การใช้เทคนิคเอ็นแกรมช่วยในการเพิ่มประสิทธิภาพการตัดสินใจแปลอักษรเบรลล์ ในกรณีที่มีคำควบกล้ำ สระประสมและอักษรเบรลล์สองเซลล์

Enhancing Thai Braille Translation with n-gram for Decision Making in the Cases of Compound Consonants, Vowels and Characters

ทศวัฒน์ ชุณหวิทยะธีระ

Totsawat Chunhawitayatera

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Computer Engineering
Prince of Songkla University
2555

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์ การใช้เทคนิคเอ็นแกรมช่วยในการเพิ่มประสิทธิภาพการตัดสินใจแปลอักษรเบรลล์ในกรณีที่มีคำควบกล้ำ สระประสมและอักษรเบรลล์สองเซลล์

ผู้เขียน นายทศวัฒน์ ชุณหวิทยะธีระ

สาขาวิชา วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

……..

(ผู้ช่วยศาสตราจารย์ ดร.พิชญา ตัณฑัยย์)

คณะกรรมการสอบ

……..ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.อำนาจ เปาะทอง)

……..กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.พิชญา ตัณฑัยย์)

……..กรรมการ

(ดร.เดือนเพ็ญ กชกรจารุพงศ์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

……..

(ศาสตราจารย์ ดร.อมรรัตน์ พงศ์ดารา)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ การใช้เทคนิคเอ็นแกรมช่วยในการเพิ่มประสิทธิภาพการตัดสินใจแปลอักษรเบรลล์ในกรณีที่มีคำควบกล้ำ สระประสมและอักษรเบรลล์สองเซลล์
ผู้เขียน นายทศวัฒน์ ชุณหวิทยะธีระ
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2554

บทคัดย่อ

ในปัจจุบันมีการพัฒนาโปรแกรมเพื่อช่วยแปลงข้อความอักษรเบรลล์ภาษาไทยให้เป็นข้อความภาษาไทยเพื่ออำนวยความสะดวกในการสื่อสารระหว่างผู้พิการทางสายตาและคนสายตาปกติ ทั้งนี้เพราะลักษณะการเขียนอักษรเบรลล์ภาษาไทยระดับ 1 มีความแตกต่างจากลักษณะการเขียนภาษาไทยอยู่พอสมควร จึงไม่สามารถใช้วิธีการแปลงโดยอาศัยการเปรียบเทียบแบบตัวอักษรต่อตัวอักษรหรือคำต่อคำได้ ซึ่งโปรแกรมที่ได้มีการพัฒนามาก่อนหน้านี้มักใช้กฎไวยากรณ์ทางภาษาเข้ามากำกับวิธีการแปลง แต่ยังคงประสบปัญหาการแปลงคำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์ เนื่องจากลักษณะการเขียนอักษรเบรลล์ภาษาไทยระดับ 1 มีลักษณะการเขียนอักขระเรียงติดต่อกันไปจนกระทั่งจบประโยคและใช้การเว้นวรรคเพื่อแบ่งแต่ละประโยคออกจากกัน คำประเภทดังกล่าวมีรูปพ้องกัน จึงสามารถอ่านได้หลายแบบและแต่ละรูปแบบก็มีความหมายที่แตกต่างกันไป จึงทำให้เกิดความกำกวมขึ้นและไม่สามารถเขียนโปรแกรมให้ครอบคลุมทุก ๆ กรณีที่เกิดขึ้นได้

งานวิจัยนี้จึงได้นำเสนอวิธีแก้ไขปัญหาการแปลงข้อความอักษรเบรลล์ภาษาไทยให้เป็นข้อความภาษาไทยเพื่อใช้กับประโยคที่มีคำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์ โดยใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำก่อนที่จะแปลงให้เป็นภาษาไทยเพื่อลดความกำกวมของภาษาและนำเสนอจำนวนแกรมที่เหมาะสมสำหรับใช้ตัดคำอักษรเบรลล์ภาษาไทย ซึ่งผลการทดลองสรุปว่าสามารถแปลงข้อความได้ถูกต้องมากขึ้นและใช้ปริมาณหน่วยความจำน้อยลง แต่ใช้เวลาแปลงมากกว่าเดิมโดยอยู่ในเกณฑ์ที่ยอมรับได้ และจำนวนแกรมที่เหมาะสมคือ 4-แกรม

คำสำคัญ: อักษรเบรลล์ภาษาไทย, เอ็นแกรมโมเดล, โปรแกรมแปลงข้อความอักษรเบรลล์ภาษาไทยเป็นข้อความภาษาไทย

Thesis Title	Enhancing Thai Braille translation with n-gram for decision making in the cases of compound consonants, vowels and characters
Author	Mr. Totsawat Chunhawitayatera
Major Program	Computer Engineering
Academic Year	2011

ABSTRACT

There have been several developments for Thai Braille to Thai translation for assisting the communication between the visually impaired and the normal sighted. The grammar rules of Thai Braille and Thai are different. So, it is not applicable to use character-to-character or word-to-word translation techniques. Generally, Thai Braille to Thai translation programs apply grammar rules to control the translation process but there are problems about translating compound consonants and vowels, two-cell Braille characters and transliterated words because Thai Braille Grade 1 write characters continuously until the end of the sentence and uses spaces for separating sentences. There are such words that have the same character sequence with other words but have several meanings. This causes ambiguity. Therefore, it is difficult to write a program to cover all conditions.

This Thesis proposes a method to solve Thai Braille to Thai translation problems in the cases that the sentences contain compound consonants, vowels, two-cell Braille characters and transliterated words by using the N-gram model technique to wrap words before being translated into Thai in order to reduce the ambiguity of the language and propose the optimal N-gram number for wrapping Thai Braille words. The new method can improve the correctness of the translation and reduce the memory consumption better than the old method but the new method requires more translation time than the old method. The appropriate N-gram number is 4-gram for the Braille Thai text word wrap.

Keywords: Thai Braille, N-gram model, Thai Braille to Thai translation program

กิตติกรรมประกาศ

ขอแสดงคำขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. พิชญา ตัณฑัยย์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ที่ได้กรุณาอุทิศเวลาให้คำปรึกษา ให้การสนับสนุนการทำวิจัย แนะนำความรู้ในด้านการทำวิจัย เอกสาร ข้อมูลต่างๆเป็นอย่างดี รวมทั้งแนวความคิดและกำลังใจในการแก้ปัญหาตลอดจนตรวจทานแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. อำนาจ เปาะทอง ประธานกรรมการสอบวิทยานิพนธ์ที่ได้กรุณาให้คำปรึกษาคำแนะนำ และให้การช่วยเหลือแก้ไขปัญหาในงานวิจัย และตรวจทานวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณ ดร. เดือนเพ็ญ กชกรจารุพงศ์ กรรมการสอบวิทยานิพนธ์ที่ได้กรุณาให้คำปรึกษาคำแนะนำ และให้การช่วยเหลือแก้ไขปัญหาในงานวิจัย

ขอขอบคุณ พี่ๆ เพื่อน ๆ และน้องๆ นักศึกษาปริญญาโท ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ โดยเฉพาะกลุ่มงานวิจัยห้อง WIG ประกอบด้วย นางสาวสุภาวดี มากอ้น, นายพิทักษ์ เศวตสุนทร, นายณัฐวุฒิ วิจิตร์ และนางสาวพรวิกา สร้อยทอง และทุกท่านที่ได้ให้คำแนะนำ คำปรึกษา และกำลังใจที่ดีมาโดยตลอด

ขอขอบพระคุณ คณาจารย์และเจ้าหน้าที่ในภาควิชาวิศวกรรมคอมพิวเตอร์ทุกๆ ท่าน ที่ให้ความช่วยเหลือในด้านต่างๆมาโดยตลอด จนกระทั่งงานสำเร็จลุล่วง

สุดท้ายนี้ ข้าพเจ้าขอน้อมรำลึกถึงพระคุณของนายอภิชัย ชุณหวิทยะธีระ (บิดา) และนางวารุณี ชุณหวิทยะธีระ (มารดา) บุคคลในครอบครัว ที่ส่งเสริมสนับสนุน ให้คำแนะนำ ให้คำปรึกษา ให้กำลังใจ และทุนทรัพย์แก่ข้าพเจ้าตลอดมาจนกระทั่งทำให้ข้าพเจ้าประสบความสำเร็จ

ทศวัฒน์ ชุณหวิทยะธีระ

สารบัญ

หน้า

บทคัดย่อ..... (3)

ABSTRACT..... (4)

กิตติกรรมประกาศ..... (5)

สารบัญ..... (6)

สารบัญตาราง..... (10)

สารบัญภาพประกอบ..... (12)

สัญลักษณ์คำย่อและตัวย่อ..... (14)

ศัพทานุกรม..... (15)

บทที่ 1 บทนำ 1

1.1 ความสำคัญ และที่มาของวิทยานิพนธ์..... 1

1.2 วัตถุประสงค์..... 4

1.3 ขอบเขตของการวิจัย..... 4

1.4 ขั้นตอนการวิจัยและวิธีการวิจัย..... 5

1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 5

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง..... 7

2.1 อักษรเบรลล์..... 7

2.1.1 อักษรเบรลล์ภาษาไทย..... 8

2.1.2 หลักการอ่านเขียนอักษรเบรลล์ภาษาไทยระดับ 1..... 10

2.2 โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file……... 12

2.3 เทคนิคการตัดคำ..... 13

2.3.1 เทคนิคการตัดคำโดยใช้กฎพื้นฐาน..... 14

2.3.2 เทคนิคการตัดคำโดยใช้พจนานุกรม..... 15

2.3.3 เทคนิคการตัดคำโดยใช้คลังข้อมูล..... 17

2.4 คลังข้อมูลภาษา..... 18

2.5 เอ็นแกรมโมเดล..... 21

2.5.1 หลักการทำงานของเอ็นแกรมโมเดล..... 21

สารบัญ (ต่อ)

หน้า

2.6 Smoothing Techniques..... 22

2.7 CMU-SLM toolkit..... 24

2.8 โปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์..... 25

บทที่ 3 การออกแบบและการพัฒนาระบบ..... 39

3.1 การวิเคราะห์ปัญหาการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยในปัจจุบัน..... 39

3.2 แนวคิดในการแก้ไขปัญหาในงานวิจัย..... 30

3.3 การคัดเลือกประโยคจากคลังข้อมูล BEST และการสร้างไฟล์โมเดล
อักษรเบรลล์ภาษาไทย..... 31

3.3.1 การคัดเลือกประโยคที่ต้องการจากคลังข้อมูล BEST..... 32

3.3.2 การสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทย..... 36

3.4 ภาพรวมของระบบที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้..... 39

3.5 อินพุตและเอาท์พุทของระบบ..... 41

3.6 โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรม
เข้ามาช่วยในการตัดคำที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้..... 41

3.6.1 ขั้นตอน pre_ngram_generate..... 41

3.6.2 ขั้นตอน generate_ngram_pattern..... 44

3.6.3 ขั้นตอน select_output_pattern..... 48

3.6.4 ขั้นตอน translate_to_thai..... 56

3.7 การประยุกต์เอ็นแกรมโมเดล..... 61

3.8 Smoothing Techniques ที่เลือกใช้..... 66

3.9 รายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทย..... 69

บทที่ 4 ผลการทดลอง..... 73

4.1 ด้านความถูกต้องของการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทย..... 73

สารบัญ (ต่อ)

หน้า

4.1.1 เครื่องมือที่ใช้ในการทดสอบ..... 73

4.1.2 ภาพรวมการทดสอบด้านความถูกต้องการแปลงอักษรเบรลล์ภาษาไทย ไปเป็นภาษาไทย..... 74

4.1.3 ชุดข้อมูลทดสอบเปรียบเทียบด้านความถูกต้อง..... 75

4.1.4 ผลการทดลองเปรียบเทียบด้านความถูกต้อง..... 77

4.2 ด้านปริมาณการใช้หน่วยความจำในการแปลง..... 79

4.2.1 เครื่องมือที่ใช้ในการทดสอบ..... 79

4.2.2 ภาพรวมการทดสอบปริมาณการใช้หน่วยความจำในการแปลง..... 79

4.2.3 ชุดข้อมูลทดสอบด้านปริมาณการใช้หน่วยความจำ..... 81

4.2.4 ผลการทดลองเปรียบเทียบปริมาณการใช้หน่วยความจำ..... 82

4.3 ระยะเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทย..... 83

4.3.1 เครื่องมือที่ใช้ในการทดสอบ..... 83

4.3.2 ภาพรวมการทดสอบระยะเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทย ไปเป็นภาษาไทย..... 84

4.3.3 ชุดข้อมูลทดสอบเปรียบเทียบด้านเวลาที่ใช้ในการแปลง..... 86

4.3.4 ผลการทดลองเปรียบเทียบด้านเวลาที่ใช้ในการแปลง..... 89

4.4 สรุปผลการทดสอบ..... 92

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ..... 93

5.1 สรุปผลการวิจัย..... 93

5.2 สรุปปัญหา อุปสรรค และสาเหตุข้อผิดพลาดในงานวิจัยนี้..... 94

5.3 ข้อเสนอแนะ..... 97

เอกสารอ้างอิง..... 99

ภาคผนวก..... 103

สารบัญ (ต่อ)

หน้า

ภาคผนวก ก. ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์..... 104
ภาคผนวก ข. ตารางผลการทดลอง..... 114
ภาคผนวก ค. รายการคำที่มีความผิดพลาดประเภทต่างๆ..... 121
ภาคผนวก ง. ตารางอักษรเบรลล์ประเภทต่างๆ..... 156
ประวัติผู้เขียน..... 158

สารบัญตาราง

หน้า

ตารางที่ 2.1 พยัญชนะภาษาไทยที่มีรูปอักษรเบรลล์แบบ 2 เซลล์..... 8
ตารางที่ 2.2 รูปอักษรเบรลล์ของสระประสม..... 9
ตารางที่ 2.3 สระและเครื่องหมายที่มีรูปอักษรเบรลล์แบบ 2 เซลล์..... 10
ตารางที่ 3.1 อักษรเบรลล์สองเซลล์ที่มีรูปพ้องกันกับอักษรเบรลล์เซลล์เดียว
สองตัวที่เขียนติดกัน..... 28
ตารางที่ 3.2 รายละเอียดการคัดเลือกประโยคที่ต้องการจากคลังข้อความ BEST..... 35
ตารางที่ 3.3 รายละเอียดของตัวแปรในขั้นตอน pre_ngrams_generate..... 42
ตารางที่ 3.4 รายละเอียดของตัวแปรที่ใช้ในขั้นตอน gen_ngram_pattern..... 44
ตารางที่ 3.5 รายละเอียดตัวแปรต่างๆ ที่ใช้ในขั้นตอน select_output_pattern..... 48
ตารางที่ 3.6 ตัวแปรต่างๆ ที่ใช้ในขั้นตอน translate_to_thai..... 58
ตารางที่ 3.7 ตัวแปรต่างๆ ที่ใช้ในฟังก์ชัน load_ngram_word..... 61
ตารางที่ 3.8 ตัวแปรต่างๆ ที่ใช้ในขั้นตอน calculate_smoothing..... 67
ตารางที่ 3.9 ตัวแปรชนิดต่างๆ ที่ใช้ในการสร้างรายการคำ..... 71
ตารางที่ 4.1 รายละเอียดชุดข้อมูลทดสอบด้านความถูกต้อง..... 76
ตารางที่ 4.2 ผลการทดลองเปรียบเทียบความถูกต้องของวิธีการเดิมและวิธีการใหม่..... 77
ตารางที่ 4.3 รายละเอียดชุดข้อมูลทดสอบด้านปริมาณการใช้หน่วยความจำ..... 82
ตารางที่ 4.4 ผลการทดลองปริมาณการใช้หน่วยความจำ..... 83
ตารางที่ 4.5 รายละเอียดประเภทคำควบกล้ำ..... 87
ตารางที่ 4.6 รายละเอียดประเภทสระประสม..... 88
ตารางที่ 4.7 รายละเอียดประเภทอักษรเบรลล์สองเซลล์..... 88
ตารางที่ 4.8 รายละเอียดประเภทคำทับศัพท์..... 88
ตารางที่ 4.9 รายละเอียดประเภทข้อความทั่วไป..... 89
ตารางที่ 4.10 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลง..... 90
ตารางที่ 1 ผลการทดลองปริมาณการใช้หน่วยความจำ..... 114
ตารางที่ 2 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์คำควบกล้ำ..... 115
ตารางที่ 3 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์สระประสม..... 116

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 4 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภท
ไฟล์อักษรเบรลล์สองเซลล์..... 117
ตารางที่ 5 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์คำทับศัพท์..... 118
ตารางที่ 6 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประโยคทั่วไปของวิธีการเดิม..... 119
ตารางที่ 7 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประโยคทั่วไปของวิธีการใหม่..... 120
ตารางที่ 8 รายการคำที่โปรแกรม Thai2brl แปลงผิดพลาด..... 121
ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทย
ที่โปรแกรม SWATH ตัดคำผิดพลาด..... 122
ตารางที่ 10 รายการคำของคลังข้อความ BEST เขียนผิดพลาด..... 154
ตารางที่ 11 อักษรเบรลล์ภาษาไทยแบบ 6 จุด..... 156
ตารางที่ 12 อักษรเบรลล์ภาษาไทยแบบ 8 จุด..... 157

สารบัญภาพประกอบ

หน้า

รูปที่ 2.1 การเรียงจุดในอักษรเบรลล์..... 7
รูปที่ 2.2 กราฟเปรียบเทียบจำนวนครั้งที่กดคีย์เมื่อใช้ข้อมูลและจำนวนแกรมต่างๆ..... 20
รูปที่ 3.1 ขนาดของไฟล์ที่มีการใช้คำประเภทต่างๆ
เปรียบเทียบกับขนาดของไฟล์ทั้งหมด..... 30
รูปที่ 3.2 ภาพรวมวิธีการคัดเลือกประโยคที่ต้องการจากคลังข้อความ BEST..... 34
รูปที่ 3.3 ตัวอย่างประโยคที่คัดเลือกจากคลังข้อความ BEST..... 36
รูปที่ 3.4 ประโยคที่แปลงให้อยู่ในรูปของอักษรเบรลล์ภาษาไทย..... 36
รูปที่ 3.5 ขั้นตอนการสร้างโมเดลอักษรเบรลล์ภาษาไทย
ด้วยชุดเครื่องมือ CMU-SLM toolkit..... 37
รูปที่ 3.6 ตัวอย่างไฟล์โมเดลอักษรเบรลล์ภาษาไทย..... 38
รูปที่ 3.7 ภาพรวมโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย
ที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ..... 40
รูปที่ 3.8 ขั้นตอน prepare_ngram_pattern..... 43
รูปที่ 3.9 ขั้นตอน gen_ngram_pattern..... 45
รูปที่ 3.10 การสร้างรูปแบบเอ็นแกรม..... 47
รูปที่ 3.11 ขั้นตอนการทำงานของฟังก์ชัน select_output..... 49
รูปที่ 3.12 ขั้นตอนการทำงานของฟังก์ชัน ngram_to_word..... 51
รูปที่ 3.12a กรณีที่มีการเรียงตัวอักขระเหมือนกันทั้งหมด..... 53
รูปที่ 3.12b กรณีที่มีการเรียงตัวอักขระไม่เหมือนกันทั้งหมด..... 54
รูปที่ 3.13 ขั้นตอนการทำงานของฟังก์ชัน find_ngram..... 55
รูปที่ 3.14 ภาพรวมวิธีการแปลงแบบเทียบคำ..... 57
รูปที่ 3.15 การทำงานของขั้นตอน translate_to_thai..... 59
รูปที่ 3.16 การทำงานของฟังก์ชัน match_brl..... 60
รูปที่ 3.17 ขั้นตอนการทำงานของฟังก์ชัน load_ngram_word..... 62
รูปที่ 3.18 วิธีการจัดเก็บข้อมูลในตัวแปรประเภทต่างๆ..... 65
รูปที่ 3.19 ภาพรวมการทำงานของ Interpolation..... 66

สารบัญภาพประกอบ (ต่อ)

หน้า

รูปที่ 3.20 การทำงานของฟังก์ชัน calculate_smoothing..... 68
รูปที่ 3.21 ขั้นตอนการสร้างรายการคำภาษาไทย..... 70
รูปที่ 4.1 ภาพรวมวิธีการเปรียบเทียบด้านความถูกต้อง..... 74
รูปที่ 4.2 ภาพรวมวิธีการทดลองด้านปริมาณการใช้หน่วยความจำ..... 80
รูปที่ 4.3 ภาพรวมวิธีการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลง..... 85
รูปที่ 5.1 ขั้นตอนการสร้างรายการคู่คำภาษาไทยและอักษรเบรลล์ภาษาไทย..... 95

สัญลักษณ์คำย่อและตัวย่อ

ASCII	American Standard Code for Information Interchange
BCC	Braille Computer Code
BEST	Benchmark for Enhancing the Standard of Thai Language processing
BRL	Braille
CMU-SLM	The Carnegie Mellon University Statistical Language Modeling
Chr	Character
G II	Braille grade 2
HLT	Human Language Technology
NECTEC	National Electronics and Computer Technology Center
SWATH	Smart Word Analysis for Thai

ศัพทานุกรม

คลังข้อมูลฝึก	Training corpus
ช่องว่าง	Space
ฝึกฝนเอ็นแกรมโมเดล	N-gram model Training
รูปแบบเอ็นแกรม	N-gram pattern
สตริง	String
อาร์เรย์	Array
อักษรเบรลล์	Braille character
เอ็นแกรมโมเดล	N-gram model

บทที่ 1

บทนำ

1.1 ความสำคัญ และที่มาของวิทยานิพนธ์

ในปัจจุบันมีการพัฒนาโปรแกรมช่วยแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยแบบ File-to-file เพื่อช่วยอำนวยความสะดวกในการสื่อสารระหว่างผู้พิการทางสายตาและคนสายตาปกติ ทั้งนี้เพราะลักษณะการเขียนอักษรเบรลล์ภาษาไทยมีความแตกต่างจากการเขียนภาษาไทยอยู่พอสมควรกล่าวคือ ใช้สัญลักษณ์, ตัวเลขอารบิคและตัวอักษรภาษาอังกฤษเพื่อแทนพยัญชนะ, สระและวรรณยุกต์ในภาษาไทย (ในที่นี้จะกล่าวถึงเฉพาะการเขียนอักษรเบรลล์ภาษาไทยระดับ 1 ในรูปแบบ Braille Computer Code (BCC) เท่านั้น) เช่น ใช้สัญลักษณ์ “?” เพื่อแทนตัวอักษร “พ”, ใช้ตัวเลขอารบิค “6” เพื่อแทนสระเอา, ใช้ตัวอักษรภาษาอังกฤษ “g” เพื่อแทนตัวอักษรภาษาไทย “ก” เป็นต้น นอกจากนี้ลักษณะการเขียนอักษรเบรลล์ภาษาไทยยังมีการเรียงพยัญชนะ, สระและวรรณยุกต์ที่แตกต่างไปจากลักษณะการเขียนภาษาไทยอีกด้วย ยกตัวอย่างเช่น คำว่า “เพลีย” เมื่อเขียนให้อยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 จะมีการเรียงสระและพยัญชนะเป็น “พ+ล+สระเอีย” และเมื่อเขียนในรูปแบบ BCC จะได้ “?+l+(” เป็นต้น (รายละเอียดการเขียนอักษรเบรลล์จะอธิบายในหัวข้อที่ 2.1.2 การเขียนอักษรเบรลล์ภาษาไทยระดับ 1)

นอกจากนี้การแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยแบบ File-to-file ยังคงประสบปัญหาการหาขอบเขตของคำเช่นเดียวกันกับการแปลงภาษาไทยไปเป็นภาษาอังกฤษ ซึ่งปัญหานี้มีสาเหตุมาจากลักษณะการเขียนภาษาไทยที่เขียนอักขระเรียงติดกันไป โดยไม่มีการเว้นวรรคระหว่างคำ แต่จะเว้นวรรคเมื่อเขียนจบประโยคหรือขึ้นย่อหน้าใหม่และคำบางคำในภาษาไทยมีรูปที่พ้องกัน ทำให้สามารถอ่านได้หลายแบบและมีความหมายที่แตกต่างกันจึงก่อให้เกิดความกำกวมขึ้น และอักษรเบรลล์ภาษาไทยก็มีลักษณะการเขียนเช่นเดียวกับลักษณะการเขียนของภาษาไทยด้วย จึงทำให้ประสบปัญหาการหาขอบเขตของคำเช่นเดียวกันกับภาษาไทย โดยกระบวนการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยแบบ File-to-file ทั่วไปมักประสบปัญหาในการแปลงคำที่เกี่ยวข้องกับคำควบกล้ำ คำที่มีการใช้สระประสม และคำที่มีการใช้อักษรเบรลล์สองเซลล์ (รายละเอียดการเขียนอักษรเบรลล์สองเซลล์จะอธิบายในหัวข้อที่ 2.1.1 อักษรเบรลล์ภาษาไทย) ซึ่งงานวิจัยเรื่อง “การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด” โดยมหาวิทยาลัยสงขลานครินทร์ ได้

นำเสนอโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file [1] แต่ยังคงมีความผิดพลาดในการแปลงที่เกี่ยวกับประเภทของคำที่ได้กล่าวไว้ข้างต้น ตัวอย่างเช่น

การแปลงคำประเภทคำควบกล้ำหรือคำที่มีการใช้สระประสม เช่นคำว่า “|*gl(}4” (เป็นคำว่า “ตาเกลี้ยง” ที่เขียนอยู่ในรูปแบบ BCC ระดับ 1) ซึ่งโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม จะแปลงเป็น “ตากเลี้ยง” ทั้งนี้เพราะคำดังกล่าวเมื่อเขียนในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 จะสามารถอ่านได้สองแบบคือ “ตาเกลี้ยง” และ “ตากเลี้ยง” ซึ่งโปรแกรมส่วนใหญ่มักถูกกำหนดให้แปลงให้ได้คำที่มีตัวอักษรมากที่สุด หรือเลือกคำที่มีพยางค์ยาวที่สุด จึงดึงอักษร “g” (“ก”) ไปรวมกับคำ “|*” (“ตา”) ซึ่งเมื่อดึงไปรวมแล้วทำให้เกิดคำว่า “ตาก” ซึ่งมีความหมายในภาษาไทย จึงทำให้แปลงออกมาเป็นคำว่า “ตากเลี้ยง” ซึ่งแม้จะถูกต้องตามหลักการเขียนไวยากรณ์อักษรเบรลล์และแต่ละคำคือ “ตาก” และ “เลี้ยง” ต่างก็มีความหมายในภาษาไทยด้วยกันทั้งคู่ แต่เมื่อนำมารวมกันแล้วกลับทำให้ความหมายโดยรวมผิดไป

การแปลงคำที่มีการใช้อักษรเบรลล์สองเซลล์ เช่นคำว่า “?rao}u0)r}” (เป็นคำว่า “พระองค์ทรง” ที่เขียนอยู่ในรูปแบบ BCC ระดับ 1) ซึ่งโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม จะแปลงเป็น “พระองธรง” ทั้งนี้เป็นเพราะโปรแกรมจะไม่ทราบว่าเป็นอักษรเบรลล์สองเซลล์ที่มีรูปพ้องกันกับอักษรเบรลล์เซลล์เดียวสองตัวหรือเป็นอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกัน โดยในกรณีที่ยกตัวอย่างนี้ ตัวอักษร “ธ” เป็นอักษรเบรลล์สองเซลล์ที่มีรูปพ้องกันกับอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกัน ซึ่งก็คือ “การันต์+ท” จึงทำให้เกิดการแปลงที่ผิดพลาดดังกล่าว

การแก้ไขปัญหาเหล่านี้ไม่สามารถกระทำได้โดยกำหนดเงื่อนไขเพิ่มเติม หากแต่ต้องพิจารณาความหมายโดยรวมหรือความถี่ในการใช้งาน จากปัญหาดังกล่าวผู้วิจัยจึงมีแนวคิดนำเทคนิคเอ็นแกรมเข้ามาแก้ไขปัญหา แบบจำลองเอ็นแกรม (N-Gram model) [2] เป็นแบบจำลองที่ใช้กระบวนการทางสถิติในการคำนวณหาค่าความถี่หรือค่าความน่าจะเป็นของชุดอักขระ (Character sequence) ที่เกิดขึ้นร่วมกันเป็นคำ หรือค่าความน่าจะเป็นของคำที่เขียนเรียงกัน (Word sequence) ที่เกิดขึ้นร่วมกันเป็นประโยค โดยค่าความถี่หรือความน่าจะเป็นสามารถคำนวณได้จากการนับจำนวนคำที่มีอยู่ในคลังข้อมูลฝึก (Training corpus) [13] ซึ่งแบบจำลองเอ็นแกรมนี้มีงานวิจัยต่างๆ ได้นำมาประยุกต์ เช่น “การระบุคำไทยและคำทับศัพท์ด้วยแบบจำลองเอ็นแกรม” [3] เป็นงานวิจัยที่นำเทคนิคของเอ็นแกรมเข้ามาช่วยระบุชนิดของภาษา, “การตัดคำและกำกับหมวดคำภาษาไทยแบบเบ็ดเสร็จด้วยคอมพิวเตอร์” [4] เป็นงานวิจัยที่นำเทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำภาษาไทยและระบุประเภทของคำ, “การประยุกต์เอ็นแกรมและเว็กเตอร์โมเดลสำหรับระบบย่อข้อความภาษาไทย” [5] เป็นงานวิจัยที่นำเทคนิคเอ็นแกรมเข้ามาช่วยในการย่อความภาษาไทย เป็นต้น ซึ่ง

งานวิจัยเหล่านี้เป็นการนำเอาเทคนิคเอ็นแกรมเข้ามาประยุกต์ในด้านประมวลผลทางภาษาไทยทั้งสิ้น แต่ยังไม่มีงานวิจัยใดเลยที่นำเอาเทคนิคเอ็นแกรมเข้ามาประยุกต์กับการประมวลผลอักษรเบรลล์ภาษาไทย นอกจากนี้อักษรเบรลล์ภาษาไทยมีความคล้ายคลึงกันกับภาษาไทยและมีปัญหาการหาขอบเขตคำเช่นเดียวกันกับภาษาไทยอีกด้วย

ดังนั้นผู้วิจัยจึงมีแนวความคิดนำเทคนิคเอ็นแกรมเข้ามาช่วยหาขอบเขตของคำอักษรเบรลล์ภาษาไทย ซึ่งจะทำให้ทราบขอบเขตที่แน่นอนของคำแต่ละคำจึงส่งผลให้การแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยทำได้ง่ายขึ้นและถูกต้องมากยิ่งขึ้น อีกทั้งการนำเทคนิคเอ็นแกรมมาประยุกต์ใช้นี้ ทำให้กระบวนการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยมีความซับซ้อนน้อยลงเมื่อเปรียบเทียบกับการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคกฎไวยากรณ์ทางภาษาเข้ามาควบคุมการแปลง โดยให้เอ็นแกรมคำนวณหาค่าความน่าจะเป็นของคำที่เกิดขึ้นร่วมกันเมื่อแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย เพื่อช่วยในการระบุขอบเขตของคำอักษรเบรลล์ภาษาไทยว่าควรแบ่งคำออกมาเป็นรูปแบบไหนจึงจะเหมาะสม และมีความถูกต้องมากที่สุด

วิทยานิพนธ์นี้ จึงมุ่งเน้นวิจัยและพัฒนาเพื่อศึกษาการนำเทคนิคเอ็นแกรมเข้ามาประยุกต์กับการแปลงอักษรเบรลล์ภาษาไทย เพื่อช่วยปรับปรุงกระบวนการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยให้ถูกต้องมากยิ่งขึ้น โดยมีประเด็นที่ศึกษาดังต่อไปนี้

1) การนำเทคนิคเอ็นแกรมเข้ามาประยุกต์ใช้ร่วมกับการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทย จะช่วยเพิ่มประสิทธิภาพในการแปลงได้หรือไม่ โดยเปรียบเทียบกับโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม ซึ่งได้นำเสนอไว้ในวิทยานิพนธ์เรื่อง “การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด” ซึ่งเป็นโปรแกรมที่ได้มีการพัฒนามาก่อนหน้านี้แล้ว
2) การหาค่าของจำนวนแกรมที่เหมาะสม เพื่อหาค่าของจำนวนแกรมที่ให้ผลลัพธ์ในการแบ่งคำที่ดีที่สุดโดยเปรียบเทียบระหว่าง 3-แกรม, 4-แกรม และ 5-แกรม โดยใช้วิธีพิจารณาจากความถูกต้องของการแปลง ซึ่งจะต้องถูกต้องตามหลักไวยากรณ์ของอักษรเบรลล์และเมื่อนำไปแปลงให้เป็นภาษาไทยแล้วจะต้องถูกต้องตามหลักไวยากรณ์ของภาษาไทยด้วย และด้านปริมาณการใช้หน่วยความจำในการแปลงและด้านเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยมาประกอบกัน

1.2 วัตถุประสงค์

1) ศึกษาและพัฒนาการนำเอ็นแกรมเข้ามาช่วยการแบ่งคำอักษรเบรลล์และแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยที่เกี่ยวข้องกับ คำควบกล้ำ, คำที่มีการใช้สระประสม และคำที่มีการใช้อักษรเบรลล์สองเซลล์และคำทับศัพท์
2) เพื่อทดสอบและเปรียบเทียบความถูกต้องการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยระหว่างโปรแกรมแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยตัดสินใจด้านการแบ่งคำและการแปลงอักษรอักษรเบรลล์ที่ได้เสนอไว้ในวิทยานิพนธ์นี้กับโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิมที่ได้มีการพัฒนามาก่อนหน้านี้
3) เพื่อศึกษาจำนวนแกรมที่เหมาะสมสำหรับใช้แบ่งคำอักษรเบรลล์ภาษาไทยโดยพิจารณาเปรียบเทียบจากด้านความถูกต้องของการแปลง, ด้านปริมาณการใช้หน่วยความจำในการแปลงและด้านเวลาที่ใช้ในการแปลง

1.3 ขอบเขตของการวิจัย

1) ศึกษา วิเคราะห์ ออกแบบและพัฒนาโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยตัดสินใจด้านการแบ่งคำและการแปลงอักษรเบรลล์
2) โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยตัดสินใจด้านการแบ่งคำและการแปลงอักษรเบรลล์ที่ได้นำเสนอไว้ในวิทยานิพนธ์นี้เน้นในด้านการแก้ไขปัญหาการแปลงอักษรเบรลล์ภาษาไทยที่เกี่ยวข้องกับ การแปลงคำควบกล้ำ, การแปลงคำที่มีการใช้สระประสม, การแปลงคำที่มีการใช้อักษรเบรลล์สองเซลล์และการแปลงคำที่มีการใช้คำทับศัพท์
3) เปรียบเทียบประสิทธิภาพระหว่างโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิมกับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยตัดสินใจด้านการแบ่งคำและการแปลงอักษรเบรลล์ในด้านความถูกต้องของการแปลง, ด้านปริมาณการใช้หน่วยความจำและด้านเวลาที่ใช้ในการแปลง
4) เปรียบเทียบจำนวนแกรมที่เหมาะสมสำหรับใช้แบ่งคำอักษรเบรลล์ภาษาไทยโดยพิจารณาจากด้านความถูกต้องของการแปลง, ด้านปริมาณการใช้หน่วยความจำในการแปลงและ

ด้านเวลาที่ใช้ในการแปลงและเปรียบเทียบระหว่างจำนวนแกรม 3-แกรม, 4-แกรม และ 5-แกรม

5) โปรแกรมที่พัฒนาขึ้นมานี้ครอบคลุมเฉพาะการแปลงอักษรเบรลล์ภาษาไทยระดับ 1

1.4 ขั้นตอนการวิจัยและวิธีการวิจัย

1) ศึกษางานวิจัยที่เกี่ยวข้องกับเอ็นแกรมโมเดลและเทคนิคการประมาณค่าความน่าจะเป็น
2) ศึกษางานวิจัยที่เกี่ยวข้องกับการประยุกต์เอ็นแกรมโมเดลและเทคนิคการตัดคำ
3) ศึกษางานวิจัยที่เกี่ยวข้องกับคลังข้อมูลและเทคนิคการฝึกฝนเอ็นแกรมโมเดล
4) ออกแบบและพัฒนาโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ
5) ออกแบบและพัฒนาการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยแบบวิธีจับคู่คำ หลังจากใช้เทคนิคเอ็นแกรมโมเดลช่วยในการตัดคำอักษรเบรลล์ภาษาไทยแล้ว
6) ทดสอบเปรียบเทียบประสิทธิภาพระหว่างโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ ในด้านความถูกต้องของการแปลง, ด้านปริมาณการใช้หน่วยความจำและด้านเวลาที่ใช้ในการแปลง
7) ทดสอบเปรียบเทียบหาจำนวนแกรมที่เหมาะสมของโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ เมื่อใช้จำนวนแกรมเท่ากับ 3-แกรม, 4-แกรม และ 5-แกรม
8) ตรวจสอบรายละเอียดความถูกต้องของโปรแกรมที่ได้พัฒนา
9) รวบรวมผลการทดสอบ สรุปผล จัดทำวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ช่วยแก้ไขปัญหาในการแปลงอักษรเบรลล์ภาษาไทยที่เกี่ยวข้องกับการแปลงคำควบกล้ำ, การแปลงคำที่มีการใช้สระประสม, การแปลงคำที่มีการใช้อักษรเบรลล์สองเซลล์และการแปลงคำที่มีการใช้คำทับศัพท์
2) ช่วยแก้ไขปัญหาการหาขอบเขตคำของอักษรเบรลล์ภาษาไทยระดับ 1

ในบทที่ 2 จะกล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้อง ในบทที่ 3 จะกล่าวถึงการออกแบบและพัฒนาระบบ ในบทที่ 4 จะกล่าวถึงผลการทดลอง และในบทที่ 5 จะกล่าวถึงสรุปผลการวิจัยและข้อเสนอแนะ

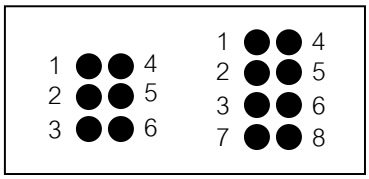
บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ในบทนี้จะนำเสนอทฤษฎีและหลักการที่เกี่ยวข้อง ซึ่งประกอบด้วย อักษรเบรลล์, โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file, เทคนิคการตัดคำ, คลังข้อมูลภาษา, เอ็นแกรมโมเดล, Smoothing Technique, CMU-SLM Toolkit และโปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์ (Thai2brl)

2.1 อักษรเบรลล์

หลุยส์ เบรลล์ (Louis Braille) ครูตาบอดชาวฝรั่งเศส ได้ประดิษฐ์อักษรสำหรับคนตาบอดขึ้นในปี ค.ศ.1834 [29] โดยประกอบด้วยจุดนูน 6 จุด ใน 1 ช่อง โดยแบ่งเป็น 2 แถวในแนวตั้ง แถวละ 3 จุด ซึ่งสามารถจัดกลุ่มของจุดได้ถึง 63 รูปแบบ และได้แพร่หลายไปทั่วโลก จนมีผู้ดัดแปลงนำไปใช้ในกลุ่มประเทศที่ใช้อักษรโรมันเป็นหลักทั่วยุโรป จากนั้นในปี ค.ศ.1932 ประเทศอังกฤษและสหรัฐอเมริกา ได้ร่วมมือกันจัดทำมาตรฐานอักษรเบรลล์ภาษาอังกฤษ (English Braille Standard) ขึ้นในปี ค.ศ.1932 ต่อมามีการพัฒนาให้เป็น 8 จุด [35] เพื่อให้สอดคล้องกับรหัสอักษรในระบบคอมพิวเตอร์ โดยเพิ่มจุดสองจุดลงไปในแต่ละคอลัมน์ของอักษรเบรลล์เดิม ดังรูปที่ 2.1



รูปที่ 2.1 การเรียงจุดในอักษรเบรลล์

อักษรเบรลล์ภาษาอังกฤษ ถือว่าเป็นอักษรเบรลล์สากลที่ใช้กันทั่วโลก และแต่ละประเทศนำมาดัดแปลง ปรับปรุงแก้ไข เพื่อนำไปใช้ในประเทศของตนเองเพื่อใช้สื่อสารระหว่างคนตาบอดด้วยกันรวมทั้งประเทศไทยด้วย โดยนอกเหนือจากการเขียนเป็นอักษรแทนพยัญชนะหรือสระในภาษาต่างๆ แล้ว ยังสามารถเขียนแทนสัญลักษณ์ต่างๆ เช่น คณิตศาสตร์, วิทยาศาสตร์, ดนตรี และเครื่องหมายต่างๆ ได้อีกด้วยด้วย โดยเรียกการอ่าน-เขียนอักษรเบรลล์แบบปกติว่า “อักษรเบรลล์

ระดับ 1” และเรียกการอ่าน-เขียนอักษรเบรลล์แบบย่อคำว่า “อักษรเบรลล์ระดับ 2” ซึ่งทั้งอักษรเบรลล์ภาษาอังกฤษและอักษรเบรลล์ภาษาไทยก็มีการอ่าน-เขียนในแบบระดับ 2 ด้วย

2.1.1 อักษรเบรลล์ภาษาไทย

นางสาวเจเนวีฟ คอลฟิลด์ (Miss Gevevive Caulfield) สุภาพสตรีตาบอดชาวอเมริกัน เป็นคนแรกที่นำความรู้เกี่ยวกับอักษรเบรลล์มาเผยแพร่ในประเทศไทย [29] พร้อมกับจัดตั้งโรงเรียนสอนคนตาบอดกรุงเทพขึ้นเมื่อปี พ.ศ.2482 และได้ร่วมกับชาวไทยคณะหนึ่งจัดตั้ง มูลนิธิช่วยคนตาบอดแห่งประเทศไทย และกำหนดอักษรเบรลล์ภาษาไทย โดยดัดแปลงมาจากอักษรเบรลล์อังกฤษมาตรฐาน ซึ่งผู้วิจัยได้แสดงรูปแบบอักษรเบรลล์ภาษาไทยไว้ในตารางที่ 11 ในภาคผนวก ง

อักษรเบรลล์ภาษาไทย ถูกกำหนดโดยการเลียนเสียงเทียบกับอักษรภาษาอังกฤษ เช่น ‘ก’ จะมีรูปแบบอักษรเบรลล์เหมือนอักษร ‘g’ ในภาษาอังกฤษ คือ จุด 1-2-4-5 (⠛) เป็นต้น และผู้พิการทางสายตาเรียกข้อมูลที่เก็บอยู่ในรูปอักษรรหัสแอสกีภาษาอังกฤษ ที่มีรูปแบบอักษรเบรลล์เหมือนกับข้อความภาษาไทยว่า ข้อมูล “Braille Computer Code”

เนื่องจากอักษรเบรลล์แบบ 6 จุด สามารถแสดงรูปแบบที่แตกต่างกันได้เพียง 63 รูปแบบ ซึ่งไม่เพียงพอที่จะใช้แทนอักษรภาษาไทยได้ทั้งหมด จึงได้มีการกำหนด อักษรเบรลล์แบบสองเซลล์ขึ้น โดยพยัญชนะภาษาไทยซึ่งมีทั้งสิ้น 44 ตัว เมื่อเทียบเสียงเหมือนกันแล้วจะมี 28 เสียง โดยกำหนดให้ พยัญชนะที่ใช้กันมากที่สุดเป็นพยัญชนะพื้นฐาน และกำหนดรูปอักษรเบรลล์เป็นแบบหนึ่งเซลล์ ส่วนพยัญชนะที่มีเสียงเหมือนกันกับพยัญชนะพื้นฐานให้ใช้ อักษรเบรลล์แบบ 2 เซลล์ โดยมีเซลล์หลังเหมือนกับพยัญชนะพื้นฐาน และมีเซลล์หน้าเป็น จุด 6 (⠠) หรือ จุด 3-6 (⠤) หรือ จุด 3-5-6 (⠴) ซึ่งผู้วิจัยได้รวบรวมพยัญชนะภาษาไทยที่มีรูปอักษรเบรลล์แบบ 2 เซลล์แสดงไว้ในตารางที่ 2.1

ตารางที่ 2.1 พยัญชนะภาษาไทยที่มีรูปอักษรเบรลล์แบบ 2 เซลล์

เซลล์หน้า	ตัวอักษร
, (⠠)	ศ(⠠⠎), ฆ(⠠⠥), ฌ (⠠⠬), ฬ(⠠⠇), ณ(⠠⠝), ฑ(⠠⠾), ญ(⠠⠽), ฎ(⠠⠙), ฏ(⠠⠸), ฐ (⠠⠞), ภ(⠠⠦)
- (⠤)	ษ(⠤⠎), ค(⠤⠥), ฒ(⠤⠾)
อ (⠴)	ธ(⠴⠾), ฃ(⠴⠅)

สระในภาษาไทยกำหนดรูปอักษรเบรลล์ตามเสียงสระ 32 เสียง โดยมีการใช้งานรหัสอักษรสระประสมซึ่งไม่มีในอักษรรหัสแอสกีด้วย โดยผู้วิจัยได้แสดงรูปสระประสมของอักษรเบรลล์ทั้งหมดไว้ในตารางที่ 2.2 ซึ่งจะเห็นได้ว่ามีการใช้งานสระประสมแบบ 2 เซลล์ด้วย โดยมีการกำหนดอักษรเบรลล์ให้กับสระพื้นฐานทุกตัว และกำหนดให้ใช้อักษรเบรลล์สองเซลล์แทนสระเสียงสั้นที่มีรูปสระเสียงยาว เช่น สระเอะ, สระแอะ โดยใช้สระอะต่อท้ายรูปสระเสียงยาว เช่น สระเอะ (fa fa) จะเขียนอยู่ในรูปสระเอ (f f) และสระอะ (a a) ต่อกันเป็นต้น

นอกจากพยัญชนะที่มีรูปอักษรเบรลล์แบบ 2 เซลล์แล้ว ยังมีการใช้สระและเครื่องหมายที่ใช้รูปอักษรเบรลล์ 2 เซลล์ด้วย โดยผู้วิจัยได้แสดงเครื่องหมายที่มีรูปอักษรเบรลล์แบบ 2 เซลล์ไว้ในตารางที่ 2.3

ตารางที่ 2.2 รูปอักษรเบรลล์ของสระประสม

สระ	รูปอักษรเบรลล์	Braille Computer Code
อัว	e	e
อัวะ	ea	ea
เอา	6	6
เอีย	(	(
เอือ	q	q
เออ	%	%
เอะ	fa	fa (เ-ะ)
แอะ	<a	<a (แ-ะ)
โอะ	ia	ia (โ-ะ)
เอียะ	(a	(a (เอีย-ะ)
เอือะ	qa	qa (เอือ-ะ)
เออะ	%a	%a (เออ-ะ)
เอาะ	oa	oa (อ-ะ)

ตารางที่ 2.3 สระและเครื่องหมายที่มีรูปอักษรเบรลล์แบบ 2 เซลล์

แอสกี	รูปอักษรเบรลล์	Braille Computer Code
ใ	:1	:1 (ไๆ)
ฤ	r1	r1 (รๆ)
ฤๅ	r1*	r1* (รๆา)
ฦ	l1	l1 (ลๆ)
ฦๅ	l1*	l1* (ลๆา)
ฯ	;2	;2
ฯลฯ	;l	;l

2.1.2 หลักการอ่านเขียนอักษรเบรลล์ภาษาไทยระดับ 1

หลักการอ่าน-เขียนอักษรเบรลล์ภาษาไทยระดับ 1 [33] [34] สามารถสรุปกฎเป็นข้อๆ ได้ดังนี้

ข้อ 1: คำที่ไม่ประกอบด้วยสระประสม วางรูปคำเหมือนกับการเขียนปกติตามหลักภาษาศาสตร์

พยัญชนะต้น+สระ+วรรณยุกต์+ตัวสะกด

เช่น ลิ้น ➔ ล + อิ + ้ + น lb4n lb4n

ข้อ 2: สระต่อไปนี้จะเขียนหลังพยัญชนะเสมอ ได้แก่ อะ, อา, อิ, อี, อึ, อือ, อุ, อู, เอะ, แอะ, โอะ, เอาะ, ออ, อัวะ, อัว, เอียะ, เอีย, เอือะ, เอือ, เออะ, เออ, อำ, เอา และเมื่อคำที่ใช้สระเหล่านี้มีรูปวรรณยุกต์ประสมอยู่ด้วยจะต้องเขียนหลังสระ ยกเว้น สระอา กับ สระ ออ ให้เขียนรูปวรรณยุกต์ก่อนสระ

พยัญชนะต้น+สระ+วรรณยุกต์+ตัวสะกด

เช่น เรียน ➔ ร + เอีย + น r(n r(n

ข้อ 3: คำที่เขียนเหมือนกันในรูป สระเอ + สระอา ที่ออกเสียง ได้ทั้งสระอาและสระเอาให้ใช้เป็นสระเอา

พยัญชนะ + สระเอา + วรรณยุกต์

เช่น เพลา ➔ พ + ล + เอา ?l6 ?l6

ข้อ 4: คำที่มีสระ เอ แอ โอ ไอ ใอ ในอักษรเบรลล์ให้เขียนเรียงตามปกติเช่นเดียวกับการเขียนภาษาไทย

สระ + พยัญชนะ + ตัวสะกด + วรรณยุกต์

เช่น ไหล่ ➔ ไ + ห + ล + อ่ :hl4 :hl4

จากหลักการอ่าน-เขียนอักษรเบรลล์ภาษาไทยระดับ 1 จะเห็นได้ว่าอักษรเบรลล์ และอักษรในรหัสแอสกี ไม่สามารถที่จะเทียบกันได้แบบตัวต่อตัว โดยกรณีที่ไม่สามารถเทียบข้อมูลได้แบบตัวต่อตัวได้แก่

- การใช้อักษรเบรลล์ 2 เซลล์แทนอักษรในรหัสแอสกี 1 ตัว โดยมีการใช้งานอักษรเบรลล์แบบ 2 เซลล์ทั้งพยัญชนะ สระ และเครื่องหมายในภาษาไทย
- การใช้งานอักษรเบรลล์ 1 เซลล์เพื่อแทนอักษรในรหัสแอสกีหลายตัวในการใช้งานสระประสม
- การอ่าน-เขียนคำที่ใช้ สระอะ หรือสระอำ ร่วมกับวรรณยุกต์ มีการเรียงอักษรรหัสแอสกี และอักษรเบรลล์แตกต่างกัน โดยอักษรเบรลล์จะเขียนเป็น พยัญชนะต้น + สระ + วรรณยุกต์ แต่ ในรหัสแอสกี จะเขียนเป็น พยัญชนะต้น + วรรณยุกต์ + สระ เช่น “จ๊ะ” (จ-อ๊-ะ) เขียนอักษรเบรลล์เป็น “ja7” (จ-ะ-อ๊) “ค้ำ” (ค-อ้-อำ) เขียนอักษรเบรลล์เป็น “u4z” (ค-อำ-อ้)
- อักษรเบรลล์มีการใช้สระประสม แต่ไม่มีในอักษรรหัสแอสกี ทำให้มีการเรียงอักษรแตกต่างกัน และเนื่องจากการเขียนข้อความภาษาไทยไม่มีการเว้นวรรคระหว่างคำ จึงทำให้มีความกำกวมเกิดขึ้นว่า คำๆ นั้นจะเป็นคำที่ใช้สระประสมหรือคำหลายคำต่อกันดังนี้
  - คำที่ใช้สระเออและสระเออะที่มีพยัญชนะต้น 1 ตัว เช่นคำว่า “เลอะ” เมื่อเขียนเป็นอักษรเบรลล์จะเขียนเป็นสระประสมได้เป็น “l%a” (“ล-เออะ”) แต่คำว่า “ทะเลอะไร” (“ทะเล”-“อะไร”) ซึ่งพ้องรูปกับคำว่า “ทะ-เลอะ-ไร” เมื่อเขียนเป็นอักษรเบรลล์จะเขียนเป็น “)afloa:r” (“ท-ะ-เ-ล-อ-ะ-ไ-ร”) ไม่ใช่ “)a l%a:r” (“ท-ะ-ล-เออะ-ไ-ร”)
  - คำที่ใช้พยัญชนะต้น 2 ตัว เช่นคำว่า “เจริญ” เขียนแบบอักษรเบรลล์เป็น “jr%,y” (“จ-ร-เออ-ญ”) ไม่ใช่ “fjrb,y” (“เ-จ-ร-อิ-ญ”) คำว่า “เกอิชา” เขียนแบบอักษรเบรลล์เป็น “fgob+*” (“เ-ก-อ-อิ-ช-า”) ไม่ใช่ “go%+a” (“ก-อ-เออ-ช-า”) และไม่ใช่ “g%b+*” (“ก-เออ-อิ-ช-า”) คำว่า

"เทมือ" เขียนแบบอักษรเบรลล์เป็น "f)m5o" ("เ-ท-ม-อื-อ") ไม่ใช่ ")m%" ("ท-ม-เอือ") เป็นต้น

การเขียนอักษรเบรลล์จะใช้งานสระประสม ซึ่งต่างจากการเขียนในภาษาไทยปกติ และเป็นปัญหาสำคัญในการแปลงข้อมูลระหว่างไทยและเบรลล์ โดยในปัจจุบันมีการใช้งานอักษรเบรลล์ภาษาไทย โดยใช้การอ่าน-เขียนอักษรเบรลล์แบบระดับ 1 แต่ไม่ใช้สระประสมและผู้พิการทางสายตาเรียกการอ่าน-เขียนเช่นนี้อย่างไม่เป็นทางการว่า "อักษรเบรลล์ภาษาไทยระดับ 0"

นอกเหนือจากอักษรเบรลล์แบบ 6 จุด แล้ว ยังได้มีการกำหนดรูปแบบของอักษรเบรลล์ภาษาไทยแบบ 8 จุดขึ้น [35] เพื่อให้ให้สอดคล้องกับระบบคอมพิวเตอร์ทั่วไปด้วย โดยกำหนดให้สามารถเทียบอักขระภาษาไทยแบบปกติและอักษรเบรลล์แบบ 8 จุดได้แบบตัวต่อตัว โดยผู้วิจัยได้แสดงรูปแบบอักษรเบรลล์ภาษาไทยทั้งแบบ 6 จุด และ 8 จุดไว้ในตารางที่ 11-12 ในภาคผนวก ง

2.2 โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file ของมหาวิทยาลัยสงขลานครินทร์

โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file [1] ได้นำเสนอไว้ในงานวิจัยเรื่อง "การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด" ของมหาวิทยาลัยสงขลานครินทร์ ซึ่งโปรแกรมดังกล่าวทำหน้าที่แปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยในรูปแบบ File-to-file โดยกระบวนการทำงานของโปรแกรมนี้ ได้ใช้วิธีการแปลงแบบใช้กฎไวยากรณ์ทางภาษาของอักษรเบรลล์ภาษาไทยมาควบคุมการแปลง ซึ่งสามารถให้ผลลัพธ์ในการแปลงที่ถูกต้องในระดับหนึ่ง

จากการศึกษางานวิจัยนี้พบว่ายังคงมีปัญหาในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยอยู่ด้วยกัน 2 ประการคือ 1) ปัญหาการแปลงคำควบกล้ำและสระประสม ซึ่งปัญหานี้เกิดจากการต่อคำทำให้เกิดความกำกวมของคำขึ้น เช่น คำว่า "เครือ" เขียนอยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 เป็น "urq" ซึ่งก็คือ "ค+ร+เอือ" แต่เนื่องจากคำว่า "rq" หรือ "ร+เอือ" เป็นคำที่มีความหมายว่า "เรือ" ก็เป็นคำที่มีความหมายในภาษาไทยเช่นกัน และอักษรเบรลล์ 'u' ซึ่งก็คืออักษร "ค" ในภาษาไทยของว่า "เครือ" อาจจะเป็นตัวควบกล้ำของคำปัจจุบันหรือเป็นตัวสะกดของคำก่อนหน้าก็ได้

และ 2) ปัญหาการแปลงคำที่มีการใช้อักษรเบรลล์สองเซลล์ ซึ่งปัญหานี้เกิดจากมีอักษรเบรลล์สองเซลล์บางตัวที่มีรูปพ้องกันกับการเขียนอักษรเบรลล์เซลล์เดียวสองตัวเขียนติดกัน ซึ่งปัญหานี้โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม ยังคงมีการแปลงที่ผิดพลาดอยู่เนื่องจาก การแปลงแบบ File-to-file นั้นโปรแกรมเดิมจะไม่สามารถทราบได้ว่าตรงไหนเป็นอักษรเบรลล์สองเซลล์หนึ่งตัวหรือเป็นอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกัน ซึ่ง

จะแตกต่างกับกรณีของโปรแกรมการแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด [1] ที่สามารถแก้ไขปัญหาเหล่านี้ได้โดยใช้วิธีให้ผู้ใช้กดคีย์พิเศษก่อนจะป้อนอักษรเบรลล์สองเซลล์และการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยแบบ File-to-file นั้นผู้ใช้จะไม่สามารถโต้ตอบกับโปรแกรมเพื่อระบุว่าตรงไหนเป็นอักษรเบรลล์สองเซลล์ได้ ยกตัวอย่างเช่นคำว่า “พระองค์ทรง” เขียนในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 คือ “?rao}u0)r}” ซึ่งก็คือ “พ+ร+ะ+อ+ง+ค+การันต์+ท+ร+ง” โดยอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกันแล้วมีรูปพ้องกันกับอักษรเบรลล์สองเซลล์ในที่นี้คือ ‘0+)’ ซึ่งก็คืออักษร ‘การันต์+ท’ โดยมีรูปพ้องกันกับอักษรเบรลล์สองเซลล์คืออักษร ‘0)’ ซึ่งก็คืออักษร ‘ธ’ เมื่อโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิมทำการแปลงจะได้เป็นคำว่า “พระองคธรง” เพราะโปรแกรมเดิมจะไม่ทราบว่าเป็นอักษรเบรลล์สองเซลล์หนึ่งตัวหรืออักษรเบรลล์เซลล์เดียวสองตัว ดังสาเหตุที่ได้กล่าวไว้แล้วในข้างต้น

ดังนั้นงานวิจัยนี้จึงได้มุ่งเน้นแก้ไขปัญหาทั้งสองประการที่ได้กล่าวไว้ในข้างต้นโดยใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการแบ่งคำ เพื่อช่วยให้โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยทราบขอบเขตของคำแต่ละคำอย่างชัดเจน ทำให้ไม่มีความกำกวมที่เกิดขึ้นจากการต่อคำและความกำกวมที่เกิดจากการไม่ทราบว่าเป็นอักษรเบรลล์สองเซลล์หนึ่งตัวหรือเป็นอักษรเบรลล์เซลล์เดียวสองตัว

2.3 เทคนิคการตัดคำ

เทคนิคการตัดคำ [6] ได้ถูกนำมาใช้ในงานด้านประมวลผลภาษาธรรมชาติ [7] เพื่อทำให้คอมพิวเตอร์สามารถเข้าใจความหมายของคำในภาษานั้นๆ ได้และเนื่องจากภาษาบางภาษา เช่น ภาษาไทย, ภาษาจีน, ภาษาญี่ปุ่น ภาษาลาว เป็นต้น มีลักษณะการเขียนประโยคที่ประกอบไปด้วยคำย่อยๆ หลายคำเรียงติดกันโดยไม่มีการเว้นช่องว่างระหว่างคำเหมือนกับภาษาอังกฤษทำให้เกิดปัญหา “การหาขอบเขตของคำ” เมื่อต้องการนำข้อมูลเหล่านั้นไปประมวลผลด้วยคอมพิวเตอร์ ซึ่งคอมพิวเตอร์ไม่สามารถทราบได้ว่าในประโยคนั้นๆ ประกอบด้วยคำกี่คำ และแต่ละคำประกอบด้วยตัวอักษรอะไรบ้าง จึงจำเป็นต้องตัดคำเสียก่อน ก่อนที่จะนำไปประมวลผลต่อ เพื่อแบ่งขอบเขตของคำแต่ละคำอย่างชัดเจน

อักษรเบรลล์ภาษาไทยมีลักษณะการเขียนเช่นเดียวกันกับภาษาไทยคือ เขียนคำแต่ละคำติดๆ กันเป็นประโยค และเว้นวรรคระหว่างประโยค แต่จะเรียงตัวอักษรนำ ตัวอักษรตาม สระ ตัวสะกด และวรรณยุกต์ที่แตกต่างไปจากภาษาไทย จากความคล้ายคลึงดังกล่าวทำให้ผู้วิจัยมีแนวความคิดว่าหากใช้เทคนิคการตัดคำเข้ามาช่วยแบ่งคำอักษรเบรลล์ภาษาไทย เพื่อให้ทราบขอบเขตของคำที่ชัดเจนแล้ว จะทำให้ไม่เกิดความกำกวมขึ้นอีกต่อไปและทำให้การแปลงอักษร

เบรลล์ภาษาไทยเป็นภาษาไทยทำได้ง่ายขึ้นและถูกต้องมากยิ่งขึ้นอีกด้วย นอกจากนี้ยังไม่มีงานวิจัยใดๆ เลยที่นำวิธีการตัดคำมาใช้กับอักษรเบรลล์ภาษาไทย ซึ่งในปัจจุบันมีเพียงงานวิจัยที่เกี่ยวข้องกับการตัดคำภาษาไทยเท่านั้น และจากการศึกษาพบว่าเทคนิคการตัดคำนั้นมีด้วยกันหลายเทคนิค [8][9][10][11] ซึ่งสามารถแบ่งเป็นประเภทต่างๆ ได้ดังนี้ 1) เทคนิคการตัดคำโดยใช้กฎพื้นฐาน 2) เทคนิคการตัดคำโดยใช้พจนานุกรม 3) เทคนิคการตัดคำโดยวิธีเทียบคำที่ยาวที่สุด 4) เทคนิคการตัดคำโดยวิธีตัดคำให้ได้จำนวนคำและคำที่ไม่พบในพจนานุกรมน้อยที่สุด 5) เทคนิคการตัดคำโดยใช้คลังข้อมูล ซึ่งรายละเอียดของแต่ละเทคนิคมีดังต่อไปนี้

2.3.1 เทคนิคการตัดคำโดยใช้กฎพื้นฐาน

เทคนิคการตัดคำโดยใช้กฎพื้นฐาน [8] เป็นการตัดคำโดยอาศัยกฎที่ได้ตั้งเอาไว้มาใช้เป็นเกณฑ์ในการแบ่งขอบเขตของคำ โดยกฎที่นำมาใช้นั้นได้นำมาจากหลักไวยากรณ์ภาษาไทย ซึ่งลักษณะของกฎสามารถแบ่งได้เป็น 2 ชนิดคือ กฎการหาขอบเขตหน้า (Front boundary recognition rule) และกฎการหาขอบเขตหลัง (Tail boundary recognition rule) และในแต่ละกฎยังแบ่งออกเป็น 2 กลุ่มย่อยๆ คือ แบ่งตามคุณสมบัติของตัวอักษรโดยกฎที่ได้นี้จะจัดอยู่ในกลุ่มเอ (Group A) และแบ่งตามคุณสมบัติของรูปแบบการใช้สระแต่ละตัว ซึ่งกฎที่ได้นี้จะแบ่งให้อยู่ในกลุ่มบี (Group B) ตัวอย่างกฎกลุ่มเอ เช่น

ตัวอย่างกฎที่ได้จากคุณสมบัติของอักษรในการหาขอบเขตหน้าของพยางค์ เช่น

- กฎ A – 1F : สระต่างๆ เหล่านี้ ะ า ิ ี ึ ื ุ ู ็ ำ ู ั และวรรณยุกต์ ่ ้ ๊ ๋ จะต้องมีพยัญชนะอยู่ข้างหน้าอย่างน้อย 1 ตัวอักษรเสมอ
- กฎ A – 2F : สระ เ แ ไ โ ส่วนใหญ่จะเป็นตัวอักษรแรกในพยางค์เสมอ ยกเว้นคำบางคำ ตัวอย่างเช่น ขโมย,ทแยง,จเร,สไบ เป็นต้น
- กฎ A – 3F : สระ ใ จะเป็นตัวอักษรแรกของพยางค์เสมอ โดยไม่มีข้อยกเว้น

ตัวอย่างกฎที่ได้จากคุณสมบัติของอักษรในการหาขอบเขตหลังของพยางค์ เช่น

- กฎ A – 1T : พยัญชนะต่อไปนี้ ศ ณ ญ ฐ ฎ ฏ ฒ ฬ ฌ จะเป็นตัวสะกดเสมอ ยกเว้นพยางค์ดังต่อไปนี้ ศก ศร ศพ ฐก ฬส ญวน ณรงค์ ศตวรรษ และพยางค์อื่นๆ แต่พยางค์เหล่านั้นสามารถจัดการได้โดยใช้กฎ A – 1F
- กฎ A – 2T : สระ ึ จะตัวมีสะกดหนึ่งตัวเสมอ ยกเว้น พยางค์เหล่านี้ “ ฤ ฬ ฮึ ”
- กฎ A – 3T : ไม้หันอากาศ (ั) จะต้องตามด้วยตัวสะกดอย่างน้อย 1 ตัวอักษรเสมอ

จากตัวอย่างกฎที่แสดงข้างต้นจะเห็นได้ว่าการหาขอบเขตของพยางค์สามารถจะนำคุณสมบัติของตัวอักษรต่างๆ มาสร้างเป็นกฎเพื่อใช้ในการแบ่งพยางค์ได้ นอกจากนี้มีการสร้างกฎที่ใช้ในการแบ่งพยางค์ โดยกฎที่ได้จะพิจารณาจากคุณสมบัติของรูปแบบการใช้สระแต่ละตัวหรือกฎกลุ่มบี ซึ่งจะพิจารณาจากลักษณะของพยางค์ในภาษาไทย โดยลักษณะของพยางค์ในภาษาไทยนั้นจะมีรูปแบบที่แน่นอนและตายตัว ทำให้งานวิจัยนี้มีการสร้างกฎต่างๆ โดยอ้างอิงจากรูปแบบของพยางค์ต่างๆ ที่ปรากฏในภาษาไทย ตัวอย่างกฎกลุ่มบี เช่น

ตัวอย่างกฎที่ได้จากคุณสมบัติของรูปแบบการใช้สระใช้ในการหาขอบเขตหลังของพยางค์ เช่น

- กฎ B – 1T : สระเหล่านี้ “ ั ิ ึ ี” ถ้ามีวรรณยุกต์ แล้วจะต้องมีตัวสะกด 1 ตัวเสมอ
- กฎ B – 4T : สระในรูปแบบดังต่อไปนี้ -ัว -ัย -ัวะ -ือ เ-าเ-าะเ-ะ แ-ะ โ–ะ เ–ียะ เ-อะ ไม่ต้องการตัวสะกด ยกเว้นบางพยางค์ในรูปแบบดังนี้ เ-า- และ เ-อ- โดยเครื่องหมาย - แทนตัวพยัญชนะ 1 ตัวอักษร

และในงานวิจัยที่ใช้เทคนิคการตัดคำโดยใช้กฎพื้นฐานมีการสร้างกฎสำหรับการกำกับพยางค์ที่ไม่ใช่ลักษณะของพยางค์ภาษาไทย ซึ่งพยางค์เหล่านั้นอาจประกอบไปด้วย อักขระพิเศษ ตัวเลข คำย่อ หรือ อักขระจากภาษาต่างชาติ จากการทดลองความถูกต้องในการตัดพยางค์ [8] ได้นำกฎเกณฑ์เหล่านี้ทดลองกับเอกสารต่างๆ จำนวน 100 เล่ม โดยเอกสารนั้นได้นำมาจากเอกสารชนิดต่างๆ จำนวน 10 ชนิด เช่น หนังสือพิมพ์ ปรัชญา วิทยาศาสตร์ ศาสนา ภาษาศาสตร์ ฯลฯ และจากการทดสอบปรากฏว่าสามารถตัดพยางค์ได้ถูกต้องถึง 96%

2.3.2 เทคนิคการตัดคำโดยใช้พจนานุกรม

เทคนิคการตัดคำโดยใช้พจนานุกรม [9][10] เป็นวิธีการตัดคำที่นำมาใช้กันต่อมา วิธีการนี้จำเป็นต้องเก็บข้อมูลของพจนานุกรมทั้งหมดไว้ในหน่วยความจำ ความถูกต้องในการตัดคำจะขึ้นอยู่กับจำนวนคำในพจนานุกรม และความเร็วในการทำงานขึ้นอยู่กับวิธีที่ใช้ในการค้นหาคำจากพจนานุกรม และโครงสร้างข้อมูลพจนานุกรม

หลักการตัดคำโดยใช้พจนานุกรมนี้ สามารถตัดคำได้ถูกต้องมากกว่าการใช้กฎพื้นฐาน ซึ่งวิธีการตัดคำโดยใช้พจนานุกรมนี้ที่มีการใช้งานอย่างแพร่หลาย คือ วิธีการเทียบคำที่ยาวที่สุด (Longest Matching) และวิธีการตัดคำให้ได้จำนวนคำและคำที่ไม่พบในพจนานุกรมน้อยที่สุด (Maximal Matching)

2.3.2.1 เทคนิคการตัดคำโดยวิธีเทียบคำที่ยาวที่สุด (Longest Matching)

เทคนิคการตัดคำโดยวิธีเทียบคำที่ยาวที่สุด [9] เป็นการตัดพยางค์ที่นำพจนานุกรมเข้ามาใช้ โดยจะจัดเก็บพยางค์ต่างๆ ไว้ในพจนานุกรม และนำกฎไวยากรณ์ต่างๆ จำนวน 18 กฎเข้ามาช่วยในกรณีที่ไม่พบพยางค์ในพจนานุกรม

หลักการทำงานของวิธีตัดพยางค์ด้วยพจนานุกรมนี้ก็คือ ตรวจสอบสายอักขระ (String) ที่เข้ามาจากซ้ายไปขวากับพยางค์ที่ได้เก็บไว้ในพจนานุกรม ในกรณีที่ตรวจสอบแล้วปรากฏว่าพบพยางค์มากกว่า 1 พยางค์ในพจนานุกรม ให้ทำการเลือกแบ่งพยางค์โดยเลือกพยางค์ที่ยาวที่สุด แล้วทำต่อไปเรื่อยๆ จนจบสายอักขระ แต่ในกรณีที่เลือกพยางค์ที่ยาวที่สุดแล้ว ทำให้เกิดพยางค์ที่ไม่ปรากฏในพจนานุกรมให้ใช้วิธีการย้อนรอย (Back Tracking) กลับไปเลือกพยางค์ที่ยาวรองลงมาแทน

จากการเปรียบเทียบความรวดเร็วในการแบ่งพยางค์กับเทคนิคการแบ่งพยางค์แบบใช้กฎพื้นฐาน ซึ่งสรุปผลได้ว่าเมื่อนำพจนานุกรมเข้ามาใช้ในการแบ่งพยางค์จะสามารถตัดพยางค์ได้รวดเร็วกว่าการใช้กฎ โดยที่ความถูกต้องของการตัดพยางค์นั้นสามารถตัดได้ถูกต้องมากกว่า 99% [9] แต่วิธีการนี้ก็ยังมีข้อเสียคือ ต้องใช้เนื้อที่หน่วยความจำเป็นจำนวนมากในการจัดเก็บรายการคำในพจนานุกรม และความถูกต้องของการแบ่งพยางค์จะขึ้นอยู่กับจำนวนคำที่มีอยู่ในพจนานุกรม และความรวดเร็วในการค้นหารายการคำในพจนานุกรมที่เก็บไว้ในหน่วยความจำ

2.3.2.2 เทคนิคการตัดคำโดยวิธีตัดคำให้ได้จำนวนคำและคำที่ไม่พบในพจนานุกรมน้อยที่สุด (Maximal Matching)

เทคนิคการตัดคำโดยวิธีตัดคำให้ได้จำนวนคำและคำที่ไม่พบในพจนานุกรมน้อยที่สุด [10] เป็นเทคนิคที่ลดข้อด้อยของวิธีการตัดคำโดยใช้พจนานุกรม ซึ่งจากการศึกษาวิจัยพบว่าหากประโยคหรือข้อความมีคำที่ไม่มีอยู่ในพจนานุกรมแล้ว จะทำให้ความถูกต้องในการตัดคำลดลงโดยมาก ซึ่งสาเหตุนั้นมาจากการตัดคำโดยอาศัยพจนานุกรมนี้ความถูกต้องจะขึ้นอยู่กับจำนวนคำที่มีอยู่ในพจนานุกรม ดังนั้นเทคนิคการตัดคำโดยวิธีตัดคำให้ได้จำนวนคำและคำที่ไม่พบในพจนานุกรมน้อยที่สุดจึงเป็นวิธีแก้ไขข้อด้อยเมื่อไม่มีคำนั้นๆ อยู่ในพจนานุกรม ซึ่งวิธีการตัดคำแบบนี้เป็นการหารูปแบบในการตัดคำทั้งหมดที่สามารถจะเป็นไปได้ เช่น เมื่อมีข้อความว่า "ไปหามเหสี" เมื่อใช้วิธีการนี้จะสามารถสร้างรูปแบบในการตัดคำได้ 2 แบบ คือ

1. ไป|หาม|เห|สี
2. ไป|หา|มเหสี

จากนั้นวิธีการนี้จะเลือกข้อความที่แบ่งแล้วมีจำนวนคำน้อยที่สุดคือ แบบที่ 2 ซึ่งมีจำนวนคำที่ตัดได้ 3 คำ ในขณะที่แบบที่ 1 มี 4 คำ แต่ในกรณีที่แบ่งคำออกมาแล้วมีจำนวนคำที่เท่ากัน จะใช้เทคนิคการตัดคำแบบยาวที่สุด (Longest Matching) เข้ามาช่วยตัดคำ เช่น ข้อความว่า "ฉันนั่งตากลมที่หน้าบ้าน" เมื่อใช้วิธีการนี้จะสามารถสร้างรูปแบบในการตัดคำได้ 2 แบบ คือ

1. ฉัน|นั่ง|ตาก|ลม|ที่|หน้า|บ้าน
2. ฉัน|นั่ง|ตา|กลม|ที่|หน้า|บ้าน

แต่ทั้ง 2 แบบนี้มีจำนวนคำที่เท่ากัน ในกรณีนี้จึงควรใช้วิธีการตัดคำแบบยาวที่สุด (Longest Matching) เข้ามาช่วยจะทำให้ผลลัพธ์ในการตัดคำประโยคดังกล่าวเป็นแบบที่ 1 เพราะเมื่อเปรียบเทียบจากคำที่ตัดได้ต่างกันนั้น (ตา / ตาก) จะเห็นได้ว่า “ตาก” มีตัวอักษรมากกว่าจึงเลือกแบบที่ 1

2.3.3 เทคนิคการตัดคำโดยใช้คลังข้อมูล

เทคนิคการตัดคำโดยใช้คลังข้อมูล [11] เป็นเทคนิคที่นำเอาวิธีการทางสถิติเข้ามาช่วยในการแก้ไขปัญหาความกำกวม โดยใช้ค่าสถิติที่เกิดขึ้นจากลำดับของหน้าที่ของคำแต่ละคำหรือประเภทย่อยของคำแต่ละคำ หรืออาจกล่าวได้ว่าเป็นการนำเอาไวยากรณ์มาใช้งานร่วมกับสถิติเพื่อแก้ปัญหาความกำกวม โดยงานวิจัยนี้ใช้แบบจำลองไตรแกรมและนำเอาค่าสถิติซึ่งพิจารณาจากความต่อเนื่องของหน้าที่ของคำหรือประเภทย่อยของคำ และเลือกเอาประโยคที่มีความน่าจะเป็นสูงที่สุดโดยการหาความน่าจะเป็นของแต่ละประโยคสามารถคำนวณตามสมการที่ 1

$$P(W_i) = \sum_T P(W_i, T_i)$$
$$= \sum_T \prod_i P(t_i|t_{i-1}, t_{i-2}) \times P(w_i|t_i) \quad \text{..... (1)}$$

จากสมการที่ 1 จะเป็นสูตรสำหรับการคำนวณหาค่าความน่าจะเป็นของแต่ละประโยคที่ตัดได้ โดยใช้วิธีตัดคำให้ได้จำนวนคำและคำที่ไม่พบในพจนานุกรมน้อยที่สุด (Maximal Matching) จากนั้นนำแต่ละรูปแบบของประโยคที่ตัดได้มาคำนวณหาค่าความน่าจะเป็น โดยพิจารณาจากผลรวมความน่าจะเป็นของหน้าที่ของคำ (T_i) หรือประเภทย่อยของคำ (W_i) ทุกแบบที่เป็นไปได้ของแต่ละประโยค และมีข้อกำหนดว่าความน่าจะเป็นของการเกิดหน้าที่คำหรือประเภทย่อยของคำที่ตำแหน่งปัจจุบันจะขึ้นอยู่กับหน้าที่ของคำ หรือประเภทย่อยของคำสองคำก่อนหน้าเท่านั้น ทำให้วิธีการนี้เหมาะสำหรับงานที่ต้องการทราบขอบเขตคำเพียงอย่างเดียวเท่านั้น ซึ่งวิธีนี้สามารถแก้ไขปัญหาความกำกวมได้ดี เนื่องจากมีการพิจารณาถึงหน้าที่ของคำหรือประเภทย่อยของคำเข้ามาประกอบด้วย

อย่างไรก็ตามในกรณีที่ข้อความมีความกำกวมโดยมีหน้าที่ของคำหรือประเภทย่อยของคำเหมือนกัน วิธีการนี้จะไม่สามารถตัดคำได้ดีนักและข้อจำกัดอีกประการหนึ่งคือจะต้องเก็บค่าสถิติจากคลังข้อความ โดยที่คลังข้อความที่ดีควรจะนำมาจากเอกสารหลายๆ ประเภท และจะต้องมีขนาดใหญ่พอสมควร

ดังนั้นประสิทธิภาพในการตัดคำแบบนี้จะขึ้นอยู่กับคลังข้อความด้วย ความถูกต้องสูงสุดสำหรับวิธีการนี้คือ 85% และวิธีนี้ยังคงเป็นวิธีการที่นิยมใช้มากที่สุด เนื่องจากความแม่นยำสูงและใช้อัลกอริทึมในการพัฒนาโปรแกรมทำได้ง่ายที่สุด

จากวิธีการตัดคำโดยใช้เทคนิคต่างๆ ที่ได้ศึกษามาพบว่าวิธีการตัดคำโดยใช้คลังข้อมูลสามารถตัดคำได้ถูกต้องมากที่สุดและมีข้อด้อยน้อยที่สุด ดังนั้นในงานวิทยานิพนธ์นี้จึงได้เลือกใช้วิธีการตัดคำโดยใช้คลังข้อมูล ซึ่งจะอยู่ในรูปแบบการตัดคำของอักษรเบรลล์ภาษาไทยระดับ 1

2.4 คลังข้อมูลภาษา

คลังข้อมูลภาษา [12] คือ ข้อมูลภาษาเขียนหรือภาษาพูดที่เป็นภาษาที่ใช้จริง ซึ่งถูกรวบรวมขึ้นมาในปริมาณที่มากเพียงพอตามเงื่อนไขที่กำหนดเพื่อให้ได้คลังข้อมูลที่เป็นตัวแทนของภาษาที่ต้องการ คลังข้อมูลภาษาจะถูกนำไปใช้งานในด้านภาษาศาสตร์ การสร้างแบบจำลองภาษา รวมถึงการประมวลผลภาษาธรรมชาติอีกด้วย เช่น นำไปใช้ในการสร้างพจนานุกรม นำข้อมูลด้านสถิติของคำไปสร้างเป็นแบบจำลองเพื่อนำมาใช้ในงานประมวลผลภาษาธรรมชาติด้วยคอมพิวเตอร์

จากการศึกษาพบว่าคลังข้อมูลภาษาสามารถแบ่งได้เป็น 2 ประเภทหลักๆ ด้วยกันคือ คลังคำ และคลังข้อความ โดยงานวิจัยในวิทยานิพนธ์นี้จะนำเอาคลังข้อความอักษรเบรลล์ภาษาไทยมาใช้ในการสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทยเพื่อให้เอ็นแกรมโมเดลคำนวณค่าความน่าจะเป็นของคำที่เกิดขึ้นร่วมกันตามหลักการสร้างไฟล์โมเดลภาษา (การสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทยจะอธิบายในหัวข้อที่ 3.3.2) แต่ในปัจจุบันยังไม่มีการสร้างคลังข้อความอักษรเบรลล์ภาษาไทย มีเพียงแต่คลังข้อความภาษาไทยเท่านั้น

ดังนั้นในงานวิทยานิพนธ์นี้จึงได้นำเอาคลังข้อความภาษาไทยมาประยุกต์ในการสร้างโมเดลอักษรเบรลล์ภาษาไทย โดยได้เลือกคลังข้อความ BEST [13] ซึ่งเป็นคลังข้อความภาษาไทยนำมาแปลงให้อยู่ในรูปแบบของอักษรเบรลล์ภาษาไทยระดับ 1 เสียก่อนจึงจะสามารถนำไปสร้างโมเดลอักษรเบรลล์ภาษาไทยได้

คลังข้อความ BEST เป็นคลังข้อความภาษาไทยที่มีการเก็บรวบรวมคำอย่างเป็นระบบและมีขนาดใหญ่ที่สุดในปัจจุบัน ซึ่งพัฒนาโดยหน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา (Human

Language Technology: HLT) ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (National Electronics and Computer Technology Center: NECTEC) ซึ่งมีรายละเอียดดังนี้

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติสร้างคลังข้อความภาษาไทยแบบแบ่งคำ “BEST 2009” โดยนำงานเขียนที่มีลักษณะต่างๆ 3 ประเภทมารวบรวมไว้เพื่อให้เป็นตัวแทนของภาษาไทยที่ใช้กันโดยทั่วไปในปัจจุบัน ได้แก่ ข้อความจากหนังสือประเภทนวนิยายในฐานะเป็นตัวแทนของภาษาพูดทั่วไป ข้อความจากเว็บไซต์ www.midnightuniv.org [14] และข้อความจากสารานุกรมสำหรับเยาวชนไทยในฐานะเป็นตัวแทนของภาษาเขียนอย่างเป็นทางการ และข้อความจากหนังสือพิมพ์บนอินเตอร์เน็ตในฐานะเป็นตัวแทนของภาษาข่าว ข้อความในเอกสารต่างๆ เหล่านี้ได้ถูกนำมาวิเคราะห์เพื่อกำหนดขอบเขตของคำ โดยอาศัยหลักการทางภาษาศาสตร์โดยคำนึงถึงการประมวลผลด้วยคอมพิวเตอร์ร่วมด้วย ทั้งนี้ไม่มีการแก้ไขดัดแปลงเนื้อหาสาระของผลงานต้นฉบับแต่อย่างใด ซึ่งคลังข้อความ BEST มีรายละเอียดดังต่อไปนี้

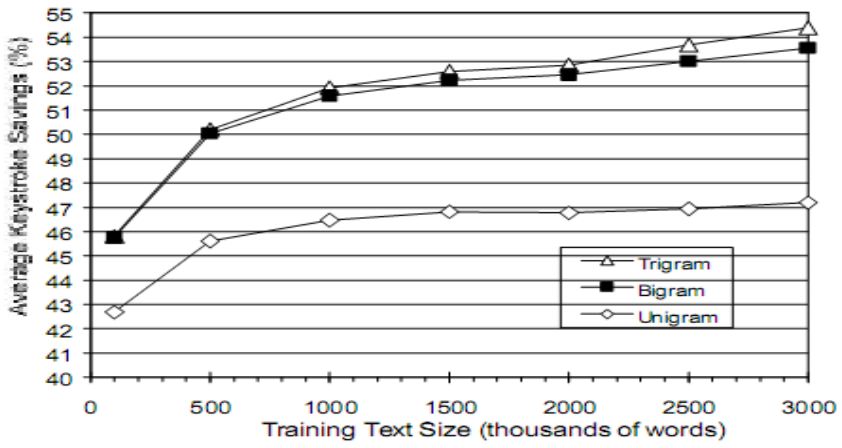
- ชื่อไฟล์ : ข้อมูลในคลังข้อความจะกำหนดขึ้นตามประเภทของภาษา ดังนี้
  - ภาษาพูดทั่วไปจะเก็บอยู่ในไฟล์ชื่อ novel-000xx.txt
  - ภาษาเขียนอย่างเป็นทางการจะเก็บอยู่ในไฟล์ชื่อ encyclopedia-000xx.txt และ miduniv-000xx.txt
  - ภาษาข่าวจะเก็บอยู่ในไฟล์ชื่อ news-000xx.txt
- ขนาด : คลังข้อความมีขนาดประมาณ 5,600,000 คำ โดยนับจากเครื่องหมายกำหนดขอบเขตของคำ “|” ซึ่งคำในที่นี้รวมถึงเครื่องหมายวรรคตอน นิพจน์ระบุนาม (Named-entity) และอักษรย่อด้วย ทั้งนี้จะแบ่งคลังข้อความเป็น 7 ชุด คือชุดฝึกฝน 6 ชุด ประมาณ 5,000,000 คำ และชุดทดสอบ 2 ชุด ประมาณ 600,000 คำ โดยแต่ละชุดมีจำนวนคำดังต่อไปนี้
  - ชุดฝึกฝนที่ 1 ประมาณ 500,000 คำ (สร้างขึ้นในเดือน กรกฎาคม 2551)
  - ชุดฝึกฝนที่ 2 ประมาณ 5000,000 คำ (สร้างขึ้นในเดือน สิงหาคม 2551)
  - ชุดฝึกฝนที่ 3 ประมาณ 1,000,000 คำ (สร้างขึ้นในเดือน กันยายน 2551)
  - ชุดฝึกฝนที่ 4 ประมาณ 1,000,000 คำ (สร้างขึ้นในเดือน ตุลาคม 2551)
  - ชุดฝึกฝนที่ 5 ประมาณ 1,000,000 คำ (สร้างขึ้นในเดือน พฤศจิกายน 2551)
  - ชุดฝึกฝนที่ 6 ประมาณ 1,000,000 คำ (สร้างขึ้นในเดือน ธันวาคม 2551)

- ชุดทดสอบชุดที่ 1 ประมาณ 100,000 คำ (สร้างขึ้นในเดือน มกราคม 2552)
- ชุดทดสอบชุดที่ 2 ประมาณ 500,000 คำ (สร้างขึ้นในเดือน กุมภาพันธ์ 2552)

- Metadata: ไฟล์ข้อมูลทั้งหมดเป็นข้อความล้วนที่ไม่มีการกำกับ metadata เพื่อความสะดวกในการใช้งาน
- ย่อหน้าและประโยค: ข้อความในไฟล์แบ่งย่อยที่ระดับย่อหน้าตามต้นฉบับเดิม โดยไม่กำกับเครื่องหมายแสดงย่อหน้าและไม่กำกับของเขตของประโยค
- รหัสอักขระ: UTF-8
- เครื่องหมายกำหนดขอบเขตคำ: | ในตำแหน่งท้ายของคำ
- เครื่องหมายกำหนดนิพจน์ระบุนาม (Named-entity) : <NE>…</NE>
- เครื่องหมายกำหนดอักษรย่อ: <AB>…</AB>
- เครื่องหมายกำหนดข้อความที่เป็นร้อยกรอง: <POEM>…</POEM>

จากข้อดีต่างๆ ของคลังข้อมูล BEST เช่น ถูกออกแบบมาให้สามารถนำไปใช้ประมวลผลด้วยคอมพิวเตอร์ได้ มีความน่าเชื่อถือในการตัดคำ ได้รับอนุญาตให้สามารถนำไปใช้งานในด้านการศึกษาวิจัยได้ จึงเป็นคลังข้อความที่เหมาะสมที่จะนำมาประยุกต์ใช้งาน ในงานวิจัยนี้

และจากการศึกษางานวิจัยการใช้จำนวนแกรมต่างๆ และขนาดของคลังข้อความในการฝึกฝนเอ็นแกรมโมเดลที่แตกต่างกันเพื่อช่วยในการคาดเดาคำศัพท์ [15] พบว่าเมื่อเพิ่มขนาดของข้อมูลที่ใช้ฝึกฝนเอ็นแกรมโมเดลจะสามารถช่วยลดการกดคีย์เพื่อพิมพ์คำแต่ละคำลงได้ ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 กราฟเปรียบเทียบจำนวนครั้งที่กดคีย์เมื่อใช้ข้อมูลและจำนวนแกรมต่างๆ [15]

Lesher และคณะ [15] ได้ทดลองเพื่อเปรียบเทียบประสิทธิภาพของจำนวนของเอ็นแกรม เมื่อใช้จำนวนแกรมเท่ากับ 1-gram, 2-gram และ 3-gram โดยในแต่ละแกรมใช้ข้อมูลขนาดต่างๆ ตั้งแต่ 100,000 – 3,000,000 คำในการฝึกฝนเอ็นแกรมโมเดล และใช้เอ็นแกรมโมเดลเหล่านี้ร่วมกันกับโปรแกรมช่วยพิมพ์เพื่อให้เอ็นแกรมโมเดลคาดเดาคำที่จะพิมพ์ โดยผลการทดสอบเปรียบเทียบพบว่าเมื่อใช้ขนาดของข้อมูลในการฝึกฝนเอ็นแกรมโมเดลเพิ่มขึ้นทำให้สามารถคาดเดาคำที่จะพิมพ์ได้ถูกต้องมากยิ่งขึ้น ซึ่งจะช่วยลดจำนวนคำที่ต้องพิมพ์ด้วยตนเองลงได้ และการใช้จำนวนแกรมเท่ากับ 3-gram จะให้ผลลัพธ์ในการคาดเดาคำได้ดีที่สุด

ดังนั้นงานวิจัยนี้จึงนำเอาคลังข้อความ BEST ซึ่งเป็นคลังข้อความที่มีขนาดใหญ่ที่สุดในปัจจุบันมาประยุกต์ใช้งาน แต่จะคัดเลือกเฉพาะประโยคที่มีการใช้คำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์เท่านั้น ทั้งนี้เพราะโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม สามารถแปลงประโยคทั่วๆ ไปได้ถูกต้องอยู่แล้วและวิทยานิพนธ์นี้ได้มุ่งเน้นที่จะแก้ไขปัญหาที่เกี่ยวข้องกับการแปลงคำควบกล้ำ, สระประสมและอักษรเบรลล์สองเซลล์เท่านั้น และในหัวข้อที่ 3.3.1 จะกล่าวถึงวิธีคัดเลือกประโยคที่ต้องการจากคลังข้อมูล BEST

2.5 เอ็นแกรมโมเดล

เอ็นแกรมโมเดล (N-gram model) [2] คือ แบบจำลองที่ใช้คำนวณค่าความน่าจะเป็นของชุดอักขระ (character sequence) ที่เกิดขึ้นร่วมกันเป็นคำ หรือค่าความน่าจะเป็นของคำที่เขียนเรียงกัน (word sequence) ที่เกิดขึ้นร่วมกันเป็นประโยค โดยค่าความน่าจะเป็นของชุดอักขระหรือคำ คำนวณได้จากคลังข้อมูลที่สร้างไว้

Gram คือ หน่วยที่ใช้ในการสร้างแบบจำลองอาจจะเป็นเสียง คำ หรืออักขระก็ได้และแกรมมีได้หลายขนาดแล้วแต่จะกำหนด ตั้งแต่ 1 จนถึง N โดย N เป็นจำนวนนับตั้งแต่ 1, 2, 3 ..., n

2.5.1 หลักการทำงานของ N-gram model

เมื่อมีกลุ่มของอักขระที่เขียนเรียงติดกัน ที่เกิดขึ้นร่วมกันเป็นคำ $S = (c_1 \ldots c_n)$ หรือกลุ่มของคำที่เขียนเรียงติดกัน ที่เกิดขึ้นร่วมกันเป็นประโยค $W = (w_1 \ldots w_n)$ ให้ N-gram model คำนวณค่าความน่าจะเป็นของการเกิดชุดของอักขระหรือคำใดๆ ที่เกิดขึ้นร่วมกัน โดยการพิจารณาจากอักขระหรือคำก่อนหน้า ตัวอย่างเช่น

การประมาณค่าด้วย 3- Gram (Probability trigram) คือ การประมาณค่าความน่าจะเป็นของชุดอักขระที่เกิดขึ้นร่วมกันเป็นคำ มีค่าเท่ากับผลคูณของความน่าจะเป็นที่จะพบอักขระ(คำ) ทีละ 3 ตัว(คำ) ติดกันในชุดอักขระนั้น เช่น ประโยค “ฉันกินข้าว” ความน่าจะเป็นที่คำว่า “ฉัน”, “กิน” และ

“ข้าว” เขียนเรียงติดกันแล้วเกิดขึ้นร่วมกันเป็นประโยค = P(ข้าว|กิน,ฉัน) หากเขียนให้อยู่ในรูปทั่วไปจะมีสมการคำนวณค่าความน่าจะเป็นดังสมการที่ 2 และ 3

ค่าความน่าจะเป็นที่จะเกิดคำหรืออักขระ x_i โดยมีชุดของคำหรือชุดอักขระ $x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1}$ นำหน้า $= P(x_i| x_{i-1},x_{i-2},x_{i-3},\ldots,x_{i-n})$ …….. (2)

ซึ่ง $P(x_i| x_{i-1},x_{i-2},x_{i-3},\ldots,x_{i-n}) = c(x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1},x_i)/c(x_{i-1},x_{i-2},x_{i-3},\ldots,x_{i-n})$ ……..(3)
โดยที่ $c(x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1},x_i)$ คือจำนวนครั้งใน Training Corpus ที่เกิด x_i ร่วมกับ $x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1}$
$c(x_{i-1},x_{i-2},x_{i-3},\ldots,x_{i-n})$ คือจำนวนครั้งที่เกิด $x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1}$ ใน Training Corpus

แต่ในความเป็นจริงแล้วเป็นไปไม่ได้ที่จะบรรจุคำทุกๆ คำหรือรูปแบบคำทุกๆ รูปแบบที่เกิดขึ้นลงในคลังข้อมูลฝึกได้ทั้งหมด

ดังนั้นจึงต้องปรับปรุงให้ค่าความน่าจะเป็นแก่คำหรือชุดของอักขระ ที่ไม่มีอยู่ในคลังข้อมูลฝึก ซึ่งเรียกการเพิ่มค่าความน่าจะเป็นให้กับคำที่ไม่พบนี้ว่าการ Smoothing [17] ทั้งนี้เพื่อป้องกันปัญหาการหารด้วยศูนย์ แม้ว่าในความเป็นจริงแล้วจะไม่มีคำนั้นๆ อยู่ก็ตาม

งานวิจัยในวิทยานิพนธ์นี้จะนำแบบจำลองภาษาอักษรเบรลล์ภาษาไทยเข้ามาช่วยในการตัดสินเลือกคำที่ได้จากการแปลงอักษรเบรลล์ไปเป็นภาษาไทย โดยใช้วิธีการคำนวณหาค่าความน่าจะเป็นของคำที่เกิดขึ้นร่วมกัน และเนื่องจากหลักไวยากรณ์ของอักษรเบรลล์ จะไม่เหมือนกับหลักไวยากรณ์ของภาษาไทยจึงไม่สามารถนำเอาแบบจำลองเอ็นแกรมของภาษาไทยมาใช้ได้ จึงจำเป็นต้องสร้างแบบจำลองเอ็นแกรมของอักษรเบรลล์ภาษาไทยขึ้นมาใหม่

2.6 Smoothing Techniques

Smoothing techniques [17] คือ เทคนิคการประมาณค่าความน่าจะเป็นให้กับคำหรือชุดของคำที่ไม่มีอยู่หรือไม่พบในคลังข้อความ ซึ่งเทคนิคนี้อาศัยวิธีการประมาณค่าทางสถิติในการคำนวณหาค่าความน่าจะเป็นของคำที่ไม่พบในคลังข้อความ และเทคนิคนี้เป็นกระบวนการที่จำเป็นสำหรับงานด้านสถิติ อีกทั้งเป็นการป้องกันปัญหาการหารด้วยศูนย์อีกด้วย นอกจากนี้แล้วจากข้อจำกัดของคลังข้อความที่ไม่สามารถจัดเก็บคำและคู่ของคำทั้งหมดที่เกิดขึ้นจริงไว้ในคลังข้อมูลได้ทั้งหมดทำให้คำบางคำหรือคู่ของคำบางคำอาจไม่มีอยู่ในคลังข้อมูล โดยจะขอยกตัวอย่างสมการการคำนวณค่าความน่าจะเป็นของคำเมื่อใช้จำนวนแกรมเท่ากับ 2 (bigram) ดังสมการที่ 4

$$P(w_2|w_1) = C(w_1w_2)/C(w_1) \text{ ……..…… (4)}$$

จากสมการที่ 4 หมายความว่า ค่าความน่าจะเป็นที่คำ w_2 เกิดขึ้นก็ต่อเมื่อมีคำ w_1 เขียนนำหน้า เท่ากับจำนวนคำของ w_1w_2 ซึ่งเขียนเรียงติดกันที่เกิดขึ้นร่วมกันในคลังข้อมูลฝึก ($C(w_1w_2)$) หารด้วยจำนวนคำ w_1 ที่พบในคลังข้อมูลฝึก ($C(w_1)$)

เมื่อพิจารณาสมการที่ 4 แล้วพบว่าหากคำ w_1 ไม่พบในคลังข้อมูลจะทำให้ค่าของจำนวนคำ w_1 เท่ากับศูนย์ ($C(w_1) = 0$) และเมื่อคำนวณค่าความน่าจะเป็นที่คำ w_2 เกิดขึ้นก็ต่อเมื่อมีคำ w_1 เขียนนำหน้า ($P(w_2|w_1)$) จะทำให้ไม่สามารถคำนวณค่าความน่าจะเป็นได้ ทำให้เกิดปัญหาการหารด้วยศูนย์ [22] หรืออีกกรณีหนึ่งคือคู่ของคำ w_1w_2 ไม่พบในคลังข้อมูลทำให้ค่าของจำนวนคำของ w_1w_2 ซึ่งเขียนเรียงติดกันเท่ากับศูนย์ ($C(w_1w_2) = 0$) และเมื่อคำนวณค่าความน่าจะเป็นที่คำ w_2 เกิดขึ้นก็ต่อเมื่อมีคำ w_1 เขียนนำหน้า ($P(w_2|w_1)$) จะทำให้ได้ค่าความน่าจะเป็นเท่ากับศูนย์ ซึ่งในความเป็นจริงแล้วอาจจะเป็นไปได้ว่ามีโอกาสที่คำ w_1 และ w_2 เกิดขึ้นร่วมกัน ซึ่งกรณีนี้จะให้ผลลัพธ์ในการคำนวณที่ผิดพลาดออกมา

ดังนั้นเพื่อป้องกันปัญหาดังกล่าวจึงได้มีการนำเสนอเทคนิควิธีการที่เรียกว่า smoothing เข้ามาเพื่อป้องกันปัญหาดังกล่าว และจากการศึกษาพบว่าวิธีการ smoothing มีอยู่ด้วยกันหลายวิธี [17] เช่น Baseline Smoothing, Additive Smoothing, Katz Smoothing, Church-Gale Smoothing, Jelinek-Mercer Smoothing และ Novel Smoothing

จากการศึกษาพบว่าการประมาณค่าด้วยวิธี Jelinek-Mercer Smoothing (Interpolation) จะให้ผลลัพธ์ดีที่สุดเมื่อเปรียบเทียบกับวิธีอื่นๆ [17] ซึ่งการทดสอบเปรียบเทียบได้สร้างเอ็นแกรมโมเดล โดยใช้จำนวนแกรมเท่ากับสองแกรม (Bigram) และสามแกรม (Trigram) และใช้คลังข้อมูล TIPSTER, Brown และ Wall Street Journal ในการฝึกฝนเอ็นแกรมโมเดลและแบ่งการฝึกฝนเอ็นแกรมโมเดลออกเป็นสองชุดด้วยกันคือ ชุดแรกฝึกฝนเอ็นแกรมโมเดลด้วยข้อมูลขนาดประมาณ 50,000 ประโยค ชุดที่สองฝึกฝนเอ็นแกรมโมเดลด้วยข้อมูลขนาดประมาณ 10,000,000 ประโยค และวัดค่าความน่าจะเป็นเมื่อใช้เทคนิคการ Smoothing แบบต่างๆ โดยเปรียบเทียบความเบี่ยงเบนจากเส้นฐาน (Difference in test cross-entropy from baseline)

จากผลการทดลองดังกล่าวสามารถสรุปได้ว่าเทคนิคการ Smoothing แบบ Jelinek-Mercer Smoothing (interpolation) จะให้ผลลัพธ์ที่ดีที่สุดเพราะมีความเบี่ยงเบนจากเส้นฐานน้อยที่สุด ซึ่งหมายถึงเทคนิคนี้สามารถประมาณค่าความน่าจะเป็นได้ใกล้เคียงกับค่าความน่าจะเป็นที่คำนวณได้จริงๆ ตามสมการที่ 2 โดยวิธี Jelinek-Mercer Smoothing (Interpolation) มีสมการในการประมาณค่าความน่าจะเป็นดังแสดงในสมการที่ 5

$$P_{interp}\left(w_i \mid w_{i-n+1}^{i-1}\right) = \lambda_{w_{i-n+1}^{i-1}} P_{ML}\left(w_i \mid w_{i-n+1}^{i-1}\right) + \left(1-\lambda_{w_{i-n+1}^{i-1}}\right) P_{interp}\left(w_i \mid w_{i-n+2}^{i-1}\right) \text{.....(5)}$$

โดยที่ $P_{ML}\left(w_i \mid w_{i-n+1}^{i-1}\right) = C(w_{i-n+1}^{i-1}\ w_i) / C(w_{i-n+1}^{i-1})$

และจากงานวิจัยของ Bahl และคณะ [17] แนะนำให้กำหนดค่าของ $\lambda_{w_{i-n+1}^{i-1}} = \frac{C\left(w_{i-n+1}^{i-1}\right)}{|W_i : C\left(w_{i-n+1}^{i-1}\right) > 0|}$

ดังนั้นในงานวิจัยที่นำเสนอนี้ได้เลือกใช้เทคนิคการประมาณค่าแบบ Jelinek-Mercer Smoothing (Interpolation) เพื่อใช้คำนวณค่าความน่าจะเป็นให้กับคำหรือคู่ของคำที่เกิดขึ้นร่วมกัน (อยู่ในรูปแบบของอักษรเบรลล์ภาษาไทยระดับ 1) ซึ่งไม่ได้ปรากฏอยู่ในคลังข้อมูลและนอกจากนี้แล้วชุดเครื่องมือ CMU-SLM toolkit [21] ที่นำมาใช้ในงานวิจัยนี้มีความสามารถในการคำนวณค่าความน่าจะเป็นโดยใช้เทคนิคการ Smoothing แบบ Jelinek-Mercer Smoothing (Interpolation) อีกด้วย

2.7 CMU-SLM Toolkit

ชุดเครื่องมือ CMU-SLM Toolkit [20] เป็นชุดเครื่องมือที่ใช้ในการสร้างโมเดลภาษา (Language model) หรือเรียกว่า “การฝึกฝนเอ็นแกรมโมเดล” โดยชุดเครื่องมือนี้ได้รับการพัฒนาโดยมหาวิทยาลัย Carnegie Mellon โดยในงานวิจัยที่นำเสนอนี้ได้เลือกใช้เครื่องมือ CMU-SLM Toolkit เพื่อช่วยอำนวยความสะดวกในการสร้างโมเดลภาษาของอักษรเบรลล์ภาษาไทยระดับ 1 เนื่องจากมีการใช้งานกันอย่างแพร่หลาย ชุดเครื่องมือนี้มีการเผยแพร่เป็นสาธารณะ มีคู่มืออธิบายการใช้งานชัดเจน ทำให้ใช้งานได้ง่าย และเป็นชุดเครื่องมือที่ครบถ้วนสมบูรณ์ โดยในชุดเครื่องมือนี้จะประกอบด้วยโปรแกรมย่อยจำนวนหลายโปรแกรมด้วยกันเช่น text2wfreq, wfreq2vocab, text2wngram, text2idngram, ngram2mgram, wngram2idngram, idngram2stats, mergeidngram, idngram2lm, binlm2arpa, evallm, interpolate เป็นต้น

สำหรับในวิทยานิพนธ์นี้ได้ใช้เฉพาะโปรแกรมย่อยบางตัวเท่านั้นในการสร้างโมเดลภาษา ซึ่งมีดังต่อไปนี้

1. โปรแกรม text2wfreq จะทำหน้าที่สร้างรายการคำที่มีอยู่ในไฟล์ที่ผู้ใช้ได้นำมาฝึกฝนเอ็นแกรมโมเดลทั้งหมด พร้อมทั้งนับความถี่ของแต่ละคำ
2. โปรแกรม text2wfreq จะทำหน้าที่สร้างรายการของคำ (Vocabulary) ทั้งหมดที่มีอยู่ในไฟล์ที่ผู้ใช้นำมาฝึกฝนเอ็นแกรมโมเดล

3. โปรแกรม text2idngram ทำหน้าที่สร้างไบนารีไฟล์ที่เก็บว่ามีคำในไฟล์ที่ผู้ใช้นำมาฝึกฝนเอ็นแกรมโมเดล คำๆ นั้นจะสามารถเกิดขึ้นร่วมกับคำใดได้บ้างและไม่สามารถเกิดร่วมกับคำใดได้บ้างโดยเก็บอยู่ในลักษณะเป็นดัชนี (index) ว่าเกิดขึ้นร่วมกับคำอื่นได้หรือไม่ โดยแทนค่าด้วย 1 หากสามารถเกิดขึ้นร่วมกันได้และในกรณีที่ไม่สามารถเกิดขึ้นร่วมกันได้จะแทนค่าด้วย 0
4. โปรแกรม idngram2lm ทำหน้าที่สร้างไฟล์โมเดลภาษา ซึ่งเป็นไฟล์ที่จะนำไปใช้งานต่อไป โดยตัวอย่างไฟล์โมเดลภาษาจะแสดงไว้ในรูปที่ 3.6 ในหัวข้อ 3.3.1 วิธีการฝึกฝนเอ็นแกรมโมเดล

เมื่อใช้โปรแกรมย่อยเหล่านี้แล้วก็จะได้ผลลัพธ์ออกมาเป็นไฟล์โมเดลภาษา ซึ่งขึ้นอยู่กับข้อความที่ผู้ใช้ในนำมาใช้ในการสร้างโมเดลภาษาว่าเป็นภาษาอะไร จากนั้นผู้วิจัยจะนำเอาไฟล์โมเดลภาษานี้มาใช้กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำตามที่ได้ออกแบบไว้ ซึ่งจะอธิบายใน บทที่ 3 การออกแบบและพัฒนาระบบ

2.8 โปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์ ของมหาวิทยาลัยสงขลานครินทร์

โปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์ (Thai2brl) [1] ได้นำเสนอไว้ในงานวิจัยเรื่อง “การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด” ของมหาวิทยาลัยสงขลานครินทร์ เช่นเดียวกันกับโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file ที่ได้นำมาเปรียบเทียบประสิทธิภาพกับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำซึ่งได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ โดยโปรแกรมดังกล่าวทำหน้าที่แปลงภาษาไทยไปเป็นอักษรเบรลล์ภาษาไทยในรูปแบบ File-to-file ซึ่งกระบวนการทำงานของโปรแกรมนี้ ได้ใช้วิธีการแปลงแบบใช้กฎไวยากรณ์ทางภาษาของภาษาไทยและอักษรเบรลล์ภาษาไทยเข้ามาควบคุมการแปลง โดยให้ผลลัพธ์ในการแปลงที่ถูกต้องในระดับหนึ่ง

ซึ่งในงานวิจัยนี้ได้นำโปรแกรมดังกล่าวมาใช้เพื่อแปลงข้อความภาษาไทยให้เป็นข้อความอักษรเบรลล์ภาษาไทยระดับ 1 เพื่อนำไปใช้งานต่อไป โดยได้นำมาใช้ในขั้นตอนต่างๆ ดังนี้

1. ใช้แปลงประโยคภาษาไทยที่ได้คัดเลือกมาจากคลังข้อความ BEST ให้เป็นประโยคอักษรเบรลล์ภาษาไทยระดับ 1 เพื่อนำไปสร้างโมเดลอักษรเบรลล์ภาษาไทยต่อไป

(รายละเอียดการคัดเลือกประโยคที่ต้องการจากคลังข้อความ BEST จะอธิบายในหัวข้อ 3.3.1 และวิธีการสร้างโมเดลอักษรเบรลล์ภาษาไทยจะอธิบายในหัวข้อ 3.3.2)

2. ใช้แปลงชุดข้อมูลทดสอบให้อยู่ในรูปของอักษรเบรลล์ภาษาไทยระดับ 1 เพื่อนำไปทดสอบเปรียบเทียบประสิทธิภาพในด้านต่างๆ ระหว่างโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file (วิธีการเดิม) กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ (วิธีการใหม่) ซึ่งได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ (รายระเอียดของชุดข้อมูลทดสอบจะอธิบายในหัวข้อ 4.1.3, 4.2.3 และ 4.3.3)
3. ใช้แปลงรายการคำภาษาไทยให้อยู่ในรูปแบบของอักษรเบรลล์ภาษาไทยระดับ 1 เพื่อนำไปใช้ในขั้นตอน Translate_to_thai เนื่องจากเอ็นแกรมโมเดลไม่สามารถนำไปประยุกต์ด้านการแปลงให้เป็นภาษาไทยได้ จึงต้องมีขั้นตอนดังกล่าวเพื่อทำหน้าที่แปลงประโยคอักษรเบรลล์ภาษาไทยที่ตัดคำแล้วโดยใช้เทคนิคเอ็นแกรมให้อยู่ในรูปของประโยคภาษาไทย และโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file (วิธีการเดิม) ให้เอาท์พุทเป็นประโยคภาษาไทยจึงต้องมีขั้นตอนดังกล่าวเพื่อแปลงให้เอาท์พุทเป็นประโยคภาษาไทยเช่นกัน เพื่อให้สามารถเปรียบเทียบกันได้โดยตรง (รายละเอียดของขั้นตอน Translate_to_thai จะอธิบายในหัวข้อ 3.6.4 และรายการคำภาษาไทยและรายการคำอักษรเบรลล์ภาษาไทยจะอธิบายในหัวข้อ 3.9)

แต่ทั้งนี้เนื่องจากโปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์ดังกล่าว ยังคงมีการแปลงที่ผิดพลาดอยู่บ้างเนื่องจากการใช้กฎไวยากรณ์ทางภาษาของภาษาไทยและอักษรเบรลล์ภาษาไทยเข้ามาควบคุมการแปลง ซึ่งผู้วิจัยได้พยายามค้นหาคำที่แปลงผิดและแก้ไขให้ถูกต้องให้มากที่สุดเท่าที่จะเป็นไปได้ โดยได้รวบรวมรายการคำดังกล่าวไว้ในตารางที่ 8 ในภาคผนวก ค

บทที่ 3

การออกแบบและการพัฒนาระบบ

ในบทนี้จะกล่าวถึงรายละเอียดของปัญหาการแปลงอักษรเบรลล์ภาษาไทยระดับ 1 ไปเป็นภาษาไทย, ขั้นตอนการสร้างไฟล์โมเดลภาษาและเครื่องมือที่ใช้ เพื่อนำไปใช้งานร่วมกับโปรแกรมที่ได้นำเสนอและหลักการทำงานของโปรแกรมที่ได้นำเสนอในวิทยานิพนธ์นี้ ซึ่งประกอบไปด้วย 1) การวิเคราะห์ปัญหาการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยในปัจจุบัน 2) แนวคิดในการแก้ไขปัญหาในงานวิจัย 3) การสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทย 4) ภาพรวมของโปรแกรมที่นำเสนอ 5) อินพุตและเอาท์พุทของโปรแกรมที่นำเสนอ 6) โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำซึ่งได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ 7) เอ็นแกรมโมเดล 8) Smoothing Techniques และ 9) รายการคำอักษรเบรลล์ภาษาไทยและรายคำภาษาไทย

3.1 การวิเคราะห์ปัญหาการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยในปัจจุบัน

ในการเขียนอักษรเบรลล์ภาษาไทยระดับ 1 นั้นมีความคล้ายคลึงกับการเขียนภาษาไทยอยู่มาก เช่น การเขียนคำแต่ละคำติดๆ กันเป็นประโยคและการเว้นวรรคประโยคเป็นช่วงๆ และไม่มีการแบ่งขอบเขตของคำแต่ละคำอย่างชัดเจนเหมือนกับภาษาอังกฤษ แต่ก็มีความแตกต่างกันในด้านของหลักไวยากรณ์อยู่พอสมควรและไม่สามารถแปลโดยเทียบตัวอักษรแบบตัวต่อตัวได้

ดังนั้นจึงจำเป็นต้องอาศัยซอฟต์แวร์แปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยเข้ามาช่วยเพื่อเป็นตัวกลางในการสื่อสารกันระหว่างผู้พิการทางสายตากับคนปกติ ซึ่งซอฟต์แวร์สำหรับการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ได้มีการพัฒนามาก่อนหน้านี้ ได้ใช้หลักการแปลงที่แตกต่างกัน เช่น การแปลงโดยอาศัยการเปรียบเทียบแบบคำต่อคำ การแปลงโดยอาศัยกฎไวยากรณ์ทางภาษา การแปลงโดยอาศัยการเทียบแบบตัวอักษรเบรลล์กับอักษรไทยแบบต่อตัวตามรูปแบบการเขียนภาษาไทยโดยทั่วไปที่เรียกอย่างไม่เป็นทางการว่า “การเขียนอักษรเบรลล์ภาษาไทยระดับ 0” ซึ่งไม่ใช่มาตรฐานในการสื่อสารโดยปกติระหว่างกันของผู้พิการทางสายตา แต่เพื่อให้สื่อสารกับคนสายตาปกติได้

ในการพัฒนาโปรแกรมแปลงอักษรเบรลล์มักใช้การแปลงโดยอาศัยกฎไวยากรณ์ทางภาษาซึ่งได้รับความนิยมนำมาใช้มากที่สุด เนื่องจากสามารถใช้กฎไวยากรณ์ทางภาษาเป็นเงื่อนไขในการ

เขียนโปรแกรมได้เลยและไม่จำเป็นต้องใช้องค์ประกอบอื่นๆ เช่น ฐานข้อมูลคำ คลังข้อมูลคำ หรือเครื่องมือพิเศษอื่นๆ

แต่การแปลงโดยใช้กฎไวยากรณ์ทางภาษานั้นมีความซับซ้อนในการเขียนเงื่อนไขการแปลงให้ครอบคลุมกรณีการแปลงทั้งหมดและมักจะมีปัญหาที่เกิดขึ้นจากการต่อคำ เช่น คำว่า “เครือ” เขียนอยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 เป็น “urq” ซึ่งก็คือ “ค+ร+เอือ” แต่เนื่องจากคำว่า “rq” หรือ “ร+เอือ” เป็นคำที่มีความหมายว่า “เรือ” ก็เป็นคำที่มีความหมายในภาษาไทยเช่นกัน และอักษรเบรลล์ ‘u’ ซึ่งก็คืออักษร “ค” ในภาษาไทยของว่า “เครือ” อาจจะเป็นตัวควบกล้ำของคำปัจจุบันหรือเป็นตัวสะกดของคำก่อนหน้าก็ได้ เช่นวลีที่ว่า “ไททานิคเรือที่ไม่มีวันจม” (อักษร “ค” เป็นตัวสะกดของคำก่อนหน้า) หรือประโยค “กล้วยเครือนี้ผลมีขนาดใหญ่” (อักษร “ค” เป็นตัวควบกล้ำของคำปัจจุบัน) ซึ่งปัญหาการต่อคำเหล่านี้โดยทั่วไปแล้วจะเกิดขึ้นกับเฉพาะคำควบกล้ำและคำที่มีการใช้สระประสม เพราะอักษรนำของคำๆ นั้นอาจจะเป็นตัวควบกล้ำของคำปัจจุบันหรือเป็นตัวสะกดของคำก่อนหน้าก็ได้ ซึ่งปัจจุบันปัญหานี้ยังคงเป็นสาเหตุหลักที่โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่มีการพัฒนามาก่อนหน้านี้ ยังคงมีการแปลงที่ผิดพลาดอยู่

ปัญหาการแปลงอีกประการหนึ่งที่สำคัญคือ การแปลงคำที่มีการใช้อักษรเบรลล์สองเซลล์ ซึ่งปัญหานี้เกิดจากการที่มีอักษรเบรลล์สองเซลล์บางตัวที่มีรูปพ้องกันกับการเขียนอักษรเบรลล์เซลล์เดียวสองตัวติดกัน โดยอักษรเบรลล์สองเซลล์ที่มีปัญหาในการแปลงดังกล่าวแสดงดังตารางที่ 3.1

ตารางที่ 3.1 อักษรเบรลล์สองเซลล์ที่มีรูปพ้องกันกับอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกัน

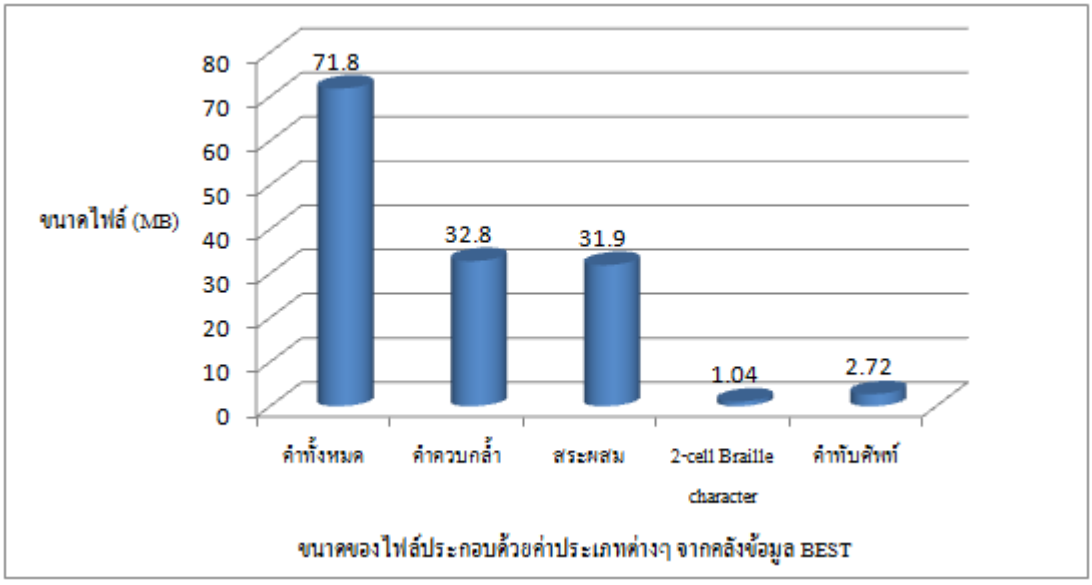
อักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกัน	อักษรเบรลล์สองเซลล์
r+1 (ร+ๆ)	r1 (ฤ)
l+1 (ล+ๆ)	l1 (ฦ)
o+a (อ+สระอะ)	oa (สระเอาะ)
0) (การันต์+ท)	0) (ธ)

จากตารางที่ 3.1 อักษรเบรลล์ ‘r1’ ซึ่งก็คืออักษร ‘ฤ’ จะพ้องรูปกับอักษรเบรลล์ ‘r’ (ร) ตามด้วย ‘1’ (ๆ) และอักษรเบรลล์ ‘l1’ ซึ่งก็คืออักษร ‘ฦ’ จะพ้องรูปกับอักษรเบรลล์ ‘l’ (ล) ตามด้วย ‘1’ (ๆ) และอักษรเบรลล์ ‘oa’ ซึ่งก็คือ สระเอาะ จะพ้องรูปกับอักษรเบรลล์ ‘o’ (อ) ตามด้วย ‘a’

(สระอะ) และอักษรเบรลล์ ‘0)’ ซึ่งก็คืออักษร ‘ธ’ จะพ้องรูปกับอักษรเบรลล์ ‘0’ (การันต์) ตามด้วย ‘)’ (ท)

โดยส่วนมากแล้วจะพบกับปัญหาการแปลงอักษรเบรลล์ ‘0’ (การันต์) ตามด้วย ‘)’ (ท) ที่พ้องรูปกันกับอักษรเบรลล์ ‘0)’ (ธ) มากที่สุดเพราะเป็นตัวอักษรที่มีการใช้งานในภาษาไทยหรืออักษรเบรลล์ภาษาไทยมากกว่าตัวอักษรอื่นๆ เช่นคำว่า “พระองค์ทรง” เขียนในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 คือ “?rao}u0)r}” ซึ่งก็คือ “พ+ร+ะ+อ+ง+ค+การันต์+ท+ร+ง” โดยอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกันแล้วมีรูปพ้องกันกับอักษรเบรลล์สองเซลล์ในที่นี้คือ ‘0+)’ ซึ่งก็คืออักษร ‘การันต์+ท’ โดยมีรูปพ้องกันกับอักษรเบรลล์สองเซลล์คืออักษร ‘0)’ ซึ่งก็คืออักษร ‘ธ’ ด้วยเหตุนี้ทำให้โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม [1] ไม่สามารถทราบได้ว่าเป็นอักษรเบรลล์สองเซลล์หนึ่งตัวหรือเป็นอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกัน ซึ่งจะแตกต่างกับกรณีของโปรแกรมการแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด [1] ที่สามารถแก้ไขปัญหาเหล่านี้ได้โดยใช้วิธีให้ผู้ใช้กดคีย์พิเศษก่อนจะป้อนอักษรเบรลล์สองเซลล์ ซึ่งการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยแบบ File-to-file นั้นผู้ใช้จะไม่สามารถโต้ตอบกับโปรแกรมเพื่อระบุว่าตรงไหนเป็นอักษรเบรลล์สองเซลล์ได้ แม้ว่าปัญหาดังกล่าวอาจสามารถแก้ไขได้ด้วยการแบ่งคำที่เหมาะสมก็ตาม แต่ก็ยังมีคำบางคำที่ไม่สามารถแบ่งได้ เนื่องจากเป็นคำที่เป็นชื่อเฉพาะไม่สามารถแบ่งออกเป็นคำย่อยๆ ได้และจะทำให้ความหมายโดยรวมนั้นผิดไปจากเดิม เช่น คำว่า “นวลจันทร์ทะเล”, “พิมพ์ทอง” เป็นต้น

จากการศึกษาผู้วิจัยได้คัดเลือกประโยคที่ประกอบไปด้วยคำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์จากคลังข้อมูล BEST [13] ขนาด 71.8 เมกกะไบต์ (รายละเอียดของคลังข้อมูล BEST และวิธีการคัดเลือกประโยคจากคลังข้อมูล BEST จะอธิบายในหัวข้อที่ 2.4 และ 3.3.1 ตามลำดับ) พบว่าประโยคภาษาไทยที่มีการใช้คำควบกล้ำและสระประสมมีจำนวนประมาณเกือบครึ่งหนึ่งของจำนวนประโยคทั้งหมด และประโยคที่มีการใช้คำอักษรเบรลล์สองเซลล์และคำทับศัพท์มีขนาดประมาณ 1.04 เมกกะไบต์และ 2.72 เมกกะไบต์ตามลำดับดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ขนาดของไฟล์ที่มีการใช้คำประเภทต่างๆ เปรียบเทียบกับขนาดของไฟล์ทั้งหมด

จากรูปที่ 3.1 เห็นได้ว่าในภาษาไทยประกอบด้วยคำควบกล้ำและสระประสมประมาณครึ่งหนึ่งของประโยคทั้งหมด ดังนั้นจึงส่งผลให้การเขียนอักษรเบรลล์ภาษาไทยประกอบด้วยคำประเภทคำควบกล้ำและสระประสมเป็นจำนวนมากเช่นกัน ดังนั้นปัญหาการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ได้กล่าวไว้ข้างต้นจึงนับว่าเป็นปัญหาที่สำคัญมากทีเดียว และจำเป็นต้องได้รับการปรับปรุงแก้ไข

3.2 แนวคิดในการแก้ไขปัญหาในงานวิจัย

จากปัญหาในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ได้กล่าวไว้ในข้างต้น จึงมีแนวคิดที่จะนำเทคนิคเอ็นแกรมโมเดลเข้ามาช่วยในการตัดคำอักษรเบรลล์ภาษาไทยก่อนที่จะแปลงไปเป็นภาษาไทย เพื่อแบ่งขอบเขตแต่ละคำของอักษรเบรลล์ภาษาไทยให้ชัดเจน ทั้งนี้เพื่อเป็นการลดความกำกวมที่จะเกิดขึ้นกับการต่อคำ เช่นเดียวกันกับในกรณีของภาษาไทย

ซึ่งในงานวิจัยที่นำเสนอนี้ได้ใช้เอ็นแกรมโมเดลในการคำนวณหาค่าความน่าจะเป็นที่คำปัจจุบัน ซึ่งอยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 จะเกิดขึ้นได้ก็ต่อเมื่อมีคำบางคำที่เป็นรูปแบบเฉพาะ ซึ่งเขียนอยู่ก่อนหน้าคำๆ นี้ ยกตัวอย่างคำในรูปแบบภาษาไทยเพื่อง่ายต่อการทำความเข้าใจ เช่น คำปัจจุบันเป็นคำว่า “เรียน” คำที่เขียนอยู่ก่อนหน้าคำๆ นี้ก็จะมีความเป็นไปได้คือคำว่า “โรง” ตามด้วยคำว่า “เรียน” รวมกันเป็นคำว่า “โรงเรียน”, “นัก” ตามด้วยคำว่า “เรียน” รวมกันเป็นคำว่า “นักเรียน”, “การ” ตามด้วยคำว่า “เรียน” รวมกันเป็นคำว่า “การเรียน” เป็นต้น หรือในทางกลับกัน เมื่อมีส่วนของประโยค เช่น “ฉันไป” ซึ่งคำส่วนใหญ่ที่ตามหลังส่วนของประโยคดังกล่าวก็จะเป็นคำจำพวกสถานที่ เช่น คำว่า “โรงเรียน”, “โรงงาน”, “บ้าน” เป็นต้น ซึ่งมี

ความเป็นไปได้น้อยที่จะเป็นคำประเภทอื่นๆ เช่น คำว่า “กินข้าว”, “เที่ยว” เป็นต้น ซึ่งคำเหล่านี้ก็สามารถพบได้บ้าง แต่จะไม่มีโอกาสพบว่าเป็นคำที่ให้ความหมายขัดแย้งหรือไม่เกี่ยวข้องกับส่วนของประโยคนั้น เช่นคำว่า “เงิน”, “เขี่ย”, “แดง” เป็นต้น

ดังนั้นจึงได้นำเอาหลักการนี้มาประยุกต์ใช้กับอักษรเบรลล์ภาษาไทยระดับ 1 เพราะมีลักษณะบางประการที่เหมือนกับภาษาไทย คือมีการเขียนเป็นคำๆ เรียงติดกันเป็นประโยคและมีการเว้นวรรคระหว่างประโยค เช่นเดียวกับภาษาไทย แต่จะมีความแตกต่างในเรื่องของลำดับการวางอักขระให้เป็นคำขึ้นมาและมีการใช้สัญลักษณ์แทนตัวอักขระที่ไม่เหมือนกับภาษาไทย แต่ก็สามารถใช้หลักการของเอ็นแกรมโมเดลได้เช่นเดียวกับภาษาไทย ซึ่งงานวิจัยที่ได้นำเสนอนี้จะนำไฟล์โมเดลอักษรเบรลล์ภาษาไทยมาเปรียบเทียบเพื่อหาคำถัดไป เช่น พิจารณาจากคำที่เขียนนำหน้าก่อนหน้าคำปัจจุบันจำนวนสองคำ เมื่อใช้จำนวนแกรมเท่ากับ 3-gram เพื่อเปรียบเทียบว่าคำที่ตามหลังนั้นเป็นคำใดได้บ้างและมีค่าความน่าจะเป็นมากน้อยแค่ไหน และมีคำนี้อยู่ในประโยคอินพุตหรือไม่ หากมีจะตัดคำๆ นี้แยกออกมาโดยใช้สัญลักษณ์ <pipe> (ใช้สัญลักษณ์ “<pipe>”ในการแบ่งคำเพื่อไม่ให้ซ้ำกับสัญลักษณ์ “|” ในอักษรเบรลล์ภาษาไทยซึ่งใช้แทนตัวอักษร “ต” ในภาษาไทย) ในการแบ่งขอบเขตของคำแต่ละคำและดำเนินการแบบนี้ไปเรื่อยๆ จนกระทั่งจบประโยคหนึ่งๆ

สำหรับวิธีการคำนวณค่าความน่าจะเป็นจะเป็นไปตามหลักของการสร้างเอ็นแกรมโมเดลซึ่งได้อธิบายไว้ในหัวข้อ 3.3.2 การสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทย โดยในวิทยานิพนธ์นี้ได้เลือกใช้ชุดเครื่องมือ CMU-SLM Toolkit ในการสร้างไฟล์โมเดลภาษา

3.3 การคัดเลือกประโยคจากคลังข้อมูล BEST และการสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทย

การสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทยนี้ เป็นการสร้างแบบจำลองทางภาษาของอักษรเบรลล์ภาษาไทยระดับ 1 เพื่อนำไปใช้งานร่วมกับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ โดยการสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทยหรือการฝึกฝนเอ็นแกรมโมเดล เป็นกระบวนการทางสถิติเพื่อให้เอ็นแกรมโมเดลคำนวณค่าความน่าจะเป็นของคำหนึ่งๆ ว่าสามารถที่จะเกิดขึ้นร่วมกับคำใดได้บ้างด้วยค่าความน่าจะเป็นเท่าใด ซึ่งผู้วิจัยได้แบ่งขั้นตอนการสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทยออกเป็น 2 ขั้นตอนหลัก คือ 1) ขั้นตอนคัดเลือกประโยคที่ต้องการจากคลังข้อมูล BEST ทั้งนี้เพราะโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม [1] สามารถแปลงคำทั่วๆ ไปได้ถูกต้องอยู่แล้วจึงไม่มีความจำเป็นที่ต้องนำคลังข้อมูล BEST ทั้งหมดมาใช้สร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทย 2) ขั้นตอนการสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทย ในขั้นตอนนี้

เป็นการสร้างไฟล์แบบจำลองภาษาอักษรเบรลล์ภาษาไทยระดับ 1 โดยใช้ชุดเครื่องมือ CMU-SLM toolkit (รายละเอียดของชุดเครื่องมือ CMU-SLM toolkit จะอธิบายในหัวข้อ 2.7) ซึ่งมีรายละเอียดของขั้นตอนดังกล่าวดังต่อไปนี้

3.3.1 การคัดเลือกประโยคจากคลังข้อมูล BEST

ในงานวิจัยที่ได้นำเสนอนี้ได้เลือกใช้คลังข้อความ BEST ซึ่งเป็นคลังข้อความภาษาไทยที่มีขนาดใหญ่ที่สุดในปัจจุบันนำมาประยุกต์ใช้งาน และงานวิจัยนี้ได้มุ่งเน้นการแก้ไขปัญหาเฉพาะการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทย ที่เกี่ยวข้องกับคำควบกล้ำ, สระประสมและอักษรเบรลล์สองเซลล์ของโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิมโดยได้นำเสนอไว้ในวิทยานิพนธ์เรื่อง “การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด” [1] และโปรแกรมดังกล่าวสามารถแปลงประโยคโดยทั่วไปได้ถูกต้องอยู่แล้ว

ดังนั้นจึงไม่มีความจำเป็นที่จะต้องนำเอาคลังข้อความ BEST ทั้งหมดมาใช้สร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทย ผู้วิจัยจึงได้คัดเลือกเอาเฉพาะประโยคที่มีการใช้คำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์ และคำทับศัพท์ จากคลังข้อความ BEST เท่านั้น ซึ่งการคัดเลือกประเภทของประโยคดังกล่าวจะประกอบไปด้วย 4 ประเภทดังนี้

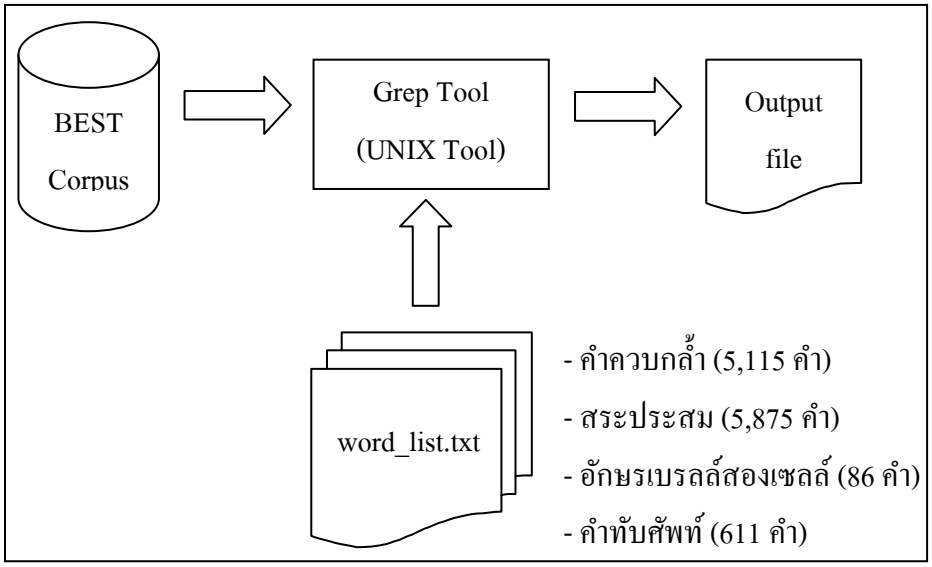
1) ประโยคที่มีการใช้คำควบกล้ำแบ่งออกเป็น 2 ประเภทด้วยกันคือ
    1. ประโยคที่ประกอบด้วยคำควบกล้ำแท้ ซึ่งจะประกอบด้วยตัวอักษรควบกล้ำดังนี้คือ กร, กล, กว, ขร, ขล, ขว, คร, คล, คว, ปร, ปล, พร และ พล
    2. ประโยคที่ประกอบด้วยคำควบกล้ำเทียม ซึ่งจะประกอบด้วยตัวอักษรควบกล้ำดังนี้คือ จร, ซร, ตร, ทร, ศร และ สร
2) ประโยคที่ประกอบด้วยสระประสม ประกอบไปด้วยสระประสมดังต่อไปนี้คือ สระอัว , สระอัวะ, สระเออ, สระเออะ,สระเอะ, สระเอา, สระเอาะ, สระเอีย, สระเอียะ, สระเอือ , สระแอะ และสระโอะ
3) ประโยคที่ประกอบด้วยอักษรเบรลล์สองเซลล์ จะคัดเลือกเฉพาะอักษรเบรลล์สองเซลล์ที่มีรูปพ้องกันกับอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกัน คือ “**ธ**” (พ้องรูปกับอักษรเบรลล์เซลล์เดียว “การันต์+ท”), “**ฃ**” (พ้องรูปกับอักษรเบรลล์เซลล์เดียว “การันต์+ข”) , “**สระเอาะ**” (พ้องรูปกับอักษรเบรลล์เซลล์เดียว “อ+สระอะ), “ฤ” (พ้องรูปกับอักษรเบรลล์เซลล์เดียว “ร+ๆ”), “ฦ” (พ้องรูปกับอักษรเบรลล์เซลล์เดียว “ล+ๆ”)

4) ประโยคที่ประกอบด้วยคำทับศัพท์ จะคัดเลือกเฉพาะคำทับศัพท์ที่บรรจุไว้ในพจนานุกรมฉบับราชบัณฑิตยสถาน พ.ศ.2542 [16] เท่านั้น

ในการเลือกประโยคประเภทต่างๆ ดังที่ได้กล่าวไว้ในข้างต้นจากคลังข้อความ BEST ผู้วิจัยเลือกใช้เครื่องมือที่ชื่อว่า grep ซึ่งเป็นยูนิกซ์ทูล (UNIX tool) หรือเครื่องมือหนึ่งที่เป็นโปรแกรมย่อยของระบบปฏิบัติการยูนิกซ์ ที่สามารถคัดเลือกข้อความที่ต้องการได้โดยใช้นิพจน์ (Regular expression) หรือใช้รายการคำ (Word list) ในการค้นหาประโยคที่ตรงกับเงื่อนไขที่ผู้ใช้สร้างไว้

ในงานวิจัยนี้ได้เลือกใช้วิธีคัดเลือกประโยคที่ต้องการโดยใช้รายการคำ เพราะไม่สามารถสร้างนิพจน์ให้ครอบคลุมชนิดของประโยคที่ต้องการคัดเลือกจากคลังข้อความ BEST ได้ทั้งหมด และเนื่องจากปัจจุบันยังไม่มีการรวบรวมรายการคำประเภทคำควบกล้ำ, คำที่มีการใช้สระประสม, คำที่มีการใช้อักษรเบรลล์สองเซลล์และคำทับศัพท์

ดังนั้นผู้วิจัยจึงจำเป็นต้องสร้างรายการคำดังกล่าวด้วยตนเอง ซึ่งรายการคำเหล่านี้ได้นำมาจากพจนานุกรมฉบับราชบัณฑิตยสถาน พ.ศ.2542 [16] โดยคัดเลือกเฉพาะคำที่เป็นคำควบกล้ำ, คำที่มีการใช้สระประสม, คำที่มีการใช้อักษรเบรลล์สองเซลล์และคำทับศัพท์ และเก็บไว้ในไฟล์ชื่อ word_list.txt โดยประกอบไปด้วยคำควบกล้ำจำนวน 5,115 คำ, สระประสมจำนวน 5,875 คำ, อักษรเบรลล์สองเซลล์จำนวน 86 คำและคำทับศัพท์จำนวน 611 คำ เมื่อสร้างรายการคำที่ต้องการได้แล้ว ผู้วิจัยจึงนำรายการคำดังกล่าวเป็นเงื่อนไขในการคัดเลือกประโยคที่ต้องการจากคลังข้อความ BEST โดยใช้โปรแกรมย่อย grep ในการค้นหาประโยคที่ต้องการจากคลังข้อความ BEST และใช้งานบนระบบปฏิบัติการลีนุกซ์อูบันตู (Linux Ubuntu) เวอร์ชัน 9.04 โดยมีภาพรวมการทำงานดังแสดงในรูปที่ 3.2



รูปที่ 3.2 ภาพรวมวิธีการคัดเลือกประโยคที่ต้องการจากคลังข้อความ BEST

จากภาพรวมวิธีการคัดเลือกประโยคที่ต้องการในรูปที่ 3.2 ผู้วิจัยได้เรียกใช้เครื่องมือ grep โดยใช้คำสั่ง "**grep –f word_list.txt file_name.txt > output_file.txt**" ใน Shell prompt โดยระบุออฟชัน –f หมายถึงระบุว่าให้นำรายการคำที่อยู่ในไฟล์ word_list.txt คือไฟล์ที่บรรจุรายการคำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์ที่ผู้วิจัยได้คัดเลือกมาจากพจนานุกรมฉบับราชบัณฑิตยสถาน พ.ศ.2542 มาใช้ค้นหาประโยคในคลังข้อความ BEST ที่มีการใช้คำเหล่านั้นอยู่ในประโยค โดยไฟล์ชื่อ file_name.txt คือไฟล์ในคลังข้อความ BEST และสัญลักษณ์ > หมายถึงกำหนดให้โปรแกรมย่อย grep เขียนเอาท์พุทที่ได้ใส่ไว้ในไฟล์ที่ชื่อว่า output_file.txt และมีรายละเอียดของผลการคัดเลือกประโยคที่ต้องการจากคลังข้อความ BEST ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 รายละเอียดการคัดเลือกประโยคที่ต้องการจากคลังข้อความ BEST

ประเภทของคำที่ใช้คัดเลือกข้อความ	จำนวนคำที่ใช้คัดเลือก (คำ)	ประเภทข้อความที่คัดเลือกจากคลังข้อความ BEST (คลังข้อความ BEST มีขนาดไฟล์ทั้งหมด 71.8 MB)				รวมทั้งหมดคิดเป็นเปอร์เซ็นต์
		ประเภทบทความ (ขนาด)	ประเภทสารานุกรมไทยสำหรับเยาวชน (ขนาด)	ประเภทบทความข่าว(ขนาด)	ประเภทนวนิยาย (ขนาด)	
คำควบกล้ำ	5,115	7.76 MB	7.50 MB	9.59 MB	7.95 MB	45.68 %
สระประสม	5,875	7.59 MB	7.29 MB	9.31 MB	7.73 MB	44.43 %
อักษรเบรลล์สองเซลล์	86	97.2 KB	32.6 KB	173 KB	768 KB	1.45 %
คำทับศัพท์	611	529 KB	807 KB	861 KB	589 KB	3.79 %

จากตารางที่ 3.2 เห็นได้ว่าคำควบกล้ำและสระประสมมีการใช้งานในประโยคประเภทต่างๆ ในภาษาไทยมากที่สุดโดยคิดเป็น 45.68 % และ 44.43 % ตามลำดับ ซึ่งมีขนาดประมาณเกือบครึ่งหนึ่งของคลังข้อความ BEST โดยคำควบกล้ำบางคำก็อาจรวมอยู่ในคำที่มีสระประสมและแม้ว่าจะมีการใช้งานอักษรเบรลล์สองเซลล์และคำทับศัพท์น้อยมาก แต่ก็ยังมีจำนวนการใช้งานอยู่พอสมควร จึงทำให้ปัญหาการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยดังที่ได้กล่าวไว้แล้ว นับได้ว่าเป็นปัญหาที่มีความสำคัญเช่นกัน

เมื่อได้ไฟล์เอาท์พุทตามที่ต้องการแล้วผู้วิจัยจึงนำไฟล์ทั้งหมดดังกล่าวไปแปลงให้อยู่ในรูปของอักษรเบรลล์ภาษาไทยระดับ 1 โดยใช้โปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์ [1] ซึ่งเป็นโปรแกรมที่มีการพัฒนามาก่อนแล้ว (รายละเอียดของโปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์จะอธิบายในหัวข้อที่ 2.8) เพื่อนำไปสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทยต่อไป ซึ่งตัวอย่างประโยคที่ได้คัดเลือกมาจากคลังข้อความ BEST และตัวอย่างประโยคที่ถูกแปลงให้อยู่ในรูปของอักษรเบรลล์ภาษาไทยระดับ 1 แสดงในรูปที่ 3.3 และ 3.4 ตามลำดับ

ตัวอย่างประโยคภาษาไทย

|การ|พัฒนา|วิจัย|รถ|แทรกเตอร์|หลาย|แบบ|

|ประกอบ|ด้วย|กระบะ|เมล็ด|พืช|ที่|บรรจุ|อยู่|บน|ตะแกรง|

|กลาย|เป็น|ผู้|กระทำ|ความ|ผิด|ฐาน|บุกรุก|อุทยาน|แห่ง|ชาติ|

|มี|การ|เคลื่อนย้าย|ทรัพยากร|ที่|เป็น|วัตถุ|ดิบ|ของ|ระบบ|การ|ผลิต|

รูปที่ 3.3 ตัวอย่างประโยคที่คัดเลือกจากคลังข้อความ BEST

ตัวอย่างประโยคที่แปลงเป็นอักษรเบรลล์ภาษาไทยระดับ 1

|g*r|?>-)n*|wbj>y|rt|<)rg|%r0|hl*y|<vv|

|&ragov|d4wy|grava|fml'd|?5+|)29|vrrjc|oy39|vn||a<gr}|

|gl*y|f&'n|p34|gra)z|uw*m|pbd|,t*n|vcgrcg|oc)y*n|<h9}|+*|b|

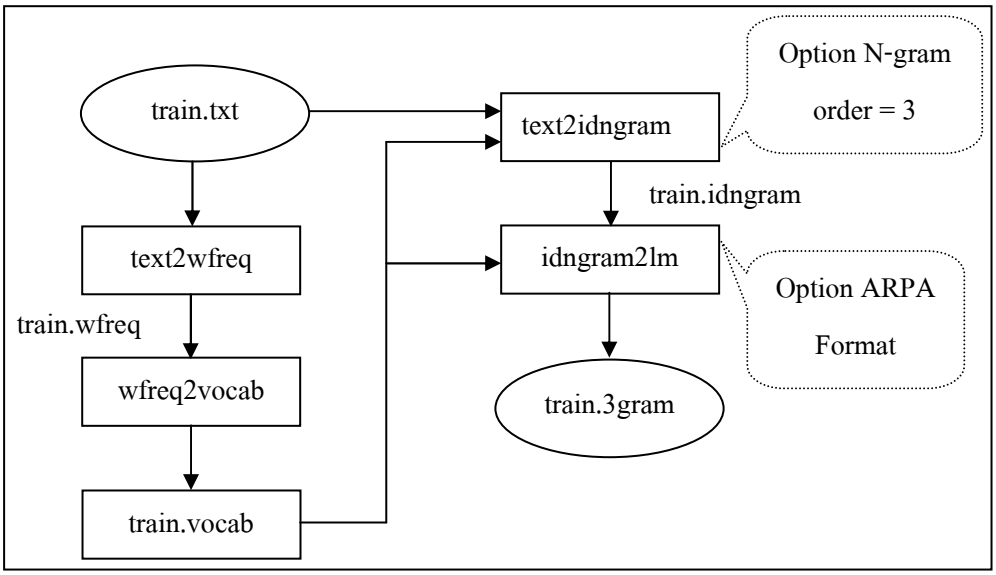
|m2|g*r|ulq9ny4*y|)r>?y*gr|)29|f&'n|w>|tc|dbv|ko}|ravv|g*r|plb|

รูปที่ 3.4 ประโยคที่แปลงให้อยู่ในรูปของอักษรเบรลล์ภาษาไทย

3.3.2 การสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทย

การสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทยหรือการฝึกฝนเอ็นแกรมโมเดล [18] คือ กระบวนการทางสถิติเพื่อให้เอ็นแกรมโมเดลคำนวณค่าความน่าจะเป็นของคำหนึ่งๆ ว่าสามารถที่จะเกิดขึ้นร่วมกับคำใดได้บ้าง โดยมีขั้นตอนย่อยๆ อยู่หลายขั้นตอนด้วยกัน ซึ่งจะอธิบายโดยใช้แผนภาพในภายหลังและผลที่ได้จากการฝึกฝนเอ็นแกรมโมเดลนี้จะได้เป็นไฟล์นามสกุล .gram และในไฟล์ .gram นี้จะบรรจุค่าความน่าจะเป็นของคำนั้นๆ ว่ามีโอกาสที่จะเกิดขึ้นร่วมกับคำใดได้บ้างด้วยค่าความน่าจะเป็นเท่าไร และบรรจุค่าดัชนีถ่วงน้ำหนัก (Back-off weight) ที่จะนำไปใช้ในกระบวนการ Smoothing อีกด้วย โดยมีการจัดเก็บให้อยู่ในรูปแบบของ ARPA format [19] ซึ่งเป็นรูปแบบไฟล์มาตรฐานของโมเดลภาษา (Language Model) และในงานวิจัยที่นำเสนอนี้ได้เลือกชุดเครื่องมือที่เรียกว่า CMU-SLM toolkit [20] ซึ่งเป็นชุดเครื่องมือที่ช่วยสร้างโมเดลภาษา อีกทั้งชุดเครื่องมือนี้มีการเผยแพร่เป็นสาธารณะ และมีการนำไปใช้งานกันอย่างแพร่หลาย โดยสามารถใช้งานได้ง่าย มีคู่มืออธิบายชัดเจน และเป็นชุดเครื่องมือที่ครบถ้วนสมบูรณ์ (รายละเอียดของชุด

เครื่องมือ CMU-SLM toolkit จะอธิบายในหัวข้อ 2.7) โดยมีขั้นตอนการสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทยดังแสดงในรูปที่ 3.5



รูปที่ 3.5 ขั้นตอนการสร้างโมเดลอักษรเบรลล์ภาษาไทยด้วยชุดเครื่องมือ CMU-SLM toolkit [18]

จากรูปที่ 3.5 มีขั้นตอนการสร้างโมเดลอักษรเบรลล์ภาษาไทยดังต่อไปนี้

1. สร้างไฟล์ train.wfreq จาก train.text ด้วยฟังก์ชัน text2wfreq โดยไฟล์ train.wfreq จะเป็นไฟล์ที่เก็บคำที่มีทั้งหมดใน train.text พร้อมทั้งค่าความถี่ของคำแต่ละคำ
2. สร้างไฟล์ train.vocab จาก train.wfreq ด้วยฟังก์ชัน text2wfreq โดยไฟล์ train.vocab จะเป็นรายการของคำ (Vocabulary) ทั้งหมดที่มีอยู่ในไฟล์ train.text
3. สร้างไฟล์ train.idngram โดยอาศัยไฟล์ train.vocab และไฟล์ train.text ด้วยฟังก์ชัน text2idngram โดยไฟล์ train.idngram จะเป็น Binary file ที่เก็บ Index ของ N-gram ที่เกิดใน train.text ซึ่งจะต้องระบุขั้น Order ของ N-gram โดยระบุหลัง option –n เช่น –n 3 คือ กำหนดว่าจะสร้าง N-gram order ที่ N เท่ากับ 3 ซึ่งก็คือ Trigram
4. สร้างไฟล์ train.3gram จาก train.idngram และ train.vocab ด้วยฟังก์ชัน idngram2lm โดยไฟล์ train.3gram จะเป็นรูปแบบไฟล์ N-gram ที่นำไปใช้ ซึ่งถูกจัดเก็บให้อยู่ในรูปแบบของ ARPA format ซึ่งการระบุ Option -arpa จะได้ไฟล์เอาต์พุตในรูปแบบของ

ARPA format และการระบุ Option –binary จะได้ไฟล์เอาต์พุตในรูปแบบของ Binary format

เมื่อสร้างโมเดลอักษรเบรลล์ภาษาไทยเสร็จแล้วจะได้ไฟล์นามสกุล .gram ซึ่งผู้วิจัยจะนำไฟล์นี้ไปใช้งานต่อไป โดยตัวอย่างของไฟล์โมเดลอักษรเบรลล์ภาษาไทยแสดงในรูปที่ 3.6

ตัวอย่างไฟล์โมเดลอักษรเบรลล์ภาษาไทย (gram=3)

```
\1-grams:
. . .
-5.1338 !*g1 0.2118
-5.6109 !*ivr!* 0.3348
. . .
\2-grams:
. . .
-1.5478 )zn(vr>,tv*l gra)rw}fg-s|rl<ashgr,n0 0.5339
-1.5478 )zn(vr>,tv*l sn*mhlw} -0.1462
. . .
\3-grams:
. . .
-0.5000 <m9 ;l <n9non
-0.3346 <m9 ?29 n4o}
. . .
```

รูปที่ 3.6 ตัวอย่างไฟล์โมเดลอักษรเบรลล์ภาษาไทย

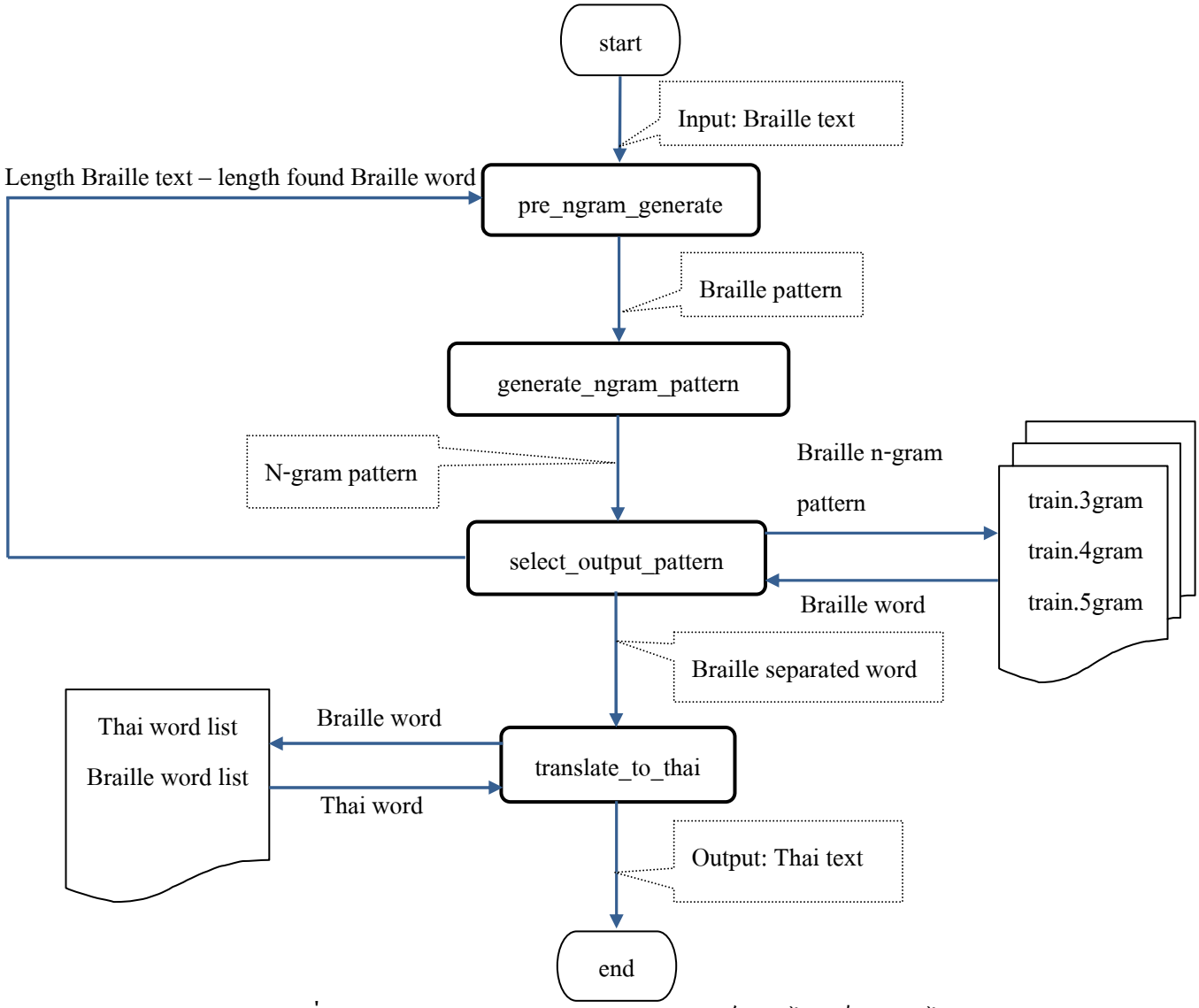
ในรูปที่ 3.6 เป็นตัวอย่างไฟล์โมเดลอักษรเบรลล์ภาษาไทยโดยใช้จำนวนแกรมเท่ากับ 3-แกรม ซึ่งภายในไฟล์จะมีการเรียงลำดับตั้งแต่ 1-gram, 2-gram ไปจนถึง 3-gram และหากในขั้นตอนที่ 3 ในการสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทยถ้าผู้ใช้ได้ระบุตัวเลือกจำนวนแกรมเป็น

n แกรม (n เป็นจำนวนนับ) ภายในไฟล์เอ็นแกรมโมเดลที่ฝึกฝนเสร็จแล้วก็จะมีการเรียงลำดับตั้งแต่ 1-gram, 2-gram ไปเรื่อยๆ จนถึงจำนวนแกรมที่ผู้ใช้ได้กำหนดไว้นั่นเอง และในคอลัมน์ซ้ายสุดจะเป็นค่าความน่าจะเป็นที่พบรูปแบบคำนั้นๆ ซึ่งค่าความน่าจะเป็นนี้จะเห็นได้ว่าติดลบ นั่นก็เพราะค่าความน่าจะเป็นนั้นมีค่าที่น้อยมากๆ เมื่อนำไปคำนวณแล้วจะทำให้มีค่าน้อยมากจนแทบจะเป็นศูนย์ จึงมีการจัดรูปแบบใหม่โดยใช้ลอการิทึมฐานสิบ ($\log_{10}$) จึงทำให้ค่าความน่าจะเป็นติดลบ คอลัมน์ตรงกลาง (เป็นส่วนที่มีตัวอักษรและสัญลักษณ์) เป็นรายการของคำอักษรเบรลล์ภาษาไทยระดับ 1 ที่มีอยู่ในคลังข้อมูลที่ผู้วิจัยได้นำมาสร้างไฟล์โมเดลอักษรเบรลล์ภาษาไทย ซึ่งรายการคำที่แสดงอยู่นี้มีความหมายว่า คำที่อยู่หลังสุดซึ่งแบ่งโดยการเว้นวรรคนั้น จะเกิดขึ้นร่วมกับคำอื่นๆ ที่อยู่ก่อนหน้าด้วยค่าความน่าจะเป็นที่แสดงไว้ในคอลัมน์ซ้ายสุด และในคอลัมน์ขวาสุดที่เป็นตัวเลขเป็นค่าที่แสดงถึงค่าดัชนีถ่วงน้ำหนัก (Back-off weight) คือค่าถ่วงน้ำหนักเมื่อคำหรือชุดของคำไม่มีอยู่หรือไม่พบในคลังข้อความ โดยจะนำค่านี้ไปใช้ในการประมาณค่าความน่าจะเป็นตามหลักวิธีการของ Smoothing ซึ่งค่าดัชนีถ่วงน้ำหนักนี้จะนำไปแทนค่าตัวแปร $\lambda_{w_{i-n+1}^{i-1}}$ ในสมการที่ 5 แต่จะไม่มีค่าดัชนีถ่วงน้ำหนักในกลุ่มจำนวนแกรมที่ผู้ใช้ได้ระบุไว้ในขั้นตอนที่ 3 (ในที่นี้คือกลุ่มคำที่อยู่ในช่วง 3-gram) นั่นเป็นเพราะว่าหากพบชุดของคำที่เขียนเรียงติดกันที่อยู่ในช่วง 3-gram ก็จะมีค่าความน่าจะเป็นเท่ากับค่าในคอลัมน์ซ้ายสุดนั่นเอง และหากไม่พบจะนำเอาค่าดัชนีถ่วงน้ำหนักในช่วง 2-gram และ 1-gram ไปคำนวณหาค่าความน่าจะเป็นโดยวิธีการของ Smoothing technique ต่อไป

3.4 ภาพรวมของระบบที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้

ในวิทยานิพนธ์นี้ได้นำเสนอวิธีการปรับปรุงประสิทธิภาพในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยในรูปแบบ File-to-file โดยนำเทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำอักษรเบรลล์ภาษาไทยเพื่อลดความกำกวมที่เกิดขึ้นจากลักษณะการเขียนของอักษรเบรลล์ภาษาไทยระดับ 1 จากนั้นแปลงให้เป็นภาษาไทยโดยใช้วิธีเทียบแบบคำต่อคำ ซึ่งจำเป็นจะต้องสร้างรายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยไว้ล่วงหน้า

ในรูปที่ 3.7 แสดงองค์ประกอบต่างๆ ของระบบที่ได้ออกแบบไว้และลักษณะการเชื่อมต่อกันของฟังก์ชันหลักรวมทั้งการเชื่อมต่อองค์ประกอบอื่นๆ ที่จำเป็นต้องใช้ร่วมในการแปลงอักษรเบรลล์ภาษาไทยระดับ 1 ไปเป็นภาษาไทย



รูปที่ 3.7 ภาพรวมโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย
ที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ

ในรูปที่ 3.7 เป็นภาพรวมของโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ โดยโปรแกรมนี้รับอินพุตเข้ามาเป็นข้อความที่เขียนอยู่ในรูปอักษรเบรลล์ภาษาไทยระดับ 1 ซึ่งจะแบ่งประโยคออกเป็นส่วนๆ ตามจำนวนแกรมที่ได้กำหนดไว้ (pre_ngram_generate) และนำไปสร้างรูปแบบเอ็นแกรม (generate_ngram_pattern) จากนั้นนำรูปแบบเอ็นแกรมไปเปรียบเทียบกับไฟล์โมเดลอักษรเบรลล์ภาษาไทยเพื่อตัดคำอักษรเบรลล์ภาษาไทย (select_output_pattern) และแปลงให้เป็นประโยค

ภาษาไทย (translate_to_thai) โดยวิธีเทียบคำซึ่งใช้งานร่วมกับรายการคำภาษาไทยและรายการคำอักษรเบรลล์ภาษาไทย โดยแต่ละส่วนจะมีการเชื่อมต่อกันดังที่แสดงในรูปที่ 3.7 ซึ่งรายละเอียดการทำงานของขั้นตอนต่างๆ จะอธิบายไว้ในหัวข้อที่ 3.6 และในหัวข้อที่ 3.5 จะอธิบายถึงรายละเอียดของอินพุตและเอาท์พุทของโปรแกรมที่นำเสนอนี้

3.5 อินพุตและเอาท์พุทของระบบ

โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้รับอินพุตเป็นไฟล์ข้อความธรรมดา ซึ่งภายในได้บรรจุข้อความที่เขียนให้อยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 โดยการทำงานของโปรแกรมนี้จะประมวลผลข้อความอักษรเบรลล์ภาษาไทยระดับ 1 ครั้งละ 1 ประโยค โดยจะแบ่งแต่ละประโยคออกจากกันด้วยการเว้นวรรค ทั้งนี้เนื่องมาจากลักษณะไวยากรณ์ของอักษรเบรลล์ภาษาไทยระดับ 1 จะใช้การเว้นวรรคในการแบ่งประโยคออกจากกันเช่นเดียวกันกับการเขียนภาษาไทย เมื่อโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำประมวลผลเสร็จแล้ว จะได้เอาท์พุทเป็นประโยคภาษาไทย และนำประโยคภาษาไทยที่ได้นั้นเขียนใส่ในไฟล์เอาท์พุทตามที่ผู้ใช้งานได้ตั้งชื่อไฟล์เอาไว้ ซึ่งจะเป็นไฟล์ข้อความธรรมดา และการทำงานของโปรแกรมจะเป็นเช่นนี้ไปเรื่อยๆ จนกว่าข้อความอักษรเบรลล์ภาษาไทยระดับ 1 ในไฟล์อินพุตจะหมดลง สำหรับวิธีการประมวลผลของโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอนี้ได้อธิบายไว้ในหัวข้อที่ 3.6

3.6 โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้

ในหัวข้อนี้จะอธิบายถึงขั้นตอนต่างๆ ของโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้รวมถึงชื่อและชนิดของตัวแปรต่างๆ ที่ใช้ในการเก็บข้อมูลและใช้ในการประมวลผล ซึ่งประกอบไปด้วย 4 ขั้นตอนดังนี้

3.6.1 ขั้นตอน pre_ngrams_generate

ขั้นตอน pre_ngram_generate นี้เป็นขั้นตอนเพื่อตระเตรียมข้อมูลให้เหมาะสมสำหรับสร้างรูปแบบเอ็นแกรม (N-gram pattern) โดยรับอินพุตเข้ามาเป็นประโยคอักษรเบรลล์ภาษาไทยระดับ 1 และตัดออกเป็นส่วนๆ ตามขนาดจำนวนแกรมที่ได้กำหนดไว้ เช่น หากกำหนดไว้เป็น 4 แกรม

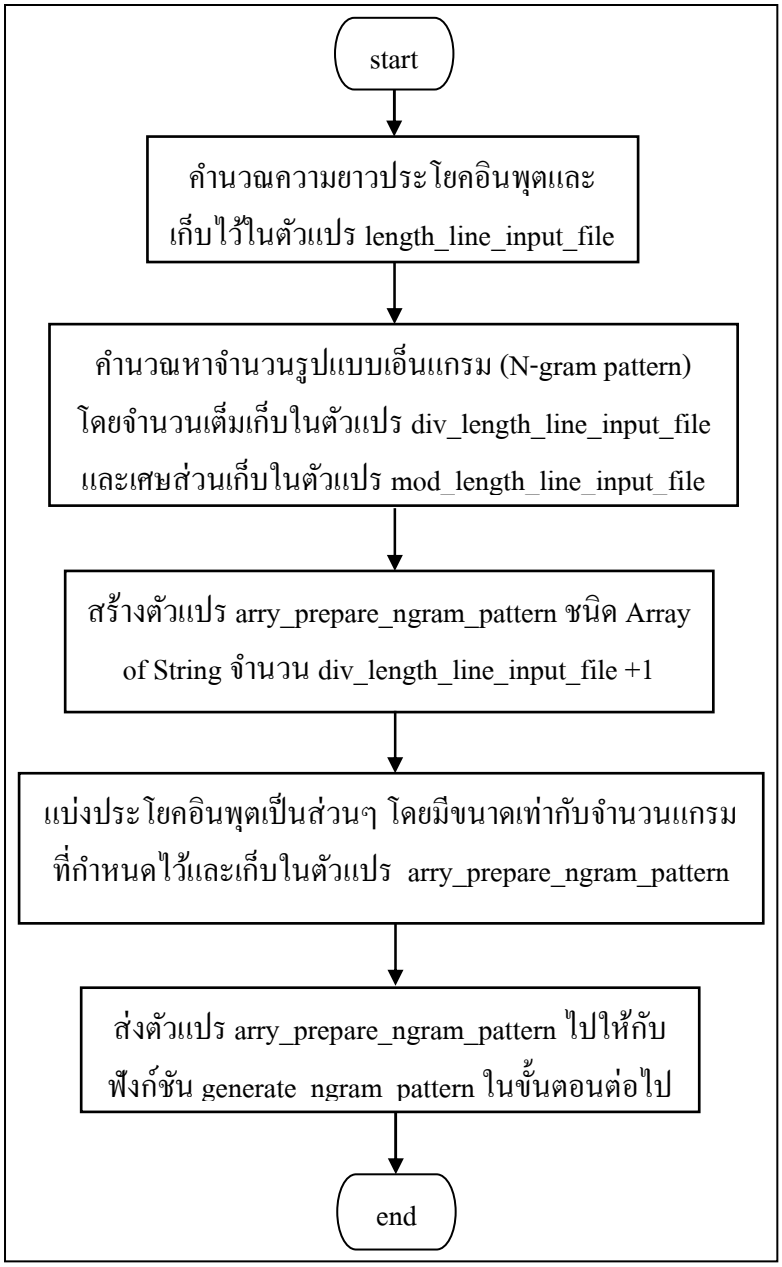
ขั้นตอนนี้ก็จะแบ่งประโยคอินพุตดังกล่าวออกเป็นส่วนๆ และแต่ละส่วนจะมีจำนวนอักขระเท่ากับ 4 อักขระ และแบ่งเช่นนี้ไปเรื่อยๆ จนจบความยาวของประโยค หากประโยคอินพุตเหลือเศษของอักขระอยู่ส่วนสุดท้ายที่แบ่งออกมาก็จะเหลือเท่ากับจำนวนอักขระที่เหลือนั้น

ผู้วิจัยได้ออกแบบให้ขั้นตอนนี้ให้เป็นฟังก์ชันหนึ่งๆ ซึ่งรับอินพุตเข้ามาครั้งละหนึ่งประโยคโดยอยู่ในรูปแบบของประโยคอักษรเบรลล์ภาษาไทยระดับ 1 และให้เอาท์พุทเป็นอาร์เรย์ของสตริง (Array of string) ที่บรรจุส่วนของประโยคที่ถูกแบ่งออกมาจากประโยคอินพุต ซึ่งมีการประกาศฟังก์ชันเป็น int prepare_ngram_pattern(string line_input_file) และมีการกำหนดตัวแปรพร้อมทั้งหน้าที่ดังแสดงในตารางที่ 3.3

ตารางที่ 3.3 รายละเอียดของตัวแปรในขั้นตอน pre_ngrams_generate

ชนิดของตัวแปร	ชื่อของตัวแปร	หน้าที่ของตัวแปร
Integer	length_line_input_file	เก็บค่าความยาวของประโยคอินพุต
Integer	index_start	เก็บค่าตำแหน่งเริ่มต้นของรูปแบบเอ็นแกรม
Integer	div_length_line_input_file	เก็บค่าจำนวนเต็มของความยาวประโยคอินพุต
Integer	mod_length_line_input_file	เก็บค่าจำนวนเศษของความยาวประโยคอินพุต
Global Array of String	arry_prepare_ngram_pattern	เก็บรูปแบบเอ็นแกรมที่ได้สร้างไว้ (N-gram pattern)
Global Integer	index_arry_prepare_ngram_pattern	เก็บค่าตำแหน่งของเอ็นแกรมที่ได้สร้างไว้

และผู้วิจัยได้ออกแบบวิธีการตระเตรียมข้อมูลโดยมีขั้นตอนการทำงานต่างๆ ดังแสดงในรูปที่ 3.8



รูปที่ 3.8 ขั้นตอน prepare_ngram_pattern

จากรูปที่ 3.8 มีรายละเอียดการทำงานต่างๆ ดังต่อไปนี้

1. คำนวณหาค่าความยาวประโยคอินพุตซึ่งอยู่ในรูปของประโยคอักษรเบรลล์ภาษาไทยระดับ 1 โดยใช้ฟังก์ชัน length() และเก็บไว้ในตัวแปร length_line_input_file
2. คำนวณจำนวนรูปแบบเอ็นแกรม (N-gram pattern) โดยหาจากนำค่าความยาวประโยคอินพุตที่เก็บไว้ในตัวแปร length_line_input_file มาหารด้วยจำนวนแกรมที่ได้กำหนดไว้

โดยหารเอาจำนวนเต็มและเก็บไว้ในตัวแปร div_length_line_input_file และหารเอาเศษและเก็บไว้ในตัวแปร mod_length_line_input_file

3. สร้างตัวแปร arry_prepare_ngram_pattern ชนิด Array of String เพื่อเก็บรูปแบบเอ็นแกรม (N-gram pattern) จำนวน div_length_line_input_file+1
4. แบ่งประโยคอินพุตเป็นส่วนๆ โดยแต่ละส่วนมีจำนวนอักขระเท่ากับจำนวนแกรมที่ได้กำหนดไว้และเก็บไว้ในตัวแปร arry_prepare_ngram_pattern ที่ได้สร้างเอาไว้ และส่งค่าตัวแปร arry_prepare_ngram_pattern ไปให้กับขั้นตอน generate_ngram_pattern

3.6.2 ขั้นตอน generate_ngram_pattern

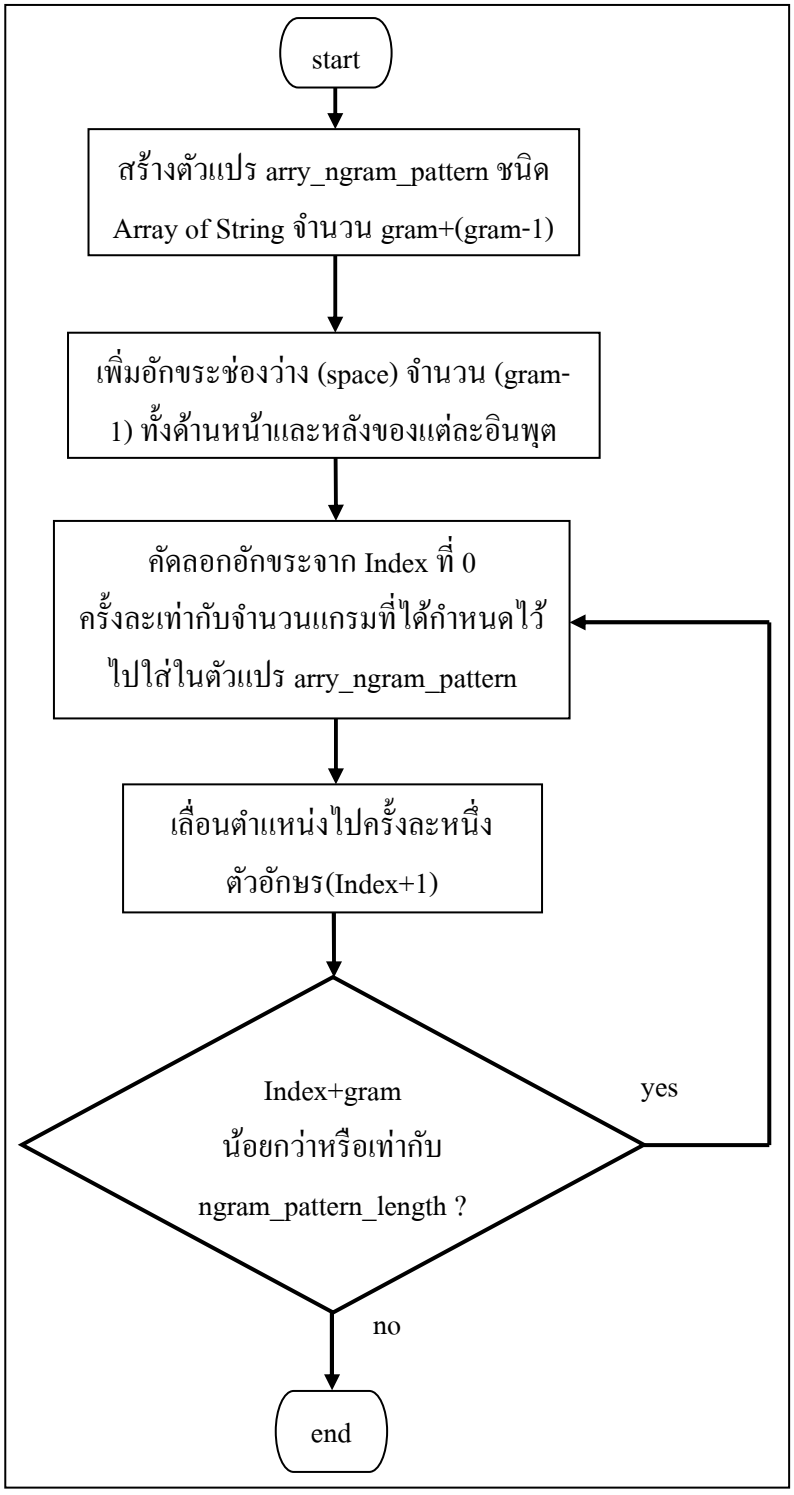
ขั้นตอน generate_ngram_pattern นี้เป็นขั้นตอนสร้างรูปแบบเอ็นแกรม (N-gram pattern) เพื่อนำไปเปรียบเทียบกับไฟล์โมเดลอักษรเบรลล์ภาษาไทยซึ่งจะอธิบายในหัวข้อที่ 3.7 เอ็นแกรมโมเดล โดยรับอินพุตเข้ามาเป็นส่วนของประโยคที่ได้จากขั้นตอน prepare_ngram_pattern และเพิ่มอักขระช่องว่าง (Space) ทั้งด้านหน้าและด้านหลังจำนวน gram-1 เพื่อเพิ่มอัตราจับคู่คำโดยขั้นตอนการสร้างรูปแบบเอ็นแกรมได้นำมาจากเว็บไซต์ ejeepss.wordpress.com [28] และส่งชุดของรูปแบบเอ็นแกรมที่ได้ไปให้กับขั้นตอน select_output_pattern เพื่อเปรียบเทียบกับไฟล์โมเดลอักษรเบรลล์ภาษาไทยและเลือกเอาท์พุทต่อไป

ผู้วิจัยได้ออกแบบให้ขั้นตอนนี้รับอินพุตเข้ามาครั้งละหนึ่งชุดโดยเป็นส่วนประโยคที่ได้จากขั้นตอน prepare_ngram_pattern และให้เอาท์พุทเป็นอาร์เรย์ ของสตริง (Array of String) ที่บรรจุชุดรูปแบบเอ็นแกรม (N-gram pattern) ซึ่งมีการประกาศฟังก์ชันเป็น int gen_ngram_pattern(string ngram_pattern) และมีการกำหนดตัวแปรพร้อมทั้งหน้าที่ดังแสดงในตารางที่ 3.4

ตารางที่ 3.4 รายละเอียดของตัวแปรที่ใช้ในขั้นตอน gen_ngram_pattern

ชนิดของตัวแปร	ชื่อของตัวแปร	หน้าที่ของตัวแปร
Integer	ngram_pattern_length	เก็บค่าความยาวของอินพุต
Global Array of String	arry_ngram_pattern	เก็บชุดรูปแบบของเอ็นแกรม
Global Integer	index_arry_ngram_pattern	เก็บตำแหน่งของชุดรูปแบบเอ็นแกรม

และผู้วิจัยได้ออกแบบวิธีการสร้างรูปแบบเอ็นแกรม (N-gram pattern) โดยมีขั้นตอนการทำงานต่างๆ ดังแสดงในรูปที่ 3.9

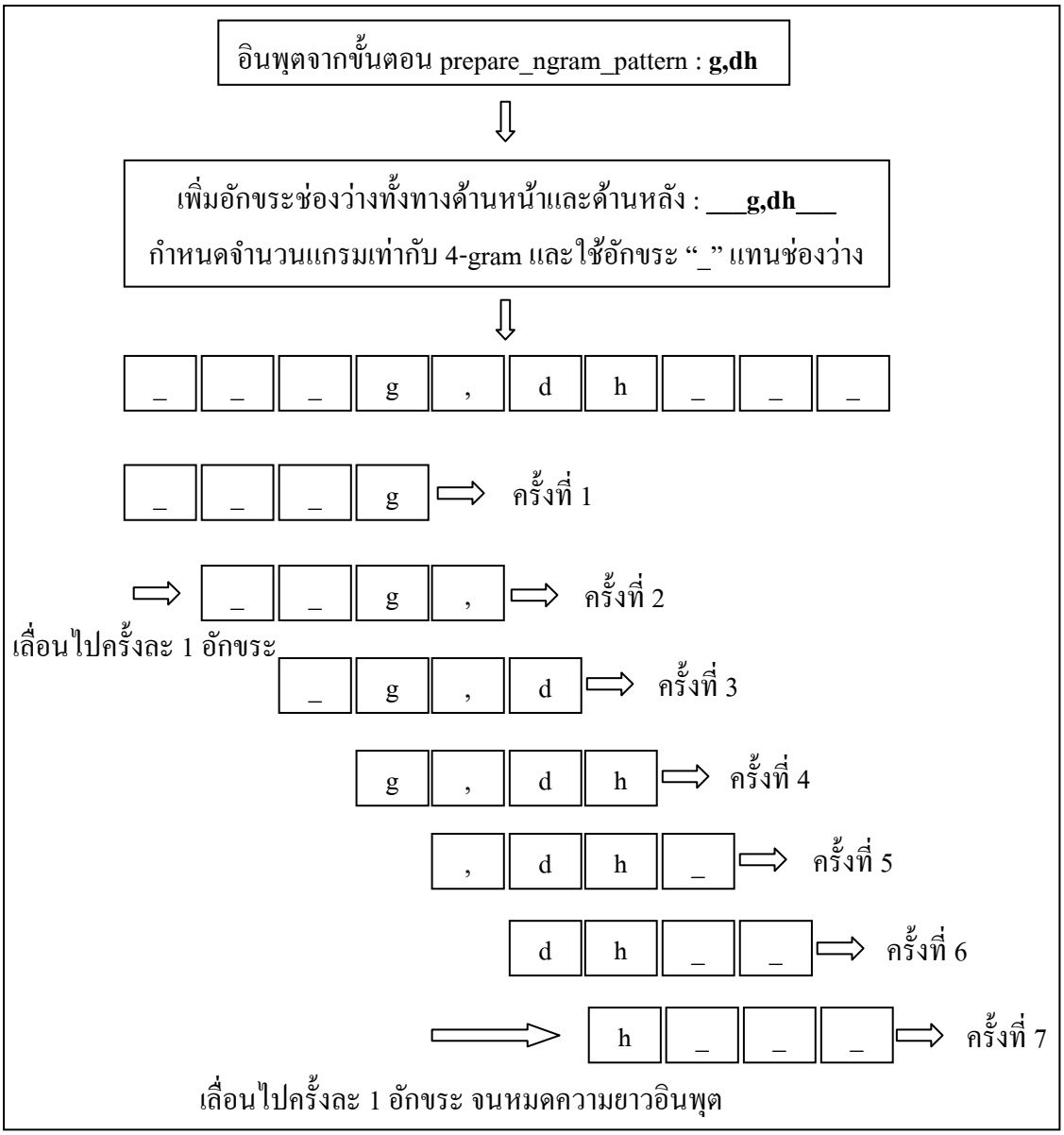


รูปที่ 3.9 ขั้นตอน gen_ngram_pattern

จากรูปที่ 3.9 มีรายละเอียดการทำงานต่างๆ ดังต่อไปนี้

1. สร้างตัวแปร arry_ngram_pattern ชนิด Array of String จำนวน gram+(gram-1) ตามที่ได้กำหนดจำนวนแกรมเอาไว้
2. เพิ่มอักขระช่องว่างทั้งทางด้านหน้าและด้านหลังของอินพุตจำนวน (gram-1) เพื่อเพิ่มอัตราการจับคู่ของคำ ยกตัวอย่างเช่น มีอินพุตเป็น g,dh และกำหนดจำนวนแกรมเท่ากับ 4-แกรม จากขั้นตอนนี้จะได้เอาท์พุทเป็น “___g,dh___” (เพื่อให้สังเกตได้ง่ายจะใช้สัญลักษณ์ “_” แทนอักขระช่องว่าง)
3. คัดลอกอักขระครั้งละเท่ากับจำนวนแกรมที่ได้กำหนดไว้ไปใส่ในตัวแปร arry_ngram_pattern ครั้งละ 1 ชุดและเลื่อนไปทางด้านขวาครั้งละ 1 อักขระและคัดลอกอักขระจำนวนครั้งละเท่ากับจำนวนแกรมที่ได้กำหนดไว้ไปใส่ในตัวแปร array_ngram_pattern อีกครั้งจนกว่าจะหมดความยาวของอินพุต
4. ส่งค่าตัวแปร arry_ngram_pattern ไปให้กับขั้นตอน select_output_pattern ต่อไป

และเพื่อให้ง่ายต่อการทำความเข้าใจผู้วิจัยจึงใช้แผนภาพแสดงขั้นตอนการสร้างรูปแบบเอ็นแกรม ดังแสดงในรูปที่ 3.10



รูปที่ 3.10 การสร้างรูปแบบเอ็นแกรม (N-gram pattern)

ในรูปที่ 3.10 เป็นวิธีการสร้างรูปแบบเอ็นแกรม (N-gram pattern) โดยจะมีลักษณะการทำงานเหมือนกับกล่องที่บรรจุตัวอักษรได้ครั้งละเท่ากับจำนวนแกรมที่ได้กำหนดไว้ ซึ่งในตัวอย่างนี้ได้กำหนดไว้ที่ 4-แกรม โดยกล่องนี้จะคัดลอกอักขระครั้งละ 4 อักขระรวมทั้งอักขระช่องว่างด้วย และเมื่อคัดลอกเสร็จแล้วจะได้เป็น 1 รูปแบบเอ็นแกรมจากนั้นเลื่อนไปด้านขวาครั้งละ 1 อักขระจนจบความยาวของอินพุตก็จะได้รูปแบบเอ็นแกรมทั้งหมดที่เป็นไปได้

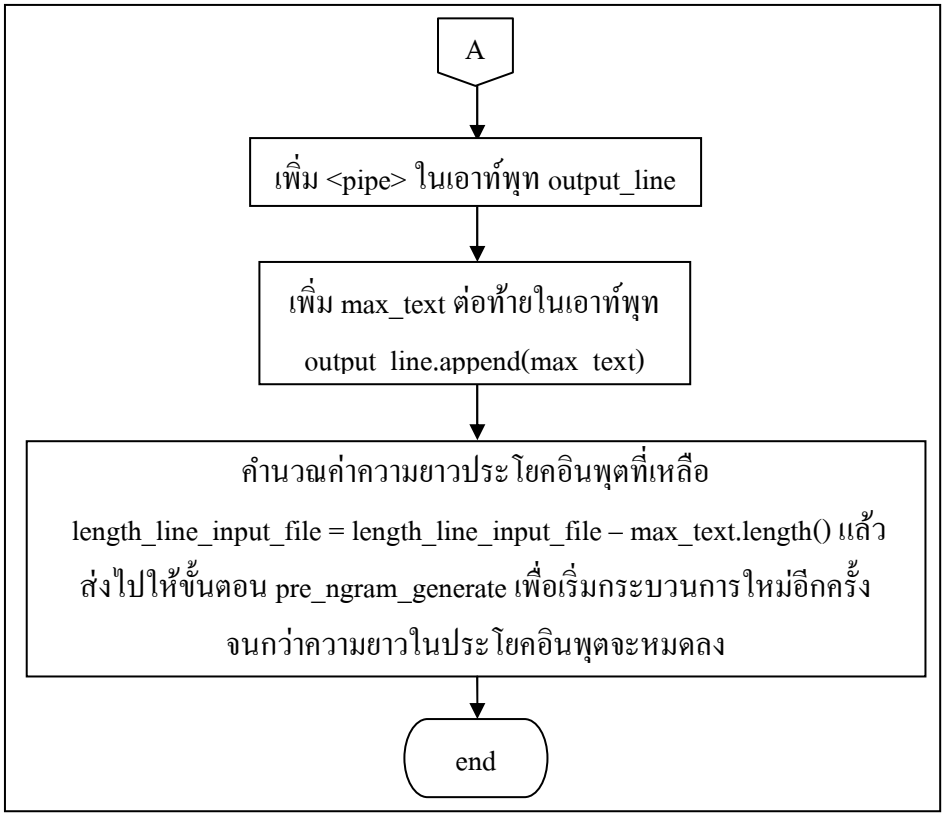
3.6.3 ขั้นตอน select_output_pattern

ขั้นตอน select_output_pattern ทำหน้าที่เปรียบเทียบรูปแบบเอ็นแกรม (N-gram pattern) ที่สร้างมาจากขั้นตอน generate_ngram_pattern เพื่อค้นหาคำอักษรเบรลล์ภาษาไทยระดับ 1 ที่มีความคล้ายคลึงกันกับรูปแบบเอ็นแกรมดังกล่าวและค้นหาว่ารูปแบบเอ็นแกรมใดในไฟล์โมเดลอักษรเบรลล์ภาษาไทยจะมีค่าความน่าจะเป็นสูงที่สุด โดยเลือกคำอักษรเบรลล์ภาษาไทยระดับ 1 เป็นเอาท์พุทและแบ่งแต่ละคำออกจากกันด้วยสัญลักษณ์ <pipe> (ใช้สัญลักษณ์ “<pipe>” เพื่อไม่ให้ซ้ำกับสัญลักษณ์ “|” ในอักษรเบรลล์ภาษาไทยที่ใช้แทนตัวอักษร “ต” ในภาษาไทย) เพื่อลดความกำกวมของรูปแบบการเขียนอักษรเบรลล์ภาษาไทยระดับ 1 และแปลงให้เป็นประโยคภาษาไทยโดยขั้นตอน translate_to_thai อีกครั้ง โดยมีการประกาศฟังก์ชันเป็น void select_output(string dictionary[],int number_dictionary_index,string arry_ngram_pattern[],int index_arry_ngram_pattern) และมีรายละเอียดของตัวแปรและหน้าที่ต่างๆ ของแต่ละตัวแปรดังแสดงในตารางที่ 3.5

ตารางที่ 3.5 รายละเอียดตัวแปรต่างๆ ที่ใช้ในขั้นตอน select_output_pattern

ชนิดของตัวแปร	ชื่อของตัวแปร	หน้าที่ของตัวแปร
Global Integer	max_score	เก็บคะแนนที่มากที่สุด
Global String	max_text	เก็บคำอักษรเบรลล์ที่มีคะแนนมากที่สุด
Global String	max_pattern	เก็บรูปแบบเอ็นแกรมที่ได้คะแนนมากที่สุด
Global Integer	max_text_index	เก็บ index คำอักษรเบรลล์ที่มีคะแนนมากที่สุด
Integer	ngram_pattern_length	เก็บความยาวของรูปแบบเอ็นแกรม
Integer	dictionary_length	เก็บความยาวของคำในรายการคำอักษรเบรลล์ภาษาไทย
Integer	score	เก็บค่าคะแนนของแต่ละคำในปัจจุบัน
Integer	num_ngram_pattern	เก็บจำนวนรูปแบบของเอ็นแกรม
Array of String	array_ngram_pattern	เก็บรูปแบบของเอ็นแกรม
Global String	line_input_file	เก็บประโยคอินพุต
Global String	output_line	เก็บประโยคเอาท์พุท
Global Integer	length_line_input_file	เก็บค่าความยาวของประโยคอินพุต
Global Array of String	word_list_brl	เก็บรายการคำอักษรเบรลล์ภาษาไทย
Global Integer	index_word_list_brl	เก็บ index ของรายการคำอักษรเบรลล์ภาษาไทย
Global Float	max_prob	เก็บค่าความน่าจะเป็นที่มากที่สุด
Float	current_prob	เก็บค่าความน่าจะเป็นของแต่ละคำในปัจจุบัน

ผู้วิจัยได้ออกแบบวิธีการสร้างเอาท์พุทโดยอาศัยข้อมูลต่างๆ จากฟังก์ชันอื่นๆ ซึ่งได้แบ่งฟังก์ชันออกเป็น 3 ฟังก์ชันหลักคือ 1) ฟังก์ชัน select_output ทำหน้าที่สร้างเอาท์พุทซึ่งเป็นประโยคอักษรเบรลล์ภาษาไทยระดับ 1 ที่แบ่งคำแล้วโดยอาศัยข้อมูลจากฟังก์ชัน ngram_to_word และฟังก์ชัน find_ngram เพื่อใช้ประกอบในการตัดสินใจตัดคำ 2) ฟังก์ชัน ngram_to_word ทำหน้าที่คำนวณคะแนนเพื่อเปรียบเทียบรูปแบบเอ็นแกรมว่าคล้ายคลึงกับคำใดมากที่สุด และ 3) ฟังก์ชัน find_ngram ทำหน้าที่ค้นหารูปแบบของคำในไฟล์โมเดลอักษรเบรลล์ภาษาไทยว่ารูปแบบคำใดมีค่าความน่าจะเป็นสูงที่สุด โดยมีขั้นตอนการทำงานต่างๆ ดังแสดงในรูปที่ 3.11-3.13 ตามลำดับ



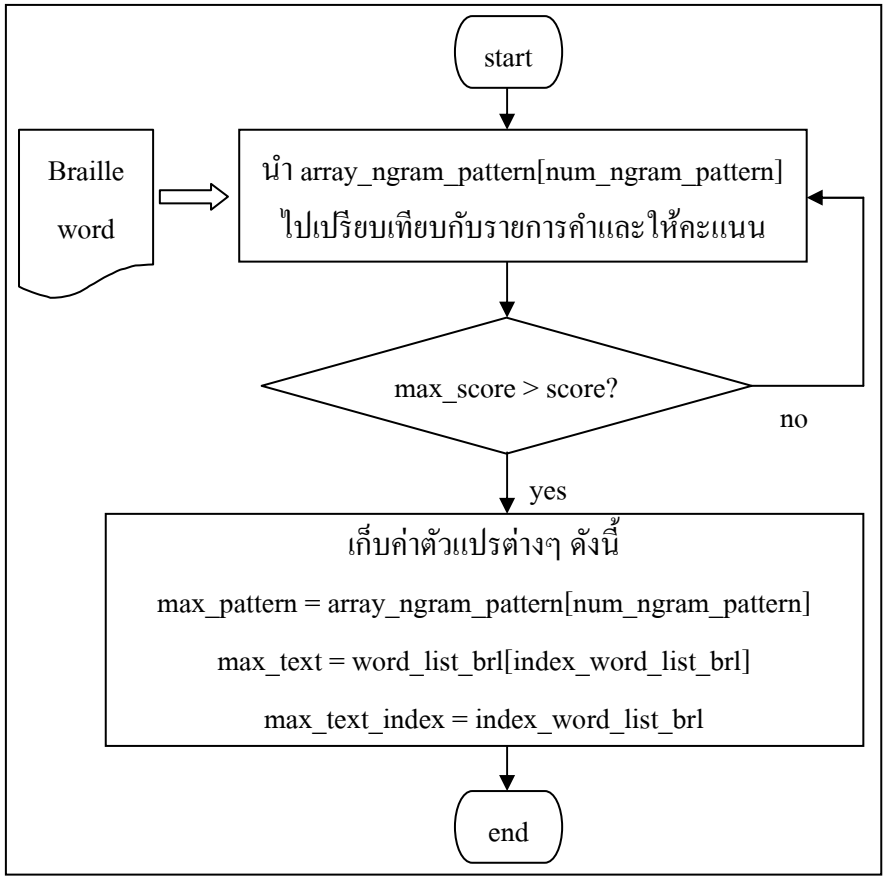
รูปที่ 3.11 ขั้นตอนการทำงานของฟังก์ชัน select_output

จากรูปที่ 3.11 แสดงขั้นตอนการทำงานของฟังก์ชัน select_output ซึ่งมีลำดับการทำงานดังนี้

1. คำนวณหาจำนวนรูปแบบของเอ็นแกรมจาก gram+(gram-1) ตามที่ได้กำหนดจำนวนแกรมเอาไว้แล้ว
2. ส่งรูปแบบเอ็นแกรมแต่ละรูปแบบที่เก็บไว้ในตัวแปร array_ngram_pattern [num_ngram_pattern] ไปให้กับฟังก์ชัน ngram_to_word
3. ทำซ้ำขั้นตอนที่ 1 และ 2 จนกระทั่งครบรูปแบบเอ็นแกรมทั้งหมดซึ่งจะได้คำที่มีความคล้ายคลึงกับรูปแบบเอ็นแกรมมากที่สุดซึ่งเก็บไว้ในตัวแปร max_text
4. ตรวจสอบว่าคำอักษรเบรลล์ภาษาไทยที่เลือกมานั้นพบในประโยคอินพุตหรือไม่ หากไม่พบให้กลับไปเริ่มขั้นตอน pre_ngram_generate ใหม่เพื่อสร้างชุดรูปแบบเอ็นแกรมใหม่ และหากพบว่ามีคำนั้นมีอยู่ในประโยคอินพุตให้ส่งคำๆ นั้นไปให้กับฟังก์ชัน find_ngram
5. เมื่อส่งค่า max_text ซึ่งเป็นคำอักษรเบรลล์ภาษาไทยระดับ 1 ที่เลือกจากวิธีเปรียบเทียบความคล้ายคลึงไปให้กับฟังก์ชัน find_ngram เพื่อค้นหารูปแบบของคำๆ นั้นว่าเกิดขึ้น

ร่วมกับคำๆ ใดได้บ้างและมีค่าความน่าจะเป็นเท่าไร โดยเมื่อผ่านฟังก์ชันนี้แล้วก็จะได้รูปแบบคำที่ใช้ฝึกฝนเอ็นแกรมโมเดลที่มีค่าความน่าจะเป็นสูงที่สุด

6. เพิ่มสัญลักษณ์ “<pipe>” เพื่อเป็นสัญลักษณ์ในการแบ่งคำแต่ละคำออกจากกันลงในประโยคเอาท์พุทซึ่งเก็บไว้ในตัวแปร output_line
7. เพิ่มคำที่เลือกมาโดยอาศัยข้อมูลจากฟังก์ชัน ngram_to_word และฟังก์ชัน find_ngram ซึ่งเก็บไว้ในตัวแปร max_text ลงในประโยคเอาท์พุท
8. คำนวณค่าความยาวประโยคอินพุตที่เหลือหลังจากแบ่งคำปัจจุบันเสร็จแล้วกลับไปให้กับขั้นตอน pre_ngram_generate เพื่อเริ่มกระบวนการใหม่
9. ทำซ้ำตั้งแต่ขั้นตอน pre_ngram_generate จนถึงขั้นตอน select_output จนกว่าความยาวของประโยคอินพุตที่เก็บไว้ในตัวแปร length_line_input_file จะหมดลง

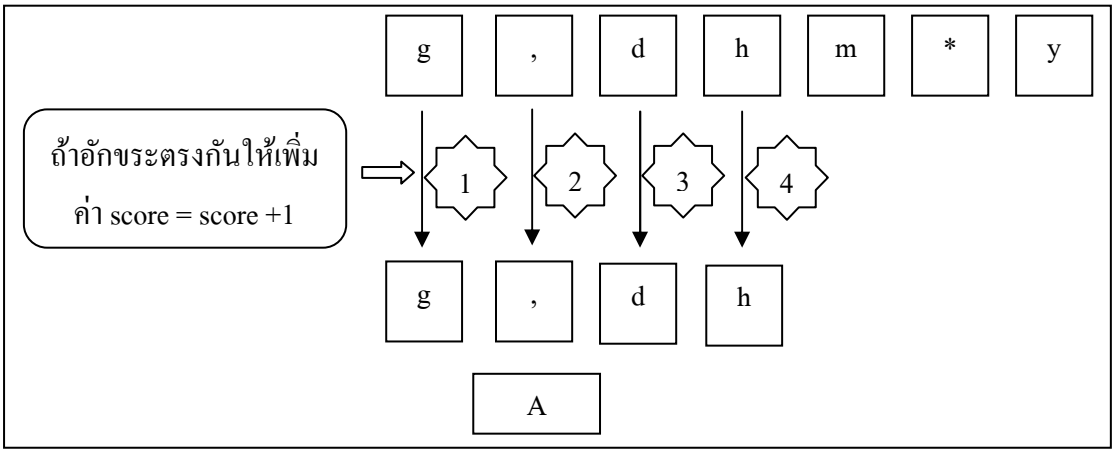


รูปที่ 3.12 ขั้นตอนการทำงานของฟังก์ชัน ngram_to_word

ผู้วิจัยได้ออกแบบฟังก์ชัน ngram_to_word เพื่อทำหน้าที่เปลี่ยนจากรูปแบบเอ็นแกรม (N-gram pattern) ให้ไปเป็นคำอักษรเบรลล์ภาษาไทยระดับ 1 โดยใช้หลักการของ Pattern Matching และการให้คะแนนความคล้ายคลึง ทั้งนี้เพราะต้องการนำคำอักษรเบรลล์ภาษาไทยดังกล่าวไปค้นหารูปแบบของคำในไฟล์โมเดลอักษรเบรลล์ภาษาไทยที่บ่งบอกว่าคำอักษรเบรลล์คำใดสามารถเกิดร่วมกันกับคำใดได้บ้างและแต่ละรูปแบบมีค่าความน่าจะเป็นเท่าไหร่ ซึ่งวิธีการทำงานของฟังก์ชัน ngram_to_word มีรายละเอียดดังนี้

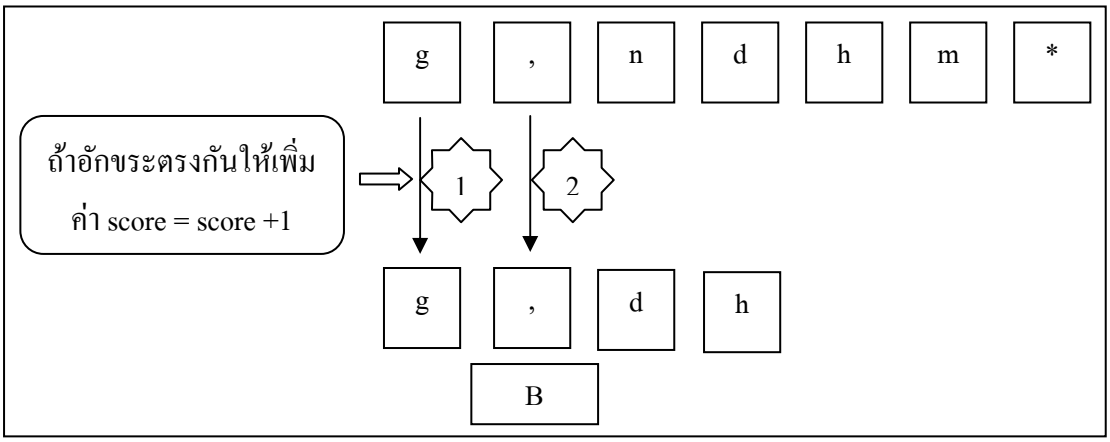
1. นำแต่ละรูปแบบเอ็นแกรม (N-gram pattern) ไปเปรียบเทียบกับรายการคำอักษรเบรลล์ภาษาไทยที่ได้สร้างเอาไว้แล้ว โดยใช้หลักการเช่นเดียวกันกับ Pattern Matching แต่อักขระตัวแรกของรูปแบบเอ็นแกรมจะต้องตรงกันกับอักขระตัวแรกของรายการคำอักษรเบรลล์ภาษาไทยเท่านั้น
2. ให้คะแนนแต่ละรูปแบบเอ็นแกรมโดยหากอักขระตัวแรกตรงกันกับรายการคำอักษรเบรลล์ภาษาไทยก็ให้คะแนน 1 คะแนนและหากตัวถัดๆ ไปตรงกันอีกก็เพิ่มคะแนนเช่นนี้ไปเรื่อยๆ จนหมดความยาวรูปแบบเอ็นแกรมและอักขระของรูปแบบเอ็นแกรมและอักขระของรายการคำอักษรเบรลล์ภาษาไทยที่เหมือนกันต้องมีลักษณะเรียงติดกันไปตลอดไม่มีอักขระอื่นๆ ที่ไม่ตรงกันมาคั่นกลาง
3. หากรูปแบบเอ็นแกรมใดมีค่าคะแนนมากกว่าค่าคะแนนมากที่สุด (max_score) ให้เก็บค่าต่างๆ ดังนี้
   - เก็บค่ารูปแบบเอ็นแกรมที่มีคะแนนมากที่สุดไว้ในตัวแปร max_pattern
   - เก็บค่าคะแนนมากที่สุดไว้ในตัวแปร max_score
   - เก็บคำอักษรเบรลล์ภาษาไทยที่มีความคล้ายคลึงกับรูปแบบเอ็นแกรมมากที่สุดไว้ในตัวแปร max_text
   - เก็บค่า Index ของคำอักษรเบรลล์ภาษาไทยที่มีความคล้ายคลึงกับรูปแบบเอ็นแกรมมากที่สุดไว้ในตัวแปร max_text_index
4. ทำซ้ำจนกว่าจะครบรูปแบบเอ็นแกรมทั้งหมด

และเพื่อให้ง่ายต่อการทำความเข้าใจผู้วิจัยจึงใช้กรณีตัวอย่างในรูปที่ 3.12a และ 3.12b อธิบายวิธีการคิดคะแนนความคล้ายคลึงกันระหว่างรูปแบบเอ็นแกรมและรายการคำอักษรเบรลล์ภาษาไทยระดับ 1 ดังต่อไปนี้



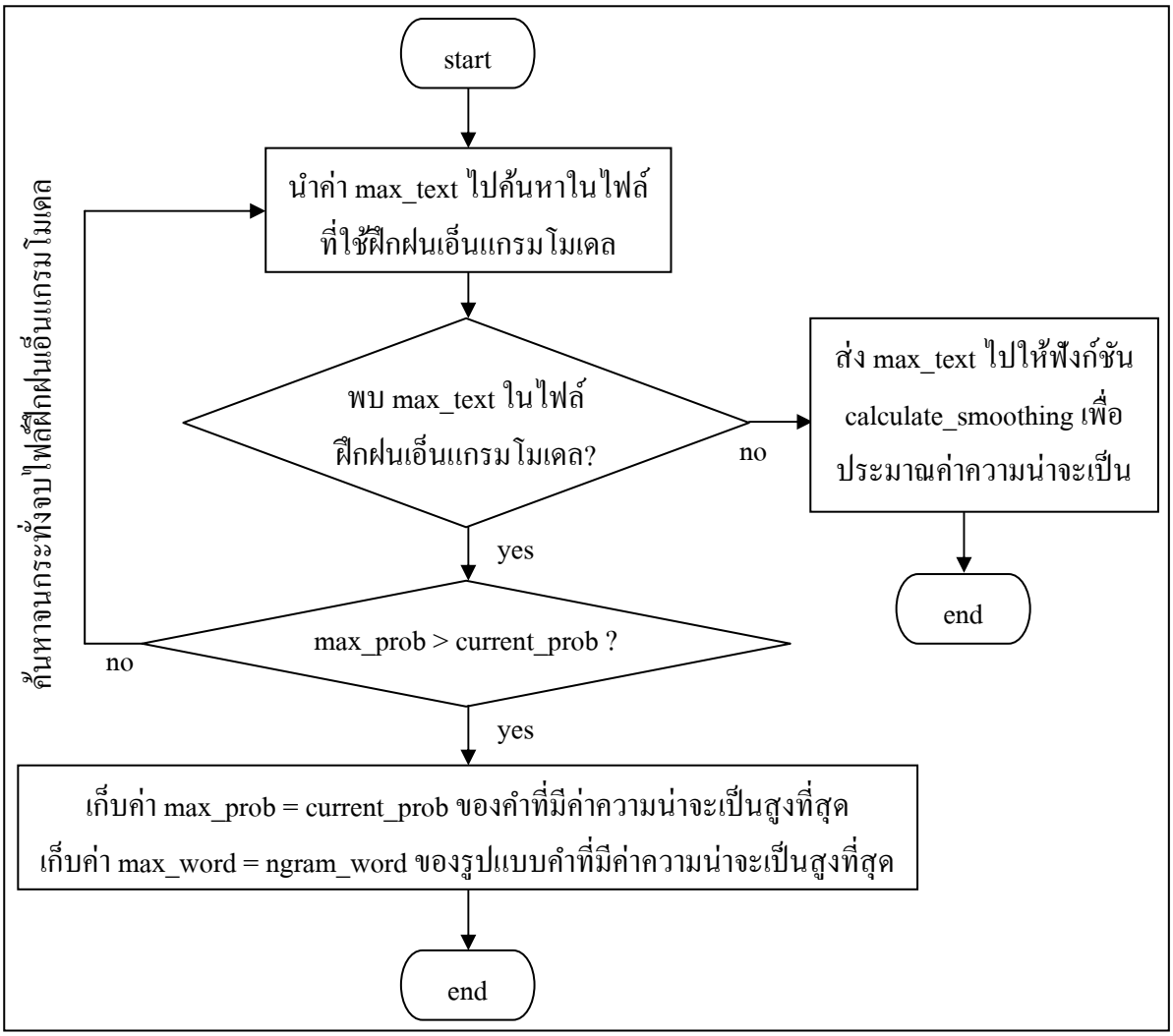
รูปที่ 3.12a กรณีที่มีการเรียงตัวอักขระเหมือนกันทั้งหมด

ในกรณีรูปที่ 3.12a นี้เป็นกรณีที่มีการเรียงอักขระของรูปแบบเอ็นแกรมและการเรียงอักขระของคำอักษรเบรลล์ภาษาไทยระดับ 1 ที่ได้สร้างเอาไว้แล้วเหมือนกันจนกระทั่งหมดความยาวของรูปแบบเอ็นแกรม ซึ่งจะคิดคะแนนตามจำนวนลักษณะการเรียงอักขระที่เหมือนกันและต่อเนื่องกันไปเรื่อยๆ กล่าวคือ เมื่ออักขระของรูปแบบเอ็นแกรมตำแหน่งแรกคือตัวอักขระ “g” ตรงกันกับอักขระตัวแรกของคำอักษรเบรลล์ภาษาไทยซึ่งเป็นตัวอักขระ “g” เช่นเดียวกันก็จะให้คะแนน 1 คะแนนและเก็บค่าคะแนนไว้ในตัวแปร score และอักขระตัวถัดไปซึ่งเป็นตัวอักขระ “,” ตรงกันอีกก็จะเพิ่มคะแนนอีก 1 คะแนนรวมเป็น 2 คะแนนและทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะพบอักขระที่เรียงไม่เหมือนกันจึงหยุดการเพิ่มคะแนน



รูปที่ 3.12b กรณีที่มีการเรียงตัวอักขระไม่เหมือนกันทั้งหมด

ในกรณีรูปที่ 3.12b นี้การเรียงตัวอักขระระหว่างรูปแบบเอ็นแกรมและคำอักษรเบรลล์ภาษาไทยเหมือนกันบางส่วน กล่าวคือตัวอักขระตัวแรกของรูปแบบเอ็นแกรมเป็นตัวอักษร “g” เช่นเดียวกันจึงให้คะแนนเพิ่ม 1 คะแนนและตัวอักขระตัวที่ 2 เป็นตัวอักษร “,” เช่นเดียวกันอีกก็เพิ่มคะแนนให้อีก 1 คะแนนรวมเป็น 2 คะแนนและตัวอักขระตัวที่ 3 ของรูปแบบเอ็นแกรมเป็นตัว “d” แต่ตัวอักขระตัวถัดไปของคำอักษรเบรลล์ภาษาไทยเป็นตัว “n” ซึ่งไม่ตรงกันจึงหยุดการเพิ่มคะแนนเพียงเท่านี้แม้ว่าตัวอักขระที่ 4 และ 5 ของคำอักษรเบรลล์ภาษาไทยจะตรงกันกับตัวอักษรที่เหลือของรูปแบบเอ็นแกรมก็ตาม



รูปที่ 3.13 ขั้นตอนการทำงานของฟังก์ชัน find_ngram

จากรูปที่ 3.13 แสดงขั้นตอนการทำงานของฟังก์ชัน find_ngram ซึ่งมีรายละเอียดดังต่อไปนี้

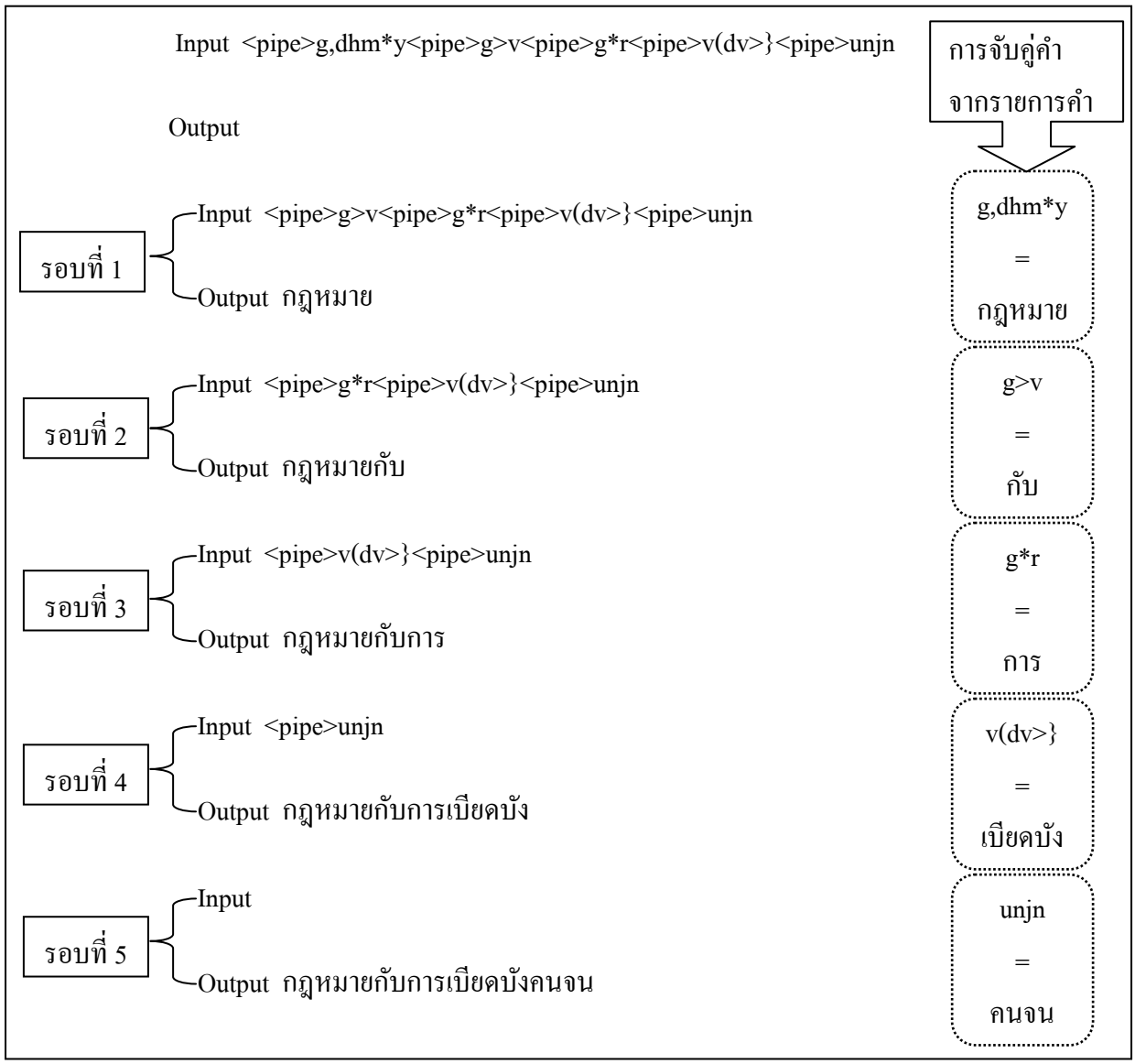
1. นำคำที่เก็บไว้ในตัวแปร max_text ไปค้นหาในรายการคำโมเดลอักษรเบรลล์ภาษาไทยซึ่งเก็บไว้ในตัวแปร array_ngram_word_list (รายละเอียดของรายการโมเดลอักษรเบรลล์ภาษาไทยจะอธิบายในหัวข้อที่ 3.7 เอ็นแกรมโมเดล)
2. หากพบคำ max_text ให้เก็บค่าความน่าจะเป็นของรูปแบบคำๆ นั้นในตัวแปร current_prob หากไม่พบให้ส่งคำ max_text ไปให้กับฟังก์ชัน calculate_smoothing เพื่อประมาณค่าความน่าจะเป็นของรูปแบบคำๆ นั้น จากนั้นจึงจบการทำงาน (รายละเอียดของการประมาณค่าความน่าจะเป็นจะอธิบายในหัวข้อที่ 3.8 Smoothing Techniques)

3. ค้นหาจนกระทั่งหมดรายการคำโมเดลอักษรเบรลล์ภาษาไทยเพื่อค้นหารูปแบบของคำที่มีค่าความน่าจะเป็นสูงที่สุดโดยเก็บไว้ในตัวแปร max_prob
4. เก็บค่าความน่าจะเป็นที่สูงที่สุดของรูปแบบคำไว้ในตัวแปร max_prob และเก็บรูปแบบคำที่มีค่าความน่าจะเป็นสูงที่สุดไว้ในตัวแปร max_word
5. ส่งค่า max_prob และ max_word กลับไปให้ฟังก์ชัน select_output เพื่อประกอบการตัดสินใจแบ่งคำต่อไป

3.6.4 ขั้นตอน translate_to_thai

ในขั้นตอน translate_to_thai นี้ทำหน้าที่แปลงจากประโยคอักษรเบรลล์ที่ได้ตัดคำแล้วให้อยู่ในรูปของประโยคภาษาไทย โดยใช้งานร่วมกันกับ รายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทย ซึ่งจะอธิบายรายละเอียดในหัวข้อที่ 3.9 รายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทย

ซึ่งวิธีการแปลงประโยคอักษรเบรลล์ภาษาไทยระดับ 1 ที่ได้ตัดคำไว้แล้วไปเป็นประโยคภาษาไทย จะใช้วิธีเทียบคำคือ แทนที่คำอักษรเบรลล์ภาษาไทยระดับ 1 ด้วยคำในภาษาไทยที่ตรงกัน โดยค้นหารายการของคำที่ตรงกันจากรายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทย ที่ได้สร้างไว้ก่อนแล้ว ดังแสดงในรูปที่ 3.14

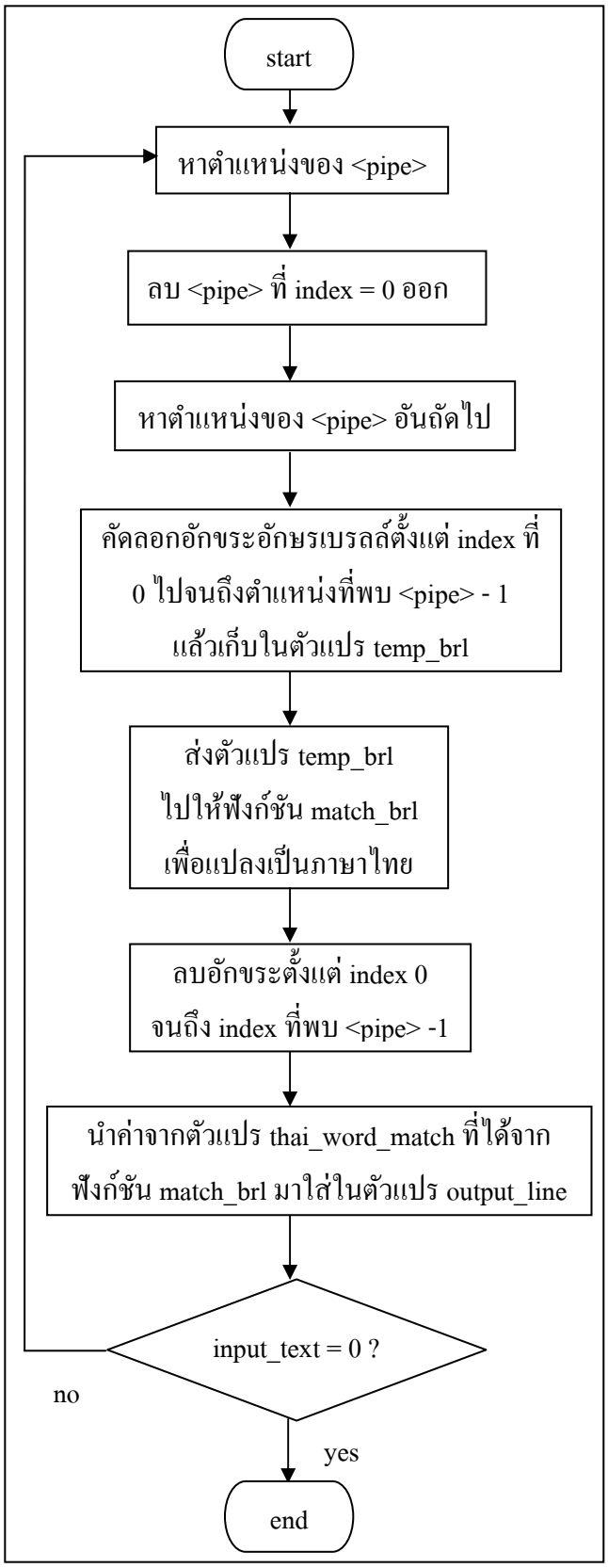


รูปที่ 3.14 ภาพรวมวิธีการแปลงแบบเทียบคำ

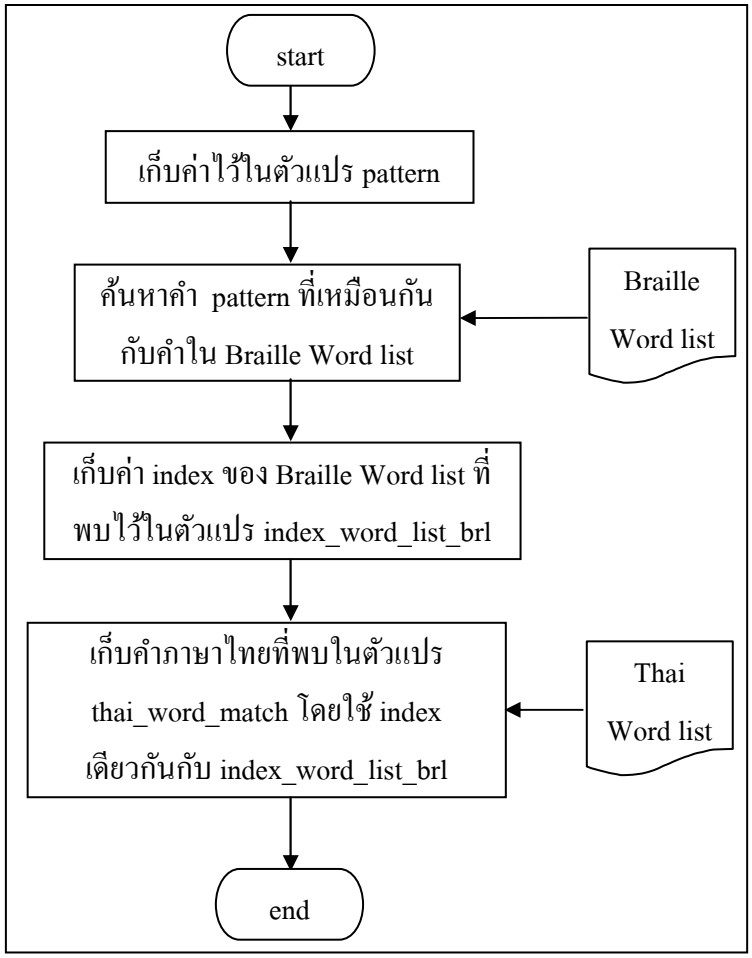
ในวิธีการแปลงแบบเทียบคำนี้ผู้วิจัยได้ออกแบบฟังก์ชันหลักที่ชื่อว่า mapping_translate เพื่อทำหน้าที่ในการแปลงแบบเทียบคำ ซึ่งมีการประกาศฟังก์ชันเป็น string mapping_translate (string text) โดยมีการรับ Parameter เป็นตัวแปรชนิด String คือประโยคอักษรเบรลล์ภาษาไทยระดับ 1 ที่ได้แบ่งคำไว้เรียบร้อยแล้วนั่นเอง และมีการคืนค่าเป็นชนิด String เช่นกันคือประโยคภาษาไทย โดยฟังก์ชัน mapping_translate มีรายละเอียดของตัวแปรชนิดต่างๆ ที่ใช้งานร่วมกันดังแสดงในตารางที่ 3.6 และมีขั้นตอนการทำงานดังแสดงในรูปที่ 3.15 และ 3.16 โดยในรูปที่ 3.15 เป็นฟังก์ชันหลักและรูปที่ 3.16 เป็นฟังก์ชัน match_brl ซึ่งทำหน้าที่ค้นหาคำอักษรเบรลล์ภาษาไทย

ตารางที่ 3.6 ตัวแปรต่างๆ ที่ใช้ในขั้นตอน translate_to_thai

ชื่อของตัวแปร	ชนิดของตัวแปร	หน้าที่ของตัวแปร
text	String	เก็บประโยคอักษรเบรลล์ภาษาไทยระดับ 1 ที่ตัดคำเรียบร้อยแล้ว
length_text	Integer	เก็บค่าดั้งเดิมความยาวของประโยคอักษรเบรลล์ภาษาไทยระดับ 1 ที่ตัดคำเรียบร้อยแล้ว
temp_length_text	Integer	เก็บค่าความยาวปัจจุบันของประโยคอักษรเบรลล์ภาษาไทยระดับ 1 ที่กำลังแปลงให้เป็นประโยคภาษาไทย
length_match_brl	Integer	เก็บค่าความยาวของคำอักษรเบรลล์ภาษาไทยปัจจุบันที่นำไปแปลงให้เป็นคำภาษาไทย
length_word_copy	Integer	เก็บค่าความยาวทั้งหมดของคำอักษรเบรลล์ภาษาไทยที่แปลงไปเป็นคำภาษาไทยแล้ว
start_index_copy	Integer	เก็บค่าตำแหน่งอักขระอักษรเบรลล์ภาษาไทยที่จะนำไปแปลงเป็นคำภาษาไทย
output_line	String	เก็บประโยคภาษาไทยที่แปลงมาจากประโยคอักษรเบรลล์ภาษาไทย
temp_brl	String	เก็บอักษรเบรลล์ภาษาไทยและส่งไปให้ฟังก์ชัน match_brl แปลงเป็นภาษาไทย
thai_word_match	Global String	เก็บคำภาษาไทยที่ค้นหาในรายการคำแล้วตรงกันกับคำอักษรเบรลล์ภาษาไทยที่ได้ส่งไปแปลง



รูปที่ 3.15 การทำงานของขั้นตอน translate_to_thai



รูปที่ 3.16 การทำงานของฟังก์ชัน match_brl

จากรูปที่ 3.15 และ 3.16 มีรายละเอียดการทำงานต่างๆ ดังต่อไปนี้

1. หาตำแหน่งของสัญลักษณ์ <pipe> จากประโยคอินพุต
2. ลบสัญลักษณ์ <pipe> ที่ตำแหน่งเริ่มต้นออก
3. หาตำแหน่งของสัญลักษณ์ <pipe> อันถัดไป
4. คัดลอกอักขระอักษรเบรลล์ที่แบ่งคำเสร็จแล้วตั้งแต่ตำแหน่งเริ่มต้นจนถึงตำแหน่งที่พบ <pipe> - 1 อันถัดไปและส่งให้กับฟังก์ชัน match_brl เพื่อค้นหาคำภาษาไทยที่ตรงกัน
5. นำคำภาษาไทยที่ค้นหาพบจากฟังก์ชัน match_brl ไปใส่ในตัวแปร output_line
6. ลบอักขระอักษรเบรลล์ตั้งแต่ตำแหน่งแรกจนถึงตำแหน่งที่พบ <pipe> - 1
7. ทำซ้ำตั้งแต่ข้อ 1-6 จนกว่าประโยคอินพุตจะหมดลง

3.7 การประยุกต์เอ็นแกรมโมเดล

เทคนิคเอ็นแกรมโมเดลเป็นเทคนิคทางด้านสถิติที่ใช้สร้างโมเดลภาษาและคำนวณหาค่าความน่าจะเป็นที่คำๆ หนึ่งสามารถเกิดร่วมกับคำๆ ใดได้บ้างด้วยค่าความน่าจะเป็นเท่าไหร่ (รายละเอียดด้านทฤษฎีได้อธิบายไว้ในหัวข้อที่ 2.5 เอ็นแกรมโมเดล) ซึ่งในงานวิจัยนี้ได้นำเทคนิคดังกล่าวมาประยุกต์ใช้ตัดคำอักษรเบรลล์ภาษาไทยก่อนที่จะแปลงไปเป็นภาษาไทยเพื่อลดความกำกวมอันเนื่องมาจากลักษณะไวยากรณ์การเขียนของอักษรเบรลล์ภาษาไทยระดับ 1 ที่เขียนคำแต่ละคำติดกันต่อเนื่องกันไปและใช้การเว้นวรรคเพื่อแบ่งแต่ละประโยคออกจากกันแต่จะมีการเรียงอักษรนำ อักษรตาม สระ วรรณยุกต์และตัวสะกดที่แตกต่างไปจากลักษณะไวยากรณ์ของภาษาไทย โดยวิธีการไฟล์โมเดลอักษรเบรลล์ภาษาไทยหรือการฝึกฝนเอ็นแกรมโมเดลได้อธิบายไว้ในหัวข้อที่ 3.3.2 และรายละเอียดของเครื่องมือที่ใช้ได้อธิบายไว้ในหัวข้อที่ 2.7 CMU-SLM toolkit

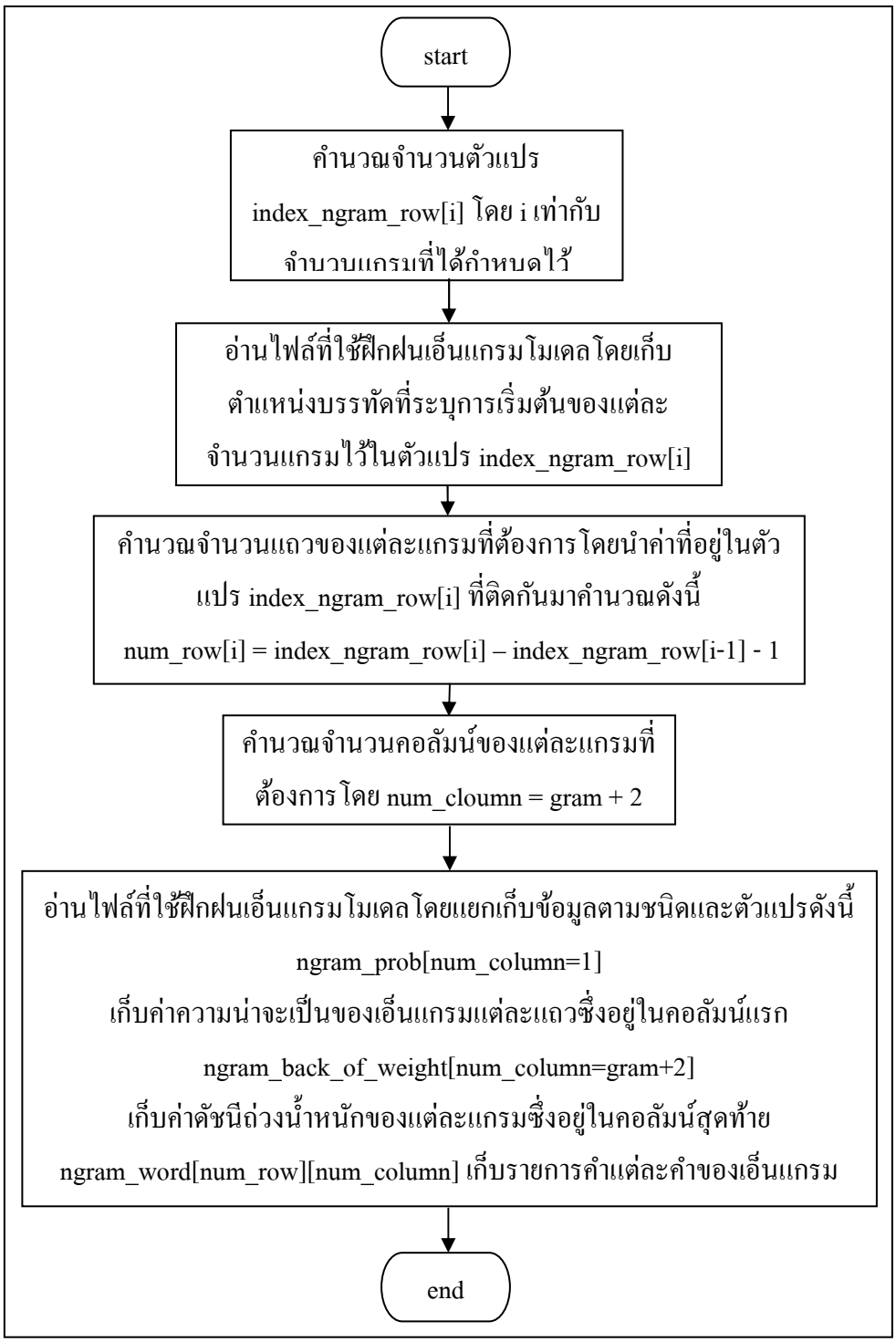
โดยในหัวข้อนี้จะกล่าวถึงวิธีการจัดเตรียมข้อมูลดังกล่าวและจัดเก็บไว้ในหน่วยความจำหลักและการนำไปใช้งาน ซึ่งผู้วิจัยได้ออกแบบฟังก์ชันชื่อว่า load_ngram_word ซึ่งจะทำหน้าที่อ่านไฟล์ที่ฝึกฝนเอ็นแกรมโมเดลและประมวลผลเพื่อแยกชนิดของข้อมูลและเก็บข้อมูลดังกล่าวไปไว้ในหน่วยความจำหลัก โดยมีการประกาศฟังก์ชันเป็น void load_ngram_word() โดยมีรายละเอียดของตัวแปรต่างๆ ที่ใช้ในฟังก์ชันนี้ดังแสดงในตารางที่ 3.7

ตารางที่ 3.7 ตัวแปรต่างๆ ที่ใช้ในฟังก์ชัน load_ngram_word

ชื่อของตัวแปร	ชนิดของตัวแปร	หน้าที่ของตัวแปร
index_ngram_row	Array of Integer	เก็บตำแหน่งบรรทัดของแต่ละแกรม
num_row	Array of Integer	เก็บจำนวนแถวของแต่ละแกรม
num_column	Array of Integer	เก็บจำนวนคอลัมน์ของแต่ละแกรม
row	Global Integer	เก็บ index ของแถว
column	Global Integer	เก็บ index ของคอลัมน์
ngram_word	Global 2D Array of String	เก็บรายการคำของเอ็นแกรม
ngram_prob	Global Double	เก็บค่าความน่าจะเป็นของเอ็นแกรม
ngram_back_of_weight	Global Double	เก็บค่าดัชนีถ่วงน้ำหนักของเอ็นแกรม

ผู้วิจัยได้ออกแบบการทำงานของฟังก์ชัน load_ngram_word เป็น 2 ช่วง ดังนี้ 1) อ่านข้อมูลในไฟล์ฝึกฝนเอ็นแกรมโมเดลเพื่อคำนวณจำนวนแถวและคอลัมน์ของแต่ละแกรมเพื่อจองขนาดของตัวแปรที่ใช้เก็บข้อมูลดังกล่าวได้ถูกต้อง 2) อ่านข้อมูลดังกล่าวและจัดเก็บไว้ในตัวแปรแต่ละ

ประเภทตามที่ได้ออกแบบไว้ โดยฟังก์ชัน load_ngram_word มีขั้นตอนการทำงานดังแสดงในรูปที่ 3.17



รูปที่ 3.17 ขั้นตอนการทำงานของฟังก์ชัน load_ngram_word

จากรูปที่ 3.17 มีรายละเอียดขั้นตอนการทำงานต่างๆ ดังต่อไปนี้

1. คำนวณจำนวนตัวแปรที่ใช้เก็บตำแหน่งเริ่มต้นของแต่ละแกรมซึ่งมีจำนวนเท่ากับขนาดของจำนวนแกรมที่ได้กำหนดไว้ กล่าวคือ หากกำหนดขนาดแกรมไว้ที่ 3-แกรม ก็ต้องกำหนดจำนวนตัวแปร index_ngram_row[gram] ให้มีจำนวนเท่ากับ 3 ชุดด้วยเพื่อที่จะเก็บตำแหน่งเริ่มต้นของแกรมคือ 1-gram, 2-gram และ 3-gram ว่าอยู่ที่บรรทัดไหนบ้าง เป็นต้น
2. อ่านไฟล์โมเดลอักษรเบรลล์ภาษาไทยโดยเก็บตำแหน่งเริ่มต้นของแต่ละแกรมไว้ในตัวแปร index_ngram_row[gram] โดยสามารถตรวจสอบได้จากการเปรียบเทียบคำ กล่าวคือ ตำแหน่งเริ่มต้นของ 1-แกรม จะมีคำกำกับเป็น \1-grams: และตำแหน่งเริ่มต้นของ 2-แกรม จะมีคำกำกับเป็น \2-grams: เป็นลักษณะเช่นนี้ไปเรื่อยๆ จนกระทั่งครบตามจำนวนแกรมที่ได้กำหนดไว้ ซึ่งจะทำให้ทราบได้ว่าตำแหน่งเริ่มต้นของแต่ละแกรมอยู่ที่บรรทัดไหนบ้าง และเก็บตำแหน่งบรรทัดเหล่านั้นไว้ในตัวแปร index_ngram_row[gram]
3. คำนวณจำนวนแถวของแต่ละแกรมที่ต้องการโดยจัดเก็บไว้ในตัวแปร num_row[i] ซึ่งคำนวณได้จาก num_row[i] = index_ngram_row[i] – index_ngram_row[i-1] – 1 ซึ่งจะได้จำนวนแถวที่ต้องการเพื่อจัดเก็บรูปแบบเอ็นแกรมของแต่ละขนาดแกรมตั้งแต่ 1-แกรม ไปจนถึงขนาดแกรมที่ได้กำหนดไว้
4. คำนวณจำนวนคอลัมน์ของแต่ละรูปแบบเอ็นแกรมที่ต้องการโดยจัดเก็บไว้ในตัวแปร num_column[i] ซึ่งคำนวณได้จาก num_column[i] = gram + 2 คือในแต่ละแกรมจะมีคอลัมน์ที่เพิ่มขึ้นมาสำหรับจัดเก็บค่าความน่าจะเป็นของแต่ละรูปแบบเอ็นแกรมและค่าดัชนีถ่วงน้ำหนักของแต่ละรูปแบบเอ็นแกรม (ตามที่ได้แสดงไว้ในรูปที่ 3.18) ยกเว้นจำนวนแกรมสุดท้ายตามที่ได้กำหนดไว้ซึ่งในตัวอย่างนี้ได้กำหนดขนาดของแกรมเท่ากับ 3-แกรม โดยจำนวนแกรมสุดท้ายที่ได้กำหนดไว้ก็คือจำนวนแกรมที่ 3-gram ซึ่งจะมีเพิ่มขึ้นมาอีกเพียง 1 คอลัมน์เท่านั้นสำหรับจัดเก็บค่าความน่าจะเป็นของแต่ละรูปแบบเอ็นแกรม
5. อ่านไฟล์โมเดลอักษรเบรลล์ภาษาไทยโดยแยกจัดเก็บข้อมูลชนิดต่างๆ ลงในตัวแปรแต่ละตัวแปรตามที่ได้กำหนดไว้ กล่าวคือ
   5.1. เก็บค่าความน่าจะเป็นของแต่ละรูปแบบเอ็นแกรมแต่ละแถวซึ่งอยู่ในคอลัมน์แรกเสมอไว้ในตัวแปร ngram_prob[num_column=1]

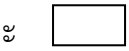
5.2. เก็บค่าดัชนีถ่วงน้ำหนักของแต่ละรูปแบบเอ็นแกรมในแต่ละแถวซึ่งจะอยู่ที่คอลัมน์สุดท้ายเสมอ ไว้ในตัวแปร ngram_back_of_weight[num_column=gram+2] ยกเว้นจำนวนแกรมสุดท้ายที่ไม่มีค่าดัชนีถ่วงน้ำหนัก

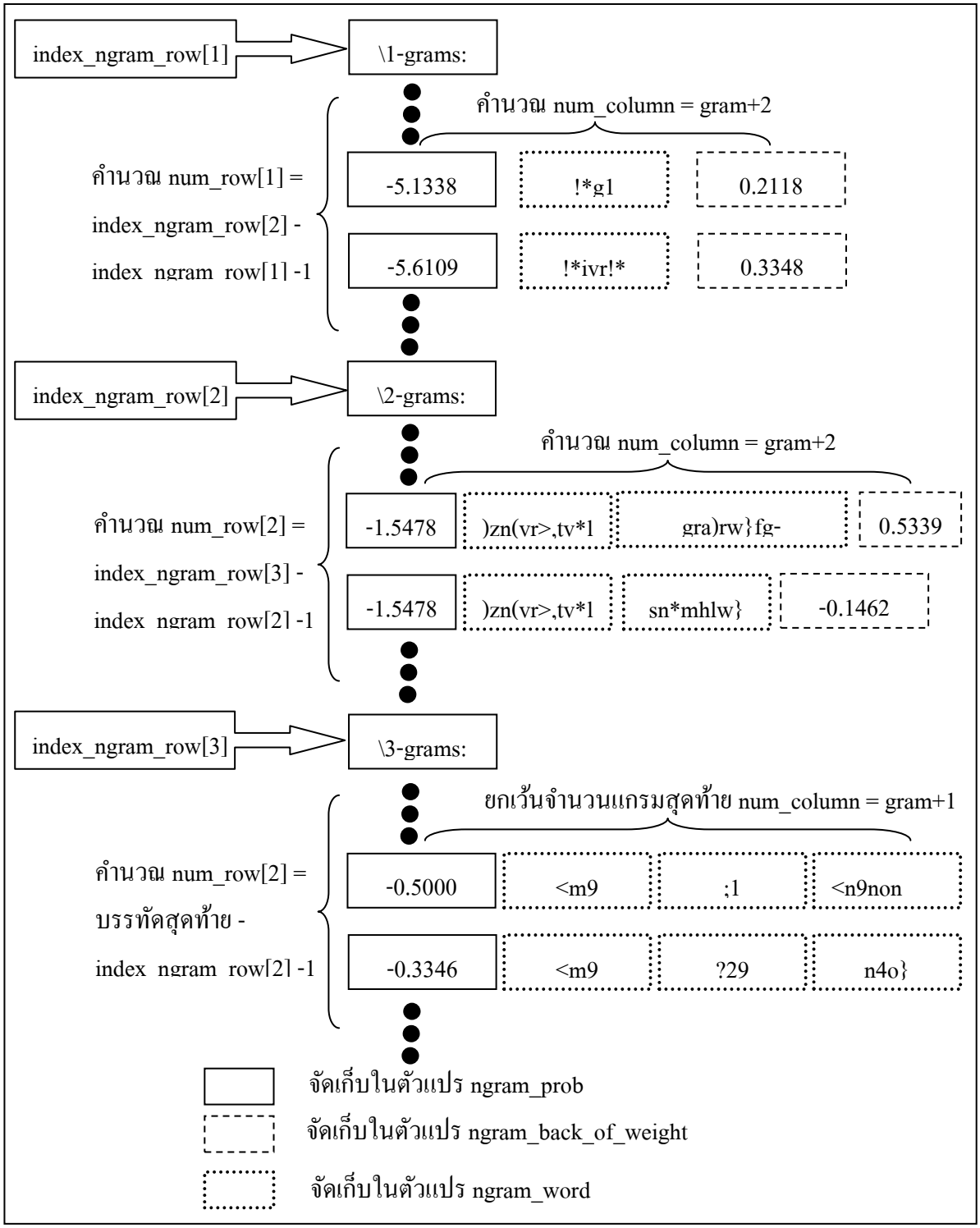
5.3. เก็บรายการคำของแต่ละเอ็นแกรมในตัวแปร ngram_word[num_row][num_column] โดยแบ่งตามแถวและคอลัมน์ตามที่ได้ออกแบบไว้

6. จบการของฟังก์ชันนี้โดยออกแบบให้ฟังก์ชันเริ่มต้นการทำงานพร้อมกับการเริ่มโปรแกรมหลัก

และเพื่อเป็นการง่ายต่อการทำความเข้าใจผู้วิจัยจึงใช้แผนภาพเพื่ออธิบายการจัดเก็บข้อมูลจากไฟล์โมเดลอักษรเบรลล์ภาษาไทยลงในตัวแปรประเภทต่างๆ ดังแสดงในรูปที่ 3.18 โดยสมมุติว่ากำหนดจำนวนแกรมไว้เป็น 3-แกรม

โดยมีการแบ่งการจัดเก็บข้อมูลชนิดต่างๆ ในตัวแปรประเภทต่างๆ ดังต่อไปนี้

1. เก็บลำดับที่บรรทัดของตำแหน่งเริ่มต้นของแต่ละแกรมไว้ในตัวแปร index_ngram_row[i] โดยใช้วิธีค้นหาจากการเปรียบเทียบคำคือ ตำแหน่งเริ่มต้นของแต่ละแกรมจะมีลักษณะเป็น \1-grams:, \2-grams: เป็นเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงจำนวนแกรมที่ได้กำหนดไว้
2. เก็บค่าความน่าจะเป็นของแต่ละรูปแบบเอ็นแกรมไว้ในตัวแปร ngram_prob ซึ่งจะอยู่ในคอลัมน์แรกของแต่ละแถวที่ไม่ใช่แถวของตำแหน่งเริ่มต้นของแต่ละแกรม ซึ่งจากรูปที่ 3.18 คือค่าตัวเลขที่แสดงอยู่ในกรอบลักษณะ 
3. เก็บค่าดัชนีถ่วงน้ำหนัก (Back-off weight) ของแต่ละรูปแบบเอ็นแกรมไว้ในตัวแปร ngram_back_of_weight ซึ่งจะอยู่คอลัมน์ท้ายสุดของแต่ละแถวเสมอยกเว้นจำนวนแกรมสุดท้ายจะไม่มีค่าดัชนีถ่วงน้ำหนัก ซึ่งจากรูปที่ 3.18 คือค่าตัวเลขที่แสดงอยู่ในกรอบลักษณะ
4. เก็บค่าคำของแต่ละรูปแบบเอ็นแกรมซึ่งเป็นคำแต่ละคำที่เขียนอยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 โดยเก็บไว้ในตัวแปร ngram_word[row][column] ตามตำแหน่งของแถวและคอลัมน์ที่ได้กำหนดไว้ ซึ่งจากรูปที่ 3.18 คือตัวอักษรเบรลล์ที่แสดงอยู่ในกรอบลักษณะ



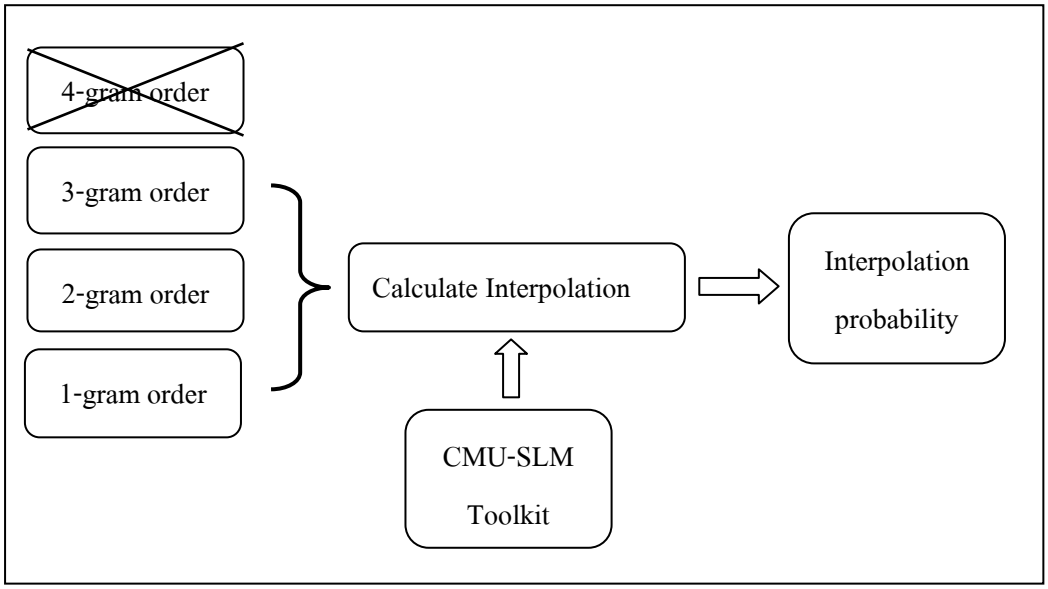
รูปที่ 3.18 วิธีการจัดเก็บข้อมูลในตัวแปรประเภทต่างๆ

3.8 Smoothing Techniques ที่เลือกใช้

เทคนิค Smoothing Technique ที่เลือกใช้ในวิทยานิพนธ์นี้ใช้วิธีการของ Jelinek-Mercer Smoothing (Interpolation) ในการประมาณค่าความน่าจะเป็น ซึ่งวิธีการแบบ Interpolation คือการนำเอาค่าความน่าจะเป็นของจำนวนแกรมอื่นๆ มาคำนวณร่วมกันเพื่อประมาณค่าความน่าจะเป็น โดยมีสมการคำนวณค่าความน่าจะเป็นดังสมการที่ 5

$$P_{interp}\left(w_i \mid w_{i-n+1}^{i-1}\right) = \lambda_{w_{i-n+1}^{i-1}} P_{ML}\left(w_i \mid w_{i-n+1}^{i-1}\right) + \left(1-\lambda_{w_{i-n+1}^{i-1}}\right) P_{interp}\left(w_i \mid w_{i-n+2}^{i-1}\right) \quad \text{..... (5)}$$

และเพื่อให้ง่ายต่อการทำความเข้าใจผู้วิจัยจึงใช้แผนภาพในการอธิบายดังแสดงในรูปที่ 3.19



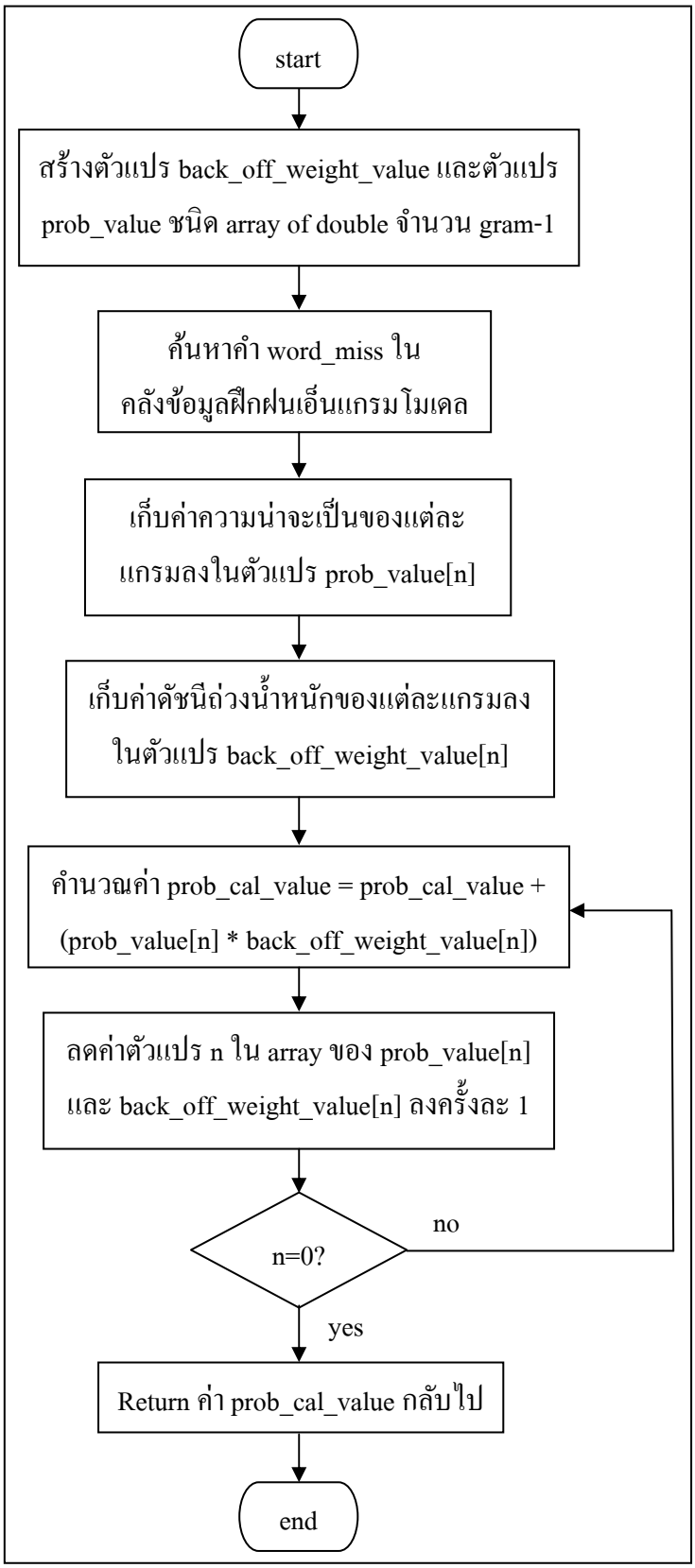
รูปที่ 3.19 ภาพรวมการทำงานของ Interpolation

จากรูปที่ 3.19 สมมุติว่ากำหนดการฝึกฝนเอ็นแกรมโมเดลด้วยจำนวนแกรมเท่ากับ 4-แกรม และในคลังข้อมูลฝึกไม่พบคำที่ต้องการเมื่อค้นหาที่จำนวนแกรมเท่ากับ 4-แกรม ดังนั้นจึงใช้วิธีการประมาณค่าความน่าจะเป็นแบบ Interpolation โดยใช้ชุดเครื่องมือ CMU-SLM Toolkit ประมาณค่าความน่าจะเป็นโดยการใช้จำนวนแกรมที่ถัดไปเรื่อยๆ จนถึง 1-แกรมมาคำนวณร่วมกัน กล่าวคือค่าความน่าจะเป็นที่ได้จากการประมาณค่าของคำที่ไม่พบในจำนวนแกรมเท่ากับ 4-แกรม มีค่าเท่ากับค่า Back-off weight ของ 3-แกรมคูณด้วยค่าความน่าจะเป็นของจำนวนแกรม 3-แกรม บวกด้วย ค่า Back-off weight ของ 2-แกรมคูณด้วยค่าความน่าจะเป็นของจำนวนแกรม 2-แกรม บวกด้วย ค่า Back-off weight ของ 1-แกรมคูณด้วยค่าความน่าจะเป็นของจำนวนแกรม 1-แกรม

และในวิทยานิพนธ์นี้ผู้วิจัยได้สร้างฟังก์ชัน calculate_smoothing เพื่อประมาณค่าความน่าจะเป็นในกรณีที่ไม่พบคำหรือชุดของคำในคลังข้อมูลฝึก โดยมีการประกาศฟังก์ชันเป็น double calculate_smooth (string word) โดยมีการรับพารามิเตอร์ (Parameter) เป็นตัวแปรชนิด String คือ คำอักษรเบรลล์ภาษาไทยระดับ 1 ที่ไม่พบในคลังข้อมูลฝึกและมีการคืนค่าเป็นชนิด Double คือค่าความน่าจะเป็นที่ได้จากการประมาณค่าโดยฟังก์ชัน calculate_smoothing และมีรายละเอียดของตัวแปรชนิดต่างๆ ที่ใช้งานร่วมกันดังแสดงในตารางที่ 3.8 และมีขั้นตอนการทำงานแสดงในรูปที่ 3.20

ตารางที่ 3.8 ตัวแปรต่างๆ ที่ใช้ในขั้นตอน calculate_smoothing

ชนิดของตัวแปร	ชื่อของตัวแปร	หน้าที่ของตัวแปร
Double	prob_cal_value	เก็บค่าความน่าจะเป็นที่ได้จากการประมาณค่า
String	word_miss	เก็บค่าของคำที่ไม่พบในคลังข้อมูลฝึก
Array of Double	back_off_weight_value	เก็บค่าดัชนีถ่วงน้ำหนักของแต่ละจำนวนแกรม
Array of Double	prob_value	เก็บค่าความน่าจะเป็นของแต่ละจำนวนแกรม



รูปที่ 3.20 การทำงานของฟังก์ชัน calculate_smoothing

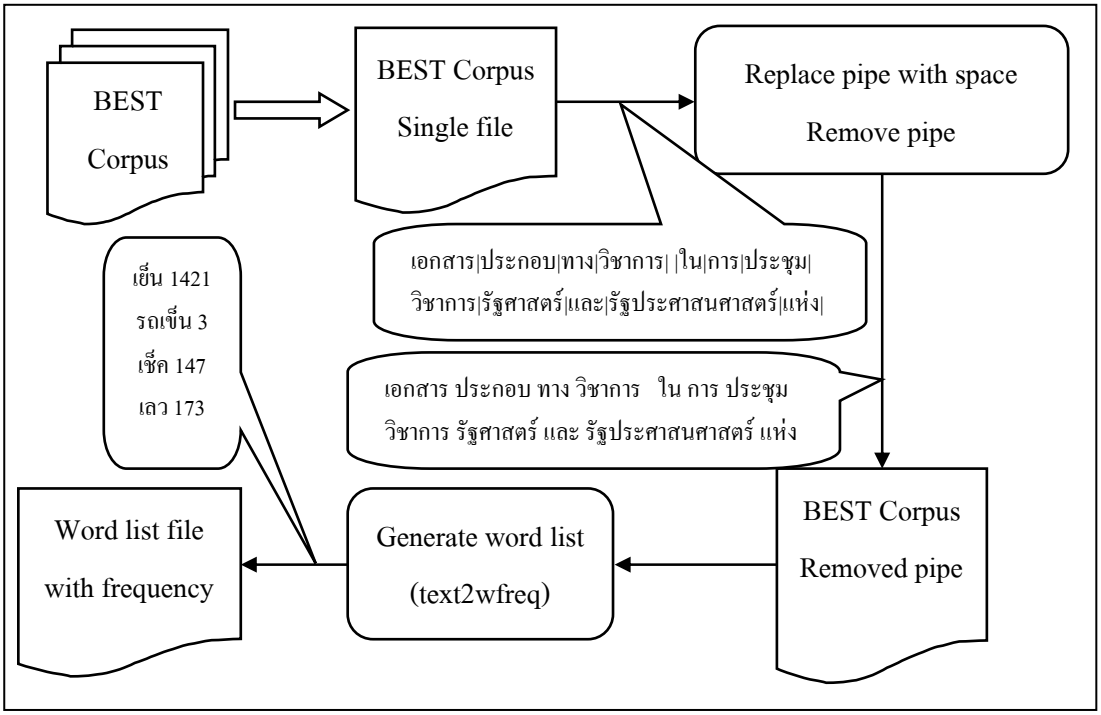
จากรูปที่ 3.20 มีรายละเอียดการทำงานต่างๆ ดังต่อไปนี้

1. สร้างตัวแปร back_off_weight_value และ prob_value ชนิด Array of Double จำนวน gram-1 ตามที่ได้กำหนดไว้
2. ค้นหารูปแบบคำที่ไม่พบในคลังข้อมูลฝึกในขนาดแกรมที่ gram-1, gram-2 ไปเรื่อยๆ จนถึงขนาด 1-gram
3. เก็บค่าดัชนีถ่วงน้ำหนัก (Back-off weight) ของแต่ละแกรมลงในตัวแปร back_off_weight_value[n] และเก็บค่าความน่าจะเป็นของแต่ละแกรมลงในตัวแปร prob_value[n]
4. คำนวณค่าความน่าจะเป็นแบบ Interpolation โดยนำผลรวมของค่าดัชนีถ่วงน้ำหนัก (Back-off weight) ของแต่ละแกรมคูณด้วยค่าความน่าจะเป็นของแต่ละแกรมและเก็บไว้ในตัวแปร prob_cal_value และส่งคืนค่าความน่าจะเป็นแบบ Interpolation กลับคืนไปให้คำส่งที่เรียกใช้ฟังก์ชันนี้

3.9 รายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทย

รายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยที่ผู้วิจัยได้จัดเตรียมขึ้นมานี้จะนำไปใช้ในขั้นตอน Translate_to_Thai และเนื่องจากเอ็นแกรมโมเดลนั้นสามารถนำไปใช้ในด้านการหาขอบเขตของคำเท่านั้น ไม่สามารถนำไปใช้แปลงอักษรเบรลล์ภาษาไทยให้เป็นภาษาไทยได้ นอกจากนี้ผู้วิจัยต้องการเปรียบเทียบประสิทธิภาพระหว่างโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file [1] เดิม กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมช่วยในการตัดคำที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ ซึ่งโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิมนั้นจะให้เอาท์พุทเป็นข้อความภาษาไทยทำให้จำเป็นต้องมีขั้นตอน Translate_to_Thai เพิ่มเติมเข้ามาเพื่อแปลงให้เอาท์พุทเป็นข้อความภาษาไทยเหมือนกันเพื่อเปรียบเทียบกันได้โดยตรง

ดังนั้นผู้วิจัยจึงออกแบบให้การแปลงจากอักษรเบรลล์ภาษาไทยเป็นภาษาไทยใช้วิธีการแบบเทียบเป็นคำต่อคำ เช่น เมื่อตัดคำอักษรเบรลล์ภาษาไทยได้เป็นคำว่า “g,dhm*y” ซึ่งตรงกันกับคำว่า “กฎหมาย” ในภาษาไทย เมื่อใช้วิธีเทียบเป็นคำต่อคำก็จะแทนที่คำ “g,dhm*y” ด้วยคำว่า “กฎหมาย” ลงไปแทนในตำแหน่งเดียวกันของข้อความ ซึ่งเมื่อใช้วิธีการนี้แล้วทำให้จำเป็นต้องสร้างทั้งรายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยขึ้นมาเพื่อที่จะนำไปใช้กับวิธีการเทียบเป็นคำต่อคำ โดยขั้นตอนการสร้างรายการคำภาษาไทยแสดงไว้ในรูปที่ 3.21



รูปที่ 3.21 ขั้นตอนการสร้างรายการคำภาษาไทย

จากรูปที่ 3.21 มีรายละเอียดวิธีการสร้างรายการคำภาษาไทยเพื่อนำไปใช้ในขั้นตอน Translate_to_Thai ดังต่อไปนี้

1. นำไฟล์คลังข้อความ BEST ทั้งหมดรวมเข้าเป็นไฟล์ๆ เดียวกัน เพื่อความสะดวกต่อการนำไปใช้งานและเพื่อใช้ในขั้นตอนการสร้างรายการคำ (Generate word list)
2. เนื่องจากข้อจำกัดของโปรแกรม text2wfreq ที่อยู่ในชุดเครื่องมือ CMU-SLM toolkit ไม่สามารถประมวลผลภาษาไทยได้โดยตรง ทำให้ต้องมีการดัดแปลงเครื่องหมายคั่นระหว่างคำเป็นช่องว่าง (Space) แทนเพื่อให้เหมือนกับการเขียนคำในภาษาอังกฤษที่ใช้ช่องว่างเป็นตัวบ่งบอกขอบเขตของคำ ดังนั้นผู้วิจัยจึงได้แทนที่เครื่องหมายคั่นระหว่างคำ (“|”) ที่ใช้ในคลังข้อความ BEST ด้วยช่องว่างแทน
3. สร้างรายการคำโดยเรียกใช้โปรแกรม text2wfreq ด้วยคำสั่ง “text2wfreq -hash 1000000 <input.text> output.wfreq” โดย text2wfreq เป็นโปรแกรมที่ใช้สร้างรายการคำและออฟชั่น –hash 1000000 หมายถึงให้จองพื้นที่หน่วยความจำไว้สำหรับการสร้าง

Hash-table เพื่อใช้ในการจัดเก็บรายการคำ ถัดไปจะเป็นชื่อของไฟล์อินพุตและชื่อของไฟล์เอาท์พุทตามลำดับ โดยนามสกุลของไฟล์เอาท์พุทจะเป็น *.wfreq ซึ่งก็คือรายการความถี่ของคำนั่นเอง

เมื่อได้รายการคำภาษาไทยเรียบร้อยแล้วจึงนำรายการคำดังกล่าวไปแปลงให้อยู่ในรูปแบบของอักษรเบรลล์ภาษาไทยระดับ 1 โดยใช้โปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์ [1] (รายละเอียดของโปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์จะอธิบายในหัวข้อ 2.8) ซึ่งให้เอาท์พุทเป็นข้อความที่อยู่ในรูปแบบอักษรเบรลล์ภาษาไทย โดยใช้คำสั่ง Thai2brl <[Input file] >[OutputFile] –g 1 ใน Shell prompt โดยโปรแกรม Thai2brl เป็นโปรแกรมที่ทำหน้าที่แปลงจากภาษาไทยเป็นอักษรเบรลล์ภาษาไทย และถัดไปเป็นชื่ออินพุตไฟล์และเอาท์พุทไฟล์ตามลำดับ และออฟชั่น –g 1 หมายถึงระบุให้เอาท์พุทอยู่ในรูปแบบของอักษรเบรลล์ภาษาไทยระดับ 1

เมื่อสร้างทั้งรายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยแล้ว จึงนำไปใช้งานในโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ได้ออกแบบไว้ต่อไป โดยผู้วิจัยได้ออกแบบให้จัดเก็บรายการคำดังกล่าวไว้ในหน่วยความจำ โดยมีรายละเอียดต่างๆ ดังแสดงในตารางที่ 3.9 ดังนี้

ตารางที่ 3.9 ตัวแปรชนิดต่างๆ ที่ใช้ในการสร้างรายการคำ

รายละเอียด/ประเภทรายการคำ	รายการคำอักษรเบรลล์ภาษาไทย	รายการคำภาษาไทย
ชื่อฟังก์ชันที่ใช้โหลดข้อมูล	void load_braille_word_list()	void load_word_list()
ประเภทข้อมูลที่ใช้จัดเก็บ	global array of string	global array of string
ชื่อข้อมูลที่ใช้จัดเก็บ	word_list_brl[]	word_list[]
ประเภทข้อมูลที่จัดเก็บดัชนี	global integer	global integer
ชื่อข้อมูลที่ใช้จัดเก็บดัชนี	index_word_list_brl	index_word_list
เรียกใช้งานฟังก์ชันเมื่อ	เริ่มโปรแกรมหลัก	เริ่มโปรแกรมหลัก

ในตารางที่ 3.9 จะเห็นได้ว่าลักษณะของฟังก์ชันและข้อมูลที่ใช้จัดเก็บรายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยนั้นจะคล้ายคลึงกัน แต่จะแตกต่างกันเพียงแค่ลักษณะข้อมูลที่ถูกจัดเก็บไว้ในหน่วยความจำเท่านั้น ซึ่งก็คือเป็นรายการคำอักษรเบรลล์ภาษาไทยกับรายการคำภาษาไทยเท่านั้นที่แตกต่างกัน

รายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยนี้ผู้วิจัยได้ออกแบบให้มีฟังก์ชันสำหรับจัดเก็บรายการคำดังกล่าวให้อยู่ในหน่วยความจำหลัก โดยมีฟังก์ชันชื่อว่า load_braille_word_list และ load_word_list ทำหน้าที่โหลดรายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยไปไว้ในหน่วยความจำหลัก ทั้งนี้เพื่อความสะดวกรวดเร็วในการเรียกใช้เมื่อตัดคำอักษรเบรลล์ภาษาไทยในแต่ละประโยคเสร็จสิ้น และทั้งสองฟังก์ชันนี้จะเรียกใช้เมื่อมีการเริ่มโปรแกรมหลักเหมือนกัน

ในบทถัดไปจะกล่าวถึงผลการทดลองโดยเป็นการเปรียบเทียบประสิทธิภาพด้านความถูกต้องของการแปลงอักษรเบรลล์ภาษาไทยระดับ 1 ไปเป็นภาษาไทย, ด้านปริมาณการใช้หน่วยความจำในระหว่างการแปลงและด้านเวลาที่ใช้ในการแปลง ระหว่างโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file (วิธีการเดิม) กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ (วิธีการใหม่) ซึ่งได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้

บทที่ 4

ผลการทดลอง

ในบทนี้ได้นำเสนอผลการทดลองที่ได้ออกแบบไว้ ซึ่งสอดคล้องกันกับวัตถุประสงค์ของวิทยานิพนธ์ที่ได้นำเสนอโดยมี 3 ประการด้วยกันคือ 1) ศึกษาและพัฒนาการนำเทคนิคเอ็นแกรมเข้ามาช่วยแบ่งคำอักษรเบรลล์ภาษาไทยและแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยที่เกี่ยวข้องกับคำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์ 2) เปรียบเทียบประสิทธิภาพในด้านความถูกต้อง, ประมาณการใช้หน่วยความจำและเวลาที่ใช้ในการแปลงระหว่างโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file (วิธีการเดิม) กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ (วิธีการใหม่) 3) จำนวนแกรมที่เหมาะสมสำหรับตัดคำอักษรเบรลล์ภาษาไทยระดับ 1 ทั้งด้านความถูกต้อง, ปริมาณการใช้หน่วยความจำและระยะเวลาที่ใช้ในการแปลง

ผู้วิจัยได้ออกแบบการทดลองทั้งหมดเป็น 3 ด้านเพื่อให้สอดคล้องกันกับจุดประสงค์ดังกล่าวข้างต้น ซึ่งประกอบไปด้วย 1) ด้านความถูกต้องของการแปลง 2) ด้านปริมาณการใช้หน่วยความจำในการแปลง และ 3) ด้านระยะเวลาที่ใช้ในการแปลง โดยในแต่ละด้านมีรายละเอียดดังนี้

4.1 ด้านความถูกต้องของการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย

4.1.1 เครื่องมือที่ใช้ในการทดสอบ

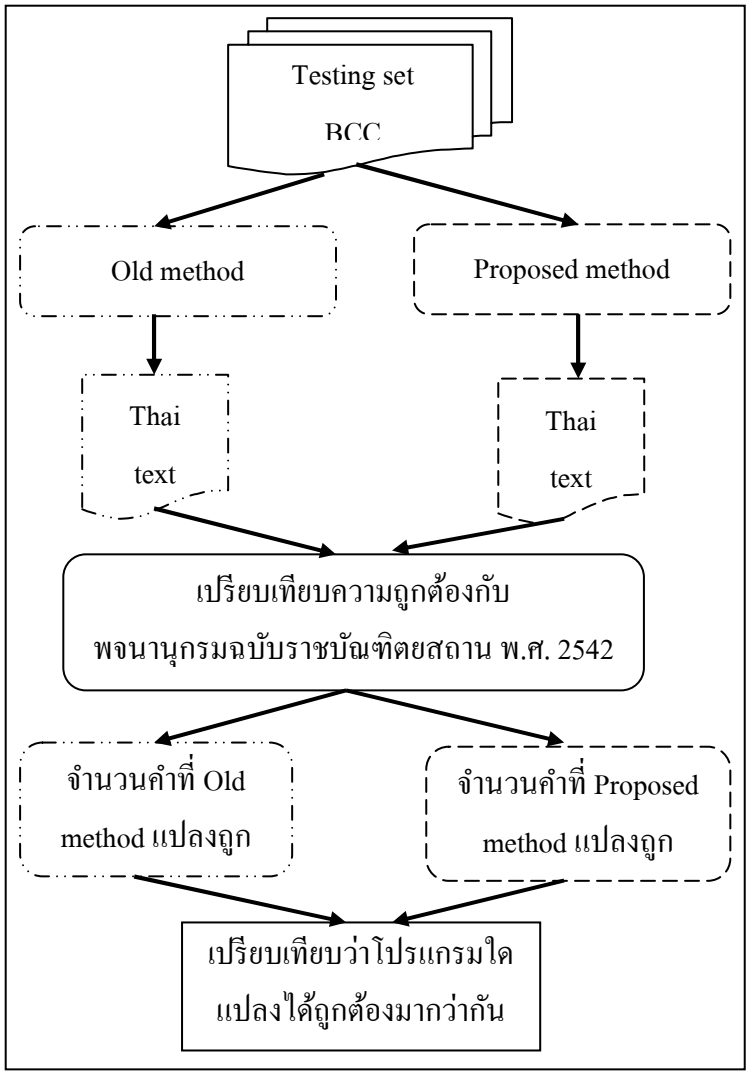
1) โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file [1]
2) โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ
3) ชุดข้อมูลทดสอบในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 (รายละเอียดของชุดข้อมูลทดสอบจะอธิบายในหัวข้อ 4.1.3)
4) เครื่องคอมพิวเตอร์ติดตั้งระบบปฏิบัติการ Linux Ubuntu version 9.04 1 เครื่อง ติดตั้ง CPU Intel รุ่น Core2Quad (Q9400) ความเร็ว 2.66 GHz ติดตั้งหน่วยหน่วยความจำที่ใช้งานได้ 3.25 GB

5) C/C++ compiler

6) Gedit Text editor

4.1.2 ภาพรวมการทดสอบด้านความถูกต้องการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย

ผู้วิจัยได้ออกแบบวิธีการทดสอบเปรียบเทียบความถูกต้องการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยระหว่างโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ซึ่งได้นำเสนอไว้ในวิทยานิพนธ์นี้ โดยได้ใช้แผนภาพเพื่ออธิบายซึ่งง่ายต่อการทำความเข้าใจ โดยภาพรวมวิธีการทดสอบด้านความถูกต้องแสดงในรูปที่ 4.1



รูปที่ 4.1 ภาพรวมวิธีการเปรียบเทียบด้านความถูกต้อง

จากรูปที่ 4.1 มีขั้นตอนวิธีการทดสอบเปรียบเทียบด้านความถูกต้องดังต่อไปนี้

1) นำชุดข้อมูลทดสอบ (Testing set) ซึ่งเป็นชุดของไฟล์ข้อความประโยคอักษรเบรลล์ภาษาไทยระดับ 1 โดยแบ่งแยกย่อยออกเป็น 4 ประเภทได้แก่
   1.1. ประโยคที่มีการใช้คำควบกล้ำ
   1.2. ประโยคที่มีการใช้สระประสม
   1.3. ประโยคที่มีการใช้อักษรเบรลล์สองเซลล์
   1.4. ประโยคที่มีการใช้คำทับศัพท์

   โดยรายละเอียดของชุดข้อมูลทดสอบจะกล่าวในหัวข้อที่ 4.1.3 จากนั้นให้โปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file และโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ แปลงจากข้อความอักษรเบรลล์ภาษาไทยเป็นภาษาไทย
2) เมื่อโปรแกรมทั้งสองแปลงเสร็จแล้วก็จะได้ประโยคภาษาไทย จากนั้นจึงนำประโยคดังกล่าวไปเปรียบเทียบกับพจนานุกรมฉบับราชบัณฑิตยสถาน พ.ศ. 2542 เพื่อตรวจสอบความถูกต้อง
3) นับจำนวนประโยคที่โปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file แปลงถูกต้องและนับจำนวนประโยคที่โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ แปลงถูกต้อง
4) เปรียบเทียบว่าโปรแกรมไหนสามารถแปลงข้อความได้ถูกต้องมากกว่ากัน

4.1.3 ชุดข้อมูลทดสอบเปรียบเทียบด้านความถูกต้อง

ชุดข้อมูลทดสอบนี้ผู้วิจัยได้ออกแบบให้ครอบคลุมและสอดคล้องกันกับจุดประสงค์ของงานวิจัยนี้มากที่สุด โดยแบ่งออกเป็น 4 ประเภทด้วยกันคือ

1. ประโยคที่มีการใช้คำควบกล้ำ
2. ประโยคที่มีการใช้สระประสม
3. ประโยคที่มีการใช้อักษรเบรลล์สองเซลล์
4. ประโยคที่มีการใช้คำทับศัพท์

และเพื่อให้ชุดข้อมูลทดสอบดังกล่าวสอดคล้องกันกับประเภทภาษาเขียน ผู้วิจัยจึงได้นำเอาวิธีการรวบรวมข้อความแบบคลังข้อความ BEST [14] มาใช้กับชุดข้อมูลทดสอบนี้ด้วย โดยในแต่ละประเภทจะมีลักษณะการเขียนประโยคทั้ง 3 รูปแบบภาษาคือ

1) รูปแบบภาษาเขียนแบบเป็นทางการ โดยรวบรวมจากประโยคในหนังสือสารานุกรมไทยสำหรับเยาวชน, หนังสือเรียนวิชาภาษาไทยและหนังสือเรียนวิชาสังคมศึกษา
2) รูปแบบภาษาเขียนแบบไม่เป็นทางการ โดยรวบรวมจากประโยคในนวนิยายทั่วไปและบทความละครจากหนังสือพิมพ์
3) รูปแบบภาษาข่าว โดยรวบรวมจากบทความหนังสือพิมพ์ในอินเตอร์เน็ต

โดยรายละเอียดของชุดข้อมูลทดสอบสรุปได้ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 รายละเอียดชุดข้อมูลทดสอบด้านความถูกต้อง

ประเภทไฟล์	คำควบกล้ำ	สระประสม	เบรลล์สองเซลล์	คำทับศัพท์
ขนาดไฟล์ (KB)	9,923	9,700	315	820
ค่าเฉลี่ยจำนวนอักขระต่อบรรทัด	42.96	44.35	35.82	43.44
จำนวนคำเฉพาะ	231,115	219,022	8,713	18,881
จำนวนคำทั้งหมด	2,324,072	2,273,127	85,010	190,609
จำนวนคำเฉพาะคิดเป็นเปอร์เซ็นต์ของจำนวนคำทั้งหมด	9.94	9.64	10.25	9.91

ชุดข้อมูลทดสอบเหล่านี้เป็นประโยคภาษาไทยที่มีการใช้คำเฉพาะคือ คำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์ บรรจุอยู่ในแต่ละประโยค ซึ่งแต่ละประโยคนั้นอาจจะมีคำเฉพาะเหล่านี้มากกว่าหนึ่งคำก็ได้แต่อย่างน้อยทุกๆ ประโยคจะต้องมีคำเฉพาะเหล่านี้อย่างน้อยหนึ่งคำ จากนั้นผู้วิจัยได้ใช้โปรแกรมแปลงภาษาไทยเป็นอักษรเบรลล์ [1] (Thai2brl) ซึ่ง

เป็นโปรแกรมที่ทำหน้าที่แปลงภาษาไทยไปเป็นอักษรเบรลล์ระดับ 1 ซึ่งได้มีผู้พัฒนามาก่อนแล้ว เพื่อแปลงให้ชุดข้อมูลทดสอบดังกล่าวอยู่ในรูปแบบของอักษรเบรลล์ภาษาไทยระดับ 1 จากนั้นจึงนำชุดข้อมูลทดสอบดังกล่าวให้โปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file และโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ แปลงซึ่งจะได้ผลลัพธ์ออกมาเป็นประโยคภาษาไทย และนำไปเปรียบเทียบความถูกต้องกับชุดข้อมูลทดสอบซึ่งเป็นประโยคภาษาไทยต้นฉบับ เพื่อตรวจสอบความถูกต้องและหาข้อผิดพลาดต่อไป

4.1.4 ผลการทดลองเปรียบเทียบด้านความถูกต้อง

ผลการทดลองเปรียบเทียบการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยระหว่างโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ เมื่อใช้จำนวนแกรมเท่ากับ 3-แกรม, 4-แกรมและ 5-แกรม ซึ่งได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ พบว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ เมื่อใช้จำนวนแกรมเท่ากับ 4-แกรมและ 5-แกรม สามารถช่วยเพิ่มความถูกต้องในการแปลงประโยคที่มีการใช้คำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์ ได้มากที่สุด โดยผลการทดลองสามารถสรุปได้ดังตารางที่ 4.2

ตารางที่ 4.2 ผลการทดลองเปรียบเทียบความถูกต้องของวิธีการเดิมและวิธีการใหม่

ประเภทข้อมูล	ประเภทของประโยค			
	คำควบกล้ำ	สระประสม	อักษรเบรลล์สองเซลล์	คำทับศัพท์
จำนวนประโยคทั้งหมด	231,115	219,022	8,713	18,881
จำนวนประโยคที่วิธีการเดิมแปลงถูกต้อง	183,562	172,340	4,995	14,003
อัตราการแปลงถูกต้องของวิธีการเดิม (%)	79.42%	78.69%	57.33%	74.16%

ตารางที่ 4.2 ผลการทดลองเปรียบเทียบความถูกต้องของวิธีการเดิมและวิธีการใหม่ (ต่อ)

ประเภทข้อมูล	ประเภทของประโยค			
	คำควบกล้ำ	สระประสม	อักษรเบรลล์สองเซลล์	คำทับศัพท์
จำนวนประโยคที่วิธีการใหม่ขนาด 3-แกรมแปลงถูกต้อง	204,979	194,817	8,098	16,829
อัตราการแปลงถูกต้องของวิธีการใหม่ขนาด 3-gram (%)	88.69%	88.95%	92.94%	89.13%
จำนวนประโยคที่วิธีการใหม่ขนาด 4-แกรมแปลงถูกต้อง	215,768	205,071	8,525	17,715
อัตราการแปลงถูกต้องของวิธีการใหม่ขนาด 4-gram (%)	93.36%	93.63%	97.84%	93.82%
จำนวนประโยคที่วิธีการใหม่ขนาด 5-แกรมแปลงถูกต้อง	215,768	205,071	8,525	17,715
อัตราการแปลงถูกต้องของวิธีการใหม่ขนาด 5-gram (%)	93.36%	93.63%	97.84%	93.82%
วิธีการเดิม: โปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file วิธีการใหม่: โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ				

จากการทดลองเปรียบเทียบความถูกต้องของการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยพบว่าเทคนิคเอ็นแกรมที่นำมาใช้ในการตัดคำอักษรเบรลล์ภาษาไทยก่อนที่จะแปลงไปเป็นภาษาไทย จำนวนแกรมที่เหมาะสมคือ 4-แกรม เนื่องจากเมื่อใช้จำนวนแกรมเท่ากับ 4-แกรม สามารถช่วยลดปัญหาความกำกวมของการเขียนอักษรเบรลล์ภาษาไทยได้เป็นอย่างดี และสามารถแปลงได้ถูกต้องในระดับที่มากกว่า 93 เปอร์เซ็นต์ในทุกประเภทของประโยคเมื่อเปรียบเทียบกับโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิมที่ใช้เทคนิคการเขียน

กฎไวยากรณ์ทางภาษาเข้ามากำกับการแปลง ซึ่งไม่ครอบคลุมลักษณะรูปแบบการเขียนอักษรเบรลล์ภาษาไทยทั้งหมดได้ และมีอัตราความถูกต้องในการแปลงต่ำกว่า หากใช้จำนวนแกรมเป็น 3-แกรม อัตราความถูกต้องในการแปลงจะลดลง หากใช้จำนวนแกรมเป็น 5-แกรมอัตราความถูกต้องในการแปลงจะไม่เพิ่มขึ้น และใช้เวลาในการประมวลผลมากกว่าการใช้จำนวนแกรมเป็น 4-แกรม

ดังนั้นสรุปได้ว่าการนำเทคนิคเอ็นแกรมเข้ามาช่วยตัดคำอักษรเบรลล์ภาษาไทยก่อนที่จะแปลงไปเป็นภาษาไทยสามารถลดความกำกวมของภาษาลงไปได้ ซึ่งทำให้โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยทราบขอบเขตที่ชัดเจนของคำแต่ละคำ และแปลงได้ถูกต้องมากยิ่งขึ้น และจำนวนแกรมที่เหมาะสมคือ 4-แกรม

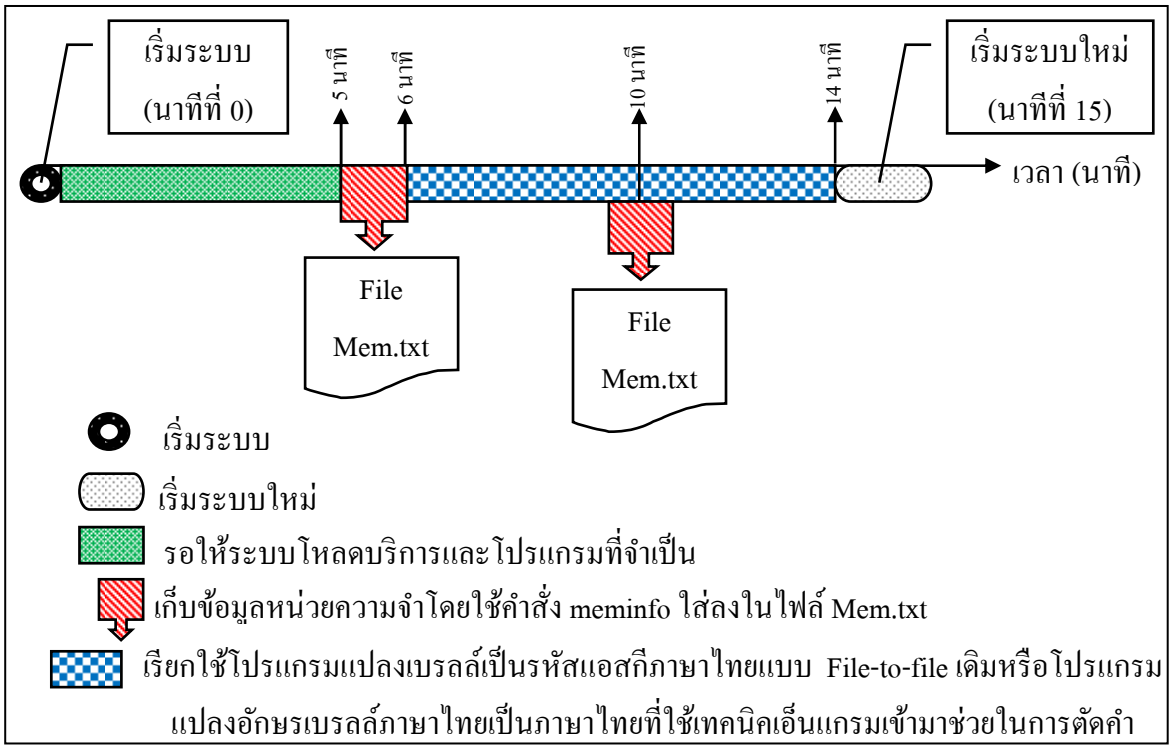
4.2 ด้านปริมาณการใช้หน่วยความจำในการแปลง

4.2.1 เครื่องมือที่ใช้ในการทดสอบ

1) โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file
2) โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ
3) ชุดข้อมูลทดสอบในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 (รายละเอียดของชุดข้อมูลทดสอบจะอธิบายในหัวข้อ 4.2.3)
4) เครื่องคอมพิวเตอร์ติดตั้งระบบปฏิบัติการ Linux Ubuntu version 9.04 1 เครื่อง ติดตั้ง CPU Intel รุ่น Core2Quad (Q9400) ความเร็ว 2.66 GHz ติดตั้งหน่วยหน่วยความจำที่ใช้งานได้ 3.25 GB
5) C/C++ compiler
6) Gedit Text editor

4.2.2 ภาพรวมการทดสอบปริมาณการใช้หน่วยความจำในการแปลง

ผู้วิจัยได้ออกแบบวิธีการทดสอบเปรียบเทียบปริมาณการใช้หน่วยความจำในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยระหว่างโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ โดยภาพรวมวิธีการทดลองด้านปริมาณการใช้หน่วยความจำแสดงในรูปที่ 4.2



รูปที่ 4.2 ภาพรวมวิธีการทดลองด้านปริมาณการใช้หน่วยความจำ

จากรูปที่ 4.2 ผู้วิจัยได้แสดงภาพรวมวิธีการทดลองวัดปริมาณการใช้หน่วยความจำของโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม และโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ซึ่งมีขั้นตอนดังต่อไปนี้

1) เริ่มระบบเครื่องคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการ Linux Ubuntu เวอร์ชัน 9.04
2) รอให้ระบบปฏิบัติการเรียกใช้งานบริการที่จำเป็นและโปรแกรมอื่นๆ ให้เสร็จสิ้นก่อน โดยต้องการให้มีผลกระทบกับการทดลองด้านปริมาณการใช้หน่วยความจำของโปรแกรมแปลงให้น้อยที่สุด โดยรอเป็นระยะเวลา 5 นาที
3) หลังจากรอเป็นระยะเวลา 5 นาทีแล้ว ผู้วิจัยได้ใช้คำสั่ง cat /proc/meminfo >> mem.txt ใน Terminal ซึ่งเป็นคำสั่งให้แสดงข้อมูลเกี่ยวกับหน่วยความจำของระบบออกมาและเก็บข้อมูลดังกล่าวไว้ในไฟล์ข้อความที่ชื่อว่า mem.txt โดยผู้วิจัยจะใช้ข้อมูลเฉพาะปริมาณหน่วยความจำที่ว่าง MemFree : โดยเปรียบเทียบปริมาณหน่วยความจำที่ว่าง

ก่อนเรียกใช้งานโปรแกรมแปลงและระหว่างโปรแกรมแปลงกำลังทำงาน โดยผลต่างของปริมาณหน่วยความจำที่ว่างก็คือหน่วยความจำที่โปรแกรมแปลงต้องใช้นั่นเอง

4) ในนาทีที่ 6 ผู้วิจัยเรียกใช้โปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม หรือโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ เพื่อให้แปลงข้อความจากอักษรเบรลล์ภาษาไทยเป็นภาษาไทย โดยใช้ชุดข้อมูลทดสอบที่ได้เตรียมเอาไว้ ซึ่งรายละเอียดของชุดข้อมูลทดสอบจะกล่าวในหัวข้อที่ 4.2.3
5) ในนาทีที่ 10 ผู้วิจัยได้ใช้คำสั่ง cat /proc/meminfo >> mem.txt ใน Terminal อีกครั้งเพื่อวัดปริมาณการใช้หน่วยความจำในขณะที่โปรแกรมแปลงกำลังทำงาน
6) ในนาทีที่ 15 หรือเมื่อโปรแกรมแปลงทำงานเสร็จสิ้นแล้วผู้วิจัยจะเริ่มระบบปฏิบัติการใหม่อีกครั้ง (Restart) เพื่อเริ่มการทดลองใหม่อีกครั้งและเป็นการป้องกันปัญหาการคืนหน่วยความจำให้แก่ระบบและผลกระทบอื่นๆ จากบริการและโปรแกรมอื่นๆ ที่ระบบปฏิบัติการเรียกขึ้นมาใช้งาน
7) ทำการทดลองซ้ำแต่ละโปรแกรมทั้งหมด 10 ครั้ง และในแต่ละครั้งเมื่อทดลองเสร็จสิ้นจะเริ่มเครื่องคอมพิวเตอร์ใหม่ทุกครั้ง

ซึ่งในการทดลองเปรียบเทียบปริมาณการใช้หน่วยความจำนี้ เพื่อต้องการเปรียบเทียบว่าโปรแกรมใดเหมาะสมที่จะนำไปใช้บนอุปกรณ์คอมพิวเตอร์แบบพกพาสำหรับผู้พิการทางสายตา เช่น Braille Note [1] มากกว่ากัน ทั้งนี้เพราะอุปกรณ์ Braille Note นั้นมีทรัพยากรที่จำกัดทั้งในด้านความเร็วของหน่วยประมวลผลกลางและปริมาณหน่วยความจำหลัก

4.2.3 ชุดข้อมูลทดสอบด้านปริมาณการใช้หน่วยความจำ

ชุดข้อมูลสำหรับทดสอบเปรียบเทียบปริมาณการใช้หน่วยความจำนี้ ผู้วิจัยได้นำชุดข้อมูลทดสอบนี้มาจากการชุดข้อมูลทดสอบเปรียบเทียบความถูกต้อง แต่พบว่าไฟล์ที่บรรจุประโยคที่มีการใช้อักษรเบรลล์สองเซลล์และคำทับศัพท์มีจำนวนคำน้อยมาก และโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file แปลงได้รวดเร็วมากจนไม่สามารถเรียกใช้คำสั่ง cat /proc/meminfo >> mem.txt ใน Terminal เพื่อเก็บข้อมูลปริมาณการใช้หน่วยความจำได้ทัน จึงไม่สามารถวัดปริมาณการใช้หน่วยความจำเมื่อใช้ไฟล์ประเภทดังกล่าวทดสอบ ทำให้ในการทดลองนี้

ใช้เฉพาะไฟล์ที่บรรจุประโยคที่มีการใช้คำควบกล้ำและสระประสมเท่านั้น โดยสามารถสรุปได้ดังตารางที่ 4.3

ตารางที่ 4.3 รายละเอียดชุดข้อมูลทดสอบด้านปริมาณการใช้หน่วยความจำ

ประเภทไฟล์	คำควบกล้ำ	สระประสม
ขนาดไฟล์ (KB)	9,923	9,700
ค่าเฉลี่ยจำนวนอักขระต่อบรรทัด	42.96	44.35
จำนวนคำเฉพาะ	231,115	219,022
จำนวนคำเฉพาะคิดเป็นเปอร์เซ็นต์ของจำนวนทั้งหมด	51.34	48.66

4.2.4 ผลการทดลองเปรียบเทียบปริมาณการใช้หน่วยความจำ

ผลการทดลองเปรียบเทียบปริมาณการใช้หน่วยความจำในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยระหว่างโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำเมื่อใช้จำนวนแกรมเท่ากับ 3-แกรม, 4-แกรม และ 5-แกรม ซึ่งได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ โดยทดลองซ้ำทั้งหมด 10 ครั้งและหาค่าเฉลี่ย พบว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำใช้ปริมาณหน่วยความจำน้อยกว่าโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file มาก ซึ่งมีผลการทดลองดังแสดงในตารางที่ 4.4 และผลการทดลองด้านปริมาณการใช้หน่วยความจำในการแปลงทั้งหมดจะแสดงไว้ในตารางที่ 1 ในภาคผนวก ข

ตารางที่ 4.4 ผลการทดลองปริมาณการใช้หน่วยความจำ

ครั้งที่	วิธีการเดิม		วิธีการใหม่					
			3-แกรม		4-แกรม		5-แกรม	
	คำควบกลั้ำ (MB)	สระประสม (MB)	คำควบกลั้ำ (MB)	สระประสม (MB)	คำควบกลั้ำ (MB)	สระประสม (MB)	คำควบกลั้ำ (MB)	สระประสม (MB)
เฉลี่ย	178.9	178.9	8.19	8.21	8.89	8.9	8.79	8.81
SD.	0.1729	0.2424	0.1524	0.1197	0.1449	0.2108	0.1197	0.0994
Max	179.2	179.3	8.5	8.4	9.2	9.3	9	9
Min	178.7	178.6	8	8	8.7	8.7	8.6	8.7

จากผลการทดลองเปรียบเทียบด้านปริมาณการใช้หน่วยความจำระหว่างโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ เมื่อใช้จำนวนแกรมเท่ากับ 3-แกรม, 4-แกรม และ 5-แกรม จะเห็นได้ว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำมีปริมาณการใช้หน่วยความจำเฉลี่ยแล้วอยู่ในช่วงระหว่างประมาณ 8.19-8.9 MB ซึ่งน้อยกว่าโปรแกรมเดิมมากประมาณ 20 เท่า ทั้งนี้เนื่องจากโปรแกรมเดิมได้ใช้ลักษณะการทำงานแบบรีเคอร์ซีฟ (Recursive) ซึ่งใช้ปริมาณหน่วยความจำมากแตกต่างจากโปรแกรมที่นำเสนอนี้ที่ใช้ลักษณะการทำงานแบบวนรอบ (For loop)เพื่อค้นหารูปแบบเอ็นแกรม (N-gram pattern) ที่คล้ายคลึงกัน

ดังนั้นจึงสรุปได้ว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ มีความเหมาะสมกับการนำไปใช้บนอุปกรณ์ Braille note [32] ที่มีทรัพยากรจำกัดมากกว่าโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม โดยเฉพาะอย่างยิ่งหน่วยความจำ

4.3 ระยะเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย

4.3.1 เครื่องมือที่ใช้ในการทดสอบ

1) โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file
2) โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ

3) ชุดข้อมูลทดสอบในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 (รายละเอียดของชุดข้อมูลทดสอบจะอธิบายในหัวข้อ 4.2.3)

4) เครื่องคอมพิวเตอร์ติดตั้งระบบปฏิบัติการ Linux Ubuntu version 9.04 1 เครื่อง ติดตั้ง CPU Intel รุ่น Core2Quad (Q9400) ความเร็ว 2.66 GHz ติดตั้งหน่วยหน่วยความจำที่ใช้งานได้ 3.25 GB

5) C/C++ compiler

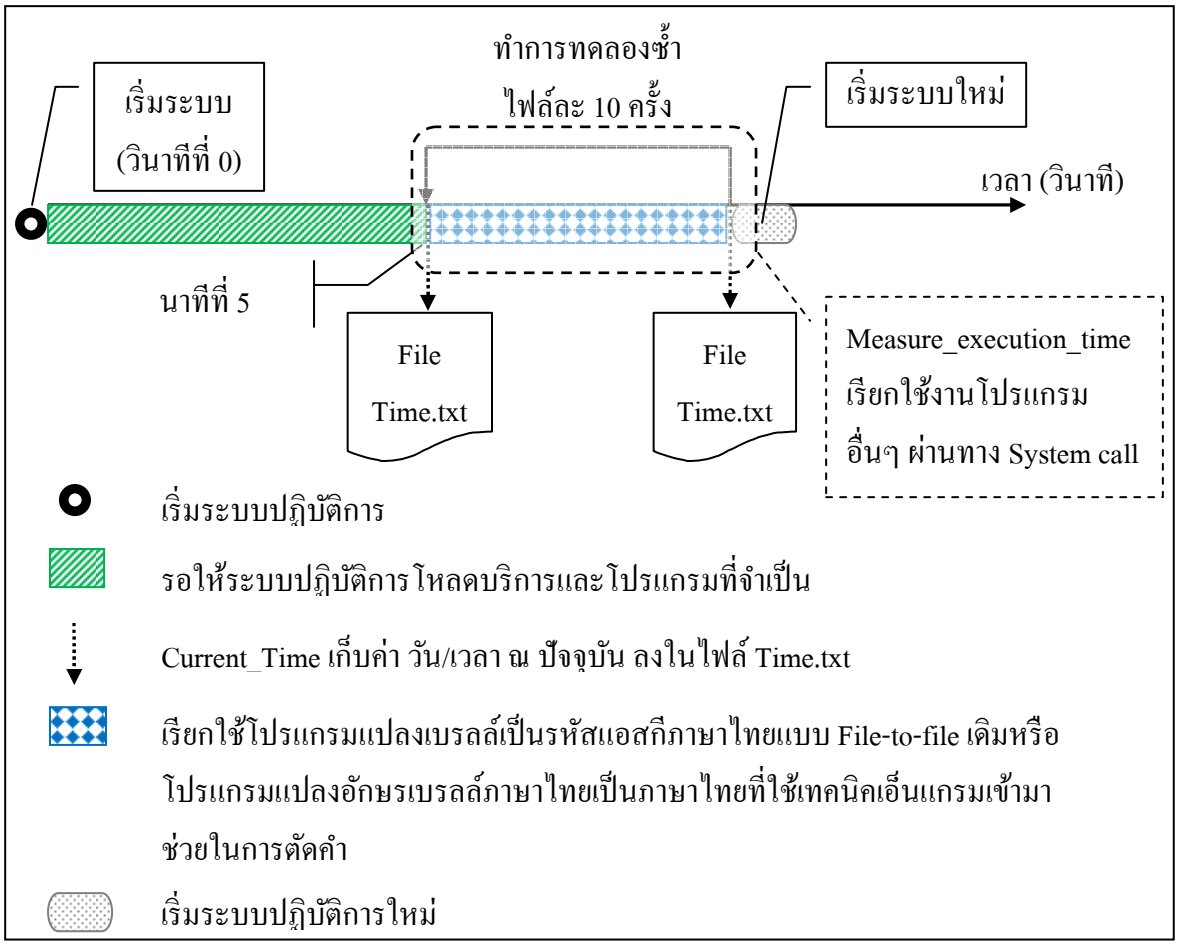
6) Gedit Text editor

4.3.2 ภาพรวมการทดสอบระยะเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย

ผู้วิจัยได้ออกแบบวิธีการทดสอบเปรียบเทียบความรวดเร็วในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยระหว่างโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ โดยใช้วิธีสร้างโปรแกรมที่ชื่อว่า measure_execution_time เพื่อทำหน้าที่เรียกใช้งานโปรแกรมแปลงเหล่านี้ผ่านทางการเรียกผ่านระบบ (System call) เนื่องจากผู้วิจัยต้องการให้โปรแกรมอื่นๆ มีการทำงานที่ต่อเนื่องกันไปตลอด ทั้งการจับเวลาและการแปลงข้อความอักษรเบรลล์ภาษาไทยเป็นภาษาไทย โดยมีลำดับการเรียกใช้งานโปรแกรมอื่นๆ ดังนี้

1) เรียกใช้งานโปรแกรมเก็บค่า วัน/เวลา ณ ปัจจุบัน (Current_Time) ครั้งแรก (ก่อนแปลง) แล้วเก็บค่าลงในไฟล์ Time.txt โดยโปรแกรม Current_Time เป็นโปรแกรมที่ผู้วิจัยสร้างขึ้นมา เพื่อเรียกใช้คำสั่ง date ผ่านทางการเรียกผ่านระบบซึ่งแสดงค่า วัน/เวลา ณ ปัจจุบันของระบบปฏิบัติการออกมา
2) เรียกใช้โปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file หรือโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำเพื่อแปลงชุดข้อมูลทดสอบจากอักษรเบรลล์ภาษาไทยเป็นภาษาไทย
3) เรียกใช้งานโปรแกรมเก็บค่า วัน/เวลา ณ ปัจจุบัน ครั้งที่สอง (หลังแปลง) แล้วเก็บค่าลงในไฟล์ Time.txt

โดยผลต่างเวลาครั้งแรกและครั้งที่สองก็คือเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยของโปรแกรมแปลงนั่นเอง โดยมีภาพรวมวิธีการการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลง ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 ภาพรวมวิธีการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลง

จากรูปที่ 4.3 ผู้วิจัยได้แสดงภาพรวมวิธีการทดลองวัดเวลาในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยของโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file และโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ซึ่งมีขั้นตอนดังต่อไปนี้

1. เริ่มระบบเครื่องคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการ Linux Ubuntu เวอร์ชัน 9.04
2. รอให้ระบบปฏิบัติการเรียกใช้งานบริการที่จำเป็นและโปรแกรมอื่นๆ ให้เสร็จสิ้นก่อน โดยต้องการให้มีผลกระทบกับการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงของ

โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยให้น้อยที่สุด โดยรอเป็นระยะเวลา 300 วินาที (5 นาที)

3. หลังจากรอเป็นระยะเวลา 300 วินาทีแล้ว ผู้วิจัยได้เรียกใช้โปรแกรม measure_execution_time ที่ได้สร้างขึ้นมาไว้ก่อนหน้านี้แล้ว เพื่อเริ่มกระบวนการจับเวลาในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยของโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม หรือโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ โดยในแต่ละไฟล์จะทดลองจับเวลาทั้งหมด 10 ครั้ง
4. หลังจากจับเวลาในการแปลงของแต่ละไฟล์เสร็จแล้วผู้วิจัยจะเริ่มระบบปฏิบัติการใหม่อีกครั้ง เพื่อเริ่มการทดลองจับเวลากับไฟล์อื่นๆ โดยใช้ชุดข้อมูลทดสอบที่ได้เตรียมเอาไว้ ซึ่งรายละเอียดของชุดข้อมูลทดสอบจะกล่าวในหัวข้อที่ 4.3.3

ซึ่งในการทดลองเปรียบเทียบความรวดเร็วในการแปลงนี้ เพื่อต้องการเปรียบเทียบว่าโปรแกรมใดสามารถแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยได้รวดเร็วกว่ากัน

4.3.3 ชุดข้อมูลทดสอบเปรียบเทียบด้านเวลาที่ใช้ในการแปลง

ชุดข้อมูลทดสอบเปรียบเทียบเวลาที่ใช้ในการแปลงนี้ผู้วิจัยได้ออกแบบให้สอดคล้องกันกับลักษณะไฟล์ที่ผู้ใช้นิยมนำมาใช้งานในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย โดยมีขนาดประมาณ 10 KB (ประมาณ 10 หน้า) โดยเป็นไฟล์ข้อความล้วน และเขียนอยู่ในรูปของอักษรเบรลล์ภาษาไทยระดับ 1 รวมทั้งมีการใช้สัญลักษณ์ต่างๆ ในการเขียนด้วยเช่น เครื่องหมายอัญประกาศ, นขลิขิต, ปรัศนี เป็นต้น โดยแบ่งออกเป็น 5 ประเภทด้วยกันคือ

1. ประโยคที่มีการใช้คำควบกล้ำ จำนวน 5 ไฟล์
2. ประโยคที่มีการใช้สระประสม จำนวน 5 ไฟล์
3. ประโยคที่มีการใช้อักษรเบรลล์สองเซลล์ จำนวน 5 ไฟล์
4. ประโยคที่มีการใช้คำทับศัพท์ จำนวน 5 ไฟล์
5. ประโยคทั่วไป จำนวน 10 ไฟล์

โดยประเภทประโยคทั่วไป เป็นการรวบรวมเอาประโยคที่มีการใช้คำควบกล้ำ, อักษรเบรลล์สองเซลล์, สระประสมและคำทับศัพท์เข้าไว้ด้วยกัน และเพื่อให้ชุดข้อมูลทดสอบดังกล่าว

ใกล้เคียงกันกับประโยคที่ผู้ใช้งานได้ใช้งานจริงและสอดคล้องกันกับประเภทของภาษาเขียน ผู้วิจัยจึงได้นำเอาวิธีการรวบรวมข้อความแบบคลังข้อความ BEST [14] มาใช้ในชุดข้อมูลทดสอบนี้ด้วย โดยในแต่ละประเภทจะประกอบไปด้วยลักษณะการเขียนทั้ง 3 รูปแบบภาษาคือ

1. รูปแบบภาษาเขียนแบบเป็นทางการ นำมาจากประโยคในหนังสือสารานุกรมไทยสำหรับเยาวชน, หนังสือเรียนวิชาภาษาไทยและหนังสือเรียนวิชาสังคมศึกษา
2. รูปแบบภาษาเขียนแบบไม่เป็นทางการ นำมาจากประโยคในนวนิยายทั่วไปและบทความละครจากหนังสือพิมพ์
3. รูปแบบภาษาข่าว นำมาจากบทความหนังสือพิมพ์ในอินเตอร์เน็ต

ซึ่งในแต่ละประเภททั้ง 5 ประเภทที่นำมาทดสอบเปรียบเทียบเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยใน โดยในแต่ละไฟล์จะมีลักษณะการเขียนทั้ง 3 รูปแบบดังกล่าว เพื่อให้ชุดข้อมูลทดสอบดังกล่าวมีความใกล้เคียงกันกับลักษณะการเขียนที่ผู้ใช้ได้ใช้งานจริงมากที่สุดเท่าที่จะเป็นไป โดยรายละเอียดของชุดข้อมูลทดสอบเปรียบเทียบเวลาที่ใช้ในการแปลงสามารถสรุปได้ดังแสดงในตารางที่ 4.5 – 4.9

ตารางที่ 4.5 รายละเอียดประเภทคำควบกล้ำ

ประเภทคำควบกล้ำ					
ไฟล์ที่	1	2	3	4	5
จำนวนบรรทัด	198	205	200	178	182
ขนาด(KB)	9.76	10	10	9.83	9.95
ค่าเฉลี่ยจำนวนตัวอักษรต่อบรรทัด	49.53	48.97	50.33	55.6	54.99
จำนวนคำควบกล้ำ	198	205	200	178	182

ตารางที่ 4.6 รายละเอียดประเภทสระประสม

ประเภทสระประสม					
ไฟล์ที่	1	2	3	4	5
จำนวนบรรทัด	197	197	191	202	198
ขนาด(KB)	9.86	9.64	9.17	9.19	9.76
ค่าเฉลี่ยจำนวนตัวอักษรต่อบรรทัด	50.26	49.11	48.18	45.62	49.52
จำนวนคำสระประสม	197	197	191	202	198

ตารางที่ 4.7 รายละเอียดประเภทอักษรเบรลล์สองเซลล์

ประเภทอักษรเบรลล์สองเซลล์					
ไฟล์ที่	1	2	3	4	5
จำนวนบรรทัด	211	211	245	214	217
ขนาด(KB)	9.14	9.17	9.6	9.92	9.87
ค่าเฉลี่ยจำนวนตัวอักษรต่อบรรทัด	43.39	43.54	39.16	46.51	45.59
จำนวนคำอักษรเบรลล์สองเซลล์	211	211	245	214	217

ตารางที่ 4.8 รายละเอียดประเภทคำทับศัพท์

ประเภทคำทับศัพท์					
ไฟล์ที่	1	2	3	4	5
จำนวนบรรทัด	193	191	186	196	185
ขนาด(KB)	9.98	9.68	9.96	10	9.19
ค่าเฉลี่ยจำนวนตัวอักษรต่อบรรทัด	51.98	50.9	53.88	51.26	49.9
จำนวนคำทับศัพท์	193	191	186	196	185

ตารางที่ 4.9 รายละเอียดประเภทข้อความทั่วไป

ประเภทประโยคทั่วไป										
ไฟล์ที่	1	2	3	4	5	6	7	8	9	10
ขนาด (KB)	9.92	9.97	9.98	9.91	9.96	10.00	10.00	10.00	9.95	9.98
จำนวนบรรทัด	198	224	214	201	214	219	218	214	221	195
จำนวนคำเฉพาะ	198	224	214	201	214	219	218	214	221	195
ค่าเฉลี่ยจำนวนตัวอักษรต่อบรรทัด	50.33	44.61	46.79	49.53	46.68	45.91	46.05	46.98	45.11	51.45

4.3.4 ผลการทดลองเปรียบเทียบด้านเวลาที่ใช้ในการแปลง

ผลการทดลองเปรียบเทียบเปรียบเทียบเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยระหว่างโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ เมื่อใช้จำนวนแกรมเท่ากับ 4-แกรม ซึ่งเป็นจำนวนแกรมที่มีความเหมาะสมมากที่สุดในการตัดคำอักษรเบรลล์ภาษาไทย เนื่องจากผลการทดลองด้านความถูกต้องในการแปลงในหัวข้อ 4.1.4 เมื่อจำนวนแกรมเท่ากับ 4-แกรม ก็สามารถให้ผลการแปลงที่มีความถูกต้องมากที่สุดแล้วโดยไม่จำเป็นที่จะต้องใช้จำนวนแกรมที่มากขึ้นแต่อย่างใด และผลการทดลองด้านปริมาณการใช้หน่วยความจำในหัวข้อ 4.2.4 เมื่อใช้จำนวนแกรมเท่ากับ 4-แกรม ก็มีการใช้ปริมาณหน่วยความจำที่ใกล้เคียงกันกับจำนวนแกรมเท่ากับ 5-แกรม ดังนั้นในการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงจึงใช้จำนวนแกรมเท่ากับ 4-แกรมเท่านั้นในการเปรียบเทียบกับโปรแกรมเดิม

โดยผลการทดลองพบว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ เมื่อใช้จำนวนแกรมเท่ากับ 4-แกรม ใช้เวลาในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่มากกว่าโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม อยู่พอสมควรโดยผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยไปเป็นภาษาไทยระหว่างโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ แสดงในตารางที่ 4.10 และผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงทั้งหมดจะแสดงไว้ในตารางที่ 2-7 ในภาคผนวก ข

ตารางที่ 4.10 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลง

ประเภทประโยค	วิธีการเดิม	วิธีการใหม่
คำควบกล้ำ (ค่าเฉลี่ยเวลา : วินาที)	0.5786	158.28
S.D.	0.141248	9.012554
Max	0.76	176
Min	0.41	148
Average file size (KB)	9.908	9.908
สระประสม (ค่าเฉลี่ยเวลา : วินาที)	0.59	158.12
S.D.	0.077249	19.88502
Max	0.74	196
Min	0.51	140
Average file size (KB)	9.524	9.524
เบรลล์สองเซล์ (ค่าเฉลี่ยเวลา : วินาที)	0.634	171.58
S.D.	0.10218	9.731267
Max	0.78	187
Min	0.53	160
Average file size (KB)	9.54	9.54

ตารางที่ 4.10 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลง (ต่อ)

ประเภทประโยค	วิธีการเดิม	วิธีการใหม่
คำทับศัพท์ (ค่าเฉลี่ยเวลา : วินาที)	0.5356	159.6
S.D.	0.065781	8.111468
Max	0.65	172
Min	0.41	146
Average file size (KB)	9.762	9.762
ทั่วไป (ค่าเฉลี่ยเวลา : วินาที)	0.5476	160.1
S.D.	0.009653	5.418822
Max	0.56	168
Min	0.53	150
Average file size (KB)	9.967	9.967

จากการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย พบว่าโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม สามารถแปลงได้รวดเร็วมากกว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ เมื่อใช้จำนวนแกรมเท่ากับ 4-แกรม เนื่องจากโปรแกรมเดิมใช้เทคนิคการแปลงแบบใช้กฎไวยากรณ์ทางภาษาเข้ามาควบคุมการแปลงร่วมกับวิธีการเขียนโปรแกรมแบบรีเคอร์ซีฟ (Recursive) ซึ่งแตกต่างจากวิธีที่นำเอาเทคนิคเอ็นแกรมเข้ามาช่วยตัดคำอักษรเบรลล์ภาษาไทย

ก่อนที่จะแปลงเป็นภาษาไทยและใช้วิธีการเขียนโปรแกรมแบบวนรอบ เพื่อค้นหารูปแบบเอ็นแกรม (N-gram pattern) ที่คล้ายคลึงกัน ทำให้โปรแกรมที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้สามารถแปลงได้ช้ากว่าโปรแกรมเดิม

อย่างไรก็ตามโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ ใช้เวลาในการแปลงข้อความอักษรเบรลล์ภาษาไทยเป็นภาษาไทย อยู่ในเกณฑ์ที่สามารถยอมรับได้ คือใช้เวลาประมาณไม่เกิน 3 นาที ในการแปลงเอกสารข้อความประมาณ 10 หน้า และให้ผลการแปลงที่ถูกต้องมากกว่าโปรแกรมเดิมเป็นอย่างมาก

4.4 สรุปผลการทดสอบ

การทดลองเปรียบเทียบประสิทธิภาพด้านความถูกต้องในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทย, การเปรียบเทียบด้านปริมาณการใช้หน่วยความจำและการเปรียบเทียบเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยระหว่างโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้กับโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file พบว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำมีประสิทธิภาพที่ดีกว่าด้านความถูกต้องในการแปลงและด้านปริมาณการใช้หน่วยความจำ แต่ใช้เวลาในการแปลงมากกว่าโปรแกรมเดิมมาก จึงเหมาะสำหรับนำไปใช้แปลงเอกสารทั่วไปที่ไม่ได้มีความยาวมากนักและไม่เหมาะที่จะนำไปใช้แปลงไฟล์ที่มีขนาดใหญ่มาก ซึ่งทำให้ใช้เวลาแปลงนานเกินไป ในบทถัดไปจะกล่าวถึงการสรุปผลการวิจัย, ปัญหาและอุปสรรคที่พบและข้อเสนอแนะต่างๆ ที่ควรปรับปรุงในงานวิจัยนี้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้มุ่งเน้นการแก้ไขปัญหาการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่เกี่ยวข้องกับปัญหาการแปลงประโยคที่มีการใช้คำควบกล้ำ สระประสม อักษรเบรลล์สองเซลล์และคำทับศัพท์ ซึ่งเป็นปัญหาที่พบในการทำงานของโปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file เดิม โดยใช้เทคนิคเอ็นแกรมโมเดลประยุกต์ในการตัดคำอักษรเบรลล์ภาษาไทยก่อนที่จะแปลงไปเป็นภาษาไทยเพื่อลดความกำกวมของลักษณะการเขียนอักษรเบรลล์ภาษาไทย โดยผลการวิจัยสามารถกล่าวโดยสรุปได้ดังต่อไปนี้

1) การนำเทคนิคเอ็นแกรมโมเดลเข้ามาช่วยในการตัดคำอักษรเบรลล์ภาษาไทยสามารถช่วยลดความกำกวมของลักษณะการเขียนอักษรเบรลล์ภาษาไทยระดับ 1 ได้เป็นอย่างดี โดยเทคนิคนี้จะช่วยหาขอบเขตของคำแต่ละคำ ทำให้โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ สามารถแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยได้อย่างถูกต้องมากยิ่งขึ้น โดยสามารถแปลงประโยคอักษรเบรลล์ภาษาไทยระดับ 1 ที่มีการใช้คำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์ได้ถูกต้องมากกว่า 93 เปอร์เซ็นต์ทุกประเภท
2) จากการทดลองทั้งด้านความถูกต้องในการแปลง ด้านปริมาณการใช้หน่วยความจำและเปรียบเทียบเวลาที่ใช้ในการแปลง สามารถสรุปได้ว่าเมื่อใช้จำนวนแกรม 4-แกรม มีความเหมาะสมที่สุดในการนำไปใช้งานการตัดคำอักษรเบรลล์ภาษาไทยระดับ 1 เนื่องจากเมื่อใช้จำนวนแกรม 4-แกรม สามารถแปลงได้ถูกต้องมากกว่าจำนวนแกรม 3-แกรม และใช้เวลาแปลงน้อยกว่าเมื่อใช้จำนวนแกรม 5-แกรม และจากการเปรียบเทียบปริมาณการใช้หน่วยความจำพบว่าใกล้เคียงกันกับจำนวนแกรม 3-แกรม และ 5-แกรม

5.2 สรุปปัญหา อุปสรรค และสาเหตุข้อผิดพลาดในงานวิจัยนี้

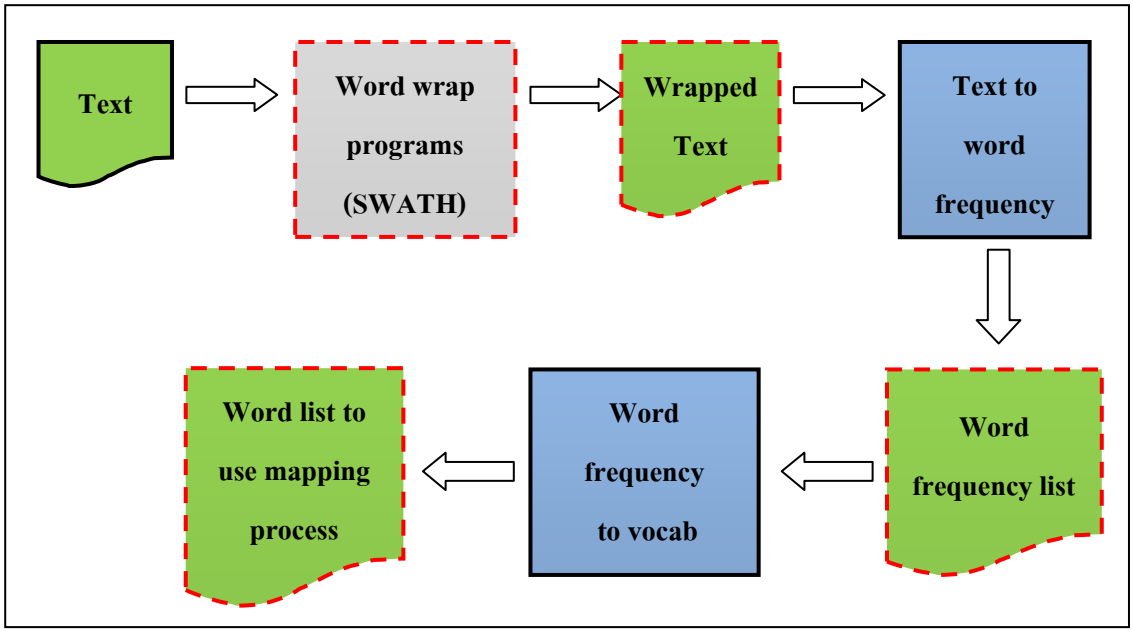
1) คลังข้อความสำหรับฝึกฝนเอ็นแกรมโมเดล

ในการสร้างเอ็นแกรมโมเดลนั้นจำเป็นต้องมีคลังข้อมูลหรือคลังข้อความสำหรับฝึกฝนเอ็นแกรมโมเดลให้สามารถสร้าง N-gram pattern ที่บ่งบอกลักษณะการเรียงกันของคำที่เกิดขึ้นเป็นประโยคหรือลักษณะการเรียงของอักขระที่เกิดขึ้นเป็นคำ ด้วยค่าความน่าจะเป็นหนึ่งๆ ของแต่ละภาษานั้นๆ ซึ่งในปัจจุบันนี้ไม่มีคลังข้อความใดเลยที่เขียนอยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 ที่จะนำมาใช้ฝึกฝนเอ็นแกรมโมเดลได้โดยตรง จึงจำเป็นต้องประยุกต์จากคลังข้อความภาษาไทยแทน โดยได้ใช้คลังข้อความ BEST

โดยผู้วิจัยได้พบว่ายังคงมีคำบางคำในคลังข้อความนี้เขียนผิดอยู่ก่อนแล้วและผู้วิจัยได้พยายามค้นหาและแก้ไขคำที่เขียนผิดเหล่านี้ให้ถูกต้อง โดยผู้วิจัยได้พยายามแก้ไขให้ถูกต้องและครอบคลุมมากที่สุดเท่าที่จะเป็นไปได้แต่อาจจะไม่ครอบคลุมคำทั้งหมดที่มีอยู่เนื่องจากคลังข้อความ BEST นั้นมีขนาดใหญ่มาก (รายละเอียดของคลังข้อความ BEST กล่าวไว้ในหัวข้อที่ 2.3.3) โดยผู้วิจัยได้รวบรวมรายการคำที่คลังข้อความ BEST เขียนผิดพลาดอยู่ก่อนแล้วในตารางที่ 10 ในภาคผนวก ค

2) โปรแกรมตัดคำและรายการคู่คำภาษาไทยกับคำอักษรเบรลล์ภาษาไทย

ในงานวิจัยนี้ผู้วิจัยได้นำโปรแกรมตัดคำ SWATH [29] มาใช้ในขั้นตอนการสร้างรายการคู่คำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยเพื่อนำไปใช้ในขั้นตอน Translate_to_thai เนื่องจากเทคนิคเอ็นแกรมโมเดลสามารถตัดคำอักษรเบรลล์ภาษาไทยได้แต่ไม่สามารถแปลงอักษรเบรลล์ภาษาไทยให้เป็นภาษาไทยได้ จึงต้องใช้วิธีการแปลงแบบจับคู่คำ ซึ่งจำเป็นต้องสร้างรายการคำที่ใช้ในการจับคู่คำขึ้นมาก่อน โดยผู้วิจัยได้ออกแบบวิธีการสร้างรายการคำดังกล่าวดังแสดงในรูปที่ 5.1



รูปที่ 5.1 ขั้นตอนการสร้างรายการคู่คำภาษาไทยและอักษรเบรลล์ภาษาไทย

จากรูปที่ 5.1 ผู้วิจัยได้ใช้โปรแกรมตัดคำ SWATH แบ่งคำเพื่อนำไปสร้างรายการคู่คำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยต่อไป แต่เนื่องจากโปรแกรมตัดคำ SWATH ยังคงมีความผิดพลาดในการตัดคำอยู่ โดยมีความถูกต้องในการตัดคำที่ 99 เปอร์เซ็นต์ [29] ทำให้ประโยคที่ตัดคำเรียบร้อยแล้ว (Wrapped Text ในรูปที่ 5.1) มีคำบางคำที่ตัดคำผิดพลาดรวมอยู่ด้วย ดังนั้นเมื่อนำไปใช้ในขั้นตอนถัดไปจึงก่อให้เกิดความผิดพลาดตามมา และทำให้รายการคู่คำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยผิดพลาดไปด้วย

ดังนั้นผู้วิจัยจึงได้พยายามค้นหาและแก้ไขรายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยดังกล่าว ให้ถูกต้องมากที่สุดโดยใช้วิธีเปรียบเทียบกับพจนานุกรมฉบับราชบัณฑิตยสถาน พ.ศ.2542 [16] และผู้วิจัยได้รวบรวมรายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาดในตารางที่ 9 ในภาคผนวก ค โดยได้ตัดรายการคำดังกล่าวออกไปจากรายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยที่ได้สร้างไว้

3) โปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์ (Thai2brl)

ผู้วิจัยได้นำโปรแกรม Thai2brl [1] มาใช้ในงานวิจัยนี้ในขั้นตอนต่างๆ ที่ต้องแปลงข้อความจากภาษาไทยให้เป็นอักษรเบรลล์ภาษาไทยระดับ 1 เช่น การสร้างคลังข้อความในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 สำหรับฝึกฝนเอ็นแกรมโมเดล,

การสร้างรายการคู่คำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยสำหรับขั้นตอน Translate_to_thai อย่างไรก็ตามโปรแกรมนี้ยังคงมีความผิดพลาดในการแปลงจากภาษาไทยไปเป็นอักษรเบรลล์ภาษาไทยและชุดข้อมูลทดสอบต่างๆ ที่นำเสนอในบทที่ 4 ทำให้ข้อความอักษรเบรลล์ภาษาไทยระดับ 1 ที่แปลงได้มีการแปลงที่ผิดพลาดรวมอยู่ด้วย จึงเป็นสาเหตุให้โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ มีการทำงานที่ผิดพลาดไปด้วย

โดยผู้วิจัยได้พยายามค้นหาคำที่แปลงผิดและแก้ไขในส่วนที่พบให้มากที่สุดเท่าที่จะเป็นไปได้ แต่อาจจะไม่ใช่ทั้งหมดที่โปรแกรม Thai2brl แปลงผิดพลาด จึงทำให้ข้อความอักษรเบรลล์ภาษาไทยระดับ 1 ที่แปลงได้ยังคงมีความผิดพลาดรวมอยู่ด้วย และหากไม่สามารถแก้ไขคำผิดได้ ก็จะตัดคำนั้นออกไปจากฐานข้อมูล โดยผู้วิจัยได้รวบรวมคำที่โปรแกรม Thai2brl แปลงผิดพลาดและได้แก้ไขไปแล้วในตารางที่ 8 ในภาคผนวก ค

4) โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ที่ได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้

เนื่องจากโปรแกรมที่ผู้วิจัยได้ออกแบบไว้นี้ได้ใช้เทคนิคเอ็นแกรมโมเดลเข้ามาช่วยในการตัดคำ โดยอ้างอิงจากค่าความน่าจะเป็นของคำที่เขียนเรียงติดกันที่เกิดขึ้นร่วมกันเป็นประโยคในคลังข้อมูลฝึก ซึ่งเอ็นแกรมโมเดลนี้เป็นลักษณะการตัดคำโดยใช้ค่าสถิติของคำเข้ามาเป็นเกณฑ์ตัดสิน ซึ่งบางครั้งก็มีความผิดพลาดในการตัดคำเกิดขึ้นได้โดยเฉพาะประโยคที่มีการใช้ชื่อเฉพาะ(ชื่อบุคคล, ชื่อสถานที่, ชื่อสิ่งของ) ซึ่งในความเป็นจริงคำที่เป็นชื่อเฉพาะเหล่านี้จะสามารถพบได้น้อยกว่าคำทั่วๆ ไปมากจึงทำให้มีค่าทางสถิติที่น้อยกว่าคำทั่วๆ ไปจึงเป็นสาเหตุให้โปรแกรมที่พัฒนาขึ้นมานี้ตัดคำผิดพลาดนั่นเอง ยกตัวอย่างเช่น ประโยค “คุณอาร์มจะไปไหนค่ะ” ซึ่งโดยทั่วไปแล้วค่าความน่าจะเป็นที่คำว่า “คุณ” เขียนนำหน้าแล้วตามด้วยคำว่า “อา” จะมีมากกว่าคำว่า “คุณ” เขียนนำหน้าแล้วตามด้วยคำว่า “อาร์ม” เสมอดังนั้นโปรแกรมจึงเลือกตัดคำให้เป็น “คุณอา” แทนที่จะเป็น “คุณอาร์ม” เป็นต้น

5.3 ข้อเสนอแนะ

1) โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ ซึ่งได้นำเสนอไว้ในวิทยานิพนธ์ฉบับนี้ ยังคงทำงานได้ช้าเมื่อเปรียบเทียบกับโปรแกรมเดิมและไม่เหมาะสมที่จะนำไปใช้แปลงไฟล์ข้อความขนาดใหญ่ จึงควรนำไปใช้ร่วมกับโปรแกรมอื่นๆ เพื่อให้แปลงข้อความเฉพาะประโยคที่มีการใช้คำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์

2) รายการคำที่นำมาจากพจนานุกรมฉบับราชบัณฑิตยสถานที่ใช้คัดเลือกประโยคที่มีการใช้คำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์จากคลังข้อความ BEST เพื่อนำไปสร้างโมเดลอักษรเบรลล์ภาษาไทยในหัวข้อที่ 3.3.1 ควรใช้พจนานุกรมฉบับราชบัณฑิตยสถานที่ปรับปรุงล่าสุดให้เป็นปัจจุบันเพื่อเพิ่มความถูกต้องของรายการคำดังกล่าวและครอบคลุมรายการคำให้มากที่สุดเท่าที่จะเป็นไปได้ ซึ่งจะทำให้ไฟล์โมเดลภาษาที่สร้างขึ้นครอบคลุมรูปแบบคำต่างๆ มากขึ้นและจะมีความถูกต้องในการเลือกคำมากขึ้นเช่นกัน

3) คลังข้อความที่นำมาใช้ฝึกฝนเอ็นแกรมโมเดล ควรมีความถูกต้อง 100 เปอร์เซ็นต์และเป็นไปตามกฎไวยากรณ์ภาษาของอักษรเบรลล์ภาษาไทยระดับ 1 แต่ทั้งนี้เนื่องจากในวิจัยนี้ ไม่มีคลังข้อความใดเลยที่เขียนอยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 ดังนั้นผู้วิจัยจึงต้องนำเอาคลังข้อความ BEST ซึ่งเป็นคลังข้อความภาษาไทย นำมาแปลงให้อยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 โดยใช้โปรแกรม Thai2brl [1] ในการแปลง ซึ่งยังคงมีความผิดพลาดในการแปลงอยู่ ซึ่งส่งผลให้เอ็นแกรมโมเดลที่สร้างขึ้นมานั้นมีข้อมูลที่ผิดพลาดรวมอยู่ด้วยและส่งผลต่อความถูกต้องของการตัดคำอักษรเบรลล์ภาษาไทย

ดังนั้นจึงควรมีการสร้างคลังข้อความอักษรเบรลล์ภาษาไทยที่มีความถูกต้องครบถ้วนสมบูรณ์เพื่อใช้ในการฝึกฝนเอ็นแกรมโมเดล ซึ่งจำเป็นต้องมีผู้เชี่ยวชาญในการเขียนอักษรเบรลล์ระดับ 1 เพื่อตรวจสอบแก้ไขข้อมูลดังกล่าวให้เป็นไปตามกฎไวยากรณ์ของอักษรเบรลล์ และต้องใช้เวลาในการรวบรวมข้อความประเภทต่างๆ

ให้ครบถ้วนเพื่อเป็นตัวแทนของประเภทภาษาเขียนแบบต่างๆ เพื่อสร้างเป็นคลังข้อความอักษรเบรลล์ภาษาไทย

4) คลังข้อความที่นำมาใช้ฝึกฝนเอ็นแกรมโมเดลควรมีการปรับปรุงให้มีขนาดที่กะทัดรัดมากขึ้นโดยใช้เทคนิค N-gram Weighting Technique [30] ซึ่งสามารถลดขนาดของคลังข้อมูลที่นำมาฝึกฝนเอ็นแกรมโมเดลลงได้และสามารถลดอัตราความผิดพลาดที่เกิดจากการเขียนคำที่ผิด (Word error rate) ลงได้ 1.2 เปอร์เซ็นต์ โดยจะส่งผลให้ใช้ปริมาณหน่วยความจำน้อยลงและสามารถทำงานได้รวดเร็วมากขึ้นเนื่องจากเอ็นแกรมโมเดลมีขนาดที่เล็กลงนั่นเอง

5) รายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยที่ใช้ในขั้นตอน Translate_to_Thai ควรมีการปรับปรุงให้มีความถูกต้องทุกๆ คำ เนื่องจากจะส่งผลต่อการเทียบคำจากอักษรเบรลล์ภาษาไทยไปเป็นคำภาษาไทยทำให้เกิดความผิดพลาดได้ ซึ่งสาเหตุนี้เกิดขึ้นจากผู้วิจัยได้ใช้โปรแกรมตัดคำภาษาไทย (SWATH) ซึ่งยังคงมีความผิดพลาดในการตัดคำอยู่บ้าง (SWATH มีความถูกต้องอยู่ในระดับที่ 99 เปอร์เซ็นต์) นำมาใช้ตัดคำประโยคภาษาไทยเพื่อนำไปสร้างเป็นรายการคำภาษาไทยและใช้โปรแกรมแปลงภาษาไทยไปเป็นอักษรเบรลล์ระดับ 1 (โปรแกรม Thai2brl) จึงทำให้มีความผิดพลาดดังกล่าวเกิดขึ้น

ดังนั้นจึงควรตรวจสอบรายการคู่คำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยให้ถูกต้องทุกๆ คำ โดยผู้ที่มีความรู้ความเชี่ยวชาญในการเขียนภาษาไทยและอักษรเบรลล์ภาษาไทยระดับ 1 และควรเพิ่มคู่ของคำอื่นๆ ที่อาจจะไม่พบในรายการคู่คำดังกล่าวเพื่อให้สามารถครอบคลุมการแปลงได้มากยิ่งขึ้น

เอกสารอ้างอิง

[1] วรพล ทินกรสูติบุตร, “การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด,” วิทยานิพนธ์มหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์, 2551.

[2] หน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, “Principle of N-gram model,” *หลักการทำงานของเอ็นแกรมโมเดล*, 19-July-2009. [Online]. Available: http://www.hlt.nectec.or.th/speech/index.php?option=com_content&view=article&id=66&Itemid=92&lang=en. [Accessed: 29-Feb-2012].

[3] อัครพล เอกวงศ์อนันต์, “การระบุคำไทยและคำทับศัพท์ด้วยแบบจำลองเอ็นแกรม,” วิทยานิพนธ์มหาบัณฑิต สาขาวิชาภาษาศาสตร์ ภาควิชาภาษาศาสตร์ คณะอักษรศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2548.

[4] นัฐวุฒิ ไชยเจริญ, “การตัดคำและกำกับหมวดคำภาษาไทยแบบเบ็ดเสร็จด้วยคอมพิวเตอร์,” วิทยานิพนธ์มหาบัณฑิต สาขาวิชาภาษาศาสตร์ ภาควิชาภาษาศาสตร์ คณะอักษรศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2544.

[5] วีรยา อมรพงษ์กุล,อัษฎาภรณ์ พิมพ์สมบูรณ์, จันทิมา พลพินิจ, และอุมาภรณ์ สายแสงจันทร์, “การประยุกต์เอ็นแกรมและเว็กเตอร์โมเดลสำหรับระบบย่อข้อความภาษาไทย,” Proceedings of Thailand Conference of Computer Science (ThCSC), pp. 48-54, Bangkok, Thailand, ธันวาคม พ.ศ. 2547.

[6] สารานุกรมไทยสำหรับเยาวชนโดยพระราชประสงค์ในสมเด็จพระเจ้าอยู่หัว, “Thai Word Segmentation,” *โปรแกรมตัดคำภาษาไทย*, 5-Nov-1983. [Online]. Available: http://kanchanapisek.or.th/kp6/New/sub/book/book.php?book=25&chap=7&page=t25-7-infodetail05.html. [Accessed: 15-Jan-2012].

[7] Thai Computational Linguistics Laboratory, “Natural language processing,” *การประมวลผลภาษาธรรมชาติ*, 14-Nov-2002. [Online]. Available: http://www.tcllab.org. [Accessed: 15-Jan-2012].

[8] ดวงแก้ว สวามิภักดิ์, “การประยุกต์ใช้เครื่องมือพัฒนา การประมวลผลภาษาภายใต้ระบบยูนิกซ์เพื่อสร้างซอฟต์แวร์วิเคราะห์ไวยากรณ์ภาษาไทย,” สถาบันไทยคดีศึกษา มหาวิทยาลัยธรรมศาสตร์, 2531.

[9] ยืน ภู่วรวรรณ และ วิวรรธน์ อึ่มอารมณ์, “การแบ่งพยางค์ไทยด้วยดิกชันนารี,” รายงานการประชุมวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 9, 2529.

[10] สัมพันธ์ ระรื่นรมย์, “การแบ่งคำไทยด้วยพจนานุกรม,” โครงงานวิศวกรรม ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2534.

[11] ไพศาล เจริญพรสวัสดิ์, “การตัดคำโดยใช้คุณลักษณะ,” วิทยานิพนธ์มหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2541.

[12] คณะอักษรศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, “คลังข้อมูลภาษา,” *คลังข้อมูลภาษา*, 31-Aug-2008. [Online]. Available: http://ling.arts.chula.ac.th/TNC/. [Accessed: 15-Jan-2012].

[13] หน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, “BEST corpus,” *คลังข้อความ BEST*, 19-July-2009. [Online]. Available: http://www.hlt.nectec.or.th/speech/index.php?option=com_jumi&fileid=14&Itemid=121&lang=en. [Accessed: 15-Jan-2012].

[14] นิธิ เอียวศรีวงศ์และคณะ, “Midnight University,” *มหาวิทยาลัยเที่ยงคืน*, Sep-1997. [Online]. Available: http://www.midnightuniv.org. [Accessed: 15-Jan-2012].

[15] G. W. Lesher, B. J. Moulton, D. J. Higginbotham, “Effect of N-gram order and training text size on word prediction,” In Proceedings of the Rehabilitation Engineering and Assistive Technology Society of North America (RESNA), pp. 52-54, 1999.

[16] ราชบัณฑิตยสถาน, “The Royal Institute,” *พจนานุกรมฉบับราชบัณฑิตยสถาน*, 31-Mar-1933. [Online]. Available: http://www.royin.go.th/th/home/. [Accessed: 15-Jan-2012].

[17] Stanley F. Chen และ Joshua Goodman, “An Empirical Study of Smoothing Techniques for Language Modeling,” Proceedings of the 34th annual meeting on Association for Computational Linguistics, pp. 310-318, 1996.

[18] หน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, “Training N-gram model,” *การฝึกฝนเอ็นแกรมโมเดลด้วยชุดเครื่องมือ CMU-SLM toolkit*, 17-Jan-2012. [Online]. Available: http://www.hlt.nectec.or.th/speech/index.php?option=com_content&view=article&id=70&Itemid=94. [Accessed: 4-Feb-2012].

[19] Doug Paul (MIT Lincoln Labs for research) และ U.S. Department of Defense Advanced Research Project Agency (ARPA), "ARPA format," *ARPA format*, 2004. [Online]. Available: http://www.speech.sri.com/projects/srilm/manpages/ngram-format.5.html. [Accessed: 15-Jan-2012].

[20] P.R. Clarkson และ R. Rosenfeld Cambridge University, "CMU-SLM toolkit," *CMU-SLM toolkit*, June-1999. [Online]. Available: http://www.speech.cs.cmu.edu /SLM/toolkit.html. [Accessed: 15-Jan-2012].

[21] Pisit Thai Natural Language Processing Lab, "โครงสร้างพจนานุกรมแบบทรัย," *ReoCities*, Aug-1998. [Online]. Available: http://reocities.com/ResearchTriangle/ thinktank/5593/Trie/trie.htm. [Accessed: 15-Jan-2012].

[22] The Math Forum (Drexel University), "Division by zero," *ปัญหาการหารด้วยศูนย์*, Jan-1994. [Online]. Available: http://mathforum.org/dr.math/faq/faq.divideby0.html. [Accessed: 4-Feb-2012].

[23] หน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, "การสร้างโมเดลภาษาด้วย CMU-SLM toolkit," *การสร้างโมเดลภาษาด้วยชุดเครื่องมือ CMU-SLM toolkit*, 17-Jan-2012. [Online]. Available: http://www.hlt.nectec.or.th/speech/index.php?option=com_content&view=article&id=70&Itemid=94. [Accessed: 4-Feb-2012].

[24] Krisda Khankasikam และ Nuttanart Muansuwan, "Thai Word Segmentation a Lexical Semantic Approach," The 10th Machine Translation Summit (MT Summit X), September 12-16, Phuket, Thailand, p. 331, 2005.

[25] Hitoshi Isahara Kansai Advanced Research Center และศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, "ORCHID corpus," *คลังข้อความ ORCHID*, Oct-1999. [Online]. Available: http://www.hlt.nectec.or.th/orchid/ [Accessed: 15-Jan-2012].

[26] กิตติชน แม้นสมุทร, "การตัดคำแบบเอ็นแกรม," *สร้าง Search Engine กัน*, 20-Jun-2007. [Online]. Available: http://ejeepss.wordpress.com/2007/06/20/สร้าง-search-engine-กันไหม/. [Accessed: 15-Jan-2012].

[27] หน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, "โปรแกรมตัดคำ SWATH," *Thai word segmentation*, 28-Jul-

2005. [Online]. Available: http://www.hlt.nectec.or.th/products/swath.php. [Accessed: 15-Jan-2012].

[28] B.P. Hsu and J.R. Glass, "N-gram Weighting: Reducing Training Data Mismatch in Cross-Domain Language Model Estimation," In Proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2008, pp. 829-838, 2008.

[29] สำนักงานคณะกรรมการการศึกษาขั้นพื้นฐาน, "ประวัติอักษรเบรลล์," *อักษรเบรลล์*, Oct-2008. [Online]. Available: http://school.obec.go.th/speced4/speced4/load/Borsure/4History of braille.pdf. [Accessed: 6-Feb-2012].

[30] BANA member, "American English Braille," Braille formats, 1996. [Online]. Available: http://www.brl.org/formats/index.html. [Accessed: 6-Feb-2012].

[31] Eitan Gurari, "North American Braille Computer Code," *North American Braille Computer Code Formats*, 3-Sep-2010. [Online]. Available: http://www.dotlessbraille.org/index.html. [Accessed: 6-Feb-2012].

[32] BrailleNote Users members, "BrailleNote," *BrailleNote*, 18-Feb-2012. [Online]. Available: http://www.braillenoteusers.info. [Accessed: 3-Mar-2012].

[33] สมทรง พันธ์สุวรรณ , "การอ่าน เขียน พิมพ์ อักษรเบรลล์," ,พิมพ์ครั้งที่ 1 หจก. จงเจริญการพิมพ์ พ.ศ. 2538.

[34] กมลวรรณ อินอร่าม, "การเรียนรู้อักษรเบรลล์," คณะครุศาสตร์ มหาวิทยาลัยราชภัฏสวนดุสิต, ISBN 974-223-404-3, 2547.

[35] The Braille Authority of North America (BANA), "Unified 8 dot Braille Code," *Unified 8 dot Braille Code*, Dec-2010. [Online]. Available: http://www.brailleauthority.org. [Accessed: 4-Mar-2012].

ภาคผนวก

ภาคผนวก ก : ผลงานที่ได้รับการตีพิมพ์เผยแพร่

ทศวัฒน์ ชุณหวิทยะธีระ และพิชญา ตัณฑัยย์ , “การใช้เอ็นแกรมช่วยในการตัดสินใจแปลอักษรเบรลล์ที่ใช้คำควบกล้ำ สระประสม และอักษรเบรลล์สองเซลล์,” ในการประชุมวิชาการเสนอผลงานวิจัย มหาวิทยาลัยพายัพ พ.ศ. 2555, เชียงใหม่, ประเทศไทย, 17 กุมภาพันธ์ 2555, รางวัลผลงานวิจัยดีเด่น (Best Paper Award) กลุ่มสาขาวิทยาศาสตร์เทคโนโลยี

ชื่อเรื่อง **การใช้เอ็นแกรมช่วยในการตัดสินใจแปลอักษรเบรลล์ที่ใช้คำควบกล้ำ สระประสม และอักษรเบรลล์สองเซลล์**

Enhancing Thai Braille translation with n-gram for decision making in the cases of compound consonants, vowels and characters

ชื่อคณะผู้วิจัย นายทศวัฒน์ ชุณหวิทยะธีระ และ ผศ.ดร. พิชญา ตัณฑัยย์

Mr. Totsawat Chunhawitayatera and Asst. Prof. Dr. Pichaya Tandayya

หน่วยงานต้นสังกัด ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

บทคัดย่อ

การแปลงข้อความรหัสอักษรเบรลล์ภาษาไทยให้เป็นข้อความรหัสแอสกีภาษาไทยในปัจจุบัน มักใช้กฎไวยากรณ์ทางภาษาเข้ามากำกับวิธีการแปลง แต่ยังคงประสบปัญหาการแปลงคำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์ เนื่องจากไวยากรณ์ที่แตกต่างกันและความไม่แน่นอนในการใช้ภาษา จึงไม่สามารถเขียนโปรแกรมให้ครอบคลุมทุกๆ กรณีที่เกิดขึ้นได้ งานวิจัยนี้ได้นำเสนอวิธีแก้ไขปัญหาการแปลงข้อความรหัสอักษรเบรลล์ภาษาไทยให้เป็นข้อความรหัสแอสกีภาษาไทยเพื่อใช้กับประโยคที่มีคำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์ โดยใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำก่อนที่จะแปลงให้เป็นภาษาไทยเพื่อลดความกำกวมของภาษาและนำเสนอจำนวนแกรมที่เหมาะสมสำหรับใช้ตัดคำอักษรเบรลล์ภาษาไทย ซึ่งผลการทดลองสรุปว่าสามารถแปลงข้อความได้ถูกต้องมากขึ้นและใช้ปริมาณหน่วยความจำน้อยลง แต่ใช้เวลาแปลงมากกว่าเดิมโดยอยู่ในเกณฑ์ที่ยอมรับได้ และจำนวนแกรมที่เหมาะสมคือ 4 แกรม

คำสำคัญ : อักษรเบรลล์ภาษาไทย, เอ็นแกรมโมเดล, โปรแกรมแปลงข้อความรหัสอักษรเบรลล์ภาษาไทยเป็นรหัสแอสกีภาษาไทย

ABSTRACT

In the present, the Braille Thai to Thai translation applies grammar rules to control the translation process but there are problems about translating compound consonants and vowels, two-cell Braille characters and transliterated words. The causes are too many grammar rules and the ambiguity of the language. Therefore, it is difficult to write a program to cover all conditions. This paper proposes a method to solve Braille Thai to Thai translation problems in the cases that the sentences contain compound consonants, vowels, two-cell Braille characters and transliterated words by using the N-gram model technique to wrap words before being translated into Thai in order to reduce the ambiguity of the language and propose the optimal N-gram number for wrapping Braille Thai words. The new method can improve the translation correctness and reduce the memory consumption better than the old method but the new method requires more translation time than the old method. The appropriate N-gram number is 4-gram for the Braille Thai text word wrap.

Key Words: Thai Braille, N-gram model, Braille to Thai translation program

บทนำ

ในประเทศไทยได้มีการพัฒนาโปรแกรมช่วยแปลงรหัสอักษรเบรลล์ภาษาไทยเป็นรหัสแอสกีภาษาไทยแบบ File-to-file มาอย่างต่อเนื่อง แต่โปรแกรมเหล่านั้นยังคงประสบปัญหาการแปลงคำที่เกี่ยวข้องกับคำควบกล้ำ, คำที่มีการใช้สระประสม, คำที่มีการใช้อักษรเบรลล์สองเซลล์และคำทับศัพท์ ซึ่งในวิทยานิพนธ์เรื่อง "การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด" วรพล ทินกรสูติบุตร (2551) โดยมหาวิทยาลัยสงขลานครินทร์ ได้นำเสนอโปรแกรม แปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file ซึ่งโปรแกรมนี้เป็นโปรแกรมแปลงข้อความรหัสอักษรเบรลล์ภาษาไทยเป็นข้อความรหัสแอสกีภาษาไทยแบบ File-to-file ซึ่งยังคงมีความผิดพลาดในการแปลงคำประเภทดังกล่าว

ในบทความนี้ได้เสนอวิธีแก้ไขปัญหาดังกล่าว โดยนำเทคนิคเอ็นแกรมเข้ามาช่วยในการหาขอบเขตของคำอักษรเบรลล์ภาษาไทย ซึ่งจะทำให้ทราบขอบเขตที่แน่นอนของคำแต่ละคำ จึงส่งผลให้การแปลงเบรลล์ภาษาไทยเป็นแอสกีภาษาไทยทำได้ง่ายขึ้นและถูกต้องมากยิ่งขึ้น อีกทั้งการนำเทคนิคเอ็นแกรมมาประยุกต์ใช้นี้ ทำให้กระบวนการแปลงเบรลล์ภาษาไทยเป็นแอสกีภาษาไทยมีความซับซ้อนน้อยลง เมื่อเปรียบเทียบกับการแปลงเบรลล์ภาษาไทยเป็นแอสกีภาษาไทยที่ใช้เทคนิคกฎไวยากรณ์ทางภาษา โดยให้เอ็นแกรมคำนวณหาค่าความน่าจะเป็นของอักขระที่เขียนเรียงติดกัน (Character sequence) ที่เกิดขึ้นร่วมกันเป็นคำอักษรเบรลล์ภาษาไทย เพื่อช่วยในการระบุขอบเขตของคำอักษรเบรลล์ภาษาไทยว่าควรแบ่งคำออกมาเป็นรูปแบบอย่างไรจึงจะเหมาะสม และเพิ่มความถูกต้องในการแปลง

วัตถุประสงค์ของงานวิจัย

งานวิจัยนี้มุ่งเน้นวิจัยและพัฒนาเพื่อศึกษาการนำเทคนิคเอ็นแกรม เข้ามาประยุกต์ใช้กับการแปลงอักษรเบรลล์ภาษาไทย เพื่อช่วยปรับปรุงกระบวนการแปลงรหัสอักษรเบรลล์ภาษาไทยไปเป็นรหัสแอสกีภาษาไทยให้ถูกต้องมากยิ่งขึ้น โดยมีประเด็นที่ศึกษาดังต่อไปนี้

1) นำเทคนิคเอ็นแกรมเข้ามาประยุกต์ใช้ร่วมกับการแปลงเบรลล์ภาษาไทยเป็นแอสกีภาษาไทย จะช่วยเพิ่มความถูกต้องในการแปลงได้หรือไม่ โดยเปรียบเทียบกับโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม ซึ่งได้นำเสนอไว้ในวิทยานิพนธ์เรื่อง "การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด"

2) หาค่าของจำนวนแกรมที่เหมาะสม เพื่อหาค่าของจำนวนแกรมที่ให้ผลลัพธ์ในการแบ่งคำที่ดีที่สุดโดยเปรียบเทียบระหว่าง 3-แกรม, 4-แกรม และ 5-แกรม โดยใช้วิธีพิจารณาจากความถูกต้องของการแบ่งคำอักษรเบรลล์ภาษาไทยระดับ 1 ซึ่งจะต้องถูกต้องตามหลักไวยากรณ์ของอักษรเบรลล์ภาษาไทยและเมื่อนำไปแปลงให้เป็นภาษาไทยแล้วจะต้องถูกต้องตามหลักไวยากรณ์ของภาษาไทยด้วย

บทความนี้นำเทคนิคเอ็นแกรมมาตัดคำอักษรเบรลล์ภาษาไทยและแปลงให้เป็นภาษาไทยโดยใช้วิธีจับคู่คำ และเสนอจำนวนแกรมที่เหมาะสมในการตัดคำอักษรเบรลล์ภาษาไทย

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

งานวิจัยและเทคนิคต่างๆ ที่เกี่ยวข้อง ซึ่งได้นำมาใช้ในการวิจัยนี้โดยประกอบไปด้วย 1) โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม 2) เทคนิคการตัดคำ 3) เอ็นแกรมโมเดล 4) Smoothing Technique 5) คลังข้อมูลภาษา และ 6) CMU-SLM Toolkit

1. โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม

โปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file ได้นำเสนอไว้ในวิทยานิพนธ์เรื่อง "การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด" วรพล ทินกรสูติบุตร (2551) ของมหาวิทยาลัยสงขลานครินทร์ ซึ่งโปรแกรมดังกล่าวทำหน้าที่แปลงเบรลล์ภาษาไทยเป็นรหัสแอสกีภาษาไทยในรูปแบบ File-to-file โดยกระบวนการทำงานของโปรแกรมนี้ ใช้กฎไวยากรณ์ทาง

ภาษาของอักษรเบรลล์ภาษาไทยมาควบคุมการแปลง ซึ่งให้ผลลัพธ์ในการแปลงถูกต้องระดับหนึ่ง

จากการศึกษางานวิจัยนี้พบว่ายังคงมีปัญหาในการแปลงเบรลล์ภาษาไทยเป็นแอสกีภาษาไทยอยู่ด้วยกัน 2 ประการคือ 1) ปัญหาการแปลงคำควบกล้ำและสระประสม ซึ่งปัญหานี้เกิดจากการตัดคำในประโยคอักษรเบรลล์ภาษาไทยที่มีความกำกวม ทำให้โปรแกรมดังกล่าวไม่สามารถแยกแยะได้ว่าอักขระของคำควบกล้ำที่เขียนนั้นเป็นอักขระตัวสะกดของคำๆ หน้าหรือเป็นอักขระอักษรนำของคำๆ หลัง 2) ปัญหาการแปลงคำที่มีการใช้อักษรเบรลล์สองเซลล์ ซึ่งปัญหานี้เกิดจากมีอักษรเบรลล์สองเซลล์บางตัวมีรูปฟ้องกันกับอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกัน ทำให้โปรแกรมดังกล่าวไม่สามารถทราบได้ว่าอักขระเหล่านี้ควรเป็นอักษรเบรลล์สองเซลล์หนึ่งตัว หรือเป็นอักษรเบรลล์เซลล์เดียวสองตัวที่เขียนติดกัน

2. เทคนิคการตัดคำ

เทคนิคการตัดคำ ใช้ในการหาขอบเขตของคำในงานด้านประมวลผลภาษาธรรมชาติ เพื่อทำให้คอมพิวเตอร์สามารถเข้าใจความหมายของคำในภาษานั้นๆ ได้และเนื่องจากภาษาบางภาษา เช่น ภาษาไทย ภาษาจีน ภาษาญี่ปุ่น ภาษาลาว เป็นต้น มีลักษณะการเขียนประโยคที่ประกอบไปด้วยคำย่อยๆ หลายคำเรียงติดกันโดยไม่มีการเว้นช่องว่างระหว่างคำเหมือนกับภาษาอังกฤษทำให้เกิดปัญหา “ขอบเขตของคำ” เมื่อนำข้อมูลเหล่านั้นไปประมวลผลด้วยคอมพิวเตอร์ ทำให้คอมพิวเตอร์ไม่สามารถทราบได้ว่าในประโยคนั้นๆ ประกอบด้วยคำกี่คำ และแต่ละคำประกอบด้วยตัวอักษรอะไรบ้าง จึงจำเป็นต้องตัดคำเสียก่อน ก่อนที่จะนำไปประมวลผลต่อ เพื่อแบ่งขอบเขตของคำแต่ละคำอย่างชัดเจน

อักษรเบรลล์ภาษาไทยมีลักษณะการเขียนเช่นเดียวกันกับภาษาไทยคือ เขียนคำแต่ละคำติดๆ กันเป็นประโยคและเว้นวรรคระหว่างประโยค แต่จะเรียงตัวอักษรนำ ตัวอักษรตาม สระ ตัวสะกด และวรรณยุกต์ที่แตกต่างไปจากภาษาไทย

ดังนั้นหากใช้เทคนิคการตัดคำเข้ามาช่วยตัดคำอักษรเบรลล์ภาษาไทย เพื่อให้ทราบขอบเขตของคำที่ชัดเจน และลดความกำกวม ทำให้การแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยทำได้ง่ายขึ้นและถูกต้องมากยิ่งขึ้นอีกด้วย นอกจากนี้ยังไม่มีงานวิจัยใดๆ เลยที่จะนำวิธีการตัดคำมาใช้กับอักษรเบรลล์ภาษาไทย ซึ่งในปัจจุบันมีเพียงงานวิจัยที่เกี่ยวข้องกับการตัดคำภาษาไทยเท่านั้น

ในปัจจุบันพบว่าเทคนิคการตัดคำโดยใช้คลังข้อมูลมีความถูกต้องแม่นยำมากที่สุด และเทคนิคการตัดคำโดยใช้เอ็นแกรมโมเดลเป็นเทคนิคการตัดคำโดยใช้คลังข้อมูลที่นิยมใช้ในปัจจุบันมากที่สุด

3. เอ็นแกรมโมเดล (N-gram model)

เอ็นแกรมโมเดล (N-gram model) คือ แบบจำลองที่ใช้คำนวณค่าความน่าจะเป็นของชุดอักขระ (character sequence) ที่เกิดขึ้นร่วมกันเป็นคำ หรือค่าความน่าจะเป็นของคำที่เขียนเรียงกัน (word sequence) ที่เกิดขึ้นร่วมกันเป็นประโยค โดยค่าความน่าจะเป็นของชุดอักขระหรือคำสามารถคำนวณได้จากคลังข้อมูลฝึกที่สร้างไว้

แกรม (Gram) คือ หน่วยที่ใช้ในการสร้างแบบจำลองอาจจะเป็นเสียง คำ หรืออักขระก็ได้และแกรมมีได้หลายขนาดแล้วแต่จะกำหนด ตั้งแต่ 1 จนถึง N โดย N เป็นจำนวนนับตั้งแต่ 1, 2, 3 ..., n

หลักการทำงานของเอ็นแกรมอาศัยหลักความเป็นจริงที่ว่าเมื่อมีการเขียนอักขระใดๆ หรือคำใดๆ เรียงติดกันเพื่อสร้างเป็นคำหรือเป็นประโยค แต่ละอักขระหรือคำที่เขียนนั้น จะมีความสัมพันธ์กับอักขระหรือคำที่เขียนไว้ก่อนหน้าด้วยค่าความน่าจะเป็นค่าหนึ่ง ตัวอย่างเช่น มีคำ 3 คำคือ ”ข้าว” “กิน”และ“ฉัน” เมื่อนำมาเขียนเรียงกันเพื่อสร้างเป็นประโยค จะสามารถเรียงได้หลายรูปแบบ คือ “ข้าว+กิน+ฉัน” “กิน+ฉัน+ข้าว” “ฉัน+กิน+ข้าว” เป็นต้น แต่รูปแบบ “ฉัน+กิน+ข้าว” จะสามารถพบได้มากที่สุดหรืออีกนัยหนึ่งก็คือรูปแบบนี้มีค่าความน่าจะเป็นสูงที่สุดนั่นเอง แต่ทั้งนี้เทคนิคแบบเอ็นแกรมเป็นการคำนวณในเชิงสถิติเท่านั้นและไม่ได้

นำเอาเรื่องของกฎไวยากรณ์ภาษาเข้ามาเกี่ยวข้องด้วย โดยค่าความน่าจะเป็นมีสมการคำนวณดังนี้

ค่าความน่าจะเป็นที่จะเกิดคำหรืออักขระ x_i โดยมีชุดของคำหรือชุดอักขระ $x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1}$ นำหน้า $= P(x_i| x_{i-1},x_{i-2},x_{i-3},\ldots,x_{i-n})$

ซึ่ง $P(x_i| x_{i-1},x_{i-2},x_{i-3},\ldots,x_{i-n}) = c(x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1},x_i)/c(x_{i-1},x_{i-2},x_{i-3},\ldots,x_{i-n})$

โดยที่ $c(x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1},x_i)$ คือจำนวนครั้งในคลังข้อมูลฝึกที่เกิด x_i ร่วมกับ $x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1}$ $c(x_{i-1},x_{i-2},x_{i-3},\ldots,x_{i-n})$ คือจำนวนครั้งที่เกิด $x_{i-n},\ldots,x_{i-3},x_{i-2},x_{i-1}$ ในคลังข้อมูลฝึก

4. Smoothing Techniques

Smoothing techniques สำหรับงานวิจัยทางด้านโมเดลภาษาคือ เทคนิคประมาณค่าความน่าจะเป็นให้กับคำหรือชุดของคำที่ไม่พบในคลังข้อมูลฝึก ซึ่งเทคนิคนี้อาศัยวิธีการประมาณค่าทางสถิติในการคำนวณหาค่าความน่าจะเป็นของคำที่ไม่พบในคลังข้อความ และเทคนิคนี้เป็นกระบวนการที่จำเป็นสำหรับงานด้านสถิติอีกทั้งเป็นการป้องกันปัญหาการหารด้วยศูนย์อีกด้วย และจากข้อจำกัดของคลังข้อความที่ไม่สามารถจัดเก็บคำและคู่ของคำทั้งหมดที่เกิดขึ้นจริงไว้ในคลังข้อมูลได้ทั้งหมด ทำให้คำบางคำหรือคู่ของคำบางคำอาจจะไม่มีอยู่ในคลังข้อมูล จากการศึกษาเปรียบเทียบเทคนิควิธีประมาณค่าความน่าจะเป็น Smoothing techniques ของ Stanley F. Chen และ Joshua Goodman (1996) พบว่าเทคนิคของ Jelinek และ Mercer Smoothing (Interpolation) จะให้ผลลัพธ์ดีที่สุดเมื่อเปรียบเทียบกับวิธีอื่นๆ โดยมีสมการคำนวณประมาณค่าความน่าจะเป็นดังนี้

$$P_{interp}\left(w_i \mid w_{i-n+1}^{i-1}\right)= \lambda_{w_{i-n+1}^{i-1}} P_{ML}\left(w_i \mid w_{i-n+1}^{i-1}\right)+(1-\lambda_{w_{i-n+1}^{i-1}}) P_{interp}\left(w_i \mid w_{i-n+2}^{i-1}\right)$$

โดยค่าความน่าจะเป็นของ Interpolation เท่ากับค่าถ่วงน้ำหนัก (Back-off weight) คูณด้วยค่าความน่าจะเป็นของจำนวนแกรมที่ n-1

โดยที่ $P_{ML}\left(w_i \mid w_{i-n+1}^{i-1}\right) = C(w_{i-n+1}^{i-1}\ w_i) / C(w_{i-n+1}^{i-1})$

คือ ค่าความน่าจะเป็นของจำนวนแกรมที่ n-1 ซึ่งหาได้จากจำนวนคำที่ w_i เกิดขึ้นร่วมกับ w_{i-n+1}^{i-1} หารด้วยจำนวนคำของ w_{i-n+1}^{i-1} ที่พบในคลังข้อมูลฝึก

และ $\lambda_{w_{i-n+1}^{i-1}} = \dfrac{C\left(w_{i-n+1}^{i-1}\right)}{|\ W_i : C\left(w_{i-n+1}^{i-1}\right) > 0\ |}$

คือค่าถ่วงน้ำหนัก (Back-off weight) เท่ากับจำนวนคำของ w_{i-n+1}^{i-1} ที่พบในคลังข้อมูลฝึกหารด้วยค่าสัมบูรณ์ของจำนวนคำ w_i ต่อจำนวนคำของ w_{i-n+1}^{i-1}

5. คลังข้อมูลภาษา

คลังข้อมูลภาษา คือ ข้อมูลภาษาเขียนหรือภาษาพูดที่เป็นภาษาที่ใช้จริง ซึ่งรวบรวมขึ้นมาในปริมาณที่มากเพียงพอตามเงื่อนไขที่กำหนดเพื่อให้ได้คลังข้อมูลที่เป็นตัวแทนของภาษาที่ต้องการ คลังข้อมูลภาษานำไปใช้งานในด้านภาษาศาสตร์ การสร้างแบบจำลองภาษารวมถึงการประมวลผลภาษาธรรมชาติอีกด้วย เช่น นำไปใช้สร้างพจนานุกรม นำข้อมูลด้านสถิติของคำไปสร้างเป็นแบบจำลองเพื่อนำมาใช้ในงานประมวลผลภาษา-ธรรมชาติด้วยคอมพิวเตอร์

ในงานวิจัยนี้ได้นำคลังข้อความภาษาไทยมาประยุกต์ใช้งานในการฝึกฝนเอ็นแกรมโมเดล โดยได้เลือกคลังข้อความ BEST ซึ่งเป็นคลังข้อความภาษาไทยและได้คัดเลือกเฉพาะประโยคที่มีการใช้คำควบกล้ำ, สระประสม, อักษรเบรลล์สองเซลล์และคำทับศัพท์เท่านั้น ทั้งนี้เพราะโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิมสามารถแปลงประโยคทั่วๆ ไปได้ถูกต้องอยู่แล้วและต้องนำคลังข้อความมาแปลงให้อยู่ในรูปแบบของอักษรเบรลล์ภาษาไทยระดับ 1 เสียก่อน จึงจะสามารถนำไปฝึกฝนเอ็นแกรมโมเดลได้ โดยในที่นี้จะใช้โปรแกรมแปลงภาษาไทยเป็นอักษรเบรลล์ (Thai2brl) ซึ่งได้นำเสนอไว้ในวิทยานิพนธ์เรื่อง “การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด” วรพล ทินกรสูติบุตร (2551) ของมหาวิทยาลัยสงขลานครินทร์

คลังข้อความ BEST เป็นคลังข้อความภาษาไทยที่มีการเก็บรวบรวมคำอย่างเป็นระบบและมีขนาดใหญ่ที่สุดในปัจจุบัน ซึ่งพัฒนาโดยหน่วย

ปฏิบัติการวิจัยวิทยาการมนุษยภาษา (HLT) ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) โดยคัดเลือกข้อความจากงานเขียนลักษณะต่าง ๆ 3 ประเภทคือ 1) ข้อความจากหนังสือประเภทนวนิยายในฐานะเป็นตัวแทนของภาษาพูดทั่วไป 2) ข้อความจากเว็บไซต์ www.midnightuniv.org และข้อความจากสารานุกรมสำหรับเยาวชนไทยในฐานะเป็นตัวแทนของภาษาเขียนอย่างเป็นทางการ 3) ข้อความจากหนังสือพิมพ์บนอินเตอร์เน็ตในฐานะเป็นตัวแทนของภาษาข่าว

6. CMU-SLM Toolkit

ชุดเครื่องมือ CMU-SLM Toolkit เป็นชุดเครื่องมือที่ใช้ในการสร้างโมเดลภาษา (Language model) ซึ่งชุดเครื่องมือนี้ได้รับการพัฒนาโดยมหาวิทยาลัย Carnegie Mellon โดยในงานวิจัยนี้ได้เลือกใช้ชุดเครื่องมือ CMU-SLM Toolkit เพื่อช่วยอำนวยความสะดวกในการสร้างโมเดลภาษาของอักษรเบรลล์ภาษาไทยระดับ 1 เนื่องจากชุดเครื่องมือนี้ มีการใช้งานกันอย่างแพร่หลายและมีเทคนิคการ Smoothing แบบ Interpolation ชุดเครื่องมือนี้มีการเผยแพร่เป็นสาธารณะ มีคู่มืออธิบายการใช้งานชัดเจน ทำให้ใช้งานได้ง่ายและเป็นชุดเครื่องมือที่ครบถ้วนสมบูรณ์ โดยในชุดเครื่องมือนี้จะประกอบด้วยโปรแกรมย่อยจำนวนหลายโปรแกรมด้วยกัน เช่น text2wfreq, wfreq2vocab, text2wngram, text2idngram, idngram2lm, interpolate เป็นต้น

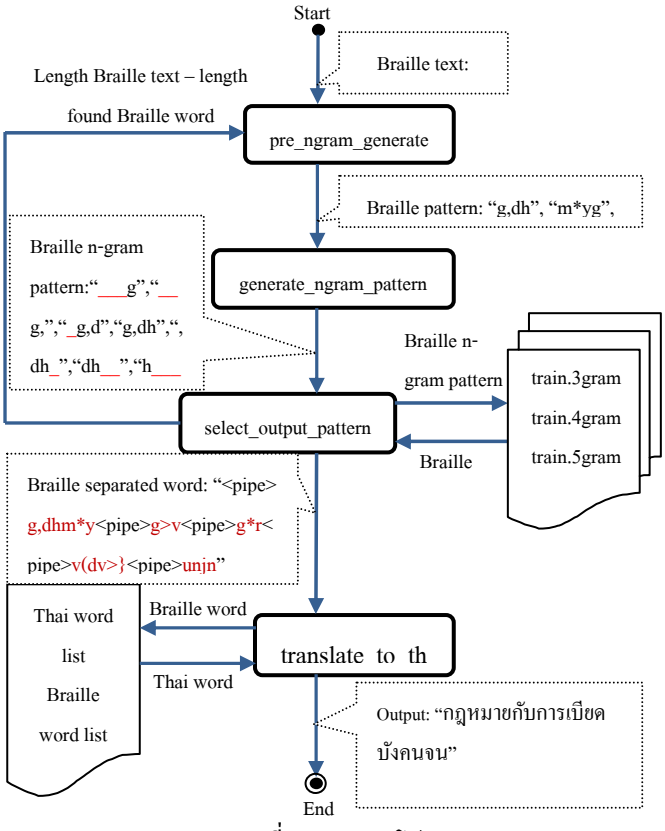
อุปกรณ์และวิธีวิจัย

เครื่องมือที่ใช้ในการวิจัยประกอบด้วย

- CMU-SLM toolkit
- คลังข้อความ BEST
- โปรแกรมแปลงภาษาไทยเป็นอักษรเบรลล์ภาษาไทย
- ชุดข้อมูลสำหรับทดสอบ
- พัฒนาโปรแกรมโดยใช้ภาษา C/C++

ภาพรวมของระบบ

ในบทความนี้ได้นำเสนอวิธีการปรับปรุงประสิทธิภาพในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยในรูปแบบ File-to-file โดยนำเทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำอักษรเบรลล์ภาษาไทยเพื่อลดความกำกวมที่เกิดขึ้นจากลักษณะการเขียนของอักษรเบรลล์ภาษาไทยระดับ 1 หลังจากนั้นจะแปลงให้เป็นภาษาไทยโดยใช้วิธีเทียบแบบคำต่อคำ ซึ่งจำเป็นต้องสร้างรายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยไว้ล่วงหน้า ในภาพที่ 1 แสดงภาพรวมของระบบที่ได้ออกแบบไว้



ภาพที่ 1 ภาพรวมโปรแกรม

ภาพที่ 1 แสดงองค์ประกอบต่างๆของระบบที่ได้ออกแบบไว้และลักษณะการเชื่อมต่อกันของฟังก์ชันหลักรวมทั้งการเชื่อมต่อองค์ประกอบอื่นๆ ที่จำเป็นต้องใช้ร่วมในการแปลงอักษรเบรลล์ภาษาไทยระดับ 1 ให้เป็นภาษาไทย ซึ่งประกอบไปด้วยส่วนต่างๆ ทั้งหมด 6 ส่วนด้วยกันคือ

1) ขั้นตอน pre_ngram_generate
2) ขั้นตอน generate_ngram_pattern
3) ขั้นตอน select_output_pattern
4) ไฟล์ฝึกฝนเอ็นแกรมโมเดล
5) ขั้นตอน translate_to_thai

6) รายการคำภาษาไทยและรายการคำอักษรเบรลล์ภาษาไทย

โดยอินพุต/เอาท์พุทและรายละเอียดแต่ละส่วนมีดังนี้

อินพุตและเอาท์พุทของระบบ

โปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยแบบ File-to-file ที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอไว้ในบทความนี้จะรับอินพุตเป็นไฟล์ข้อความธรรมดา ซึ่งภายในได้บรรจุข้อความที่เขียนให้อยู่ในรูปแบบอักษรเบรลล์ภาษาไทยระดับ 1 โดยการทำงานของโปรแกรมนี้จะประมวลผลข้อความอักษรเบรลล์ภาษาไทยระดับ 1 ครั้งละ 1 ประโยคโดยแบ่งแต่ละประโยคออกจากกันด้วยการเว้นวรรค เช่นเดียวกันกับการเขียนภาษาไทย เมื่อโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ได้นำเสนอนี้ประมวลผลเสร็จแล้ว จะได้เอาท์พุทเป็นประโยคภาษาไทยและนำประโยคภาษาไทยที่ได้นั้นเขียนใส่ในไฟล์เอาท์พุทตามที่ผู้ใช้งานโปรแกรมนี้ได้ตั้งชื่อไฟล์เอาไว้ ซึ่งเป็นไฟล์ข้อความภาษาไทยธรรมดา และการทำงานของโปรแกรมก็จะเป็นเช่นนี้ไปเรื่อยๆ จนกว่าข้อความอักษรเบรลล์ภาษาไทยระดับ 1 ในไฟล์อินพุตจะหมดลง

1. ขั้นตอน pre_ngram_generate

ขั้นตอนนี้รับอินพุตเป็นข้อความอักษรเบรลล์ภาษาไทยและแบ่งออกเป็นแต่ละประโยคโดยใช้การเว้นวรรค จากนั้นตัดประโยคที่รับมาเป็นส่วนๆ ตามขนาดของจำนวนแกรมที่ได้กำหนดไว้ เพื่อเพิ่มอัตราการจับคู่คำของแต่ละอักขระให้มากขึ้น และให้เอาท์พุทเป็นชุดของอักขระอักษรเบรลล์ภาษาไทยโดยแต่ละชุดมีความยาวเท่ากับจำนวนแกรมที่ได้กำหนดไว้

2. ขั้นตอน generate_ngram_pattern

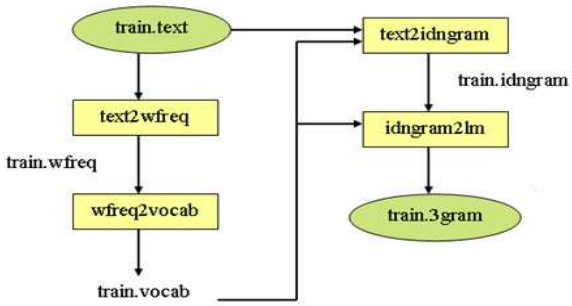
ขั้นตอนนี้รับอินพุตเป็นชุดของอักขระอักษรเบรลล์ภาษาไทยจากขั้นตอน pre_ngram_generate และนำมาสร้างรูปแบบเอ็นแกรม (n-gram pattern) โดยอ้างอิงกระบวนการสร้างรูปแบบเอ็นแกรมจากเว็บไซต์ http://www.hlt.nectec.or.th เพื่อนำไปใช้ค้นหาเปรียบเทียบรูปแบบของคำที่เขียนเรียงติดกันตามหลักการของเอ็นแกรมโมเดลในขั้นตอนต่อไป โดยให้เอาท์พุทเป็นชุดของรูปแบบ n-gram pattern

3. ขั้นตอน select_output_pattern

ขั้นตอนนี้รับอินพุตเป็นชุดรูปแบบของ n-gram pattern จากนั้นเปรียบเทียบแต่ละ pattern กับไฟล์ฝึกฝนเอ็นแกรมโมเดลโดยพิจารณาว่า pattern นั้นๆ ใกล้เคียงกับคำใดมากที่สุดและ pattern รูปแบบไหนมีค่าความน่าจะเป็นสูงที่สุด และหากมี pattern นั้นปรากฏอยู่ในประโยคก็เลือกเอา pattern นั้นเป็นคำเอาท์พุท หากไม่มีก็จะเลือกเอาคำที่มีความคล้ายคลึงกับ pattern นั่นมากที่สุดมาแทน แต่จะต้องมี pattern นั้นปรากฏอยู่ในประโยคด้วย จากนั้นใส่เครื่องหมาย “<pipe>” เพื่อเป็นสัญลักษณ์ในการแบ่งคำแต่ละคำออกจากกัน และส่งค่าความยาวของประโยคที่เหลือกลับไปให้ขั้นตอน pre_ngram_generate เพื่อสร้างชุดของอักขระใหม่และทำแบบนี้วนไปเรื่อยๆ จนกว่าจะจบประโยค เมื่อจบขั้นตอนนี้แล้วจะให้เอาท์พุทเป็นประโยคอักษรเบรลล์ที่มีการแบ่งคำแต่ละคำออกจากกันโดยใช้เครื่องหมาย “<pipe>” คั่นคำแต่ละคำ

4. ไฟล์ฝึกฝนเอ็นแกรมโมเดล

ไฟล์ที่ใช้ฝึกฝนเอ็นแกรมโมเดลนี้ สร้างขึ้นจากการใช้ชุดเครื่องมือ CMU-SLM toolkit โดยในการทดลองได้ใช้ขนาดของจำนวนแกรมเท่ากับ 3-gram, 4-gram และ 5-gram เพื่อเปรียบเทียบว่าจำนวนแกรมเท่าใดจึงจะให้ผลลัพธ์ที่ดีที่สุด ซึ่งวิธีการฝึกฝนเอ็นแกรมโมเดล (Training N-gram model) คือ กระบวนการทางสถิติเพื่อให้เอ็นแกรมโมเดลคำนวณค่าความน่าจะเป็นของคำหนึ่งๆ ว่าสามารถเกิดขึ้นร่วมกับคำใดได้บ้าง โดยขั้นตอนการฝึกฝนเอ็นแกรมโมเดลอ้างอิงมาจากเว็บไซต์ www.hlt.nectec.or.th ดังแสดงในภาพที่ 2



ภาพที่ 2 การฝึกฝนเอ็นแกรมโมเดล

โดยการฝึกฝนเอ็นแกรมโมเดลจะได้เป็นไฟล์นามสกุล .gram และในไฟล์ .gram นี้จะบรรจุค่าความน่าจะเป็นของคำนั้นๆ ว่ามีโอกาสที่จะเกิดขึ้นร่วมกับคำใดได้บ้างด้วยค่าความน่าจะเป็นเท่าไร และบรรจุค่าถ่วงน้ำหนัก (Back-off weight) ที่จะนำไปใช้ในกระบวนการ smoothing อีกด้วย ซึ่งมีการจัดเก็บให้อยู่ในรูปแบบของ ARPA format ซึ่งเป็นรูปแบบไฟล์มาตรฐานของโมเดลภาษา (Language Model)

5. ขั้นตอน translate_to_thai

ขั้นตอนนี้เป็นขั้นตอนแปลงข้อความอักษรเบรลล์ภาษาไทยที่ได้ตัดคำแล้วโดยใช้สัญลักษณ์ “<pipe>” คั่นระหว่างคำ แปลงให้เป็นภาษาไทย โดยใช้เทคนิควิธีการจับคู่คำ ซึ่งอาศัยรายการคู่คำอักษรเบรลล์ภาษาไทยและภาษาไทยที่สร้างไว้ก่อนหน้านี้แล้ว โดยแทนที่คำอักษรเบรลล์ภาษาไทยแต่ละคำด้วยคำภาษาไทยที่ได้จับคู่เอาไว้และลบเครื่องหมาย “<pipe>” ออก ซึ่งผลลัพธ์ท้ายสุดก็จะได้ข้อความภาษาไทยเพื่อนำไปเปรียบเทียบประสิทธิภาพกับโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม

6. รายการคู่คำอักษรเบรลล์ภาษาไทยและภาษาไทย

รายการคำอักษรเบรลล์ภาษาไทยและรายการคำภาษาไทยที่จัดเตรียมขึ้นมานี้ จะนำไปใช้ในขั้นตอน Translate_to_Thai เพราะเอ็นแกรมโมเดลนั้นมีความสามารถเฉพาะการหาขอบเขตของคำเท่านั้น ไม่สามารถใช้ในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยได้ นอกจากนี้ต้องการเปรียบเทียบประสิทธิภาพระหว่างโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม กับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอนี้ ซึ่งโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิมนั้นให้เอาท์พุทเป็นข้อความภาษาไทย ทำให้จำเป็นต้องมีขั้นตอน Translate_to_Thai เพิ่มเติมเข้ามาเพื่อทำให้เอาท์พุทเป็นข้อความภาษาไทยเหมือนกันเพื่อที่จะเปรียบเทียบกันได้โดยตรง

ดังนั้นจึงได้ออกแบบการแปลงจากอักษรเบรลล์ภาษาไทยเป็นภาษาไทยโดยใช้วิธีเทียบคำต่อคำหลังจากที่ใช้เทคนิคเอ็นแกรมในการตัดคำแล้ว ตัวอย่างเช่น เมื่อตัดคำอักษรเบรลล์ภาษาไทยได้เป็นคำว่า “g,dhm*y” ซึ่งตรงกันกับคำว่า “กฎหมาย” ในภาษาไทย เมื่อใช้วิธีเทียบคำต่อคำก็จะแทนที่คำ “g,dhm*y” ด้วยคำว่า “กฎหมาย” ลงไปแทนในตำแหน่งเดียวกันของข้อความทำให้ไฟล์เอาท์พุทเป็นข้อความภาษาไทย

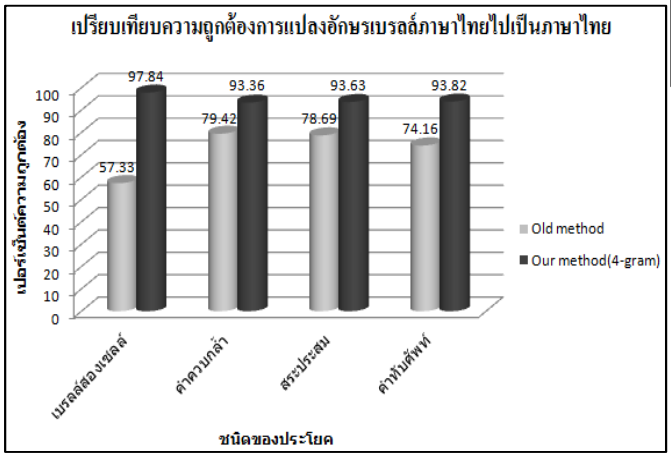
ผลการทดสอบเปรียบเทียบประสิทธิภาพ

ในการทดสอบเปรียบเทียบประสิทธิภาพระหว่างโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิมกับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำที่ได้นำเสนอนี้ จะทดสอบเปรียบเทียบทั้งหมด 3 ด้าน ด้วยกันคือ 1) ความถูกต้องของการแปลง 2) ปริมาณการใช้หน่วยความจำ และ 3) เวลาที่ใช้ในการแปลง ซึ่งมีรายละเอียดดังนี้

1) ด้านความถูกต้องของการแปลง

ในการทดสอบด้านความถูกต้องของการแปลงข้อความอักษรเบรลล์ภาษาไทยเป็นภาษาไทยได้ใช้ชุดข้อมูลทดสอบซึ่งแบ่งเป็น 4 ชุดด้วยกันคือ ชุดที่ 1) ข้อความที่ใช้อักษรเบรลล์สองเซลล์ จำนวน 8,713 บรรทัด ชุดที่ 2) ข้อความที่ใช้คำควบกล้ำ จำนวน 231,115 บรรทัด ชุดที่ 3) ข้อความที่ใช้สระประสม จำนวน 219,022 บรรทัด ชุดที่ 4) ข้อความที่ใช้คำทับศัพท์ จำนวน 18,881 บรรทัด โดยแต่ละบรรทัดจะมี 1 ประโยค และในแต่ละประโยคจะบรรจุคำประเภทนั้นๆ อย่างน้อย 1 คำ

โดยชุดข้อมูลทดสอบแต่ละชุดจะมีลักษณะการใช้ภาษาเขียนแบบต่างๆ แบ่งเป็น 3 ประเภทคือ 1) ภาษาเขียนแบบเป็นทางการ 2) ภาษาเขียนแบบไม่เป็นทางการและ 3) ภาษาข่าว ทั้งนี้เพื่อต้องการให้ข้อมูลทดสอบมีลักษณะใกล้เคียงกับการเขียนโดยทั่วไปมากที่สุด โดยอ้างอิงจากโครงสร้างการจัดเก็บข้อความของคลังข้อความ BEST และผลการทดลองแสดงในภาพที่ 3

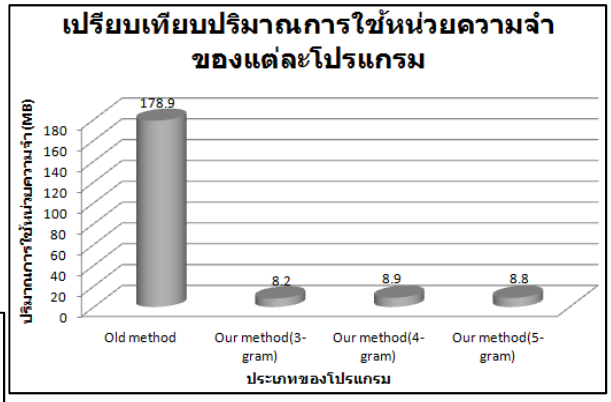


ภาพที่ 3 ผลการเปรียบเทียบความถูกต้อง

โดยผลการทดลองพบว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยตัดคำสามารถแปลงข้อความที่มีการใช้คำควบกล้ำ , สระประสม, อักษรเบรลล์สองเซลล์ และคำทับศัพท์ได้ถูกต้องมากกว่าโปรแกรมเดิมคิดเป็น 13.94%, 14.94%, 40.51% และ 19.66% ตามลำดับเมื่อใช้จำนวนแกรมเท่ากับ 4-gram

2) ด้านปริมาณการใช้หน่วยความจำ

ในการทดสอบด้านปริมาณการใช้หน่วยความจำจะเริ่มวัดปริมาณหน่วยความจำที่โปรแกรมต้องใช้ตั้งแต่เริ่มเรียกใช้งานโปรแกรมจนกระทั่งโปรแกรมสิ้นสุดการทำงาน โดยใช้ชุดข้อมูลทดสอบเช่นเดียวกันกับวิธีการวัดความถูกต้องของการแปลง และวัดทั้งหมด 10 ครั้งและหาค่าเฉลี่ยปริมาณการใช้หน่วยความจำของโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิมกับโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ โดยแบ่งเป็น เมื่อใช้จำนวนแกรมเท่ากับ 3-gram, 4-gram และ 5-gram ซึ่งผลที่ได้แสดงในภาพที่ 4

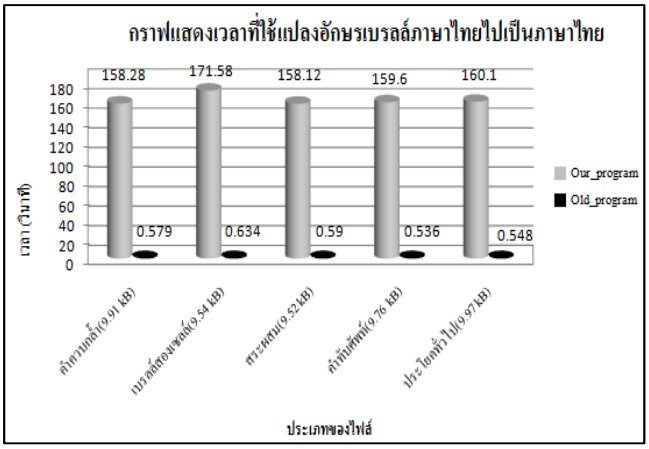


ภาพที่ 4 ผลเปรียบเทียบปริมาณการใช้หน่วยความจำ

โดยผลการทดลองพบว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยตัดคำมีปริมาณการใช้หน่วยความจำน้อยกว่าโปรแกรมเดิมคิดเป็น 21.81 เท่า ซึ่งเหมาะสมต่อการนำไปใช้บนอุปกรณ์ Braille note ซึ่งมีทรัพยากรที่จำกัด

3) ด้านเวลาที่ใช้ในการแปลง

ในด้านเวลาที่ใช้ในการแปลงจะใช้วิธีจับเวลาตั้งแต่เรียกใช้โปรแกรมจนกระทั่งโปรแกรมสิ้นสุดการทำงาน โดยไฟล์ที่นำมาทดสอบจะแบ่งออกเป็น 5 ประเภทคือ 1) ประโยคที่ใช้คำควบกล้ำ 2) ประโยคที่ใช้อักษรเบรลล์สองเซลล์ 3) ประโยคที่ใช้สระประสม 4) ประโยคที่ใช้คำทับศัพท์ และ 5) ประเภทประโยคทั่วไป ซึ่งแต่ละประเภทจะใช้ไฟล์ทั้งหมด 5 ไฟล์และประเภทประโยคทั่วไปใช้ทั้งหมด 10 ไฟล์โดยมีขนาดประมาณ 10 KB (ประมาณ 220 บรรทัด) โดยแต่ละบรรทัดจะมี 1 ประโยคและในแต่ละประโยคมีคำประเภทนั้นๆ อย่างน้อย 1 คำ และทดสอบทั้งหมด 10 ครั้งและหาค่าเฉลี่ย ทั้งนี้เพื่อต้องการให้ใกล้เคียงกับการนำโปรแกรมไปใช้งานจริงมากที่สุด ซึ่งผลที่ได้แสดงในภาพที่ 5



ภาพที่ 5 ผลการเปรียบเทียบเวลาที่ใช้ในการแปลง

โดยผลการทดลองพบว่าโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยตัดคำใช้เวลาในการแปลงมากกว่าโปรแกรมเดิมคิดเป็น 272.71 เท่า จึงเหมาะในการใช้แก้ปัญหาเฉพาะจุดร่วมกับโปรแกรมแปลงตัวอื่น

ผลการวิจัยและการอภิปรายผล

จากการพัฒนาโปรแกรมแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมเข้ามาช่วยในการตัดคำ เพื่อแก้ไขปัญหาของโปรแกรมแปลงเบรลล์เป็นแอสกีภาษาไทยแบบ File-to-file เดิม โดยสามารถแก้ไขปัญหาการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยได้เป็นอย่างดี และใช้ปริมาณหน่วยความจำน้อยกว่าโปรแกรมเดิมมากซึ่งใช้กฎไวยากรณ์ทางภาษาเข้ามาควบคุมการแปลง จึงมีความเหมาะสมมากกว่าในการนำไปใช้ในอุปกรณ์ Braille Note ที่มีทรัพยากรอยู่อย่างจำกัด

อย่างไรก็ตามการใช้งานโปรแกรมที่ได้พัฒนาขึ้นมาใหม่นี้ก็มีข้อด้อย คือเวลาที่ใช้ในการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยใช้เวลาแปลงที่มากกว่าโปรแกรมเดิมมาก ทำให้โปรแกรมนี้เหมาะกับการใช้แปลงไฟล์ที่มีขนาดไม่ใหญ่มาก ซึ่งผู้ใช้งานสามารถยอมรับได้ แต่หากเป็นไฟล์ที่มีขนาดใหญ่จะใช้เวลาแปลงนาน จึงนับว่าเป็นข้อด้อยของโปรแกรมนี้ที่ควรนำไปปรับปรุงให้ดีกว่าเดิม

สรุป

บทความนี้เสนอการนำเทคนิคเอ็นแกรมโมเดลมาประยุกต์ใช้ตัดคำอักษรเบรลล์ภาษาไทยเพื่อลดความกำกวมของภาษาก่อนที่จะแปลงไปเป็นภาษาไทย ซึ่งวิธีการนี้สามารถช่วยแก้ปัญหาการแปลงอักษรเบรลล์ภาษาไทยเป็นภาษาไทยได้เป็นอย่างดี โดยเฉพาะปัญหาที่เกี่ยวข้องกับการแปลงคำควบกล้ำ คำที่มีการใช้สระ-ประสม อักษรเบรลล์สองเซลล์และคำทับศัพท์ และจำนวนแกรมที่เหมาะสมสำหรับใช้ตัดคำอักษรเบรลล์ภาษาไทยคือ 4 แกรม

ข้อเสนอแนะ

การนำเทคนิคเอ็นแกรมเข้ามาช่วยตัดคำอักษรเบรลล์ภาษาไทยก่อนที่จะแปลงให้เป็นภาษาไทยสามารถช่วยลดความกำกวมของคำได้เป็นอย่างดี แต่ยังคงมีประเด็นเรื่องเวลาที่ใช้ในการแปลงที่ทำได้ช้ากว่าโปรแกรมเดิมมาก ซึ่งไม่เหมาะสมในการนำไปใช้แปลงไฟล์ขนาดใหญ่ทั้งหมด แต่ควรใช้งานร่วมกับวิธีการอื่นๆ เพื่อเพิ่มความถูกต้องในการแปลงคำควบกล้ำ คำที่มีการใช้สระประสม อักษรเบรลล์สองเซลล์และคำทับศัพท์

เอกสารอ้างอิง

วรพล ทินกรสูติบุตร. (2551). การแปลงเบรลล์และแอสกีภาษาไทยแบบทันทีทันใด. วิทยานิพนธ์มหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลา-นครินทร์

Stanley F. Chen and Joshua Goodman. (1996). An Empirical Study of Smoothing Techniques for Language Modeling, Aiken Computational Laboratory Harvard University.

N-gram. (2011). Retrieved September 5, 2011, from http://en.wikipedia.org/wiki/N-gram

BEST Corpus. (2011). Retrieved September 5, 2011, from http://thailang.nectec.or.th/2009/index.php

Philip Clarkson. (2011). CMU-SLM language model toolkit. Retrieved

September 26, 2011, from http://mi.eng.cam.ac.uk/~prc14/toolkit.html

Training N-gram model. (2011). Retrieved September 5, 2011, from http://www.hlt.nectec.or.th/speech/index.php?option=com_content&view=article&id=70&Itemid=94

Word wrap. (2011). Retrieved September 5, 2011, from http://en.wikipedia.org/wiki/Word_wrap

Language model. (2011). Retrieved September 5, 2011, from http://www.hlt.nectec.or.th/speech/index.php?option=com_content&view=article&id=70&Itemid=94

ภาคผนวก ข : ตารางผลการทดลอง

ตารางที่ 1 ผลการทดลองปริมาณการใช้หน่วยความจำ

ครั้งที่	วิธีการเดิม		วิธีการใหม่					
			3-แกรม		4-แกรม		5-แกรม	
	คำควบกล้ำ (MB)	สระประสม (MB)	คำควบกล้ำ (MB)	สระประสม (MB)	คำควบกล้ำ (MB)	สระประสม (MB)	คำควบกล้ำ (MB)	สระประสม (MB)
1	179	178.9	8.2	8.2	8.9	8.9	8.8	8.8
2	178.8	179.2	8.2	8.3	8.8	9.3	8.9	8.9
3	179.1	178.6	8	8.4	8.7	8.7	8.7	8.7
4	178.7	178.8	8.5	8	8.8	8.9	8.6	8.8
5	178.9	178.9	8.2	8.1	9.2	8.8	8.8	8.8
6	178.7	178.7	8.1	8.2	9	9.2	8.8	8.7
7	179.2	179.1	8.3	8.1	8.9	9	8.7	9
8	178.9	179.3	8	8.3	8.8	8.7	8.7	8.7
9	178.7	178.8	8.3	8.2	8.8	8.7	8.9	8.8
10	178.9	178.6	8.1	8.3	9	8.8	9	8.9
เฉลี่ย	178.9	178.9	8.19	8.21	8.89	8.9	8.79	8.81
SD.	0.1729	0.2424	0.1524	0.1197	0.1449	0.2108	0.1197	0.0994
Max	179.2	179.3	8.5	8.4	9.2	9.3	9	9
Min	178.7	178.6	8	8	8.7	8.7	8.6	8.7

ตารางที่ 2 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์คำควบกล้ำ

ครั้งที่	ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์คำควบกล้ำ (เวลา : วินาที)									
	วิธีการเดิม					วิธีการใหม่				
	ไฟล์ที่					ไฟล์ที่				
	1	2	3	4	5	1	2	3	4	5
1	0.55	0.75	0.74	0.44	0.43	149	174	160	155	150
2	0.54	0.74	0.73	0.43	0.42	148	173	162	156	152
3	0.53	0.73	0.72	0.42	0.41	150	174	161	156	152
4	0.53	0.73	0.73	0.43	0.42	149	174	161	156	152
5	0.56	0.76	0.75	0.45	0.44	148	175	160	156	152
6	0.54	0.74	0.73	0.43	0.42	149	175	160	155	152
7	0.56	0.76	0.75	0.45	0.44	149	176	161	156	152
8	0.54	0.74	0.73	0.43	0.42	148	174	160	157	153
9	0.55	0.75	0.74	0.44	0.43	150	174	161	156	152
10	0.56	0.76	0.75	0.45	0.44	149	173	161	155	151
เฉลี่ย	0.546	0.746	0.737	0.437	0.427	148.9	174.2	160.7	155.8	151.8
SD.	0.0117	0.0117	0.0106	0.0106	0.0106	0.7379	0.9189	0.6749	0.6325	0.7888
Max	0.56	0.76	0.75	0.45	0.44	150	176	162	157	153
Min	0.53	0.73	0.72	0.42	0.41	148	173	160	155	150
จำนวนบรรทัด	198	205	200	178	182	198	205	200	178	182
Size(KB)	9.76	10	10	9.83	9.95	9.76	10	10	9.83	9.95

ตารางที่ 3 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์สระประสม

ครั้งที่	ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์สระประสม (เวลา : วินาที)									
	วิธีการเดิม					วิธีการใหม่				
	ไฟล์ที่					ไฟล์ที่				
	1	2	3	4	5	1	2	3	4	5
1	0.55	0.54	0.53	0.73	0.63	158	155	142	194	140
2	0.54	0.53	0.52	0.72	0.62	158	155	142	195	141
3	0.54	0.53	0.52	0.72	0.62	158	155	142	196	140
4	0.53	0.52	0.51	0.71	0.61	159	155	142	195	140
5	0.56	0.55	0.54	0.74	0.64	158	155	142	194	141
6	0.53	0.52	0.51	0.71	0.61	159	155	141	195	141
7	0.54	0.53	0.52	0.72	0.62	158	155	143	195	140
8	0.55	0.54	0.53	0.73	0.63	159	155	142	195	140
9	0.56	0.55	0.54	0.74	0.64	158	154	142	195	141
10	0.54	0.53	0.52	0.72	0.62	158	156	142	195	140
เฉลี่ย	0.544	0.534	0.524	0.724	0.624	158.3	155	142	194.9	140.4
SD.	0.0107	0.0107	0.0107	0.0107	0.0107	0.4830	0.4714	0.4714	0.5676	0.5164
Max	0.56	0.55	0.54	0.74	0.64	159	156	143	196	141
Min	0.53	0.52	0.51	0.71	0.61	158	154	141	194	140
จำนวนบรรทัด	197	197	191	202	198	197	197	191	202	198
Size(KB)	9.86	9.64	9.17	9.19	9.76	9.86	9.64	9.17	9.19	9.76

ตารางที่ 4 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์อักษรเบรลล์สองเซลล์

ครั้งที่	ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์อักษรเบรลล์สองเซลล์ (เวลา : วินาที)									
	วิธีการเดิม					วิธีการใหม่				
	ไฟล์ที่					ไฟล์ที่				
	1	2	3	4	5	1	2	3	4	5
1	0.55	0.56	0.57	0.77	0.76	161	162	168	184	182
2	0.53	0.54	0.55	0.75	0.74	160	164	170	184	182
3	0.56	0.57	0.58	0.78	0.77	160	162	170	184	180
4	0.54	0.55	0.56	0.76	0.75	161	162	168	185	182
5	0.53	0.54	0.55	0.75	0.74	160	163	169	184	178
6	0.55	0.56	0.57	0.77	0.76	163	162	168	184	181
7	0.54	0.55	0.56	0.76	0.75	162	163	169	185	181
8	0.55	0.56	0.57	0.77	0.76	160	164	169	184	182
9	0.53	0.54	0.55	0.75	0.74	161	163	167	184	187
10	0.54	0.55	0.56	0.76	0.75	160	164	168	184	179
เฉลี่ย	0.542	0.552	0.562	0.762	0.752	160.8	162.9	168.6	184.2	181.4
SD.	0.0103	0.0103	0.0103	0.0103	0.0103	1.0328	0.8756	0.9661	0.4216	2.4129
Max	0.56	0.57	0.58	0.78	0.77	163	164	170	185	187
Min	0.53	0.54	0.55	0.75	0.74	160	162	167	184	178
จำนวนบรรทัด	211	211	245	214	217	211	211	245	214	217
Size(KB)	9.14	9.17	9.6	9.92	9.87	9.14	9.17	9.6	9.92	9.87

ตารางที่ 5 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์คำทับศัพท์

ครั้งที่	ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทไฟล์คำทับศัพท์ (เวลา : วินาที)									
	วิธีการเดิม					วิธีการใหม่				
	ไฟล์ที่					ไฟล์ที่				
	1	2	3	4	5	1	2	3	4	5
1	0.53	0.52	0.51	0.61	0.41	163	160	159	171	146
2	0.56	0.55	0.54	0.64	0.44	161	160	157	172	147
3	0.55	0.54	0.53	0.63	0.43	162	160	157	171	147
4	0.54	0.53	0.52	0.62	0.42	163	161	157	172	146
5	0.56	0.55	0.54	0.64	0.44	162	160	157	172	148
6	0.56	0.56	0.55	0.65	0.45	162	160	158	171	146
7	0.56	0.56	0.55	0.65	0.45	163	161	157	170	146
8	0.55	0.54	0.53	0.63	0.43	161	161	157	172	148
9	0.53	0.52	0.51	0.61	0.41	162	161	158	171	146
10	0.54	0.53	0.52	0.62	0.42	162	161	157	172	146
เฉลี่ย	0.548	0.54	0.53	0.63	0.43	162.1	160.5	157.4	171.4	146.6
SD.	0.0123	0.0149	0.0149	0.0149	0.0149	0.7379	0.5270	0.6992	0.6992	0.8433
Max	0.56	0.56	0.55	0.65	0.45	163	161	159	172	148
Min	0.53	0.52	0.51	0.61	0.41	161	160	157	170	146
จำนวนบรรทัด	193	191	186	196	185	193	191	186	196	185
Size(KB)	9.98	9.68	9.96	10	9.19	9.98	9.68	9.96	10	9.19

ตารางที่ 6 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประโยคทั่วไปของวิธีการเดิม

ครั้งที่	ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทประโยคทั่วไป (เวลา : วินาที)									
	โปรแกรมแปลงเบรลล์เป็นรหัสแอสกีภาษาไทยแบบ File-to-file (วิธีการเดิม)									
	ไฟล์ที่									
	1	2	3	4	5	6	7	8	9	10
1	0.54	0.55	0.55	0.56	0.55	0.54	0.54	0.55	0.56	0.56
2	0.54	0.55	0.54	0.55	0.54	0.55	0.55	0.54	0.55	0.54
3	0.53	0.56	0.54	0.54	0.56	0.55	0.54	0.54	0.54	0.53
4	0.56	0.53	0.53	0.56	0.55	0.56	0.54	0.56	0.56	0.55
5	0.54	0.54	0.54	0.55	0.54	0.54	0.56	0.55	0.53	0.56
6	0.56	0.56	0.56	0.56	0.54	0.55	0.56	0.56	0.54	0.56
7	0.55	0.56	0.54	0.56	0.56	0.54	0.55	0.55	0.55	0.53
8	0.54	0.54	0.53	0.55	0.55	0.55	0.54	0.55	0.56	0.54
9	0.53	0.56	0.56	0.54	0.54	0.54	0.55	0.54	0.55	0.55
10	0.54	0.55	0.56	0.55	0.54	0.56	0.55	0.56	0.53	0.54
เฉลี่ย	0.543	0.55	0.545	0.552	0.547	0.548	0.548	0.55	0.547	0.546
SD.	0.0106	0.0105	0.0118	0.0079	0.0082	0.0079	0.0079	0.0082	0.0116	0.0117
Max	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
Min	0.53	0.53	0.53	0.54	0.54	0.54	0.54	0.54	0.53	0.53
จำนวนบรรทัด	198	224	214	201	214	219	218	214	221	195
Size(KB)	9.92	9.97	9.98	9.91	9.96	10.00	10.00	10.00	9.95	9.98

ตารางที่ 7 ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประโยคทั่วไปของวิธีการใหม่

ครั้งที่	ผลการทดลองเปรียบเทียบเวลาที่ใช้ในการแปลงประเภทประโยคทั่วไป (เวลา : วินาที)									
	โปรแกรมแปลงอักษรเบรลล์ภาษไทยเป็นภาษาไทยที่ใช้เทคนิคเอ็นแกรมในการตัดคำ(วิธีการใหม่)									
	ไฟล์ที่									
	1	2	3	4	5	6	7	8	9	10
1	155	159	163	162	153	159	151	167	165	166
2	156	159	162	162	153	159	151	168	166	167
3	156	159	163	162	154	159	150	168	166	165
4	156	161	162	161	154	159	150	166	167	167
5	155	159	162	163	153	159	150	167	166	165
6	155	159	162	162	154	159	150	167	166	165
7	156	159	164	161	154	160	150	168	166	166
8	155	160	162	162	153	159	150	166	165	167
9	156	159	163	163	154	158	150	167	166	165
10	155	160	162	161	154	160	150	166	166	166
เฉลี่ย	155.5	159.4	162.5	161.9	153.6	159.1	150.2	167	165.9	165.9
SD.	0.5270	0.6992	0.7071	0.7379	0.5164	0.5676	0.4216	0.8165	0.5676	0.8756
Max	156	161	164	163	154	160	151	168	167	167
Min	155	159	162	161	153	158	150	166	165	165
จำนวนบรรทัด	198	224	214	201	214	219	218	214	221	195
Size(KB)	9.92	9.97	9.98	9.91	9.96	10.00	10.00	10.00	9.95	9.98

ภาคผนวก ค : รายการคำที่มีความผิดพลาดประเภทต่างๆ

ตารางที่ 8 รายการคำที่โปรแกรม Thai2brl แปลงผิดพลาด

คำไทย	Thai2brl แปลงผิด	ที่ถูกต้องควรเป็น
	เบรลล์ไทยระดับ 1	เบรลล์ไทยระดับ 1
ข้างเป้ออกมา	k4*}&%4ogm*	k4*}f&4oogm*
มาเทอะไร	m*)%a:r	m*f)oa:r
เฉลี่ยกันไป	f/l29yg>n:&	/l(9yg>n:&
ไกล่เกลี่ย	:gl9fgl29y	:gl9gl(9y
กระเป๋าเป้อย่าง	gra&68&%4y9*}	gra%68f&4oy9*}
เกลี่ยดิน	fgl29ydbn	gl(9dbn
ปนเปอยู่	&n&%y39	&nf&oy39
ลังเลอยู่	l>}l%y39	l>}floy39
อ่อนเพลีย	o9onf?l2y	o9on?l(
เกลี่ย	fgl29y	gl(9y
ร่อนเร่อยู่	r9onr%9y39	r9onfr9oy39
ทะเลอยู่ไหน	)al%y39:hn	)afaoy39:hn
สังเคราะห์	s>}ur*ah0	s>}uroah0
เพลียกาย	f?l2yg*y	?l(g*y
คำว่า	u*w9*	uzw9*
โอ้เอ้อยู่	io4o%4y39	io4fo4oy39
เมื่อ	fm59o	mq9
คะเนอยู่	uan%y39	uafnoy39
จะเข้อยู่	jak%y39	jafk4oy39
เต้อยาก	|%4y*g	f|4oy*g
กะลิ้มกะเหลี่ย	galb4myafhl29y	galb4mgahl(9
คลอเคลีย	uloful2y	uloul(
ระเหยอยู่	rahy%y39	rah%yoy39

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ก.....-ุ.....ง	g.....-c.....}	พกลม	?glm	รีโน	r2in
สังคราะห์แสง	s>}ur*ah0<s}	ชะมบ	+amv	วิภว	wb,?w
กฎุกกโรหินี	g,|cggirhbn2	รู้ธ	r340)	รียก	r2yg
ใจจรดใจจ่อ	:1jjrd:1jj9o	ตร่ำ	|r9z	รีรา	r2r*
เพราะฉนั้น	?roa/n>4n	คันธ	u>n0)	ไฮเด	:=fd
แม่ม่าย	ffm9m9*y	ตริว	|rbw	กู๊ก	g37g
กรรมการก	grrmg*rg	วิสย	wbsy	รึ้ง	r{4}
แหยแฝ่น	<hy<fx9n	ถามะ	t*ma	แหวด	<hwd
เข้าเกีย	k64g(	เมรย	fmry	รื้น	r54n
กรธนกิจ	gr0)ngbj	เมลก	fmlg	ช่อย	+9oy
เร้าโลม	r64ilm	บแรง	v<r}	มูฬห	m3,lh
บ้านเปา	v4*n&6	ปฤษฐ	&r1-s,t	ไฮ้ส	:=4s
พัฒนากร	?>-)n*gr	ปล่ง	&l9}	โหนะ	ihna
ไหวพวิบ	:hw?wbv	เมหน	fmhn	โหม่	ihm9
เบอร์นี	v%r0n2	พชระ	?+ra	จ้าน	j4*n
นั้นเนอ	n>4nn%	ถ่าว	t9*w	ชอว์	+ow0
ประกาศก	&rag*,sg	จ่าว	j9*w	ฝอ.	xo.
วันจักร	w>nj>gr	สารท	s*r)	จรส	jrs
ปากไปล่	&*g:&l9	พญาณ	?,y*,n	ปร-	&r-
ชูเมือง	+3mq}	เม้า	m64	กำม	gzm
โอ๊ดโท้	io7di)4	เรขา	frk*	ปรด	&rd
ซองแมว	!o}<mw	โขมย	ikmy	รชะ	r+a
เบอร์น	v%r0n	เมาะ	moa	รณ์	r,n0
จัดสัน	j>ds>n	โขลน	ikln	ปาป	&*&
เสียกล	s(gl	โขษม	ik-sm	อาด	o*d
ไซยยาฟ	:!yy*$	วุ้ง	wc4}	แจะ	j<a
ประสาร	&ras*r	เมิล	m%l	วฤก	wr1g
โสเหล่	isfhl9	พณิช	?,nb+	ปรม	&rm
ละเมิน	lam%n	ถิ่ง	tb9}	มบธ	mv0)
กับแก้	g>v<g4	เร่ม	fr9m	วศค	w,su

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ลูกล่า	l3gl9*	พู้น	?34n	พีร	?2r
ปัญญาส	&>,y,y*s	เมือ	mq	มมข	mmk
ข้าวกะ	k4*wga	พนจร	?njr	ร้น	r4n
เป็นไฟ	f&'n:$	ชังฆ	+>},u	รนด	rnd
เหลือม	hlqm	ปลาก	&l*g	คาธ	u*0)
ว่ายตา	w9*y|*	เราะ	roa	รบส	rvs
เด็กชา	fd'g+*	เวตร	fw|r	รปภ	r&,?
วิชาธร	wb+*0)r	บอตอ	vo|o	ทอย	)oy
วิทยุต	wb)yc|	ถึง่	t{}9	วหา	wh*
เดินหน	d%nhn	ธรฺม	0)rm	วอค	wou
มาเรีย	m*r(	พนาย	?n*y	ฆุน	,kcn
กรี็ดก	gr27dg	ธอโร	0)oir	มรณ	mr,n
โดโด้	idid4	บ้อม	v4om	เฆา	,k6
มีเรีย	m2r(	บ๋อม	v8om	ปาร	&*r
น้ำตาย	n4z|*y	ธังก	0)>}g	โตก	i|g
เผ่าไท	p69:)	แครต	<ur|	คคน	uun
โศกน้ำ	i,sgn4z	เวย์	fwy0	วอล	wol
บรรดาก	vrrd*g	เวรา	fwr*	ขอน	kon
เกลือก	glqg	เพตร	f?|r	คฤห	ur1h
อุ้มสม	oc4msm	ปลาต	&l*|	นัฑ	n>,)
เคลิบ	ul%v	กูรม	g3rm	คล-	ul-
เยียน	y(n	ชัปน	+>&n	นัน	n>n
เยียบ	y(v	เพรา	?r6	รพา	r?*
ทาบทา	)*v)*	กูวา	g3w*	อาม	o*m
เจ๊ยุ	fj7yc	เวสิ	fwsb	โตย	i|y
เจรี็ด	fjr'd	เวหน	fwhn	gur	gur
เยียว	y(w	พยาล	?y*l	กลี	gl2
หนั่น	hn>9n	ไขมา	:km*	คงธ	u}0)
หน้ำ	hn>4*	ข้อน	k4on	มลน	mln
ทารพี	)*r?2	แคละ	ul<a	ศูล	,s3l

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
เคห์ล	fuh0l	แคแล	<u<l	รยะ	rya
กุเทพ	gcf)?	ธารณ	0)*r,n	จลา	jl*
ผัวนา	pen*	ไข้ว	:k4w	ชาด	+*d
แย่แส	<y9<s	พรพล	?r?l	ฝี่	x29
เวย์น	fwy0n	เวีย	w(	ฝึก	x2g
เวหาส	fwh*s	ไขษย	:k-sy	คาม	u*m
เวิ้ก	w%4g	ถุ้ง	tc4}	ใต่	:1|9
เวียก	w(g	ถู๊ก	t37g	มวก	mwg
กุมุท	gcmc)	ธิณะ	0)b,na	แห้	<h4
เพอิญ	f?ob,y	แวดี	<wd2	นัว	ne
กทรรป	g)rr&	แวตร	<w|r	ซาซ	!*!
เคียร	u(r	ปลีก	&l2g	มสด	msd
หมดพก	hmd?g	ขึ้น	,k{4n	ษัฏ	-s>,|
เจะกุ	jfagc	ข้อย	k4oy	ษัฑ	-s>,)
เจะคะ	jfada	ค้าบ	u4*v	ษัณ	-s>,n
กุรุง	gcrc}	คคนะ	uuna	ษัษ	-s>-s
เคื้อ	uq4	ธีโอ	0)2io	แ-ะ	<-a
แคแกล	<u<gl	ขอลอ	kolo	โ-ะ	i-a
กุรุร	gcrcr	ธุมา	0)cm*	จิร	jbr
โวการ	iwg*r	จิตต	jb||	คณน	u,nn
โวทาน	iw)*n	ไวโพ	:wi?	รสก	rsg
กุเรา	gcr6	ชากา	+*g*	แม๊	<m7
ก้นขบ	g4nkv	เพิ่ม	?%m	สกฏ	sg,|
ไวคอม	:wuom	ชาคร	+*ur	ชาร	+*r
ผี้ว์	p24w0	ปวาฬ	&w*,l	กรา	gr*
ทำนูล	)zn3l	เธนุ	f0)nc	ปีบ	&2v
แพทยา	<?)y*	คงที	u})2	ใหน	:1hn
กุเวร	gcfwr	ปสาท	&s*)	กัว	ge
แพรมน	<?rmn	ปสาน	&s*n	ปีป	&2&
แคบิด	<uvbd	โธวน	i0)wn	ใหม	:1hm

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ศตปที	,s|&)2	เพื่อ	?q	ขัช	k>+
รงกุ์	r}gc0	อจลา	ojl*	โรษ	ir-s
รจเลข	rjflk	ปหาร	&h*r	ใหว	:1hw
ทำไห้	)z:h4	อจิร	ojbr	ชาล	+*l
คลิ๊ป	ulb7&	ดยุก	dycg	ปีพ	&2,l
นายดุ	n*ydc	ดยุค	dycu	สขะ	ska
โพรดิ	i?rdb	เก่น	fg9n	สขา	sk*
กูซิน	g3!bn	แพทย	<?)y	สขิ	skb
เผนิก	fpnbg	แม่เ	<m9f	ถกล	tgl
เผลี๊จ	fpl'j	ศยนะ	,syna	กกก	ggg
เผลี๊ด	fpl'd	อทระ	o)ra	กรร	grr
กูดกบ	g3dgv	ศยาม	,sy*m	รอน	ron
นายติ	n*y|b	พรอก	?rog	ก๊า	g7*
รมณีย	rm,n2y	ศรัท	,sr>)	ปุฏ	&c,|
เผลิด	fplbd	อธิป	o0)b&	ฤชุ	r1+c
ทิพยา	)b?y*	ขะยง	kay}	ฤตุ	r1|c
รมิดา	rmbd*	ป้อย	&4oy	ฉนำ	/nz
เผอิล	fpobl	พรอด	?rod	ฉบบ	/vv
เผิ่ง	p%9}	ขะยม	kaym	ฤษภ	r1-s,?
เผิ้ง	p%4}	ศราพ	,sr*?	รอม	rom
เผียน	p(n	อนงค	on}u	กกพ	gg?
ไพเรอ	:?r%	อนยะ	onya	หตะ	h|a
ศวศุร	,sw,scr	กั่ว	ge9	หทย	h)y
ศวัสน	,sw>sn	อนัม	on>m	ฉบำ	/vz
นายทร	n*y)r	ทนอด	)nod	ฉมบ	/mv
ไพศาข	:?,s*k	แม้ง	<m4}	หนง	hn}
ศอกกำ	,soggz	จินต	jbn|	กกล	ggl
บาปวน	v*&wn	กัษณ	g>-s,n	เกล	fgl
คลุ้ม	ulc4m	อนึก	on{g	ฉมา	/m*
หย็อย	hy'oy	เกร่	fgr9	ฉมำ	/mz

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
หย้าว	hy4*w	อนุช	onc+	ลกร	lgr
บ่าลง	v9*l}	ปักข	&>gk	ปุร	&cr
บาลอก	v*log	ปักษ	&>g-s	ลขข	lkk
บาลาน	v*l*n	กาคำ	g*uz	ลคร	lur
กระลา	gral*	อนุส	oncs	ส่ญ	s9,y
บาสมา	v*sm*	ศลัถ	,sl>t	ลฆุ	l,uc
หรัสว	hr>sw	ไพโค	:?iu	สดก	sdg
รโหคต	rihu|	สำหา	szh*	กกะ	gga
กูดภู	g3d,?3	ศวัส	,sw>s	สดน	sdn
ทีแท้	)2<)4	ศศธร	,s,s0)r	ล่ง	l9}
กระลำ	gralz	ทรรป	)rr&	นีร	n2r
กมเลศ	gmfl,s	ตะกู	|ag3	สดำ	sdz
หลงจิ	hl}jb	ตะแก	|a<g	สดี	sd2
หลงตา	hl}|*	บาจก	v*jg	สต์	s|0
กระลี	gral2	ทรวด	)rwd	สตก	s|g
หลงลม	hl}lm	ทรหน	)rhn	สตน	s|n
ตักษณ	|>g-s,n	ทรหู	)rh3	ก่ง	g9}
ฝนฝ้า	xnx4*	ปัญจ	&>,yj	ก้ง	g4}
ฟ้ตวา	$>|w*	กาซิ	g*!b	ซิน	!bn
เซราะ	f!r*a	ศัพท	,s>?)	สตอ	s|o
ฝรั้ง	xr>4}	ศัลก	,s>lg	นุง	nc}
กูรมะ	g3rma	บาทบ	v*)v	ซิม	!bm
กูวัง	g3w>}	เกลศ	fgl,s	ฉศก	/,sg
กระวี	graw2	ศากย	,s*gy	ล่น	l9n
กรณีย	gr,n2y	ศาฎก	,s*,dg	พฤฒ	?r1-)
รอบคี	rovu2	ศานต	,s*n|	พฤต	?r1|
ทีโรส	)2irs	ช่าว	+9*w	ซิล	!bl
กรบูร	grv3r	ทราก	)r*g	มหู	mh3
บินมะ	vbnma	ศาพร	,s*?r	ลบม	lvm
คว่าว	uw9*w	ศารท	,s*r)	ลปก	l&g

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
บินยา	vbny*	อปยศ	o&y,s	ลปน	l&n
บิลญา	vbl,y*	ตะบม	|avm	ตาด	|*d
บิลบี	vblv2	คาเล	u*fl	กฉา	g/*
บิศาจ	vb,s*j	ค่าว	u9*w	ฉัฐ	/>,t
กระสน	grasn	ทรุม	)rcm	โคล	iul
กระสบ	grasv	ทรู่	)r39	คุบ	ucv
หล็อก	hl'og	ทรูท	)r3)	สถน	stn
ศิกษก	,sbg-sg	ก๊าน	g7*n	สถล	stl
หล็อน	hl'on	ทลบม	)lvm	คุป	uc&
ร่อยๆ	r9oy1	ทลิท	)lb)	ล้ว	l4w
ศิเขา	,sbk6	กานน	g*nn	ตาน	|*n
ทีสุด	)2scd	กลู่	gl39	กอย	goy
ฟืนฟู	$5n$3	ท่วย	)9wy	ธปท	0)&)
กระสะ	grasa	ท้วย	)4wy	ธพน	0)?n
ฟุดูล	$cd3l	ทวยะ	)wya	เพะ	?fa
ระงิ่	ra}29	แมลบ	<mlv	ธรณ	0)r,n
กอผจญ	gopj,y	ทวัย	)w>y	สาต	s*|
ทุกขำ	)cgk	โครส	iurs	วาก	w*g
ขึ้นง	k{4n}	ทวิช	)wb+	ตาบ	|*v
หลาหล	hl*hl	โครำ	iurz	จุน	jcn
เก๊น	fg7n	ตะละ	|ala	ลวย	lwy
หลิ่ง	hlb9}	ขาญุ	k*,nc	จะะ	jaa
บือซา	v5o!*	เกศว	fg,sw	ซีท	!2)
ศิลปี	,sbl&2	นกหก	nghg	กาฐ	g*,t
บือโด	v5oid	ดอฆอ	do,uo	โคส	ius
บือแน	v5o<n	ด้อง	d4o}	กอส	gos
ฝีจัก	x2j>g	สิ่ง	s29}	เกษ	fg-s
ศุทธิ	,sc)0)b	โคลด	iuld	กฎิ	g,|b
ศุนัก	,scn>g	สีตล	s2|l	ไคร	:ur
ศุษิร	,sc-sbr	ก๋วย	g8wy	ถนป	tn&

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ระหวย	rahwy	อย่ง	oy9}	พ.ศ	?.,s
เกยลา	fgyl*	อยัส	oy>s	คว-	uw-
ระแอน	ra<on	กาบู	g*v3	ถนำ	tnz
รักขา	r>gk*	ศาสก	,s*sg	ลออ	loo
เกยูร	fgy3r	ดอปอ	do&o	โมฆ	im,u
รักดป	r>gd&	สีมา	s2m*	ถมอ	tmo
กะฉอด	ga/od	ฟ่าม	$9*m	ถลน	tln
เกร็บ	fgr'v	อยุ่	oyc9	ถลม	tlm
เกร็ม	fgr'm	ทหระ	)hra	แน้	<n4
ตัวโค	|eiu	กว้า	gw4*	พกา	?g*
กะซี่	ga!29	ศาสน	,s*sn	พงศ	?},s
รังเก	r>}fg	สีสะ	s2sa	res	res
โซลชา	i!l+*	สีสา	s2s*	คูถ	u3t
กะดัด	gad>d	โมจน	imjn	บิฐ	vb,t
ทูเรน	)3frn	กามท	g*m)	ฆาฏ	,u*,|
กะตอก	ga|og	ปากี	&*g2	วาร	w*r
บุตร์	vc|r0	สิ่ง	s59}	ฆาต	,u*|
กะต่า	ga|9*	ป่าง	&9*}	ฆาน	,u*n
เศกฐี	f,sg,t2	โมริ	imrb	ตาล	|*l
เศรษฐ	f,sr-s,t	โมโร	imir	ซุช	!c+
ญญ่าย	,y,y9*y	ข่าม	k9*m	ไกร	:gr
ตางค์	|*}u0	โมษก	im-sg	แพส	<?s
บุทคล	vc)ul	โมษณ	im-s,n	ตาว	|*w
โคโรค	iuiru	ชี้ต	+24|	ใม้	:1m4
ขุนลู	kcnl3	เกัน	fg>n	ก็ต	g'|
กะตุด	ga|cd	กวาน	gw*n	พชร	?+r
เศวาล	f,sw*l	ปาทป	&*)&	ทาษ	)*-s
แจ๊คแ	<j7uf	จูรณ	j3r,n	วาฬ	w*,l
กะตูก	ga|3g	อวจร	owjr	โนน	inn
บุรณะ	vcr,na	งงิด	}}bd	ควา	uw*

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
โศรตร	i,sr|r	ศีตล	,s2|l	โนม	inm
เกลิง	fglb}	ด้วย	dey	บีต	v2|
หัวตำ	he|z	ด้าง	d4*}	วิง	wb}
กะถัว	gate	สุงก	sc}g	ตาฬ	|*,l
ฐาสุก	,t*scg	สุจะ	scja	สธน	s0)n
กะทัง	ga)>}	กาโย	g*iy	กถา	gt*
สกขรา	sgkr*	สุจิ	scjb	พฐู	?0)3
กะทุน	ga)cn	การก	g*rg	ธุ์	0)c0
ณอคุณ	,nouc,n	สุญญ	sc,y,y	ลัน	l>n
ภารยา	,?*ry*	ศุนิ	,scnb	งอด	}od
เทรีย	)r(	นพศก	n?,sg	ขาบ	k*v
รันธะ	r>n0)a	นภจร	n,?jr	กทม	g)m
ฆอชนี	,uo+n2	ศูนย	,s3ny	เคก	fug
รากษส	r*g-ss	อเสข	ofsk	ขขว	kkw
สงเษป	s}f-s&	ลกลา	lgl*	ธุว	0)cw
กะบุด	gavcd	ลการ	lg*r	อนธ	on0)
ตามิน	|*mbn	ลกุจ	lgcj	ลัภ	l>,?
โจษจน	ij-sjn	ลคุฑ	luc,)	งอย	}oy
โฆเซ่	i,uf!9	ลคุพ	luc,l	อนล	onl
คอสอง	uoso}	ตัตว	|>|w	ลัล	l>l
หายยา	h*yy*	ออกเ	oogf	แฉ่	</9
เบรีย	vr(	คิ่น	ub9n	พบู	?v3
หึงสา	h{}s*	นมผา	nmp*	พพธ	??0)
ฉทวาร	/)w*r	ง้อม	}4om	แฉว	</w
ฉนั้น	/n>4n	คิมห	ubmh	อนำ	onz
มฆวัน	m,uw>n	ป่าย	&9*y	ลาช	l*+
ราปีก	r*&2g	ดาโบ	d*iv	ลาญ	l*,y
พญายา	?,y*y*	นมาซ	nm*!	บงก	v}g
รายวา	r*yw*	นยนะ	nyna	พลว	?lw
มณฑิร	m,n,)br	นยนา	nyn*	มับ	m>v

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
พนจรก	?njrg	ดามพ	d*m?	มัม	m>m
ฉะฉิ่	/a/29	บาบอ	v*vo	มัย	m>y
กรัณฑ	gr>,n,)	กษีร	g-s2r	พลอ	?lo
ฉะต้า	/a|4*	ลพุช	l?c+	พร-	?r-
แบนโจ	<vnij	บาบี	v*v2	ยกน	ygn
ฉะนอง	/ano}	ป้าว	&4*w	กนน	gnn
เหม้อ	fhm4o	ป๊าว	&7*w	ขฉพ	k/?
เหมอะ	fhmoa	นฤดม	nr1dm	พรบ	?rv
เหมือ	hmq	นฤตย	nr1|y	ยขว	ykw
นายร.	n*yr.	ดาลุ	d*lc	ฟอจ	$oj
ริดวน	rbdwn	บาพก	v*?g	ยฉร	y/r
เหริน	fhrbn	บ๋าย	v8*y	ชงฆ	+},u
มนทิร	mn)br	ทอรา	)or*	ยธย	y0)-s
มนุษย	mnc-sy	ไฟ้า	:$4*	ยมก	ymg
สธ.ก็	s0).g'	ด่าว	d9*w	ยมล	yml
สธุสะ	s0)csa	ง่าน	}9*n	ยวน	ywn
ฉัยยา	/>yy*	ง้าน	}4*n	ยวา	yw*
โทบิน	i)vbn	ทะทา	)a)*	ขนด	knd
รื่อง	r59o}	ลมาด	lm*d	ลาล	l*l
คากัน	u*g>n	ลลนา	lln*	ร่า	r9*
มรขวด	mrkwd	ลลาฏ	ll*,|	ร้า	r4*
สนเดก	snfdg	ทะนน	)ann	ลาส	l*s
ฉิมคง	/bmu}	fund	fund	ขนน	knn
กะแหะ	gah<a	ยมท.	ym).	มาฆ	m*,u
สนาดก	sn*dg	ลวณะ	lw,na	สพท	s?)
มรีจิ	mr2jb	ลวนะ	lwna	ขาล	k*l
กะอิด	gaobd	ล่วม	l9wm	โปป	i&&
ใบสัจ	:1vs>j	ล้วย	l4wy	ตร-	|r-
แหร่น	<hr9n	เจมะ	fjma	ชนม	+nm
แหร้ว	<hr4w	ย้อง	y4o}	นาด	n*d

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
รูดี้	r3d24	ทะแย	)a<y	ถิร	tbr
ตำหระ	|zhra	คื้น	u54n	มาด	m*d
ดรธาน	dr0)*n	ก่อง	g9o}	อปร	o&r
มลิ้น	mlb4n	ภฤดก	,?r1dg	นาท	n*)
มลื่น	ml59n	ดาอี	d*o2	ยอบ	yov
มวกผา	mwgp*	ดำกล	dzgl	มาต	m*|
แหว้ง	<hw4}	เศรฐ	f,sr,t	ชู่	+39
แหว้น	<hw4n	ลอนิ	lonb	ดีน	d2n
สนุ่น	snc9n	อะแจ	oa<j	ดีป	d2&
เขยอะ	fkyoa	ดำดง	dzd}	กว-	gw-
มสารก	ms*rg	งายู	}*y3	งาย	}*y
มหัคฆ	mh>u,u	ลอแม	lo<m	พลำ	?lz
ดลภาค	dl,?*u	ล่อย	l9oy	อภว	o,?w
รูสึก	r3s{g	เศลษ	f,sl-s	คอค	uou
กริตย	grb|y	นหาน	nh*n	หยุ	hyc
ค่าตา	u9*|*	ดำนู	dzn3	มาท	m*)
ป๋งมา	&8}m*	นหุต	nhc|	มาน	m*n
เขษตร	fk-s|r	บำรู	vzr3	ชูบ	+3v
มหาพน	mh*?n	เศาจ	,s6j	หยู	hy3
ดไวท์	d:w)0	ตาขอ	|*ko	หรณ	hr,n
โง่งๆ	i}9}1	คณิน	u,nbn	งิน	}bn
ปฏิรพ	&,|br?	ดำบล	dzvl	งิม	}bm
ตีกาก	|2g*g	ละกล	lagl	อุร	ocr
พร่อย	?r9oy	ษัษฐ	-s>-s,t	ชมบ	+mv
ปตานี	&|*n2	โษฑศ	i-s,),s	ปล-	&l-
ปถมัง	&tm>}	ทะไล	)a:l	งึน	}{n
ปปัญจ	&&>,yj	ดำรี	dzr2	ไปย	:&y
เรือก	rqg	ละมา	lam*	ติก	|bg
ปมเปา	&m&6	ดำรู	dzr3	มาบ	m*v
เกียง	g(}	สูรย	s3ry	อมต	om|

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
เกียด	g(d	ละมุ	lamc	เช่	f+9
เกียน	g(n	เสกข	fsgk	งุย	}cy
ชนะมี	+nam2	เส็ง	fs'}	มาย	m*y
ดอนพล	don?l	ทักษ	)>g-s	ยัญ	y>,y
ประกล	&ragl	เง็น	f}'n	กวม	gwm
ตึกเค	|{gfu	ละไม	la:m	ทีก	)2g
กันได	g>n:d	ปีพก	&2,lg	กึง	g{}
มหิมา	mhbm*	ปีพา	&2,l*	คุ่	uc9
มหิยี	mhb-s2	ชุษณ	+c-s,n	กวย	gwy
ประจญ	&raj,y	อังส	o>}s	ผกก	pgg
ดักดน	d>gdn	ยัชน	y>+n	พสก	?sg
มโหษธ	mih-s0)	เสนง	fsn}	พสน	?sn
ดักษก	d>g-sg	ยัฐิ	y>,tb	พสุ	?sc
ดักษณ	d>g-s,n	ยั่น	y>9n	อยน	oyn
กรรดิ	grrdb	ภาณก	,?*,ng	ย้า	y4*
ประชน	&ra+n	เสนห	fsnh	กวะ	gwa
ตุนกู	|cng3	อัฐม	o>,tm	ยาค	y*u
ประชา	&ra+*	อัณฑ	o>,n,)	ผคม	pum
กันยา	g>ny*	ตาธร	|*0)r	ผงม	p}m
กันลง	g>nl}	ต่าน	|9*n	ผงร	p}r
กรรทบ	grr)v	เสบย	fsvy	อยู	oy3
แกงได	<g}:d	กร้อ	gr4o	เจะ	jfa
ตุรงค	|cr}u	เกีย	g(	วีท	w2)
ตุลขำ	|clkz	สงกร	s}gr	ทีน	)2n
จแจ้น	j<j4n	อัทธ	o>)0)	วีร	w2r
ประณต	&ra,n|	ลัคน	l>un	กวา	gw*
ประณม	&ra,nm	สงคม	s}um	ยาด	y*d
ชรแรง	+r<r}	สงคร	s}ur	ทีบ	)2v
ประดง	&rad}	ลั่ง	l>9}	ผญา	p,y*
ประดน	&radn	สงคอ	s}uo	ดุด	dcd

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
อกเลา	ogl6	อันต	o>n|	ผทม	p)m
ชระงม	+ra}m	ยาตร	y*|r	ยาน	y*n
ชระงำ	+ra}z	ลัญฉ	l>,y/	เอช	fo+
เตงกู	f|}g3	ลัฐิ	l>,tb	แซต	<!|
ประดิ	&radb	คนทา	un)*	กุง	gc}
ชระมด	+ramd	ลัดา	l>d*	ลี่	l29
ชระลอ	+ralo	ภาสา	,?*s*	อรช	or+
ธันวา	0)>nw*	เสาว	s6w	ลี้	l24
กัปนก	g>&ng	อันธ	o>n0)	อรณ	or,n
ชระลุ	+ralc	คนธ์	un0)0	ลีช	l2+
มะดูก	mad3g	สณฑ์	s,n,)0	แซท	<!)
มะแตก	ma<|g	ภิงส	,?b}s	แซน	<!n
สมปัก	sm&>g	ทันต	)>n|	เอด	fod
สมผุส	smpcs	เสิบ	s%v	พอน	?on
ประติ	&ra|b	สดูป	sd3&	ขบถ	kvt
มะแน่	ma<n9	สตอล	s|ol	ยาบ	y*v
ธารณะ	0)*r,na	ลาขา	l*k*	กปณ	g&,n
ธารณา	0)*r,n*	ขิ่ง	kb9}	วู่	w39
มะปิน	ma&bn	ตามพ	|*m?	วู๊	w37
มะผาบ	map*v	สติก	s|bg	มาศ	m*,s
มะฝ่อ	max9o	อัมพ	o>m?	ดุล	dcl
ธารธร	0)*r0)r	ลาชา	l*+*	มาส	m*s
สมมติ	smm|b	ลาชิ	l*+b	กฤช	gr1+
กัมดี	g>md2	เปตา	f&|*	กฤด	gr1d
มะพูด	ma?3d	ทัปน	)>&n	เว้	fw8
มะแพน	ma<?n	นะแส	na<s	เวก	fwg
มะแฟน	ma<$n	ล่าน	l9*n	เซส	f+s
มะเมอ	mam%	ทัศน	)>,sn	กพง	g?}
ส้มลม	s4mlm	นักร	n>gr	แฟต	<$|
กัมปน	g>m&n	ดีผา	d2p*	กมง	gm}

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ธิคุณ	0)buc,n	ดีพร	d2?r	ผรณ	pr,n
ประทะ	&ra)a	ตายอ	|*yo	เชา	+6
องคุล	o}ucl	ล้าย	l4*y	กมจ	gmj
เตรีย	|r(	ดียว	d2yw	ผล-	pl-
ประทำ	&ra)z	ดียา	d2y*	ผลก	plg
ธุดงค	0)cd}u	ทัพห	)>,lh	ผลง	pl}
มะวาร	maw*r	ขีณะ	k2,na	อโล	oil
จยุติ	jyc|b	คมน์	umn0	อวก	owg
ธุระะ	0)craa	กรัก	gr>g	กุณ	gc,n
ประเท	&raf)	ตาลอ	|*lo	ลุต	lc|
คำชุม	uz+cm	ท้าง	)4*}	อวท	ow)
มักตา	m>g|*	ย้าว	y4*w	ลุท	lc)
ดาวเน	d*wfn	โง้น	i}4n	แถล	<tl
ดาวบส	d*wvs	ลายู	l*y3	เวย	fwy
มังกง	m>}g}	ตาลุ	|*lc	ผลอ	plo
จรมูก	jrm3g	ลารา	l*r*	มิต	mb|
มังกุ	m>}gc	คร่ง	ur9}	กล-	gl-
จรรจา	jrrj*	ลาลส	l*ls	อสภ	os,?
คำตัก	uz|>g	ก๊าส	g7*s	นขา	nk*
อนวัช	onw>+	ตาวา	|*w*	ที่	)59
เขียะ	k(a	ลาวก	l*wg	มิน	mbn
จรวัก	jrw>g	แก้ง	<g4}	เบท	fv)
ดำนาณ	dzn*,n	แกง	<g}	ลู้	l34
คำนวร	uznwr	กราง	gr*}	ท่ง	)9}
จระทก	jra)g	เภรว	f,?rw	ปลู	&l3
จระนำ	jranz	นันท	n>n)	อเส	ofs
อนุญา	onc,y*	แสนย	<sny	ขมม	kmm
เขือน	kqn	แสบก	<svg	อ๊อ	o7o
คำรวจ	dzrwj	แสยก	<syg	กลธ	gl0)
มันตา	m>n|*	แสยง	<sy}	หวว	hww

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
มันนก	m>nng	แสอก	<sog	ทธิ	)0)b
ดำหนิ	dzhnb	แสอง	<so}	ทนต	)n|
แกร้ง	<gr4}	ดุรค	dcru	ใช้	:1!4
คำปุน	uz&cn	โสดม	isdm	ทนม	)nm
ดิแซะ	db!<a	บุปผ	vc&p	รี่	r29
พระใบ	?ra:1v	กราล	gr*l	กยร	gyr
ชอื้อ	+o54o	ยี่น	y29n	แวก	<wg
แกร้ม	<gr4m	อาจจ	o*jj	ตฤณ	|r1,n
ดินหู	dbnh3	บุพพ	vc??	กรก	grg
ดินอะ	dbnoa	กำชำ	gz+z	สิร	sbr
ชะต้า	+a|4*	กำซำ	gz!z	คปค	u&u
มากชู	m*g+3	ครัน	ur>n	ยำๅ	yz*
คำสมอ	uzsmo	มกรา	mgr*	ตฤท	|r1)
มาคธี	m*u0)2	ดุอา	dco*	ภรต	,?r|
มางสะ	m*}sa	บุยอ	vcyo	ตฤป	|r1&
คำแสด	uz<sd	ยี้ม	y{4m	รีฟ	r2$
ดิอาซ	dbo*!	บุรพ	vcr?	มี่	m29
ชะเลย	+afly	ดูกร	d3gr	มี้	m24
มาเซา	m*!6	แกแน	<g<n	หสน	hsn
ชะวาก	+aw*g	เปศล	f&,sl	มีเ	m2f
คำแหง	uz<h}	เปสล	f&sl	แวฟ	<w$
อปการ	o&g*r	ยี้ม	y54m	สี่	s29
สรบับ	srv>v	นาคร	n*ur	รีส	r2s
อปมาน	o&m*n	หงอด	h}od	แง่	<}8
ดีฉัน	d2/>n	อาซิ	o*!b	ยิฐ	yb,t
โต๊ระ	i|7ra	บุษย	vc-sy	สีฆ	s2,u
สรลอด	srlod	อาณา	o*,n*	ยิน	ybn
อภิณห	o,?b,nh	ดูรา	d3r*	ทมก	)mg
สรลอน	srlon	ลิ้ม	lb4m	รึง	r{}
จองคช	jo}u+	นาคู	n*u3	ทมน	)mn

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
เขือม	kqm	สถาพ	st*?	ทมบ	)mv
มานุษ	m*nc-s	สถาล	st*l	สีด	s2d
กัศมล	g>,sml	ดูอา	d3o*	ภฤศ	,?r1,s
มาฟูส	m*$3s	มดาย	md*y	หอง	ho}
อมตบท	om|v)	ดูเอ	d3fo	สีต	s2|
ไต้ไฟ	:|4:$	อาเด	o*fd	ทมอ	)mo
ดีหิน	d2hbn	อาตม	o*|m	ยิบ	ybv
น้ำขุ	n4*yc	ยุ่น	yc9n	ตลบ	|lv
จอดดด	joddd	ยุ้น	yc4n	โว่	iw9
ประศม	&ra,sm	ยุบล	ycvl	ปสพ	&s?
ประศุ	&ra,sc	อานก	o*ng	สมา	sm*
ประสก	&rasg	พรัด	?r>d	ทยา	)y*
สลิ้ง	slb4}	อานะ	o*na	ภัก	,?>g
กร่อม	gr9om	หญิบ	h,ybv	ใว้	:1w4
ดึงสะ	d{}sa	พรับ	?r>v	ไวเ	:wf
อยาลา	oy*l*	อานำ	o*nz	ทร-	)r-
กลไกล	gl:gl	ยุวา	ycw*	ปสุ	&sc
คิลาน	ubl*n	กริว	grbw	คมช	um+
คิ้ว่	ub4w9	ท่าว	)9*w	มีน	m2n
มารยา	m*r-s*	ยูคา	y3u*	ป๊อ	&7o
แขมพง	<km?}	ยู้ด	y34d	กสท	gs)
ม้ารำ	m4*rz	ลุ่ก	lc9g	คมน	umn
กาจชม	g*j+m	ตำแบ	|z<v	กสิ	gsb
ดุริย	dcrby	มนุญ	mnc,y	ยีด	y2d
นกเกร	ngfgr	บูละ	v3la	แข้	<k4
สวภาพ	sw,?*?	อ่าม	o9*m	ฌอง	,+o}
มาวาร	m*w*r	ลุ้ง	lc4}	สีห	s2h
น้ำชน	n4z+n	น้าว	n4*w	ขิต	kb|
สว้าน	sw4*n	อายต	o*y|	ฌอน	,+on
จะขาบ	jak*v	โชตก	i+|g	คร-	ur-

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
อวลอบ	owlov	อายน	o*yn	มึก	m{g
อวลอล	owlol	แกล่	<gl9	เดง	fd}
มาแอะ	m*o<a	ยูษะ	y3-sa	ทรท	)r)
ลวงค์	lw}u0	เยซุ	fy!c	รุย	rcy
อเสกข	ofsgk	เด็จ	fd'j	รุส	rcs
ลวิตร	lwb|r	ครี้	ur24	ญ้อ	,y4o
กลเหย	glfhy	ครีค	ur2u	ทุพ	)c?
แกะกะ	g<aga	แปทู	<&)3	รู่	r39
มิดลอ	mbdlo	มมาก	mm*g	ลูม	l3m
ถนนตก	tnn|g	มยุง	myc}	รูจ	r3j
มิถุน	mbtcn	แปร๊	<&r8	สือ	s5o
นโคทร	niu)r	มยูข	my3k	ครม	urm
นงพงา	n}?}*	ตำรุ	|zrc	ก๊อ	g7o
โกญจน	ig,yjn	สนิธ	snb0)	ขยด	kyd
มิลลิ	mbllb	สนุข	snck	ศกฎ	,sg,|
นนตรา	nn|r*	สนุต	snc|	ศกล	,sgl
โกฐสอ	ig,tso	สนุน	sncn	ขยม	kym
สอปูน	so&3n	โชรม	i+rm	จัม	j>m
สอยไร	soy:r	สบกก	svgg	รูต	r3|
กล้อม	gl4om	กำมา	gzm*	ปอล	&ol
ละคอน	lauon	สบทบ	sv)v	ศจี	,sj2
ละงิด	la}bd	พฤกษ	?r1g-s	ศดก	,sdg
ละโบม	laivm	นาฬิ	n*,l2	คอล	uol
มีคณา	m2u,n*	กำรอ	gzro	ศตก	,s|g
สะตือ	sa|5o	หนาง	hn*}	ขยล	kyl
สะทอน	sa)on	แกแล	<g<l	ญาน	,y*n
มีแซม	m2<!m	แกว่	<gw9	รูธ	r30)
สะนิง	sanb}	มริจ	mrbj	รูน	r3n
ละลอก	lalog	เยิง	y%}	ศนิ	,snb
ละวาง	law*}	เยิด	y%d	เดธ	fd0)

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ละเวง	lafw}	เยิน	y%n	ศพล	,s?l
คุณจี	uc,nj2	เยิบ	y%v	ศมน	,smn
อะจีน	oaj2n	เยิม	y%m	ญิบ	,ybv
ละเหย	lafhy	หนาน	hn*n	เลณ	fl,n
โกรญจ	igr,yj	เยีย	y(	เลด	fld
กลันท	gl>n)	กำรู	gzr3	แอม	<om
มีศุข	m2,sck	ติงส	|b}s	ศรภ	,sr,?
ละแอน	la<on	กรีษ	gr2-s	ญี่	,y29
นิกษา	nbg-s*	มลโค	mliu	กอก	gog
ลักขะ	l>gka	สพาบ	s?*v	เลท	fl)
ลักขี	l>gk2	อาศุ	o*,sc	ทุส	)cs
นิกอฮ	nbgo=	มล่น	ml9n	ฏฐิ	,|,tb
ลักภา	l>g,?*	ติซี	|b!2	จ๊า	j7*
นิกะห	nbgah	กรึง	gr{}	สมี	sm2
ลักษะ	l>g-sa	เยือ	yq	พาก	?*g
นพปฎล	n?&,dl	หนาย	hn*y	หัง	h>}
ลันเต	l>nf|	มลวก	mlwg	ยุด	ycd
สังขต	s>}k|	มลัก	ml>g	ยุต	yc|
มือสี	m5os2	ทำงน	)z}n	รูม	r3m
ลัพธ์	l>?0)0	มล้า	ml4*	พาง	?*}
ลัพธิ	l>?0)b	มลาก	ml*g	เคห	fuh
ลัภย์	l>,?y0	กรุก	grcg	กรบ	grv
ลัยคต	l>yu|	มลาน	ml*n	ตวน	|wn
ลากษา	l*g-s*	มลาว	ml*w	ภาม	,?*m
มุขลด	mckld	มลิน	mlbn	เลพ	fl?
ลางที	l*})2	อาหร	o*hr	ตวว	|ww
คุณดม	uc,ndm	โป๊ป	i&7&	ตุม	|cm
ลางาด	l*}*d	หนี่	hn29	โอ๊	io7
เดี้ก	d%4g	จมรี	jmr2	พาต	?*|
เดี่ด	d%9d	อำนด	oznd	พาท	?*)

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
เดิ้ด	d%4d	มสาร	ms*r	ยุพ	yc?
อัดธา	o>d0)*	เจิด	j%d	เทก	f)g
เดิ่น	d%9n	มหัล	mh>l	คอส	uos
เดิ้น	d%4n	เจิน	j%n	เรข	frk
มุทิน	mc)bn	เด่า	d69	พาธ	?*0)
อัทธา	o>)0)*	จรบน	jrvn	แบ้	<v4
นิทาฆ	nb)*,u	ซงคำ	!}dz	พาน	?*n
ลาพุช	l*?c+	โปสก	i&sg	สุณ	sc,n
ลาเพา	l*?6	ติตถ	|b|t	เลส	fls
นิธาน	nb0)*n	ติปา	|b&*	เลห	flh
อันตร	o>n|r	ทำไม	)z:1m	สยง	sy}
นินนะ	nbnna	สภาค	s,?*u	สยด	syd
มุรธา	mcr0)*	จรรม	jrrm	เทค	f)u
มุลุต	mclc|	ติมิ	|bmb	มุน	mcn
พระหา	?rah*	หม่อ	hm9o	กลศ	gl,s
อันเต	o>nf|	ใยไพ	:1y:?	สยา	sy*
คุณถม	uc,ntm	ซ้วย	!4wy	พาม	?*m
อันทุ	o>n)c	ติลก	|blg	ยู่	y39
มุหุต	mchc|	ครูด	ur3d	ยู้	y34
อันวย	o>nwy	เลกี	flg2	สรก	srg
มูรธา	m3r0)*	ไปล่	:&l9	สรณ	sr,n
มูรัค	m3r>u	จรวจ	jrwj	สรบ	srv
เดิ่ม	d%9m	จรอก	jrog	เคะ	ufa
เดิ้ม	d%4m	ซอแซ	!o<!	สรพ	sr?
นวปฎล	nw&,dl	สภาว	s,?*w	มุส	mcs
นวพฐู	nw?0)3	เล็น	fl'n	เคา	u6
ลาหู่	l*h39	หมอย	hmoy	สรม	srm
เดี้ย	d(4	ผณิน	p,nbn	โอร	ior
เดียง	d(}	จราส	jr*s	คระ	ura
ล่ำซำ	l9z!z	เล่ย	fl9y	ศลภ	,sl,?

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
เดียด	d(d	หม่า	hm9*	ศวา	,sw*
ลำเทง	lzf)}	หม้า	hm4*	เทท	f))
สันถว	s>ntw	ทิพย	)b?y	ศศะ	,s,sa
ช่ำใจ	+9z:1j	แจตร	<j|r	เทน	f)n
คุณฐู	uc,n0)3	เลื่ว	fl'w	ศศิ	,s,sb
ชำนรร	+znrr	เลวง	flw}	ตู๊	|37
นวาระ	nw*ra	เขดา	fkd*	ขร-	kr-
เดื่อ	dq9	เลส์	fls0	ศศี	,s,s2
นหาดก	nh*dg	คลวง	ulw}	ศสา	,ss*
นหารุ	nh*rc	คล้อ	ul4o	ณัฐู	,n>,t
ลำพัน	lz?>n	อิกอ	obgo	พาส	?*s
นพการ	n,lg*r	ทิพา	)b?*	ทฤฆ	)r1,u
คุณนน	uc,nnn	เลหะ	flha	พาห	?*h
สัปปี	s>&&b	จริม	jrbm	ณิก	,nbg
สัมนา	s>mn*	แจรก	<jrg	ผีต	p2|
แดกงา	<dg}*	แจรง	<jr}	สุภ	sc,?
ลิกขา	lbgk*	อิชท	ob+)	นฤป	nr1&
การจร	g*rjr	ตีระ	|2ra	ชอย	+oy
อัสดม	o>sdm	แบ้ง	<v4}	เต่	f|9
ปสันน	&s>nn	นิกร	nbgr	นวก	nwg
ปสัยห	&s>yh	hนี้	hn24	โชด	i+d
กล้าย	gl4*y	จริว	jrbw	เตง	f|}
ปหังส	&h>}s	ทิวส	)bws	ขล-	kl-
คุณนุ	uc,nnc	I"II	I"II	ขลบ	klv
มูลฟู	m3l$3	นิคห	nbuh	จาบ	j*v
สัยน์	s>yn0	ผรสุ	prsc	ศาณ	,s*,n
การเต	g*rf|	หมืน	hm5n	ศาต	,s*|
คุณบท	uc,nv)	ก้อม	g4om	ขลม	klm
สากิย	s*gby	โจรา	ijr*	ปัฏ	&>,|
นิษฐา	nb-s,t*	โจละ	ijla	สุย	scy

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
นิยาท	nb-s*)	ที้ง	)24}	สุร	scr
ปอทม.	&o)m.	รชนิ	r+nb	ศาป	,s*&
นิสาท	nbs*)	รชนี	r+n2	กาศ	g*,s
ปอพาน	&o?*n	โกญจ	ig,yj	ผุ้	pc4
ปอมท.	&om).	ก็อย	g'oy	ผุก	pcg
ถะกัด	tag>d	ผลิก	plbg	โบน	ivn
ถะโกน	taign	มหิป	mhb&	ไอน	:on
ปะกัง	&ag>}	ผลิน	plbn	กุส	gcs
ถะถับ	tat>v	มหิษ	mhb-s	ขลา	kl*
คุณปก	uc,n&g	รถยา	rty*	แลง	<l}
อาจมน	o*jmn	ผลือ	pl5o	คัง	u>}
ลีพหา	l2,lh*	รบาญ	rv*,y	โชพ	i+?
ถัมภ์	t>m,?0	กอรี	gor2	แลบ	<lv
จัตุร	j>|cr	โดมร	idmr	ขีร	k2r
นอนนก	nonng	และ	flfa	แลป	<l&
ปะบุก	&avcg	แล่ง	<l9}	ตอย	|oy
ปะป่น	&a&9n	จ่อง	j9o}	จ่ง	j9}
กลิ่ง	glb9}	ม่อน	m9on	จ๋ง	j8}
ปะโยค	&aiyu	อิลู	obl3	หาด	h*d
เมนกา	fmng*	สมมต	smm|	บเอ	vfo
นื่อง	n59o}	ซั้น	!>4n	แค้	<u4
โดอาส	ido*s	กรุณ	grc,n	เทล	f)l
สามสบ	s*msv	ม็อม	m'om	ขว-	kw-
นูฮอส	n3=os	โดรณ	idr,n	ขวง	kw}
กลิ้ม	glb4m	หยอน	hyon	หาน	h*n
ลุ่นๆ	lc9n1	สมรด	smrd	เทว	f)w
ลุฟตี	lc$|2	หยัก	hy>g	สุว	scw
ชีต้น	+2|4n	หยัง	hy>}	ไช่	:+9
เมาโร	m6ir	และะ	l<aa	กูฏ	g3,|
นะญิส	na,ybs	หยัด	hy>d	นพป	n,l&

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ลุลาย	lcl*y	แล้ว	<le	น้อ	n4o
กลิ้ง	gl{4}	หยัน	hy>n	ไชย	:+y
นักตะ	n>g|a	หยับ	hy>v	กูด	g3d
เมียร	m(r	ผอวด	powd	คร่ำ	urz
เมื่อ	fm29o	ผอูน	po3n	ต๊ะ	|a7
เนเมธ	fnfm0)	หยาง	hy*}	น่ำ	n9z
ปัญจก	&>,yjg	พลวก	?lwg	ขี้	k{4
เนษาท	fn-s*)	หยาน	hy*n	เมก	fmg
ถ้ำชา	t4z+*	ทีรฆ	)2r,u	ฮอก	=og
อามลก	o*mlg	หยาว	hy*w	ทศม	),sm
สารทา	s*r)*	ตุมพ	|cm?	ฮอง	=o}
คุณพี	uc,n?2	หยุง	hyc}	ก่ำ	g9z
อายาน	o*y*n	หยุน	hycn	เภา	,?6
อายุธ	o*yc0)	หยุย	hycy	ฮอบ	=ov
อายุร	o*ycr	หรบๆ	hrv1	ทศา	),s*
อายุษ	o*yc-s	จ่อม	j9om	ป๊ว	&e
ปัตนี	&>|n2	หรับ	hr>v	หาส	h*s
นังธร	n>}0)r	นิยต	nby|	ทสา	)s*
สารูป	s*r3&	เขตร	fk|r	เผน	fpn
ชุนดู	+cnd3	โกรง	igr}	ม.ค	m.u
นันตา	n>n|*	หรุน	hrcn	ทอก	)og
อาวาห	o*w*h	ซาคา	!*u*	บัฏ	v>,|
อาวุต	o*wc|	โกรด	igrd	ฮัท	=>)
นันทิ	n>n)b	ซ่าง	!9*}	ฮัฟ	=>$
นันลา	n>nl*	หฤทย	hr1)y	โลล	ill
เลขผา	flkp*	กรูม	gr3m	โลห	ilh
เลขยะ	flkya	คลอบ	ulov	ฮาม	=*m
อายาฒ	o*-s*-)	กะชะ	ga+a	ซ่ก	!9g
ขมหิน	kmhbn	นิรา	nbr*	ซ้น	!4n
คุณยุ	uc,nyc	ไลโป	:li&	ซ่ม	!9m

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
คุณรจ	uc,nrj	โกรม	igrm	ซม.	!m.
กาวาง	g*w*}	จ่อย	j9oy	ฮาว	=*w
เลฑฑุ	fl,),)c	รศนา	r,sn*	und	und
เลณฑุ	fl,n,)c	กะดง	gad}	โยด	iyd
เลดงา	fld}*	อุคห	ocuh	โยต	iy|
อาสูร	o*s3r	ทุกข	)cgk	โยธ	iy0)
เลดัม	fld>m	อุณห	oc,nh	พิร	?br
กระดุ	gradc	จอแส	jo<s	ปฏล	&,dl
ฤลละ	tclla	โกลง	igl}	วขฉ	wk/
เลนจง	flnj}	ไจ้ๆ	:j41	จ้น	j4n
กาสาว	g*s*w	ทุกฏ	)cg,|	แดส	<ds
เลปกร	fl&gr	โกลฟ	igl$	โยส	iys
นาคจู	n*uj3	ทุกร	)cgr	โด้	id4
เลยยย	flyyy	สมีป	sm2&	กลา	gl*
เลษฏุ	fl-s,|c	สมีร	sm2r	หูก	h3g
ขยั้น	ky>4n	ผ่าย	p9*y	มกร	mgr
ชุมชม	+cm+m	ผ้าย	p4*y	ไย่	:y9
เลิ่ก	l%9g	คล้า	ul4*	มกส	mgs
เลิ่ด	l%9d	คลาก	ul*g	วงศ	w},s
เลิ่ศ	l%9,s	คลาด	ul*u	ฮูธ	=30)
คุณศก	uc,n,sg	ผ่าว	p9*w	เฮก	f=g
แถมพก	<tm?g	มะฮ์	ma=0	ซอน	!on
ป้านง	&4*n}	ผิดก	pbdg	เทห	f)h
เมือบ	mqv	ฉมัน	/m>n	เฮด	f=d
เลียม	l(m	ร่อย	r9oy	ขอก	kog
ปานนุ	&*nnc	หลาบ	hl*v	จ้บ	j4v
บงบอก	v}vog	วยาม	wy*m	เมศ	fm,s
เลียว	l(w	มังส	m>}s	เมษ	fm-s
เลือก	fl2og	สร่ง	sr9}	คัว	ue
ขวยใจ	kwy:1j	นิ้ด	n24d	นอม	nom

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
กำนัด	gzn>d	สรบบ	srvv	เมห	fmh
ทกล้า	)gl4*	นาโง	n*i}	เมอ	m%
ทดถอย	)dtoy	สรภู	sr,?3	ศิว	,sbw
ชูเซ่	+3f!9	นิ้น	n24n	ค้ำ	u7*
ชูร่า	+3r9*	ซำฮะ	!z=a	กูน	g3n
ชู้ว์	+34w0	ปถพี	&t?2	กูบ	g3v
ทยาลุ	)y*lc	ฉลาย	/l*y	ดนย	dny
กระใด	gra:1d	สรลน	srln	แกล	<gl
ขวั้น	kw>4n	สรลม	srlm	วจน	wjn
บรทาร	vr)*r	ฉลีก	/l2g	วจา	wj*
จามปา	j*m&*	นาฏก	n*,|g	โดล	idl
ขว้าว	kw4*w	วรูถ	wr3t	ปาฐู	&*,t
ทรทึง	)r){}	พอตา	?o|*	วชะ	w+a
ทรธึก	)r0){g	วฤษภ	wr1-s,?	ปาณ	&*,n
จ่าร.	j9*r.	วฤษล	wr1-sl	ใด้	:1d4
โลงอก	il}og	นาฏย	n*,|y	วฏะ	w,|a
ปิงคล	&b}ul	วสลี	wsl2	ได่	:d9
ปิจดี	&bjd2	หวัว	hwe	วตธ	w|0)
ทรสาย	)rs*y	มัธย	m>0)y	จรก	jrg
ทรสุม	)rscm	พ้อม	?4om	ศุน	,scn
โลลุป	illc&	สริต	srb|	วธก	w0)g
ทรหวล	)rhwl	สรีร	sr2r	มตก	m|g
ทรหึง	)rh{}	นิ้ว	n54w	ศุภ	,sc,?
ทรเหล	)rfhl	จะไร	ja:r	มตะ	m|a
โลหัช	ilh>+	หวุม	hwcm	ปณต	&,n|
นางทิ	n*})b	ทุรน	)crn	บ้ำ	v7*
ไลโซล	:li!l	อูว์	o3w0	วนา	wn*
ทรอึง	)ro{}	สลวน	slwn	พีท	?2)
ขอแกว	ko<gw	คลิง	ulb}	มธุ	m0)c
ทรัพย	)r>?y	คลิด	ulbd	วยง	wy}

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ไลปิด	:l&bd	ปรวด	&rwd	แกว	<gw
สำบอก	szvog	เอกม	fogm	ปืน	&'n
จ่าอ.	j9*o.	เอกส	fogs	วรท	wr)
พ่อธร	?9o0)r	มัลก	m>lg	เก่	fg9
เชิ่ง	+%9}	ทุษฐ	)c-s,t	มนท	mn)
เชิ้ง	+%4}	กล่น	gl9n	เผ	fp
จำงาย	jz}*y	เน่ง	fn9}	อข	ok
จำทับ	jz)>v	ระมา	ram*	นย	ny
แม-ม้	<m-m4	กลบท	glv)	ภว	,?w
แมร์ก	<mr0g	ก้ง	g4}}	อฆ	o,u
แม่รจ	<m9rj	กะทำ	ga)z	ลอ	lo
กำหนก	gzhng	ฉันท	/>n)	นร	nr
เชิ่ด	+%9d	ม้าน	m>*n	วข	wk
บราลี	vr*l2	ทูษก	)3-sg	วู	w3
อีควิ	o2uwb	คุ่น	uc9n	al	al
ทฤษฎี	)r1-s,d>	เขลง	fkl}	ชค	+u
วฏุมะ	w,|cma	กล่ม	gl9m	หห	hh
วฑฺฒน	w,)-)n	เขละ	fkla	แซ	<!
ทลอึง	)lo{}	พะอง	?ao}	ศว	,sw
จำบ่ม	jzv9m	ระลง	ral}	ปบ	&v
เชิ่ล	+%9l	เขษม	fk-sm	บย	vy
เชี้ย	+(4	มาแก	m*<g	สผ	sp
วนจรก	wnjrg	พักร	?>gr	ศศ	,s,s
จำบับ	jzv>v	น่าย	n9*y	อย	oy
ปืนยา	&5ny*	กลมภ	glm,?	ชช	++
คดรวย	udrwy	ฉาทน	/*)n	ตะ	|a
บริรม	vrbrm	มาจา	m*j*	ผอ	po
จำยาม	jzy*m	คลุบ	ulcv	ศห	,sh
จำราญ	jzr*,y	ไกรพ	:gr?	f.	f.
พัญจก	?>,yjg	กชกร	g+gr	มค	mu

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
อุจฉุ	ocj/c	เท้ง	f)4}	หย	hy
เชือม	+qm	สวยม	swym	มฆ	m,u
วังชา	w>}+*	มาดา	m*d*	ปร	&r
กระทด	gra)d	กลละ	glla	ศฮ	,s=
สือดี	s5od2	เท้ด	f)4d	มง	m}
วัญจก	w>,yjg	กล้ว	gl4w	กข	gk
คุดซู	ucd!3	เอฟ)	fo$)	เช	f+
คุปติ	uc&|b	มาตร	m*|r	กค	gu
โชโหว	i+ihw	คุยห	ucyh	เล	fl
ขะแถก	ka<tg	เนาะ	noa	ชท	+)
ต๊กโต	|7gi|	มาตฤ	m*|r1	คี	u2
เปรอง	&r%}	มาตา	m*|*	นฤ	nr1
ครวัก	urw>g	สว่า	sw9*	พด	?d
กว่าง	gw9*}	เผือ	pq	บษ	v-s
ตกว่า	|gw9*	ฉินท	/bn)	มจ	mj
ตกหมก	|ghmg	หั้น	h>4n	โม	im
พ่าห์	?9*h0	ควอช	uwo+	กม	gm
ตงุ่น	|}c9n	ม้าน	m4*n	di	di
คูเบค	u3fvu	กะปู	ga&3	ริ	rb
ตจสาร	|js*r	ฉียง	/2y}	คย	uy
พำพวก	?z?wg	มานี	m*n2	ผะ	pa
พิกโต	?bgi|	ควาก	uw*g	คร	ur
พิก๊ล	?bg7l	มาปก	m*&g	พำ	?z
นางสุ	n*}sc	มาภา	m*,?*	สห	sh
ยมท.)	ym).)	ม้าย	m4*y	แผ	<p
ยรรยง	yrry}	กล้อ	gl4o	ลิ	lb
ยวกสา	ywgs*	ขจอก	kjog	ทส	)s
ซลาโว	!l*iw	รัชด	r>+d	วจ	wj
ต้นกล	|4ngl	แต้ว	<|4w	ไก	:g
ย้องๆ	y4o}1	เฉลว	f/lw	กจ	gj

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
บฤงคพ	vr1}u?	ขจาว	kj*w	ถด	td
เปศัส	f&,s>s	ขจุย	kjcy	ถน	tn
พิดรก	?bdrg	ซีไอ	!2:o	ลฯ	l;2
พิดาน	?bd*n	ขเจา	kj6	วณ	w,n
พิโดร	?bidr	ขณะเ	k,naf	คณ	u,n
เปสุญ	f&sc,y	โต่ง	i|9}	ชะ	+a
คระหน	urahn	มารค	m*ru	วต	w|
กินตา	gbn|*	ต.ปง	|.&}	ระ	ra
ทะนุก	)ancg	แน่ว	<n9w	อช	o+
วัทน์	w>)n0	โตฎก	i|,dg	วธ	w0)
วัที่	w>)29	ขตอย	k|oy	เม	fm
วันทน	w>n)n	เทวอ	f)wo	ทก	)g
วันทย	w>n)y	โผอน	ipon	มต	m|
วันทิ	w>n)b	มาระ	m*ra	แร	<r
สุหฤท	schr1)	มารู	m*r3	มท	m)
สุหลง	schl}	ก็ด้	g'd4	ทด	)d
สุหัท	sch>)	ส่อน	s9on	ทท	))
สุไหง	sc:h}	กะแม	ga<m	เค	fu
สูกษม	s3g-sm	กะยุ	gayc	ยู	y3
วันฟู	w>n$3	กดชา	gd+*	ทน	)n
ทะลอก	)alog	มาลำ	m*lz	ตจ	|j
ยันตร	y>n|r	มาเล	m*fl	พุ	?c
จีนแส	j2n<s	โนว์	inw0	ไม	:m
กว่าน	gw9*n	พานร	?*nr	ฮก	=g
วัปปะ	w>&&a	มาสก	m*sg	ตึ	|{
ต้นัง	|4n>}	เค้ง	fu4}	ฮง	=}
บอนนา	vonn*	เคจฉ	fuj/	ฮด	=d
เปิดส	&%ds	มาห์	m*h0	เห	fh
วัวร์	wer0	ไตบา	:|v*	ทุ	)c
ยัษฎิ	y>-s,|b	ขนอง	kno}	ซง	!}

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
เสฏฐี	fs,|,t2	แฉละ	</la	พะ	?a
เอ๊าฟ	o67$	มาฬก	m*,lg	นฬ	n,l
กว้าว	gw4*w	คหัฐ	uh>,t	วภ	w,?
วากรา	w*gr*	ขนาก	kn*g	ตต	||
ยากส์	y*gs0	ขนาง	kn*}	ซู	!3
วางตา	w*}|*	มิคา	mbu*	โศ	i,s
ยาจนก	y*jng	มิคี	mbu2	ตท	|)
ยาจนา	y*jn*	มิ้ง	mb4}	แต	<|
ยาไฉน	y*:/n	เค้ป	fu4&	โร	ir
ยาตนา	y*|n*	หามะ	h*ma	ตล	|l
วาจาล	w*j*l	ชคดี	+ud2	รณ	r,n
บอเลา	vol6	เทห์	f)h0	บะ	va
บอว์ม	vow0m	มิญช	mb,y+	ฟุ	$c
กินนี	gbnn2	เค้ย	fu4y	ซล	!l
ยานยา	y*ny*	มิดะ	mbda	ทพ	)?
บะดัร	vad>r	กลัง	gl>}	รุ	rc
ตยุติ	|yc|b	ซูรอ	!3ro	เด	fd
เสริช	sr%+	กุกๆ	gcg1	ษฐ	-s,t
เสริด	sr%d	ชฎิล	+,dbl	ษน	-sn
ทะเวน	)afwn	หาริ	h*rb	ษม	-sm
กระทอ	gra)o	เคยู	fuy3	อน	on
ยายจี	y*yj2	มิตี	mb|2	โส	is
เสวตร	fsw|r	หาสก	h*sg	จิ	jb
พิหาร	?bh*r	พิขา	?bk*	ทม	)m
พิฬาร	?b,l*r	พิ่ง	?b9}	ธ์	0)0
ตรมวล	|rmwl	พิจย	?bjy	ลี	l2
วายิบ	w*ybv	รัมก	r>mg	du	du
นาที่	n*)29	พิจล	?bjl	นข	nk
ขับซอ	k>v!o	แอลก	<olg	หม	hm
ตรลอด	|rlod	บงสุ	v}sc	รท	r)

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ตรลาด	|rl*d	พิชย	?b+y	นค	nu
ขับพล	k>v?l	หิรอ	hbro	สะ	sa
วาลธิ	w*l0)b	หีตา	h2|*	ใค	:1u
เคร็ด	fur'd	หึงส	h{}s	ทร	)r
เคร็น	fur'n	โอโซ	ioi!	รพ	r?
นาเนก	n*fng	เขาอ	k6o	ขย	ky
ตระกล	|ragl	รางๆ	r*}1	ซิ	!b
ยายยุ	y*yyc	กทลี	g)l2	จญ	j,y
วิกจะ	wbgja	ถงาด	t}*d	วล	wl
เคร่า	ur69	เผือ	xq	ป่า	&z
ยายวี	y*yw2	มี้ด	m24d	ผก	pg
ยายศิ	y*y,sb	บโทน	vi)n	ปิ	&b
วิกัย	wbg>y	พิธุ	?b0)c	ฐู	0)3
วิการ	wbg*r	มีนา	m2n*	วว	ww
พี่ธม	?290)m	ขบวร	kvwr	ธจ	0)j
เสียน	s(n	ขบูร	kv3r	ธช	0)+
ตระดก	|radg	กล่า	gl9*	วษ	w-s
กินลี	gbnl2	กุณิ	gc,nb	ธน	0)n
ตระไน	|ra:n	กุณี	gc,n2	ชุ	+c
ตระบก	|ravg	โอษฐ	io-s,t	ขข	kk
ทังวล	)>}wl	กุดา	gcd*	ดร	dr
ซะอัด	!ao>d	ขมวน	kmwn	ขง	k}
โอทีส	io)2s	กนลา	gnl*	ชย	+y
โอ้โธ	io4i0)	มีพห	m2,lh	ไส	:s
โปรโส	i&ris	วาบา	w*v*	สก	sg
วิจิน	wbjbn	เขือ	kq	ฮิ	=b
ยำยาม	yzy*m	วามน	w*mn	วอ	wo
โอลอฟ	iolo$	วามะ	w*ma	มร	mr
กระทุ	gra)c	มี้ด	m54d	อห	oh
ทัดทา	)>d)*	กุมภ	gcm,?	ฒน	,un

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ยิฏฐะ	yb,|,ta	มื่น	m59n	ผำ	pz
จุมที	jcm)2	มื่อ	m59o	ห้	h4
บัพชา	v>?+*	วาโย	w*iy	ฤณ	r1,n
ไปศาจ	:&,s*j	วารณ	w*r,n	นง	n}
จุมพฏ	jcm?,|	กะแอ	ga<o	หง	h}
ตรังค	|r>}u	พิมพ	?bm?	ฤต	r1|
ยิสต์	ybs|0	กลาด	gl*d	นฐ	n,t
บัลลพ	v>ll?	เชิล	!%l	นท	n)
จุมพล	jcm?l	ขยอง	kyo}	กี	g2
ซัลมา	!>lm*	ไอนุ	:onc	นน	nn
ทัพพะ	)>??a	กังก	g>}g	wo	wo
พี่ออ	?29oo	ค้อย	u4oy	ถี	t2
ซาญอล	!*,yol	กบทู	gv)3	นี	n2
ซ้าตา	!4*|*	สัตย	s>|y	นบ	nv
แสนิง	<snb}	มุขย	mcky	ลี	l'
ผเดิน	pd%n	คอร์	uor0	ลก	lg
กระนก	grang	มุคร	mcur	ตร	|r
วิทูร	wb)3r	ว่าา	w9**	มุ	mc
ยุกติ	ycg|b	ขยาบ	ky*v	สค	su
ซาเนน	!*fnn	สัตว	s>|w	กน	gn
จูดัง	j3d>}	ร๊าด	r7*d	วั	w>
จูดิท	j3db)	กุรร	gcrr	ตห	|h
พุงจง	?c}j}	ราตร	r*|r	สุ	sc
พุงดอ	?c}do	แซ้ง	<!4}	ปง	&}
จูเทศ	j3f),s	ราโท	r*i)	ฎิ	,|b
กิ๋งบ	g28}v	มุจน	mcjn	สล	sl
ซาเยด	!*fyd	มุญช	mc,y+	คห	uh
หงสรถ	h}srt	มุทค	mc)u	ฐม	,tm
ผรุพก	prc?g	ตถตา	|t|*	ศข	,sk
ครูลพ	ur3l?	มุทา	mc)*	ศฉ	,s/

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
หงอดๆ	h}od1	กบี่	gv29	ศช	,s+
ผลคุน	plucn	มุทุ	mc)c	ศฎ	,s,d
วิโยค	wbiyu	ตนกู	|ng3	ศฐ	,s,t
หงุ่ย	h}c9y	มุนิ	mcnb	ศต	,s|
หจช.,	hj+.,	กังส	g>}s	แย	<y
บางสุ	v*}sc	ไทกร	:)gr	แ-	<-
กระบา	grav*	ฮเลบ	=flv	ขู	k3
ตรึงศ	|r{},s	มุ่ม	mc9m	ลู	l3
เจดีย	fjd2y	มุ้ม	mc4m	บค	vu
ผลึ่ง	pl{9}	ชลวร	+lwr	บง	v}
ตฤตีย	|r1|2y	กโบร	givr	โ-	i-
วิวระ	wbwra	คอริ	uorb	ก้	g4
บาเซล	v*f!l	พิหค	?bhu	จู	j3
บาเซะ	v*!fa	กโบล	givl	ศม	,sm
หตุผล	h|cpl	กปณก	g&,ng	ศย	,sy
วิวาห	wbw*h	กปณา	g&,n*	ผู	p3
บาดรี	v*dr2	มุรช	mcr+	ลฐ	l,t
บาแดง	v*<d}	ขล้ง	kl4}	อป	o&
วิษาณ	wb-s*,n	แซ่ว	<!9w	ยง	y}
วิสฤต	wbsr1|	มุลู	mcl3	เต	f|
กุขุน	gckcn	แข่น	<k9n	บจ	vj
บาทรช	v*)r+	ขลวน	klwn	บฉ	v/
เจโทร	fji)r	มุสะ	mcsa	สฐ	s,t
เยภุย	fy,?cy	ราแม	r*<m	สว	sw
ซิงเย	!b}fy	ขลาย	kl*y	ฟง	$}
ซิงห์	!b}h0	แขนะ	<kna	ยม	ym
ผะออบ	paoov	ขลึง	kl{}	ฉง	/}
วีกลี	w2gl2	กปอ.	g&o.	บฎ	v,|
เยาวน	y6wn	มูดอ	m3do	กุ	gc
เคล็น	ful'n	รายา	r*y*	แช	<+

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
เคล้น	ful4n	แขละ	<kla	สต	s|
เคล่า	ul69	มูรธ	m3r0)	โฑ	i,)
ซินตา	!bn|*	โซเย	i!fy	ยล	yl
เคลิก	ul%g	โซรม	i!rm	ชว	+w
วีแวว	w2<ww	ราสี	r*s2	ยว	yw
รือน	r5on	ฮารี	=*r2	บต	v|
กูฎา	g3,|*	ราหุ	r*hc	ลพ	l?
กัน.	g>n.	คะได	ua:d	พข	?k
โหมะ	ihma	ฮาเว	=*fw	ถุ	tc
รุกข	rcgk	คะนน	uann	ปุ	&c
ขวาด	kw*d	ตมูก	|m3g	กส	gs
ฐกัด	,tg>d	โซโห	i!ih	ภ7	,?#g
กู้ด	g34d	ตยาค	|y*u	ยห	yh
ถั่ง	t>9}	กัณห	g>,nh	เน	fn
ขวาว	kw*w	ถนัน	tn>n	ภค	,?u
ขยณะ	k-s,na	ถนิม	tnbm	ฉม	/m
ขษัย	k-s>y	สัสต	s>s|	ทล	)l
ขษีร	k-s2r	ตรณี	|r,n2	นภ	n,?
สามล	s*ml	ถลาก	tl*g	มู	m3
ถัทธ	t>)0)	คะมู	uam3	งะ	}a
กมาร	gm*r	คะยา	uay*	ถบ	tv
รุจา	rcj*	ตรลบ	|rlv	ภฐ	,?,t
ขสมก	ksmg	ธตรฐ	0)|r,t	ภต	,?|
กมุท	gmc)	ตรลา	|rl*	ดล	dl
ชะชะ	+a+a	จากร	j*gr	ภภ	,?,?
ข้อก	k4og	วินย	wbny	ม่	m9
พื่น	?59n	คัคน	u>un	ด้	d4
ตรับ	|r>v	เคอย	u%y	บธ	v0)
ธนุร	0)ncr	ส้าง	s4*}	ภย	,?y
ไหม่	:hm9	ริปุ	rb&c	ภร	,?r

ตารางที่ 9 รายการคำภาษาไทยกับคำอักษรเบรลล์ภาษาไทยที่โปรแกรม SWATH ตัดคำผิดพลาด (ต่อ)

ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
รุ่ม	rc9m	ริปู	rb&3	ถล	tl
พื่อ	?59o	ตรวด	|rwd	วิ	wb
ส้าย	s4*y	ตรวย	|rwy	ร่	r9
รุรุ	rcrc	แหนว	<hnw	พจ	?j
ไห้ว	:h4w	ตรอน	|ron	บบ	vv
รุหะ	rcha	แหระ	hr<a	บพ	v?
ฑังส	,)>}s	ฌ็อง	,+'o}		
คันถ	u>nt	ถ่อน	t9on		

ตารางที่ 10 รายการคำของคลังข้อความ BEST เขียนผิดพลาด

คลังข้อความ BEST เขียนผิด		ที่ถูกต้องควรเป็น	
ภาษาไทย	เบรลล์ไทยระดับ 1	ภาษาไทย	เบรลล์ไทยระดับ 1
ทรงสีเหลี่ยม	)r}s2hl(9m	ทรงสี่เหลี่ยม	)r}s29hl(9m
คลอบ	ulov	ครอบ	urov
กึงผ่าน	g{}p9*n	กึ่งผ่าน	g{9}p9*n
ใว้ใน	:1w4:1n	ไว้ใน	:w4:1n
สัญญาน	s>,y,y*n	สัญญาณ	s>,y,y*,n
ไฟ้า	:$4*	ไฟฟ้า	:$$4*
ตรงที	|r})2	ตรงที่	|r})29
พื่นที	?59n)2	พื้นที่	?54n)29
เลือก	fl2og	เลือก	lqg
ก็พาน	g'?*n	ก็พาล	g'?*l
เจรจาพาที่	fjrj*?*)29	เจรจาพาที	fjrj?*)2
ไม่มี่	:m9m29	ไม่มี	:m9m2
ก.	g.	ก็	g'
นาที่	n*)29	นาที	n*)2
ผุ้ชาย	pc4+*y	ผู้ชาย	p34+*y
แซ่	<!9	แซ่ด	<!9d
ทีมี	)2m2	ที่มี	)29m2
ก้	g4	ก็	g'
เมื่อ	fm29o	เมื่อ	mq9
ใด้	:1d4	ได้	:d4
คื้น	u54n	คืน	u5n
ดีดีที่	d2d2)29	ดีดีที	d2d2)2
เหมือ	fhm5o	เหมือน	hmqn
ม้าน	m>*n	มัน	m>n
ก๊า	g7*	กับ	g>v
สังคราะห์	s>}ur*ah0	สังเคราะห์	s>}uroah0
ต้นัง	|4n>}	ต้น	|4n
สักที่	s>g)29	สักคนที่	s>gun)29
ก๊อ	g7o	ก็	g'

ตารางที่ 10 รายการคำของคลังข้อความ BEST เขียนผิดพลาด (ต่อ)

คลังข้อความ BEST เขียนผิด		ที่ถูกต้องควรเป็น	
ห้องแลบ	h4o}<lv	ห้องแล็ป	h4o}<l’&
ผุ้	pc4	ผู้	p34
ประสมัย	&rasm>y	ประชุมสมัย	&ra+cmsm>y
ไป๊	:&7	ไป	:&
ลูกบาตร	l3gv*|r	ลูกบาศก์	l3gv*,sg0
ชีวิตยู่	+2wb|y39	ชีวิตอยู่	+2wb|oy39
ผ่าวร้อน	p9*wr4on	ร้อนผ่าว	r4onp9*w
ชอกช้ำก่ำทรวง	+og+z4gz9)rw}	ชอกช้ำระกำทรวง	+og+z4ragz)rw}
มาเกิน	m*g%n	มาเกิด	m*g%d
ใว้	:1w4	ไว้	:w4
น้ำมันก๊าส	nz4m>ng7*s	น้ำมันก๊าด	nz4m>ng7*d
ยู่	y39	อยู่	oy39
รักทีมี	r>g)2m2	รักที่มี	r>g)29m2
ขอมัน	kom>n	ของมัน	ko}m>n
รู้สีสา	r34s2s*	รู้สึกรู้สา	r34s{gr34s*
โทร	i)r	โทรศัพท์	i)r,s>?)0
มีดู	m2d3	มาดู	m*d3

ภาคผนวก ง : ตารางอักษรเบรลล์ประเภทต่างๆ

ตารางที่ 11 อักษรเบรลล์ภาษาไทยแบบ 6 จุด

ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ
G	K	0k	U	-u	,u	]	J	/
ช	ซ	ฌ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ
+	!	,+	,y	,d	,|	,t	,)	-)
ณ	ด	ต	ถ	ท	ธ	น	บ	ป
,n	D	|	T	)	0)	N	V	&
ผ	ฝ	พ	ฟ	ภ	ม	ย	ร	ล
P	X	?	$	,?	M	Y	R	L
ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	
W	,s	-s	S	H	,l	O	=	
อะ	อา	อิ	อี	อึ	อื	อุ	อู	
A	*	B	2	[	5	C	3	
เอะ	เอ	แอะ	แอ	โอะ	โ	เอาะ	ออ	
Fa	f	<t	<	ia	i	Oa	O	
เออะ	เออ	เอียะ	เอีย	เอือะ	เอือ	อัวะ	อัว	
%a	%	(a	(	Qa	Q	Ea	E	
อำ	ไอ	ใอ	เอา	ฤ	ฦ	ฤๅ	ฦๅ	
Z	:	:1	6	R1	L1	R1*	L1*	
อ่	อ้	อ๊	อ๋	อั	อ์	ๆ		
9	4	7	8	>	0	1		

ตารางที่ 12 อักษรเบรลล์ภาษาไทยแบบ 8 จุด

ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ	ฐ
G	K	k	U	u	u	]	J	\	0	8	0	Y	D	8	T
ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ	ภ
}	}	N	D	8	T	}	}	N	V	&	P	X	4	6	4
ม	ย	ร	ฤ	ล	ฦ	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ	อะ
M	Y	R	R	L	L	W	4	4	S	H	l	O	%	'	A
อั	อา	อำ	อิ	อี	อึ	อื	อุ	อู	อฺ	เอ	แอ	โ	ใอ	ไอ	า
@	*	Z	B	;	9	?	C	:	.	f	2	i	5	5	*
ๆ	อี๋	อ่	อ้	อ๊	อ๋	อ์	อำ	๐	๑	๒	๓	๔	๕	๖	๗
'	'	*	/	=	(	)	!	j	A	B	C	D	e	f	g
๘	๙														
h	i														

ประวัติผู้เขียน

ชื่อ สกุล นายทศวัฒน์ ชุณหวิทยะธีระ

รหัสประจำตัวนักศึกษา 5010120025

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2550

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนผู้ช่วยสอน ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัย-สงขลานครินทร์

การตีพิมพ์เผยแพร่ผลงาน

ทศวัฒน์ ชุณหวิทยะธีระ และ ผศ.ดร. พิชญา ตัณฑัยย์, “การใช้เอ็นแกรมช่วยในการตัดสินใจแปลอักษรเบรลล์ที่ใช้คำควบกล้ำ สระประสม และอักษรเบรลล์สองเซลล์,” ในการประชุมวิชาการเสนอผลงานวิจัย มหาวิทยาลัยพายัพ พ.ศ. 2555, เชียงใหม่, ประเทศไทย, 17 กุมภาพันธ์ 2555, รางวัลผลงานวิจัยดีเด่น (Best Paper Award) กลุ่มสาขาวิทยาศาสตร์เทคโนโลยี.