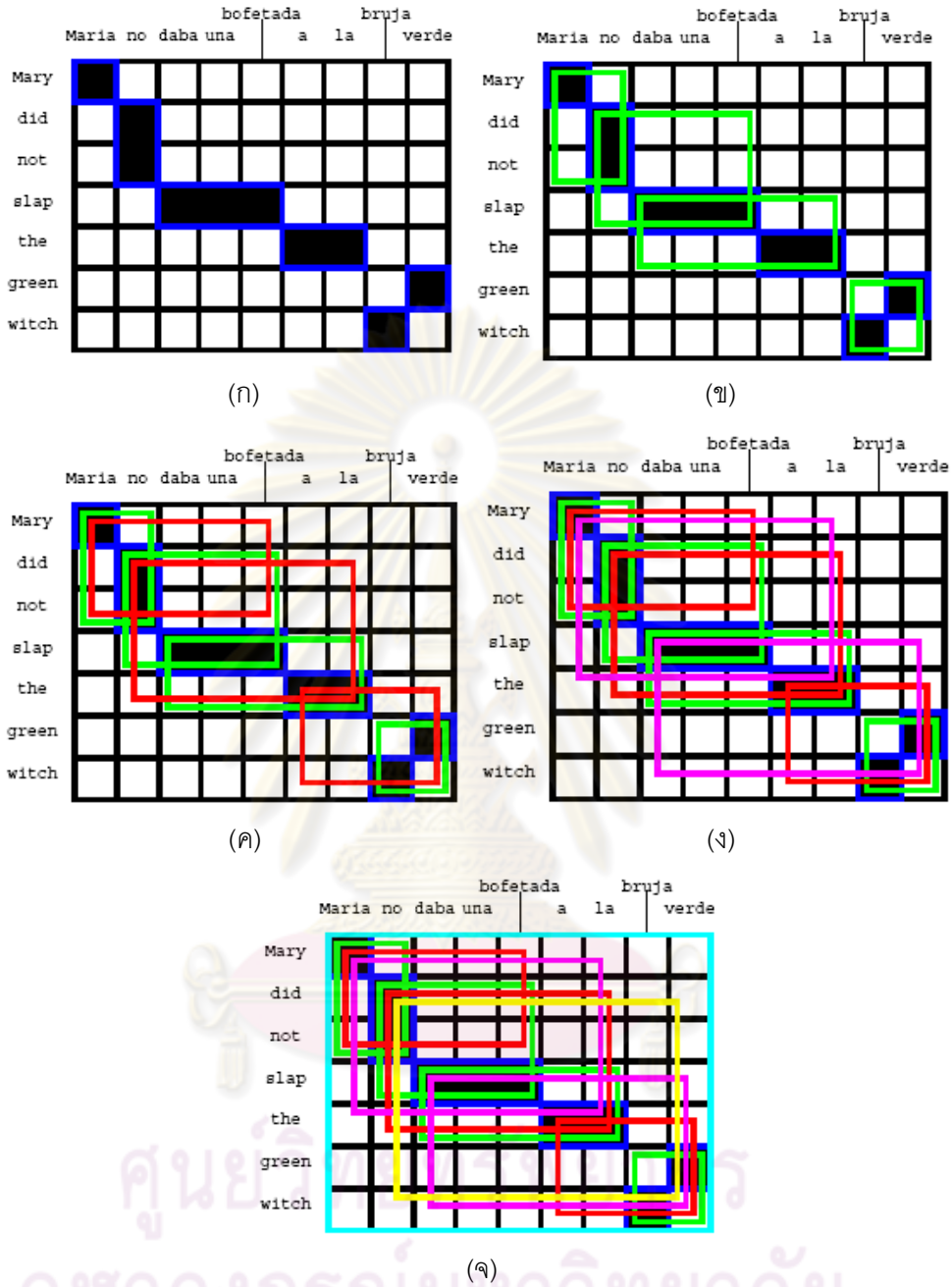
รอบที่ 1 (Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green)

รอบที่ 2 (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch)

รอบที่ 3 (Maria no daba una bofetada, Mary did not slap), (no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch)

รอบที่ 4 (Maria no daba una bofetada a la, Mary did not slap the),(daba una bofetada a la bruja verde, slap the green witch)

รอบที่ 5 (no daba una bofetada a la bruja verde, did not slap the green witch), (Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)



รูปที่ 9 แสดงขั้นตอนการจับคู่ระหว่างวลีภาษาฝรั่งเศษกับวลีภาษาอังกฤษ (จาก Knight [5])

เมื่อเราได้การจับคู่ของวลีได้แล้ว เราสามารถกำหนดคู่วลีที่มีอยู่ในตารางและหาค่าความน่าจะเป็นของคู่วลีนั้นได้จาก

$$\phi(\bar{f} \mid \bar{e}) = \frac{count(\bar{f}, \bar{e})}{\sum_{f} count(\bar{f} \mid \bar{e})}$$

2.7 งานวิจัยที่เกี่ยวข้อง

การถอดอักษรนั้นแบ่งขั้นตอนการทำออกเป็น 2 ขั้น คือ การแบ่งข้อความออกเป็นหน่วยย่อย และ การแปลงหน่วยย่อยให้เป็นระบบการเขียนของภาษาปลายทาง ในขั้นการแบ่งข้อความนั้นเป็นการแบ่งข้อความเริ่มต้นออกเป็นสายลำดับของหน่วยย่อย โดยหน่วยย่อยอาจจะเป็นพยางค์ คำ วลี หรือเป็นหน่วยย่อยพื้นฐานในการสะกดอื่นๆ ทั้งนี้เพื่อให้สัมพันธ์กับภาษาและวิธีการนำหน่วยย่อยที่สะกดด้วยตัวอักษรของภาษาต้นทางเหล่านี้ไปแปลงให้อยู่ในระบบการเขียนของภาษาปลายทางในขั้นตอนที่ 2

2.7.1 งานวิจัยเกี่ยวกับการแบ่งข้อความภาษาไทย

ในภาษาไทยได้มีงานวิจัยเกี่ยวข้องกับการแบ่งข้อความออกเป็นคำ และพยางค์หลายงาน ได้แก่ Poovarawan [8] ได้เสนอวิธีการตัดคำด้วยวิธีการจับคู่คำที่ยาว (Longest Matching) ที่สุดก่อนแบบอ่านจากซ้ายไปขวาโดยอาศัยพจนานุกรมช่วยในการแบ่งคำไทย เนื่องจากในงานนี้ใช้อัลกอริทึมเชิงละโมล (Greedy Algorithm) จึงเกิดข้อผิดพลาดได้บ่อย เช่น “ไปหามเหสี” จะแบ่งได้เป็น “ไป-หาม-เห-สี” แทนที่ควรจะเป็น “ไป-หา-มเหสี” เนื่องจากว่าเราจะพบคำว่า “หาม” ก่อนคำว่า “หา” เสมอ Sornlertlamvanich [9] ได้เสนอวิธีการตัดคำให้ได้จำนวนคำและคำที่ไม่มีในพจนานุกรมน้อยที่สุด (Maximal Matching) เป็นการแก้ไขปัญหาของวิธีจับคู่คำที่ยาวที่สุด วิธีการที่ 2 นี้ให้ผลดีกว่าวิธีการแรก แต่มีข้อด้อยคือไม่สามารถแก้ปัญหาในกรณีการแบ่งที่ทำให้เกิดจำนวนคำเท่ากันได้และยังคงต้องพึ่งพจนานุกรม ซึ่งการใช้พจนานุกรมเข้ามาช่วยในการแบ่งคำนั้นข้อด้อยที่สำคัญคือปัญหาการไม่มีคำบางคำอยู่ในพจนานุกรม และก็จะเป็นการยากมากที่เราจะสามารถทำให้พจนานุกรมของเรามีคำได้คลอบคลุมภาษาและพอเพียงในการนำไปใช้ตัดคำ และจะยากมากถ้าต้องพบกับคำที่เป็นชื่อเฉพาะ Theeramunkong [10] ได้เสนอวิธีการใช้ต้นไม้ตัดสินใจโดยไม่พึ่งพจนานุกรม ด้วยวิธีการนี้ช่วยให้แก้ปัญหาคำที่ไม่มีอยู่ในพจนานุกรมได้ แต่ก็ยังคงไม่สามารถแก้ปัญหาความกำกวมของคำได้

งานของ Aroonmanakun [11] การแบ่งพยางค์ด้วยวิธีการทางสถิติช่วยให้สามารถแก้ไขปัญหาความกำกวม โดยใช้แบบจำลองภาษาแบบไตรแกรมกับฐานข้อมูลแบบแบ่งพยางค์ (syllable segmentation corpus) ได้มีการนิยามรูปแบบของพยางค์ที่เป็นไปได้ทั้งหมดในภาษาไทย 200 แบบ เช่น CRTาะ CVTอะ CRTY โดยที่ X C R Y T เป็นสัญลักษณ์แทนอักษรในแต่ละกลุ่ม ผลในการจับจับคู่รูปแบบกับพยางค์ในขั้นตอนนี้ยังมีความกำกวมอยู่ เช่น "กรรมการกรมพลศึกษารอยกร่าง" สามารถมีรูปแบบที่จับคู่ได้ถึง 36 ประเภทแต่ด้วยการนำไตรแกรมเข้ามาช่วยทำให้แบ่งพยางค์ได้เหลือเพียง "กรรม-การ-กรม-พล-ศึก-ษา-รอ-ยก-ร่าง" จากการศึกษาใช้ฐานข้อมูลขนาด 553,372 พยางค์ ใช้วิธีการทำให้ราบเรียบวิทเทนเบลล์ใช้เวอติบิอัลกอริทึม

(Viterbi algorithm) ในการตัดสินการแบ่งคำที่ดีที่สุดทดสอบบทฐานข้อมูลอื่นที่มีขนาด 30,498 พยางค์ พบ 52 ข้อผิดพลาดของการแบ่งดังนั้นผลการทดลองถูกต้อง 99.8% ซึ่งมี 22 จาก 52 ข้อผิดพลาดมาจากชื่อและคำในภาษาต่างชาติที่เขียนแบบไทย วิธีการนี้ยังมีข้อผิดพลาดเกิดขึ้นบ้างเมื่อพบคำที่ไม่อยู่ในชุดข้อมูลฝึก ในงานวิจัยนี้ได้นำวิธีการแบ่งพยางค์ของ Aroonmanakun [11] เข้ามาช่วยในการเริ่มต้นแบ่งชื่อบุคคลที่สะกดด้วยอักษรไทยออกเป็นสายลำดับของพยางค์ด้วย แต่เนื่องจากวิธีการนี้ไม่ได้สร้างขึ้นมาให้รองรับการถอดอักษร ตัวอย่างเช่น เมื่อข้อความในภาษาอังกฤษจงใจที่จะสะกดตัวอักษรในรูปแบบที่รักษาลักษณะทางภาษาเอาไว้มากกว่าที่จะสะกดให้ออกเสียงตามคำอ่าน เช่น “โชติ” ที่ออกเสียงพยางค์เดียว แต่โดยมากจะถอดอักษร (transliteration) เป็นอักษรภาษาอังกฤษที่ออกเสียง 2 พยางค์ “choti” ซึ่งอาจเป็นได้ใน “โชติรส”

2.7.2 งานวิจัยเกี่ยวกับการถอดอักษร

การถอดอักษรมีวิธีการทำใกล้เคียงกันกับการถ่ายเสียง (transcription) โดยเริ่มจากการเปลี่ยนข้อความภาษาไทยไปเป็นสายลำดับของหน่วยเสียงในภาษาไทยก่อน แล้วจึงเทียบสายลำดับหน่วยเสียงภาษาไทยไปเป็นสายลำดับเสียงในภาษาอังกฤษก่อนที่จะเปลี่ยนไปเป็นข้อความที่เขียนในระบบการเขียนของภาษาอังกฤษ Aroonmanakun [12] ได้รวบรวมความยากในการแบ่งข้อความภาษาไทยเพื่อถอดอักษรไว้หลายประการ ได้แก่ ตำแหน่งของพยัญชนะออกเสียงต่างกันเมื่อเป็นพยัญชนะต้นกับพยัญชนะตัวสะกด และในบางคำพยัญชนะ 1 ตัวเป็นทั้งพยัญชนะต้นและพยัญชนะสะกดด้วย ตำแหน่งของสระที่มีทั้งอยู่หน้า หลัง บน ล่าง ของพยัญชนะทั้งยังสามารถประสมกันแล้วออกเสียงต่างจากเดิม การลดรูปของสระ การมีตัวควบกล้ำ การมีเสียง อะ กึ่งเสียงระหว่างคำ การมีเสียงอักษรนำ คำพ้องรูป และขอบเขตของคำ ด้วยปัญหาเหล่านี้ทำให้เกิดความกำกวมเกิดขึ้นในการแบ่งพยางค์รวมไปถึงการออกเสียงของพยางค์นั้นๆ และ Aroonmanakun [11] ได้เสนอวิธีการทางสถิติบนพื้นฐานของการใช้ฐานข้อมูลขนาดใหญ่มาทำการแบ่งข้อความภาษาไทยออกเป็นสายลำดับของพยางค์ จากนั้นใช้วิธีการทางสถิติในการหาความน่าจะของหน่วยเสียงจากสายลำดับของพยางค์นั้น และสุดท้ายทำการเปลี่ยนสายลำดับของหน่วยเสียงให้เป็นข้อความที่สะกดด้วยตัวอักษรอังกฤษโดยการใช้กฎการถอดอักษรแบบถ่ายเสียงที่บัญญัติขึ้นโดยราชบัณฑิต 2527

วิธีการนี้ไม่เหมาะสมสำหรับการถอดอักษรของชื่อบุคคลไทย เนื่องจากชื่อของบุคคลไทยแม้จะเสียงเหมือนกันแต่เมื่อถอดอักษรแล้วก็ได้ชื่อที่สะกดด้วยตัวอักษรอังกฤษแตกต่างกันซึ่งวิธีการนี้ไม่สามารถทำได้ด้วยกฎ หรือตารางการจับคู่เสียงกับตัวอักษร เราสังเกตได้ว่าจะเป็นการยากถ้าเราจะใช้วิทยาการศึกษาสำนึก (heuristic) เข้ามาช่วยในการทำโรมันไนเซชั่นกับชื่อบุคคลไทยที่มีลักษณไม่เป็นระบบ ดังนั้นงานวิจัยนี้จึงเสนอวิธีการขับเคลื่อนด้วยข้อมูล (data-driven) ซึ่ง

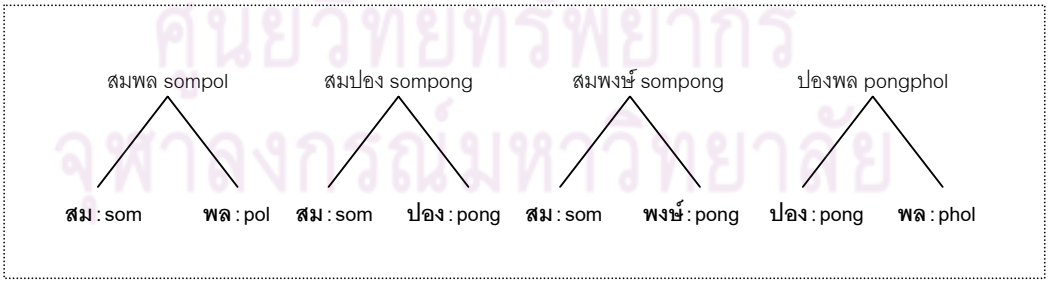
เป็นวิธีการให้คอมพิวเตอร์เรียนรู้การจับคู่อักษรระหว่างภาษาไทยและภาษาอังกฤษที่มีการแบ่งหน่วยย่อยแบบใหม่ซึ่งเรียกว่า “แกรม” หน่วยย่อยนี้มีความแตกต่างกันกับหน่วยพยางค์ทั่วไปโดยที่แต่ละแกรมมีลักษณะคล้ายกับการเก็บคำอ่านไว้คู่กับระบบการเขียนของทั้งสองภาษาเอาไว้ด้วยกัน

บทที่ 3

ขั้นตอนการดำเนินงานวิจัย

3.1 การแบ่งชื่อออกเป็นแกรม

ในงานวิจัยนี้เราจะมองการสะกดของแต่ละชื่อเป็นสายลำดับที่เรียงต่อกันของหน่วยย่อยพื้นฐาน โดยที่คุณลักษณะของแต่ละหน่วยย่อยพื้นฐานเหล่านี้มาจากสายลำดับของอักษรทั้งในภาษาต้นทางและภาษาปลายทางซึ่งในงานวิจัยนี้ คือ ภาษาไทย และ ภาษาอังกฤษตามลำดับ ด้วยการจัดเรียงหน่วยย่อยที่ต่อกันแบบนี้จะทำให้การสะกดชื่อถูกบังคับจากทั้ง 2 ภาษา แต่ละหน่วยย่อยเราพยายามจะให้มีความหมายในทางภาษา แต่อาจจะมีความหมายหรือไม่มีก็ได้ทั้งนี้เพื่อให้สายลำดับอักษรในภาษาไทยกับภาษาอังกฤษออกเสียงได้สอดคล้องกัน ในงานวิจัยนี้จะเรียกหน่วยย่อยพื้นฐานนี้ว่า “แกรม” โดยให้กลุ่มอักษร ก:ข เป็นแกรมที่สะกดในภาษาไทยเป็นกลุ่ม ก และสะกดในภาษาอังกฤษเป็นกลุ่ม ข ดังตัวอย่างในรูปที่ 10 แสดงการแตกองค์ประกอบของชื่อออกเป็นแกรม โดยชื่อทางด้านซ้ายสุดที่สะกดด้วยตัวอักษรภาษาไทย “สมพล” และสะกดด้วยตัวอักษรภาษาอังกฤษ “sompol” สามารถแบ่งเป็นสายลำดับของ 2 แกรม “สม:som” และ “พล:pol” เรียงต่อกัน เนื่องจากคุณสมบัติของแกรมถูกบังคับจากการสะกดของทั้ง 2 ภาษาดังนั้นแกรม 2 แกรมจะแตกต่างกันเมื่อการะสะกดในภาษาไทยหรือการสะกดในภาษาอังกฤษอย่างใดอย่างหนึ่งหรือทั้งสองอย่างแตกต่างกัน จากการแตกองค์ประกอบของตัวอย่างชื่อทั้ง 4 ชื่อในภาพจะทำให้ได้แกรมที่แตกต่างกันทั้งสิ้น 5 แกรม ได้แก่ “สม:som” “พล:pol” “พล:phol” “ปอง:pong” และ “พงษ์:pong”



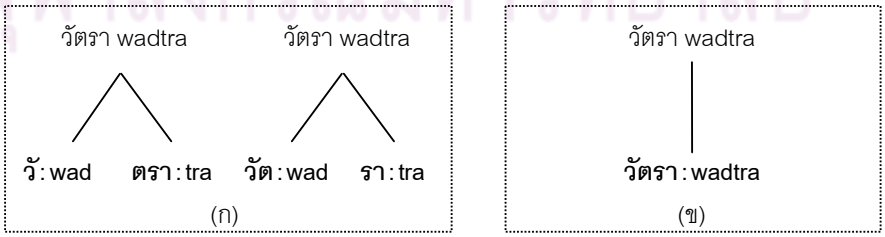
รูปที่ 10 ตัวอย่างการแบ่งชื่อออกเป็นสายลำดับของแกรม

สังเกตว่าความสัมพันธ์ของสายลำดับอักษรไทยกับกับสายลำดับอักษรอังกฤษมีลักษณะเป็นแบบ 1 ต่อ 1 เช่น สายลำดับอักษรไทย “พล” อาจถอดได้เป็น pon หรือ phol และในทางกลับกัน สายลำดับอักษรอังกฤษ “pong” อาจถอดมาจากสายลำดับอักษร ปอง หรือ พงษ์ ดังนั้น

การนิยามแกรมที่ต้องถูกบังคับจากการสะกดทั้ง 2 ภาษาจะช่วยรองรับความหลากหลายในการถอดอักษรของชื่อบุคคล

การสร้างพจนานุกรมแกรมสะสมมีวิธีการง่ายๆ เราจะเพิ่มแกรมเข้าไปในพจนานุกรมในระหว่างที่กำลังประมวลผลชื่อในชุดข้อมูลฝึก โดยแต่ละแกรมจะต้องประกอบด้วยตัวอักษรที่เป็นเสียงพยัญชนะต้นและตัวอักษรที่เป็นเสียงสระหรืออักษรที่ทำให้รู้ถึงสระที่ลดรูปไปเป็นอย่างน้อย ในการแบ่งชื่อออกเป็นสายลำดับของแกรมเราต้องการให้แกรมมีขนาดเล็กที่สุดหรือมีลักษณะเป็นพยางค์เดี่ยวๆ 1 พยางค์ แต่ในบางกรณีแกรมสามารถมีได้มากกว่า 1 พยางค์เพื่อป้องกันไม่ให้เกิดแกรมที่มีการสะกดในภาษาไทยเป็นตัวอักษรเดี่ยวเพียง 1 ตัว ยกเว้น 2 ตัวอักษรได้แก่ “ณ” และ “ฤ” ซึ่งจะกล่าวถึงในเวลาถัดไป การที่เราไม่ต้องการให้เกิดแกรมที่มีการสะกดในภาษาไทยเป็นอักษรเพียงตัวเดียวเพื่อป้องกันไม่ให้เกิดความเป็นไปได้ของคำตอบที่กว้างมากเกินไปในการถอดอักษร และนอกจากนั้นภาษาไทยยังมีสระบางตัวที่ต้องการตัวสะกด ทำให้สระเหล่านี้และอักษรที่ทำหน้าที่เป็นตัวสะกดไม่สามารถอยู่แยกกันได้ ตัวอย่าง 2 ตัวอย่างด้านล่างจะเป็นการแสดงปัญหาที่จะเกิดขึ้นถ้าเราพยายามจะบังคับให้การแบ่งแกรมต้องเป็น 1 พยางค์ โดยตัวอย่างในรูปที่ 11 เป็นปัญหาจากการลดรูปสระ ทำให้การแบ่งคำว่าจรัญออกเป็น 2 พยางค์จะมีอักษรที่เป็นสระ “-ะ” ของพยางค์แรกหายไปและเหลือเพียงอักษร “จ” ที่เป็นพยัญชนะเท่านั้นดังรูป (ก) เราจึงเลือกที่ให้มีมากกว่า 1 พยางค์ต่อ 1 แกรม ดังนั้นเราจึงแบ่งชื่อตามรูป 11 (ข) คือ แบ่งออกเป็นสายลำดับของ 2 แกรม “จรัญ : charan” และ “กุล : kul” ที่เรียงต่อกัน

รูปที่ 11 ภาพการเลือกใช้สายลำดับของแกรมที่มีหลายพยางค์ (ข) แทนการใช้สายลำดับที่แต่ละแกรมมีเพียง 1 พยางค์ (ก)



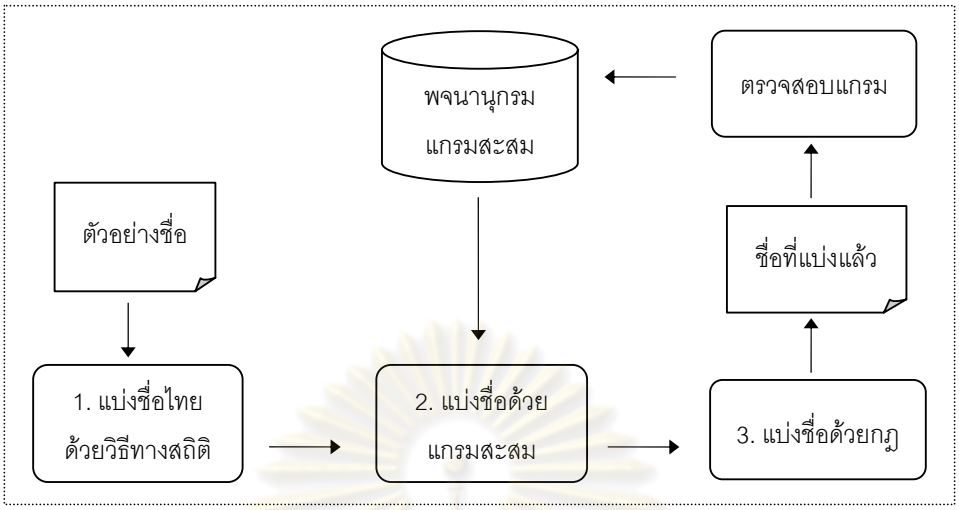
รูปที่ 12 ภาพตัวอย่างอื่นๆ ในการใช้สายลำดับของแกรมที่มีหลายพยางค์ (ข) แทนการใช้สายลำดับที่แต่ละแกรมมีเพียง 1 พยางค์ (ก)

ตัวอย่างในรูปที่ 12 เป็นปัญหาของชื่อที่มี 2 พยางค์ติดกันโดยพยางค์ทั้ง 2 มีการใช้ตัวอักษรพยัญชนะร่วมกัน โดยพยางค์แรกใช้ “ต” เป็นพยัญชนะตัวสะกดและพยางค์หลังใช้ “ต” เป็นพยัญชนะต้นควบกล้ำกันกับ “ร” เพื่อทำหน้าที่ร่วมกันเป็นพยัญชนะต้นของพยางค์ การแบ่งชื่อโดยบังคับให้เป็น 1 แกรมต่อ 1 พยางค์จะทำให้แกรมมีลักษณะเป็นอย่างใดอย่างหนึ่งในรูปที่ 12 (ก) โดยการแบ่งชื่อแบบตัวอย่าง 12 (ก) รูปซ้ายจะทำให้เกิดแกรม “วั:wad” เป็นแกรมที่ผิดพลาดเนื่องจากอักษรของสระที่ใช้เป็น " ั " ซึ่งจำเป็นต้องมีพยัญชนะสะกดต่อท้าย ในขณะที่การแบ่งชื่อแบบตัวอย่าง 12 (ก) รูปขวาจะทำให้แกรม “รา:tra” ซึ่งเป็นลักษณะที่ไม่เหมาะสมเนื่องจากพยัญชนะต้น “ร” จะไม่สามารถออกเสียงเป็นหน่วยเสียงภาษาไทย “/tr/” ที่ตรงกับหน่วยเสียงควบในภาษาอังกฤษ “tr” ได้ ดังนั้นการแบ่งชื่อในตัวอย่างที่ 12 เราเลือกที่จะแบ่งตามแบบรูป 13 (ข) คือ แบ่งเป็น 1 แกรม “วัตรา : wadtra”

ตามที่ได้กล่าวไว้ข้างต้นว่ามีข้อยกเว้นสำหรับกฎห้ามแบ่งแกรมที่ทำให้มีการสะกดในภาษาไทยเป็นพยัญชนะเพียงตัวเดียว กรณีแรกคือชื่อมีที่ใช้พยัญชนะ “ณ” เป็นคำที่สมบูรณ์ซึ่งเป็นเรื่องปกติที่เราจะพบในนามสกุลของคนไทย ซึ่งจะมีเพียง 1 แกรมเท่านั้นสำหรับกรณีนี้คือ “ณ:na” กรณีที่สองได้แก่ “ฤ” เมื่อถูกออกเสียงเป็นพยางค์ที่สมบูรณ์ในตัวเองซึ่งมักปรากฏในชื่อ เช่น “มนต์ฤดี” แต่กรณีของพยัญชนะ “ฤ” เดี่ยวๆ นี้สามารถพบได้ในหลายๆ แกรม เช่น “ฤ : rue”, “ฤ : reu” หรือ “ฤ : rhu” และอื่นๆ

3.2 การสร้างพจนานุกรม

งานสร้างพจนานุกรมมักเป็นงานที่ต้องใช้เวลานานและน่าเบื่อถ้าเราต้องทำการแบ่งชื่อเองกับฐานข้อมูลชื่อขนาดใหญ่ ในงานวิจัยนี้เสนอวิธีการกึ่งอัตโนมัติในการแบ่งชื่อในชุดข้อมูลฝึก (รายละเอียดเกี่ยวกับฐานข้อมูลชื่อที่ถูกใช้ในงานวิจัยนี้จะกล่าวในภายหลัง) การแบ่งชื่อในชุดข้อมูลฝึกจะประมวลผลหลายรอบ โดยแต่ละรอบจะประมวลผลทีละไม่กี่พันชื่อ หลังจากการประมวลผลในแต่ละรอบเราจะนำแกรมที่ได้รับจากการแบ่งชื่อในรอบนั้นมาเพิ่มลงในพจนานุกรม ซึ่งจะนำไปใช้กับการแบ่งชื่ออย่างอัตโนมัติในรอบต่อไป แต่ทั้งนี้ชื่อที่ถูกแบ่งนั้นต้องได้รับการตรวจสอบและแก้ไขด้วยมือก่อนที่จะนำแกรมที่ได้เหล่านี้ไปเพิ่มลงในพจนานุกรมดังกระบวนการในรูปที่ 13

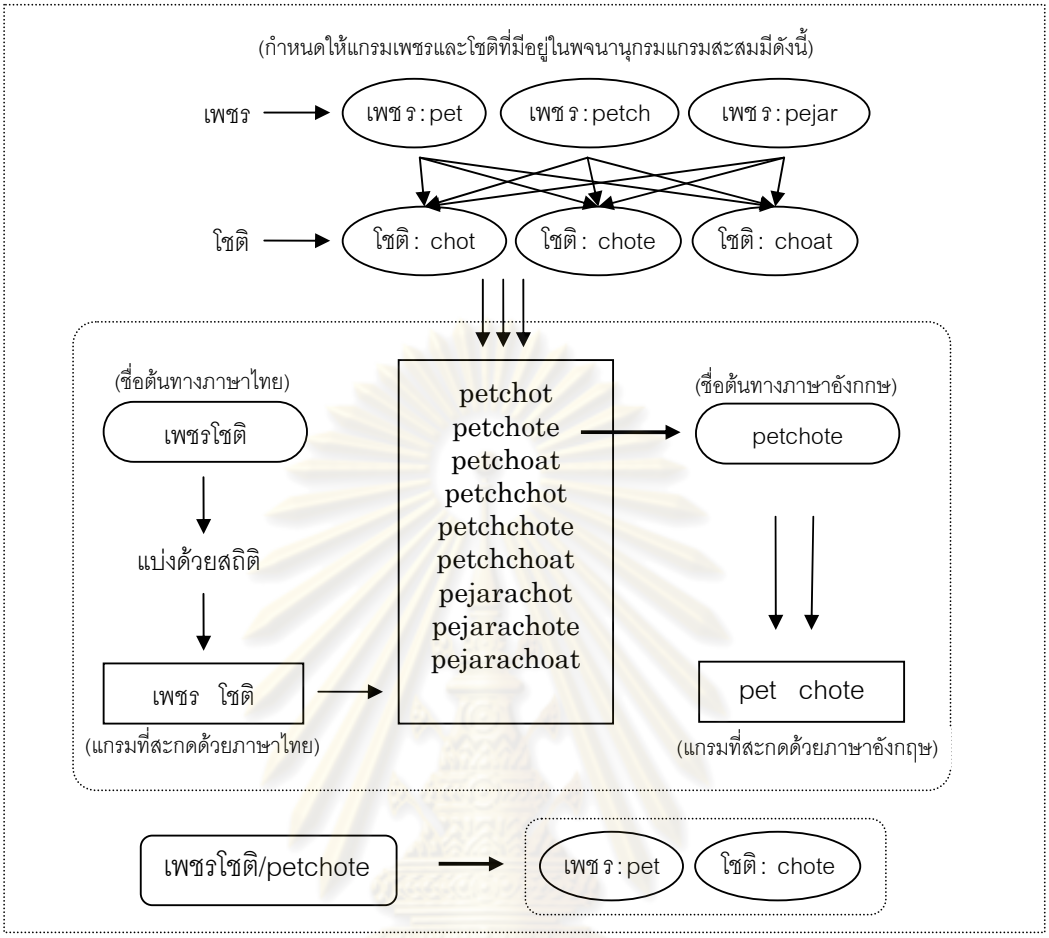


รูปที่ 13 กระบวนการสร้างพจนานุกรมแกรมสะสม

เมื่อเริ่มต้นการแบ่งชื่อในแต่ละรอบ เราจะแบ่งชื่อในส่วนที่สะกดด้วยอักษรไทยออกเป็นสายลำดับของพยางค์ก่อนด้วยวิธีการของ Aroonmanakun [11] จากนั้นใช้แกรมที่สะสมไว้ในรอบก่อนๆ มาช่วยในการแบ่งชื่อ โดยเราจะแบ่งชื่อที่สะกดด้วยตัวอักษรอังกฤษด้วยการใช้แกรมสะสมที่เราได้จากรอบก่อนๆ เข้ามาช่วยค้นหาการแบ่งแบบขวาไปซ้าย ซึ่งในเวลาเดียวกันนี้แต่ละคำตอบจะเป็นแกรมที่ซ้ำกับแกรมสะสมที่อยู่ในพจนานุกรม

3.2.1 การแบ่งชื่อด้วยพจนานุกรมแกรมสะสม

รูปที่ 14 แสดงวิธีการแบ่งชื่อ “เพรชโชติ/petchote” ด้วยพจนานุกรมแกรมสะสม เริ่มจากนำชื่อที่สะกดด้วยอักษรไทย "เพรชโชติ” มาแบ่งออกเป็นสายลำดับของพยางค์ด้วยวิธีการทางสถิติ นำพยางค์ “เพชร” และ “โชติ” ที่เป็นภาษาไทยนี้ไปค้นหาแกรมที่มีการสะกดด้วยอักษรไทยตรงกับพยางค์นั้นๆ ในพจนานุกรมแกรมสะสม จากนั้นนำสายลำดับอักษรที่สะกดด้วยอักษรอังกฤษของแต่ละแกรมเหล่านั้นมาเรียงต่อกันตามตำแหน่งของสายลำดับพยางค์ เนื่องจากแต่ละพยางค์ที่นำไปค้นหาในพจนานุกรมได้ผลรับออกมาหลายแกรม เราจะแบ่งชื่อตามสายลำดับของแกรมที่สร้างสายลำดับของอักษรอังกฤษได้ตรงกับชื่อของบุคคลนั้นที่สะกดด้วยอักษรอังกฤษ

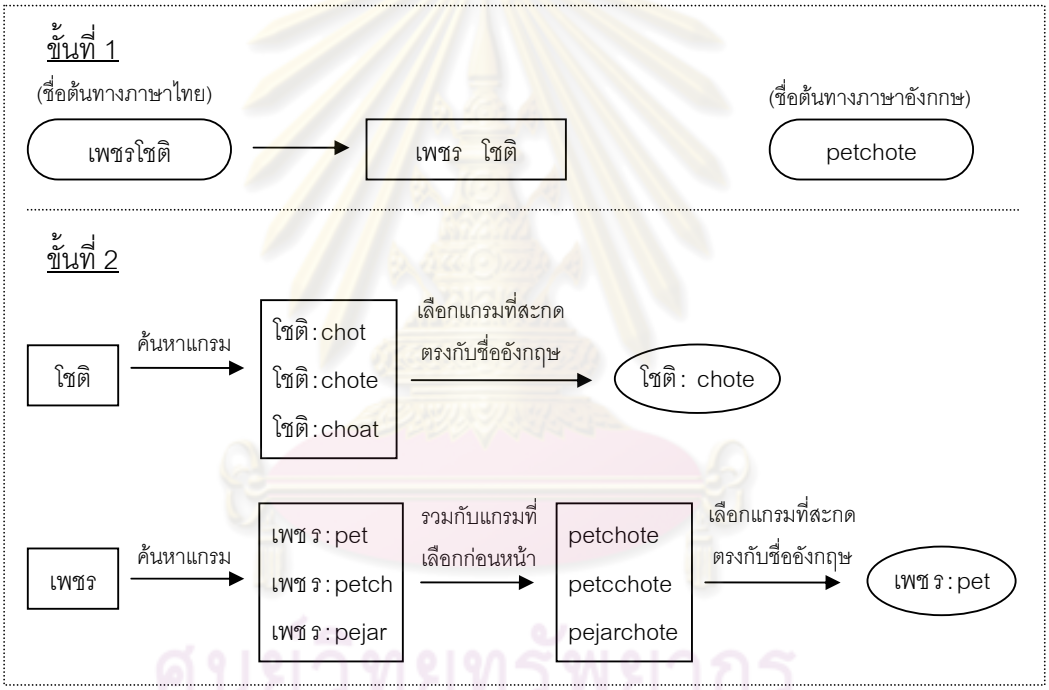


รูปที่ 14 แสดงการแบ่งชื่อ "เพชรโชติ/petchote" ด้วยพจนานุกรมแกรมสะสม

ในตัวอย่างรูปที่ 14 เราสมมุติว่าในพจนานุกรมแกรมสะสมมีแกรมที่สะกดด้วยอักษรภาษาไทยเป็น "เพชร" และ "โชติ" อย่างละ 3 แกรมทำให้ในตัวอย่างมีเพียง 9 สายลำดับอักษรที่เป็นตัวเลือก แต่ในการทดลองพบว่าแกรมที่สะกดด้วยอักษรไทยตรงกับ "เพชร" มีถึง 34 แกรม หรือสะกดด้วยอักษรอังกฤษต่างๆ กันดังนี้ "beth phetchara petchara petchra pechara peachla phetcha pejara patcha phetch phetra petcha pechra peacha bejara paetch pechr patch pheth phech petch phate bejra pacha pecha pach pech phet peth phat pet pej pat ped" และ "โชติ" มีถึง 20 แกรม หรือ สะกดด้วยภาษาอังกฤษต่างๆ กันดังนี้ "choad chotes choate chotte chost chong chort chote chode chout choat chose shote choot choth jote chot chod cho" เมื่อนำมาสร้างสายลำดับอักษรที่เป็นตัวเลือกจึงมาตัวเลือกเกิดขึ้นถึง 680 ตัวเลือกจากชื่อที่มีเพียง 2 พยางค์ ในกรณีของนามสกุลที่มีจำนวนพยางค์ตั้งแต่ 5 พยางค์ขึ้นไปเราจะเริ่มพบว่าตัวเลือกอาจมากกว่า 1 ล้านตัวเลือกได้ เช่น "ศุภ เวช รัก ษา กุล" ในพจนานุกรมพบ "ศุภ" 16 แกรม "เวช" 40 แกรม "รัก" 13 แกรม "ษา" 3 แกรม และ "กุล" 41 แกรม

สามารถสร้างสายลำดับอักษรได้ทั้งสิ้น 1,023,360 ตัวเลือก วิธีการนี้ให้ตัวเลือกมากเกินไป จึงต้องมีการลดทอนแกรมที่เป็นตัวเลือกจากพยางค์ภาษาไทย

การลดทอนแกรม ทำเพื่อให้สายลำดับแกรมที่ถูกสร้างเป็นตัวเลือกมีน้อยที่สุดเท่าที่จะเป็นไปได้ เป็นเสมือนการตัดสายลำดับแกรมที่เป็นตัวเลือกที่ไม่มีโอกาสถูกต้องออกไป ทำให้การแบ่งชื่อมีประสิทธิภาพดีขึ้น การลดทอนแกรมจะทำในระหว่างการแบ่งชื่อโดยแบ่งชื่อด้วยแกรมในพจนานุกรมสะสมจากขวาไปซ้าย โดยยึดตามสายลำดับของพยางค์ในภาษาไทยเป็นหลัก จำนวนแกรมที่เป็นตัวเลือกจะถูกลดทอนลงเมื่อเริ่มประมวลผลพยางค์ในลำดับถัดไปเรื่อยๆ และยังช่วยให้มีแนวโน้มที่จะพบแกรมใหม่จากการจับคู่สายลำดับอักษรภาษาไทยกับอักษรภาษาอังกฤษที่อยู่หน้าสุดของชื่อ ซึ่งเป็นแกรมที่ยังไม่มีอยู่ในพจนานุกรม



รูปที่ 15 วิธีการแบ่งชื่อ เพชรโชติ ด้วยการเลือกแกรมแบบลดทอน

รูปที่ 15 เป็นการแสดงการแบ่งชื่อ เพชรโชติ เป็นสายลำดับของแกรมด้วยการเลือกแกรมแบบลดทอน โดยในขั้นที่ 1 จะนำชื่อที่สะกดด้วยอักษรไทยมาแบ่งออกเป็นสายลำดับของพยางค์ด้วยวิธีการของ Aroonmanakun [11] ได้เป็นสายลำดับ “เพชร โชติ” จากนั้นขั้นที่ 2 จะทำการเลือกแกรมโดยใช้วิธีการบังคับจากชื่อทั้ง 2 ภาษาช่วยในการลดทอนแกรมที่พบในพจนานุกรม โดยในขั้นที่ 2 นี้จะเริ่มประมวลผลจากพยางค์ทางขวาสุดก่อน จากตัวอย่างในรูปคือ “โชติ” จากพยางค์โชติ จะพบ 3 แกรมที่มีด้านที่สะกดด้วยอักษรไทย โชติ แต่เมื่อเทียบกับชื่อภาษาอังกฤษแล้ว จะพบว่ามีเพียง 1 แกรมเท่านั้นคือ โชติ:chote ที่สะกดตรงกับชื่อภาษาอังกฤษ เมื่อได้แกรมจาก

พยางค์ทางขวาสุดแล้วก็จะประมวลผลกับแกรมถัดไปทางซ้าย จากตัวอย่างในรูปคือ “เพชร” เมื่อค้นหาแกรมด้านที่สะกดด้วยอักษรไทยตรงกับ “เพชร” จะได้ 3 แกรม นำด้านที่สะกดด้วยอักษรอังกฤษของแต่ละแกรมมาต่อท้ายด้วย chote ซึ่งเป็นสายลำดับอักษรอังกฤษที่ได้จากแกรมที่ประมวลผลมาก่อน ในตัวอย่างนี้คือแกรม โชติ:chot นำสายลำดับอักษรอังกฤษของแต่ละแกรมที่รวมกับ chote แล้วนี้ไปเปรียบเทียบกับชื่ออังกฤษ จะพบว่ามีเพียงแกรมเดียว คือ เพชร:pet ที่เมื่อรวมกับแกรมโชติแล้ว ด้านสะกดด้วยอักษรอังกฤษตรงกับชื่ออังกฤษ เพื่อให้เห็นภาพชัดเจนยิ่งขึ้น จะแสดงลำดับในการเลือกแกรมมาใช้ในการแบ่งชื่อสกุลที่มักจะมีความยาวมากว่าชื่อจริงใน ตัวอย่างรูปที่ 16

ชื่อ จิระวัฒนเอก แบ่งออกเป็นสายลำดับพยางค์ → จิ ระ วัฒน เอก		ชื่ออังกฤษ chirawattanaek
ครั้งที่ 1) เอก	→ “เอก:aek”, “เอก:ek”	Chirawattanaek
ครั้งที่ 2) วัฒน เอก	→ “wattana:วัฒน” + “เอก:ek”	chirawattana ek
ครั้งที่ 3) ระ วัฒน เอก	→ “ระ:ra“ + ”wattana:วัฒน” + “เอก:ek”	chira wattana ek
ครั้งที่ 4) จิ ระ วัฒน เอก	ไม่พบแกรมที่สะกดด้วยอักษรไทย “จิ” ที่รวมกับสายลำดับแกรมที่ได้ในครั้งที่ 3 แล้วตรงกับชื่อที่สะกดด้วยภาษาอังกฤษ	chi ra wattana ek
→ เราสามารถหวังได้ว่าพยางค์ “จิ” ตรงกับสายลำดับอักษรของชื่อภาษาอังกฤษว่า “chi”		

รูปที่ 16 ขั้นตอนในการเลือกแกรมมาใช้ในการแบ่งชื่อ จิระวัฒนเอก/chirawttanaek

ดังในตัวอย่างที่ 16 ชื่อ “จิระวัฒนเอก/chirawattanaek“ ขั้นตอนเริ่มด้วยการนำชื่อภาษาไทย “จิระวัฒนเอก” มาแบ่งออกเป็นสายลำดับของพยางค์ “จิ ระ วัฒน เอก” จากนั้นเริ่มประมวลผลครั้งที่ 1 จากขวาไปซ้ายโดยการหาแกรมที่มีสายลำดับอักษรที่สะกดด้วยภาษาไทยเป็น “เอก” และสายลำดับอักษรที่สะกดด้วยภาษาอังกฤษเป็นข้อความใดๆ ที่สะกดตรงกับชื่อภาษาอังกฤษจากด้านขวาสุดซึ่งพบ 2 แกรมจาก 43 แกรม คือ “เอก:aek” และ “เอก:ek” นำ 2 แกรมนี้มาประมวลผลต่อในครั้งที่ 2

ครั้งที่ 2 คือ หาค้นหาแกรมที่เมื่อต่อท้ายด้วยแกรมใดๆ จากครั้งที่ 1 แล้ว นำสายลำดับอักษรที่สะกดด้วยภาษาไทยของทั้ง 2 แกรมมารวมกันตรงกับ “วัฒนเอก” และนำสายลำดับอักษรที่สะกดด้วยภาษาอังกฤษของทั้ง 2 แกรมมารวมกันตรงกับสายลำดับอักษรในชื่อภาษาอังกฤษจากหลังสุด ซึ่งพบ 1 แกรมจาก 21 แกรมที่มีสายลำดับอักษรภาษาไทยว่า “วัฒน” คือ “วัฒน:wattana” ที่สามารถต่อด้วยแกรม “เอก:ek” แล้วทำให้ตรงกับส่วนของสายลำดับชื่อภาษาอังกฤษจากด้วยขวาว่า “wattanaek”

ในครั้งที่ 3 ค้นหาได้เพียง 1 แกรมคือ “ระ:ra” และสุดท้ายในครั้งที่ 4 ค้นหาไม่พบแกรมที่ต่อด้วย 3 แกรมก่อนหน้าแล้วตรงกับชื่อที่สะกดด้วยภาษาอังกฤษ จึงสามารถคาดหวังได้ว่าแกรม “จิ:chi” เป็นแกรมที่ยังไม่มีในพจนานุกรมสะสมของเรา แต่เพื่อความถูกต้องจึงต้องตรวจด้วยผู้เชี่ยวชาญอีกครั้งก่อนนำแกรมไปใส่ในพจนานุกรมสะสม ถ้าในกรณีที่พจนานุกรมเรามีแกรม “จิ:chi” อยู่ก่อนแล้วจะทำให้ชื่อ “จิระวัฒนเอก/chirawattanaek“ นี้สามารถแบ่งได้ด้วยพจนานุกรมแกรมสะสมอย่างสมบูรณ์ คือ แบ่ง เป็น 4 แกรม “จิ:chi”, “ระ:ra”, “วัฒน:wattana”, และ “เอก:ek” ในทางกลับกันถ้าในครั้งที่ 3 ในพจนานุกรมไม่มีแกรม “ระ:ra” เราจะหยุดการแบ่งชื่อด้วยพจนานุกรม และตัดเฉพาะชื่อส่วนที่เหลือคือ “จิระ/chira” ไปทำการแบ่งด้วยกฎต่อไป

ผลจากการแบ่งชื่อด้วยพจนานุกรมแกรมสะสมจะทำให้เราได้คำตอบของชื่อแบ่งออกได้เป็น 3 ประเภทที่เป็นไปได้ ได้แก่

1) จับคู่ได้อย่างสมบูรณ์ คือ เมื่อชื่อนั้นสามารถแบ่งออกเป็นลายลำดับของแกรมซึ่งแกรมเหล่านั้นมีอยู่ในแกรมสะสมที่มีอยู่ในพจนานุกรมได้ทั้งหมด และเราไม่จำเป็นต้องแก้ไขเอง กรณีนี้จะไม่มีแกรมใหม่เพิ่มเข้าไปในพจนานุกรม

2) จับคู่ได้ไม่สมบูรณ์แบบผิดพลาด 1 จุด คือ ผลของการค้นหาแบบขวาไปซ้าย ถ้าสามารถแบ่งบางส่วนของชื่อออกเป็นสายลำดับของแกรมที่มีอยู่ในพจนานุกรมได้ ยกเว้นเพียงพยางค์แรกของชื่อที่สะกดในภาษาไทยและกลุ่มอักษรที่สะกดในภาษาอังกฤษที่ไม่สามารถจับคู่ได้ที่อยู่ตอนต้นของชื่อ กรณีนี้เราต้องการการตัดสินว่าเราจะจับคู่กันระหว่างพยางค์แรกของชื่อภาษาไทยกับกลุ่มอักษรที่เหลืออยู่ตอนต้นชื่อภาษาอังกฤษเป็นแกรมใหม่และเพิ่มลงในพจนานุกรม หรือเราจะทำการแก้ไขการแบ่งชื่อนี้ด้วยมือก่อนที่จะนำแกรมใหม่เพิ่มในพจนานุกรม

3) จับคู่ได้ไม่สมบูรณ์แบบผิดพลาดมากกว่า 1 จุด คือ เหลือพยางค์ในภาษาไทยมากกว่า 1 พยางค์ที่ไม่สามารถจับคู่ได้ กรณีนี้เราจะทำการแบ่งชื่อด้วยกฎต่อไป

สำหรับรอบแรกของการแบ่งชื่อเมื่อเรายังไม่มีแกรมสะสมในพจนานุกรมนั้น ผลลัพธ์การแบ่งชื่อจะเป็นแบบกรณีสุดท้ายคือเกิดจุดผิดพลาดมากกว่า 1 จุด และเราต้องทำการตรวจสอบและแก้ไขเองทั้งหมด

3.2.2 การแบ่งชื่อโดยการใช้กฎ

การแบ่งชื่อโดยการใช้กฎจะทำเมื่อการแบ่งชื่อด้วยแกรมสะสมเกิดความผิดพลาดมากกว่า 1 จุด และทำกับชื่อที่สะกดด้วยตัวอักษรภาษาอังกฤษโดยใช้ชื่อภาษาไทยที่แบ่งแล้วมาช่วยในการแบ่ง วิธีการคือสร้างกฎง่ายๆ ที่ได้จากการสังเกตการแบ่งชื่อหลายๆ ชื่อด้วยมือและใช้ร่วมกับตารางเทียบอักษรไทยอังกฤษ ขั้นตอนเริ่มด้วยการแบ่งสายลำดับอักษรของชื่อออกเป็นกลุ่มๆ โดยดูจากสระในภาษาอังกฤษจากนั้นจึงรวมแต่ละกลุ่มเข้าด้วยกันตามกฎ หรือจัดเรียงกลุ่มที่อยู่ติดกัน

ใหม่ด้วยตารางเทียบอักษร สุดท้ายผลของการแบ่งชื่อจะทำให้จับกันได้แบบ 1 ต่อ 1 ระหว่างแต่ละพยางค์ของภาษาไทยกับแต่ละกลุ่มอักษรของภาษาอังกฤษแบบตามลำดับเพื่อทำเป็น 1 แกรม ดังรูปที่ 17

ชื่ออังกฤษ	meesaplak	ชื่อไทย "มี ทรัพย์ หลาก"
กระจายกลุ่ม	→ mee, sa, plak	
จัดตัวสะกด	→ mee, sap, lak	
จัดเรียงครั้งที่ 1	→ mee, sap, lak	→ มี, ทรัพย์, หลาก

รูปที่ 17 แสดงการแบ่งชื่อ "meesaplak" ให้สัมพันธ์กับสายลำดับพยางค์ "มี ทรัพย์ หลาก"

ชื่ออังกฤษ	amornmedwarintara	ชื่อไทย "อมร เมศ วรินทร์"
กระจายกลุ่ม	→ amo, rnme, dwa, ri, nta, ra	
จัดตัวสะกด	→ amorn, me, dwa, ri, nta, ra	
จัดเรียงครั้งที่ 1	→ amorn, med, wa, ri, nta, ra	
จัดเรียงครั้งที่ 2	→ amorn, med, warintara	→ อมร, เมศ, วรินทร์

รูปที่ 18 การแบ่งชื่อ "amornmedwarintara" ให้สัมพันธ์กับสายลำดับพยางค์ "อมร, เมศ, วรินทร์"

รูปที่ 18 แสดงตัวอย่างการแบ่งชื่อ "amornmedwarintara" ให้สัมพันธ์กับสายลำดับพยางค์ "อมร, เมศ, วรินทร์" โดยแบ่งเป็น 3 ขั้นได้แก่

1) การกระจายกลุ่ม เป็นการประมวลผลแบบซ้ายไปขวา ด้วยการใช้กฎการแบ่งกลุ่มทุกครั้งที่พบอักษรที่เป็นสระในภาษาอังกฤษ ได้แก่ "a", "e", "i", "o", และ "u" ทำให้แบ่งกลุ่มได้เป็น "a, mo, rnme, dwa, ri, nta, ra" จากการสังเกตการแบ่งชื่อหลายๆ ชื่อด้วยมือทำให้พบข้อยกเว้น 3 กรณี คือ

- อักษรตัวแรกของสายลำดับขึ้นต้นด้วยสระ เราจะไม่แบ่งกลุ่มเป็นอักษรตัวแรกกับสายลำดับอักษรที่เหลือ ในตัวอย่างนี้คือ จะไม่แยกเป็น "a, mornmedwarintara"
- อักษรตัวสุดท้ายเป็นพยัญชนะต้น เราจะนำไปรวมกลุ่มกับ กลุ่มที่อยู่ด้านซ้าย ในรูปที่ 16 คือ "pla, k" รวมเป็น "plak"
- เมื่ออักษรที่เป็นสระเรียงต่อกัน รูปที่ 16 คือ "mee" จะไม่แยกเป็น "me, e"

2) การจัดตัวสะกด คือ การตัดพยัญชนะที่อยู่ด้านหน้าของสระ (พยัญชนะต้น) ของกลุ่มทางขวาไปเป็นพยัญชนะท้ายของกลุ่มทางซ้าย โดยเริ่มจัดเรียงจาก 2 กลุ่มที่อยู่ทางซ้ายสุดก่อนแล้วประมวลผลไปทางขวา จากการสังเกตการแบ่งชื่อหลายๆ ชื่อด้วยมือทำให้สร้างกฎการตัดง่ายๆ ดังนี้

- ถ้าพยัญชนะต้นของกลุ่มทางขวามี 2 ตัว หรือ 4 ให้แบ่งครึ่งหนึ่งไปต่อท้ายกับกลุ่มทางซ้าย เช่น “sa, plak” รวมจัดเป็น “sap, lak”
- ถ้าพยัญชนะต้นของกลุ่มทางขวามีมากกว่า 2 ตัว และ 2 ตัวแรกเป็น “ng” ให้แบ่งตัว 2 แรกไปต่อท้ายกับกลุ่มทางซ้าย (โดยมากจะเป็นตัวสะกด “ง”)
- ถ้าพยัญชนะต้นของกลุ่มทางขวามีมากกว่า 2 ตัว และตัวแรกเป็น “r” ให้แบ่ง 2 ตัวแรกไปต่อท้ายกับกลุ่มทางซ้าย เช่น “ka, rnda” รวมจัดเป็น “karn, da” (กานต์ดา) “le, rtkun” รวมเป็น “lert, kun” (เลิศกุล)

3) การจัดเรียง คือ การรวม 2 กลุ่มที่อยู่ติดกันเข้าด้วยกันโดยอาศัยตารางเทียบอักษรไทยอังกฤษที่สร้างขึ้น จากการสังเกตการแบ่งชื่อหลายๆ ชื่อด้วยมือทำให้สร้างกฎการจัดเรียงในแต่ละคู่มีลำดับดังนี้

- ถ้าจำนวนกลุ่มของสายลำดับอักษรที่สะกดด้วยภาษาอังกฤษเท่ากับจำนวนพยางค์ของสายลำดับพยางค์ที่สะกดด้วยอักษรภาษาไทย ให้หยุดการจัดเรียง
- ถ้าพยัญชนะของกลุ่มทางขวา สัมพันธ์กับพยางค์ในภาษาไทยลำดับที่ก่อนหน้า ให้นำกลุ่มทางซ้ายไปรวมกับกลุ่มทางเป็นกลุ่มเดียวกัน แล้วเลื่อนไปพิจารณากลุ่มถัดไปทางขวา เช่น “ปัญ สุทธิ์” กับ “pun, ya, suith” เราจะนำกลุ่ม “pun” และ “ya” รวมกันเป็น “punya, suith” เนื่องจาก “y” สัมพันธ์กับ “ญ” ของพยางค์ ปัญ ซึ่งเป็นพยางค์ลำดับที่ 1 ก่อนหน้ากลุ่ม “ya” ที่มีลำดับกลุ่ม ที่ 2 และหยุดการจัดเรียงต่อเนื่องจากจำนวนพยางค์เท่ากับจำนวนกลุ่มอักษรภาษาอังกฤษแล้ว
- ถ้าพยัญชนะท้ายของกลุ่มทางซ้ายเมื่อรวมกับพยัญนะต้นของกลุ่มทางขวาแล้ว สัมพันธ์กับพยางค์ภาษาไทยลำดับที่ที่ติดกันทางขวา ให้ตัดพยัญชนะท้ายของกลุ่มซ้ายไปรวมด้านหน้าของกลุ่มด้านขวาเพื่อเป็นพยัญชนะต้นควบกล้ำ เช่น “เบี้ยกลาง” กับ “biak, lang” ให้ตัด k ไปรวมกับกลุ่ม “lang” เป็น “bia, klang” เนื่องจาก “kl” สัมพันธ์กับพยางค์ “กลาง” ซึ่งมีลำดับที่ 2 โดยกลุ่มอักษร “biak” เป็นกลุ่มลำดับที่ 1
- ถ้า ณ ขณะที่ประมวลกลุ่มปัจจุบันเป็นกลุ่มลำดับที่ตรงกับพยางค์สุดท้ายของสายลำดับพยางค์ในภาษาไทยแล้ว แต่จำนวนกลุ่มที่ยังไม่ได้ประมวลผลยังเหลืออีก ให้นำกลุ่มอักษรภาษาอังกฤษที่เหลือมารวมเป็นกลุ่มเดียว เช่น “amorn, med, wa, ri, nta, ra” กับ “อมร, เมศ, วรินทร์” ถ้ากำลังประมวลผลอยู่ที่กลุ่ม “wa” ให้รวม “wa, ri, nta, ra” เป็นกลุ่มเดียวกันได้เป็น “amorn, med, warintara” เนื่องจากจะทำให้มี 3 กลุ่มเท่ากับจำนวนพยางค์ในสายลำดับพยางค์ภาษาไทย

ถึงแม้ว่าวิธีนี้จะทำให้เกิดความผิดพลาดขึ้นในบางกรณี แต่ก็ยังแบ่งชื่อได้ถูกต้องเป็นส่วนมากและทำให้เราสะดวกขึ้นมากในการแบ่งชื่อ และการตรวจสอบแก้ไข

3.2.3 การสร้างตารางเทียบอักษร

การสร้างตารางเทียบอักษรเริ่มสร้างจากตารางที่ว่างเปล่า เมื่อผู้วิจัยพบว่าชื่อที่เขียนด้วยอักษรภาษาไทยตัวใด มักเขียนด้วยอักษรภาษาอังกฤษตัวใดก็จะนำมาจับคู่กันและใส่ไว้ในตารางเทียบอักษร ดังนั้นวิธีการนี้อาจจะไม่ตรงกับตารางการถอดอักษรที่ราชบัญฑิตแนะนำในหลายตัวอักษร และการจับคู่ระหว่างอักษรภาษาไทยกับอักษรภาษาอังกฤษยังมีลักษณะเป็นแบบ 1 อักษรอังกฤษต่อหลายอักษรไทย ดังแสดงดังตาราง 6

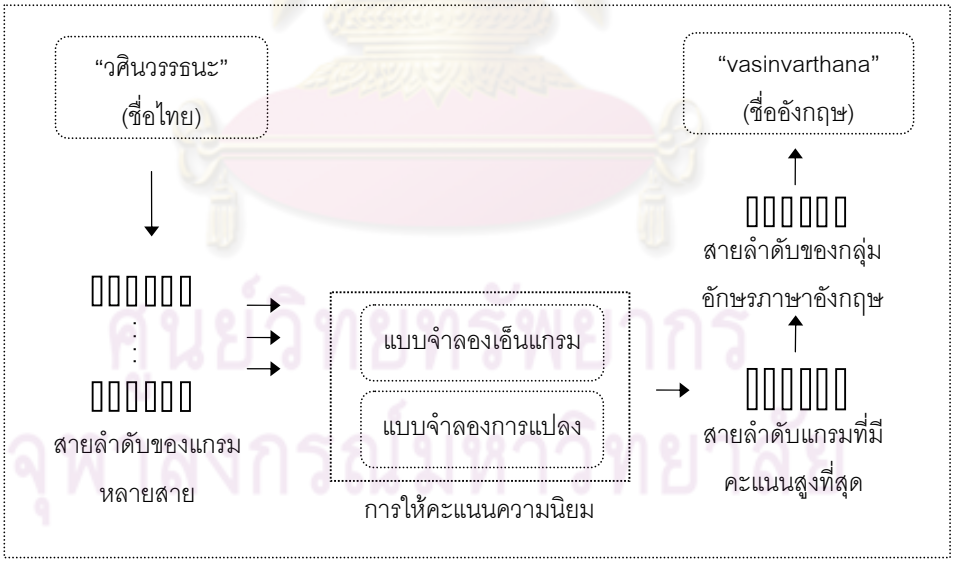
ตารางที่ 6 ตารางเทียบอักษรระหว่างอักษรภาษาไทยและอักษรภาษาอังกฤษ

อักษรภาษาอังกฤษ	อักษรภาษาไทย	อักษรภาษาอังกฤษ	อักษรภาษาไทย
a	อ	kh	ก, ข, ฃ, ค, ฅ, ฆ, ง
b	บ	kl	กล, คล
c	ซ, ทร, ค, ก, ฌ	kr	กฤ, กร, คร
d	ด, ฎ, ฏ, ฑ	gr	กฤ, กร, คร,
e	อ	gl	กล,
f	ฟ, ฝ	ng	ง
g	ก, ค, จ	ch	จ, ฉ, ช, ฌ
h	ฮ, ห	sh	ฉ, ช,
i	อ	th	ฐ, ฑ, ฒ, ถ, ท, ธ
j	จ, ฉ	ph	ผ, พ, ภ, ฝ
k	ก, ค, ข	rue	ฤ, ฤๅ
l	ล, ฬ, ร, ฤ	ri	ฤ
m	ม	roe	ฤ
n	ณ, น	lue	ฦ, ฦๅ
o	อ	sr	สร, ศร, ษร, ซร,
p	ป, พ, บ, ภ, ผ	tr	ทร, ถร, ฐร, ธร, ตร, ธร,
q		sl	สล, ศล, ษล, ซล,
r	ร, ล, ฤ	tl	ทล, ถล, ฐล, ธล,
s	ซ, ศ, ษ, ส, ทร	pr	ปร, ปฤ, ป, ผ, พร, พฤ
t	ฏ, ต, ท, ธ, ฒ, ถ, ฐ, ฎ, ฑ	pl	ปล, พล,
u	อ,	th	ฐ, ท, ธ, ถ, ฒ,

อักษรภาษาอังกฤษ	อักษรภาษาไทย	อักษรภาษาอังกฤษ	อักษรภาษาไทย
v	ว	jh	จ,
w	ว	au	อ,
x	อ	wh	ว, หว,
y	ญ, ย	kw	ข,
z	ซ, ศ, ษ, ส, ทร		

3.3 การถอดอักษรโดยใช้แกรม

การถอดอักษรชื่อคนไทยโดยการใช้พจนานุกรมแกรมสะสมแสดงไว้ดังรูปที่ 19 ชื่อบุคคลไทยที่จะนำมาถอดอักษรเป็นอักษรอักฤษจะถูกแบ่งออกเป็นสายลำดับของแกรมหลายๆ สาย สำหรับเป็นตัวเลือกโดยทุกสายจะตรงกับการสะกดในภาษาไทยของชื่อนั้น แต่ละสายลำดับของแกรมจะมีคะแนนความนิยมซึ่งได้รับมาจากแบบจำลองภาษาและแบบจำลองแปลงวลี ระบบจะเลือกสายลำดับที่มีค่าความนิยมสูงสุดมาเป็นคำตอบของเหล่าตัวเลือกทั้งหมด จากนั้นการถอดอักษรเป็นอักษรอังกฤษจะทำโดยการนำอักษรอังกฤษของสายลำดับแกรมที่เป็นคำตอบมาเชื่อมต่อเข้าด้วยกัน



รูปที่ 19 แสดงการถอดอักษรชื่อบุคคลไทยโดยใช้พจนานุกรมแกรมสะสม

กำหนดให้ ชื่อคนไทย $R = r_0 r_1 \ldots r_N$ เมื่อแต่ละ r_i เป็นตัวอักษรไทยและ N เป็นจำนวนตัวอักษรที่ใช้สะกดชื่อในภาษาไทยนั้น ระบบจะเริ่มแปลง R เป็นสายลำดับของ K กลุ่มตัวอักษรไทย จะได้ว่า $T = t_0 t_1 \ldots t_K$ เมื่อแต่ล่ะ t_i เป็นกลุ่มตัวอักษรไทยที่ใช้สะกดชื่อคนไทยของแกรม

ซึ่งปรากฏอยู่ในพจนานุกรมแกรมโดยใช้วิธีการเลือกจากขวาไปซ้ายของชื่อ และเลือกแกรมที่มีความยาวมากกว่าก่อน

จากนั้นระบบจะนำสายลำดับของ T ซึ่งตรงกันกับชื่อคนไทย R ไปคำนวณหาคะแนนความนิยมของการถอดอักษรของแต่ล่ะตัวเลือก $E = e_0e_1...e_K$ โดยที่แต่ล่ะ e_i คือ กลุ่มอักษรภาษาอังกฤษที่ใช้สะกดในแต่ละแกรม จะพบว่าจำนวนของ E ที่จะต้องเท่ากับ K ด้วยเนื่องจากแต่ละส่วนเป็นการจับคู่กันแบบ 1 ต่อ 1 นั่นคือ $t_i : e_i$ โดยที่ i = 0, 1, …, K ต้องเป็นแกรมที่อยู่ในพจนานุกรม ระบบจะทำการหาค่าความน่าจะเป็นของสายลำดับ E เมื่อกำหนด T ให้ ด้วยวิธีการเลือก MAP จะทำให้เราจะเลือกตัวเลือกที่ดีที่สุด E* จาก T ที่กำหนดให้

$$\boldsymbol{E}^* = \arg\max_{\boldsymbol{E}} p(\boldsymbol{E} \mid \boldsymbol{T}) = \arg\max_{\boldsymbol{E}} p(\boldsymbol{T} \mid \boldsymbol{E})\, p(\boldsymbol{E})$$

เมื่อพิจารณาค่า p(E) เป็น “N-Gram score” ซึ่งสามารถคำนวณค่าได้ด้วยแบบจำลองเอ็นแกรมของกลุ่มอักษรภาษาอังกฤษ ในงานวิจัยนี้เราใช้แบบจำลองไบแกรมซึ่งสร้างจากตัวอย่างชื่อในชุดข้อมูลฝึกที่เราแบ่งโดยเราจะพิจารณาเฉพาะชื่อในภาษาอังกฤษเท่านั้น หรือในอีกนัยหนึ่งคือการนับจำนวนของแกรมในแบบจำลองไบแกรม แกรมที่แตกต่างกันจะถูกมองเหมือนว่าเหมือนกันถ้าการสะกดในภาษาอังกฤษเหมือนกัน

จากปัญหาทั่วไปของปัญหาการแปลด้วยเครื่อง (Machine Translation) ในระดับประโยค เราจะมอง p(T|E) เป็นเหมือนคะแนนการแปลง โดยพิจารณาแบบมีความขึ้นต่อกันของ T โดยแต่ละ ti มีความขึ้นต่อกันเฉพาะกับ ei

$$p(\boldsymbol{T} \mid \boldsymbol{E}) = p(t_0t_1...t_K \mid e_0e_1...e_K) = \prod_{i=0}^{K} p(t_i \mid e_i)$$

และ

$$p(t_i \mid e_i) = \frac{N(t_i : e_i)}{\sum_{\text{all } \tau} N(\tau : e_i)}$$

เมื่อ $N(t_i{:}e_i)$ เป็นจำนวนของแกรม $t_i : e_i$ ในชุดข้อมูลฝึก และ $\sum_{\text{all } \tau} N(\tau : e_i)$ เป็นจำนวนของแกรมทั้งหมดของแกรม τ ซึ่งตรงกับการสะกดในภาษาอังกฤษเป็น e_i

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลอง

4.1.1 ฐานข้อมูลชื่อ

ในงานวิจัยนี้ได้นำฐานข้อมูลชื่อและนามสกุลของคนไทยมาใช้ในการทดลอง เพื่อเป็นทั้งชุดข้อมูลฝึกให้กับแบบจำลองทางสถิติและชุดข้อมูลสำหรับทดสอบประเมินผล ฐานข้อมูลชื่อบุคคลนี้ได้มาจากฐานข้อมูลของนักศึกษาที่ลงทะเบียนที่จุฬาลงกรณ์มหาวิทยาลัยในช่วง 10 ปีที่ผ่านมา (พ.ศ. 2541-2551) โดยตัดชื่อของนักศึกษาต่างชาติออกในระหว่างการประมวลผล ฐานข้อมูลนี้มีชื่อทั้งสิ้น 178,612 ชื่อซึ่งแต่ละชื่อมีชื่อที่สะกดด้วยตัวอักษรอังกฤษกำกับด้วย

ในการทดลองเราจะสุ่มเลือกชื่อออกมา 20% ของชื่อทั้งหมด โดยการนำชื่อมาเรียงลำดับตามตัวอักษรไทยก่อน จากนั้นชื่อในลำดับที่ 5 หารลงตัวจะถูกแยกออกมาก่อนเพื่อเก็บไว้เป็นชุดข้อมูลทดสอบ วิธีการนี้จะทำให้ได้ชื่อทดสอบที่มีการกระจายในทุกๆ ตัวอักษรไทย ชื่ออีก 80% ที่เหลือเราจะนำมาเป็นชุดข้อมูลฝึกซึ่งชื่อเหล่านี้จะนำมาใช้ในการสร้างพจนานุกรมแกรมสะสม และใช้ฝึกแบบจำลองทางสถิติในการหาค่าคะแนนเอ็นแกรม (N-Gram) รวมทั้ง ค่าคะแนนการแปล (Translation) ด้วย

4.1.1 การประเมินผล

ในงานวิจัยนี้วัดผลโดยทำการเปรียบเทียบประสิทธิภาพของวิธีการถอดอักษรที่ได้เสนอข้างต้นกับประสิทธิภาพการถอดอักษรของโปรแกรมประยุกต์ ซียูโรมันไนเซชั่น (CU Romanization) ที่มีอยู่ในปัจจุบัน เสนอโดย วิโรจน์ อรุณมานะกุล [9] ซึ่งในงานวิจัยนี้ใช้เป็นฐาน เราจะให้ชื่อที่สะกดโดยอักษรไทยแต่ละชื่อในชุดข้อมูลทดสอบเป็นข้อมูลเข้าของการถอดอักษรในทั้งวิธีการที่เราใช้เป็นฐาน และวิธีการที่เสนอในงานวิจัยนี้ เราจะพิจารณาว่าการถอดชื่อบุคคลจากอักษรไทยเป็นอักษรภาษาอังกฤษ ถูกต้อง ก็ต่อเมื่อผลลัพธ์ของชื่อที่สะกดด้วยตัวอักษรภาษาอังกฤษตรงกับชื่อภาษาอังกฤษในฐานข้อมูลที่คู่กันกับชื่อที่สะกดด้วยอักษรไทยชื่อนั้น

การประเมินผลของวิธีการที่เสนอแบ่งออกเป็น 2 กลุ่มตามการแบ่งชื่อออกเป็นสายลำดับของแกรม กลุ่มแรกคือกลุ่มที่ใช้การแบ่งชื่อบุคคลออกเป็นสายลำดับของแกรมโดยวิธีการที่ใช้ใน Aroonmanakun [11] และกลุ่มที่ 2 คือกลุ่มที่ใช้การแบ่งชื่อบุคคลโดยวิธีการจับคู่ชื่อของบุคคลกับแกรมสะสมในพจนานุกรมแบบขวาไปซ้าย และใช้แกรมที่มีจำนวนอักษรไทยมากกว่าก่อนแกรมที่

มีจำนวนตัวอักษรไทยน้อย จากนั้นชื่อของทั้งสองกลุ่มจะถูกนำไปถอดอักษรเป็นตัวเลือกของชื่อที่สะกดด้วยตัวอักษรในภาษาอังกฤษด้วยวิธีการทางสถิติและคำนวณออกมาเป็นคะแนนของเอ็นแกรม (N-Gram) และคะแนนของการแปลอักษรของแต่ละตัวเลือกตามที่กล่าวไว้ในบทที่ 3 เราจะนำตัวเลือกที่มีคะแนนสูงที่สุดจะมาเปรียบเทียบกับชื่อที่สะกดด้วยตัวอักษรภาษาอังกฤษที่ถูกต้อง

นอกจากการเปรียบเทียบเปอร์เซ็นความถูกต้อง (Accuracy) ของการถอดอักษรในภาพรวมของวิธีการที่นำเสนอกับวิธีการที่ใช้เป็นฐานแล้ว งานวิจัยนี้ยังสนใจในการสังเกตอัตราการค้นคืน (recall rate) ของการถอดอักษรเมื่อเราให้มีการใช้หลายตัวเลือกที่คะแนนสูงสุดกับแต่ละชื่อบุคคลมาประเมินแทนการใช้แค่เพียงตัวเลือกเดียว ชื่อที่สะกดด้วยอักษรภาษาอังกฤษจะถูกค้นคืนเมื่อตัวเลือกที่ถูกต้องจากการถอดอักษรของชื่อนั้นอยู่ระหว่าง N ตัวเลือกแรกที่คะแนนดีที่สุดจากวิธีการที่เสนอ โดยค่าของ N คือตั้งแต่ 1 ถึง 15 เราจะสังเกตการเปลี่ยนแปลงอัตราเปอร์เซ็นต์ค่าค้นคืนของทั้ง 2 กลุ่มการประเมิณที่กล่าวถึงก่อนหน้า

4.2 ผลการทดลอง

4.2.1 ผลการสร้างพจนานุกรม

ในการทดลองทำการแบ่งชื่อทีละรอบทั้งหมด 22 รอบ จำนวนชื่อที่นำมาทำการแบ่งในแต่ละรอบไม่เท่ากันเพื่อความสะดวกในการตรวจสอบและแก้ไขหลังจากที่ใช้โปรแกรมช่วยในการแบ่งแล้ว ในการแบ่งแต่ละรอบจะมีการนำแกรมที่ได้จากการแบ่งรอบก่อนหน้าทั้งหมดมาช่วยในการแบ่งได้ผลดังแสดงในตารางที่ 7

ผลการแบ่งพยางค์ของชื่อในแต่ละรอบแบ่งออกเป็น 3 กลุ่มได้แก่

กลุ่ม 1 แบ่งได้สมบูรณ์ คือ จำนวนชื่อที่สามารถแบ่งได้สมบูร์ด้วยแกรมในพจนานุกรมเท่านั้น

กลุ่ม 2 แบ่งผิดพลาด 1 แกรม คือ จำนวนชื่อที่แบ่งด้วยแกรมสะสมแล้ว แกรมแรกไม่มีอยู่ในพจนานุกรม

กลุ่ม 3 คือ แบ่งผิดพลาดมากกว่า 1 แกรม คือ จำนวนชื่อที่แบ่งด้วยแกรมในพจนานุกรมแล้ว มีมากกว่า 1 แกรมที่ไม่อยู่ในพจนานุกรม

คอลัมน์เปอร์เซ็นต์แบ่งสมบูรณ์ หมายถึง อัตราส่วนระหว่างจำนวนชื่อในกลุ่มที่ 1 ต่อจำนวนชื่อทั้งหมดในการแบ่งรอบนั้น ซึ่งสะท้อนให้เห็นถึงผลการแบ่งชื่อที่สมบูรณ์จากจำนวนแกรมที่เพิ่มขึ้นในพจนานุกรม

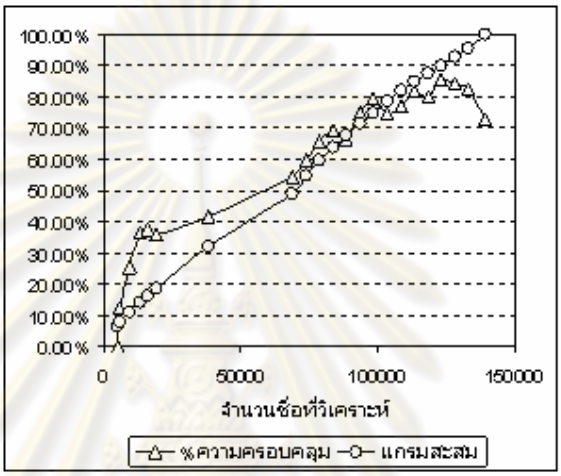
ตารางที่ 7 จำนวนชื่อที่ทำการแบ่งในแต่ละรอบ

รอบที่	ผลการแบ่ง (ชื่อ)			จำนวน (ชื่อ)	จำนวนชื่อสะสม	% แบ่งสมบรูณ์	จำนวนแกรมสะสม
	กลุ่ม 1	กลุ่ม 2	กลุ่ม 3				
1	0	498	4,509	5,007	5,007	0.00	1,567
2	125	607	266	998	6,005	12.53	1,969
3	868	1,688	840	3,396	9,401	25.56	2,694
4	1,458	1,756	760	3,974	13,375	36.69	3,464
5	1,102	1,498	329	2,929	16,304	37.62	3,938
6	1,058	1,620	260	2,938	19,242	36.01	4,564
7	7,929	8,492	2570	18,991	38,233	41.75	7,804
8	16,713	11,010	3003	30,726	68,959	54.39	11,917
9	2,954	659	1,351	4,964	73,923	59.51	13,404
10	3,215	787	908	4,910	78,833	65.48	14,536
11	3,452	810	720	4,982	83,815	69.29	15,547
12	3,243	874	744	4,861	88,676	66.71	16,457
13	3,711	631	625	4,967	93,643	74.71	17,395
14	3,918	323	705	4,946	98,589	79.22	18,213
15	3,706	617	642	4,965	103,554	74.64	19,159
16	3,829	598	523	4,950	108,504	77.35	19,972
17	3,996	416	468	4,880	113,384	81.89	20,674
18	3,958	422	577	4,957	118,341	79.85	21,356
19	4,242	195	523	4,960	123,301	85.52	21,950
20	4,190	340	444	4,974	128,275	84.24	22,606
21	4,089	449	420	4,958	133,233	82.47	23,264
22	4,566	1,031	670	6,267	139,500	72.86	24,385

จากกราฟรูปที่ 20 เปรียบเทียบระหว่างเปอร์เซ็นต์การแบ่งสมบูรณ์ กับ จำนวนแกรมสะสมในพจนานุกรมที่แสดงไว้ดังตารางด้านบน เราจะสังเกตเห็นว่าหลังจากการประมวลผลในชุดข้อมูลฝึกครบทั้ง 22 รอบ รวมทั้งทำการตรวจ/แก้ไขการแบ่งชื่อที่เกิดความผิดพลาดในแต่ละรอบเองแล้ว มี 24,385 แกรมที่ได้จากชื่อทั้งสิ้น 139,500 ชื่อ ประกอบด้วย 24,385 แกรม ในจำนวนนี้เป็นแกรมที่เขียนเป็นภาษาไทยแบบไม่ซ้ำกันประมาณ 8,000 ที่ใช้ในการแบ่งชื่อที่เขียนด้วย

อักษรไทย และเป็นแกรมที่เขียนเป็นอักษรอังกฤษที่ไม่ซ้ำกันประมาณ 12,000 แกรมที่ใช้ในการแบ่งชื่อที่เขียนด้วยอักษรอังกฤษในชุดข้อมูลฝึก

นอกจากนี้เมื่อพิจารณาจากกราฟ แล้วพบว่าจำนวนแกรมสะสมมีการเพิ่มขึ้นใกล้เคียงสมการเส้นตรง โดยเพิ่มขึ้นควบคู่ไปกับจำนวนชื่อที่เพิ่มขึ้นเมื่อแกรมสะสมในพจนานุกรมมีปริมาณได้ระดับหนึ่งแล้ว (ประมาณ 35 % ในการทดลองนี้) และยังพบว่าเปอร์เซ็นการแบ่งชื่อได้สมบูรณ์มีการเพิ่มขึ้นแบบสมการเส้นตรงเมื่อจำนวนของชื่อที่แบ่งแล้วเพิ่มขึ้นอีกด้วย



รูปที่ 20 กราฟความสัมพันธ์ของแกรมสะสมที่เพิ่มขึ้นกับเปอร์เซ็นต์ความครอบคลุมเมื่อจำนวนชื่อที่ถูกประมวลผลเพิ่มขึ้น

4.2.2 ผลการถอดอักษร

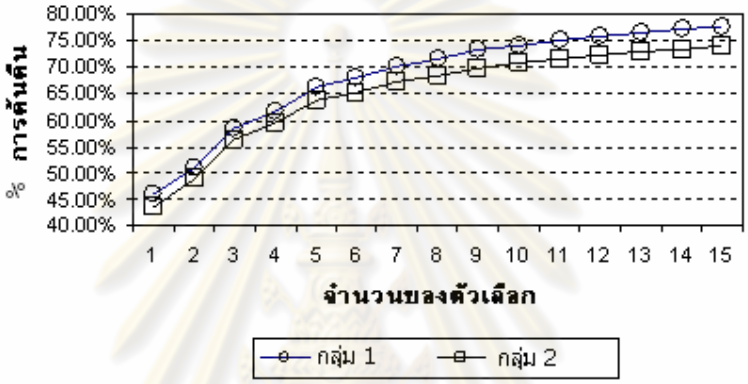
จากตารางที่ 8 แสดงการเปรียบเทียบความถูกต้อง (Accuracy) ของการถอดอักษรระหว่างวิธีที่ใช้เป็นฐานและวิธีการที่งานวิจัยนี้เสนอโดยเปรียบเทียบจากผลการถอดอักษรชื่อบุคคลในชุดทดสอบจำนวน 35,722 ชื่อ

ตารางที่ 8 ผลความถูกต้อง (Accuracy) ของการถอดอักษรกับชื่อในชุดทดสอบ

	วิธีที่ใช้เป็นฐาน (Baseline)	วิธีการที่เสนอ	
		กลุ่มที่ 1	กลุ่มที่ 2
ความถูกต้องของถอดอักษร (Accuracy)	18.20%	46.13%	43.66%

ทั้งกลุ่มที่ 1 และ กลุ่มที่ 2 ใช้วิธีการถอดอักษรที่เสนอในงานวิจัยนี้ แต่ต่างกันที่ขั้นตอนการแบ่งชื่อภาษาไทยโดย กลุ่มที่ 1 แบ่งโดยวิธีการของ Aroonmanakun [11] ส่วนกลุ่มที่ 2 แบ่งโดยใช้

พจนานุกรมแกรมสะสม ผลปรากฏว่าการถอดชื่อบุคคลของวิธีการที่ใช้เป็นฐานให้ความถูกต้องของการถอดอักษรประมาณ 18% แสดงให้เห็นว่ามากกว่า 80% ของชื่อบุคคลทำการถอดอักษรแตกต่างไปจากที่บุคคลเหล่านั้นใช้จริงๆ เป็นการสะท้อนความซับซ้อนในการถอดชื่อบุคคลของไทย อย่างไรก็ตามความแตกต่างกันอย่างมากของธรรมชาติการถอดอักษรของชื่อคนไทย ทำให้เราเห็นข้อด้อยของวิธีการถอดอักษรของระบบที่ใช้เป็นฐาน ซึ่งมองเพียงการจับคู่เสียงที่ใกล้เคียงกันที่สุดของ 2 ระบบเสียง แต่ไม่ได้พิจารณาถึงความยืดหยุ่นที่มีในการใช้จริง ต่างจากวิธีที่นำเสนอซึ่งให้ผลเกือบ 50% หรือมากกว่าวิธีการที่ใช้เป็นฐานกว่าเท่าตัวของชื่อทดสอบชุดเดียวกัน



รูปที่ 21 กราฟเปอร์เซ็นต์การค้นคืนของวิธีที่เสนอทั้ง 2 กลุ่ม เมื่อจำนวนตัวเลือก (hypotheses) เพิ่มขึ้น

รูปที่ 21 แสดงเปอร์เซ็นต์อัตราการค้นคืนเมื่อใช้ N จำนวนของตัวเลือกที่ดีที่สุด จะเห็นได้ว่ากลุ่มที่หนึ่งของวิธีการที่เสนอให้ผลอัตราการค้นคืนสูงกว่าในทุกๆ ค่าของ N ที่ทดสอบ และการทดลองทั้งสองกลุ่มมีแนวโน้มที่เหมือนกันคืออัตราการค้นคืนมีการเพิ่มขึ้นจากที่ระดับประมาณ 44% - 46% เปอร์เซ็นถึงประมาณ 75 % - 80 % เมื่อใช้ 15 ตัวเลือกที่ดีที่สุด ถึงแม้ว่าอัตราการค้นคืนจะเพิ่มขึ้นเมื่อจำนวนตัวเลือกที่ใช้เพิ่มขึ้นแต่เราจะสังเกตเห็นว่ากราฟเริ่มจะคงตัวแล้ว ดังนั้นเราสามารถคาดเดาการณ์ได้ว่าการใช้ตัวเลือกมากขึ้นกว่านี้จะไม่เป็นการเพิ่มอัตราการค้นคืนให้มากขึ้นจากที่แสดงในกราฟ กล่าวได้ว่าการใช้ N ตัวเลือกแรกที่ดีที่สุดไม่ได้เป็นการเลือกชื่อที่ทำการถอดอักษรได้ถูก เราอาจสามารถหวังเปอร์เซ็นต์อัตราการค้นคืนได้ถึง 80% เมื่อเราประยุกต์เข้ากับการใช้โปรแกรมค้นหาประยุกต์ (Search Application) ซึ่งไม่มีตัวเลือกของการทำถอดอักษรเป็นคำค้นหาที่ไม่มากจนเกินไปจากที่เสนอ

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้เสนอวิธีการถอดอักษร (Thai Romanization) ที่มีความยืดหยุ่นและเหมาะสมกับการถอดอักษรกับชื่อคนไทยด้วยวิธีการใช้ความนิยมเป็นฐานในการถอด และได้เสนอวิธีการสร้างพจนานุกรมจากฐานข้อมูลชื่อบุคคลขนาดใหญ่ และทำการแบ่งชื่อเป็นสายลำดับของหน่วยย่อยพื้นฐานไปในขณะเดียวกัน ในงานวิจัยนี้ชื่อบุคคลจะถูกมองเป็นสายลำดับของแกรมซึ่งเป็นหน่วยย่อยที่ประกอบด้วยการสะกดด้วยตัวอักษรไทยและสะกดด้วยอักษรอังกฤษที่ออกเสียงคล้ายกัน แบบจำลองความน่าจะเป็นถูกสร้างขึ้นบนพื้นฐานของชุดของแกรม วิธีการนี้ให้ผลที่น่าพอใจในการค้นหาผลลัพธ์ของการถอดอักษรและเช่นเดียวกับกรอบงานการสร้างพจนานุกรมแกรมสะสมจากตัวอย่างของชื่อบุคคลมากกว่าแสนชื่อ

5.2 อภิปรายผลการวิจัย

ในงานวิจัยนี้เสนอวิธีการถอดชื่อบุคคลโดยอาศัยความนิยมในการใช้เป็นฐาน ให้ผลความถูกต้องของการถอดชื่อบุคคล 46 – 75 % ซึ่งให้ผลดีกว่าเมื่อเทียบกับวิธีที่ใช้เป็นฐานซึ่งให้ผลความถูกต้องในการถอดชื่อบุคคล 18% สาเหตุที่วิธีการที่เสนอให้ผลดีกว่านั้นเป็นเพราะการถอดชื่อของบุคคลไทยมีความหลากหลายในการถอด แต่วิธีการถอดอักษรของงานวิจัยที่ใช้เป็นฐานถอดตามกฎการถอดอักษรแบบถ่ายเสียงที่ราชบัณฑิตแนะนำ ทำให้การถอดชื่อบุคคลไทยของงานวิจัยที่ใช้เป็นฐานไม่ตรงกับที่บุคคลนั้นถอด นอกจากนี้ข้อมูลที่ใช้ฝึกของงานวิจัยที่ใช้เป็นฐานยังเป็นบริบททั่วไป ทำให้เกิดความผิดพลาดในการออกเสียง อะ กึ่งเสียงที่เชื่อมระหว่างคำในชื่อของบุคคลไทย เช่น ชื่อบุคคล รัฐพล (รัด-ทะ-พน) มี 3 พยางค์โดยมี ทะ เป็นพยางค์ที่มีสระกึ่งเสียงแต่เนื่องจากในบริบททั่วไปคำว่า รัฐ มักออกเสียงพยางค์เดียว งานวิจัยที่ใช้เป็นฐานจึงถอดอักษรเป็น Rat Phon ทำให้เกิดความผิดพลาดขึ้น

ผลการถอดอักษรของวิธีที่งานวิจัยนี้เสนอมีความแตกต่างกันมากเมื่อจำนวนตัวเลือกเป็น 1 ตัวเลือกและ 15 ตัวเลือก แสดงให้เห็นว่าวิธีการถอดชื่อของบุคคลไทยมีความหลากหลาย ชื่อบุคคลไทย 1 ชื่อ สามารถถอดได้หลายแบบ ความผิดพลาดที่เกิดขึ้นของวิธีที่เสนอนี้ส่วนใหญ่ขึ้นอยู่กับการถอดชื่อของบุคคล ว่าถอดตามแบบที่บุคคลส่วนมากนิยมหรือไม่ และสังเกตได้ว่าการใช้เพียง 1 ตัวเลือกที่ดีที่สุดของการถอดชื่อบุคคลให้ความถูกต้อง 46 % ดังนั้นในการนำชื่อที่ได้รับการถอดอักษรด้วยวิธีการที่เสนอไปค้นหาเอกสารภาษาอังกฤษ ชื่อที่เป็นตัวเลือกจำนวน 15 ชื่อจะ

ทำให้ผลลัพธ์การค้นหาไม่ดีเท่าที่ควร การใช้ความรู้ทางภาษาศาสตร์เข้ามาช่วยจะทำให้ลดจำนวนชื่อที่เป็นตัวเลือกได้

ปัญหาที่พบในงานวิจัยนี้ส่วนมากอยู่ในช่วงของการสร้างพจนานุกรมแกรมสะสม เมื่อมีการแบ่งแกรมผิดพลาดเกิดขึ้นและตรวจสอบไม่พบ จะทำให้มีแกรมสะสมที่ไม่เหมาะสมเข้าไปอยู่ในพจนานุกรมและถูกใช้ในการแบ่งชื่อบุคคลในการแบ่งชื่อข้อมูลรอบต่อๆ ไป การพบแกรมที่ไม่เหมาะสมในภายหลังจะทำให้เสียเวลาในการกลับมาแก้ไข นอกจากนี้ยังมีบางชื่อที่ต้องถูกคัดออกจากข้อมูลฝึกเนื่องจากไม่สามารถแบ่งแกรมได้อย่างเหมาะสม เช่น ชื่อ มัชณมน ถอดเป็น mashamon จะเห็นว่าชื่อภาษาไทยกับชื่อภาษาอังกฤษอ่านออกเสียงไม่สัมพันธ์กัน

5.3 **ข้อเสนอแนะ**

ในการวิจัยนี้ยังมีความผิดพลาดส่วนหนึ่งที่เกิดขึ้นจากแกรมที่ไม่มีอยู่ในพจนานุกรม การเพิ่มฐานข้อมูลชื่ออาจเป็นทางแก้หนึ่งที่ช่วยลดความผิดพลาดได้เมื่อพจนานุกรมแกรมสะสมมีขนาดใหญ่ระดับหนึ่ง แต่ไม่สามารถแก้ปัญหาชื่อใหม่ๆ ที่เกิดขึ้นได้โดยเฉพาะชื่อที่มีรากศัพท์มาจากภาษาบาลีและสันสกฤตซึ่งเป็นที่นิยมของคนไทย การสร้างแกรมขึ้นเลียนแบบแกรมในพจนานุกรมที่มีความใกล้เคียงกันน่าจะเป็นแนวทางหนึ่งที่ช่วยแก้ไขปัญหาได้

ในการวิจัยนี้ผู้ทดลองไม่ได้ทดลองนำชื่อที่ได้จากการถอดอักษรไปทดลองหาเอกสารภาษาอังกฤษ การเพิ่มเงื่อนไขในการค้นหาบางอย่าง เช่น ใช้รูปแบบการค้นหาด้วย ชื่อจริง.อักษรของนามสกุลตัวแรก หรือ อักษรตัวแรกของชื่อจริง.นามสกุล จะช่วยเพิ่มโอกาสความถูกต้องในการค้นหามากยิ่งขึ้น แต่ทั้งนี้ไม่ควรใช้ชื่อบุคคลที่เป็นตัวเลือกจากการถอดอักษรเกิน 15 ชื่อ เนื่องจากจะทำให้เอกสารที่ได้จากการค้นหามีมากขึ้นและทำให้ความถูกต้องของการค้นหาลดลง

รายการอ้างอิง

[1] Pie, A.M., and Gaynor, F. Dictionary of Linguishes. London : Peter Owen, 1958

[2] ดร. กาญจนา นาคสกุล. ระบบเสียงภาษาไทย. กรุงเทพมหานคร: โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2520

[3] ภาทิพ ศรีสุทธิ์ "อักษรไทย และการผันวรรณยุกต์." [ออนไลน์]. แหล่งที่มา: http://www.st.ac.th/bhatips/gramma3.htm [29 เมษายน 2549]

[4] Luksaneeyanawin, S. Speech computing and speech technology in Thailand. In Proceedings of the Symposium on Natural Language Processing. 1993, 276--321.

[5] Knight, K., and Philipp, K. What's New in Statistical Machine Translation, Tutorial at HLT/NAACL 2004.

[6] “Statistical Machine Translation System.” [online]. Available from: http://www.statmt.org/moses/ 2007 [2009, Apr 29]

[7] Junrafsky, D., and Marin, J.H. Speech and language processing : An introduction to Natural Language Processing, Computational Lingu. Upper Saddle River, N.J. : Prentice Hall, c2002.

[8] Poowarawan, Y., Dictionary-based Thai Syllable Separation, Proceedings of the Ninth Electronics Engineering Conference, 1986.

[9] Sornlertlamvanich, V., Word Segmentation for Thai in a Machine Translation System (in Thai), Papers on Natural Language processing, NECTEC, Thailand, 1995.

[10] Theeramunkong, T., Tanhermhong, T., Phatharakittikul, D., and Sangvareethip, A. Non-Dictionary-Based Word Segmentation Using Local Context Statistics. Proceedings of the 5th Symposium on Natural Language Processing and Oriental COCOSDA Workshop, May 2002, Hua Hin, Thailand, pp. 81-88.

[11] Aroonmanakun, W. 2002. Collocation and Thai Word Segmentation. In Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop, Hua Hin, Thailand, pp. 68-75.

[12] Aroonmanakun, W., and Rivepiboon, W. A Unified Model of Thai Word Segmentation and Romanization. In Proceedings of The 18th PACLIC, Tokyo, Japan. 2004.

[13] อุไรรัตน์ บุญภานนท์. การถอดอักษรอังกฤษเป็นไทยโดยใช้หลักวิชาภาษาศาสตร์. วิทยานิพนธ์ปริญญามหาบัณฑิต, ภาควิชาภาษาศาสตร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2529

ภาคผนวก

ภาคผนวก ก.
การถอดอักษรไทยเป็นอักษรโรมันแบบถ่ายเสียงราชบัณฑิตยสถาน

ตารางที่ 9 ตารางการถอดอักษรไทยเป็นภาษาโรมันแบบถ่ายเสียงราชบัณฑิตยสถาน

พยัญชนะ	รูปโรมัน		รูปสระ	รูปโรมัน
	พยัญชนะต้น	ตัวสะกด		
ก	k	k	อะ, ั (reduced form of อะ), รร (with final consonant), อา	a
ข ฃ ค ฅ ฆ	kh	k	รร (without final consonant)	an
ง	ng	ng	อำ	am
ซ ทร (pronounced ซ) ศ ษ ส	s	t	อิ, อี	i
ญ	y	t	อึ, อื	ue
ฎ ฑ (pronounced ด) ด	d	t	อุ, อู	u
ฏ ต	t	t	เอะ, เ็ (reduced form of เอะ), เอ	e
ฐ ฑ ฒ ถ ท ธ	th	t	แอะ, แอ	ae
ณ น	n	n	โอะ, –(reduced form of โอะ), โอ, เอาะ, ออ	o
บ	b	p	เออะ, เ ิ (reduced form of เออะ), เออ	oe
ป	p	p	เอียะ, เอีย	ia
ผ พ ภ	ph	p	เอือะ, เอือ	uea
ฝ ฟ	f	p	อัวะ, อัว, –ว– (reduced form of อัว)	ua
ม	m	m	ใอ, ไอ, อัย, ไอย, อาย	ai
ย	y		เอา, อาว	ao
ร	r	n	อุย	ui
ล ฬ	l	n	โอย, ออย	oi
ว	w		เอย	oei
ห ฮ	h		เอือย	ueai
			อวย	uai
			อิว	io
			เอ็ว, เอว	eo
			แอ็ว, แอว	aeo
			เอียว	iao
			ฤ (pronounced รึ), ฤๅ	rue
			ฤ (pronounced ริ)	ri
			ฤ (pronounced เรอ)	roe
			ฦ, ฦๅ	lue

ภาคผนวก ข
ตัวอย่าง 600 รายชื่อในชุดข้อมูลฝึกที่แบ่งเป็นสายลำดับของแกรม

ชื่อภาษาไทย	ชื่อภาษาอังกฤษ	ชื่อภาษาไทย	ชื่อภาษาอังกฤษ
กร ชุ ลี	korn shu lee	กฤติ ยา	kriti ya
กรณ์ ดนัย	korn danai	กฤติ ยา	kritti ya
กรณ์ นภา	korn napar	กฤติ ยา	kitti ya
กร ณพ	korra nop	กฤติ ยา ณี	kritti ya nee
กรณ์ พิ มล	korn phi mol	กฤติ ยา ภรณ์	kritti ya pon
กร ณิ กา	korn ni ka	กฤป นนท์	krippa non
กร ณิ การ์	korn ni ka	กฤศ ณ ภัทร	krit na pat
กร ณิ การ์	korn ni kar	กฤศ ณ ภัทร	krit na pat
กร ณิ การ์	korn ni kar	กฤศ นัน	krisa nan
กร ณิศ	kara nis	กฤศ วรรณ	gridsa wan
กร ณิศ	kora nit	กฤษ กมล	kris kamol
กร ณิศ	kora nis	กฤษ กร	kritsa korn
กร ดา	kora da	กฤษ กร	krisa korn
กร ทอง	korn thong	กฤษ กร	kritsa korn
กร ทอง	korn thong	กฤษ กร	kitsa korn
กร ทอง	gorn thong	กฤษ กาญจน์	kritsa kan
กร ทิพ	korn tip	กฤษ ฎา	krisa da
กร ทิพย์	korn tip	กฤษ ฎา	kritsa da
กร ทิพย์	korn tip	กฤษ ฎา	kritsa da
กร ทิพย์	korn tip	กฤษ ฎา	kritsa da
กร ทิพย์	korn thip	กฤษ ฎา	kritsa da
กร ทิพย์	korn tip	กฤษ ฎา	kitsa da
กร ทิพย์	korn thip	กฤษ ฎา	kris da
กร ทิพย์	korn thip	กฤษ ฎา	krissa da
กร ธนินทร์	kon tanin	กฤษ ฎา	gridsa da
กร นด	korra nod	กฤษ ฎา	krissa da

กร นบ	korn nop	กฤษ ฎา	kritsa da
กร นันท์	korra nun	กฤษ ฎา	kritsa da
กร นันท์	kora nan	กฤษ ฎา	kritsa da
กร นันท์	kora nand	กฤษ ฎา	krisa da
กร นันท์	kora nan	กฤษ ฎา	kridsa da
กร นิ กา	korn ni ka	กฤษ ฎา	kritsa da
กร นิจ	gora nij	กฤษ ฎา	kris da
กร นิจ	gora nij	กฤษ ฎา	kritsa da
กร นิ ภา	kor ni pa	กฤษ ฎา	krissa da
กร นุช	kora nuch	กฤษ ฎา	kritsa da
กร บง กช	korn bong kot	กฤษ ฎา	kritsa da
กร บง กช	korn bong kot	กฤษ ฎา	kritsa da
กร บง กช	korn bong koj	กฤษ ฎา	kidsa da
กร ประ ภา	korn pra bha	กฤษ ฎา	kritsa da
กร ปรี ยา	korn pree ya	กฤษ ฎา	krisa da
กร ปรี ยา	gron pree ya	กฤษ ฎา	krissa da
กร พงศ์	korn pong	กฤษ ฎา	kridsa da
กร พงษ์	korn pong	กฤษ ฎา	krissa da
กร พรรณ	korra pun	กฤษ ฎา	krisa da
กร พล	korn pol	กฤษ ฎา	krissa da
กร พิน	kora pin	กฤษ ฎา	krissa da
กร พินธ์	kora pin	กฤษ ฎา	kridsa da
กร พินธ์	kora pin	กฤษ ฎา	kridsa da
กร พินธ์	kora pin	กฤษ ฎา	krids da
กร พินธุ์	kora pin	กฤษ ฎา	kritsa da
กร พินธุ์	kora pin	กฤษ ฎา	kris da
กร พินธุ์	kora pin	กฤษ ฎา	kridsa da
กร พินธุ์	khora pin	กฤษ ฎา	krisa da
กร พินธุ์	kora pin	กฤษ ฎา	gridsa da
กร พินธุ์	korra pin	กฤษ ฎา	krisa da

กร พินธุ์	kora pin	กฤษ ฎา	krisa da
กร เพชร	korn petch	กฤษ ฎา	krissa da
กร ภัค	korn pak	กฤษ ฎา	krisa da
กร ภัทร	kora pat	กฤษ ฎา	kritsa da
กร ภัทร	kora pat	กฤษ ฎา	kritsa da
กร ภัทร	kora pat	กฤษ ฎา	krisa da
กร ภัทร	korn pat	กฤษ ฎา	krissa da
กรม ทรัพย์	krom sub	กฤษ ฎา	kriss da
กร มิษฐ์	kora mit	กฤษ ฎา	krisa da
กรรช รส	kancha ros	กฤษ ฎา	krisa dar
กรรช รส	kancha rose	กฤษ ฎา	krissa da
กรรช รัตน์	kancha ratana	กฤษ ฎา	kritsa da
กรรณ กมล	kan kamol	กฤษ ฎา	krissa da
กรรณ งาม	kan ngam	กฤษ ฎา	kritsa da
กรรณ ทิพย์	kan tip	กฤษ ฎา	krisa da
กรรณ ภรณ์	kunna porn	กฤษ ฎา	krisa da
กรรณ วัฒน์	kanna wat	กฤษ ฎา	kritsa da
กรร ณา ภรณ์	kan na porn	กฤษ ฎา	kitsa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	krisa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	krissa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	kritsa da
กรร ณิ กา	kun ni ka	กฤษ ฎา	krissa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	kritsa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	kritsa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	kritsa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	krids da
กรร ณิ กา	kan ni ka	กฤษ ฎา	krissa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	kritsa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	krisa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	kritsa da

กรร ณิ กา	kan ni kar	กฤษ ฎา	kritsa da
กรร ณิ กา	ka ni ka	กฤษ ฎา	krissa da
กรร ณิ กา	kan ni ka	กฤษ ฎา	krisa da
กรร ณิ กา	kun ni ka	กฤษ ฎา	krissa da
กรร ณิ กา	kan ni ka	กฤษ ฎา กร	krisa da korn
กรร ณิ กา	kun ni ka	กฤษ ฎา กร	krisa da korn
กรร ณิ กา	kan ni ka	กฤษ ฎา กร	kritsa da korn
กรร ณิ กา	kan ni ka	กฤษ ฎา กรณ์	kidsa da gon
กรร ณิ กา	kan ni ga	กฤษ ฎางค์	krisa dang
กรร ณิ กา	kan ni ka	กฤษ ฎา พร	kritsa da porn
กรร ณิ กา	kan ni ka	กฤษ ฎา ภรณ์	kridsa da porn
กรร ณิ กา	kan ni ka	กฤษ ฎา ภรณ์	kridsa da porn
กรร ณิ กา	kan ni ka	กฤษฎี์ รุจ	kritsa rhut
กรร ณิ กา	kan ni ka	กฤษ ฎี	krisa dee
กรร ณิ กา	kan ni ka	กฤษ ฎี	kritsa dee
กรร ณิ กา	kan ni ka	กฤษ ฎี	kritsa dee
กรร ณิ กา	kan ni ka	กฤษ ฎี กร	krissa dee korn
กรร ณิ กา	kan ni ka	กฤษฎ พัฒ์	krita pat
กรร ณิ กา	kan ni ka	กฤษ ฎา	krisa da
กรร ณิ กา	kan ni ka	กฤษฎี์ ญา ณพ	christ ya nov
กรร ณิ กาณ์	kan ni ka	กฤษณ	krissana
กรร ณิ กาณ์	kan ni ka	กฤษณ	kitsana
กรร ณิ การ์	kan ni kar	กฤษณ	kitsana
กรร ณิ การ์	kan ni ka	กฤษ ณ กมล	kritsa na kamon
กรร ณิ การ์	gun ni gar	กฤษณ์ กร	krisa korn
กรร ณิ การ์	kan ni kar	กฤษณ์ กร	kritsa korn
กรร ณิ การ์	kan ni ka	กฤษณ์ ก้อง	krit kong
กรร ณิ การ์	kan ni kar	กฤษณ์ ชัย	krit chai
กรร ณิ การ์	kun ni ka	กฤษณ ชัย	krisna chai
กรร ณิ การ์	kan ni ka	กฤษณ ณัฐ	kritsana nut

กรร ณิ การ์	ka ni kar	กฤษณ นันท์	krisana nan
กรร ณิ การ์	kan ni kar	กฤษณ พงศ์	krissana pong
กรร ณิ การ์	kan ni ka	กฤษณ พงศ์	krissana pong
กรร ณิ การ์	kan ni kar	กฤษณ พงศ์	kritsana phong
กรร ณิ การ์	kan ni kar	กฤษณ พงศ์	krissana pong
กรร ณิ การ์	kan ni kar	กฤษณ์ พงษ์	krisa pong
กรร ณิ การ์	kan ni ka	กฤษณ พงษ์	kritsana pong
กรร ณิ การ์	kan ni ka	กฤษณ พงษ์	kitsana pong
กรร ณิ การ์	kan ni kar	กฤษณ พล	krishna pol
กรร ณิ การ์	kan ni kar	กฤษณ พล	kritsana pol
กรร ณิ การ์	kun ni ka	กฤษณ พล	krishna pol
กรร ณิ การ์	kan ni ka	กฤษ ณรงค์	gris narong
กรร ณิ การ์	kan ni ka	กฤษ ณรงค์	gris narong
กรร ณิ การ์	kan ni kar	กฤษณ รัฐ	krissana rut
กรร ณิ การ์	kan ni ka	กฤษณ วรรณ	kitsana wan
กรร ณิ การ์	kan ni ka	กฤษณ ศักดิ์	krissana sak
กรร ณิ การ์	kan ni ka	กฤษ ณะ	krisa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	krish na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	krissa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	krish na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	klis sana
กรร ณิ การ์	kan ni kar	กฤษ ณะ	krissa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	kritsa na
กรร ณิ การ์	kun ni ka	กฤษ ณะ	krissa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	krisa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	kriss na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	krish na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	krisa na
กรร ณิ การ์	kun ni ka	กฤษ ณะ	kritsa na

กรร ณิ การ์	kan ni ka	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	kritsa na
กรร ณิ การ์	kun ni ka	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	kridsa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	krisha na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	kidsa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	kridsa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	krisha na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	krissa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	krissa na
กรร ณิ การ์	gun ni gar	กฤษ ณะ	kritsa na
กรร ณิ การ์	kun ni ka	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	krisa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	krissa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni ka	กฤษ ณะ	kritsa na
กรร ณิ การ์	kan ni kar	กฤษ ณะ	kritsa na
กรร ณิ การ์ ทิพย์	kun ni gar tip	กฤษ ณะ	grisa na
กรรม มันต์	kum mun	กฤษ ณะ	krisa na
กรร วิ ภา	kan wi pa	กฤษ ณะ	kritsa na
กร รัฐา	korn ratha	กฤษ ณะ	kritsa na
กร ลักษณ์	kora luk	กฤษ ณัฐา	krit nuttha
กร ลักษณ์	korn lak	กฤษ ณันท์	kritsa nun
กร วรรณ	korra wan	กฤษ ณัส	kritsa nat
กร วรรณ	kora wan	กฤษณา ภรณ์	kritsana porn

กร วรรธน์	kora vat	กฤษ ณี	kitsa nee
กร วรรธน์	kora vat	กฤษ ณี	krisa nee
กร วรา	korn wara	กฤษ ณี	krisa nee
กร วฤนท์	kora warin	กฤษ ณี	kritsa nee
กร วฤนท์	kora warin	กฤษ ณี	krisa nee
กร วลัย	korn valai	กฤษ ณี	kritsa nee
กร วลัย	korn valai	กฤษ ณี	krisa nee
กร วสา	korn wasa	กฤษ ณี	kritsa nee
กร วัลลิ์	kora wan	กฤษ ณี	kritsa nee
กร วิก	gara vig	กฤษ ณี	kritsa nee
กร วิก	kora vik	กฤษ ณีย์	krissa nee
กร วิก	gara vig	กฤษ ดา	kris da
กร วิ กร	kon vi kon	กฤษ ดา	kritsa da
กร วิ กา	korn vi ca	กฤษ ดา	krissa da
กร วิ กา	korn vi ka	กฤษ ดา	krisa da
กร วิ กา	khon wi ka	กฤษ ดา	krisa da
กร วิ กา	korn vi ka	กฤษ ดา	krisa da
กร วิ กา	korn wi ga	กฤษ ดา	grisa da
กร วิ กา	korn vi ka	กฤษ ดา	krisa da
กร วิ กา	korn wi ka	กฤษ ดา	kritsa da
กร วิ กา	korn wi ka	กฤษ ดา	krisa da
กร วิ กา	korn vi ka	กฤษ ดา	krisa da
กร วิ กา	kara vi kar	กฤษ ดา	kritsa da
กร วิ กา	korn vi ka	กฤษ ดา	krisa da
กร วิ กา	korn vi ga	กฤษ ดา	kritsa da
กร วิ กา	korn wi ka	กฤษ ดา	kridsa da
กร วิ กา	korn wi ka	กฤษ ดา	krissa da
กร วิ กา	korn wi ka	กฤษ ดา	kris da
กร วิ กา	korn wi ka	กฤษ ดา	chris da
กร วิชช์	kora wij	กฤษ ดา	krissa da

กร วิชญ์	khora wit	กฤษ ดา	krisa da
กร วิชญ์	kora wit	กฤษ ดา	krisa da
กร วิทย์	korra vit	กฤษ ดา	krisa da
กร วิทย์	korn wit	กฤษ ดา	krisa da
กร วิ ภา	korn wi pa	กฤษ ดา	kridsa da
กร วิ ภา	kon wi pha	กฤษ ดา	kritsa da
กร วิ ภา	korn vi pa	กฤษ ดา	kritsa da
กร วีร์	kora wee	กฤษ ดา	kris da
กร วีร์	korra wee	กฤษ ดา	kritsa da
กร วีร์	kora wee	กฤษ ดา	kitsa da
กร ศจี	korn sajee	กฤษ ดา	kritsa da
กร ศิ ริ	korn si ri	กฤษ ดา	krissa da
กร ศุ กล	korn su kol	กฤษ ดา	kitsa da
กร ศุ ลี	korn su lee	กฤษ ดา	kritsa da
กร สิ นี	korn si nee	กฤษ ดา กร	kritsa da korn
กร สุทธิ์	gala sut	กฤษ ดา กร	kris da korn
กร สุ มา	korn su ma	กฤษ ดา กร	kritsa da korn
กรอง กมล	krong kamol	กฤษ ดา กร	kritsa da korn
กรอง กมล	krong kamol	กฤษ ดา พันธ์	kris da bhan
กรอง กมล	krong kamol	กฤษ ดา รัตน์	krisa da rat
กรอง กาญจน์	krong kan	กฤษ ดี	krisa dee
กรอง กาญจน์	krong karn	กฤษ ตยา	kritsa taya
กรอง กาญจน์	krong kan	กฤษ ติ กรณ์	krit ti korn
กรอง กาญจน์	krong karn	กฤษ ทศักดิ์	krid tasak
กรอง กาญจน์	krong kan	กฤษ ธิ โชค	kit ti chok
กรอง กาญจน์	krong karn	กฤษ ธิ พร	krit ti porn
กรอง กานต์	krong kan	กฤษ นะ	krissa na
กรอง แก้ว	krong kaew	กฤษ นันท์	krisa nun
กรอง แก้ว	krong kaew	กฤษ นันท์	krisa nan
กรอง แก้ว	krong kaew	กฤษ นันทร์	kridsa nun

กรอง แก้ว	krong kaew	กฤษ พงศ์	kritsa pong
กรอง แก้ว	krong kaew	กฤษ พร	krisa porn
กรอง จิตต์	krong jit	กฤษ พร	kris porn
กรอง ทอง	krong tong	กฤษ ยา พร	khritsa ya porn
กรอง ทอง	krong thong	กฤษ รัตน์	grissa rat
กรอง ทอง	krong thong	กฤษ วัฒน์	kritsa wat
กรอง ทอง	krong tong	กฤษ์ สุ ภา	krit su pa
กรอง ทอง	krong thong	กฤ ษา	kri sa
กรอง พร	krong porn	กฤ สุ ภางค์	kri su pang
กรอง พร	krong porn	กล กิจ	kolla kij
กรอง รวี	grong rawee	กล จรัส	klon jaras
กรอง วลัย	krong walai	กล ชัย	kola chai
กร อนงค์	korn anong	กล ชัย	kolla chai
กร อร	ko raon	กล ชัย	kon chai
กร อร	ko raon	กลย์ ชัย	kon chai
กร อุ มา	korn u ma	กลย์ ธัช	kon that
กร เอก	Korn -ake	กลย์ มนัส	kol manas
กระ ทรวง	kra suang	กล ยุทธ์	konla yut
กระ เทม	kra sem	กล ยุทธ	konla yut
กระ เทียร	kra sean	กล ยุทธ	kola yoot
กระ สุน	kra soon	กล วัชร	kolla wach
กระ แสร์	kra sae	กล วิชญ์	kolla wit
กรัณ ญา	karun ya	กล ศาสตร์	konla sart
กรัณฑ์ รัตน์	karun rat	กล สกรรจ์	kon sagun
กรัณฑ์ รัตน์	karun lat	กลอย กมล	kloy kamon
กรัณย์ ธร	karan torn	กลอย กมล	kloy kamol
กรัณ ยา	karan ya	กลอย กาญจน์	kloy kan
กรัส วัน	kras wan	กนก นุช	kanok nuch
กนก นารถ	kanok nart	กลัน ทิ กา	kalan thi ka
กราน เลิศ	gran lert	กลับ เมือง	glub mung

กรา วิ ฑัต	gra vi tat	กลาง กมล	klang kamon
กริช ชัย	krit chai	กล้า ศักดิ์	kla sak
กริช ชัย	kris chai	กลิ กา	kali ka
กริช เพชร	krit pet	กลิ่น เก สร	klin ka sorn
กริ ชา	kari cha	กลิ่น แก้ว	klin kaew
กริณ	garin	กลิ่น ประ ทุม	klin pra toom
กริ ตา	kari ta	กลิ่น ผกา	klin phaka
กรินทร์ วิชช์	karine vidch	กลิ่น สุ คนธ์	klin su kon
กรีฑ ชนม์	kreeta chon	กลิ่น สุ คนธ์	klin su kon
กรี ฑา	kre tha	กลิ่น สุ คนธ์	klin su kon
กรี ฑา	kre tha	กวิ เชษฐ์	kavi chet
กรี ฑา พล	kree tha pol	กวิญ ญา	kawin ya
กรี ฑา พล	kree tha phol	กวิ ฏา	kawi ta
กรี ธา พร	kree ta porn	กวิ ตา	kavi ta
กรุง ธน	krung thon	กวิ ตา	kavi ta
กรุง บดินทร์	krung bordin	กวิ ตา	kavi ta
กรุง พล	krung phon	กวิ ตา	kwi ta
กรุง ศรี	krung sri	กวิ ตา	kawi ta
กรุณ รัตน์	karun rat	กวิ ตา	kavi ta
กรุ ณา	karu na	กวิน เจตน์	kavin jet
กรุ ณา	karu na	กวิน ดา	kawin da
กรุ ณา	karu na	กวิน ทิพย์	kavin thip
กรุ ณา	karu na	กวิ รัช	kavi rach
กรุ ณา	karu na	กวิ วัณย์	kawi wan
กรุ ณา	karu na	กวิศ รา	kawisa ra
กรุ ณา	karu na	กวิ ศา	kawi sa
กรุ ณา	karu na	กวี นา	gavee na
กรุ ณา	karu na	กวี นา	kave na
กรุ ณา	karu na	กวี นา	kawee na
กรุ ณา	karu na	กวี นุช	kawee nuch

กรุ ณา	karu na	กวี พงษ์	kawee pong
กรุ ณา	karu na	กวี พงษ์	kawee pong
กรุ ณา	karu na	กวี พจน์	kawee poj
กรุ ณี	karu nee	กวี พจน์	kawee poj

ภาคผนวก ค

ผลงานตีพิมพ์

Romanization of Thai Proper Names Based on Popularity of Usages

Akegapon Tangverapong, Atiwong Suchato, and Proadpran Punyabukkana

Spoken Language Systems Research Group,
Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University, Bangkok, 10330, Thailand
Akegapon.T@student.chula.ac.th, atiwong@cp.eng.chula.ac.th,
Proadpran.P@Chula.ac.th

Abstract. The lack of standards for Romanization of Thai proper names makes searching activity a challenging task. This is particularly important when searching for people-related documents based on orthographic representation of their names using either solely Thai or English alphabets. Romanization based directly on the names' pronunciations often fails to deliver exact English spellings due to the non-1-to-1 mapping from Thai to English spelling and personal preferences. This paper proposes a Romanization approach where popularity of usages is taken into consideration. Thai names are parsed into sequences of grams, units of syllable-sized or larger governed by pronunciation and spelling constraints in both Thai and English writing systems. A Gram lexicon is constructed from a corpus of more than 130,000 names. Statistical models are trained accordingly based on the Gram lexicon. The proposed method significantly outperformed the current Romanization approach. Approximately 46% to 75% of the correct English spellings are covered when the number of proposed hypotheses increases from 1 to 15.

Keywords: Thai Romanization, Statistical Language Processing, Machine Translation.

1 Introduction

Thai *Romanization* refers to the method of writing Thai word with English alphabet, which somehow is not standardized. The problem is more prominent for Thai proper names. Personal preferences affect how ones spell names using English alphabets significantly. In some cases, name pronunciations are strictly preserved (transcription) while in many cases spellings reflecting the roots of the names are preferred over how they would sound (transliteration). In this work, we propose a Thai Romanization approach where popularity of Romanization patterns is taken into consideration. A Romanization approach that is capable of producing a list of probable hypotheses of Romanized string given a Thai proper name would be beneficial in searching activities relating to retrieving people-related documents published in either Thai or English when names are used as the search keywords.

T. Theeramunkong et al. (Eds.): PAKDD 2009, LNAI 5476, pp. 580–587, 2009.

2 Literature Review

Romanization is typically approached in two steps: text segmentation and writing system translation. The first one segments the original text string into sequences of method-specific units such as words, syllables, or some other orthographic-based units. The latter proposes the spelling of the original string in the destination writing system.

For Thai, there have been some researches related to Thai text segmentation. Poovarawan [1], Sornlertlamvanich [2] proposed a dictionary-based. Unregistered words will not be recognized. Theeramunkong [3] proposed a method based on a decision tree model without using dictionaries. This method can solve the unregistered word problem. Still, it cannot handle cases with ambiguituies. Aroonmanakun [4] proposed a segmentation method based on syllable trigram models. Romanization based on this segmentation method yields reasonable result. However, the method is not designed to handle the transliteration type of Romanization where resulting English strings aim at preserving some original linguistic information rather than retaining the closest pronunciations. An example of such names includes "โชติ", originally pronounced as a single syllable, while one of the most popular transliteration is a double-syllable Romanized string "choti".

Apart from transliteration, Romanization could be done via transcription, where Thai graphemes are converted to Thai phoneme sequences before they are then mapped to their closest matched English phoneme sequences. Finally, the best sequence of English graphemes is then hypothesized based on the phonemes. Charoenporn et al. [5] and Aroonmanakun et al [6] both deployed corpus-based statistical segmentation methods to segment Thai words into syllables. However, the former chose to perform the writing system translation step by relying on a set of hand-written rules based on the Romanization guide for Thai script defined by the Royal Institute in 1984 together with some deterministic mapping tables.

Obviously, it is difficult and time consuming for such heuristic approaches to handle the unsystematic character of Thai proper name Romanization. Therefore, we propose a data-driven approach that automatically learns the character mapping between Thai and English via a set of newly proposed units generically called "Gram". These units are distinct from typical syllable-based units in the aspect that each Gram contains attributes related to the writing systems (as well as the indirectly embedded pronunciations) of both languages at the same time.

3 Background on Thai Writing System

44 alphabets are used to represent 21 Thai consonant sounds. All of them can be used for consonantal phonemes in the syllable-initial position. Three of them can be combined with other consonants to form true consonant clusters. Although, theoretically, all 44 consonantal alphabets can be used to represent 8 syllable-final consonantal phonemes, some are more popular than the others. In many occasions, a string of multiple consonants could represent only one phoneme, and many occurrences are governed by exceptions rather than precise rules. 34 symbols, some of which are constructed from multiple alphabets, together with three of the alphabets used for the

consonants, are used to represent Thai vowel phonemes including monophthongs as well as diphthongs. 5 Tone symbols are superimposed to Thai syllabic representation to govern the tonal aspect of each syllable, which is also affects by the base consonantal alphabets representing the phoneme in the syllable-initial position.

4 Proposed Methods

4.1 Name Decomposition and Gram Accumulation

In this work, the spelling of each name is looked as if it is the concatenation of a sequence of basic units. The attributes of each one of these units are their alphabet sequences in both the original and the destination systems, which in this case are Thai and English respectively. These arrangements of these units will constrain the spelling of the names in both languages. Although resulting units with meaning are preferable, it is not a requirement for units to be meaningful. Therefore, these units are referred to by using a generic name called *Gram*s, in contrary to the name *Morph* adopted in many linguistic literatures. The string **A**:**B** will be used in this paper to refer to a Gram whose Thai spelling is **A** and its English spelling is **B**. Figure 1 shows some examples of the name decomposition.

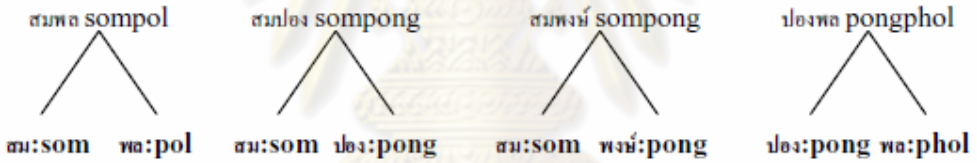


Fig. 1. Some examples of the name decomposition

Items in the Gram lexicon were added while names in the training corpus were analyzed. Each Gram must at least consist of alphabets for the corresponding syllable-initial phoneme and alphabets for the vowel phoneme. Although single-syllable Grams are preferable, multi-syllable Grams are also allowed to ensure that the decomposition of names do not produce any Grams that their Thai spellings only contain a single alphabet apart from two exceptions to be mentioned later. Such cases would have made the decomposition generate too many possible hypotheses. Furthermore, if the vowel alphabet of a syllable is a type of vowel that requires mandatory coda consonants, the alphabets for the consonant and the vowel cannot be separated. This is, again, to prevent the over-generation problem.

Building Gram lexicon is a tedious task if the decomposition is performed manually on a large name corpus. Here, we semi-automatically process names in our training/development set (Details about the name corpus used in this work will be described later in this paper.) in several batches of a couple thousands names. Grams obtained from one batch are added to the lexicon which will be used for the automatic decomposition of the names in the next batch. Manual adjustment is performed on the result of the automatic decomposition of each batch before new Grams are added to the lexicon. The process can be illustrated in Figure 2.

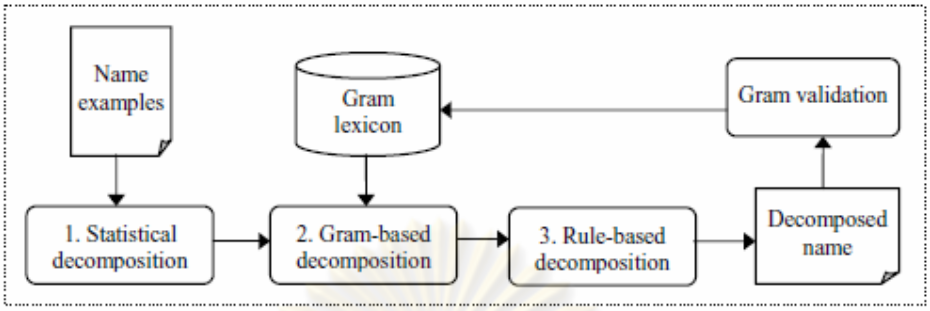


Fig. 2. Gram lexicon construction process

At the beginning of each batch, the Thai spelling of each name is decomposed into a syllable sequences using the method proposed by Aroonmanakun et al. [4]. Then, given the accumulated Grams from previous batches, a right-to-left searching is performed in the English alphabet string associated with each name in order to match each Thai syllable with English alphabets and at the same time, each resulting pair must replicates a Gram in the accumulated Gram lexicon. This search attempt can lead to three possible results: 1) A complete matching, 2) A single failure, and 3) A multiple failure.

A complete matching is when the name can be decomposed entirely into a sequence of Grams in the accumulated Gram lexicon. No manual adjustments are needed and no new Grams are added into the Gram lexicon.

From the result of the right-to-left searching, if the name can be partially decomposed into a sequence of Grams in the lexicon except for the first Thai syllable and some unmatched English alphabets at the beginning of the name. An expert judgment is required whether to pair up the Thai syllable with the remaining English alphabets and add a new Gram into the Gram lexicon. Otherwise, a manual adjustment has to be made on the entire name.

The last case is when more than one syllables cannot be represented using existing Grams. In this case, an alphabet-mapping decomposition is tried. This method of decomposition first tokenized the associated sequence of English alphabets so that each resulting token is either a group of alphabets representing the English vowel, a group of alphabets representing syllable-initial consonants, or a group of alphabets representing consonants in the coda positions. A simple algorithm based on a Thai-English alphabet-wise mapping is then used to combine some English tokens together so that it results in a one-to-one mapping between the Thai syllables and the English tokens. An expert judgment is then needed to verify the validity of the resulting Grams. Manual adjustments are sometimes needed before the new Grams are added to the Gram lexicon.

Although manual adjustments are still required in many cases, such manual burdens are less severe when the Gram lexicon grows larger. Note that in the first batch when the Gram lexicon is empty, the decomposition of every name will result in the multiple failure case and will be treated accordingly.

4.2 Gram-Based Romanization

With the Gram lexicon, A Thai name to be romanized can be decomposed into multiple sequences of Grams constrained by its corresponding Thai spelling. Statistical language and phase transition models are used to give a popularity score to each possible Gram sequence. The system picks the sequence with the highest score to be the hypothesis of choice. Then, Romanization can be completed simply by concatenating the English alphabets of the Grams in the corresponding sequence together.

Given a Thai name $\boldsymbol{R} = r_0 r_1 \ldots r_N$ where each r_i is a Thai alphabet and N is the number of alphabets in the Thai spelling of that name, the system first converts $\boldsymbol{R}$ to a sequence of K Thai alphabet strings, $\boldsymbol{T} = t_0 t_1 \ldots t_K$ where each t_i is the Thai spelling (string of Thai alphabets) of a Gram appearing in our Gram lexicon using a right-to-left longest matching approach.

Then, for the sequence $\boldsymbol{T}$ corresponding with the name $\boldsymbol{R}$, an associated popularity score is computed for each possible Romanization hypothesis, $\boldsymbol{E} = e_0 e_1 \ldots e_K$ where each e_i is the English spelling (string of English alphabets) of a Gram. Note that the length of a possible $\boldsymbol{E}$ must also be K since the mapping between each term in both sequences must be one-to-one. This also means that each of $t_i{:}e_i$ for $i = 0, 1, \ldots, K$ must be a Gram in the lexicon. Here, we deploy the conditional probability of the sequence $\boldsymbol{E}$ given $\boldsymbol{T}$. The MAP criterion is deployed such that we will choose the best hypothesis $\boldsymbol{E}^*$ for the given $\boldsymbol{T}$ such that:

$$\boldsymbol{E}^* = \arg\max_{\boldsymbol{E}} p(\boldsymbol{E} \mid \boldsymbol{T}) = \arg\max_{\boldsymbol{E}} p(\boldsymbol{T} \mid \boldsymbol{E})\, p(\boldsymbol{E})$$

We will refer to the term $p(\boldsymbol{E})$ as the "N-Gram score" which can be computed using the typical N-gram models of English alphabet strings. In this work, bigram models are created from the name decomposition examples in the training/development set. However, only the English alphabets of each Gram are considered. In other words, when counting the number of Gram pairs to obtain the bigram models, two different Grams considered the same unit if their English spellings are the same.

Similar to the problem of the machine translation problems in the sentence level, we view the term $p(\boldsymbol{T} \mid \boldsymbol{E})$ as the translation score. Assuming independency among the terms in $\boldsymbol{T}$ and that t_i only depends on e_i we have:

$$p(\boldsymbol{T} \mid \boldsymbol{E}) = p(t_0 t_1 \ldots t_K \mid e_0 e_1 \ldots e_K) = \prod_{i=0}^{K} p(t_i \mid e_i)\,,$$

and

$$p(t_i \mid e_i) = \frac{N(t_i : e_i)}{\sum_{\text{all } \tau} N(\tau : e_i)},$$

where $N(t_i{:}e_i)$ is the number of the Gram $t_i{:}e_i$ in the training/development set and $\sum_{\text{all } \tau} N(\tau : e_i)$ is the total number of any Grams τ that their associated English spelling is e_i.

5 Experiment Details

5.1 Name Corpus

A corpus of Thai first names and surnames were used for the training of all statistical models as well as for evaluating the proposed method. The corpus was constructed from a database of the names of students registered at Chulalongkorn University, mostly, in the last 10 years. Foreign names were excluded from the corpus. The corpus contains Thai spellings of 178,612 names together with their English counterparts. 20% of the names were randomly picked as the test data set while the rest 80% were put in the training/development set, in which names were used for the building of our Gram lexicon, as well as to train statistical models for the N-gram scores as well as the translation scores mentioned earlier. The names in this set were also observed in the process of introducing our Grams.

6 Results and Discussion

6.1 Construction of the Gram Lexicon

Figure 3 compares the coverage percentages and the number of accumulated Grams (as the percentage of the total Grams in the final Gram lexicon) with the number of names analyzed.

As we can observe from the table, after analyzing the 22 batches of data in the training/development set, including manual adjustments that handle decomposition failures, 24,385 Grams cover 139,500 names in the set. Note that each of these Grams is constrained by both the Thai and English spelling. Our analysis also showed that, among these 24,385 Grams, there were around eight thousands unique tokens if only Thai alphabet sequences were considered, while the number is around twelve thousand if only English alphabet sequences were considered.

From figure 3, we can see that the number of Grams grows rather linearly with the increasing of the number of names after some Grams (which, in this case, are around 35% of the total Grams) are already in the lexicon. The coverage percentage also grows linearly with the increasing of the number of names already analyzed.

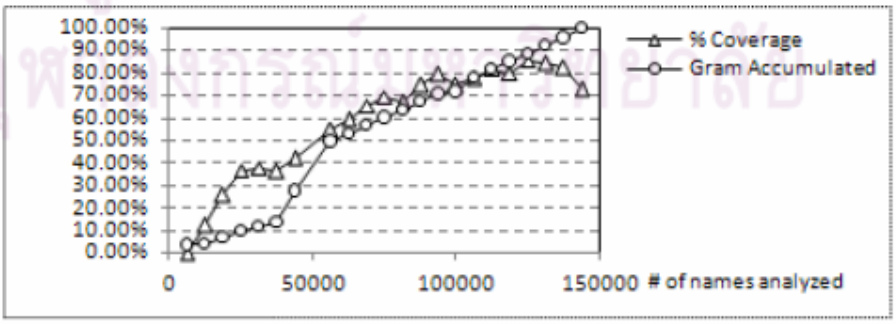


Fig. 3. The tendency of %Coverage and the number of accumulated Grams when more names are analyzed in batches

6.2 Romanization Results

Table 1 compares the Romanization accuracies of the baseline and the proposed methods evaluated on the 35,722 names in the test data set.

Table 1. Romanization accuracies evaluated on the test data set

	Baseline	Proposed Method	
		1st Variation	2nd Variation
Romanization Accuracy	18.20%	46.13%	43.66%

The Romanization accuracy percentage of the baseline method is around 18%. The fact that more than 80% of the names are romanized differently from what are listed in the test data set reflects the degree of complexity of Thai name Romanization. However, the highly-varied nature of the Thai name Romanization also exposes the weakness of the baseline method that it only focuses on the closest matching of the sounds between the two sound systems and it lacks the flexibility to handle real-life preferences. The proposed method yield the best performance of almost 50% evaluated on the same data set.

Figure 4 shows the recall rate percentages when *N*-best hypotheses are used. We can see from the figure that the 1st variation of our proposed method yields higher recall rates for every values of *N* tested. Both variations show a similar tendency where the recall rate grows from the values shown in table 4 (≈44% – 46%) to around 75% to 80% when 15-best hypotheses are used. Although the recall rate keeps increasing as the number of hypothesis increases, we can observe that the curves tend to saturate. Consequently, we can expect that using more hypotheses is not likely to push the recall rate any higher. It is worth mentioning that even though using *N*-best hypotheses does not single out the correct Romanization of a name, we can reasonably expect it to recall the correct one almost 80% of the time when applied to a search application in which not too many Romanization hypotheses from the proposed method are used as the search queries.

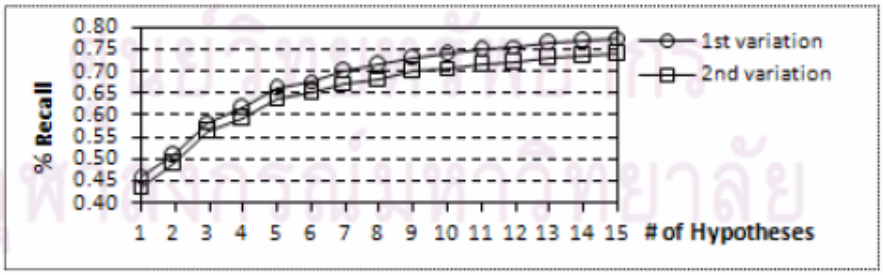


Fig. 4. Recall rate percentages of the proposed method when N-best hypotheses are used

7 Conclusion

This paper presents a Romanization method whose flexibility is more suitable for the Romanization of Thai names than a conventional Romanization method. Although it

might not be surprising that taking into consideration real-life preferences in the forms of a large number of Romanized name examples yield a better Romanization performance, practical statistical models have to be formulated in order for an automatic method to hypothesize the most likely results instead of having to rely on the burden of creating sets of heuristics from such a large name database. In this work, the proposed Gram-based models yield a satisfactory finding in terms of the Romanization results as well as the framework adopted in the construction of the Gram lexicon from examples of more than one-hundred thousand names.

References

[1] Poowarawan, Y.: Dictionary-based Thai Syllable Separation. In: Proceedings of the Ninth Electronics Engineering Conference (1986)
[2] Sornlertlamvanich, V.: Word Segmentation for Thai in a Machine Translation system (in Thai), Papers on Natural Language processing, NECTEC, Thailand (1995)
[3] Thanaruk, T., Thanasan, T., Duangrumol, P., Arunthep, S.: Non-Dictionary-Based Word Segmentation Using Local Context Statistics. In: Proceedings of the 5th Symposium on Natural Language Processing and Oriental COCOSDA Workshop, Hua Hin, Thailand, pp. 81–88 (May 2002)
[4] Aroonmanakun, W.: Collocation and Thai Word Segmentation. In: Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop, pp. 68–75. Sirindhorn International Institute of Technology, Pathumthani (2002)
[5] Thatsanee Charoenporn, Ananlada Chotimongkol, and Virach Sornlertlamvanich, Automatic Romanization For Thai, Bangkok, Thailand (1999)
[6] Aroonmanakun, W., Rivepiboon, W.: A Unified Model of Thai Word Segmentation and Romanization. In: Proceedings of The 18th PACLIC, Tokyo, Japan (2004)
[7] Karoonboonyanan, T.: Standardization and Implementations of Thai Language. In: The Seminar on Enhancement of the International Standardization Activities in Asia Pacific Region (AHTS-1) held on at CICC, Japan (March 1999)

ประวัติผู้เขียนวิทยานิพนธ์

นายเอกพล ตั้งวีระพงษ์ เกิดเมื่อวันที่ 22 เมษายน พ.ศ. 2524 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ จากภาควิชาวิทยาศาสตร์คอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2546 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2549