JIT Mahasarakham University, Thailand
http://jit.it.msu.ac.th

Journal of Applied Informatics and Technology, 7(2), 268-282.

Research Article

Leveraging PyThaiNLP for Sentiment Analysis of Thai Online Text: A Comparative Study of Logistic Regression and Support Vector Machine

Sunisa Duangtham[1], **Setthaphong Lertritrungrot**[1], **Nattavadee Hongboonmee**[1], **Wansuree Massagram**[1,*]

[1] Department of Computer Science and Information Technology, Faculty of Science, Naresuan University, Phitsanulok 65000, Thailand

* Corresponding author: Wansuree Massagram, wansureem@nu.ac.th

Received:
3 May 2024
Revised:
28 June 2024
Accepted:
19 September 2024

Keywords:

PyThaiNLP, Sentiment Analysis, Thai Online Text.

Abstract: The objective of this study is to compare the performance of sentiment analysis models for Thai online text using the existing PyThaiNLP libraries. For extracting text from online sources to create a dataset, the text was manually categorized into positive, neutral, and negative sentiments. Data preprocessing involved removing punctuation marks, tokenizing, removing non-Thai characters, and bag of words creation. The data was then divided into training and testing sets to build models using three algorithms: logistic regression, logistic regression with stochastic gradient descent (SGD), and support vector machine (SVM). Upon comparison, the logistic regression model was found to perform the best – achieving accuracy of 80.73% with a 90:10 train-test split using the newmm word tokenization tool and the augmented dictionary. The accuracy for analyzing positive sentiment was 81.10%, for neutral sentiment, 80.16%, and for negative sentiment, 80.97%.

1. Introduction

The popularity of social media and online platforms in Thailand has fueled a vast amount of user-generated data expressed in Thai. Analyzing the sentiment within this data offers valuable insights into public opinion, brand perception, and social trends. However, sentiment analysis for online Thai

text presents unique challenges due to the dissimilarity from the formal written Thai. Thai language used in online media mostly does not adhere to the correct principles of the Royal Institute Dictionary. Instead, it often consists of intentional misspellings, newly coined words, and phonetic alterations to convey emotions (Noro-a *et al.*, 2018). Previous sentiment analysis models may not fully support the analysis of sentiment from Thai text in online media.

Furthermore, Thai relies on diacritics for tone differentiation, which can be easily omitted in an informal online setting. Additionally, Thai sentiment often incorporates cultural nuances and sarcasm, making automated analysis even more intricate.

The unique challenges in sentiment analysis for online Thai text stem from its informal nature, including intentional misspellings and phonetic alterations, complicating accurate interpretation. Omission of diacritics, crucial for tone differentiation, and the integration of cultural nuances and sarcasm further hinder automated systems from effectively analyzing sentiment.

This research aims to leverage the existing PyThaiNLP library of Python, which already includes functions for text preparation, such as word tokenization, Thai character checking, a Thai dictionary, and stopword removal. The study will explore the effectiveness of the following sentiment analysis techniques: logistic regression, stochastic gradient descent, and support vector machine on online Thai text, in order to create a modern sentiment analysis model for Thai language used on the internet, addressing the challenges of informal text, omitted diacritics, cultural nuances and sarcasm.

2. Related Work

This section provides an overview of the background for this study which includes challenges of online sentiment analysis, PyThaiNLP, literature review and the three chosen sentiment analysis models.

2.1 Online Sentiment Analysis

Online sentiment analysis is challenging for several reasons. Online text often utilizes informal language, slang, abbreviations, and emoticons that can be difficult for traditional sentiment analysis models to interpret accurately. These elements don't always adhere to formal grammar rules, making it harder for machines to understand the true sentiment.

Sarcasm and irony in online communication can also be hard for sentiment analysis tools to detect. Sarcastic statements might use positive words to express negative sentiment, confusing the model. Similarly, online communication can be ambiguous. Some words or phrases can have multiple meanings depending on the context. For example, "that's cool" could be positive or negative depending on the situation, posing a challenge for sentiment analysis models.

Understanding the sentiment of online text often requires considering the broader context of the conversation or post. Sentiment can be influenced by humor, cultural references, or the relationship between the participants. Moreover, online discussions within specific communities (e.g., gamers, K-Pop, financial forums, etc.) might have their own jargon and slang, requiring domain-specific knowledge for accurate sentiment analysis. Pang, Lee, & Vaithyanathan (2002) explored the difficulties of sentiment analysis for online movie reviews, highlighting the impact of informal language, subjectivity, and domain-specific vocabulary -- laying the groundwork for further research on sentiment analysis techniques that can handle the complexities of online communication.

Online Thai writing style vastly differs from formal Thai writing style. Noro-a *et al* (2018) discussed that formal Thai writing uses more complex vocabulary and adheres to stricter grammatical rules. Online Thai often incorporates slang, abbreviations, and emoticons for a more casual tone. It also uses shorter sentences, fragments, and informal sentence structures for quicker communication. Formal Thai writing employs a system of honorifics to show respect to the recipient. This involves using specific pronouns and sentence starters depending on the social hierarchy between the writer and reader. Online Thai tends to be less strict with honorifics, especially in casual conversations with less frequent courtesy markers. The majority of negative comments online tends to be very impolite. Table 1 summarizes the key differences between formal and online Thai writing styles.

2.2 PyThaiNLP

PyThaiNLP (2024) is an open-source Python library specifically designed for processing and analyzing Thai text. Its tool set consists of word tokenization, part-of-speech tagging, named entity recognition, text normalization, stemming and lemmatization, machine translation, and sentiment analysis. While PyThaiNLP offers sentiment analysis functionalities, online Thai text presents its own challenges mentioned previously. This study uses PyThaiNLP word tokenization during the data preparation process to break down text into individual words or meaningful units.

Table 1. Summary of key differences in formal vs online Thai

Feature	Formal Thai Writing	Online Thai Writing
Vocabulary	Complex, formal, correctly spelled.	Slang, abbreviations, emoticons, newly coined, intentionally misspelled
Sentence Structure	Complete, proper grammar	Shorter, informal structures
Honorifics	Used extensively	Less strict, more casual
Courtesy Language	Frequent politeness markers	Less frequent, less polite, more direct

2.3 Literature Review

Bowornlertsutee & Paireekreng (2022) proposed a study titled " The Model of Sentiment Analysis for Classifying the Online Shopping Reviews" utilizing techniques such as word tokenization, bag of words, and Stop-word Removal with the PyThaiNLP library in Python. They employed machine learning techniques including LSTM, SGD, logistic regression, and SVM for text analysis. The variables considered were positive sentiment, neutral sentiment, and negative sentiment. The accuracy of the analysis results was as follows: LSTM 81.27%, logistic regression 69%, SGD 66%, and SVM 65%. However, the model had limitations in analyzing ambiguous words.

Lisirikul & Numpradit (2018) proposed a study titled " Opinion Analysis System to Business by Text Mining On Twitter" utilizing techniques such as word tokenization and stop-word removal with the THSplitlib library in PHP. They employed naïve Bayes and SVM techniques for text analysis using the RapidMiner program. The variables considered were positive sentiment and negative sentiment. The accuracy of the analysis results was as follows: naïve Bayes 56.66% and SVM 76.47%. However, the model had limitations in analyzing neutral sentiment.

Chaisanguan & Romsaiyud (2018) proposed a study titled " Development of a Real-time Sentiment Analysis System of Students on Facebook Using Naive Bayes Classifier in Thai Language" They utilized the longest matching and maximal matching techniques for data preparation, and naïve Bayes classification for sentiment analysis with the RapidMiner program. Variables included positive sentiment, neutral sentiment, and negative sentiment. Data were collected using the Facebook Graph API. The analysis yielded the following accuracy metrics: Accuracy 97.60%, Precision 96.61%, Recall 96.50%, and F-measure 96.55%. However, the system had limitations: 1. It couldn't analyze conflicting sentences, and 2. It couldn't analyze incorrect text, leading to potential misinterpretations.

Aliman *et al* (2022) proposed a study titled "Sentiment Analysis using Logistic Regression," analyzing potential mental health crisis tweets using data preprocessing techniques based on Singh & Kumari (2016), including removing URLs, slang words, misspelled words, and "@" mentions. They employed support vector classifier, SGD, naive bayes, and logistic regression models for text analysis, The variables considered were positive sentiment and negative sentiment. The accuracy of the analysis results was as follows: logistic regression 81%, naive Bayes 77%, SGD 71%, and support vector classifier 69%. However, a limitation of this model is that it was developed for English, Filipino, and Taglish, not for Thai language.

2.4 Sentiment Analysis Model

Sentiment analysis models are algorithms that automatically identify the emotional tone (positive, negative, or neutral) within a piece of text. Learning from labeled

data, the models can be more adaptable to different types of text and can improve over time with more data. Common machine learning models used for sentiment analysis (Wankhade, Rao, & Kulkarni, 2022; Scikit-learn, 2024) are logistic regression, support vector machine (SVM), naïve Bayes, and neural networks. The latter two models can be effective for sentiment analysis; however, they require significant amount of data. Logistic regression estimates the probability of a text belonging to a specific sentiment class and can be effective for smaller datasets. SVM excels at finding clear boundaries between different sentiment classes, even for non-linear data – making it suitable for complex online text analysis.

1) Logistic Regression

Logistic regression is a statistical method commonly used in machine learning for classification tasks, especially when dealing with binary outcomes and interpretability is a concern. The core of logistic regression lies in finding the optimal coefficients for the linear equation that best separates the classes. This is achieved through an iterative process called gradient descent, where the model continuously adjusts the coefficients to minimize the error between predicted probabilities and actual class labels even in the small training data set – making it suitable to estimate the probability of an online text belonging to a specific sentiment class (positive, negative, neutral).

One approach to enhance logistic regression models is to use an optimization algorithm such as stochastic gradient descent (SGD) to train various models. SGD iteratively adjusts the model's internal parameters to minimize the error between predicted sentiment and actual labels in the training data. However, it can be slow to converge and often requires careful tuning of learning rate parameter to avoid overfitting.

2) SVM

SVM is another statistical method commonly used for classification tasks in machine learning. Unlike logistic regression, which excels with linearly separable data, SVMs can effectively handle both linearly separable and non-linearly separable data. An SVM aims to find a clear hyperplane that separates these positive, negative, and neutral classes with the maximum margin. A larger margin indicates a clearer separation between the classes -- making it easier to classify new data points accurately and is better for generalization on unseen data.

Moreover, when dealing with non-linearly separable data, SVMs employ a kernel trick that involves transforming the data into a higher-dimensional space where it becomes linearly separable. The SVM then operates in this higher dimension to create the optimal separation hyperplane. Common kernels used include linear, polynomial, and radial basis function (RBF).

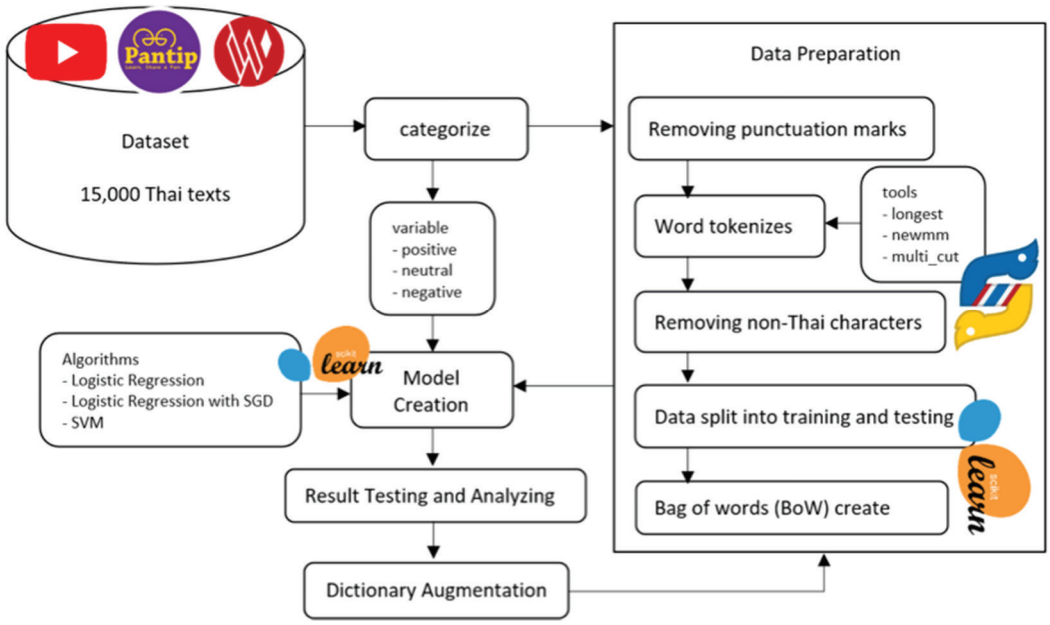

Figure 1. Illustration of system overview diagram

In sentiment analysis of online Thai text, both logistic regression and SVM models have their merits. If the dataset is small with potentially linear relationships, logistic regression might be effective enough. However, for complex, non-linear language nuances often present in online text, SVMs could be a more robust choice, especially with larger datasets. Hence, this study experiments with both models and compares their performance on our data to determine the best approach for this sentiment analysis task for online Thai text.

3. Materials and Methods

This section details the approach employed to analyze sentiment within the chosen dataset. The steps taken to prepare the text for analysis are described. Following this, the specific sentiment analysis techniques utilized are explained, along with the evaluation metrics chosen to assess the model's performance, as summarized in the system overview diagram in Figure 1 and elaborated further in sections 3.1 to 3.5.

3.1 Dataset

The data collection process involved amassing a corpus of 15,000 Thai-language texts. These texts were sourced from comments on YouTube, Pantip, and a subset from the Wisesight Sentiment dataset (Suriyawongkul *et al*., 2019). The collected texts were free topics to ensure that the sentiment analysis model created could comprehensively predict online texts. The collection period spanned from March 2023 to September 2023. These texts were manually edited to correct the wrong typing, except for those intentionally

wrong typing to convey emotions. They were manually categorized into three sentiment categories: positive, neutral, and negative, with each category comprising 5,000 records. The text that expresses admiration, care, interests, and likes were classified as positive. Questions, commands, and ordinary narratives were classified as neutral. Texts containing insults or provocations were classified as negative. These texts were then recorded into a .csv file in their corresponding category. Examples of each text category can be seen in Figure 2. for positive, neutral, and negative respectively.

3.2 Data Preparation

1) Removing punctuation marks such as ?, . ; : ! “ๆ ฯ > < / - . These punctuation marks were omitted in this study to simplify the model; however, they will be taken into consideration in our future works.

2) Tokenization using the word tokenize function from the PyThaiNLP library (Kanoktipsatharporn, 2020). The words were tokenized according to the existing PyThaiNLP dictionary with three word-tokenization tools, which are already available in PyThaiNLP and are low computational resource: longest matching, newmm (maximal matching with Thai Character Cluster), and multi_cut (maximal matching). The accuracies of each tokenization method were then compared.

3) Parsing only Thai characters by removing non-Thai characters was achieved using the Thai character checking function from PyThaiNLP library. This step also involved removing emojis since they are not considered Thai characters.

The example results from the first three steps of data preparation are compared with their original text in Table 2.

4) Data splitting by dividing the dataset into training and testing sets using three different ratios: 80% training and 20% testing, 85% training and 15% testing, and 90% training and 10% testing, to compare their accuracy.

5) Bag of words (BoW) creation by constructing a vocabulary from the words in the training data. The BoW was achieved by converting the words into numerical

ของดีมากครับร้านเจ้า	pos
ของแถมอีกอย่างนึงคือตรงที่ช่วยยืดอายุแบตเตอรี่แหละ ผมคิดมาจะ 3 ปี และแบยังใช้อยู่	pos
ของโปรด. อร่อยทุกอย่างงๆๆ. หิวทีไรก็นึกถึงทุกทีเลย	pos
ขอชื่นชม สว.อำพค่ะ	pos
ขอชื่นชมพนักงานส่งพัสดุพิเศษและพนักงานรับยอดเดือนได้มากค่ะ ส่งที่อยุธยา เก่งมากที่หาบ้านเราเจอ	pos

(a)

ชุดความคุมพร้อุปสมาจากนอกไม่ได้ถูกปรับแต่งมาสำหรับทีมส์ที่ดีดในเมืองไทย	neu
ชุดคัสสรร.ราคาเท่าไหร่คะ	neu
ชุดดอกไม้ยังมีอยู่ไหมคะ	neu
ชุดเท่าไรคะแบบนี้	neu
ชุดนี้กิน 3 คนจะอิ่มป่าว	neu

(b)

กรรมการเลือกตั้งไม่ใช่กรรมการวิจารณ์ที่จะมาโกงกันครับ	neg
กรรมาฯสั่งข้าราชการปิดโอกาสประชาชนและประชาชน	neg
กรรมใดใครก่อกรรมมันย่อมตามสนอง	neg
กรรมดอนแท้	neg
กรรมย้อนหลังของมีผ้านาจากศิลธรรมมานำดอนของตอนนี้	neg

(c)

Figure 2. Example of (a) positive, (b) neutral, and (c) negative text

representations. Then, using this vocabulary as features to count the occurrences of each word in the text, resulting in a sparse matrix representation.

3.3 Model Creation

To conduct a comparative analysis, three sentiment analysis models for Thai text were constructed utilizing the existing PyThaiNLP libraries for the algorithms mentioned in section 2.4: logistic regression, logistic regression with SGD, and SVM. The selection of these three algorithms is based on their low computational resource requirements for training, their ability to handle high-dimensional datasets, and the ease of using their functions from scikit-learn. Their parameter settings with the following details.

1) Logistic Regression

The models were created with penalty = 'l2', dual = False, tol = 0.0001, C = 1.0, fit intercept = True, intercept_scaling = 1, class_weight = None, random_state = None, solver = 'lbfgs', max_iter = 10,000, multi_class = 'auto', verbose = 0, warm_start = False, n_jobs = None, l1_ratio = None. These parameters are all default values from scikit-learn, except for max_iter, which was set to 10,000 to increase the iterations and ensure the solutions converge with the scale of the dataset.

2) Logistic Regression with SGD

The models were created with loss = 'hinge', penalty = 'l2', alpha = 0.0001, l1_ratio

Table 2. Example results from first three steps of data preparation

Data preparation step		Example 1	Example 2	Example 3
Original text		ซื้อของจากออนไลน์ราคาน่ารักส์ๆ ไม่แพง:D	พวกคุณเป็นความหวังของพวกเรานะคะ <3	ไปไกลๆตีนเลยย 😠
1) Removing punctuation		ซื้อของจากออนไลน์ราคาน่ารักส์ๆไม่แพงD	พวกคุณเป็นความหวังของพวกเรานะคะ 3	ไปไกลตีนเลยย 😠
2) Tokenization	longest	['ซื้อ', 'ของ', 'จาก', 'ออนไลน์', 'ราคา', 'น่า', 'รัก', 'ส์', 'ไม่', 'แพง', 'd']	['พวกคุณ', 'เป็นความ', 'หวัง', 'ของ', 'พวกเรา', 'นะคะ', ' ', '3']	['ไป', 'ไกล', 'ตีน', 'เลย', 'ย', ' 😠']
	newmm	['ซื้อ', 'ของ', 'จาก', 'ออนไลน์', 'ราคา', 'น่า', 'รักส์', 'ไม่', 'แพง', 'D']	['พวกคุณ', 'เป็น', 'ความหวัง', 'ของ', 'พวกเรา', 'นะคะ', ' ', '3']	['ไป', 'ไกล', 'ตีน', 'เลย', 'ย', ' ', '😠']
	multi_cut	['ซื้อ', 'ของ', 'จากออนไลน์', 'ราคาน่ารัก', 'ส์', 'ไม่', 'แพง', 'D']	['พวกคุณ', 'เป็นความหวัง', 'ของ', 'พวกเรา', 'นะคะ', ' ', '3']	['ไป', 'ไกล', 'ตีน', 'เลย', 'ย', ' ', '😠']
3) Parsing only Thai characters		ซื้อ ของ จาก ออนไลน์ ราคา น่า รักส์ ไม่ แพง เลย	พวกคุณ เป็น ความหวัง ของ พวกเรา นะคะ	ไป ไกล ตีน เลย ย

= 0.15, fit_intercept = True, max_iter = 10,000, tol = 0.001, shuffle = True, verbose = 0, epsilon = 0.1, n_jobs = None, random_state = None, learning_rate = 'optimal', eta0 = 0.0, power_t = 0.5, early_stopping = False, validation_fraction = 0.1, n_iter_no_change = 5, class_weight = None, warm_start = False, average = False. These parameters are all default values from scikit-learn, except for max_iter, which was set to 10,000 to increase the iterations and ensure the solutions converge with the scale of the dataset.

3) SVM

The models were created with C = 1.0, kernel = 'rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None. These parameters are all default values from scikit-learn.

3.4 Dictionary Augmentation for Word Tokenization

After the models were built, it was observed that words prefixed with "ไม่» (meaning "not"), e.g. "ไม่ชอบ» (not like), "ไม่หล่อ» (not handsome), "ไม่แพง» (not expensive), and "ไม่รัก» (not love) were incorrectly segmented. Additionally, some words, such as abbreviations, slang, proper nouns, and commonly misspelled words, e.g. "อิดอก» (bitch), "ทปอ» (CUPT), "กกต» (ECT), "ลาซาด้า» (Lazada), and "แปลงสีฟัน» (toothbrush) were also incorrectly segmented. Forty of these words, comprising 14 words prefixed with "not", 2 abbreviations, 2 slangs, 5 loanwords, 11 proper nouns, and 6 commonly misspelled words, were augmented into the

Table 3. Accuracy and precision results before dictionary augmentation

Precision (%)	Trian (%)	Logistic Regression				Logistic Regression with SGD				Support Vector Machine (SVM)			
		Accuracy (%)	Precision (%)			Accuracy (%)	Precision (%)			Accuracy (%)	Precision (%)		
			pos	neu	neg		pos	neu	neg		pos	neu	neg
longest	80	78.80	80.10	77.16	79.15	77.80	80.20	76.27	76.92	74.13	71.74	73.19	77.53
	85	79.47	80.96	78.84	78.54	78.93	80.96	79.51	76.25	74.62	71.71	73.85	78.41
	90	80.33	81.30	79.38	80.36	79.53	80.49	79.38	78.74	75.47	73.17	73.74	79.55
newmm	80	78.00	79.40	76.46	78.14	78.37	79.70	76.46	78.95	74.30	71.54	72.99	78.44
	85	79.11	80.83	78.44	78.00	78.62	80.44	77.49	77.87	74.67	72.23	73.32	78.54
	90	80.33	81.30	79.18	80.57	80.33	80.08	80.35	80.57	75.47	73.98	72.76	79.76
multi_cut	80	76.07	77.01	74.18	77.02	76.07	77.01	75.17	76.01	72.37	67.46	72.10	77.63
	85	77.11	78.36	75.61	77.33	77.07	79.66	75.61	75.84	73.07	68.71	73.18	77.46
	90	77.87	78.66	76.65	78.34	77.33	77.85	76.26	77.94	73.80	69.51	73.35	78.54

Table 4. Accuracy and precision results after dictionary augmentation

Tokeniza-tion	Trian (%)	Logistic Regression				Logistic Regression with SGD				Support Vector Machine (SVM)			
		Accu-racy (%)	Precision (%)			Accu-racy (%)	Precision (%)			Accu-racy (%)	Precision (%)		
			pos	neu	neg		pos	neu	neg		pos	neu	neg
longest	80	78.77	80.10	77.36	78.85	78.10	80.70	75.77	77.83	74.80	72.34	73.49	78.64
	85	80.36	81.49	79.38	80.16	80.09	81.49	80.19	78.54	74.98	72.62	73.99	78.41
	90	80.40	81.10	79.38	80.77	80.47	81.30	79.18	80.97	75.87	73.98	73.15	80.57
newmm	80	78.40	79.70	76.96	78.54	78.40	80.20	76.96	78.04	74.60	71.94	73.49	78.44
	85	79.51	80.70	78.44	79.35	79.42	81.62	78.03	78.54	74.89	72.62	73.05	79.08
	90	80.73	81.10	80.16	80.97	80.67	80.69	79.96	81.38	76.00	74.19	73.54	80.36
multi_cut	80	76.67	77.31	75.17	77.53	76.57	78.21	75.27	76.21	72.83	67.96	73.09	77.53
	85	78.00	78.10	77.49	78.41	77.56	78.62	75.88	78.14	73.56	69.10	74.26	77.46
	90	78.27	78.86	76.26	79.76	78.00	78.05	76.07	79.96	74.07	69.92	73.74	78.54

original tokenization dictionary. These forty words were identified by examining the test data to see whether the model's predictions were correct. All models were then recreated using this enhanced dictionary. The analysis results were compared with those obtained from the original dictionary to assess the accuracy improvement.

3.5 Result Testing and Analyzing

Accuracy and precision are commonly used matrices for sentiment analysis. There mathematical representations are shown in equations (1) and (2), respectively. Accuracy measures the overall correctness of the model by indicating what percentage of sentiment classifications the model is correct. Precision focuses on the positive predictive value, i.e. out of the instances the model classified as positive sentiment, what percentage were, indeed, positive. Thus, while accuracy is desirable, precision is also needed to analyze the performance of the models.

For example, in sentiment analysis for social media, a model with high accuracy might classify most tweets correctly, but also label many neutral tweets as positive. This inflates its accuracy but misses the mark. A lower accuracy model with high precision might miss some relevant tweets, but excels at identifying true positive sentiment, which might be more valuable for the company.

$$Accuracy = \frac{Numberof CorrectPrediction}{TotalNumberofPredictions} \quad (1)$$

$$Precision = \frac{TruePositives}{TruePositives+FalsePositives} \quad (2)$$

4. Experimental Results

As explained in the previous section, three different algorithms were used to create the sentiment analysis models. The tokenization dictionary was also enhanced to improve the model performance. The accuracy and precision performance of each model is reported here.

4.1 Model Comparison

Upon building sentiment analysis models for Thai text from the existing PyThaiNLP libraries to perform word tokenization, various train-test splits, algorithm selections, and dictionary augmentations, the comparison yields the following accuracy percentages (%). The results of comparison can be seen in Tables 3 and 4 for the before and after dictionary augmentations. From Tables 3 and 4, it can be observed that the model providing the highest accuracy is generated from the logistic regression algorithm after dictionary augmentation, using the newmm tokenization tool with a 90:10 split for training and testing data. Additionally, the model yielding the highest precision for predicting positive and negative texts is generated from the logistic regression with SGD algorithm after dictionary augmentation, using the newmm tokenization tool with 85:15 and 90:10 splits for training and testing data, respectively. Moreover, the model providing the highest precision for neutral texts is generated from the logistic regression algorithm before dictionary augmentation, using the newmm tokenization tool with a 90:10 split for training and testing data.

4.2 Results of Sentiment Analysis Precision Selected Model

Based on the comparison of models, it was found that the Logistic Regression model with a 90:10 train-test split using the newmm word tokenization tool and an augmented dictionary consistently exhibits the highest accuracy for both before and after dictionary augmentation. Therefore, this model was selected to evaluate precision, recall, and f-measure using equations (2), (3), and (4) to determine if the values were sufficiently high for practical use. The model's computational time per text was also assessed using equation (5) to measure the speed of text prediction.

$$Recall = \frac{TruePositives}{TruePositives+FalseNagative} \quad (3)$$

$$F-measure = 2\frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

$$time = \frac{totalcomputationaltimefor the testset}{numberoftextsinthetestset} \quad (5)$$

The confusion matrix of the best model is shown in Figure 3. From the 1,500 test samples, this model correctly predicted 399 positive texts, misclassifying 66 as neutral and 27 as negative. It correctly predicted 412 neutral texts, misclassifying 53 as positive and 49 as negative. It correctly predicted 400 negative texts, misclassifying 27 as positive and 67 as neutral. The precision of predicting positive, neutral, and negative texts, calculated

using equation (2), was 81.10%, 80.16%, and 80.97%, respectively. The recall for positive, neutral, and negative texts, calculated using equation (3), was 83.30%, 75.60%, and 84.30%, respectively. The f-measure for positive, neutral, and negative texts, calculated using equation (4), was 82.18%, 77.81%, and 82.47%, respectively.

The model calculated the total computational time for the test set as 0.0010001659393310547 seconds with 1,500 texts in the test set. Therefore, using equation (5), the computational time per text was 0.0000006668 seconds.

The model's performance before and after dictionary augmentation were improved throughout the board as a result of only additional forty new words in the existing data. Furthermore, the accuracy, precision, recall, and f-measure are high for the best performance model indicating that the model performs well across different sentiment categories. Additionally, the computational time per text is very low, making this model suitable for practical use.

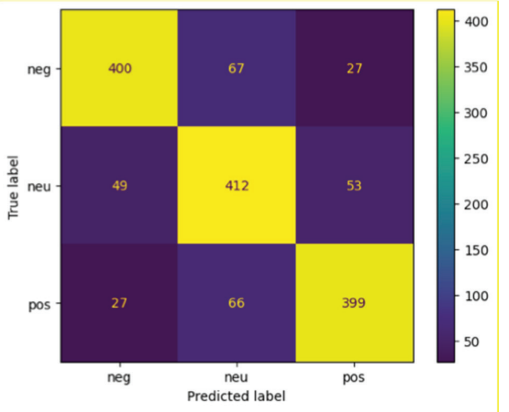

Figure 3. Confusion Matrix: Results of the analysis of texts by the best performance model.

5. Discussion

5.1 Comparison of Data Splitting

From the comparison of accuracy values of sentiment analysis models for online Thai text, it was found that the amount of data used for training has an impact on accuracy. The more data used for training, the higher the accuracy of the model. As seen from Tables 3 and 4, models trained with 90% of the data have higher accuracy than those trained with 85%, and models trained with 85% of the data have higher accuracy than those trained with 80%.

5.2 Comparison of Word Tokenization

In comparing the accuracy values of sentiment analysis models for online Thai text, it was found that the multi_cut tokenization tool, which employs maximal matching, resulted in the lowest accuracy. This was observed when compared with models using longest matching and newmm tokenization tools, which provided similar accuracy values. The models using the newmm tokenization tool consistently showed the highest accuracy, as seen in Table 4.

5.3 Comparison of Dictionaries

The addition of vocabulary words to the existing dictionary from the PyThaiNLP

library resulted in an increase in the accuracy of the model. This can be seen from the results in Table 3 (before dictionary augmentation) and Table 4 (after dictionary augmentation). The same algorithm, tokenization tool, and train-test split were used, but the accuracy in Table 4 is higher because the model in Table 4 had an augmented dictionary before tokenization. This allowed the model to recognize more words, leading to more accurate tokenization and predictions. For example, before the dictionary augmentation, the text "ไม่หล่อเลย" ("not handsome at all") was tokenized as "[ไม่, หล่อ, เลย]" ("[not, handsome, at all]"), leading to an incorrect prediction of positive sentiment because the word "หล่อ" (handsome) was present in the text. However, the true sentiment of this text is negative as it conveys an insult. With the augmented dictionary, the text was tokenized as "[ไม่หล่อ, เลย]" ("[not handsome, at all]") and correctly predicted as negative sentiment. This illustrates why models with an augmented dictionary before tokenization predict texts more accurately and have higher accuracy compared to models without dictionary augmentation.

5.4 Comparison of Algorithms

In comparing the accuracy values of sentiment analysis models for Thai text, it was found that the SVM algorithm yielded the lowest accuracy. This was observed that the models using logistic regression (with or without SGD) provided similar accuracy values. Logistic regression without SGD yielded the highest accuracy, while logistic regression with SGD achieved the highest precision for each text category.

Algorithms used in the model comparisons utilized default parameters from scikit-learn. This resulted in the SVM algorithm having the lowest accuracy, followed by logistic regression with SGD and then without SGD. Adjusting the parameters in future research might yield different results.

6. Conclusion

The comparison of models revealed that the model generated from the logistic regression algorithm, with a 90:10 split for training and testing data, using the newmm tokenization tool with an augmented dictionary, achieved the highest accuracy of 80.73%. Additionally, it exhibited a precision of 81.10% for analyzing text with positive sentiment, 80.16% for text with neutral sentiment, and 80.97% for text with negative sentiment. The dictionary augmentation shows a clear improvement for all the models despite adding only a few words. The future direction of this work will be improving the dictionary for better performance of the online Thai text sentiment analysis.

This model is suitable for predicting sentiment in Thai texts on online media. However, it may not be appropriate for predicting sentiment in academic articles or articles requiring deep understanding. This is because the dataset consists of Thai language texts from online media with free topics, and

the data preparation was specifically tailored for such texts.

Acknowledgments

I would like to express my gratitude to the PyThaiNLP development team for their providing a convenient platform for processing Thai language text. I am also thankful to the owners of online content whose texts served as dataset for this research project.

References

Aliman, G. B., *et al*. (2022). Sentiment analysis using logistic regression. *Journal of Computational Innovations and Engineering Applications*, *11*(7), 36–40. https://doi.org/10.9790/9622-1107023640

Bowornlertsutee, P., & Paireekreng, W. (2022). The model of sentiment analysis for classifying the online shopping reviews. *Journal of Engineering and Digital Technology (JEDT)*, 10(1), 71-79. https://ph01.tci-thaijo.org/index.php/TNIJournal/article/view/246375 [In Thai].

Chaisanguan, S., & Romsaiyud, W. (2018). Development of a real-time sentiment analysis system of students on Facebook using Naive Bayes classifier in Thai language. *Proceedings of the 8th STOU National Research Conference* (pp. 521–535). Sukhothai, Thailand, November 23, 2018 [In Thai].

Kanoktipsatharporn, S. (2020). Python ตัดคำภาษาไทย ด้วย PyThaiNLP API ตัดคำ Word Tokenize ภาษาไทย ตัวอย่างการตัดคำภาษาไทย อัลกอริทึม deepcut, newmm, longest, pyicu, attacut – PyThaiNLP ep.2. Retrieved 12 October 2023, from https://www.bualabs.com/archives/3740/python-word-tokenize-pythainlp-example-algorithm-deepcut-newmm-longest-python-pythainlp-ep-2/ [In Thai]

Lisirikul, C., & Numpradit, J. (2018). Opinion analysis system to business by text mining on Twitter. *Proceedings of the 14th National Conference on Computing and Information Technology* (pp. 408–413). Chiang Mai, Thailand, July 5–6, 2018.

Noro-a, S., Charong, N., Musigcharoen, N., & Yossiri, V. (2018). Features of language used of Thai teenagers on social media. *Proceedings of the 9th Hatyai National and International Conference* (pp. 940–952). Songkhla, Thailand, July 20, 2018. [In Thai]

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)* (pp. 79–86). Association for Computational Linguistics. https://aclanthology.org/W02-1011/

PyThaiNLP. (2024). PyThaiNLP: Thai natural language processing in python. Retrieved 31 March 2024, from https://github.com/PyThaiNLP/pythainlp

Scikit-learn. (2024). 1. Supervised learning. Retrieved 14 March 2024, from https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

Singh, T., & Kumari, M. (2016). Role of text pre-processing in Twitter sentiment analysis. *Procedia Computer Science*, *89*, 549–254. https://doi.org/10.1016/j.procs.2016.06.095

Suriyawongkul, A., Chuangsuwanich, E., Chormai, P., & Polpanumas, C. (2019). Wisesight sentiment corpus. Retrieved 11 October 2023, from https://github.com/PyThaiNLP/wisesight-sentiment

Wankhade, M., Rao, A. C., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review, 55*(7), 5731-5780. https://link.springer.com/article/10.1007/s10462-022-10144-1