การเปรียบเทียบประสิทธิภาพของตัวแบบสำหรับการวิเคราะห์ความคิดเห็นจากบทวิจารณ์เกม
Evaluating model performance for sentiment analysis on game reviews

ณัฐภูมินทร์ มานุจำ[1], อนิรุทธ์ โชติถนอม[2], วรารัตน์ สงฆ์แป้น[3] และ อาทิตยาพร โรจรัตน์[4,*]
Natthapumin Manucham[1], Anirut Chottanom[2], Wararat Songpan[3] and Artitayaporn Rojarath[4,*]
Received: 9 December 2024 ; **Revised** 19 December 2024 ; **Accepted**: 18 April 2025

บทคัดย่อ
การศึกษาวิจัยนี้มีวัตถุประสงค์เพื่อเข้าใจความคิดเห็นของผู้เล่นที่มีต่อเกมและต้องการเปรียบเทียบให้เห็นลักษณะของตัวแบบที่ใช้ในการจำแนกประเภทโดยเป็นข้อมูลภาษาไทยในกรณีถ้ามีทรัพยากรที่จำกัด จากการประเมินประสิทธิภาพของตัวแบบ Machine learning ได้แก่ วิธีการ Naïve bayes และ Support vector machine ในส่วนของตัวแบบ Convolutional neural network model ได้แก่ 1D-CNN ซึ่งนำมาเปรียบเทียบกับตัวแบบ Transformer ได้แก่ เทคนิค BERT Multilingual และ WangchanBERTa ในการวิเคราะห์ความคิดเห็นเกี่ยวกับเกมซึ่งใช้ชุดข้อมูลที่เป็นการแสดงความคิดเห็นที่ใช้ภาษาไทยเป็นหลัก ในงานวิจัยได้ใช้เทคนิค Bag of words, TF-IDF และ Word2Vec ในการสกัดคุณลักษณะพิเศษจากข้อความสำหรับตัวแบบ Machine learning ในขณะที่ตัวแบบ Transformer ใช้การ Embedding ที่ผ่านการ pre-trained มาก่อน จากผลการทดลองแสดงให้เห็นว่า WangchanBERTa เป็นตัวแบบที่มีประสิทธิภาพดีที่สุด โดยมีค่าความถูกต้องสูงถึง 82.16% ค่า Precision ที่ 87.06% ค่า Recall ที่ 86.18% และ ค่า F1-score ที่ 86.62% ในขณะที่วิธีการ Support vector machine ที่ใช้ เทคนิค BoW เป็นตัวแบบที่แสดงผลลัพธ์ที่ดีที่สุดในกลุ่มของตัวแบบ Machine learning ด้วยค่าความถูกต้องที่ 81.26% ส่วนผลการทดลองในกลุ่มของเทคนิคการแปลงข้อความแสดงให้เห็นว่าการสกัดคุณลักษณะพิเศษจากข้อความมีความสำคัญต่อตัวแบบ Machine learning โดยจากการศึกษาพบว่าการสกัดคุณลักษณะพิเศษจากข้อความที่สนใจเฉพาะคำ เช่นเทคนิค BoW และ TF-IDF สามารถสกัดคุณลักษณะพิเศษจากข้อความได้ดีกว่าการสนใจบริบทของคำ เช่นเทคนิค Word2Vec เมื่อวิเคราะห์ถึงผลการทดลองจะเห็นว่าค่าความถูกต้องของวิธีการที่ดีที่สุดของทั้ง 2 ประเภทตัวแบบห่างกันเพียงประมาณ 1% โดยตัวแบบ Machine learning ได้ค่าความถูกต้องที่ใกล้เคียงกันกับตัวแบบ Transformer แต่ใช้ทรัพยากรน้อยกว่าจึงเหมาะสำหรับสถานการณ์ที่มีข้อจำกัดด้านทรัพยากร

คำสำคัญ: การวิเคราะห์ความรู้สึก, การเรียนรู้ของเครื่อง, การสกัดคุณลักษณะพิเศษ, การประมวลผลภาษาธรรมชาติ, บทวิจารณ์เกม

Abstract
This research aimed to examine game reviews and compare the characteristics of classification models using Thai language data, particularly in contexts with limited resources. The experiment evaluates the performance of machine

[1] นิสิตปริญญาตรี สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
[2] ผู้ช่วยศาสตราจารย์ สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
[3] รองศาสตราจารย์ วิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น
[4] อาจารย์ หน่วยวิจัยห้องปฏิบัติการมัลติเอเจนท์ ระบบอัจฉริยะ และการจำลองสถานการณ์ สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
[1] Bachelor Student, Department of Information Technology, Faculty of Informatics, Mahasarakham University
[2] Assistant Professor, Department of Information Technology, Faculty of Informatics, Mahasarakham University
[3] Associate Professor, College of Computing, Khon Kaen University
[4] Lecturer, Multi-agent Intelligent Simulation Laboratory (MISL) Research Unit, Department of Information Technology, Faculty of Informatics, Mahasarakham University
* Corresponding author E-mail: artitayaporn.r@msu.ac.th

learning models, including Naïve Bayes, support vector machines, and 1D-CNN, and compares them with transformer-based models, namely BERT Multilingual and WangchanBERTa, in analyzing game-related reviews using a dataset primarily consisting of Thai-language comments. This research employs Bag of words, TF-IDF, and word2vec techniques to generate text transformations for machine learning models and CNN model. In contrast, transformer models employ pre-trained embeddings. The experimental results indicate that WangchanBERTa achieves the highest overall performance, with an accuracy of 82.16%, a precision of 87.06%, a recall of 86.18%, and an F1-score of 86.62%. Meanwhile, the support vector machine method employing the Bag of Words technique demonstrates the best performance among the machine learning models, with an accuracy of 81.26%. The experimental findings within the text transformation methods indicate that feature extraction is a critical factor in optimizing the performance of machine learning models. The study reveals that feature extraction methods focusing exclusively on individual words, such as Bag-of-Words and TF-IDF, demonstrate greater effectiveness in feature extraction compared to context-aware approaches such as word2vec. An analysis of the experimental results reveals that the accuracy of the best-performing models in both categories differs by approximately 1%. The machine learning models achieve accuracy levels comparable to those of the transformer models while requiring fewer resources, making them well-suited for resource-constrained environments.

Keywords: Sentiment analysis, machine learning, feature extraction, natural language processing, game reviews

บทนำ

ในปัจจุบัน ประเทศไทยมีผู้เล่นเกมผ่านคอมพิวเตอร์ส่วนตัว (PC) ประมาณ 16.3 ล้านคน ซึ่งประมาณหนึ่งในสามของผู้เล่นเกมเหล่านี้เป็นผู้หญิง (Gimme, 2018) นอกจากนี้ ข้อมูลจาก Marketeer Online ยังแสดงให้เห็นว่าอุตสาหกรรมเกมในประเทศไทยมีการใช้จ่ายเงินเล่นเกมเฉลี่ยปีละ 2,200 บาทต่อคน (Eukeik, 2011) สำหรับการอัตราการเติบโตของผู้ชมอีสปอร์ตในประเทศไทย มีการคาดการณ์ว่าจะเพิ่มขึ้นประมาณ 30% ระหว่างปี 2560 ถึง 2564 นี่แสดงให้เห็นถึงการเติบโตที่รวดเร็วของตลาดเกมในประเทศไทย ไม่เพียงแต่ในด้านของการเล่นเกมเท่านั้น แต่ยังรวมถึงด้านการชมการแข่งขันอีสปอร์ตด้วย เนื่องด้วยการเติบโตของตลาดเกมในประเทศไทย ทำให้แพลตฟอร์มจำหน่ายเกมชื่อดังอย่าง Steam มีผู้ใช้งานจากประเทศไทยมากขึ้นในทุกๆ ปี ตามข้อมูลในปี 2024 แพลตฟอร์มเกม Steam เป็นแพลตฟอร์มร้านค้าออนไลน์สำหรับตลาดเกม PC ในการจัดจำหน่ายจากค่ายเกมต่างๆ ให้สามารถเข้าถึงกันได้ทั่วโลก มีผู้ใช้งานในประเทศไทยประมาณ 1.2 ล้านคน ซึ่งสะท้อนถึงวัฒนธรรมการเล่นเกมที่แข็งแกร่งในประเทศ และในระดับโลก (Wresearch, 2022) Steam ยังคงเติบโตอย่างต่อเนื่องโดยมีจำนวนผู้ใช้งานที่เพิ่มขึ้น ซึ่งจำนวนผู้ใช้งานพร้อมกัน 31 ล้านคนทั่วโลกในปี 2024 และ Steam มีเกมให้เลือกมากมายโดยมีเกมให้เล่นเกือบ 50,361 เกม และมีเกมใหม่ๆ ที่กำลังเปิดตัวในปี 2024 (Shewale, 2024) ด้วยการที่มีเกมให้เล่นมากมายนี้ ทำให้ผู้บริโภคส่วนใหญ่จะทำการสืบค้นข้อมูลเกี่ยวกับตัวเกม เช่น ประเภทของเกม, คุณภาพของเกม, ราคาของเกม และปัจจัยที่สำคัญที่ส่งผลกระทบต่อการเลือกซื้อเกมของผู้บริโภคนั้นก็คือ ข้อมูลการแสดงความคิดเห็น หรือบทวิจารณ์จากผู้ที่เคยซื้อไปแล้ว Steam จะมีการเปิดให้มีการเขียนบทวิจารณ์เกมจากผู้ซื้อต่อตัวเกมดังกล่าว ทำให้ผู้ที่สนใจที่จะซื้อตัวเกมนั้นสามารถเข้าไปอ่านเพื่อประกอบการตัดสินใจ ซึ่งบทวิจารณ์จากผู้ซื้อนั้นเป็นการแสดงความรู้สึกในรูปแบบข้อความที่มีต่อตัวเกมพร้อมกับการให้คะแนนความนิยมเพื่อสรุปว่าแนะนำหรือไม่แนะนำให้ซื้อเกม จากกรณีนี้มักพบปัญหาที่เกิดขึ้น คือบทวิจารณ์นั้นและคะแนนที่ให้มีความขัดแย้งกันจากการวิจารณ์แนวประชดประชัน การใช้ถ้อยคำเสียดสี และความคลุมเครือในการใช้ภาษา ทำให้สร้างความสับสนทั้งผู้ซื้อที่ต้องการข้อมูลเพื่อประกอบการตัดสินใจซื้อและนักพัฒนาเกมที่จะนำข้อมูลบทวิจารณ์เหล่านี้ไปประกอบการปรับปรุงในตัวเกม

Bais *et al*. (2017) ศึกษาเกี่ยวกับการจำแนกความรู้สึกของรีวิวเกมบน Steam โดยการจำแนกความรู้สึกในข้อความ มีความน่าสนใจโดยเฉพาะการใช้ภาษาที่มีความซับซ้อน เช่น การใช้ภาษาที่แสดงถึงการเสียดสี, การใช้ภาษาอังกฤษที่ไม่ถูกหลักไวยากรณ์ และความรู้สึกที่ผสมผสามกัน ในงานวิจัยนี้มีการจำแนกข้อความที่เป็นเชิงบวกและเชิงลบโดยพิจารณ์จากภาษาเขียนของผู้ใช้ Steam ผู้วิจัยได้มีการใช้อัลกอริทึม ได้แก่ Support Vector Machine (SVM), Logistic Regression, Naïve Bayes และTurney Algorithm ซึ่งข้อมูลทดสอบเป็นแบบ Unsupervised Sentiment นอกจากนี้ยังพิจารณาลักษณะที่สำคัญอื่นๆ ในการวิเคราะห์ความรู้สึก เช่น

เปอร์เซ็นของผู้ที่พบว่ารีวิวมีประโยชน์หรือตลก และจำนวนชั่วโมงที่ผู้รีวิวเล่นเกมที่รีวิว ผลการทดลองพบว่า Logistic Regression มีค่าความถูกต้องสูงที่สุด ที่ 93.5% รองลงมาคือ Linear SVM ที่ค่าความถูกต้อง 93.1%

วนัสวรรณ มีประเสริฐ และเอกรัฐ รัฐกาญจน์ (2564) ได้ศึกษาเกี่ยวกับการวิเคราะห์ความคิดเห็นจากทวิตเตอร์ของลูกค้าบริษัทช้อปปี้ประเทศไทย โดยใช้เทคนิคการเรียนรู้ของเครื่อง (Machine Learning) โดยใช้ Random Forest, SVM และ Logistic Regression สำหรับการจำแนกหัวข้อ ใช้ Model ที่มีผู้พัฒนาไว้แล้วในการแยกความรู้สึก คือ Wangchan BERTa และมีการคัดเลือกคุณลักษะด้วย Bag of Words และ TF-IDF สำหรับการจำแนกความรู้สึก ซึ่งให้ประสิทธิภาพที่ดีที่สุดสำหรับข้อความภาษาไทย จากการวิจัยพบว่าวิธีการวิเคราะห์ข้อมูลเพื่อจัดหมวดหมู่ประเด็นปัญหาที่มีประสิทธิภาพมากที่สุดคือวิธี Random Forest ส่วนวิธีการวิเคราะห์ข้อมูลที่มีประสิทธิภาพมากที่สุดเพื่อวิเคราะห์ความรู้สึกแต่ละประเด็นสำคัญ คือ WangchanBERTa

ปราชญภาคย์ เหล่าสังข์สุข และคณะ (2560) ในงานวิจัยนี้เสนอระบบวิเคราะห์และสรุปความคิดเห็นเกี่ยวกับร้านอาหารจากเว็บไซต์รีวิวโดยอัตโนมัติที่ใช้เทคนิคการตัดคำในประโยคโดยอาศัยฐานคำศัพท์ การวิเคราะห์ประเภทคำและรูปประโยค เพื่อหาความหมายเชิงบวกหรือเชิงลบของประโยค และนำมาคำนวณเป็นค่าคะแนนความพึงพอใจสำหรับปัจจัยด้านการบริการต่างๆ ซึ่งจากผลการประเมินความพึงพอใจของผู้ใช้ในด้านความสอดคล้องของผลสรุปจากระบบและผลสรุปจากผู้ใช้พบว่าการสรุปความคิดเห็นด้านบริการและอาหารอยู่ในระดับดี ขณะที่ด้านอื่นๆ อยู่ในระดับพอใช้

รวิสุดา เทศเมือง และนิเวศ จิระวิชิตชัย (2560) นำเสนอวิธีการจำแนกความคิดเห็นโดยการวิเคราะห์ความคิดเห็นภาษาไทย เกี่ยวกับการรีวิวสินค้าออนไลน์ ด้านการบริการห้องพัก โรงแรม รีสอร์ท จาก Agoda Thailand และ Twitter Thailand โดยใช้เทคนิคเหมืองข้อความวิเคราะห์ความคิดเห็นภาษาไทยเกี่ยวกับการรีวิวการบริการห้องหัก และสร้างแบบจำลองด้วยอัลกอริทึม 4 วิธี ได้แก่ ซัพพอร์ตเวกเตอร์แมทชีน ต้นไม้ตัดสินใจ นาอีฟเบย์ และเคเนียเรสเนเบอร์ เพื่อเปรียบเทียบประสิทธิภาพการวิเคราะห์ความคิดเห็นภาษาไทยเกี่ยวกับการรีวิวบริการห้องพัก มีการใช้การแทนค่าดัชนีด้วยค่า TF-IDF และลดคุณลักษณะด้วยค่า ChiSquare พบว่า อัลกอริทึมซัพพอร์ทเวกเตอร์แมชชีนให้ประสิทธิภาพในการจำแนกดีที่สุดที่ 83.38%

วสวัตติ์ อินทร์แปลง และจารี ทองคำ (2563) งานวิจัยนี้มีวัตถุประสงค์เพื่อค้นหาเทคนิคการจำแนก จาก 5 เทคนิคที่มีประสิทธิภาพ คือ เทคนิค Naïve Bayes, SVM, K-Nearest Neighbor เทคนิคต้นไม้ตัดสินใจ C4.5 และเทคนิค Random Forest โดยเก็บรวบรวมข้อมูลความคิดเห็นต่อเกมมือถือผับจี จำนวน 3,798 ข้อความ ในกระบวนการคัดเลือกคำาบ่งชี้เพื่อใช้ในการแยกคุณลักษณะได้เลือกใช้คำาวิเศษณ์ และคำาสแลงบางคำาที่ความหมายของคำาเป็นคำาวิเศษณ์เพื่อทำการแยกคุณลักษณะเชิงบวกและเชิงลบ งานวิจัยยังมีการปรับความสมดุลของข้อมูลด้วยวิธี SMOTEและใช้หลักการ 10-fold cross validation ในการแบ่งกลุ่มข้อมูลเป็นชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบ เมื่อทำการทดสอบและวัดประสิทธิภาพของตัวแบบพบว่า เทคนิค K-Nearest Neighbor ให้ผลดีที่สุดในการวิเคราะห์ความคิดเห็น โดยให้ค่าความแม่นยำา 99.75% ค่าความระลึก 100% และค่าความถูกต้อง 99.87%

การศึกษาครั้งนี้ผู้วิจัยมีแนวคิดที่จะสร้างตัวแบบการวิเคราะห์ความรู้สึกทางอารมณ์ สำหรับจำแนกบทวิจารณ์จากผู้ซื้อ โดยใช้เทคนิค Machine Leaning แบบการเรียนรู้แบบ Supervised Leaning ส่วนข้อมูลที่ใช้ คือบทวิจารณ์เกมจากแพลตฟอร์ม steam ที่เป็นภาษาไทย เพื่อให้การวิเคราะห์มีความแม่นยำ ข้อมูลที่ใช้ในการฝึกตัวแบบถูกแปลงเป็นคุณลักษณะหลายประเภท ได้แก่ Bag of Words, TF-IDF และ Word Embedding (Word2Vec) ซึ่งเป็นวิธีการที่ช่วยในการแทนค่าคำในลักษณะที่สามารถนำไปใช้ในการฝึกตัวแบบได้อย่างมีประสิทธิภาพ ในการศึกษาได้เลือกใช้เทคนิคการเรียนรู้ของเครื่องที่หลากหลายเพื่อเปรียบเทียบผลลัพธ์ เช่น ตัวแบบ Machine Learning อย่าง Naïve Bayes, SVM และ 1D-CNN (One-dimensional Convolutional Neural Network) ซึ่งเป็นเทคนิคที่มีความหลากหลายในการจัดการกับข้อมูลที่มีลักษณะต่างๆ นอกจากนี้ยังได้ใช้ตัวแบบ Transformer อย่าง BERT Multilingual และ WangchanBERTa ซึ่งถูกออกแบบมาเพื่อจัดการกับข้อมูลภาษาธรรมชาติในหลายภาษาและโดยเฉพาะสำหรับภาษาไทย

งานวิจัยฉบับนี้ จะนำเสนอการวิเคราะห์ความรู้สึกจากบทวิจารณ์เกมบน Steam โดยมุ่งเน้นที่การรับรู้และทำนายความรู้สึกหรือทัศนคติที่ปรากฏในบทวิจารณ์ โดยใช้เทคนิคด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) การวิเคราะห์ความรู้สึก (Sentiment Analysis) และเทคนิคการจำแนกข้อความ (Text Classification) เพื่อสร้างตัวแบบที่เหมาะสมสำหรับการวิเคราะห์ความรู้สึกจากบทวิจารณ์เกมบน Steam เนื่องจากบทวิจารณ์เกมมักจะเต็มไปด้วยคำแสลง คำหยาบ หรือคำเฉพาะกลุ่มที่เกี่ยวข้องกับการเล่นเกม ซึ่งทำให้การวิเคราะห์และการทำนายความรู้สึกเป็นเรื่องที่ท้าทาย ดังนั้นจึงจำเป็นต้องพัฒนาและ

ปรับแต่งตัวแบบการวิเคราะห์ความรู้สึกนี้เพื่อให้สามารถจัดการกับความหลากหลายของภาษาที่พบในบทวิจารณ์เกมได้อย่างมีประสิทธิภาพ

ในงานวิจัยมีส่วนที่เป็น Research Contribution ดังต่อไปนี้

1) ในงานวิจัยมีการเปรียบเทียบให้เห็นประสิทธิภาพของตัวแบบ 3 ประเภทในการวิเคราะห์ความรู้สึกกับความคิดเห็นที่เป็นภาษาไทย ซึ่งประกอบด้วย machine learning model, CNN model และ transformer model

2) ขั้นตอนของการเตรียมข้อมูลมีกระบวนการในจัดการข้อมูลที่เป็น imbalanced data ด้วยวิธี oversampling ซึ่งทำให้ประสิทธิภาพด้านความถูกต้องของตัวแบบทั้ง 3 ประเภทเพิ่มขึ้น

3) เปรียบเทียบวิธีการสกัดคุณลักษณะพิเศษที่จะนำไปใช้งานร่วมกับตัวแบบ machine learning โดยคุณลักษณะพิเศษที่สนใจจำนวนคำที่ปรากฏในเอกสาร เช่น BoW และ TF-IDF และคุณลักษณะพิเศษที่สนใจบริบทของคำ เช่น Word2Vec

วิธีการดำเนินการวิจัย

ขั้นตอนการดำเนินการจะประกอบไปด้วย 5 ส่วนที่สำคัญ ดังแสดงที่ Figure 1 ซึ่งมีรายละเอียดดังนี้

1. การรวบรวมข้อมูล (Data collection)

การวิเคราะห์ความรู้สึกจากบทวิจารณ์เกมบนสตรีม ใช้ข้อมูลจากเว็บไซต์ Kaggle (https://www.kaggle.com/datasets/boatlnwza/steam-review-thai) ซึ่งเป็นข้อมูลการแสดงความคิดเห็นเกี่ยวกับเกม โดยชุดข้อมูลมีจำนวน 45,891 แถวแบ่งเป็น 2 คลาส ได้แก่ คลาส True ที่แสดงถึงความคิดเห็นเชิงบวก มีจำนวน 31,238 แถว และคลาส false ที่แสดงถึงความคิดเห็นเชิงลบ มีจำนวน 14,653 แถว แสดงดังตัวอย่างที่ Table 1 โดยที่ชุดข้อมูลทั้งหมดยังไม่ได้ผ่านการเตรียมข้อมูล

2. การเตรียมข้อมูล (Preprocessing Data)

2.1 การทำความสะอาดข้อความ (Text Cleaning)

เป็นขั้นตอนที่ช่วยจัดการและปรับปรุงข้อความให้อยู่ในรูปแบบที่เหมาะสมสำหรับการนำไปใช้วิเคราะห์หรือสร้างตัวแบบ โดยเริ่มต้นจากการลบข้อความที่มีอีโมจิ เนื่องจากอีโมจิมักจะไม่สอดคล้องกับการวิเคราะห์เนื้อหา

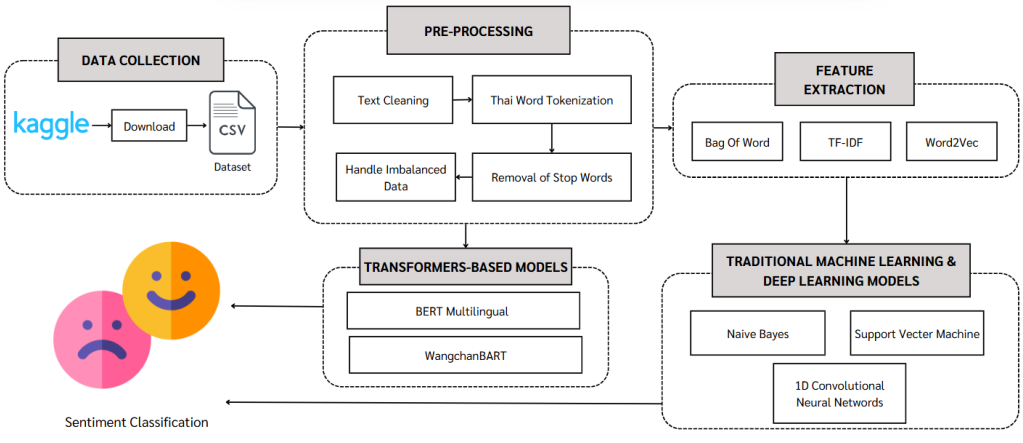

Figure 1 Framework of the sentiment analysis process based on game reviews on Steam

นอกจากนี้ยังเลือกเก็บไว้เฉพาะตัวอักษรโดยการลบสัญลักษณ์ หรือตัวอักษรที่ไม่ใช่ภาษาไทยและภาษาอังกฤษออกเพื่อให้ข้อมูลเป็นไปในรูปแบบที่ถูกต้องชัดเจนยิ่งขึ้น และคำภาษาอังกฤษจะเปลี่ยนเป็นตัวพิมพ์เล็กทั้งหมดเพื่อให้อยู่ในรูปแบบเดียวกัน ดังแสดงตัวอย่างที่ Table 2

หลังจากการทำความสะอาดข้อความ หากในคอลัมน์ของข้อมูลที่ทำความสะอาดแล้วมีข้อความใดที่ว่างเปล่าไม่มีข้อมูล ข้อมูลแถวนั้นจะถูกลบออกซึ่งมีจำนวนทั้งหมด 404 แถว ข้อมูลจะเหลือทั้งหมด 45,487 แถว จากการลบแถวที่ว่าง 404 แถว โดยแบ่งเป็น 2 คลาส ได้แก่ true ที่แสดงถึงความคิดเห็นเชิงบวก มีจำนวน 30,910 แถว และ false ที่แสดงถึงความคิดเห็นเชิงลบ มีจำนวน 14,577 แถว แสดงดัง Figure 2

2.2 การตัดคำ (Word Tokenization)

การตัดคำเป็นกระบวนการสำคัญในงานประมวลผลภาษาธรรมชาติ โดยมีเป้าหมายในการแยกข้อความออกเป็นหน่วยย่อยหรือ “Token” ซึ่งแต่ละ Token มักจะเป็นคำหรือวลีที่มีความหมายในตัวเอง แต่สำหรับภาษาไทยซึ่งไม่มีการเว้นวรรคระหว่างคำ กระบวนการนี้จึงมีความซับ

ซ้อนมากขึ้น จำเป็นต้องใช้อัลกอริทึมเฉพาะในการแยกคำออกจากกัน โดยมีเทคนิคที่น่าสนใจ 3 เทคนิค ได้แก่ newmm, multi-cut และ deepcut ดังแสดงใน Table 3 จากตารางแสดงให้เห็นการเปรียบเทียบการตัดคำระหว่าง newmm, multi-cut และ deepcut แต่ละเทคนิคมีจุดเด่นต่างกัน โดยเทคนิคnewmm เหมาะกับการแยกคำที่พบในรูปแบบทั่วไปและ

คำที่ตรงกับพจนานุกรม มีการแยกคำค่อนข้างครบ แต่บางครั้งอาจไม่เหมาะสำหรับรีวิวที่มีการใช้คำผสมไทยและอังกฤษหรือคำที่ไม่ได้อยู่ในพจนานุกรม เช่น “2D” เทคนิค multi-cut มีความสามารถแยกคำที่รวมกันเป็นวลีได้ เช่น “เกมดีมากครับ” ทำให้เห็นเนื้อหาโดยรวมได้ดีขึ้น แต่จะมีปัญหากับการแยกคำเชิงละเอียด ซึ่งอาจจำเป็นในการวิเคราะห์รีวิวที่เน้นไปที่คำแสดงอารมณ์หรือคำแสดงความรู้สึกเชิงบวกและลบ เทคนิค deepcut สามารถแยกคำละเอียดได้มากที่สุด โดยเฉพาะประโยคที่ผสมระหว่างคำภาษาไทยและภาษาอังกฤษที่ปะปนกัน ช่วยให้สามารถจับคำในเชิงรายละเอียดได้ดี เหมาะกับงานที่ต้องการข้อมูลแบบคำต่อคำเพื่อนำไปวิเคราะห์ต่อ

โดยสรุปหากต้องการการวิเคราะห์รีวิวเกมหรือความคิดเห็นที่ต้องการความละเอียดสูงในเชิงอารมณ์ และมีการผสมคำภาษาไทยและภาษาอังกฤษ จึงเหมาะสมที่จะเลือกใช้เทคนิค deepcut เทคนิค deepcut สามารถแยกคำได้ละเอียดโดยเฉพาะคำภาษาไทย เช่น “ดี”, “มาก” และ “ครับ” ที่ตัดออกมาเป็นคำเดี่ยว ซึ่งทำ

Table 1 Example of game reviews.

Review	Recommended
เกมดีมากครับ เล่นได้ยาวๆ เล่นกับเพื่อนสนุกมาก	True
สนุกมากๆแนะนำเลย	True
โคตรดี โคตรมันส์ โคตรเพลิน	True
โลกกว้างก็จริง แต่ไม่ได้รู้สึกตื่นตากับอะไรเลย	False
โคตรของโคตรของโคตรบัค ทั้งบัคทั้งหน่วง	False

Table 2 Text cleaning example.

Text	Text cleaning
เกมดีมากครับ เล่นได้ยาวๆ เล่นกับเพื่อนสนุกมาก	เกมดีมากครับ เล่นได้ยาว เล่นกับเพื่อนสนุกมาก
ผู้พัฒนาแย่มากที่ปล่อย DLC ที่เสียหายอยู่เสมอ	ผู้พัฒนาแย่มากที่ปล่อย dlc ที่เสียหายอยู่เสมอ
เป็นเกมที่ดี ฝึกความ Creative สมเป็นเกมต้นแบบ 2D	เป็นเกมที่ดี ฝึกความ creative สมเป็นเกมต้นแบบ 2d
เกมไม่ดี!!!!!!!!!	เกมไม่ดี
	null

ให้สามารถจับคำแสดงอารมณ์หรือคำคุณศัพท์ได้ดีกว่า นอกจากนี้ deepcut สามารถตัดคำผสมภาษาไทยและภาษาอังกฤษได้ดี เช่น “2d” ซึ่งมีความสำคัญในรีวิวเกมที่อาจมีคำศัพท์เฉพาะในภาษาอังกฤษ การแยกคำออกมาได้ชัดเจนทำให้การวิเคราะห์ในขั้นถัดไปแม่นยำขึ้น

2.3 การลบคำหยุด (Stop words removal)

การลบคำหยุดเป็นกระบวนการที่เป็นส่วนสำคัญในการประมวลผลข้อความ เพื่อลดขนาดของข้อมูลและเพิ่มประสิทธิภาพในการทำนาย ทำให้สามารถเน้นคำที่สำคัญและมีความหมายมากขึ้น โดยใน stop word list มีทั้งหมด 1030 คำ ซึ่งจะแบ่งออกเป็น 1) คำสรรพนาม เช่น “ฉัน”, “คุณ”, “เธอ” 2) คำเชื่อม เช่น “และ”, “กับ”, “แต่” 3) คำบุพบท เช่น “ใน”, “บน”, “ที่” 4) คำบ่งบอกเวลา เช่น “เมื่อ”, “ก่อน”, “หลัง” 5) คำช่วยหรือคำบอกอารมณ์ เช่น “ครับ”, “ค่ะ”, “นะ” 6) คำแสดงความเป็นเจ้าของ เช่น “ของ”, “นั้น”, “นี้” และ 7) คำทั่วไปที่ไม่จำเป็นต่อความหมายหลัก เช่น “แล้ว”, “ก็”, “ด้วย” เป็นต้น ดังแสดงตัวอย่างที่ Table 4

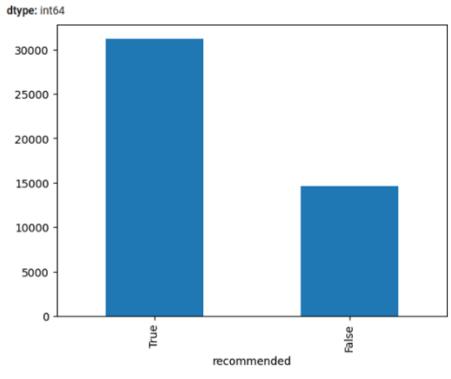

Figure 2 Graph comparing the number of instances in both classes after text cleaning

เมื่อทำการลบคำหยุดจะทำให้บางแถวเป็นค่าว่าง ซึ่งมีจำนวนทั้งหมด 1,908 แถว จากนั้นจะทำการลบแถวที่เป็นค่าว่างนั้นทิ้ง ชุดข้อมูลจะเหลือทั้งหมด 43,579 แถว โดยมีแถวที่เป็นคลาส true ที่แสดงถึงความคิดเห็นเชิงบวก มีจำนวน 29,402 แถว และคลาส false ที่แสดงถึงความคิดเห็นเชิงลบ มีจำนวน 14,177 แถว

2.4 การแบ่งข้อมูล (Data splitting)

กระบวนการนี้เริ่มจากการนำข้อมูลทั้งหมดแบ่งออกเป็นชุดฝึกสอน (train) 70% และชุดทดสอบ (test) 30% หลังจากการแบ่งข้อมูลจากชุดข้อมูล 43,579 แถวจะได้ชุด train 70% จำนวน 30,505 แถว โดยแบ่งเป็นคลาส true จำนวน 20,640 แถว และคลาส false จำนวน 9,865 แถว และ

ชุด test 30% จำนวน 13,074 แถว โดยแบ่งเป็นคลาส true จำนวน 8,762 แถว และคลาส false จำนวน 4,312 แถว ดังแสดง Table 5

2.5 การจัดการข้อมูลที่ไม่สมดุล (Improving imbalanced data)

การจัดการข้อมูลที่ไม่สมดุลเป็นกระบวนการที่สำคัญต่อ machine learning model เนื่องจากข้อมูลที่ไม่สมดุลอาจทำให้ตัวแบบเรียนรู้ข้อมูลไม่เต็มที่และมีแนวโน้มที่จะคาดการณ์ไปยังคลาสที่มีตัวอย่างมากกว่า ส่งผลให้ความแม่นยำของการทำนายลดลง โดยเทคนิคที่นิยมใช้ในการแก้ไขปัญหานี้คือ Random Oversampling (ROS) และ Random Undersampling (RUS) (Rojarath *et al*., 2024)

วิธีการ ROS เป็นวิธีการเพิ่มจำนวนตัวอย่างข้อมูลในคลาสที่มีจำนวนน้อย โดยการคัดลอกตัวอย่างที่มีอยู่แล้วในคลาสนั้นมาเพิ่ม ทำให้คลาสที่มีจำนวนน้อยมีจำนวนตัวอย่างเพิ่มขึ้นจนใกล้เคียงกับคลาสที่มีจำนวนมาก วิธีการนี้ช่วยให้ตัวแบบสามารถเรียนรู้จากตัวอย่างที่มากขึ้นจากคลาสที่มีจำนวนน้อย ซึ่งจะช่วยลดความลำเอียง (bias) ของตัวแบบต่อคลาสที่มีจำนวนมาก ในงานวิจัยมี training dataset จำนวน 30,505 แถว เป็นคลาส true จำนวน 20,640 แถว

Table 3 Examples of word tokenization

Word segmentation techniques	Word segmentation
newmm	['เกม', 'ดีมาก', 'ครับ', 'เล่น', 'ได้', 'ยาว', 'เล่น', 'กับ', 'เพื่อน', 'สนุก', 'มาก']
	['เป็น', 'เกม', 'ที่', 'ดี', 'ฝึก', 'ความ', 'creative', 'สม', 'เป็น', 'เกม', 'ต้น', 'แบบ', '2', 'd']
Multi_cut	['เกมดีมากครับ', 'เล่น', 'ได้', 'ยาว', 'เล่น', 'กับ', 'เพื่อน', 'สนุก', 'มาก']
	['เป็น', 'เกมที่', 'ดี', 'ฝึก', 'ความ', 'creative', 'สม', 'เป็น', 'เกมต้น', 'แบบ', '2', 'd']
deepcut	['เกม', 'ดี', 'มาก', 'ครับ', 'เล่น', 'ได้', 'ยาว', 'เล่น', 'กับ', 'เพื่อน', 'สนุก', 'มาก']
	['เป็น', 'เกม', 'ที่', 'ดี', 'ฝึก', 'ความ', 'creative', 'สม', 'เป็น', 'เกม', 'ต้น', 'แบบ', '2d']

คลาส false จำนวน 9,865 แถว เมื่อใช้ ROS แล้วชุดฝึกสอนจะมีจำนวน 41,280 แถว โดยมีคลาส true จำนวน 20,640 แถว และ คลาส false จำนวน 20,640 แถว เท่ากัน ดังแสดงที่ Figure 3 วิธีการ RUS เป็นวิธีการลดจำนวนตัวอย่างในคลาสที่มีจำนวนมากลง โดยการสุ่มลบตัวอย่างบางส่วนออกจากคลาสที่มีจำนวนมาก เพื่อให้จำนวนข้อมูลในคลาสนั้นมี

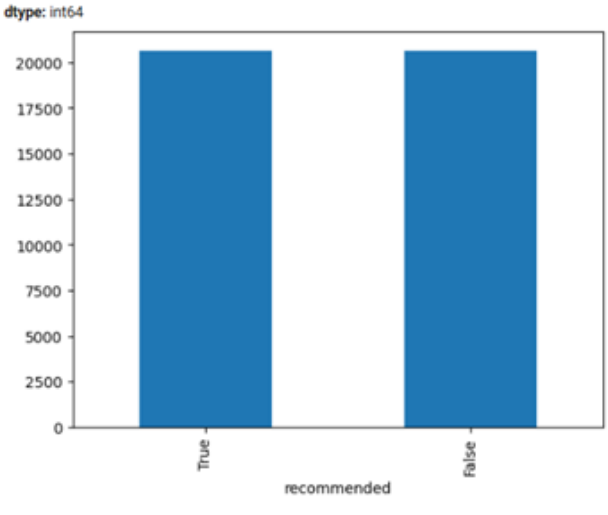

Figure 3 Graph after performing Random Oversampling (ROS) technique

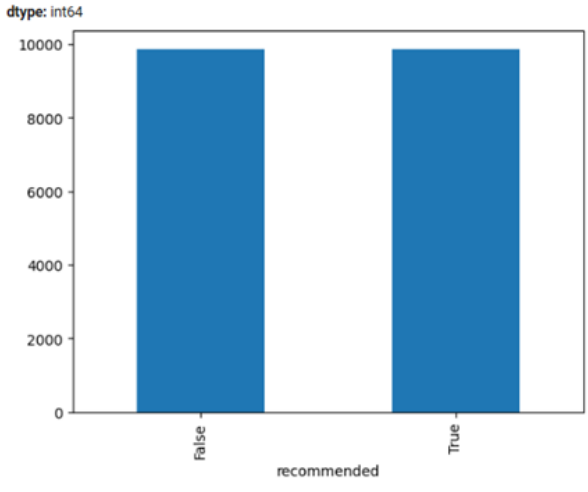

Figure 4 Graph after performing Random Undersampling (RUS) technique

ความสมดุลกับคลาสที่มีจำนวนน้อย วิธีนี้มักใช้เมื่อมีจำนวนข้อมูลในคลาสใดคลาสหนึ่งมากเกินไปและส่งผลให้ขนาดข้อมูลทั้งหมดใหญ่เกินไป ซึ่งอาจทำให้การประมวลผลและการฝึกตัวแบบใช้เวลานาน การลดขนาดข้อมูลด้วยวิธีการ RUS จะทำให้กระบวนการฝึกฝนตัวแบบเร็วขึ้นและลดการใช้ทรัพยากรจำนวนมากในการประมวลผล จากชุดฝึกสอนจำนวน 30,505 แถว โดยมีคลาส true จำนวน 20,640

แถว และ คลาส false จำนวน 9,865 แถว เมื่อใช้ RUS แล้ว ชุดฝึกสอนจะมีจำนวน 19,730 แถว โดยมีคลาส true จำนวน 9,865 แถว และ คลาส false จำนวน 9,865 แถวเท่ากัน ดัง Figure 4

3. การสกัดคุณลักษณะพิเศษ (Feature Extraction)

3.1 การสร้างคลังคำศัพท์ (Bag of Words)

โดยใช้ฟังก์ชัน CountVectorizer จากไลบรารี scikit-learn ซึ่งจะทำการแปลงข้อความในแต่ละเอกสารให้อยู่ในรูปแบบของ Bag of Words โดยการนับจำนวนคำที่ปรากฏในแต่ละเอกสาร กระบวนการนี้ไม่สนใจลำดับของคำหรือความหมายในเชิงไวยากรณ์ แต่สนใจว่าคำใดปรากฏและปรากฏกี่ครั้งในแต่ละเอกสาร โดยมีคำที่พบมากที่สุด 10 อันดับ ดังแสดงที่ Figure 5

3.2 การคัดเลือกคุณลักษณะด้วย TF-IDF (Term Frequency-Inverse Document Frequency)

ใช้ฟังก์ชัน TfidfVectorizer จากไลบรารี scikit-learn เป็นเทคนิคที่ใช้การประเมินความสำคัญของคำศัพท์ในชุดข้อมูลที่เป็น text โดยการให้ค่าน้ำหนักแก่คำศัพท์ตามความถี่ของคำนั้นๆ ในเอกสารแต่ละรายการ

Table 4 Examples of stop word removal.

words	words + stop words
[เกม, ดี, มาก, ครับ, เล่น, ได้, ยาว, เล่น, กับ, เพื่อน, สนุก, มาก]	[เกม, ดี, เล่น, เล่น, เพื่อน, สนุก]
[สนุก, มาก, เกม, ก็, เล่น, ลื่น, แต่, คือ, ยัง, ไง, อยาก, เล่น, ต่อ, นะ, ขอร้อง, ล่ะ]	[สนุก, เกม, เล่น, ลื่น, เล่น, ขอร้อง]

Table 5 Data splitting.

Dataset	Instances	True	False
Training set	30,505	20,640	9,865
Test set	13,074	8,762	4,312

และยังคำนวณค่าน้ำหนักให้แก่คำศัพท์ที่มีความสำคัญสูงในเอกสารทั้งหมดของชุดข้อมูล

Term Frequency (TF) เป็นค่าที่บอกความถี่ของคำศัพท์ที่ปรากฏในเอกสาร ดังสมการที่ 1

$$TF(t,d) = \frac{\text{จำนวนครั้งที่คำศัพท์ } t \text{ ปรากฏในเอกสาร } d}{\text{จำนวนคำทั้งหมดในเอกสาร } d} \quad (1)$$

Inverse Document Frequency (IDF) เป็นการคำนวณค่าน้ำหนักที่แสดงถึงความถี่ในการใช้คำหนึ่งคำโดยคำที่ใช้บ่อยในเอกสารจะมีค่า IDF ต่ำ ซึ่งหมายถึงคำนั้นไม่สำคัญมากในเชิงความหมายของเอกสารนั้นๆ ส่วนคำที่ถูกนำมาใช้น้อยในเอกสารจะมีค่า IDF สูง ดังสมการที่ 2

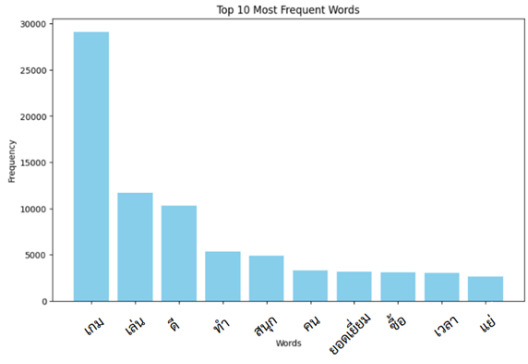

Figure 5 Top 10 most frequent words

$$IDF(t,D) = \log\left(\frac{\text{จำนวนเอกสารทั้งหมดในชุดข้อมูล } D}{\text{จำนวนเอกสารที่มีคำศัพท์ } t \text{ ปรากฏในนั้น}}\right) \quad (2)$$

ดังนั้น นำการคำนวณทั้งสองส่วนมารวมกัน จะได้การคำนวณหาค่า TF-IDF โดยจะนำค่า TF กับค่า IDF มาคูณเข้าด้วยกัน ซึ่งเป็นการคัดแยกคำตามความสำคัญโดยการให้น้ำหนักคำในแต่ละคำในเอกสารแต่ละรายการ ดังสมการที่ 3

$$TF-IDF(t,d,D) = TF(t,d) \cdot IDF(t,D) \quad (3)$$

3.3 การคัดเลือกคุณลักษณะด้วย Word2Vec

เป็นหนึ่งในเทคนิคหลักที่ใช้ในการสร้าง Word Embedding ซึ่งเป็นวิธีการที่แปลง text ให้เป็นเวกเตอร์ ซึ่งสามารถใช้ในการจับความหมายของคำและความสัมพันธ์ระหว่างคำได้ดีขึ้น โดยใช้ Continuous Bag of Words (CBoW) ในการเรียนรู้ word embeddings โดยตัวแบบจะพยายามทำนายคำที่เป็นเป้าหมาย หรือคำตรงกลางจากคำที่อยู่รอบข้างที่อยู่รอบๆ คำนั้นในประโยค ซึ่งช่วยให้ตัวแบบเรียนรู้การแทนความหมายของคำในบริบทที่หลากหลาย ดังสมการที่ 4

$$Context = \{w_{t-c}, w_{t-c+1}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c}\} \quad (4)$$

สำหรับคำเป้าหมาย w_t ในประโยค คำรอบๆ จะถูกกำหนดเป็น C ซึ่งอาจจะมีขนาด C ตัวอักษรที่อยู่ด้านซ้ายและด้านขวา โดยคำรอบๆ จะถูกแทนที่ด้วยเวกเตอร์ CBoW ใช้เวกเตอร์ของคำใน context เพื่อคาดการณ์คำเป้าหมาย W_t ดังสมการที่ 5

$$P(w_t|C) = \frac{e^{v_{w_t \cdot h}^T}}{\sum_{w \in V} e^{v_{w_t \cdot h}^T}} \quad (5)$$

โดยที่ h คือเวกเตอร์เฉลี่ยของ context words, V_{wt} คือ เวกเตอร์ของคำเป้าหมาย และ V คือชุดของคำทั้งหมดในคลังคำ

4. การสร้างตัวแบบ

4.1 Naïve Bayes Multinomial

Naïve Bayes Multinomial เป็นอัลกอริทึมการจำแนกประเภทที่ใช้ในการประมวลผลข้อมูลประเภทข้อความ เช่น การจัดประเภทเอกสาร หรือการวิเคราะห์ความคิดเห็น โดยอิงตามทฤษฎีของ Bayes และมีสมมติฐานว่าแต่ละฟีเจอร์เป็นอิสระจากกัน หลักการพื้นฐานของทฤษฎี Bayes สมการสำหรับการคำนวณความน่าจะเป็นที่ข้อมูล x จะอยู่ในคลาส C_k คือ

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)} \quad (6)$$

สำหรับ Naïve Bayes Multinomial วิธีนี้จะใช้ตัวแบบการคำนวณความน่าจะเป็นของการเกิดฟีเจอร์ คือ คำในข้อความ โดยการนับจำนวนครั้งของคำแต่ละคำที่ปรากฏในคลาส C_k ซึ่งคำนวณด้วยสมการที่ 7

$$P(x_i|C_k) = \frac{N_{x_i,C_k} + \alpha}{N_{C_k} + \alpha \cdot V} \quad (7)$$

หลังจากคำนวณความน่าจะเป็นของคำแต่ละคำแล้ว ความน่าจะเป็นของทั้งข้อความ x สามารถคำนวณได้จากสมการที่ 8

$$P(x_i|C_k) = \prod_{i=1}^{n} P(x_i|C_k) \quad (8)$$

โดย x_i คือ ลำดับคำที่ปรากฏในข้อความนั้น และ n คือ จำนวนคำทั้งหมดในข้อความ

จากนั้นตัวแบบ Naïve Bayes Multinomial จะทำการคำนวณความน่าจะเป็นของแต่ละคลาส $P(C_k|x)$ และทำการเลือกคลาสที่มีค่าความน่าจะเป็นสูงสุดเป็นคำตอบ

4.2 Support Vector Machine แบบ Linear

Support Vector Machine แบบ linear เป็นอัลกอริทึมการจำแนกประเภทที่ใช้เส้นตรงในการแยกข้อมูลออกเป็นสองคลาส โดยมีวัตถุประสงค์ในการหาขอบเขตการตัดสินใจที่ดีที่สุดซึ่งเรียกว่า hyperplane โดย hyperplane ที่ดีที่สุด คือเส้นที่ทำให้ระยะห่างระหว่างข้อมูลจากสองคลาสที่ใกล้ hyperplane ที่สุดมีค่ามากที่สุด ซึ่งข้อมูลที่อยู่ใกล้ hyperplane นี้เรียกว่า support vectors หรือจุดข้อมูลที่อยู่ใกล้กับเส้นแบ่งระหว่างคลาสและถูกใช้ในการกำหนดขอบเขตของการตัดสินใจ

SVM พยายามหาค่าของเวกเตอร์น้ำหนัก (w) และค่า bias (b) เพื่อหาฟังก์ชันสมการของเส้นตรง ดังสมการที่ 9

$$f(x) = w^T x + b \quad (9)$$

โดยเส้น hyperplane นี้จะแยกข้อมูลจากสองคลาสได้ ข้อมูลจากคลาสหนึ่งต้องอยู่ฝั่งหนึ่งของเส้นและข้อมูลจากอีกคลาสหนึ่งต้องอยู่ฝั่งตรงข้าม ซึ่งสามารถเขียนเงื่อนไขการจำแนกได้เป็น ดังสมการที่ 10

$$y_i(w^T x_i + b) \geq 1 \quad (10)$$

สำหรับข้อมูลทุกตัว x_i ในชุดข้อมูลการฝึก โดยที่ y_i คือคลาสของข้อมูล ซึ่ง $y_i = 1$ หรือ $y_i = -1$ ขึ้นอยู่กับคลาสของข้อมูล ซึ่ง svm พยายามที่จะหา hyperplane ที่ทำให้ระยะห่างระหว่าง support vectors จากสองคลาสมีค่าสูงสุด โดยระยะห่างนี้เรียกว่า margin ซึ่งสมการในการหาค่า margin ดังสมการที่ 11

$$Margin = \frac{2}{||w||} \quad (11)$$

โดยการหา $||w||$ หมายถึง การหาขนาดของเวกเตอร์น้ำหนัก w ดังนั้น SVM จะพยายามหาค่า w ที่ทำให้ margin มากที่สุด ซึ่งแปลงเป็นการแก้ปัญหาในรูปแบบฟังก์ชัน ได้ดังสมการที่ 12

$$\min_{w,b} \frac{1}{2}||w||^2 \quad (12)$$

ภายใต้เงื่อนไข $y_i(w^T x_i + b) \geq 1$

4.3 1D-CNN (1D Convolutional Neural Networks)

1D-CNN เป็นโครงข่ายประสาทเทียมรูปแบบหนึ่ง ที่ใช้สำหรับประมวลผลข้อมูลที่เป็นลำดับ จะทำการ convolution กับข้อมูลในรูปของเวกเตอร์ข้อมูล แบบ 1 มิติ ประกอบด้วยหลายขั้นตอนซึ่งรวมถึงการทำคอนโวลูชัน การทำ max pooling การใช้งาน fully connected layers และการใช้ activation functions เพื่อดึงลักษณะเด่นที่เกี่ยวข้องกับลำดับข้อมูล

เลเยอร์ Conv1D แรก จะทำการคำนวณการคอนโวลูชัน 1D บนข้อมูลลำดับ การคำนวณการคอนโวลูชันสามารถแสดงเป็นสมการได้ดังสมการที่ 13

$$y(t) = (x \cdot w)(t) = \sum_{i=0}^{k-1} x\,(t+i)w(i) \qquad (13)$$

โดยที่ $x(t)$ คือข้อมูลอินพุตที่ตำแหน่ง t และ $w(i)$ คือฟิลเตอร์คอนโวลูชันที่มีขนาด k และใช้ฟังก์ชันการกระตุ้นแบบ ReLU เพื่อเพิ่มความ non-linearity ให้กับตัวแบบหลังจากการคอนโวลูชันครั้งแรก จะใช้ MaxPooling เพื่อย่อขนาดของ feature map การทำ MaxPooling จะลดมิติข้อมูลโดยเลือกค่ามากที่สุดจากทุกตำแหน่งที่กำหนดที่อยู่ติดกันบน feature map ซึ่งช่วยสรุปฟีเจอร์และลดความซับซ้อนของข้อมูล การทำ pooling นี้ช่วยคงฟีเจอร์สำคัญไว้ขณะที่ลดขนาดของ input เพื่อให้เหมาะสมกับเลเยอร์ถัดไป ดังสมการที่ 14

$$y(t) = max\,(x(t), x(t+1)) \qquad (14)$$

จากนั้นจะทำการคอนโวลูชันและ pooling ซ้ำ โดยใช้ฟิลเตอร์ 64 และ 32 ตัวตามลำดับ และยังคงใช้ขนาดฟิลเตอร์เท่ากับ 5 เลเยอร์ ซึ่งจะช่วยดึงลักษณะซับซ้อนมากขึ้นจากข้อมูลเมื่อประมวลผลต่อไปในเครือข่าย ขั้นตอนต่อไปจะทำการลดมิติข้อมูลด้วยการนำค่ามากที่สุดจาก feature map ทั้งหมดออกมา การทำ global max pooling นี้จะย่อข้อมูลทั้งหมดให้เป็นเวกเตอร์เดียว โดยคงฟีเจอร์ที่สำคัญที่สุดจากข้อมูลลำดับทั้งหมด

หลังจากการ pooling แล้ว ข้อมูลที่ได้จะถูกส่งผ่านไปยัง fully connected layers สองเลเยอร์ โดยเลเยอร์แรกมีหน่วยประมวลผล 64 หน่วยพร้อมการกระตุ้นแบบ ReLU และเลเยอร์ที่สองมีหน่วยประมวลผล 1 หน่วยพร้อมการกระตุ้นแบบ sigmoid ฟังก์ชัน sigmoid ดังสมการที่ 15

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad (15)$$

จากสมการที่ 15 ฟังก์ชันนี้จะเปลี่ยนผลลัพธ์ให้อยู่ในรูปแบบความน่าจะเป็น ซึ่งเหมาะสำหรับการจำแนกประเภทแบบ binary classification คือแบ่งข้อมูลออกเป็นสองประเภท ตัวแบบจะถูกคอมไพล์โดยใช้ตัวปรับค่า Adam และ loss function แบบ binary crossentropy ซึ่งเหมาะสำหรับปัญหาการจำแนกประเภทแบบ binary classification ระหว่างการฝึกตัวแบบจะปรับน้ำหนักของเลเยอร์ต่างๆ เพื่อทำให้ข้อผิดพลาดในการจำแนกลดลงอย่างต่อเนื่อง

4.4 BERT Multilingual (mBERT)

mBERT เป็นตัวแบบภาษาที่ใช้พื้นฐานจากโครงสร้างของ BERT แต่มีการปรับปรุงและออกแบบมาให้เหมาะสมกับบริบทที่แตกต่างกันของภาษาและงานประมวลผลข้อมูล

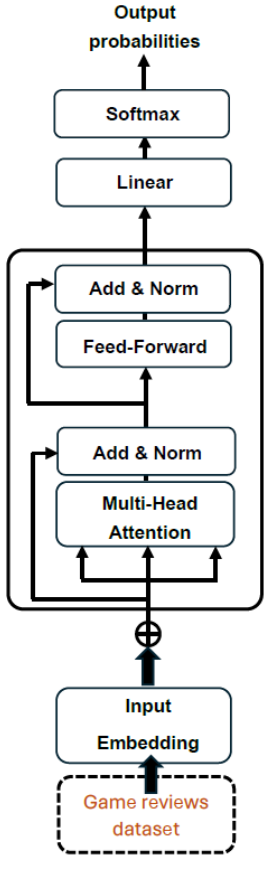

Figure 6 The structure of the BERT model

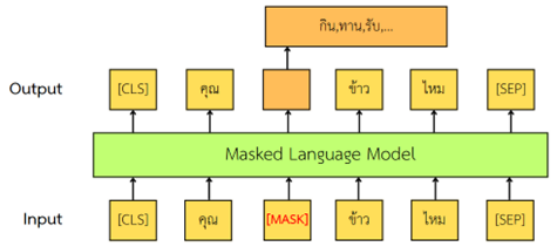

Figure 7 The structure of Mask Language Model

จาก Figure 6 แสดงการวิเคราะห์ผลลัพธ์จะเป็นการทำนายว่าข้อความนั้นมีความคิดเห็นเชิงบวกหรือเชิงลบ โดยโครงสร้างของ BERT จะใช้กระบวนการ embedding สามแบบประกอบด้วยกระบวนการ ได้แก่ 1) token embedding กระบวนการแปลงคำหรือ token ในข้อความให้อยู่ในรูปแบบของเวกเตอร์ เช่น [CLS], [SEP] และคำอื่น ๆ ซึ่งแต่ละเวกเตอร์จะมีข้อมูลเชิงความหมายที่ช่วยให้ตัวแบบเข้าใจบริบทของคำในประโยค 2) position embedding เวกเตอร์นี้ระบุตำแหน่งของคำในประโยค ทำให้ตัวแบบสามารถเข้าใจลำดับของคำในข้อความได้ เช่น ตำแหน่งของ [CLS] ที่แสดงถึงจุดเริ่มต้นของข้อความ 3) segment embedding ใช้เพื่อระบุว่า token ใดอยู่ในประโยคไหน หากมีการเปรียบเทียบประโยคหรือวิเคราะห์ความสัมพันธ์ระหว่างข้อความสองชุด เช่น ประโยคคำถามและคำตอบ เมื่อ embeddings ทั้งสามส่วนรวมกันจะถูกส่งผ่านเข้าตัวแบบ BERT โดยที่ BERT จะคำนวณและทำการเรียนรู้จากข้อมูลในทุกตำแหน่งของคำในข้อความ จากนั้นผลลัพธ์ที่ได้จากตำแหน่ง [CLS] ซึ่งเป็นตำแหน่งพิเศษที่ใช้สำหรับการทำนายประเภทของข้อความ จะถูกใช้ในการพิจารณาว่าข้อความนั้นมีความคิดเห็นเชิงบวกหรือเชิงลบ โดยปกติจะใช้ชั้นสุดท้ายของ BERT เพื่อคำนวณค่า ความน่าจะเป็นของแต่ละคลาส

4.5 WangchanBERTa

WangchanBERTa เป็น Thai Transformer -based NLP Model ใช้หลักการของ RoBERTa ได้รับการพัฒนาโดยเฉพาะสำหรับการประมวลผลภาษาธรรมชาติในภาษาไทยโดยใช้สองแนวทางหลักในการฝึกฝนตัวแบบ คือ encoder only model แสดงที่ Figure 6 ซึ่งเป็นสถาปัตยกรรมแบบ transformers รูปแบบหนึ่งที่ใช้งานส่วน encoder เท่านั้น โดยมักจะถูก pre-train ด้วยวิธี Masked Language Modeling (MLM) แสดงที่ Figure 7 ซึ่งเป็นการเติมคำที่หายไปในวลีหรือประโยค โดยตัวแบบภาษาประเภทนี้จะมีความถนัดในการทำงานที่เกี่ยวข้องกับการทำความเข้าใจภาษาธรรมชาติ เช่น ตัวแบบการจำแนกประเภทของข้อความ หรือตัวแบบภาษาและวิธี MLM หรือการทำนายคำในประโยคที่ถูกแทนที่ด้วย masked token โดยชุดข้อมูลที่ใช้ในการเทรนตัวแบบ จะเป็นชุดข้อมูลจากแหล่งต่างๆ เช่น วิกิพีเดียภาษาไทย ข่าวจากสำนักข่าวในประเทศไทย โพสท์ คอมเมนท์จากสื่อโซเชียลมีเดีย ซับไตเติลจากโครงการ OpenSubtitles และชุดข้อมูลอื่นๆ ที่มีการเผยแพร่และเปิดข้อมูลสู่สาธารณะจะสำหรับการฝึกฝนตัวแบบการประมวลผลภาษาไทยในโจทย์ต่างๆ เช่น machine translation, sentiment analysis และ text classification รวมเป็นข้อมูลขนาด 78.5 GB เนื่องด้วยการนำชุดข้อมูลเข้าสู่ตัวแบบเพื่อทำการทำนาย masked token จะต้องผ่านกระบวนการตัดแบ่งคำ

โดยการตัดแบ่งคำที่ใช้นั้นจะเป็น 4 รูปแบบ การตัดแบ่งคำ คือ 1) การตัดแบ่งหน่วยคำย่อย (subword-level tokenization) ด้วยไลบรารี่ SentencePiece (spm) โดยตัวตัดแบ่งหน่วยคำย่อยอาศัยข้อมูลทางสถิติของการปรากฏร่วมกันของตัวอักษรในชุดข้อมูลในการกำหนดขอบเขตของหน่วยคำย่อย 2) การตัดแบ่งคำจาก dictionary ของคำในภาษาไทยด้วย maximal matching algorithm (newmm) ด้วยไลบรารี่ PyThaiNLP 3) การตัดแบ่งพยางค์ในภาษาไทยจาก dictionary ของพยางค์ในภาษาไทยด้วย maximal matching algorithm (ซื่อย่อว่า syllable) ด้วยไลบรารี่ PyThaiNLP 4) การตัดแบ่งคำจากตัวแบบ machine learning (sefr)

ผลการวิจัย

จากการทดลองจากตัวแบบที่ใช้สถาปัตยกรรม Transformer และได้รับการพัฒนาโดยเฉพาะสำหรับการประมวลผลภาษาธรรมชาติในภาษาไทย

1. การตั้งค่าพารามิเตอร์

1.1 การตั้งค่าพารามิเตอร์ Naïve Bayes Multinomial

ในการทำงานกับข้อมูลข้อความสามารถมีอิทธิพลต่อประสิทธิภาพของตัวแบบได้ โดยในที่นี้มีการตั้งค่าหลักๆ ได้แก่ 1) alpha หลักการ คือควบคุมการป้องกันการเกิด overfitting และควบคุมความ bias ของตัวแบบต่อคำที่หายาก เป็นค่าที่ใช้ในการคำนวณความน่าจะเป็นใช้แก้ปัญหาคำที่ไม่เคยพบในชุดข้อมูล train โดยสามารถทดลองค่าต่างๆ เช่น 0.01, 0.1, 1.0, และ 10.0 เพื่อดูว่าค่าที่ดีที่สุดสำหรับชุดข้อมูล ค่าที่สูงจะทำให้การ smoothing มีมากขึ้น แต่ก็อาจทำให้ตัวแบบสูญเสียความสามารถในการเรียนรู้จากข้อมูลจริงไป 2) fit_prior ค่าที่ใช้กำหนด prior probabilities ของแต่ละคลาสที่ได้จากข้อมูลฝึก ถ้าตั้งเป็น true ตัวแบบจะใช้ prior probabilities ที่ได้จากข้อมูลฝึก ถ้าเป็น false ตัวแบบจะสมมุติว่าแต่ละ

คลาสมี prior ที่เท่ากัน3) class_prior เป็นค่าที่ใช้กำหน prior probability สำหรับแต่ละคลาส ซึ่งสามารถใช้ค่าที่กำหนดเป็นลิสต์หรือ none หากเป็น none ตัวแบบจะคำนวณ prior จากข้อมูลการฝึก ค่าที่กำหนดในลิสต์จะช่วยให้สามารถปรับความน่าจะเป็นให้เหมาะสมกับการกระจายของข้อมูลในชุดฝึก

ตัวแบบ Naïve bayes แบบ bag of words ค่าพารามิเตอร์ที่ดีที่สุดคือ alpha ที่ 0.1, ไม่กำหนด class_prior (เป็น none) และ fit_prior ตั้งเป็น true โดยได้ค่าความถูกต้องดีที่สุดคือ 0.8323 ต่อมาตัวแบบ Naïve bayes แบบ TF-IDF ค่าพารามิเตอร์ที่ดีที่สุดจากการตั้งค่า alpha ที่ 0.1, ไม่กำหนด class_prior และ fit_prior ตั้งเป็น true โดยได้ค่าความถูกต้องที่ดีที่สุดคือ 0.8313 และสุดท้ายการทดลองด้วยตัวแบบ Naïve bayes แบบ word2vec ค่าพารามิเตอร์ที่ดีที่สุดคือ alpha ที่ 15.0, ไม่กำหนด class_prior และ fit_prior ตั้งเป็น false โดยได้ค่าความถูกต้องที่ดีที่สุด คือ 0.6820

1.2 การตั้งค่าพารามิเตอร์ Support Vector Machines (Linear Kernel)

การตั้งค่าพารามิเตอร์หลักสำหรับ SVM ได้แก่ ค่า C ค่านี้ คือพารามิเตอร์หลักสำหรับ SVM ที่ควบคุมการกระจายของ hyperplane ในการแยกคลาส หากค่า C มีค่าที่สูงจะหมายถึงการให้ความสำคัญกับการลดข้อผิดพลาดในการจำแนกประเภทมากขึ้น ซึ่งอาจนำไปสู่การ overfitting ในขณะที่ค่าที่ต่ำจะทำให้ตัวแบบมีความยืดหยุ่นมากขึ้น แต่ก็อาจทำให้เกิดการ underfitting ได้เช่นกัน การทดลองใช้ค่าหลากหลาย เช่น 0.1, 1, 10, และ 100 จะช่วยให้สามารถค้นหาค่าที่เหมาะสมที่สุดสำหรับข้อมูลได้

สำหรับตัวแบบ SVM แบบ BoW มีค่า C ที่ดีที่สุด คือ 1 โดยได้ค่าความถูกต้อง คือ 0.8424 ส่วนตัวแบบ SVM แบบ TF-IDF ก็มีค่า C ที่ดีที่สุด คือ 0.1 โดยได้ค่าความถูกต้องที่ดีที่สุดคือ 0.8357 และตัวแบบ SVM แบบ Word2Vec มีค่า C ที่ดีที่สุด คือ 0.1 โดยได้ค่าความถูกต้องที่ดีที่สุดคือ 0.7388

1.3 การตั้งค่าพารามิเตอร์ 1D-CNN

ในการประมวลผลข้อความ มีหลายองค์ประกอบที่สำคัญซึ่งส่งผลต่อประสิทธิภาพของตัวแบบ การใช้ Conv1D มีบทบาทในการจับลักษณะของข้อมูลที่เป็นลำดับ โดยในที่นี้ได้กำหนดจำนวนฟิลเตอร์ที่ 128 ขนาดของฟิลเตอร์ที่ 5 และใช้ฟังก์ชันการเปิดใช้งาน ReLU เพื่อเพิ่ม non-linear ให้กับตัวแบบ ข้อมูลที่ผ่านการ convolutions จะถูกส่งไปยัง Max Pooling1D ซึ่งช่วยลดขนาดของข้อมูลและเน้นคุณลักษณะที่สำคัญ

จากนั้นมีการใช้ Conv1D อีกสองครั้ง โดยลดจำนวนฟิลเตอร์ลงเหลือ 64 และ 32 ตามลำดับ และใช้ MaxPooling1D ในแต่ละชั้นเพื่อช่วยลดมิติของข้อมูลให้เล็กลงอีก หลังจากการ convolutions และ pooling หลายชั้น ข้อมูลจะถูกส่งไปยัง GlobalMaxPooling1D ซึ่งช่วยในการดึงค่าที่มีความสำคัญสูงสุดจากลำดับทั้งหมดในแต่ละฟีเจอร์ และข้อมูลจะถูกเชื่อมต่อเข้าสู่ชั้น dense ที่มี 64 นิวรอนและใช้ฟังก์ชันการเปิดใช้งาน ReLU เพื่อให้ตัวแบบสามารถเรียนรู้ลักษณะความสัมพันธ์ที่ซับซ้อนได้ดีขึ้น ชั้นสุดท้ายคือ dense ที่มีนิวรอน 1 ตัวและใช้ฟังก์ชันการเปิดใช้งาน sigmoid เพื่อให้ผลลัพธ์ที่เป็นค่าความน่าจะเป็นระหว่าง 0 และ 1 สำหรับการจำแนกประเภทที่เป็น binary

สุดท้ายการใช้ Adam เป็น optimizer ที่เป็นที่นิยมในงาน deep learning ซึ่งช่วยในการหาค่าที่เหมาะสมได้อย่างรวดเร็ว โดยใช้ loss function เป็น binary_crossentropy ซึ่งเหมาะสมสำหรับปัญหาการจำแนกประเภท binary และการตั้งค่า epochs และ batch_size เป็นส่วนสำคัญในการฝึกตัวแบบ 1D-CNN ซึ่งมีผลต่อประสิทธิภาพและความสามารถในการเรียนรู้ของตัวแบบ โดยการตั้งค่า epochs จำนวนรอบที่ตัวแบบจะผ่านข้อมูลในการฝึก โดยการตั้งค่าเป็น 5, 10, 15, หรือ 20 ช่วยให้เห็นถึงผลกระทบของการ train ในระยะเวลาต่างๆ การ train นานขึ้นสามารถช่วยให้ตัวแบบเรียนรู้ลักษณะของข้อมูลได้ดีขึ้น แต่ก็มีความเสี่ยงที่จะ overfit ถ้า train data นานเกินไป ในส่วนของ batch_size จำนวนตัวอย่างที่ตัวแบบจะนำมาใช้ในการอัปเดตน้ำหนักในแต่ละรอบ การตั้งค่าเป็น 8, 16, หรือ 32 จะแสดงให้เห็นถึงความสมดุลระหว่างประสิทธิภาพและความเร็วในการ train ขนาด batch ที่เล็กจะทำให้ตัวแบบสามารถอัพเดตค่า weight ได้บ่อย แต่จะใช้เวลาในการ train ที่มากกว่า

เมื่อตัวแบบถูก train ด้วย epochs ที่ตั้งค่าเป็น 10 และ batch_size ที่ตั้งค่าเป็น 16 ผลลัพธ์ที่ได้คือค่าความถูกต้องที่ 0.76 ซึ่งให้ค่าความถูกต้องที่ดีที่สุดเมื่อเทียบกับตั้งค่าแบบอื่นๆ

1.4 การตั้งค่าพารามิเตอร์ BERT multilingual และ WangchanBERTa

ทั้ง 2 วิธีการมีความสำคัญต่อการฝึกตัวแบบเพื่อให้สามารถประมวลผลภาษาต่างๆ ได้อย่างมีประสิทธิภาพ โดยการตั้งค่าที่สำคัญ ดังนี้ num_train_epochs เป็นการกำหนดจำนวนรอบที่ตัวแบบจะผ่านข้อมูลในระหว่างการ train โดยการตั้งค่าที่เหมาะสมช่วยให้ตัวแบบเรียนรู้จากข้อมูลได้ดีขึ้น แต่ถ้าตั้งค่าสูงเกินไปอาจทำให้เกิดการ overfitting การตั้งค่า train_batch_size สำหรับควบคุมจำนวนตัวอย่างที่ใช้ใน

การอัพเดตค่า weight ในแต่ละรอบขนาดที่เหมาะสมจะจะช่วยเพิ่มความเร็วในการ train

Table 6 Performance comparison of different imbalanced data handling methods for Naïve Bayes classification using bag of words representation.

Evaluation Metrics	Imbalanced Data	Balanced Data	
		ROS	RUS
Accuracy	82.00	79.53	79.56
Precision	83.69	87.12	88.31
Recall	90.84	81.51	80.10
F1-Score	87.12	84.22	84.00

และให้ตัวแบบเรียนรู้จากข้อมูลได้ดี สำหรับตัวแบบ BERT multilingual มีค่า num_train_epochs ที่ดีที่สุด คือ 6 และค่า train_batch_size ที่ดีที่สุด คือ 64 โดยได้ค่าความถูกต้องที่ดีที่สุด คือ 0.78 ส่วนตัวแบบ WangchanBERTa มีค่า num_train_epochs ที่ดีที่สุด คือ 3 และค่า train_batch_size ที่ดีที่สุด คือ 64 โดยได้ค่าความถูกต้องที่ดีที่สุด คือ 0.90

2. ผลการเปรียบเทียบการจัดการข้อมูลที่ไม่สมดุล

ในงานวิจัยนี้วัดประสิทธิภาพของตัวแบบด้วย ค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) ค่าความถ่วงดุล (F1-score) (Khruahong *et al*., 2022; Phiphitphatphaisit & Surinta, 2024)

จากการเปรียบเทียบวิธีการจัดการข้อมูลที่ไม่สมุดล ได้แก่ วิธี ROS และวิธี RUS โดยใช้วิธีการจำแนกประเภทด้วย ตัวแบบ Naïve bayes แบบ BoW และวัดประสิทธิภาพบนข้อมูลทดสอบได้ผลการเปรียบเทียบค่าวัดประสิทธิภาพต่าง ๆ แสดงดัง Table 6

จาก Table 6 พบว่าค่าความถูกต้องของชุดข้อมูลที่ไม่สมดุลอยู่ที่ 82.00% เมื่อเทียบกับการใช้เทคนิค ROS และ RUS ซึ่งให้ค่า accuracy ใกล้เคียงกัน โดย ROS อยู่ที่ 79.53% และ RUS อยู่ที่ 79.56% สำหรับค่า precision ของข้อมูลเริ่มต้นอยู่ที่ 83.69% ขณะที่ ROS มีค่า precision สูงขึ้นเป็น 87.12% และ RUS สูงสุดที่ 88.31% ในขณะที่ค่า recall ของชุดข้อมูลที่ไม่สมดุลสูงที่สุดที่ 90.84% ตามด้วย ROS ที่มี recall 81.51% และ RUS ที่ 80.10% ค่า F1-score ซึ่งเป็นค่าที่สะท้อนทั้ง precision และ recall พบว่าชุดข้อมูลที่ไม่สมดุลมีค่า F1-score อยู่ที่ 87.12% ซึ่งลดลงเล็กน้อยเมื่อใช้ ROS โดยมีค่า F1-score อยู่ที่ 84.22% และ RUS ค่า F1-score ที่ 84.00%

เมื่อพิจารณาจากผลลัพธ์พบว่า ROS เป็นเทคนิคที่เหมาะสมที่สุดในการจัดการข้อมูลที่ไม่สมดุล เนื่องจาก ROS ช่วยเพิ่มจำนวนตัวอย่างในคลาสที่มีจำนวนน้อยโดยการคัดลอกข้อมูลจากคลาสนั้น ซึ่งทำให้ตัวแบบเรียนรู้ลักษณะของกลุ่มน้อยได้ดีขึ้น ผลที่ได้คือค่า precision ที่สูงขึ้นจาก 83.69% เป็น 87.12%

Table 7 Performance comparison of machine learning models.

Models	Features	Evaluation metrics					
		5-CV (STD)		Accuracy	Precision	Recall	F1-Score
Naïve Bayes	BoW	0.81	0.0094	**79.53**	87.12	81.51	84.22
	TF-IDF	0.82	0.0097	79.00	87.99	79.52	83.54
	Word2Vec	0.68	0.0032	67.12	69.97	89.24	78.44
SVM	BoW	0.83	0.0111	**81.26**	85.80	86.33	86.06
	TF-IDF	0.82	0.0100	80.70	86.98	83.73	85.32
	Word2Vec	0.74	0.0051	72.79	86.03	70.91	77.74

เนื่องจากตัวแบบสามารถทำนายคลาสกลุ่มน้อยได้แม่นยำขึ้น แม้ว่า recall จะลดลงเมื่อเทียบกับชุดข้อมูลที่ไม่สมดุลจาก 90.84% เป็น 81.51% แต่ค่า recall ของ ROS ยังสูงกว่าค่า recall ของ RUS ที่ 80.10% ซึ่งแสดงว่า ROS ยังคงสามารถจับคลาสกลุ่มน้อยได้ดีแม้จะเพิ่มจำนวนตัวอย่างในกลุ่มน้อยขึ้น F1-Score ของ ROS อยู่ที่ 84.22% ซึ่งแม้จะลดลงจากข้อมูลเริ่มต้นที่ 87.12% แต่ยังคงอยู่ในระดับที่เหมาะสมเมื่อเทียบกับการใช้ RUS ที่มีค่า F1-score ที่ 84% ซึ่งหมายความว่า ROS สามารถรักษาสมดุลระหว่าง precision และ recall ได้ดีกว่า และเนื่องจาก ROS ไม่ได้ลบข้อมูลจากคลาสกลุ่มใหญ่เหมือนกับ RUS การใช้ ROS จึงไม่ทำให้สูญเสียข้อมูลที่อาจมีความสำคัญจากคลาสใหญ่ไป ทำให้ตัวแบบยังคงสามารถเรียนรู้จากข้อมูลทั้งหมดได้

ดังนั้น จากผลลัพธ์ที่ได้นี้ ROS จึงเป็นวิธีที่เหมาะสมกว่าในการจัดการกับข้อมูลที่ไม่สมดุล เพราะช่วยเพิ่มความแม่นยำในการทำนายคลาสกลุ่มน้อย โดยไม่ทำให้ตัวแบบเสียสมดุลในการเรียนรู้จากข้อมูลทั้งสองคลาส

3. ผลการวิจัย Machine Learning Model

ส่วนนี้จะเป็นการเปรียบเทียบผลลัพธ์ที่ได้จากการประเมินประสิทธิภาพของ machine learning model ได้แก่ Naïve bayes และ SVM สำหรับการจำแนกความคิด

เห็นของผู้เล่นเกม โดยใช้วิธี feature extraction ร่วมกันกับตัวแบบ machine learning model ได้แก่ Bag of Words, TF-IDF และ Word2ฮec และ CNN model ได้แก่ 1D-CNN จาก Table 7 ที่แสดงการเปรียบเทียบประสิทธิภาพระหว่าง 2 ประเภทตัวแบบพบว่า machine learning model ได้แก่ วิธี SVM ที่ประมวลผลร่วมกันกับเทคนิค BoW (SVM+BoW) มีค่าเฉลี่ยของค่า accuracy ที่ได้จากการทดสอบตัวแบบแบบ 5 fold cross validation เท่ากับ 0.83 หรือ 83% ซึ่งมากที่สุดและมีค่า standard deviation เท่ากับ 0.0111 หมายความได้ว่าการเปลี่ยนแปลงของค่า accuracy ระหว่างแต่ละ fold มีค่าเบี่ยงเบนเพียงเล็กน้อยแสดงว่าตัวแบบมีความเสถียรดีเมื่อทดสอบกับ fold ที่ต่างกัน และเมื่อเทียบประสิทธิภาพกับ CNN model ที่แสดงบน Table 9 การวัดค่าเมตริกซ์ของทั้ง 7 ตัวแบบแสดงให้เห็นว่า SVM+BoW เป็นวิธีการที่ให้ค่า accuracy สูงที่สุดที่ 81.26% ส่วน CNN model ซึ่งก็คือ 1D-CNN ให้ค่า accuracy ที่ต่ำกว่าที่ 72.96%

4. ผลการวิจัย Transformer Model

การวิเคราะห์ประสิทธิภาพของตัวแบบ transformer ในการจำแนกประเภทความคิดเห็น โดยได้ทดสอบกับ 2 ตัวแบบ ได้แก่ BERT Multilingual และ WangchanBERTa ผลลัพธ์ที่ได้จากการทดลองแสดงดัง Table 8 ซึ่งแสดงถึงค่า mean และค่า standard deviation ของ 5-fold cross validation จากชุด train และค่า accuracy, ค่า precision, ค่า recall และค่า F1-score จากชุดทดสอบ

Table 8 Performance comparison of transformer models.

Transformer Models	Evaluation metrics				
	5-CV (STD)	Accuracy	Precision	Recall	F1-Score
BERT Multilingual	0.79 0.0079	78.03	83.74	83.57	78.03
WangchanBERTa	0.79 0.1492	**82.16**	**87.06**	**86.18**	**86.62**

Table 9 Performance comparison of 1D-CNN model.

Model	Features	Evaluation metrics			
		Accuracy	Precision	Recall	F1-Score
1D-CNN	BoW	60.31	68.85	74.45	71.54
	TF-IDF	50.85	73.03	42.28	53.55
	Word2Vec	72.96	85.90	71.37	77.97

จาก Table 8 ที่แสดงการเปรียบเทียบประสิทธิภาพของทั้ง 2 ตัวแบบพบว่า WangchanBERTa และ mBERT มีค่าเฉลี่ยของค่า accuracy ที่ได้จากการทดสอบตัวแบบแบบ 5 fold cross validation เท่ากัน คือ 0.79 หรือ 79% แต่เมื่อพิจารณาที่ค่า standard deviation ซึ่ง mBERT มีค่า STD ที่น้อยกว่า คือ 0.0079 ซึ่งหมายความว่า mBERT เป็นตัวแบบที่มีความเสถียรมากกว่าในการทดสอบด้วย 5-CV ในส่วนของการวัดประสิทธิภาพของตัวแบบพบว่า WangchanBERTa ให้ค่า accuracy ที่สูงกว่าคือ 82.16% ซึ่งสูงที่สุดเมื่อเปรียบเทียบกันแล้ว ดังนั้นจึงอาจวิเคราะห์ได้ว่า WangchanBERTa เป็นตัวแบบที่มีค่า accuracy ที่สูงที่สุดในตัวแบบประเภท transformer model สามารถประมวลผลได้ดีในบาง fold แต่ประสิทธิภาพไม่คงที่เมื่อทำการ cross-validation ระหว่างแต่ละ fold และเมื่อเปรียบเทียบกับ 1D-CNN ตัวแบบประเภท CNN model ที่ Table 9 แล้ว WangchanBERTa มีประสิทธิภาพของการจำแนกคลาสที่ดีกว่าถึง 10% โดย 1D-CNN มีค่า accuracy อยู่ที่ 72.96% เมื่อพิจารณาที่ตัวแบบทั้ง 2 ประเภทแล้วพบว่า transformer model มีค่า accuracy ที่มากที่สุด

5. ผลการเปรียบเทียบ Training Time ในงานวิจัยนี้ได้ทำการเปรียบเทียบระยะเวลา

ในการ train ตัวแบบที่ใช้ในการจำแนกข้อความคิดเห็น โดยได้ทำการเปรียบเทียบตัวแบบที่แตกต่างกัน 3 ตัวแบบ ได้แก่ SVM แบบ BoW, 1D-CNN แบบ Word2Vec และ WangchanBERTa เพื่อวิเคราะห์ประสิทธิภาพในด้านของ training time โดยการทดลองทั้งหมดดำเนินการบน google colab โดยใช้ T4 GPU เพื่อเพิ่มประสิทธิภาพในการประมวลผล

จากผลการทดลอง Table 10 พบว่า SVM แบบ BoW ใช้เวลาในการ train ที่ 56.30 วินาที ซึ่งน้อยที่สุดเมื่อเทียบกับตัวแบบอื่น เนื่องจาก SVM เป็นวิธีการที่มีโครงสร้างที่เข้าใจง่ายและรวมเข้ากับคุณลักษณะการแปลงข้อความจากเทคนิค BoW ที่ไม่ต้องการการเรียนรู้เชิงลึกของลำดับข้อมูล ในขณะที่ 1D-CNN แบบ Word2Vec ใช้เวลา 218.62 วินาที ซึ่งมีการประมวลผลที่มากขึ้นเนื่องจากต้องทำการเรียนรู้เชิงลึกจากเวกเตอร์ของคำที่ได้จาก Word2Vec

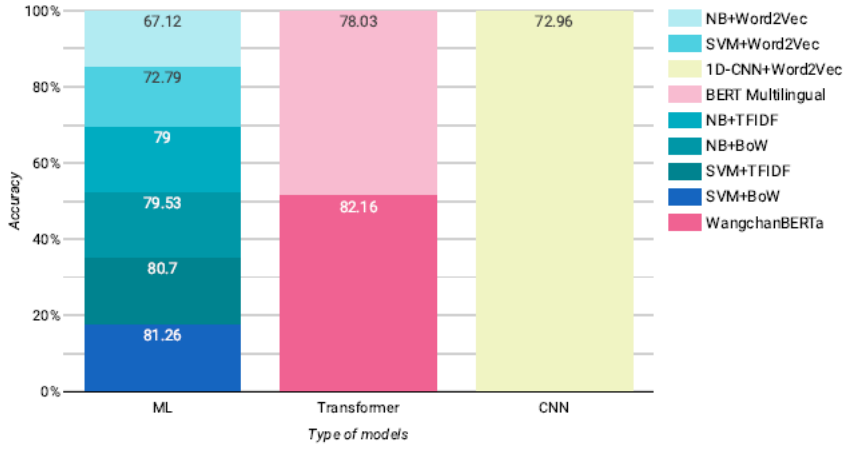

Figure 13 Comparative analysis of the accuracy performance across the three model categories

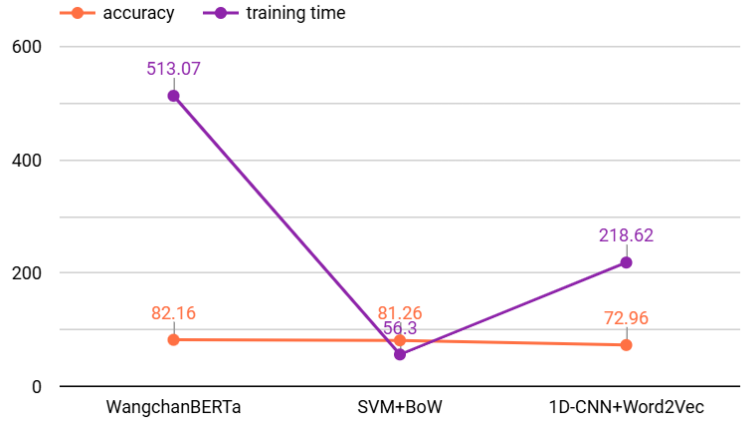

Figure 14 Comparison of accuracy and training time across the best-performing model categories

Table 10 A comparison of the training time of sentiment analysis models

Models	Training time (seconds)
SVM (Bag of Words)	56.30
1D-CNN (Word2Vec)	218.62
WangchanBERTa	513.07

ดังนั้น ส่วนของตัวแบบที่ใช้เวลาฝึกสอนมากที่สุด คือ WangchanBERTa โดยมี training time ที่ 513.07 วินาที ซึ่งเกิดจากโครงสร้างของตัวแบบที่เป็นแบบ transformer และมีจำนวนพารามิเตอร์สูง ส่งผลให้ต้องใช้ทรัพยากรในการคำนวณมากขึ้น ถึงแม้ว่าตัวแบบ WangchanBERTa จะใช้เวลาในการ train มากสุดแต่เมื่อเทียบค่าความถูกต้องในการเรียนรู้บริบทของภาษาไทยแล้ว จะเห็นได้ว่า Wangchan-BERTa ให้ค่าความถูกต้องที่ดีที่สุดเมื่อเทียบกับตัวแบบอื่น

วิจารณ์ผลการวิจัย

จาก Figure 13 แสดงการเปรียบเทียบค่าความถูกต้องจากทุกตัวแบบด้วยกราฟ stack column ซึ่งในงานวิจัยของเรามีทั้งหมด 9 ตัวแบบ จากรูปภาพแสดงการเปรียบเทียบแบ่งเป็น 3 ประเภทตัวแบบ จะเห็นได้ว่า WangchanBERTa ซึ่งอยู่ในประเภท transformer model มีค่าความถูกต้องสูงที่สุดที่ 82.16% รองลงมาคือ SVM แบบ BoW ค่าความถูกต้องอยู่ที่ 81.26% ซึ่งเป็นตัวแบบประเภท machine learning model ส่วนตัวแบบประเภท CNN ให้ค่าความถูกต้องในการประมวลผลชุดข้อมูลภาษาไทยต่ำสุดในสามประเภทตัวแบบอยู่ที่ 72.96% คือ 1D-CNN

ส่วนของวิธีการแปลงข้อความให้อยู่ในรูปแบบเวกเตอร์ที่สามารถนำไปใช้กับตัวแบบต่างๆ ทั้ง 3 วิธี ได้แก่ BoW, TF-IDF และ Word2Vec จากกราฟ stack column แล้ว BoW เมื่อนำไปใช้กับ machine learning เดียวกัน โดย SVM+BoW ให้ค่าความถูกต้องที่ 81.26% ซึ่งมีค่าสูงกว่า SVM+TFIDF ที่

มีค่าความถูกต้องที่ 80.7% และตัวแบบ Naïve bayes ก็ให้ค่าความถูกต้องที่สูงกว่าเช่นเดียวกัน เนื่องจากวิธีการ BoW อาจจะเหมาะสำหรับการทำ sentiment analysis แบบพื้นฐาน ไม่ใช้การตัดคำที่ละเอียดเกินไปจึงเหมาะสมกับการนำไปใช้กับประโยคสั้นๆ อย่างเช่นข้อความแสดงความคิดเห็นได้ดี ถึงแม้ว่า TF-IDF จะเป็นวิธีการที่พัฒนาเพิ่มเติมให้มีประสิทธิภาพที่ดีขึ้นจาก BoW แต่จากผลการทดลองจะเห็นได้ว่าเมื่อนำไปใช้กับ machine learning ตัวแบบเดียวกันวิธีการ BoW ให้ค่าความถูกต้องสูงกว่า TF-IDF

เมื่อวิเคราะห์ถึงหลักการของวิธีการแปลงข้อความทั้ง 2 วิธี และการทดลองของเราใช้ชุดข้อมูลที่เป็นภาษาไทย โดยข้อความในภาษาไทยจะไม่มีการเว้นวรรคระหว่างคำ และมักจะมีคำที่ใช้บ่อย ซ้ำๆ สำคัญต่อความหมาย ด้วยหลักการของ BoW ที่ไม่ได้ทำการปรับ weight ด้วย IDF จึงสามารถจับลักษณะพิเศษนี้ได้ดี เพราะ BoW จะให้ความสำคัญกับคำที่ปรากฏบ่อยตามธรรมชาติของภาษา แต่ TF-IDF จะลดความสำคัญของคำที่ปรากฏบ่อยในชุดข้อมูลข้อความคิดเห็น ซึ่งบางครั้งคำที่ปรากฏบ่อยๆ ที่ถูกตัดออกไปตามหลักการอาจมีความสำคัญต่อการจำแนกคำในข้อความคิดเห็นได้ จึงอาจส่งผลให้ประสิทธิภาพของ machine learning เมื่อรวมกับ TF-IDF แล้วอาจจะทำให้ค่าความถูกต้องลดลงได้ อีกหนึ่งเหตุผลในการทดลองของเรา TF-IDF จะเหมาะสมกับ deep learning models และ Transformers model ที่สามารถเรียนรู้จากข้อมูลที่ปรับ weight ได้มากกว่า

ความแตกต่างระหว่าง BERT และ WangchanBERTa เป็นเรื่องที่สำคัญในงานวิจัยที่ใช้เทคนิค transformer เพื่อประมวลผลภาษาธรรมชาติ โดย BERT มีความสามารถในการเรียนรู้บริบทของคำทั้งด้านซ้ายและขวา โดยใช้โครงสร้าง transformer ที่ช่วยให้มันสามารถจับลำดับและความสัมพันธ์ระหว่างคำในประโยคได้ดี ข้อมูลจะถูกประมวลผลแบบพร้อมกันทั้งประโยค BERT ถูกออกแบบมาเพื่อการใช้งานทั่วไปและไม่ได้เน้นเฉพาะกับภาษาใดภาษาหนึ่ง ซึ่งอาจทำให้ไม่เหมาะสมกับการประมวลผลภาษาไทยที่มีลักษณะเฉพาะที่แตกต่างจากภาษาอังกฤษ ในขณะที่ WangchanBERTa ได้รับการพัฒนาโดยเน้นการปรับปรุงการประมวลผลภาษาไทยโดยเฉพาะ โดยจะทำการ train ตัวแบบให้สามารถเข้าใจลักษณะเฉพาะของภาษาไทย เช่น การใช้พยางค์ที่ซับซ้อนและการสะกดคำที่ไม่เป็นระเบียบ แต่ข้อมูลที่ใช้ต้องเป็นข้อความล้วนไม่มี HTML หรือ markup ทำให้ WangchanBERTa มีประสิทธิภาพดีกว่า BERT ในการทำงานกับภาษาไทย ดังแสดง Table 8

จาก Table 8 แสดงให้เห็นว่า WangchanBERTa มีประสิทธิภาพในการประมวลผลชุดข้อมูลรีวิวเกมที่ดีกว่า BERT ในทุกค่าการวัดประสิทธิภาพ เพราะคุณสมบัติหลักของ WangchanBERTa สามารถจับความหมายของคำและโครงสร้างภาษาไทยได้แม่นยำกว่า BERT ที่ไม่ได้รับการ train โดยเฉพาะกับภาษาไทย ดังนั้นการใช้ WangchanBERTa จึงเหมาะสมกว่าในการประมวลผลภาษาไทย เพราะเป็นเทคนิคที่เหมาะกับภาษาที่ไม่มีการเว้นวรรคระหว่างคำอย่างเช่นภาษาไทย เนื่องจาก WangchanBERTa ใช้ SentencePiece แบบ Unigram ที่ไม่ต้องอาศัยการเว้นวรรคในการตัดคำและภาษาไทยไม่มีช่องว่างระหว่างคำ อีกทั้ง WangchanBERTa ใช้แนวทางของ RoBERTa ซึ่งเป็นวิธีการที่พัฒนาจาก BERT ให้มีประสิทธิภาพของการประมวลผลข้อความที่ดีขึ้น นอกจากนี้ WangchanBERTa ยังสามารถเข้าใจคำแสลง คำย่อ และศัพท์โซเชียลที่มักจะอยู่ในข้อความรีวิวต่างๆ นั้นได้ดีกว่า BERT ที่เน้นภาษาอังกฤษที่เป็นทางการ จึงทำให้ WangchanBERTa มีประสิทธิภาพในการวิเคราะห์ข้อมูลภาษาไทยได้ดีกว่า BERT

นอกจากการวัดประสิทธิภาพด้วยเมตริกซ์ต่างๆ แล้ว ระยะเวลาที่ใช้ในการ train ตัวแบบก็เป็นส่วนสำคัญโดยเฉพาะการประมวลผลบนอุปกรณ์ที่มีข้อจำกัดเรื่องทรัพยากร Figure 14 แสดงการเปรียบเทียบด้วยกราฟเส้นจากตัวแบบที่มีค่าความถูกต้องสูงสุดของแต่ละประเภทตัวแบบจะเห็นได้ว่า WangchanBERTa เป็นตัวแบบที่มีประสิทธิภาพดีที่สุด โดยมีค่าความถูกต้องสูงถึง 82.16% ในขณะที่วิธีการ Support Vector Machines ที่ใช้ เทคนิค BoW เป็นตัวแบบที่แสดงผลลัพธ์ที่ดีที่สุดในกลุ่มของ machine learning model ด้วยค่าความถูกต้องที่ 81.26% เมื่อวิเคราะห์ถึงผลการทดลองจะเห็นว่าค่าความถูกต้องของวิธีการที่ดีที่สุดของทั้ง 2 ประเภทตัวแบบมีค่าห่างกันเพียงเล็กน้อย โดยตัวแบบ machine learning ได้ค่าความถูกต้องที่ใกล้เคียงกันกับ transformer model แต่ใช้ทรัพยากรน้อยกว่า ในกรณีนี้จึงเหมาะสำหรับสถานการณ์ที่มีข้อจำกัดด้านทรัพยากร ข้อมูลเชิงลึกนี้มีประโยชน์สำหรับงานการวิเคราะห์ความคิดเห็นที่มีลักษณะเป็นข้อความรีวิว สามารถใช้ตัวแบบ machine learning ในการประมวลผลข้อมูลที่เป็นข้อความคิดเห็นได้

สรุปผลการวิจัย

จากผลการทดลองแสดงให้เห็นถึงประสิทธิภาพของตัวแบบทั้งในกลุ่ม machine learning และ transformer พบว่าในกลุ่ม machine learning ตัวแบบที่ทำงานได้ดีที่สุดคือ SVM ที่ใช้ BoW ซึ่งมีค่า accuracy ที่ 81.26% และมีค่า

precision ค่า recall ค่า F1-score ที่ให้ผลลัพธ์ที่ดี โดยค่า precision และค่า recall อยู่ที่ 85.80% และ 86.33% ตามลำดับ โดยวิธีแบบ BoW มีคุณสมบัติที่ดีกับการจับคำโดยไม่ต้องเข้าใจโครงสร้างประโยคเหมาะกับข้อความที่เป็นรีวิว การแสดงความคิดเห็น จากผลการทดลองจึงแสดงให้เห็นว่าตัวแบบนี้สามารถจับข้อมูลได้แม่นยำและมีการจำแนกข้อมูลได้ดีส่วนในกลุ่ม transformer ตัวแบบที่มีประสิทธิภาพดีที่สุดคือ WangchanBERTa โดยมีค่า accuracy ที่ 82.16% ซึ่งสูงที่สุดในกลุ่ม transformer models และมีค่า precision ค่า recall และค่า F1-score ที่ดีที่สุดเมื่อเทียบกับ BERT Multilingual เนื่องจากข้อความรีวิวที่เป็นภาษาไทยไม่มีช่องว่างระหว่างคำในประโยค จึงไม่ต้องทำการTokenization จึงอาจทำให้ tokenizer ของ BERT มีการตัดคำผิดพลาดได้และส่งผลให้การประมวลผลของ BRET เข้าใจ context ผิด

นอกจากนี้ WangchanBERTa สามารถจับความหมายของคำได้ดีโดยไม่ต้องเข้าใจความหมายของทั้งประโยคและจับความหมายในประโยคที่ไม่เป็นทางการได้ดีกว่าจากข้อมูลที่เป็นข้อความรีวิวที่เป็นลักษณะของประโยคที่มีคำที่เป็นภาษาโซเชียล หรือมีคำสแลงอยู่ด้วย จากผลการทดลองและคุณสมบัติของตัวแบบจึงแสดงให้เห็นว่า WangchanBERTa เหมาะสมที่สุดในการวิเคราะห์ข้อมูลภาษาไทยในชุดข้อมูลนี้ แม้ว่าตัวแบบ machine learning จะให้ผลลัพธ์ที่ดี แต่ตัวแบบ transformer อย่าง WangchanBERTa แสดงถึงความสามารถที่ดีกว่าในด้านความถูกต้องและการจำแนกประเภทข้อความที่ซับซ้อนโดยเฉพาะข้อมูลภาษาไทย ผลการทดลองแสดงให้เห็นว่าการทำ feature extraction มีความสำคัญต่อ machine learning model โดยจากการศึกษาพบว่า วิธีการ feature extraction ที่สนใจเฉพาะคำซึ่งได้แก่ BoW และ TF-IDF สามารถแยกคุณลักษณะได้ดีกว่าวิธีการที่สนใจบริบทของคำ เช่น Word2Vec

งานวิจัยในอนาคตมีแนวคิดที่จะเพิ่มการทดลองในส่วนของกระบวนการสร้าง feature ด้วยวิธีการ word embeddings ที่หลากหลายขึ้น เช่น Word2Vec, GloVe รวมถึงการผสมผสานระหว่างวิธีการต่าง ๆ เพื่อเพิ่มความแม่นยำในการจับความหมายโดยเฉพาะข้อมูลที่เป็นภาษาไทย และส่วนของตัวแบบ transformer อาจจะเพิ่มการทดลองโดยใช้ตัวแบบที่หลากหลายขึ้น เช่น GPT, RoBERTa เพื่อเปรียบเทียบและปรับปรุงประสิทธิภาพให้ดีขึ้นต่อไป

กิตติกรรมประกาศ

โครงการวิจัยนี้ได้รับการสนับสนุนจากเงินทุนอุดหนุนการวิจัยจากงบประมาณเงินรายได้ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคามประจำปีงบประมาณ 2568

เอกสารอ้างอิง

ปราชญภาคย์ เหล่าสังข์สุข, อนัส จินดา และสิรยา สิทธิสาร. (2560). การวิเคราะห์ความคิดเห็นเกี่ยวกับร้านอาหารบนเว็บไซต์รีวิว. *วารสารมหาวิทยาลัยทักษิณ*, *20*(1), 39–47. https://ph02.tci-thaijo.org/index.php/tsujournal/issue/view/6788

รวิสุดา เทศเมือง และนิเวศ จิระวิชิตชัย. (2560). การวิเคราะห์ความคิดเห็นภาษาไทยเกี่ยวกับการรีวิวสินค้าออนไลน์โดยใช้ขั้นตอนวิธีซัพพอร์ตเวกเตอร์แมทชีน. *วารสารวิศวกรรมศาสตร์ มหาวิทยาลัยสยาม*, *18*(1), 1–12.

วนัสวรรณ มีประเสริฐ และเอกรัฐ รัฐกาญจน์. (2564). การวิเคราะห์ความคิดเห็นจากทวิตเตอร์ของลูกค้าบริษัทช้อปปี้ประเทศไทย. *วารสารระบบสารสนเทศด้านธุรกิจ*, *7*(3), 6–18. https://doi.org/10.14456/jisb.2021.11

วสวัตติ์ อินทร์แปลง และจารี ทองคำ. (2563). การวิเคราะห์ความคิดเห็นต่อเกมมือถือพับจีด้วยเหมืองข้อความ. *วารสารวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยมหาสารคาม*, *39*(5), 523–531. https://li01.tci-thaijo.org/index.php/scimsujournal/article/view/235350/172284

Bais, R., Odek, P., & Ou, S. (2017). *Sentiment classification on Steam reviews*. Stanford University. https://cs229.stanford.edu/proj2017/final-posters/ 5147938.pdf

Eukeik. (2011). *ตลาดเกมมูลค่าใหญ่แค่ไหนในประเทศไทย* [How big is the game market value in Thailand?]. Marketeer Online. https://marketeeronline.co/archives/241572

Gimme. (2018). *เปิดตัวเลขคนเล่นเกมในไทย* [Revealing the number of gamers in Thailand]. Droidsans. https://droidsans.com/thai-gamer-esport-viewer-number

Khruahong, S., Surinta, O., & Lam, S. C. (2022). Sentiment analysis of local tourism in Thailand from YouTube comments using BiLSTM. In M. H. H. N. D. C. L. T. T. N. Vo (Eds.), *Lecture Notes in Computer Science: Vol. 13732. Multi-disciplinary trends in artificial intelligence* (pp. 169–177). Springer. https://doi.org/10.1007/978-3-031-20992-5_15

Phiphitphatphaisit, S., & Surinta, O. (2024). Multi-layer adaptive spatial-temporal feature fusion network for efficient food image recognition. *Expert Systems with Applications*, *255*, 124834. https://doi.org/10.1016/j.eswa.2024.124834

Shewale, R. (2024). *Steam statistics for 2024*. Demand Sage. https://www.demandsage.com/steam-statistics

Wresearch. (2022). *Thailand gaming market outlook (2022-2028)*. 6Wresearch. http://www.6wresearch.com/industry-report/thailand-gaming-market-outlook