การตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรในข้อความสื่อสังคมออนไลน์และแปลงให้เป็นบรรทัดฐาน

นางสาวปวันรัตน์ หิรัญกาญจน์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

DETECTION AND NORMALIZATION OF WORDPLAY GENERATED BY REPRODUCTION OF LETTERS IN ONLINE SOCIAL MEDIA TEXTS

Miss Pawanrat Hirankan

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

หัวข้อวิทยานิพนธ์	การตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรในข้อความสื่อสังคมออนไลน์และแปลงให้เป็นบรรทัดฐาน
โดย	นางสาวปวันรัตน์ หิรัญกาญจน์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.อติวงศ์ สุชาโต
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ผู้ช่วยศาสตราจารย์ ดร.โปรดปราน บุณยพุกกณะ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

……. ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม)

…….. อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.อติวงศ์ สุชาโต)

……. อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(ผู้ช่วยศาสตราจารย์ ดร.โปรดปราน บุณยพุกกณะ)

……. กรรมการภายนอกมหาวิทยาลัย
(ดร.ชัย วุฒิวิวัฒน์ชัย)

ปวันรัตน์ หิรัญกาญจน์ : การตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรในข้อความสื่อสังคมออนไลน์และแปลงให้เป็นบรรทัดฐาน (DETECTION AND NORMALIZATION OF WORDPLAY GENERATED BY REPRODUCTION OF LETTERS IN ONLINE SOCIAL MEDIA TEXTS) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.อติวงศ์ สุชาโต, อ. ที่ปรึกษาวิทยานิพนธ์ร่วม : ผศ.ดร.โปรดปราน บุณยพุกกณะ, 78 หน้า.

การเล่นคำด้วยวิธีซ้ำตัวอักษรจากคำเดิมเป็นวิธีการเล่นคำที่พบมากในเว็บไซต์เครือข่ายทางสังคม ซึ่งการเล่นคำโดยส่วนใหญ่จะสร้างความกำกวมให้กับระบบประมวลผลทางภาษามนุษย์ เช่น ระบบสังเคราะห์เสียง งานวิจัยนี้แสดงสถิติการเกิดของการเล่นคำด้วยวิธีซ้ำตัวอักษรจากข้อความในเว็บไซต์เครือข่ายทางสังคมจำนวน 102,586 ชิ้นข้อความ โดยเสนอลักษณะเด่นที่ใช้ในการจำแนกประเภท และกรอบงานสำหรับการจำแนกประเภทเพื่อตรวจจับโทเค็นที่เป็นการเล่นคำด้วยวิธีซ้ำตัวอักษรจากข้อความภาษาไทยในเว็บไซต์เครือข่ายทางสังคม ซึ่งถูกแบ่งเป็นโทเค็นย่อยในระดับคำด้วยเครื่องมือการตัดคำภาษาไทย ที่เรียนรู้จากแบบจำลองคอนดิชันแนลแรนดอมฟิลด์ จากนั้นนำเสนอระบบในการแปลงข้อความให้เป็นบรรทัดฐาน โดยคำนึงถึงการแปลงเป็นคำอ่านโดยเสนอวิธีการที่ใช้ในการจัดการโทเค็นที่แตกต่างกัน กรอบงานสำหรับการจำแนกประเภทวิธีการจัดการแปลงให้เป็นบรรทัดฐานที่เหมาะสมกับลักษณะการซ้ำตัวอักษรของโทเค็น ซึ่งจากการวัดผลด้วยชิ้นข้อความจำนวน 48,949 ชิ้นข้อความ แล้วพบว่าระบบตรวจจับการเล่นคำมีความแม่นยำถึง 98.45% ซึ่งมีประสิทธิภาพสูงขึ้นจากการใช้กฎและวิธีเส้นแบ่งฐาน และระบบแปลงให้เป็นบรรทัดฐานสามารถแปลงข้อความที่ตรวจจับได้ได้ถูกต้อง 99.19 % เมื่อตรวจสอบโดยผู้เชี่ยวชาญ

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2555 ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม .....

5470271021 : MAJOR COMPUTER ENGINEERING

KEYWORDS : NATURAL LANGUAGE PROCESSING/ ONLINE SOCIAL NETWORK LANGUAGE PROCESSING/ WORDPLAY DETECTION/ WORDPLAY NORMALIZATION/ TEXT NORMALIZATION/ FACEBOOK/ DECISION TREE

PAWANRAT HIRANKAN : DETECTION AND NORMALIZATION OF WORDPLAY GENERATED BY REPRODUCTION OF LETTERS IN ONLINE SOCIAL MEDIA TEXTS ADVISOR: ASSOC.PROF. ATIWONG SUCHATO, Ph.D., CO-ADVISOR: ASST.PROF. PROADPRAN PUNYABUKKANA, Ph.D., 78 pp.

Wordplay generated by letters of its original word being repeated is commonly found in social network texts. Most of the time, wordplay items of this type are ambiguous to machines in language processing tasks such as Text-to-Speech. This research shows some statistics on the number of letters found in 102,586 real social network text items and proposes a set of classification features together with a few classification frameworks to detect repeated-letter wordplay tokens from Thai social network texts, which were tokenized by CRF-based Thai word segmentation. Then proposed an original word pronunciation based normalization system by handling method classification framework. Evaluation on 48,949 text items shows that the proposed method achieves the detection accuracy of 98.45% which is an improvement over simple rule-based and some previously proposed methods. In addition normalized detected wordplay tokens achieve 99.19 % accuracy evaluated by expert checking.

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

Field of Study : Computer Engineering Advisor's Signature .....

Academic Year : 2012 Co-advisor's Signature .....

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก รองศาสตราจารย์ ดร.อติวงศ์ สุชาโต และ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ผู้ช่วยศาสตราจารย์ ดร.โปรดปราน บุณยพุกกณะ ที่ให้คำปรึกษา ความรู้และโอกาสที่ดีต่าง ๆ ในระหว่างระยะเวลาที่ศึกษาและดำเนินการวิจัย ขอขอบคุณ ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม อาจารย์ ดร.ชัย วุฒิวิวัฒน์ชัย ที่ให้คำแนะนำเพื่อใช้ในการปรับปรุงแก้ไขในการทำวิทยานิพนธ์นี้

นอกจากนี้ขอขอบคุณเพื่อนร่วมงานในห้องปฏิบัติการระบบภาษาพูดและห้องปฏิบัติการเทคโนโลยีช่วยเหลือผู้พิการ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้ข้อแนะนำและกำลังใจเสมอมากับข้าพเจ้า ขอขอบคุณ คุณแม่ และครอบครัวที่เข้าใจและสนับสนุนการทำวิจัยในครั้งมาโดยตลอด ทำให้การจัดทำวิทยานิพนธ์ในครั้งนี้ประสบความสำเร็จได้

สารบัญ

หน้า

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เครื่องมือสังเคราะห์เสียงถูกนำมาใช้อย่างกว้างขวางในการสื่อสารในปัจจุบัน ซึ่งข้อความที่ปรากฏในภาษาต่างๆที่ใช้สื่อสารกันโดยทั่วไปจะประกอบด้วยอักขระของภาษานั้นๆ สัญลักษณ์ต่างๆ ตัวเลข รวมถึงอักขระในภาษาอื่นๆ เช่น ภาษาอังกฤษ ที่ถูกนำมาใช้ในการสื่อสารเพื่อการอธิบายที่ชัดเจนยิ่งขึ้น ข้อความของอักขระภาษาไทยที่เขียนไม่ถูกต้องตามหลักภาษา สัญลักษณ์ ตัวเลข และอักขระภาษาอังกฤษต่างๆที่แทรกเข้ามานี้ รวมเรียกว่า คำที่ไม่เป็นมาตรฐาน (Non-Standard Word : NSW) [1] ซึ่งเป็นส่วนที่สร้างความกำกวม (Ambiguous) ในการอ่าน

ข้อความที่ผู้ใช้สื่อสารกันในเว็บไซต์เครือข่ายทางสังคม (Social network website) มีการเล่นคำ หรือสร้างรูปแบบการเขียนที่ไม่ถูกต้องตรงตัวในแบบต่างๆ [6] ซึ่งจะมีผลทำให้ใช้วิธีการอ่านที่ไม่ตรงตามหลักภาษาที่แตกต่างกันออกไป ทำให้เกิดรูปแบบความไม่เป็นทางการต่างๆที่ทำให้ระบบประมวลผลภาษามนุษย์ (Natural language processing) ทำงานได้ยากขึ้นและมีความผิดพลาดมากขึ้น ในกรณีที่มีข้อความประเภทนี้ป้อนเข้าสู่ระบบสังเคราะห์เสียงพูด (Text-to-speech system) จะทำให้ระบบสังเคราะห์เสียงอ่านข้ามข้อความที่เขียนไม่ตรงตามกฎที่สร้างไว้ในระบบ หรืออ่านทุกข้อความตามกฎซึ่งอาจทำให้ผลจากการอ่านไม่ตรงกับความต้องการของผู้เขียน ทำให้ข้อมูลขาดหายและสื่อสารไปยังผู้ฟังได้ไม่ตรงตามความต้องการ ลักษณะที่ผู้เขียนนิยมใช้มากที่สุดเพื่อสื่อถึงการสร้างเสียงอ่านที่ไม่ตรงกับอักขรวิธีคือการซ้ำตัวอักษรเพื่อเพิ่มพื้นที่คำ ซึ่งสื่อถึงการออกเสียงที่ยาวขึ้น เช่น อ้าววว เห้ยยยย เป็นต้น ลักษณะต่างๆเหล่านี้ที่เกิดขึ้นในภาษาไทย ทำให้การแปลงจากข้อความเป็นคำอ่านมิได้ตรงตามอักขรวิธีทั้งหมดอีกต่อไปซึ่งงานวิจัยที่เสนอวิธีการในการจัดการข้อความไม่เป็นทางการจะจัดการข้อความเหล่านี้ด้วยวิธีการที่แตกต่างกันออกไป โดยส่วนมากมีวัตถุประสงค์เพื่อแปลงให้เป็นคำที่เป็นทางการ เช่น การวิเคราะห์ข้อความที่มีการพิมพ์ผิดเปลี่ยนแปลงตัวสะกด และเว้นวรรคไม่ถูกต้องและใช้ตัวเปลี่ยนแปรสโทแคสติกแปลงเป็นข้อความที่มีในพจนานุกรม [7] หรือใช้หลักการทางมาชีนทรานสเลชัน (Machine translation) ในการสร้างฐานข้อมูลโดยมองว่าภาษาที่ไม่เป็นทางการนั้นเป็นอีกภาษาหนึ่งเพื่อแปลเป็นภาษาที่เป็นทางการ [8,9] อย่างไรก็ตามการวิเคราะห์รูปแบบและจำแนกประเภทการเขียนข้อความไม่เป็นทางการจะช่วยให้สามารถจัดการตามรูปแบบของความไม่เป็นทางการได้อย่างเหมาะสมมากขึ้น [6,10]

ระบบสังเคราะห์เสียงในภาษาต่างๆ ได้นำเทคนิคการทำข้อความให้เป็นบรรทัดฐานเดียวกัน (Text normalization) มาใช้ในการแก้ปัญหาความกำกวมในการอ่านอย่างแพร่หลาย ซึ่งเมื่อพิจารณางานวิจัยต่างๆที่ทำเกี่ยวกับการทำข้อความให้เป็นบรรทัดฐานเดียวกันและการขจัดความกำกวม (Disambiguation) ของแต่ละภาษา เช่น ภาษาฮินดี [2] ภาษาบังคลาเทศ [3] ภาษาซองคา [4] ภาษาจีน [1,5] และ ภาษาญี่ปุ่น [1] แล้วจะพบว่ามีขั้นตอนที่คล้ายกัน [3] คือ เริ่มต้นจากการแบ่งข้อความเป็นแต่ละส่วน ตามอักขรวิธีของแต่ละภาษา เรียกว่า โทเค็น (Token) จากนั้นทำการจำแนกประเภทของแต่ละโทเค็น ขจัดความกำกวมของโทเค็นที่เป็นข้อความที่ไม่เป็นมาตรฐาน เพื่อให้ข้อความเป็นบรรทัดฐานเดียวกัน ซึ่งสามารถช่วยเพิ่มประสิทธิภาพในการทำงานของระบบสังเคราะห์เสียงให้อ่านคำที่ไม่ใช่อักขระในภาษานั้นๆที่แทรกอยู่ในข้อความได้ถูกต้องมากขึ้น

อย่างไรก็ตาม ข้อความภาษาไทยในสังคมออนไลน์มีความกำกวมและไม่เป็นทางการหลายประเภทที่ยังไม่สามารถแก้ไขเพียงด้วยใช้กฎการเขียนในภาษาไทย หรือการทำให้เป็นบรรทัดฐานเดียวกันเพียงอย่างเดียว เนื่องจากยังต้องอาศัยการระบุว่าเป็นข้อความกำกวมประเภทใด และจัดการความกำกวมให้ถูกต้องตามประเภทของการเขียนไม่ถูกต้องตรงตัว ดังตัวอย่างในตารางที่ 1-1

ตารางที่ 1-1 ตัวอย่างข้อความที่ไม่เป็นทางการ ประเภทความกำกวม และวิธีการแก้ไข

ข้อความ	**ประเภทความกำกวม**	**วิธีการแก้ไข**
ย๊ากส์ยาก	ผันวรรณยุกต์ไม่ถูกต้องตามอักขรวิธี	เลือกแปลงเป็นรูปวรรณยุกต์ที่ถูกต้องตามอักขรวิธี ที่ตรงตามความต้องการของผู้เขียนที่สุด
ม่ายยยยยย	เล่นคำด้วยวิธีซ้ำตัวอักษรซึ่งเกิดจากการที่ผู้เขียนสื่อให้อ่านเป็นเสียงยาวขึ้น	แปลงให้เป็นรูปที่สามารถอ่านตามอักขรวิธีได้ตรงตามความต้องการของผู้เขียน
ไอพอด	การเขียนคำอ่านทับศัพท์ ที่นำมาอ่านตามเสียงไทยแล้วไม่ตรงตามความต้องการของผู้เขียน	อ่านแบบทับศัพท์ หรืออ่านด้วยวิธีแปลงเป็นหน่วยเสียงภาษาอังกฤษ

จากการวิเคราะห์ข้อมูลตัวอย่าง การเล่นคำด้วยวิธีซ้ำตัวอักษรซึ่งเป็นรูปแบบที่เกิดมากที่สุด ส่งผลให้ระบบสังเคราะห์เสียง [11] พยายามสังเคราะห์อักษรที่ถูกซ้ำตามหลักการประสมอักษร ซึ่งแท้จริงแล้วผู้เขียนเพียงต้องการสื่อด้วยเสียงที่ยาวขึ้นหรือเน้นข้อความ เช่นคำว่า “ม่ายยยยย” ระบบจะอ่านเป็น “ม่าย-ยย-ยย” (/ m aɪ̯/ / j ʊj/ / j ʊj/) ตามหลักการประสมอักษร แต่ผู้เขียนเพียงต้องการให้อ่านว่า “ม่าย” (/ m aɪ̯ /) ด้วยเสียงที่ยาวขึ้นหรือเน้นข้อความที่ซ้ำอักษรเท่านั้น

งานวิจัยนี้จะทำการวิเคราะห์และสร้างแบบจำลองในการจำแนกข้อความภาษาไทยที่เขียนไม่ถูกต้องตรงตัวโดยการเล่นคำด้วยวิธีซ้ำตัวอักษร ที่แทรกอยู่ในข้อความภาษาไทยจากการสื่อสารในเว็บไซต์เครือข่ายทางสังคมโดยคำนึงถึงวิธีในการแปลงเป็นคำอ่าน เพื่อสร้างกฎสำหรับจัดการข้อความและแปลงข้อความให้เป็นบรรทัดฐานที่เหมาะสม เพื่อเพิ่มประสิทธิภาพในการแปลงข้อความในภาษาไทยเป็นสัทอักษร และพัฒนาประสิทธิภาพของระบบสังเคราะห์เสียงพูดภาษาไทยให้ดียิ่งขึ้น

1.2 วัตถุประสงค์ของการวิจัย

เพื่อทำการวิเคราะห์ลักษณะเด่นของข้อความภาษาไทยที่ไม่เป็นทางการ ที่เกิดจากการเล่นคำด้วยวิธีซ้ำตัวอักษรจากข้อความในเว็บไซต์สังคมออนไลน์ และสร้างแบบจำลองในการจำแนกประเภทของข้อความดังกล่าว โดยคำนึงถึงวิธีการในการแปลงเป็นคำอ่าน เพื่อนำไปตัดสินใจและเลือกวิธีการแปลงให้เป็นบรรทัดฐานที่เหมาะสม

1.3 ขอบเขตของการวิจัย

1. งานวิจัยนี้จะทำการจำแนกประเภทในข้อความภาษาไทยเท่านั้น ไม่รวมถึงภาษาอื่นๆ
2. งานวิจัยนี้จะสร้างแบบจำลองสำหรับตรวจจับข้อความที่เป็นการเล่นคำด้วยวิธีซ้ำตัวอักษรเพื่อแปลงให้เป็นบรรทัดฐานที่เหมาะสมโดยคำนึงถึงวิธีการอ่าน ซึ่งจะแปลงเป็นข้อความต้นฉบับที่ถูกนำมาเขียนแบบซ้ำตัวอักษรเท่านั้น ไม่สามารถแก้ไขข้อความที่พิมพ์ผิดในกรณีอื่นได้
3. ข้อความที่นำมาใช้ในการเรียนรู้ของระบบ เป็นข้อความที่ใช้สื่อสารบนเว็บไซต์เครือข่ายทางสังคม ซึ่งมีทั้งภาษาพูด ภาษาเขียน และภาษาสัญลักษณ์

1.4 ลำดับขั้นตอนในการเสนอผลการวิจัย

1. ขั้นตอนการศึกษาเบื้องต้น
    1.1. ศึกษางานวิจัยเกี่ยวกับจัดการข้อความที่ไม่เป็นทางการ
    1.2. ศึกษาหลักการทำข้อความให้เป็นมาตรฐานเดียวกัน
    1.3. ศึกษาอัลกอริทึมและเทคนิคการเรียนรู้ด้วยเครื่องจักร (Machine Learning) ที่ใช้ในการจำแนกประเภท
    1.4. ศึกษาเครื่องมือที่ใช้ในงานวิจัย เช่น เครื่องมือสังเคราะห์เสียงพูดภาษาไทย(text-to-speech system) และเครื่องมือในการจำแนกประเภท (Classification Tool)
2. ขั้นตอนการออกแบบระบบและทำการทดลอง
    2.1. ออกแบบการทดลอง
    2.2. เก็บข้อมูลข้อความภาษาไทยจากเว็บไซต์เครือข่ายทางสังคม
    2.3. วิเคราะห์ข้อมูลเพื่อหาลักษณะเด่นที่เหมาะสม
    2.4. สร้างแบบจำลองในการจำแนกประเภท เพื่อตรวจจับข้อความที่มีการเล่นคำแบบซ้ำอักษร
    2.5. ทดสอบและวัดผลความแม่นยำของระบบในการตรวจจับข้อความที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษร
    2.6. สร้างแบบจำลองในการแปลงเป็นบรรทัดฐานที่เหมาะสม
    2.7. ทดสอบและวัดผลความถูกต้องของคำอ่านของคำที่ได้จากวิธีการแปลงเป็นบรรทัดฐานที่เหมาะสม
    2.8. วิเคราะห์ผลการทดลอง
3. สรุปผลและเรียบเรียงวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเพิ่มประสิทธิภาพการอ่านให้กับระบบสังเคราะห์เสียงพูดภาษาไทย ให้สามารถอ่านข้อความในเว็บไซต์สังคมเครือข่าย ลดการอ่านข้ามและอ่านผิดความหมาย
2. สามารถตรวจจับข้อความที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษรเพื่อแปลงข้อความเป็นบรรทัดฐานโดยคำนึงถึงเสียงอ่านที่ตรงตามความต้องการของผู้สื่อสาร และสามารถนำไปปรับใช้ในระบบสังเคราะห์เสียงพูดภาษาไทยได้อย่างเหมาะสม

3. สามารถนำกรอบงานนี้ไปประยุกต์ใช้กับภาษาอื่นหรือการเขียนแบบไม่เป็นทางการในรูปแบบอื่นๆ

1.6 ผลงานตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง "*Detection of Wordplay Generated by Reproduction of Letters in Social Media Texts*" จัดทำโดย "Pawanrat Hirankan, Atiwong Suchato and Proadpran Punyabukkana" ถูกนำเสนอในงานประชุมวิชาการ "The 10th International Joint Conference on Computer Science and Software Engineering: JCSSE'2013" ณ มหาวิทยาลัยมหาสารคาม ประเทศไทย ในวันที่ 30-31 พฤษภาคม 2556

1.7 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

ในวิทยานิพนธ์นี้ได้แบ่งเนื้อหาออกเป็น 5 บท คือ บทที่ 1 บทนำ กล่าวถึง ความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการวิจัย ขอบเขตของการวิจัย ประโยชน์ที่คาดว่าจะได้รับ ลำดับขั้นตอนในการเสนอผลการวิจัย และผลงานตีพิมพ์จากวิทยานิพนธ์ บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง กล่าวถึง แนวคิดและทฤษฎี ประกอบด้วย ทฤษฎีทางภาษาศาสตร์ ซึ่งได้แก่ อักษรภาษาไทยและระบบการเขียนภาษาไทย ไตรยางค์และการผันอักษร ทฤษฎีเกี่ยวกับการประมวลผลภาษามนุษย์ ซึ่งได้แก่ การแปลงข้อความภาษาไทยเป็นสัทอักษร และการตัดคำภาษาไทย รวมถึงการเรียนรู้ของเครื่องจักรที่เกี่ยวข้องกับงานวิจัย อีกส่วนคือ งานวิจัยที่เกี่ยวข้องกับการจัดการข้อความที่ไม่เป็นทางการ และการทำให้เป็นบรรทัดฐาน บทที่ 3 ขั้นตอนการดำเนินงานวิจัย กล่าวถึง ข้อมูลที่ใช้ในการวิจัย เครื่องมือที่ใช้ในงานวิจัย ขั้นตอนการดำเนินงานวิจัย ประกอบด้วย ขั้นตอนการวิเคราะห์ข้อมูลเบื้องต้นจากเว็บไซต์เครือข่ายทางสังคม ขั้นตอนวิธีในการการสร้างระบบตรวจจับการเล่นคำ และขั้นตอนการสร้างเครื่องมือในการตรวจจับข้อความที่มีการเล่นคำ และการจัดการทำข้อความให้เป็นบรรทัดฐานเพื่อส่งต่อให้ระบบสังเคราะห์เสียงสร้างเสียงอ่านได้อย่างถูกต้อง บทที่ 4 การทดลอง และอภิปรายผล กล่าวถึง การทดลองของวิทยานิพนธ์นี้และผลการทดลอง บทที่ 5 บทสรุปผลการวิจัยและข้อเสนอแนะ กล่าวถึง การสรุปผลการวิจัย ข้อเสนอในจุดเด่นจุดด้อยของงานวิจัย และงานวิจัยในอนาคต

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้อง ซึ่งแบ่งออกเป็น 2 ส่วนคือส่วนแรกจะกล่าวถึง ทฤษฎีที่เกี่ยวข้อง ได้แก่ ทฤษฎีทางภาษาศาสตร์ที่เกี่ยวข้องกับวิทยานิพนธ์นี้ ซึ่งได้แก่ อักษรภาษาไทยและระบบการเขียนภาษาไทยรวมถึงการผันวรรณยุกต์ภาษาไทย ทฤษฎีเกี่ยวกับการประมวลผลภาษามนุษย์ ได้แก่ การตัดคำภาษาไทย และการแปลงข้อความเป็นสัทอักษร และทฤษฎีเกี่ยวกับการเรียนรู้ของเครื่องจักรที่เกี่ยวข้องกับงานวิจัยนี้ ได้แก่ การเรียนรู้แบบต้นไม้ตัดสินใจ ในส่วนที่ 2 จะกล่าวถึงวรรณกรรมที่เกี่ยวข้องกับวิทยานิพนธ์นี้ ซึ่งได้แก่ งานวิจัยที่เกี่ยวข้องกับการจัดการกับข้อความที่ไม่เป็นทางการ (Casual text) และงานวิจัยที่เกี่ยวกับการทำให้เป็นบรรทัดฐาน (Normalization)

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ทฤษฎีทางภาษาศาสตร์

2.1.1.1 อักษรภาษาไทยและระบบการเขียนภาษาไทย

อักษรภาษาไทยเป็นรูปเขียนที่ใช้แทนเสียงในภาษาไทย ประกอบด้วยพยัญชนะ สระ และวรรณยุกต์ [12]

- **พยัญชนะ**

  พยัญชนะไทย มี 44 รูป ได้แก่ ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ

<u>หน้าที่ของพยัญชนะ</u>

- เป็นพยัญชนะต้น ซึ่งทำให้เกิด 21 หน่วยเสียง ดังแสดงในตารางที่ 2-1

ตารางที่ 2-1 สัญลักษณ์หน่วยเสียงพยัญชนะต้นในภาษาไทย [21]

พยัญชนะต้นในภาษาไทย	สัทอักษรสากล	สัญลักษณ์หน่วยเสียง
ก	/k/	K
ข ฃ ค ฅ ฆ	/k^h/	Kh
ง	/ŋ/	Ng

พยัญชนะต้นในภาษาไทย	สัทอักษรสากล	สัญลักษณ์หน่วยเสียง
จ	/c/	C
ฉ ช ฌ	/c^h/	Ch
ซ ศ ษ ส	/s/	S
ญ ย	/j/	J
ฎ ด	/d/	D
ฏ ต	/t/	T
ฐ ฑ ฒ ถ ท ธ	/t^h/	Th
ณ น	/n/	N
บ	/b/	B
ป	/p/	P
พ ภ ผ	/p^h/	Ph
ฟ ฝ	/f/	F
ม	/m/	M
ร	/r/	R
ล ฬ	/l/	L
ว	/w/	W
ห ฮ	/h/	H
อ	/z/	Z

- เป็นพยัญชนะท้าย หรือตัวสะกด ซึ่งสามารถแบ่งได้ดังตารางที่ 2-2

ตารางที่ 2-2 สัญลักษณ์หน่วยเสียงตัวสะกดในภาษาไทย [21]

มาตราตัวสะกด	พยัญชนะท้าย	สัทอักษรสากล	สัญลักษณ์หน่วยเสียง
แม่กก	ก ข ค ฆ	/k*/	k^
แม่กด	จ ช ซ ฎ ฏ ฑ ฒ ด ต ถ ท ธ ศ ษ ส	/t*/	t^
แม่กบ	บ ป ภ พ ฟ	/p*/	p^

มาตราตัวสะกด	พยัญชนะท้าย	สัทอักษรสากล	สัญลักษณ์หน่วยเสียง
แม่กน	น ณ ญ ร ล ฬ	/n/	n^
แม่กง	ง	/ŋ/	ng^
แม่กม	ม	/m/	m^
แม่เกอย	ย	/j/	j^
แม่เกอว	ง	/w/	w^

- เป็นอักษรควบกล้ำ ซึ่งจะเป็นพยัญชนะสองตัวเขียนเรียงกันอยู่ต้นพยางค์ และใช้สระเดียวกัน เวลาอ่านออกเสียงกล้ำเป็นพยางค์เดียวกัน เสียงวรรณยุกต์ของพยางค์นั้นจะผันเป็นไปตามเสียงพยัญชนะตัวหน้า ซึ่งจะมีพยัญชนะต้นควบ ร ล ว เท่านั้น ซึ่งจะทำให้เกิดหน่วยเสียงตามตารางที่ 2-3

ตารางที่ 2-3 ตารางสัญลักษณ์หน่วยเสียงพยัญชนะควบกล้ำ [21]

พยัญชนะควบกล้ำ	สัทอักษรสากล	สัญลักษณ์หน่วยเสียง
ปร-	/pr/	Pr
ปล-	/pl/	Pl
ตร-	/tr/	Tr
กร-	/kr/	Kr
กล-	/kl/	Kl
กว-	/kw/	Kw
พร-	/p^hr/	Phr
พล-, ผล-	/p^hl/	Phl
ทร-	/t^hr/	Thr
คร-, ขร-	/k^hr/	Khr
คล-, ขล-	/k^hl/	Khl
คว-	/k^hw/	Khw

- เป็นอักษรนำ โดยการใช้พยัญชนะ 2 ตัวประสมสระเดียวกัน มีหลักการอ่านดังนี้

1. อ่านออกเสียงร่วมกันสนิทเป็นพยางค์เดียว ได้แก่
   - เมื่อ ห นำอักษรต่ำ เช่น หยุด หวาน หลอก หญิง เหงา หรูหรา
   - เมื่อ อ นำ ย มี 5 คำ คือ อย่า อยู่ อย่าง อยาก
2. อ่านออกเสียง 2 พยางค์ พยางค์แรกออกเสียง อะ กึ่งเสียง พยางค์หลังออกเสียงตามสระที่ประสมอยู่และออกเสียงเหมือน ห นำ เช่น ตลาด, สนาม เป็นต้น

- เป็นสระหรือส่วนหนึ่งของสระ ได้แก่ พยัญชนะ อ, ย, ว, ร ในบางกรณีจะมีการใช้พยัญชนะ แทนรูปสระเช่น

  ใช้ รร ทำหน้าที่แทนวิสรรชนีย์ หรือสระอะ เช่น สรรค์
  ใช้ ว เป็นสระ อัว ลดรูป เช่น กวน ชวน
  ใช้ ย เป็นส่วนประกอบของสระ เช่น เสีย เสย
  ใช้ อ เป็นสระและส่วนหนึ่งของสระ เช่น รอ เออ
- เป็นตัวการันต์ ซึ่งทำให้ไม่ออกเสียงพยัญชนะที่เป็นตัวการันต์

- **สระ**

สระในภาษาไทยมี 21 รูป สามารถประกอบได้เป็น 32 เสียง ดังนี้

1. สระเดี่ยว (Monophthong) ในภาษาไทยมีทั้งหมด 18 หน่วยเสียง ซึ่งสามารถแบ่งได้เป็น สระเสียงสั้น 9 หน่วยเสียง และสระเสียงยาว 9 หน่วยเสียง
2. สระประสม (Diphthong) ในภาษาไทยมีทั้งหมด 6 หน่วยเสียง ซึ่งสามารถแบ่งได้เป็น สระเสียงสั้น 3 หน่วยเสียง และสระเสียงยาว 3 หน่วยเสียง
3. สระเกิน คือ สระที่มีเสียงซ้ำกับสระแท้แต่มีพยัญชนะท้ายผสมอยู่ด้วย ซึ่งมีอยู่ 8 เสียงด้วยกัน

ซึ่งรูปและเสียงสระเดี่ยวและสระประสมแสดงในตารางที่ 2-4 และสระเกินแสดงในตารางที่ 2-5 ตามลำดับ

ตารางที่ 2-4 ตารางแสดงสัญลักษณ์ของหน่วยเสียงสระในภาษาไทย [21]

ประเภทสระ	เสียงสั้น		เสียงยาว	
	รูปสระ	สัญลักษณ์หน่วยเสียง	รูปสระ	สัญลักษณ์หน่วยเสียง
สระเดี่ยว	–ะ	a	–า	aa
	–ิ	i	–ี	ii
	–ึ	v	–ือ	vv
	–ุ	u	–ู	uu
	เ–ะ	e	เ–	ee
	แ–ะ	x	แ–	xx
	โ–ะ	o	โ–	oo
	เ–าะ	@	–อ	@@
	เ–อะ	q	เ–อ	qq
สระประสม	–ัวะ	ua	–ัว	uua
	เ–ียะ	ia	เ–ีย	iia
	เ–ือะ	va	เ–ือ	Vva

ตารางที่ 2-5 ตารางแสดงสระเกินในภาษาไทย

เสียงสั้น	เสียงยาว
อำ (อะ+ม)	-
ไอ (อะ+ย)	-
ใอ (อะ+ย)	-
เอา (อะ+ว)	-
ฤ (ร+อึ)	ฤา (ร+อึ)
ฦ (ล+อึ)	ฦา (ล+อึ)

ตำแหน่งการวางสระในภาษาไทย

- หน้าพยัญชนะต้น เช่น แต่ โธ่
- หลังพยัญชนะต้น เช่น ตา มา
- หน้าและและพยัญชนะต้น เช่น เรา เบาะ

- บนพยัญชนะต้น เช่น ลืม ดี
- หน้าและบนและหลังพยัญชนะต้น เช่น เสียง เกลือ
- ล่างพยัญชนะต้น เช่น ลุง คุณ

- **วรรณยุกต์**

วรรณยุกต์ไทยมี 4 รูปได้แก่ ◌่ ◌้ ◌๊ ◌๋ และมี 5 เสียง ได้แก่เสียง สามัญ เอก โท ตรี และจัตวา

ไตรยางค์ และการผันอักษร

ไตรยางค์หรืออักษรสามหมู่คือการแบ่งพยัญชนะไทยออกเป็นสามหมวด ตามลักษณะการผันวรรณยุกต์ เนื่องจากพยัญชนะไทยเมื่อกำกับด้วยวรรณยุกต์หนึ่งๆ แล้วจะมีเสียงวรรณยุกต์ที่แตกต่างกัน ซึ่งไตรยางค์จะแบ่งพยัญชนะไทยออกเป็น 3 ประเภท ดังนี้

- อักษรสูง มี 11 ตัว ได้แก่ ข ฃ ฉ ฐ ถ ผ ฝ ศ ษ ส ห
- อักษรกลาง มี 9 ตัว ได้แก่ ก จ ฎ ฏ ด ต บ ป อ
- อักษรต่ำ มี 24 ตัว ได้แก่ ค ฅ ฆ ง ช ซ ฌ ญ ฑ ฒ ณ ท ธ น พ ฟ ภ ม ย ร ล ว ฬ ฮ

หลักการผันอักษรจะขึ้นอยู่กับหมู่อักษร คำเป็น คำตาย และเสียงสั้น หรือ เสียงยาว ดังตารางที่ 2-6

ตารางที่ 2-6 การผันอักษรของอักษรสามหมู่ [12]

อักษรสามหมู่	เสียงสามัญ	เสียงเอก	เสียงโท	เสียงตรี	เสียงจัตวา	หมายเหตุ
อักษรกลาง : ก จ ด ฎ ต ฏ บ ป อ คำเป็น คำตาย	ปา	ป่า กัด	ป้า กั้ด	ป๊า กั๊ด	ป๋า กั๋ด	คำเป็น : พื้นเสียงเป็นเสียงสามัญ คำตาย : พื้นเสียงเป็นเสียงเอก
อักษรสูง : ข ฃ ฉ ฐ ถ ผ ฝ ศ ษ ส ห คำเป็น คำตาย	- -	ข่า ขัด	ข้า ขั้ด	- -	ขา -	คำเป็นพื้นเสียงเป็นเสียงจัตวา คำตายพื้นเสียงเป็นเสียงเอก

อักษรสามหมู่	เสียงสามัญ	เสียงเอก	เสียงโท	เสียงตรี	เสียงจัตวา	หมายเหตุ
อักษรต่ำ(อักษรที่เหลือ 24 ตัว) คำเป็น คำตายเสียงยาว คำตายเสียงสั้น	 คา - -	 - - -	 ค่า คาด ค่ะ	 ค้า ค้าด คะ	 - - -	คำเป็นพื้นเสียงเป็นเสียงสามัญ หากผันร่วมกับอักษรสูงจะผันได้ครบ 5 เสียง เช่น คา ข่า ค่า (ข้า) ค้า ขา

2.1.1.2 **พยางค์และการประสมอักษร**

พยางค์คือเสียงที่เปล่งออกมาครั้งหนึ่งๆ จะมีความหมาย หรือไม่มีความหมายก็ได้ ซึ่งจะเกิดจากการเปล่งเสียงพยัญชนะ เสียงสระ และเสียงวรรณยุกต์ตามกันออกมาอย่างต่อเนื่อง จนฟังดูเหมือนกับเปล่งเสียงออกมาในครั้งเดียวกัน ซึ่งเรียกว่าการประสมเสียงในภาษา [13]

ซึ่งพยางค์จะประกอบด้วยโครงสร้าง 4 รูปแบบดังนี้

1. การประสมอักษร 3 ส่วน ได้แก่ พยางค์ที่เกิดจากการประสมของ
   พยัญชนะต้น + สระ + วรรณยุกต์
   เช่น ฝา ค้า เป็นต้น
2. การประสมอักษร 4 ส่วนปกติได้แก่ พยางค์ที่เกิดจากการประสมของ
   พยัญชนะต้น + สระ+ พยัญชนะตัวสะกด+ วรรณยุกต์
   เช่น มาด ร้าย
3. การประสมอักษร 4 ส่วนพิเศษ ได้แก่ พยางค์ที่เกิดจากการประสมของ
   พยัญชนะต้น + สระ+ วรรณยุกต์+ การันต์
   เช่น เล่ห์ เบียร์ เป็นต้น
4. การประสมอักษร 5 ส่วน ได้แก่ พยางค์ที่เกิดจากการประสมของ
   พยัญชนะต้น + สระ + พยัญชนะตัวสะกด+ วรรณยุกต์+ การันต์
   เช่น ลักษณ์ ขันธ์ สังข์ จันทร์ เป็นต้น

2.1.2 ทฤษฎีเกี่ยวกับการประมวลผลภาษามนุษย์

2.1.2.1 การตัดคำภาษาไทย (Thai Word Segmentation)

การตัดคำภาษาไทยเป็น การระบุขอบเขตของคำเนื่องจากภาษาไทยเป็นภาษาที่ไม่มีการเว้นวรรคระหว่างคำ การหาขอบเขตของคำจึงเป็นสิ่งที่จำเป็นที่จะต้องทำเป็นอันดับแรกก่อนนำไปประมวลผลทางภาษาขั้นตอนอื่นๆ ในปัจจุบันวิธีที่ใช้ในการระบุขอบเขตของคำมี 2 วิธีใหญ่ๆได้แก่ [14]

1.การตัดคำโดยวิธีใช้พจนานุกรม (Dictionary-based: DCB) วิธีการนี้อาศัยการค้นคำในพจนานุกรมมาเป็นหลักในการตัดคำซึ่งวิธีการที่เป็นที่นิยมได้แก่

- การตัดคำแบบเลือกคำยาวที่สุด (Longest Maching): หลักการทำงานของกระบวนการตัดคำด้วยพจนานุกรมวิธีนี้ จะทำการตรวจสอบสายอักขระ (String) ที่เข้ามาจากซ้ายไปขวา กับคำที่อยู่ในพจนานุกรมในกรณีที่มีการตรวจสอบแล้วพบว่าพบคำมากกว่า 1 คำในพจนานุกรมจะทำการเลือกคำที่ยาวที่สุด ทำไปเรื่อยๆจนจบ แต่ในกรณีที่เลือกคำที่ยาวที่สุดไปแล้ว ทำให้เกิดคำที่ไม่ปรากฏในพจนานุกรม ก็ยอมให้มีการย้อนรอย (Back Trancking) กลับไปเลือกคำที่ยาวรองลงมาแทน

- การตัดคำโดยเลือกแบบเหมือนที่สุด (Maximum Matching): วิธีการนี้สามารถแก้ไขความบกพร่องของการตัดคำแบบเลือกคำยาวที่สุดได้ โดยจุดบกพร่องที่กล่าวนี้คือขั้นตอนวิธีการตัดคำแบบเลือกคำยาวที่สุด จะเลือกคำที่ยาวเกินไปตั้งแต่ครั้งแรก ทำให้ข้อความที่ตามมาเกิดข้อผิดพลาดได้ หลักการของการตัดคำโดยเลือกแบบเหมือนมากที่สุดคือ ขั้นตอนแรกจะทำการตัดคำที่เป็นไปได้ในทุกๆแบบก่อน แล้วหลังจากนั้นเลือกประโยคที่มีจำนวนคำน้อยที่สุด เช่น คำว่า ไปหามเหสี สามารถตัดได้เป็น “ไป-หาม-เห-สี” กับ “ไป-หา-มเหสี” ซึ่งเมื่อพิจารณาจากจำนวนคำแล้ว วิธีการนี้จะเลือก ไป หา มเหสี ซึ่งเป็นประโยคที่มีจำนวนคำน้อยสุดเป็นประโยคที่ถูกต้อง

2. การตัดคำโดยใช้วิธีการเรียนรู้ของเครื่องจักร (Machine learning-based: MLB)

วิธีนี้จะใช้อัลกอริทึมในการเรียนรู้ของเครื่องจักรมาสร้างแบบจำลองที่เรียนรู้จากคลังข้อมูลข้อความที่มีการตัดคำและป้ายระบุ ซึ่งแบบจำลองที่ได้จะเป็นตัวจำแนกประเภทแบบไบนารี ที่ใช้ทำนายว่าตัวอักษรนั้นๆเป็นจุดเริ่มต้นใหม่ของคำหรือไม่ โดยไม่ต้องอาศัยพจนานุกรม

วิธีการตัดคำโดยใช้วิธีการเรียนรู้ของเครื่องจักรที่มีความแม่นยำสูงที่สุดสำหรับภาษาไทยในปัจจุบันคือเครื่องมือตัดคำที่สร้างจากแบบจำลองคอนดิชันแนลแรนดอมฟิลด์ (Conditional Random Field : CRF) [14,15]

2.1.2.2 การแปลงข้อความเป็นสัทอักษร

การแปลงข้อความเป็นสัทอักษร [16] เป็นการวิเคราะห์หาคำอ่านจากข้อความ ซึ่งผลที่ได้จากการแปลงจะเป็นสัญลักษณ์แทนหน่วยเสียง การประมวลผลวิเคราะห์คำอ่านจากข้อความนั้นทำได้หลายวิธีการ ซึ่งเครื่องมือในการสร้างโมเดลการออกเสียง มีสองวิธีได้แก่

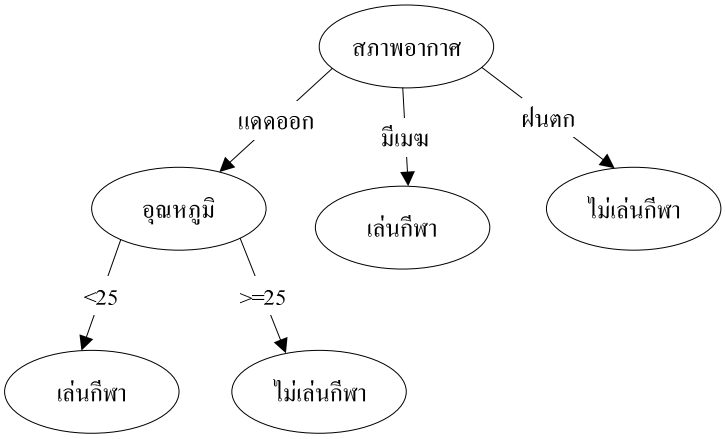
1.ใช้ข้อมูลจากพจนานุกรมคำอ่าน (Pronunciation dictionary) ซึ่งวิธีนี้จะรวดเร็วและถูกต้องแม่นยำสูง แต่ต้องอาศัยการเก็บข้อมูลจำนวนมาก และไม่สามารถจัดการกับข้อความที่ไม่มีในพจนานุกรมได้

2. ใช้วิธีการวิเคราะห์ทางสัทศาสตร์ (Phonological analysis) และนำไปสร้างโมเดลในการอ่านออกเสียง ซึ่งอาจจะเป็นวิธีใช้กฎ วิธีต้นไม้ตัดสินใจ วิธีพีจีแอลอาร์พาสเซอร์ (probabilistic generalized LR parser: PGLR Parser) หรือ ใช้เทคนิคการขับเคลื่อนด้วยข้อมูลตัวอย่าง (Example Based G2P: EBG2P)

2.1.3 การเรียนรู้ของเครื่องจักรที่เกี่ยวข้องกับงานวิจัย

2.1.3.1 การเรียนรู้แบบต้นไม้ตัดสินใจ

การเรียนรู้ต้นไม้ตัดสินใจ (Decision Tree Learning) [17] เป็นเทคนิคการเรียนรู้ที่ความสัมพันธ์ระหว่างข้อมูลขาเข้าและผลลัพธ์สามารถเขียนแทนด้วยต้นไม้ตัดสินใจ ดังตัวอย่างในภาพที่ 2-1 โดยทั่วไปแล้วต้นไม้ตัดสินใจจะประกอบด้วยโหนด (node) และเส้นเชื่อม (edge) โดยโหนดแทนลักษณะเด่นของข้อมูล (feature) เข้าที่ต้องพิจารณา และเส้นเชื่อมแทนค่าที่เป็นไปได้ของลักษณะเด่นนั้นๆและโหนดใบ (leaf node) หรือโหนดที่ไม่มีลูกแทนผลลัพธ์ที่เป็นคำตอบ การใช้ต้นไม้ตัดสินใจในการหาผลลัพธ์ของข้อมูลเข้านั้น เริ่มจากการพิจารณาลักษณะเด่นที่อยู่ในโหนดราก (root node) ของต้นไม้ตัดสินใจ หลังจากนั้นจึงท่องต้นไม้ตามค่าลักษณะเด่นของข้อมูลเข้าว่ามีค่าเป็นเท่าใด และทำกระบวนการดังกล่าวซ้ำไปเรื่อยๆ จนถึงโหนดใบ ซึ่งค่าที่โหนดใบจะเป็นคำตอบของข้อมูลเข้านั่นเอง ตัวอย่างเช่น ในกรณีที่ข้อมูลเข้าคือ สภาพอากาศมีแดดออก และอุณหภูมิเป็น 24 องศานั้น การทำนายผลลัพธ์เริ่มจากการพิจารณาลักษณะเด่นของสภาพอากาศซึ่งมีค่าเป็นแดดออก ทำให้ต้องพิจารณาลักษณะเด่นอุณหภูมิต่อ ซึ่ง 24 องศาน้อยกว่า 25 องศา ทำให้ได้ผลลัพธ์เป็น เล่นกีฬา



ภาพที่ 2-1 ต้นไม้ตัดสินใจสำหรับทำนายว่าจะเล่นกีฬาหรือไม่

สำหรับเทคนิคการสร้างต้นไม้ตัดสินใจจากชุดตัวอย่างข้อมูลนั้น โดยทั่วไปแล้วทำได้โดยการเลือกลักษณะเด่น จากลักษณะเด่นทั้งหมดมาสร้างเป็นโหนดราก หลังจากนั้นจึงแบ่งข้อมูลออกเป็นกลุ่มๆตามค่าของลักษณะเด่นดังกล่าว แล้วจึงทำการเลือกลักษณะเด่นเพื่อแบ่งข้อมูลในแต่ละกลุ่ม เป็นกลุ่มย่อยไปเรื่อยๆ จนข้อมูลในแต่ละกลุ่มมีคำตอบเหมือนกันหมด ในกรณีที่ลักษณะเด่นเป็นค่าต่อเนื่องต้องทำการแปลงลักษณะเด่นดังกล่าวให้เป็นค่าไม่ต่อเนื่องเสียก่อน

การเลือกลักษณะเด่นนั้นมีหลายแนวทางด้วยกัน แนวทางที่นิยมใช้คือค่าเกน (Information Gain) ซึ่งแสดงถึงความสามารถในการแบ่งข้อมูลออกจากกันของแต่ละลักษณะเด่น ลักษณะเด่นที่สามารถแยกข้อมูลที่มีคำตอบต่างกันออกจากกันได้ จะต้องมีค่าเกนมากกว่าลักษณะเด่นที่ไม่สามารถแยกข้อมูลที่มีคำตอบต่างกันออกจากกัน โดยค่าเกนของลักษณะเด่น X นั้นคำนวณจากค่าเอนโทรปี (Entropy) ทั้งหมดของชุดข้อมูลนั้นลบด้วยค่าเอนโทรปีหลังจากเลือกลักษณะเด่น X เพื่อแบ่งข้อมูลออกเป็นกลุ่มๆ ซึ่งค่าเอนโทรปีหลังจากการเลือกลักษณะเด่น X นั้นคือผลรวมของผลคูณระหว่างค่าเอนโทรปีของแต่ละโหนดกับอัตราส่วนของตัวอย่างในแต่ละกิ่งต่อตัวอย่างทั้งหมดในโหนดนั้นๆ

ถ้ากำหนดให้ชุดข้อมูลสำหรับการเรียนรู้คือ T และลักษณะเด่นที่โหนดคือ X ที่มีค่าที่เป็นไปได้ n ค่า โหนดปัจจุบันจะแบ่งตัวอย่าง T ออกตามแต่ละเส้นเชื่อมเป็น $\{t_1,t_2,\ldots,t_n\}$ ตามค่าที่เป็นไปได้ของ X ค่าเอนโทรปีหลังจากแบ่งชุดข้อมูลเป็นกลุ่มๆตามคุณรฆสมบัติ X คำนวณได้จากสมการ 2.1

$$I_x(T) = \sum_{i=1}^{n} \left|\frac{t_i}{T}\right| I(t_i) \qquad (2.1)$$

และค่าเกนของลักษณะเด่น X คำนวณได้ด้วยจากสมการ 2.2

$$Gain(X) = I(T) - I_x(T) \quad (2.2)$$

เมื่อค่าเอนโทรปีของชุดข้อมูล T ที่ประกอบไปด้วยค่าผลลัพธ์ที่เป็นไปได้ $\{m_1, m_2, \ldots, m_n\}$ คำนวณได้จากสมการ 2.3

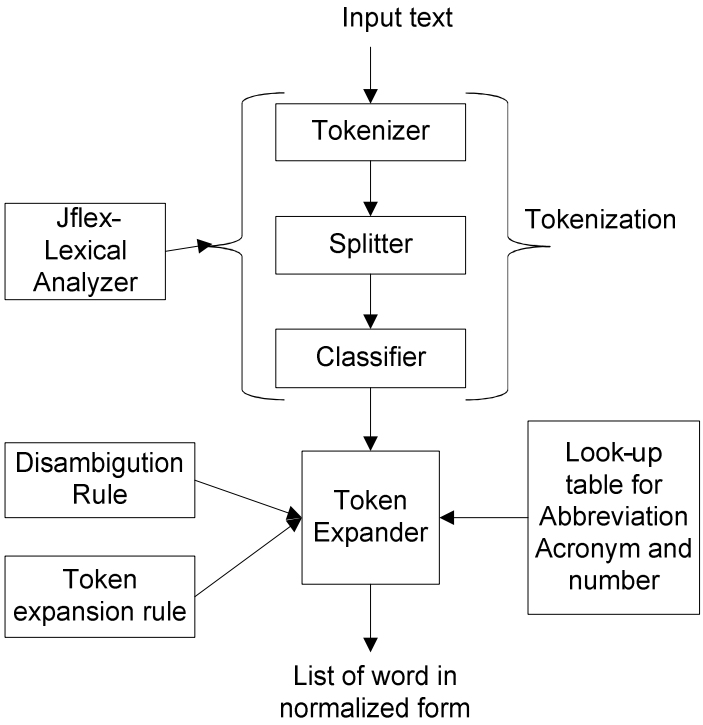
$$I(m) = \sum_{i=1}^{n} -P(m_i) \log_2 P(m_2) \quad (2.3)$$

ตัวอย่างของขั้นตอนวิธีสำหรับสร้างต้นไม้ตัดสินใจ ได้แก่ CART, ID3, C4.5, C5.0 เป็นต้น ซึ่งข้อเสียของวิธีการแบบนี้คือ ไม่เหมาะกับระบบที่มีจำนวนคุณลักษณะมากๆ ข้อดีของแบบจำลองต้นไม้ตัดสินใจคือ กฎที่สร้างจากแบบจำลองนี้จะง่ายต่อการเข้าใจและตีความผลลัพธ์ที่ได้จากการเรียนรู้ อีกทั้งยังสามารถรองรับข้อมูลได้หลายประเภททั้งแบบต่อเนื่องและแบบไม่ต่อเนื่องอีกด้วย

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยเกี่ยวกับการทำให้ข้อความเป็นบรรทัดฐาน

การจัดการกับคำที่ไม่เป็นมาตรฐาน เช่น ข้อความที่ไม่เป็นทางการ สัญลักษณ์ ตัวเลข และอักษรในภาษาอื่นที่แทรกอยู่ในข้อความ เพื่อสร้างกระบวนการทำให้ข้อความเป็นบรรทัดฐานเดียวกัน ถูกนำมาใช้กับภาษาต่างๆ อย่างแพร่หลายเพิ่มมากขึ้น ซึ่งกระบวนการในการทำให้เป็นบรรทัดฐานเดียวกันส่วนใหญ่จะคล้ายคลึงกันดังภาพที่ 2-2 คือเริ่มต้นจากการตัดแบ่งข้อความเป็นส่วนๆเรียกว่าโทเค็น จากนั้นทำการจำแนกประเภทโทเค็น และการสร้างกฎหรือวิธีการจัดการกับโทเค็นประเภทต่างๆเพื่อขจัดกำกวม และการแปลงเป็นคำมาตรฐาน (Standard word) เพื่อส่งต่อไปยังส่วนแปลงข้อความเป็นคำอ่าน แต่กระบวนการและอัลกอริทึมในการจำแนกและจัดการกับความกำกวมนั้น แตกต่างกันออกไปในแต่ละภาษา



ภาพที่ 2-2 ระบบการข้อความให้เป็นบรรทัดฐานเดียวกันสำหรับภาษาบังคลาเทศ [3]

ในงานวิจัยของ Craig Olinsky และ Alan W Black [1], (2000) ได้เสนอวิธีการจัดการกับคำที่ไม่เป็นมาตรฐานและคำพ้องรูปโดยใช้อัลกอริทึม คาร์ท (CART : Classification and regression trees) ซึ่งเป็นรูปแบบหนึ่งของต้นไม้ตัดสินใจ ในการแบ่งประเภทของข้อความในภาษาญี่ปุ่นและภาษาจีน แล้วทำการแปลงเป็นข้อความที่เป็นบรรทัดฐานเดียวกัน ด้วยตัวขยายของแท็ก (Tag Expander) ตามประเภทแท็ก โดยในงานนี้ใช้ข้อมูลจากหนังสือพิมพ์ธุรกิจ จากงานวิจัยนี้ได้แบ่งข้อความที่ไม่เป็นมาตรฐานเป็น 3 ประเภทได้แก่ NSW ที่เป็นตัวเลข ตัวอักษร และอื่นๆ ซึ่งจากข้อมูลทำให้พบว่ามีการกระจายตัวของข้อมูลที่เป็นตัวเลขที่อ่านตามหลักมากที่สุด ตามมาด้วยวันที่ และจำนวนเงิน ซึ่งรวมกันสามประเภทนี้มีถึง 71% ของข้อมูล NSW ทั้งหมด และจากการทดลองนี้มีค่าความแม่นยำในข้อความที่เป็นตัวอักษร 93.6% และข้อความที่เป็นตัวเลข 72.9%

ซึ่งต่อมาในงานวิจัยของ K. Panchapagesan และคณะ [2] , (2004) ใช้ เฟล็กซ์ (Flex) ซึ่งเป็นเครื่องมือที่ใช้หลักการของเรกูลาร์เอ็กซ์เพรสชั่น(Regular Expression) ในการตัดข้อความ แบ่งข้อความและจำแนกประเภทของข้อความในภาษาฮินดี และ ใช้ต้นไม้ตัดสินใจและลิสการตัดสินใจ (Decision tree และ Decision list) ในการแปลงหรือขยายข้อความให้เป็นบรรทัดฐานเดียวกัน

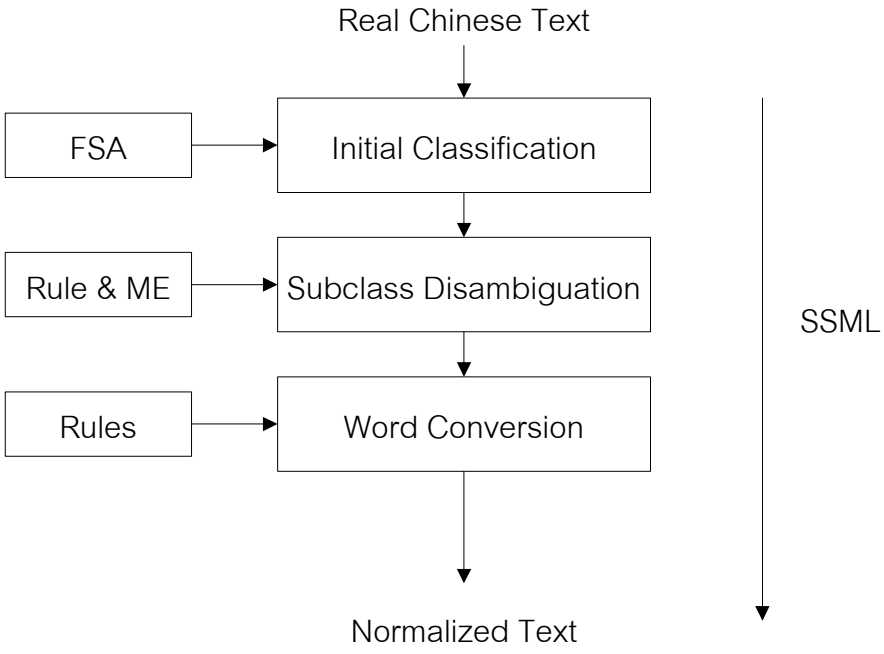
ตัวอย่าง เมื่อ X เป็นตัวอักษรในภาษาฮินดี

Input : XXXXX Test@gmail.com XXXXX

เฟล็กซ์จะใช้หลักการของเรกูลาร์เอ็กซ์เพรสชั่นในการสแกนข้อมูลและระบุว่าส่วนใดเป็นข้อความประเภทไม่เป็นมาตรฐาน (Non-standard word)ประเภทใด

Output : XXXXX<EMAIL>Test@gmail.com</EMAIL>XXXXX

ต่อมาในงานวิจัยของ Tao และคณะ [5] ,(2008) มีการสร้างระบบการทำให้ข้อความเป็นบรรทัดฐานเดียวกันที่ซับซ้อนขึ้น สำหรับภาษาจีนโดยแบ่งการประมวลผลเป็นสามระยะ ดังภาพที่ 2-3 โดยระยะแรกจะใช้ ไฟไนท์สเตท ออโตมาตา (Finite State Automata: FSA)ในการแบ่งกลุ่มประเภทใหญ่ๆของข้อความที่ไม่เป็นมาตรฐานก่อน จากนั้นระยะที่สองใช้ตัวจำแนกประเภทแม็กซ์เอ็นโทรปี (Max Entrypy Classifier: ME) ร่วมกับการใช้กฎในการแปลงข้อความเป็นกลุ่มย่อย จากนั้นระยะที่สามใช้กฎในการแปลงเป็นข้อความที่มีมาตรฐานเดียวกัน และนำไปทดลองกับระบบสังเคราะห์เสียง BaiLing



ภาพที่ 2-3 ขั้นตอนการทำข้อความให้เป็นบรรทัดฐานเดียวกันโดยแบ่งเป็นสามระยะ

ซึ่งจากการทดสอบระบบสังเคราะห์เสียงที่ใช้วิธีการแปลงข้อความให้เป็นบรรทัดฐานเดียวกันโดยแบ่งเป็นสามระยะนี้ได้ผลความแม่นยำที่ค่อนข้างสูง ดังตารางที่ 2-7

ตารางที่ 2-7 ประสิทธิภาพของระบบสังเคราะห์เสียงจากระบบที่ผ่านการทำให้เป็นบรรทัดฐานโดยแบ่งการประมวลผลเป็นสามระยะ [5]

	Precision	Recall
Simple Rules	90.77 %	88.59 %
FSA+ME+Rules	93.81 %	91.58 %
FSA+ME&Rule+Rules	96.02 %	93.74 %

ต่อมาในงานวิจัยของ Firoj Alam (2009) [3] ได้ทำการสร้างกระบวนการแปลงข้อความให้เป็นมาตรฐานเดียวกัน (text normalization) ในภาษาบังคลาเทศ ด้วยการระบุกฎ (Rule based) ของประเภทสัญลักษณ์ต่างๆจาก corpus ภาษาบังคลาเทศ หลังจากที่ระบุกฎของประเภทสัญลักษณ์แล้ว จะสร้างกฎขึ้นมากลุ่มหนึ่งสำหรับ จัดการกับโทเค็นแต่ละประเภท ซึ่งได้นิยามตามความหมาย และการแปลงเป็นคำพูดตามวิธีของแต่ละกลุ่ม ดังภาพที่ 2-2 ซึ่งประสิทธิภาพของระบบการใช้กฎนี้ ให้ผลความแม่นยำ 99% ในการอ่านตัวเลขที่มีทศนิยม (100%) เวลา (67%) และจำนวนเงิน (100%) ซึ่งถูกแสดงอยู่ในรูปของทศนิยมเหมือนกัน

การเรียนรู้แบบต้นไม้ตัดสินใจเป็นวิธีการที่ถูกนำมาใช้ได้ผลดีในการจำแนกประเภทข้อความที่มีความกำกวมในหลายภาษาดังกล่าวข้างต้น ซึ่งในกรณีที่ภาษาไม่ซับซ้อนมากต้นไม้ตัดสินใจยังคงให้ผลที่มีประสิทธิภาพสูงและสามารถตีความหมายจากแบบจำลองที่ได้ ทั้งยังง่ายต่อการตีความผลลัพธ์ที่ได้จากการเรียนรู้

2.2.2 **งานวิจัยเกี่ยวกับการจัดการกับข้อความที่ไม่เป็นทางการ** (Casual text Processing)

การขยายตัวของการสื่อสารบนเว็บไซต์สังคมออนไลน์ทำให้มีการใช้ข้อความที่ไม่เป็นทางการด้วยการเล่นคำในรูปแบบต่างๆเพิ่มมากขึ้น งานวิจัยที่เกี่ยวข้องกับการจัดการข้อความที่ไม่เป็นทางการเริ่มต้นขึ้นจากการวิเคราะห์ข้อความสื่อสาร หรือ SMS ที่ใช้ในโทรศัพท์มือถือ ไปจนถึงยุคที่ใช้ข้อความสื่อสารในกระทู้หรือเว็บบอร์ดต่างๆ มาจนถึงข้อความในเว็บไซต์สังคมเครือข่าย

งานของ Alexander Clark [7] เป็นงานวิจัยเกี่ยวกับการวิเคราะห์ข้อความที่มีการพิมพ์ผิด เปลี่ยนแปลงตัวสะกด และเว้นวรรคไม่ถูกต้อง โดยงานวิจัยนี้ใช้ตัวเปลี่ยนแปรสโทแคสติก

(Stochastic transducers) มาวิเคราะห์เพื่อหารูปแบบการพิมพ์ผิดจากคลังข้อความ 100 ล้านคำของ Usenet news และเสนอแบบจำลองในการแก้ไขข้อความ

ซึ่งจากงานวิจัยนี้ได้พบว่า การปรากฏของข้อความที่เขียนถูกจะพบได้มากกว่าข้อความที่เขียนผิด และการเขียนผิดมีสองรูปแบบ คือความผิดพลาดจากการพิมพ์ (Typing error) ซึ่งความผิดพลาดจากการพิมพ์นั้นเกิดจากการกดปุ่มบนแป้นพิมพ์ผิดไป เนื่องจากข้อความหรือตัวอักษรที่ต้องการพิมพ์อยู่ใกล้กับตัวอักษรที่ต้องการ กับความผิดพลาดจากการสะกด (Spelling error) ซึ่งเกิดจากการพิมพ์ผิดที่เกิดจากความเข้าใจผิดในการสะกดคำจริงๆ หรือตั้งใช้ในการสะกดในรูปแบบที่ผิดนั้น

อย่างไรก็ตามงานวิจัยนี้ยังไม่สามารถวัดประสิทธิภาพของระบบได้ และสรุปว่าการพิมพ์ผิดดังกล่าวอาจเป็นได้ทั้งความผิดพลาดโดยตั้งใจหรือไม่ตั้งใจก็ได้ แต่ไม่ได้นำเสนอวิธีการจำแนกความตั้งใจของผู้เขียนดังกล่าว

งานวิจัยของ AiTi Aw, Min Zhang, Juan Xiao และ Jian Su [8] ได้ศึกษาเกี่ยวกับการแปลข้อความเอสเอ็มเอส (SMS text) ซึ่งมีการแตกต่างไปจากการเขียนปกติ และมีปรากฏการณ์พิเศษต่างๆที่ทำให้การแปลข้อความยุ่งยากมากขึ้นในการทำระบบแปลข้อความอัตโนมัติ ซึ่งในงานวิจัยนี้นำเสนอ อีกวิธีการหนึ่งในการแก้ไขความไม่เป็นทางการของข้อความเอสเอ็มเอส (SMS text) โดยการสร้างแบบจำลองในการแปลงข้อความเอสเอ็มเอส ให้เป็นบรรทัดฐานก่อนนำเข้าระบบมาชีนทรานสเลชันโดยนิยามของการทำข้อความเอสเอ็มเอสให้เป็นบรรทัดฐาน เป็นอีกภาษาหนึ่งที่จะแปลเป็นภาษาอังกฤษและพัฒนาระบบมาชีนทรานสเลชันในการแปลข้อความในระดับวลีขึ้นมา โดยประเมินผลบนคลังข้อมูลคู่ขนานกับข้อความเอสเอ็มเอสที่ถูกทำให้เป็นบรรทัดฐาน 5,000 ประโยค อย่างไรก็ตาม ประสิทธิภาพของระบบยังคงขึ้นกับขนาดของคลังข้อมูลที่นำมาใช้อยู่มากเกินไป ทำให้ระบบไม่สามารถทำงานได้ดีในกรณีที่คลังข้อมูลไม่ครอบคลุมรูปแบบปรากฏการณ์พิมพ์ผิดอื่นๆ

ซึ่งต่อมาในงานวิจัยของ Carlos A. Henr´ıquez Q. [9] ได้นำวิธีเอ็นแกรม (n-gram) มาใช้ในการทำข้อความที่ไม่เป็นทางการให้เป็นบรรทัดฐานโดยใช้หลักการทางมาชีนทรานสเลชันมาเป็นหลักในการจัดการกับข้อความที่ไม่เป็นทางการ อย่างไรก็ตามการใช้หลักการทางมาชีน-ทรานสเลชันมาเป็นหลักในการจัดการกับข้อความที่ไม่เป็นทางการยังคงต้องการฐานข้อมูลขนาดใหญ่ และประสิทธิภาพของระบบยังคงขึ้นกับขนาดของฐานข้อมูลอยู่มาก

ต่อมาในงานวิจัยของ Eleanor Clark [6] ได้เสนอการทำข้อความให้เป็นบรรทัดฐานสำหรับภาษาที่ไม่เป็นทางการในข้อความจากเว็บไซต์สังคมเครือข่ายโดยนำเสนอหมวดหมู่ของความไม่เป็น

ทางการในรูปแบบต่างๆ ซึ่งอาจเกิดจากความสร้างสรรค์หรือการแสดงตัวตนของผู้เขียน ที่ทำให้เกิดความไม่เป็นทางการสองระดับคือข้อความจะไม่สามารถนำไปเป็นอินพุตให้กับระบบประมวลผลทางภาษาได้เนื่องจากความไม่ถูกต้อง และผู้เขียนอาจจะต้องการสื่อสารด้วยภาษาที่เข้าใจเฉพาะกลุ่ม เป็นต้น งานวิจัยนี้กล่าวถึงการแปลงข้อความภาษาอังกฤษในสื่อสังคมออนไลน์ให้เป็นบรรทัดฐาน และการนำไปใช้โดยวัดผลจากระบบตรวจการสะกดคำ ซึ่งข้อมูลที่นำมาใช้ในงานวิจัยนี้จะเป็นข้อมูลจากทวิตเตอร์ (Micro blogging service : Twitter) ซึ่งจะวัดประสิทธิภาพของระบบโดยเปรียบเทียบความถูกต้องของข้อความก่อนและหลังการประมวลผลก่อนด้วยระบบตรวจสอบการสะกด (Spell Checker) ซึ่งในงานวิจัยนี้นำเสนอระบบการแปลงข้อความภาษาอังกฤษที่ไม่เป็นทางการ (Casual English Conversion System: CECS) ซึ่งประกอบด้วย

1. ระบบจำแนกประเภทภาษาอังกฤษที่ไม่เป็นทางการ (Casual English classification system)

ระบบ CECS จะแบ่งรูปแบบข้อความภาษาอังกฤษที่ไม่เป็นทางการออกเป็นหมวดหมู่ และจัดการด้วยวิธีการแก้ไขสำหรับรูปแบบความไม่เป็นทางการในแต่ละหมวดหมู่ ซึ่งในงานวิจัยนี้ระบบจะแบ่งความไม่เป็นทางการในภาษาอังกฤษออกเป็น 8 หมวดหมู่ดังนี้

1) การเขียนเป็นรูปย่อ เช่น nite (night) , saying (saying) และอาจมีตัวเลขแทรกมาเช่น gr8(great)
2) การเขียนด้วยอักษรย่อ เช่น lol (“laugh out loud”), iirc (“if I remember correctly”)
3) การพิมพ์ผิดหรือสะกดผิด เช่นwouls (“would”), rediculous (“ridiculous”)
4) การตัดหรือละเครื่องหมายแบ่งวรรคตอน เช่น im (“I’m”), dont (“don’t”)
5) การใช้ศัพท์สแลงที่ไม่มีในพจนานุกรม เช่น that was well mint (“that was very good”). It also includes specific cultural reference or in group-memes.
6) การเล่นคำ คือ การตั้งใจสะกดผิดเพื่อเน้นเสียงหรือเพิ่มผลกระทบบางอย่างทางการออกเสียง เช่น that was soooooo great (“that was so great”).
7) การป้องกันการถูกเซ็นเซอร์ คือการพยายามใช้ตัวเลขหรือเครื่องหมายวรรคตอนแทรกในข้อความแทนการพิมพ์คำเต็ม เช่น sh1t, f***
8) การพิมพ์สัญลักษณ์แทนอารมณ์ด้วยอักษรภาพ หรือ อีโมติคอน เช่น :) หน้ายิ้ม <3 หัวใจ เป็นต้น

2. การสร้างฐานข้อมูลในระบบ CECS (Database construction) ฐานข้อมูลที่ใช้ในระบบ CECS แปลผลและตรวจสอบด้วยคน ซึ่งมีทั้งหมด 1,043 รายการ ซึ่งข้อมูลเหล่านี้เป็นคำเดี่ยวหรือวลี ซึ่งแต่ละรายการจะนำมาจากข้อมูลชุดฝึกฝนซึ่งมาจาก Twitter , Youtube comment , Wiktionary และ Urban Dictionary แต่ละ record ของจะประกอบด้วยข้อมูล 4 คอลัมน์ได้แก่
    1) คำที่มีความผิดพลาด (error word) คือคำไม่เป็นทางการที่ปรากฏในข้อความ
    2) คำปกติ (regular word) เป็นคำที่ถูกต้องตามพจนานุกรมภาษาอังกฤษ
    3) ประเภทของความไม่เป็นทางการ (category)
    4) บันทึก (notes) ข้อมูลเพิ่มเติมเกี่ยวกับที่มาหรือวิธีการเกิดความไม่เป็นทางการหรือความตั้งใจของผู้เขียน

    โดยระบบ CECS จะใช้ทำการตัดแบ่งข้อความเป็นส่วนย่อยๆ ที่เรียกว่าโทเคน แล้วทำการตรวจสอบคำในฐานข้อมูล และทำการแปลงข้อความตามวิธีการที่เหมาะสมสำหรับแต่ละหมวดหมู่
3. Phrase matching rules ระบบในการแปลงวลีให้เป็นข้อความที่เป็นทางการ จากฐานข้อมูลอาจแปลงคำที่ไม่เป็นทางการต้นแบบหนึ่งคำ ไปเป็นคำที่เป็นทางการมากกว่าหนึ่งคำขึ้นกับบริบท

การจำแนกประเภทความไม่เป็นทางการจากงานวิจัยนี้ทำให้การทำให้เป็นมาตรฐานเดียวกันมีประสิทธิภาพสูงขึ้น อย่างไรก็ตามฐานข้อมูลที่นำมาใช้ และการจำแนกประเภทของคำที่ไม่เป็นทางการในงานวิจัยนี้ยังคงถูกสร้างโดยใช้คนตรวจสอบ ซึ่งในกรณีถ้าหากสามารถจำแนกประเภทของคำที่ไม่เป็นทางการได้อย่างอัตโนมัติ จะสามารถลดระยะเวลาหรือรองรับความผิดพลาดที่จะเกิดขึ้นใหม่ได้ในกรณีที่มีรูปแบบเดียวกัน โดยไม่จำเป็นต้องใช้ฐานข้อมูลขนาดใหญ่ขึ้น

ซึ่งต่อมาในงานวิจัยเกี่ยวกับการวิเคราะห์อารมณ์ในข้อความ ของ Hemalatha [10] ได้ใช้กฎการซ้ำของตัวอักษรอย่างน้อย 4 ตัวมาเป็นกฎในการจำแนกประเภทข้อความที่เป็นการเล่นคำด้วยวิธีซ้ำตัวอักษร ซึ่งได้ผลดีในภาษาอังกฤษ แต่เนื่องจากในภาษาไทย มีความกำกวมในแง่ของขอบเขตคำ และการซ้ำของพยัญชนะที่ยังคงเป็นโครงสร้างของการประสมอักษรได้ ทำให้การใช้กฎในการจำแนกข้อความที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษรยังไม่เพียงพอต่อข้อความที่พบในภาษาไทย

เนื่องจากการเล่นคำด้วยวิธีซ้ำตัวอักษร เป็นรูปแบบความไม่เป็นทางการที่มีมากที่สุดในข้อมูลตัวอย่างจากเว็บไซต์สังคมเครือข่ายของผู้ใช้งานภาษาไทยกลุ่มตัวอย่าง และมีผลทำให้การอ่านออกเสียงของระบบสังเคราะห์เสียงไม่ตรงกับความต้องการของผู้เขียน

งานวิจัยนี้จึงเลือกที่จะนำเทคนิคการเรียนรู้โดยใช้ต้นไม่ตัดสินใจ มาประยุกต์ใช้กับการตรวจจับข้อความที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษรจากข้อความภาษาไทยที่ไม่เป็นทางการในเว็บไซต์สังคมเครือข่าย เพื่อทำการแปลงเป็นบรรทัดฐานที่เหมาะสม เพื่อให้ระบบสังเคราะห์เสียงมีประสิทธิภาพมากขึ้น

บทที่ 3

ขั้นตอนการสร้างระบบตรวจจับการเล่นคำและทำให้เป็นบรรทัดฐาน

ในบทนี้จะกล่าวถึงรูปแบบความไม่เป็นทางการที่พบในข้อความภาษาไทยจากเว็บไซต์เครือข่ายทางสังคม และนำเสนอขั้นตอนวิธีการในการสร้างแบบจำลองตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรเพื่อเน้นคำหรือเพื่อตั้งใจให้อ่านด้วยการลากเสียงยาว ซึ่งข้อมูลนำเข้าเป็นข้อความจากเว็บไซต์เครือข่ายทางสังคม โดยนิยามให้ 1 ประกาศ (Post) เป็นตัวแทนของการพูด 1 ครั้ง (Utterance) ซึ่งในงานวิจัยนี้จะเรียกว่าชิ้นข้อความ (Text item) จากนั้นจะนำเสนอวิธีการทำให้เป็นบรรทัดฐาน โดยการแปลงกลับเป็นข้อความที่มีคำอ่านตรงกับที่ผู้เขียนต้องการสื่อ ซึ่งจากรายละเอียดที่กล่าวข้างต้น เราจำเป็นต้องนิยามขอบเขตของคำหรือส่วนข้อความ(Text segment) เพื่อนำไปจำแนกว่าคำนั้นๆมีลักษณะเด่นของการเล่นคำด้วยวิธีซ้ำตัวอักษรหรือไม่ จากนั้นทำการสกัดลักษณะเด่นเพื่อสร้างแบบจำลองการตรวจจับการเล่นคำ และสร้างกฎในการทำให้เป็นบรรทัดฐานต่อไป

3.1 เครื่องมือที่ใช้ในการวิจัย

1. เครื่องมือตัดคำภาษาไทย ซึ่งในที่นี้จะใช้ ทีเล็กซ์ ซึ่งเป็นเครื่องมือตัดคำที่ใช้แบบจำลองคอนดิชันแนลแรนดอมฟิลด์ เรียนรู้จากฐานข้อมูล เบสท์ 2010 (BEST2010) ซึ่งมีข้อความตัวอย่าง 5 ล้านคำ [15]

2. เครื่องมือทำเหมืองข้อมูล (Data mining tool) ซึ่งในงานวิจัยนี้ใช้โปรแกรมเวก้า(WEKA) เวอร์ชั่น 3.6 [18]

3. ระบบฐานข้อมูล Microsoft Access 2010

4. โปรแกรม Microsoft Excel 2010

5. เครื่องมือพัฒนาโปรแกรมภาษา C#, Microsoft Visual Studio 2010

3.2 รูปแบบความไม่เป็นทางการที่พบในข้อความภาษาไทยจากเว็บไซต์เครือข่ายทางสังคม

จากการวิเคราะห์ข้อที่ใช้สื่อสารในเว็บไซต์เครือข่ายทางสังคมของผู้ใช้งานที่ใช้ภาษาไทย จำนวน 8554 ชิ้นข้อความ พบว่ามีทั้งการใช้อักขระภาษาไทย อักขระภาษาอื่นๆโดยเฉพาะภาษาอังกฤษ ตัวเลข และสัญลักษณ์อื่นๆในการสื่อสาร อย่างไรก็ตามเมื่อพิจารณาเฉพาะอักขระภาษาไทย จะพบว่ามีโทเค็นมีความที่ไม่เป็นทางการถึง 7004 โทเค็น ซึ่งรูปแบบความไม่เป็นทางการที่เกิดจากการใช้อักขระภาษาไทยจะแบ่งได้เป็น 8 กลุ่มดังนี้

3.2.1 การเล่นคำด้วยการเปลี่ยนแปลงเสียงสระหรือพยัญชนะบางตัว เพื่อสร้างเสียงใหม่หรือเสียงที่ใกล้เคียงกับคำต้นฉบับ (Sound-transformation wordplay: ST-WP) โดยการเล่นคำลักษณะนี้ยังคงสะกดได้ถูกต้องตามหลักการประสมคำ ตัวอย่างเช่น สวดยวด (สุดยอด) ม่าย(ไม่) เปง(เป็น) จุงเบย(จังเลย)เป็นต้น

3.2.2 การเล่นคำด้วยวิธีซ้ำตัวอักษร (Repeated-letter wordplay: RPT-WP) เป็นการกดแป้นคีย์บอร์ดค้างเพื่อให้พื้นที่ข้อความยาวขึ้นอาจทำให้สระหรือตัวสะกดของพยางค์นั้นๆมีมากกว่า 1 ตัว ซึ่งผิดหลักการประสมอักษร ทำให้ระบบสังเคราะห์เสียงแปลงเป็นคำอ่านของพยางค์ต่อไป เช่น แล้ววววว อ่านว่า แล้ว-วะ-วะ- วอ

3.2.3 การใช้วรรณยุกต์ไม่ถูกต้อง (Miss tone: MTON) เช่น คำว่า อะ เป็น คำที่มีพื้นเสียงเป็นเสียงเอกอยู่แล้วนำไปผันวรรณยุกต์เอก เป็น อะ หรือการใช้วรรณยุกต์ตรีแทนเสียงตรี ซึ่งในกรณีที่เป็นการผันอักษรต่ำคำเป็นหรือเสียงยาว จะมีพื้นเสียงสามัญ ผันวรรณยุกต์เอกได้เสียงโท ผันวรรณยุกต์โทได้เสียงตรี ไม่มีการผันด้วยวรรณยุกต์ตรี เช่น ค๊า(ค้า) ย๊ากยาก(ย้ากยาก) เป็นต้น

3.2.4 การเขียนทับศัพท์ภาษาอังกฤษ (Transliterate: TLS) โดยใช้คำแตกต่างกันออกไปและไม่มีในพจนานุกรมทับศัพท์หรือข้อมูลชุดฝึกสอนของระบบประมวลผลทางภาษา เช่น facebook ถูกเขียนด้วยภาษาไทยในหลายรูปแบบเช่น เฟซ เฟซบุค เฟซบุ๊ค เป็นต้น

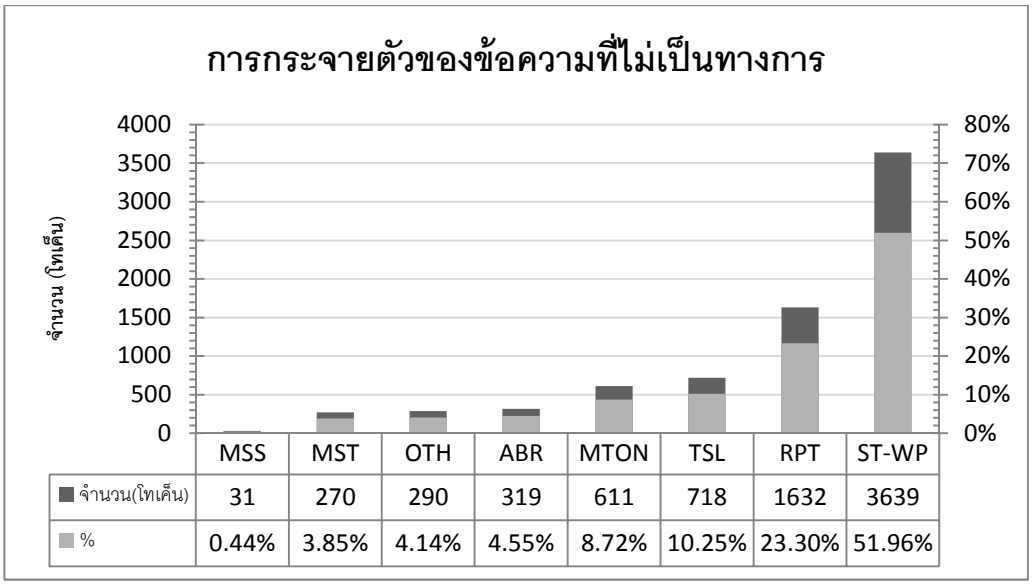
3.2.5 การใช้อักษรย่อที่กำหนดขึ้นเอง ไม่ตรงตามพจนานุกรม (Abbreviation: ABR) เช่น อจ. แทนคำว่าอาจารย์ เป็นต้น

3.2.6 การพิมพ์ผิด (Miss type: MST) ทำให้คำนั้นกลายเป็นคำที่ไม่สามารถอ่านออกเสียงได้ เช่น เด๋ยว, โ๙คดีๆ

3.2.7 การสะกดผิด ที่ไม่ทำให้เปลี่ยนแปลงเสียงอ่าน เช่น สัญญาน อนุญาติ

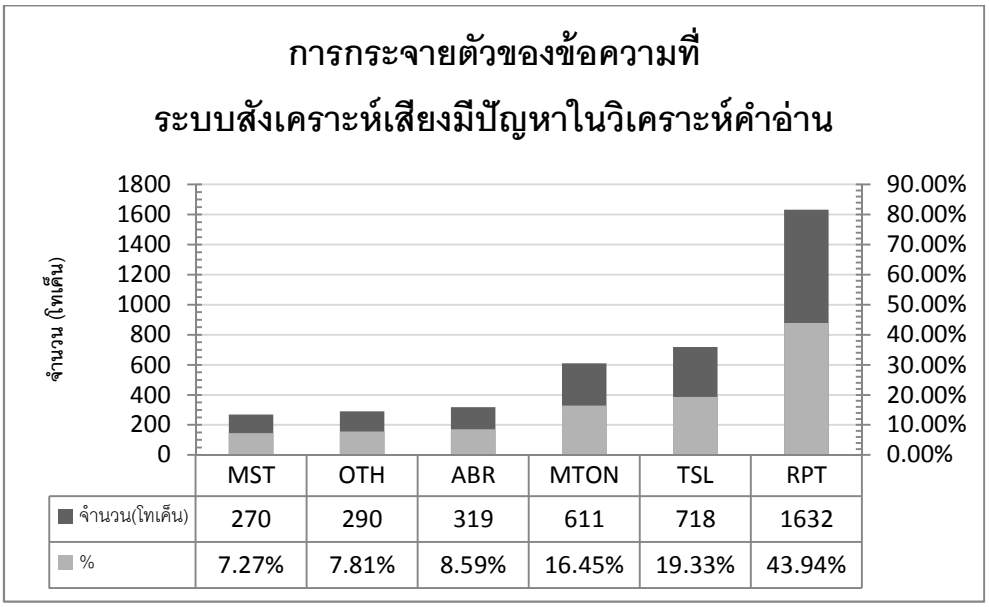
3.2.8 ความไม่เป็นทางการแบบอื่นๆ ได้แก่ ชื่อเฉพาะต่างๆ เช่น แอฟ, อิ่งอิ๊ง, เนียนไพร์ ซึ่งส่วนใหญ่เป็นการเขียนทับศัพท์ที่ไม่ทราบเสียงวรรณยุกต์

อย่างไรก็ตามในการเกิดคำหรือข้อความที่ไม่เป็นทางการครั้งหนึ่งๆอาจมีความไม่เป็นทางการเกิดขึ้นพร้อมกันหลายรูปแบบ เช่น งายยยย มีทั้งการเล่นคำด้วยการแปลงเสียงและซ้ำอักษร ซึ่งจากข้อมูลโทเค็นที่ไม่เป็นทางการจำนวน 7004 โทเค็นจะมีการกระจายตัวของความไม่เป็นทางการแบบต่างๆดังภาพ 3-1



ภาพที่ 3-1 การกระจายตัวของข้อความที่ไม่เป็นทางการประเภทต่างๆ

ซึ่งจากการวิเคราะห์ผลการอ่านด้วยระบบสังเคราะห์เสียงนั้น ข้อความที่ไม่เป็นทางการบางกลุ่มไม่มีผลกระทบต่อระบบสังเคราะห์เสียง ได้แก่ กลุ่มที่มีการแปลงเสียงแต่ยังคงสะกดด้วย และการสะกดคำผิดในกรณีที่ยังคงอ่านได้ ดังนั้นการกระจายตัวของข้อมูลที่ระบบสังเคราะห์เสียงมีปัญหาในการวิเคราะห์ข้อความ จะเหลือเพียง 3,714 โทเค็น ซึ่งมีการกระจายตัว ดังภาพ 3-2

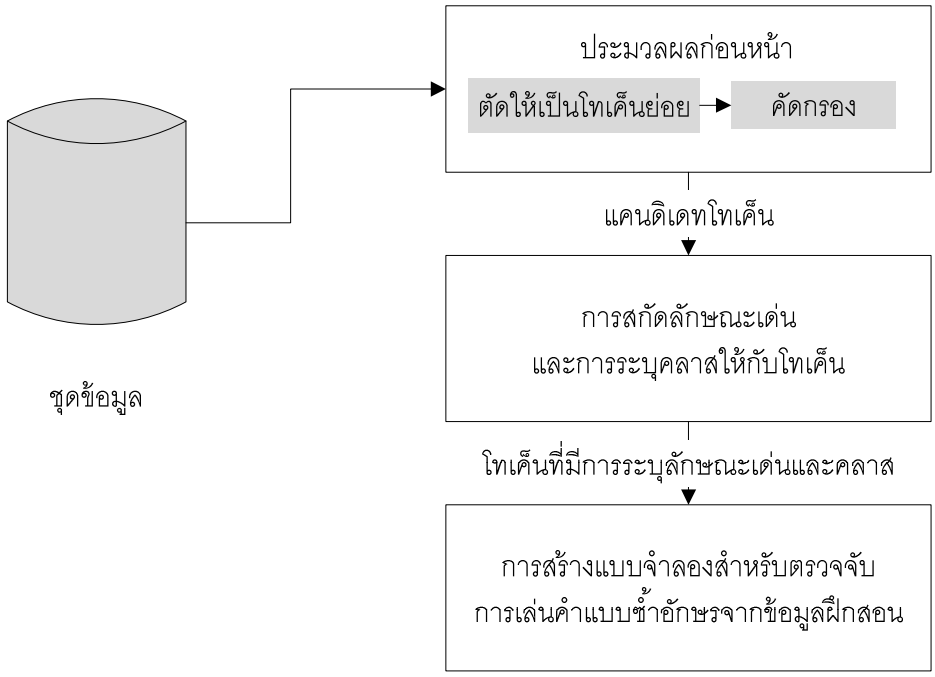


ภาพที่ 3-2 การกระจายตัวของข้อความที่ระบบสังเคราะห์เสียงมีปัญหาในการวิเคราะห์คำอ่าน

ในงานวิจัยนี้จึงเลือกที่จะนำเสนอแบบจำลองในการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษร เนื่องจากเป็นวิธีการเขียนแบบไม่เป็นทางการที่พบมากที่สุดในกรณีที่ระบบสังเคราะห์เสียงมีปัญหาในการวิเคราะห์เสียงอ่าน ซึ่งในงานวิจัยนี้จะทำการแปลงข้อความที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษรให้เป็นคำต้นฉบับที่ระบบสังเคราะห์เสียงสามารถอ่านออกเสียงได้ตรงกับความต้องการของผู้ใช้งาน

3.3 ขั้นตอนการสร้างระบบตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษร

ขั้นตอนที่ใช้การสร้างระบบตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรแบ่งเป็น 3 ส่วนย่อย ดังภาพ 3-3 ได้แก่



ภาพที่ 3-3 แผนภาพแสดงขั้นตอนการสร้างระบบตรวจจับการเล่นคำด้วยวิธีซ้ำอักษร

3.3.1 การประมวลผลก่อนหน้า (Pre-processing)

การประมวลผลก่อนหน้าเป็นขั้นตอนที่ระบบทำการเตรียมข้อความแบ่งให้อยู่ในระดับโทเค็นที่เหมาะสม ที่มีโอกาสเป็นการเล่นคำด้วยวิธีซ้ำตัวอักษร ระบบจะนำข้อมูลที่เก็บรวมรวมจากเว็บไซต์สังคมเครือข่ายมาประมวลผลเบื้องต้น ซึ่งประกอบด้วยขั้นตอนย่อย 2 ขั้นตอนดังนี้

1. การทำให้เป็นโทเค็นย่อย (Tokenization) แต่ละชิ้นข้อความจะถูกนำมาแบ่งย่อยเป็นโทเค็น ก่อนที่จะนำไปสกัดลักษณะเด่น ในงานวิจัยนี้จะนิยามระดับคำหนึ่งคำแทนหนึ่งโทเค็นที่จะถูกจำแนกกลุ่ม โดยใช้เครื่องมือตัดคำในการระบุขอบเขตของคำ ซึ่งเครื่องมือตัดคำที่ใช้ในที่นี้ได้แก่ ทีเล็กซ์ (TLEX : Thai Lexeme Analyser based on the Conditional Random Fields) เป็นเครื่องมือตัดคำที่สร้างโดยแบบจำลองคอนดิ

ชันแนลแรนดอมฟิลด์ ซึ่งเป็นวิธีที่มีประสิทธิภาพสูงสุดสำหรับการตัดคำภาษาไทยในปัจจุบัน [14,15] และเนื่องจากข้อความที่เป็นการเล่นคำนั้น ไม่ได้อยู่ในข้อมูลชุดฝึกฝนที่ใช้ในการสร้างเครื่องมือตัดคำ ทำให้ผลในการตัดคำด้วยทีเล็กซ์ ของข้อความที่เป็นการเล่นคำด้วยวิธีซ้ำตัวอักษร แสดงเป็น 2 รูปแบบดังนี้

- คำที่มีการเล่นคำแบบซ้ำอักษร 1 คำถูกตัดให้อยู่ในโทเค็นเดียวกัน เช่น แล้ว ถูกตัดเป็น |แล้ววววววว| ซึ่งระบบสังเคราะห์เสียงจะอ่านเป็น แล้ว วะ วะ วะ วะ วะ
- คำที่มีการเล่นคำแบบซ้ำอักษร 1 คำ ถูกตัดออกเป็นหลายส่วนตามหลังข้อความที่เป็น formal thai เช่น มากกกกกก -> |มาก|กก|กกก| ซึ่งแต่ละโทเคนจะเป็นส่วนหนึ่งของคำที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษร

2. การกรองให้เหลือเฉพาะโทเคนที่มีโอกาสเป็นการเล่นคำแบบซ้ำอักษร (Fast match filter)ขั้นตอนนี้ระบบจะใช้เรกูลาร์เอ็กซ์เพรสชั่นในการตรวจจับโทเค็นของคำที่เขียนด้วยอักษรไทยที่มีการซ้ำของอักษรอย่างน้อย 2 ตัวขึ้นไปเป็นคำที่มีโอกาสจะเป็นการเล่นคำด้วยวิธีซ้ำตัวอักษรซึ่งในที่นี้เราจะเรียกว่า แคนดิเดทโทเค็น ระบบจะพิจารณากรองให้เหลือเฉพาะโทเค็นที่มีการซ้ำของพยัญชนะ และการซ้ำของสระในบรรทัดเท่านั้น เนื่องการซ้ำของสระบนล่างหรือวรรณยุกต์ ไม่ได้เป็นการขยายพื้นที่ของข้อความ จึงไม่ได้แสดงการเน้นข้อความ และเก็บข้อมูลคำที่มีโอกาสเป็นการเล่นคำนั้นคู่กับประโยคตามตัวอย่างที่มีการตัดคำแบ่งเป็นโทเค็นย่อยแล้ว ตามตารางที่ 3-1 ซึ่งแคนดิเดทโทเค็นที่เป็นการเล่นคำจะแสดงเป็นตัวหนา

ตารางที่ 3-1 ตัวอย่างประโยคที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษรและแคนดิเดนโทเค็น

ผลการทำให้เป็นโทเค็นย่อย	แคนดิเดทโทเค็น
|@pang| | |ไป|เกาหลี|มา|จ้าา|	**จ้าา**
|แจ่มมมมมมมม| |๕๕๕| |ว่า| แล้ว|อยาก|ไป|ทะเล| |เด๋ว|เรา| จะ|เลี้ยง|ชาย|กลาง|ให้|อ้วนพี| | เร็วๆ| |นี้| |ปล.| |ขี้|โรค|มาก|กก|	**แจ่มมมมมมมม กก**
|เออ|...|เพลง|มัน|เก่า|มาก|กก| จน|เรา|ก็|ร้อง|ได้|แค่|...| |ครั้งนี้|	เออ **กก**

ผลการทำให้เป็นโทเค็นย่อย	แคนดิเดทโทเค็น
คง|ถูกใจ|	
|โถววววว| |ทำ|งาน|หนัก|เก็บ|เงิน| แต่งงาน|สินะ|ๆ |สุขสันต์|วัน| เกิด|น๊า|คน|สวยยยยยย|	**โถววววว** แต่งงาน **สวยยยยยย**
|Jub| |หลายจุ้ฟ|ให้|แล้ว|นะ| | ขอบคุณ|เพื่อน|ออม|คน|สวย| มาก|ๆ|เลย|น้าาา|	ออม **น้าาา**
|เทอม|สุดท้าย|แย้วววววววววว| | แต่|ผึ้ง|ยัง|คิด|อยู่|เลย|ว่า|ควร| จะ|ทำ|งานไง|ดี| |หรือ|ว่า|ควร| เรียน|ต่อเรย|ดี|ม้ายย| |พี่เชอร์รี| หน้า|สวย| |เด็ก| |ใส|เหมือน|เดิม| ไม่|เปลี่ยนเรย|น้าาา|	**แย้วววววววว มั้ยย น้าาา**

3.3.2 **การสกัดลักษณะเด่นและการระบุกลุ่มให้กับข้อมูล** (Feature Extraction & Class Annotation)

3.3.2.1 **การสกัดลักษณะเด่นของโทเค็น**

ลักษณะที่เด่นที่ใช้ในการสร้างแบบจำลองในการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรที่ใช้ในงานวิจัยนี้ได้แก่

1. ตัวอักษรที่พิมพ์ซ้ำ (Expanded letter : ExpL) ลักษณะเด่นนี้จะระบุว่าในโทเค็นที่สนใจตัวอักษรใดที่ถูกผู้ใช้งานพิมพ์ซ้ำมากกว่า 1 ตัว เช่น คำว่า ม่ายยยย จะมี ExpL เท่ากับ ‘ย’

2 .จำนวนอักษรที่ซ้ำ (Expansion count : ExpC) ลักษณะเด่นนี้จะนับจำนวนตัวอักษรที่ถูกพิมพ์ซ้ำในคำนั้น ซึ่งจะมีค่าตั้งแต่ 2 ขึ้นไป เช่น คำว่า ม่ายยยย จะมี ExpC เท่ากับ 4

3.ตัวอักษรเริ่มต้นของโทเค็น (Starting letter: SL) ลักษณะเด่นนี้จะดึงตัวอักษรเริ่มต้นของโทเค็นนั้น ตัวอย่างเช่น คำว่า มากกกก จะมี SL เท่ากับ ‘ม’

4. คลาสของตัวอักษรเริ่มต้น (Class of starting letter: CSL) ลักษณะเด่นในข้อนี้จะทำการตรวจสอบว่า ตัวอักษรเริ่มต้นของโทเค็นเป็นสระ พยัญชนะ หรืออักขระอื่นๆ ซึ่งในกรณีที่อักษร

เริ่มต้นโทเค็นเป็นพยัญชนะ ระบบจะทำการติดป้ายเป็น CSN (Consonant) และในกรณีที่ตัวอักษรเริ่มต้นโทเค็นเป็นสระ ระบบจะทำการติดป้ายเป็น VOW (Vowel) ซึ่งในกรณีที่ไม่ใช่ทั้งพยัญชนะและสระ ระบบจะทำการติดป้ายเป็น OTH (Other) ตัวอย่างเช่น คำว่า “แย้วววววววว” จะมีค่า CSL เท่ากับ VOW คำว่า “ม้ยย” จะมีค่า CSL เท่ากับ CSN เป็นต้น

5. ผลจากการค้นคำในพจนานุกรม (Dictionary-lookup result: Dict) พจนานุกรมที่ใช้ในงานวิจัยนี้คือ LEXiTRON ซึ่งผลจากการค้นคำในพจนานุกรม มีค่าที่เป็นไปได้ 3 กรณีคือ

1) กรณีที่เป็นโทเค็นของคำที่มีในพจนานุกรม จะมีค่าของลักษณะเด่น Dict เป็น “COR” ซึ่งหมายถึงว่าคำนั้นเป็นคำที่ถูกต้องเป็นทางการ (Correct Word)

2) ในกรณีที่โทเค็นนั้นไม่มีในพจนานุกรม ระบบจะลองทำการลบอักษรที่เขียนซ้ำออกให้เหลือเพียงตัวเดียว แล้วนำไปค้นในพจนานุกรมอีกครั้ง กรณีที่ลบอักษรซ้ำแล้วพบในพจนานุกรม จะมีค่าลักษระเด่น Dict เป็น “COP” ซึ่งหมายความว่าโทเค็นนั้นเป็นคำเป็นทางการที่ถูกซ้ำอักษร (Correct with processing)

3) ในกรณีที่ลบอักษรซ้ำแล้วยังไม่พบในพจนานุกรม ระบบจะระบุค่าลักษณะเด่นของโทเค็นนั้นเป็น “UNK” ซึ่งหมายความว่าโทเค็นนั้นเป็นโทเค็นของคำที่ไม่ทราบความหมาย (Unknown)

6. อักษรที่ตามหลังการพิมพ์ซ้ำ (Letter after expansion: AExp) เป็นลักษณะเด่นที่แสดงค่าเป็นตัวอักษรที่ตามหลังตัวอักษรที่ถูกพิมพ์ซ้ำ ในกรณีที่ตัวอักษรที่ถูกพิมพ์ซ้ำเป็นตัวสุดท้ายของโทเค็น ให้ค่าของลักษณะเด่นเป็น ‘E’ ที่หมายถึง อักษรหลังการพิมพ์ซ้ำเป็นค่าว่าง (Empty)

7. อักษรที่มาก่อนการพิมพ์ซ้ำ (Letter before expansion: BExp) เป็นลักษณะเด่นที่แสดงค่าเป็นตัวอักษรที่อยู่ก่อนหน้าตัวอักษรที่ถูกพิมพ์ซ้ำ ในกรณีที่ตัวอักษรที่ถูกพิมพ์ซ้ำเป็นแรกของโทเค็น ให้ค่าของลักษณะเด่นเป็น ‘E’ ที่หมายถึง อักษรก่อนการพิมพ์ซ้ำเป็นค่าว่าง (Empty)

8. อักษรตัวสุดท้ายของโทเค็น (End letter: End) เป็นลักษณะเด่นที่แสดงอักษรตัวสุดท้ายของโทเค็นนั้นๆ เช่น ออมมมี่ จะมีค่า End เป็น ◌่ (ไม้เอก)

9. อักษรสุดท้ายของโทเค็นก่อนหน้า (Last letter of preceding token : LP) แสดงอักษรตัวสุดท้ายของโทเค็นก่อนหน้า ซึ่งในกรณีที่โทเค็นนั้นเป็นโทเค็นแรกของประโยค จะมีค่า LP เท่ากับ ‘NONE’

10.อักษรสุดท้ายของโทเค็นก่อนหน้า เทียบกับอักษรตัวแรกของโทเค็นที่พิจารณา (Across token germination: G) ซึ่งในกรณีที่ อักษรตัวสุดท้ายของโทเค็นก่อนหน้าเหมือนกับอักษรตัวแรก

ของโทเค็นที่พิจารณา จะมีค่า G เท่ากับ 1 ในกรณีอื่นๆรวมถึงกรณีที่ไม่มีโทเค็นก่อนหน้า จะมีค่าเท่ากับ 0

จากลักษณะเด่นที่นำเสนอ แคนดิเดทโทเค็นทุกตัวจะถูกสกัดลักษณะเด่นเพื่อเป็นข้อมูลป้อนให้ระบบเพื่อสร้างแบบจำลองในการจำแนกประเภท ซึ่งตัวอย่างการสกัดลักษณะเด่น แสดงในตารางที่ 3-2

ตารางที่ 3-2 ตัวอย่างผลจากการสกัดลักษณะเด่นจากโทเค็น แสดงพร้อมกับโทเค็นก่อนหน้า

Prev Token	Token	ExpL	ExpC	SL	CSL	Dict	AExp	BExp	End	LP	G
มาก	กก	ก	2	ก	CSN	COR	E	E	ก	ก	1
ผม	ยาววววว	ว	4	ย	CSN	COP	E	า	ว	ม	0

3.3.2.2 **การระบุโทเค็นที่เป็นการเล่นคำ** (Positive token)

แคนดิเดทโทเคนจะถูกตัดสินโดยผู้เชี่ยวชาญ ซึ่งจะโทเค็นใดๆ จะถูกนับเป็นโทเค็นที่มีการเล่นคำก็ต่อเมื่อโทเค็นนั้นเป็นโทเค็นที่มีการซ้ำตัวอักษร หรือเป็นส่วนหนึ่งของการเล่นคำด้วยวิธีซ้ำตัวอักษรเพื่อขยายความยาวเพื่อผลกระทบอื่นๆ ซึ่งในกระบวนการทำให้เป็นบรรทัดฐาน โทเค็นที่เป็นส่วนหนึ่งของการเล่นคำในหน่วยเดียวกันจะถูกเชื่อมรวมให้เป็นโทเค็นเดียวกันก่อนการนำไปทำให้เป็นบรรทัดฐาน

3.2.3 **สร้างแบบจำลองในการตรวจจับโทเค็นที่เป็นการเล่นคำแบบซ้ำอักษร**

ขั้นตอนนี้เป็นขั้นตอนการสร้างแบบจำลองแบบต้นไม้ตัดสินใจ ที่ใช้ในการทำนายว่าแต่ละโทเค็นเป็นการเล่นคำด้วยวิธีซ้ำตัวอักษรหรือไม่ ซึ่งในงานวิจัยนี้จะใช้อัลกอริทึม C4.5 หรือตัวจำแนกประเภทชื่อ j48 ซึ่งเป็นประเภทหนึ่งของต้นไม้ตัดสินใจจากเวก้า [18] ซึ่งเป็นเครื่องมือในการทำเหมืองข้อมูล ซึ่งประกอบด้วยขั้นตอนต่อไปนี้

3.2.3.1 **แปลงข้อมูลชุดฝึกฝนที่มีการสกัดลักษณะเด่นเป็นรูปแบบที่ใช้สำหรับโปรแกรมเวก้า**

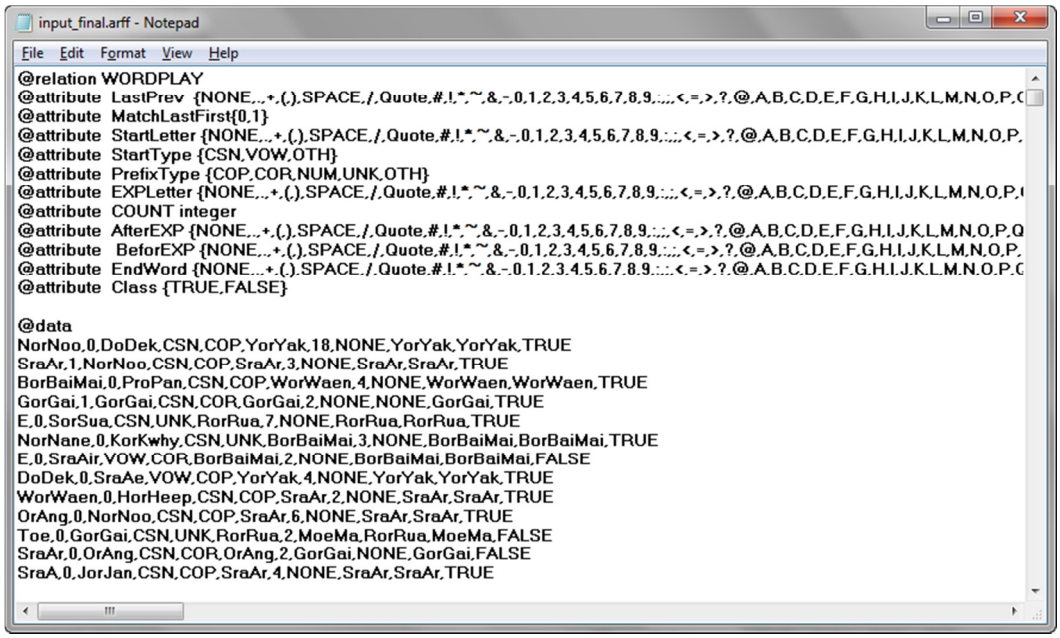
รูปแบบที่ใช้สำหรับการเรียนรู้ด้วยโปรแกรมเวก้าจะมีชื่อไฟล์เป็น *.arff โดยโครงสร้างการนำเข้าข้อความจะต้องคำอธิบายส่วนหัวและส่วนข้อมูลดังนี้

- เริ่มต้นด้วย @relation WORDPLAY แสดงชื่อตารางหรือชุดข้อมูที่ใช้ในการสร้างแบบจำลองดังภาพที่ 3-4

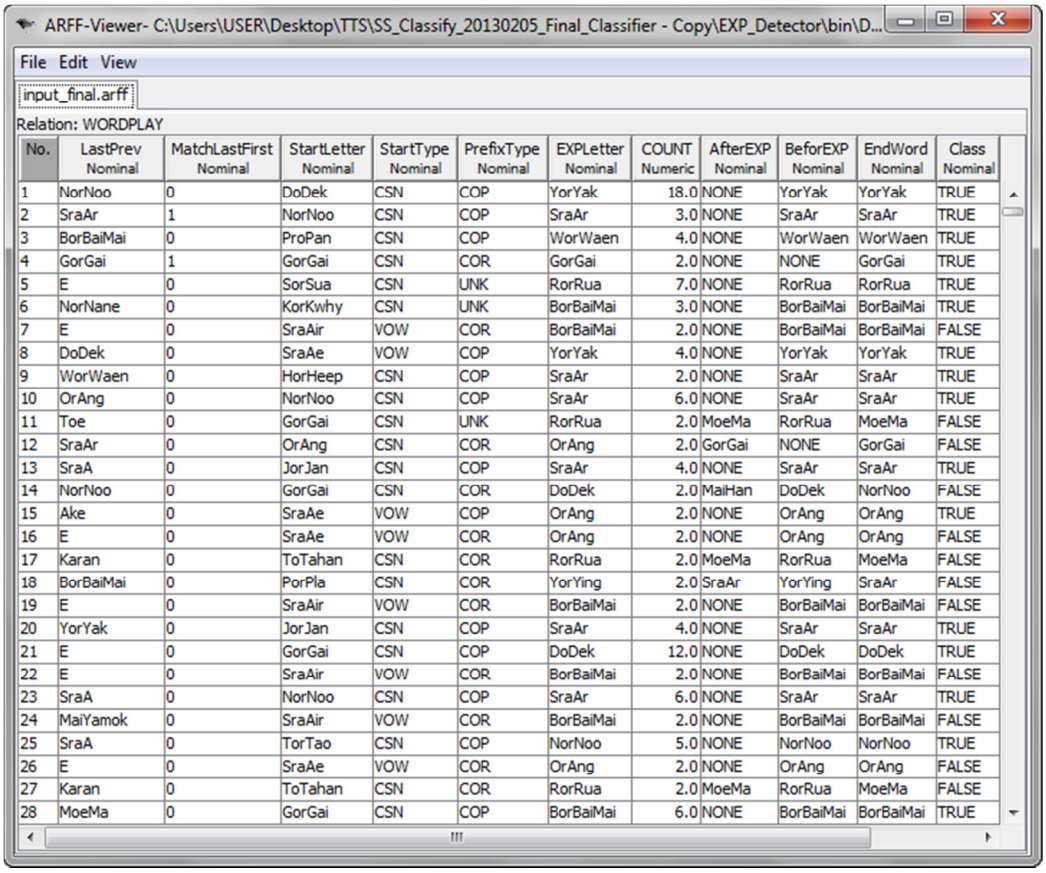
- ตามด้วย @attribute ATTRIBUTENAME แสดงรายการลักษณะเด่นที่สกัดจากโทเค็นซึ่งเมื่อใช้เป็นข้อมูลนำเข้าจะแสดงเป็นคอลัมน์ของข้อมูล หลังชื่อ Attribute จะตามด้วยชนิดของ Attribute ซึ่งในกรณีที่คุณลักษณะเป็นรูปแบบ Nominal จะแสดงค่าที่เป็นไปได้ทั้งหมดในวงเล็บ และในกรณีที่เป็นจำนวนเต็ม จะประกาศว่าเป็น Integer หลังชื่อ Attribute โดยที่ Attribute สุดท้ายจะเป็นคลาสของโทเค็น ระบุว่าเป็นการเล่นคำด้วยวิธีซ้ำตัวอักษรหรือไม่ ดังภาพที่ 3-4

- ส่วนของข้อมูลจะเริ่มต้นด้วย @data ซึ่งแต่ละรายการจะแสดงโดยมีเครื่องหมายจุลภาคคั่นโดยใน 1 รายการจะมีต้องมีจำนวนคอลัมน์เท่ากับจำนวน Attribute ดังภาพที่ 3-4

ภาพที่ 3-4 ตัวอย่างรูปแบบไฟล์ .arff ในการสร้างแบบจำลองตรวจจับการเล่นคำ



ภาพที่ 3-5 ตัวอย่างแคนดิเดทโทเค็นชุดฝึกฝนเมื่อป้อนเข้าสู่โปรแกรมเวก้า



ARFF-Viewer- C:\Users\USER\Desktop\TTS\SS_Classify_20130205_Final_Classifier - Copy\EXP_Detector\bin\D...

File Edit View

input_final.arff

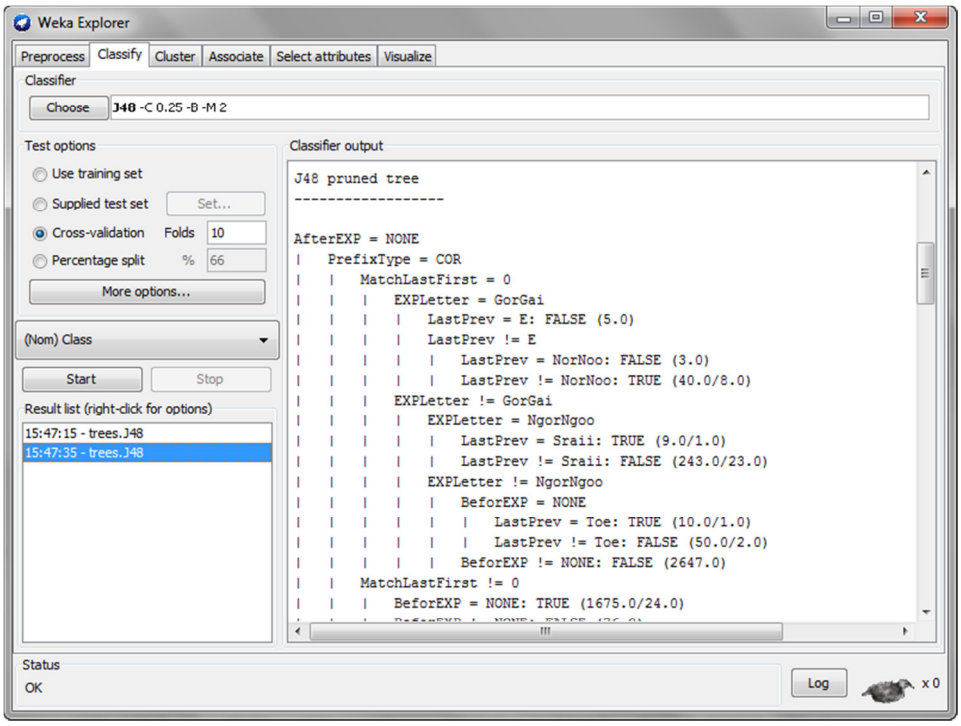
Relation: WORDPLAY

No.	LastPrev Nominal	MatchLastFirst Nominal	StartLetter Nominal	StartType Nominal	PrefixType Nominal	EXPLetter Nominal	COUNT Numeric	AfterEXP Nominal	BeforEXP Nominal	EndWord Nominal	Class Nominal
1	NorNoo	0	DoDek	CSN	COP	YorYak	18.0	NONE	YorYak	YorYak	TRUE
2	SraAr	1	NorNoo	CSN	COP	SraAr	3.0	NONE	SraAr	SraAr	TRUE
3	BorBaiMai	0	ProPan	CSN	COP	WorWaen	4.0	NONE	WorWaen	WorWaen	TRUE
4	GorGai	1	GorGai	CSN	COR	GorGai	2.0	NONE	NONE	GorGai	TRUE
5	E	0	SorSua	CSN	UNK	RorRua	7.0	NONE	RorRua	RorRua	TRUE
6	NorNane	0	KorKwhy	CSN	UNK	BorBaiMai	3.0	NONE	BorBaiMai	BorBaiMai	TRUE
7	E	0	SraAir	VOW	COR	BorBaiMai	2.0	NONE	BorBaiMai	BorBaiMai	FALSE
8	DoDek	0	SraAe	VOW	COP	YorYak	4.0	NONE	YorYak	YorYak	TRUE
9	WorWaen	0	HorHeep	CSN	COP	SraAr	2.0	NONE	SraAr	SraAr	TRUE
10	OrAng	0	NorNoo	CSN	COP	SraAr	6.0	NONE	SraAr	SraAr	TRUE
11	Toe	0	GorGai	CSN	UNK	RorRua	2.0	MoeMa	RorRua	MoeMa	FALSE
12	SraAr	0	OrAng	CSN	COR	OrAng	2.0	GorGai	NONE	GorGai	FALSE
13	SraA	0	JorJan	CSN	COP	SraAr	4.0	NONE	SraAr	SraAr	TRUE
14	NorNoo	0	GorGai	CSN	COR	DoDek	2.0	MaiHan	DoDek	NorNoo	FALSE
15	Ake	0	SraAe	VOW	COP	OrAng	2.0	NONE	OrAng	OrAng	TRUE
16	E	0	SraAe	VOW	COR	OrAng	2.0	NONE	OrAng	OrAng	FALSE
17	Karan	0	ToTahan	CSN	COR	RorRua	2.0	MoeMa	RorRua	MoeMa	FALSE
18	BorBaiMai	0	PorPla	CSN	COR	YorYing	2.0	SraAr	YorYing	SraAr	FALSE
19	E	0	SraAir	VOW	COR	BorBaiMai	2.0	NONE	BorBaiMai	BorBaiMai	FALSE
20	YorYak	0	JorJan	CSN	COP	SraAr	4.0	NONE	SraAr	SraAr	TRUE
21	E	0	GorGai	CSN	COP	DoDek	12.0	NONE	DoDek	DoDek	TRUE
22	E	0	SraAir	VOW	COR	BorBaiMai	2.0	NONE	BorBaiMai	BorBaiMai	FALSE
23	SraA	0	NorNoo	CSN	COP	SraAr	6.0	NONE	SraAr	SraAr	TRUE
24	MaiYamok	0	SraAir	VOW	COR	BorBaiMai	2.0	NONE	BorBaiMai	BorBaiMai	FALSE
25	SraA	0	TorTao	CSN	COP	NorNoo	5.0	NONE	NorNoo	NorNoo	TRUE
26	E	0	SraAe	VOW	COR	OrAng	2.0	NONE	OrAng	OrAng	FALSE
27	Karan	0	ToTahan	CSN	COR	RorRua	2.0	MoeMa	RorRua	MoeMa	FALSE
28	MoeMa	0	GorGai	CSN	COP	BorBaiMai	6.0	NONE	BorBaiMai	BorBaiMai	TRUE

3.3.3.1 เลือกใช้อัลกอริทึมและการสร้างแบบจำลอง

ในที่นี้จะเลือกใช้วิธีการเรียนรู้แบบต้นไม้ตัดสินใจ ประเภท C4.5 ซึ่งเป็นฟังก์ชั่น j48 ในโปรแกรมเวก้า ดังภาพที่ 3-6 จากนั้นจะนำเงื่อนไขจากแบบจำลองที่ได้มาสร้างระบบตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษร

ภาพที่ 3-6 การเลือกอัลกอริทึมในการสร้างแบบจำลองและผลจากข้อมูลชุดฝึกฝน



3.4 สร้างระบบแปลงข้อความที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษรให้เป็นบรรทัดฐาน

3.4.1 แบบจำลองของวิธีการจัดการกับโทเค็นเพื่อแปลงให้เป็นบรรทัดฐาน

ขั้นตอนในการสร้างระบบแปลงข้อความให้เป็นบรรทัดฐานประกอบด้วยการนำข้อความที่ถูกตรวจจับว่าเป็นการเล่นคำด้วยวิธีซ้ำตัวอักษรมาทำการติดป้ายว่าการจัดการกับโทเค็นดังกล่าวต้องดำเนินการอย่างไร จากนั้นนำกฎหรือแบบจำลองที่ได้ มาสร้างระบบการแปลงข้อความให้เป็นบรรทัดฐานซึ่งค่าที่จะทำการติดป้ายให้โทเค็นจะมีดังต่อไปนี้

1. ให้เชื่อมโทเค็นนั้นๆกับโทเค็นก่อนหน้าก่อนการทำให้เป็นบรรทัดฐาน (Merge-Replace)

2. ให้แทนที่ตัวอักษรซ้ำด้วยตัวอักษร 1 ตัว (ReplaceWith1)

3.ให้แทนที่ตัวอักษาซ้ำด้วยตัวอักษร 2 ตัว (ReplaceWith2)

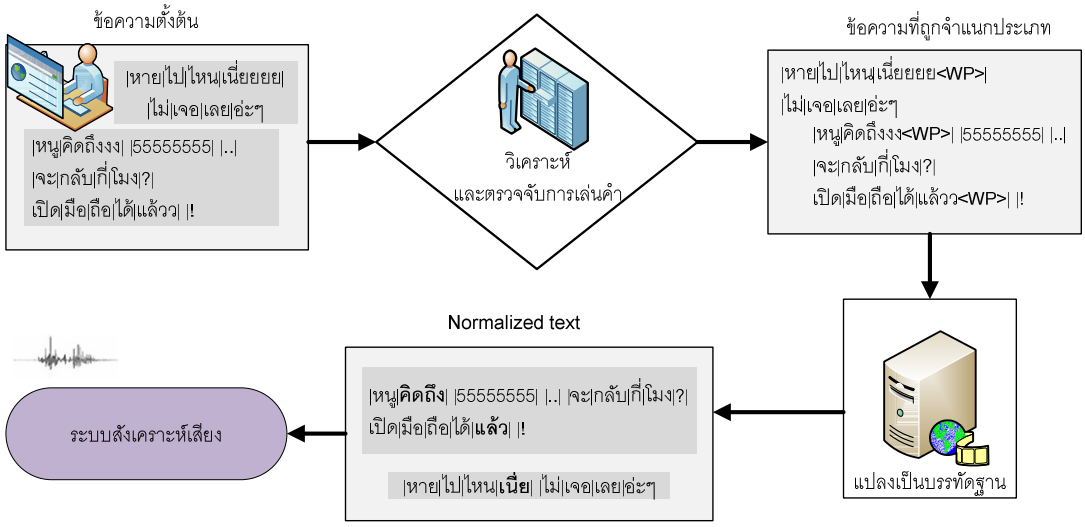
4.ให้ลบตัวอักษรซ้ำทั้งหมดออกไป (RemoveAll)

5.ไม่ทำการแก้ไขใดๆ (NoProcess)

เมื่อทำการติดป้ายให้กับข้อมูลชุดฝึกฝนจนครบแล้วข้อมูลจะถูกแปลงเป็นรูปแบบ .arff และป้อนเป็นข้อมูลอินพุตเพื่อสร้างต้นไม้ตัดสินใจด้วยขั้นตอนวิธีเดียวกับที่สร้างแบบจำลองที่ใช้สำหรับตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษร

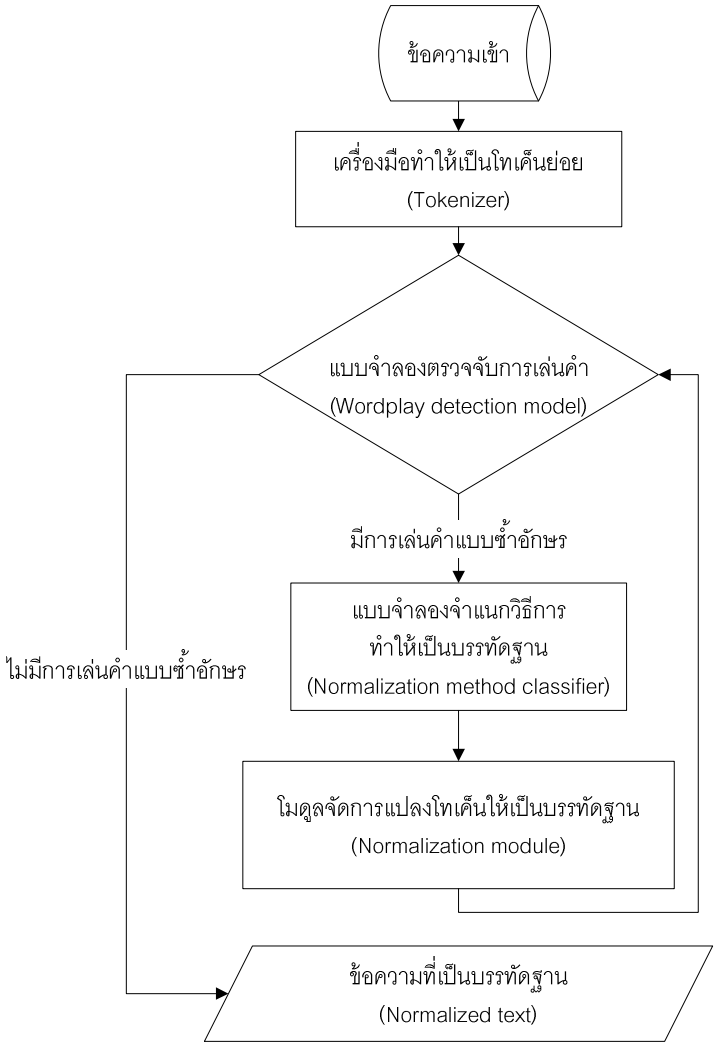
3.4.2 ระบบแปลงข้อความให้เป็นบรรทัดฐาน

ระบบแปลงข้อความให้เป็นบรรทัดฐานจะมีหน้าที่รับข้อความตั้งต้นจากเว็บไซต์เครือข่ายทางสังคม และนำไปตรวจสอบเพื่อวิเคราะห์และจำแนกประเภทการเล่นคำ จากนั้นจึงแปลงให้เป็นบรรทัดฐานเพื่อส่งต่อเป็นข้อมูลขาเข้าของระบบสังเคราะห์เสียง ซึ่งมีโครงสร้างของระบบดังภาพ 3-7



ภาพที่ 3-7 โครงสร้างระบบแปลงข้อความให้เป็นบรรทัดฐาน

ซึ่งการทำงานของระบบจะประกอบด้วยขั้นตอนต่างๆ ดังภาพ 3-8 ดังนี้



ภาพที่ 3-8 แผนภาพขั้นตอนการทำงานของระบบการแปลงข้อความให้เป็นบรรทัดฐาน

1.นำเข้าชิ้นข้อความและใช้ *เครื่องมือในการทำให้เป็นโทเค็นย่อย* ซึ่งในที่นี้คือเครื่องมือตัดคำภาษาไทย ทำการตัดชิ้นข้อความให้เป็นหน่วยของคำ ซึ่งจะใช้โทเค็นที่เป็นตัวแทนของคำในการตรวจจับการเล่นคำด้วยแบบจำลองสำหรับตรวจจับการเล่นคำ

2.ทำการสกัดลักษณะเด่นและใช้แบบจำลองตรวจจับการเล่นคำในการจำแนกแคนดิเดทโทเค็นว่าเป็นการเล่นคำแบบซ้ำอักษรหรือไม่

- กรณีที่โทเค็นนั้นเป็นการเล่นคำให้ส่งต่อไปยังแบบจำลองในการจำแนกวิธีการแปลงให้เป็นมาตรฐาน
- กรณีที่โทเค็นนั้นไม่เป็นการเล่นคำไม่ต้องทำการแปลง ถือเป็นโทเค็นที่เป็นบรรทัดฐานแล้ว

3.นำโทเค็นที่มีการเล่นคำพร้อมลักษณะเด่นป้อนสู่แบบจำลองการจำแนกวิธีการในการทำให้เป็นบรรทัดฐานเพื่อติดป้ายว่าจะให้ระบบใช้วิธีการใดในการทำให้เป็นบรรทัดฐาน

4.ป้อนโทเค็นที่มีการติดป้ายวิธีในการทำให้เป็นบรรทัดฐานไปยังโมดูลแปลงข้อความเพื่อดำเนินการแปลงให้เป็นบรรทัดฐานด้วยวิธีที่เหมาะสม จากนั้นส่งไปยังแบบจำลองตรวจจับการเล่นคำอีกครั้ง จนกว่าจะถูกแปลงเป็นบรรทัดฐานที่เหมาะสม

บทที่ 4

การทดลอง การเตรียมการทดลอง และวิธีการวัดผลการทดลอง

ในบทนี้จะกล่าวถึงการเตรียมข้อมูลสำหรับทำการทดลองตามขั้นตอนวิธีที่นำเสนอ และวิธีการวัดผลของการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษร รวมไปถึงการวัดผลของการทำให้เป็นบรรทัดฐาน โดยจะทำการทดสอบกับชุดทดสอบซึ่งเป็นข้อความจากเว็บไซต์สังคมออนไลน์ ซึ่งในบทนี้จะประกอบด้วยเนื้อหาดังต่อไปนี้ 1. ข้อมูลที่ใช้ทำการทดลอง 2. ระบบเส้นเชื่อมฐาน 3.การวัดผลของการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรจากลักษณะเด่นที่นำเสนอ 4.การวัดผลของการทำข้อความให้เป็นมาตรฐานเทียบจากการตัดสินของผู้เชี่ยวชาญ

4.1 ข้อมูลที่ใช้ในการทดลอง

ชิ้นข้อความที่ใช้ในงานนี้เป็นข้อความจำนวน 150,626 ประกาศ จากเว็บไซต์เครือข่ายทางสังคม [19] ผู้พูดหรือผู้ประกาศข้อความเป็นผู้ใช้งานจำนวน 2,080 คนซึ่งใช้ภาษาไทยเป็นหลักและเป็นเจ้าของภาษา ชิ้นข้อความทั้งหมดจะถูกแบ่งออกเป็นสองชุดคือชุดข้อมูลฝึกสอน (training data) และชุดข้อมูลทดสอบ (test data) ชุดข้อมูลฝึกสอนประกอบด้วย 102,585 ชิ้นข้อความใช้ในการประมวลผลและสร้างต้นไม้ตัดสินใจที่ใช้ในการจำแนกประเภทข้อมูล ส่วนชุดข้อมูลทดสอบประกอบด้วย 48,040 ชิ้นข้อความ ซึ่งจะใช้ในการวัดผลการจำแนกประเภท ตารางที่ 4-1 แสดงปริมาณและการกระจายตัวของข้อมูลในชุดฝึกสอนและชุดทดสอบ ตารางที่ 4-2 และตารางที่ 4-3 แสดงการกระจายตัวของแคนดิเดทโทเค็นที่มีการซ้ำอักษรเพียงสองตัวและแคนดิเดทโทเคนที่มีการซ้ำของตัวอักษรตั้งแต่ 3 ตัวขึ้นไปตามลำดับ ซึ่งจะแสดงเป็น 2 ตารางเพื่อให้ผู้อ่านสามารถเปรียบเทียบการกระจายตัวของข้อมูลทั้งสองชุด อย่างไรก็ตามการกระจายตัวของข้อมูลชุดทดสอบไม่ได้ถูกนำมาใช้ในการออกแบบการทดลอง ซึ่งข้อมูลชุดทดสอบจะถูกสงวนไว้สำหรับการวัดผลแบบจำลองหรือความรู้เชิงสถิติจะได้มาจากชุดฝึกสอนเท่านั้น

ตารางที่ 4-1 การกระจายตัวเชิงปริมาณของข้อมูล

จำนวนของ	ข้อมูลชุดฝึกสอน	ข้อมูลชุดทดสอบ
ชิ้นข้อความ (ประกาศ)	102,586	48,040
โทเค็นทั้งหมด (โทเค็น)	1,450,228	607,363
แคนดิเดทโทเค็น (โทเค็น)	27,486	12,551
โทเค็นที่เป็นการเล่นคำแบบซ้ำอักษร (โทเค็น)	16,435	7,776

ตารางที่ 4-2 การกระจายตัวของโทเค็นที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษร 2 ตัว

จำนวนของ	ข้อมูลชุดฝึกสอน	ข้อมูลชุดทดสอบ
แคนดิเดทโทเค็นทั้งหมดที่มีการซ้ำอักษร 2 ตัว(โทเค็น)	16,115	7,172
โทเค็นที่เป็นการเล่นคำแบบซ้ำอักษร 2 ตัว (โทเค็น)	5,065	2,403
โทเค็นที่มีการซ้ำอักษร 2 ตัวแต่ไม่มีการเล่นคำ (โทเค็น)	11,050	4,769

ตารางที่ 4-3 การกระจายตัวของโทเค็นที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษร 3 ตัวขึ้นไป

จำนวนของ	ข้อมูลชุดฝึกสอน	ข้อมูลชุดทดสอบ
แคนดิเดทโทเค็นทั้งหมดที่มีการซ้ำอักษร 3 ตัวขึ้นไป (โทเค็น)	11,371	5,379

จำนวนของ	ข้อมูลชุดฝึกสอน	ข้อมูลชุดทดสอบ
โทเค็นที่เป็นการเล่นคำแบบซ้ำอักษร 3 ตัวขึ้นไป (โทเค็น)	11,370	5,373
โทเค็นที่มีการซ้ำอักษร 3 ตัวขึ้นไปแต่ไม่มีการเล่นคำ (โทเค็น)	1	6

จากตารางที่ 4-2 และตารางที่ 4-3 จะพบว่าแคนดิเดทโทเค็นที่มีการซ้ำอักษรตั้งแต่ 3 ตัวขึ้นไปส่วนใหญ่จะเป็นการเล่นคำ ซึ่งผลจากการวิเคราะห์ข้อมูลดังกล่าวจะนำไปใช้เป็นส่วนหนึ่งของการทดลองในส่วนถัดไป

4.2 **ระบบเส้นเชื่อมฐาน** (Baseline Systems)

ในส่วนนี้จะนำเสนอ ระบบเส้นเชื่อมฐานที่จะนำมาใช้สร้างแบบจำลองการตรวจจับข้อความที่เป็นการเล่นคำด้วยวิธีซ้ำตัวอักษร เพื่อวัดประสิทธิภาพเปรียบเทียบกับขั้นตอนวิธีที่นำเสนอในงานวิจัยนี้ ซึ่งวิธีเส้นเชื่อมฐานที่นำมาวัดผลแบ่งออกเป็น 3 วิธีใหญ่ๆ ดังนี้

1. เส้นเชื่อมฐานวิธีที่ 1 : ตรวจจับการเล่นคำโดยอ้างอิงจากพจนานุกรม

เส้นเชื่อมฐานวิธีแรกที่จะนำมาใช้ในการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรจะขึ้นกับค่าลักษณะเด่นจากการค้นคำในพจนานุกรม หรือลักษณะเด่นที่มีชื่อว่า Dict ซึ่งในกรณีที่ ค่า Dict ของโทเค็นนั้น เท่ากับ “COR” หมายถึงเป็นข้อความที่ค้นเจอในพจนานุกรม (ในที่นี้หมายถึง Lexitron) ระบบนี้จะจำแนกโทเค็นนั้นๆเป็นคำปกติที่ไม่ใช่การเล่นคำ ในขณะที่ถ้าโทเค็นนั้นมีค่า Dict เป็น COP หมายถึงเมื่อตัดตัวอักษรที่ซ้ำออกแล้ว ค้นเจอในพจนานุกรม แสดงว่าโทเค็นนั้นมาจากการนำคำปกติมาซ้ำอักษร ระบบจะจำแนกโทเค็นนั้นเป็นคำที่มีการเล่นคำด้วยวิธีซ้ำอักษรทันที ส่วนโทเค็นที่มีค่า Dict เป็น “UNK” ระบบนี้จะแบ่งเป็นสองกรณีย่อย คือกรณีที่ 1-1 ระบบจำแนกโทเค็น “UNK” ให้เป็นโทเค็นที่มีการเล่นคำ และกรณีที่ 1-2 ระบบจะจำแนกโทเค็น “UNK” ให้เป็นโทเค็นที่ไม่ได้เป็นการเล่นคำ

2. เส้นเชื่อมฐานวิธีที่ 2 : ตรวจจับการเล่นคำด้วยกฎการซ้ำอักษร 4 ตัวร่วมกับการอ้างอิงจากพจนานุกรม

ระบบนี้จะใช้กฎที่อ้างอิงจากงานวิจัยของ Hemalatha [10] แคนดิเดทโทเค็นที่มีการซ้ำของอักษรมากกว่า 3 ตัวจะถูกพิจารณาเป็นโทเค็นที่มีการเล่นคำ ดังนั้นระบบนี้จะตรวจจับแคนดิเดทโทเค็นที่มีการซ้ำของตัวอักษร 4 ตัวขึ้นไป (ค่าลักษณะเด่น ExpC ตั้งแต่ 4 ขึ้นไป) และจำแนกเป็นโทเค็นที่มีการเล่นคำ ส่วนแคนดิเดนโทเค็นที่มีการซ้ำอักษร 2-3 ตัว จะแบ่งเป็น 2 กรณีตาม กรณีที่ 1 คือ ให้โทเค็นที่มี Dict เป็น “COR” ถูกจำแนกเป็นคำปกติ โทเค็นที่เป็น “COP” ถูกจำแนกเป็นโทเค็นที่มีการเล่นคำ และโทเค็นที่เป็น “UNK” จะถูกแบ่งเป็น 2 กรณีเหมือนกับวิธีที่ 1 คือกรณีที่ 2-1 ระบบจำแนกโทเค็น “UNK” ให้เป็นโทเค็นที่มีการเล่นคำ และกรณีที่ 2-2 ระบบจะจำแนกโทเค็น “UNK” ให้เป็นโทเค็นที่ไม่ได้เป็นการเล่นคำ

3. เส้นเชื่อมฐานวิธีที่ 3 : ตรวจจับการเล่นคำด้วยกฎการซ้ำอักษร 3 ตัวร่วมกับการอ้างอิงจากพจนานุกรม

ระบบนี้จะคล้ายกับวิธีที่สอง แต่จากการพิจารณาข้อมูลชุดฝึกฝนทำให้พบว่าโทเค็นที่มีการซ้ำอักษรตั้งแต่ 3 ตัวขึ้นไปจะถูกจำแนกเป็นการเล่นคำมากกว่า 99% ระบบนี้จึงใช้กฎคล้ายกับวิธีที่ 2 แต่เปลี่ยนเป็นแคนดิเดทโทเค็นที่มีการซ้ำของอักษรตั้งแต่ 3 ตัวขึ้นไป (ค่าลักษณะเด่น ExpC มีค่าตั้งแต่ 3 ขึ้นไป) จะถูกพิจารณาและจำแนกเป็นโทเค็นที่มีการเล่นคำ ส่วนแคนดิเดนโทเค็นที่มีการซ้ำอักษร 2 ตัว จะแบ่งเป็น 2 กรณีตาม กรณีที่ 1 คือ ให้โทเค็นที่มี Dict เป็น “COR” ถูกจำแนกเป็นคำปกติ โทเค็นที่เป็น “COP” ถูกจำแนกเป็นโทเค็นที่มีการเล่นคำ และโทเค็นที่เป็น “UNK” จะถูกแบ่งเป็น 2 กรณีเหมือนกับวิธีที่ 1 และ 2 คือกรณีที่ 3-1 ระบบจำแนกโทเค็น “UNK” ให้เป็นโทเค็นที่มีการเล่นคำ และกรณีที่ 3-2 ระบบจะจำแนกโทเค็น “UNK” ให้เป็นโทเค็นที่ไม่ได้เป็นการเล่นคำ

4.3 การวัดผลวิธีการสร้างตัวจำแนกประเภทโดยการใช้ลักษณะเด่นที่นำเสนอ

จากลักษณะเด่นที่นำเสนอในบทที่แล้ว จะถูกนำมาใช้ในการทดลองในกรอบการจำแนกประเภทที่แตกต่างกันในไปในรายละเอียดดังนี้

1. การทดลองที่ 1 การจำแนกประเภทโทเค็นโดยใช้แบบจำลองการตรวจจับที่สร้างสำหรับโทเค็นที่ไม่มีในพจนานุกรมเท่านั้น

การทดลองนี้จะจัดการกับโทเค็นที่ไม่พบในพจนานุกรม โดยสร้างแบบจำลองต้นไม้ตัดสินใจให้เฉพาะโทเค็นที่มีลักษณะเด่น Dict เป็น “UNK” แทนการตัดสินให้อยู่ในคลาสใดคลาสหนึ่ง เช่นเดียวกับระบบเส้นเชื่อมฐาน วิธีนี้จะสกัดลักษณะเด่นอื่นๆจากแคนดิเดทโทเค็นเฉพาะที่มีค่า Dict เท่ากับ “UNK” จากข้อมูลชุดฝึกสอน แล้วนำไปสร้างแบบจำลองต้นไม้ตัดสินใจ สำหรับโทเค็นที่ไม่สามารถค้นเจอในพจนานุกรม ในขณะที่ โทเค็นที่สามารถพบในพจนานุกรม ทั้งกรณีที่เป็น “COR” และ “COP” จะดำเนินการเหมือนกับระบบเส้นเชื่อมฐานคือระบุว่า “COR” ไม่ใช่การเล่นคำ และ “COP” เป็นการเล่นคำ

2. การทดลองที่ 2 การจำแนกโทเค็นทั้งหมดจากแบบจำลองที่สร้างจากชุดข้อมูลฝึกฝนบนการสกัดลักษณะเด่นทั้งหมด

การทดลองนี้จะสร้างแบบจำลองต้นไม้ตัดสินใจสำหรับตรวจจับโทเค็นที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษรด้วยการเรียนรู้จากชุดข้อมูลฝึกสอน โดยใช้ลักษณะเด่นที่นำเสนอทั้งหมดรวมถึงความสามารถในการพบในพจนานุกรมด้วย ซึ่งวิธีนี้จะใช้แบบจำลองการตรวจจับการเล่นคำที่ได้จากข้อมูลชุดฝึกสอนดังกล่าวในการทำนายคลาสของทุกๆแคนดิเดทโทเค็นในชุดทดสอบว่าเป็นการเล่นคำด้วยวิธีซ้ำตัวอักษรหรือไม่

3. การทดลองที่ 3 การจำแนกโทเค็นโดยใช้แบบจำลองที่ได้จากการทดลองที่ 2 ร่วมกับการตรวจจับก่อนหน้า (Pre-screening)

จากการพิจารณาข้อมูลชุดฝึกฝนแล้วพบว่ามากกว่า 99% ของโทเค็นที่มีการซ้ำอักษร 3 ตัวขึ้นไปถูกจำแนกโดยผู้เชี่ยวชาญว่าเป็นการเล่นคำแบบซ้ำอักษร การทดลองนี้จึงใช้กฎการกรองโทเค็นที่มีการซ้ำตัวอักษรตั้งแต่ 3 ตัวขึ้นไปเพื่อตัดสินทันทีว่าเป็นการเล่นคำด้วยวิธีซ้ำตัวอักษร ในกรณีที่โทเค็นมีการซ้ำอักษรเพียง 2 ตัวอักษรจะใช้แบบจำลองที่ได้จากข้อ 2 ในการตัดสินว่าเป็นการเล่นคำแบบซ้ำอักษรหรือไม่

จากระบบเส้นเชื่อมฐานและการทดลองทั้ง 3 กรณี ในบทต่อไปจะแสดงผลการทดลอง โดยใช้ข้อมูลจากชุดทดสอบ วัดความถูกต้องโดยเปรียบเทียบกับกับคลาสที่ถูกติดป้ายโดยผู้เชี่ยวชาญ

ในกรณีที่ระบบจำแนกประเภทได้ตรงกับคลาสที่ถูกตัดสินด้วยผู้เชี่ยวชาญให้นับเป็นโทเค็นที่ระบบจำแนกได้ถูกต้อง และคำนวณความแม่น(accuracy) ของระบบด้วยสมการ 4.1

$$\text{ความแม่นของระบบ (Accuracy)} = \frac{\text{จำนวนโทเค็นที่ระบบจำแนกได้ถูกต้อง}}{\text{จำนวนโทเค็นทั้งหมด}} \qquad (4.1)$$

4.4 การแปลงให้เป็นบรรทัดฐานเดียวกัน

หลังจากวัดผลการทดลองแล้ว ระบบจะเลือกวิธีการตรวจจับที่ดีที่สุด มาประยุกต์ใช้กับการสร้างเครื่องมือในการแปลงให้เป็นบรรทัดฐานโดยคำนึงถึงวิธีการอ่าน ซึ่งคำอ่านของคำที่มีการเล่นคำแบบซ้ำอักษรนั้น จะเหมือนกับคำอ่านของคำต้นฉบับก่อนการซ้ำตัวอักษร

โดยสร้างระบบแปลงให้เป็นบรรทัดฐานเดียวกันจาก 2 วิธีการดังต่อไปนี้

1. สร้างแบบจำลองต้นไม้ตัดสินใจโดยการติดป้ายด้วยคลาสของวิธีการจัดการในการแปลงให้เป็นบรรทัดฐาน แล้วนำมาประมวลผลกับข้อมูลทดสอบและตรวจสอบความแม่นยำของระบบด้วยการตัดสินจากผู้เชี่ยวชาญโดยนับจากโทเค็นที่แปลงได้ถูกต้อง เทียบกับการเล่นคำ 1 ครั้งของผู้เขียน

2. นำแบบจำลองวิธีการจัดการในการแปลงข้อความให้เป็นบรรทัดฐานไปใช้กับข้อมูลฝึกฝนเพื่อตรวจสอบความแม่นยำบนชุดข้อมูลฝึกฝน แล้วสร้างกฎเพิ่มเติมในกรณีที่มีการจัดการกับโทเค็นไม่ถูกต้อง จากนั้นจึงนำแบบจำลองควบคู่กับกฎที่สร้างขึ้นไปใช้ในการติดป้ายวิธีการแปลงข้อความให้เป็นบรรทัดฐาน ประมวลผลกับข้อมูลทดสอบและตรวจสอบความแม่นยำของระบบด้วยการตัดสินจากผู้เชี่ยวชาญโดยนับจากโทเค็นที่แปลงได้ถูกต้อง เทียบกับการเล่นคำ 1 ครั้งของผู้เขียน

4.5 การวัดความประสิทธิภาพของระบบแปลงข้อความเป็นบรรทัดฐาน

การวัดความประสิทธิภาพของระบบแปลงข้อความเป็นบรรทัดฐานจะวัดผลจากการตัดสินของผู้เชี่ยวชาญในการตรวจสอบว่าระบบแปลงข้อความที่มีการเล่นคำของผู้เขียนในแต่ละครั้งให้เป็นบรรทัดฐานได้ถูกต้องหรือไม่ โดยไม่คำนึงถึงจำนวนโทเค็น โดยสนใจเฉพาะโทเค็นที่ถูกตรวจจับจากระบบตรวจจับข้อความที่มีการเล่นคำเท่านั้น

ตัวอย่างเช่น

|เออ|...|เพลง|มัน|เก่า|มาก|กก|จน|เรา|ก็|ร้อง|ได้|แค่|...| |ครั้งนี้|คง|ถูกใจ|

|ไม|วัน|นี้|หนู|ไม่|เห็น|พี่โก้|เนี่ยยยยยยยยยยยยยยยย|

ข้อความที่ถูกขีดเส้นใต้จะถูกนับเป็นการเล่นคำ 1 ครั้งซึ่งระบบการแปลงข้อความให้เป็นบรรทัดฐานจะต้องรวมโทเค็นทั้งสองให้เป็นโทเค็นเดียวกันและแปลงเป็นคำที่คำอ่านตรงกับที่ผู้เขียนต้องการคือ |มาก| ซึ่งในกรณีนี้ระบบจะต้องแปลงข้อความเป็น “|เออ|...|เพลง|มัน|เก่า|มาก|จน|เรา|ก็|ร้อง|ได้|แค่|...| |ครั้งนี้|คง|ถูกใจ|” ผู้เชี่ยวชาญจึงจะตัดสินว่าแปลงถูกหนึ่งข้อความ

บทที่ 5

ผลการทดลองและวิเคราะห์ผลการทดลอง

ในบทนี้จะแสดงผลการทดลองตามขั้นตอนการทดลองที่ได้จากบทที่แล้ว ดังต่อไปนี้

1. ประเมินผลการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรของระบบเส้นเชื่อมฐาน
2. ประเมินผลการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรจากขั้นตอนวิธีที่นำเสนอ
3. วิเคราะห์ผลการทดลองของการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษร
4. วัดผลการแปลงข้อความเป็นบรรทัดฐานด้วยวิธีการที่นำเสนอ

5.1 ประเมินผลการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรของระบบเส้นเชื่อมฐาน

ผลการทดสอบการตรวจจับการเล่นคำจากแคนดิเดทโทเค็นในชุดทดสอบจำนวน 12,551 โทเค็น ด้วยระบบเส้นเชื่อมฐานวิธีต่างๆเทียบการตัดสินจากผู้เชี่ยวชาญได้ค่าความแม่นยำที่แตกต่างกันดังตารางที่ 5-1

ตารางที่ 5-1 ค่าความแม่นยำของระบบเส้นเชื่อมฐานในการตรวจจับการเล่นคำ

เส้นเชื่อมฐานวิธีที่	สรุปเงื่อนไขของวิธีเส้นเชื่อมฐานโดยย่อ			จำนวนโทเค็นที่ระบบจำแนกประเภทได้ถูกต้อง	เปอร์เซ็นต์ความถูกต้องของระบบเส้นเชื่อมฐาน
	การคัดเลือกเบื้องต้น	ผลจากการค้นในพจนานุกรม	การจำแนกโทเค็นที่ไม่มีในพจนานุกรม		
1-1	ไม่มี	กรณีที่โทเค็นมีค่าลักษณะเด่น Dict เป็น ‘COP’: เป็นการเล่นคำ ‘COR’: ไม่เป็นการเล่นคำ	ถูกจำแนกเป็นโทเค็นที่มีการเล่นคำ	10,296	82.03%
1-2			ถูกจำแนกเป็นโทเค็นที่ไม่มีการเล่นคำ	8,298	66.11%
2-1	กรณี ExpC>3	สำหรับโทเค็นที่ ExpC <=3	ถูกจำแนกเป็นโทเค็นที่มีการเล่นคำ	10,296	82.03%

เส้นเชื่อมฐานวิธีที่	สรุปเงื่อนไขของวิธีเส้นเชื่อมฐานโดยย่อ			จำนวนโทเค็นที่ระบบจำแนกประเภทได้ถูกต้อง	เปอร์เซ็นต์ความถูกต้องของระบบเส้นเชื่อมฐาน
	การคัดเลือกเบื้องต้น	ผลจากการค้นในพจนานุกรม	การจำแนกโทเค็นที่ไม่มีในพจนานุกรม		
2-2	ถูกจำแนกว่ามีการเล่นคำ	กรณีที่โทเค็นมีค่าลักษณะเด่น Dict เป็น 'COP': เป็นการเล่นคำ 'COR': ไม่เป็นการเล่นคำ	ถูกจำแนกเป็นโทเค็นที่ไม่มีการเล่นคำ	9,708	77.35%
3-1	กรณี ExpC>2 ถูกจำแนกว่ามีการเล่นคำ	สำหรับโทเค็นที่ ExpC <=2 case 'COP':WP, case 'COR':NWP	ถูกจำแนกเป็นโทเค็นที่มีการเล่นคำ	10,296	82.03%
3-2			ถูกจำแนกเป็นโทเค็นที่ไม่มีการเล่นคำ	10,860	86.53%

5.2 ประเมินผลการตรวจจับการเล่นคำด้วยวิธีซ้ำตัวอักษรจากขั้นตอนวิธีที่นำเสนอ

ผลการทดสอบการตรวจจับการเล่นคำจากแคนดิเดทโทเค็นในชุดทดสอบจำนวน 12,551 โทเค็น ด้วยขั้นตอนวิธีที่นำเสนอทั้ง 3 วิธี ได้ค่าความแม่นยำที่แตกต่างกันดังตาราง

- ผลการทดลองที่ 1 การจำแนกประเภทโทเค็นโดยใช้แบบจำลองการตรวจจับที่สร้างสำหรับโทเค็นที่ไม่มีในพจนานุกรมเท่านั้น ดังแสดงในตารางที่ 5-2

ตารางที่ 5-2 ผลจากการจำแนกด้วยแบบจำลองจากการทดลองที่ 1

ผลจากการจำแนกด้วยแบบจำลองจากการทดลองที่ 1	**คลาสจริงของโทเค็น**	
	เป็นโทเค็นที่มีการเล่นคำ	**เป็นโทเค็นที่ไม่มีการเล่นคำ**
เป็นโทเค็นที่มีการเล่นคำ	87.36%	6.45%
เป็นโทเค็นที่ไม่มีการเล่นคำ	12.64%	93.55%
ความแม่นโดยรวมของระบบ (Total Accuracy)		89.71%

- ผลการทดลองที่ 2 การจำแนกโทเค็นทั้งหมดจากแบบจำลองที่สร้างจากชุดข้อมูลฝึกฝนบนการสกัดลักษณะเด่นทั้งหมด

ตารางที่ 5-3 ผลจากการจำแนกด้วยแบบจำลองจากการทดลองที่ 2

ผลจากการจำแนกด้วยแบบจำลองจากการทดลองที่ 2	**คลาสจริงของโทเค็น**	
	เป็นโทเค็นที่มีการเล่นคำ	**เป็นโทเค็นที่ไม่มีการเล่นคำ**
เป็นโทเค็นที่มีการเล่นคำ	98.95%	2.37%
เป็นโทเค็นที่ไม่มีการเล่นคำ	1.05%	97.63%
ความแม่นโดยรวมของระบบ (Total Accuracy)		98.45%

- ผลการทดลองที่ 3 การจำแนกโทเค็นโดยใช้แบบจำลองที่ได้จากการทดลองที่ 2 ร่วมกับการตรวจจับก่อนหน้า

ตารางที่ 5-4 ผลจากการจำแนกด้วยแบบจำลองจากการทดลองที่ 3

ผลจากการจำแนกด้วยแบบจำลองจากการทดลองที่ 3	**คลาสจริงของโทเค็น**	
	เป็นโทเค็นที่มีการเล่นคำ	**เป็นโทเค็นที่ไม่มีการเล่นคำ**
เป็นโทเค็นที่มีการเล่นคำ	98.96%	2.37%
เป็นโทเค็นที่ไม่มีการเล่นคำ	1.04%	97.63%
ความแม่นโดยรวมของระบบ (Total Accuracy)		98.45%

วิเคราะห์ผลการตรวจจับการเล่นคำด้วยวิธีซ้ำอักษร

จากผลการจำแนกประเภทของระบบจากทุกๆกรอบงานของการทดลอง จะพบว่าการสร้างแบบจำลองต้นไม้ตัดสินใจจากลักษณะเด่นที่นำเสนอ สามารถจำแนกการเล่นคำด้วยวิธีซ้ำอักษร ได้ดีกว่าวิธีเส้นเชื่อมฐาน โดยวิธีการทดลองที่ 2 และ 3 มีประสิทธิภาพเท่าเทียมกันและสูงกว่าการทดลองวิธีที่ 1 อย่างมีนัยสำคัญ อย่างไรก็ตามสิ่งที่แตกต่างกันระหว่างกรอบงานของการทดลองทั้ง 3 วิธีคือลักษณะของแคนดิเดทที่จะถูกนำไปจำแนกด้วยแบบจำลองต้นไม้ตัดสินใจ

ในการทดลองที่ 1 เฉพาะโทเค็นไม่มีในพจนานุกรม (Dict="UNK") เท่านั้นที่จะถูกจำแนกด้วยแบบจำลองตามลักษณะเด่นที่นำเสนอ ซึ่งเป็นกรอบงานที่พยายามพัฒนาประสิทธิภาพขึ้นจากวิธีเส้นเชื่อมฐานวิธีที่ 1 ซึ่งใช้วิธีตัดสินให้โทเค็นเหล่านี้ไปอยู่ในคลาสใดคลาสหนึ่ง ในกรณีนี้ลักษณะเด่นที่นำเสนอช่วยให้ค่าความผิดพลาดลดลงจากเดิมถึง 42.73% (จากความผิดพลาด 17.97% เหลือเพียง 10.29%) และ 69.64% (จากความผิดพลาด 33.89% เหลือเพียง 10.29%)

ในการทดลองที่ 2 และการทดลองที่ 3 ค่าลักษณะเด่น Dict หรือการมีอยู่ในพจนานุกรมไม่ได้ถูกพิจารณาก่อน แต่ถูกนำไปใช้เป็นลักษณะเด่นหนึ่งในการสร้างแบบจำลอง ซึ่งทั้งสองวิธีให้ค่าความแม่นยำที่ 98.45% หมายถึงโทเค็นที่ถูกตรวจจับไม่ถูกต้องมีเพียงไม่เกิน 200 โทเค็นจาก 12,551 โทเค็นเท่านั้น อย่างไรก็ตาม เมื่อเปรียบเทียบจำนวนแคนดิเดทโทเค็นผ่านทำนายคลาสด้วยแบบจำลองต้นไม้ตัดสินใจของการทดลองที่ 2 กับการทดลองที่ 3 จะพบว่า ถ้าสามารถพิจารณาจากความยาวของโทเค็นก่อนการใช้แบบจำลองในการทำนายคลาสของโทเค็น 42.85% ของแคนดิเดทโทเค็น ซึ่งเป็นโทเค็นที่มีการซ้ำของตัวอักษรตั้งแต่ 3 ตัวขึ้นไปจะสามารถตัดสินว่าเป็นการเล่นคำด้วยวิธีซ้ำตัวอักษรได้เลยโดยที่ไม่ต้องผ่านการตัดสินใจจากแบบจำลองต้นไม้ตัดสินใจ และระบบยังคงมีระดับความสามารถในการตรวจจับข้อความที่เป็นการเล่นคำด้วยวิธีซ้ำตัวอักษรสูงเท่ากับการใช้แบบจำลองนี้กับทุกๆโทเค็น

และเมื่อวิเคราะห์ข้อมูลในส่วนที่ทำให้ผลของการทดลองที่ 2 และ 3 ตรวจจับได้ดีกว่าการทดลองที่ 1 และการทดลองของระบบเส้นเชื่อมฐาน พบว่ารูปแบบของข้อมูลที่ทำให้แบบจำลองที่สร้างจากลักษณะเด่นที่นำเสนอตรวจจับได้ดีกว่าระบบเส้นเชื่อมฐานและการทดลองที่ 1 คือโทเค็นที่มีในพจนานุกรมหลายกรณีเป็นส่วนหนึ่งของการเล่นคำ เช่นคำว่า ยาก|กกก| หรือ บ้าง|งง| ที่ปรากฏในข้อความจะถูกผู้เชี่ยวชาญตัดสินเป็นการเล่นคำ แต่การทดลองที่ 1 จะตัดสินว่าไม่เป็นการเล่นคำ เนื่องจากคำว่า กก และ งง เป็นคำที่มีในพจนานุกรม ซึ่งข้อมูลในรูปแบบดังกล่าวคิดเป็น 87% ของการเล่นคำที่ระบบไม่สามารถตรวจจับได้ (Miss detection) และในขณะที่กรณีที่การทดลองที่ 1

ตรวจจับคำหรือข้อความที่ไม่มีการเล่นคำ มักเกิดจากกรณีที่มีการใช้สระเอ 2 ตัวแทนการใช้สระแอ ซึ่งเป็นการพิมพ์ผิด ไม่ใช่การเล่นคำ แต่การแทนที่ด้วยสระเอ 1 ตัวทำให้บังเอิญค้นเจอในพจนานุกรม เช่น และ -> เละ แต่ง->เต่ง หรือ แก -> เก แต่ในกรณีที่แบบจำลองยังไม่สามารถแก้ไขได้ส่วนใหญ่จะเกิดจากข้อความที่มีรูปแบบคล้ายการเล่นคำ แต่เป็นโทเค็นที่มี 2 คำในโทเค็นเดียวกัน เช่น

5.3 วัดผลประสิทธิภาพในการแปลงข้อความเป็นบรรทัดฐาน

ผลการแปลงข้อความเป็นบรรทัดฐานจากการเล่นคำโดยผู้เขียนทั้งหมด 7,313 ครั้ง จากข้อความชุดทดสอบ 48,040 ชิ้นข้อความ วัดผลโดยระบบแปลงข้อความชุดทดสอบให้เป็นข้อความที่เป็นบรรทัดฐานและตัดสินความถูกต้องโดยผู้เชี่ยวชาญ โดยเปรียบเทียบระหว่างวิธีใช้แบบจำลองต้นไม้ตัดสินใจ และวิธีการใช้แบบจำลองต้นไม้ตัดสินใจร่วมกับการใช้กฎดังตารางที่ 5.5

ตารางที่ 5-5 ประสิทธิภาพของระบบในการแปลงข้อความเป็นบรรทัดฐาน

วิธีที่ใช้จำแนกวิธีการแปลงข้อความเป็นบรรทัดฐาน	จำนวนข้อมูลที่แปลงเป็นบรรทัดฐานได้ถูกต้อง (ครั้ง)	เปอร์เซนต์ความถูกต้องในการแปลงข้อความที่ตรวจจับถูกต้องเป็นบรรทัดฐาน
แบบจำลองต้นไม้ตัดสินใจ	7,149	97.75%
แบบจำลองต้นไม้ตัดสินใจ+การใช้กฎ	7,254	99.19 %

วิเคราะห์ผลในการแปลงข้อความเป็นบรรทัดฐาน

ประสิทธิภาพในการแปลงข้อความที่ตรวจจับถูกต้องเป็นบรรทัดฐานจากการสร้างแบบจำลองต้นไม้ตัดสินใจร่วมกับการใช้กฎในข้อความที่จำแนกผิดของชุดฝึกฝน ทำให้ผลที่ได้ในการแปลงเป็นบรรทัดฐานของข้อมูลในชุดทดสอบ สามารถลดความผิดพลาดได้ถึง 64% (จากค่าความผิดพลาดเดิม 2.25% เหลือเพียง 0.81%) อย่างไรก็ตามทั้ง 2 วิธียังคงให้ค่าความแม่นยำในการแปลงข้อความเป็นบรรทัดฐานที่สูง ซึ่งสามารถนำไปประยุกต์ใช้เพื่อลดความกำกวมในการอ่านข้อความที่ไม่เป็นทางการประเภทที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษรในข้อความจากเว็บไซต์เครือข่ายทางสังคม เพื่อเพิ่มประสิทธิภาพในการอ่านข้อความที่ไม่เป็นทางการให้กับระบบสังเคราะห์เสียงได้

บทที่ 6
บทสรุปผลการวิจัย และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้นี้ได้เสนอวิธีการในการตรวจจับข้อความที่มีการเล่นคำด้วยวิธีซ้ำตัวอักษรโดยใช้ข้อความภาษาไทยที่ผู้ใช้งานสื่อสารกันในเว็บไซต์เครือข่ายทางสังคมในการวิเคราะห์และนำเสนอลักษณะเด่นที่ใช้ในสร้างแบบจำลองในการจำแนกประเภท และนำโทเค็นที่มีลักษณะเป็นการเล่นคำด้วยวิธีซ้ำอักษรมาประมวลผลเพื่อแปลงเป็นคำที่มีคำอ่านตรงกับคำที่ผู้เขียนต้องการ เพื่อให้ระบบสังเคราะห์เสียงสร้างสัทอักษรของข้อความที่มีเสียงอ่านตรงตามความต้องการของผู้เขียน ซึ่งลักษณะเด่นที่ระบบนำเสนอช่วยทำให้ระบบในการตรวจจับการเล่นคำด้วยวิธีซ้ำอักษร จะมีความสำคัญและเป็นประโยชน์ต่อการสร้างระบบสังเคราะห์เสียงที่มีประสิทธิภาพสูงขึ้น

6.2 ข้อเสนอแนะ

จากความผิดพลาดในการตรวจจับข้อความที่มีการเล่นคำด้วยวิธีซ้ำอักษรส่วนหนึ่งพบว่ามาจากการตัดคำซึ่งการใช้เครื่องมือตัดคำที่เรียนรู้จากชุดข้อมูลฝึกฝนที่เป็นคำที่เป็นทางการไม่มีการเล่นคำ ทำให้เกิดกรณีที่ตัดคำแล้วในโทเค็นเดียวกันมี 2 คำ เช่น |กูงง| |ไอออม| ทำให้ลักษณะไปเหมือนกับข้อความที่มีการเล่นคำแบบซ้ำอักษรทั้งๆที่เป็นคำปกติ ในขณะที่ บางคำมีการเล่นคำ แต่มีลักษณะเด่นและรูปแบบการเขียนเหมือนคำปกติเช่น สอบแทนน นกกก

และในบางกรณี มีการตัดคำแล้วทำให้คำที่มีการซ้ำมากกว่า 2 ตัวอักษรขึ้นไปถูกตัดแยกเป็นคนละโทเค็นในบางกรณี เช่นคำว่า น่ารักจังงงงงงงงงง เครื่องมือในการตัดคำจะตัดแยกเป็น |น่า|รัก|จัง|งงงงงงงง|งง| โทเค็น “งง” ซึ่งเป็นโทเค็นหลังสุดจะมีการซ้ำของอักษรเพียง 2 ตัว ซึ่งถ้าออกแบบระบบให้มีการตรวจจับการซ้ำอักษรตั้งแต่ 3 ตัวขึ้นไปก่อนมีการตัดคำ อาจช่วยให้สามารถตรวจจับข้อความที่มีการเล่นคำแบบซ้ำอักษรได้รวดเร็วขึ้น

และความผิดพลาดจากการทำข้อความให้เป็นบรรทัดฐานพบว่ากรณีที่สระที่ใช้เป็นสระที่เป็นอักขระเริ่มต้นคำ คือเป็นสระที่เขียนอยู่หน้าพยัญชนะและพยัญชนะต้นกับตัวสะกดเป็นตัวเดียวกัน เช่น แกก แงง แบบบบ แนนนน ทำให้เกิดความผิดพลาดเนื่องจากบางคำผู้ใช้ต้องการให้

ออกเสียงแบบไม่มีตัวสะกดเช่น แก , แง แต่ในบางครั้งต้องการให้ออกเสียงแบบมีตัวสะกดเช่น แบบ ในกรณีนี้ระบบต้องการข้อมูลฝึกฝนเหล่านี้เพิ่มเติมเพื่อสร้างกฎหรือแบบจำลองที่รองรับกรณีนี้ เนื่องจากข้อมูลฝึกฝนที่นำมาใช้สร้างแบบจำลองมีข้อมูลในคลาสที่แก้ไขด้วยการแทนที่ด้วยตัวอักษร 2 ตัวน้อยมากเมื่อเทียบกับการแทนที่ด้วยตัวอักษร 1 ตัว ระบบจึงสร้างแบบจำลองที่ไม่รอบรับกรณีที่ต้องแก้ไขด้วยการแทนที่ด้วยตัวอักษร 2 ตัว (ดูในภาคผนวก ก) ซึ่งอย่างไรก็ตามผลจากงานวิจัยนี้เป็นเพียงส่วนหนึ่งของการจัดการข้อความที่ไม่เป็นทางการจากการซ้ำอักษร ในกรณีที่ต้องการระบบสังเคราะห์เสียงที่มีประสิทธิภาพสูงขึ้นอาจจะพัฒนาไปสู่การนำข้อความที่ตรวจจับได้ว่าเป็นการเล่นคำ ไปปรับแต่งรูปแบบการสังเคราะห์เสียงเช่น สังเคราะห์เสียงข้อความเหล่านี้ด้วยการลากเสียงยาว หรือเน้นข้อความให้ดังขึ้น หรือนำวิธีการนี้ไปประยุกต์ใช้กับความไม่เป็นทางการรูปแบบอื่นๆที่พบในภาษาไทยต่อไป เช่นการเขียนทับศัพท์ การใช้วรรณยุกต์ไม่ถูกต้อง เพื่อช่วยให้ระบบสังเคราะห์เสียงมีประสิทธิภาพสูงขึ้นต่อไป

รายการอ้างอิง

[1] Olinsky, C. and Black, A. Non-Standard Word and Homograph Resolution for AsiaLanguage Text Analysis. International Conference of Spoken Language Processing (ICSLP 2000).

[2] Panchapagesan, K., Partha, P.T., N. Sridhar, K., Kalika B., and Ramakrishnan, A.G. Hindi Text Normalization. Proceeding of Knowledge Based Computing Systems (KBCS 2004): 19-22.

[3] Firoj, A., S.M. Murtoza, H. and Mumit, K. Bangla Text Normalization Conference on Language and Technology (CLT 2009).

[4] Uden, S. and Pema C. Dzongkha Text Normalization Algorithm. PAN Localization Project (2010).

[5] Tao, Z., Yuan, D., Dezhi, H., Wu, L. and Haila, W. A Three-Stage Text Normalization Strategy for Mandarin Text-to-Speech Systems. 6th International Symposium of Chinese Spoken Language Processing (ISCSLP 2008).

[6] Clark, E. and Araki, K. Text normalization in social media: Progress, problems and applications for a pre-processing system of casual English. Procedia Social and Behavioral Sciences (2011): 2-11.

[7] Clark, A. Pre-processing very noisy text Proceedings of Workshop on Shallow Processing of Large Corpora (2003): 12-22.

[8] Aw, A., Zhang, M., Xiao, J. and Su, J. A phrase-based statistical model for SMS text normalization. Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions (2006): 33-40.

[9] Henriquez, CA. and Hernandez, A. A ngram-based statistical machine translation approach for text normalization onchat-speak style communications. Proceedings of CAW2.0 (2009): 1-5.

[10] Hemalatha, I., Saradhi Varma, G.P. and Govardhan, A. Preprocessing the Informal Text for efficient Sentiment Analysis. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS 2012)

[11] NECTEC. VAJA6.0 text-to-speech. [Online]. 2011 Available from: http://vaja.nectec.or.th [2012, November 1]

[12] ภาทิพ ศรีสุทธิ์. อักษรไทยและการผันอักษร [ออนไลน์]. 2550. แหล่งที่มา: http://www.st.ac.th/bhatips/gramma3.htm [1 พฤษภาคม 2556]

[13] สมศักดิ์ ทองช่วย. เอกสารประกอบการจัดการเรียนการสอน รายวิชา ท40105 หลักภาษาไทยในชีวิตประจำวัน [ออนไลน์]. แหล่งที่มา: http://www.mwit.ac.th/~saktong/learn6/75.pdf [1 พฤษภาคม 2556]

[14] Haruechaiyasak, C., Kongyoung S. and Dailey, M. A comparative study on Thai word segmentation approaches. Proceeding of the ECTI-CON (2008): 125-128.

[15] Haruechaiyasak, C. and Kongyoung S. TLex: Thai Lexeme Analyser Based on the Conditional Random Fields. Proceedings of Eighth International Symposium on Natural Language Processing (SNLP2009).

[16] Wutiwiwatchai, C. and Furui, S.,Thai speech processing technology: A review. Speech Communication 49, 1 (2007): 8–27

[17] อัศนีย์ ก่อตระกูล. การประมวลผลภาษามนุษย์ด้วยคอมพิวเตอร์ : เส้นทางสู่การพัฒนาระบบสารสนเทศอัจฉริยะ. หน่วยปฏิบัติการวิจัยเชี่ยวชาญเฉพาะการประมวลผลภาษาธรรมชาติและเทคโนโลยีสารสนเทศอัจฉริยะ (NAiST) ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, 2549.

[18] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. The WEKA Data Mining Software: An Update SIGKDD Explorations 11, 1 (2009).

[19] Spoken Language System and Assistive Technology Laboratory, Faculty of Engineering, Chulalongkorn University. Thai Speech Resource Builder [ออนไลน์]. แหล่งที่มา:http://www.facebook.com/apps/application.php?id=328409973860155 [5 มกราคม 2556]

[20] สมนึก ธนการ. อักษรนำ. [ออนไลน์]. แหล่งที่มา: http://www.w-nikro.com/index.php?lay=show&ac=article&Id=538980100 [1 พฤษภาคม 2556]

[21] คณะอักษรศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย. Principles of Romanization for Thai Script by transcription method. [ออนไลน์]. แหล่งที่มา: http://www.arts.chula.ac.th/~ling/tts/principles_eng.pdf [1 พฤษภาคม 2556]

ภาคผนวก

ภาคผนวก ก

แบบจำลองต้นไม้ตัดสินใจที่ได้จากการทดลอง

1.แบบจำลองต้นไม้ตัดสินใจที่สร้างจากโทเค็นเฉพาะที่ไม่มีในพจนานุกรมจากการทดลองที่ 1 เพื่อใช้กับโทเค็นที่ไม่มีในพจนานุกรมเท่านั้น

=== Run information ===

cheme:weka.classifiers.meta.CVParameterSelection -P "C 0.1 0.5 21.0" -P "M 2.0 10.0 9.0" -X 10 -S 1 -W weka.classifiers.trees.J48 -- -C 0.25 -B -M 2

Relation: WORDPLAY

Instances: 4044

Attributes: 11

LastPrev

MatchLastFirst

StartLetter

StartType

PrefixType

EXPLetter

COUNT

AfterEXP

BeforEXP

EndWord

Class

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Cross-validated Parameter selection.

Classifier: weka.classifiers.trees.J48

Cross-validation Parameter: '-C' ranged from 0.1 to 0.5 with 21.0 steps

Cross-validation Parameter: '-M' ranged from 2.0 to 10.0 with 9.0 steps

Classifier Options: -C 0.46000000000000013 -M 4 -B

J48 pruned tree

AfterEXP = NONE

| EXPLetter = NorNoo

| | LastPrev = NgorNgoo: TRUE (8.0/1.0)

| | LastPrev != NgorNgoo

| | | StartType = CSN

| | | | StartLetter = LorLing: TRUE (4.0)

| | | | StartLetter != LorLing

| | | | | StartLetter = ChorChang: TRUE (5.0/1.0)

| | | | | StartLetter != ChorChang

| | | | | | LastPrev = Ake: TRUE (7.0/2.0)

| | | | | | LastPrev != Ake: FALSE (118.0/16.0)

| | | StartType != CSN

| | | | StartLetter = SraAe: TRUE (32.0/12.0)

| | | | StartLetter != SraAe

| | | | | StartLetter = SraAir: TRUE (4.0/1.0)

| | | | | StartLetter != SraAir: FALSE (8.0/2.0)

| EXPLetter != NorNoo

| | EXPLetter = RorRua: FALSE (6.0)

| | EXPLetter != RorRua

| | | StartLetter = SorSo: FALSE (6.0/2.0)

| | | StartLetter != SorSo

| | | | BeforEXP = PorPla

| | | | | LastPrev = E: FALSE (4.0/1.0)

| | | | | LastPrev != E: TRUE (7.0/3.0)

| | | | BeforEXP != PorPla

| | | | | EXPLetter = BorBaiMai

| | | | | | StartLetter = SorSua: FALSE (6.0)

| | | | | | StartLetter != SorSua

| | | | | | | StartLetter = SraAe: FALSE (21.0/6.0)

| | | | | | | StartLetter != SraAe: TRUE (197.0/14.0)

| | | | | EXPLetter != BorBaiMai: TRUE (1000.0/35.0)

AfterEXP != NONE

| EXPLetter = SraAr: TRUE (8.0)

| EXPLetter != SraAr

| | EndWord = .

| | | StartLetter = HorHeep: TRUE (5.0)

| | | StartLetter != HorHeep

| | | | LastPrev = GorGai: TRUE (9.0/2.0)

| | | | LastPrev != GorGai

| | | | | EXPLetter = YorYak: TRUE (7.0/3.0)

| | | | | EXPLetter != YorYak

| | | | | | BeforEXP = GorGai: TRUE (5.0/2.0)

| | | | | | BeforEXP != GorGai: FALSE (96.0/3.0)

| | EndWord != .

| | | AfterEXP = MaiYamok

| | | | BeforEXP = NONE: TRUE (4.0/1.0)

| | | | BeforEXP != NONE: FALSE (4.0)

| | | AfterEXP != MaiYamok

| | | | AfterEXP = SorSua

| | | | | EndWord = Karan: TRUE (4.0/1.0)

| | | | | EndWord != Karan: FALSE (22.0/2.0)

| | | | AfterEXP != SorSua: FALSE (2447.0/34.0)

Number of Leaves : 27

Size of the tree : 53

Time taken to build model: 78.44 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	3859	95.4253 %
Incorrectly Classified Instances	185	4.5747 %
Kappa statistic	0.8949	
Mean absolute error	0.0701	
Root mean squared error	0.2003	
Relative absolute error	16.106 %	
Root relative squared error	42.935 %	
Total Number of Instances	4044	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.928	0.033	0.929	0.928	0.928	0.963	TRUE
	0.967	0.072	0.966	0.967	0.966	0.963	FALSE
Weighted Avg.	0.954	0.06	0.954	0.954	0.954	0.963	

=== Confusion Matrix ===

a	b	<-- classified as
1201	93	a = TRUE
92	2658	b = FALSE

2. แบบจำลองต้นไม้ตัดสินใจที่สร้างจากโทเค็นชุดฝึกสอนทั้งหมดโดยใช้ลักษณะเด่นที่นำเสนอทั้งหมด จากการทดลองที่ 2

ผลลัพธ์จากโปรแกรมเวก้า

=== Run information ===

Scheme:weka.classifiers.meta.CVParameterSelection -P "M 2.0 4.0 3.0" -P "C 0.3 0.5 10.0" -X 10 -S 1 -W weka.classifiers.trees.J48 -- -C 0.25 -B -M 2

Relation: WORDPLAY

Instances: 27480

Attributes: 11

LastPrev

MatchLastFirst

StartLetter

StartType

PrefixType

EXPLetter

COUNT

AfterEXP

BeforEXP

EndWord

Class

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Cross-validated Parameter selection.

Classifier: weka.classifiers.trees.J48

Classifier Options: -M 2 -C 0.3222222222222222 -B

J48 pruned tree

AfterEXP = NONE

| PrefixType = COR

| | MatchLastFirst = 0

| | | EXPLetter = GorGai

| | | | LastPrev = E: FALSE (5.0)

| | | | LastPrev != E

| | | | | LastPrev = NorNoo: FALSE (3.0)

| | | | | LastPrev != NorNoo: TRUE (40.0/8.0)

| | | EXPLetter != GorGai

| | | | EXPLetter = NgorNgoo

| | | | | LastPrev = Sraii: TRUE (9.0/1.0)

```
| | | | |  LastPrev != Sraii: FALSE (243.0/23.0)
| | | |  EXPLetter != NgorNgoo
| | | | |  BeforEXP = NONE
| | | | | |  LastPrev = Toe: TRUE (10.0/1.0)
| | | | | |  LastPrev != Toe: FALSE (50.0/2.0)
| | | | |  BeforEXP != NONE: FALSE (2647.0)
| |  MatchLastFirst != 0
| | |  BeforEXP = NONE: TRUE (1675.0/24.0)
| | |  BeforEXP != NONE: FALSE (76.0)
|  PrefixType != COR
| |  COUNT <= 2.0
| | |  EXPLetter = RorRua
| | | |  StartType = CSN: FALSE (27.0/3.0)
| | | |  StartType != CSN: TRUE (7.0/1.0)
| | |  EXPLetter != RorRua
| | | |  EXPLetter = NorNoo
| | | | |  PrefixType = COP: TRUE (173.0/4.0)
| | | | |  PrefixType != COP
| | | | | |  StartType = CSN
```

| | | | | | | StartLetter = LorLing: TRUE (4.0)

| | | | | | | StartLetter != LorLing

| | | | | | | | LastPrev = PorPla: TRUE (2.0)

| | | | | | | | LastPrev != PorPla

| | | | | | | | | LastPrev = NgorNgoo: TRUE (6.0/1.0)

| | | | | | | | | LastPrev != NgorNgoo

| | | | | | | | | | StartLetter = RorRua: TRUE (2.0)

| | | | | | | | | | StartLetter != RorRua

| | | | | | | | | | | StartLetter = ChorChang: TRUE (5.0/1.0)

| | | | | | | | | | | StartLetter != ChorChang

| | | | | | | | | | | | LastPrev = Ake: TRUE (6.0/2.0)

| | | | | | | | | | | | LastPrev != Ake

| | | | | | | | | | | | | LastPrev = DoDek

| | | | | | | | | | | | | | StartLetter = JorJan: TRUE (3.0)

| | | | | | | | | | | | | | StartLetter != JorJan: FALSE (4.0)

| | | | | | | | | | | | | LastPrev != DoDek: FALSE (108.0/10.0)

| | | | | | StartType != CSN

| | | | | | | LastPrev = Ake: FALSE (3.0)

| | | | | | | LastPrev != Ake: TRUE (43.0/16.0)

```
|  |  |  |  EXPLetter != NorNoo
|  |  |  |  |  BeforEXP = PorPla
|  |  |  |  |  |  LastPrev = GorGai: FALSE (3.0)
|  |  |  |  |  |  LastPrev != GorGai
|  |  |  |  |  |  |  LastPrev = OrAng: TRUE (4.0/1.0)
|  |  |  |  |  |  |  LastPrev != OrAng
|  |  |  |  |  |  |  |  LastPrev = E
|  |  |  |  |  |  |  |  |  PrefixType = COP: TRUE (14.0/5.0)
|  |  |  |  |  |  |  |  |  PrefixType != COP: FALSE (4.0/1.0)
|  |  |  |  |  |  |  |  LastPrev != E
|  |  |  |  |  |  |  |  |  StartLetter = SraAir: FALSE (10.0/3.0)
|  |  |  |  |  |  |  |  |  StartLetter != SraAir: TRUE (10.0/2.0)
|  |  |  |  |  BeforEXP != PorPla
|  |  |  |  |  |  StartLetter = SorSo
|  |  |  |  |  |  |  LastPrev = SraA: FALSE (5.0/1.0)
|  |  |  |  |  |  |  LastPrev != SraA: TRUE (4.0)
|  |  |  |  |  |  StartLetter != SorSo
|  |  |  |  |  |  |  StartLetter = SPACE
|  |  |  |  |  |  |  |  LastPrev = E: FALSE (2.0)
```

```
|  |  |  |  |  |  |  |  LastPrev != E: TRUE (3.0)
|  |  |  |  |  |  |  StartLetter != SPACE
|  |  |  |  |  |  |  |  EXPLetter = ChorChang
|  |  |  |  |  |  |  |  |  StartLetter = PorPla: FALSE (2.0)
|  |  |  |  |  |  |  |  |  StartLetter != PorPla: TRUE (4.0)
|  |  |  |  |  |  |  |  EXPLetter != ChorChang
|  |  |  |  |  |  |  |  |  EXPLetter = TorTao
|  |  |  |  |  |  |  |  |  |  StartLetter = TorTao: FALSE (2.0)
|  |  |  |  |  |  |  |  |  |  StartLetter != TorTao: TRUE (4.0)
|  |  |  |  |  |  |  |  |  EXPLetter != TorTao
|  |  |  |  |  |  |  |  |  |  PrefixType = COP: TRUE (1854.0/32.0)
|  |  |  |  |  |  |  |  |  |  PrefixType != COP
|  |  |  |  |  |  |  |  |  |  |  StartLetter = SorSua
|  |  |  |  |  |  |  |  |  |  |  |  EXPLetter = BorBaiMai: FALSE (6.0)
|  |  |  |  |  |  |  |  |  |  |  |  EXPLetter != BorBaiMai: TRUE (14.0/2.0)
|  |  |  |  |  |  |  |  |  |  |  StartLetter != SorSua
|  |  |  |  |  |  |  |  |  |  |  |  EXPLetter = BorBaiMai
|  |  |  |  |  |  |  |  |  |  |  |  |  StartLetter = SraAe
|  |  |  |  |  |  |  |  |  |  |  |  |  |  LastPrev = Ake: TRUE (3.0/1.0)
```

```
| | | | | | | | | | | | | | LastPrev != Ake: FALSE (18.0/5.0)
| | | | | | | | | | | | | StartLetter != SraAe
| | | | | | | | | | | | | | LastPrev = .: FALSE (3.0/1.0)
| | | | | | | | | | | | | | LastPrev != .
| | | | | | | | | | | | | | | StartLetter = BorBaiMai: FALSE (3.0/1.0)
| | | | | | | | | | | | | | | StartLetter != BorBaiMai: TRUE (189.0/8.0)
| | | | | | | | | | | | EXPLetter != BorBaiMai: TRUE (979.0/33.0)
| | COUNT > 2.0: TRUE (11065.0)
AfterEXP != NONE
| COUNT <= 2.0
| | EXPLetter = SraAr: TRUE (15.0)
| | EXPLetter != SraAr
| | | EXPLetter = YorYak
| | | | EndWord = .: TRUE (11.0/3.0)
| | | | EndWord != .
| | | | | AfterEXP = NorNoo: TRUE (2.0)
| | | | | AfterEXP != NorNoo: FALSE (95.0/2.0)
| | | EXPLetter != YorYak
| | | | EXPLetter = GorGai
```

```
|  |  |  |  |  BeforEXP = NONE
|  |  |  |  |  |  LastPrev = E: FALSE (7.0)
|  |  |  |  |  |  LastPrev != E
|  |  |  |  |  |  |  AfterEXP = MaiYamok: TRUE (3.0)
|  |  |  |  |  |  |  AfterEXP != MaiYamok
|  |  |  |  |  |  |  |  AfterEXP = .: TRUE (10.0/2.0)
|  |  |  |  |  |  |  |  AfterEXP != .: FALSE (15.0/4.0)
|  |  |  |  |  BeforEXP != NONE
|  |  |  |  |  |  AfterEXP = .
|  |  |  |  |  |  |  StartLetter = Por+Pung: FALSE (3.0/1.0)
|  |  |  |  |  |  |  StartLetter != Por+Pung: TRUE (2.0)
|  |  |  |  |  |  AfterEXP != .: FALSE (308.0/11.0)
|  |  |  |  EXPLetter != GorGai: FALSE (7349.0/36.0)
|  COUNT > 2.0: TRUE (304.0/1.0)

Number of Leaves  :  61

Size of the tree :  121

Time taken to build model: 80.12 seconds

=== Stratified cross-validation ===

=== Summary ===
```

Correctly Classified Instances 27171 98.8755 %

Incorrectly Classified Instances 309 1.1245 %

Kappa statistic 0.9766

Mean absolute error 0.0182

Root mean squared error 0.0999

Relative absolute error 3.7955 %

Root relative squared error 20.3676 %

Total Number of Instances 27480

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.992	0.015	0.99	0.992	0.991	0.996	TRUE
0.985	0.008	0.987	0.985	0.986	0.996	FALSE
Weighted Avg. 0.989	0.013	0.989	0.989	0.989	0.996	

=== Confusion Matrix ===

a	b	<-- classified as
16296	138	a = TRUE
171	10875	b = FALSE

3. แบบจำลองต้นไม้ตัดสินใจในการจำแนกวิธีการทำให้เป็นบรรทัดฐานที่สร้างขึ้นจากข้อมูลชุดฝึกสอน

=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -B -M 2

Relation: WORDPLAY

Instances: 16417

Attributes: 12

LastPrev

MatchLastFirst

StartLetter

StartType

PrefixType

EXPLetter

COUNT

AfterEXP

BeforEXP

EndWord

Class

Handling

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

BeforEXP = NONE

| LastPrev = E

| | AfterEXP = NONE

| | | StartLetter = NgorNgoo: Merge-Replace (4.0)

| | | StartLetter != NgorNgoo: RemoveAllExp (32.0/12.0)

| | AfterEXP != NONE: ReplaceWith1 (3.0/1.0)

| LastPrev != E

| | AfterEXP = Karan: RemoveAllExp (3.0/1.0)

| | AfterEXP != Karan

| | | StartType = CSN: Merge-Replace (3326.0/46.0)

| | | StartType != CSN: ReplaceWith1 (3.0/1.0)

BeforEXP != NONE

| StartLetter = SPACE

| | LastPrev = E: ReplaceWith1 (2.0)

| | LastPrev != E: RemoveAllExp (6.0/1.0)

```
|   StartLetter != SPACE
|   |   EXPLetter = OrAng
|   |   |   LastPrev = E
|   |   |   |   StartType = CSN: ReplaceWith1 (101.0)
|   |   |   |   StartType != CSN
|   |   |   |   |   AfterEXP = NONE: ReplaceWith1 (164.0/38.0)
|   |   |   |   |   AfterEXP != NONE: Merge-Replace (16.0/6.0)
|   |   |   LastPrev != E
|   |   |   |   StartLetter = SraO: Merge-Replace (3.0/1.0)
|   |   |   |   StartLetter != SraO: ReplaceWith1 (355.0/2.0)
|   |   EXPLetter != OrAng: ReplaceWith1 (12399.0/27.0)

Number of Leaves  :  14

Size of the tree :  27

Time taken to build model: 0.41 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances       16260               99.0437 %
Incorrectly Classified Instances       157               0.9563 %
Kappa statistic                          0.971
```

Mean absolute error 0.0056

Root mean squared error 0.0543

Relative absolute error 5.0659 %

Root relative squared error 23.0787 %

Total Number of Instances 16417

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.986	0.004	0.984	0.986	0.985	0.993	Merge-Replace
0	0	0	0	0	0.15	no
0.997	0.024	0.994	0.997	0.995	0.989	ReplaceWith1
0	0	0	0	0	0.657	ReplaceWith2
0.436	0.001	0.545	0.436	0.485	0.786	RemoveAllExp
0	0	0	0	0	?	ReplaceWithSraAir
Weighted Avg. 0.99	0.02	0.988	0.99	0.989	0.988	

=== Confusion Matrix ===

a	b	c	d	e	<-- classified as
3282	0	39	0	9	a = Merge-Replace
0	0	1	0	1	b = no
31	0	12954	0	9	c = ReplaceWith1

7 0 28 0 1 | d = ReplaceWith2

17 0 14 0 24 | e = RemoveAllExp

4. กฎที่ใช้ในการแปลงให้เป็นบรรทัดฐาน สำหรับกรณีที่แทนที่ตัวอักษรซ้ำด้วยอักษร 2 ตัว

กรณีที่การอักษรที่ถูกซ้ำถูกประสมด้วยสระเอ และสระแอ ให้แทนที่ด้วยอักษร 2 ตัว

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวปวันรัตน์ หิรัญกาญจน์ เกิดวันจันทร์ที่ 25 พฤศจิกายน พ.ศ.2528 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้นที่ โรงเรียนสุราษฎร์ธานี จังหวัดสุราษฎร์ธานี สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจาก โรงเรียนมหิดลวิทยานุสรณ์ จังหวัดนครปฐม และจบการศึกษาระดับปริญญาตรีจากหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ กรุงเทพมหานคร เป็นนิสิตปริญญาโทในหลักสูตรในหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย กรุงเทพมหานคร