Sentiment Analysis of Thai Stock Market Opinions through Pantip.com

ศิริประภา อุปภาค

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
สถาบันวิทยาการหุ่นยนต์ภาคสนาม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีการศึกษา 2567

สารบัญ

บทที่ 1 บทนำ

1.1 ที่มา ความสำคัญ

ในปัจจุบัน การตัดสินใจลงทุนในตลาดหุ้นไม่ได้จำกัดเพียงการวิเคราะห์ข้อมูลพื้นฐานหรือข้อมูลเชิงเทคนิคเท่านั้น แต่ยังอาศัยความคิดเห็นและประสบการณ์ที่เผยแพร่ในสื่อโซเชียลมีเดียเป็นอีกหนึ่งแหล่งข้อมูลสำคัญ เนื่องจากความคิดเห็นเหล่านี้สามารถสะท้อนอารมณ์และความรู้สึกของนักลงทุนในสภาวะตลาดที่เปลี่ยนแปลงอย่างรวดเร็ว

อย่างไรก็ตาม การวิเคราะห์ความคิดเห็นในภาษาไทยบนสื่อโซเชียลมีเดียยังคงเผชิญกับความท้าทายหลายประการ เนื่องจากลักษณะของข้อความที่ไม่เป็นทางการ เช่น การเว้นวรรคที่ไม่สม่ำเสมอ การใช้คำแสลง และการผสมผสานระหว่างภาษาไทยกับภาษาอังกฤษ ซึ่งแตกต่างจากภาษาอังกฤษที่มีโครงสร้างที่ชัดเจนและเครื่องมือประมวลผลที่ได้รับการพัฒนาอย่างต่อเนื่อง นอกจากนี้ การขาดชุดข้อมูลที่มีการกำหนดป้ายกำกับอย่างครอบคลุมยังเป็นอุปสรรคสำคัญที่จำกัดความแม่นยำในการฝึกโมเดลวิเคราะห์ความคิดเห็น

ด้วยความท้าทายดังกล่าว งานวิจัยนี้จึงมีความสำคัญในการพัฒนาโมเดล Sentiment Analysis ที่สามารถรองรับลักษณะเฉพาะของข้อความบนสื่อโซเชียลมีเดียภาษาไทยได้อย่างมีประสิทธิภาพ โดยมุ่งเน้นการนำโมเดลเชิงลึก เช่น ThaiBERT มาใช้ เพื่อจับความสัมพันธ์เชิงบริบทที่ซับซ้อนในข้อความ ซึ่งจะช่วยให้นักลงทุนและผู้เกี่ยวข้องสามารถใช้ข้อมูลความคิดเห็นเหล่านี้เป็นเครื่องมือประกอบการตัดสินใจลงทุนได้อย่างมีประสิทธิภาพและมีความน่าเชื่อถือ

การศึกษานี้จึงไม่เพียงแต่เป็นการพัฒนาเครื่องมือทางเทคโนโลยี NLP สำหรับภาษาไทย แต่ยังมีศักยภาพในการสนับสนุนการตัดสินใจในตลาดหุ้นโดยการให้ข้อมูลเชิงลึกที่สะท้อนความรู้สึกและแนวโน้มของนักลงทุนในสภาวะตลาดที่ผันผวน

1.2 ประโยคปัญหางานวิจัย (Problem Statement)

การวิเคราะห์ความคิดเห็น (Sentiment Analysis) ในภาษาไทยจากข้อมูลโซเชียลมีเดียนั้นมียังมีความซับซ้อนเนื่องจากการใช้คำที่ไม่ได้เป็นทางการ และมีรูปแบบการเขียนที่ไม่ตายตัว ทั้งการเว้นวรรค การใช้คำแสลง และการใช้คำที่มีการผสมระหว่างภาษาไทยและภาษาอังกฤษ ซึ่งแตกต่างจากภาษาอังกฤษที่มีโครงสร้างชัดเจนและมีเครื่องมือวิเคราะห์ที่พัฒนามาอย่างยาวนาน โดยปัญหานี้ส่งผลให้กระบวนการต่าง ๆ ในการวิเคราะห์ เช่น กระบวนการตัดคำ (Word Segmentation) ยุ่งยากยิ่งขึ้น รวมถึงขั้นตอนการประมวลผลล่วงหน้า (Preprocessing) มีความซับซ้อน อีกทั้งยังพบว่าขาดชุดข้อมูลขนาดใหญ่และเป็นมาตรฐาน (Standard Dataset) ที่เหมาะสมสำหรับฝึกโมเดล ซึ่งจะส่งผลโดยตรงต่อความแม่นยำของงานวิเคราะห์ความคิดเห็น ไม่ว่าจะเป็นโมเดลเชิงสถิติแบบดั้งเดิม (เช่น Naïve Bayes, SVM) หรือโมเดลเชิงลึก (Deep Learning)

ถึงแม้ในปัจจุบันจะเครื่องมือประมวลผลภาษาไทย (Thai NLP Tools) แต่พบว่าส่วนใหญ่ถูกพัฒนามาเพื่อรองรับข้อความทางการ (Formal Text) ในขณะที่ข้อมูลโซเชียลมีเดียมักมีรูปแบบการพิมพ์ที่หลากหลายและขาดมาตรฐาน จึงทำให้เครื่องมือเหล่านี้เมื่อถูกนำมาใช้งานจริงยังไม่สามารถจัดการคำแสลง การลากเสียง หรือการใช้สัญลักษณ์แทนอารมณ์ได้อย่างมีประสิทธิภาพ นอกจากนี้ยังไม่มีเกณฑ์วัด (Benchmark) ที่ชัดเจนและได้รับการยอมรับอย่างแพร่หลายสำหรับเปรียบเทียบประสิทธิภาพของโมเดลต่าง ๆ ในบริบทของภาษาไทยโดยเฉพาะ

การนำโมเดลที่มีโครงสร้างเชิงลึกอย่าง Bidirectional Encoder Representations from Transformers (BERT) เข้ามาใช้งานในบริบทของภาษาไทย ถือเป็นแนวทางที่มีศักยภาพในการแก้ปัญหานี้ เนื่องจาก BERT สามารถเข้าใจความสัมพันธ์เชิงบริบทในระดับลึกได้ดี ทั้งในข้อความทางการและไม่เป็นทางการ นอกจากนี้ การใช้ BERT ที่ผ่านการพัฒนาสำหรับภาษาไทย (ThaiBERT) ยังช่วยเพิ่มความแม่นยำในการจัดการข้อมูลเชิงความหมายที่ซับซ้อน อย่างไรก็ตาม การปรับแต่ง (Fine-tuning) และการจัดการชุดข้อมูลที่เหมาะสมยังคงเป็นความท้าทายสำคัญ ดังนั้น ปัญหาหลักของงานวิจัยนี้คือ การออกแบบและพัฒนาโมเดล Sentiment Analysis ภาษาไทยที่สามารถจัดการความซับซ้อนของข้อความที่ไม่เป็นทางการบนโซเชียลมีเดีย โดยผสานการใช้ BERT หรือโมเดลเชิงลึกอื่น ๆ ที่เหมาะสมกับภาษาไทย เพื่อเพิ่มความแม่นยำและความน่าเชื่อถือของผลลัพธ์

1.3 ผลผลิตและผลลัพธ์ (Outputs and Outcomes)

ผลผลิต

1. ชุดข้อมูลความคิดเห็นจากสื่อโซเชียล (Pantip.com) ที่รวบรวมโพสต์และคอมเมนต์เกี่ยวกับหุ้น พร้อมป้ายกำกับอารมณ์ (Positive, Negative, Neutral) ในช่วง 1–5 วันก่อนและหลังข่าว
2. โมเดลวิเคราะห์ความคิดเห็น (Sentiment Analysis Model) สำหรับข้อความภาษาไทย
3. ระบบต้นแบบ (Prototype/Dashboard) สำหรับแสดงผลการวิเคราะห์ความสัมพันธ์ระหว่างข่าวกับปัจจัยทางการเงิน

ผลลัพธ์

1. เพิ่มประสิทธิภาพในการวิเคราะห์ข่าวตลาดหุ้น ช่วยให้นักลงทุนและผู้เกี่ยวข้องประเมินความเสี่ยงได้
2. เพิ่มความเข้าใจในแนวโน้มและอารมณ์ของนักลงทุนจากความคิดเห็นในสื่อโซเชียล
3. ขยายองค์ความรู้ด้าน Sentiment Analysis ภาษาไทยในโดเมนการเงิน

1.4 ความต้องการของระบบ (Requirements)

1. ข้อมูลความคิดเห็นเกี่ยวกับหุ้นที่เก็บรวบรวมจาก Pantip.com
2. ข้อมูลข่าวต้องประกอบด้วยเนื้อหา วันที่เผยแพร่ และหัวข้อที่เกี่ยวข้องกับหุ้นไทย
3. โมเดล Sentiment Analysis สำหรับภาษาไทยที่จำแนกความคิดเห็นออกเป็น Positive, Negative และ Neutral
4. ระบบจัดเก็บข้อมูลและผลการวิเคราะห์เพื่อใช้ในการตรวจสอบและเปรียบเทียบในอนาคต
5. การแสดงผลผ่านทาง dashboard ในรูปแบบของกราฟ และตาราง

1.5 ขอบเขตของงานวิจัย (Scopes)

1. พัฒนาโมเดลสำหรับข่าวสารการเงินภาษาไทยเท่านั้น
2. วิเคราะห์ความคิดเห็นที่มีต่อหุ้นไทยในสื่อโซเชียล (Pantip.com) โดยไม่จำกัดเฉพาะหุ้นใดหุ้นหนึ่ง
3. ข้อมูลที่เก็บจะอยู่ในรูปแบบของข้อความเต็ม (paragraph text)
4. วิเคราะห์ Sentiment ของใน 3 ระดับ ได้แก่ Positive, Negative, Neutral
5. วิเคราะห์ผลกระทบระยะสั้นในช่วง 1–5 วันหลังการลงกระทู้ (blog) เพื่อระบุหุ้นที่ถูกกล่าวถึงมากที่สุด 10 อันดับแรกในช่วงนั้น
6. วัดผลความแม่นยำของโมเดลด้วยเกณฑ์การวัด (Evaluation Metrics) ได้แก่
    - Accuracy: ความถูกต้องในการจำแนก Sentiment ของข่าว
    - Precision, Recall, F1-score ในการประเมินความสามารถของโมเดลในการจำแนก Positive, Negative, และ Neutral
    - Correlation Analysis ตรวจสอบความสัมพันธ์ระหว่าง Sentiment ของข่าวกับการเปลี่ยนแปลงของราคาหุ้นและปริมาณการซื้อขาย
7. ไม่รวมข้อมูลที่ไม่ได้อยู่ในรูปแบบข้อความ เช่น รูปภาพ เสียงหรือวิดีโอ

1.6 ข้อกำหนดของงานวิจัย (Assumptions)

1. ความคิดเห็นจาก Pantip.com ที่เกี่ยวข้องกับการลงทุนในหุ้นถือว่ามีความเป็นตัวแทนและมีคุณภาพพอเพียงสำหรับการวิจัย
2. ชุดข้อมูลความคิดเห็นจะประกอบด้วยข้อความเต็ม (paragraph text) พร้อมข้อมูล metadata เช่น วันที่โพสต์ เวลา และแท็ก/หัวข้อที่เกี่ยวข้อง
3. ทรัพยากรด้านการประมวลผล (Processing Power) และเทคนิคการประมวลผลภาษาไทย (Thai NLP) ที่มีอยู่เพียงพอสำหรับการพัฒนาและทดสอบโมเดล
4. โมเดลวิเคราะห์ความคิดเห็นที่พัฒนาจะสามารถปรับปรุงหรือฝึกซ้ำ (Retraining) ได้หากมีการเพิ่มข้อมูลข่าวหรือปรับขอบเขตการวิเคราะห์ในอนาคต
5. ระยะเวลา 1–5 วันหลังเผยแพร่ข่าวถือว่าเหมาะสมในการวัดผลกระทบระยะสั้นต่อราคาหุ้นและปริมาณการซื้อขาย ทั้งนี้ไม่ได้ครอบคลุมผลกระทบระยะยาว

1.7 ขั้นตอนการดำเนินงาน

แผนการดำเนินโครงการเริ่มตั้งแต่วันที่ 18 มกราคม พ.ศ. 2568 และสิ้นสุดในวันที่ 5 พฤษภาคม พ.ศ. 2568 โดยแสดงรายละเอียดการดำเนินงานดังนี้

หัวข้อและรายละเอียด	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ศึกษา Research และกำหนดขอบเขต - ค้นคว้างานวิจัยและบทความที่เกี่ยวข้อง - กำหนดเป้าหมาย/วัตถุประสงค์ และขอบเขตของโครงงาน															
เก็บรวบรวมชุดข้อมูล (Data Collection) - รวบรวมข่าวการเงินในไทยจากแหล่งต่าง ๆ - จัดหมวดหมู่และโครงสร้างข้อมูลเบื้องต้น															
เตรียม/ทำความสะอาดข้อมูล (Data Preprocessing) - กำจัดข้อมูลซ้ำ/ข้อมูลขาดหาย/สัญลักษณ์พิเศษ - ตัดคำ (Tokenization)															
วิเคราะห์ข้อมูลเบื้องต้น - สำรวจการกระจายตัวของคำ/อารมณ์ (Positive, Negative, Neutral) - ดูค่าเฉลี่ย ความถี่ หรือ Distribution เบื้องต้น															

หัวข้อและรายละเอียด	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ออกแบบ/เลือกโมเดลต้นแบบ (Model Selection) - ทดลองโมเดล Machine Learning / Deep Learning - เปรียบเทียบจุดเด่น/จุดด้อยในบริบทข่าวการเงินไทย															
ฝึกอบรมและปรับจูนโมเดล (Training & Tuning) - ปรับ Hyperparameters ให้เหมาะสม - Overfitting/Underfitting															
ประเมินและปรับปรุงโมเดล (Evaluation & Improvement) - ประเมินโมเดลด้วย Metrics (Accuracy, Precision, Recall, F1-score เป็นต้น) - Fine-tune เพื่อปรับปรุงผลลัพธ์															
สรุปผลและจัดทำรายงาน (Final Report) - วิเคราะห์ผลการทดลอง สรุปข้อสรุปและข้อเสนอแนะ - จัดทำรายงานโครงงานฉบับสมบูรณ์															

หมายเหตุ : สีเทา คือช่วงเวลาในการดำเนินงาน

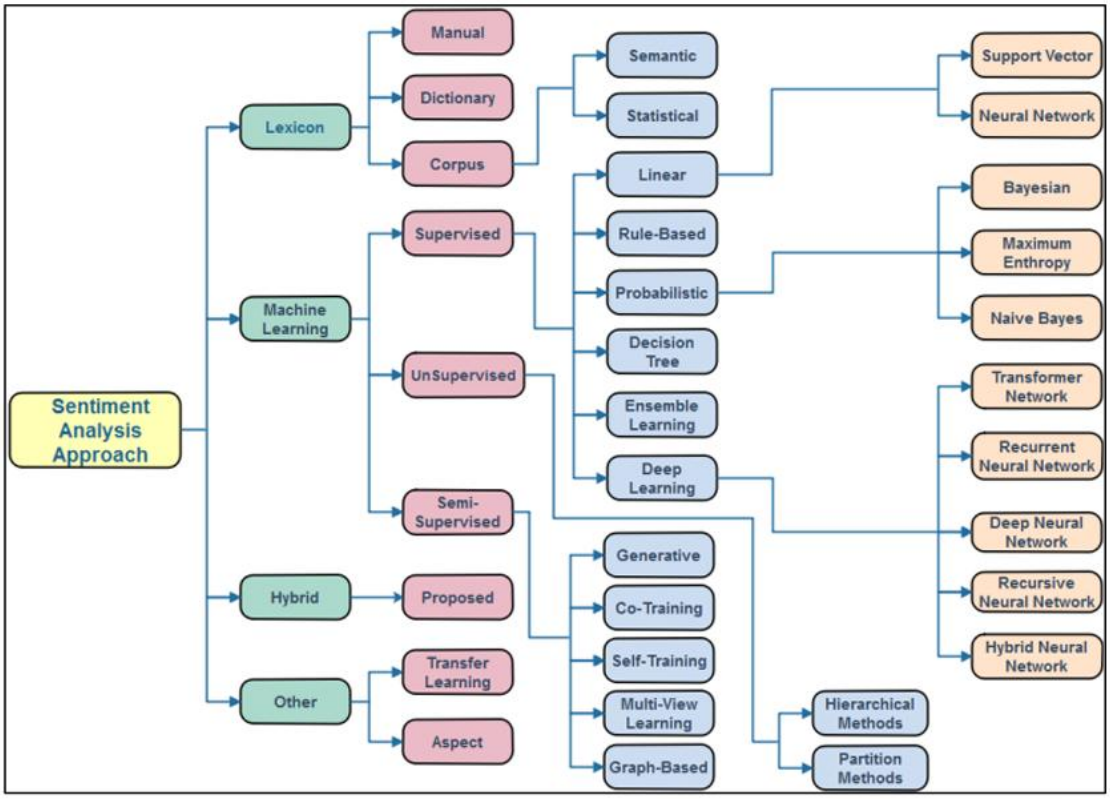
สีแดงคือช่วงที่มีการส่งรายงาน

บทที่ 2 ทฤษฎี/งานวิจัย/การศึกษาที่เกี่ยวข้อง

2.1 การวิเคราะห์ความรู้สึกของข่าวการเงินด้วยโมเดล BERT [1]

งานวิจัยดังกล่าวใช้ข้อมูลข่าวการเงินจำนวน 5,842 รายการ โดยแบ่งออกเป็น 3 กลุ่ม (บวก, ลบ, เป็นกลาง) หลังจากเตรียมข้อมูลด้วยการทำความสะอาดและตัดคำ จากนั้นจึงนำข้อมูลเข้าสู่กระบวนการ Fine-Tuning ของโมเดล BERT เพื่อจำแนกความรู้สึก ผลการประเมินด้วยตัวชี้วัดทางสถิติ เช่น Accuracy, Precision, Recall และ F1-score แสดงถึงความแม่นยำที่สูงของโมเดล

2.1.1 ขั้นตอนการประมวลผลและปรับแต่งโมเดล



รูปภาพ 1 วิธีการวิเคราะห์ความรู้สึก

1. การเก็บรวบรวมข้อมูลและการแบ่งประเภท (Data Collection)
   - รวบรวมข่าวการเงินจากแหล่งข้อมูลที่น่าเชื่อถือ จำนวนรวม 5,842 รายการ
2. การเตรียมข้อมูล (Data Preprocessing)
   - ลบอักขระพิเศษและองค์ประกอบที่ไม่เกี่ยวข้อง (Text Cleaning)
   - แบ่งข้อความเป็นหน่วยคำเพื่อให้โมเดลสามารถประมวลผลได้ (Tokenization)

- ปรับความยาวของข้อความให้คงที่ เพื่อให้สอดคล้องกับ input ของโมเดล (Padding)
- แบ่งชุดข้อมูล เป็น Training, Validation และ Test เพื่อประเมินประสิทธิภาพและป้องกัน Overfitting

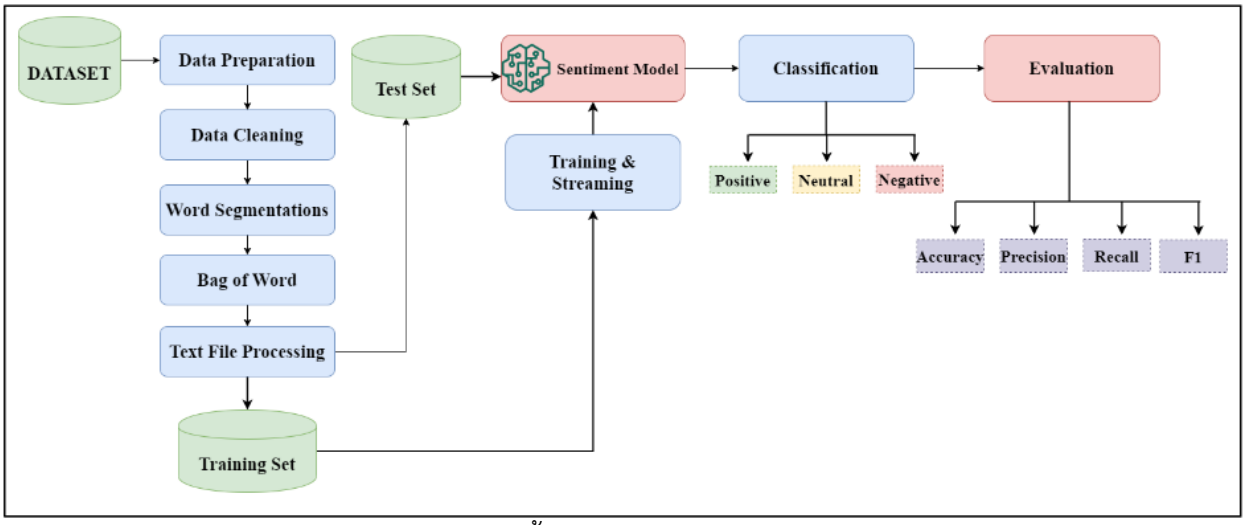
3. การปรับแต่งโมเดล (Fine-Tuning)
   - ตั้งค่า Hyperparameters โดยกำหนดค่า Learning Rate Batch Size และจำนวน Epochs
   - โหลดโมเดลและปรับแต่งชั้นจำแนก (Classification Head)
   - ฝึกโมเดลด้วยข้อมูล Training พร้อมติดตามค่า Loss บน Validation Set
   - ใช้เทคนิค Early Stopping เมื่อ Loss ไม่ลดลง
4. การประเมินผลโมเดล (Model Evaluation)
   - กำหนดตัวชี้วัดหลัก
     - Accuracy คำนวณอัตราการทำนายที่ถูกต้องจากจำนวนตัวอย่างทั้งหมด
     - Precision วัดความถูกต้องของการทำนายในแต่ละประเภท (ลด False Positives)
     - Recall วัดความสามารถในการจับข้อมูลที่เป็นจริง (ลด False Negatives)
     - F1-Score ค่าเฉลี่ยถ่วงน้ำหนักระหว่าง Precision และ Recall เพื่อให้เห็นความสมดุล
   - ใช้ชุดข้อมูลสำหรับการทดสอบ เพื่อแยกชุดข้อมูลที่ไม่เคยใช้ในการฝึก (Test Set) เพื่อประเมินความสามารถในการ Generalize ของโมเดล
   - ดำเนินการประเมินผล โดยป้อนชุดข้อมูล Test ลงในโมเดล คำนวณค่าตัวชี้วัดที่กำหนดไว้ในแต่ละประเภทความรู้สึก
   - ประเมินด้วยวิธี Cross Validation ด้วยวิธี k-fold cross validation (เช่น k=5 หรือ k=10) เพื่อให้ผลการประเมินมีความเสถียรและเป็นกลาง
5. วิเคราะห์ผลลัพธ์
   - เปรียบเทียบค่า Accuracy, Precision, Recall และ F1-Score เพื่อระบุจุดแข็งและจุดที่ต้องปรับปรุง
   - ตรวจสอบกราฟของ Loss และ Accuracy ระหว่างช่วงฝึกและ Validation

ผลการทดลองแสดงให้เห็นว่าโมเดล BERT สามารถจำแนกความรู้สึกของข่าวการเงินได้อย่างยอดเยี่ยม โดยได้ค่า Accuracy อยู่ที่ 95.29%, Precision 95.37%, Recall 95.24% และค่า F1-score 95.32% พร้อมทั้งมีค่า Loss ต่ำเพียง 9.07% ซึ่งบ่งบอกถึงประสิทธิภาพสูงในการจับบริบทและจำแนกประเภทความรู้สึกในข่าวการเงินอย่างแม่นยำ ทำให้โมเดลนี้สามารถนำไปประยุกต์ใช้เพื่อสนับสนุนการตัดสินใจในตลาดการเงินได้อย่างมีประสิทธิผล

2.2 งานวิจัยด้านการวิเคราะห์ความรู้สึกในบทความแนะนำสินค้าออนไลน์ [2]

ตัวต้นแบบสำหรับวิเคราะห์ความรู้สึกจากบทความและความคิดเห็นออนไลน์ โดยมีเป้าหมายจำแนกความคิดเห็นออกเป็น 3 ระดับ ได้แก่ เชิงบวก (Positive), เป็นกลาง (Neutral) และเชิงลบ (Negative) ด้วยการใช้เทคนิค Web Scraping ร่วมกับการประมวลผลข้อความและการจำแนกประเภทด้วย Machine Learning

2.2.1 ขั้นตอนการดำเนินงานและการประมวลผลข้อมูล



รูปภาพ 2 แสดงขั้นตอนกระบวนการสร้างตัวแบบ

1. การเก็บรวบรวมและเตรียมข้อมูล (Data Preparation Phase)
   - ดึงข้อมูลบทความและความคิดเห็นจากเว็บไซต์ (www.blognone.com) โดยใช้ภาษา Python และไลบรารี BeautifulSoup
   - รวบรวมข้อมูลจาก 252 บทความและ 1,412 ความคิดเห็น รวมเป็น 83,670 คำ
   - ทำการทำความสะอาดข้อมูล (Data Cleaning) ด้วย Regular Expression เพื่อลบ HTML Tag, URL และสัญลักษณ์พิเศษออกจากข้อความ
2. การประมวลผลข้อความและการตัดคำ (Text File Processing Phase)
   - ใช้เทคนิคการตัดคำ (Word Segmentation) สำหรับภาษาไทยโดยใช้ 3 อัลกอริทึม ได้แก่ NewMM Engine, Longest Engine และ AttaCut Engine
   - สร้างชุดคุณลักษณะ (Feature Extraction) โดยการสร้าง Bag of Words เพื่อแปลงข้อความให้เป็นเวกเตอร์สำหรับการเรียนรู้ของเครื่อง
3. การฝึกอบรมและจำแนกประเภท (Training & Classification Phase)

- แบ่งชุดข้อมูลออกเป็นชุดฝึก (Training Set) 80% และชุดทดสอบ (Testing Set) 20%
- นำข้อมูลที่ผ่านการตัดคำและสร้างคุณลักษณะเข้าสู่กระบวนการจำแนกประเภทด้วยเทคนิคต่าง ๆ ได้แก่
    - K-Nearest Neighbors (KNN)
    - Random Forest
    - Logistic Regression
    - Support Vector Machines (SVM)
- เปรียบเทียบผลลัพธ์ของแต่ละเทคนิคเพื่อหาตัวแบบที่ให้ความแม่นยำสูงสุด

4. การประเมินผลตัวแบบ (Model Evaluation Phase)
    - ใช้ตัวชี้วัดหลัก เช่น Accuracy, Precision, Recall และ F1-Score ในการวัดประสิทธิภาพ
    - ผลการทดลองพบว่าเทคนิค SVM ร่วมกับการตัดคำแบบ Longest ให้ผลลัพธ์ที่ดีที่สุด โดยมีค่า Accuracy อยู่ที่ประมาณ 79%

ตัวต้นแบบวิเคราะห์ความรู้สึกในบทความแนะนำสินค้าออนไลน์ภาษาไทย โดยใช้เทคนิค web scraping ดึงข้อมูลจากเว็บไซต์ จากนั้นประมวลผลข้อความด้วยการตัดคำ (NewMM, Longest, AttaCut) และสร้าง Bag-of-Words เพื่อฝึกโมเดลจำแนกประเภท (KNN, Random Forest, Logistic Regression, SVM) ผลการทดลองพบว่าโมเดล SVM ร่วมกับการตัดคำแบบ Longest ให้ความแม่นยำสูงสุดประมาณ 79% ซึ่งสรุปได้ว่าแนวทางนี้มีประสิทธิภาพในการวิเคราะห์ความรู้สึกของข้อมูลภาษาไทยและสามารถนำไปประยุกต์ใช้ในงานวิจัยด้านการตลาดออนไลน์ได้อย่างมีประสิทธิผล

บทที่ 3 ระเบียบวิธีวิจัย

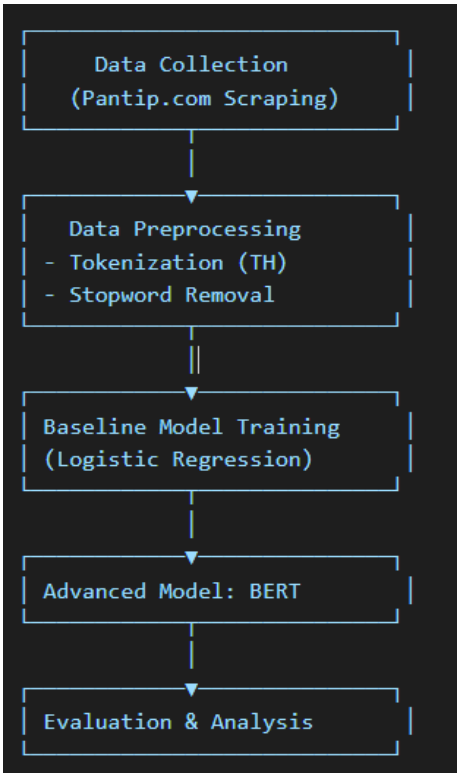
จากแนวคิดการทำโครงการวิเคราะห์ตลาดหุ้นของไทยใน Pantip.com ด้วยกระบวนการวิเคราะห์ความรู้สึก (Sentiment Analysis) มีกระบวนการการดำเนินงานดังนี้

3.1 ภาพรวมและขอบเขตของการวิจัย

ต้นแบบโครงการการวิเคราะห์ความคิดเห็นที่เป็นข้อความภาษาไทยเกี่ยวกับข่าวสารการเงินในตลาดหุ้นไทยจาก Pantip.com เพื่อจำแนกอารมณ์หรือแนวโน้มความรู้สึกออกเป็น 3 ส่วนหลัก ได้แก่ บวก (Positive), ลบ (Negative) และ กลาง (Neutral) โดยลำดับขั้นตอนของระบบประกอบด้วย 5 ส่วนหลัก ได้แก่

1. การเก็บข้อมูลจากแหล่งที่เกี่ยวข้อง (Web Scraping)
2. การเตรียมข้อมูลล่วงหน้า (Preprocessing)
3. การประเมินคุณภาพข้อมูลด้วยโมเดล Logistic Regression
4. การพัฒนาโมเดลขั้นสูงด้วย BERT
5. การประเมินผลการทำนาย

3.1.1 แผนภาพต้นแบบระบบ (Overview Pipeline)



รูปภาพ 3 Pipeline ของระบบ Sentiment Analysis จากความคิดเห็นใน Pantip.com

3.2 รายละเอียดทางเทคนิค

3.2.1 การเก็บรวบรวมข้อมูล (Data Collection)

นำเข้าข้อมูลด้วยวิธีการดึงข้อมูล (Web Scraping) เพื่อให้ได้ข้อมูลที่เพียงพอและสะท้อนความคิดเห็นจริงของผู้ใช้งาน Pantip.com

3.2.1.1 เหตุผลในการเลือกใช้ Pantip.com

1. ปริมาณผู้ใช้งานสูง (High Traffic)

Pantip.com เป็นเว็บบอร์ดภาษาไทยที่มีผู้ใช้งานจำนวนมากและต่อเนื่อง (ETDA, 2019) ส่งผลให้สามารถเก็บข้อมูลที่หลากหลายทั้งด้านเนื้อหาและอารมณ์ รวมถึงติด 10 อันดับ Social Platform ที่คนไทยเลือกใช้ในการปรึกษา หาข้อมูลต่าง ๆ

2. เปรียบเทียบกับแพลตฟอร์มอื่น

ตาราง 1 เปรียบเทียบแพลตฟอร์มที่พูดคุยเรื่องข่าวตลาดหุ้น

Plattform	ข้อดี	ข้อเสีย	ความเกี่ยวข้องกับข่าวการเงิน	ข้อจำกัด
Pantip.com	- ข้อมูลเปิดเผย - แท็กช่วยคัดกรอง	- โครงสร้างเว็บมีการเปลี่ยนเป็นช่วง	สูง	- ไม่มี API
Line OpenChat	- ข้อมูลเฉพาะกลุ่ม - การแลกเปลี่ยนเชิงลึก	ไม่มี API, ข้อมูลในกลุ่มปิด	สูง (เฉพาะกลุ่ม)	- ข้อจำกัดความเป็นส่วนตัว
X (Twitter)	- ข้อมูลเรียลไทม์ - แฮชแท็กช่วยค้นหา	- ข้อความสั้น - ไม่มีความเห็นนักลงทุนมาก	สูง (เหตุการณ์)	- Rate limit - นโยบาย API
Facebook	- ข้อมูลเชิงลึกจากกลุ่ม/เพจ - แฮชแท็กช่วยค้นหา	- ข้อมูลส่วนตัว - เข้าถึงยาก (ต้องหากลุ่ม)	สูง (เฉพาะกลุ่ม)	- กฎหมายความเป็นส่วนตัว - API จำกัด

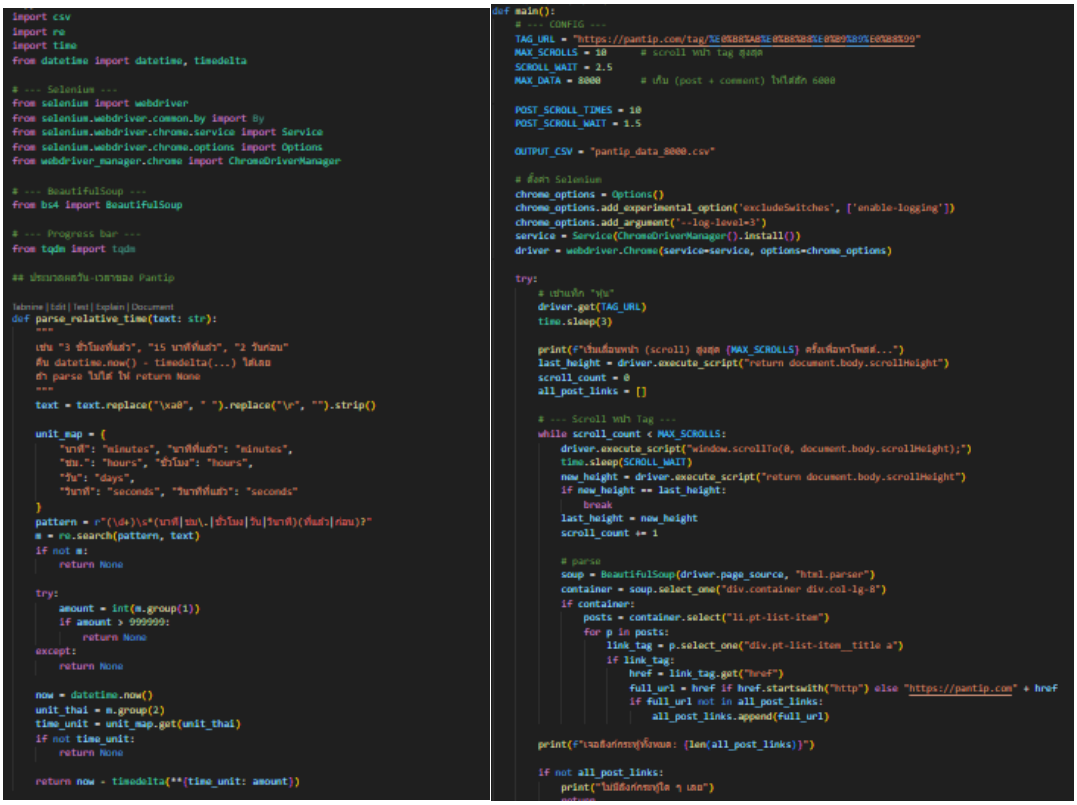
3.2.1.2 ช่วงเวลาและปริมาณข้อมูล

1. การเก็บข้อมูลในช่วง 5 วัน
   - การเก็บข้อมูลเกี่ยวกับหุ้นควรเก็บเป็นช่วงเวลาเนื่องจากทิศทางของหุ้นต้องอิงตามช่วงเวลา ไม่ใช่ช่วงวัน โดยตลาดหุ้นจะมีช่วงหยุดโดยเฉลี่ย 2 วัน/สัปดาห์
   - สอดคล้องกับการศึกษาผลกระทบข่าวระยะสั้น หรือ Short Window Event Study (MacKinlay,2022)
2. ปริมาณข้อมูล

ในงานวิจัยนี้ได้ทำการเก็บข้อมูล 8,000 samples โดยอิงจากการศึกษางานวิจัยต่าง ๆ พบว่าการเก็บข้อมูลจะอยู่ในช่วง 3,000 – 6,000 samples แต่จะมีงานวิจัยของโซนเอเชียที่มีการเก็บข้อมูลมากกว่า ซึ่งข้อมูลจะอยู่ในช่วง 6,000- 10,000 samples เนื่องจากปัญหาของภาษาที่ไม่ใช่ภาษาอังกฤษ ทำให้ต้องใช้ข้อมูลในการวิเคราะห์มากกว่าปกติ

3.2.1.3 วิธีการ Web Scraping

ใช้ภาษา Python ร่วมกับไลบรารี Selenium, BeautifulSoup และ TQDM โดยมีการควบคุมการ scroll หน้าเว็บ, การ parse วันเวลาแบบไทย และการเขียนผลลัพธ์ออกไฟล์ CSV ดังนี้



```python
import csv
import re
import time
from datetime import datetime, timedelta

# --- Selenium ---
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from webdriver_manager.chrome import ChromeDriverManager

# --- BeautifulSoup ---
from bs4 import BeautifulSoup

# --- Progress bar ---
from tqdm import tqdm

## ฟังก์ชันแปลงวัน-เวลาของ Pantip

Tabnine | Edit | Test | Explain | Document
def parse_relative_time(text: str):
    """
    เช่น "3 ชั่วโมงที่แล้ว", "15 นาทีที่แล้ว", "2 วันก่อน"
    คืน datetime.now() - timedelta(...) ให้เลย
    ถ้า parse ไม่ได้ ให้ return None
    """
    text = text.replace("\xa0", " ").replace("\r", "").strip()

    unit_map = {
        "นาที": "minutes", "นาทีที่แล้ว": "minutes",
        "ชม.": "hours", "ชั่วโมง": "hours",
        "วัน": "days",
        "วินาที": "seconds", "วินาทีที่แล้ว": "seconds"
    }
    pattern = r"(\d+)\s*(นาที|ชม\.|ชั่วโมง|วัน|วินาที)(ที่แล้ว|ก่อน)?"
    m = re.search(pattern, text)
    if not m:
        return None

    try:
        amount = int(m.group(1))
        if amount > 999999:
            return None
    except:
        return None

    now = datetime.now()
    unit_thai = m.group(2)
    time_unit = unit_map.get(unit_thai)
    if not time_unit:
        return None

    return now - timedelta(**{time_unit: amount})
```

```python
def main():
    # --- CONFIG ---
    TAG_URL = "https://pantip.com/tag/[illegible]"
    MAX_SCROLLS = 10       # scroll หน้า tag สูงสุด
    SCROLL_WAIT = 2.5
    MAX_DATA = 8000        # เก็บ (post + comment) ไฟล์สัก 6000

    POST_SCROLL_TIMES = 10
    POST_SCROLL_WAIT = 1.5

    OUTPUT_CSV = "pantip_data_8000.csv"

    # ตั้งค่า Selenium
    chrome_options = Options()
    chrome_options.add_experimental_option('excludeSwitches', ['enable-logging'])
    chrome_options.add_argument('--log-level=3')
    service = Service(ChromeDriverManager().install())
    driver = webdriver.Chrome(service=service, options=chrome_options)

    try:
        # เข้าแท็ก "ทัน"
        driver.get(TAG_URL)
        time.sleep(3)

        print(f"เริ่มเลื่อนหน้า (scroll) สูงสุด {MAX_SCROLLS} ครั้งเพื่อหาโพสต์...")
        last_height = driver.execute_script("return document.body.scrollHeight")
        scroll_count = 0
        all_post_links = []

        # --- Scroll หน้า Tag ---
        while scroll_count < MAX_SCROLLS:
            driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
            time.sleep(SCROLL_WAIT)
            new_height = driver.execute_script("return document.body.scrollHeight")
            if new_height == last_height:
                break
            last_height = new_height
            scroll_count += 1

            # parse
            soup = BeautifulSoup(driver.page_source, "html.parser")
            container = soup.select_one("div.container div.col-lg-8")
            if container:
                posts = container.select("li.pt-list-item")
                for p in posts:
                    link_tag = p.select_one("div.pt-list-item__title a")
                    if link_tag:
                        href = link_tag.get("href")
                        full_url = href if href.startswith("http") else "https://pantip.com" + href
                        if full_url not in all_post_links:
                            all_post_links.append(full_url)

        print(f"เจอลิงก์กระทู้ทั้งหมด: {len(all_post_links)}")

        if not all_post_links:
            print("ไม่มีลิงก์กระทู้ให้ ๆ เลย")
```

Full code https://github.com/Ppreawsr/Opentopics/blob/main/Test/bigdata.py

หมายเหตุด้านเทคนิค

- มีการจัดการเวลาการลงกระทู้ที่อยู่ในรูปแบบ "3 ชั่วโมงที่แล้ว" และ "6 เม.ย. 2568 เวลา 13:45 น." โดยใช้ฟังก์ชัน parse_thai_datetime(text)
- มีปริมาณข้อมูลที่ได้สามารถควบคุมโดยการกำหนด MAX_SCROLL และ MAX_DATA

3.2.1.4 ผลลัพธ์ของการ scraping ข้อมูลจากเว็บไซต์ Pantip.com

type	datetime	content
post	2025-04-07	(.....ในคืนอันมืดมิด เรายังมีความหวัง.....)
post	2025-04-07	ไทยโดนภาษีจากทรัมป์ 37% มีโอกาสทำให้คนตกงานไหมครับ
post	2025-04-07	ต้องระวังมากกว่าเดิม ใครรีบ ไปก่อนเลย
comment	2025-04-07	
comment	2025-04-07	ความคิดเห็นที่ 1ไม่ต้องรีบเจรจาก้อได้ สุดท้าย คุยกันอย่ดี สุดท้าย หุ้นเด้งกลับสู่ที่เดิม- ทน 2 ปี กำไร 1 เด้งแน่นอนตอบกลับ01ถูกใจให้พอยต์สมา
comment	2025-04-07	ความคิดเห็นที่ 2ต้องเข้าออกระดับ TF 1 นาที ; Super Bipolar...ตอบกลับ00ถูกใจให้พอยต์Canslim202017 นาทีที่แล้วร่วมแสดงความรู้สึก:ถูกใจ
comment	2025-04-07	ความคิดเห็นที่ 3Daytrade ชกลม จับแมวมือเปล่า ไม่หอบหุ้นกลับบ้าน หนุกหนานแน่ 555ตอบกลับ01ถูกใจให้พอยต์สมาชิกหมายเลข 5803662
comment	2025-04-07	ความคิดเห็นที่ 4ตอบกลับ00ถูกใจให้พอยต์Canslim202011 นาทีที่แล้วร่วมแสดงความรู้สึก:ถูกใจ0ขำกลิ้ง0หลงรัก0ซึ้ง0สยอง0ทึ่ง0
post	2025-04-07	เมื่อกี้แพทย์เจ้าของไข้เพิ่งแจ้งว่า ปู่ผมพ้นขีดอันตรายแล้ว
comment	2025-04-07	
comment	2025-04-07	ความคิดเห็นที่ 1Daytrade ชกลม จับแมวมือเปล่า ไม่หอบหุ้นกลับบ้าน หนุกหนานแน่ 555ตอบกลับ00ถูกใจให้พอยต์สมาชิกหมายเลข 5803662
comment	2025-04-07	ความคิดเห็นที่ 2ขาสั้นไม่ถูกใจในโพสนี้เลยน่ะๆๆตอบกลับ00ถูกใจให้พอยต์สมาชิกหมายเลข 779950140 วินาทีที่แล้วร่วมแสดงความรู้สึก:ถูกใจ0ขำ
post	2025-04-07	ห้ระยะยาวที่เพื่อนๆคิดว่าเก็บกินปันผลได้ยาวๆ
comment	2025-04-07	
comment	2025-04-07	ความคิดเห็นที่ 1ptt ผมรอ 25 บาท ไม่รู้จะลงมาไหมนะตอบกลับ00ถูกใจให้พอยต์สมาชิกหมายเลข 677329320 นาทีที่แล้วร่วมแสดงความรู้สึก:ถู
comment	2025-04-07	ความคิดเห็นที่ 2ลงซื้อ ขึ้นขายอย่างเดียวตอบกลับ00ถูกใจให้พอยต์สมาชิกหมายเลข 580366214 นาทีที่แล้วร่วมแสดงความรู้สึก:ถูกใจ0ขำกลิ้ง0ข
comment	2025-04-07	ความคิดเห็นที่ 3ส่วนตัวระยะยาวคง advanc แต่ยังไม่ใช่ราคานี้ตอบกลับ00ถูกใจให้พอยต์any-every12 นาทีที่แล้วร่วมแสดงความรู้สึก:ถูกใจ0ขำกลิ้
post	2025-04-07	$$฿฿ ผมอยากจะเสนอว่า ให้กลต. ประกาศปิดตลาดหุ้นไทย หยุดการซื้อขายสัก 10 วัน.. น่าจะลดความเสียหายได้มาก ฿฿$$

รูปภาพ ตัวอย่างข้อมูลที่ได้จากการ scrape ข้อมูลจาก Pantip.com

จากการเก็บข้อมูลในรูปแบบ csv ที่มี 3 คอลัมม์ได้แก่ 1. ประกอบด้วยประเภทของข้อความ (post หรือ comment) 2. เวลาในการลงโพสต์ และ 3. content เพื่อดูเนื้อหาที่ผู้คนกำลังสนใจใน ณ ขนาดนั้น โดยสามารถแบ่งประเภทของกระทู้และคอมเมนต์ ออกเป็น 6 ประเภท

Type of comment	Example	Tokenization
ปกติ	หุ้นไทยวันนี้ลบ 0.42 จุด	PyThaiNLP, deepcut
คำทับศัพท์/อังกฤษ	Stock Index กำลังวิ่งแรง	nltk, spaCy
แสลง	เม่ารับเละ	Custom dict, fastText
คำหยาบ	แม่งเอ้ย	Dictionary-based
คำผิด/เพี้ยน	ขอบคุนครับ	SymSpell, pyspellchecker
สัญลักษณ์/อีโมจิ	🔥 หุ้นพุ่ง 555	emoji, re

รูปภาพ ประเภทขอคอมเมนต์ใน Pantip.com

3.2.2 การเตรียมข้อมูล (Data Preprocessing)

นำข้อมูลจากจากWeb Scraping ทำความสะอาดและเตรียมข้อมูลให้อยู่ในรูปแบบที่เหมาะสมสำหรับการนำไปฝึกโมเดล เนื่องจากข้อมูลจาก Pantip มักมีลักษณะไม่เป็นทางการ เช่น การใช้คำซ้ำ การเว้นวรรคผิด หรือคำสแลงต่าง ๆ ก

3.2.2.1 การนำเข้าข้อมูล (Import and Setup)

ติดตั้งและเรียกใช้ไลบรารีหลักที่จำเป็นสำหรับงาน NLP และ Data Processing จากนั้นอัปโหลดไฟล์ความคิดเห็นจาก Pantip ที่เก็บไว้ในรูปแบบ CSV เพื่อนำมาใช้งาน

1. การ import ไลบรารี

```python
import pandas as pd
import re
import emoji
import unicodedata

from transformers import AutoTokenizer
from datasets import Dataset
```

2. การอัปโหลดชุดข้อมูล

```python
df = pd.read_csv('/content/dataset.csv')
print("Number of rows:", len(df))
print("Columns:", df.columns.to_list())
df.head()
```

3.2.2.1 การทำความสะอาดข้อมูล (Cleaning & Normalization)

ชุดข้อมูลนี้ประกอบด้วยคำสแลง คำสะกดผิด URL emoji HTML และอักขระพิเศษอื่น ๆ ที่ต้องลบหรือแปลงให้เป็นคำมาตรฐาน รวมถึงการ normalize unicode และเปลี่ยนเป็นตัวพิมพ์เล็กทั้งหมด โดยขั้นตอนจะเรียงลำดับดังนี้

1. Dictionary คำสแลงและคำสะกดผิด

```python
slang_dict = {
    "เม่า": "นักลงทุนรายย่อย",
    "โคตร": "มาก",
    "โคตรปัง": "ยอดเยี่ยม",
    "ปังมาก": "ยอดเยี่ยม",
    "เทพ": "เก่งมาก",
    "กาก": "แย่",
    "คูล": "เจ๋ง",
    "อวย": "ชมเชยเกินจริง",
    "ดิดดอย": "ขาดทุน",
    "มโน": "จินตนาการไปเอง",
    "งงเต้": "งงมาก",
    "จอย": "ร่วมสนุก",
    "แกง": "หลอก/ล้อ",
    "สับ": "วิจารณ์อย่างแรง",
    "ชิง": "อวด",
    "เหลา": "เล่า",
    "ม่าย": "ไม่",
    "เฟียส": "มั่นใจสุดๆ",
    "เลิศ": "ดีมาก",
}
misspell_dict = {
    "ม่าย": "ไม่",
    "ย่างงี้": "อย่างนี้",
}
```

```python
def replace_slang(text: str, slang_dictionary: dict) -> str:
    for s, std in slang_dictionary.items():
        if s in text:
            text = text.replace(s, std)
    return text

def correct_misspell(text: str, misspell_dictionary: dict) -> str:
    for wrong, right in misspell_dictionary.items():
        if wrong in text:
            text = text.replace(wrong, right)
    return text

```

2. ฟังก์ชันทำความสะอาด (Data Clean)

```python
def clean_and_normalize(text: str) -> str:
    if not isinstance(text, str):
        return ""

    # 1. ตัดช่องว่างหัว-ท้าย
    text = text.strip()

    # 2. ลบข้อความ
    text = re.sub(r'\[spoil\].*?ข้อความที่ซ่อนไว้', '', text, flags=re.IGNORECASE)

    text = re.sub(r'NaN', '', text, flags=re.IGNORECASE)

    # 3. แทนที่คำสแลง
    for slang, std in slang_dict.items():
        text = text.replace(slang, std)

    # 4. แก้คำสะกดผิด
    for wrong, right in misspell_dict.items():
        text = text.replace(wrong, right)

    # 5. ลบ URL
    text = re.sub(r'https?://\S+|www\.\S+', '', text)

    # 6. ลบ emoji
    text = emoji.replace_emoji(text, replace="")

    # 7. ลบ HTML tags
    text = re.sub(r'<.*?>', '', text)

    # 8. Normalize Unicode (NFC) - ลดสระลอย
    text = unicodedata.normalize("NFC", text)

    # 9. ลบสัญลักษณ์พิเศษ (ยกเว้นตัวอักษรไทย/อังกฤษ/ตัวเลข)
    text = re.sub(r"[^\w\sก-ฮะ-๙a-zA-Z0-9]", "", text)

    # 10. ลบช่องว่างซ้ำ
    text = re.sub(r"\s+", " ", text)

    # 11. เปลี่ยนเป็น lowercase
    text = text.lower().strip()

```

กำจัดสิ่งรบกวน (noise) และแปลงรูปแบบข้อความให้เป็นมาตรฐาน โดยฟังก์ชัน clean_and_normalize() ที่ใช้นั้นจะรวมการทำงานทั้งหมดไว้ในฟังก์ชันเดียว ดังนี้

1. ตรวจสอบข้อมูลว่าเป็นข้อความหรือไม่ หากไม่ใช่จะคืนค่ากลับเป็นสตริงว่าง
2. ลบช่องว่างหัวท้ายด้วย .strip()
3. ลบข้อความที่ไม่จำเป็น เช่น [spoil]...ข้อความที่ซ่อนไว้ หรือข้อความว่า NaN
4. แทนที่คำสแลงเป็นภาษากลาง และแก้คำสะกดผิดตามพจนานุกรมที่เตรียมไว้
5. ลบ URL ที่ขึ้นต้นด้วย http://, https:// หรือ www.
6. ลบ emoji ทั้งหมดด้วยฟังก์ชันจากไลบรารี emoji
7. ลบ HTML tag ที่อยู่ในข้อความด้วย regex <.*?>

8. แปลงข้อความให้เป็นมาตรฐาน Unicode แบบ NFC เพื่อป้องกันปัญหาสระลอยหรือการเข้ารหัสผิด
9. ลบสัญลักษณ์พิเศษที่ไม่ใช่ตัวอักษรไทย/อังกฤษ/ตัวเลข
10. ลบช่องว่างซ้ำซ้อน เช่น เว้นวรรคเกิน 1 ช่อง
11. แปลงข้อความทั้งหมดให้เป็นตัวพิมพ์เล็ก (lowercase)

3.3.2.3 การเตรียม Tokenizer สำหรับ WangchanBERTa

WangchanBERTa เป็น Pretrained Transformer Model ที่ถูกฝึกบนข้อมูลภาษาไทยจำนวนมากโดยสถาบันวิจัยปัญญาประดิษฐ์ประเทศไทย (AIResearch) โมเดลนี้ถูกออกแบบมาให้เข้าใจโครงสร้างภาษาไทยโดยเฉพาะ จึงมีความสามารถในการประมวลผลคำซ้อน คำผสม และบริบทของข้อความได้ดีโดยไม่ต้องพึ่งการตัดคำแบบดั้งเดิม เช่น PyThaiNLP

Tokenizer	จุดเด่น	ข้อจำกัด
PyThaiNLP (newmm/attacut)	ตัดคำได้แม่นในเชิงภาษา	ไม่สอดคล้องกับ vocab ของ BERT, ไม่รองรับ subword
Multilingual BERT Tokenizer	รองรับหลายภาษา	ไม่สอดคล้องกับภาษาไทยโดยเฉพาะ, precision ต่ำกว่า
WangchanBERTa Tokenizer ✅	สร้างมาคู่กับโมเดล BERT ไทย	ไม่มี

1. ผ่านการฝึกจาก corpus ขนาดใหญ่ของภาษาไทย
2. รองรับคำทับศัพท์ สแลง คำที่ใช้ในบริบทโซเชียล
3. ใช้ subword encoding ที่แม่นยำกว่า word/token level → รองรับคำใหม่หรือคำผิดได้ดีกว่า

3.3.3 Labeling

3.3.4 การสร้างและฝึกโมเดล (Model Construction & Training)

เริ่มต้นการทดลองด้วย **โมเดล Classical ML** (เช่น SVM, Logistic Regression) เพื่อเป็น baseline และถ้ามีเวลาหรือทรัพยากรเพียงพอ จะขยายสู่ **Deep Learning** (BERT/ThaiBERT) ซึ่งมีงานก่อนหน้า (Devlin et al., 2019; Medhi et al., 2020) ยืนยันว่ามีประสิทธิภาพสูงสำหรับ Sentiment Analysis ภาษาไทย แต่ต้องอาศัย GPU และข้อมูลขนาดใหญ่กว่าระดับพัน

3.3.5 การประเมินผล (Evaluation)

ประเมินด้วยชุดข้อมูลทดสอบ (Test Set) ที่แยกออกชัดเจน ชุดตัวชี้วัดที่ใช้ในงานวิจัยนี้ประกอบด้วยค่าความแม่นยำ (Accuracy), Precision, Recall, F1-Score ของทั้ง 3 คลาส (Positive, Negative, Neutral) รวมถึงดู Confusion Matrix เพื่อวิเคราะห์รายคลาส

Code in Colab :

https://colab.research.google.com/drive/1PS4ZpPOIvm9bA5_PbxZCBCPs7aML1hJ-?usp=sharing

บทที่ 4 การทดลองและผลการทดลอง/วิจัย

[เนื้อหา]

4.1[หัวข้อ]

[เนื้อหา]

4.1.1 [หัวข้อย่อย]

1. เนื้อหา
2. เนื้อหา

4.2[หัวข้อ]

[เนื้อหา]

บทที่ 5 บทสรุป

[เนื้อหา]

5.1[หัวข้อ]

[เนื้อหา]

5.1.1 [หัวข้อย่อย]

1. เนื้อหา
2. เนื้อหา

5.2[หัวข้อ]

[เนื้อหา]

เอกสารอ้างอิง