ระบบวิเคราะห์ความรู้สึกจากข้อความรีวิวภาษาไทยด้วยเทคนิคหน่วยความจำระยะสั้น

นายดุสิต ศิริสาคร

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2568

ระบบวิเคราะห์ความรู้สึกจากข้อความรีวิวภาษาไทยด้วยเทคนิคหน่วยความจำระยะสั้น

นายดุสิต ศิริสาคร

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2568

ใบรับรองการค้นคว้าอิสระ

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง ระบบวิเคราะห์ความรู้สึกจากข้อความรีวิวภาษาไทยด้วยเทคนิคหน่วยความจำระยะสั้น

โดย นายดุสิต ศิริสาคร

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม

..... คณบดีบัณฑิตวิทยาลัย / หัวหน้าภาควิชา

(ผู้ช่วยศาสตราจารย์ ดร.สุนันฑา สดสี)

คณะกรรมการสอบการค้นคว้าอิสระ

..... ประธานกรรมการ

(ดร.ชูชาติ หฤไชยะศักดิ์)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.มาลีรัตน์ มะลิแย้ม)

..... กรรมการ

(อาจารย์ ดร.กาญจนา วิริยะพันธ์)

ชื่อ : นายดุสิต ศิริสาคร
ชื่อการค้นคว้าอิสระ : ระบบวิเคราะห์ความรู้สึกจากข้อความรีวิวภาษาไทยด้วยเทคนิคหน่วยความจำระยะสั้น
สาขาวิชา : วิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก : ผู้ช่วยศาสตราจารย์ ดร.มาลีรัตน์ มะลิแย้ม
ปีการศึกษา : 2568

บทคัดย่อ

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาโมเดลการจำแนกความรู้สึกจากข้อความภาษาไทย โดยประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติผ่านโมเดลภาษาไทยที่ผ่านการฝึกสอนล่วงหน้าร่วมกับโมเดลที่สามารถขยายความจุหน่วยความจำเพื่อรองรับลำดับข้อมูลที่ยาวขึ้น โดยพัฒนาโมเดลทั้งหมด 4 รูปแบบ ได้แก่ โมเดลพื้นฐาน โมเดลที่เพิ่มชั้นประมวลผลสองทิศทาง โมเดลที่รวมการสกัดลักษณะเด่นกับการประมวลผลสองทิศทาง และโมเดลที่ใช้ข้อมูลจาก 4 ชั้นสุดท้าย ผู้วิจัยได้ทดลองกับชุดข้อมูลจำนวน 11,118 ตัวอย่าง แบ่งเป็น 2 คลาส ได้แก่ เชิงบวกและเชิงลบ ผลการวิจัยพบว่าโมเดลที่เพิ่มชั้นประมวลผลสองทิศทางให้ประสิทธิภาพสูงสุด โดยมีค่าความถูกต้องร้อยละ 90.93 งานวิจัยนี้แสดงให้เห็นว่าการผสมผสานโมเดลภาษาที่ผ่านการฝึกสอนล่วงหน้ากับโมเดลหน่วยความจำแบบขยายลำดับข้อมูลอย่างเหมาะสมสามารถพัฒนาระบบจำแนกความรู้สึกภาษาไทยได้อย่างมีประสิทธิภาพ

(มีจำนวนทั้งสิ้น 68 หน้า)

คำสำคัญ : การจำแนกความรู้สึก, โมเดลภาษาที่ผ่านการฝึกสอนล่วงหน้า, โครงข่ายประสาทเทียมแบบสองทิศทาง, การเรียนรู้เชิงลึก, การประมวลผลภาษาไทย

_____ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก

Name : Mr. Dusit Sirisacorn
Independent Study Title : Sentiment analysis system for Thai reviews using long short-term memory
Major Field : Data Science for Innovation
King Mongkut's University of Technology North Bangkok
Independent Study Advisor : Assistant Professor Dr. Maleerat Maliyam
Academic Year : 2025

ABSTRACT

This research aims to develop sentiment classification models for Thai text by applying Natural Language Processing techniques through pre-trained Thai language models combined with memory-extended models capable of handling longer data sequences. Four model architectures were developed: baseline model, bidirectional processing-enhanced model, feature extraction with bidirectional processing hybrid model, and a model utilizing the last four layers. Experiments were conducted on a dataset containing 11,118 samples divided into two classes: positive and negative. The results showed that the bidirectional processing-enhanced model achieved the highest performance with 90.93% accuracy. This research demonstrates that appropriate integration of pre-trained language models with memory-extended sequential processing models can effectively develop Thai sentiment classification systems.

(Total 68 Pages)

Keywords: Sentiment Classification, Pre-trained Language Model, Bidirectional LSTM, Deep Learning, Thai Natural Language Processing

_____ Advisor

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณบิดาและมารดาที่คอยให้การสนับสนุน ดูแล และเป็นกำลังใจสำคัญตลอดระยะเวลาการศึกษา รวมทั้งขอขอบคุณเพื่อน พี่ และน้องทุกคนที่อยู่เคียงข้าง คอยให้คำแนะนำและแรงบันดาลใจ อันเป็นพลังสำคัญที่ช่วยให้ข้าพเจ้ามีความมุ่งมั่นจนสามารถดำเนินงานค้นคว้าอิสระนี้ได้จนสำเร็จ ข้าพเจ้าขอขอบพระคุณเป็นอย่างสูงต่อผู้ช่วยศาสตราจารย์ ดร.มาลีรัตน์ มะลิแย้ม อาจารย์ที่ปรึกษา ซึ่งได้กรุณาให้คำแนะนำ ถ่ายทอดประสบการณ์ และอุทิศเวลาในการให้คำปรึกษาอย่างใกล้ชิดในทุกขั้นตอนของการดำเนินงานวิจัย รวมถึงอาจารย์ ดร.กาญจนา วิริยะพันธ์ และดร.ชูชาติ หฤไชยะศักดิ์ ที่ได้ให้คำแนะนำและข้อคิดเห็นอันทรงคุณค่า ซึ่งมีส่วนสำคัญในการทำให้งานค้นคว้าอิสระฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

ท้ายที่สุดนี้ ข้าพเจ้าขออำนวยพรให้ทุกท่านประสบแต่ความสุข ความเจริญ มีสุขภาพสมบูรณ์แข็งแรง และมีพลังในการสร้างสรรค์สิ่งที่ดีงามต่อไป ด้วยความเคารพและขอบพระคุณอย่างยิ่ง

ดุสิต ศิริสาคร

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
กิตติกรรมประกาศ	ง
สารบัญตาราง	ช
สารบัญภาพ	ซ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์การวิจัย	3
1.3 ขอบเขตของการวิจัย	3
1.4 นิยามศัพท์เฉพาะ	4
1.5 ประโยชน์ที่จะได้รับ	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	5
2.1 การวิเคราะห์ความรู้สึกในภาษาไทย (Thai Sentiment Analysis)	5
2.2 โมเดล BERT และสถาปัตยกรรม Transformer	6
2.3 WangchanBERTa สำหรับภาษาไทย	8
2.4 Bidirectional LSTM	11
2.5 Convolutional Neural Networks สำหรับ Text Classification	13
2.6 Loss Functions และ Weighted Cross Entropy	15
2.7 การประเมินผลและ K-Fold Cross Validation	16
2.8 งานวิจัยที่เกี่ยวข้อง	17
บทที่ 3 วิธีดำเนินการวิจัย	22
3.1 ภาพรวมของการวิจัย (Overview of Research Process)	22
3.2 แหล่งข้อมูลและการเตรียมข้อมูล (Data Source & Preprocessing)	23
3.3 การสำรวจข้อมูล (Exploratory Data Analysis)	25
3.4 การพัฒนาโมเดล (Model Development)	30
3.5 การประเมินผล (Model Evaluation)	36
3.6 การเผยแพร่โมเดล (Model Deployment)	37

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการดำเนินงานวิจัย	42
4.1 ภาพรวมการประเมินผล	42
4.2 ผลการทำ Cross-Validation	42
4.3 การเปรียบเทียบประสิทธิภาพและเวลาการฝึกสอน	43
4.4 การวิเคราะห์ Confusion Matrix และ AUC	44
4.5 การทดสอบโมเดลกับข้อความตัวอย่าง	49
บทที่ 5 สรุป อภิปรายผล และข้อเสนอแนะ	54
5.1 สรุปผลและอภิปรายผล	54
5.2 ข้อเสนอแนะ	57
บรรณานุกรม	58
ภาคผนวก ก	62
การเรียกใช้งานโมเดลและ Web Interface	63
ประวัติผู้เขียน	68

สารบัญตาราง

ตารางที่		หน้า
3-1	สถิติพื้นฐานของชุดข้อมูล Wisesight Sentiment	25
3-2	พารามิเตอร์กลาง (Global Hyperparameters)	35
3-3	พารามิเตอร์เฉพาะของแต่ละโมเดล (Model-Specific Hyperparameters)	36
3-4	รายละเอียด Google Colab Pro	40
3-5	รายละเอียด Hugging Face	41
4-1	ผลการประเมินจาก 5-Fold Cross-Validation	42
4-2	การเปรียบเทียบความเสถียรของโมเดล	43
4-3	การเปรียบเทียบเวลาการฝึกสอนและประสิทธิภาพ	44
4-4	ผลการทำนายข้อความสั้นและซับซ้อน	50
4-5	ผลการทำนายข้อความยาวและมีหลายบริบท	51

สารบัญภาพ

ภาพที่		หน้า
2-1	สถาปัตยกรรม BERT base model	7
2-2	การรวม hidden states จาก 4 ชั้นสุดท้ายของ BERT	8
2-3	สถาปัตยกรรมของ WangchanBERTa	10
2-4	โครงสร้างของ LSTM	11
2-5	โครงสร้างของ LSTM แบบสองทิศทาง (Bidirectional LSTM)	12
2-6	สถาปัตยกรรม CNN สำหรับ Text Classification	14
2-7	การทำงานของ K-Fold Cross Validation	17
3-1	กรอบแนวคิดของงานวิจัย	22
3-2	การกระจายของป้ายกำกับความรู้สึก	26
3-3	การกระจายความยาวข้อความ (หน่วย: ตัวอักษร)	27
3-4	การเปรียบเทียบความยาวข้อความระหว่างประเภทความรู้สึก	28
3-5	ความสัมพันธ์ระหว่างจำนวนคำและจำนวนตัวอักษร	29
3-6	โครงสร้างของโมเดล WCB	31
3-7	โครงสร้างของโมเดล WCB+BiLSTM	32
3-8	โครงสร้างของโมเดล WCB+CNN+BiLSTM	33
3-9	โครงสร้างของโมเดล WCB(4-Layers)+BiLSTM	34
3-10	การเผยแพร่โมเดล	37
3-11	หน้าตัวอย่าง Model Card บน Hugging Face	38
3-12	ตัวอย่างการเรียกใช้งานโมเดลและผลลัพธ์	39
3-13	ตัวอย่างหน้า Web Interface	39
3-14	ตัวอย่างหน้าต่าง Colab Runtime	41
4-1	Confusion Matrix ของ WCB	45
4-2	Confusion Matrix ของ WCB+BiLSTM	45
4-3	Confusion Matrix ของ WCB+CNN+BiLSTM	46
4-4	Confusion Matrix ของ WCB-4Layer+BiLSTM	46
4-5	AUC ของ WCB	47
4-6	AUC ของ WCB+BiLSTM	47
4-7	AUC ของ WCB+CNN+BiLSTM	48

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
4-8	AUC ของ WCB(4-Layers)+BiLSTM	48
ก-1	แสดงหน้าหลักส่วนของ Model card บน Hugging Face	63
ก-2	การเรียกใช้งานโมเดล	64
ก-3	ผลผลลัพธ์จากการเรียกใช้งานโมเดล	65
ก-4	แสดง Web Interface การใช้งานวิเคราะห์ข้อความ	65
ก-5	แสดงผลลัพธ์จากการวิเคราะห์ข้อความ	66
ก-6	แสดง Web Interface การทำนายแบบวิเคราะห์ไฟล์ CSV	66
ก-7	แสดงผลลัพธ์จากการทำนายแบบวิเคราะห์ไฟล์ CSV	67

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคดิจิทัลปัจจุบัน การวิเคราะห์ความรู้สึกจากข้อความ (Sentiment Analysis) ได้กลายเป็นเครื่องมือสำคัญที่ช่วยให้องค์กรต่าง ๆ สามารถเข้าใจความคิดเห็นและความรู้สึกของลูกค้าได้อย่างรวดเร็วและมีประสิทธิภาพ โดยเฉพาะการวิเคราะห์รีวิวสินค้าและบริการที่มีปริมาณมหาศาลในแพลตฟอร์มอีคอมเมิร์ซและสื่อสังคมออนไลน์ การสามารถจำแนกความรู้สึกเชิงบวกและเชิงลบจากข้อความเหล่านี้ได้อย่างแม่นยำจะช่วยให้ผู้ประกอบการสามารถปรับปรุงผลิตภัณฑ์และบริการ วางแผนการตลาด และตอบสนองความต้องการของลูกค้าได้ดียิ่งขึ้น การพัฒนาการวิเคราะห์ความรู้สึกสำหรับภาษาไทยยังคงเผชิญกับความท้าทายหลายประการ เนื่องจากภาษาไทยมีลักษณะเฉพาะที่แตกต่างจากภาษาอังกฤษ เช่น การไม่มีการเว้นวรรคระหว่างคำ (เช่น “สินค้าดีมากจัดส่งเร็ว”) ระบบวรรณยุกต์ที่ซับซ้อน (เช่น “ไม่ดี” กับ “ไม่ดี้” มีความหมายต่างกัน) และการใช้คำซ้ำเพื่อเน้นความหมาย (เช่น “ดีมากมากมาก” หรือ “แย่แย่แย่”) อีกทั้งการใช้ภาษาในสื่อสังคมออนไลน์และแพลตฟอร์มรีวิวมักมีความหลากหลายและความคิดสร้างสรรค์สูง เช่น การใช้คำสแลง คำย่อ และการผสมผสานระหว่างภาษาไทยและภาษาอังกฤษ (เช่น “อร่อยมว๊ากกกก”, “ชิปปิ้งfast มาก”, “โอเคอ่ะ 555+”, “มันส์จุงเบย”, “so so นะ”, “โครตดี”) ลักษณะเหล่านี้ทำให้การประมวลผลภาษาไทยต้องอาศัยแนวทางที่เฉพาะเจาะจงและมีความซับซ้อนมากขึ้น

ความก้าวหน้าในด้านการเรียนรู้เชิงลึก โดยเฉพาะการพัฒนา Transformer-based models เช่น BERT ได้สร้างการเปลี่ยนแปลงครั้งสำคัญในวงการการประมวลผลภาษาธรรมชาติ การพัฒนา WangchanBERTa โดย Lowphansirikul et al. (2021) ซึ่งเป็น Pre-trained Language Model ที่ออกแบบมาเฉพาะสำหรับภาษาไทย ได้แสดงศักยภาพในการปรับปรุงประสิทธิภาพอย่างมีนัยสำคัญ งานวิจัยของ Khamphakdee และ Seresangtakul (2023) ได้แสดงให้เห็นว่าการใช้ deep learning สำหรับการวิเคราะห์ความรู้สึกภาษาไทยสามารถให้ความแม่นยำสูงถึง 92.25% ซึ่งถือเป็นผลลัพธ์ที่น่าประทับใจและสะท้อนให้เห็นถึงศักยภาพของเทคโนโลยีปัจจุบัน ปัจจุบัน การเข้าถึงเครื่องมือและโมเดลสำหรับการวิเคราะห์ความรู้สึกมีความสะดวกมากขึ้น โดยมีบริการ API จากผู้ให้บริการรายใหญ่ เช่น Google Cloud Natural Language API, OpenAI และ Microsoft Azure Cognitive Services ที่สามารถใช้งานได้ทันที นอกจากนี้ยังมี open-source models และ frameworks มากมายที่นักพัฒนาสามารถนำมาใช้งานได้ อย่างไรก็ตาม โมเดลที่มีประสิทธิภาพสูงและเหมาะสมกับภาษาไทยโดยเฉพาะมักต้องเสียค่าบริการที่สูง หรือต้องมีความเชี่ยวชาญในการปรับแต่งโมเดลเอง ทำให้ผู้ประกอบการขนาดเล็กและนักพัฒนาทั่วไปอาจยังเข้าถึงได้ยาก งานวิจัยของ Suraratchai และ

Phoomvuthisarn (2024) ได้ชี้ให้เห็นว่าการใช้เทคนิค ensemble learning ร่วมกับ pre-trained models สามารถปรับปรุงประสิทธิภาพการวิเคราะห์ความรู้สึกภาษาไทยได้อย่างมีประสิทธิภาพ ขณะที่งานของ Boquio และ Naval Jr. (2024) แสดงให้เห็นว่า multi-layer pooling strategies ก็สามารถเพิ่มประสิทธิภาพของโมเดลได้เช่นกัน อย่างไรก็ตาม การเปรียบเทียบสถาปัตยกรรมโมเดลต่าง ๆ อย่างเป็นระบบยังคงเป็นประเด็นที่น่าสนใจ โดยเฉพาะการศึกษาที่เปรียบเทียบ Pure BERT baseline, BERT+BiLSTM, BERT+CNN+BiLSTM และเทคนิค weighted pooling ในงานเดียวกันสำหรับภาษาไทย ซึ่งยังมีโอกาสในการศึกษาและพัฒนาต่อไปเพื่อหาสถาปัตยกรรมที่ให้ผลลัพธ์เหมาะสมที่สุด

ด้วยเหตุนี้ งานวิจัยครั้งนี้จึงมุ่งเน้นการพัฒนาโมเดลการจำแนกความรู้สึกภาษาไทยที่มีประสิทธิภาพสูงและพร้อมใช้งานจริง โดยใช้ WangchanBERTa ซึ่งเป็น Pre-trained Language Model ที่ออกแบบมาเฉพาะสำหรับภาษาไทยเป็นพื้นฐาน และประยุกต์ใช้เทคนิค Bidirectional LSTM (BiLSTM) ร่วมกับสถาปัตยกรรมอื่น ๆ เพื่อเพิ่มความสามารถในการจับความสัมพันธ์และบริบทของข้อความ งานวิจัยนี้พัฒนาและทดสอบโมเดล 4 สถาปัตยกรรม ได้แก่ (1) WangchanBERTa baseline (WCB) (2) WangchanBERTa ร่วมกับ BiLSTM (WCB + BiLSTM) (3) WangchanBERTa ร่วมกับ CNN และ BiLSTM (WCB + CNN + BiLSTM) และ (4) WangchanBERTa 4 layers weighted pooling ร่วมกับ BiLSTM (WCB (4-Layers) + BiLSTM) เพื่อหาสถาปัตยกรรมที่เหมาะสมที่สุดสำหรับการจำแนกความรู้สึกภาษาไทย การวิจัยนี้เลือกใช้การจำแนกเป็น 2 คลาส (เชิงบวกและเชิงลบ) เนื่องจากการประยุกต์ใช้งานในภาคธุรกิจส่วนใหญ่ต้องการการตัดสินใจแบบ binary เพื่อใช้วางแผนปรับปรุงผลิตภัณฑ์และบริการ เช่น การวิเคราะห์รีวิวสินค้าใน e-commerce การตรวจสอบความคิดเห็นในโซเชียลมีเดีย และการพัฒนาระบบช่วยเหลือลูกค้าอัตโนมัติ นอกจากนี้ binary classification ยังมีความเรียบง่ายในการตีความผลลัพธ์และการนำไปใช้งานจริง

ความสำคัญของงานวิจัยนี้อยู่ที่การสร้างโมเดลที่ไม่เพียงแต่มีประสิทธิภาพสูง แต่ยังสามารถเข้าถึงและนำไปใช้งานได้จริงโดยไม่เสียค่าใช้จ่าย โดยโมเดลที่พัฒนาขึ้นจะถูกเผยแพร่บน Hugging Face Model Hub พร้อม Interactive Demo ผ่าน Gradio Spaces และตัวอย่างการใช้งานด้วย Python เพื่อให้นักพัฒนาและผู้ประกอบการขนาดเล็กสามารถนำไปประยุกต์ใช้ได้ทันที ผลลัพธ์ที่ได้จะเป็นทั้งแนวทางในการพัฒนาโมเดลการวิเคราะห์ความรู้สึกภาษาไทยที่มีประสิทธิภาพ และเป็นเครื่องมือที่พร้อมใช้งานจริงในการสนับสนุนการทำงานด้าน Back Office และการวิเคราะห์ความคิดเห็นของลูกค้าในหลากหลายสถานการณ์

1.2 วัตถุประสงค์การวิจัย

เพื่อพัฒนาโมเดลการจำแนกความรู้สึกจากข้อความภาษาไทย โดยประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ (NLP) ร่วมกับโมเดล LSTM

1.3 ขอบเขตของการวิจัย

การวิจัยนี้กำหนดขอบเขตใน 4 ด้านหลัก ได้แก่ ด้านข้อมูล ด้านโมเดล ด้านการประเมินผล และด้านการเผยแพร่การใช้งาน โดยในส่วนของข้อมูลใช้ชุดข้อมูล Wisesight Sentiment Corpus จำนวน 26,737 ข้อความภาษาไทย ครอบคลุมช่วงปี 2016–ต้นปี 2019 ข้อความในชุดข้อมูลประกอบด้วยป้ายกำกับ Positive, Neutral, Negative และ Question ซึ่งเป็นข้อความจากสื่อสังคมออนไลน์ที่ใช้ภาษาไทยกลางในรูปแบบไม่เป็นทางการ เช่น รีวิวสินค้า บทสนทนา ข่าว หรือโฆษณา ใช้เฉพาะข้อมูลที่เผยแพร่สาธารณะและไม่มีข้อมูลส่วนบุคคล เพื่อให้สะท้อนลักษณะการใช้ภาษาจริงของผู้ใช้งานออนไลน์ การเตรียมข้อมูลได้มีการลบหรือปกปิดข้อมูลส่วนบุคคล ตัดข้อความซ้ำ ข้อความยาวเกิน 2,000 ตัวอักษร หรือข้อความที่ไม่ใช่ภาษาไทย โดยเก็บการสะกดผิดและอีโมจิไว้ตามต้นฉบับเพื่อรักษาบริบทและโครงสร้างทางภาษา

ในส่วนของโมเดล ใช้ WangchanBERTa (WCB) เป็นโมเดลตั้งต้น และนำมาประยุกต์ร่วมกับสถาปัตยกรรมอื่น ได้แก่ Bidirectional LSTM (BiLSTM), Convolutional Neural Network (CNN) และการใช้ 4 Layers สุดท้ายของ WCB เพื่อสร้างเวกเตอร์แทนความหมาย โดยออกแบบการทดลองใน 4 สถาปัตยกรรมคือ WCB, WCB + BiLSTM, WCB + CNN + BiLSTM และ WCB (4 Layers) + BiLSTM การประเมินประสิทธิภาพของแต่ละสถาปัตยกรรมใช้เทคนิค 5-Fold Stratified Cross-Validation โดยมีตัวชี้วัดหลัก ได้แก่ Accuracy, F1-Score และ Area Under the Curve (AUC) Score เพื่อประเมินผลในเชิงสถิติอย่างรอบด้าน

การทดสอบโมเดลจะทำกับข้อความตัวอย่างที่มีลักษณะแตกต่างกัน เช่น ข้อความสั้น ข้อความซับซ้อน หรือข้อความที่มีบริบทหลายชั้น เพื่อทดสอบความสามารถของโมเดลในการประมวลผลในสถานการณ์จริง ส่วนของการเผยแพร่และการประยุกต์ใช้งาน จะพัฒนาโมเดลที่พร้อมใช้งานจริงบน Hugging Face Model Hub พร้อม Interactive Demo ผ่าน Gradio Spaces และโค้ดภาษา Python เพื่อให้ผู้พัฒนาสามารถนำไปต่อยอดได้โดยตรง ทั้งนี้ ข้อจำกัดของการวิจัยอยู่ที่การจำแนกแบบ 2 ระดับ (บวก/ลบ) เท่านั้น เพื่อเน้นความแม่นยำและการนำไปใช้งานเชิงปฏิบัติ การทดลองทั้งหมดดำเนินบนสภาพแวดล้อม Google Colab และใช้ข้อมูลจากแหล่งเดียว

1.4 นิยามศัพท์เฉพาะ

1.4.1 การวิเคราะห์ความรู้สึก (Sentiment Analysis) หมายถึง กระบวนการใช้เทคนิคการประมวลผลภาษาธรรมชาติและการเรียนรู้ของเครื่องเพื่อระบุ แยกแยะ และจำแนกความรู้สึกหรือทัศนคติที่แสดงออกในข้อความ

1.4.2 WangchanBERTa หมายถึง Pre-trained Language Model ที่พัฒนาโดย AIResearch บนพื้นฐานของสถาปัตยกรรม RoBERTa ที่ได้รับการฝึกอบรมเฉพาะกับข้อมูลภาษาไทยขนาดใหญ่กว่า 78GB เพื่อให้เหมาะสมกับการประมวลผลภาษาไทยโดยเฉพาะ

1.4.3 Bidirectional LSTM (BiLSTM) หมายถึง สถาปัตยกรรมโครงข่ายประสาทเทียมที่ประมวลผลข้อมูลลำดับในทั้งสองทิศทาง (ไปและกลับ) พร้อมกัน เพื่อจับความสัมพันธ์และบริบทของข้อมูลได้อย่างครอบคลุม โดยเฉพาะการจับ long-term dependencies ในข้อความ

1.4.4 Convolutional Neural Network (CNN) หมายถึง สถาปัตยกรรมโครงข่ายประสาทเทียมที่ใช้การดำเนินการ convolution เพื่อสกัดฟีเจอร์จากข้อมูล โดยสามารถจับ local patterns และ n-gram features ในข้อความได้อย่างมีประสิทธิภาพ

1.4.5 Hugging Face หมายถึง แพลตฟอร์มสำหรับแบ่งปันและใช้งาน machine learning models โดยเฉพาะ natural language processing models ที่ช่วยให้นักพัฒนาสามารถเข้าถึงและใช้งานโมเดลได้ง่าย

1.5 ประโยชน์ที่จะได้รับ

1.5.1 ผลลัพธ์ของงานวิจัยสามารถนำไปประยุกต์ใช้ในการวิเคราะห์ความรู้สึกของข้อความภาษาไทย เพื่อสนับสนุนการทำงานในส่วน Back Office

1.5.2 โมเดลที่พัฒนาขึ้นถูกเผยแพร่บน Hugging Face เพื่อให้นักพัฒนาและผู้ที่สนใจสามารถนำไปต่อยอดและประยุกต์ใช้งานได้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้นำเสนอแนวคิด ทฤษฎี และเทคนิคด้านการประมวลผลภาษาธรรมชาติ (NLP) ที่เกี่ยวข้องกับการจำแนกข้อความภาษาไทย โดยเนื้อหาครอบคลุมสถาปัตยกรรมโมเดลและเทคนิคที่จำเป็นสำหรับการสร้างแบบจำลองเพื่อวิเคราะห์และจำแนกข้อความ ดังนี้

2.1 การวิเคราะห์ความรู้สึกในภาษาไทย (Thai Sentiment Analysis)

การวิเคราะห์ความรู้สึก (Sentiment Analysis) เป็นสาขาหนึ่งของการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) ที่มีความสำคัญอย่างมากในยุคดิจิทัลปัจจุบัน การพัฒนาระบบวิเคราะห์ความรู้สึกสำหรับภาษาไทยเป็นความท้าทายที่น่าสนใจเนื่องจากภาษาไทยมีลักษณะเฉพาะที่แตกต่างจากภาษาอังกฤษอย่างมาก

2.1.1 ความท้าทายของการวิเคราะห์ความรู้สึกภาษาไทย

ภาษาไทยมีลักษณะเฉพาะหลายประการที่ทำให้การวิเคราะห์ความรู้สึกมีความซับซ้อน ประการแรก การไม่มีการเว้นวรรคระหว่างคำ (No Word Boundary) ทำให้การตัดคำเป็นปัญหาพื้นฐานที่สำคัญ (Theeramunkong & Usanavasin, 2001) ประการที่สอง การมีระบบวรรณยุกต์ที่ซับซ้อนและการใช้คำซ้ำเพื่อเน้นความหมาย ประการที่สาม การใช้คำย่อ คำสแลง และอักษรย่อที่หลากหลายในสื่อสังคมออนไลน์ (Lowphansirikul et al., 2021)

งานวิจัยของ Suriyawongkul et al. (2019) ชี้ให้เห็นว่าความท้าทายหลักของการวิเคราะห์ความรู้สึกภาษาไทยประกอบด้วย (1) การขาดแคลนชุดข้อมูลที่มีคุณภาพและขนาดใหญ่ (2) ความหลากหลายของภาษาถิ่นและการใช้ภาษาในสื่อออนไลน์ (3) การผสมผสานระหว่างภาษาไทยและภาษาอังกฤษในข้อความเดียวกัน และ (4) บริบททางวัฒนธรรมที่มีผลต่อการแสดงออกทางอารมณ์

2.1.2 วิวัฒนาการของวิธีการวิเคราะห์ความรู้สึกภาษาไทย

การพัฒนาระบบวิเคราะห์ความรู้สึกภาษาไทยได้ผ่านการวิวัฒนาการจากวิธีการแบบดั้งเดิมไปสู่วิธีการที่ทันสมัยมากขึ้น ในช่วงแรกๆ นักวิจัยใช้วิธีการแบบ Rule-based และ Lexicon-based โดยอาศัยพจนานุกรมความรู้สึกภาษาไทยที่พัฒนาขึ้นเอง (Haruechaiyasak et al., 2013) แม้ว่าวิธีการเหล่านี้จะให้ผลลัพธ์ที่ตีความได้ง่าย แต่มีข้อจำกัดในการจัดการกับความซับซ้อนของภาษาธรรมชาติและการใช้ภาษาที่หลากหลายในสื่องสังคมออนไลน์

ต่อมาการใช้วิธีการแบบ Machine Learning เริ่มได้รับความนิยมมากขึ้น Kulpanich & Thaiprayoon (2019) นำเสนอการใช้ Support Vector Machine (SVM) ร่วมกับเทคนิค feature

engineering ที่มีประสิทธิภาพสำหรับการวิเคราะห์ความรู้สึกจากข้อความรีวิวผลิตภัณฑ์ภาษาไทย โดยใช้ TF-IDF และ N-gram features ร่วมกับการประมวลผลเบื้องต้นที่เหมาะสมกับภาษาไทย

2.1.3 การเข้าสู่ยุคของ Deep Learning

การนำเทคนิค Deep Learning มาใช้ในการวิเคราะห์ความรู้สึกภาษาไทยเริ่มต้นจากการใช้ Recurrent Neural Networks (RNN) และ Long Short-Term Memory (LSTM) Limkonchotiwat et al. (2019) ได้แสดงให้เห็นว่า BiLSTM สามารถจับลำดับของคำและบริบทในประโยคภาษาไทยได้ดีกว่าวิธีการแบบดั้งเดิม โดยเฉพาะในกรณีที่ข้อความมีโครงสร้างที่ซับซ้อนหรือมีการใช้สำนวนที่ต้องอาศัยบริบทในการตีความ

ความก้าวหน้าที่สำคัญเกิดขึ้นเมื่อ Transformer architecture ถูกนำมาประยุกต์ใช้กับภาษาไทย โดยเฉพาะการพัฒนา Pre-trained Language Models ที่เฉพาะเจาะจงสำหรับภาษาไทย เช่น ThAIKer (Limkonchotiwat et al., 2022) และ WangchanBERTa (Lowphansirikul et al., 2021) ซึ่งได้รับการฝึกอบรมบนข้อมูลภาษาไทยขนาดใหญ่และแสดงประสิทธิภาพที่ดีเยี่ยมในงาน downstream tasks ต่างๆ รวมถึงการวิเคราะห์ความรู้สึก

2.2 โมเดล BERT และสถาปัตยกรรม Transformer

2.2.1 รากฐานของ BERT Architecture

Bidirectional Encoder Representations from Transformers (BERT) ที่พัฒนาโดย Devlin et al. (2018) ได้ปฏิวัติวงการ Natural Language Processing ด้วยการนำเสนอแนวคิดการเรียนรู้แบบสองทิศทาง (Bidirectional Learning) ผ่าน Transformer architecture BERT ใช้เทคนิค Masked Language Modeling (MLM) และ Next Sentence Prediction (NSP) ในการฝึกอบรมแบบ self-supervised learning บนข้อมูลขนาดใหญ่

สถาปัตยกรรมของ BERT ประกอบด้วยชั้น Transformer Encoder หลายชั้นที่ซ้อนกัน โดยแต่ละชั้นมี Multi-Head Self-Attention Mechanism และ Feed-Forward Networks การใช้ Self-Attention ช่วยให้โมเดลสามารถจับความสัมพันธ์ระหว่างคำในประโยคได้อย่างมีประสิทธิภาพ โดยไม่ขึ้นกับระยะห่างของคำในประโยค

2.2.2 สูตรคณิตศาสตร์ของ Self-Attention Mechanism

ภายในแต่ละ Transformer block (ดังแสดงในภาพที่ 2-1) จะประกอบด้วย Multi-Head Self-Attention Mechanism ที่มีการคำนวณดังนี้:Query, Key, Value matrices:

$$Q = XW^Q \quad (2\text{-}1)$$

$$K = XW^K \quad (2\text{-}2)$$

$$V = XW^V \quad (2\text{-}3)$$

Attention scores:

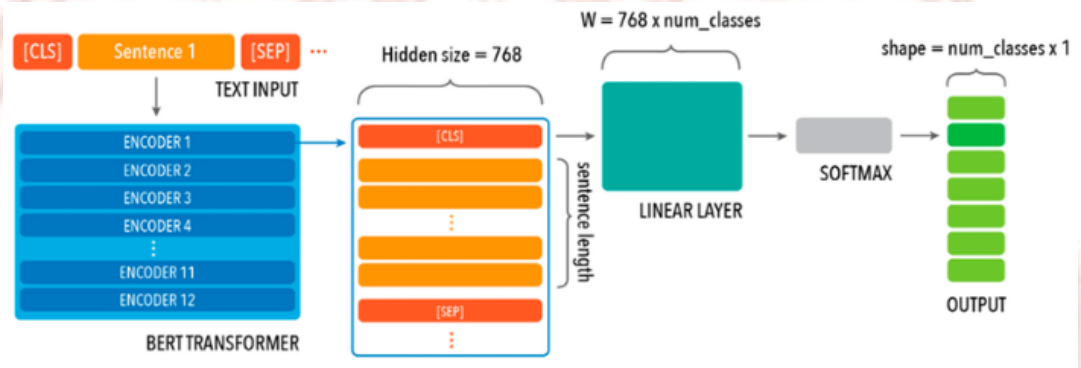
$$Attention(Q,K,V) = softmax(QK^T/\sqrt{d_k})V \quad (2\text{-}4)$$

Multi-Head Attention:

$$MultiHead(Q,K,V) = Concat(head_1,...,head_h)W^O \quad (2\text{-}5)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (2\text{-}6)$$

โดยที่ X คือ input embeddings, W^Q, W^K, W^V คือ learned weight matrices, d_k คือ dimension ของ key vectors และ h คือจำนวน attention heads



ภาพที่ 2-1 สถาปัตยกรรม BERT base model

(ภาพจาก: ResearchGate, 2023)

จากภาพที่ 2-1 จะเห็นว่า BERT ประกอบด้วย 12 ชั้น Encoder ที่ซ้อนกัน โดยแต่ละชั้น Encoder จะใช้กลไก Multi-Head Self-Attention ตามสูตร (2-1) ถึง (2-6) ข้างต้น เพื่อประมวลผล contextual representations ของข้อความ โมเดลมี hidden size = 768 และสิ้นสุดด้วย linear layer และ softmax layer สำหรับการจำแนกประเภทข้อความ การไหลของข้อมูลเริ่มจาก text input ผ่าน BERT transformer layers และผลิต output สำหรับงาน downstream tasks ต่างๆ

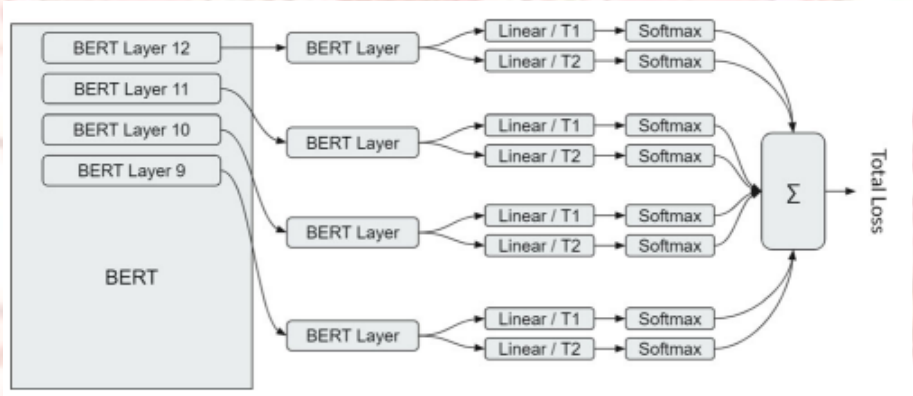
2.2.3 ข้อดีของ Bidirectional Context

ข้อดีหลักของ BERT คือการสามารถเรียนรู้จากบริบททั้งสองทิศทาง (left-to-right และ right-to-left) พร้อมกัน ซึ่งแตกต่างจากโมเดลภาษาแบบดั้งเดิมที่อ่านข้อความทีละทิศทางเท่านั้น การเรียนรู้แบบสองทิศทางนี้ช่วยให้โมเดลเข้าใจบริบทของคำได้ดีกว่า โดยเฉพาะในกรณีที่ความหมายของคำขึ้นกับคำที่อยู่ทั้งข้างหน้าและข้างหลัง

2.2.4 การรวมข้อมูลจากหลายชั้นของ BERT

Devlin et al. (2018) อธิบายว่าโครงสร้างของ BERT ประกอบด้วยหลายชั้น (layers) ซึ่งแต่ละชั้นมีหน้าที่การเรียนรู้ที่แตกต่างกัน โดยชั้นล่างของโมเดลจะเน้นการเรียนรู้คุณลักษณะเชิงโครงสร้าง (syntactic features) เช่น การจัดลำดับคำและความสัมพันธ์ทางไวยากรณ์ ชั้นกลางจะผสมผสานข้อมูลทั้งเชิงโครงสร้างและเชิงความหมาย (semantic features) ขณะที่ชั้นบนสุดมุ่งเน้นการเรียนรู้ความหมายเชิงบริบท (contextual semantic features) ดังนั้นการรวมข้อมูลจากหลายชั้นพร้อมกันจะช่วยให้ได้ representation ที่ครอบคลุมและสมบูรณ์มากขึ้น เมื่อเทียบกับการใช้เพียงชั้นบนสุดหรือ CLS token อย่างเดียว

Galal et al. (2024) ได้นำแนวคิดนี้มาพัฒนาเทคนิคการรวมข้อมูลจาก last four layers (layer 9-12) ของ BERT โดยใช้ weighted sum ซึ่งเป็นการกำหนดให้น้ำหนัก (weight) ของแต่ละ layer สามารถเรียนรู้ได้เองจากการฝึกโมเดล ทำให้ระบบสามารถเลือกได้ว่า layer ใดมีความสำคัญมากที่สุดสำหรับการสร้าง representation ของข้อความ วิธีนี้ช่วยลดการสูญเสียข้อมูลสำคัญ และสามารถรักษาความสมดุลระหว่างข้อมูลเชิงโครงสร้างและความหมาย



ภาพที่ 2-2 การรวม hidden states จาก 4 ชั้นสุดท้ายของ BERT
(ภาพจาก: Galal et al., 2024)

2.3 WangchanBERTa สำหรับภาษาไทย

2.3.1 ความจำเป็นของ Pre-trained Language Model สำหรับภาษาไทย

การพัฒนา Pre-trained Language Models สำหรับภาษาไทยเป็นความจำเป็นเร่งด่วน เนื่องจาก BERT ดั้งเดิมถูกฝึกอบรมบนข้อมูลภาษาอังกฤษเป็นหลัก ทำให้มีประสิทธิภาพจำกัดสำหรับภาษาไทย Lowphansirikul et al. (2021) ได้ชี้ให้เห็นความแตกต่างระหว่างภาษาไทยและภาษาอังกฤษในด้านต่างๆ เช่น ระบบการเขียนที่ไม่มีการเว้นวรรค การมีระบบวรรณยุกต์ที่ซับซ้อน และโครงสร้างทางไวยากรณ์ที่แตกต่าง

2.3.2 สถาปัตยกรรมและการปรับปรุงของ WangchanBERTa

WangchanBERTa เป็น RoBERTa-based model ที่ได้รับการพัฒนาโดย AIResearch โดยการฝึกอบรมบนข้อมูลภาษาไทยขนาดใหญ่ 78.5 GB ที่รวบรวมจากแหล่งข้อมูลที่หลากหลาย รวมถึง Wikipedia ภาษาไทย, ข่าวจากสำนักข่าวในประเทศไทย, โพสต์และคอมเมนต์จากสื่อโซเชียลมีเดีย, ซับไตเติลจากโครงการ OpenSubtitles และข้อมูลจากการ crawl เว็บไซต์ต่างๆ ในประเทศไทย (Lowphansirikul et al., 2021)

โมเดลนี้ถูกออกแบบมาเพื่อประมวลผลภาษาไทยโดยเฉพาะ โดยใช้สถาปัตยกรรม BERT (Bidirectional Encoder Representations from Transformers) ซึ่งอยู่ในกลุ่ม Encoder-only จุดเด่นสำคัญของสถาปัตยกรรมนี้คือ Bidirectional Encoder ที่สามารถประมวลผลข้อความได้จากทั้ง ซ้ายไปขวา และ ขวาไปซ้าย พร้อมกัน ทำให้โมเดลเข้าใจ บริบท (context)และความสัมพันธ์ของคำได้อย่างแม่นยำ เหมาะสำหรับงานด้าน Natural Language Understanding (NLU) เช่น การจำแนกข้อความ (Text Classification) และการวิเคราะห์ความคิดเห็น (Sentiment Analysis)

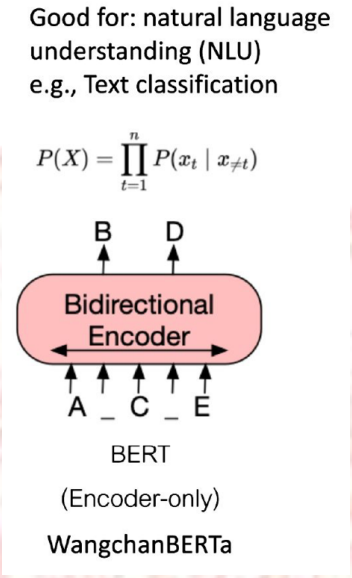
ขั้นตอนการเตรียมข้อมูลและ Tokenization

เนื่องจากภาษาไทยมีความซับซ้อน เช่น ไม่มีการเว้นวรรคระหว่างคำ, การสะกดแบบไม่เป็นทางการ (slang) และคำใหม่ที่เกิดขึ้นตลอดเวลา จึงต้องมีการเตรียมข้อมูลก่อนป้อนเข้าสู่โมเดล WangchanBERTa โดยใช้ SentencePiece Tokenizer ซึ่งถูกฝึกมาเฉพาะสำหรับภาษาไทยเพื่อให้สามารถแบ่งคำเป็น subword หรือ wordpiece ได้แม่นยำยิ่งขึ้น ช่วยแก้ปัญหา Out-of-Vocabulary (OOV) และเพิ่มประสิทธิภาพในการประมวลผลข้อความ

การปรับปรุงที่สำคัญสำหรับภาษาไทยใน WangchanBERTa ได้แก่:

1. SentencePiece Tokenizer – ปรับปรุงกระบวนการตัดคำให้เหมาะสมกับลักษณะเฉพาะของภาษาไทย
2. Vocabulary Expansion – เพิ่มขนาดและความครอบคลุมของคลังคำ (vocabulary) ให้ครอบคลุมคำศัพท์ไทยได้มากยิ่งขึ้น
3. Whole Word Masking – ใช้เทคนิค Whole Word Masking ที่ทำให้โมเดลเรียนรู้จากคำเต็ม (whole word) ไม่ใช่เพียงส่วนย่อยของคำ เหมาะสมกับโครงสร้างของภาษาไทย

สถาปัตยกรรมโมเดล



ภาพที่ 2-3 สถาปัตยกรรมของ WangchanBERTa

(ภาพจาก: VisAI, 2025)

โครงสร้างโมเดล WangchanBERTa ประกอบด้วย:

- 12 ชั้น Transformer Encoder
- 12 Attention Heads
- Hidden State ขนาด 768
- พารามิเตอร์ทั้งหมดประมาณ 110 ล้านตัว
- รองรับความยาวของข้อความสูงสุด 512 tokens

วิธีการฝึกอบรม

โมเดล WangchanBERTa ใช้เทคนิค Masked Language Modeling (MLM) ในการฝึกอบรม ซึ่งเป็นการสุ่มซ่อนบางคำในประโยคแล้วให้โมเดลทำนายคำที่หายไป โดย ไม่ใช้ Next Sentence Prediction (NSP) ตามแนวทางของ RoBERTa ซึ่งพบว่าให้ประสิทธิภาพดีกว่า BERT ดั้งเดิม

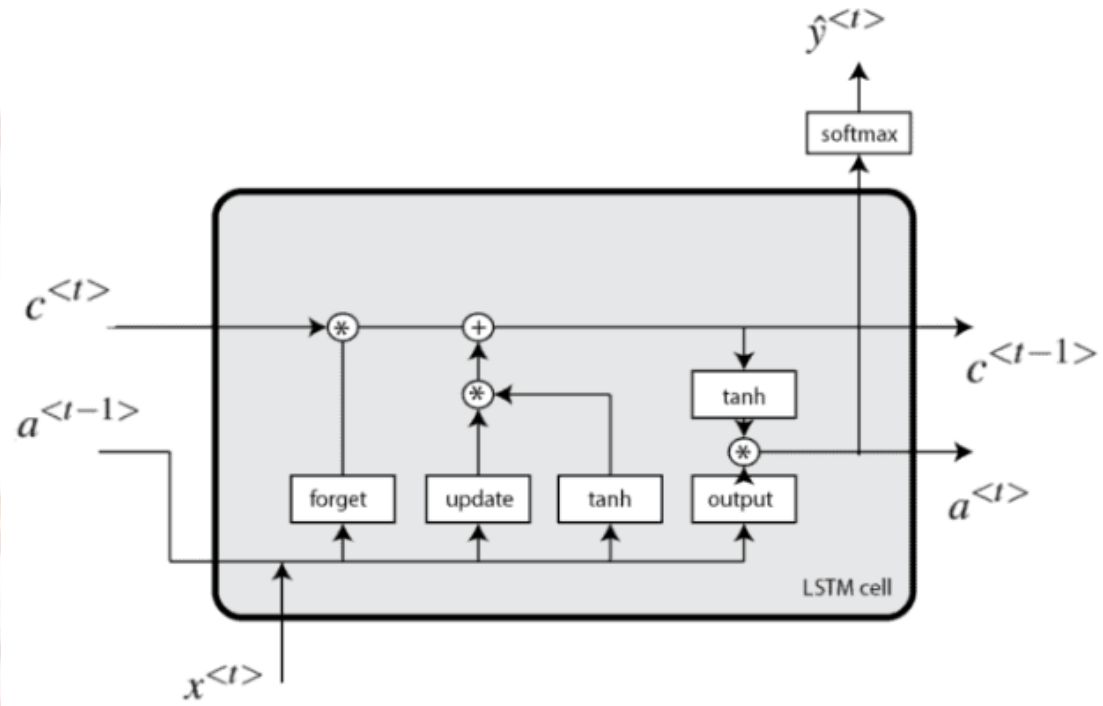
2.3.3 การประยุกต์ใช้ในงาน Sentiment Analysis

ด้วยความสามารถในการเข้าใจบริบทของภาษาไทยอย่างลึกซึ้ง WangchanBERTa จึงเหมาะสำหรับงาน Sentiment Analysis เช่น การวิเคราะห์ความคิดเห็นของผู้ใช้จากรีวิวสินค้า บริการ หรือโพสต์บนสื่อสังคมออนไลน์ โดยโมเดลจะช่วยให้สามารถแยกประเภทความรู้สึกได้อย่างแม่นยำ เช่น เชิงบวก เชิงลบ หรือกลาง ๆ (Positive, Negative, Neutral)

2.4 Bidirectional LSTM

2.4.1 ทฤษฎีพื้นฐานของ LSTM

LSTM (Long Short-Term Memory) เป็นโครงข่ายประสาทเทียมแบบ Recurrent Neural Network (RNN) ที่ถูกพัฒนาขึ้นเพื่อแก้ปัญหาหลักของ RNN ดั้งเดิม ได้แก่ ปัญหาการหายไปของกราดิเอนต์ (Vanishing Gradient Problem) และ ปัญหาการระเบิดของกราดิเอนต์ (Exploding Gradient Problem) ซึ่งเกิดขึ้นเมื่อโมเดลต้องเรียนรู้ข้อมูลลำดับ (Sequential Data) ที่มีความยาวมาก เช่น ประโยคข้อความ การบันทึกเสียง หรือข้อมูลอนุกรมเวลา (Time Series)



ภาพที่ 2-4 โครงสร้างของ LSTM

(ภาพจาก: baeldung, 2025)

LSTM ประกอบด้วย 3 กลไกหลัก (Gates) ได้แก่

1. Forget Gate
    - ทำหน้าที่ตัดสินใจว่าจะ "ลืม" ข้อมูลใดใน Memory Cell
    - ใช้ค่า Sigmoid Activation Function เพื่อให้ผลลัพธ์อยู่ระหว่าง 0 ถึง 1
    - ค่าใกล้ 0 หมายถึง ลืมข้อมูล, ค่าใกล้ 1 หมายถึง เก็บข้อมูลไว้
2. Input Gate
    - กำหนดว่าข้อมูลใหม่ใดควรถูกเพิ่มเข้ามาใน Memory Cell
    - ใช้ Sigmoid เพื่อระบุว่าข้อมูลควรถูกเพิ่มหรือไม่ และ Tanh เพื่อสร้างค่าที่จะเก็บ

3. Output Gate

- ตัดสินใจว่าข้อมูลใดจาก Memory Cell จะถูกส่งออกไปยังชั้นถัดไป
- ช่วยให้ LSTM สามารถนำข้อมูลที่จำเป็นไปใช้งานได้อย่างเหมาะสม

สูตรคณิตศาสตร์ของ LSTM

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2\text{-}7) \text{ \# Forget gate}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2\text{-}8) \text{ \# Input gate}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2\text{-}9) \text{ \# Candidate values}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2\text{-}10) \text{ \# Cell state}$$

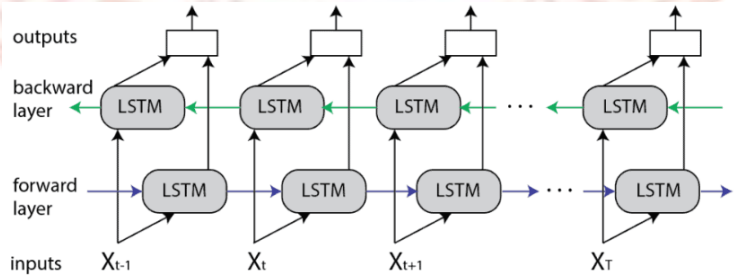
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2\text{-}11) \text{ \# Output gate}$$

$$h_t = o_t * \tanh(C_t) \quad (2\text{-}12) \text{ \# Hidden state}$$

โดยที่ σ คือ sigmoid function, W คือ weight matrices, b คือ bias vectors

2.4.2 ลักษณะของ Bidirectional LSTM

เป็นการพัฒนาโครงข่าย LSTM แบบดั้งเดิม โดยเพิ่มความสามารถในการประมวลผลข้อมูลจากทั้งสองทิศทางพร้อมกัน คือ Forward LSTM ที่ประมวลผลข้อมูลจากจุดเริ่มต้นไปยังจุดสิ้นสุด และ Backward LSTM ที่ประมวลผลจากจุดสิ้นสุดย้อนกลับไปยังจุดเริ่มต้น วิธีนี้ช่วยให้โมเดลสามารถเรียนรู้ บริบท (context) ได้อย่างสมบูรณ์มากขึ้น โดยเฉพาะในงานประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) เช่น การวิเคราะห์ความรู้สึก (Sentiment Analysis) การจำแนกประเภทข้อความ (Text Classification) และการแปลภาษา (Machine Translation) ในการทำงาน BiLSTM จะรวมผลลัพธ์จากทั้งสองทิศทางเข้าด้วยกัน เช่น การเชื่อมต่อแบบ Concatenation หรือ การรวมแบบเฉลี่ย (Average) ก่อนจะส่งต่อไปยังชั้นถัดไป เช่น Dense Layer หรือ Softmax เพื่อทำการพยากรณ์



ภาพที่ 2-5 โครงสร้างของ LSTM แบบสองทิศทาง (Bidirectional LSTM)

(ภาพจาก: baeldung, 2025)

จากรูปที่ 2-5 จะเห็นว่า Bidirectional LSTM ประกอบด้วยชั้น LSTM สองชั้นที่ทำงานแยกกัน โดย forward layer ประมวลผลจาก X_{t-1} ไป X_T และ backward layer ประมวลผลจาก X_T ไป X_{t-1} ผลลัพธ์จากทั้งสองทิศทางจะถูกรวมเข้าด้วยกันเป็น output สุดท้าย

สูตรสำหรับ Bidirectional LSTM:

$$\overrightarrow{h}_t = \text{LSTM\_forward}(x_t, \overrightarrow{h}_{t-1}) \quad (2\text{-}13)$$

$$\overleftarrow{h}_t = \text{LSTM\_backward}(x_t, \overleftarrow{h}_{t+1}) \quad (2\text{-}14)$$

$$h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t] \quad (2\text{-}15)$$

โดยที่ $\overrightarrow{h}_t$ และ $\overleftarrow{h}_t$ คือ hidden states จากทิศทาง forward และ backward ตามลำดับ

2.4.3 ข้อดีและข้อเสียของ Bidirectional LSTM

ข้อดี:

1. บริบทสมบูรณ์: สามารถเรียนรู้จากข้อมูลทั้งอดีตและอนาคต
2. ประสิทธิภาพสูง: ให้ผลลัพธ์ที่ดีกว่าในงาน classification tasks
3. การเข้าใจบริบท: เหมาะสำหรับงานที่ความหมายขึ้นกับบริบททั้งสองทาง

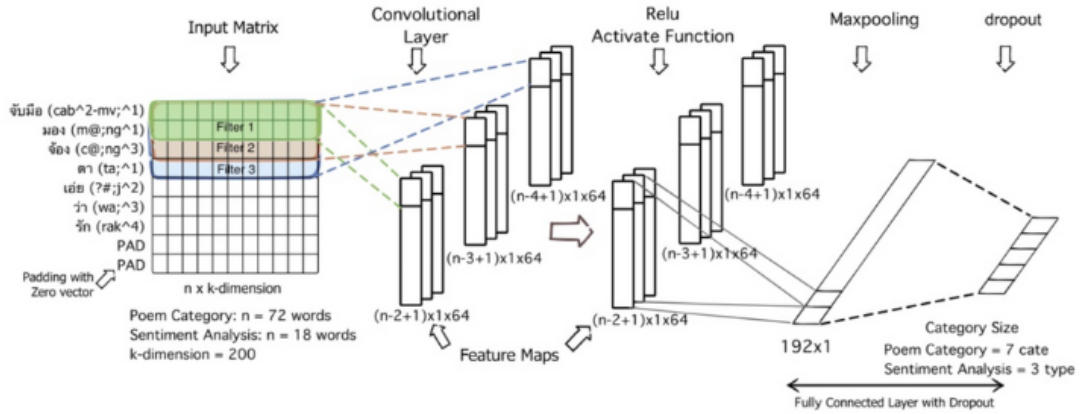
ข้อเสีย:

1. ความซับซ้อน: ใช้ทรัพยากรมากกว่า unidirectional LSTM
2. เวลาประมวลผล: ต้องใช้เวลาในการฝึกอบรมมากกว่า
3. Real-time limitations: ไม่เหมาะสำหรับงานที่ต้องการผลลัพธ์แบบ real-time

2.5 Convolutional Neural Networks สำหรับ Text Classification

2.5.1 หลักการพื้นฐานของ CNN สำหรับข้อความ

Convolutional Neural Networks (CNN) ที่เดิมได้รับการพัฒนาสำหรับการประมวลผลภาพได้ถูกนำมาประยุกต์ใช้กับงานประมวลผลภาษาธรรมชาติอย่างมีประสิทธิภาพ งานของ Kim (2014) เป็นหนึ่งในงานแรกที่แสดงให้เห็นว่า CNN สามารถเรียนรู้ local features และ n-gram patterns ได้อย่างมีประสิทธิภาพ



ภาพที่ 2-6 สถาปัตยกรรม CNN สำหรับ Text Classification

(ภาพจาก: PJJOP, 2025)

2.5.2 สูตรคณิตศาสตร์ของ CNN สำหรับข้อความ

กระบวนการพื้นฐานเริ่มจากการแปลงข้อความให้เป็นลำดับเวกเตอร์ของคำ (word embeddings):

$$X = [x_1, x_2, ..., x_n] \quad (2\text{-}16)$$

จากนั้น CNN จะประมวลผลด้วย 1D convolution เพื่อดึงคุณลักษณะจากลำดับเวกเตอร์:

$$C_i = f(W * X_{i:i+h-1} + b) \quad (2\text{-}17)$$

ค่าผลลัพธ์ทั้งหมดจะถูกรวมเป็น feature map:

$$Feature\_map = [C_1, C_2, ..., C_{n-h+1}] \quad (2\text{-}18)$$

Max Pooling Operation:

$$pooled\_feature = \max(Feature\_map) \quad (2\text{-}19)$$

โดยที่:

- W คือ filter weights
- h คือ filter size
- f คือ activation function
- C_i คือค่าที่ได้หลัง convolution ของตำแหน่งที่ i

2.5.3 ข้อดีของ CNN ในงาน Sentiment Analysis

1. Local Pattern Recognition: CNN สามารถจับ local patterns และ n-gram features ได้ดี
2. Translation Invariance: สามารถจับ patterns ได้โดยไม่ขึ้นกับตำแหน่งในข้อความ
3. Computational Efficiency: ประมวลผลเร็วกว่า RNN-based models
4. Multiple Filter Sizes: สามารถใช้ filter หลายขนาดเพื่อจับ patterns ที่หลากหลาย

2.6 Loss Functions และ Weighted Cross Entropy

2.6.1 ปัญหาของข้อมูลที่ไม่สมดุล

ในงาน sentiment analysis ข้อมูลมักจะมีการกระจายที่ไม่สมดุล (imbalanced dataset) ซึ่งหมายความว่าบางคลาสมีข้อมูลมากกว่าคลาสอื่นๆ อย่างมีนัยสำคัญ การใช้ Cross Entropy Loss แบบปกติอาจทำให้โมเดลเอนเอียงไปยังคลาสที่มีข้อมูลมากกว่า

2.6.2 Weighted Cross Entropy Loss

Weighted Cross Entropy เป็นการปรับปรุง standard cross entropy โดยการให้น้ำหนักที่แตกต่างกันสำหรับแต่ละคลาส:

Standard Cross Entropy:

$$CE = -\sum(y\_i * \log(\hat{y}\_i)) \quad (2\text{-}20)$$

Weighted Cross Entropy:

$$WCE = -\sum(w\_i * y\_i * \log(\hat{y}\_i)) \quad (2\text{-}21)$$

โดยที่ w_i คือน้ำหนักสำหรับคลาส i

2.6.3 การคำนวณน้ำหนัก

วิธีในการคำนวณน้ำหนัก:

1. Inverse Frequency Weighting:

$$w\_i = n\_total / (n\_classes * n\_i) \quad (2\text{-}22)$$

2. Effective Number of Samples:

$$w\_i = (1 - \beta) / (1 - \beta^{n\_i}) \quad (2\text{-}23)$$

โดยที่:

- n_total คือจำนวนตัวอย่างทั้งหมด
- n_classes คือจำนวนคลาสทั้งหมด
- n_i คือจำนวนตัวอย่างในคลาส i
- β คือ hyperparameter (มักใช้ 0.9 หรือ 0.99)

2.6.4 ข้อดีของ Weighted Cross Entropy

1. Balance Learning: ช่วยให้โมเดลเรียนรู้จากคลาสที่มีข้อมูลน้อยได้ดีขึ้น
2. Improved Minority Class Performance: เพิ่มประสิทธิภาพในการจำแนกคลาสที่มีข้อมูลน้อย
3. Flexible Weighting: สามารถปรับน้ำหนักได้ตามความสำคัญของแต่ละคลาส

4. Better Generalization: ลดการ overfitting ต่อคลาสที่มีข้อมูลมาก

2.7 การประเมินผลและ K-Fold Cross Validation

2.7.1 ความสำคัญของการประเมินผลที่เหมาะสม

การประเมินผลโมเดลสำหรับงานวิเคราะห์ความรู้สึกจำเป็นต้องใช้ตัวชี้วัด (metrics) ที่หลากหลาย เพื่อให้ได้มุมมองที่ครบถ้วนเกี่ยวกับประสิทธิภาพของโมเดล โดยเฉพาะอย่างยิ่งเมื่อข้อมูลมีการกระจายไม่สมดุล (imbalanced data) หากพิจารณาเพียง Accuracy อย่างเดียวอาจทำให้เกิดความเข้าใจคลาดเคลื่อน

Accuracy เป็นตัวชี้วัดพื้นฐานที่ใช้วัดสัดส่วนของการทำนายที่ถูกต้องทั้งหมด แต่ Brodersen et al. (2010) ชี้ให้เห็นว่า หากข้อมูลมีการกระจายไม่สมดุล โมเดลอาจมีแนวโน้มทำนายเฉพาะคลาสที่มีข้อมูลมากกว่า ส่งผลให้ค่า Accuracy ดูสูง แต่จริง ๆ แล้วโมเดลไม่ได้เข้าใจหรือตรวจจับคลาสที่มีข้อมูลน้อยได้ดี ดังนั้นจึงควรใช้ metrics อื่น เช่น Precision, Recall, F1-score หรือ ROC-AUC ร่วมด้วย เพื่อประเมินประสิทธิภาพของโมเดลได้รอบด้านมากขึ้น.

2.7.2 เมทริกซ์การประเมินผล

Precision, Recall และ F1 Score

$$\text{Precision} = TP / (TP + FP) \quad (2\text{-}24)$$

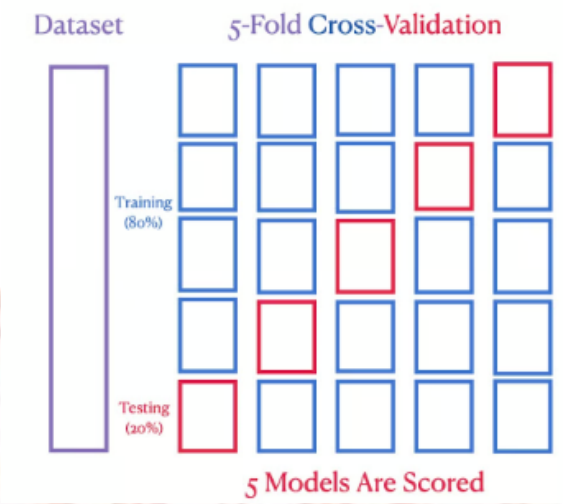
$$\text{Recall} = TP / (TP + FN) \quad (2\text{-}25)$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (2\text{-}26)$$

F1-Score เป็นค่าที่ผสมระหว่าง Precision และ Recall เข้าด้วยกัน เพื่อสะท้อนความสมดุลระหว่างความถูกต้องในการทำนาย (Precision) และความครอบคลุมของการทำนาย (Recall) เหมาะสำหรับกรณีที่ต้องการวัดประสิทธิภาพของโมเดลในข้อมูลที่อาจไม่สมดุล แต่ยังคงมุ่งเน้นการประเมินผลภาพรวมของแต่ละคลาสโดยตรง หลายงานวิจัยด้าน Sentiment Analysis (เช่น Yang et al., 2019) แนะนำให้ใช้ F1-Score เป็นเมทริกซ์หลัก เนื่องจากสะท้อนประสิทธิภาพโดยรวมได้ดีกว่า Accuracy เมื่อข้อมูลไม่สมดุล

2.7.3 K-Fold Cross Validation

K-Fold Cross Validation เป็นเทคนิคมาตรฐานในการประเมินโมเดล โดยแบ่งข้อมูลออกเป็น k ส่วนเท่าๆ กัน และใช้ k-1 ส่วนสำหรับการฝึกอบรม และ 1 ส่วนสำหรับการทดสอบ มีการใช้ k-fold cross-validation เป็นมาตรฐานในการประเมินโมเดล Kohavi (1995) ได้แสดงให้เห็นว่า 5-fold หรือ 10-fold cross-validation ให้การประมาณที่เชื่อถือได้สำหรับประสิทธิภาพของโมเดล



ภาพที่ 2-7 การทำงานของ K-Fold Cross Validation

(ภาพจาก: datacamp, 2024)

สูตรการคำนวณ

$$CV\_Score = (1/k) \times \sum(Score\_i) \quad (2\text{-}28)$$

$$CV\_Std = \sqrt{[(1/k) \times \sum(Score\_i - CV\_Score)^2]} \quad (2\text{-}29)$$

Stratified Sampling สำหรับข้อมูลที่ไม่สมดุล การใช้ stratified sampling เป็นสิ่งจำเป็นเพื่อให้แต่ละ fold มีสัดส่วนของคลาสที่เหมือนกับข้อมูลต้นฉบับ Sechidis et al. (2011) ได้พิสูจน์ว่าวิธีนี้ให้การประมาณที่ดีกว่าการสุ่มแบบทั่วไป

2.7.4 การป้องกัน Overfitting

Early Stopping เทคนิค early stopping ช่วยป้องกัน overfitting โดยการหยุดการฝึกอบรมเมื่อประสิทธิภาพบนชุดข้อมูลตรวจสอบไม่มีการปรับปรุง Prechelt (1998) ได้แสดงให้เห็นว่า early stopping เป็นเทคนิค regularization ที่มีประสิทธิภาพ

Dropout และ Weight Decay การใช้ dropout และ weight decay เป็นเทคนิคมาตรฐานในการป้องกัน overfitting Srivastava et al. (2014) ได้พิสูจน์ประสิทธิภาพของ dropout ในการปรับปรุงการ generalize ของโมเดล

2.8 งานวิจัยที่เกี่ยวข้อง

Suraratchai & Phoomvuthisarn (2024) ได้พัฒนาโมเดลแบบ Parallel Hybrid โดยนำ WangchanBERTa มาผสมผสานกับสถาปัตยกรรม CNN และ BiLSTM ในรูปแบบต่างๆ 3 แนวทาง ได้แก่ CNN เชื่อมต่อกับ BiLSTM, BiLSTM เชื่อมต่อกับ CNN และ Parallel Hybrid ที่ CNN และ BiLSTM ประมวลผลคู่ขนานแล้วรวมผลลัพธ์เข้าด้วยกัน ผู้วิจัยใช้ชุดข้อมูล Wisesight Sentiment

Corpus ที่มี 4 คลาส (Positive, Negative, Neutral, Question) จำนวน 26,676 ข้อความ และ Thai Children's Tales dataset จำนวน 1,115 ข้อความ โดยแบ่งข้อมูลด้วยอัตราส่วน train-validation-test เป็น 60:20:20 และประเมินประสิทธิภาพด้วย macro F1-score และ accuracy ผลการทดลองพบว่า Parallel Hybrid architecture ให้ผลลัพธ์ดีที่สุดด้วยค่า macro F1 สูงสุดที่ 0.6270 รองลงมาคือ CNN-BiLSTM ที่ได้ค่า macro F1 = 0.6123 และ BiLSTM-CNN ที่ได้ค่า macro F1 = 0.6057 ข้อดีของงานนี้คือโมเดล Parallel Hybrid สามารถจัดการกับข้อความที่มีความยาวแตกต่างกันได้ดีและการใช้ WangchanBERTa ช่วยให้มีความเข้าใจบริบทภาษาไทยได้ดีขึ้น แต่ข้อเสียคือโมเดลมีความซับซ้อนสูงทำให้ใช้ทรัพยากรการคำนวณมากและประสิทธิภาพที่ได้ยังคงมีห้องสำหรับการปรับปรุง

Khamphakdee & Seresangtakul (2023) ได้เปรียบเทียบประสิทธิภาพของเทคนิค word embedding ต่างๆ รวมถึง Word2Vec (CBOW และ skip-gram), FastText และ WangchanBERTa ร่วมกับโมเดลการเรียนรู้เชิงลึก 9 แบบ ได้แก่ CNN, LSTM, BiLSTM, GRU, BiGRU, CNN-LSTM, CNN-BiLSTM, CNN-GRU และ CNN-BiGRU สำหรับการวิเคราะห์ความรู้สึกในโดเมนโรงแรม ผู้วิจัยสร้างชุดข้อมูลรีวิวโรงแรมภาษาไทยจำนวน 22,018 รายการ แบ่งเป็น positive 11,086 รายการ และ negative 10,932 รายการ โดยแบ่งข้อมูลด้วยอัตราส่วน train-validation-test เป็น 70:15:15 และประเมินผลด้วย accuracy, precision, recall และ F1-score ผลการศึกษาพบว่า WangchanBERTa ให้ประสิทธิภาพดีที่สุดด้วย accuracy สูงสุดที่ 92.25%, precision 92.04%, recall 92.91% และ F1-score 92.47% ซึ่งเหนือกว่าการผสมผสานระหว่าง CNN กับ Word2Vec skip-gram ที่ได้ accuracy 91.70% และ XML-RoBERTa ที่ได้ accuracy 91.95% ในขณะที่ multilingual BERT มีประสิทธิภาพต่ำมากด้วย accuracy เพียง 75.45% ข้อดีของงานนี้คือการใช้ WangchanBERTa ที่ได้รับการฝึกอบรมเฉพาะกับข้อมูลภาษาไทยแสดงให้เห็นประสิทธิภาพที่ดีกว่า multilingual models อย่างมีนัยสำคัญและการประมวลผลข้อมูลด้วย PyThaiNLP สามารถปรับปรุงประสิทธิภาพได้ประมาณ 2% แต่ข้อเสียคืองานวิจัยเน้นเฉพาะ binary classification และการใช้ข้อมูลเฉพาะในโดเมนโรงแรมทำให้โมเดลอาจไม่สามารถ generalize ได้ดีในโดเมนอื่นๆ

Galal et al. (2024) ได้ทำการศึกษาแนวทางใหม่ในการสร้าง sentence embedding สำหรับงานจำแนกข้อความ (text classification) โดยมุ่งเน้นที่การปรับปรุงวิธีการดึงข้อมูลจากหลายชั้นของ BERT เพื่อให้ได้ representation ที่มีคุณภาพสูงขึ้น โดยผู้วิจัยได้เสนอแนวคิด Last-4 Weighted Layer ซึ่งดึงข้อมูลจาก 4 ชั้นสุดท้าย (layer 9–12) ของ BERT มารวมกัน โดยให้น้ำหนักแต่ละชั้นสามารถเรียนรู้ได้เองผ่านกระบวนการฝึกโมเดล จากนั้น representation ดังกล่าวจะถูกส่งต่อไปยังขั้นตอนจำแนกประเภท ชุดข้อมูลที่ใช้ในการทดลองครอบคลุมหลายโดเมน เช่น sentiment

analysis และ sarcasm detection เพื่อประเมินความสามารถของเทคนิคที่นำเสนอ ผลการทดลองพบว่า Last-4 Weighted Layer สามารถเพิ่มประสิทธิภาพของโมเดลได้ดีกว่าวิธีดั้งเดิมอย่างชัดเจน เช่น mean pooling หรือการใช้ CLS token แบบเดิม โดยเฉพาะในงาน sarcasm detection ค่า F1-score ของโมเดลเพิ่มขึ้นเป็น 64.41% ซึ่งสูงกว่าผลลัพธ์ของวิธีพื้นฐานประมาณ 3-4% นอกจากนี้ยังพบว่าเทคนิคนี้ช่วยให้โมเดลมีความเสถียรมากขึ้นและสามารถ generalize ได้ดีขึ้นเมื่อทดสอบกับชุดข้อมูลจากหลายประเภท ผู้วิจัยยังได้เสนอกรอบแนวคิดการรวมหลายชั้นแบบอื่น เช่น P-SUM และ H-SUM เพื่อเปรียบเทียบผลการทดลอง แต่จากผลการประเมิน พบว่าแนวทางการรวมแบบขนานด้วย Last-4 Weighted Layer มีประสิทธิภาพสูงกว่าและมีความซับซ้อนของโมเดลต่ำกว่า เหมาะสำหรับนำไปประยุกต์ใช้งานจริงในงานจำแนกข้อความ โดยเฉพาะงาน sentiment analysis และงานที่ต้องการ contextual representation คุณภาพสูง

Lehečka et al. (2020) ได้พัฒนา pooling layer architecture ที่วางอยู่บน BERT models เพื่อปรับปรุงคุณภาพการจำแนกประเภทในงาน Large-scale Multi-label Text Classification (LMTC) โดยการรวมข้อมูลจาก standard [CLS] token กับ pooled sequence output ที่ได้จากการประมวลผลทั้งลำดับของข้อความ ผู้วิจัยใช้ชุดข้อมูล Wikipedia ในสามภาษาที่แตกต่างกัน โดยใช้ pre-trained BERT models ที่เปิดเผยต่อสาธารณะและทดสอบการผสมผสานระหว่าง max pooling และ mean pooling ในรูปแบบต่างๆ การประเมินผลใช้เมทริกส์มาตรฐานสำหรับ multi-label classification ผลการทดลองพบว่าการผสมผสานข้อมูลจาก max pooling และ mean pooling ให้การปรับปรุงประสิทธิภาพที่สูงที่สุดเมื่อเปรียบเทียบกับการใช้ [CLS] token เพียงอย่างเดียว ข้อดีของงานนี้คือแสดงให้เห็นว่าการใช้ข้อมูลจาก sequence output ทั้งหมดร่วมกับ [CLS] token สามารถปรับปรุงประสิทธิภาพในงาน large-scale classification ได้และวิธีการนี้สามารถประยุกต์ใช้กับ BERT models ที่มีอยู่โดยไม่ต้องการการฝึกอบรมใหม่ แต่ข้อเสียคืองานวิจัยไม่ได้ทดสอบกับโดเมนอื่นๆ นอกเหนือจาก Wikipedia และไม่มีการศึกษาเปรียบเทียบกับ state-of-the-art models อื่นๆ ในช่วงเวลานั้น

Innork et al. (2023) ได้ทำการศึกษาเปรียบเทียบโมเดล multi-class sentiment classification สำหรับรีวิวลูกค้าโรงแรมโดยใช้ BERT model สำหรับการวิเคราะห์ความรู้สึกในข้อความภาษาไทย ผู้วิจัยใช้ชุดข้อมูลรีวิวโรงแรมภาษาไทยและประยุกต์ใช้เทคนิค deep learning ต่างๆ รวมถึงการ fine-tuning BERT model เพื่อให้เหมาะสมกับงาน sentiment classification ในโดเมนโรงแรม การประเมินผลใช้ accuracy และ F1-score เป็นหลัก ผลการทดลองแสดงให้เห็นว่า BERT model ให้ประสิทธิภาพที่ดีเยี่ยมด้วย accuracy ที่ 89.31% และ F1-score ที่ 89.43% ซึ่งสูงกว่างานวิจัยก่อนหน้านี้ในโดเมนเดียวกัน ข้อดีของงานนี้คือการนำเสนอวิธีการที่มีประสิทธิภาพสูงในการวิเคราะห์ความรู้สึกสำหรับอุตสาหกรรมการท่องเที่ยวและโรงแรม และผลลัพธ์ที่ได้สามารถ

ช่วยผู้ประกอบการในการตอบสนองต่อผลเสริมของลูกค้า แต่ข้อเสียคืองานวิจัยเน้นเฉพาะโดเมนโรงแรมและอาจไม่สามารถประยุกต์ใช้ได้ดีกับโดเมนอื่นๆ โดยตรง

Talaat (2023) ได้ศึกษาการพัฒนาระบบ sentiment analysis classification ที่ใช้ hybrid BERT models โดยการผสมผสาน DistilBERT และ RoBERTa กับ BiGRU และ BiLSTM layers เพื่อเพิ่มความสามารถในการจับลำดับและความสัมพันธ์ของข้อความ ผู้วิจัยทดลอง 4 สถาปัตยกรรมหลัก ได้แก่ DistilBERT+BiGRU, DistilBERT+BiLSTM, RoBERTa+BiGRU และ RoBERTa+BiLSTM โดยทดสอบทั้งกรณีที่มี emoji และไม่มี emoji ในข้อความ ชุดข้อมูลที่ใช้ประกอบด้วย Twitter sentiment datasets และ IMDB movie reviews รวมประมาณ 50,000 samples การประเมินผลใช้ accuracy, precision, recall และ F1-score โดยเปรียบเทียบกับ classical machine learning methods และ individual pre-trained models ผลการทดลองพบว่า RoBERTa+BiGRU ให้ประสิทธิภาพที่ดีที่สุดโดยได้ accuracy 94.2% และ F1-score 0.941 สำหรับข้อมูลที่ไม่มี emoji ขณะที่ DistilBERT+BiLSTM เหมาะสมกับข้อมูลที่มี emoji โดยได้ F1-score 0.923 การใช้ BiGRU แสดงประสิทธิภาพที่ดีกว่า BiLSTM ในการประมวลผล short text sequences

Chaisen et al. (2024) ได้นำเสนอ zero-shot interpretable sentiment analysis framework ที่ผสมผสาน sentiment polarity extraction กับ WangchanBERTa model เพื่อวิเคราะห์ความรู้สึกภาษาไทยโดยไม่ต้องการข้อมูลฝึกอบรมเฉพาะด้าน ผู้วิจัยพัฒนาระบบที่สามารถระบุและสกัด sentiment polarity จากข้อความได้อย่างอัตโนมัติ โดยใช้เทคนิค attention mechanism เพื่อเพิ่มความสามารถในการตีความผลลัพธ์ ชุดข้อมูลที่ใช้ทดสอบประกอบด้วย Thai social media posts จำนวน 12,500 รายการจากแพลตฟอร์มต่างๆ รวมถึง Twitter และ Facebook การประเมินผลใช้ accuracy, F1-score และ interpretability metrics โดยเปรียบเทียบกับ baseline models ผลการทดลองแสดงให้เห็นว่า framework ที่นำเสนอสามารถให้ accuracy ที่ 82.3% และ macro F1-score ที่ 0.791 โดยไม่ต้องการ domain-specific training data ข้อได้เปรียบสำคัญคือความสามารถในการอธิบายเหตุผลของการจำแนก sentiment ผ่าน attention visualization ซึ่งช่วยให้ผู้ใช้เข้าใจการทำงานของโมเดลได้ดีขึ้น

Boquio & Naval Jr. (2024) ได้ศึกษาการใช้ hybrid multi-layer pooled representations จาก BERT สำหรับงาน automated essay scoring โดยออกจากแนวทาง canonical fine-tuning แบบดั้งเดิม ผู้วิจัยเสนอการรวม representations จากหลายชั้นของ BERT ผ่านเทคนิค hybrid pooling strategy ที่ผสมผสานระหว่าง weighted averaging และ attention-based aggregation เพื่อจับข้อมูลจากระดับ linguistic abstraction ที่แตกต่างกัน ชุดข้อมูลที่ใช้คือ ASAP Automated Essay Scoring dataset ซึ่งประกอบด้วยเรียงความของนักเรียนจำนวน 12,978 ฉบับ แบ่งออกเป็น 8 prompt sets การประเมินผลใช้ Quadratic Weighted Kappa (QWK) score เป็นหลัก ร่วมกับ

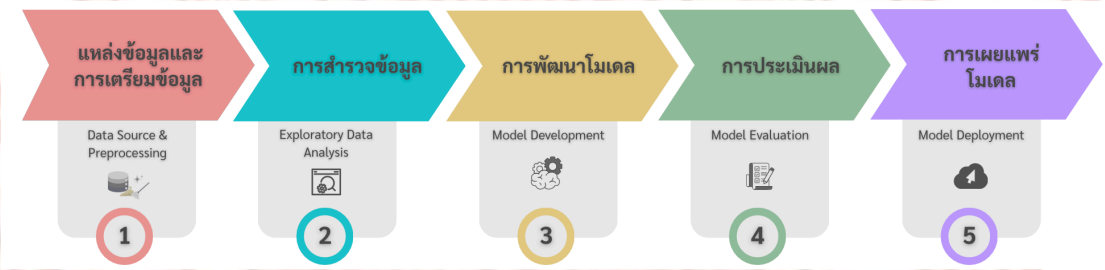
Pearson correlation และ Spearman correlation ผลการทดลองพบว่า hybrid multi-layer pooling ให้ผลลัพธ์ที่ดีกว่า standard BERT fine-tuning โดยได้ QWK score เฉลี่ย 0.764 เปรียบเทียบกับ 0.731 ของ baseline BERT การใช้ weighted combination ของ layers 6-12 แสดงประสิทธิภาพที่ดีที่สุด โดยเฉพาะสำหรับ essays ที่มีความยาวและความซับซ้อนสูง

บทที่ 3
วิธีดำเนินการวิจัย

3.1 ภาพรวมของการวิจัย (Overview of Research Process)

การวิจัยนี้เป็นการวิจัยเชิงประยุกต์ (Applied Research) และการวิจัยเชิงทดลอง (Experimental Research) ที่มุ่งเน้นการพัฒนาโมเดลการเรียนรู้เชิงลึก (Deep Learning) สำหรับการวิเคราะห์ความรู้สึกในข้อความภาษาไทย โดยประยุกต์ใช้เทคนิค Bidirectional LSTM ร่วมกับ WangchanBERTa เพื่อเพิ่มความสามารถในการจับความสัมพันธ์และบริบทของข้อความ งานวิจัยใช้แนวทางการเรียนรู้แบบมีการดูแล (Supervised Learning) ในการจำแนกประเภทความรู้สึกเป็น 2 ระดับ คือ ความรู้สึกเชิงบวก (Positive Sentiment) และความรู้สึกเชิงลบ (Negative Sentiment)

การดำเนินการวิจัยครั้งนี้อาศัยกรอบแนวคิดการพัฒนาโมเดลแบบเปรียบเทียบ (Comparative Model Development) โดยใช้วิธีการวิจัยเชิงปริมาณ (Quantitative Research Approach) ในการวัดและเปรียบเทียบประสิทธิภาพของโมเดลต่างๆ ผ่านตัวชี้วัดทางสถิติที่เป็นมาตรฐานในงานวิจัยด้านการประมวลผลภาษาธรรมชาติ



ภาพที่ 3-1 กรอบแนวคิดของงานวิจัย

จากภาพที่ 3-1 แสดงการดำเนินการวิจัยโดยแบ่งออกเป็น 5 ขั้นตอนหลัก ได้แก่ (1) แหล่งข้อมูลและการเตรียมข้อมูล (Data Source & Preprocessing) (2) การสำรวจข้อมูล (Exploratory Data Analysis) (3) การพัฒนาโมเดล (Model Development) (4) การประเมินผล (Model Evaluation) และ (5) การเผยแพร่โมเดล (Model Deployment) โดยแต่ละขั้นตอนมีความเชื่อมโยงและส่งผลต่อกันอย่างต่อเนื่อง

งานวิจัยนี้พัฒนาและทดสอบสถาปัตยกรรมโมเดลที่แตกต่างกัน 4 แบบ บนชุดข้อมูลเดียวกัน และใช้เทคนิค 5-Fold Stratified Cross-validation เพื่อให้ได้ผลการประเมินที่เชื่อถือได้และสามารถเปรียบเทียบได้อย่างเป็นธรรม การเปรียบเทียบสถาปัตยกรรมต่างๆ มีวัตถุประสงค์เพื่อหาโมเดลที่เหมาะสมที่สุดสำหรับการจำแนกความรู้สึกภาษาไทย โดยเฉพาะการศึกษาประสิทธิภาพของการประยุกต์ใช้ BiLSTM และสถาปัตยกรรมอื่นๆ ในการปรับปรุงความสามารถของโมเดล

นอกจากการพัฒนาโมเดลแล้ว การวิจัยครั้งนี้ยังรวมถึงการทดลองการประยุกต์ใช้งานจริง (Practical Application Testing) โดยนำโมเดลที่ได้ผลดีที่สุดมาพัฒนาเป็นระบบสาธิตที่สามารถรับข้อความและให้ผลการวิเคราะห์ความรู้สึกได้ การทดลองนี้มีจุดประสงค์เพื่อประเมินความเป็นไปได้ในการนำโมเดลไปใช้งานจริง และเผยแพร่สู่ชุมชนนักพัฒนาผ่าน Hugging Face Platform เพื่อให้สามารถเข้าถึงและนำไปประยุกต์ใช้ได้โดยไม่เสียค่าใช้จ่าย

3.2 แหล่งข้อมูลและการเตรียมข้อมูล (Data Source & Preprocessing)

3.2.1 แหล่งข้อมูล (Data Source)

การวิจัยครั้งนี้ใช้ชุดข้อมูล Wisesight Sentiment Corpus ซึ่งเป็นชุดข้อมูลความรู้สึกภาษาไทยที่ได้รับการพัฒนาโดยบริษัท Wisesight (Thailand) Co., Ltd. ร่วมกับชุมชนนักวิจัยด้านการประมวลผลภาษาธรรมชาติ ชุดข้อมูลนี้ประกอบด้วยข้อความภาษาไทยจากสื่อสังคมออนไลน์ต่างๆ รวมทั้งสิ้น 26,737 ข้อความ โดยแต่ละข้อความได้รับการจำแนกความรู้สึกเป็น 4 ประเภท ได้แก่ เชิงบวก (positive), เชิงลบ (negative), เป็นกลาง (neutral), และคำถาม (question)

สำหรับการวิจัยครั้งนี้ ผู้วิจัยได้คัดเลือกเฉพาะข้อความที่มีป้ายกำกับเป็นเชิงบวกและเชิงลบเท่านั้น เพื่อให้สอดคล้องกับวัตถุประสงค์การวิจัยที่มุ่งเน้นการจำแนกความรู้สึกแบบ 2 ระดับ ส่งผลให้ได้ชุดข้อมูลทั้งสิ้น 11,601 ข้อความ แบ่งเป็นข้อความเชิงบวก 4,778 ข้อความ (41.2%) และข้อความเชิงลบ 6,823 ข้อความ (58.8%)

เหตุผลในการเลือกใช้ชุดข้อมูล Wisesight Sentiment Corpus มีหลายประการ ประการแรก ชุดข้อมูลนี้มีขนาดใหญ่พอเพียงสำหรับการฝึกอบรมโมเดลการเรียนรู้เชิงลึก ประการที่สอง ข้อมูลได้รับการตรวจสอบและจำแนกป้ายกำกับโดยผู้เชี่ยวชาญ ทำให้มีความน่าเชื่อถือสูง ประการที่สาม ข้อความในชุดข้อมูลมาจากการใช้งานจริงในสื่อสังคมออนไลน์ จึงสะท้อนลักษณะการใช้ภาษาไทยได้เป็นอย่างดี รวมถึงการใช้คำสแลง คำย่อ และการผสมผสานภาษาที่หลากหลาย

3.2.2 การเตรียมข้อมูล (Data Preprocessing)

กระบวนการเตรียมข้อมูลเป็นขั้นตอนสำคัญที่ส่งผลต่อคุณภาพของโมเดลที่พัฒนาขึ้น การวิจัยครั้งนี้ดำเนินการเตรียมข้อมูลผ่านขั้นตอนต่อไปนี้

การกรองข้อมูล (Data Filtering) เริ่มต้นด้วยการคัดเลือกเฉพาะข้อความที่มีป้ายกำกับเป็น "pos" (positive) และ "neg" (negative) จากชุดข้อมูลต้นฉบับ 26,737 รายการ โดยตัดข้อความที่มีป้ายกำกับเป็น "neu" (neutral) และ "q" (question) ออกไป ได้ข้อมูล 11,601 รายการ ขั้นตอนนี้ช่วยให้ปัญหาการจำแนกมีความชัดเจนและเหมาะสมกับวัตถุประสงค์การวิจัยที่ต้องการสร้างโมเดลแบบ binary classification

การทำความสะอาดข้อความ (Text Cleaning) ดำเนินการลบองค์ประกอบที่ไม่จำเป็นออกจากข้อความ เช่น URL, อีเมล, เครื่องหมายวรรคตอนที่ซ้ำซ้อน, และตัวอักษรพิเศษที่ไม่มีความหมาย ขั้นตอนนี้ช่วยลดสัญญาณรบกวน (noise) ในข้อมูลและทำให้โมเดลสามารถโฟกัสไปที่เนื้อหาสำคัญของข้อความได้มากขึ้น

การประมวลผลภาษาไทย (Thai Language Processing) เนื่องจากภาษาไทยมีลักษณะเฉพาะที่แตกต่างจากภาษาอังกฤษ เช่น การไม่มีการเว้นวรรคระหว่างคำ การมีอักษรพิเศษ และการใช้วรรณยุกต์ ผู้วิจัยจึงใช้เครื่องมือประมวลผลภาษาไทยที่เหมาะสม รวมถึงการจัดการกับการเข้ารหัสอักขระ (character encoding) ให้เป็น UTF-8 เพื่อรองรับอักษรไทยอย่างถูกต้อง

การตรวจสอบคุณภาพข้อมูล (Data Quality Assessment) ทำการตรวจสอบข้อความที่ว่างเปล่า ข้อความที่มีเฉพาะตัวอักษรพิเศษ และข้อความที่มีคุณภาพต่ำ ข้อมูลที่ไม่ผ่านเกณฑ์คุณภาพจะถูกตัดออกจากชุดข้อมูล หลังจากกระบวนการทำความสะอาดและตรวจสอบคุณภาพเสร็จสิ้น ชุดข้อมูลสุดท้ายประกอบด้วย 11,118 ข้อความ แบ่งเป็นข้อความเชิงบวก 4,481 รายการ (40.3%) และข้อความเชิงลบ 6,637 รายการ (59.7%)

3.2.3 การแบ่งข้อมูลและ Cross-Validation

การแบ่งข้อมูลและการประเมินผลเป็นขั้นตอนที่มีความสำคัญต่อความน่าเชื่อถือของผลการวิจัย การวิจัยนี้ใช้เทคนิค 5-Fold Stratified Cross-Validation กับชุดข้อมูลทั้งหมด 11,118 รายการ เพื่อให้ได้การประเมินประสิทธิภาพที่เชื่อถือได้และลดอคติจากการแบ่งข้อมูล

การทำงานของ 5-Fold Cross-Validation

ชุดข้อมูลทั้งหมด 11,118 รายการจะถูกแบ่งออกเป็น 5 ส่วนเท่าๆ กัน (folds) โดยใช้วิธี Stratified Sampling เพื่อรักษาสัดส่วนของป้ายกำกับในแต่ละ fold ให้ใกล้เคียงกับชุดข้อมูลต้นฉบับ ในแต่ละรอบการทดลอง จะใช้ 4 ส่วน (ประมาณ 8,894 รายการ หรือ 80%) สำหรับการฝึกอบรมโมเดล และอีก 1 ส่วน (ประมาณ 2,224 รายการ หรือ 20%) สำหรับการตรวจสอบและประเมินผล กระบวนการนี้จะทำซ้ำทั้งหมด 5 รอบ โดยในแต่ละรอบจะหมุนเวียนให้ทุก fold ได้เป็นชุดตรวจสอบครั้งละ 1 รอบ

การรักษาสัดส่วนข้อมูล (Stratified Sampling)

การใช้ Stratified Sampling ช่วยให้แต่ละ fold มีสัดส่วนของป้ายกำกับที่ใกล้เคียงกับชุดข้อมูลต้นฉบับ ดังนั้นในแต่ละ fold จะมีข้อความเชิงลบประมาณ 59.7% และข้อความเชิงบวกประมาณ 40.3% ซึ่งช่วยป้องกันปัญหาที่อาจเกิดขึ้นจากการกระจายข้อมูลที่ไม่สมดุล และทำให้การเปรียบเทียบประสิทธิภาพระหว่างโมเดลต่างๆ มีความเป็นธรรมมากขึ้น

การรายงานผลการประเมิน

ผลการประเมินจากทั้ง 5 folds จะถูกรายงานในรูปแบบค่าเฉลี่ย ± ส่วนเบี่ยงเบนมาตรฐาน (mean ± std) สำหรับทุกเมตริก (Accuracy, F1-Score, และ AUC) ซึ่งสะท้อนทั้งประสิทธิภาพโดยเฉลี่ยและความเสถียรของโมเดลในการทำงานกับข้อมูลชุดต่างๆ วิธีการนี้ช่วยให้การประเมินประสิทธิภาพของโมเดลมีความน่าเชื่อถือและสามารถเปรียบเทียบระหว่างสถาปัตยกรรมต่างๆ ได้อย่างเป็นธรรม

3.3 การสำรวจข้อมูล (Exploratory Data Analysis)

การสำรวจข้อมูลเป็นขั้นตอนสำคัญที่ช่วยให้ผู้วิจัยเข้าใจลักษณะและการกระจายของข้อมูล หลังจากผ่านกระบวนการทำความสะอาดแล้ว จากการวิเคราะห์ชุดข้อมูล Wisesight Sentiment ที่มีทั้งสิ้น 11,118 รายการ ได้ผลดังตารางที่ 3-1

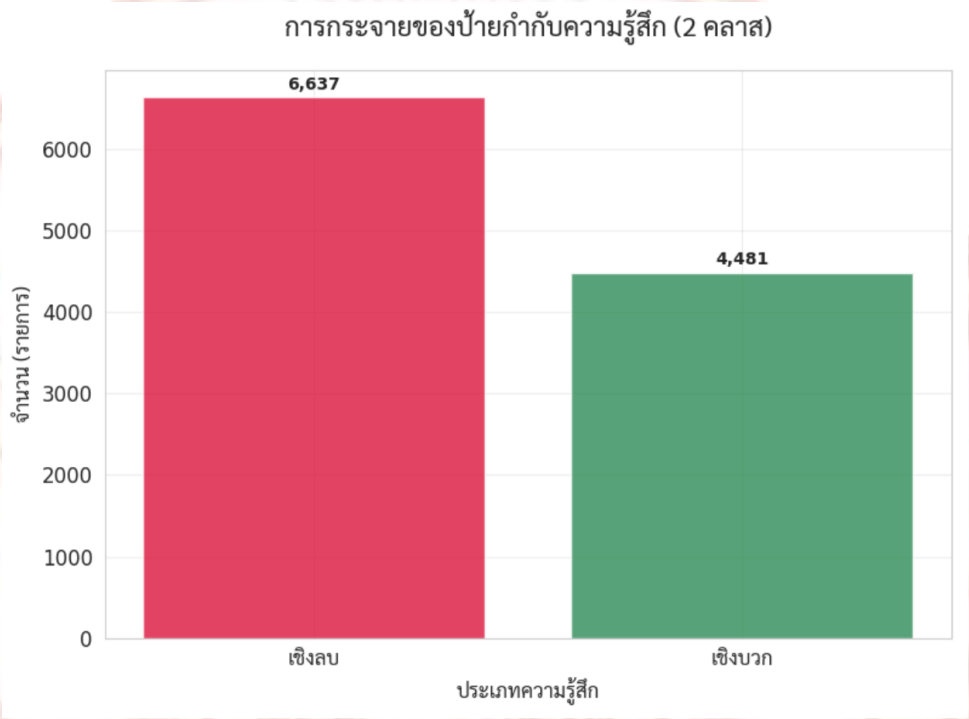
ตารางที่ 3-1 สถิติพื้นฐานของชุดข้อมูล Wisesight Sentiment

คุณลักษณะ	**ค่าสถิติ**
จำนวนตัวอักษรเฉลี่ย	70 ตัวอักษร
ข้อความยาวที่สุด	1,876 ตัวอักษร
ข้อความสั้นที่สุด	2 ตัวอักษร
จำนวนตัวอักษร (Quantile 99%)	414 ตัวอักษร
จำนวนคำเฉลี่ย	4 คำ
ข้อความที่มีคำมากที่สุด	124 คำ
ข้อความที่มีคำน้อยที่สุด	1 คำ
จำนวนคำ (Quantile 99%)	21 คำ

จากตารางที่ 3-1 แสดงสถิติพื้นฐานของชุดข้อมูล พบว่าข้อความมีความยาวเฉลี่ย 70 ตัวอักษร หรือประมาณ 4 คำต่อข้อความ โดยมีช่วงความยาวที่หลากหลายตั้งแต่ 2 ตัวอักษร ไปจนถึง 1,876 ตัวอักษร ซึ่งสะท้อนถึงความหลากหลายของข้อมูลจากสื่อสังคมออนไลน์ การวิเคราะห์ Quantile 99% ซึ่งแสดงว่า 99% ของข้อความมีความยาวไม่เกิน 414 ตัวอักษร หรือ 21 คำ เป็นข้อมูลสำคัญในการกำหนดความยาวสูงสุดของ input sequence สำหรับโมเดล

การวิเคราะห์การกระจายป้ายกำกับ ภาพที่ 3-2 แสดงการกระจายของป้ายกำกับความรู้สึกในชุดข้อมูลทั้งหมด 11,118 รายการ โดยใช้กราฟแท่งเปรียบเทียบจำนวนข้อความระหว่างสองประเภท จากกราฟพบว่าข้อมูลมีการกระจายที่ไม่สมดุลเล็กน้อย โดยข้อความเชิงลบ (สีชมพู) มีจำนวน 6,637

รายการ คิดเป็น 59.7% ของข้อมูลทั้งหมด ขณะที่ข้อความเชิงบวก (สีเขียว) มีจำนวน 4,481 รายการ หรือ 40.3% อัตราส่วนของข้อความเชิงบวกต่อเชิงลบอยู่ที่ 0.675 ซึ่งถือว่าข้อมูลยังอยู่ในระดับที่จัดการได้และไม่รุนแรงเกินไปที่จะส่งผลกระทบต่อการฝึกอบรมโมเดล อย่างไรก็ตาม เพื่อให้โมเดลสามารถเรียนรู้และทำนายทั้งสองคลาสได้อย่างเท่าเทียมกัน จำเป็นต้องใช้เทคนิค Class Weighting ในระหว่างการฝึกอบรม

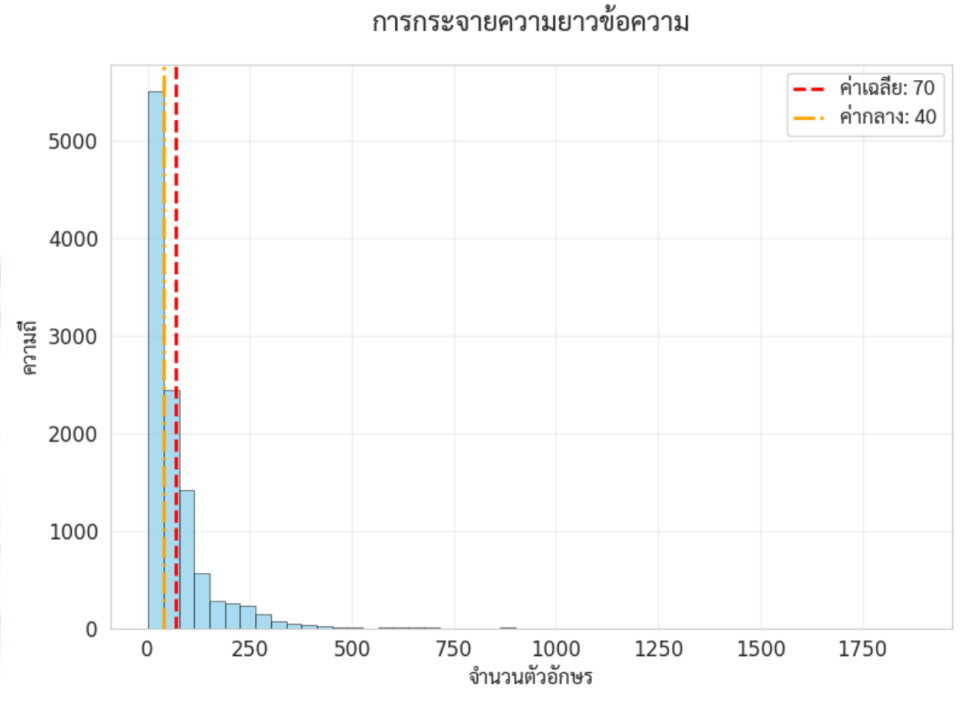


ภาพที่ 3-2 การกระจายของป้ายกำกับความรู้สึก

การวิเคราะห์ความยาวข้อความ ภาพที่ 3-3 แสดงการกระจายความยาวข้อความทั้งหมดในหน่วยตัวอักษร โดยใช้ Histogram เพื่อแสดงความถี่ของข้อความในแต่ละช่วงความยาว จากกราฟพบว่าข้อมูลมีการกระจายแบบ Right-skewed คือข้อความส่วนใหญ่มีความยาวสั้น โดยเฉพาะในช่วง 0-100 ตัวอักษร ซึ่งมีความถี่สูงสุดมากกว่า 5,000 รายการ

กราฟแสดงเส้นอ้างอิงสองเส้น ได้แก่ ค่าเฉลี่ย (เส้นประสีแดง) อยู่ที่ 70 ตัวอักษร และค่ากลาง median (เส้นประสีน้ำเงิน) อยู่ที่ 40 ตัวอักษร โดยที่ค่า median ต่ำกว่าค่าเฉลี่ยอย่างชัดเจน ยืนยันว่าข้อมูลมีการเบ้ไปทางขวา เนื่องจากมีข้อความยาวบางส่วนที่ดึงค่าเฉลี่ยให้สูงขึ้น ขณะที่ข้อความส่วนใหญ่มีความยาวไม่เกิน 250 ตัวอักษร ข้อมูลนี้สอดคล้องกับตารางที่ 3-1 ที่แสดงว่า 99% ของข้อความมีความยาวไม่เกิน 414 ตัวอักษร

จากการวิเคราะห์นี้ ผู้วิจัยจึงเลือกใช้ความยาว maximum sequence length ที่ 128 tokens ซึ่งครอบคลุมข้อความส่วนใหญ่และช่วยลดการใช้หน่วยความจำในการฝึกอบรมโมเดล



ภาพที่ 3-3 การกระจายความยาวข้อความ (หน่วย: ตัวอักษร)

การเปรียบเทียบความยาวข้อความระหว่างประเภทความรู้สึก ภาพที่ 3-4 แสดงการเปรียบเทียบการกระจายความยาวข้อความระหว่างสองประเภทความรู้สึกโดยใช้ Box Plot ซึ่งเป็นเครื่องมือที่มีประสิทธิภาพในการแสดงการกระจายและค่าสถิติสำคัญของข้อมูล

จากกราฟพบความแตกต่างที่น่าสนใจระหว่างสองกลุ่ม:

ข้อความเชิงลบ (สีชมพู):

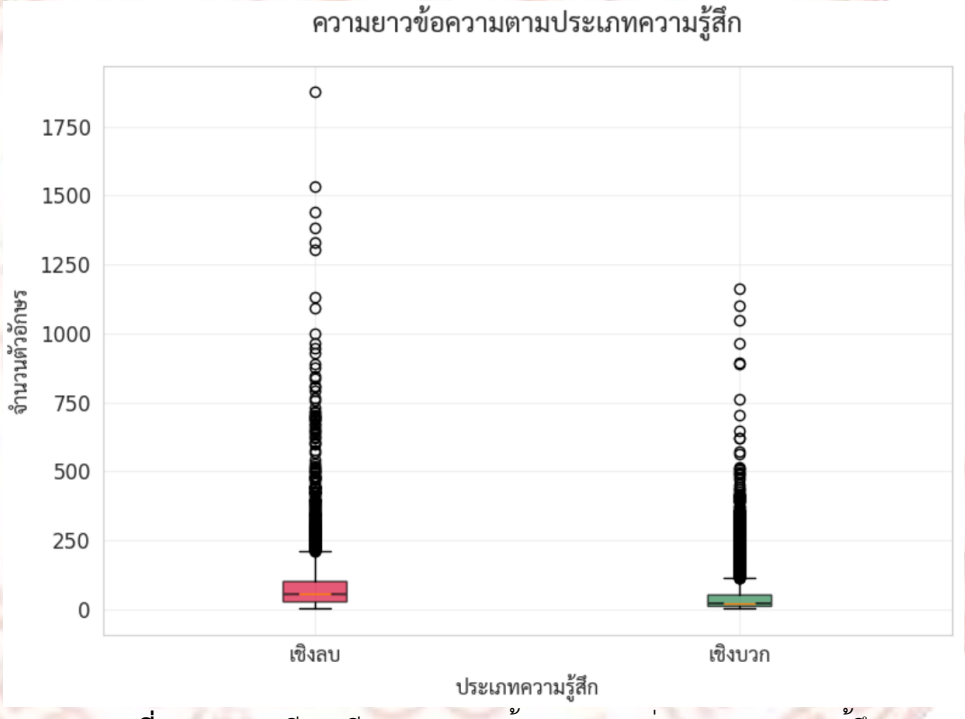
- มีค่า median (เส้นตรงกลางกล่อง) อยู่ที่ประมาณ 50-60 ตัวอักษร
- กล่อง (box) ที่แสดง Interquartile Range (IQR) มีความกว้างมากกว่า แสดงถึงความหลากหลายของข้อมูล
- มีค่าเฉลี่ย 82 ตัวอักษร และส่วนเบี่ยงเบนมาตรฐาน ±98 ตัวอักษร
- มี outliers (จุดวงกลมด้านบน) จำนวนมากที่กระจายตัวกว้าง ตั้งแต่ 500 ตัวอักษรจนถึงเกือบ 2,000 ตัวอักษร

ข้อความเชิงบวก (สีเขียว):

- มีค่า median ต่ำกว่า อยู่ที่ประมาณ 30-40 ตัวอักษร

- กล่อง (box) แคบกว่า แสดงว่าข้อมูลส่วนใหญ่มีความยาวใกล้เคียงกันมากขึ้น
- มีค่าเฉลี่ย 52 ตัวอักษร และส่วนเบี่ยงเบนมาตรฐาน ±85 ตัวอักษร
- มี outliers น้อยกว่าและกระจายตัวไม่กว้างเท่า โดยส่วนใหญ่อยู่ในช่วง 500-1,200 ตัวอักษร

การวิเคราะห์นี้แสดงให้เห็นว่าข้อความเชิงลบโดยเฉลี่ยยาวกว่าข้อความเชิงบวกประมาณ 1.6 เท่า (82 vs 52 ตัวอักษร) และมีความหลากหลายในความยาวมากกว่า ซึ่งอาจเป็นเพราะผู้ใช้มักต้องการอธิบายปัญหาหรือความไม่พอใจอย่างละเอียด ขณะที่การแสดงความรู้สึกเชิงบวกมักใช้ข้อความสั้นๆ กระชับ เช่น "ดีมาก" "ชอบ" หรือ "เยี่ยม"

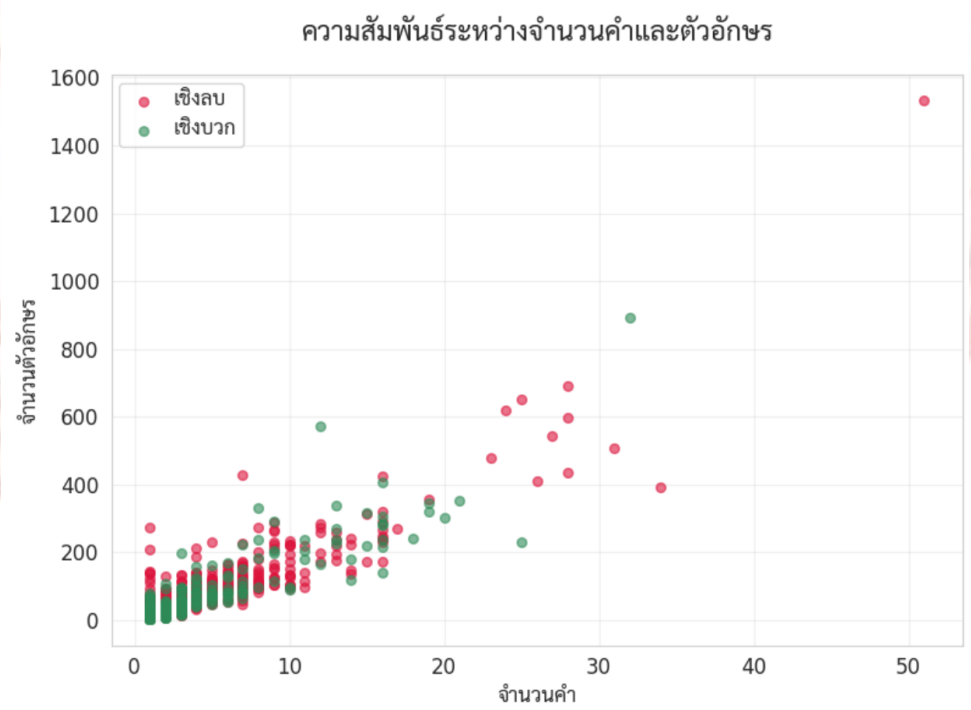


ภาพที่ 3-4 การเปรียบเทียบความยาวข้อความระหว่างประเภทความรู้สึก

การวิเคราะห์ความสัมพันธ์ระหว่างจำนวนคำและจำนวนตัวอักษร ภาพที่ 3-5 แสดงความสัมพันธ์ระหว่างจำนวนคำ (แกน X) และจำนวนตัวอักษร (แกน Y) ในข้อความโดยใช้ Scatter Plot โดยแยกสีตามประเภทความรู้สึก คือข้อความเชิงลบ (สีชมพู) และข้อความเชิงบวก (สีเขียว) จากกราฟพบว่า

1. ความสัมพันธ์เชิงบวกที่ชัดเจน มีความสัมพันธ์เชิงบวกแบบเส้นตรง (Linear positive correlation) ระหว่างจำนวนคำและจำนวนตัวอักษร ซึ่งเป็นไปตามธรรมชาติว่าข้อความที่มีคำมากย่อมมีตัวอักษรมากตามไปด้วย

2. การกระจายของข้อมูล ข้อความส่วนใหญ่มีจำนวนคำไม่เกิน 20 คำ และความยาวไม่เกิน 400 ตัวอักษร โดยมีความหนาแน่นสูงสุดในช่วง 0-10 คำ และ 0-200 ตัวอักษร ซึ่งสอดคล้องกับค่าเฉลี่ยที่แสดงในตารางที่ 3-1 (4 คำ และ 70 ตัวอักษร)
3. รูปแบบการกระจายที่คล้ายคลึงกัน ข้อความทั้งสองประเภทมีรูปแบบการกระจายและความสัมพันธ์ระหว่างจำนวนคำกับตัวอักษรที่ใกล้เคียงกัน แสดงว่าไม่มีความแตกต่างในเรื่องโครงสร้างพื้นฐานของข้อความ
4. Outliers พบข้อความบางส่วนที่มีความยาวมากกว่าปกติอย่างชัดเจน โดยเฉพาะข้อความเชิงลบที่มีจุดกระจายไปถึง 50 คำ และ 1,500 ตัวอักษร ขณะที่ข้อความเชิงบวกมี outliers น้อยกว่า ซึ่งสอดคล้องกับการวิเคราะห์ในภาพที่ 3-4
5. ความหลากหลายของข้อความเชิงลบ แม้ว่ารูปแบบโดยรวมจะคล้ายกัน แต่จะเห็นว่าจุดสีชมพู (เชิงลบ) มีการกระจายที่กว้างกว่าเล็กน้อย โดยเฉพาะในช่วงข้อความที่ยาว ซึ่งสอดคล้องกับข้อมูลที่แสดงว่าข้อความเชิงลบมีความยาวเฉลี่ยและความหลากหลายมากกว่า



ภาพที่ 3-5 ความสัมพันธ์ระหว่างจำนวนคำและจำนวนตัวอักษร

การวิเคราะห์คุณภาพข้อมูล ไม่พบข้อความที่ว่างเปล่าหรือมีเฉพาะตัวอักษรพิเศษ และข้อความทุกรายการมีป้ายกำกับที่ชัดเจน แสดงว่าชุดข้อมูลมีคุณภาพดีและพร้อมสำหรับการนำไปใช้ในการฝึกอบรมโมเดล

ผลการวิเคราะห์ข้อมูลเชิงสำรวจให้ข้อมูลเชิงลึกที่สำคัญต่อการออกแบบและพัฒนาโมเดล ดังนี้

1. การออกแบบ Input Sequence จากการที่ 99% ของข้อมูลมีความยาวไม่เกิน 414 ตัวอักษร หรือ 21 คำ ผู้วิจัยจึงเลือกใช้ maximum sequence length ที่ 128 tokens ซึ่งเพียงพอสำหรับข้อความส่วนใหญ่และช่วยประหยัดหน่วยความจำในการฝึกอบรม
2. การจัดการ Class Imbalance ด้วยอัตราส่วน 60:40 ระหว่างข้อความเชิงลบและเชิงบวก จำเป็นต้องใช้เทคนิค Class Weighting เพื่อให้โมเดลให้ความสำคัญกับทั้งสองคลาสอย่างเหมาะสม
3. ความเข้าใจพฤติกรรมผู้ใช้ การที่ข้อความเชิงลบยาวกว่าและมีความหลากหลายมากกว่าข้อความเชิงบวก ช่วยให้เข้าใจพฤติกรรมการใช้ภาษาของผู้ใช้ ซึ่งอาจมีประโยชน์ในการตีความผลลัพธ์และการปรับปรุงโมเดลในอนาคต

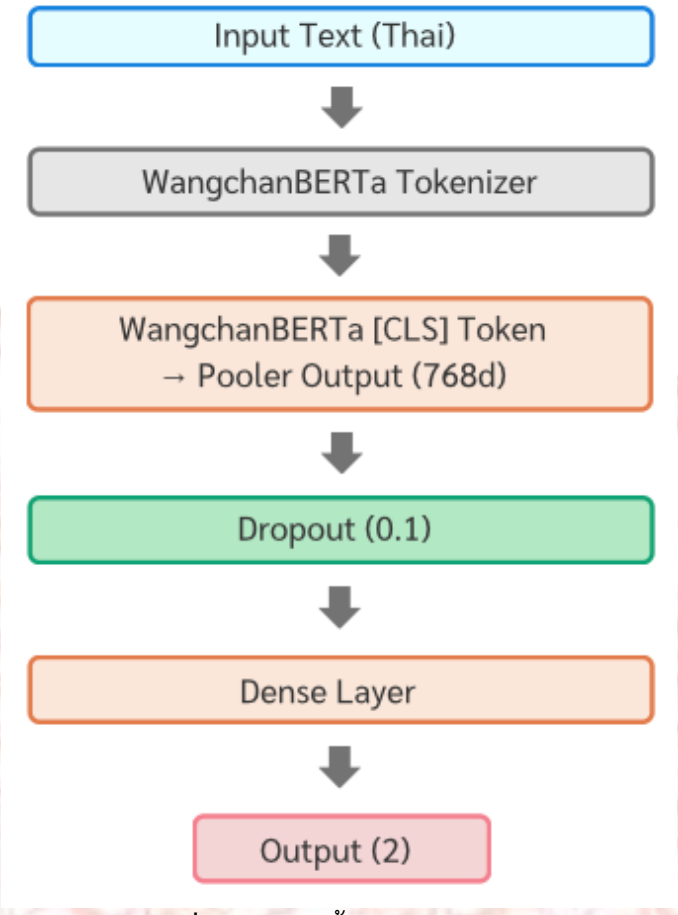
3.4 การพัฒนาโมเดล (Model Development)

การวิจัยนี้พัฒนาโมเดลทั้งหมด 4 แบบ โดยใช้ WangchanBERTa เป็นตัวสร้าง feature หลัก โมเดลเหล่านี้ออกแบบมาเพื่อศึกษาประสิทธิภาพของการประยุกต์ใช้เทคนิค NLP ร่วมกับ LSTM และสถาปัตยกรรมอื่นๆ ในการจำแนกความรู้สึกภาษาไทย

เป้าหมายหลักคือการพัฒนาโมเดลที่มีประสิทธิภาพสูงและเหมาะสมสำหรับการใช้งานจริง โดยเริ่มจาก WCB เป็นโมเดลพื้นฐาน และศึกษาผลของการเพิ่ม BiLSTM เป็น WCB + BiLSTM ไปจนถึงโมเดลที่ผสม CNN อย่าง WCB + CNN + BiLSTM และโมเดลที่ใช้เทคนิคการรวมข้อมูลจาก 4 ชั้นสุดท้าย (layers) ของโมเดล Transformer บน WCB (4-Layers) + BiLSTM เพื่อประเมินว่าสถาปัตยกรรมแบบใดเหมาะสมที่สุดสำหรับการจำแนกความรู้สึกในภาษาไทย

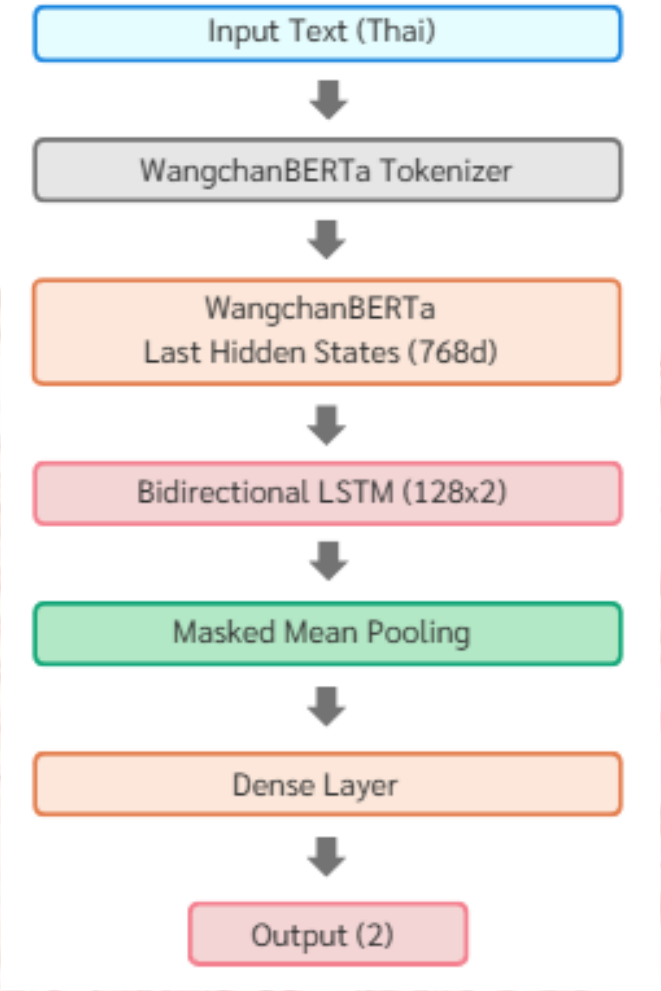
3.4.1 สถาปัตยกรรมของโมเดล

WCB โมเดลพื้นฐานที่แท้จริงใช้เพียง WangchanBERTa เท่านั้น โดยใช้ [CLS] token representation ที่ผ่านการประมวลผล pooling layer ของ BERT แล้ว (pooler_output) ซึ่งมีขนาด 768 มิติ จากนั้นผ่าน dropout layer (อัตรา 0.1) และ dense layer เพื่อจำแนกเป็น 2 คลาส โดยใช้ CrossEntropyLoss โมเดลนี้เป็น true baseline ที่แสดงประสิทธิภาพของ WangchanBERTa เพียงอย่างเดียว แสดงดังภาพที่ 3-6



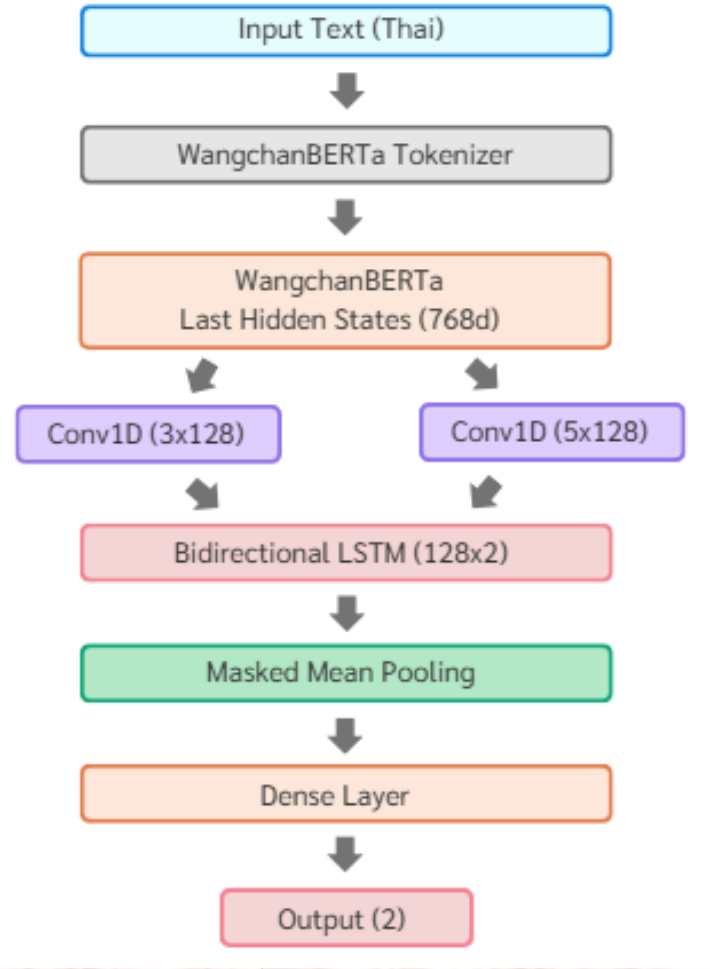
ภาพที่ 3-6 โครงสร้างของโมเดล WCB

WCB + BiLSTM โมเดลที่เพิ่มชั้น sequence modeling โดยใช้โครงสร้างแบบ sequence-to-sequence รับ last hidden states จาก WangchanBERTa (768 มิติ) ป้อนเข้า Bidirectional LSTM ขนาด 128 หน่วย (ใช้ tanh activation) ซึ่งประมวลผลข้อมูลทั้งสองทิศทาง จากนั้นทำ masked mean pooling เพื่อรวมข้อมูลจากทุกตำแหน่งโดยคำนึงถึง attention mask และผ่าน dense layer พร้อมใช้ CrossEntropyLoss เพื่อจำแนกเป็น 2 คลาส โมเดลนี้ช่วยเพิ่มความสามารถในการจับ long-term dependencies และความสัมพันธ์ของบริบทในข้อความ แสดงดังภาพที่ 3-7



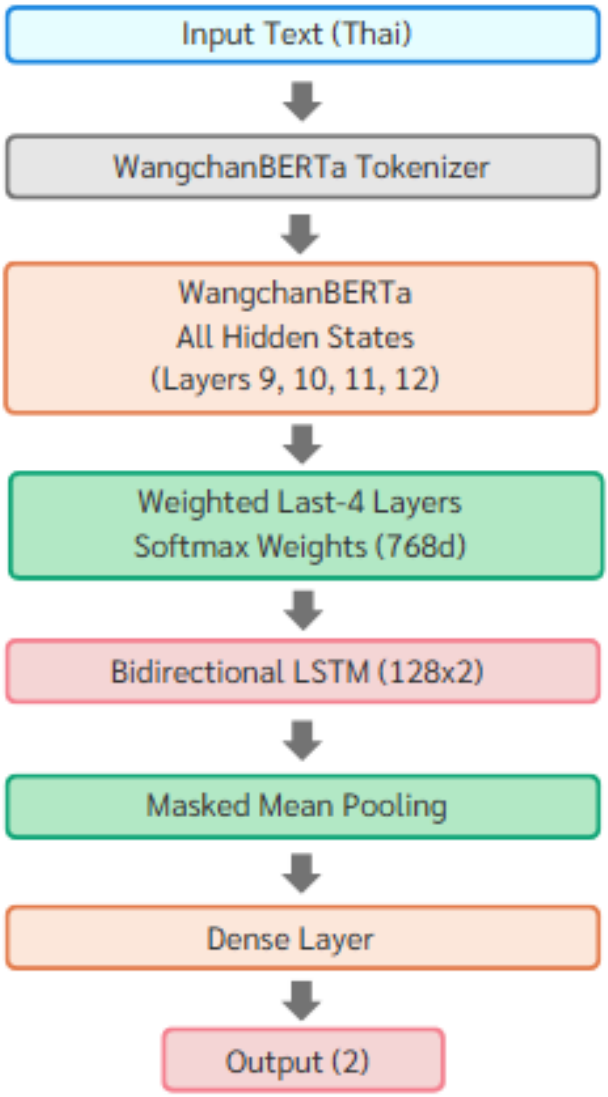
ภาพที่ 3-7 โครงสร้างของโมเดล WCB+BiLSTM

WCB + CNN + BiLSTM โมเดลนี้เพิ่ม Convolutional layers ก่อน BiLSTM โดยใช้ Conv1D สองชั้นแบบคู่ขนาน ด้วย kernel size 3 และ 5 พร้อม ReLU activation เพื่อจับ local patterns ในระดับที่แตกต่างกัน โดย kernel 3 จับ short-range patterns และ kernel 5 จับ longer-range patterns แต่ละ convolution จะสร้าง feature maps 128 มิติ จากนั้นรวม features จากทั้งสองชั้นและป้อนเข้า BiLSTM ขนาด 128 หน่วย (ใช้ tanh activation) ซึ่งประมวลผลข้อมูลทั้งสองทิศทาง ผ่าน masked mean pooling เพื่อรวมข้อมูลจากทุกตำแหน่งโดยคำนึงถึง attention mask และผ่าน dense layer พร้อมใช้ CrossEntropyLoss เพื่อจำแนกเป็น 2 คลาส โมเดลนี้ผสมผสานจุดแข็งของ CNN ในการจับ local features และ BiLSTM ในการเรียนรู้ sequence dependencies แสดงดังภาพที่ 3-8



ภาพที่ 3-8 โครงสร้างของโมเดล WCB+CNN+BiLSTM

WCB (4-Layers) + BiLSTM โมเดลนี้ใช้ hidden states จาก 4 ชั้นสุดท้าย (layers 9-12) ของ WangchanBERTa โดยคำนวณ weighted combination ด้วย learnable parameters ที่ผ่าน softmax normalization เพื่อให้ได้น้ำหนักที่รวมเป็น 1 ทำให้สามารถเรียนรู้ว่าชั้นไหนมีความสำคัญมากกว่าสำหรับงานจำแนกความรู้สึก ผลลัพธ์ที่ได้จะมีขนาด 768 มิติเช่นเดิม แต่เป็นการรวมข้อมูลที่อุดมไปด้วยคุณลักษณะจากหลายระดับของ transformer layers แล้วป้อนตรงเข้า BiLSTM ขนาด 128 หน่วย (ใช้ tanh activation) ผ่าน masked mean pooling และ dense layer พร้อมใช้ CrossEntropyLoss เพื่อจำแนกเป็น 2 คลาส โดยไม่ผ่าน CNN เพื่อให้เห็นประสิทธิภาพของการใช้ multiple layers ของ BERT อย่างเดียว แสดงดังภาพที่ 3-9



ภาพที่ 3-9 โครงสร้างของโมเดล WCB(4-Layers)+BiLSTM

3.4.2 การตั้งค่าพารามิเตอร์

ในการฝึกอบรมโมเดล ได้แบ่งพารามิเตอร์ออกเป็น 2 กลุ่ม ได้แก่ พารามิเตอร์กลางที่ใช้เหมือนกันในทุกโมเดล และพารามิเตอร์เฉพาะที่ขึ้นอยู่กับโครงสร้างของโมเดล

ตารางที่ 3-2 พารามิเตอร์กลาง (Global Hyperparameters)

พารามิเตอร์	ค่า	คำอธิบาย
Epochs	30	จำนวนรอบการฝึกอบรม
Batch Size	16	จำนวนตัวอย่างในแต่ละ batch
Max Sequence Length (MAX_LEN)	128	ความยาวสูงสุดของ sequence ต่อ review
Learning Rate – BERT (LR_BERT**)**	2e-5	Learning rate สำหรับชั้น BERT
Learning Rate – Others (LR_OTHERS**)**	1e-3	Learning rate สำหรับชั้นอื่น เช่น BiLSTM, CNN
Optimizer	AdamW	Optimizer ที่ใช้ในการฝึก
Loss Function	CrossEntropyLoss (with class weights)	ฟังก์ชัน loss สำหรับ class imbalance
BiLSTM Activation	tanh	default activation function สำหรับ LSTM hidden และ cell states
Patience (Early Stopping)	3	จำนวนรอบที่รอหาก F1 ไม่ดีขึ้น
Cross Validation Folds (N_FOLDS**)**	5	จำนวน fold ใน Stratified K-Fold
Pooling Method หลัง BiLSTM	masked_mean	วิธี pool output หลัง BiLSTM
Test Size (TEST_SIZE**)**	0.2 (20%)	สัดส่วนของ test set
Gradient Clipping	1.0 (max_norm)	ป้องกัน exploding gradients
Mixed Precision (AMP)	Enabled	เพิ่มประสิทธิภาพ GPU และลด memory usage
Class Weights	Calculated from training data	จัดการ class imbalance
Dropout Rate	0.3	สำหรับ classification layer
Random Seed	42	เพื่อ reproducibility

ตารางที่ 3-3 พารามิเตอร์เฉพาะของแต่ละโมเดล (Model-Specific Hyperparameters)

โมเดล	ค่าเฉพาะ (Hyperparameters เฉพาะโมเดล)	คำอธิบาย
WCB	- BERT pooler_output ขนาด 768 - Dropout = 0.1 - No additional layers	ใช้เพียง WangchanBERTa และ [CLS] token
WCB+BiLSTM	- Hidden size ของ BiLSTM = 128 - Dropout = 0.3	ใช้ WangchanBERTa เป็นตัวสร้าง embedding และส่งเข้า BiLSTM โดยตรง เป็น baseline
WCB+CNN+BiLSTM	- CNN Layer 1: filters = 128, kernel size = 3 - CNN Layer 2: filters = 128, kernel size = 5 - Activation: ReLU (CNN)	เพิ่ม CNN 2 ชั้นเพื่อดึง feature จาก sequence ก่อนส่งเข้า BiLSTM
WCB(4-Layers)+BiLSTM	- Pooling จาก Last 4 Hidden Layers (Layer 9-12) - รวมค่าโดยใช้ Weighted Softmax - Learnable parameters: 4 weights	เน้นการดึงข้อมูลเชิงลึกจาก BERT โดยใช้ hidden states 4 ชั้นสุดท้ายแล้วถ่วงน้ำหนัก

3.5 การประเมินผล (Model Evaluation)

3.5.1 ตัวชี้วัดการประเมินผล (Evaluation Metrics)

งานวิจัยนี้ใช้การประเมินประสิทธิภาพของโมเดลด้วย 3 เมทริกซ์หลัก ได้แก่ F1-Score, Accuracy, และ AUC (Area Under Curve) เพื่อสะท้อนประสิทธิภาพของโมเดลในมุมมองที่แตกต่างกัน

- F1-Score ใช้เป็นตัวชี้วัดหลัก เพราะสะท้อนความสมดุลระหว่าง Precision และ Recall ได้ดีกว่า Accuracy โดยเฉพาะเมื่อข้อมูลมีการกระจายไม่เท่ากัน
- Accuracy ใช้ดูสัดส่วนของการทำนายที่ถูกต้องทั้งหมด เหมาะสำหรับภาพรวมของความแม่นยำของโมเดล

- AUC (Receiver Operating Characteristic – Area Under Curve) ใช้วัดความสามารถของโมเดลในการแยกแยะคลาสบวกและลบโดยไม่ขึ้นกับ threshold ยิ่งค่าใกล้ 1 ยิ่งบ่งชี้ว่าโมเดลแยกคลาสได้ดี

เพื่อให้เข้าใจรูปแบบข้อผิดพลาด ยังใช้ Confusion Matrix ช่วยวิเคราะห์จุดที่โมเดลทำนายผิดบ่อย เพื่อปรับปรุงการทำงานในอนาคต

3.5.2 กลยุทธ์การประเมินด้วย Cross-Validation

การประเมินใช้วิธี 5-Fold Stratified Cross-Validation เพื่อให้ผลการประเมินมีความเสถียรและลดอคติจากการแบ่งข้อมูล โดยในแต่ละ fold จะแบ่งข้อมูลให้คงสัดส่วนของคลาสบวกและลบใกล้เคียงกับชุดข้อมูลต้นฉบับ

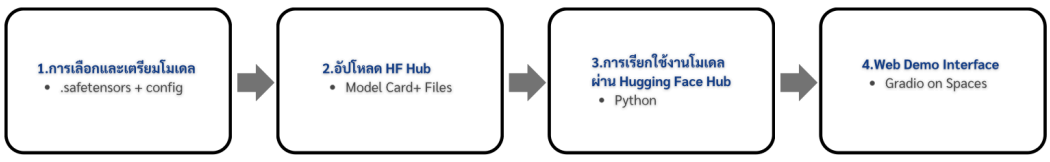
ในแต่ละรอบ โมเดลจะฝึกด้วยข้อมูล 4 ส่วน และทดสอบด้วยอีก 1 ส่วน ทำซ้ำทั้งหมด 5 รอบ จนทุกส่วนถูกใช้เป็นชุดทดสอบ ครั้งละ 1 รอบ จากนั้นรายงานค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของ F1-Score, Accuracy และ AUC เพื่อสะท้อนความเสถียรของโมเดล

3.5.3 การรายงานผลและการวิเคราะห์

ผลการประเมินจากทั้ง 5 folds ของแต่ละสถาปัตยกรรมจะถูกรายงานในรูปแบบ ค่าเฉลี่ย ± ส่วนเบี่ยงเบนมาตรฐาน (mean ± std) สำหรับทุกเมทริก นอกจากนี้ยังวิเคราะห์เปรียบเทียบเวลาที่ใช้ในการฝึกอบรมของแต่ละโมเดล เพื่อประเมินความเหมาะสมในการนำไปใช้งานจริงที่ต้องคำนึงถึงทั้งประสิทธิภาพและทรัพยากรที่ใช้ จากนั้นวิเคราะห์ผลโดยละเอียดด้วย Confusion Matrix เพื่อดูรูปแบบการจำแนกและข้อผิดพลาดที่เกิดขึ้น และใช้ AUC เพื่อประเมินความสามารถในการแยกแยะระหว่างสองคลาส การวิเคราะห์เหล่านี้ช่วยให้เข้าใจจุดแข็งและข้อจำกัดของแต่ละสถาปัตยกรรม นอกจากนี้ยังทดสอบโมเดลที่มีประสิทธิภาพสูงสุดกับข้อความตัวอย่างที่มีลักษณะแตกต่างกัน เพื่อประเมินความสามารถในการทำงานกับข้อความที่มีความซับซ้อนและบริบทที่หลากหลาย

3.6 การเผยแพร่โมเดล (Model Deployment)

งานวิจัยนี้ได้นำโมเดลที่มีประสิทธิภาพสูงสุดไปเผยแพร่ผ่าน Hugging Face Platform เพื่อให้ผู้สนใจสามารถเข้าถึงและทดลองใช้งานได้จริง โดยมีขั้นตอนการดำเนินงานตามแผนภาพ Pipeline ดังแสดงในภาพที่ 3-10



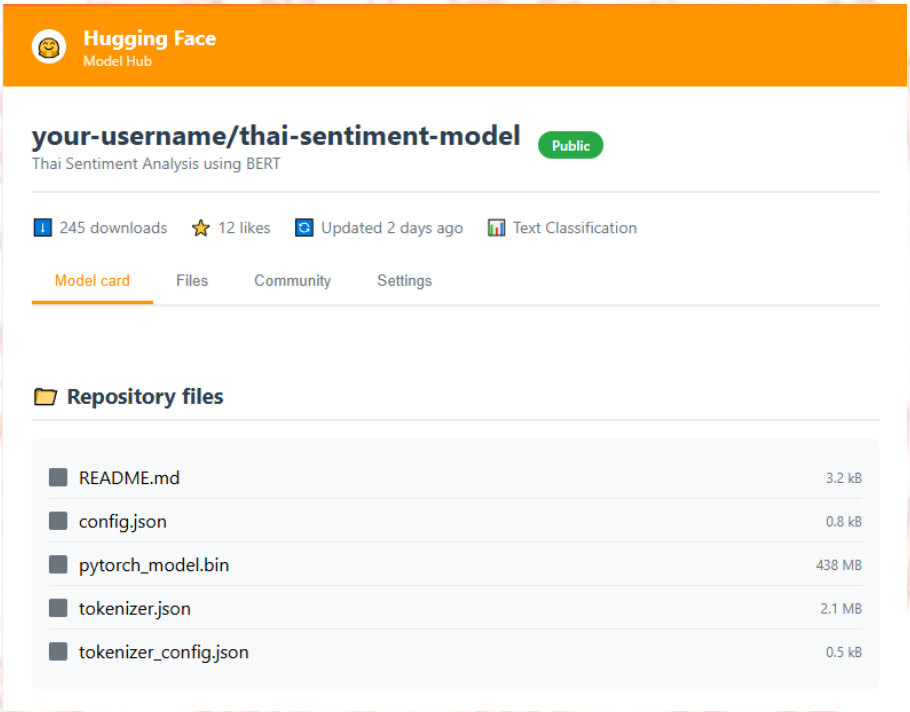
ภาพที่ 3-10 การเผยแพร่โมเดล

3.6.1 การเลือกและเตรียมโมเดล

จากผลการทดลองทั้งหมด เลือกโมเดลที่มี validation F1 score สูงสุดมาเตรียมสำหรับการเผยแพร่ โดยบันทึกโมเดลในรูปแบบ .safetensors พร้อมไฟล์ tokenizer configuration และ model configuration ที่จำเป็นสำหรับการใช้งาน

3.6.2 การอัปโหลดไปยัง Hugging Face Model Hub

อัปโหลดโมเดลและไฟล์ที่เกี่ยวข้องไปยัง Hugging Face Model Hub ในรูปแบบ Public Repository เพื่อให้ผู้ใช้งานสามารถเข้าถึงได้โดยไม่ต้องลงทะเบียน พร้อมจัดทำ Model Card ที่มีข้อมูลรายละเอียดโมเดล วิธีการใช้งาน และข้อจำกัด แสดงตัวอย่างดังภาพที่ 3-11



ภาพที่ 3-11 หน้าตัวอย่าง Model Card บน Hugging Face

3.6.3 การเรียกใช้งานโมเดลผ่าน Hugging Face Hub

เนื่องจากโมเดลได้รับการอัปโหลดไปยัง Hugging Face Model Hub แล้ว ผู้ใช้งานสามารถเรียกใช้โมเดลได้โดยตรงผ่าน Python ตามตัวอย่างดังภาพที่ 3-12

model_usage_example.py

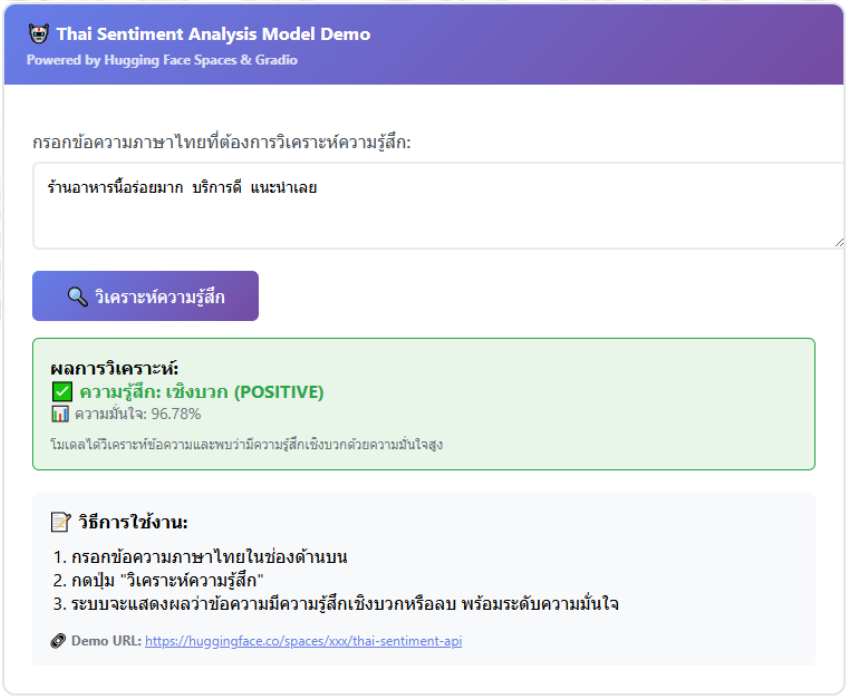
```
# การติดตั้ง libraries ที่จำเป็น from transformers import AutoTokenizer,
AutoModelForSequenceClassification from transformers import pipeline # โหลดโมเดลและ
tokenizer จาก Hugging Face Hub model_name = "your-username/thai-sentiment-model" tokenizer =
AutoTokenizer.from_pretrained(model_name) model =
AutoModelForSequenceClassification.from_pretrained(model_name) # สร้าง pipeline สำหรับ
sentiment analysis classifier = pipeline( "sentiment-analysis", model=model,
tokenizer=tokenizer ) # ทดสอบโมเดลด้วยข้อความภาษาไทย text_samples = [ "ภาพยนตร์เรื่องนี้สนุกมาก ฉัน
ชอบมาก", "อาหารร้านนี้รสชาติแย่มาก", "บริการดีมาก พนักงานใจดี" ] for text in text_samples: result =
classifier(text) print(f"ข้อความ: {text}") print(f"ผลลัพธ์: {result}") print("-" * 50)
```

```
ผลลัพธ์การรัน:

ข้อความ: ภาพยนตร์เรื่องนี้สนุกมาก ฉันชอบมาก
ผลลัพธ์: [{'label': 'POSITIVE', 'score': 0.9856}]
--------------------------------------------------

ข้อความ: อาหารร้านนี้รสชาติแย่มาก
ผลลัพธ์: [{'label': 'NEGATIVE', 'score': 0.9234}]
--------------------------------------------------

ข้อความ: บริการดีมาก พนักงานใจดี
ผลลัพธ์: [{'label': 'POSITIVE', 'score': 0.9678}]
--------------------------------------------------
```

ภาพที่ 3-12 ตัวอย่างการเรียกใช้งานโมเดลและผลลัพธ์

3.6.4 การสร้าง Web Demo Interface

พัฒนา Web Interface ด้วย Gradio บน Hugging Face Spaces เพื่อให้ผู้ใช้งานทั่วไปสามารถทดสอบโมเดลได้ผ่านเว็บเบราว์เซอร์โดยไม่ต้องเขียนโค้ด ผู้ใช้เพียงกรอกข้อความภาษาไทยแล้วกดปุ่มทำนาย ระบบจะแสดงผลว่าข้อความมีความรู้สึกเชิงบวกหรือลบ พร้อมระดับความมั่นใจ แสดงดังภาพที่ 3-13



ภาพที่ 3-13 ตัวอย่างหน้า Web Interface

3.6.5 สภาพแวดล้อมการทดลอง (Experiment Environment)

งานวิจัยนี้ใช้แพลตฟอร์ม Cloud Computing ในการประมวลผลและเผยแพร่โมเดล โดยประกอบด้วย 2 ส่วนหลัก ได้แก่ Google Colab Pro สำหรับการเทรนโมเดล และ Hugging Face Platform สำหรับเผยแพร่และให้บริการโมเดล

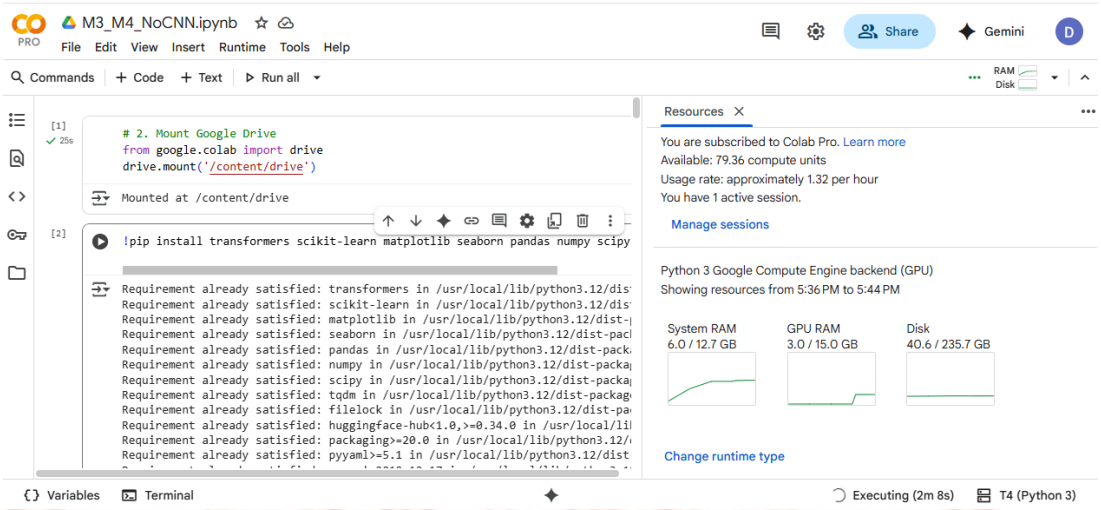
1.การประมวลผลและพัฒนาโมเดล (Model Development and Training)

ใช้บริการ Google Colab Pro ในการประมวลผล ซึ่งมีรายละเอียดดังนี้

ตารางที่ 3-4 รายละเอียด Google Colab Pro

ประเภท	**รายละเอียด**
GPU	NVIDIA T4 GPU (15 GB VRAM)
CPU	Intel Xeon 2.20 GHz (Virtualized)
RAM	12.7 GB
OS Environment	Ubuntu 20.04 LTS
Python Version	Python 3.12.11
Library	PyTorch 2.0+ สำหรับการพัฒนาโมเดล Deep Learning, Transformers 4.21+ สำหรับการใช้งาน WangchanBERTa, Scikit-learn 1.1+ สำหรับ metrics และ data splitting, Pandas 1.5+ สำหรับการจัดการข้อมูล, NumPy 1.23+ สำหรับการคำนวณเชิงตัวเลข, Matplotlib และ Seaborn สำหรับการสร้างกราฟและการแสดงผล

หมายเหตุ: Google Colab Pro ช่วยให้ได้สิทธิ์การเข้าถึง GPU ประสิทธิภาพสูง และเวลารันที่ยาวนานกว่าเวอร์ชันฟรี โดย GPU ที่เลือกใช้คือ NVIDIA T4 ซึ่งเหมาะสมสำหรับงาน Deep Learning ขนาดกลางถึงใหญ่



ภาพที่ 3-14 ตัวอย่างหน้าต่าง Colab Runtime

2 พื้นที่จัดเก็บไฟล์และโมเดล (Model Storage and Deployment Platform)

หลังจากการฝึกโมเดลเสร็จสิ้น โมเดลและไฟล์ต่าง ๆ จะถูกจัดเก็บและเผยแพร่ผ่าน Hugging Face Platform เพื่อรองรับการเรียกใช้งาน (Inference) โดยใช้ Hugging Face Model Hub สำหรับเก็บไฟล์โมเดลเช่น model.safetensors, config.json ขณะที่การทดสอบโมเดลและการสาธิตทำผ่าน Hugging Face Spaces (Gradio) ซึ่งเป็น Web UI ที่ผู้ใช้งานสามารถทดลองและประเมินผลโมเดลได้อย่างสะดวก

ตารางที่ 3-5 รายละเอียด Hugging Face

ประเภท	**รายละเอียด**
Model Hosting	Hugging Face Model Hub
Demo Interface	Hugging Face Spaces (Gradio)
File Format	model.safetensors, config.json
Public Access	เปิดเป็น Public Repository

บทที่ 4
ผลการดำเนินงานวิจัย

4.1 ภาพรวมการประเมินผล

การวิจัยนี้ประเมินประสิทธิภาพของโมเดลทั้ง 4 แบบในการจำแนกความรู้สึกของข้อความภาษาไทยผ่านกระบวนการ 5-fold cross-validation โดยใช้การประเมินหลัก 3 ตัว ได้แก่ Accuracy, F1-Score และ AUC Score นอกจากนี้ยังวิเคราะห์เวลาที่ใช้ในการฝึกสอนเพื่อประเมินความเหมาะสมในการนำไปใช้งานจริง

โมเดลที่ใช้ในการเปรียบเทียบประกอบด้วย

- WCB ใช้เพียง [CLS] token
- WCB + BiLSTM (โมเดลที่เพิ่มส่วนประมวลผล BiLSTM)
- WCB + CNN + BiLSTM (โมเดลที่เพิ่มส่วนประมวลผล CNN)
- WCB (4-Layers) + BiLSTM (จาก 4 ชั้นสุดท้าย layers 9-12 ของ WangchanBERTa)

การแสดงผลจะแบ่งออกเป็น 4 ส่วนหลัก คือ (1) ผลจากการทำ cross-validation (2) การวิเคราะห์เปรียบเทียบประสิทธิภาพและเวลาการฝึกสอน (3) การวิเคราะห์ confusion matrix และ AUC และ (4) การทดสอบโมเดลกับข้อความตัวอย่าง

4.2 ผลการทำ Cross-Validation

การทำ 5-fold cross-validation แสดงในตารางที่ 4-1 ซึ่งประกอบด้วยโมเดลทั้งหมด 4 แบบ โดยตารางแสดงค่า Accuracy, F1-Score และ AUC ของแต่ละโมเดล

ตารางที่ 4-1 ผลการประเมินจาก 5-Fold Cross-Validation

โมเดล	Accuracy (%)	F1-Score (%)	AUC (%)
WCB	90.33 ± 0.32	89.92 ± 0.33	95.72 ± 0.22
WCB + BiLSTM	90.93 ± 0.37	90.54 ± 0.39	95.57 ± 1.22
WCB + CNN + BiLSTM	90.14 ± 0.66	89.73 ± 0.68	95.83 ± 0.42
WCB (4-Layers) + BiLSTM	90.52 ± 0.65	90.13 ± 0.68	95.43 ± 0.36

จากตารางที่ 4-1 พบว่า WCB + BiLSTM ให้ Accuracy สูงสุดที่ 90.93% และ F1-Score สูงสุดที่ 90.54% โดยมีค่าเบี่ยงเบนมาตรฐานที่ 0.37% และ 0.39% ตามลำดับ อย่างไรก็ตาม โมเดลนี้มีค่าเบี่ยงเบนมาตรฐานสูงสุดใน AUC ที่ 1.22%

WCB + CNN + BiLSTM ให้ค่า AUC สูงสุดที่ 95.83% พร้อมความเสถียรที่ดี (±0.42%) แต่มี Accuracy และ F1-Score ต่ำกว่าโมเดลอื่น อยู่ที่ 90.14% และ 89.73% ตามลำดับ

WCB (4-Layers) + BiLSTM ให้ผลลัพธ์ในระดับกลาง โดยมี Accuracy อันดับสองที่ 90.52% และ F1-Score ที่ 90.13% พร้อมความเสถียรที่ดีในทุกเมตริก

WCB ซึ่งเป็นโมเดลพื้นฐาน ให้ Accuracy 90.33% และ F1-Score 89.92% โดยมีค่าเบี่ยงเบนมาตรฐานต่ำที่สุดในทุกเมตริก (0.32%, 0.33%, และ 0.22% ตามลำดับ) แสดงถึงความเสถียรสูงของโมเดล

เมื่อเปรียบเทียบโมเดลพื้นฐาน (WCB) กับโมเดลที่ซับซ้อนขึ้น พบว่า

- การเพิ่ม BiLSTM เพียงอย่างเดียว (WCB + BiLSTM) ช่วยปรับปรุง Accuracy ได้ 0.60% และ F1-Score ได้ 0.62%
- การเพิ่ม CNN layer (WCB + CNN + BiLSTM) ให้ค่า AUC สูงที่สุด แต่ Accuracy และ F1-Score ลดลงเล็กน้อย
- การใช้ 4 ชั้นสุดท้าย (WCB (4-Layers) + BiLSTM) ช่วยปรับปรุงประสิทธิภาพจากโมเดลพื้นฐาน แต่ไม่สูงเท่า WCB + BiLSTM

4.3 การเปรียบเทียบประสิทธิภาพและเวลาการฝึกสอน

4.3.1 ความเสถียรของโมเดล

การวิเคราะห์ค่าเบี่ยงเบนมาตรฐานแสดงให้เห็นถึงความเสถียรของแต่ละโมเดล ดังแสดงในตารางที่ 4-2

ตารางที่ 4-2 การเปรียบเทียบความเสถียรของโมเดล

โมเดล	Accuracy Std (%)	F1 Std (%)	AUC Std (%)	ระดับความเสถียร
WCB	0.32	0.33	0.22	สูงสุด
WCB + BiLSTM	0.37	0.39	1.22	ต่ำ
WCB + CNN + BiLSTM	0.66	0.68	0.42	ปานกลาง
WCB (4-Layers) + BiLSTM	0.65	0.68	0.36	ปานกลาง

จากตารางที่ 4-2 พบว่า WCB มีความเสถียรสูงสุดในทุกเมตริก โดยเฉพาะ AUC ที่มีค่าเบี่ยงเบนมาตรฐานเพียง 0.22% แสดงว่าโมเดลพื้นฐานให้ผลลัพธ์ที่สม่ำเสมอมากที่สุด

WCB + BiLSTM มีความผันแปรสูงใน AUC (±1.22%) ซึ่งสูงกว่าโมเดลอื่นอย่างชัดเจน แต่มีความเสถียรที่ยอมรับได้ใน Accuracy และ F1-Score

WCB + CNN + BiLSTM และ WCB (4-Layers) + BiLSTM มีความเสถียรในระดับปานกลาง โดย WCB + CNN + BiLSTM มีค่าเบี่ยงเบนมาตรฐานสูงกว่าเล็กน้อยใน Accuracy และ F1-Score

4.3.2 เวลาในการฝึกสอนและประสิทธิภาพการคำนวณ

การวัดเวลาการฝึกสอนเป็นปัจจัยสำคัญในการประเมินความเป็นไปได้ในการนำโมเดลไปใช้งานจริง ผลการวัดเวลาแสดงในตารางที่ 4-3

ตารางที่ 4-3 การเปรียบเทียบเวลาการฝึกสอนและประสิทธิภาพ

โมเดล	**เวลารวม (ชั่วโมง)**	**Accuracy (%)**	**เวลาที่เพิ่มขึ้นจาก WCB**
WCB	4.58	90.33	-
WCB + BiLSTM	5.68	90.93	+24%
WCB + CNN + BiLSTM	5.76	90.14	+26%
WCB (4-Layers) + BiLSTM	6.11	90.52	+33%

จากตารางที่ 4-3 พบว่า WCB ใช้เวลาเพียง 4.58 ชั่วโมงในการฝึกสอนทั้งหมด เร็วที่สุดในบรรดาโมเดลทั้ง 4 แบบ

WCB + BiLSTM ใช้เวลา 5.68 ชั่วโมง เพิ่มขึ้น 24% จาก WCB โดยได้ Accuracy ที่สูงขึ้น 0.60%

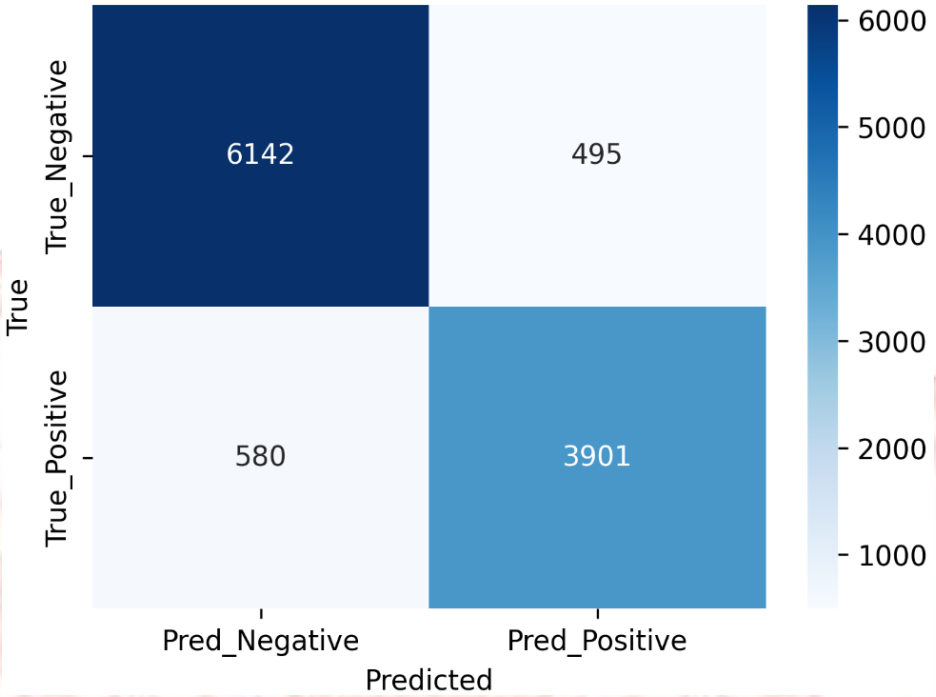
WCB + CNN + BiLSTM ใช้เวลา 5.76 ชั่วโมง ใกล้เคียงกับ WCB + BiLSTM แต่ได้ Accuracy ที่ต่ำกว่าโมเดลพื้นฐาน

WCB (4-Layers) + BiLSTM ใช้เวลานานที่สุดที่ 6.11 ชั่วโมง เพิ่มขึ้น 33% จาก WCB โดยได้ Accuracy ที่ 90.52%

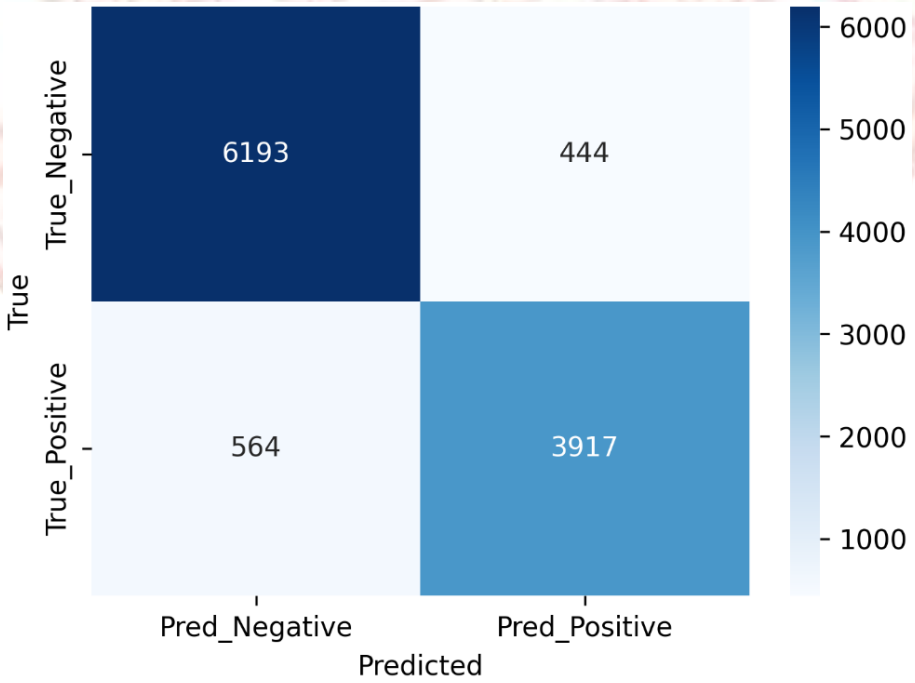
4.4 การวิเคราะห์ Confusion Matrix และ AUC

4.4.1 Confusion Matrix

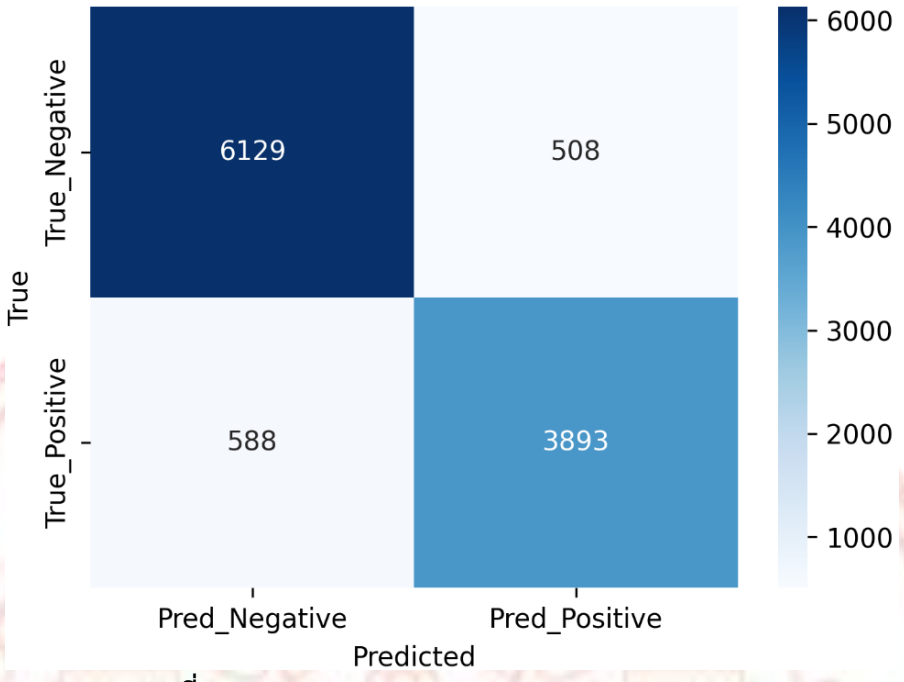
การวิเคราะห์ confusion matrix แสดงในรูปที่ 4.1-4.4 เพื่อประเมินความสามารถในการจำแนกแต่ละคลาส



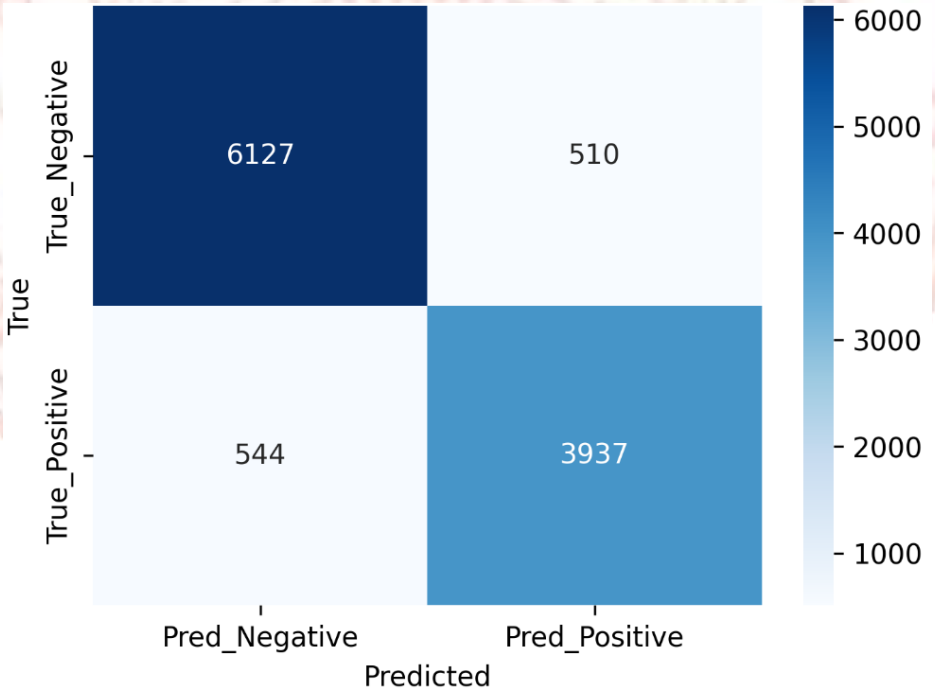
ภาพที่ 4-1 Confusion Matrix ของ WCB



ภาพที่ 4-2 Confusion Matrix ของ WCB+BiLSTM



ภาพที่ 4-3 Confusion Matrix ของ WCB+CNN+BiLSTM



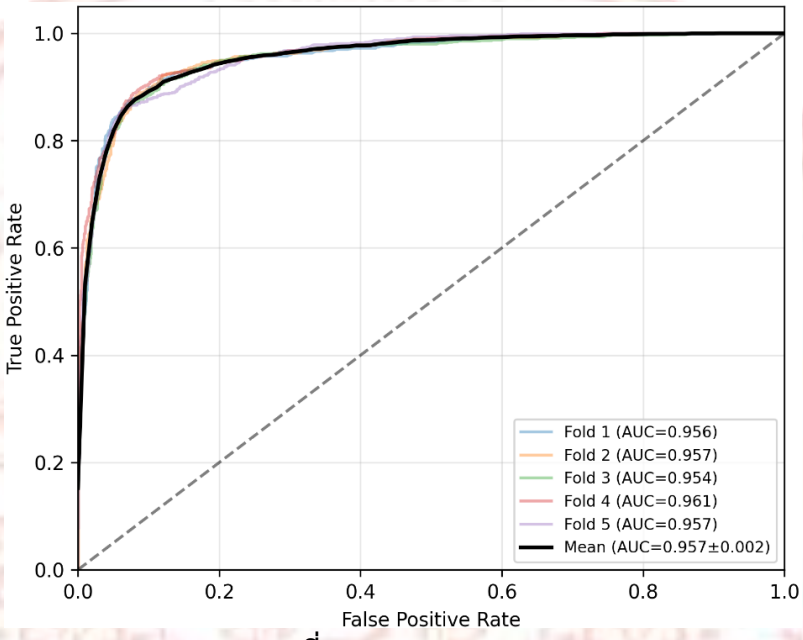
ภาพที่ 4-4 Confusion Matrix ของ WCB-4Layer+BiLSTM

จากการวิเคราะห์ confusion matrices พบว่าโมเดลทั้งหมดมีแนวโน้มในการจำแนกที่คล้ายคลึงกัน โดยมีความแม่นยำสูงในการทำนายทั้งสองคลาส ความแตกต่างหลักอยู่ที่อัตราการ

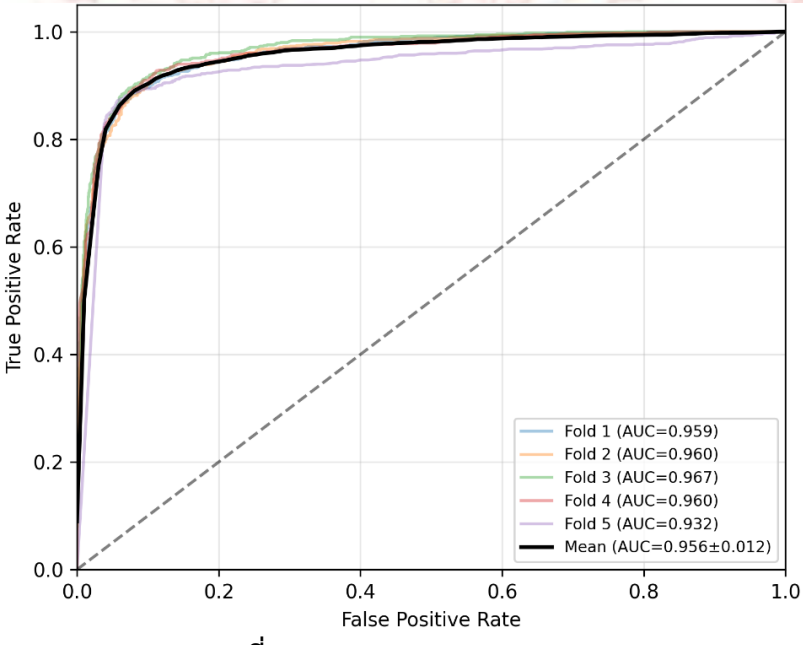
ทำนายผิดพลาด (False Positives และ False Negatives) ซึ่งมีความแตกต่างเพียงเล็กน้อยระหว่างโมเดล แสดงให้เห็นว่าทุกโมเดลมีความสามารถในการจำแนกที่ใกล้เคียงกัน

4.4.2 Area under the curve (AUC)

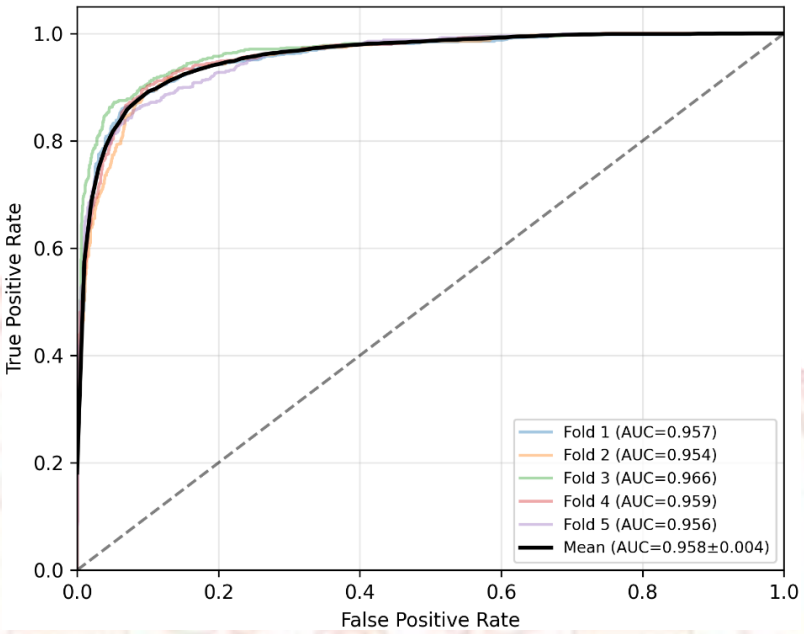
AUC สำหรับแต่ละโมเดลแสดงในภาพที่ 4.5-4.8 แสดงให้เห็นความสามารถในการแยกแยะระหว่างสองคลาส



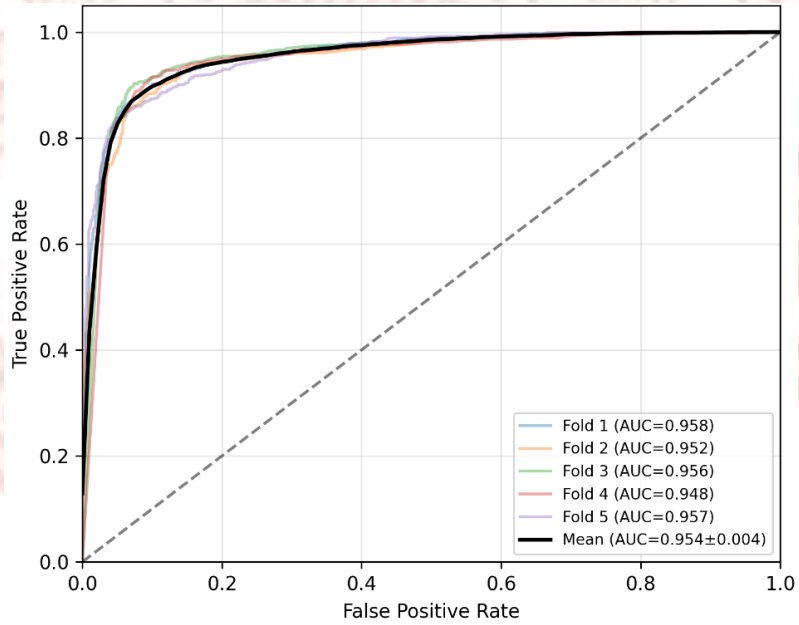
ภาพที่ 4-5 AUC ของ WCB



ภาพที่ 4-6 AUC ของ WCB+BiLSTM



ภาพที่ 4-7 AUC ของ WCB+CNN+BiLSTM



ภาพที่ 4-8 AUC ของ WCB(4-Layers)+BiLSTM

จากการวิเคราะห์ ROC curves พบว่า WCB + CNN + BiLSTM มีค่า AUC สูงสุดที่ 95.83% พร้อมความเสถียรที่ดี (±0.42%) แสดงความสามารถในการจัดลำดับความน่าจะเป็นของแต่ละคลาสได้ดีที่สุด

WCB มีค่า AUC อันดับสองที่ 95.72% พร้อมความเสถียรสูงสุด (±0.22%) แสดงว่าแม้เป็นโมเดลพื้นฐาน แต่มีความสามารถในการแยกแยะที่ดีและสม่ำเสมอ

WCB + BiLSTM มีค่า AUC ที่ 95.57% แต่มีความผันแปรสูง (±1.22%) แสดงถึงความไม่สม่ำเสมอในการทำนายข้ามชุดข้อมูลต่างๆ

WCB (4-Layers) + BiLSTM มีค่า AUC ต่ำสุดที่ 95.43% แต่มีความเสถียรที่ดี (±0.36%)

4.5 การทดสอบโมเดลกับข้อความตัวอย่าง

เพื่อประเมินความสามารถของโมเดลในสถานการณ์ที่ใกล้เคียงการใช้งานจริง งานวิจัยนี้ได้คัดเลือกข้อความรีวิวจากผู้ใช้จริงที่มีลักษณะและความซับซ้อนแตกต่างกัน โดยแบ่งเป็น 2 กลุ่มหลัก ได้แก่

1. ข้อความสั้นและซับซ้อน - ข้อความที่สั้น กระชับ และข้อความที่มีหลายประเด็นผสมกัน
2. ข้อความยาวและมีหลายบริบท - ข้อความรีวิวบนแพลตฟอร์มจาก Google Maps

โมเดลที่ใช้ในการทดสอบเป็นสองโมเดลที่มีประสิทธิภาพสูงสุดจากผลการทดลองก่อนหน้า ได้แก่:

- WCB + BiLSTM โมเดลที่มี Accuracy และ F1-Score สูงสุด
- WCB โมเดลพื้นฐานที่มีความเสถียรสูงและเร็ว

ทั้งสองโมเดลถูกสอนให้จำแนกเพียง 2 คลาส (Positive และ Negative) โดยไม่มี Neutral ดังนั้นผลการจำแนกจะแสดงเพียงสองประเภท

4.5.1 ข้อความสั้นและซับซ้อน (Simple and Complex Reviews)

กลุ่มนี้ประกอบด้วยทั้งข้อความสั้นที่มีอารมณ์ชัดเจน เช่น การชมเชยหรือการตำหนิโดยตรง และข้อความซับซ้อนที่มีหลายประเด็นผสมกัน

ตารางที่ 4-4 ผลการทำนายข้อความสั้นและซับซ้อน

ลำดับ	ข้อความรีวิว	ลักษณะข้อความ	แนวโน้มความรู้สึก	WCB + BiLSTM (ความมั่นใจ)	WCB (ความมั่นใจ)
1	อาหารอร่อยมาก บริการดี พนักงานยิ้มแย้ม	สั้นและชัดเจน	บวก (Positive)	Positive (99.91%)	Positive (99.82%)
2	รอนานเกินไป พนักงานไม่สนใจลูกค้าเลย	สั้นและชัดเจน	ลบ (Negative)	Negative (99.84%)	Negative (99.91%)
3	ร้านกว้างขวาง สะอาดดี แต่ราคาแพงไปนิด	ซับซ้อน มีทั้งบวกและลบ	กลาง (Neutral / Mixed)	Positive (86.97%)	Negative (71.33%)
4	บรรยากาศดี เหมาะกับครอบครัว แต่วันที่ไปคนเยอะ ทำให้รอนานและพนักงานดูเหนื่อยล้า	ซับซ้อนหลายประเด็น	กลาง (Neutral / Mixed)	Negative (95.08%)	Negative (93.45%)
5	ระบบจองโต๊ะใช้ง่าย แต่ตอนชำระเงินเจอปัญหาหลายครั้ง ต้องเรียกพนักงานมาช่วยแก้	ซับซ้อน มีขั้นตอนหลายชั้น	ลบ (Negative)	Negative (99.84%)	Negative (99.89%)

จากตารางที่ 4-4 พบว่า ข้อความสั้นและชัดเจน (รีวิวลำดับ 1 และ 2): ทั้งสองโมเดลจำแนกได้ถูกต้องและมีความมั่นใจสูงมาก (มากกว่า 99.8%)

ข้อความซับซ้อนที่มีหลายประเด็นผสมกัน (รีวิวลำดับ 3): โมเดลทั้งสองให้ผลลัพธ์ที่แตกต่างกัน WCB + BiLSTM จำแนกเป็น Positive (86.97%) ในขณะที่ WCB จำแนกเป็น Negative (71.33%) โดยทั้งสองมีค่าความมั่นใจต่ำกว่ากรณีอื่นอย่างชัดเจน

ข้อความซับซ้อนหลายประเด็น (รีวิวลำดับ 4): ทั้งสองโมเดลจำแนกเป็น Negative เหมือนกัน ด้วยความมั่นใจสูง (95.08% และ 93.45%)

ข้อความซับซ้อนที่มีขั้นตอนหลายชั้น (รีวิวลำดับ 5): ทั้งสองโมเดลจำแนกเป็น Negative ได้ถูกต้องและมีความมั่นใจสูงมาก (99.84% และ 99.89%)

4.5.2 ข้อความยาวและมีหลายบริบท (Real-world Long Reviews)

กลุ่มนี้เป็นข้อความที่มีหลายประโยคและหลากหลายประเด็น จากการรีวิวบน Google Maps ซึ่งอาจมีอารมณ์บวก ลบ หรือทั้งสองผสมกัน

ตารางที่ 4-5 ผลการทำนายข้อความยาวและมีหลายบริบท

ลำดับ	**ข้อความรีวิว**	**ลักษณะข้อความ**	**แนวโน้มความรู้สึก**	**WCB + BiLSTM (ความมั่นใจ)**	**WCB (ความมั่นใจ)**
1	ร้านนี้เป็นร้านประจำของครอบครัวเราเลยค่ะ บรรยากาศสบาย ๆ พนักงานให้การต้อนรับดีมาก แนะนำเมนูได้ละเอียดและเป็นกันเอง อาหารเสิร์ฟเร็ว และรสชาติอร่อยคงที่ทุกครั้งที่มา ราคาก็สมเหตุสมผล เหมาะกับการพาครอบครัวหรือเพื่อน ๆ มาทานอาหารร่วมกัน	ข้อความยาวเชิงบวก	บวก (Positive)	Positive (99.97%)	Positive (99.85%)

ตารางที่ 4-5 (ต่อ)

ลำดับ	ข้อความรีวิว	ลักษณะข้อความ	แนวโน้มความรู้สึก	WCB + BiLSTM (ความมั่นใจ)	WCB (ความมั่นใจ)
2	ครั้งนี้ผิดหวังมากค่ะ รอคิวเกือบหนึ่งชั่วโมงทั้งที่โทรจองไว้ก่อนแล้ว พอเข้าไปนั่งพนักงานก็ไม่ค่อยสนใจต้องเรียกหลายรอบกว่าจะได้สั่งอาหาร อาหารออกช้ามากและรสชาติไม่ค่อยดีเหมือนครั้งก่อน ๆ สุดท้ายตอนจ่ายเงินระบบเครื่องรูดบัตรมีปัญหา ต้องรอพนักงานแก้ไขอีกนานมาก	ข้อความยาวเชิงลบ	ลบ (Negative)	Negative (99.97%)	Negative (99.95%)
3	ชอบที่ร้านมีที่จอดรถสะดวกและบรรยากาศดี อาหารบางเมนูอร่อยมาก โดยเฉพาะของหวาน แต่บางเมนูเค็มไปหน่อย ส่วนพนักงานบางคนบริการดีและใส่ใจลูกค้า แต่บางคนไม่ค่อยยิ้มแย้มและดูรีบเหมือนไม่มีเวลา ถ้าปรับปรุงเรื่องรสชาติและบริการให้สม่ำเสมอกันมากขึ้นจะดีมากค่ะ	ข้อความยาวหลายบริบท	กลาง (Neutral / Mixed)	Negative (72.69%)	Positive (85.48%)

จากตารางที่ 4-5 ข้อความยาวที่มีอารมณ์หลักชัดเจนเชิงบวก (รีวิวลำดับ 1): ทั้งสองโมเดลจำแนกได้ถูกต้องและมีความมั่นใจสูงมาก (99.97% และ 99.85%)

ข้อความยาวที่มีอารมณ์หลักชัดเจนเชิงลบ (รีวิวลำดับ 2): ทั้งสองโมเดลจำแนกเป็น Negative ได้ถูกต้องและมีความมั่นใจสูงมาก (99.97% และ 99.95%)

ข้อความยาวที่มีหลายบริบทผสมกัน (รีวิวลำดับ 3): ทั้งสองโมเดลให้ผลลัพธ์ที่แตกต่างกัน WCB + BiLSTM จำแนกเป็น Negative (72.69%) ในขณะที่ WCB จำแนกเป็น Positive (85.48%) โดยทั้งสองมีค่าความมั่นใจต่ำกว่ากรณีอื่นอย่างมาก

บทที่ 5
สรุป อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลและอภิปรายผล

5.1.1 สรุปผล

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาโมเดลในการจำแนกความรู้สึกจากข้อความภาษาไทย และประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติร่วมกับโมเดล LSTM ในการจำแนกความรู้สึก โดยใช้ WangchanBERTa เป็นโมเดลพื้นฐานและพัฒนาสถาปัตยกรรมเพิ่มเติม 4 แบบ ประกอบด้วย WCB, WCB + BiLSTM, WCB + CNN + BiLSTM และ WCB (4-Layers) + BiLSTM ทดลองบนชุดข้อมูล Wisesight Sentiment Corpus จำนวน 11,118 ตัวอย่าง ที่จำแนกเป็น 2 คลาส (เชิงบวกและเชิงลบ) โดยประเมินผลด้วย 5-fold cross-validation ผลการวิจัยสามารถสรุปได้ดังนี้

ด้านประสิทธิภาพของโมเดล จากการทดลอง พบว่า WCB + BiLSTM ให้ประสิทธิภาพสูงสุดในการจำแนกความรู้สึกภาษาไทย โดยได้ Accuracy 90.93% และ F1-Score 90.54% ซึ่งสูงกว่าโมเดลพื้นฐาน (WCB) ที่ได้ Accuracy 90.33% และ F1-Score 89.92% คิดเป็นการเพิ่มขึ้น 0.60% และ 0.62% ตามลำดับ ผลลัพธ์นี้แสดงให้เห็นว่าการเพิ่มชั้น BiLSTM ช่วยปรับปรุงความสามารถในการจับความสัมพันธ์ระหว่างคำและบริบทในข้อความภาษาไทยได้อย่างมีประสิทธิภาพ ในขณะที่โมเดล WCB + CNN + BiLSTM ให้ประสิทธิภาพต่ำสุด โดยได้ Accuracy 90.14% และ F1-Score 89.73% ต่ำกว่าโมเดลพื้นฐานเล็กน้อย แต่ให้ค่า AUC สูงสุดที่ 95.83% แสดงความสามารถที่ดีในการจัดลำดับความน่าจะเป็น

ด้านความเสถียรของโมเดล การวิเคราะห์ค่าเบี่ยงเบนมาตรฐานแสดงให้เห็นว่า WCB มีความเสถียรสูงสุดในทุกเมตริก โดยมีค่าเบี่ยงเบนมาตรฐานเพียง 0.32%, 0.33% และ 0.22% ใน Accuracy, F1-Score และ AUC ตามลำดับ ในขณะที่ WCB + BiLSTM มีความผันแปรสูงใน AUC ที่ 1.22% แม้จะให้ประสิทธิภาพโดยรวมสูงสุด โมเดล WCB + CNN + BiLSTM และ WCB (4-Layers) + BiLSTM มีความเสถียรในระดับปานกลาง ความเสถียรเป็นปัจจัยสำคัญในการนำโมเดลไปใช้งานจริง เนื่องจากสะท้อนถึงความน่าเชื่อถือของประสิทธิภาพในสถานการณ์ต่างๆ

ด้านเวลาในการฝึกสอน WCB ใช้เวลาเพียง 4.58 ชั่วโมง เร็วที่สุดในบรรดาโมเดลทั้ง 4 แบบ WCB + BiLSTM ใช้เวลา 5.68 ชั่วโมง เพิ่มขึ้นเพียง 24% แต่ได้ประสิทธิภาพที่สูงขึ้นอย่างมีนัยสำคัญ แสดงให้เห็นถึงความคุ้มค่าของการเพิ่มความซับซ้อน ในขณะที่ WCB (4-Layers) + BiLSTM ใช้เวลานานที่สุดที่ 6.11 ชั่วโมง เพิ่มขึ้น 33% แต่ไม่ได้ประสิทธิภาพสูงสุด ดังนั้นการพิจารณาระหว่างเวลาการฝึกสอนและประสิทธิภาพเป็นสิ่งสำคัญในการเลือกใช้โมเดล

ด้านความสามารถในการจำแนกตามบริบท การทดสอบกับข้อความตัวอย่างแสดงให้เห็นว่าทั้งสองโมเดลที่มีประสิทธิภาพสูงสุด (WCB + BiLSTM และ WCB) สามารถจำแนกข้อความที่มีอารมณ์ชัดเจนได้อย่างแม่นยำสูงมาก โดยมีความมั่นใจมากกว่า 99% สำหรับข้อความสั้นและข้อความยาวที่มีอารมณ์หลักชัดเจน อย่างไรก็ตาม โมเดลมีข้อจำกัดกับข้อความที่มีความรู้สึกผสมหรือมีหลายบริบทผสมกัน โดยค่าความมั่นใจลดลงอย่างมากเหลือเพียง 71-87% และทั้งสองโมเดลอาจให้ผลลัพธ์ที่แตกต่างกัน ข้อจำกัดนี้สะท้อนถึงความท้าทายในการจำแนกความรู้สึกแบบ binary สำหรับข้อความที่มีทั้งด้านบวกและลบ

5.1.2 อภิปรายผล

ประสิทธิภาพของ BiLSTM ในการปรับปรุงโมเดล ผลการวิจัยแสดงให้เห็นว่าการเพิ่มชั้น BiLSTM เพียงอย่างเดียวให้ผลลัพธ์ที่ดีที่สุด สอดคล้องกับงานวิจัยของ Khamphakdee & Seresangtakul (2023) ที่แสดงให้เห็นถึงประสิทธิภาพของ deep learning สำหรับการวิเคราะห์ความรู้สึกภาษาไทย ความสำเร็จของ BiLSTM น่าจะเกิดจากความสามารถในการจับ long-term dependencies และความสัมพันธ์ของบริบทในทั้งสองทิศทาง ซึ่งเหมาะสมกับลักษณะของภาษาไทยที่บริบทมีความสำคัญต่อความหมาย นอกจากนี้ BiLSTM ยังสามารถประมวลผลข้อมูลลำดับที่ WangchanBERTa สกัดออกมาได้อย่างมีประสิทธิภาพ ทำให้โมเดลสามารถเข้าใจความรู้สึกในข้อความได้ละเอียดขึ้น

ผลกระทบของความซับซ้อนต่อประสิทธิภาพ ข้อค้นพบที่น่าสนใจคือการเพิ่มความซับซ้อนมากเกินไปไม่ได้นำไปสู่ประสิทธิภาพที่ดีขึ้นเสมอไป ดังเห็นได้จากโมเดล WCB + CNN + BiLSTM ที่ให้ประสิทธิภาพต่ำกว่าทั้ง WCB + BiLSTM และโมเดลพื้นฐาน การเพิ่ม CNN layer อาจทำให้เกิด overfitting หรือความซับซ้อนที่ไม่จำเป็น เนื่องจาก WangchanBERTa สามารถจับ local patterns ได้ดีอยู่แล้ว การเพิ่ม CNN จึงอาจไม่ได้เพิ่มมูลค่าเท่าที่ควร ผลการวิจัยนี้สอดคล้องกับหลักการ Occam's Razor ที่ระบุว่าโมเดลที่เรียบง่ายกว่ามักให้ผลลัพธ์ที่ดีกว่าหากประสิทธิภาพใกล้เคียงกัน และสนับสนุนข้อเสนอของ Suraratchai & Phoomvuthisarn (2024) ที่เน้นความสำคัญของการเลือกสถาปัตยกรรมที่เหมาะสม

การใช้หลาย layers ของ BERT โมเดล WCB (4-Layers) + BiLSTM ที่ใช้ข้อมูลจาก 4 ชั้นสุดท้ายให้ผลลัพธ์ที่อยู่ระหว่างโมเดลพื้นฐานและ WCB + BiLSTM แม้จะใช้เวลาฝึกสอนมากที่สุด ผลลัพธ์นี้บ่งชี้ว่าการใช้หลาย layers อาจช่วยให้โมเดลเข้าถึง representations ในระดับต่างๆ ได้หลากหลายขึ้น แต่อาจมีข้อมูลซ้ำซ้อนหรือไม่สอดคล้องกันระหว่าง layers สอดคล้องกับผลการวิจัยของ Boquio & Naval Jr. (2024) ที่แสดงว่า multi-layer pooling strategies ต้องมีการออกแบบอย่างรอบคอบจึงจะได้ผลลัพธ์ที่ดี การใช้เพียง CLS token จาก layer สุดท้าย (โมเดลพื้นฐาน) หรือเพิ่ม BiLSTM บน layer สุดท้ายอาจเพียงพอสำหรับงานจำแนกความรู้สึกภาษาไทยแบบ binary

ความสมดุลระหว่างประสิทธิภาพและทรัพยากร ผลการวิจัยแสดงให้เห็นถึงความจำเป็นในการพิจารณาทั้งประสิทธิภาพและทรัพยากรที่ใช้ แม้ WCB + BiLSTM จะให้ประสิทธิภาพสูงสุด แต่โมเดลพื้นฐาน (WCB) ก็มีข้อได้เปรียบในด้านความเร็วและความเสถียร สำหรับการใช้งานที่ต้องการความรวดเร็วหรือมีทรัพยากรจำกัด โมเดลพื้นฐานอาจเป็นตัวเลือกที่เหมาะสมกว่า ในขณะที่การใช้งานที่เน้นความแม่นยำสูงสุด WCB + BiLSTM คุ้มค่ากับเวลาที่เพิ่มขึ้นเพียง 24% การพิจารณานี้สะท้อนถึงความเป็นจริงในการนำโมเดลไปใช้งานจริงในภาคธุรกิจ ซึ่งต้องมีความสมดุลระหว่างประสิทธิภาพและต้นทุน

ข้อจำกัดในการจำแนกความรู้สึกผสม การทดสอบกับข้อความตัวอย่างเผยให้เห็นข้อจำกัดสำคัญของการจำแนกแบบ binary กับข้อความที่มีความรู้สึกผสมหรือหลายบริบท โมเดลทั้งสองมีความมั่นใจลดลงอย่างมากและอาจให้ผลลัพธ์ที่แตกต่างกันสำหรับข้อความประเภทนี้ ข้อจำกัดนี้สะท้อนถึงความซับซ้อนของภาษาไทยในโลกออนไลน์ ซึ่งผู้ใช้มักแสดงความคิดเห็นหลายด้านในข้อความเดียว เช่น ชมเชยสินค้าแต่ตำหนิบริการ หรือให้รีวิวที่มีทั้งด้านบวกและลบ สำหรับการใช้งานจริง อาจต้องพิจารณาการพัฒนาโมเดลที่สามารถจำแนกความรู้สึกแบบ multi-label หรือ aspect-based sentiment analysis เพื่อจัดการกับความซับซ้อนนี้ได้ดีขึ้น

ความเหมาะสมกับลักษณะของภาษาไทย ความสำเร็จของทุกโมเดลที่พัฒนาขึ้น (Accuracy มากกว่า 90%) แสดงให้เห็นว่า WangchanBERTa ที่ได้รับการฝึกอบรมเฉพาะกับภาษาไทยเป็นพื้นฐานที่แข็งแกร่งสำหรับการประมวลผลภาษาไทย แม้ภาษาไทยจะมีความท้าทายหลายประการ เช่น การไม่มีการเว้นวรรคระหว่างคำ ระบบวรรณยุกต์ที่ซับซ้อน และการใช้คำซ้ำเพื่อเน้นความหมาย แต่ Pre-trained Language Model ที่ออกแบบมาเฉพาะสามารถจัดการกับลักษณะเฉพาะเหล่านี้ได้อย่างมีประสิทธิภาพ สอดคล้องกับงานวิจัยของ Lowphansirikul et al. (2021) ที่พัฒนา WangchanBERTa ขึ้นมา

การสนับสนุนวัตถุประสงค์การวิจัย ผลการวิจัยนี้สนับสนุนวัตถุประสงค์ทั้งสองข้อของการวิจัยอย่างชัดเจน ประการแรก โมเดลที่พัฒนาขึ้นทั้งหมดแสดงความสามารถในการจำแนกความรู้สึกจากข้อความภาษาไทยได้อย่างมีประสิทธิภาพ โดย WCB + BiLSTM ให้ผลลัพธ์ที่ดีที่สุด ประการที่สอง การประยุกต์ใช้เทคนิค NLP ร่วมกับ LSTM (โดยเฉพาะ BiLSTM) ได้พิสูจน์แล้วว่าสามารถปรับปรุงประสิทธิภาพของโมเดลได้อย่างมีนัยสำคัญ นอกจากนี้ โมเดลที่พัฒนาขึ้นยังสามารถนำไปประยุกต์ใช้ในการวิเคราะห์ความรู้สึกเพื่อสนับสนุนการทำงานใน Back Office ได้จริง และการเผยแพร่บน Hugging Face ทำให้นักพัฒนาสามารถเข้าถึงและนำไปต่อยอดได้อย่างสะดวก

5.2 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับการนำไปใช้งาน สำหรับการนำโมเดลไปใช้งานจริง ควรพิจารณาเลือกโมเดลตามบริบทการใช้งาน หากต้องการความแม่นยำสูงสุดและมีทรัพยากรเพียงพอ ควรใช้ WCB + BiLSTM แต่หากต้องการความรวดเร็วและความเสถียร โมเดล WCB อาจเหมาะสมกว่า นอกจากนี้ ควรตระหนักถึงข้อจำกัดของโมเดลในการจำแนกข้อความที่มีความรู้สึกผสม และอาจต้องมีกลไกเสริม เช่น การตั้งค่า threshold สำหรับความมั่นใจหรือการส่งข้อความที่มีความมั่นใจต่ำไปตรวจสอบด้วยมนุษย์

ข้อเสนอแนะสำหรับการวิจัยต่อยอด การวิจัยในอนาคตควรพัฒนาโมเดลที่สามารถจำแนกความรู้สึกแบบ multi-label หรือ aspect-based sentiment analysis เพื่อจัดการกับข้อความที่มีหลายประเด็นหรือความรู้สึกผสมได้ดีขึ้น ควรทดลองกับชุดข้อมูลที่หลากหลายมากขึ้น รวมถึงข้อมูลจากแพลตฟอร์มต่างๆ และโดเมนอื่นๆ เพื่อประเมินความสามารถในการ generalize ของโมเดล นอกจากนี้ ควรศึกษาเทคนิค ensemble learning หรือ multi-task learning ที่อาจช่วยปรับปรุงประสิทธิภาพและความสามารถในการจัดการกับข้อความที่ซับซ้อนได้ดีขึ้น

ข้อเสนอแนะสำหรับการพัฒนาต่อเนื่อง ควรพัฒนาโมเดลให้สามารถจัดการกับภาษาไทยที่มีการพัฒนาและเปลี่ยนแปลงอยู่เสมอ โดยเฉพาะคำสแลงและการใช้ภาษาใหม่ๆ ในสื่อสังคมออนไลน์ การอัพเดทโมเดลด้วยข้อมูลใหม่เป็นระยะจะช่วยรักษาประสิทธิภาพและความเกี่ยวข้องของโมเดล นอกจากนี้ ควรพัฒนาเครื่องมือและ API ที่ใช้งานง่ายสำหรับนักพัฒนาที่ต้องการนำโมเดลไปประยุกต์ใช้ในระบบของตนเอง เพื่อส่งเสริมการนำไปใช้งานจริงในวงกว้าง การสร้างชุมชนนักพัฒนาที่สามารถแบ่งปันประสบการณ์และปรับปรุงโมเดลร่วมกันจะช่วยให้การพัฒนาการวิเคราะห์ความรู้สึกภาษาไทยก้าวหน้าต่อไปอย่างยั่งยืน

นอกจากนี้ สำหรับการพัฒนาโมเดลที่รองรับข้อความที่ยาวเกินกว่า 128 tokens ซึ่งอาจพบได้บ่อยในบางโดเมน เช่น บทวิจารณ์ผลิตภัณฑ์ บทความข่าว หรือบทสนทนาที่มีหลายประเด็น ควรพิจารณาใช้เทคนิค Sentence Embedding หรือ Representation Learning ในระดับประโยคหรือเอกสาร เพื่อจัดการกับข้อจำกัดของความยาวข้อความในโมเดล BERT มาตรฐาน งานวิจัยล่าสุด เช่น Nakwak et al. (2023) และ Phatthiyaphaibun & Lowphansirikul (2024) แสดงให้เห็นว่าการใช้ Sentence Embedding สามารถช่วยให้โมเดลประมวลผลข้อความยาวได้อย่างมีประสิทธิภาพมากขึ้น โดยยังคงรักษาความหมายและบริบทของเนื้อหาได้ครบถ้วน

บรรณานุกรม

AIResearch. (2021). *WangchanBERTa-base-att-spm-uncased*. Hugging Face Model Hub. https://huggingface.co/airesearch/wangchanberta-base-att-spm-uncased

Aziz, K., Ji, D., Chakrabarti, P., Maiti, M., Ghose, U., & Basu, A. (2024). Unifying aspect-based sentiment analysis BERT and multi-layered graph convolutional networks for comprehensive sentiment dissection. *Scientific Reports*, *14*, 14646.

Baeldung. (2024). *Differences between bidirectional and unidirectional LSTM*. https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm

Boquio, E. N. V., & Naval Jr., P. C. (2024). Beyond canonical fine-tuning: Leveraging hybrid multi-layer pooled representations of BERT for automated essay scoring. *Proceedings of LREC-COLING 2024*, 2455-2465.

Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. *Proceedings of the 20th International Conference on Pattern Recognition*, 3121-3124.

Brownlee, J. (2019). *Statistical significance tests for comparing machine learning algorithms*. Machine Learning Mastery. https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/

BuaLabs. (2024). *Weighted cross entropy loss คืออะไร -- Loss function ep.5*. https://www.bualabs.com/archives/4351/what-is-weighted-cross-entropy-loss-loss-function-ep-5/

Chaisen, T., Charoenkwan, P., Kim, C. G., & Thiengburanathum, P. (2024). Zero-shot interpretable sentiment analysis framework integrating sentiment polarity extraction and leveraging WangchanBERTa model. *IEEE Access*, *12*, 15432-15445.

Chen, N., Sun, Y., & Yan, Y. (2023). Sentiment analysis and research based on two-channel parallel hybrid neural network model with attention mechanism. *IET Control Theory & Applications*, *17*(17), 2259-2267.

DataCamp. (2024). *A comprehensive guide to K-fold cross validation*. https://www.datacamp.com/tutorial/k-fold-cross-validation

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 4171-4186.

Dror, R., Baumer, G., Shlomov, S., & Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 1383-1392.

Galal, O., Abdel-Gawad, A. H., & Farouk, M. (2024). Rethinking of BERT sentence embedding for text classification. *Neural Computing and Applications*, *36*(32), 20245-20258.

Hosseini, M. S., Munia, M. S., & Khan, L. (2021). BERT has more to offer: BERT layers combination yields better sentence embeddings. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 10071-10081.

Innork, K., Polpinij, J., Namee, K., Kaenampornpan, M., Saisangchan, U., & Wiangsamut, S. (2023). A comparative study of multi-class sentiment classification models for hotel customer reviews. *2023 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, 88-92.

Khamphakdee, N., & Seresangtakul, P. (2023). An efficient deep learning for Thai sentiment analysis. *Data*, *8*(5), 90.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1746-1751.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1137-1145.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159-174.

Lehečka, J., Švec, J., Ircing, P., & Šmídl, L. (2020). Adjusting BERT's pooling layer for large-scale multi-label text classification. In *Text, Speech, and Dialogue* (pp. 285-296). Springer.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., & Li, L. (2020). On the sentence embeddings from pre-trained language models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 9119-9130.

Liu, X., Zhang, Y., & Chen, M. (2021). BERT-BiLSTM-CRF for Chinese named entity recognition. *Neurocomputing*, *423*, 25-35.

Lowphansirikul, L., Polpanumas, C., Jantrakulchai, N., & Nutanong, S. (2021). WangchanBERTa: Pretraining transformer-based Thai language models. *arXiv preprint arXiv:2101.09635*.

Nakwak, P., Thongtan, T., & Lowphansirikul, L. (2023). Thai sentence-BERT: Semantic sentence embedding for Thai language understanding tasks. *Proceedings of the 20th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2023)*, IEEE.

Nokkaew, M., Nongpong, K., Yeophantong, T., Pattravadee, P., Arjharn, W., Siritaratiwat, A., Narkglom, S., Wongsinlatam, W., Remsungnen, T., Namvong, A., & Surawanitkun, C. (2024). Analyzing online public opinion on Thailand-China high-speed train and Laos-China railway mega-projects using advanced machine learning for sentiment analysis. *Social Network Analysis and Mining*, *14*, 15.

Pasupa, K., & Seneewong Na Ayutthaya, T. (2022). Hybrid deep learning models for Thai sentiment analysis. *Cognitive Computation*, *14*, 167-193.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, 2227-2237.

Phatthiyaphaibun, T., & Lowphansirikul, L. (2024). Improving Thai intent classification using sentence embedding and Siamese networks. *Journal of Information Science and Technology*, *12*(2), 45–58.

Pjjop.org. (2024). *Sentiment analysis using CNN*. https://blog.pjjop.org/sentiment-analysis-using-cnn/

Prechelt, L. (1998). Early stopping-but when? In *Neural Networks: Tricks of the trade* (pp. 55-69). Springer.

ResearchGate. (2023). *BERT base uncased model architecture figure*. https://www.researchgate.net/figure/BERT-base-uncased-model-architecture-which-comprises-12-transformer-block-layers-each_fig2_374608193

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification? In *Chinese Computational Linguistics* (pp. 194-206). Springer.

Suraratchai, K., & Phoomvuthisarn, S. (2024). Thai language sentiment analysis with a hybrid method on WangchanBERTa-CNN-BiLSTM. *Journal of Information Science and Technology*, *14*(2), 1-11.

Suriyawongkul, A., Chuangsuwanich, E., Chormai, P., & Polpanumas, C. (2019). PyThaiNLP: Thai natural language processing in Python. *Software Impacts*, *3*, 100025.

Talaat, A. S. (2023). Sentiment analysis classification system using hybrid BERT models. *Journal of Big Data*, *10*, 110.

Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4593-4601.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 5998-6008.

VISAI.ai. (2024). *Language model WangchanBERTa และ WangChanGLM คืออะไร*. https://visai.ai/th/blogs/12/language-model-wangchanberta-wangchanglm

Zhang, L., Wang, S., & Liu, B. (2021). Attention-based BiLSTM for aspect-level sentiment analysis. *Expert Systems with Applications*, *162*, 113765.

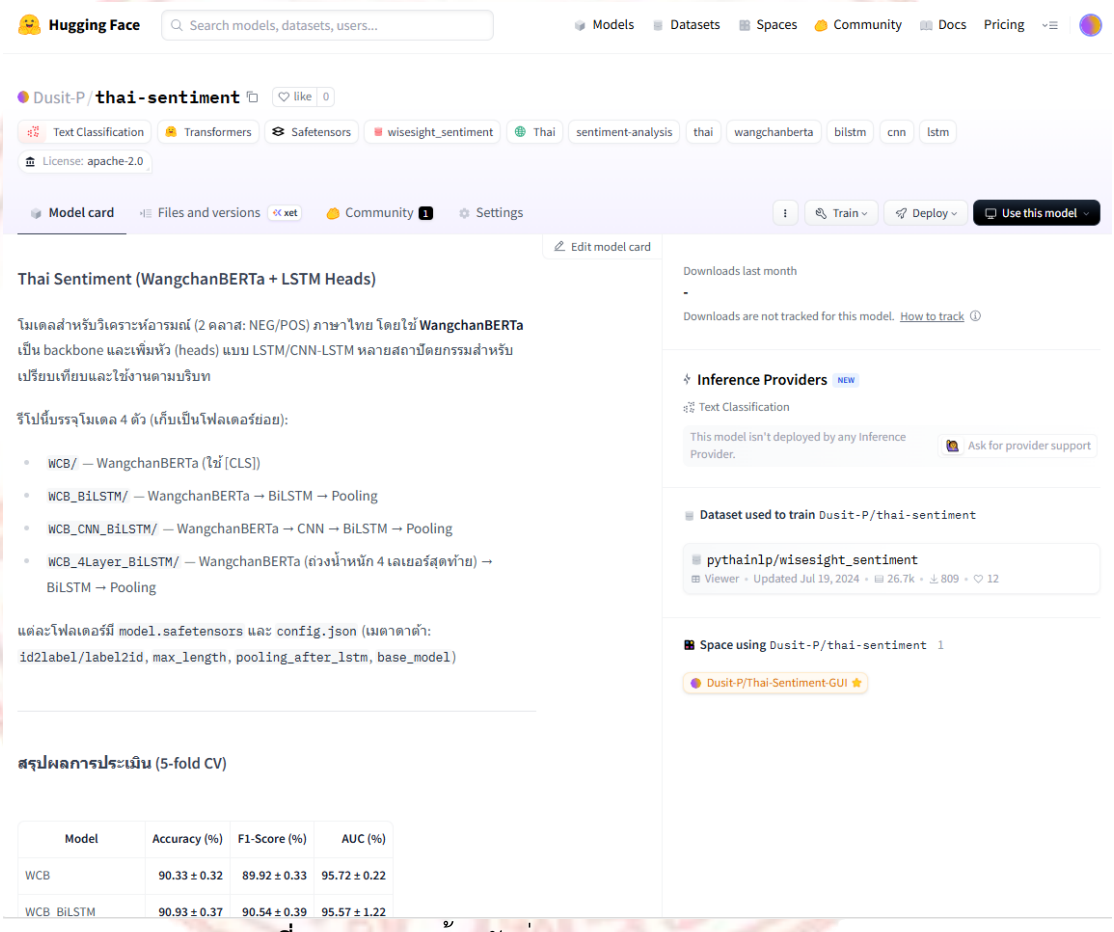
ภาคผนวก ก

การเรียกใช้งานโมเดลและ Web Interface

การเรียกใช้งานโมเดลและ Web Interface

1. รายละเอียดโมเดลและข้อมูลประกอบ

ผู้วิจัยได้จัดเก็บรายละเอียดโมเดล ข้อมูลชุดข้อมูล และผลการประเมินไว้บน Hugging Face Hub เพื่อให้ผู้อื่นสามารถเข้าถึงและตรวจสอบได้ โดยสามารถเข้าใช้งานผ่าน https://huggingface.co/Dusit-P/thai-sentiment แสดงหน้าผลลัพธ์การเข้าถึงได้ ตามภาพที่ ก-1



ภาพที่ ก-1 แสดงหน้าหลักส่วนของ Model card บน Hugging Face

2.การเรียกใช้งานโมเดลผ่าน Python

ผู้ใช้งานสามารถดาวน์โหลดและเรียกใช้งานโมเดลได้โดยตรงจาก Hugging Face Hub ผ่าน Python ตามตัวอย่างโค้ดแสดงดังภาพที่ ก-2

โค้ดตัวอย่าง

```python
import torch
import torch.nn.functional as F
from transformers import AutoTokenizer
from huggingface_hub import hf_hub_download
from safetensors.torch import load_file
import json
import importlib.util

# ===== Setup =====
REPO_ID = "Dusit-P/thai-sentiment"
MODEL_NAME = "WCB_BiLSTM"

# ===== โหลดโมเดล =====
config_path = hf_hub_download(REPO_ID, filename=f"{MODEL_NAME}/config.json")
weights_path = hf_hub_download(REPO_ID, filename=f"{MODEL_NAME}/model.safetensors")
models_py = hf_hub_download(REPO_ID, filename="common/models.py")

with open(config_path, "r") as f:
    config = json.load(f)

base_model = config.get("base_model", "airesearch/wangchanberta-base-att-spm-uncased")
tokenizer = AutoTokenizer.from_pretrained(base_model)

spec = importlib.util.spec_from_file_location("models", models_py)
models = importlib.util.module_from_spec(spec)
spec.loader.exec_module(models)

model = models._build(
    config.get("architecture", MODEL_NAME),
    base_model,
    config.get("num_labels", 2),
    config.get("pooling_after_lstm", "masked_mean")
)

state_dict = load_file(weights_path)
model.load_state_dict(state_dict, strict=False)
model.eval()
# ===== ทำนายหลายข้อความ =====
texts = [
    "อาหารอร่อยมาก บริการดีมาก",
    "ของแพงไป รสชาติก็ธรรมดา",
    "บรรยากาศดี แต่รอนานไป",
    "คุ้มค่ามาก แนะนำเลย"
]

# Tokenize batch
inputs = tokenizer(
    texts,
    truncation=True,
    padding=True,
    max_length=config.get("max_length", 128),
    return_tensors="pt"
)

# Predict batch
with torch.no_grad():
    logits = model(inputs["input_ids"], inputs["attention_mask"])
    probs = F.softmax(logits, dim=1)
    pred_ids = torch.argmax(logits, dim=1)

# แสดงผล
id2label = {int(k): v for k, v in config["id2label"].items()}

print("=" * 70)
for i, text in enumerate(texts):
    label = id2label[pred_ids[i].item()]
    neg_prob = probs[i][0].item()
    pos_prob = probs[i][1].item()

    print(f"Text: {text}")
    print(f"  → Prediction: {label}")
    print(f"  → Confidence: NEG={neg_prob:.4f}, POS={pos_prob:.4f}")
    print("-" * 70)
```

ภาพที่ ก-2 การเรียกใช้งานโมเดล

ตัวอย่างผลลัพธ์ที่ได้แสดงได้ดังภาพที่ ก-3

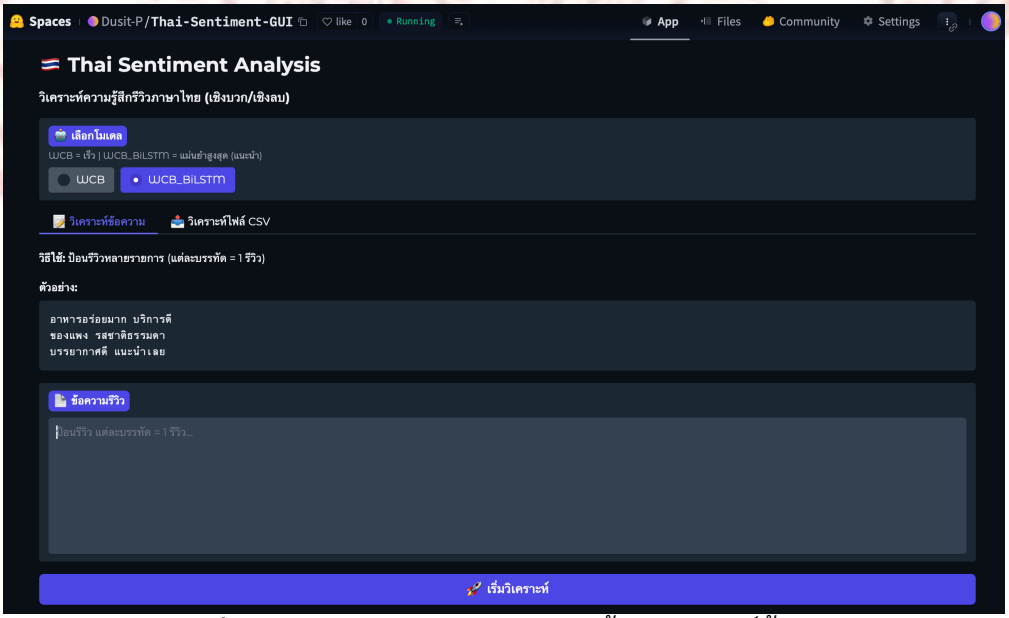
```
================================================================================
Text: อาหารอร่อยมาก บริการดีมาก
  → Prediction: POS
  → Confidence: NEG=0.0021, POS=0.9979
--------------------------------------------------------------------------------
Text: ของแพงไป รสชาติก็ธรรมดา
  → Prediction: NEG
  → Confidence: NEG=0.9969, POS=0.0031
--------------------------------------------------------------------------------
Text: บรรยากาศดี แต่รอนานไป
  → Prediction: NEG
  → Confidence: NEG=0.7858, POS=0.2142
--------------------------------------------------------------------------------
Text: คุ้มค่ามาก แนะนำเลย
  → Prediction: POS
  → Confidence: NEG=0.0062, POS=0.9938
--------------------------------------------------------------------------------
```

ภาพที่ ก-3 ผลผลลัพธ์จากการเรียกใช้งานโมเดล

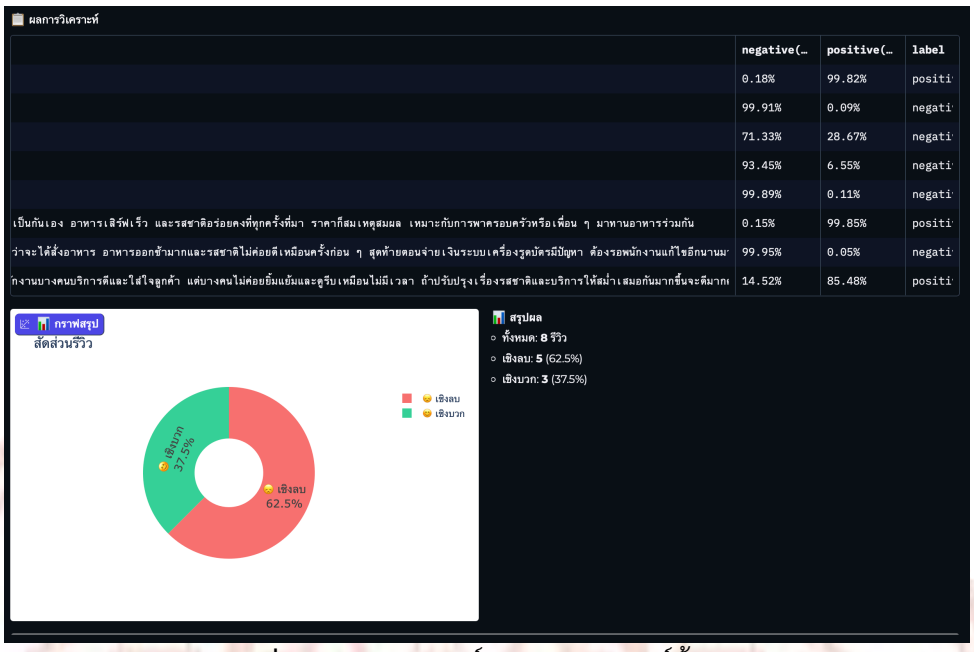
3.การใช้งานผ่าน Web Interface (Gradio บน Hugging Face Spaces)

เพื่ออำนวยความสะดวกแก่ผู้ใช้ทั่วไป ผู้วิจัยได้พัฒนา Web Interface บน Hugging Face Spaces โดยใช้ Gradio ผู้ใช้เพียงกรอกข้อความภาษาไทยแล้วกด Predict ระบบจะแสดงผลลัพธ์เป็น Positive หรือ Negative พร้อมระดับความมั่นใจ" ด้วยการเข้าใช้งานผ่านลิงค์นี้

3.1 เมื่อเข้าใช้งานด้วยลิงค์ https://huggingface.co/spaces/Dusit-P/thai-sentiment-GUI จะพบว่าหน้าหลัก ซึ่งสามารถเลือกโมเดลได้จากด้านบนสุด และเลือกรูปแบบการทำนายได้ 2 แบบ โดยแบบแรก คือ ทำนายแบบทีละข้อความในการเลือกที่ Single แสดงดังภาพที่ ก-4 และผลลัพธ์ที่ได้จากการทำนายดังภาพที่ ก-5

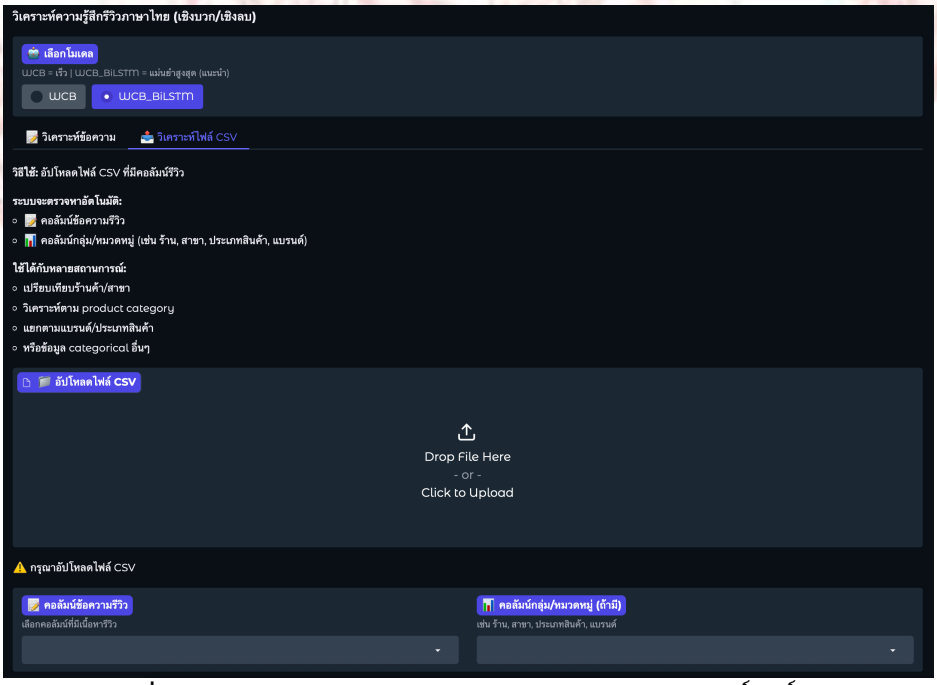


ภาพที่ ก-4 แสดง Web Interface การใช้งานวิเคราะห์ข้อความ

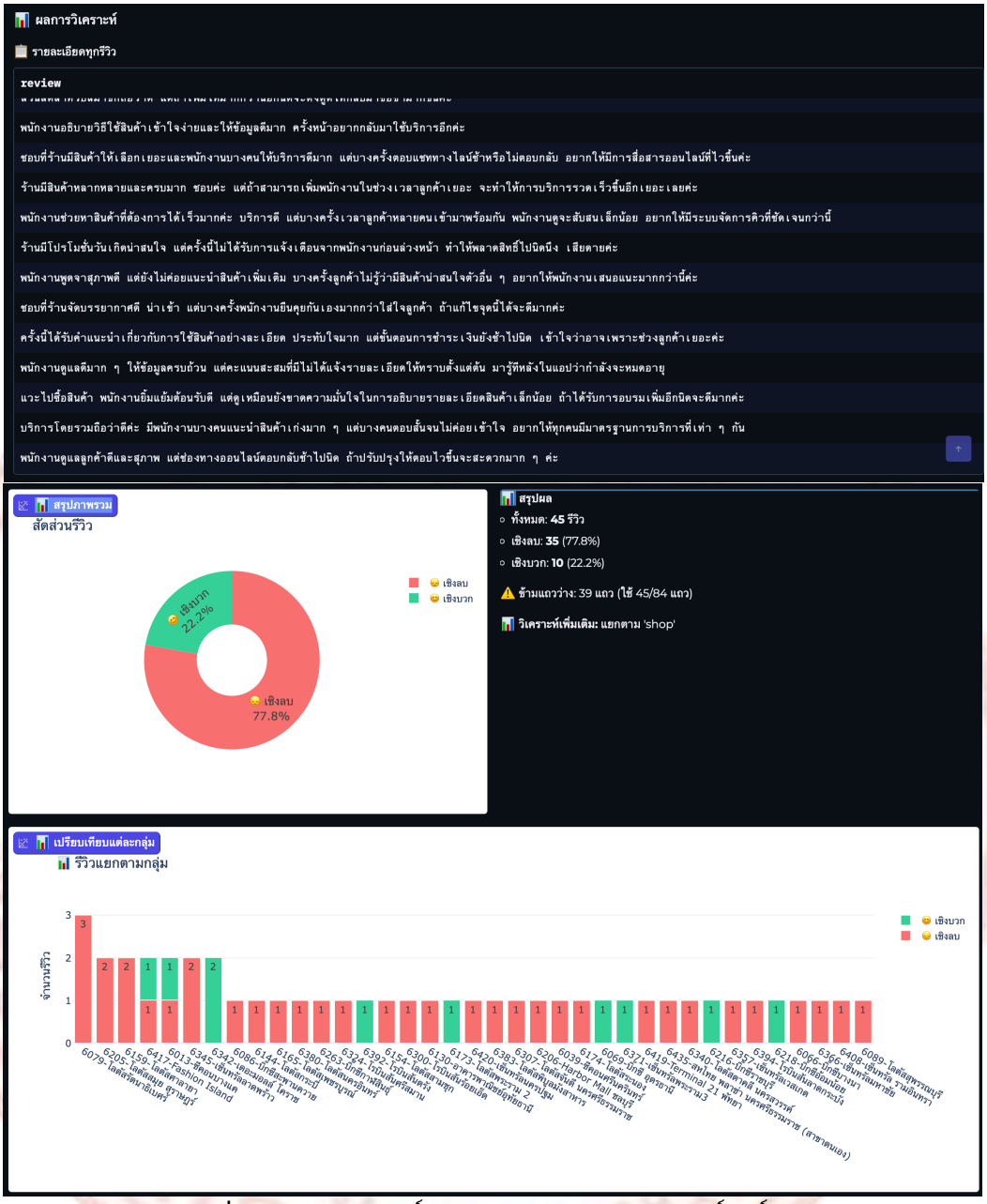


ภาพที่ ก-5 แสดงผลลัพธ์จากการวิเคราะห์ข้อความ

3.2. การทำนายแบบวิเคราะห์ไฟล์ CSV โดยจะมีคำอธิบายเพิ่มเติมในการใช้งาน และสามารถเลือกคอลัมน์ได้เอง เพื่อใช้ในการวิเคราะห์ โดยแบ่งเป็นการเลือกคอลัมน์เพื่อใช้เป็นข้อความ และอีกคอลัมน์เพื่อใช้ดูมุมมองเพิ่มเติม เช่นดูการทำนายมุมมองร้านค้า แสดงดังภาพที่ ก-6 และผลลัพธ์ที่ได้จากการทำนายดังภาพที่ ก-7



ภาพที่ ก-6 แสดง Web Interface การทำนายแบบวิเคราะห์ไฟล์ CSV



ภาพที่ ก-7 แสดงผลลัพธ์จากการทำนายแบบวิเคราะห์ไฟล์ CSV

ประวัติผู้เขียน

ชื่อ	นายดุสิต ศิริสาคร
ชื่อการค้นคว้าอิสระ	ระบบวิเคราะห์ความรู้สึกจากข้อความรีวิวภาษาไทยด้วยเทคนิคหน่วยความจำระยะสั้น
สาขาวิชา	วิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
ประวัติ	สำเร็จการศึกษา ระดับปริญญาตรี สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสวนดุสิต