**การปกปิดข้อมูลเสียงพูดเพื่อปกป้องข้อมูลที่ระบุตัวบุคคล**

**ณัฏฐณิชา ชัยศิริพานิช1 และ ประวิตรานันท์ บุตรโพธิ์2**

*1คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ*

*2คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ*

*Emails: 60070135@it.kmitl.ac.th, 60070148@it.kmitl.ac.th*

**บทคัดย่อ**

ในปัจจุบันข้อมูลนับว่าเป็นสิ่งสำคัญต่อการนำไปประยุกต์ใช้กับปัญญาประดิษฐ์เพื่อพัฒนาประสิทธิภาพขององค์กร แต่ข้อมูลเหล่านั้นมักมีข้อจำกัดในเรื่องของข้อมูลส่วนบุคคล โดยเฉพาะข้อมูลของลูกค้า ซึ่งการบันทึกเสียงการสนทนาการทำธุรกรรมกับทางธนาคารก็นับว่ามีข้อมูลส่วนบุคคลของลูกค้าเป็นจำนวนมาก จึงเสียงต่อการลักลอบนำข้อมูลไปใช้ในทางที่ไม่ถูต้อง

ผู้จัดทำได้เล็งเห็นถึงปัญหานี้และได้ทดลองสร้างระบบโดยใช้ปัญญาประดิษฐ์ โดยภายในระบบมีการแปลงไฟล์เสียงพูดให้อยู่ในรูปแบบข้อความ จากนั้นตรวจจับคำที่เป็นข้อมูลส่วนบุคคลพร้อมกับเก็บระยะเวลาที่พูดในไฟล์เสียงและทำการแทนที่คำพูดที่เป็นข้อมูลส่วนบุคคลด้วยเสียงรบกวน เพื่อให้สามารถนำผลลัพธ์ไฟล์เสียงที่ผ่านการปกปิดข้อมูลส่วนบุคคลแล้วไปใช้ประโยชน์ในการวิเคราะห์ทางด้านอื่น ๆ ต่อไป

***คำสำคัญ –*** *ข้อมูลส่วนบุคคล; ศูนย์บริการข้อมูลลูกค้าทางโทรศัพท์ (Call Center); นิพจน์ระบุนาม (Named Entities); การแปลงเสียงพูดให้อยู่ในรูปแบบข้อความ (Speech-to-Text)*

**1. บทนำ**

ปัจจุบันการละเมิดข้อมูลส่วนบุคคลนั้นเกิดขึ้นได้หลายรูปแบบ ซึ่งการนำข้อมูลที่ได้จากการบันทึกบทสนทนาการทำธุรกรรมทางธนาคารผ่านโทรศัพท์ไปวิเคราะห์ในด้านต่าง ๆ ก็มีโอกาสที่ข้อมูลเหล่านั้นจะถูกลักลอบไปใช้ในทางที่ไม่ถูกต้อง ผู้จัดทำได้เล็งเห็นถึงความสำคัญของการรักษาข้อมูลส่วนบุคคลในการทำธุรกรรมกับทางธนาคารผ่านศูนย์บริการข้อมูลลูกค้าทางโทรศัพท์ โดยมีการทดลองสร้างระบบในการปกปิดข้อมูลที่ระบุตัวบุคคล ซึ่งภายในระบบจะดำเนินการแปลงไฟล์เสียงพูดให้อยู่ในรูปแบบข้อความ จากนั้นวิเคราะห์คำที่เป็นข้อมูลส่วนบุคคลพร้อมกับเก็บค่าของระยะเวลาที่พูดในไฟล์เสียงนั้น ระบบจะทำการแทรกเสียงรบกวนแทนที่คำพูดที่เป็นข้อมูลส่วนบุคคล และผลลัพธ์ที่ได้คือไฟล์เสียงที่ผ่านการปกปิดข้อมูลส่วนบุคคลแล้ว ซึ่งสามารถนำไปใช้ประโยชน์ในการวิเคราะห์ทางด้านอื่น ๆ ต่อไป

**2. ทฤษฎี เทคโนโลยีและเครื่องมือที่เกี่ยวข้อง**

ผู้จัดทำได้แบ่งรายละเอียดที่จะนำมาใช้กับการพัฒนาระบบเป็น 3 ส่วนหลัก ๆ ดังนี้

**2.1 ทฤษฎีที่เกี่ยวข้อง**

*2.1.1 การรู้จำเสียงพูด (Speech Recognition)*

เป็นสิ่งที่ช่วยให้โปรแกรมสามารถประมวลผลคำพูดของมนุษย์ให้อยู่ในรูปแบบลายลักษณ์อักษร โดยเน้นที่การแปลงเสียงพูดจากรูปแบบคำพูดเป็นข้อความ [1]

*2.1.2 การประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP)*

เป็นศาสตร์ที่ศึกษาเกี่ยวกับการทำให้คอมพิวเตอร์สามารถสื่อสารโต้ตอบด้วยภาษาของมนุษย์ และทำให้คอมพิวเตอร์เข้าใจภาษามนุษย์มากขึ้น เช่น Siri, Google Assistant และ Alexa [2]

*2.1.3* *การรู้จำนิพจน์ระบุนาม (Named Entity Recognition)*

เป็นขั้นตอนที่สำคัญสำหรับการพัฒนาระบบประมวลผลเอกสาร โดยเฉพาะอย่างยิ่งสำหรับระบบที่เกี่ยวข้องกับการเข้าถึงข้อมูล เช่น ระบบสกัดสารสนเทศ (Information Extraction) หรือในระบบค้นคืนเอกสาร (Information Retrieval) [3]

ผู้จัดทำนำทฤษฎีนี้มาใช้ในการพัฒนาระบบส่วนของการตรวจจับคำที่เป็นข้อมูลส่วนบุคคล

*2.1.4 Jaccard’s Coefficient Similarity*

เป็นสถิติประยุกต์แนวคิดในทฤษฎีเซตเพื่อใช้เปรียบเทียบความคล้ายคลึงและความหลากหลายของกลุ่มตัวอย่าง แนวคิดของค่าสัมประสิทธิ์ Jaccard’s Coefficient Similarity คือ การวัดค่าความคล้ายคลึงระหว่างกลุ่มประชากร 2 กลุ่ม โดยคำนวณจากขนาดของประชากรที่ทั้งสองกลุ่มมีตัวอย่างร่วมกัน [4] ดังสมการที่ 1

(1)

*2.1.5 Confusion Metrix*

ผู้จัดทำนำทฤษฎีนี้มาใช้ในการประเมิลผลประสิทธิภาพของการตรวจจับข้อมูลส่วนบุคคล มีหลักการคำนวณ [5] ดังนี้



**รูปที่ 1.** ตัวอย่างตาราง Confusion Matrix

มีการแบ่งการคำนวณค่าความแม่นยำต่าง ๆ ดังสมการ

(2)

(3)

(4)

(5)

**2.2 เทคโนโลยีและเครื่องมือที่เกี่ยวข้อง**

*2.2.1 Cloud Speech to Text by Google Cloud*

กูเกิลคลาวด์แพลตฟอร์มเป็นเว็บเซิร์ฟเวอร์ที่ให้บริการคลาวด์ ภายในกูเกิลคลาวด์แพลตฟอร์มมีบริการแยกย่อยอีก เช่น Cloud Speech to Text, Cloud Storage, Compute Engine, และ Machine Learning เป็นต้น

ผู้จัดทำเลือกบริการ Cloud Storage ในการเก็บไฟล์เสียง และใช้ Cloud Speech to Text ไลบรารี Speech ในการแปลงเสียงพูดให้อยู่ในรูปแบบข้อความ [6]

*2.2.2 Stanford Named Entity Recognizer (Stanford NER)*

เป็นการประยุกต์ใช้จากภาษาจาวาสำหรับการรู้จำนิพจน์ระบุนาม (Named Entity Recognizer) ซึ่งเป็นการจัดประเภทของคำในข้อความ เช่น ชื่อสิ่งของ ชื่อบุคคล และบริษัท [7]

*2.2.3 Natural Language Toolkit (NLTK)*

เป็นแพลตฟอร์มที่นิยมในโปรแกรมภาษาไพทอนsเพื่อทำงานกับข้อมูลภาษาของมนุษย์ พร้อมกับชุดของไลบรารีที่ช่วยในการประมวลข้อความ แบ่งประเภทของคำ (Classification) การแบ่งโทเค็นของคำ (Tokenization) การตัดคำ (Stemming) การติดแท็กคำ (Tagging) และการแยกวิเคราะห์คำ (Parsing) [8]

*2.2.4 spaCy*

เป็นไลบรารีสำหรับการทำการระมวลผลภาษาธรรมชาติขั้นสูงในภาษาไพทอน สามารถใช้ในการดำเนินการสกัดข้อมูล (Information Extraction) หรือระบบการทำความเข้าใจภาษาธรรมชาติเพื่อดำเนินการประมวลผลข้อความล่วงหน้าสำหรับการเรียนรู้เชิงลึก (Deep Learning) [9]

*2.2.5 Regular Expressions*

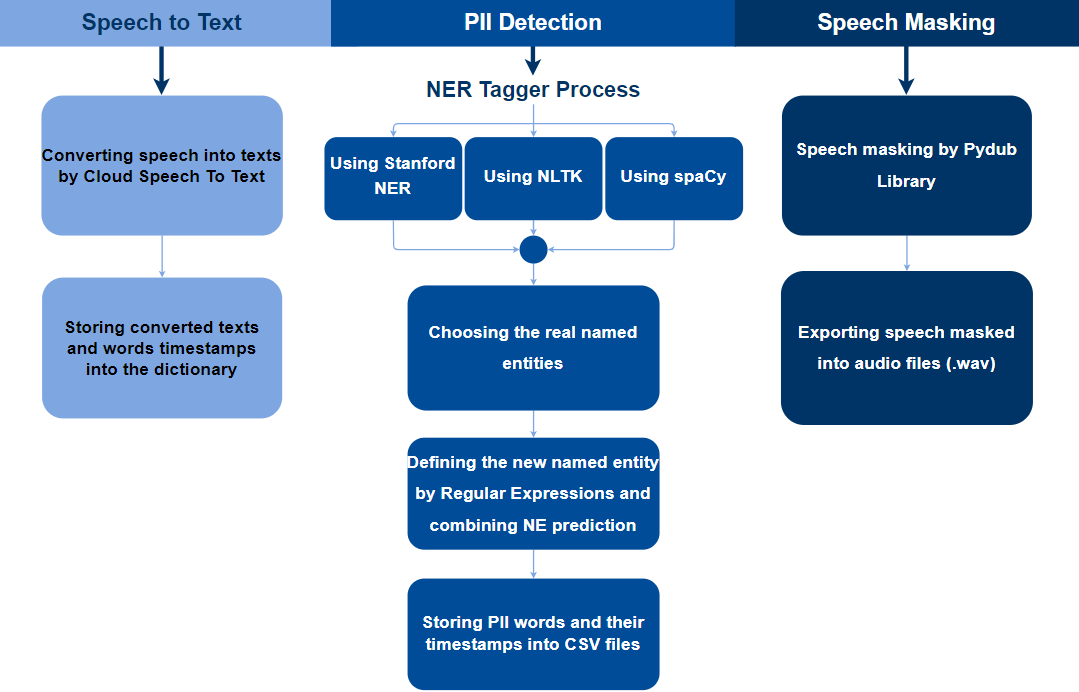
เป็นสัญลักษณ์ที่ใช้ระบุชุดของอักขระตัวอักษร เมื่อชุดของอักขระตัวอักษรที่เฉพาะเจาะจงนั้นอยู่ในชุดอักขระตัวอักษรที่มีการกำหนดให้เป็น Regular Expressions โดยทั่วไปจะใช้สัญลักษณ์ “\*”, “+”, “?”, “()” และ “|” ในการกำหนดเงื่อนไขของชุดตัวอักษร [10]

*2.2.6**Pydub*

เป็นไลบรารีหนึ่งของภาษาไพทอนที่ใช้ในการจัดการกับข้อมูลที่เป็นไฟล์เสียง

**3. ขั้นตอนการพัฒนาระบบ**

ผู้จัดทำได้นำเทคโนโลยีและเครื่องมือมาประยุกต์ใช้โดยมีการพัฒนาตามโครงร่างของระบบ ดังรูปที่ 2



**รูปที่ 2.** โครงร่างของระบบ

ภายในระบบมีการดำเนินงาน 3 ส่วน คือ การแปลงเสียงพูดให้อยู่ในรูปแบบข้อความ การตรวจจับคำที่เป็นข้อมูลส่วนบุคคลจากข้อมูลรูปแบบข้อความ และการแทนที่เสียงที่เป็นข้อมูลส่วนบุคคลด้วยเสียงรบกวน มีรายละเอียดการดำเนินงาน ดังนี้

**3.1 การแปลงเสียงพูดให้อยู่ในรูปแบบข้อความ**

*3.1.1 ประยุกต์ใช้ Cloud Speech to Text*

สร้างโปรเจกต์บนกูเกิลคลาวด์ และเปิดใช้งาน Cloud Storag API และ Cloud Speech to Text API จากนั้นอัปโหลดไฟล์เสียงขึ้นบน Cloud Storage

แปลงเสียงพูดให้อยู่ในรูปแบบข้อความโดยใช้โมดูลของไลบรารี Speech และเก็บค่าที่ได้จากการแปลงเป็น 4 ส่วน คือ บทสนทนา โทเค็นคำ เวลาเริ่มต้นของโทเค็นนั้น และเวลาที่สิ้นสุดของโทเค็นนั้น หน่วยเป็นวินาที

*3.1.2 บันทึกไฟล์ให้อยู่ในรูปแบบ Dictionary*

บันทึกเป็นประเภทไฟล์ JSON เพื่อทำการตรวจจับคำที่เป็นข้อมูลส่วนบุคคลในขั้นตอนถัดไป

**3.2 การตรวจจับคำที่เป็นข้อมูลส่วนบุคคลจากข้อมูลรูปแบบข้อความ**

นำข้อมูลผลลัพธ์จากกระบวนการก่อนหน้านี้มาประมวลผล มีวิธีดำเนินงาน ดังนี้

*3.2.1 กระบวนการตรวจจับนิพจน์ระบุนาม*

ใช้ไลบรารีทั้งหมด 3 ไลบรารี เพื่อเพิ่มความแม่นยำในการตรวจจับนิพจน์ระบุนาม ได้แก่ Stanford NER, NLTK และ spaCy มีการดำเนินงาน ดังนี้

- Stanford NER เลือกประเภทของนิพจน์ระบุนามในการติดแท็กจำนวน 5 ประเภท ได้แก่ PERSON, ORGANIZATION, LOCATION, DATE และ MONEY โดยเริ่มจากการทำ Word Tokenization ต่อมามีการติดแท็กนิพจน์ระบุนามจากอัลกอริทึมของ Stanford NER จากนั้นเก็บค่าเฉพาะโทเค็นที่เป็นนิพจน์ระบุนาม และจับคู่โทเค็นที่ไลบรารีแบ่งออกมาเทียบกับโทเค็นที่ Cloud Speech to Text แบ่งไว้ เพื่อให้แน่ใจว่าโทเค็นที่ถูกติดแท็กนั้นตรงกับระยะเวลาที่ Cloud Speech to Text ทำนายออกมา และเก็บค่าของคำที่ติดแท็กได้ พร้อมกับประเภทของนิพจน์ระบุนาม

- NLTK เลือกการติดแท็กเป็นจำนวน 6 ประเภท ได้แก่ ORGANIZATION, PERSON, LOCATION, GPE, DATE และ MONEY โดยเริ่มจากการทำ Word Tokenization จากนั้นทำการติดแท็กนิพจน์ระบุนามจากอัลกอริทึม NLTK ซึ่งต้องมีการติดแท็กส่วนของประโยค (Part-of-Speech) ก่อนจึงจะติดแท็กนิพจน์ระบุนามได้ และเลือกโทเค็นที่เป็นนิพจน์ระบุนาม จากนั้นเปลี่ยนประเภทนิพจน์ระบุนามให้สัมพันธ์กันทุกไลบรารี เช่น GPE และ LOC เปลี่ยนเป็น LOCATION เพื่อให้สะดวกต่อการประเมินผล จากนั้นจับคู่โทเค็นที่ไลบรารีแบ่งเทียบกับโทเค็นของ Cloud Speech to Text และเก็บค่าของโทเค็นนั้น ๆ

- spaCy เลือกการติดแท็กเป็นจำนวน 6 ประเภท เช่นเดียวกับ NLTK จากนั้นใช้อัลกอริทึมของ spaCy ซึ่งสามารถวิเคราะห์ข้อความต่าง ๆ อัตโนมัติ สามารถเรียกดูค่าได้จากอัลกอริทึมได้ทันที และเลือกโทเค็นที่เป็นนิพจน์ระบุนาม จากนั้นเปลี่ยนประเภทนิพจน์ระบุนามให้เหมือนกันทุกไลบรารี และจับคู่โทเค็นที่แบบจำลองแบ่งเทียบกับโทเค็นของ Cloud Speech to Text และเก็บค่าของโทเค็นเหล่านั้น

*3.2.2 กระบวนการเลือกค่าทำนายจริง*

เลือกโทเค็นคำที่ไลบรารีทำนายประเภทของนิพจน์ระบุนามเหมือนกันตั้งแต่ 2 ไลบรารีขึ้นไป และ เลือกค่าทำนายจริงประเภท PERSON และ DATE ที่ spaCy ทำนายทันทีในกรณีที่ Stanford NER และ NLTK ทำนายประเภทไม่เหมือนกัน ซึ่งเงื่อนไขนี้ได้จากการทดลองที่ 2 ในหัวข้อที่ 4 คือ การทดลอง และผลลัพธ์ที่ได้จากระบบ เนื่องจากการใช้ไลบรารีเดียวอาจไม่แม่นยำมากพอที่จะทำนายประเภทของโทเค็นคำได้อย่าง จากนั้นเก็บค่าของโทเค็นคำกับประเภทของนิพนจ์ระบุนาม เพื่อนำไปวิเคราะห์ในขั้นตอนถัดไป ในที่นี้ จะแทนผลลัพธ์ที่ได้จากกระบวนการนี้ว่า “ค่าทำนายจริง”

*3.2.3 สร้างประเภทนิพจน์ระบุนามเพิ่มเพื่อติดแท็กเลขที่เป็นข้อมูลส่วนบุคคลโดยใช้ Regular Expressions และรวมกับค่าทำนายจริง*

นำค่าโทเค็นคำของ Cloud Speech to Text มาวิเคราะห์โดยสร้างเงื่อนไขเพื่อติดแท็กเฉพาะโทเค็นที่เป็นตัวเลขโดยใช้ Regular Expressions ในการตรวจสอบ และกำหนดประเภทของเลขที่เป็นข้อมูลส่วนบุคคลให้เป็นชื่อประเภท “PIINUM” เพื่อตรวจจับเลขข้อมูลส่วนบุคคล ดังนี้ เลขบัตรประชาชน (13 หลัก) เบอร์โทรศัพท์ (10 หลัก) เลขที่บัญชี (9 หลัก) เลขบัตรเดบิต หรือบัตรเครดิต (16 หลัก) และ เลขอื่น ๆ ที่มีจำนวนตั้งแต่ 9 หลักขึ้นไป

จากนั้นรวมค่าทำนายจริงกับค่าของเลขที่เป็นข้อมูลส่วนบุคคลเข้าด้วยกัน และเก็บค่านั้นไว้ในตาราง ซึ่งมีการเก็บค่าของตัวเลขที่เป็นข้อมูลส่วนบุคคลรวมกับค่าทำนายจริงเท่านั้น ไม่ได้มีการเก็บค่าเหล่านี้รวมกับไลบรารี 3 ไลบรารีที่ใช้ในการพัฒนา

*3.2.4 เก็บค่าต่าง ๆ ให้อยู่ในรูปของไฟล์ CSV*

หลังจากได้ผลลัพธ์การทำนายนิพจน์ระบุนาม จึงจัดเก็บค่าเหล่านั้นให้อยู่ในรูปแบบตารางและบันทึกเป็นไฟล์ CSV เป็นจำนวนทั้งหมด 5 คอลัมน์ ได้แก่ ลำดับโทเค็น (indx) โทเค็นคำ (word) เวลาที่เริ่มพูดโทเค็นนั้นในไฟล์เสียง (start\_time) เวลาที่สิ้นสุด (end\_time) และประเภทของนิพจน์ระบุนาม (real\_ents)

**3.3 การแทนที่เสียงที่เป็นข้อมูลส่วนบุคคลด้วยเสียงรบกวน**

นำค่าที่ได้จากไฟล์ CSV ในขั้นตอนก่อนหน้านี้ โดยเลือกใช้เพียงคอลัมน์เวลาเริ่มต้น (start\_time) และเวลาสิ้นสุดของคำนั้น (end\_time) จากนั้นแบ่งช่วงของเวลาเริ่มต้นและสิ้นสุด แปลงค่าของเวลาให้อยู่ในหน่วยของมิลลิวินาที และแทนที่เสียงรบกวนในช่วงเวลาที่ได้คำนวณไว้ พร้อมปรับเดซิเบลของเสียงรบกวนให้อยู่ในระดับที่เหมาะสม และบันทึกไฟล์เสียงที่มีการปิดบังข้อมูลส่วนบุคคลเป็นไฟล์เสียงประเภท .wav

**3.4 การประเมินผล (Evaluation)**

มีการประเมินผลกระบวนการทั้งหมด 2 กระบวนการ คือ ประเมินผลความแม่นยำของการแปลงเสียงพูดให้อยู่ในรูปแบบข้อความจากการเทียบจากข้อมูลจริงโดยประยุกต์ใช้แนวคิด Jaccard’s Coefficient Similarity และประเมินผลความแม่นยำของการตรวจจับคำที่เป็นข้อมูลส่วนบุคคลจากข้อมูลรูปแบบข้อความจากการสร้างผลเฉลยของการทำนายข้อความและโทเค็นต่าง ๆ เพื่อใช้ตรวจสอบความแม่นยำในการทำนายของไลบรารีทั้งหมดโดยวัดจากค่า Recall เท่านั้น

**4. การทดลอง และผลลัพธ์ที่ได้จากระบบ**

ผู้จัดทำได้แบ่งการแสดงผลการทดลองและผลลัพธ์ที่ได้เป็น 3 หัวข้อหลัก มีรายละเอียด ดังนี้

**4.1 การวิเคราะห์ข้อมูลที่สร้างขึ้น**

ในการเตรียมข้อมูลเกิดปัญหาที่ไม่สามารถหาชุดข้อมูลจากแหล่งข้อมูลสาธารณะมาพัฒนาระบบได้เนื่องจากข้อมูลเหล่านี้มีข้อจำกัดในเรื่องของข้อมูลส่วนบุคคล จึงต้องสร้างชุดข้อมูลขึ้นมาเอง โดยมีรายละเอียดการสร้างข้อมูล ดังนี้

*4.1.1 สร้างบทสนทนาระหว่างลูกค้าและศูนย์บริการข้อมูลลูกค้าทางโทรศัพท์*

สร้างชุดข้อมูลในรูปแบบข้อความเป็นจำนวนทั้งหมด 23 บทสนทนาเพื่อใช้ในการพัฒนาและประเมินผลระบบ จากการวิเคราะห์ประโยคในบทสนทนาคิดเป็น 566 ประโยค และใน 1 บทสนทนามีจำนวนเฉลี่ยทั้งหมด 24.61 ประโยค แบ่งย่อยลงไปเป็นการวิเคราะห์คำที่ยังไม่ผ่านการทำความสะอาดข้อมูลมีทั้งหมด 4,095 คำ โดยที่ใน 1 บทสนทนามีจำนวนเฉลี่ย 178.04 คำ และหากวิเคราะห์คำผ่านการทำความสะอาดข้อมูลแล้ว กล่าวคือ ดำเนินการตัดเครื่องหมายวรรคตอนและ Stop words บางส่วนออก มีทั้งหมด 1732 คำ และใน 1 บทสนทนามีจำนวนเฉลี่ย 75.3 คำ

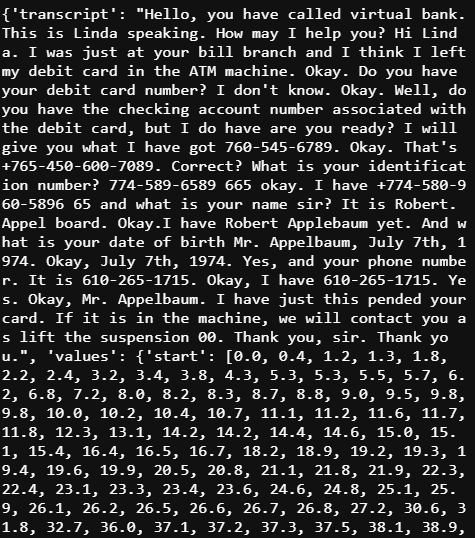
*4.1.2 นำข้อมูลในรูปแบบข้อความมาดำเนินการบันทึกเสียง*

เนื่องจากบทสนทนาที่สร้างขึ้นเป็นบทสนทนาภาษาอังกฤษ จึงต้องนำประโยคบทสนทนามาบันทึกเสียงโดยใช้เครื่องมือสำเร็จรูปในการแปลงข้อความให้อยู่ในรูปของเสียง (Text-to-Speech) โดยใช้ สิริ (Siri) ในการช่วยอ่านบทสนทนาเหล่านั้น ใน 1 บทสนทนาจะประกอบไปด้วยเสียงของพนักงานและลูกค้า โดยที่เสียงของพนักงานจะมีเพียงเพศเดียว คือ เพศหญิง โดยใช้เสียงของ “Siri Female” และในส่วนของเสียงลูกค้าจะแบ่งออกเป็น 2 เพศ ได้แก่ เพศหญิง ใช้เสียงของ “Samantha” และเพศชาย ใช้เสียงของ “Siri Male”

*4.1.3 แปลงประเภทของไฟล์เสียงบทสนทนา*

แปลงประเภทไฟล์เสียงจาก “.m4a” ให้อยู่ในประเภทไฟล์ “.wav” เพื่อที่จะสามารถนำไปประมวลผลกับ Cloud Speech to Text โดยแปลงไฟล์บนเว็บไซต์ที่ชื่อว่า “Convert MP4 to WAV” [11]

**4.2 การแปลงเสียงพูดให้อยู่ในรูปแบบข้อความ**

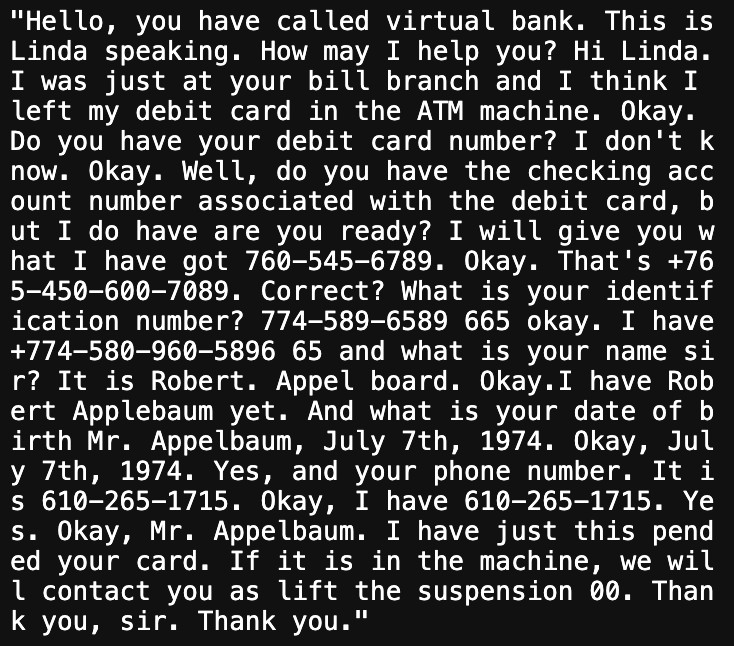


**รูปที่ 3.** ตัวอย่างการแปลงข้อมูลเสียงให้อยู่ในรูปแบบข้อความโดยใช้ Cloud Speech to Text

จากรูปที่ 3 แปลงให้อยู่ในรูปแบบของ Dictionary และสร้างคีย์ที่ชื่อว่า transcript เพื่อเก็บข้อความในบทสนทนาทั้งหมด ในส่วนของโทเค็นคำ ได้มีการสร้างคีย์ที่ชื่อว่า values เพื่อเก็บค่าของเวลาที่เริ่มพูดโทเค็นนั้น ๆ (start) เวลาที่พูดจบ (end) และโทเค็นนั้น ๆ (word) และประเมินผลความแม่นยำในการทำนายของแบบจำลอง โดยการนำข้อมูลบทสนทนาจริงเทียบกับข้อมูลที่แบบจำลองทำนายโดยใช้ Jaccard’s Coefficient Similarity ดังนี้



**รูปที่ 4.** ข้อมูลบทสนทนาจริง

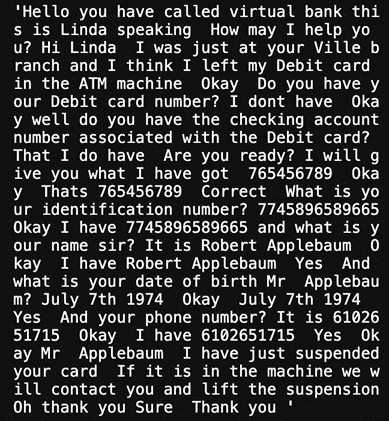


**รูปที่ 5.** บทสนทนาที่แบบจำลองทำนาย

**ตารางที่ 1.** ตารางแสดงค่าความแม่นยำของการแปลงเสียงพูดให้อยู่ในรูปของข้อความ

|  |  |
| --- | --- |
| Algorithm | Accuracy (%) |
| Cloud Speech to Text | 57.02 |

จากตารางที่ 1 ความแม่นยำในการทำนายคิดเป็นร้อยละ 57.02 ซึ่งเมื่อเทียบบทสนทนารูปที่ 4 และ 5 พบว่าสิ่งที่ส่งผลให้ค่าความแม่นยำของแบบจำลองไม่สูงนั้นส่วนใหญ่ขึ้นอยู่กับเครื่องหมายวรรคตอนของข้อมูลบทสนทนาทั้งสอง ดังนั้น จึงทำการตัดเครื่องหมายวรรคตอนของบทสนทนาทั้งสองออก เพื่อประเมินผลค่าความแม่นยำใหม่ ดังรูปที่ 6, 7 และ ตารางที่ 2



**รูปที่ 6.** ข้อมูลบทสนทนาจริงที่ผ่านการทำความสะอาด



**รูปที่ 7.** บทสนทนาที่แบบจำลองทำนาย

ที่ผ่านการทำความสะอาด

**ตารางที่ 2.** ตารางแสดงค่าความแม่นยำของการแปลงเสียงพูดให้อยู่ในรูปของข้อความ (ใหม่)

|  |  |
| --- | --- |
| Algorithm | Accuracy (%) |
| Cloud Speech to Text | 71.43 |

จากตารางที่ 2 ความแม่นยำในการทำนายคำพูดของแบบจำลองคิดเป็นร้อยละ 71.43 สามารถเห็นได้ชัดว่าค่าความแม่นยำสูงขึ้นอย่างชัดเจน เมื่อตัดเครื่องหมายวรรคตอนออกเบื้องต้น

**4.3 การตรวจจับคำที่เป็นข้อมูลส่วนบุคคลจากข้อมูลรูปแบบข้อความ**

*4.3.1 การทดลองทำนายคำที่เป็นนิพจน์ระบุนามของทั้ง 3 ไลบรารี*

ทดลองนำข้อมูลในรูปแบบข้อความที่เป็นผลลัพธ์จากการทำ Cloud Speech to Text เข้าสู่กระบวนการตรวจจับนิพจน์ระบุนามทั้ง 3 ฟังก์ชัน ของไลบรารี Stanford NER, NLTK และ spaCy จากนั้นแสดงผลลัพธ์ในรูปแบบตาราง ดังรูปที่ 8



**รูปที่ 8.** ตารางการทำนายประเภทของนิพจน์ระบุนาม

ของ 3 ไลบรารี

จากรูปที่ 8 แถวใดมีการทำนายเป็นคำว่า “O” คือ โทเค็นนั้นไม่ได้เป็นนิพจน์ระบุนาม ซึ่งมีการเก็บค่าการทำนายทั้งหมด 3 คอลัมน์ ได้แก่ stanford\_pred (ค่าที่ Stanford NER ทำนาย) nltk\_pred (ค่าที่ NLTK ทำนาย) spacy\_pred (ค่าที่ spaCy ทำนาย) นอกจากนี้ มีการประเมินผลการทำนายนิพจน์ระบุนามแต่ละประเภท เพื่อใช้ในการพิจารณาวิธีเลือกค่าทำนายจริง โดยมีการพิจารณาจากค่า Recall เท่านั้น แบ่งตามนิพจน์ระบุนาม 5 ประเภท ดังตารางต่อไปนี้

**ตารางที่ 3.** ตารางการแสดงค่าความแม่นยำของการติดแท็กคำว่า “PERSON”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Library | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
| Stanford NER | 100 | 76.7 | 86.8 | 99.2 |
| NLTK | 33.3 | 60 | 42.9 | 98.7 |
| spaCy | 93.8 | 100 | 96.8 | 100 |

**ตารางที่ 4.** ตารางการแสดงค่าความแม่นยำของการติดแท็กคำว่า “ORGANIZATION”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Library | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
| Stanford NER | 28.6 | 100 | 44.4 | 100 |
| NLTK | 0 | 0 | 0 | 99.8 |
| spaCy | 33.3 | 100 | 50 | 100 |

**ตารางที่ 5.** ตารางการแสดงค่าความแม่นยำของการติดแท็กคำว่า “LOCATION”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Library | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
| Stanford NER | 40 | 33.3 | 36.4 | 99.6 |
| NLTK | 20 | 33 | 25 | 99.6 |
| spaCy | 0 | 0 | 0 | 99.3 |

**ตารางที่ 6.** ตารางการแสดงค่าความแม่นยำของการติดแท็กคำว่า “DATE”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Library | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
| Stanford NER | 47.1 | 88.9 | 61.5 | 99.9 |
| NLTK | 0 | 0 | 0 | 99 |
| spaCy | 52.9 | 100 | 69.2 | 100 |

**ตารางที่ 7.** ตารางการแสดงค่าความแม่นยำของการติดแท็กคำว่า “MONEY”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Library | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
| Stanford NER | 100 | 100 | 100 | 100 |
| NLTK | 0 | 0 | 0 | 99.6 |
| spaCy | 100 | 100 | 100 | 100 |

จากการวิเคราะห์ตารางค่าความแม่นยำแต่ละประเภทพบว่าทั้ง 3 ไลบรารีมีค่า Recall ในการทำนายที่สูงแตกต่างกัน จึงมีการทดลองเลือกค่าทำนายจริงจากการใช้แนวคิดของ Majority Voting โดยเลือกค่านิพจน์ระบุนามที่ทำนายเหมือนกันตั้งแต่ 2 ไลบรารีขึ้นไปเป็นค่าทำนายจริง ได้ผลลัพธ์การประเมินผลดังตารางที่ 8

* การทดลองที่ 1

**ตารางที่ 8.** ตารางการแสดงค่าความแม่นยำของการทำนายนิพจน์ระบุนาม 5 ประเภทจากการทดลองที่ 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Library | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
| Stanford NER | 78 | 52 | 62.4 | 94.7 |
| NLTK | 31.2 | 26.7 | 28.8 | 88.9 |
| spaCy | 81.8 | 60 | 69.2 | 95.5 |
| Combined | 95.7 | 58.7 | 72.7 | 96.3 |

จากตารางที่ 8 ค่า Recall ที่ได้จากการทดลองที่ 1 (Combined) มีค่าต่ำกว่า spaCy ซึ่งเมื่อลองกลับไปพิจารณาค่า Recall ของ spaCy เป็นรายประเภท พบว่า ประเภท PERSON และ DATE ไลบรารีมีค่า Recall สูงกว่าไลบรารีอื่น ๆ จึงส่งผลให้เมื่อใช้แนวคิดของ Majority Voting กับทุกประเภท อาจทำให้ค่า Recall ของค่าทำนายจริงต่ำกว่า spaCy ได้ ดังนั้น จึงมีการทดลองอีกครั้งโดยใช้ Majority Voting และเลือกนิพจน์ระบุนามประเภท PERSON และ DATE เป็นค่าทำนายจริงทันที เมื่อ Stanford NER และ NLTK ไม่มีการทำนายประเภทที่เหมือนกัน แต่ spaCy ทำนายนิพจน์ระบุนามออกมาเป็น 2 ประเภทนั้น ดังการทดลองที่ 2

* การทดลองที่ 2

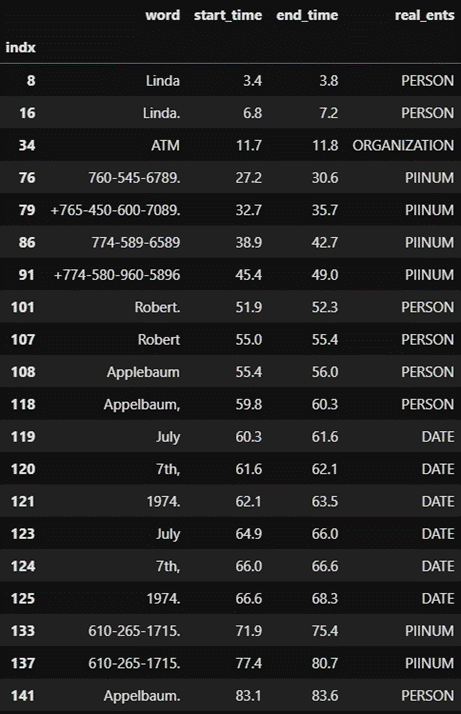
**ตารางที่ 9.** ตารางการแสดงค่าความแม่นยำของการทำนายนิพจน์ระบุนาม 5 ประเภทจากการทดลองที่ 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Library | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
| Stanford NER | 78 | 52 | 62.4 | 94.7 |
| NLTK | 31.2 | 26.7 | 28.8 | 88.9 |
| spaCy | 81.8 | 60 | 69.2 | 95.5 |
| Combined | 82.5 | 62.7 | 71.2 | 95.7 |

จากการทดลองที่ 2 ถือว่าการทดลองนี้เป็นผลสัมฤทธิ์ เนื่องจากมีค่า Recall สูงที่สุด ซึ่งคิดเป็นร้อยละ 62.7 ดังนั้น ผู้จัดทำจึงเลือกวิธีการจากการทดลองที่ 2 มาใช้ในการพัฒนากระบวนการของการตรวจจับนิพจน์ระบุนามในรูปแบบข้อความ

*4.3.2 นำค่าทำนายจริงรวมกับค่าทำนายของเลขที่เป็นข้อมูลส่วนบุคคล*

นำค่าทำนายจริงที่ได้จากการทดลองที่เป็นผลสัมฤทธิ์แล้วมารวมกับค่าทำนายของเลขที่เป็นข้อมูลส่วนบุคคล จากนั้นเก็บค่าเหล่านั้นให้อยู่ในรูปแบบตารางโดยเลือกเพียงโทเค็นที่มีการทำนายว่าเป็นนิพจน์ระบุนามเท่านั้น ดังรูปที่ 9



**รูปที่ 9.** ตารางค่าทำนายจริงทั้งหมด

**ตารางที่ 10.** ตารางการแสดงค่าความแม่นยำของการทำนายนิพจน์ระบุนามทุกประเภท

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Library | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
| Stanford NER | 78 | 52 | 62.4 | 94.7 |
| NLTK | 31.2 | 26.7 | 28.8 | 88.9 |
| spaCy | 81.8 | 60 | 69.2 | 95.5 |
| Combined | 82.5 | 62.7 | 71.2 | 95.7 |
| Combined and Regex | 87.3 | 92 | 89.6 | 98.2 |

จากตารางที่ 10 เมื่อรวมค่าทำนายจริงกับค่าของเลขที่เป็นข้อมูลส่วนบุคคลเข้าด้วยกัน (Combined and Regex) มีค่า Recall สูงถึงร้อยละ 92 ซึ่งถือว่าการทดลองพัฒนาระบบที่ได้คิดค้นขึ้นนั้นเป็นผลสัมฤทธิ์

**5. บทสรุป และการพัฒนาต่อ**

**5.1 สรุปผลโครงงาน**

ผู้จัดทำได้ทดลองสร้างชุดข้อมูลขึ้นเองทั้งรูปแบบข้อความและรูปแบบเสียงเนื่องด้วยปัญหาที่ไม่สามารถหาข้อมูลจากแหล่งข้อมูลสาธารณะได้ ซึ่งเนื้อหาของบทสนทนาประกอบด้วย ชื่อ - นามสกุล เลขที่บัญชี เลขบัตรเดบิต หรือ เครดิต เลขบัตรประชาชน วันเกิด ที่อยู่ และเบอร์โทรศัพท์

การแปลงเสียงพูดให้อยู่ในรูปแบบข้อความนั้น เมื่อทำการตัดเครื่องหมายวรรคตอน ถือว่าค่าของความแม่นยำอยู่ในระดับที่ดี แต่มีการแปลงคำผิดพลาด สาเหตุมาจากเสียงที่ใช้ในการบันทึกเสียงแต่ละบุคคลมีสำเนียงการพูดไม่เหมือนกัน เช่น นามสกุล Applebaum เมื่อเป็นเสียงของ Siri Male แบบจำลองแปลงได้เป็น 2 โทเค็น คือ “Appel” และ “board.” แต่เมื่อเป็นเสียงของ Siri Female แบบจำลองกลับแปลงคำได้ถูกต้อง และอีกกรณี คือ เมื่อสิริพูดว่า “oh” แบบจำลองจะแปลงเป็นเลข “0” ซึ่งส่งผลให้ความแม่นยำลดลง

การตรวจจับคำที่เป็นข้อมูลส่วนบุคคลจากข้อมูลรูปแบบข้อความ ได้ทำการทดลอง ทั้งหมด 3 ไลบรารี

- Stanford NER สามารถติดแท็กบุคคล และค่าเงินได้ค่อนข้างแม่นยำ แต่ในการติดแท็กวันที่ ด้วยข้อจำกัดที่ไม่มีการติดแท็กตัวเลขที่เป็นประเภท Cardinal จึงส่งผลให้มีการติดแท็กตัวเลขธรรมดา เป็นประเภทของวันที่ (Date) ทำให้ความแม่นยำลดลง

- NLTK สามารถติดแท็กบุคคลได้ในเกณฑ์ที่ดี แต่ในส่วนของนิพจน์ระบุนามอื่นนั้นแทบจะไม่สามารถติดแท็กได้อย่างถูกต้อง แต่การติดแท็กสถานที่ NLTK สามารถติดแท็กได้แม่นยำเท่ากับ Stanford NER ซึ่งเมื่อมีการใช้แนวคิดของ Majority Voting จึงสามารถทำให้การติดแท็กสถานที่นั้นมีความแม่นยำกว่าเดิม สาเหตุที่ส่งผลให้ NLTK มักมีการติดแท็กที่ผิดพลาด คือ หากโทเค็นนั้น ๆ ขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่ เช่น คำว่า “Hello” ไลบรารีจะติดแท็กเป็นสถานที่ทันที

- spaCy จากผลลัพธ์การประเมินผลความแม่นยำ จะสังเกตได้ว่าส่วนใหญ่แล้ว spaCy มักมีค่าความแม่นยำสูงในการติดแท็กโทเค็น แต่หากให้สรุปเป็นรายประเภท จะสามารถสรุปได้ว่า ไลบรารีนี้สามารถติดแท็กบุคคล วันที่ และค่าเงินได้ดีที่สุด แต่เนื่องจากการติดแท็กของไลบรารีนี้ยังมีความไม่แม่นยำอยู่บ้าง ทางผู้จัดทำจึงมีความเห็นว่าควรมีการเลือกค่าทำนายจริงโดยใช้วิธีการจากการทดลองที่ 2 ในการพัฒนาระบบเพื่อเพิ่มประสิทธิภาพในการติดแท็กนิพจน์ระบุนาม

การตรวจจับข้อมูลส่วนบุคคลทั้ง 3 ไลบรารี ยังไม่สามารถตรวจจับข้อมูลของเลขบ้านเลขที่ให้เป็นประเภทของ LOCATION ได้ จึงไม่สามารถปกปิดบ้านเลขที่ในขั้นตอนสุดท้ายได้ แต่ในส่วนของการนำค่าทำนายจริงโดยใช้วิธีการจากการทดลองที่ 2 มารวมกับการทำนายเลขที่เป็นข้อมูลส่วนบุคคลนั้น มีความแม่นยำค่อนข้างสูง ซึ่งค่าของ Recall คิดเป็นร้อยละ 92 ถือเป็นค่าความแม่นยำที่น่าพึงพอใจ

การตรวจจับเลขที่เป็นข้อมูลส่วนบุคคลโดยใช้ Regular Expressions ก็มีความแม่นยำค่อนข้างสูงเช่นกัน แต่ในบางครั้งอาจไม่แม่นยำอย่างสมบูรณ์เนื่องจากรูปแบบการแปลงตัวเลขของ Cloud Speech to Text อาจแบ่งโทเค็นได้ไม่ตรงกับตัวเลขที่ควรจะเป็น แต่หากพิจารณาถึงภาพรวมของค่าความแม่นยำแล้ว ถือเป็นที่น่าพึงพอใจ

การแทนที่เสียงที่เป็นข้อมูลส่วนบุคคลด้วยเสียงรบกวน ในบางช่วงที่มีการแทนที่เสียงรบกวนนั้นอาจเกินหรือขาดจากระยะเวลาของคำพูดที่เป็นข้อมูลส่วนบุคคล แต่โดยภาพรวมแล้วถือว่าปิดบังคำพูดที่เป็นข้อมูลส่วนบุคคลได้ดี

**5.2 แนวทางในการพัฒนาต่อ**

1) โครงงานนี้สามารถต่อยอดโดยการนำระบบที่พัฒนาไปใช้ในด้านของการปกปิดข้อมูลที่เป็นส่วนบุคคลในหน่วยงานและองค์กรที่ต้องการประยุกต์ใช้ระบบได้ทั้งในรูปแบบไฟล์เสียง และข้อมูลที่เป็นข้อความ

2) สามารถนำผลลัพธ์ที่ได้จากการพัฒนาระบบไปวิเคราะห์และพัฒนาต่อในด้านอื่น ๆ ได้ โดยไม่ต้องคำนึงถึงสิทธิส่วนบุคคลเนื่องจากมีการปกปิดข้อมูลส่วนบุคคลแล้ว

**เอกสารอ้างอิง**

[1] IBM. “What is Speech Recognition?”. [Online].

Available: <https://www.ibm.com/cloud/learn/speech-recognition>. 2020.

[2] F. Sillawat. “การเปิดใช้งาน Cloud Speech API”.

[Online]. Available: <https://bit.ly/3orAjIe>.

2018.

[3] รัฐภูมิ ตันสุตะพานิช. “**การสกัดความสัมพันธ์ระหว่างนิพจน์ระบุนามในภาษาไทย**”. (วิทยานิพนธ์ปริญญาวิทยาศาสตร์มหาบัณฑิตสาขาวิชาวิทยาการคอมพิวเตอร์, 2552).

[4] ศุภวัจน์ แต่รุ่งเรือง. “**การตรวจเทียบภายนอกหาการลักลอกในงานวิชาการโดยใช้แบบจำลองซัพพอร์ตเวกเตอร์แมชชีนและการวัดค่าความละม้ายของข้อความ**”. (วิทยานิพนธ์ปริญญาอักษรศาสตรดุษฎีบัณฑิตจุฬาลงกรณ์มหาวิทยาลัย, 2560).

[5] Y. Tang, Y., Q. Zhang, N. V Chawla, และ S. Krasse.

“**Correspondence SVMs Modeling for Highly**

**Imbalanced Classification**”. พิมพ์ครั้งที่ 1.

: CYBERNETICS.

[6] A. Geitgey. “Natural Language Processing is Fun!”. [Online]. Available: <https://bit.ly/36Vpsk8>.

2018.

[7] C. Dishmon. “Named Entity Recognition with Stanford NER Tagger”. [Online]. Available: <https://bit.ly/3lXHvKB>. 2020.

[8] “Natural Language Toolkit”. [Online]. Available:

<https://www.nltk.org/>. 2020.

[9] “spaCy 101: Everything you need to know”. [Online]. Available: <https://spacy.io/usage/spacy-101>. 2020.

[10] R. Cox. “Regular Expression Matching Can Be Simple And Fast (but is slow in Java, Perl, PHP, Python, Ruby, ...)”. [Online]. Available: <https://swtch.com/~rsc/regexp/regexp1.html>.

2007.

[11] “MP4 to WAV online file converter”. [Online].

Available: https://audio.online-convert.com/convert/mp4-to-wav.