การประยุกต์ใช้โมเดล Multinomial Naïve Bayes ในการตรวจจับคำพูดภาษาไทยที่เป็นภัยจากข้อความของระบบรู้จำเสียงพูดของ Google

Detection of Harmful Thai Speech from Google Speech Recognition Text Using Multinomial Naïve Bayes Automatic System for Room Acoustic Adjustment

ศิลป์ศรุต ทิมจ้อย[1] เนตรชนก รอดอินทร์[2]

[1]ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร rotain_n@su.ac.th

บทคัดย่อ

วิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาและออกแบบระบบตรวจจับคำพูดภาษาไทยที่เป็นภัย โดยมุ่งเน้นให้สามารถใช้ตามสถานที่ที่สำคัญหรือต้องการความปลอดภัยสูง เพื่อรักษาความปลอดภัยที่แม่นยำและทันต่อเหตุการณ์ ระบบจะเริ่มต้นจากการรับเสียงผ่านไมโครโฟน พร้อมปรับค่าความไวของระบบ ซึ่งช่วยคำนวณระดับเสียงรบกวนพื้นหลังและตั้งค่า Threshold สำหรับแยกเสียงพูดออกจากสิ่งรบกวนได้แม่นยำยิ่งขึ้น จากนั้นระบบจะทำการแปลงเสียงพูดเป็นข้อความ โดยใช้การแปลงเสียงเป็นข้อความของ Google ข้อความที่ได้จะผ่านกระบวนการประมวลผลภาษาด้วยการตัดคำภาษาไทยและลบคำฟุ่มเฟือย ก่อนแปลงเป็นเวกเตอร์ด้วยเทคนิค TF-IDF และจำแนกข้อความด้วยโมเดล Multinomial Naïve Bayes ซึ่งได้รับการปรับค่าพารามิเตอร์ Alpha โดยใช้วิธี Grid search เพื่อเพิ่มประสิทธิภาพในการจำแนก ระบบนี้พัฒนาด้วยภาษา Python พร้อมเทคโนโลยีเว็บ ได้แก่ HTML CSS และ JavaScript และรองรับการเชื่อมต่อแบบเรียลไทม์ด้วย Flask และ Socket โดยผลลัพธ์ของระบบจะแสดงผ่านหน้าเว็บไซต์ เพื่อแจ้งเตือนกรณีตรวจพบคำพูดที่เป็นภัยอย่างรวดเร็ว สำหรับผลการประเมินประสิทธิภาพของโมเดล พบว่าให้ ความแม่นยำ (Accuracy) เท่ากับ 0.917 สะท้อนถึงอัตราการทำนายถูกต้องโดยรวมของระบบ ในขณะที่ Precision เท่ากับ 0.862 หมายถึงอัตราที่โมเดลทำนายว่าเป็นภัยและถูกต้องจริง ส่วน Recall เท่ากับ 0.962 แสดงถึงความสามารถของโมเดลในการจับข้อความที่เป็นภัยได้อย่างครอบคลุม และสุดท้าย F1-score เป็นค่าเฉลี่ยถ่วงน้ำหนักระหว่าง Precision และ Recall อยู่ที่ 0.909 แสดงถึงความสมดุลของระบบในการวิเคราะห์ภัยได้อย่างแม่นยำ

คำสำคัญ: การตรวจจับคำพูดที่เป็นภัย การแปลงเสียงเป็นข้อความของ Google โมเดล Multinomial Naïve Bayes เทคนิค TF-IDF การแจ้งเตือนแบบเรียลไทม์

Abstract

This research aims to develop and design a harmful speech detection system, focusing on deployment in critical or high-security areas to ensure accurate and timely safety responses. The system begins by capturing audio input through a microphone and adjusting its sensitivity using a Threshold that is calculated based on ambient noise levels. This is done to accurately distinguish spoken words from background sounds. The speech is then transcribed into text using Google Speech Recognition. The resulting text undergoes preprocessing, including Thai word segmentation and stopword removal, followed by vectorization using the TF-IDF technique. The processed data is then classified using the Multinomial Naïve Bayes model, with the Alpha parameter optimized via Grid search to enhance classification performance. The system is primarily developed using Python, with web technologies such as HTML, CSS, and JavaScript, and supports real-time communication through Flask and Socket. Detection results are displayed through a web interface, providing rapid alerts when harmful speech is identified. The system's classification performance was evaluated using four standard metrics: accuracy, precision, recall, and F1-score. The model achieved an accuracy of 0.917, indicating a high overall correctness of classification. A precision of 0.862 reflects the proportion of predicted harmful speech that was actually harmful. The recall score of 0.962 demonstrates the model's strong ability to correctly identify most harmful speech instances. The F1-score, which balances precision and recall, reached 0.909, highlighting the system's effectiveness in providing both reliable and comprehensive detection capabilities.

Keywords: Harmful Speech Detection, Speech Recognition, Multinomial Naïve Bayes Model, TF-IDF Technique, Real-time Alert System

1. บทนำ

ในสถานที่หรือพื้นที่ที่ต้องการยกระดับความปลอดภัย จำเป็นต้องมีการเฝ้าระวังภัยโดยอาจเกิดขึ้นจากการใช้คำพูด จึงต้องมีการพัฒนาระบบที่สามารถตรวจจับคำพูดที่เป็นภัยเพื่อช่วยเพิ่มประสิทธิภาพของมาตรการรักษาความปลอดภัย โดยเฉพาะในพื้นที่ที่มีผู้คนหนาแน่นหรือมีความเสี่ยงสูงในการเกิดเหตุการณ์ไม่คาดคิด หนึ่งในแนวทางที่สำคัญคือ การนำเทคโนโลยีการแปลงเสียงเป็นข้อความ (Speech recognition) มาใช้ในการแปลงเสียงเป็นข้อความ เป็นขั้นตอนเริ่มต้นที่สำคัญของระบบ เพื่อให้สามารถวิเคราะห์และตรวจจับคำ โดยในงานวิจัยนี้เลือกใช้การแปลงเสียงเป็นข้อความของ Google ที่รองรับภาษาไทยได้อย่างมีประสิทธิภาพ [1] หลายงานวิจัยได้แสดงให้เห็นถึงความสามารถของโมเดล Multinomial

Naïve Bayes (MNB) ในการจำแนกข้อความ โดยเฉพาะในบริบทของการตรวจจับข้อความที่เข้าข่ายว่าเป็นภัย ซึ่งมีความแม่นยำและใช้งานง่ายเหมาะกับข้อมูลเชิงข้อความ (Text data) ที่มีโครงสร้างไม่ซับซ้อน [2,3] ทั้งนี้ การวิเคราะห์ข้อความที่มีคำลักษณะเป็นภัยแต่ไม่ได้เป็นภัยจริง ยังคงเป็นความท้าทายสำคัญ โดยเฉพาะกรณีที่คำบางคำปรากฏในบริบทปลอดภัย เช่น คำว่า“ตก” หรือในบริบท “ตกเครื่อง” และคำว่า “ไฟไหม้” ในการฝึกซ้อม ซึ่งอาจทำให้เกิดการแจ้งเตือนผิดพลาด (False positive) งานวิจัยหลายฉบับจึงเลือกใช้เทคนิค TF-IDF (Term Frequency–Inverse Document Frequency) ในการแปลงข้อความเป็นเวกเตอร์ โดยเน้นคำที่สำคัญในบริบท และลดน้ำหนักคำทั่วไป ช่วยให้โมเดลสามารถจำแนกข้อความได้แม่นยำมากยิ่งขึ้น [4]

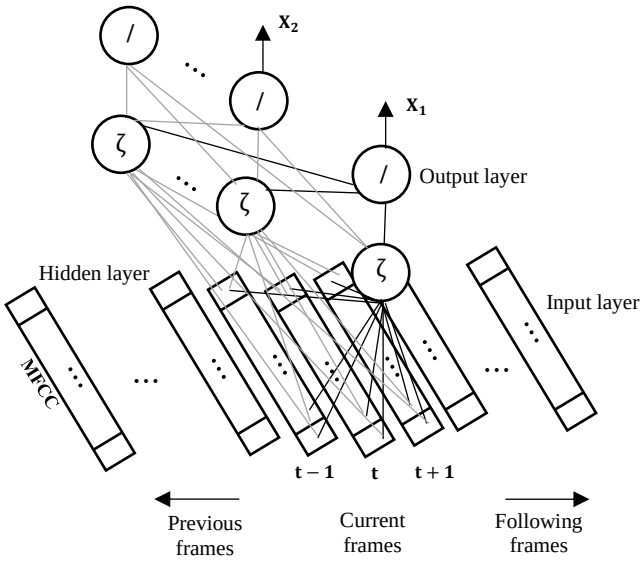
งานวิจัยนี้จึงนำเสนอการพัฒนาและออกแบบระบบตรวจจับคำพูดภาษาไทยที่เป็นภัย โดยผสานเทคนิคด้าน การแปลงเสียงเป็นข้อความ การวิเคราะห์ข้อความ และการแจ้งเตือนแบบเรียลไทม์ผ่านหน้าเว็บไซต์

2. ทฤษฎีที่เกี่ยวข้อง

2.1 การรู้จำและแปลงเสียงเป็นข้อความของ Google

การรู้จำเสียง (Speech Recognition) เป็นเทคโนโลยีที่พัฒนาโดย Google มีความสามารถในการแปลงเสียงพูดให้เป็นข้อความได้อย่างแม่นยำและรวดเร็ว รองรับภาษาไทย ถูกนำมาใช้ในหลายบริบทการสั่งงานด้วยเสียง (Voice command) ไปจนถึงการพัฒนาแอปพลิเคชันที่รองรับการโต้ตอบด้วยเสียงแบบเรียลไทม์ (Real-time Speech-to-Text) [1] ระบบนี้ถูกนำมาใช้ในงานวิจัยของ Gevaert et al. (2010) เพื่อถอดเสียงพูดที่บันทึกได้จากไมโครโฟน ก่อนเข้าสู่กระบวนการประมวลผลภาษาธรรมชาติต่อไป

โครงสร้างการแปลงเสียงเป็นข้อความประกอบด้วยหลายขั้นตอนที่ประมวลผลจากคลื่นเสียง เริ่มจากการบันทึกเสียง (Audio Capture) ระบบรับคลื่นเสียงจากไมโครโฟน โดยมีการใช้อุปกรณ์แปลงสัญญาณ (AD Converter) เพื่อเปลี่ยนเสียงเป็นข้อมูลดิจิทัล อาจได้รับผลกระทบจากสัญญาณรบกวนภายนอก เช่น เสียงแวดล้อม หรือเสียงสะท้อนภายในห้อง แล้วแปลงคลื่นเสียงเป็นสัญญาณดิจิทัล (Signal Processing) เพื่อเตรียมเข้าสู่กระบวนการประมวลผล ข้อมูลเสียงจะผ่านการกรองด้วยดิจิทัลฟิลเตอร์เพื่อลดเสียงรบกวน และทำ Bandpass filtering ในช่วง 300–3750 Hz เป็นช่วงที่มนุษย์สามารถพูดได้อย่างชัดเจน และแยกพยางค์หรือคำออกจากกัน กระบวนการแปลงเสียงเป็นข้อความ โดยทั่วไปจะเริ่มจาก การแยกลักษณะของเสียงพูด (Feature extraction) เป็นการดึงลักษณะเฉพาะของเสียง เช่น Spectrogram และ Mel Frequency Cepstral Coefficients (MFCC) ซึ่งเป็นเทคนิคยอดนิยมที่สามารถสกัดคุณลักษณะเสียงได้แม่นยำ และใช้เป็นอินพุตให้กับแบบจำลอง Neural Network และการเปลี่ยนแปลงของเสียงในช่วงเวลาออกมาเป็นชุดของคุณลักษณะ จากนั้นจะเข้าสู่ขั้นตอนแบบจำลองเสียงพูด (Acoustic Modeling) ด้วยโครงข่ายประสาทเทียม (Neural Networks) งานวิจัยได้ใช้งานโครงข่ายประสาทเทียมแบบ Feedforward Neural Network ร่วมกับ Backpropagation Algorithm เพื่อฝึกให้ระบบสามารถจดจำคำพูดที่ฝึกไว้ล่วงหน้าได้ โดยอินพุตที่ป้อนเข้าสู่โครงข่ายจะเป็นค่าของ MFCC ที่ผ่านการลดมิติแล้ว เพื่อเพิ่มความเร็วและความแม่นยำในการจำแนกเสียง [6] ตามที่แสดงในรูปที่ 1 เช่น แบบจำลองเสียงพยัญชนะและสระ เพื่อตัดสินว่าเสียงนั้นใกล้เคียงกับคำใดในภาษาที่ใช้งานอยู่ การรู้จำคำและข้อความ (Language Modeling) เช่น ไวยากรณ์หรือโครงสร้างประโยค เพื่อให้ได้ข้อความที่สมบูรณ์และถูกต้องตามบริบท สุดท้ายสร้างข้อความผลลัพธ์ (Output Text) ข้อความที่ผ่านการวิเคราะห์และตรวจสอบแล้ว จะถูกแสดงหรือส่งไปยังระบบอื่นเพื่อดำเนินการต่อ [5]



รูปที่ 1 การจำแนกเสียงด้วยโครงข่ายประสาทเทียมแบบ Feedforward Neural Network [6]

2.2 โมเดล Multinomial Naïve Bayes

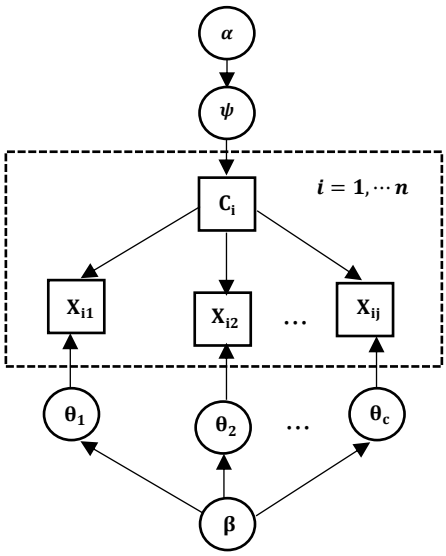
โมเดล Multinomial Naïve Bayes (MNB) เป็นหนึ่งในอัลกอริทึมที่ได้รับความนิยมมากที่สุดในตระกูล Naïve Bayes โดยเฉพาะอย่างยิ่งในงานที่เกี่ยวข้องกับ การประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) เช่น การวิเคราะห์ความคิดเห็น (Sentiment analysis), การตรวจจับถ้อยคำแสดงความเกลียดชัง (Hate speech detection) และ การตรวจจับภาษา (Language detection) เนื่องจากโมเดลมีโครงสร้างที่เรียบง่าย ความเร็วในการประมวลผลสูง และให้ผลลัพธ์ที่ดีแม้กับชุดข้อมูลขนาดเล็กหรือขนาดใหญ่ [2,4,5]

หลักการของโมเดลนี้อิงตามทฤษฎีของ Bayes โดยใช้สมการ:

$$P(C_i|X) = \frac{P(X|C_i) \cdot P(C_i)}{P(X)} \qquad (1)$$

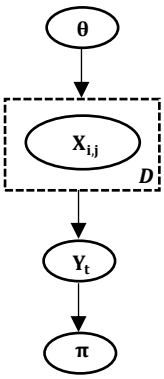
แสดงถึงการคำนวณความน่าจะเป็นของคลาส C_i ภายใต้ข้อมูลที่สังเกตได้ของ X โดยมีความหมายของแต่ละพจน์ดังนี้ $P(C_i|X)$ คือ ความน่าจะเป็นของคลาส C_i เมื่อทราบข้อมูลของ X และ $P(X|C_i)$ คือ ความน่าจะเป็นของข้อมูล X เมื่อทราบคลาส C_i และ $P(C_i)$ คือ ความน่าจะเป็นล่วงหน้าของคลาส C_i และสุดท้าย $P(X)$ คือ ความน่าจะเป็นของข้อมูล X ที่จะเหมือนกันในทุกคลาส [2]

ในการฝึกโมเดล Multinomial Naïve Bayes งานวิจัยของ Tiwari และ Agrawal (2022) แสดงให้เห็นว่าจุดเด่นของโมเดล MNB คือ สามารถทำงานได้รวดเร็วกว่าระบบที่ซับซ้อน โดยนำมาใช้ร่วมกับเทคนิค TF-IDF เพื่อแปลงข้อความเป็นเวกเตอร์ และเทคนิคการปรับพารามิเตอร์แบบ Grid search ทำการสุ่มเพื่อค้นหาค่าที่เหมาะสมที่สุดของพารามิเตอร์ α (Alpha) เป็นค่าคงที่ที่ใช้สำหรับการ Smoothing ใน Naïve Bayes โดยอาศัยเทคนิค Cross-validation [2,4] จากสมการที่ (1) ระบบจะใช้ข้อมูลตัวอย่างที่มีการระบุคลาสไว้ล่วงหน้า (label) เรียนรู้ความสัมพันธ์ระหว่างคำและคลาสเป้าหมาย โดยมีการสมมุติให้ฟีเจอร์แต่ละตัวเป็นอิสระต่อกัน เพื่อให้ง่ายต่อการคำนวณ ตามที่แสดงในรูปที่ 2



รูปที่ 2 โครงสร้างการฝึกโมเดลของ Naïve Bayes [3]

การทดสอบโมเดลของ Naïve Bayes ระบบจะรับข้อความที่แปลงมาจากเสียงพูด แล้วใช้โมเดลที่ผ่านการฝึกมาแล้วในการทำนายว่าข้อความนั้นเป็นภัยหรือไม่ โดยใช้ตัวแปรเดียวกัน ตามที่แสดงในรูปที่ 3



รูปที่ 3 โครงสร้างการทดสอบโมเดลของ Naïve Bayes [2]

3. การพัฒนาและออกแบบระบบตรวจจับคำพูดภาษาไทยที่เป็นภัยแบบเรียลไทม์

เนื้อหาเกี่ยวกับการออกแบบระบบรวมทั้งการแทรกรูปแสดงภาพรวมระบบที่ได้ออกแบบที่ประกอบด้วย ชุดข้อมูลและรายการคำที่เป็นภัย กระบวนการแปลงเสียงเป็นข้อความ การประมวลผลด้วยโมเดล Multinomial Naïve Bayes ภาพรวมการทำงานของระบบตรวจจับคำพูดที่เป็นภัย เครื่องมือและเทคโนโลยีที่ใช้ และการแสดงผลการตรวจพบคำพูดที่เป็นภัย ตามที่แสดงในรูปที่ 4

รูปที่ 4 แผนภาพแสดงกระบวนการทำงานของระบบตรวจจับคำพูดภาษาไทยที่เป็นภัยแบบเรียลไทม์

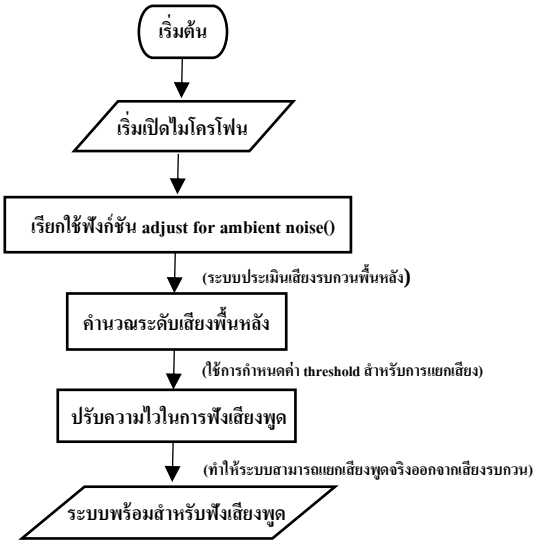
3.1 ชุดข้อมูลและรายการคำที่เป็นภัย

ระบบจะเริ่มต้นจากการเตรียมชุดข้อมูลตัวอย่างที่ประกอบด้วยข้อความภาษาไทยจำนวน 300 ประโยค แบ่งออกเป็น 2 กลุ่ม ได้แก่ ข้อความที่เข้าข่ายเป็นภัย กำหนดให้ (label = 1) เช่น "ระเบิด", "ตกกลางอากาศ" และข้อความที่ไม่เป็นภัย กำหนดให้ (label = 0) เช่น "ขอให้เดินทางโดยสวัสดิ์ภาพ", "เขาพูดถึงระเบิดในภาพยนตร์" ข้อความเหล่านี้ถูกเก็บไว้ในรูปแบบ .csv และผ่านการคัดเลือกให้มีความยาวครอบคลุมทั้งข้อความที่มีเนื้อหาน่ากังวล หรือข้อความที่มีคำสุ่มเสี่ยง แต่อยู่ในบริบทปลอดภัย เพื่อให้โมเดลสามารถเรียนรู้ จำแนกข้อความได้อย่างแม่นยำ ลดความผิดพลาดจากการแจ้งเตือนเกินจริง (False positive)

นอกจากนี้ยังมีรายการคำที่เป็นภัย (Abusive words) เป็นฐานข้อมูลของคำหรือวลีที่เกี่ยวข้องกับภัย ความรุนแรง หรืออันตราย จำนวน 145 คำ เช่น "ก่อการร้าย", "จี้", "ฆ่า", "ข่มขู่", "วางระเบิด" ถูกใช้เพื่อตรวจสอบคำ

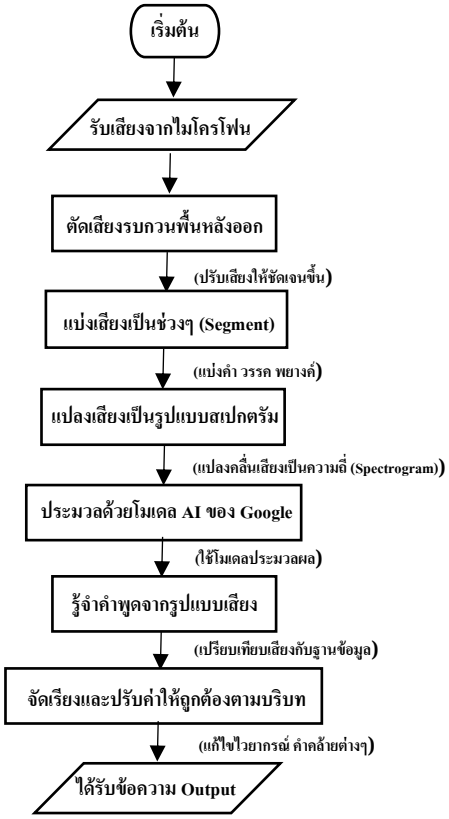
3.2 กระบวนการแปลงเสียงเป็นข้อความ

ระบบตรวจจับคำพูดที่เป็นภัยเริ่มต้นด้วยการรับเสียงผ่านไมโครโฟน เสียงที่ได้จะผ่านขั้นตอนการปรับค่าความไวของระบบจดจำเสียง เพื่อคำนวณระดับเสียงรบกวนพื้นหลัง (Ambient noise) และตั้งค่า Threshold ให้เหมาะสม ตามที่แสดงในรูปที่ 5 ช่วยให้ระบบสามารถตรวจจับเสียงพูดจริงได้แม้ในสภาพแวดล้อมที่มีเสียงรบกวนอยู่บ้าง



รูปที่ 5 ขั้นตอนการเตรียมเสียง (Threshold Calibration)

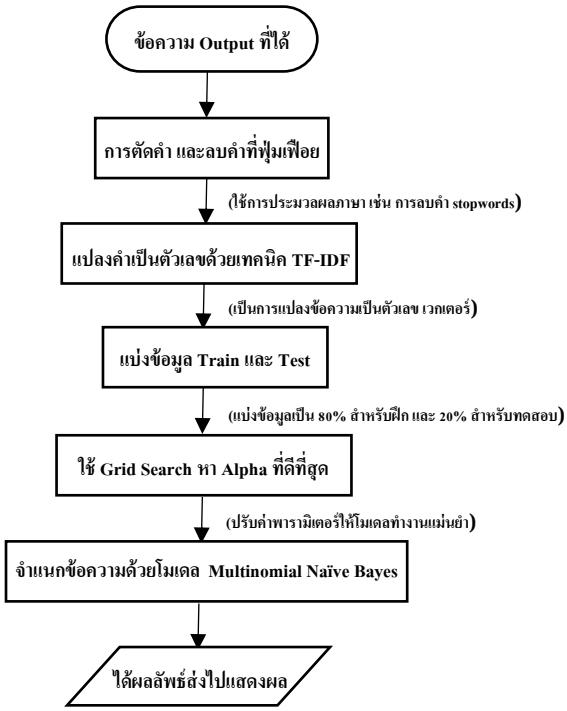
หลังจากระบบพร้อมรับเสียงพูดแล้ว ข้อมูลเสียงจะถูกตัดแบ่ง (segment) ออกเป็นช่วงๆ และแปลงเป็นรูปแบบสเปกโตรแกรม (Spectrogram) เป็นการแปลงสัญญาณเสียงให้อยู่ในรูปของภาพแสดงความถี่และเวลา ตามที่แสดงในรูปที่ 6 เพื่อให้ระบบสามารถนำไปประมวลผลผ่านโมเดล AI ของ Google ได้



รูปที่ 6 ขั้นตอนของระบบจดจำเสียงพูด

3.3 การประมวลผลด้วยโมเดล Multinomial Naïve Bayes

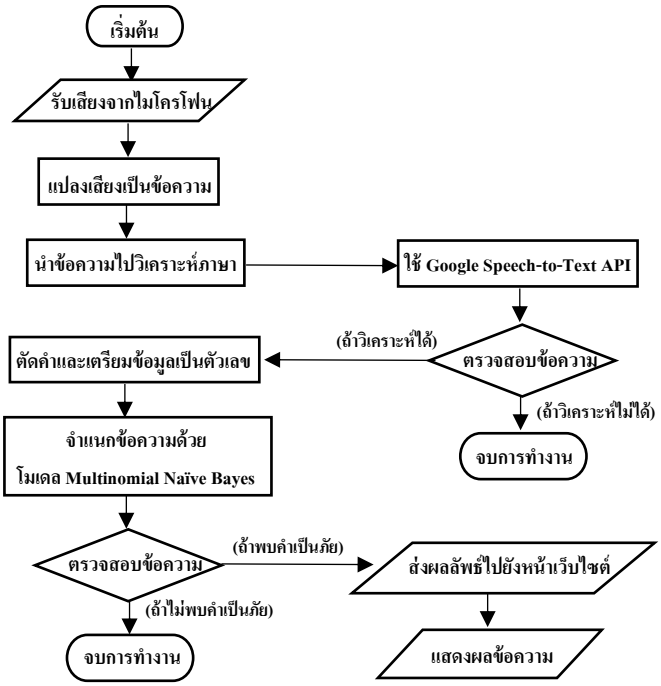
ระบบแปลงเสียงพูดเป็นข้อความแล้ว ข้อความจะถูกประมวลผลทางภาษาด้วยขั้นตอนสำคัญ ได้แก่ การตัดคำและลบคำฟุ่มเฟือยด้วย PyThaiNLP การแปลงข้อความเป็นเวกเตอร์ด้วยเทคนิค TF-IDF จากนั้นข้อมูลจะถูกแบ่งออกเป็นชุดฝึกและชุดทดสอบ และทำการปรับค่าพารามิเตอร์ Alpha ด้วย Grid search เพื่อเพิ่มประสิทธิภาพของโมเดล Multinomial Naïve Bayes ซึ่งใช้สำหรับจำแนกข้อความว่าเป็นภัยหรือไม่ ผลลัพธ์สุดท้ายจะถูกแสดงผ่านระบบแบบเรียลไทม์ ตามที่แสดงในรูปที่ 7



รูปที่ 7 การประมวลผลด้วยโมเดล Multinomial Naïve Bayes

3.4 ภาพรวมการทำงานของระบบตรวจจับคำพูดภาษาไทยที่เป็นภัย

เพื่อให้เห็นลำดับการทำงานของระบบทั้งหมดอย่างเป็นขั้นตอนที่เชื่อมโยงกันอย่างเป็นระบบ โดยเริ่มต้นจากการรับสัญญาณเสียงผ่านไมโครโฟน จากนั้นระบบจะทำการแปลงเสียงพูดเป็นข้อความโดยใช้ Google Speech-to-Text API ที่แปลงเสียงเป็นข้อความในแบบเรียลไทม์ เมื่อได้ข้อความจากเสียงพูดแล้ว ระบบจะตรวจสอบความถูกต้องเบื้องต้น หากสามารถวิเคราะห์ข้อความได้ ระบบจะนำข้อความนั้นเข้าสู่ขั้นตอนประมวลผลภาษาธรรมชาติ โดยเริ่มจากการตัดคำในภาษาไทยและลบคำฟุ่มเฟือยออก ก่อนจะแปลงข้อความเป็นเวกเตอร์ลักษณะเชิงตัวเลขด้วยเทคนิค TF-IDF จากนั้นระบบจะนำข้อมูลเวกเตอร์ดังกล่าวเข้าสู่โมเดล Multinomial Naïve Bayes ที่ผ่านการฝึกฝนมาก่อนหน้า หากระบบพิจารณาแล้วว่าข้อความมีลักษณะเป็นภัย ระบบจะทำการ แจ้งเตือนผ่านเว็บไซต์ในรูปแบบเรียลไทม์ พร้อมบันทึกข้อความ คำที่เป็นภัย และเวลาที่ตรวจพบ เพื่อใช้ในการติดตามหรือวิเคราะห์เพิ่มเติม ตามที่แสดงในรูปที่ 8



รูปที่ 8 ภาพรวมการทำงานของระบบ

3.5 เครื่องมือและเทคโนโลยีที่ใช้

ระบบนี้พัฒนาด้วยภาษา Python พร้อมเทคโนโลยีเว็บ HTML, CSS และ JavaScript โดยรองรับการทำงานแบบเรียลไทม์ผ่าน Flask และ Socket เพื่อแสดงผลลัพธ์ทันที เมื่อมีการตรวจจับคำพูดที่เป็นภัย ในส่วนของการประมวลผลเสียงและข้อความ ใช้ไลบรารี Speech Recognition สำหรับการแปลงเสียงเป็นข้อความโดยเชื่อมต่อกับ Google Speech API ข้อความจะถูกประมวลผลด้วยไลบรารี PyThaiNLP เพื่อทำการตัดคำและลบคำฟุ่มเฟือย ก่อนนำไปแปลงเป็นเวกเตอร์ด้วยเทคนิค TF-IDF และนำเข้าสู่กระบวนการจำแนกข้อความด้วยโมเดล Multinomial Naïve Bayes ซึ่งได้รับการปรับค่าพารามิเตอร์ Alpha ด้วยวิธี Grid search เพื่อเพิ่มประสิทธิภาพในการจำแนกผลลัพธ์ ยังมีการใช้ Matplotlib และ Seaborn ในการแสดงผลเชิงภาพ

3.6 การแสดงผลการตรวจพบคำพูดที่เป็นภัย

สำหรับผลลัพธ์ของระบบหลังจากทำการวิเคราะห์เสียงพูด ตรวจจับคำที่มีแนวโน้มว่าเป็นภัย และแสดงผลในรูปแบบตารางผ่านเว็บไซต์แบบเรียลไทม์ (Real-time) โดยมีรายละเอียด แสดงเวลา (Timestamp) ที่ระบบตรวจพบข้อความ ข้อความ (Message) ที่ได้ผ่านการประมวลผลแล้ว และคำที่เป็นภัยที่ระบุว่าเข้าข่ายเป็นภัย เช่น “ระเบิด”, “ก่อการร้าย” เป็นต้น ตามที่แสดงในรูปที่ 9

รูปที่ 9 แสดงผลลัพธ์ของระบบผ่านเว็บไซต์แบบเรียลไทม์

4. การทดลองและผลการทดลอง

4.1 การทดลอง

ในการทดลองของระบบตรวจจับคำพูดภาษาไทยที่เป็นภัย ระบบต้นแบบถูกทดสอบในสถานการณ์จำลองภายในห้องทำงาน ซึ่งจำลองเป็นพื้นที่ที่มีการใช้งานจริง โดยทำการทดลองภายใต้สภาพแวดล้อมที่มีเสียงรบกวนระดับปานกลาง เช่น เสียงพัดลม เสียงพื้นหลัง และเสียงพูดทั่วไป เพื่อทดสอบความสามารถของระบบในการแยกเสียงพูดออกจากเสียงรบกวน

ระบบได้รับการติดตั้งและรันผ่านเครื่องคอมพิวเตอร์ พร้อมไมโครโฟนภายในเครื่อง โดยผู้วิจัยได้ทำการพูดข้อความในบริบทที่หลากหลาย ทั้งข้อความที่เข้าข่ายภัย และข้อความปกติ เพื่อให้ระบบทำการแปลงเสียงพูดเป็นข้อความและตรวจจับคำพูดที่เป็นภัยแบบเรียลไทม์แล้วแสดงข้อความที่ตรวจพบ พร้อมคำเตือนและการบันทึกเวลาที่ตรวจจับได้

4.2 การวัดประสิทธิภาพ

Confusion Matrix เป็นเครื่องมือสำคัญที่ใช้ในการประเมินประสิทธิภาพของโมเดลจำแนกประเภท (Classification Model) การทำงานของโมเดล Multinomial Naïve Bayes ช่วยให้เข้าใจภาพรวมของการทำงานของโมเดลทั้งในด้านความแม่นยำ ความน่าเชื่อถือ และความสามารถในการตรวจจับภัยอย่างมีประสิทธิภาพ ตามที่แสดงในรูปที่ 10 มีการแสดงผลลัพธ์การทำนายของโมเดลเปรียบเทียบกับค่าจริง ประกอบด้วย True Positive (TP) คือ จำนวนข้อความที่เป็นภัยจริง และระบบทำนายว่าเป็นภัย True Negative (TN) คือ จำนวนข้อความที่ไม่เป็นภัยจริง และระบบทำนายว่าไม่เป็นภัย False Positive (FP) คือ จำนวนข้อความที่ไม่เป็นภัย แต่ระบบทำนายว่าเป็นภัย (แจ้งเตือนผิดพลาด) และ False Negative (FN) คือ จำนวนข้อความที่เป็นภัย แต่ระบบทำนายว่าไม่เป็นภัย (ตรวจจับไม่เจอ) โดยใช้ค่าทั้ง 4 มาคำนวณตัวชี้วัดที่สำคัญ ได้แก่ Accuracy (ความแม่นยำโดยรวม)โดยใช้สมการ:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2)$$

Precision (ความแม่นยำของการทำนาย) โดยใช้สมการ:

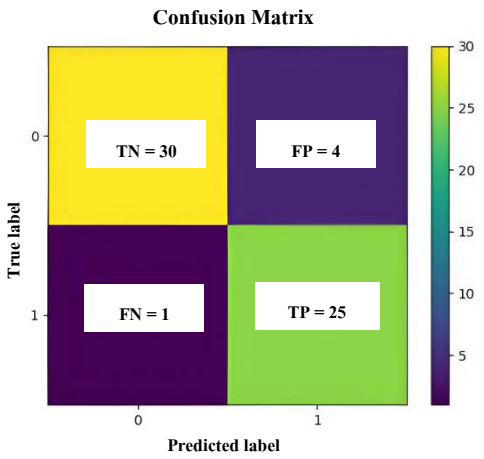
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

Recall (ความสามารถในการตรวจจับภัยได้ครบถ้วน) โดยใช้สมการ:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

และ F1-score (ค่าเฉลี่ยแบบถ่วงน้ำหนักของ Precision และ Recall) เพื่อแสดงสมดุลระหว่างความแม่นยำและความครอบคลุมโดยใช้สมการ:

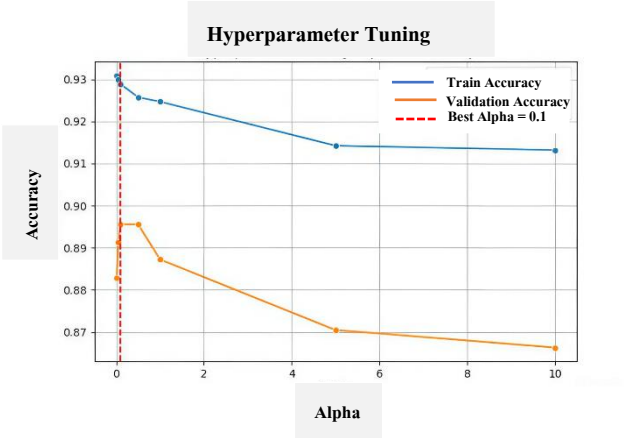
$$\text{F1} - \text{score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$



รูปที่ 10 ผลลัพธ์ของคำนวณค่าชี้วัดประสิทธิภาพ

4.3 ผลการปรับค่าพารามิเตอร์ Alpha ด้วย Grid search

ในขั้นตอนการฝึกโมเดล Multinomial Naïve Bayes การเลือกค่าพารามิเตอร์ที่เหมาะสมมีความสำคัญอย่างยิ่งต่อประสิทธิภาพของโมเดล โดยเฉพาะพารามิเตอร์ Alpha ซึ่งใช้สำหรับกระบวนการ Smoothing เพื่อลดปัญหาค่าความน่าจะเป็นเป็นศูนย์ [3] ในการทดลองนี้ใช้เทคนิค Grid search เพื่อหาค่า Alpha ที่เหมาะสมที่สุด โดยจะทดสอบโมเดลด้วยค่าหลายๆ ค่า และประเมินผลด้วยค่า Accuracy ของทั้งชุดข้อมูลฝึก (Train) และชุดข้อมูลตรวจสอบ (Validation) เพื่อหลีกเลี่ยงปัญหา Overfitting ตามที่แสดงในรูปที่ 11



รูปที่ 11 กราฟการเปรียบเทียบ Accuracy ระหว่างชุดข้อมูล Train และ Validation

5. การวิเคราะห์ผลการทดลอง

5.1 วิเคราะห์ค่าชี้วัดประสิทธิภาพ

จาก Confusion Matrix แสดงให้เห็นผลการจำแนกของโมเดลจำแนกเป็น 4 ประเภท (แสดงในรูปที่ 8) ดังนี้ True Positive (TP) = 25 แสดงว่าข้อความที่เป็นภัยจริงและโมเดลจำแนกถูกต้อง True Negative (TN) = 30 แสดงว่าข้อความที่ไม่เป็นภัยและโมเดลจำแนกถูกต้อง False positive (FP) = 4 แสดงว่าข้อความที่ไม่เป็นภัยแต่โมเดลทำนายว่าเป็นภัย (อาจมีแจ้งเตือนเกินจริง) และ False Negative (FN) = 1 แสดงว่ายังข้อความที่เป็นภัยแต่โมเดลไม่สามารถตรวจจับได้ จากค่าดังกล่าวสามารถคำนวณค่าชี้วัดต่างๆ ได้ความแม่นยำ Accuracy จากสมการที่ (2) เท่ากับ 0.917 สะท้อนถึงอัตราการทำนายถูกต้องโดยรวมของระบบ ในขณะที่ Precision จากสมการที่ (3) เท่ากับ 0.862 หมายถึง อัตราที่โมเดลทำนายว่าเป็นภัยและถูกต้องจริง ส่วน Recall จากสมการที่ (4) เท่ากับ 0.962 แสดงถึงความสามารถของโมเดลในการจับข้อความที่เป็นภัยได้อย่างครอบคลุม และสุดท้าย F1-score จากสมการที่ (5) เป็นค่าเฉลี่ยถ่วงน้ำหนักระหว่าง Precision และ Recall อยู่ที่ 0.909

ค่าชี้วัดดังกล่าวแสดงให้เห็นว่าโมเดลมีความแม่นยำสูงในการตรวจจับข้อความที่เป็นภัย โดยเฉพาะ Recall ที่สูงถึง 96.2% (เมื่อคิดเป็นเปอร์เซ็นต์) หมายความว่าระบบสามารถตรวจจับภัยได้เกือบทั้งหมด อย่างไรก็ตาม ยังมีอัตรา False positive เล็กน้อย ซึ่งอาจเกิดจากข้อความที่มีคำสุ่มเสี่ยงแต่ไม่ได้มีเจตนาเป็นภัย

5.2 วิเคราะห์จากกราฟการปรับค่าพารามิเตอร์ Alpha

จากกราฟการปรับพารามิเตอร์ Alpha ด้วย Grid search (แสดงในรูปที่ 9) พบว่า เมื่อค่า Alpha อยู่ในช่วง ต่ำกว่า 0.5 โดยเฉพาะที่ 0.1 ให้ค่า Accuracy ของ Validation สูงที่สุด และเมื่อ Alpha เพิ่มขึ้น ค่าความแม่นยำ (Accuracy) ของชุด Validation ลดลงอย่างชัดเจน ในขณะที่ชุด Train ยังคงมี Accuracy สูงอย่างสม่ำเสมอ แสดงให้เห็นว่า ค่า Alpha ที่มากเกินไปทำให้โมเดลมีการปรับเรียบเกิน (over-smoothing) จนสูญเสียรายละเอียดที่จำเป็นในการจำแนก และค่า Alpha = 0.1 ให้สมดุลที่ดีที่สุดระหว่างการเรียนรู้และการทำนาย

6. สรุป

งานวิจัยนี้ได้นำเสนอการพัฒนาและออกแบบระบบตรวจจับคำพูดที่เป็นภัยแบบเรียลไทม์ โดยใช้การแปลงเสียงเป็นข้อความผ่าน Google Speech Recognition และประมวลผลข้อความด้วยเทคนิค TF-IDF ก่อนจำแนกด้วยโมเดล Multinomial Naïve Bayes ได้รับการปรับค่าพารามิเตอร์ Alpha ด้วย Grid search เพื่อเพิ่มประสิทธิภาพ ระบบสามารถแสดงผลผ่านเว็บไซต์และแจ้งเตือนทันทีเมื่อตรวจพบข้อความที่เป็นภัย โดยผลการประเมินชี้ว่าโมเดลมีประสิทธิภาพสูง ด้วยค่า Accuracy 0.917, Precision 0.862, Recall 0.962 และ F1-score 0.909 สะท้อนถึงความสามารถของระบบในการตรวจจับภัยได้อย่างแม่นยำ ครอบคลุม และเหมาะสมต่อการใช้งานในพื้นที่ที่ต้องการความปลอดภัยสูง

7. ข้อเสนอแนะ

ระบบที่พัฒนาขึ้นสามารถตรวจจับคำพูดภาษาไทยที่เป็นภัยและแสดงผลผ่านเว็บไซต์แบบเรียลไทม์ ช่วยให้ผู้ใช้งานรับรู้เหตุการณ์ได้ทันที อย่างไรก็ตาม ระบบยังมีขอบเขตการตอบสนองเพียงในระดับการแสดงผลบนหน้าจอเท่านั้น ดังนั้น ข้อเสนอแนะในการพัฒนาต่อยอด คือ การขยายขีดความสามารถของระบบโดยการเชื่อมต่อกับ อุปกรณ์จริง ผ่านเทคโนโลยี Internet of Things (IoT) เช่น ไฟเตือนฉุกเฉิน เมื่อมีคำพูดที่เข้าข่ายอันตราย ระบบสามารถสั่งงานให้ไฟกระพริบในพื้นที่นั้นทันที เพื่อให้บุคลากรรอบข้างรับรู้สถานการณ์ หรือ ระบบควบคุมประตูอัตโนมัติ เช่น การสั่งล็อกประตู เพื่อจำกัดพื้นที่เกิดเหตุทันที เมื่อมีการตรวจพบคำพูดต้องสงสัย และการส่งข้อมูลไปยังอุปกรณ์เจ้าหน้าที่ เช่น แอปพลิเคชันที่รับแจ้งเตือนจากเซิร์ฟเวอร์ผ่านระบบ Flask + SocketIO การนำระบบนี้ไปเชื่อมต่อกับอุปกรณ์จริง จะช่วยให้การแจ้งเตือน ไม่จำกัดเพียงในระบบดิจิทัล แต่สามารถแปลงเป็น กลไกตอบสนองจริงในสถานที่

เอกสารอ้างอิง

[1] Cliff Weitzman, “Google Speech: เทคโนโลยีที่เปลี่ยนเสียงพูดให้เป็นข้อความได้แบบเรียลไทม์,” Speechify Blog, Accessed on: Mar. 27, 2025. [Online]. Available: https://speechify.com/th/blog/google-speech/?srsltid=AfmBOop7nfwldW8Q-X5pKuUTQobvTp6CwP8n7i00wAw66RTztG_SHC9B.

[2] A. Tiwari and A. Agrawal, “Comparative analysis of different machine learning methods for hate speech recognition in Twitter text data,” in Proc. 3rd Int. Conf. on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), IEEE, pp. 1–5, 2022.

[3] Mohamed P. Vijay and P. Verma, “Variants of Naïve Bayes algorithm for hate speech detection in text documents,” in Proc. Int. Conf. on Advances in Intelligent Systems and Computing (AISC), IEEE, pp. 1-4, 2023.

[4] M. M. Danyal, S. S. Khan, M. Khan, M. B. Ghaffar, B. Khan, and M. Arshad, “Sentiment analysis based on the performance of linear support vector machine and multinomial Naïve Bayes using movie reviews,” Journal of Business and Data Science (JBD), vol. 5, no. 2, pp. 1–8, 2023.

[5] Amity Solutions, “Speech Recognition: เมื่อ AI Chatbot เริ่มฉลาดใกล้เคียงสมองมนุษย์,” Amity Solutions Blog, Accessed on: Mar. 27, 2025. [Online]. Available: https://www.amitysolutions.com/th/blogs/speech-recognition-ai-chatbot-human-brain.

[6] W. Gevaert, G. Tsenov, and V. Mladenov, “Neural networks used for speech recognition,” Journal of Automatic Control, vol. 20, pp. 1–7, 2010. [Online]. Available: https://www.researchgate.net/publication/49600679