การใช้การเรียนรู้ของเครื่องสำหรับการจำแนกข้อความภาษาไทยที่เข้าข่ายหมิ่นประมาทบนเฟสบุ๊คสาธารณะ

โดย
นายปฏิภาณ วัจนาภรณ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล แผน ก แบบ ก 2 ระดับปริญญามหาบัณฑิต
ภาควิชาคอมพิวเตอร์
มหาวิทยาลัยศิลปากร
ปีการศึกษา 2566
ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

การใช้การเรียนรู้ของเครื่องสำหรับการจำแนกข้อความภาษาไทยที่เข้าข่ายหมิ่นประมาทบนเฟสบุ๊คสาธารณะ

โดย
นายปฏิภาณ วัจนาภรณ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล แผน ก แบบ ก 2 ระดับปริญญามหาบัณฑิต
ภาควิชาคอมพิวเตอร์
มหาวิทยาลัยศิลปากร
ปีการศึกษา 2566
ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

USING MACHINE LEARNING FOR THAI DEFAMATORY TEXT CLASSIFICATION ON PUBLIC FACEBOOK

By

MR. Patipan WATJANAPRON

A Thesis Submitted in Partial Fulfillment of the Requirements

for Master of Science INFORMATION TECHNOLOGY AND DIGITAL INNOVATION

Department of COMPUTER SCIENCE

Academic Year 2023

Copyright of Silpakorn University

หัวข้อ	การใช้การเรียนรู้ของเครื่องสำหรับการจำแนกข้อความภาษาไทยที่เข้าข่ายหมิ่นประมาทบนเฟสบุ๊คสาธารณะ
โดย	นายปฏิภาณ วัจนาภรณ์
สาขาวิชา	เทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล แผน ก แบบ ก 2 ระดับปริญญามหาบัณฑิต
อาจารย์ที่ปรึกษาหลัก	ผู้ช่วยศาสตราจารย์ ดร. อรวรรณ เชาวลิต

คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร ได้รับพิจารณาอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต

.....คณบดีคณะวิทยาศาสตร์
(ผู้ช่วยศาสตราจารย์ ดร. นรงค์ ฉิมพาลี)

พิจารณาเห็นชอบโดย

.....ประธานกรรมการ
(รองศาสตราจารย์ ดร. โอม ศรนิล)

.....อาจารย์ที่ปรึกษาหลัก
(ผู้ช่วยศาสตราจารย์ ดร. อรวรรณ เชาวลิต)

.....ผู้ทรงคุณวุฒิภายใน
(อาจารย์ ดร. สัจจาภรณ์ ไวจรรยา)

620720028 : เทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล แผน ก แบบ ก 2 ระดับปริญญามหาบัณฑิต
คำสำคัญ : การหมิ่นประมาท, การเรียนรู้เชิงลึก, การจำแนกประเภทข้อความ, สื่อสังคมออนไลน์, การเรียนรู้ของเครื่อง, โครงข่ายประสาทเทียมแบบคอนโวลูชัน, การพิจารณาคดี

นาย ปฏิภาณ วัจนาภรณ์: การใช้การเรียนรู้ของเครื่องสำหรับการจำแนกข้อความภาษาไทยที่เข้าข่ายหมิ่นประมาทบนเฟสบุ๊คสาธารณะ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก : ผู้ช่วยศาสตราจารย์ ดร. อรวรรณ เชาวลิต

งานวิจัยนี้มีวัตถุประสงค์เพื่อจำแนกข้อความ หรือประโยคภาษาไทยที่มีลักษณะหมิ่นประมาทบนเฟซบุ๊ก โดยอ้างอิงจากความคิดเห็นของผู้เชี่ยวชาญด้านกฎหมาย เพื่อใช้เป็นเครื่องมือในการคัดกรองข้อความสำหรับการพิจารณาฟ้องร้อง หรือดำเนินคดีทางกฎหมายในความผิดฐานหมิ่นประมาทตามประมวลกฎหมายของไทย นอกจากนี้ยังสามารถใช้เป็นตัวช่วยคัดกรองข้อความก่อนโพสต์ของผู้ใช้งานสื่อสังคมออนไลน์ได้อีกด้วย งานวิจัยนี้ใช้เทคนิคการเรียนรู้เชิงลึกเพื่อวิเคราะห์ข้อความจากการแสดงความคิดเห็น (comments) ใต้รูปภาพ หรือบทความของบุคคลที่ถูกกล่าวถึงบนเฟซบุ๊ก และใช้ข้อมูลนำเข้าที่ประกอบด้วยข้อความร่วมกับคุณลักษณะพิเศษที่ถูกสกัดจากข้อความ โดยได้สร้างแบบจำลองการเรียนรู้เชิงลึก 5 วิธีเพื่อจำแนกข้อความหมิ่นประมาท ได้แก่ 1) Long Short-Term Memory (LSTM) 2) Bidirectional Long-Short Term Memory (Bi-LSTM) 3) Convolutional Neural Networks (CNN) 4) WangchanBERTa 5) PhayaThaiBERT โดยใช้การสกัดคุณลักษณะจากการฝังคำ (word embedding) ด้วย thai2fit การนับความถี่คำศัพท์จากคำพิพากษา (Term Frequency of judges' vocabulary) การแท็กส่วนประกอบคำพูด (Part-of-Speech tagging) และการแท็กชื่อเฉพาะ (Named Entity tagging) ผลการทดลองแสดงให้เห็นว่า PhayaThaiBERT ให้ผลลัพธ์ดีที่สุดเมื่อใช้การฝังคำด้วย PhayaThaiBERT และการนับความถี่คำศัพท์จากคำพิพากษาในการสกัดคุณลักษณะของคำ ซึ่งในงานวิจัยนี้ใช้แบบจำลองพื้นฐาน (base model) และพบว่าการปรับแต่งพารามิเตอร์ของแบบจำลองรวมถึงวิธีการตัดคำ อาจส่งผลให้ประสิทธิภาพของแบบจำลองดีขึ้นได้

620720028 : Major INFORMATION TECHNOLOGY AND DIGITAL INNOVATION
Keyword : Defamatory, Deep learning, Text classification, Social media, Machine learning, Convolutional Neural Network, Judgement

MR. Patipan WATJANAPRON : Using machine learning for Thai defamatory text classification on public facebook Thesis advisor : Assistant Professor orawan chaowalit, Ph.D.

This research aims to classify Thai texts or sentences with defamatory characteristics on Facebook by referencing the opinions of legal experts. The goal is to create a tool for filtering messages in the context of legal proceedings or lawsuits concerning defamation under Thai law. Additionally, it can assist in screening posts for social media users before they publish content. This study employs deep learning techniques to analyze comments under photos or articles of individuals mentioned on Facebook, using input data that comprises text along with features extracted from the text. We developed five deep learning models to classify defamatory messages: 1) Long Short-Term Memory (LSTM) 2) Bidirectional Long Short-Term Memory (Bi-LSTM) 3) Convolutional Neural Networks (CNN) 4) WangchanBERTa 5) PhayaThaiBERT. The feature extraction methods included word embedding with thai2fit, term frequency of judges' vocabulary, part-of-speech (POS) tagging, and named entity tagging. The experimental results showed that PhayaThaiBERT provided the best performance when using word embedding with PhayaThaiBERT and term frequency of judges' vocabulary for feature extraction. In this study, we used a base model configuration and found that tuning model parameters and tokenization methods could potentially enhance the model's performance.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี ตามขอบวัตถุประสงค์ทุกประการ ผู้วิจัยขอขอบคุณบุคคลทุกท่านที่ให้ความช่วยเหลือสนับสนุนในการจัดทำวิทยานิพนธ์ฉบับนี้ขึ้นมา ผู้วิจัยรู้สึกทราบซึ้งในความกรุณาเป็นอย่างมากจึงขอใช้เนื้อกาในส่วนนี้แสดงความขอบพระคุณ

ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.อรวรรณ เชาวลิต อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ให้ความรู้ ความเมตตา คอยแนะนำ และให้คำปรึกษาผลักดันส่งเสริมการทำวิจัย ตลอดจนแก้ไขข้อบกพร่องต่าง ๆ มาโดยตลอด ผู้วิจัยขอกราบขอบพระคุณด้วยความเคารพอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณกรรมการสอบวิทยานิพนธ์ ซึ่งประกอบไปด้วย ผู้ช่วยศาสตราจารย์ ดร.ณัฐโชติ พรหมฤทธิ์ ประธานกรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร. โอม ศรนิล และอาจารย์ ดร.สัจจา ภรณ์ ไวจรรยา ที่ได้กรุณาให้เกียรติเป็นกรรมการ และเสียสละเวลาในการตรวจสอบ อีกทั้งให้คำแนะนำที่เป็นประโยชน์ในการทำวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณผู้เชี่ยวชาญทางด้านกฎหมาย ที่ให้ความรู้ และให้คำแนะนำทางด้านกฎหมาย จึงทำให้ผู้วิจัยได้รับความรู้ และประสบการณ์เพิ่มพูนมากขึ้น

นอกจากนี้ ผู้วิจัยขอกราบขอบพระคุณครอบครัว และบุคคลอันเป็นที่รักที่คอยเป็นห่วงเป็นใย เป็นกำลังใจ และสนับสนุนผู้วิจัยเรื่อยมา รวมถึงขอบคุณตัวเองที่มีความพยายามจนทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยหวังว่าวิทยานิพนธ์ฉบับนี้จักเป็นประโยชน์ต่อผู้ที่สนใจในกาลต่อไป

ปฏิภาณ วัจนาภรณ์

มหาวิทยาลัยศิลปากร

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ญ

สารบัญภาพ ..... ฏ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญ 1

1.2 วัตถุประสงค์ของงานวิจัย 4

1.3 สมมติฐานของงานวิจัย 4

1.4 ขอบเขตการวิจัย 4

1.5 ประโยชน์ที่คาดว่าจะได้รับ 5

บทที่ 2 งานวิจัยที่เกี่ยวข้อง 6

บทที่ 3 ทฤษฎีที่เกี่ยวข้อง 9

3.1 การเรียนรู้เชิงลึก (Deep Learning) 9

3.2 หน่วยความจำระยะยาว-ระยะสั้น (Long short term memory – LSTM) 10

3.3 หน่วยความจำระยะยาว-ระยะสั้นแบบสองทิศทาง (Bidirectional Long short term memory – Bi-LSTM) 11

3.4 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution neural network – CNN) 12

3.5 Bidirectional Encoder Representations from Transformer - BERT 14

3.6 การฝังคำ (Word Embedding) 18

3.7 POS tag 19
3.8 Name Entity 22
บทที่ 4 วิธีการดำเนินงานวิจัย 24
4.1 การเตรียมข้อมูลที่ใช้ในงานวิจัย 25
4.1.1 การเก็บรวบรวมข้อมูล (Data Collects) 25
4.1.2 การระบุข้อความที่เข้าข่ายหมิ่นประมาท 26
4.1.3 การทำความสะอาดข้อมูล 28
4.1.4 การตัดคำ (Tokenize) 28
4.1.5 BERT Tokenize 29
4.1.6 การแก้ปัญหาข้อมูลไม่สมดุล (Imbalanced Dataset) 30
4.2 การสกัดคุณลักษณะของข้อความ (Features Extraction) 31
4.2.1 พจนานุกรมฎีกา (Dictionary Judgement) 31
4.2.2 Part-Of-Speech Tag 35
4.2.3 Word Embedding 36
4.3 Deep Learning Model 36
4.3.1 Features Extraction Setting 36
4.3.2 การตั้งค่าพารามิเตอร์สำหรับฝึกแบบจำลอง 38
4.4 การวัดประสิทธิภาพและตัวชี้วัด 42
4.4.1 เมทริกซ์ความสับสน (Confusion Matrix) 42
4.4.2 ค่าความถูกต้อง (Accuracy) 43
4.4.3 ค่าแม่นยำ (Precision) 43
4.4.4 ค่าความครบถ้วน (Recall) 43
4.4.5 เอฟวัน สกอร์ (F1-Score) 44
บทที่ 5 ผลการดำเนินงานวิจัย 45

5.1 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า 45
5.2 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมเป็นข้อมูลเข้า 47
5.3 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ POS tag One-hot เป็นข้อมูลเข้า 48
5.4 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรม และ POS tag Embedding เป็นข้อมูลเข้า 50
5.5 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ Name Entity One-hot เป็นข้อมูลเข้า 51
5.6 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag One-hot และ Name Entity One-hot เป็นข้อมูลเข้า 53
5.7 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag Embedding และ Name Entity One-hot เป็นข้อมูลเข้า 54
บทที่ 6 สรุปผลการทดลอง และอภิปรายผลการทดลอง 56
6.1 สรุปผลการทดลองและอภิปรายผลการทดลอง 56
6.2 เปรียบเทียบประสิทธิภาพของแบบจำลองและเทคนิคที่นำเสนอกับงานวิจัยอื่น 57
6.3 ปัญหาและอุปสรรค 58
6.4 แนวทางการวิจัยในอนาคต 59
รายการอ้างอิง 60
ประวัติผู้เขียน 65

สารบัญตาราง

หน้า

ตารางที่ 1 แสดงตัวอย่างชุดข้อมูลที่ผ่านการ Label โดยผู้เชี่ยวชาญ 27

ตารางที่ 2 แสดงคำนาม 452 คำ 32

ตารางที่ 3 แสดงคำสกรรมกริยา 155 คำ 33

ตารางที่ 4 แสดงคำอกรรมกริยา 269 คำ 33

ตารางที่ 5 แสดงวลีหรือกลุ่มคำ 10 คำ 34

ตารางที่ 6 แสดงคำสบประมาท 59 คำ 34

ตารางที่ 7 แสดงคำสรรพนามบุคคลที่หนึ่ง 28 คำ 35

ตารางที่ 8 แสดงคำสรรพนามบุคคล จำนวน 63 คำ 35

ตารางที่ 9 เมทริกซ์ความสับสน 42

ตารางที่ 10 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้เฉพาะการฝังคำเป็นข้อมูลเข้า 46

ตารางที่ 11 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา 48

ตารางที่ 12 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ POS tag One-hot เป็นข้อมูลเข้า 49

ตารางที่ 13 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ POS tag Embedding เป็นข้อมูลเข้า 51

ตารางที่ 14 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ Name Entity One-hot เป็นข้อมูลเข้า 52

ตารางที่ 15 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag One-hot และ Name Entity One-hot เป็นข้อมูลเข้า 54

ตารางที่ 16 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag Embedding และ Name Entity One-hot เป็นข้อมูลเข้า 55

มหาวิทยาลัยศิลปากร

สารบัญภาพ

หน้า

ภาพที่ 1 เปรียบเทียบโครงข่ายประสาทเทียม กับโครงข่ายประสาทเชิงลึก 10
ภาพที่ 2 การทำงานของ RNN 11
ภาพที่ 3 การทำงานของ LSTM 11
ภาพที่ 4 การทำงานของ Bi-LSTM 12
ภาพที่ 5 การทำงานของ CNN 13
ภาพที่ 6 การทำของ Masked LM 14
ภาพที่ 7 การทำงานของ Next Sentence Prediction 15
ภาพที่ 8 การทำ Fine-tuning แก้ปัญหากลุ่มปัญหาระดับคำ 16
ภาพที่ 9 การทำ Fine-tuning แก้ปัญหากลุ่มปัญหาระดับประโยค 17
ภาพที่ 10 ตัวอย่างการแทนคำด้วย Vector 18
ภาพที่ 11 การหาความคล้ายคลึงระหว่างคำ (Word similarity) 19
ภาพที่ 12 แสดงการติดป้ายแท็ก POS โดยใช้ไลบารี่แบ่งคำตามคลังข้อมูล ORCHID 20
ภาพที่ 13 แสดงป้ายแท็ก POS และคำอธิบาย ของไลบารี่แบ่งคำตามคลังข้อมูล ORCHID 21
ภาพที่ 14 Name Entity Tag ที่ใช้ใน PyThaiNLP 23
ภาพที่ 15 การใช้ Name Entity Tag โดย PythaiNLP 23
ภาพที่ 16 แสดงภาพรวมกระบวนการทำงานของงานวิจัยนี้ 24
ภาพที่ 17 การแสดงความคิดเห็นเฟสบุ๊ค 25
ภาพที่ 18 จำนวนข้อมูลที่ผ่านการ Label โดยผู้เชี่ยวชาญ 27
ภาพที่ 19 ตัวอย่างการตัดคำโดย newmm 29
ภาพที่ 20 กระบวนการทำ BERT Tokenizer 30
ภาพที่ 21 การทำ Random Oversampling 30

ภาพที่ 22 คำพิพากษาศาลฎีกาที่ 2371/2522 31

ภาพที่ 23 5-fold cross validation 36

ภาพที่ 24 โครงสร้างและพารามิเตอร์ของแบบจำลอง LSTM 39

ภาพที่ 25 โครงสร้างและพารามิเตอร์ของแบบจำลอง Bi-LSTM 40

ภาพที่ 26 โครงสร้างและพารามิเตอร์ของแบบจำลอง CNN 41

ภาพที่ 27 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของการทดลองหัวข้อที่ 5.1 46

ภาพที่ 28 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของการทดลองหัวข้อที่ 5.2 47

ภาพที่ 29 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของการทดลองหัวข้อที่ 5.3 49

ภาพที่ 30 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของหัวข้อการทดลองที่ 5.4 50

ภาพที่ 31 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของหัวข้อการทดลองที่ 5.5 52

ภาพที่ 32 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของหัวข้อการทดลองที่ 5.6 53

ภาพที่ 33 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของหัวข้อการทดลองที่ 5.7 55

ภาพที่ 34 รายงาผลการทดลองเปรียบเทียบประสิทธิภาพจากงานวิจัย PhayaThaiBERT[8] 57

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบันการสื่อสารระหว่างบุคคลในสังคมผ่านทางโซเชียลมีเดียมีความสะดวกสบายและรวดเร็วมากขึ้น นับตั้งแต่เริ่มมีการพัฒนาสื่อออนไลน์ผ่านเทคโนโลยีอินเทอร์เน็ต ในปัจจุบันมีผู้ใช้สื่อสังคมออนไลน์จำนวนสูงถึง 5 พันล้านคนจากการรายงานของ Shewale [1] โดยมีแพลตฟอร์มต่าง ๆ เช่น เฟสบุ๊ค ทวิตเตอร์ อินสตาแกรม และติ๊กต็อก ที่นำเสนอฟังก์ชั่นการใช้งานที่แตกต่างกันออกไป แพลตฟอร์ม อินสตาแกรม และติ๊กต็อก เน้นการเผยแพร่เรื่องราวของผู้ใช้ด้วยสื่อประเภทรูปภาพและวิดีโอ ในขณะที่เฟสบุ๊คและทวิตเตอร์ จัดเป็นแพลตฟอร์มที่ได้รับความนิยมในหลายประเทศ ซึ่งช่วยให้ผู้ใช้สามารถเผยแพร่สื่อมัลติมีเดียได้หลากหลาย และยืดหยุ่นมากขึ้นการโต้ตอบกันผ่านการแสดงความคิดเห็นบนโพส หรือบทความของผู้ใช้รายอื่นในเฟสบุ๊ค หรือการตอบกลับข้อความในทวิตเตอร์ ได้รับความนิยม เนื่องจากช่วยให้ผู้ใช้สามารถแลกเปลี่ยนความคิดเห็น และความรู้ได้อย่างสะดวก และรวดเร็ว นอกจากนี้การสื่อสารออนไลน์ยังเป็นเครื่องมือที่สำคัญในการเชื่อมโยงบุคคลในสังคมไว้ด้วยกัน

อย่างไรก็ตาม การสื่อสารออนไลน์ก็อาจถูกใช้เป็นเครื่องมือที่ผู้กระทำผิดใช้สร้างความเสียหายกับผู้อื่นได้ ซึ่งเรียกว่าการละเมิดทางไซเบอร์ เช่น การใช้วาจาสร้างความเกลียดชัง การเหยียดเชื้อชาติ การเหยียดเพศ การคุกคามทางเพศ การเสียดสี การกลั่นแกล้ง การล้อเลียน หรือการใช้คำหยาบคาย โดยการกระทำดังกล่าวนำมาซึ่งผลกระทบต่อผู้เสียหาย หรือผู้ที่ได้รับผลกระทบไม่ว่าจะเป็นบุคคล องค์กร หน่วยงาน หรือห้างร้าน ต่าง ๆ ซึ่งในกรณีผู้เสียหายเป็นบุคคล อาจได้รับผลกระทบในหลายด้าน อาทิ

1.ความเดือดร้อนและความเสียหายทางจิตใจ คือการโดนหมิ่นประมาทอาจทำให้เหยื่อรู้สึกโกรธ หรือเศร้า ส่งผลให้เกิดความเครียด และภาวะซึมเศร้า นำมาซึ่งความคิดเกี่ยวกับการฆ่าตัวตาย โดยเรื่องนี้มีการศึกษาพฤติกรรมฆ่าตัวตาย และความคิดฆ่าตัวตายในเยาวชนที่มีสาเหตุมาจากผลกระทบจากการกลั่นแกล้งทางอินเทอร์เน็ต[2]

2.ลดคุณค่าของตนเอง เหยื่ออาจรู้สึกถูกเหยียดหยาม หรือไม่ได้รับการยอมรับจากผู้อื่น ส่งผลให้เหยื่อรู้สึกไม่มีค่า

3.การถูกกีดกัน บางครั้งการละเมิดทางไซเบอร์ อาจทำให้เหยื่อถูกปฏิเสธ หรือกีดกันในสังคมออนไลน์ หรือในชีวิตประจำวัน

4. .ผลกระทบต่อการทำงาน และการศึกษา เหยื่อที่ถูกละเมิดอาจพบความยากลำบากในการทำงาน หรือการศึกษา เช่น การโดนไล่ออกจากที่ทำงาน หรือการโดนสอบสวนจากปัญหาความน่าเชื่อถือ หรือประวัติที่เสียหาย

5.ความเสียหายทางสังคม การละเมิดอาจมีผลให้เหยื่อถูกคนในสังคมตีตราลดความน่าเชื่อถือ หรือลดความเคารพในตัวบุคคล และส่งผลให้เกิดความเสียหายต่อความสัมพันธ์ทางสังคม

ส่วนในมุมของธุรกิจ หรือองค์กรต่าง ๆ ที่ได้รับผลกระทบจากการละเมิด อาจนำมาซึ่งปัญหาต่าง ๆ เช่น ผลกระทบต่อชื่อเสียง การถูกละเมิดบนสื่อสังคมออนไลน์ อาจทำให้องค์กรถูกมองในแง่ลบ มีผลกระทบต่อชื่อเสียง และภาพลักษณ์ขององค์กร ซึ่งอาจส่งผลต่อการได้รับความไว้วางใจจากลูกค้า หรือพันธมิตรทางธุรกิจอื่น ๆ ทั้งผลกระทบทางการเงิน เนื่องมาจากลูกค้าที่ลดลง หรือใช้เงินในการฟ้องร้องคดี เป็นต้น ซึ่งจากผลกระทบต่างที่เกิดขึ้นจากที่กล่าวข้างต้น ทำให้การละเมิดทางไซเบอร์เป็นปัญหาใหญ่ที่เกิดขึ้นทั่วโลก ในหลายประเทศจึงมีกฎหมายที่ใช้ควบคุม และลงโทษผู้กระทำความผิดในการละเมิดดังกล่าว โดยในประเทศไทยมีกฎหมายที่คุ้มครองผู้ถูกละเมิดทางไซเบอร์ที่เกี่ยวข้อง ในความผิดฐานหมิ่นประมาทผู้อื่น รวมทั้งความผิดฐานดูหมิ่นผู้อื่นบนสื่อสังคมออนไลน์ ตามประมวลกฎหมายอาญา มีมาตราดังนี้

ประมวลกฎหมายอาญา มาตรา 326 บัญญัติว่า “ผู้ใดใส่ความผู้อื่นต่อบุคคลที่สาม โดยประการที่น่าจะทำให้ผู้อื่นนั้นเสียชื่อเสียง ถูกดูหมิ่น หรือถูกเกลียดชัง ผู้นั้นกระทำความผิดฐานหมิ่นประมาท ต้องระวางโทษจำคุกไม่เกินหนึ่งปี หรือปรับไม่เกินสองหมื่นบาท หรือทั้งจำทั้งปรับ”

ประมวลกฎหมายอาญา มาตรา 328 บัญญัติว่า “ถ้าความผิดฐานหมิ่นประมาทได้กระทำโดยการโฆษณาด้วยเอกสาร ภาพวาด ภาพระบายสี ภาพยนตร์ ภาพ หรือตัวอักษรที่ทำให้ปรากฏไม่ว่าด้วยวิธีใด ๆ แผ่นเสียง หรือสิ่งบันทึกเสียง บันทึกภาพ หรือบันทึกอักษรกระทำโดยการกระจายเสียงหรือการกระจายภาพ หรือโดยกระทำการป่าวประกาศด้วยวิธีอื่น ผู้กระทำต้องระวางโทษจำคุกไม่เกินสองปี และปรับไม่เกินสองแสนบาท”

ประมวลกฎหมายอาญา มาตรา 393 บัญญัติว่า ผู้ใดดูหมิ่นผู้อื่นซึ่งหน้า หรือด้วยการโฆษณาต้องระวางโทษจำคุกไม่เกินหนึ่งเดือน หรือปรับไม่เกินหนึ่งหมื่นบาท หรือทั้งจำทั้งปรับ

กฎหมายดังกล่าวบัญญัติขึ้นเพื่อเป็นมาตราการป้องกันความเสียหายจากการถูกละเมิด และปรับใช้ในการควบคุมการถูกละเมิดทางไซเบอร์ด้วย โดยในการฟ้องร้องคดีหมิ่นประมาทของเหยื่อผู้ได้รับผลกระทบตามมาตรา 326 หรือการหมิ่นประมาทต่อบุคคลที่สาม จะพิจารณาข้อความที่เข้าข่ายหมิ่นประมาทตามองค์ประกอบดังนี้

1. ผู้ใส่ความ หรือผู้กระทำการหมิ่นประมาท
2. มีการใส่ความ หรือมีการหมิ่นประมาท
3. ผู้อื่น หรือผู้ถูกใส่ความ
4. ต่อบุคคลที่สาม เป็นองค์ประกอบสำคัญที่ทำให้มีความผิดตามมาตรานี้
5. โดยประการที่น่าจะทำให้ผู้อื่นนั้นได้รับความเสียหาย เกิดความเสียหายจากการกระทำหมิ่นประมาทนั้นโดยตรง

แต่ทั้งนี้ไม่ใช่ทุกข้อความ หรือทุกประโยคที่จะเข้าองค์ประกอบความผิดของกฎหมายหมิ่นประมาทที่จะฟ้องร้อง หรือดำเนินคดีทางกฎหมายได้ แม้จะเป็นคำที่หยาบคาย ซึ่งต้องอาศัยความรู้เฉพาะทางจากผู้เชี่ยวชาญด้านกฎหมาย เพื่อพิจารณาว่าควรฟ้องร้อง หรือดำเนินคดีทางกฎหมายหรือไม่ อีกทั้งจากการศึกษาพบว่าในปัจจุบันยังมีคดีความผิดฐานหมิ่นประมาทที่เกี่ยวกับการถูกละเมิดทางไซเบอร์จำนวนมากที่ยังรอการตัดสินมากกว่า 20,000 คดี จากการรายงานขององค์กรสิทธิมนุษยชน (Article19) [3] และยังมีผู้เสียหายอีกจำนวนมาก ที่ยังไม่ได้ทำเรื่องฟ้องร้องต่อศาล

ด้วยเหตุนี้ผู้วิจัยได้เล็งเห็นถึงความสำคัญและปัญหาการถูกหมิ่นประมาทบนสื่อสังคมออนไลน์ จึงได้พัฒนาแบบจำลองที่สามารถจำแนกข้อความที่เข้าข่ายหมิ่นประมาทบนสื่อสังคมออนไลน์ขึ้น โดยในปัจจุบันสามารถพัฒนาอัลกอริธึมการประมวลผลภาษาธรรมชาติสำหรับการตรวจจับคำพูดแสดงความเกลียดชังบนโซเชียลมีเดียโดยใช้เทคนิคการเรียนรู้เชิงลึก (Deep Learning) และโครงข่ายประสาท (Neural Networks) [4-6] นอกจากนี้ในงานด้านการจำแนกข้อความภาษาไทย [7, 8] ได้นำเสนอการประยุกต์ใช้แบบจำลองทางภาษาซึ่งพัฒนาจากสถาปัตยกรรมทรานฟอร์เมอร์อย่างเบิร์ต (BERT) มาใช้ในงานด้านการจำแนกข้อความภาษาไทย ซึ่งมีความแม่นยำสูงจากแบบจำลองที่ฝึกฝนโดยชุดข้อมูลเฉพาะภาษาไทย จึงเป็นที่มาของงานวิจัยนี้ในการจำแนกข้อความ หรือประโยคภาษาไทยบนเฟสบุ๊ค ซึ่งข้อความที่นำไปเรียนรู้ได้มาจากข้อความแสดงความคิดเห็นใต้รูปภาพ หรือใต้บทความของผู้ถูกกล่าวถึง และผู้วิจัยได้จะจำแนกข้อความที่เข้าข่าย และไม่เข้าข่ายความผิดฐานหมิ่นประมาทผู้อื่นโดยอ้างอิงจากผู้เชี่ยวชาญด้านกฎหมาย โดยใช้เทคนิคการเรียนรู้เชิงลึกวิเคราะห์จากบทความ และคุณลักษณะพิเศษที่สำคัญต่อการพิจารณาฟ้องร้องคดีเป็นข้อมูลนำเข้า เพื่อเป็นระบบสำหรับช่วยคัดกรองข้อความ ให้แก่ผู้ที่ได้รับผลการกระทบจากการละเมิดทางไซเบอร์ ในการพิจารณาฟ้องร้อง หรือดำเนินคดีทางกฎหมายในความผิดฐานหมิ่นประมาทตามประมวลกฎหมายของไทยต่อไป

1.2 วัตถุประสงค์ของงานวิจัย

วัตถุประสงค์ของงานวิจัยการจำแนกข้อความหมิ่นประมาทในภาษาไทยบนเฟซบุ๊กมีดังนี้

1. เพื่อศึกษาและพัฒนาแบบจำลองด้วยเทคนิคการเรียนรู้เชิงลึกสำหรับจำแนกข้อความที่เข้าข่ายหมิ่นประมาทบนเฟสบุ๊ค

2. เพื่อเปรียบเทียบแบบจำลองการเรียนรู้เชิงลึก 5 วิธีคือ 1) Long Short-Term Memory (LSTM) 2) Bidirectional Long-Short Term Memory (Bi-LSTM) 3) Convolutional Neural Networks (CNN) 4)WangchanBERTa 5) PhayaThaiBERT

3. เพื่อเปรียบเทียบการสกัดคุณลักษณะของคำ สำหรับเป็นข้อมูลเข้าแบบจำลอง 4 วิธีคือ 1) การฝังคำ (Word Embeding) 2) การนับความถี่คำศัพท์จากคำพิพากษา (Term Frequency of judges' vocabulary) 3) การแท็กส่วนประกอบคำพูด (Part-of-Speech tagging) 4) การแท็กชื่อเฉพาะ (Named Entity tagging)

1.3 สมมติฐานของงานวิจัย

ข้อความ หรือประโยคในสื่อสังคมออนไลน์ในไทย มีลักษณะรูปประโยคที่อาจเข้าองค์ประกอบความผิดฐานหมิ่นประมาทตามประมวลกฎหมายอาญามาตราที่เกี่ยวข้อง การพัฒนาตัวแบบการเรียนรู้เชิงลึก ที่สามารถจดจำ และเรียนรู้องค์ประกอบความผิดภายใต้เงื่อนไขต่าง ๆ จึงสามารถใช้ในการจำแนกข้อความ หรือประโยคข้อความที่เข้าข่ายหมิ่นประมาทบนสื่อสังคมออนไลน์ได้ ซึ่งสามารถนำไปต่อยอดในการพัฒนาแพลตฟอร์มสำหรับตรวจสอบ หรือจำแนกข้อความที่เข้าข่ายหมิ่นประมาทได้

1.4 ขอบเขตการวิจัย

1. สร้างคลังคำศัพท์คำที่หมิ่นประมาท โดยรวบรวมมาจากคำพิพากษาศาลฎีกา โดยพิจารณาจากคำพิพากษาที่ผู้พิพากษาตัดสินว่ามีความผิดตามประมวลกฎหมายอาญา มาตรา 326 มาตรา 328 และมาตรา 393

2. ชุดข้อมูลในการฝึก(Train Dataset) และ ชุดข้อมูลทดสอบ(Test Dataset) เก็บรวบรวมมาจากการแสดงความคิดเห็นใต้รูปภาพบทความ หรือกระทู้แบบสาธารณะของผู้ใช้ผ่านแพลตฟอร์มเฟสบุ๊ค

3. แบบจำลองที่ได้จากงานวิจัยนี้ สามารถใช้ระบุประโยคความคิดเห็นภาษาไทยที่มีการแสดงความคิดเห็นกันบนสื่อสังคมออนไลน์ว่าประโยคความคิดเห็นใด อาจเข้าข่ายหมิ่นประมาทตามประมวลกฎหมายอาญามาตรา 326 มาตรา 328 และมาตรา 393 โดยเป็นการวิเคราะห์เฉพาะข้อความที่โพส ไม่พาดพิงส่งผลกระทบต่อตัวผู้แสดงความคิดเห็น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถระบุได้ว่าประโยคความคิดเห็นบนสื่อสังคมออนไลน์อาจเข้าข่ายความผิดหมิ่นประมาทตามประมวลกฎหมายอาญามาตรา 326 มาตรา 328 และมาตรา 393 เพื่อเป็นเครื่องมือสำหรับช่วยเหลือให้เหยื่อผู้ถูกละเมิดทางไซเบอร์ ใช้ในการประกอบการตัดสินใจสำหรับฟ้องร้องดำเนินคดีทางกฎหมายกับผู้โพสต์ หรือผู้แสดงความคิดเห็นที่ทำให้ตนเกิดความเสียหายได้

บทที่ 2
งานวิจัยที่เกี่ยวข้อง

ในหลาย ๆ ประเทศทั่วโลกได้เล็งเห็นถึงปัญหาและผลกระทบของการถูกละเมิดทางไซเบอร์ นักวิจัยในหลายสาขาวิชา จึงได้ใช้เทคโนโลยีที่ทันสมัยมาช่วยแก้ปัญหาดังกล่าว โดยในปี 2021 มีการศึกษาเพื่อจัดหมวดหมู่ของการละเมิดทางไซเบอร์ โดยพิจารณาจากสองประเภทหลักคือ cyber-hate (CH) และ cyberbullying (CB) [9]

Cyber-hate การแสดงความเกลียดชัง ในช่วงไม่กี่ปีที่ผ่านมา[9-11] นักวิจัยได้พัฒนาการตรวจจับคำพูดแสดงความเกลียดชังบนเฟสบุ๊ค และทวิตเตอร์ของชาวอินโดนีเซียในโซเชียลมีเดียโดยใช้การเรียนรู้ของเครื่อง และการเรียนรู้เชิงลึก ในปี 2020 [12] มีผู้วิจัยได้ใช้เทคนิค SVM, CNN, Attention based model, BERT pre-trained language model สำหรับการจำแนกประเภทการละเมิด ด้วยชุดข้อมูลที่รวบรวมจากการแสดงความคิดเห็นบนสื่อสังคมออนไลน์เฟสบุ๊ค และทวิตเตอร์ในภาษาอังกฤษและภาษาฮินดิ ในขณะเดียวกัน Mossie et al., [13] ได้ใช้เทคนิค Gradient Boosted Decision Trees, LSTM, GRU โดยใช้ 5,876 โพส และ 485,548 ข้อความ ของภาษาอัมฮาริก(Amharic) จากเฟสบุ๊ค ในการฝึกสอนแบบจำลอง ซึ่งได้ค่าความแม่นยำ 92.56% ในการตรวจหาคำพูดแสดงความเกลียดชังบนแพลตฟอร์มสื่อสังคมออนไลน์

ในปี 2018 นักวิจัยจำนวนมากได้พัฒนาระบบการเรียนรู้เชิงลึก (Deep Learning) และการถ่ายโอนการเรียนรู้(Transfer learning) เพื่อตรวจจับข้อความที่มีลักษณะการกลั่นแกล้ง(Cyber bullying) บนแพลตฟอร์มสื่อสังคมออนไลน์[14] ได้ใช้ CNN, LSTM, BiLSTM, BiLSTM with attention, logistic regression, SVM, RF, Naïve Bayes และได้เทรนโมเดลด้วยชุดข้อมูลจากทวิตเตอร์ และ Wikipedia ในขณะเดียวกัน Bu et al., [15] เสนอ CNN, LSTM ของการเรียนรู้เชิงลึกเพื่อตรวจจับข้อคิดเห็นเกี่ยวกับการกลั่นแกล้งทางอินเทอร์เน็ต และวิเคราะห์โมเดลโดยใช้อัลกอริธึม t-SNE เพื่อตรวจสอบความสัมพันธ์ syntactic และ semantic subsets.

จากการศึกษาเรื่อง Cyber-hate และ Cyber bullying ของงานวิจัยที่กล่าวมา แสดงให้เห็นว่าโมเดลการเรียนรู้เชิงลึก สามารถตรวจจับการละเมิดทางไซเบอร์ได้อย่างมีประสิทธิภาพ จากการศึกษา [16-20] พบว่า LSTM, Bi-LSTM, CNN เป็นเทคนิคที่ใช้กันอย่างแพร่หลายในการวิเคราะห์ข้อความสำหรับจำแนกประเภทการละเมิดทางไซเบอร์ โดยค่าความถูกต้องของ LSTM สูงถึง 97.19%, ค่าความถูกต้องของ Bi-LSTM สูงถึง 89.05% และค่าความถูกต้องของ CNN สูงถึง 97.06%

ทั้งนี้ขึ้นอยู่กับลักษณะของ Domain ของ Text และภาษา โดยผู้วิจัยได้เล็งเห็นว่าเป็นเทคนิคที่มีประสิทธิภาพ และสามารถนำมาพัฒนาต่อในการศึกษาของผู้วิจัยได้ต่อไป

ในภาษาไทยมีงานวิจัยเกี่ยวกับการประมวลผลภาษาธรรมชาติ โดยใช้เทคนิคการเรียนรู้เชิงลึก ในหลายด้านด้วยกัน[21-23] แต่การศึกษาข้อความที่ละเมิดทางไซเบอร์บนสื่อสังคมออนไลน์ด้วยเทคนิคการเรียนรู้เชิงลึกยังมีอย่างจำกัด เนื่องจากเป็นภาษาที่มีเอกลักษณ์เฉพาะตัว เช่น ตัวอักษรตัวสะกด อีกทั้งคำแต่ละคำอาจมีความหมายได้มากกว่าหนึ่งความหมาย ซึ่งงานวิจัยทางด้านนี้เป็นเรื่องที่ท้าทาย มีการพัฒนา PyThaiNLP เพื่อประมวลผลข้อความภาษาไทยโดยเฉพาะ โดยมีฟังก์ชั่นการทำงานหลายประเภท เช่น การตัดคำ (Tokenization) การวิเคราะห์ชนิดของคำทางไวย์กรณ์ (Part of speech) การตรวจสอบตัวสะกด (spell check) เป็นต้น ซึ่งได้ถูกนำไปใช้ในงานวิจัยต่าง ๆ ในช่วงไม่กี่ปีที่ผ่านมา [24] ได้วิเคราะห์ความรู้สึกของลูกค้าจากข้อความภาษาไทยผ่านโมเดล LSTM-CNN เพื่อวิจัยการตลาด ต่อมา Pasupa et al., [25] ได้เปรียบเทียบประสิทธิภาพของแบบจำลองการเรียนรู้เชิงลึก 3 วิธีได้แก่ CNN, LSTM และ Bi-LSTM ที่ใช้การสกัดคุณลักษณะของคำที่ต่างกัน ได้แก่ การฝังคำ (word embedding), การติดแท็กประเภทของคำ (POS-tag) และการหาขั้วความรู้สึกของคำ (sentic features) ในการวิเคราะห์ความรู้สึกของนิทานเด็กไทยด้วยชุดข้อมูลภาษาไทยจำนวน 1,152 ประโยค จากนิทานเด็กไทย 40 เรื่อง โดย CNN ที่ใช้คุณลักษณะทั้งสามได้ค่าความถูกต้องสูงถึง 81.7%

นอกจากนี้แบบจำลองเบิร์ต (BERT) ได้รับความนิยมมากในงานทางด้านการจำแนกข้อความภาษาไทย โดยได้ถูกใช้ในงานวิจัยของ Harnmetta et al., [26] ซึ่งได้นำเทคนิค TF-IDF, BERT และ WanchanBERTa มาใช้ในการสร้างแบบจำลองเพื่อเปรียบเทียบประสิทธิภาพในการวิเคราะห์ความคิดเห็นเกี่ยวกับหุ้นในไทย โดยใช้ชุดข้อมูลที่แบ่งออกเป็น 3 ประเภท ได้แก่ กลุ่มคนที่ซื้อ กลุ่มคนที่ไม่ขาย และกลุ่มคนที่ขาย รวมทั้งหมด 410 รายการ โดยผลการทดลองปรากฏว่าแบบจำลอง WangchanBERTa มีประสิทธิภาพสูงที่สุดที่ Accuracy 92.52% และในงานวิจัยของ Gatchalee et al., [27] ได้นำเสนอเทคนิค CNN Conv1D, CNN Skip-gram, CNN FastText Skip-gram, BERT, RoBERTa และ WangchanBERTa มาใช้ในการสร้างแบบจำลองเพื่อเปรียบเทียบประสิทธิภาพในการจัดประเภทบทความเกี่ยวกับการตลาดในภาษาไทย โดยใช้ชุดข้อมูลที่แบ่งออกเป็น 2 ประเภท ได้แก่ บทความประเภท Timely และบทความประเภท Timeless รวมทั้งหมด 600 บทความ ซึ่งผลการทดลองปรากฏว่าแบบจำลอง WangchanBERTa มีประสิทธิภาพสูงที่สุดที่ Accuracy เท่ากับ 93.00% ซึ่งแสดงให้เห็นว่าแบบจำลอง WangchanBERTa สามารถทำงานได้ดีในชุดข้อมูลภาษาไทย นอกจากนั้นในงานวิจัยของSriwirote et al., [8] ได้นำเสนอแบบจำลอง PayathaiBERT โดยโมเดลนี้ได้รับการพัฒนาต่อยอดจาก WangchanBERTa โมเดลภาษาไทยรุ่นก่อนหน้า โดย PhayaThaiBERT มีคลังคำศัพท์ที่ใหญ่กว่า WangchanBERTa มาก ซึ่งรวมถึงคำศัพท์

ต่างประเทศจำนวนมาก ทำให้โมเดลเข้าใจการสลับภาษาได้ดีขึ้น นอกจากนี้ PhayaThaiBERT ได้รับการฝึกอบรมบนชุดข้อมูลภาษาไทยขนาดใหญ่ เช่น หนังสือพิมพ์ เว็บไซต์ โซเชียลมีเดีย และอื่น ๆ ซึ่งทำให้มีประสิทธิภาพดีกว่า WangchanBERTa ในหลายงาน เช่น การแปลภาษา การจับใจความ การตอบคำถาม และอื่น ๆ

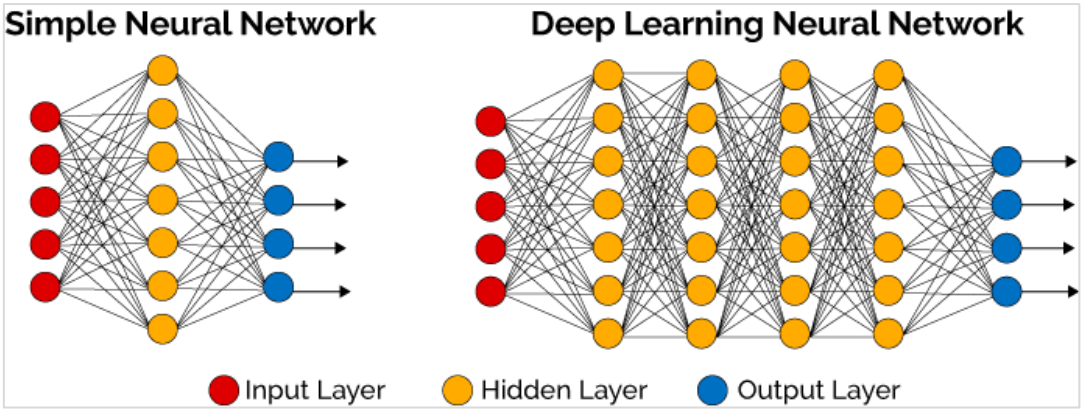
งานวิจัยต่าง ๆ ที่เกี่ยวข้องดังที่ได้กล่าวมาข้างต้น ได้พัฒนาแบบจำลองการเรียนรู้ด้วยเครื่องและแบบจำลองการเรียนรู้เชิงลึกไปใช้ในการจำแนกประโยค หรือข้อความที่เกี่ยวกับการละเมิดทางไซเบอร์ในหลากหลายภาษา และใช้การสกัดคุณลักษณะของคำที่หลากหลาย ซึ่งงานวิจัยที่เกี่ยวข้องกับการละเมิดทางไซเบอร์ที่มีความผิดตามกฎหมายหมิ่นประมาทของภาษาไทยที่ผ่านมา Arreerard et al., [28] ได้พัฒนาแบบจำลองในการจำแนกข้อความที่ละเมิดทางไซเบอร์ ด้วยการเรียนรู้ด้วยเครื่องทั้งหมด 2 วิธี ได้แก่ SVM, Naïve Bay โดยใช้ข้อความเป็นข้อมูลรับเข้าซึ่งผ่านการสกัดคุณลักษณะของคำที่แตกต่างกัน ได้แก่ Word n-grams, Charn-grams, Dependency structure, Sentiment polarity และกำหนดข้อกำหนดเฉพาะ (Specific terms) จากพจนานุกรมคำพิพากษา โดยผลการทดลองได้ประสิทธิภาพ ความแม่นยำ (Accuracy) อยู่ที่ 74% ผู้วิจัยจึงเล็งเห็นได้ว่าหากพัฒนาด้วยแบบจำลองที่ทันสมัยมากขึ้นอาจทำให้ได้ประสิทธิภาพที่ดียิ่งขึ้น งานวิจัยนี้จึงได้ศึกษาโดยการเลือกใช้เทคนิคเรียนรู้เชิงลึก 5 วิธีได้แก่ LSTM, Bi-LSTM, CNN, PayaThaiBERT, WangchanBERTa เพื่อทดลอง และพัฒนาแบบจำลองร่วมกับการสกัดคุณลักษณะของคำที่สำคัญคือการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา (Term frequency of Dictionary judgement), POS Tag และ Name Entity Tag เพื่อช่วยให้แบบจำลองเข้าใจโครงสร้างของประโยคที่มีองค์ประกอบที่เข้าข่ายความผิดฐานหมิ่นประมาทตามกฎหมาย และสามารถจำแนกข้อความที่หมิ่นประมาททางไซเบอร์ได้ เพื่อพัฒนาไปเป็นระบบสำหรับช่วยแนะนำการฟ้องร้อง หรือดำเนินคดีในความผิดฐานหมิ่นประมาททางไซเบอร์ในอนาคต

บทที่ 3
ทฤษฎีที่เกี่ยวข้อง

ในการดำเนินงานวิจัย ผู้วิจัยได้ศึกษาทฤษฎีที่เกี่ยวข้องต่าง ๆ สำหรับเป็นพื้นฐานในการดำเนินงานได้แก่ การเรียนรู้เชิงลึก, หน่วยความจำระยะยาว-ระยะสั้น, หน่วยความจำระยะยาว-ระยะสั้นแบบสองทิศทาง, โครงข่ายประสาทเทียมแบบคอนโวลูชัน, การฝังคำ, BERT

3.1 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึกเป็นกลุ่มย่อยหนึ่งของการเรียนรู้ของเครื่อง ซึ่งมันเป็นการเรียนรู้ของเครื่องประเภทหนึ่งที่จำลองการทำงานของสมองมนุษย์ เพื่อทำให้เครื่องจักรสามารถมีความคิดสำหรับตัดสินใจได้เช่นเดียวกับมนุษย์ ซึ่งสมองมนุษย์มีการกระบวนการทำงาน และกระบวนการตัดสินใจที่ซับซ้อนทำให้สามารถประมวลผลวิเคราะห์สิ่งต่าง ๆ ได้เป็นจำนวนมาก ด้วยเหตุนี้การเรียนรู้เชิงลึกจึงถูกพัฒนาขึ้นโดยเลียนแบบการทำงานของสมองมนุษย์โดยเป็นการสร้างสถาปัตยกรรมโครงข่ายประสาทเทียม จำลองเซลล์ประสาท (Node) ขึ้นมา และเชื่อมต่อเซลล์ประสาทแต่ละเซลล์เข้าด้วยกันอย่างทั่วถึง และสามารถนำมาเรียงตัวกันเป็นชั้น (Layer) ซึ่งแต่ละ Layer จะแบ่งหน้าที่การทำงานที่แตกต่างกันออกไป เช่น Input Layer, Hidden Layer และ Output Layer ซึ่งการเรียนรู้เชิงลึกโครงข่ายในชั้น Hidden Layer จะถูกซ้อนกันมากกว่า 3 ชั้นขึ้นไปไปดังแสดงในภาพที่ 1 เพื่อใช้ในการเรียนรู้ระบุรูปแบบ และจดจำข้อมูลต่าง ๆ ซึ่งปัจจุบันมีงานที่นำเอาการเรียนรู้เชิงลึกไปประยุกต์ใช้ในงานต่าง ๆ เช่น การสร้างตัวแบบจำลองเพื่อจดจำใบหน้าบุคคล รวมถึงการจดจำคำพูดของบุคคล เป็นต้น ซึ่งการเรียนรู้เชิงลึกแบบจำลองที่มีความสามารถในการประมวลผลอัตโนมัติ โดยไม่มีผู้สอนหรือแบบกึ่งผู้สอน โดยมันมีประสิทธิภาพที่ดีในการตรวจจับรูปแบบ หรือจัดหมวดหมู่ของข้อมูล ซึ่งมีประโยชน์ในการหาคุณลักษณะสำคัญของข้อมูลในงานวิจัยนี้



ภาพที่ 1 เปรียบเทียบโครงข่ายประสาทเทียม กับโครงข่ายประสาทเชิงลึก

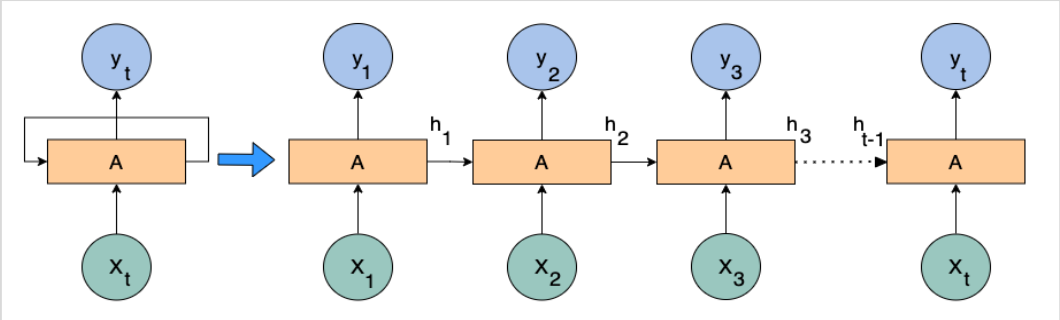
(ที่มาของภาพ https://medium.com/@nnpatel4583/what-is-deep-learning-4daa22ceea4e)

3.2 หน่วยความจำระยะยาว-ระยะสั้น (Long short term memory – LSTM)

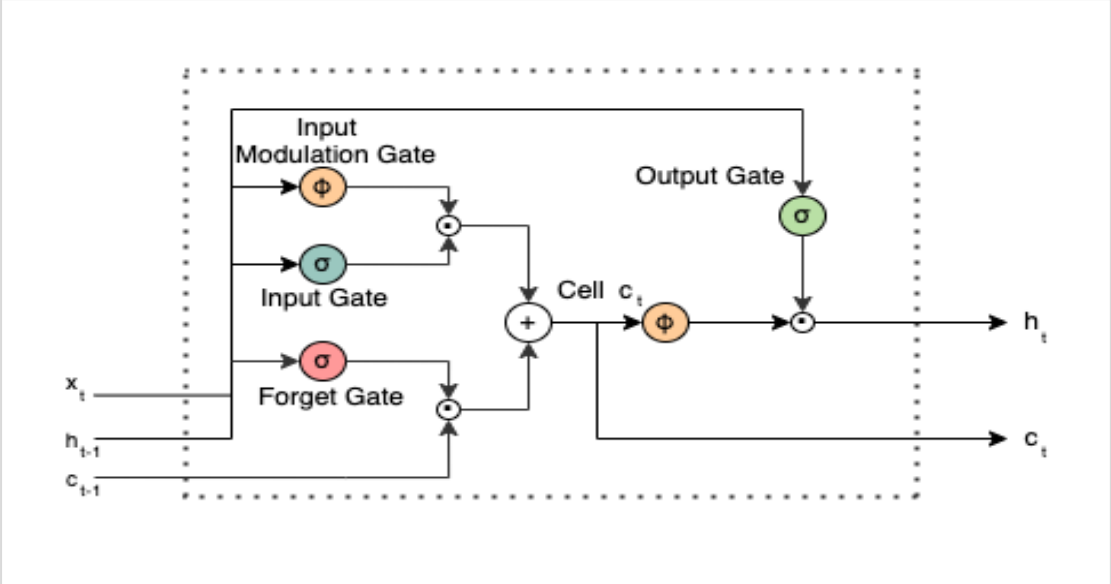
Long Short Term Memory (LSTM) [29] เป็นอัลกอริทึมที่พัฒนาต่อยอดมาจาก Recurrent Neural Network (RNN) ซึ่งเป็นอัลกอริทึมการเรียนรู้เชิงลึก ที่ใช้หลักการวิเคราะห์ข้อมูลอนุกรมลำดับ (Sequence) โดย RNN ได้รับการพัฒนามาจากโครงข่ายประสาทเทียม Neural Network(NN) แต่แตกต่างที่โดยปกติ NN จะรับค่า Inuput เข้าไปแล้วได้ค่า Output ออกมา แต่ RNN นั้นสามารถนำค่า Output ก่อนหน้า มาเก็บไว้ที่ hidden state (ht−1) เพื่อวิเคราะห์แล้วจึงกลับมาคำนวณเป็น Input ใหม่ (xt) อีกครั้ง และทำงานวนซ้ำในเครือข่ายได้ดังแสดงในภาพที่ 2 ซึ่งในการนำ RNN ไปใช้งานจริงในข้อมูลที่ Sequence มีขนาดยาวเกินไปทำให้ Gradient ที่ได้มีการเปลี่ยนแปลงจนไม่สามารถสังเกตุการเปลี่ยนแปลงเพื่อนำมาใช้วิเคราะห์ปัญหาได้ ทำให้ LSTM จึงถูกพัฒนาขึ้น เพื่อแก้ปัญหาดังกล่าว โดย LSTM ได้แก้ปัญหาโดยการออกแบบการทำงานในส่วนของ cell ใหม่ ซึ่งใน cell ของมันจะมี gate ซึ่งประกอบไปด้วย Input gate, Forgot gate, Memory Cell State Gate และ Output Gate แสดงในภาพที่ 3 โดยแต่ละ gate จะมีหน้าที่ดังต่อไปนี้

- Input gate เป็นหน่วยย่อยที่จะกำหนดข้อมูลเข้าที่จะนำมาวิเคราะห์ใน cell
- Forgot gate เป็นหน่วยย่อยที่จะกำหนดข้อมูลเข้าเพื่อวิเคราะห์เช่นเดียวกับ Input gate แต่จะกำหนดว่าข้อมูลนั้นควรถูกบันทึก หรือถูกลืม
- Memory Cell State Gate เป็นหน่วยย่อยที่กำหนดข้อมูลเข้ามาวิเคราะห์ โดยจะคำนวณค่าสถานะเพื่อใช้ในการคำนวณครั้งต่อไป

- Output Gate เป็นหน่วยย่อยสำหรับการคำนวณ Output ของ Cell โดยค่าที่ออกมาจาก cell จะมี Output และ Hidden State สำหรับการคำนวณในครั้งต่อไป



ภาพที่ 2 การทำงานของ RNN

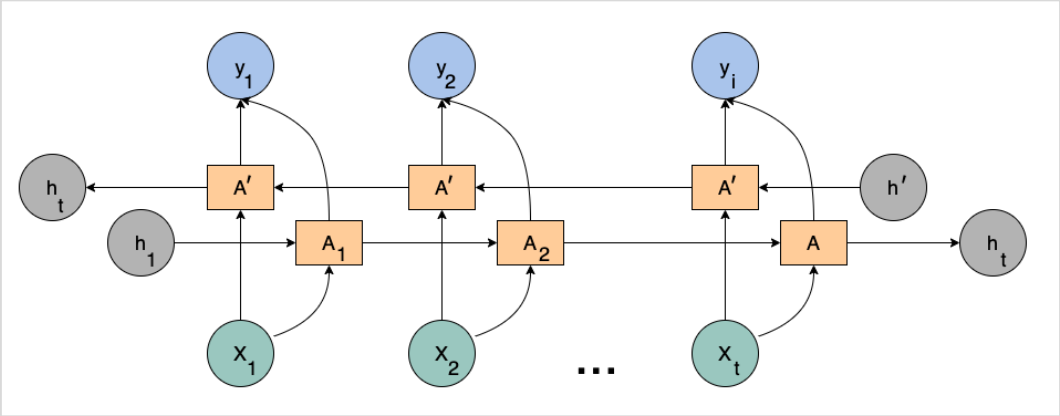


ภาพที่ 3 การทำงานของ LSTM

3.3 หน่วยความจำระยะยาว-ระยะสั้นแบบสองทิศทาง (Bidirectional Long short term memory – Bi-LSTM)

หน่วยความจำระยะยาว-ระยะสั้นแบบสองทิศทาง หรือ Bi-LSTM พัฒนาต่อยอดมาจาก LSTM มีหลักการทำงานเหมือนกัน แต่มีการเพิ่มฟังก์ชันการป้อนข้อมูลแบบสองทิศทางพร้อมกัน แสดงในภาพที่ 4 ซึ่งโดยปกติแล้วข้อมูล input จะถูกป้อนเข้าทิศทางเดียวจากซ้ายไปขวา โดย Output ที่ส่งออกมาจาก แต่การป้อนข้อมูลสองทิศทางก็ส่งผลให้การทำงานแบบ Bi-LSTM จะช้ากว่า LSTM เพราะจะมีขนาดที่ใหญ่กว่า แต่จะเหมาะกับ domain text ที่ต้องการเข้าใจความหมายของคำจาก

บริบท (context) รอบๆ ได้ เช่น ในงานวิจัยของ Khongtum et al. [30] ได้เลือกใช้ Bi-LSTM เพื่อทำ Entity Recognition กับ domain text thai poem ซึ่งมีบริบทที่ซับซ้อน ซึ่ง Bi-LSTM สามารถทำงานได้ดีใน domain text ดังกล่าว



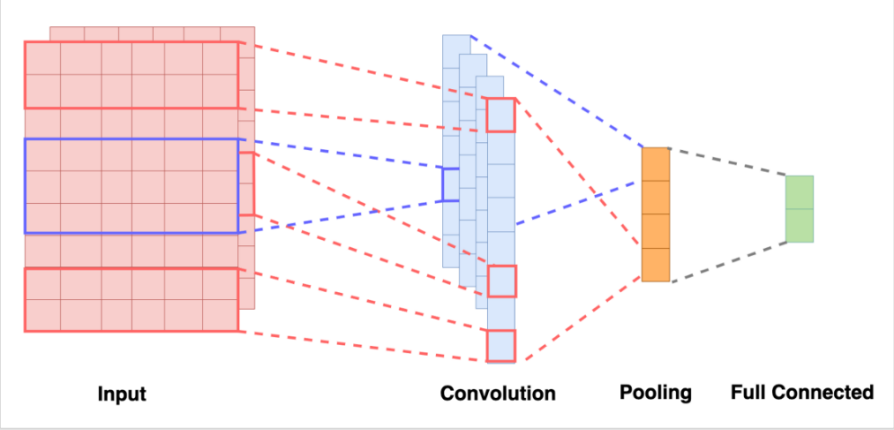
ภาพที่ 4 การทำงานของ Bi-LSTM

3.4 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution neural network – CNN)

Neural Network (NN) คือโครงข่ายประสาทเทียม ซึ่งประกอบไปด้วยเซลล์ประสาท (Node) และเส้นประสานประสาท (Dense) เป็นการจำลองโครงสร้างตามรูปแบบของเซลล์สมองมนุษย์ Convolutional Neural Network (CNN) เป็นสถาปัตยกรรม (Architecture) ที่ถูกพัฒนามาจาก Neural Network (NN) โดยมี input layer 1 ชั้น ที่ทำหน้าที่รับข้อมูล เข้ามาในโครงข่ายประสาท และส่งข้อมูลไปยัง hidden layer ชั้นถัดไป, ส่วน hidden layer มีได้มากกว่า 1 ชั้น ซึ่งการเพิ่มชั้นที่มากขึ้นส่งผลให้นิวรอล)Neuron) มีจำนวนมากขึ้น อาจทำให้มีโอกาสได้ความแม่นยำที่สูงขึ้น และส่วนสุดท้ายเป็นส่วนของ output layer โดยรับค่าจาก hidden layer ชั้นสุดท้าย ซึ่งจำนวนของ Neuron ใน output layer จะเท่ากับจำนวนของคลาสที่ต้องการจัดกลุ่ม และมีค่าคำตอบอยู่ใน Neuron โดยเมื่อ Neuron ใด ๆ มีค่าคำตอบมากที่สุดแสดงว่าคำตอบของข้อมูลที่รับเข้ามาคือคลาสนั้น

โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution neural network – CNN) จะเป็นการเพิ่มชั้น layer การประมวลผลแบบคอนโวลูชั่น (Convolutional) เข้าไปใน Neural Network การทำงานใน Convolution layer จะหาความสัมพันธ์ โดยดึงคุณลักษณะเด่นของข้อมูลเพื่อใช้ในการเรียนรู้ โดย CNN ประกอบด้วยการทำงาน 2 ส่วน แบ่งเป็น Feature Extraction และ Classification ส่วนแรก Feature Extraction จะเป็นการทำงานเพื่อคัดเลือกคุณสมบัติที่สำคัญของ

ข้อมูล โดยจะนำข้อมูลส่วนย่อย (Local Region) ของข้อมูลส่วนหนึ่งมาค่อย ๆ เรียนรู้ โดยมีตัวกรอง (Filter) หรือ เคอร์เนล (Kernel) ทำหน้าที่สกัดคุณลักษณะพิเศษสำหรับการนำไปใช้ในการจำแนกประเภทในขั้นตอน Classification ซึ่งจะทำงานโดยการวางทาบลงบนชุดข้อมูล เพื่อจะกำหนดพื้นที่ที่ต้องการใช้สำหรับวิเคราะห์ โดยเมื่อ CNN ได้ใช้ Filter ในการสร้างคุณลักษณะขึ้นมาใหม่แล้ว จะลดขนาดของคุณลักษณะที่ได้มา แต่ยังสามารถคงเอกลักษณ์ของข้อมูลเดิมไว้ได้ไม่เปลี่ยนแปลง โดยมีอัลกอริทึมที่ใช้งานได้หลายประเภท ซึ่งสองประเภทหลักที่นิยมได้แก่ Max Pooling และ Mean Pooling โดย Max Pooling เป็นตัวกรองแบบหนึ่งที่จะหาค่าสูงสุดของพื้นที่ที่ทาบอยู่ และ Mean Pooling คือการสุ่มตัวอย่างค่าเฉลี่ยของข้อมูลในขอบเขตพื้นที่ทาบอยู่ และสุดท้ายคือส่วนของการทำ Classification ซึ่งเป็นชั้นที่ทำหน้าที่จำแนก หรือทำนายผลลัพธ์ของข้อมูล โดยเป็นชั้นเชื่อมต่อแบบเต็มรูปแบบ (Fully-Connected Layer) กับชั้นก่อนหน้า เช่น ถ้าส่วนย่อย(Local Region) ของข้อมูล Input (Feature Size) 5x5 มาทาบกับ Filter Size 3x3 คือการที่ Filter จะเลื่อนทาบไปเรื่อย ๆ บนข้อมูลขนาด 5x5 และจะได้ผังคุณลักษณะ (Feature Map) ของส่วนย่อย (Local Region) ของข้อมูล จากนั้นก็นำส่วนย่อย(Local Region) ของข้อมูลถัดไปมาคูณกับ Filter เลื่อนเปลี่ยนไปเรื่อย ๆ จนกว่าจะครบทั้งข้อมูล จากนั้นเมื่อได้ชุดคุณลักษณะขึ้นมาใหม่แล้ว จะเข้าสู่ชั้นพูลลิ่ง (Pooling Layer) ซึ่งเป็นชั้นที่คั่นกลางระหว่างชั้นคอนโวลูชัน จะลดขนาด (Downsample) ของ Feature Map ให้เล็กลง แต่คงความสัมพันธ์ของข้อมูลอย่างครบถ้วน โดยมีอัลกอริทึมหลายประเภท ได้แก่ Max Pooling, Average Pooling และ Sum Pooling ต่อมาการทำงานในส่วนของ Classification จะเป็นชั้นที่ทำหน้าที่ในการจำแนกประเภทของข้อมูล ซึ่งเชื่อมโยงแบบสมบูรณ์ (Fully-Connected Layer) กับโครงสร้างของ Neural Network โดยเชื่อมโยงในชั้นคอนโวลูชัน (Convolution) และ ชั้นพูลลิ่ง (Pooling) ของ Convolution layer ดังแสดงในภาพที่ 5



ภาพที่ 5 การทำงานของ CNN

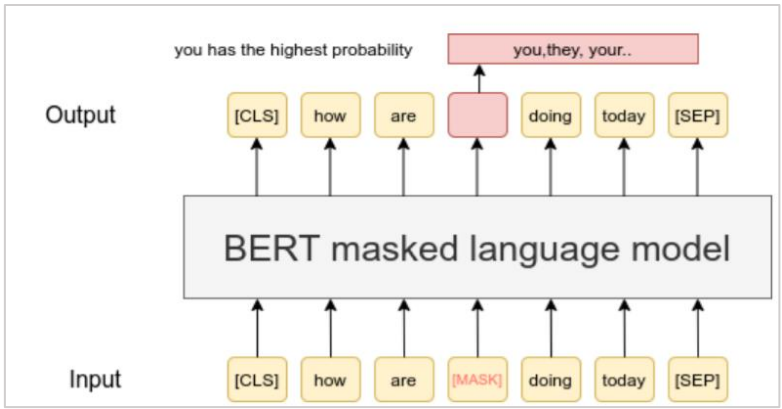
3.5 Bidirectional Encoder Representations from Transformer - BERT

ตัวเข้ารหัสแบบสองทิศทางในรูปแบบตัวแปลง หรือ BERT คือการใช้การฝึกอบรมแบบสองทิศทางของ Transformer ที่ได้การนำเสนอจากนักวิจัยของ Google AI เมื่อปี 2019 [31] โดยเป็นแบบจำลองที่ได้รับความนิยมในการสร้างแบบจำลองด้านภาษา (Language Model) เมื่อเปรียบเทียบกับแบบจำลองแบบดั้งเดิมซึ่งจะลำดับข้อมูลเข้าจะรับข้อมูลเข้าจากซ้ายไปขวา หรือเป็นการรวมกันระหว่างข้อความจากซ้ายไปขวา และขวาไปซ้าย ผลของงานวิจัยนี้ได้แสดงให้เห็นว่า BERT สามารถเข้าใจบริบท และเข้าใจความหมายของภาษาได้ลึกซึ้งมากกว่าแบบจำลองแบบดั้งเดิมที่รับข้อมูลแบบทิศทางเดียว

สำหรับแบบจำลอง Language Model จะประสบปัญหา และความท้าทายในการระบุผลลัพธ์ของการทำนาย โดยในแบบจำลองแบบดั้งเดิมจะทำนายคำถัดไปเป็นลำดับจากซ้ายไปขวา เช่น “เขากลับบ้านจาก...” ซึ่งวิธีดังกล่าวจะมีข้อจำกัดในการเรียนรู้บริบทของคำรอบ ๆ ซึ่งเพื่อจัดการปัญหานี้ BERT ได้ใช้สองวิธี ได้แก่

1. Masked LM (MLM)

MLM ได้รับการออกแบบมาเพื่อแก้ไขปัญหา Language แบบดั้งเดิม โดยในแต่ละรอบของการฝึกสอนแบบจำลอง จะนำประโยคจำนวนหนึ่งป้อนเป็นข้อมูลเข้าประมาณ 15% โดยของคำทั้งหมดจะถูกลบออก (Masked words) และแบบจำลองจำเป็นต้องเติมคำที่ถูกลบออกไปด้วยตัวเอง ซึ่งการที่จะเพิ่มคำที่ถูกต้องได้ แบบจำลองจำเป็นจะต้องเข้าใจบริบทที่ได้มาจากคำที่อยู่รอบ ๆ ในทางตรงข้ามเมื่อแบบจำลองได้เรียนรู้โดยวิธีแบบ MLM ไประยะเวลาหนึ่ง แบบจำลองจะพยายามเดาคำที่หายไปเพียงอย่างเดียว และไม่สนใจคำอื่นเลย จึงทำให้ในเวกเตอร์คำที่แบบจำลองไม่สนใจดังกล่าวนั้นไม่มีประโยชน์สำหรับการฝึกสอนแบบจำลองโดยจะแสดงการทำงานของ MLM ในภาพที่ 6

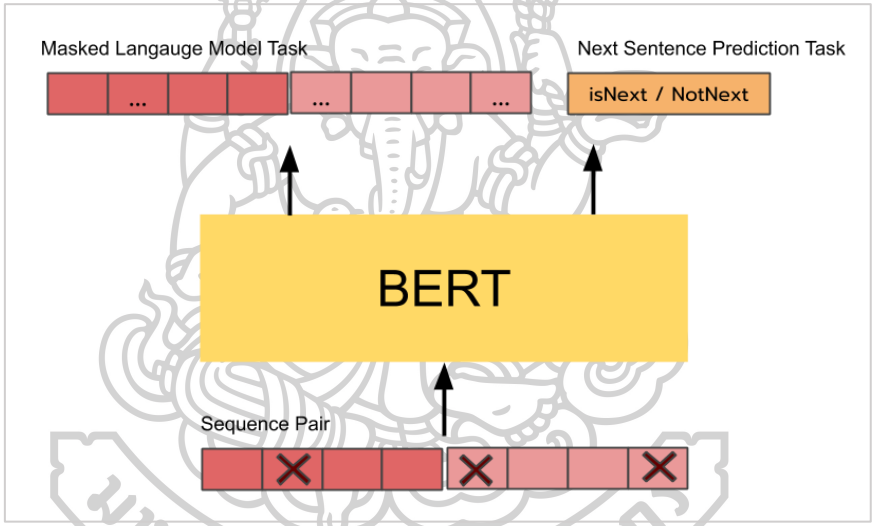


ภาพที่ 6 การทำของ Masked LM

(ที่มาภาพ https://www.sbert.net/examples/unsupervised_learning/MLM/README.html)

2. Next Sentence Prediction (NSP)

Next Sentence Prediction (NSP) เป็นเทคนิคในการฝึก และใช้งานโมเดลปัญญาประดิษฐ์ ซึ่งมักนำมาใช้ในโมเดลการเรียนรู้ของคำภาษาธรรมชาติ (NLP) เป้าหมายของการสร้างโมเดล NSP คือเพื่อให้แบบจำลองได้เรียนรู้ และสามารถเข้าใจ หรือทำนายความสัมพันธ์ระหว่างประโยค เพื่อให้แบบจำลองได้เรียนรู้ความสัมพันธ์ระหว่างประโยค BERT ได้เพิ่มการทำ NSP ไปพร้อมกับการทำ MLM โดยจะนำเข้าข้อมูล 2 ประโยค และให้แบบจำลองเลือกว่าประโยคทั้งสองนั้น อยู่ติดกัน หรือไม่ โดยจะแบ่งข้อมูลออกครึ่งนึง โดยเป็นคู่ของประโยคที่อยู่ติดกัน และอีกครึ่งเป็นประโยคที่จะจับคู่กันแบบสุ่มโดยจะแสดงการทำงานของ NSP ในภาพที่ 7



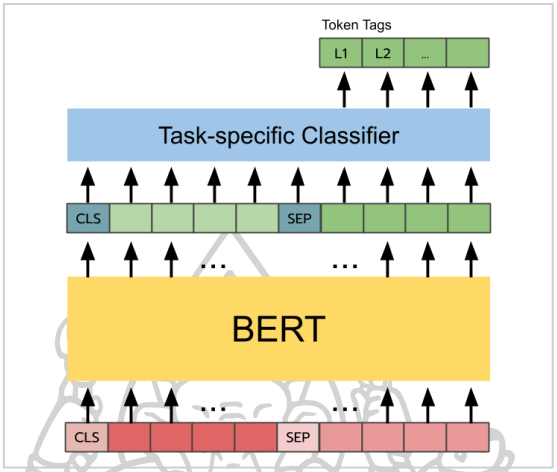
ภาพที่ 7 การทำงานของ Next Sentence Prediction

(ที่มาภาพ https://medium.com/@chameleontk/ทำความเข้าใจ-bert-98589715545)

BERT สามารถนำมาประยุกต์ใช้ในงานด้านภาษาได้หลากหลาย โดยเพิ่มชั้น Layer ไปซึ่งเรียกว่าการทำ Fine-tuning เป็นการฝึกอบรมแบบจำลองซ้ำอีกรอบด้วยข้อมูลในงานที่ต้องการ ตัวอย่างการใช้งาน BERT ในงานทางภาษาได้แก่

1. กลุ่มปัญหาระดับคำ (Token-level task) โดยในกลุ่มนี้จะอยู่ในรูปแบบของ การจำแนกแบบไบนารี (Binary classification) หรือการจำแนกประเภทหลายคลาส (Multi-Class Classification) เช่น การทำ Name Entity Recognition เพื่อหาตำแหน่งของคำ และแปะป้ายชื่อเฉพาะของคำเหล่านั้นที่อยู่ในประโยค หรือเอกสาร การทำ Part of Speech Tag เพื่อการระบุกว่าแต่ละคำคือส่วนใดของประโยค โดยในกลุ่มปัญหาระดับคำ BERT สามารถ Fine-Tuning โดยนำ

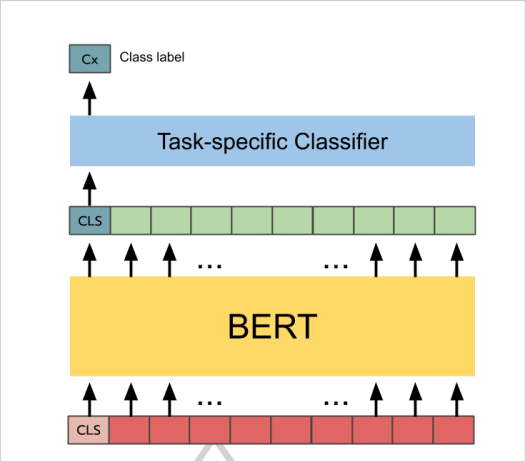
เวกเตอร์ผลลัพธ์จาก BERT ส่งผ่านไปยัง Classifier เพื่อทำนายว่าข้อมูลคำนั้น ๆ อยู่ในกลุ่มใด จากนั้นเอาคำตอบที่ได้ไปอัปเดตแบบจำลองทั้งในส่วน BERT และ Classifier ต่อไป โดยทำซ้ำจนกว่าจะได้ค่าความแม่นยำที่ต้องการแสดงการทำงานในกลุ่มปัญหาระดับคำในภาพที่ 8



ภาพที่ 8 การทำ Fine-tuning แก้ปัญหากลุ่มปัญหาระดับคำ

(ที่มาภาพ https://medium.com/@chameleontk/ทำความเข้าใจ-bert-98589715545)

2. กลุ่มปัญหาระดับประโยค (Sentence-level tasks) ในกลุ่มนี้แบบจำจองจะต้องประมวลผลภาพรวมของประโยค) เพื่อสามารถเข้าใจบริบท และความหมายของประโยคได้ เช่น การวิเคราะห์อารมณ์ของประโยค (Sentiment Analysis) การให้คะแนนความคล้ายคลึงของความหมายแต่ละประโยค (Sematic Textual Similarity) โดยในการแก้ปัญหาแบบจำลองจะนำเวกเตอร์ผลลัพธ์ [CLS] token (Classification) ซึ่งเป็นโทเค็นพิเศษที่ถูกเพิ่มเข้าไปที่จุดเริ่มต้นของทุกอินพุตประโยค หรือข้อความ ใส่ใน Classifier เพื่อประมวลผลหาคำตอบ นอกจากนั้นข้อมูลนำเข้าของกลุ่มปัญหาดังกล่าวจะมี 2 รูปแบบ คือ ประโยคเดี่ยว หรือประโยคคู่ โดยทั้งสองรูปแบบจะสามารถทำ Fine-Tuning ได้โดยการใช้ [SEP] token (Separator) ซึ่งเป็นโทเค็นพิเศษที่ใช้ในการแยกข้อความ หรือประโยคต่าง ๆ ภายในอินพุตเดียวกัน ซึ่งจะแสดงการทำงานกับกลุ่มปัญหาระดับประโยคในภาพที่ 9



ภาพที่ 9 การทำ Fine-tuning แก้ปัญหากลุ่มปัญหาระดับประโยค

(ที่มาภาพ https://medium.com/@chameleontk/ทำความเข้าใจ-bert-98589715545)

ในปัจจุบันมีการพัฒนา Pretrainted BERT Model อยู่หลายแบบจำลองที่ถูกพัฒนาขึ้นเพื่อใช้ประมวลผลภาษาธรรมชาติ เช่น Multilingual BERT (mBERT)[31] และ RoBERTa (XLM-R)[32] เป็นแบบจำลองที่ถูกพัฒนาโดยบริษัท Google ซึ่งทั้งสองถูกฝึกสอนด้วยข้อมูลจากหลายภาษาทั่วโลกมากกว่า 100 ภาษารวมถึงภาษาไทย โดยการประมวลผลข้อมูลภาษาในแบบจำลองดังกล่าวกับภาษาไทย มีข้อจำกัดในการเรียนรู้เนื่องจากต้องเรียนรู้ในหลายภาษาทำให้แบบจำลองอาจจะไม่สามารถเจาะจงในรูปแบบการใช้ภาษาได้ จึงมีการพัฒนาPretrainted BERT Model ที่ฝึกฝนกับข้อมูลภาษาไทยขึ้น ได้แก่

1. WangchanBERTa[7] คือแบบจำลองภาษาที่ถูกพัฒนาขึ้นในรูปแบบของ Encoder-model ซึ่งเป็นหนึ่งในแบบจำลองที่ได้รับความนิยมอย่างมากในวงการประมวลผลภาษาธรรมชาติ แบบจำลองนี้ถูกพัฒนาขึ้นจากความร่วมมือระหว่าง VISTEC และ PyThaiNLP ในปี 2021 และมีผู้ใช้งานมากกว่า 1.2 ล้านครั้ง โดยเฉพาะในงานที่เกี่ยวข้องกับการเข้าใจ และประมวลผลข้อความในภาษาธรรมชาติ นิยมใช้สำหรับการประมวลผลข้อมูลภาษาไทยในหลากหลายงาน เช่น การวิเคราะห์ข้อมูลทางธุรกิจ การสนับสนุนการตัดสินใจ หรือการนำไปใช้ในงานวิจัยทางการแพทย์ อีกทั้งยังมีศักยภาพในการสนับสนุนการพัฒนา และปรับปรุงในงาน NLP อื่น ๆ ในภาษาไทยด้วย

2. PhayaThaiBERT[8] คือแบบจำลองภาษาขนาดใหญ่ (LLM) ที่พัฒนาโดยทีมวิจัยจากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ร่วมกับ Google AI แบบจำลองนี้ได้รับการฝึกฝนบนชุดข้อมูลภาษาไทยขนาดใหญ่ ประกอบด้วย ข้อความจากเว็บไซต์ หนังสือ บทความ โซเชียลมีเดีย โดยแบบจำลองนี้สามารถใช้ประมวลผลข้อมูลภาษาไทยในงานดังต่อไปนี้

1) การแปลภาษา แปลภาษาไทยเป็นภาษาอื่น และแปลภาษาอื่นเป็นภาษาไทย

2) การสร้างข้อความ เขียนข้อความต่าง ๆ เช่น บทความ ข่าวสาร บทกวี

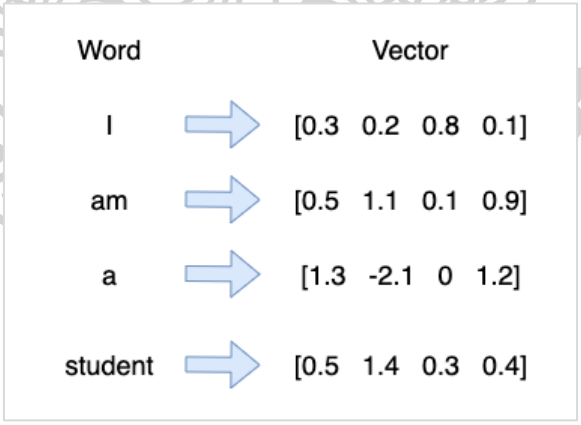
3) การตอบคำถาม ตอบคำถามภาษาไทยได้อย่างถูกต้อง

4) การสรุปข้อความ สรุปข้อความยาว ๆ ให้สั้นลง

5) การวิเคราะห์อารมณ์ วิเคราะห์ว่าข้อความนั้น ๆ แสดงอารมณ์แบบใด

3.6 การฝังคำ (Word Embedding)

การฝังคำ คือการแทนค่าของคำแต่ละคำด้วยเวกเตอร์ ซึ่งเป็นวิธีการสกัดคุณลักษณะวิธีหนึ่งในการฝึกอบรมแบบจำลองดังแสดงในภาพที่ 10 ทำให้แบบจำลองสามารถเข้าใจความหมายของคำมากขึ้น จากความสัมพันธ์ของแต่ละคำศัพท์กับคำศัพท์ข้างเคียง โดยจะมีการลดขนาดของมิติของคำลงด้วยทำให้เวกเตอร์ของคำไม่จำเป็นต้องขนาดเท่ากับจำนวนคำเสมอไปซึ่งทำให้จำนวนมิติลดลงกว่าวิธีการแบบดั้งเดิม เช่น การทำ Bag of word อย่างมาก จึงทำให้ Word embedding กินพื้นที่ในหน่วยความจำน้อยกว่า โดยในปัจจุบันมีการพัฒนา Pretrainted Word Embedding ขึ้นเพื่อช่วยให้งานทางด้านการประมวลผลทางภาษา (Natural Language Processing) ทำได้ง่ายขึ้น เช่น Word2Vec และ FastText

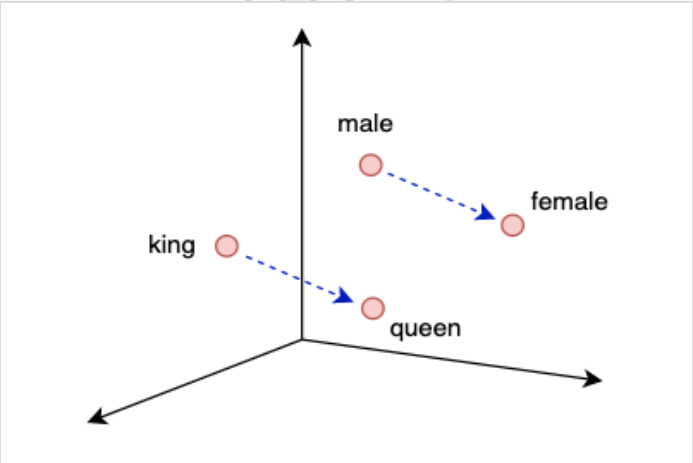


ภาพที่ 10 ตัวอย่างการแทนคำด้วย Vector

โดยซึ่งในภาษาไทยมี การพัฒนา Pretrainted Word Embedding โดย PythaiNLP คือ Thai2Fit ซึ่งเป็นแบบจำลองที่ผ่านการอบรมกับชุดข้อมูลวิกิพีเดียภาษาไทย โดยใช้วิธี Universal Language Model Fine-tuning (ULMFiT) ในการฝึกอบรมแบบจำลอง[33] โดย Thai2Fit มีคำศัพท์มากกว่า 60,000 คำในคลังคำศัพท์ ซึ่งแต่ละคำจะถูกแทนที่ด้วยเวกเตอร์ขนาด 400 มิติ

การนำ Word Embedding ไปใช้ในงาน Text Classification มีประโยชน์อย่างมาก เนื่องจากสามารถช่วยให้โมเดลสามารถเรียนรู้ และจดจำลักษณะเฉพาะของข้อความได้ดีขึ้น และนำมาใช้ในการจำแนกประเภทข้อความอย่างมีประสิทธิภาพ ซึ่งหลักการนี้สามารถปรับใช้ได้ในหลากหลายภาษา และงานที่เกี่ยวข้องกับการประมวลผลภาษาธรรมชาติ

นอกจากนี้ ยังมีการนำ Word Embedding ไปใช้ในงานอื่น ๆ เช่นการจัดกลุ่มคำ และการหาค่าความคล้ายคลึงกันของคำ (Word Similarity) ดังแสดงในภาพที่ 11 ซึ่งเป็นเครื่องมือที่มีประสิทธิภาพในการวัดความคล้ายคลึงของคำโดยพิจารณาถึงความหมายของคำด้วย การทำนี้ช่วยให้สามารถทำนายคำ หรือการค้นหาคำที่เกี่ยวข้องกับคำอื่น ๆ ได้อย่างแม่นยำ และเชื่อถือได้



ภาพที่ 11 การหาความคล้ายคลึงระหว่างคำ (Word similarity)

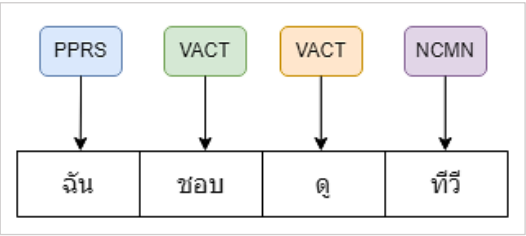
3.7 POS tag

POS tagging (Part-of-Speech tagging) เป็นกระบวนการในการประมวลผลภาษาธรรมชาติที่นำแต่ละคำในประโยคมาติดป้ายกำกับด้วยส่วนประกอบของคำ (part of speech) หรือประเภทของคำ หรือชนิดของคำที่อยู่ในประโยค เช่น นาม (noun) คำกริยา (verb) คำคุณศัพท์ (adjective) เป็นต้น เนื่องจากประโยค วลี หรือคำพูด ต่าง ๆ ที่ใช้สื่อสารกันนั้นเกิดขึ้นจากการนำคำต่าง ๆ มาประกอบกันเป็นวลี จากหลาย ๆ วลีกลายเป็นประโยคที่ทำหน้าที่แตกต่างกันในประโยค การทำ POS tagging เป็นกระบวนการสำคัญในการวิเคราะห์โครงสร้างไวยากรณ์ของประโยค ป้าย (tag) ที่กำกับให้กับแต่ละคำช่วยให้ระบบเข้าใจโครงสร้างไวยากรณ์ของประโยค และสามารถนำข้อมูลดังกล่าวไปใช้ในงานวิจัยทางด้านประมวลผลภาษาธรรมชาติต่าง ๆ ได้ เช่น การวิเคราะห์ประโยค การแยกส่วนประโยค การวิเคราะห์ความหมาย การตีความประโยค การแยกแยะที่แตกต่างของคำศัพท์ การสร้างโมเดลสำหรับประมวลผลข้อความ เป็นต้น การทำ POS tagging มีหลายวิธีได้แก่

การใช้ลักษณะ และตำแหน่งของคำในประโยค การใช้เครื่องมือวิเคราะห์ไวยากรณ์ หรือการใช้ Machine Learning เพื่อกำหนดป้ายกำกับ POS ให้กับคำแต่ละคำในประโยค เพื่อสร้างโมเดลที่สามารถจำแนกประเภทของคำในประโยคได้อย่างถูกต้อง และแม่นยำในภาษานั้น ๆ โดยทั่วไปแล้วใช้วิธีการ Supervised Learning หรือ Semi-supervised Learning

Perceptron tagger เป็นอัลกอริทึมที่ใช้ในการติดแท็ก (tagging) ประเภทของคำ (Part-of-Speech tagging) ซึ่งเป็นไลบารี่ของ Pythainlp โดยใช้แบบจำลอง Perceptron เป็นอัลกอริทึมในกลุ่มของอัลกอริทึม Discriminative learning algorithms ที่ใช้ในการเรียนรู้เพื่อพยากรณ์ (predict) หรือจำแนกแบ่งกลุ่มของข้อมูล โดยที่เน้นการเรียนรู้และคำนวณตัวแบ่ง (decision boundary) หรือฟังก์ชันที่แยกคุณลักษณะของกลุ่มข้อมูลให้ชัดเจนมากที่สุดในแต่ละกลุ่มของข้อมูล โดยไม่คำนึงถึงกระบวนการที่สร้างข้อมูลดังกล่าวมา อัลกอริทึมประมวลโดยการเรียนรู้จากชุดข้อมูลการติดแท็ก POS ที่ให้มาแล้วสร้างแบบจำลองที่สามารถใช้ในการทำนายแท็ก POS ของคำในประโยคใหม่ โดยอัลกอริทึมจะปรับค่าของพารามิเตอร์ในแต่ละรอบเพื่อให้แบบจำลองสามารถทำนายผลลัพธ์ได้ถูกต้องมากที่สุดเท่าที่เป็นไปได้ โดยใช้หลักการของการเรียนรู้เชิงคลาสิค (binary classification) ที่ใช้ใน Perceptron algorithm

Perceptron tagger ทำได้โดยรวดเร็วในการเรียนรู้ และให้ผลลัพธ์ที่มีประสิทธิภาพในการติดแท็ก POS ในภาษาต่าง ๆ โดยเฉพาะในภาษาที่มีคลังข้อมูล POS tagging ที่มีคุณภาพสูง เช่น ภาษาอังกฤษ และในกรณีของคลังข้อมูล ORCHID สำหรับภาษาไทย ORCHID นั้นคือ Online Resources for Computational HLT in Domain [34] เป็นคลังข้อมูลภาษาไทยที่ถูกพัฒนาขึ้นโดยนักวิจัยในวงการ NLP มหาวิทยาลัยเกษตรศาสตร์และมหาวิทยาลัยหัวเฉียว เพื่อวิจัยและพัฒนาในด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) ดังแสดงในภาพที่ 13 ในภาษาไทยคลังข้อมูลนี้ประกอบด้วยข้อมูลต่าง ๆ เช่น คำศัพท์, ประโยค และข้อมูลการแท็กบทบาทของคำ (POS tagging) ตัวอย่างการติดป้ายแท็ก POS โดยใช้ไลบารี่แบ่งคำตามคลังข้อมูล ORCHID ซึ่งมี 47 ประเภท เช่น ฉันชอบดูทีวี จะได้ดังภาพที่ 12



ภาพที่ 12 แสดงการติดป้ายแท็ก POS โดยใช้ไลบารี่แบ่งคำตามคลังข้อมูล ORCHID

The 47 ORCHID POS-tag	Description	The 17 UD POS-tag	Description
NCMN	Common noun	NOUN	Noun
NTTL	Title noun		
CNIT	Unit classifier		
CLTV	Collective classifier		
CMTR	Measurement classifier		
CFQC	Frequency classifier		
CVBL	Verbal classifier		
VACT	Active verb	VERB	Verb
VSTA	Stative verb		
NPRP	Proper noun	PROPN	Proper noun
NONM	Ordinal number	ADJ	Adjective
VATT	Attributive verb		
DONM	Determiner, ordinal number expression		
ADVN	Adverb with normal form	ADV	Adverb
ADVI	Adverb with iterative form		
ADVP	Adverb with prefixed form		
ADVS	Sentential adverb		
INT	Interjection	INTJ	Interjection
PPRS	Personal pronoun	PRON	Pronoun
PDMN	Demonstrative pronoun		
PNTR	Interrogative pronoun		
DDAN	Definite determiner, after noun without classifier in between	DET	Determiner
DDAC	Definite determiner, allowing classifier in between		
DDBQ	Definite determiner, between noun and classifier or preceding quantitative expression		
DDAQ	Definite determiner, following quantitative expression		
DIAC	Indefinite determiner, following noun; allowing classifier in between		
DIBQ	Indefinite determiner, between noun and classifier or preceding quantitative expression		
DIAQ	Indefinite determiner, following quantitative expression		
NCNM	Cardinal number	NUM	Numeral
NLBL	Label noun		
DCNM	Determiner, cardinal number expression		
XVBM	Pre-verb auxiliary, before negator	AUX	Auxiliary
XVAM	Pre-verb auxiliary, after negator		
XVMM	Pre-verb, before or after negator		
XVBB	Pre-verb auxiliary, in imperative mood		
XVAE	Post-verb auxiliary		
RPRE	Preposition	ADP	Adposition
JCRG	Coordinating conjunction	CCONJ	Coordinating conjunction
PREL	Relative pronoun	SCONJ	Subordinating conjunction
JSBR	Subordinating conjunction		
JCMP	Comparative conjunction		
FIXN	Nominal prefix	PART	Particle
FIXV	Adverbial prefix		
EAFF	Ending for affirmative sentence		
EITT	Ending for interrogative sentence		
NEG	Negator		
PUNC	Punctuation	PUNCT	Punctuation
		SYM	Symbol
		X	Other

ภาพที่ 13 แสดงป้ายแท็ก POS และคำอธิบาย ของไลบารี่แบ่งคำตามคลังข้อมูล ORCHID

3.8 Name Entity

Name Entity หมายถึง สิ่งที่ใช้เพื่อระบุ หรือแยกแยะชื่อของบุคคล สถานที่ หรือสิ่งของในข้อความ เป็นชนิดหนึ่งของคำนามตามหลักการภาษาไทย คือ วิสามานยนาม หรือเรียกว่า คำนามเฉพาะ เป็นคำนามที่ใช้เรียกชื่อเฉาะเฉพาะของคน สัตว์ หรือสถานที่ เป็นคำเรียกเจาะจงให้ทราบว่าเป็นใคร หรือเป็นอะไร โดยปกติแล้ว Name Entity มักนิยมใช้ในการประมวลผลภาษาธรรมชาติ การวิเคราะห์ข้อความ หรือการทำงานด้านภาษาธรรมชาติอื่น ๆ ตัวอย่างของ Name Entity ได้แก่ ชื่อคน เมือง วันที่ เวลา องค์กร และอื่น ๆ ซึ่งมักจะถูกใช้ในงานด้านการประมวลผลภาษาธรรมชาติและการทำงานด้านข้อมูลขนาดใหญ่ที่เกี่ยวข้องกับข้อความโครงสร้างทางภาษาธรรมชาติ ตัวอย่างเช่น ในประโยค “Bill Gates founded Microsoft in 1975.” ชื่อ “Bill Gates” เป็น Name Entity และ “Microsoft" เป็น Name Entity ด้วย

หลักการทำงานของ Name Entity มีขั้นตอนหลัก ๆ ดังนี้:

1. การตรวจจับ (Detection) ขั้นตอนแรกคือการตรวจจับ Name Entity ในข้อความ ซึ่งอาจใช้วิธีการต่าง ๆ เช่น การใช้กฎการตรวจจับ (rule-based approach) หรือการใช้โมเดลแบบเรียนรู้ของเครื่อง (machine learning models) เช่น โมเดลแบบที่ถูกฝึกสอนด้วยข้อมูล (data-driven models) เพื่อจำแนกประเภทของ Name Entity ที่ต้องการตรวจจับ เช่น ชื่อคน ชื่อสถานที่ เป็นต้น
2. การทำคำถาม (Question answering) เมื่อ Name Entity ถูกตรวจจับแล้ว ขั้นตอนต่อไปคือการทำคำถามเกี่ยวกับ Name Entity นั้น ๆ ในบริบทของข้อความ ซึ่งสามารถใช้วิธีการต่าง ๆ เช่น การค้นหาข้อมูลจากฐานข้อมูล การวิเคราะห์ความสัมพันธ์ระหว่าง Name Entity กับข้อมูลอื่น ๆ หรือการสร้างคำตอบจากข้อมูลที่มีอยู่ในบริบทนั้น ๆ
3. การจัดเก็บ และการใช้งาน (Storage and Utilization): สุดท้ายคือการจัดเก็บ Name Entity ที่ถูกตรวจจับ และคำตอบที่ได้จากข้อมูล ซึ่งสามารถนำไปใช้ในงานอื่น ๆ เช่น การสร้างฐานข้อมูลรวมข้อมูลที่เกี่ยวข้องกับ Name Entity การสร้างรายงาน หรือสรุปข้อมูล หรือการใช้ในการสนับสนุนการตัดสินใจในงานต่าง ๆ ตามที่เหมาะสม

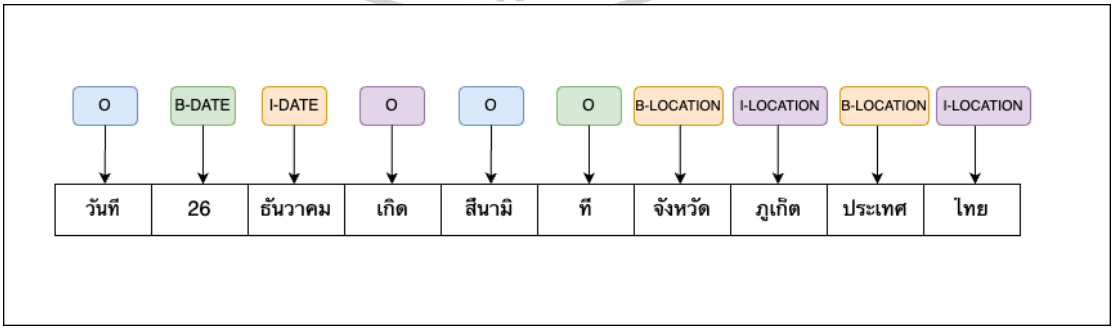
ในภาษาไทยมีเครื่องมือสำหรับการติด Name Entity Tag ของคำในประโยคว่าโดย PythaiNLP โดยจำแนกเป็น 12 กลุ่ม ได้แก่ วันที่ เวลา อีเมล ความยาว หรือระยะทาง สถานที่ องค์กร ชื่อคน เบอร์โทรศัพท์ URL รหัสไปรษณีย์ จำนวนเงิน คำเกี่ยวกับกฎหมาย ดังแสดงใน ภาพที่ 14 ซึ่ง PythaiNLP จะใช้รูปแบบ Inside-outside-beginning (IOB) เพื่อติดแท็ก Name Entity สำหรับแต่ละคำ โดยการติด Name Entity Tag ในรูปแบบ IOB จะมีการใส่คำนำหน้า Name Entity

Tag คือ B- จะแทนความหมายเป็นโทเค็นเริ่มต้นของประโยค และ I- จะแทนความหมายว่าเป็นโทเค็นภายในประโยค และ O จะเป็นการระบุว่าคำดังกล่าวไม่จัดอยู่ในหมวดหมู่ Name Entity ใด ๆ โดยตัวอย่างการใช้งาน Name Entity Tag โดย PythaiNLP แสดงในภาพที่ 15

Named Entity tag	Examples
DATE	2/21/2004, 16 ก.พ., จันทร์
TIME	16.30 น., 5 วัน, 1-3 ปี
EMAIL	info@nrpsc.ac.th
LEN	30 กิโลเมตร, 5 กม.
LOCATION	ไทย, จ.ปราจีนบุรี, กำแพงเพชร
ORGANIZATION	กรมวิทยาศาสตร์การแพทย์, อย.
PERSON	น.พ.จรัล, นางประนอม ทองจันทร์
PHONE	1200, 0 2670 8888
URL	http://www.bangkokhealth.com/
ZIP	10400, 11130
Money	2.7 ล้านบาท, 2,000 บาท
LAW	พ.ร.บ.โรคระบาด พ.ศ.2499, รัฐธรรมนูญ

ภาพที่ 14 Name Entity Tag ที่ใช้ใน PyThaiNLP

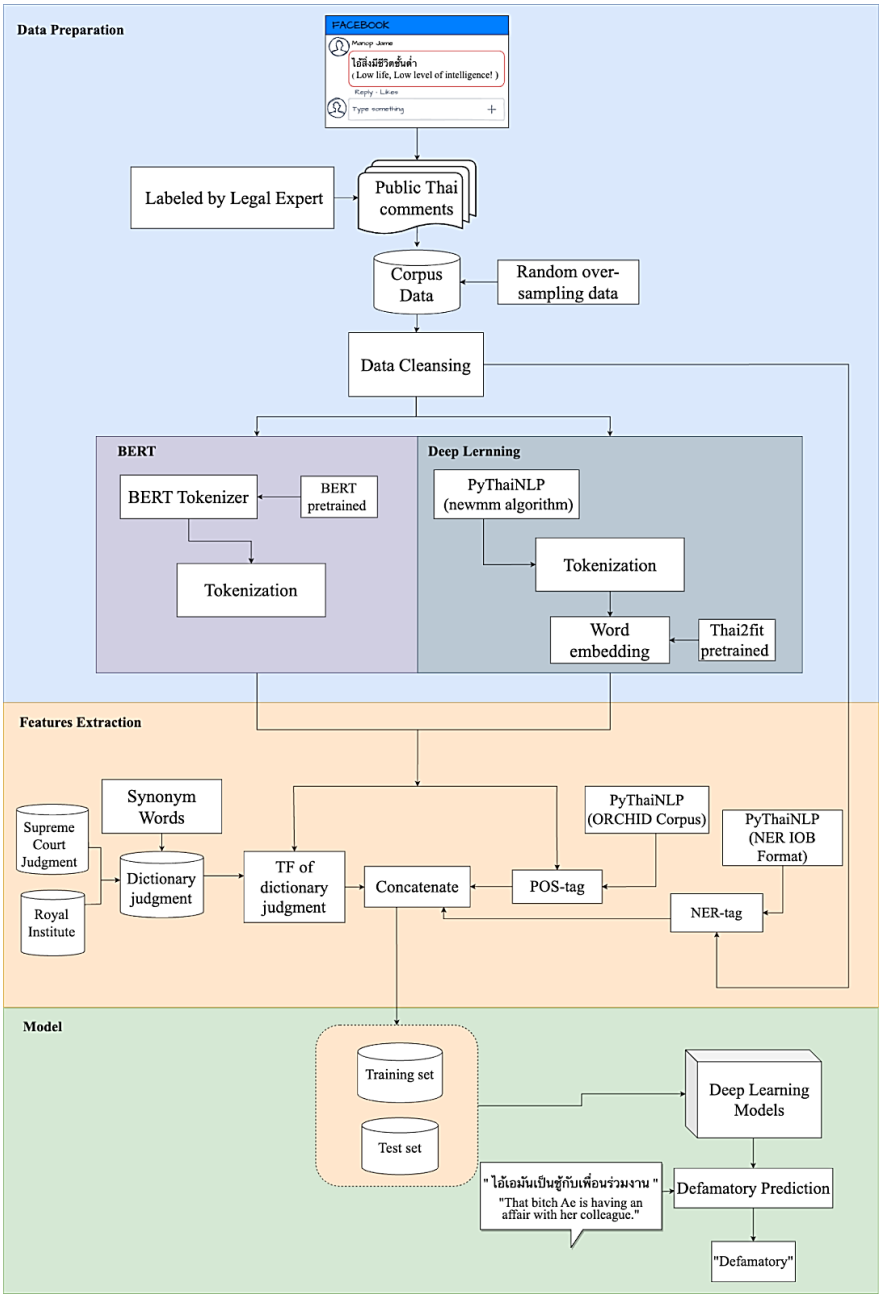
(ที่มาของภาพ https://pythainlp.github.io/dev-docs/api/tag.html)



ภาพที่ 15 การใช้ Name Entity Tag โดย PythaiNLP

บทที่ 4
วิธีการดำเนินงานวิจัย

ในบทนี้จะกล่าวถึงวิธีการดำเนินงานวิจัย โดยจะแบ่งออกเป็น 3 ขั้นตอนหลัก ดังจะกล่าวในหัวข้อ ขั้นตอนการเตรียมข้อมูล, การสกัดคุณลักษณะของข้อความ, การออกแบบ และพัฒนาแบบจำลองสำหรับจำแนกประโยค หรือข้อความที่เข้าข่ายหมิ่นประมาท แสดงภาพรวมของกระบวนการทำงานดังภาพที่ 16



ภาพที่ 16 แสดงภาพรวมกระบวนการทำงานของงานวิจัยนี้

4.1 การเตรียมข้อมูลที่ใช้ในงานวิจัย

4.1.1 การเก็บรวบรวมข้อมูล (Data Collects)

ชุดข้อมูลที่ใช้ในการเรียนรู้ของแบบจำลองสำหรับการจำแนกข้อความที่หมิ่นประมาท ผู้วิจัยได้เก็บรวบรวมมาจากแพลตฟอร์มผู้ให้บริการสื่อสังคมออนไลน์ ที่ได้รับความนิยมคือ เฟสบุ๊ค ตั้งแต่ปลายปี 2020 - 2022 โดยเก็บรวบรวมข้อความการแสดงความคิดเห็นของผู้ใช้แบบสาธารณะที่มีการโพสแบบสาธารณะ ทั้งในหน้าโปรไฟล์ของตน ที่ผู้อื่นสามารถเข้าถึงโพสดังกล่าวได้ รวมถึงโพสในกลุ่มสาธารณะ หรือในเพจเฟสบุ๊คต่าง ๆ ที่มีการตอบโต้กันในหลากหลายประเด็น ซึ่งแสดงตัวอย่างรูปแบบการความคิดเห็นในภาพที่ 17 โดยข้อความการแสดงความคิดเห็นที่เก็บรวบรวมมาจะเป็นหัวข้อเกี่ยวกับบุคคลสำคัญ บุคคลผู้มีชื่อเสียงที่ผู้คนให้ความสนใจ เช่น ดารานักแสดง นักการเมือง ข้าราชการ เน็ตไอดอล เป็นต้น รวมถึงองค์กร บริษัทห้างร้านต่าง ๆ ซึ่งข้อความส่วนใหญ่ที่รวบรวมมาจะมีลักษณะข้อความในเชิงต่อว่า ถากถาง หรือด่าทอ ซึ่งในงานวิจัยนี้ผู้วิจัยได้เก็บรวบรวมข้อความการแสดงความคิดเห็นได้รวมทั้งสิ้น 2,152 ข้อความ และให้ผู้เชี่ยวชาญระบุว่าข้อความใดเข้าข่ายหมิ่นประมาทต่อไป

ภาพที่ 17 การแสดงความคิดเห็นเฟสบุ๊ค

4.1.2 การระบุข้อความที่เข้าข่ายหมิ่นประมาท

ในขั้นตอนนี้ผู้วิจัยได้ให้ผู้เชี่ยวชาญทางด้านกฎหมาย ได้แก่ ทนายความ และอัยการ ร่วมกันลงความเห็นว่าประโยคใดเข้าข่ายหมิ่นประมาท โดยจะใส่ผลเฉลยประโยคที่เข้าข่ายหมิ่นประมาทให้มีค่าเป็น 1 และประโยคที่ไม่เข้าข่ายหมิ่นประมาทให้มีค่าเป็น 0 ซึ่งผู้เชี่ยวชาญจะพิจารณาจากองค์ประกอบความผิดของประมวลกฎหมายหมิ่นประมาทมาตรา 326 เป็นหลัก ซึ่งในมาตราดังกล่าวจะมีลักษณะองค์ประกอบที่เข้าข่ายความผิดดังต่อไปนี้

1) ผู้กระทำความผิด หรือผู้ใส่ความ เป็นได้ทั้งบุคคลธรรมดา และนิติบุคคล

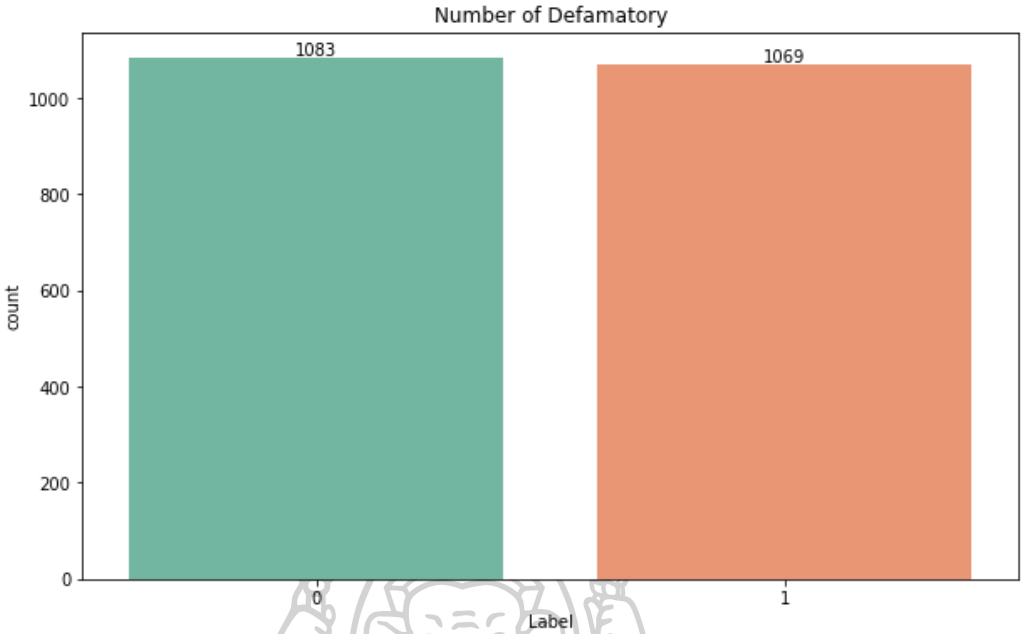
2) การใส่ความ หรือถ้อยคำในประโยคที่ทำให้ผู้อื่นได้รับความเสียหาย ซึ่งการใส่ความนั้นจะเป็นเรื่องจริง หรือเรื่องเท็จก็ตาม ถ้าผู้อื่นได้รับความเสียหาย ก็จะมีความผิดฐานหมิ่นประมาท เช่น นาย A มีเพศสัมพันธ์เชิงชู้สาวกับนาย B และนาย C มาเห็น จึงนำเรื่องราวดังกล่าวไปบอกกับผู้อื่นว่า นาย A และ นาย B เป็นชู้กัน ก็จะถือได้ว่านาย C ได้กระทำการใส่ความที่ทำให้ผู้อื่นได้รับความเสียหาย หรือเสื่อมเสียชื่อเสียง ก็จะถือว่าเข้าข่ายความผิดฐานหมิ่นประมาท

3) ผู้อื่น หรือผู้ถูกใส่ความ เป็นกรรมของการกระทำ โดยแนวคิดของคำพิพากษาศาลฎีกาจะต้องทราบได้ว่าหมายถึงผู้ใด โดยไม่จำเป็นต้องระบุชื่อบุคคลนั้น โดยจะเป็นบุคคลคนเดียว หรือหลายคนก็ได้

4) ต่อบุคคลที่สาม โดยความผิดฐานหมิ่นประมาทตาม มาตรา 326 ได้ระบุว่าการใส่ความผู้อื่นจะได้กระทำต่อบุคคลที่สาม ซึ่งหากผู้กระทำผิดได้แพร่กระจายข้อความที่ใส่ความผู้อื่น ให้บุคคลที่สามได้รับรู้ จะถือว่าเข้าข่ายในองค์ประกอบนี้

5) ทำให้ผู้อื่นได้รับความเสียหาย โดยหากคำพูดข้อความของผู้กระทำผิด ทำให้ผู้ถูกใส่ความไม่ว่าจะเป็นความจริง หรือเท็จได้รับความเสียหาย ลดคุณค่า หรือเกียรติยศของผู้ถูกใส่ความให้ลดน้อยลง ก็จะเข้าข่ายความผิดฐานหมิ่นประมาท

นอกจากนี้ผู้เชี่ยวชาญจะพิจารณาข้อความที่เข้าข่าย จากการเปรียบเทียบความผิดกับคำพิพากษาศาลฎีกาที่ได้ตัดสินว่ามีความผิด ตามประมวลกฎหมายอาญา มาตรา 326 และ มาตรา 328 ด้วย โดยหลังจากที่ผู้เชี่ยวชาญร่วมกันลงความเห็นเพื่อระบุข้อความที่หมิ่นประมาทในชุดข้อมูล สรุปว่าในชุดข้อมูลที่เก็บรวบรวมมามีข้อความที่เข้าข่ายหมิ่นประมาทจำนวน 1,069 ข้อความ และไม่เข้าข่ายหมิ่นประมาทจำนวน 1,083 ข้อความ ดังแสดงในภาพที่ 18



ภาพที่ 18 จำนวนข้อมูลที่ผ่านการ Label โดยผู้เชี่ยวชาญ

ตารางที่ 1 แสดงตัวอย่างชุดข้อมูลที่ผ่านการ Label โดยผู้เชี่ยวชาญ

Label	Sentence
0	ไร้ยางอาย บอกจะเข้ามารักษาความสงบ สุดท้ายเลือกข้างชัดเจน บอกสุเทพเป็นคนดี หมดศรัทธา เสียใจที่เคยมองว่าท่านดี
0	เป็นใครหรออิป้า ถึงมีสิทธิ์มาบอกเค้าผ่านไม่ผ่าน จะว่าอะไรใครดูน้ำหมากในปากตัวเองก่อน
0	เกิดมาเพื่อกลายเป็นขยะเปียกของโลก
1	ไอ้ธนาธรไอ้เจ๊กกบฏ
1	บิ๊กโจ๊กมึงนี่ตัวดีเลยโกงเงินรับส่วยสารพัด
0	อย่าเป็นเลยผู้นำประเทศอะ เป็นคนให้ได้ก่อน
1	ทหารเหี้ยๆๆ อย่างมึง ได้แต่โกงกินบ้านเมือง
1	ไอ้ไพร์วัลย์สึกมาเป็นควายเถอะพระเปรต
0	นิสัยเหมือนพ่อมึงไม่มีผิดไอ้แดง
0	สโลแกนแม่งประมาณนี้เลย เป็นรมว.ที่ปากดีไปวันๆ บุคลากรที่เค้าทำงาน เค้าเหนื่อยกันแค่ไหน มึงรู้ไหม เค้าต้องแบกรับความเสี่ยงมากขนาดไหน

4.1.3 การทำความสะอาดข้อมูล

ในหัวข้อนี้จะอธิบายถึงขั้นตอนการเตรียมข้อมูล เพื่ออยู่ในสภาพพร้อมสำหรับการฝึกสอน และเพื่อให้คุณภาพของแบบจำลองมีประสิทธิภาพสูงขึ้น โดยข้อความในชุดข้อมูล ต้องผ่านกระบวนการทำความสะอาด เนื่องจากประโยค และข้อความจากเฟสบุ๊คที่รวบรวมมา จะมีอักขระพิเศษ หรือสัญลักษณ์ที่ไม่จำเป็นในการเรียนรู้ของแบบจำลอง เช่น แฮชแทค ตัวเลข เครื่องหมายอักขระพิเศษต่าง ๆ เป็นต้น ซึ่งเป็นสาเหตุทำให้เกิดความผิดพลาดในขั้นตอนการตัดคำ (Tokenize) โดยในขั้นตอนการทำความสะอาดข้อมูลมีรายละเอียดขั้นตอนการทำงานดังนี้

1) ลบแฮชแทค ซึ่งแฮชแทคในข้อความ นิยมใช้เพื่อเน้นคำสำคัญเพื่อการแท็กเนื้อหาที่เกี่ยวข้อง แต่อาจไม่จำเป็นสำหรับการวิเคราะห์ความหมายของข้อความ

2) ลบข้อความที่ไม่ใช่ภาษาไทย ในงานวิจัยนี้มุ่งเน้นการจำแนกข้อความภาษาไทยที่เข้าข่ายหมิ่นประมาทตามประมวลกฎหมายอาญาของไทย ภาษาอื่น ๆ จึงไม่มีประโยชน์สำหรับการเรียนรู้ของแบบจำลอง และอาจทำให้ขั้นตอนการตัดคำผิดพลาดได้

3) ลบอักขระพิเศษ ที่ไม่มีความหมายในบริบทของข้อความ หรือไม่เกี่ยวข้องกับการวิเคราะห์จะถูกลบออก เพื่อลดความซับซ้อน และปรับปรุงคุณภาพของข้อมูลสำหรับการฝึกสอน

4) ลบสัญลักษณ์แสดงความรู้สึก (Emoji) ซึ่งใช้ในการแสดงความรู้สึก หรืออารมณ์ในข้อความ แต่จะไม่นำมาใช้ในการจำแนกข้อความที่หมิ่นประมาท ผู้วิจัยจึงลบสัญลักษณ์แสดงความรู้สึกออกจากข้อความ

เมื่อทำความสะอาดข้อมูลในชุดข้อมูลเสร็จ จึงได้ข้อมูลที่พร้อมสำหรับการตัดคำในภาษาไทยในขั้นตอนต่อไป

4.1.4 การตัดคำ (Tokenize)

ในการวิเคราะห์ข้อความเพื่อจำแนกประเภท จำเป็นต้องแยกคำในข้อความออกจากกัน และแปลงเป็นตัวเลข หรือเวคเตอร์ เพื่อนำข้อมูลคำศัพท์เป็นข้อมูลเข้าเพื่อให้แบบจำลองได้เรียนรู้ต่อไป โดยในภาษาที่มีโครงสร้างชัดเจนอาจจะไม่มีปัญหาเรื่องการตัดคำ เช่น ภาษาอังกฤษ ที่มีการแยกคำอยู่แล้ว แต่ในภาษาไทยนั้น ปัญหาหนึ่งของการทำวิจัยทางภาษาคือการตัดคำ เพราะเป็นภาษาที่เขียนเรียงติดกันไม่มีการเว้นวรรคของคำ ดังนั้นในการแบ่งคำภาษาไทยจึงจำเป็นต้องอาศัย algorithm ที่แบ่งคำอย่างแม่นยำ ในงานวิจัยนี้ผู้วิจัยจึงเลือกใช้การตัดคำโดย PyThaiNLP library[35] โดยใช้อัลกรอริทึม nemm ซึ่งเป็นอัลกอริทึมหนึ่งใน PyThaiNLP ที่ใช้สำหรับการตัดคำภาษาไทย โดยชื่อ “New Maximum Matching” มาจากอัลกอริทึม “Maximum Matching” ซึ่งเป็นวิธีการตัดคำโดยการหาคำที่ยาวที่สุดที่ตรงกับ dictionary และแบ่งคำตามนั้น โดย newmm ได้รับการปรับปรุง และเพิ่มเติมเพื่อให้สามารถจัดการกับปัญหาต่าง ๆ เช่น การแยกคำที่มีหลาย

ความหมาย หรือคำที่มีการใช้งานในหลากหลายรูปแบบได้ดีขึ้น โดยตัวอย่างการตัดคำโดยอัลกรอริทึม ดังกล่าวแสดงในภาพที่ 19

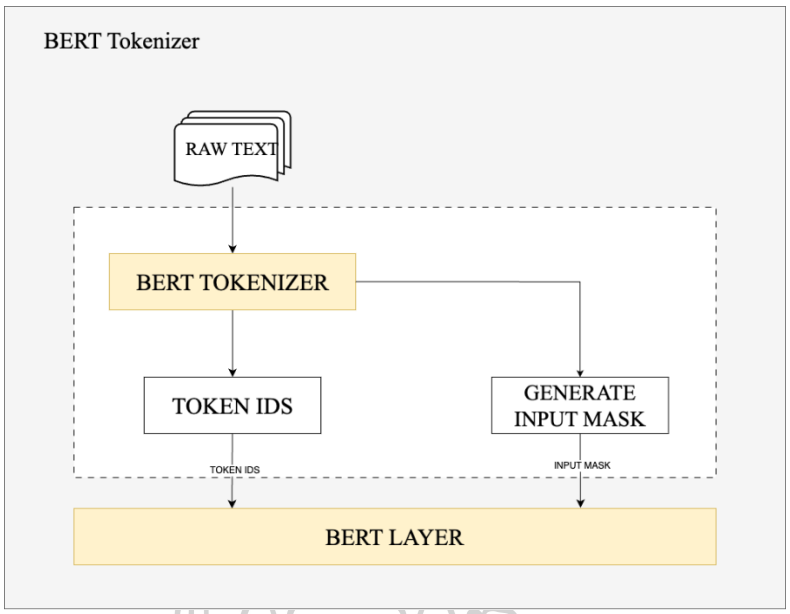
ภาพที่ 19 ตัวอย่างการตัดคำโดย newmm

4.1.5 BERT Tokenize

ในหัวข้อนี้วิจัยจะเตรียมข้อมูลให้อยู่ในรูปแบบที่เข้าใจได้ด้วยแบบจำลอง BERT โดยหลังจากผ่านกระบวนการทำความสะอาดข้อมูลแล้ว ข้อความต่าง ๆ จะเข้าสู่กระบวนการ BERT Tokenize โดยการแบ่งข้อความออกเป็นส่วนย่อยที่เรียกว่า "token" ซึ่งแต่ละ token จะมีความหมายที่เฉพาะเจาะจง โดยจะมีขั้นตอนคือ

1. แบ่งประโยคเป็น Token จะแยกประโยคออกเป็น token โดยใช้กฎเกณฑ์ที่กำหนดโดย BERT ซึ่งสามารถแบ่งออกเป็นคำ หรือส่วนย่อยที่สำคัญของข้อความได้
2. แปลง Token เป็นตัวเลขแต่ละ token จะถูกแปลงเป็นตัวเลขตามลำดับของ vocabulary ที่กำหนดไว้ล่วงหน้าจาก Pretrained BERT Model
3. สร้าง Attention Mask สร้างเมทริกซ์ที่จะระบุให้แบบจำลองรู้ว่าแต่ละ token ในประโยคเป็นส่วนสำคัญ หรือไม่ เพื่อให้โมเดลสนใจ และไม่สนใจข้อมูลที่ไม่สำคัญ

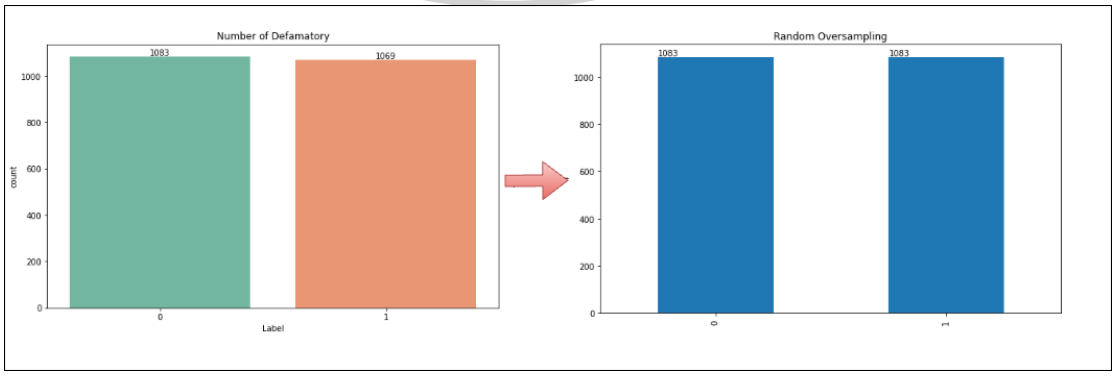
หลังจากเสร็จกระบวนการ จะได้ input_ids และ attention_mask สำหรับเป็นข้อมูลเข้า BERT Layers ต่อไป ดังแสดงในภาพที่ 20



ภาพที่ 20 กระบวนการทำ BERT Tokenizer

4.1.6 การแก้ปัญหาข้อมูลไม่สมดุล (Imbalanced Dataset)

ในหัวข้อนี้จะอธิบายวิธีการแก้ปัญหาข้อมูลไม่สมดุล เนื่องจากการเก็บรวบรวมข้อมูลข้อความที่เข้าข่ายหมิ่นประมาทนั้น มีองค์ประกอบที่จำเป็นหลายประการในข้อความจึงจะเข้าข่ายความผิด ทำให้ชุดข้อมูลที่ได้หลังจากการ Label จากผู้เชี่ยวชาญว่าเข้าข่ายหมิ่นประมาท จึงมีปริมาณน้อยกว่าที่ไม่เข้าข่ายหมิ่นประมาท ทำให้เกิดปัญหาชุดข้อมูลไม่สมดุล ผู้วิจัยจึงเลือกใช้ oversampling ในการทำชุดข้อมูลให้สมดุล เพื่อลดการเกิดปัญหาปัญหา overfiting โดยสุ่มหยิบข้อมูลจากคลาสที่น้อยกว่าคือคลาส 1 เพื่อเพิ่มข้อมูลให้เท่ากับคลาส 0 โดยสุ่มหยิบข้อมูลมาทั้งหมด 14 ข้อความ หลังจากทำ Random Oversampling ผู้วิจัยจึงได้ชุดข้อมูลที่ 2 คลาส มีประโยคเท่ากันคือคลาสละ 1,083 ข้อความ รวมทั้งหมด 2,166 ข้อความ แสดงในภาพที่ 21



ภาพที่ 21 การทำ Random Oversampling

4.2 การสกัดคุณลักษณะของข้อความ (Features Extraction)

4.2.1 พจนานุกรมฎีกา (Dictionary Judgement)

ในการพิจารณาคดีที่เกี่ยวข้องกับการหมิ่นประมาท ผู้พิพากษาจะดูองค์ประกอบของประโยคข้อความ หรือคำพูดว่าเข้าข่ายความผิด หรือไม่ ซึ่งในองค์ประกอบความผิดในเรื่องการใส่ความผู้พิพากษาจะนำคำศัพท์ที่ใช้ใส่ความมาอธิบาย และให้ความหมายของคำนั้นเพื่อชี้มูลความผิดของจำเลย แสดงในภาพที่ 22 ดังนั้นในงานวิจัยนี้ผู้วิจัยจึงเก็บรวบรวมคำศัพท์จากฎีกาในคดีที่เกี่ยวข้องกับการหมิ่นประมาทตามประมวลกฎหมายอาญามาตรา 326 และมาตรา 328

จำเลยพูดถึงผู้เสียหายซึ่งเป็นหญิงรับราชการอยู่ที่สำนักงานที่ดินอำเภอว่า กระหรี่ที่ดิน และคำว่า "กระหรี่"หมายความว่า หญิงนครโสเภณีหรือหญิงค้าประเวณี แม้จำเลยจะไม่ได้กล่าวรายละเอียดว่าค้าประเวณีกับใคร ประพฤติล่วนในทางเพศกับใครบ้าง ก็เพียงพอที่จะถือว่าเป็นคำหมิ่นประมาทแล้ว

The defendant spoke of the victim, a woman serving at the district land office that Karee land and the word "Karee" means prostitute or prostitute Although the defendant did not specify the details of prostitution with anyone. Promiscuous behavior with someone It is enough to be considered defamation.

ภาพที่ 22 คำพิพากษาศาลฎีกาที่ 2371/2522

นอกจากนั้นผู้วิจัยได้เก็บรวบรวมคำศัพท์เพิ่มเติมจากฎีกาที่เกี่ยวข้องกับประมวลกฎหมายอาญามาตรา 393 ซึ่งส่วนใหญ่มักมีคำศัพท์ประเภท คำหยาบคาย คำดูถูก หรือคำดูหมิ่นเหยียดหยาม ซึ่งการจัดกลุ่มในการทำพจนานุกรมคำพิพากษานี้ ผู้วิจัยได้ศึกษางานวิจัยของ Arreerard et al., [36] ที่เสนอการแบ่งกลุ่มพจนานุกรมไว้ โดยหลังจากการศึกษางานวิจัยดังกล่าวผู้วิจัยจึงจัดคำศัพท์เป็น 7 กลุ่มดังต่อไปนี้

1. คำนาม คำที่ใช้เรียก หรือระบุถึง คน สัตว์ สถานที่ คุณสมบัติ สถานะ อาการ หรือแนวคิด
2. สกรรมกริยา (Transitive verb) คำกริยาที่ต้องพึ่งกรรมมารองรับ เพื่อทำให้ประโยคสมบูรณ์ เช่น นาย A “สังหาร” นายB
3. อกรรมกริยา (Intransitive verb) คำกริยาที่ไม่ต้องพึ่งกรรมมารองรับ เพื่อทำให้ประโยคสมบูรณ์ เช่น นาย A “ไม่ซื่อสัตย์”
4. วลี หรือกลุ่มคำ คำที่นิยมใช้ในการประชดประชัน หรือเปรียบเปรย เป็นคำที่ใช้สบประมาทโดยอ้อม เช่น “กินบนเรือนขี้บนหลังคา” หมายถึงการอกตัญญู หรือการเนรคุณ

5. คำสบประมาท (Insult) คำที่มีความหมายในทางสบประมาทดูถูเหยียดหยามผู้อื่น คำด่า และคำหยาบ

6. คำสรรพนามบุคคลที่หนึ่ง คำที่ใช้เรียกตนเองเป็นคำที่ระบุถึงตัวของผู้หมิ่นประมาท เช่น กู ข้า

7. คำสรรพนามบุคคลที่สองและสาม คำที่ใช้เรียกผู้ถูกใส่ความทั้งคนเดียว และหลายคน เช่น มึง พวกมึง

โดยหลังจากเก็บรวบรวมคำศัพท์จากคำพิพากษาศาลมาแบ่งประเภทแล้ว ผู้วิจัยได้เพิ่มคำพ้องความหมาย(synonym) และคำที่คนมักพิมพ์ผิดบ่อย ๆ นอกจากนั้นผู้วิจัยได้เพิ่มคำศัพท์จากพจนานุกรมราชบัณฑิตยสถาน ซึ่งเขาเก็บรวบรวมคำศัพท์ภาษาไทยตั้งแต่คำโบราณ คำที่เลิกใช้แล้ว และคำที่ยังใช้อยู่ในปัจจุบัน สุดท้ายผู้วิจัยได้พจนานุกรมคำพิพากษาที่ประกอบด้วย คำนาม 452 คำ คำสกรรมกริยา 155 คำ คำอกรรมกริยา 269 คำ วลีหรือกลุ่มคำ 10 คำ คำสบประมาท 59 คำ คำสรรพนามบุคคลที่หนึ่ง 28 คำ และคำสรรพนามบุคคล จำนวน 63 คำ

ตารางที่ 2 แสดงคำนาม 452 คำ

คำนาม
คนพาล, คนสารเลว, คนไม่เอาถ่าน, คนบ้า, คนเสียจริต, ผู้วิกลจริต, คนหัวรุนแรง, พวกหัว, รุนแรง , กิ๊ก, เมียเก็บ, แมงดา, คนเกาะผู้หญิงกิน, คนให้ผู้หญิงเลี้ยง, พ่อเล้า, คนคุมซ่อง, นักเลงคุม, ซ่อง, คนไม่เต็ม, คนไม่เต็มบาท, คนโกง, คนคดโกง, คนหลอกลวง, นักต้มตุ๋น, ขโมย, หัวขโมย, ขโมยข, โจร, คนขี้ขโมย, โจรลักขโมย, คนร้าย, ผู้ร้าย, คนนอกกฎหมาย, คนทาผิดกฎหมาย, ผู้กระทาความผิด, ฆาตกร, นักฆ่า, คนเลว, คนชั่ว, ปชช., ประชาชน, คนชนบท, ชาวบ้าน, คนบ้านนอก, คนต่างจังหวัด, คนในหมู่บ้าน, ตัวโกง, นักโทษ, ผู้ต้องโทษ, ผู้ถูกคุมขัง, ซ่อง, หอนางโลม, ซ่องโสเภณี, ซ่องโจร, ที่กบดาน, ที่ซ่อนตัว, รังโจร, รังอาชญากร, อาบอบนวด, สถานอาบอบนวด, คุก, ซังเต, ห้องขัง, เรือนจำ, ของปลอม, ของเก๊, ของเทียม, ของปลอมแปลง, ชา, หลุม, คนหยาบคาย, ศัตรู, ข้าศึก, สปาย, สายลับ, คนทรยศ, ผู้ทรยศ, คนขายชาติ, คนทรยศชาติ, จราจล, การจราจล, หนี้, หนี้สิน, ปีศาจ, วิญญาณร้าย, ผี, ผีสาง, วิญญาณ, อมนุษย์, คนเลวทราม, คนถ่อย, อันธพาล, คนไร้สกุลรุน ชาติ, ตุ๊ด, กระเทย, ยาเสพติด, ยา, ไก่, อีตัว, โสเภณี, นางคณิกา, หญิงบริการ, ผู้หญิงหากิน, ผู้หญิงขายตัว, หญิงขายบริการทางเพศ, กุ๊ย, นักเลง, นักเลงโต, โจ๋อันธพาล, นักเลงหัวไม้, ผู้ก่อการร้าย, อีหนู, อนุภรรยา, นางบาเรอ, สาวฮาเร็ม, รัฐ, ชาติ, ประเทศ, รัฐชาติ, ประเทศชาติ, สัญชาติ, คนโง่, ไอ้, ควาย, คนสมองทึบ, หัวขี้เลื่อย, คนโง่เขลา, ตัวตลก, คนโง่เซ่อ, คนเซ่อซ่า, คนไร้ค่า, คนไม่เอาไหน, คนน่ารังเกียจ, คาสั่ง, มิสชั่น, โครงการ, ภาระ, หน้าที่, บ้าน, เรือน, บ้านช่อง, ศีลธรรม, จริยธรรม, คนไร้การศึกษา, มาลีฮวนน่า, ต้นกัญชา, ส่วนได้เสีย, ค่า, ประโยชน์, คอมมิวนิสต์, พวกคอมมิวนิสต์, กบฏ, ผู้ก่อกบฏ, ผู้ก่อการกบฏ, เผด็จการ, ลัทธิเบ็ดเสร็จนิยม, จอม

เผด็จการ, ผู้อยู่เหนือกฎหมาย, กฎ, ระเบียบ, กฎระเบียบ, ระเบียบข้อบังคับ, กม., กฎหมาย, สมรรถภาพ, แก๊ง, แก๊งค์อันธพาล, แก็งค์อาชญา

ตารางที่ 3 แสดงคำสกรรมกริยา 155 คำ

คำสกรรมกริยา
คบชู้สู่ชาย, จี้ปล้น, จี้ชิงทรัพย์, จี้, ข่มขืน, ข่มขืนกระทำชำเรา, ทรยศ, หักหลัง, รีดไถ, ป้ายสี, ใส่ความ, กล่าวร้าย, พูดให้ร้าย, ใส่ร้ายป้ายสี, หลีกเลี่ยง, หลบหนีจากที่คุมขัง, ทำอันตราย, หลอกลวง, คิดไม่ซื่อ, ไม่ซื่อสัตย์, โกหก, โป้ปด, ตอแหล, พูดปด, พูดโกหก, กล่าวเท็จ, ปั้นน้ำเป็นตัว, แอบอ้าง, กบฏ, โทรมหญิง, เรียงคิวขืนใจ, เรียงคิวข่มขืน, ยกเค้า, โจรกรรม, ขึ้นบ้าน, ปล้น, ชิงทรัพย์, ปล้นทรัพย์, ปล้นสะดม, ขโมย, ลักขโมย, ลักเล็กขโมยน้อย, ลอบวางเพลิง, ข่มขู่, แบล็กเมล์, เสพ, พี้, ขโมย, ลักทรัพ, ลักทรัพย์, กรรโชกทรัพ, กรรโชกทรัพย์, ต้มตุ๋น, เล่นชู้, ตีท้ายครัว, เล่นหลังบ้าน, คบชู้, สร้างบ้านเล็ก, มีบ้านเล็ก, มีเพศสัมพันธ์, กระทำชำเรา, ขืนใจ, ล่วงละเมิดทางเพศ, เสียตัว, เจาะไข่แดง, ล่าแต้ม, เก็บแต้ม, มีปากเสียง, ลงไม่ลงมือ, ชักชวนในทางที่ผิด, ทุจริต, คดโกง, ละเมิดกฎหมาย, ฝ่าฝืนกฎหมาย, แหกกฎ, ป้ายความผิด, นั่งเทียนเขียนข่าว, สร้างข่าวลือ, ก่อวินาศกรรม, ก่อเหตุอัคคีภัย, วางระเบิด, รุมประชาทัน, พรากชีวิต, ประทุษร้าย, ขู่ฆ่า, ใช้เส้นสาย, ขู่ เข็ญ, ล่วงล้า, ฝ่าเขตแดน, ละเมิดสิทธิ, รุกราน, โกงกิน, เล่นเส้นเล่นสาย, ทรยศแผ่นดิน, รับสินบน, รับใต้โต๊ะ, ฮั้วประมูล, ผิดคาพูด, เล่นของสูง, แอบอ้างเบื่องสูง, ก่อจลาจล, โกง, ขบถ, กบฏ, ตลบหลัง, เนรคุณ, แทงข้างหลัง, ขู่กระโชก, ติดสินบน, ปลิ้นปลอก, ปลิ้นปล้อน, แหกตา, สับปลับ, หลอกต้ม, ลุมโทรม, ลวนลาม, ปลอม, ปลอมแปลง, ต้ม, ตุกติก, ยุยง, ปลุกระดม, ปลุกปั่น, ปั่นหัว, เป่าหู, ยุ, วิ่งราว, ปล้นจี้, ทาชั่ว, วางเพลิง, ขู่ขวัญ, พาล, แบล็คเมล์, แบลคเมล, แบล็คเมล, แบคเมล์, แบคเมล, แบล็คเม, แบลคเม, แบคเม, คอรับชั่น, คอรัปชั่น, คอรับชัน, คอรัปชัน, วิปริต, ฉ้อโกง, ก่อกบฏ, รุมโทรม, ค้าประเวณี, ฟอกเงิน, กระแดะ, กัดไม่ปล่อย, ถ้ำมอง, รูดทรัพย์, โกงบ้านโกงเมือง

ตารางที่ 4 แสดงคำอกรรมกริยา 269 คำ

คำอกรรมกริยา
หนี, ละทิ้ง, ชอบ, โปรด, โปรดปราน, นอน, เย็ด, ตีหม้อ, ร่วมรัก, มีเซ็กส์, ร่วมประเวณี, มีเพศสัมพันธ์, ตื๊อ, ตามจีบ, ตามตื๊อ, นอกใจ, สวมเขา, โจมตี, เข้าโจมตี, ชก, อัด, ตุ๊ย, ฟาดหัว, ฟาดศีรษะ, ตีหัว, ตีศีรษะ, อม, ดูด, เลีย, ฉุด, ย่ายี, ปรนเปรอ, ขาย, ไถ, คุกคาม, เข้ามาคุกคาม, บีบ, กดดัน, บีบคั้น, บีบบังคับ, ลงแขก, สมคบคิด, ฮั้ว, ฉวย, หยิบฉวย, งัด, ดูถูก, ดูหมิ่น, เหยียดหยาม, บิดเบือน, หลอก, ล่อลวง, ก๊อป, ก๊อปปี้, เลียนแบบ, อิจฉา, ตาร้อน, อิจฉาตาร้อน, กุข่าว, ปล่อยข่าว, ประโคมข่าว, ปล่อยข่าวลือ, กุ, จับกุม, จับ, ควบคุม, คว้าตัว, จับตัว, เสือก, นั่ง, เฝ้า, หลับ

นอน, เป็น, คือ, มี, ทะเลาะ, ปะทะคารม, ไล่ออก, เถลไถล, สวม, ขึ้น, กอด, จูบ, จุบ, จุ๊บ, หอม, ได้, รับ, ได้รับ, เอา, กิน, หยิบ, ยึด, ใช้, ลุ่มหลง, หลง, ซื้อ, กินกริบ, ให้, บาเรอ, ลวงตา, ชิง, หมิ่น, แกล้ง, อาพราง, ทา, ปลด, เผา, ยิง, เชื่อ, แสวงหา, บ้า, ขัด, บริหาร, วาง, กล่าวหา, ปฏิบัติ, กระทา, สมคบ, รวมหัว, งุบงิบ, สมรู้, คบ, จ้าง, งาบ, เม้ม, ขาด, ขาดแคลน, ไร้, โอน, แอบ, หอบ, บริจาค, เข้า, ลงโทษ, กลัว, จ้างวาน, ทุบตี, เฆี่ยน, หวด, ซัด, คร่า, เตะ, ถีบ, เปิด, ตดิ, ค้า, อ้าง, บังคับ, แดก, ทิ้ง, แดรก, กก, นอนกก, คลอเคลีย, เคลียเคล้า, โอบกอด, เยด, ร่วมเพศ, ปู้ยี่ปู้ยา, ทาลาย, แย่ง, เหยด, ซี่, ซั่ม, ถูกจับ, โดนจับ, ฉกฉวย, กอบโกย, ฉกชิง, ช่วงชิง, กระชาก, จู่โจม, ต่อย, รุมสกัม, ตบ, วิวาท, ปะทะปะ ทั่ง, ตะลุมบอน, กัดกัน, ทาร้ายกัน, ทาร้าย, หยิบเอา, ทุบตี, ตี, ตบตา, ลวงหลอก, หลงผิด, หน้ามืดตามัว, ไม่ลืมหูลิมตา, ปรนนิบัติ, มั่วสุม, เสแสร้ง, แสร้งทา, รังแก, กดขี่ข่มเหง, ข่มเหงรังแก, ข่มแหง, สังหาร, ฆ่า, ลั่นไก, เหนี่ยวไก, วิ่ง, เผ่น, หลบหนี, ซ่อนตัว, กบดาน, ลักลอบ, ด้นข่าว, กาจัด, สปอย, ก่อไฟ, เผาทาลาย, ละเมิด, รุกล้ำ, ริศยา, คดิคด, ยักยอก, ฉ้อฉล, กินเศษกินเลย, กินนอกกินใน, กะล่อน, คบคิด, สุมหัว, สมรู้ร่วมคิด, ระเบิด, ฉก, ลัก, ระราน, ข่มเหง, ข่มเหงคะเนงร้าย, ขู่, ละลาบล ะล้วง, ล่วงเกิน, จาบจ้วง, ปล้า, มั่วสุม, มุสา, บุก, ขี้จุ๊, ลอกเลียนแบบ, ก็อป, กอป, ก๊อบ, ก็อบ, กอบ, ก็อปปี้, กอปปี้, ก๊อบปี้, ก็อบปี้, กอบปี้, โหน, แย่งชิง, ริษยา, อิจฉาริษยา, ฉ้อ, ล้างผลาญ, กุเรื่อง, หลอย, จิ๊ก, ใส่ไฟ, บุกรุก, กลั่นแกล้ง, ทุบ, กดหัว, ฟาดหัว, สิงสู่, เด็ดหัว, จับหี

ตารางที่ 5 แสดงวลีหรือกลุ่มคำ 10 คำ

วลีหรือกลุ่มคำ
เอาดีใส่ตัว, เอาชั่วใส่คนอื่น, ลิ้นสองแฉก, นกสองหัว, ไร้สมอง, ไม่มีสมอง, เหยียบเรือสองแคม, มวยล้ม, ล้มมวย

ตารางที่ 6 แสดงคำสบประมาท 59 คำ

คำสบประมาท
อีเหี้ย,อี,อิ,เหี้ย,เชี่ย,เฮี้ย,ควย,สัส,สัด,สิส,สืส,สิด,สัต,สัตว์,ควาย,ห่า,เย็ด,แม่ง,แม้ง,แตด,ห่า,หน้าหี,หัวควย,ดอก,เฮงซวย,ไอ้ระยำ,เบื้อก,เบื๊อก,โคตรพ่อ,โคตรแม่,ปอบ,ปรอบ,ปอป,ปอรป,เปรต,อ้วน,จัญไร,จันไร,ควย,เฮงซวย,ส้นตีน,ชาติหมา,อิสัด,มารศาสนา,อยากเย็ดสักที,ไอ้หน้าควย,อีหน้าหมา,อีหน้าหมู,อีชาติหมา,สถุล,อีร้อยควย,ไอ้หน้าโง่,พระหน้าเปรต,พระหน้าผี,ผู้หญิงต่ำๆ,ไอ้ตัวแสบ,ไอ้เบื๊อก,ไอ้ห่า,อิเหี้ย

ตารางที่ 7 แสดงคำสรรพนามบุคคลที่หนึ่ง 28 คำ

คำสรรพนามบุคคลที่หนึ่ง
ผม,ฉัน,อีฉัน,เดี๊ยน,ดิฉัน,กระผม,ข้าพเจ้า,กู,เรา,ข้าพระพุทธเจ้า,อาตมา,หม่อมฉัน,เกล้ากระหม่อม,ข้า,หนู,กุ,นู๋,ข้าเจ้า,ข้าน้อย,ฉาน,ดีฉัน,อิฉัน,ดีฉาน,ตู,ตูข้า,ลูกช้าง,เกล้ากระผม,อาตมภาพ

ตารางที่ 8 แสดงคำสรรพนามบุคคล จำนวน 63 คำ

คำสรรพนามบุคคล
มึง,ท่าน,คุณ,นาย,โจทก์,โจทก์ร่วม,ลูกสะใภ้,เจ้,เจ๊,อีนี่,อีนั่น,ไอ้นี่,ไอ้นั่น,คนนี้,คนนั้น,ไอนี่,ไอ่นี่,ไอนั่น,ไอ่นั้น,ลุง,ป้า,ไอ้แก่,อีแก่,ไอแก่,ไอ่แก่,แก,ป้า,ป้าแก,น้า,อา,เขา,เค้า,พี่,นาง,ตา,ยาย,เธอ,ไอ,ไอ้,ไอ่,เอ็ง,มัน,ไอ้เหี้ยนี่,ไอเหี้ยนี่,ไอ่เหี้ยนี่,พระคุณเจ้า,ฝ่าละอองธุลีพระบาท,ฝ่าพระบาท,ฝ่าละอองพระบาท,พระองค์,มัน,ออเจ้า,หล่อน,เจ้า,เจ้าประคุณ,เจ้าพระคุณ,เจ้าหล่อน,เผือ,แม่เจ้า,ใต้ฝ่าพระบาท,ใต้ฝ่าละอองธุลีพระบาท,ใต้ฝ่าละอองพระบาท,ใต้เท้า

4.2.2 Part-Of-Speech Tag

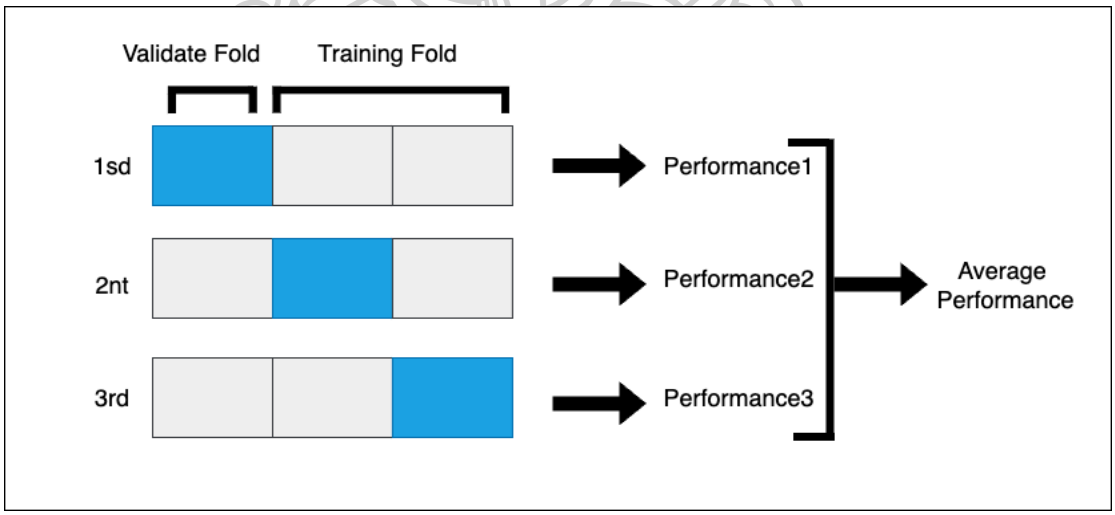
ในงานวิจัยนี้ผู้วิจัยได้เลือกใช้การติดแท็ก POS โดยหวังว่าแบบจำลองจะเข้าใจโครงสร้างของประโยคมากขึ้น โดยจะติดแท็กแต่ละคำในประโยค ว่าคำใดอยู่ส่วนใดของประโยค เช่น คำนาม คำกริยา คำสรรพนาม และคำวิเศษณ์ เป็นต้น ผู้วิจัยได้เลือกใช้ Perceptron tagger ซึ่งเป็นไลบารี่ของ Pythainlp โดยเป็นการติดแท็ก POS โดยใช้ algorithm Perceptron โดยไลบารี่นี้แบ่งคำเป็น 47 ประเภท ซึ่งแบ่งตามคลังข้อมูล ORCHID [34] ซึ่งเป็นคลังข้อความที่มีการกำกับขอบเขตของประโยค ขอบเขตของคำ และชนิดของคำต่าง ๆ ในภาษาไทย โดยในการทดลอง ผู้วิจัยได้ติดแท็ก POS เพื่อเป็นข้อมูลเข้าให้แบบทดลองเรียนรู้ทั้งหมด 2 วิธีได้แก่ pos one-hot ซึ่งผู้วิจัยได้เข้ารหัส pos โดยจะมีขนาด 47 dimensions ซึ่งเป็นขนาดเดียวกันที่ได้จากการติดแท็ก POS ของไลบารี่ Pythainlp ข้างต้น โดยหลังจากการตัดคำ (Tokenization) จากนั้นดูคำแต่ละคำของประโยค หากมีคำที่เข้าองค์ประกอบ จะถูกแทนค่าด้วย 1 ในลำดับของอาเรย์ POS และจะเป็น 0 ในกรณีที่ไม่มีคำในองค์ประกอบนั้น และวิธีที่สอง ผู้วิจัยจะฝังประเภทของคำ (POS Embedding) ซึ่งเป็นการแปลงPOS แต่ละประเภทเป็นเวคเตอร์ โดย represents ถึง POS tag แต่ละประเภท ซึ่งคล้ายกับการทำ Word Embedding ที่แต่ละคำจะ represents ถึงคำในคลังข้อมูล

4.2.3 Word Embedding

ปัจจุบันการสกัดคุณสมบัติของคำโดยการฝังคำ หรือการแปลงคำเป็นเวคเตอร์กำลังได้รับความนิยมอย่างมาก เนื่องจากทำให้เข้าใจความสัมพันธ์ของคำแต่ละคำ โดยในภาษาไทยมีการใช้ Thai2vec โดย pythainlp ซึ่งมันคือการฝึกอบรมล่วงหน้า(Pretrained)กับข้อมูลวิกิพีเดียภาษาไทย โดยใช้วิธี Universal Language Model Fine-tuning (ULMFiT)[33] โดย Thai2Vec มีคำศัพท์จำนวน 60,000 คำในคลังข้อมูล โดยแต่ละคำถูกแทนที่ด้วยเวกเตอร์ 300 มิติ โดยล่าสุดได้มีการพัฒนา และเปลี่ยนชื่อเป็น Thai2fit [37] และปรับขนาดเวคเตอร์เป็น 400 มิติ

4.3 Deep Learning Model

ในขั้นตอนการทดลองนี้เพื่อหาประสิทธิภาพของแบบจำลอง เนื่องจากชุดข้อมูลของงานผู้วิจัยนี้มีขนาดเล็กการแบ่งข้อมูลเพื่อ Train และ Validate อาจได้ข้อมูลที่ยาก หรือง่ายแตกต่างกันไป เพื่อให้ได้ค่าความแม่นยำที่น่าเชื่อถือ และมีความแปรปรวนน้อย จึงใช้การแบ่งข้อมูลเพื่อทดสอบโดยวิธี k-fold cross-validation[38] โดยเลือกแบ่งข้อมูล Train และ Validate ออกเป็น 3 ส่วนเท่า ๆ กัน (k = 3) โดยวิธีการสุ่ม แสดงในภาพที่ 23 หลังจากนั้นจะสร้างแบบจำลอง และทดสอบแบบจำลองจนกว่าข้อมูลทุก fold จะถูกนำมาฝึกสอน และทดสอบจนครบ



ภาพที่ 23 5-fold cross validation

4.3.1 Features Extraction Setting

1. Specific Term frequency of Dictionary judgement

การนำเอาพจนานุกรมคำพิพากษามาประยุกต์ใช้เป็นข้อมูลเข้าสำหรับแบบจำลอง โดยผู้วิจัยจะกำหนดเวกเตอร์ขนาด 7 มิติตามประเภทของคำในพจนานุกรมคำพิพากษาที่อธิบายไว้ในบทที่ 3

โดยคำในประโยคหลังจากการแบ่งคำ(Tokenization)แล้ว จะนำแต่ละคำมาค้นหาในพจนานุกรม หากพบคำจะนับความถี่ไว้ เช่น [3,0,2,0,2,1,1] เมื่อนับจำนวนคำที่พบในพจนานุกรมเสร็จสิ้นแล้ว ทุกค่าในเวกเตอร์จะถูกทำ Normalization โดยวิธีการหารค่าความถี่ทั้งหมด (dividing all the values of frequency) นั้นจะได้เมทริกซ์ขนาด 7 มิติ ที่แต่ละค่ามีค่าไม่เกิน 1

2. POS Tag Embedding

กระบวนการทำPOS-tag embedding สำหรับใช้ในการทดลอง โดย POS-tag embedding หรือเวกเตอร์ของประเภทของคำ (Part of Speech-tag embedding) เป็นเทคนิคที่ใช้ในการแทนคุณลักษณะ หรือประเภทของคำในประโยค เช่น คำนาม (noun) คำกริยา (verb) คำสรรพนาม (pronoun) และอื่น ๆ โดยการนำเอาประเภทคำนี้มาแทนแต่ละคำในประโยคด้วยเวกเตอร์ เพื่อแสดงถึงคุณลักษณะของคำนั้น ๆ ผู้วิจัยได้กำหนดให้เวกเตอร์ของประเภทของคำ (POS-tag embedding) มีขนาด 48 มิติ ซึ่งเป็นขนาดที่ใกล้เคียงกับ คลังข้อมูล ORCHID ที่มีขนาด 47 มิติ โดยผู้วิจัยได้เพิ่มมา 1 มิติเพื่อใส่ไว้ในกรณีไม่เจอคำที่ไม่รู้จัก

3. POS Tag One-hot

การสร้างเวกเตอร์ POS One-hot โดยจะสร้างเวกเตอร์ขนาด 47 มิติขึ้นซึ่งมีขนาดเท่ากับประเภทของคำใน ORCHID โดยหลังจากผ่านกระบวนการทำ Tokenization แล้วประโยคแต่ละชุดจะมีเวกเตอร์ขนาด 47 มิติเท่ากัน โดยจะนำคำในประโยคมาติด Tag POS โดย PythaiNLP และจะจับคู่กันกับลำดับของประเภทคำใน ORCHID หากในประโยคมีประเภทคำนั้น ๆ อยู่ เวกเตอร์ช่องนั้นจะมีค่าเท่ากับ 1 และหากไม่พบคำประเภทนั้น ๆ ในประโยคเลย เวกเตอร์ช่องนั้นจะมีค่าเท่ากับ 0

4.Name Entity One-hot

ผู้วิจัยได้สร้างเวกเตอร์สำหรับ Name Entity (NER) แบบ One-hot โดยเวกเตอร์ดังกล่าวจะมีขนาด 24 มิติ ซึ่งจะเป็นขนาดเดียวกับ NER ที่ได้จากติด tag จาก PythaiNLP ด้วยวิธี IOB format ที่อธิบายไว้ในบท 3 ซึ่งในเวกเตอร์นี้แต่ละช่องจะแทนชนิดของ tag แต่ละประเภทเช่น 'B-DATE', 'I-DATE', 'B-TIME', 'I-TIME' เป็นต้น โดยหลังจากผ่านกระบวนการทำ Data Cleansing แล้วประโยคข้อความจะถูกนำมาติด Tag จากนั้นจะจับคู่กันกับประเภทของ NER หากมีประเภทของ NER อยู่ในประโยค ช่องเวกเตอร์จะมีค่าเท่ากับ 1 และช่องเวกเตอร์ประเภทคำที่ไม่มีจะมีค่าเท่ากับ 0

5. Word Embedding

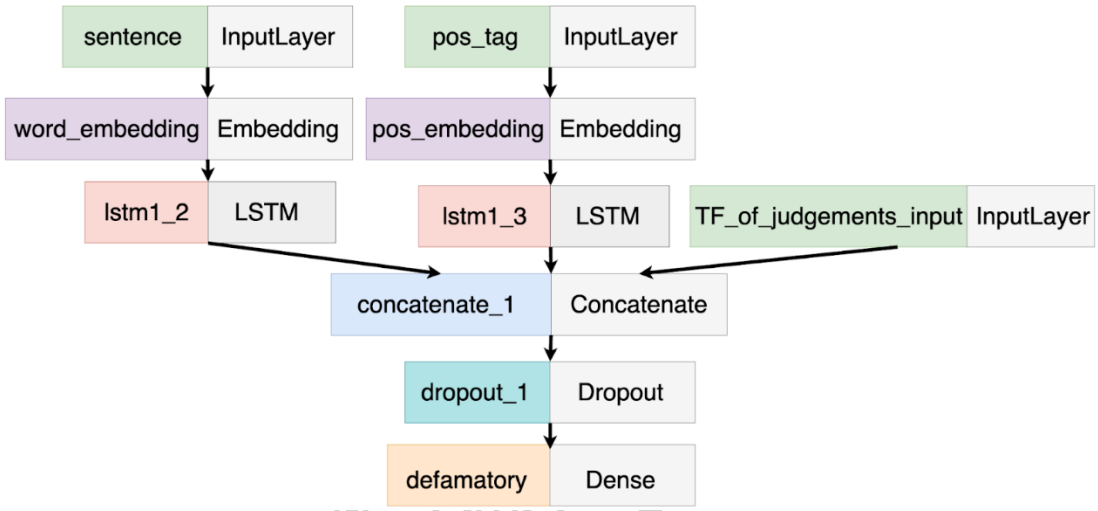
กระบวนการฝังคำที่ใช้ในงานวิจัย โดยการฝังคำที่ใช้ในแบบจำลองการเรียนรู้เชิงลึกทั้ง 3 วิธี คือ LSTM, Bi-LSTM และ CNN เพื่อให้คอมพิวเตอร์เข้าใจความหมายของคำโดยภาษาของมนุษย์

โดยวิธีการแปลงคำเป็นเวกเตอร์ ผู้วิจัยได้กำหนด word vectors ขนาด 400 มิติ ที่มาจาก Thai2fit ซึ่งเป็น Pretrained Word Embedding ที่ถูกฝึกสอนด้วยข้อมูลวิกิพีเดียภาษาไทย และการกำหนดความยาวของประโยคหลังจากการทำ Tokenization เท่ากับความยาวของประโยคที่มากที่สุด เพื่อให้ข้อมูลที่ป้อนเข้า เข้าสู่โมเดลมีขนาดเหมือนกันทุกครั้ง นอกจากนี้ผู้วิจัยจะสร้าง embedding matrix ที่ถูกนำไปใช้เป็น weights ในการทำ Word Embedding โดยการจับคู่ word vector จาก Thai2fit และการสุ่ม vector ใหม่สำหรับคำที่ไม่พบในพจนานุกรมของ Thai2fit ในการตั้งค่าพารามิเตอร์สำหรับการทำ Word Embedding ผู้วิจัยจะปิดการอัพเดทตุ้มน้ำหนักของชุดข้อมูลขณะการฝึกสอน เพื่อให้โมเดลไม่ได้รับการปรับปรุงเพิ่มเติมที่อาจทำให้เกิด overfitting ในข้อมูลได้

4.3.2 การตั้งค่าพารามิเตอร์สำหรับฝึกแบบจำลอง

1. LSTM

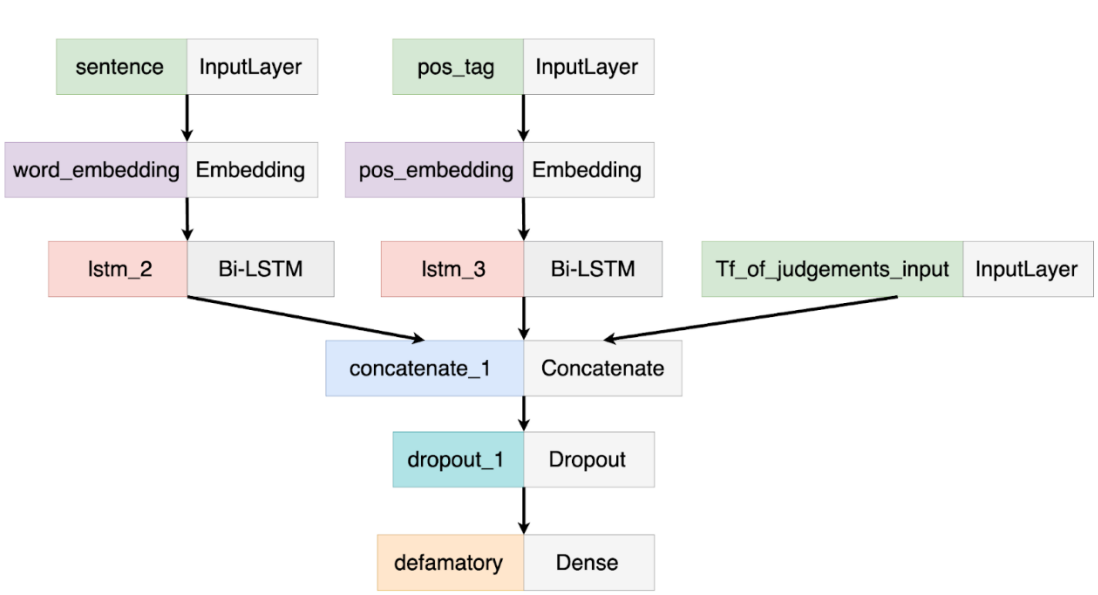
ในการสร้างแบบจำลอง LSTM ใช้ Layer LSTM เพื่อประมวลผลลำดับของข้อมูลเชิงลำดับ โดยรับ input layers ที่ป้อนข้อมูลเข้าสู่โมเดล ประกอบด้วย sentence_input เป็น input สำหรับข้อความ และ features input อื่น ๆ ที่ผู้วิจัยเสนอเป็น input เพิ่มเติม ส่วนของประโยคที่เป็นข้อมูลเข้าใช้ layers Embedding เพื่อแปลงข้อความให้อยู่ในรูปแบบเวกเตอร์ (embeddings) ซึ่งจะถูกนำเข้าไปในชั้น LSTM เพื่อเรียนรู้ และทำนายผลลัพธ์ หลังจากนั้นจะผ่านชั้น Layer เพื่อลดลำดับของคำที่ถูกฝังอยู่ในข้อความเป็นเวกเตอร์เดียวที่มีขนาด 128 มิติ โดยใช้เทคนิค LSTM จากนั้นผู้วิจัยจะรวม features ทั้งหมดเข้าไว้ในเวกเตอร์เดียวกันผ่าน layer concatenation และเพิ่มชั้น Dropout เพื่อลดการเกิด overfitting และกำหนดชั้น Dense เป็นชั้น Classification ขนาด 1 มิติ สำหรับทำนายว่าข้อความดังกล่าวเป็นหมิ่นประมาท หรือไม่ โดยใช้ activation function เป็น sigmoid โดยจะแสดงภาพการสร้างแบบจำลอง LSTM โดยรวมในภาพที่ 24



ภาพที่ 24 โครงสร้างและพารามิเตอร์ของแบบจำลอง LSTM

2. Bi-LSTM

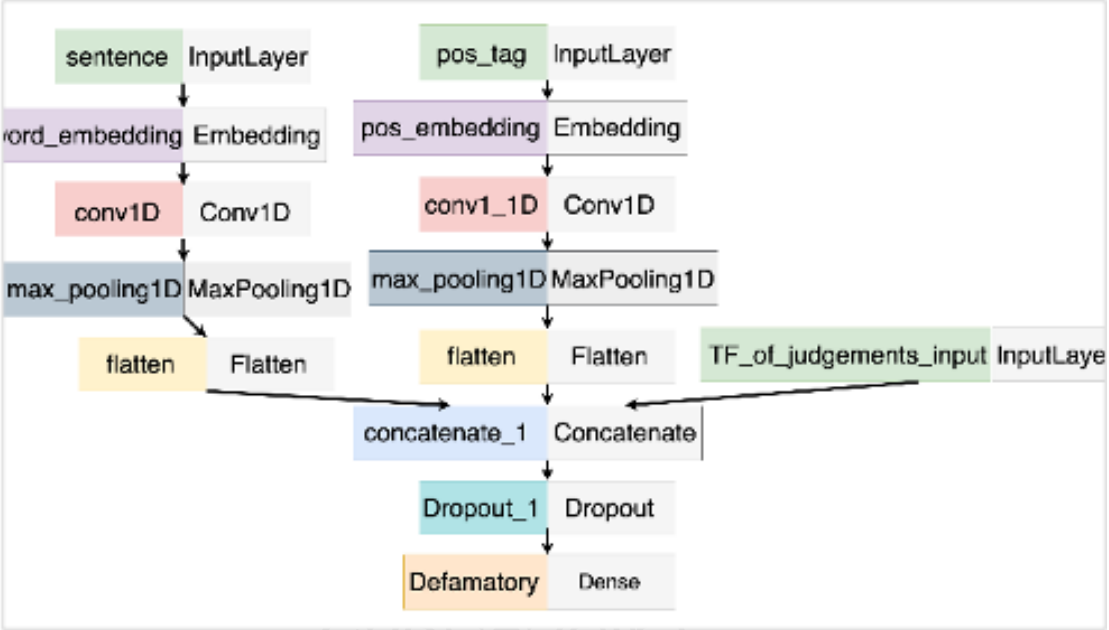
ในการสร้างแบบจำลอง Bi-LSTM จะใช้ Layer LSTM เป็นชั้นที่ใช้ในการประมวลผลลำดับของข้อมูลเชิงลำดับ โดยจะรับ input layers ที่ป้อนข้อมูลเข้าสู่โมเดล ซึ่งประกอบด้วย sentence_input ที่เป็น input สำหรับข้อความ และ features input อื่น ๆ ที่ผู้วิจัยเสนอเป็น input สำหรับข้อมูลเข้าเพิ่มเติม โดยในส่วนของประโยคที่เป็นข้อมูลเข้าจะใช้ layers Embedding เพื่อแปลงข้อความในรูปแบบเวกเตอร์ (embeddings) ซึ่งจะถูกนำเข้าไปในชั้น LSTM เพื่อให้เกิดการเรียนรู้ และการทำนายผลลัพธ์ หลังจากนั้นจะนำมาผ่านชั้น Layer เพื่อลดลำดับของคำที่ถูกฝังอยู่ในข้อความไปเป็นเวกเตอร์เดียวที่มีขนาด 128 มิติ โดยใช้เทคนิค LSTM จากนั้นผู้วิจัยจะรวม features ทั้งหมดเข้าไว้ในเวกเตอร์เดียวกันผ่าน layer concatenation จากนั้นผู้วิจัยมีการใส่ชั้น Dropout เพื่อลดการเกิด overfitting ในโมเดล และกำหนดชั้น Dense ที่เป็นชั้น Classification ขนาด 1 มิติที่เป็นจำนวนของผลลัพธ์ที่ต้องการทำนาย โดยใช้ activation function เป็น sigmoid เพื่อจำแนกว่าข้อความนั้น ๆ เข้าข่ายหมิ่นประมาท หรือไม่ต่อไป ดังแสดงในภาพที่ 25 การสร้างแบบจำลอง Bi-LSTM



ภาพที่ 25 โครงสร้างและพารามิเตอร์ของแบบจำลอง Bi-LSTM

3. CNN

ในการสร้างแบบจำลอง CNN เพื่อใช้สำหรับจำแนกข้อความที่เข้าข่ายหมิ่นประมาท จะใช้ Layer CNN1D เป็นเลเยอร์ Convolutional ที่ใช้สำหรับการทำงานกับข้อมูลที่มีลำดับ (sequence data) เช่น ข้อความ โดยในการสร้างแบบจำลอง CNN จะเริ่มจาก input layers ที่ป้อนข้อมูลเข้าสู่โมเดล ซึ่งประกอบด้วย sentence_input ที่เป็น input สำหรับข้อความ และ features input อื่น ๆ ที่ผู้วิจัยเสนอเป็น input สำหรับข้อมูลเข้าเพิ่มเติม โดยในส่วนของประโยคที่เป็นข้อมูลเข้าจะใช้ layers Embedding เพื่อแปลงข้อความในรูปแบบเวกเตอร์ (embeddings) ซึ่งจะถูกนำเข้าไปในชั้น Conv1D โดยได้กำหนด Filter ที่จะใช้ในการสร้าง feature maps ขนาด 128 และกำหนด kernel_size เท่ากับ 4 และกำหนด activation function ของ Layers ชั้นนี้ คือ ReLU เพื่อสร้างประสิทธิภาพ และความเร็วในการคำนวณ จากนั้นข้อมูลจะถูกนำไปผ่าน pooling layers เพื่อลดขนาดของข้อมูลที่ได้หลังจากการสกัดลักษณะ และส่งเข้าสู่ชั้น Flatten Layers เพื่อแปลงให้อยู่ในรูปแบบของเวกเตอร์ เพื่อเตรียมข้อมูลสำหรับการประมวลผลในชั้นต่อไป ต่อจากนี้จะคล้ายกับแบบจำลอง LSTM และ Bi-LSTM กล่าวคือผู้วิจัยจะรวม features ทั้งหมดเข้าไว้ในเวกเตอร์เดียวกันผ่าน layer concatenation และใส่ชั้น Dropout เพื่อลดการเกิด overfitting ในโมเดลชั้นสุดท้ายกำหนด Dense ที่เป็นชั้น Classification ขนาด 1 มิติ ซึ่งเป็นจำนวนของผลลัพธ์ที่ต้องการในการจำแนกข้อความที่เข้าข่ายหมิ่นประมาท โดยกระบวนการสร้างแบบจำลอง CNN แสดงดังภาพที่ 26



ภาพที่ 26 โครงสร้างและพารามิเตอร์ของแบบจำลอง CNN

ในแบบจำลองการเรียนรู้เชิงลึก LSTM Bi-LSTM และ CNN ผู้วิจัยใช้ Dropout Layer ขนาด 0.2 เพื่อหลีกเลี่ยงการเกิด overfitting นอกจากนั้นผู้วิจัยเลือกใช้ adam ในการปรับปรุงค่าน้ำหนักขณะฝึกสอนแบบจำลอง และเลือกใช้ loss function คือ binary cross entropy โดยกำหนด learning rate ที่ 1e-4 และกำหนดค่า batch size เท่ากับ 64

4. BERT Model

การใช้งาน BERT Model ในงานวิจัยนี้ได้นำเสนอการตั้งค่าพารามิเตอร์ที่ถูกปรับแต่ง โดยประกอบด้วยขั้นตอนต่อไปนี้

โหลดโมเดล BERT ในขั้นตอนแรกผู้วิจัยโหลดแบบจำลอง BERT โดยใช้คำสั่ง TFCamembertForSequenceClassification.from_pretrained() เพื่อโหลดโมเดล Camembert ที่ได้รับการฝึกแล้ว ซึ่งในงานวิจัยนี้ผู้วิจัยเลือกใช้ Pretrained BERT Model 2 แบบจำลองคือ แบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) และแบบจำลองวังจันทร์เบอร์ต้า(WangchanBERTa) ซึ่งเป็น Pretrained Model ที่ได้รับการฝึกด้วยข้อมูลภาษาไทย จากนั้นกำหนดชั้น Input สำหรับแบบจำลอง BERT เพื่อรับข้อมูลเข้าประกอบด้วย

1. input_ids ซึ่งเป็นรหัสที่แทนคำศัพท์แต่ละคำในข้อความ โมเดล BERT จะแปลงรหัสคำศัพท์เหล่านี้เป็นเส้นเวกเตอร์ที่มีความหมาย และลักษณะเฉพาะของแต่ละคำ

2. attention_mask เป็นเส้นเวกเตอร์ที่บอกให้โมเดล BERT ทราบว่าในข้อความที่ป้อนเข้ามา คำไหนเป็นคำจริง และคำไหนเป็นการเติม padding ในกรณีที่มีความยาวของข้อความไม่เท่ากัน ทำให้โมเดลสนใจ และให้ความสำคัญกับคำที่เป็นคำจริงเท่านั้น

ซึ่งทั้ง 2 ข้อมูลเข้าเป็นข้อมูลที่ใช้ในการสร้างเส้นเวกเตอร์แบบซ้อน หรือการแปลงข้อมูลเข้าเป็นเส้นเวกเตอร์ที่มีมิติน้อยลง จากนั้นจะรับ output จากโมเดล BERT ซึ่งเป็น output ชั้นที่สองของแบบจำลอง ซึ่งเป็นชั้นที่รับเส้นเวกเตอร์แบบซ้อนจากชั้นที่แล้ว และคำนวณผลลัพธ์ จากนั้นจึงเพิ่มชั้น Input สำหรับรับข้อมูลเพิ่มเติม ที่เป็น Features input ต่าง ๆ ที่ผู้วิจัยได้นำเสนอ เพื่อเตรียม concatenate เข้ากับ output จากโมเดล BERT และนำมาผ่านชั้น Dense Layer และ Dropout Layer ขนาด 0.2 เพื่อป้องกันการเกิด overfitting ในชั้นสุดท้ายผู้วิจัยกำหนด Output Layer ขนาด 1 มิติ สำหรับการจำแนกข้อความที่เข้าข่ายหมิ่นประมาท โดยใช้ Activation Function เป็น sigmoid สำหรับการจำแนกแบบ Binary Classification นอกจากนั้นผู้วิจัยเลือกใช้ adam ในการปรับค่า weight ขณะฝึกสอนแบบจำลอง และใช้ loss function คือ binary cross entropy เช่นเดียวกับแบบจำลองการเรียนรู้เชิงลึกแบบอื่น ๆ แต่ในส่วนของการกำหนด learning rate จะกำหนดเท่ากับ 2e-5 ซึ่งมาจากงานวิจัยต้นฉบับของ BERT ที่เสนอให้ใช้ค่า Learning Rate เท่ากับ 2e-5 ซึ่งเป็นค่าที่เหมาะสมสำหรับแบบจำลองขนาด 340 ล้านพารามิเตอร์ และกำหนดค่า bath size เท่ากับ 64

4.4 การวัดประสิทธิภาพและตัวชี้วัด

4.4.1 เมทริกซ์ความสับสน (Confusion Matrix)

เมทริกซ์ความสับสนเป็นตารางที่ใช้สำหรับแสดงประสิทธิภาพของโมเดลในการทำนายคลาสของข้อมูล ประกอบด้วยค่า 4 ค่า คือ True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN)

ตารางที่ 9 เมทริกซ์ความสับสน

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

โดยที่

True Positive (TP): จำนวนข้อมูลที่โมเดลทำนายถูกต้องว่าเป็น Positive

False Positive (FP): จำนวนข้อมูลที่โมเดลทำนายว่าเป็น Positive แต่จริง ๆ แล้วเป็น Negative
False Negative (FN): จำนวนข้อมูลที่โมเดลทำนายว่าเป็น Negative แต่จริง ๆ แล้วเป็น Positive
True Negative (TN): จำนวนข้อมูลที่โมเดลทำนายถูกต้องว่าเป็น Negative

4.4.2 ค่าความถูกต้อง (Accuracy)

ค่าความถูกต้อง คือหนึ่งในตัวชี้วัดที่มักถูกใช้ในการประเมินประสิทธิภาพของแบบจำลอง โดยหลักการคือการวัดประสิทธิภาพของโมเดลในการทำนายคลาสของข้อมูลที่มีอยู่ตามเมทริกซ์ความสับสน (Confusion Matrix) ซึ่งประกอบด้วยสี่ส่วนหลักคือ True Positive (TP), False Positive (FP), False Negative (FN), และ True Negative (TN) ค่าความถูกต้องถูกคำนวณโดยการหารจำนวนคำตอบที่โมเดลทำนายถูกต้อง (TP + TN) ด้วยจำนวนคำตอบทั้งหมด (TP + TN + FP + FN) ดังสมการที่ 4.4.2

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$ สมการที่ 4.4.2

4.4.3 ค่าแม่นยำ (Precision)

ค่าความแม่นยำ เป็นตัวชี้วัดที่ใช้ในการวัดประสิทธิภาพของโมเดลในการทำนายคลาสบวก หรือคลาสที่สนใจ โดยมุ่งเน้นไปที่ความแม่นยำในการทำนายว่าตัวอย่างใดมีความเป็นคลาสนั้นจริง ๆ ค่าความแม่นยำคำนวณจากสัดส่วนของตัวอย่างที่โมเดลทำนายถูกต้องว่าเป็นคลาสบวก (True Positive) เทียบกับจำนวนทั้งหมดของตัวอย่างที่โมเดลทำนายว่าเป็นคลาสบวก (True Positive และ False Positive) โดยมีสมการที่ 4.4.3

$$Precision = \frac{TP}{TP+FP}$$ สมการที่ 4.4.3

4.4.4 ค่าความครบถ้วน (Recall)

ค่าความครบถ้วน เป็นตัวชี้วัดที่ใช้ในการประเมินประสิทธิภาพของแบบจำลอง โดยการให้ความสำคัญกับการครอบคลุมของข้อมูลที่เป็นคลาสบวก (Positive class) ค่าความครบถ้วนวัดว่าโมเดลสามารถตรวจจับตัวอย่างของคลาสบวกทั้งหมดในชุดข้อมูลได้มากเพียงใด ซึ่งค่าความครบถ้วนถูกคำนวณโดยการหารจำนวนคำตอบที่โมเดลทำนายถูกต้องว่าเป็นคลาสบวก (True Positive, TP) ด้วยจำนวนทั้งหมดของตัวอย่างที่เป็นคลาสบวกจริง (True Positive และ False Negative, TP + FN) ดังสมการนี้ค่าความครบถ้วนถูกคำนวณโดยการหารจำนวนคำตอบที่โมเดลทำนายถูกต้องว่าเป็นคลาสบวก (True Positive, TP) ด้วยจำนวนทั้งหมดของตัวอย่างที่เป็นคลาสบวกจริง (True Positive และ False Negative, TP + FN) ดังสมการที่ 4.4.4

$$Recall = \frac{TP}{TP+FN}$$ สมการที่ 4.4.4

4.4.5 เอฟวัน สกอร์ (F1-Score)

F1-Score เป็นตัวชี้วัดความสมดุลระหว่างความแม่นยำ (Precision) และความครบถ้วน (Recall) ซึ่งช่วยในการประเมินประสิทธิภาพของแบบจำลองการ โดยใช้เพื่อเน้นที่ความสมดุลระหว่างการตรวจจับตัวอย่างของคลาสบวกทั้งหมด (ความครบถ้วน) และการลดจำนวนของผลลัพธ์ที่เป็นเท็จในคลาสบวก (ความแม่นยำ) F1-Score ถูกคำนวณจากความสัมพันธ์ระหว่าง Precision และ Recall ดังสมการที่ 4.4.5

$$F1 = 2\ x\ \frac{Precision\ x\ Recall}{Precision+Recall}$$ สมการที่ 4.4.5

มหาวิทยาลัยศิลปากร

บทที่ 5
ผลการดำเนินงานวิจัย

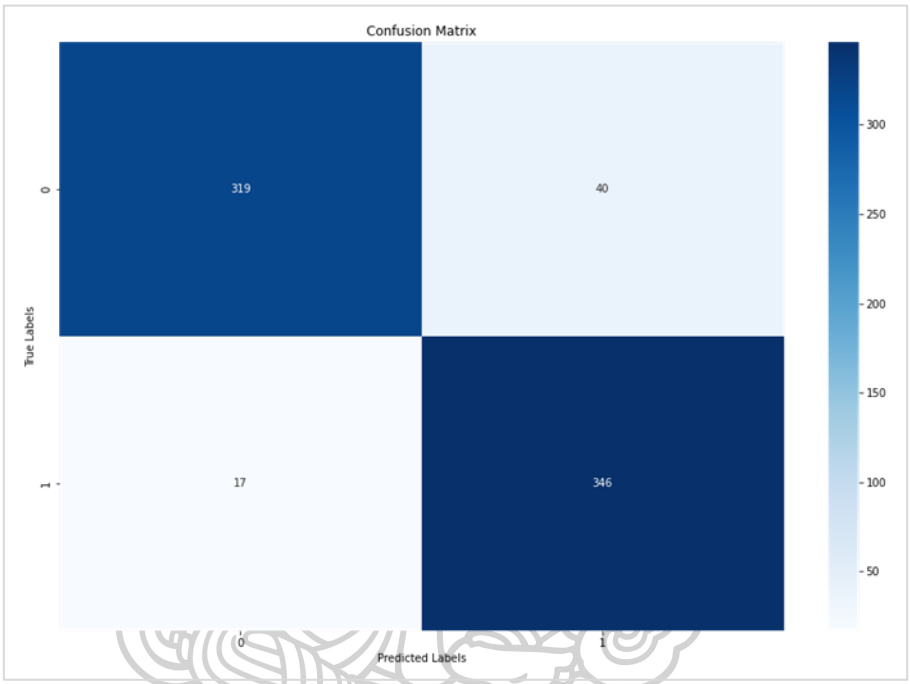
ในบทนี้จะกล่าวถึงผลการทดลองเพื่อประเมินประสิทธิภาพของวิธีการ และเทคนิคที่ใช้ในบทที่ 4 งานวิจัยนี้ได้นำแบบจำลองทั้งหมด 5 ประเภท ได้แก่ แบบจำลองหน่วยความจำระยะสั้นแบบยาว (LSTM) แบบจำลองหน่วยความจำระยะสั้นแบบยาว แบบสองทิศทาง (Bi-LSTM) แบบจำลองการเรียนรู้แบบคอนโวลูชัน (CNN) แบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ร่วมกับการนำเอาคุณลักษณะที่ได้นำเสนอคือ การฝังคำ (Word Embedding), การนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา (Term frequency of Dictionary judgement), POS Tag และ Name Entity Tag เป็นข้อมูลเข้าสำหรับการทดลองเพื่อเปรียบเทียบประสิทธิภาพในการจำแนกข้อความที่เข้าข่ายหมิ่นประมาทบนสื่อสังคมออนไลน์ โดยใช้ชุดข้อมูลที่ได้เก็บรวบรวมมา โดยผู้วิจัยจะใช้การตั้งค่าพารามิเตอร์ของแบบจำลองตามที่ได้อธิบายไว้ในบทที่ 4 ซึ่งในแบบจำลอง LSTM, Bi-LSTM และ CNN ผู้วิจัยใช้รอบการเรียนรู้จำนวน 300 รอบ และในแบบจำลอง BERT Transformer ทั้งสองแบบจำลองคือ แบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) เนื่องจากใช้ระยะเวลาการเรียนรู้ค่อนข้างมากผู้วิจัยจึงใช้รอบการเรียนรู้จำนวน 10 รอบในการทดลอง การศึกษานี้ประเมินประสิทธิภาพแบบจำลองจาก Accuracy, Precision, Recall, F1-score ซึ่งมาจากการคำนวณโดยเฉลี่ยของการทดลองเพราะใช้วิธี 3-fold cross validation สำหรับการแบ่งข้อมูลเพื่อ Train และ Validate โดยใช้ Weight average เป็นหลักในการประเมินประสิทธิภาพเฉลี่ยโดยรวม และเนื่องจากในการวิจัยนี้ทดลองโดยวิธี 3-fold cross validation ผู้วิจัยจึงแสดง Confusion Matrix จากการทดลองในรอบที่ดีที่สุดของการทดลองนั้น ๆ

5.1 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า

ในการทดลองนี้แบบจำลองทั้ง 5 ประเภทจะใช้คุณลักษณะการฝังคำที่แตกต่างกัน ได้แก่ LSTM, Bi-LSTM, CNN จะใช้คุณลักษณะการฝังคำ Word2vec คือ Thai2fit ในการฝังคำ ส่วนแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT)จะใช้คุณลักษณะการฝังคำของสถาปัตยกรรม WangchanBERTa และ PhayaThaiBERT เองตามลำดับ

ในการจำแนกข้อความที่เข้าข่ายหมิ่นประมาทบนสื่อสังคมออนไลน์ที่ใช้คุณลักษณะการฝังคำเป็นข้อมูลเข้า ผลลัพธ์ของการทดลองแสดงในตารางที่ 10 พบว่าแบบจำลองพญาไทยเบิร์ต

(PhayaThaiBERT) ซึ่งใช้เพียงคุณลักษณะการฝังคำของตัวเอง มีประสิทธิภาพดีที่สุดโดยมีค่า Accuracy เท่ากับ 88.64% ค่า Precision เท่ากับ 89.31% ค่า Recall เท่ากับ 88.64% และมีค่า F1-Score เท่ากับ 88.59% โดยแสดง Confusion Matrix ดังภาพที่ 27 รองลงมาเป็นแบบจำลอง CNN โดยมีค่า Accuracy เท่ากับ 83.61% ค่า Precision เท่ากับ 83.65% ค่า Recall เท่ากับ 83.61% และมีค่า F1-Score เท่ากับ 83.60%



ภาพที่ 27 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของการทดลองหัวข้อที่ 5.1

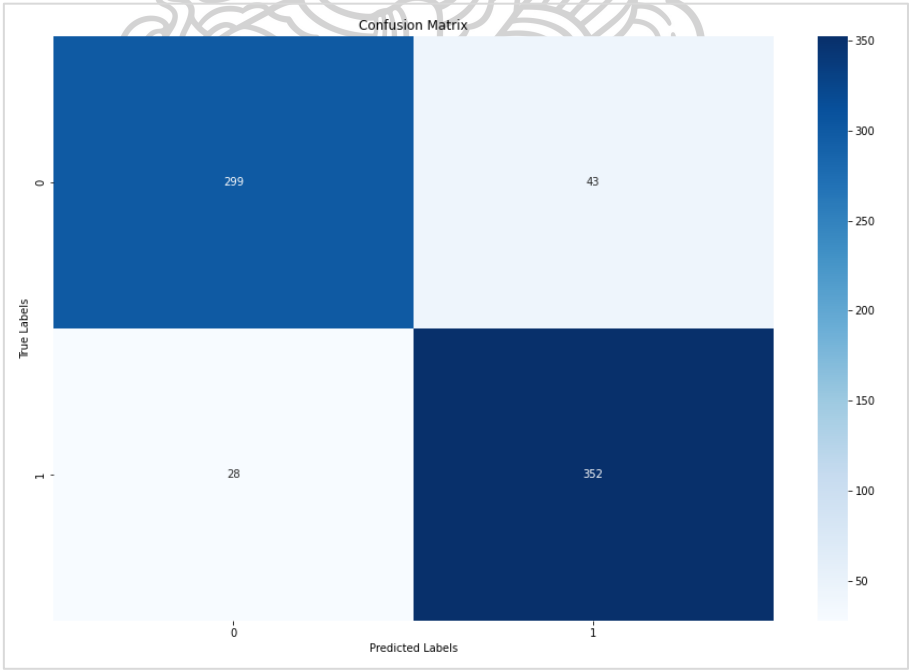
ตารางที่ 10 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้เฉพาะการฝังคำเป็นข้อมูลเข้า

Methods	Accuracy	Precision	Recall	F1-score
LSTM	80.47%	80.35%	80.47%	81.07%
Bi-LSTM	81.81%	82.36%	81.81%	81.73%
CNN	83.61%	83.65%	83.61%	83.60%
WangchanBERTa	83.24%	83.42%	83.24%	83.24%
PhayaThaiBERT	88.64%	89.31%	88.64%	88.59%

5.2 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมเป็นข้อมูลเข้า

การทดลองนี้แบบจำลองทั้ง 5 ประเภทจะใช้คุณลักษณะการฝังคำที่แตกต่างกัน ได้แก่ LSTM, Bi-LSTM, CNN จะใช้คุณลักษณะการฝังคำ Word2vec คือ Thai2fit ในการฝังคำ ส่วนแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT)จะใช้คุณลักษณะการฝังคำของสถาปัตยกรรม WangchanBERTa และ PhayaThaiBERT เองตามลำดับ ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษาเป็นข้อมูลเข้า

ผลลัพธ์ของการทดลองแสดงในตารางที่ 11 พบว่าแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ซึ่งใช้คุณลักษณะที่นำเสนอในการทดลองนี้เป็นข้อมูลเข้า มีประสิทธิภาพดีที่สุด โดยมีค่า Accuracy เท่ากับ 90.49% ค่า Precision เท่ากับ 90.62% ค่า Recall เท่ากับ 90.49% และมีค่า F1-Score เท่ากับ 90.48% โดยแสดง Confusion Matrix ดังภาพที่ 28 รองลงมาเป็นแบบจำลอง CNN โดยมีค่า Accuracy เท่ากับ 83.98% ค่า Precision เท่ากับ 84.00% ค่า Recall เท่ากับ 83.98% และมีค่า F1-Score เท่ากับ 83.98%



ภาพที่ 28 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของการทดลองหัวข้อที่ 5.2

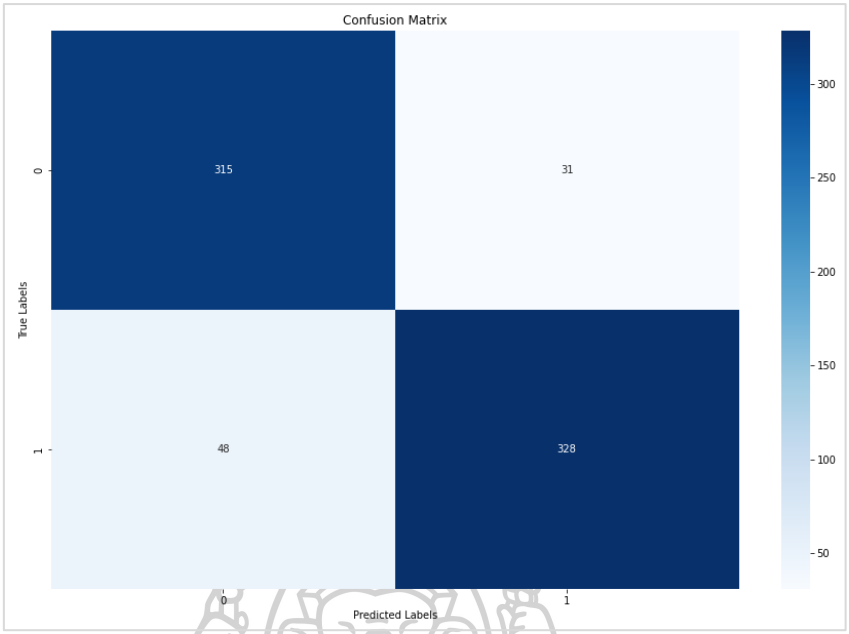
ตารางที่ 11 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา

Methods	Accuracy	Precision	Recall	F1-score
LSTM	82.13%	82.46%	82.13%	82.08%
Bi-LSTM	80.47%	81.07%	80.47%	80.35%
CNN	83.98%	84.00%	83.98%	83.98%
WangchanBERTa	68.24%	68.34%	68.24%	65.77%
PhayaThaiBERT	90.49%	90.62%	90.49%	90.48%

5.3 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ POS tag One-hot เป็นข้อมูลเข้า

การทดลองนี้แบบจำลองทั้ง 5 ประเภทจะใช้คุณลักษณะการฝังคำที่แตกต่างกัน ได้แก่ LSTM, Bi-LSTM, CNN จะใช้คุณลักษณะการฝังคำ Word2vec คือ Thai2fit ในการฝังคำ ส่วนแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) จะใช้คุณลักษณะการฝังคำของสถาปัตยกรรม WangchanBERTa และ PhayaThaiBERT เองตามลำดับ ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ POS tag One-hot เป็นข้อมูลเข้า

ผลลัพธ์ของการทดลองแสดงในตารางที่ 12 พบว่าแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ซึ่งใช้คุณลักษณะที่นำเสนอในการทดลองนี้เป็นข้อมูลเข้า มีประสิทธิภาพดีที่สุด โดยมีค่า Accuracy เท่ากับ 87.26% ค่า Precision เท่ากับ 87.76% ค่า Recall เท่ากับ 87.26% และมีค่า F1-Score เท่ากับ 87.24% โดยแสดง Confusion Matrix ดังภาพที่ 29 รองลงมาเป็นแบบจำลอง CNN โดยมีค่า Accuracy เท่ากับ 83.84% ค่า Precision เท่ากับ 83.93% ค่า Recall เท่ากับ 83.84% และมีค่า F1-Score เท่ากับ 83.82%



ภาพที่ 29 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของการทดลองหัวข้อที่ 5.3

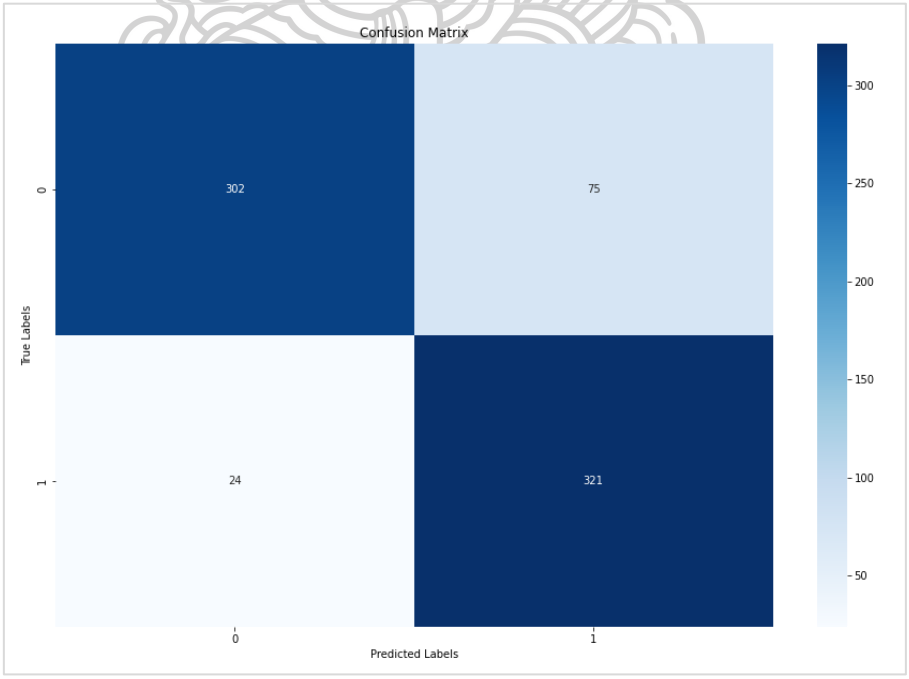
ตารางที่ 12 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ POS tag One-hot เป็นข้อมูลเข้า

Methods	Accuracy	Precision	Recall	F1-score
LSTM	82.92%	82.92%	82.99%	82.92%
Bi-LSTM	80.84%	81.34%	80.84%	80.77%
CNN	83.84%	83.93%	83.84%	83.82%
WangchanBERTa	79.18%	79.63%	79.18%	79.15%
PhayaThaiBERT	87.26%	87.76%	87.26%	87.24%

5.4 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรม และ POS tag Embedding เป็นข้อมูลเข้า

การทดลองนี้แบบจำลองทั้ง 5 ประเภทจะใช้คุณลักษณะการฝังคำที่แตกต่างกัน ได้แก่ LSTM, Bi-LSTM, CNN จะใช้คุณลักษณะการฝังคำ Word2vec คือ Thai2fit ในการฝังคำ ส่วนแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) จะใช้คุณลักษณะการฝังคำของสถาปัตยกรรม WangchanBERTa และ PhayaThaiBERT เองตามลำดับ ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ POS tag Embedding เป็นข้อมูลเข้า

ผลลัพธ์ของการทดลองแสดงในตารางที่ 13 พบว่าแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ซึ่งใช้คุณลักษณะที่นำเสนอในการทดลองนี้เป็นข้อมูล มีประสิทธิภาพดีที่สุดโดยมีค่า Accuracy เท่ากับ 87.40% ค่า Precision เท่ากับ 88.31% ค่า Recall เท่ากับ 87.40% และมีค่า F1-Score เท่ากับ 87.34% โดยแสดง Confusion Matrix ดังภาพที่ 30 รองลงมาเป็นแบบจำลอง CNN โดยมีค่า Accuracy เท่ากับ 84.03% ค่า Precision เท่ากับ 84.10% ค่า Recall เท่ากับ 84.03% และมีค่า F1-Score เท่ากับ 84.02%



ภาพที่ 30 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของหัวข้อการทดลองที่ 5.4

ตารางที่ 13 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ POS tag Embedding เป็นข้อมูลเข้า

Methods	Accuracy	Precision	Recall	F1-score
LSTM	76.82%	77.47%	76.82%	76.55%
Bi-LSTM	83.56%	83.87%	83.56%	83.52%
CNN	84.03%	84.10%	84.03%	84.02%
WangchanBERTa	75.02%	75.24%	75.02%	74.98%
PhayaThaiBERT	87.40%	88.31%	87.40%	87.34%

5.5 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ Name Entity One-hot เป็นข้อมูลเข้า

การทดลองนี้แบบจำลองทั้ง 5 ประเภทจะใช้คุณลักษณะการฝังคำที่แตกต่างกัน ได้แก่ LSTM, Bi-LSTM, CNN จะใช้คุณลักษณะการฝังคำ Word2vec คือ Thai2fit ในการฝังคำ ส่วนแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) จะใช้คุณลักษณะการฝังคำของสถาปัตยกรรม WangchanBERTa และ PhayaThaiBERT เองตามลำดับ ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ Name Entity One-hot เป็นข้อมูลเข้า

ผลลัพธ์ของการทดลองแสดงในตารางที่ 14 พบว่าแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ซึ่งใช้คุณลักษณะที่นำเสนอในการทดลองนี้เป็นข้อมูลเข้า มีประสิทธิภาพดีที่สุด โดยมีค่า Accuracy เท่ากับ 87.21% ค่า Precision เท่ากับ 88.20% ค่า Recall เท่ากับ 87.21% และมีค่า F1-Score เท่ากับ 87.14% โดยแสดง Confusion Matrix ดังภาพที่ 31 รองลงมาเป็นแบบจำลอง Bi-LSTM โดยมีค่า Accuracy เท่ากับ 83.15% ค่า Precision เท่ากับ 83.18% ค่า Recall เท่ากับ 83.15% และมีค่า F1-Score เท่ากับ 83.15%

ภาพที่ 31 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของหัวข้อการทดลองที่ 5.5

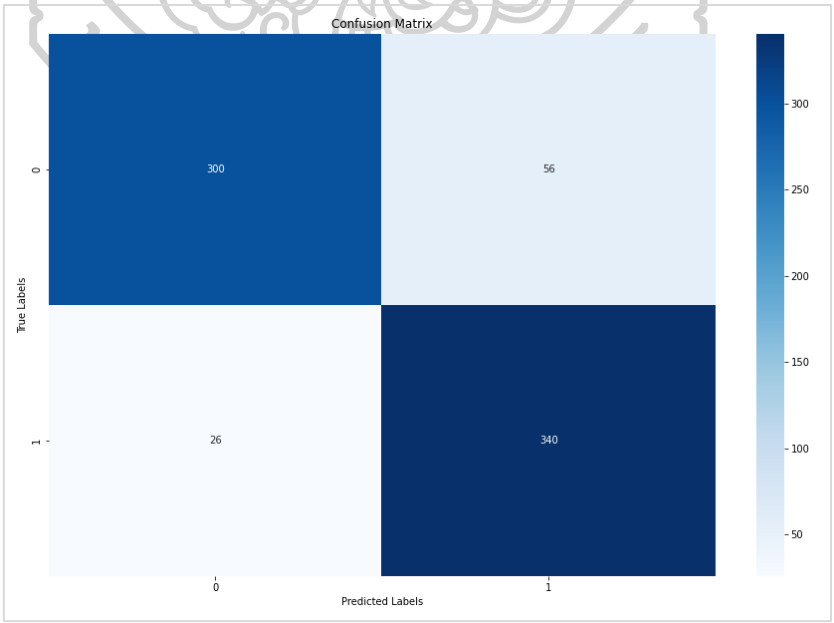
ตารางที่ 14 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา และ Name Entity One-hot เป็นข้อมูลเข้า

Methods	Accuracy	Precision	Recall	F1-score
LSTM	83.06%	83.28%	83.06%	83.03%
Bi-LSTM	83.15%	83.18%	83.15%	83.15%
CNN	82.96%	83.00%	82.96%	82.96%
WangchanBERTa	61.96%	64.64%	61.96%	54.68%
PhayaThaiBERT	87.21%	88.20%	87.21%	87.14%

5.6 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag One-hot และ Name Entity One-hot เป็นข้อมูลเข้า

การทดลองนี้แบบจำลองทั้ง 5 ประเภทจะใช้คุณลักษณะการฝังคำที่แตกต่างกัน ได้แก่ LSTM, Bi-LSTM, CNN จะใช้คุณลักษณะการฝังคำ Word2vec คือ Thai2fit ในการฝังคำ ส่วนแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) จะใช้คุณลักษณะการฝังคำของสถาปัตยกรรม WangchanBERTa และ PhayaThaiBERT เองตามลำดับ ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา Pos tag One-hot และ Name Entity One-hot เป็นข้อมูลเข้า

ผลลัพธ์ของการทดลองแสดงในตารางที่ 15 พบว่าแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ซึ่งใช้คุณลักษณะที่นำเสนอในการทดลองนี้เป็นข้อมูลเข้า มีประสิทธิภาพดีที่สุด โดยมีค่า Accuracy เท่ากับ 87.86% ค่า Precision เท่ากับ 88.20% ค่า Recall เท่ากับ 87.86% และมีค่า F1-Score เท่ากับ 87.83% โดยแสดง Confusion Matrix ดังภาพที่ 32 รองลงมาเป็นแบบจำลอง LSTM โดยมีค่า Accuracy 83.75% ค่า Precision เท่ากับ 84.19% ค่า Recall เท่ากับ 83.75% และมีค่า F1-Score เท่ากับ 83.70%



ภาพที่ 32 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของหัวข้อการทดลองที่ 5.6

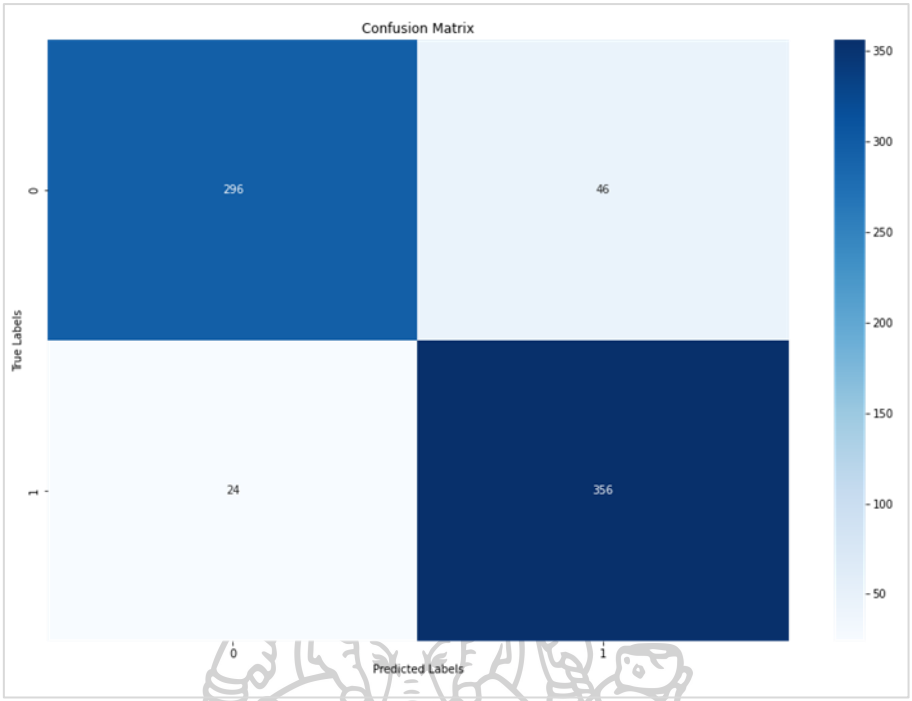
ตารางที่ 15 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag One-hot และ Name Entity One-hot เป็นข้อมูลเข้า

Methods	Accuracy	Precision	Recall	F1-score
LSTM	83.75%	84.19%	83.75%	83.70%
Bi-LSTM	81.95%	82.19%	81.95%	81.91%
CNN	82.27%	82.35%	82.27%	82.26%
WangchanBERTa	50.92%	51.24%	50.92%	49.02%
PhayaThaiBERT	87.86%	88.20%	87.86%	87.83%

5.7 การทดลองแบบจำลองโดยใช้เฉพาะคุณลักษณะการฝังคำเป็นข้อมูลเข้า ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag Embedding และ Name Entity One-hot เป็นข้อมูลเข้า

การทดลองนี้แบบจำลองทั้ง 5 ประเภทจะใช้คุณลักษณะการฝังคำที่แตกต่างกัน ได้แก่ LSTM, Bi-LSTM, CNN จะใช้คุณลักษณะการฝังคำ Word2vec คือ Thai2fit ในการฝังคำ ส่วนแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) จะใช้คุณลักษณะการฝังคำของสถาปัตยกรรม WangchanBERTa และ PhayaThaiBERT เองตามลำดับ ร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag Embedding และ Name Entity One-hot เป็นข้อมูลเข้า

ผลลัพธ์ของการทดลองแสดงในตารางที่ 16 พบว่าแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ซึ่งใช้คุณลักษณะที่นำเสนอในการทดลองนี้เป็นข้อมูลเข้า มีประสิทธิภาพดีที่สุด โดยมีค่า Accuracy เท่ากับ 88.64% ค่า Precision เท่ากับ 88.78% ค่า Recall เท่ากับ 88.64% และมีค่า F1-Score เท่ากับ 88.63% โดยแสดง Confusion Matrix ดังภาพที่ 33 รองลงมาเป็นแบบจำลอง CNN โดยมีค่า Accuracy 84.21% ค่า Precision เท่ากับ 88.78% ค่า Recall เท่ากับ 88.64% และมีค่า F1-Score เท่ากับ 88.63%



ภาพที่ 33 กราฟ Confustion Matrix ของแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ของหัวข้อการทดลองที่ 5.7

ตารางที่ 16 แสดงการเปรียบเทียบประสิทธิภาพ ของแบบจำลอง 5 ประเภทโดยใช้การฝังคำเป็นข้อมูลเข้าร่วมกับการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag Embedding และ Name Entity One-hot เป็นข้อมูลเข้า

Methods	Accuracy	Precision	Recall	F1-score
LSTM	81.53%	81.73%	81.53%	81.50%
Bi-LSTM	81.12%	81.23%	81.12%	81.10%
CNN	84.21%	84.23%	84.21%	84.21%
WangchanBERTa	57.39%	52.29%	57.39%	48.97%
PhayaThaiBERT	88.64%	88.78%	88.64%	88.63%

บทที่ 6
สรุปผลการทดลอง และอภิปรายผลการทดลอง

ในบทนี้จะกล่าวถึงการสรุปผลการทดลองการการจำแนกข้อความที่เข้าข่ายหมิ่นประมาทบนสื่อสังคมออนไลน์โดยใช้ชุดข้อมูลที่เก็บรวบรวมมาจากเฟสบุ๊คด้วย ซึ่งทดลองด้วยแบบจำลอง และการสกัดคุณลักษณะต่าง ๆ ในบทที่ 4 และนำมาอภิปรายผลการทดลอง ทั้งวิเคราะห์ปัญหาอุปสรรคในการดำเนินงานวิจัย เสนอแนวทางการวิจัยในอนาคตต่อไป

6.1 สรุปผลการทดลองและอภิปรายผลการทดลอง

วิทยานิพนธ์นี้ผู้วิจัยได้นำเสนอการสร้างแบบจำลองเพื่อจำแนกข้อความที่เข้าข่ายหมิ่นประมาทบนสื่อสังคมออนไลน์เฟสบุ๊ค โดยได้เสนอแบบจำลองทั้งหมด 5 ประเภท ได้แก่ แบบจำลองหน่วยความจำระยะสั้นแบบยาว (LSTM) แบบจำลองหน่วยความจำระยะสั้นแบบยาว แบบสองทิศทาง (Bi-LSTM) แบบจำลองการเรียนรู้แบบคอนโวลูชัน (CNN) แบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ร่วมกับการสกัดคุณลักษณะของคำ ได้แก่ การฝังคำ (Word Embedding), การนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา (Term frequency of Dictionary judgement), POS Tag และ Name Entity Tag

ผลการทดลองแสดงให้เห็นว่าแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) มีประสิทธิภาพสูงที่สุดในทุกการทดลองเมื่อเทียบกับแบบจำลองอื่น ๆ และเมื่อนำการสกัดคุณลักษณะการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา แบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) สามารถทำประสิทธิภาพสูงสุดโดยมีค่า Accuracy เท่ากับ 90.49% ค่า Precision เท่ากับ 90.62% ค่า Recall เท่ากับ 90.49% และมีค่า F1-Score เท่ากับ 90.48% ซึ่งในภาพ Confusion Matrix ที่แสดงในภาพที่ 28 ของการทดลองนี้แสดงให้เห็นว่า แบบจำลองสามารถจำแนกข้อความที่เข้าข่ายหมิ่นประมาทได้อย่างถูกต้องทั้งในการจำแนกข้อความที่เข้าข่าย และไม่เข้าข่าย นอกจากนี้ยังมีประสิทธิภาพสูงกว่าแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) ซึ่งเป็นแบบจำลองประเภท Transformer เหมือนกันในทุกการทดลอง โดยมีข้อสังเกตุว่าแบบจำลองวังจันทร์เบอร์ต้า WangchanBERTa ทำงานได้ไม่ดีในชุดข้อมูลในงานวิจัยนี้ และสอดคล้องกับผลการวิจัยของ Sriwirote et al., [8] ที่ระบุผลลัพธ์ว่า PhayaThaiBERT มีประสิทธิภาพที่ดีกว่าแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) ในหลายงานดังแสดงในรายงานผลการทดลองดังภาพที่ 34

Dataset	mBERT	XLM-R	WangchanBERTa	PhayaThaiBERT
1. wisesight_sentiment	70.57 / 55.62	71.77 / 58.29	74.35 / 65.23	**76.15 / 66.80**
2. wongnai_reviews	58.08 / 38.67	62.73 / 51.18	63.86 / **53.26**	**64.02** / 52.74
3. yelp_review_full	63.52 / 63.23	**65.19 / 64.80**	54.97 / 54.40	61.69 / 61.26
4. generated_reviews_enth	61.79 / 56.04	**65.06 / 60.28**	64.75 / 59.91	64.85 / 59.44
5. prachathai67k	63.90 / 52.95	66.63 / 58.01	67.51 / 59.05	**69.11 / 61.10**
6. thainer (ner)	79.58 / 69.87	84.95 / 72.20	84.64 / 68.19	**86.42 / 74.77**
7. lst20 (pos)	95.80 / 83.95	95.99 / 85.09	96.74 / **86.59**	**96.79** / 86.26
8. lst20 (ner)	76.48 / 70.27	**78.37** / 72.82	77.99 / **72.86**	78.11 / 72.69
9. thai_nner (layer 1)	61.19 / 23.45	63.88 / 23.28	59.31 / 22.85	**64.26 / 25.70**

ภาพที่ 34 รายงาผลการทดลองเปรียบเทียบประสิทธิภาพจากงานวิจัย PhayaThaiBERT[8]

ในทางกลับกันแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) และแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) กลับมีผลลัพธ์ที่ต่ำลงเมื่อมีการเพิ่มการสกัดคุณลักษณะของคำอื่น ๆ เพิ่มเข้าไปในการทดลอง โดยเฉพาะแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) ทำประสิทธิภาพได้ต่ำกว่าปกติ เช่นในการทดลองที่ 5.7 ที่ใช้การนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา, Pos tag Embedding และ Name Entity One-hot เป็นข้อมูลเข้า โดยได้ Accuracy 57.39% ค่า Precision เท่ากับ 52.29% ค่า Recall เท่ากับ 57.39% และ F1-score เท่ากับ 48.97% ซึ่งอาจมาจากการปรับพารามิเตอร์ของแบบจำลอง WangchanBERTa ที่ยังไม่เหมาะสมกับข้อมูล และงานวิจัยนี้ ซึ่งอาจจำเป็นต้องทำการปรับแต่งเพิ่มเติมเพื่อให้ได้ผลลัพธ์ที่ดีกว่า

จึงสรุปได้ว่าแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) มีประสิทธิภาพสูงในการจำแนกข้อความที่เข้าข่ายหมิ่นประมาทบนสื่อสังคมออนไลน์ โดยใช้เพียงการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษาเป็นข้อมูลเข้าเท่านั้น การเพิ่มการสกัดคุณลักษณะอื่น ๆ เข้าไปอาจทำให้ประสิทธิภาพลดลง จากการพิจารณาผลลัพธ์การจำแนกข้อความของการทดลองที่ใช้แบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ด้วยการนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษาเป็นข้อมูลเข้า (ภาพที่ 28) พบว่าแบบจำลองสามารถจำแนกข้อความที่เข้าข่ายหมิ่นประมาทได้ดีกว่าการจำแนกข้อความที่ไม่เข้าข่ายหมิ่นประมาท สาเหตุหนึ่งอาจมาจากชุดข้อมูลที่ใช้ในงานวิจัยนี้ได้รับการคัดเลือกที่เกี่ยวข้องกับการหมิ่นประมาทโดยตรง ทำให้แบบจำลองอาจมีแนวโน้มที่จะเข้าใจว่าข้อความที่ฝึกสอน และทดสอบเป็นข้อความที่เข้าข่ายหมิ่นประมาท แม้ว่าในความเป็นจริงอาจจะไม่หมิ่นประมาทก็ตาม

6.2 เปรียบเทียบประสิทธิภาพของแบบจำลองและเทคนิคที่นำเสนอกับงานวิจัยอื่น

ผู้วิจัยได้เปรียบเทียบประสิทธิภาพของแบบจำลองกับงานวิจัยก่อนหน้า Watjanapron et al., [39] ซึ่งใช้ชุดข้อมูลที่ได้เก็บรวบรวมก่อนหน้านี้จำนวน 1,150 ข้อความ โดยพบว่าการใช้

แบบจำลองการการเรียนรู้แบบคอนโวลูชัน (CNN) เมื่อใช้การสกัดคุณลักษณะคือ การฝังคำ (Word Embedding), การนับความถี่ของคำที่พบในพจนานุกรมคำพิพากษา (Term frequency of Dictionary judgement) และ POS Tag Embedding เป็นข้อมูลเข้า ได้ประสิทธิภาพดีที่สุดที่ Accuracy 85.91% ค่า Precision เท่ากับ 86.17% ค่า Recall เท่ากับ 85.91% และมีค่า F1-Score เท่ากับ 85.59% ซึ่งใกล้เคียงกับการทดลองเดียวกันในงานวิจัยนี้ ที่หากใช้ตัวแปรเดียวกัน จะได้ผลลัพธ์เท่ากับ Accuracy 84.03% ค่า Precision เท่ากับ 84.10% ค่า Recall เท่ากับ 84.03% และมีค่า F1-Score เท่ากับ 84.02% ซึ่งลดลงเล็กน้อยเนื่องจากมีการเพิ่มจำนวนชุดข้อมูลเกือบสองเท่า โดยมีข้อความทั้งหมด 2,166 ข้อความ

นอกจากนี้เมื่อเปรียบเทียบกับงานวิจัยของ Arreerard et al., [28] ซึ่งเป็นงานวิจัยที่ศึกษาการจำแนกข้อความที่เข้าข่ายหมิ่นประมาทภาษาไทยเช่นเดียวกัน โดยผลการทดลองที่ดีที่สุด โดยแบบจำลอง SVM ร่วมกับการสกัดคุณลักษณะที่นำเสนอในงานวิจัยคือ Word n-grams, Char n-grams, Dependency structure, Sentiment polarity และกำหนดข้อกำหนดเฉพาะจากพจนานุกรมคำพิพากษา โดยได้ค่า Accuracy เท่ากับ 74% ค่า Precision เท่ากับ 70% ค่า Recall เท่ากับ 59% และมีค่า F1-Score เท่ากับ 64% ซึ่งเมื่อเปรียบเทียบกับผลลัพธ์ที่ได้จากงานวิจัยนี้ที่ได้ประสิทธิภาพดีที่สุดจากแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) ร่วมกับการนับความถี่จากคำที่พบในพจนานุกรมคำพิพากษาโดยมีค่า Accuracy เท่ากับ 90.49% ค่า Precision เท่ากับ 90.62% ค่า Recall เท่ากับ 90.49% และมีค่า F1-Score เท่ากับ 90.48% แสดงให้เห็นว่างานวิจัยนี้มีประสิทธิภาพสูงกว่างานวิจัยก่อนหน้า

6.3 ปัญหาและอุปสรรค

ในการเก็บรวบรวมข้อมูลเพื่อใช้ในงานวิจัยนี้ ผู้วิจัยได้เก็บรวบรวมจากสื่อสังคมออนไลน์คือเฟสบุ๊คที่เป็นกลุ่มสาธารณะ โดยข้อความการแสดงความคิดเห็นดังกล่าวจำเป็นต้องอาศัยผู้เชี่ยวชาญทางด้านกฎหมายร่วมลงความเห็นว่าอาจเข้าข่ายหมิ่นประมาทตามประมวลกฎหมายที่เกี่ยวข้องหรือไม่ ทำให้ใช้ระยะเวลาในการตีความค่อนข้างนานจากผู้เชี่ยวชาญหลายท่านเนื่องจากต้องศึกษาจากฎีกาที่ศาลพิพากษาไว้ในอดีตควบคู่ในการพิจารณา จึงทำให้ล่าช้าในกระบวนการสร้างชุดข้อมูลสำหรับฝึกสอนแบบจำลอง อีกทั้งในการปรึกษาร่วมกับผู้เชี่ยวชาญทำให้พบว่าในปัจจุบันมีประโยควลีหรือคำสมัยใหม่ ๆ ที่เกิดใหม่จำนวนมากทำให้การพิจารณาตีความตัวกฎหมายทำได้ยากขึ้นอาจส่งผลต่อความแม่นยำในการจำแนกข้อความที่เข้าข่ายหมิ่นประมาทได้

6.4 แนวทางการวิจัยในอนาคต

1. ในอนาคตหากมีการเก็บรวบรวมข้อความจากสื่อสังคมออนไลน์เพิ่มเติม เพื่อสร้างชุดข้อมูลที่มีขนาดใหญ่ขึ้น เพื่อความหลากหลายของรูปแบบประโยคข้อความที่อาจเข้าข่ายหมิ่นประมาทเพิ่มมากขึ้น อีกทั้งผู้วิจัยมีความตั้งใจจะเก็บรวบรวมคำศัพท์ในฎีกาที่ตัดสินในอนาคตที่ยังอยู่ในชั้นพิจารณาคดีอีกมาก เพื่อเพิ่มเติมคำศัพท์ที่ไม่รู้จัก คำศัพท์สมัยใหม่ ในการจัดทำพจนานุกรมคำพิพากษา โดยคาดหวังว่าจะมีส่วนช่วยให้แบบจำลองเข้าใจรูปแบบประโยค และคำศัพท์ที่จะเข้าองค์ประกอบความผิดฐานหมิ่นประมาทตามประมวลกฎหมายอาญาของไทยต่อไป

2. ในขั้นตอนการตัดคำในภาษาไทย ผู้วิจัยได้เลือกใช้ PyThaiNLP ในการตัดคำเพื่อพัฒนาแบบจำลอง โดยเลือกใช้ newmm ร่วมกับการแบ่งคำในแบบจำลองวังจันทร์เบอร์ต้า (WangchanBERTa) และแบบจำลองพญาไทยเบิร์ต (PhayaThaiBERT) แต่ยังมีวิธีการตัดคำ หรือแบ่งคำวิธีอื่น ๆ อีกมากในปัจจุบัน ที่อาจนำมาประยุกต์ใช้ในการพัฒนาแบบจำลองในอนาคต ซึ่งอาจส่งผลต่อประสิทธิภาพการทดลองได้

3. นำแนวคิด และแบบจำลองที่ได้ไปพัฒนาระบบ หรือพัฒนาแพลตฟอร์มที่ใช้เพื่อช่วยตัดสินใจว่าประโยคข้อความที่โพสนั้นเข้าข่ายความผิดฐานหมิ่นประมาทตามกฎหมายไทย หรือไม่ เพื่อเป็นเครื่องมือสำหรับช่วยเหลือให้เหยื่อผู้ถูกละเมิดทางไซเบอร์ ใช้ในการประกอบการตัดสินใจสำหรับฟ้องร้องดำเนินคดีทางกฎหมายกับผู้โพสต์ หรือผู้แสดงความคิดเห็นทำให้ตนเกิดความเสียหายได้

มหาวิทยาลัยศิลปากร

รายการอ้างอิง

1. Shewale, R. Social Media Users 2024 (Global Data & Statistics). 2024 [cited 2024; Available from: https://www.demandsage.com/social-media-users/.
2. John, A., et al., Self-harm, suicidal behaviours, and cyberbullying in children and young people: Systematic review. Journal of medical internet research, 2018. 20(4): p. e9044.
3. Article19, ความจริงที่ต้องพูดถึง กรณีสนับสนุนการยกเลิกความผิดฐานหมิ่นประมาททางอาญาในประเทศไทย 2564: https://www.article19.org/wp-content/uploads/2021/03/Thailand_Truth_be_told_decriminalise_defamation_THAI.pdf.
4. Kumari, K., et al. Aggressive social media post detection system containing symbolic images. in Conference on e-Business, e-Services and e-Society. 2019. Springer.
5. Morzhov, S. Avoiding unintended bias in toxicity classification with neural networks. in 2020 26th Conference of Open Innovations Association (FRUCT). 2020. IEEE.
6. Uban, A.-S. and L.P. Dinu. On transfer learning for detecting abusive language online. in International Work-Conference on Artificial Neural Networks. 2019. Springer.
7. Lowphansirikul, L., et al., Wangchanberta: Pretraining transformer-based thai language models. arXiv preprint arXiv:2101.09635, 2021.
8. Sriwirote, P., et al., PhayaThaiBERT: Enhancing a Pretrained Thai Language Model with Unassimilated Loanwords. arXiv preprint arXiv:2311.12475, 2023.
9. Sangwan, S.R. and M.P.S. Bhatia, Soft computing for abuse detection using cyber-physical and social big data in cognitive smart cities. Expert Systems, 2021: p. e12766.
10. Pratiwi, N.I., I. Budi, and M.A. Jiwanggi. Hate speech identification using the hate codes for Indonesian tweets. in Proceedings of the 2019 2nd international conference on data science and information technology. 2019.

11. Sutejo, T.L. and D.P. Lestari. Indonesia hate speech detection using deep learning. in 2018 International Conference on Asian Language Processing (IALP). 2018. IEEE.
12. Modha, S., et al., Detecting and visualizing hate speech in social media: A cyber watchdog for surveillance. Expert Systems with Applications, 2020. 161: p. 113725.
13. Mossie, Z. and J.-H. Wang, Vulnerable community identification using hate speech detection on social media. Information Processing & Management, 2020. 57(3): p. 102087.
14. Agrawal, S. and A. Awekar. Deep learning for detecting cyberbullying across multiple social media platforms. in European conference on information retrieval. 2018. Springer.
15. Bu, S.-J. and S.-B. Cho. A hybrid deep learning system of CNN and LRCN to detect cyberbullying from SNS comments. in International Conference on Hybrid Artificial Intelligence Systems. 2018. Springer.
16. Anand, M. and R. Eswari. Classification of abusive comments in social media using deep learning. in 2019 3rd international conference on computing methodologies and communication (ICCMC). 2019. IEEE.
17. Chen, J., S. Yan, and K.-C. Wong, Verbal aggression detection on Twitter comments: Convolutional neural network for short-text sentiment analysis. Neural Computing and Applications, 2020. 32(15): p. 10809-10818.
18. Maity, S.K., et al., Opinion conflicts: An effective route to detect incivility in Twitter. Proceedings of the ACM on Human-Computer Interaction, 2018. 2(CSCW): p. 1-27.
19. Mohaouchane, H., A. Mourhir, and N.S. Nikolov. Detecting offensive language on arabic social media using deep learning. in 2019 sixth international conference on social networks analysis, management and security (SNAMS). 2019. IEEE.
20. Saeed, H.H., K. Shahzad, and F. Kamiran. Overlapping toxic sentiment classification using deep neural architectures. in 2018 IEEE international conference on data mining workshops (ICDMW). 2018. IEEE.
21. Chiewhawan, T. and P. Vateekul. Explainable deep learning for thai stock market prediction using textual representation and technical indicators. in Proceedings of

the 8th International Conference on Computer and Communications Management. 2020.

22. Horsuwan, T., et al. A comparative study of pretrained language models on thai social text categorization. in Asian Conference on Intelligent Information and Database Systems. 2020. Springer.
23. Thattinaphanich, S. and S. Prom-on. Thai named entity recognition using Bi-LSTM-CRF with word and character representation. in 2019 4th International Conference on Information Technology (InCIT). 2019. IEEE.
24. Ayutthaya, T.S.N. and K. Pasupa. Thai sentiment analysis via bidirectional LSTM-CNN model with embedding vectors and sentic features. in 2018 International joint symposium on artificial intelligence and natural language processing (iSAI-NLP). 2018. IEEE.
25. Pasupa, K. and T.S.N. Ayutthaya, Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features. Sustainable Cities and Society, 2019. 50: p. 101615.
26. Harnmetta, P. and T. Samanchuen. Sentiment analysis of thai stock reviews using transformer models. in 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE). 2022. IEEE.
27. Gatchalee, P., S. Waijanya, and P. Promrit, Thai text classification experiment using cnn and transformer models for timely-timeless content marketing. ICIC Express Letters, 2023. 19: p. 91-101.
28. Arreerard, R. and T. Senivongse. Thai defamatory text classification on social media. in 2018 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD). 2018. IEEE.
29. Hochreiter, S. and J. Schmidhuber, Long short-term memory. Neural computation, 1997. 9(8): p. 1735-1780.
30. Khongtum, O., N. Promrit, and S. Waijanya. The entity recognition of Thai poem compose by Sunthorn Phu by using the bidirectional long short term memory technique. in Multi-disciplinary Trends in Artificial Intelligence: 13th International Conference, MIWAI 2019, Kuala Lumpur, Malaysia, November 17–19, 2019, Proceedings 13. 2019. Springer.

31. Devlin, J., et al., Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
32. Conneau, A., et al., Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116, 2019.
33. Howard, J. and S. Ruder, Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146, 2018.
34. Sornlertlamvanich, V., T. Charoenporn, and H. Isahara, Orchid: Thai partof-speech tagged corpus. National Electronics and Computer Technology Center Technical Report, 1997. 12: p. 5-19.
35. Phatthiyaphaibun, W. PyThaiNLP. 2018 [cited 2021 14]; Available from: https://github.com/PyThaiNLP/pythainlp.
36. Arreerard, R., Defamatory Text Classification on Online Social Media. 2018, Chulalongkorn University.
37. Polpanumas, C., ULMFit language modeling, text feature extraction, and text classification in Thai language. GitHub. 2019.
38. Wong, T.-T. and P.-Y. Yeh, Reliable accuracy estimates from k-fold cross validation. IEEE Transactions on Knowledge and Data Engineering, 2019. 32(8): p. 1586-1594.
39. Watjanapron, P. and O. Chaowalit, USING DEEP LEARNING MODEL WITH MULTIPLE INPUTS FOR THAI DEFAMATORY TEXT CLASSIFICATION ON PUBLIC FACEBOOK COMMENTS.

มหาวิทยาลัยศิลปากร

ประวัติผู้เขียน

ชื่อ-สกุล ปฏิภาณ วัจนาภรณ์

วุฒิการศึกษา สำเร็จการศึกษาปริญญาวิทยาศาสตร์บัณฑิต สาขาเทคโนโลยีสารสนเทศ มหาวิทยาลัยศิลปากร พ.ศ.2560