INDOCHINA APPLIED SCIENCES

K. Matarat / Indochin. Appl. Sci. 14(3) (2025) 262648

Research Article

Aspect-Based Sentiment Analysis in Thai Texts: A Comparative Study of Machine Learning and Neural Network Approaches

Korakot Matarat *

Faculty of Science and Technology, Sakon Nakhon Rajabhat University, Sakon Nakhon, 47000 Thailand

***Corresponding authors**: *korakot@snru.ac.th*

https://doi.org/10.55674/ias.v14i3.262648

Received: 17 June 2025 ; **Revised:** 22 August 2025 ; **Accepted:** 30 September 2025 ; **Available online:** 14 Novembre 2025

Abstract

Efficiently classifying messages into document categories is a fundamental task in natural language processing, crucial for organizing and extracting insights from vast amounts of textual data. This paper explores the application of machine learning algorithms, particularly neural networks incorporating contextual and linguistic semantics, for the purpose of classifying texts. Unlike traditional subject-based classification, the focus here is on overall judgment, posing unique challenges. This study examines aspect-based sentiment analysis (ABSA), which depends on accurate text classification to identify entity aspects and their associated sentiments. Using Thai language review data and a list of 400K food words, the research compares several classifiers: Naive Bayes, Linear SVM, Logistic Regression, and Bag of Words (BoW) with Keras. Results show that BoW with Keras performs best, achieving 97 % accuracy after 10 training rounds, with steady improvements in accuracy and loss reduction across epochs. This paper not only presents models and methodologies applicable to Thai-language text classification but also introduces a proposed method for measuring Thai sentiment. While the study provides valuable insights, it acknowledges the necessity for considering diverse configurations and requirements, as alternative classifiers may yield comparable or superior results. The findings herein contribute to the ongoing discourse in the field and offer a foundation for further exploration and refinement of classification techniques in Thai language text processing.

Keywords: Text classification; Machine learning; Thai language algorithms; Natural language processing; Bag of words with keras

1. Introduction

The Thai language, rich in linguistic diversity, has witnessed a significant surge in scientific research focused on text categorization challenges over the past decade. While advancements in Thai text classification have been noteworthy, the escalating volume of Thai language content on the Internet presents new and evolving challenges. This is particularly evident in domains like the hospitality industry, where platforms like Agoda host over a hundred reviews for each hotel. Navigating through this abundance of information becomes a daunting task for travelers seeking relevant details about significant places.

In the realm of document categorization, the pivotal task is the selection of an optimal classification algorithm. As the volume of structured and semistructured electronic documents continues to grow, the role of text mining (TM) becomes increasingly crucial. Recently, deep learning has emerged as a powerful alternative to traditional machine learning methods for text categorization. Among these approaches, the Convolutional Neural Network (CNN), originally designed for image analysis, has proven highly effective in processing text. CNNs are capable of identifying important local features, such as word patterns and contextual phrases, which are essential for accurate classification. The integration of text classification frameworks with data processing frameworks proves instrumental in effectively categorizing diverse document classes. Notably, the Naive Bayes Classifier for Multinomial Models, Linear Support Vector Machine, and Logistic Regression have emerged as robust techniques for supervised text classification, known for their simplicity and effectiveness. Previous research on Thai text classification has explored both traditional machine learning and deep learning approaches, yet comprehensive comparisons remain limited, especially on real world Thai datasets. For example, "An Efficient Deep Learning for Thai Sentiment Analysis" [36] applied CNN and LSTM models for sentiment analysis of Thai reviews on TripAdvisor, demonstrating the effectiveness of deep learning techniques. Similarly, "A Comparative Study of Sentiment Analysis on Customer Reviews Using Machine Learning and Deep Learning" [37] evaluated Logistic Regression, Naive Bayes, Random Forest, RNN, and CNN on customer review datasets, revealing notable differences in accuracy, precision, recall, and F1-score across models. Collectively, these studies indicate a gap in systematic evaluations of traditional versus deep learning methods for Thai text. The present study addresses this gap by comparatively assessing multiple classification models on Thai language data, offering practical insights for efficient text categorization.

This research endeavors to address a fundamental question: "Which machine learning model should be employed for text categorization?" Through a comprehensive examination of various text categorization models, the study aims to identify the most suitable approach for specific circumstances. The primary contributions of this work unfold across three key steps. Firstly, the study evaluates the performance of the selected algorithm on text blocks post stop word removal and tokenization. Secondly, the authors introduce the TF-IDF approach to enhance result precision. Thirdly, a shift to Bag of Words (BoW) is proposed for comparative analysis. Further, the integration of BoW with Keras is recommended, providing a novel perspective on algorithm and token application in phrases and texts.

The models and methodologies presented herein are specifically tailored for the classification of the Thai language and lay the groundwork for future research in this domain. Leveraging Naive Bayes Classifier, Linear Support Vector Machine, Logistic Regression, and Bag of Words (BoW) with Keras, the study introduces algorithms designed for the text classification analysis of reviews related to attractions in the Wongnai data. The paper unfolds with Section 2 delving into related work, Section 3 exploring various techniques, Section 4 presenting experiments and results, and Section 5 concluding with a discussion on future research directions.

2. Literature Review

Upon examining the theory and associated research, it was discovered that there are numerous articles on text classification. In this article, he explored the categorization of multi-class text in the finance domain. Then he evaluated the added value of FinBERT, a PLM suited to the financial industry. On the other hand, he discovered that FinBERT was unable to outperform generic PLMs in his financial document classification assignment. He tested whether custom terminology could improve Finbert's performance [1]. In this study, he discusses the standard text classification models used to categorize Arabic texts, corpora, and documents into several groups. In addition, it compares various methodologies for the classification of Arabic literature [2]. In this paper, he compares several machine learning algorithms using the TF-IDF representation, such as Nave Bayes and its derivatives, SVM, and random forest classifiers. To improve the quality of the used classifiers, the recognition rate, for the tested systems, is satisfied, where the system based on naïve Bayes classifier, the TF-IDF weighting terms, and the info gain select attributes method gives 98.70% as accuracy [3]. This study suggests a comprehensive method for aspect-based sentiment analysis. For aspect-based sentiment analysis, many older systems employed single label classifiers. This system has produced superior results in comparison to other systems. This experiment demonstrates that the bidirectional encoder representation from the Transformers system gets considerable results by taking the bidirectional context of tokens in a phrase into account [4]. This paper aims to provide a detailed introduction to the Keras machine learning framework in Tensor Flow. In conjunction with PyTorch, CODEEPNEATM, and Pygame, the aforementioned package enables the integration of deep learning models in the applied domain. In addition, the author describes notable results and findings gained utilizing this methodology [5]. This study introduces a novel text classification model called attention-based BiLSTM combined CNN with gating mechanism (ABLG-CNN). Word2vec is used to train ABLG-word CNN's vector representation. The experimental results indicate that the classification performance of ABLG-CNN surpasses that of other contemporary text classification methods [6]. This study compares the representational performance of the TF-IDF and Word2Vec models for emotional text classification. He used the support vector machine (SVM) and multinomial Nave Bayes (MNB) techniques to classify emotional tweets from commuter lines and transjakarta. This study demonstrates that TF-IDF modeling outperforms Word2Vec modeling, and it enhances the classification performance of prior studies [7]. In this study, the author investigates the results of applying three distinct text feature extraction strategies while classifying short sentences and phrases into categories using a neural network. The results reveal that the TF-IDF feature extraction strategy outperforms competing methods, allowing the classifier to achieve the highest level of precision when working with larger datasets [8]. This research compares k-nearest neighbors, naive bayes, and support vector machines for news categorization. Using a number of variables and preprocessing steps, this demonstrates that k-Nearest Neighbor can compete with Support Vector Machine in terms of accuracy, while Naive Bayes produces a mediocre result, not as good as k-Nearest Neighbor and Support Vector Machine but as bad as k-Nearest Neighbor and Support Vector Machine ever achieve. K-Nearest Neighbor with the correlation measurement type yields the best outcome for this experiment [9]. In this research, he introduces a novel graph transformer-based deep learning model for large-scale multilabel text categorization. He generates a representation of the labels based on their hierarchical relationships and designs a weighted loss function based on their semantic distances. Extensive experiments conducted on three benchmark datasets demonstrated that the proposed model can realistically represent the text's hierarchy and logic and surpass existing methodologies [10]. In this paper, instead of a single graph for the entire corpus, he proposes a new GNN-based model that constructs graphs for each text input by sharing global parameters. This strategy eliminates the weight of dependence between a single text and the entire corpus, enabling online testing while preserving global data. In addition, he constructs graphs using much smaller text windows, which not only extract more local features but also significantly reduce the number of edges and memory consumption. Experiments demonstrate that, although requiring less memory, our approach outperforms existing methods on multiple text categorization atasets [11]. This article describes and evaluates the efficacy and reliability of different supervised learning models, including logistic regression (LR), decision trees (DT), support vector machines (SVM), AdaBoost (AB), random forest (RF), multinomial naive bayes (NB), and multilayer perceptrons (MLP) (MLP). A comprehensive evaluation of their performance was conducted in this paper. Further research into the usage of various SVM kernels revealed that linear kernels are superior to polynomial, sigmoid, and radial basis function kernels for text categorization. Also investigated were the effects of removing stop words on model performance; DT performed better when stop words were eliminated, whilst other models were largely unaffected by their existence or absence [12]. To train a chatbot on Islamic jurisprudence, a text classifier is required to construct a robust knowledge base. This research utilized a common methodology to develop a text classification model. To classify the text data, machine learning algorithms such as Bayesian networks and Naive Bayes were utilized. The Nave Bayes method is more accurate in all assessment models, 84.25 % when using the training set and 76.50 % when using 10-fold cross-validation, based on the results of experimental testing. In contrast, the Bayesian network technique requires less time to evaluate all models. Thus, it can be stated that the text classification model based on Nave Bayes and the String to Word Vector filter has the potential to be utilized efficiently [13]. In this paper, a novel hybrid text categorization model based on deep belief networks and softmax regression is proposed. A deep belief network is proposed in order to handle the sparse, high-dimensional matrix computation problem of text data. Deep belief networks and softmax regression are initially learned via pretraining methods. The system parameters are then optimized using a limited memory Broyden–Fletcher–Goldfarb–Shanno algorithm during the phase of fine-tuning. Experimental results on the Reuters-21,578 and 20-Newsgroup corpora demonstrate that the proposed model can converge at the fine-tuning stage and outperforms the conventional algorithms, such as SVM and KNN, by a large margin [14]. Using the Twitter Streaming API, a set of tweets was extracted for this study. The retrieved data was in text format, so he had to clean it, eliminate redundancies, and turn it into a format that computational power can utilize to determine the sentiment of tweets. He utilized Word2Vec to accomplish this and then his deep learning system to determine and categorize the tweets as positive or negative. As he is aware, he has utilized Keras, which is a Python-based, open source neural network library. It is compatible with TensorFlow, Theano, and PlaidML. In addition, he would seek out datasets from online forums such as Reddit and Quora, which would enhance the population of the data being analyzed, allowing us to get better findings and obtain a clearer picture of the current and prevalent sentiment [15]. This report outlines recent strategies and trends in the text categorization algorithm, which are explored in this overview. However, existing text classification algorithms function more effectively if you have a deeper grasp of feature extraction methods and how to accurately evaluate them. Currently, text categorization algorithms can be categorized mostly as follows: Term Frequency Inverse Document Frequency (TF-IDF), term frequency (TF), and word-embedding (e.g., Word2Vec, contextualized word representations, Global Vectors for Word Representation (GloVe), and FastText) are frequently utilized in academic and commercial applications. The focal point of this study. Explanations are provided for evaluation methods such as precision, F, the Matthew correlation coefficient (MCC), receiver operating characteristics (ROC), and area under the curve (AUC). With these measures, the method for text classification may be evaluated [16]. In this research, the author demonstrates that capsule networks have the capacity to classify text and possess various advantages over convolutional neural networks. He contrasts his suggested model with the earliest research on capsule network-based text classification. Additionally, he proposes a straightforward routing technique that efficiently minimizes the computational cost of dynamic routing [17]. In his research, he attempts to develop a deep learning model that achieves more accurate classifications of Chinese text. LSTM is a subtype of a recurrent neural network (RNN) that can process serialized data via its recurrent structure. Therefore, he incorporated two layers of LSTM and one layer of CNN into his new model. In the results, the model demonstrated exceptional text classification performance, particularly for Chinese languages [18]. In this study, he presents a strategy based on deep learning for assessing the sentiment of product reviews. The key idea of this work is that deep learning models employ word2vec to construct word embeddings and convolutional neural networks to train and classify the sentiment classes of product reviews. This program can predict the tone of brand-new product reviews [19]. In this study, six machine learning technologies were employed to accomplish the classification of multiple text classes. 5-fold cross-validation and grid search techniques were employed to identify the classifier hyperparameters. When extracting features, he utilized various n-gram models, including unigram, bigram, trigram, and fourgram. Classifiers based on machine learning were applied to each case, and results were produced. In several instances, the character level n-gram produced better results than the word level n-gram. The RBF SVM classifier with TF-IDF and the character-level four-gram feature selection technique achieved the highest accuracy of 86.88% [20]. This study revealed the findings of his research, according to which logistics regression is the most effective classifier, followed by linear SVC. A further example indicates that while the weighted-average f1 score is the most crucial metric, the performances of logistic regression, decision trees, multinomial Nave Bayes, linear SVC, and random forest are comparable [21]. This work investigates and discusses theoretical and practical experience and proposes a CNN-LSTM hybrid model-based text classification technique compared to CNN and LSTM utilizing the word2vec method. He developed a word vector dimension comparison experiment and a time comparison experiment, both of which imply that the hybrid model described in this study is preferable in the text domain [22]. This study will discuss the performance of classification algorithms for text-based data, with supervised learning as its main focus. The algorithms to be evaluated include support vector machines, logistic regression, naive bayes, random forests, and K-nearest neighbors. In terms of accuracy and training time, the outcome test suggests that SVM produces the best results [23]. In this research proposal, a strategy for selecting features based on the word frequency distribution measure is implemented. Two benchmark datasets have been utilized with the Naive Bayes and SVM classifiers (WebKB and BBC). The experimental results demonstrate that the suggested feature selection method achieves higher classification accuracy than previous feature selection techniques [24]. This article presents an improved kNN classification algorithm. Classification and Regression Tree (CART), Support Vector Machine (SVM), and k-nearest neighbor classification (kNN). Experiments demonstrate that the enhanced approach performs best with the kNN classification method, achieving an accuracy rate of 11.50% and a precision rate of 20% [25]. This study offers a simple capsule network for text categorization by means of a two-stage training process. In these methods, the global features of the sample space are often overlooked. The experimental results reveal that the proposed technique considerably improves the classification precision of the model [26]. Using the Support Vector Machines (SVM) classification technique, he classified English text and documents in this study. According to him, the Rocchio classifier delivers the best performance when the size of the feature set is small, whereas SVM excels when the size of the feature set is large. Through experimental analysis, he established that the classification rate exceeds 90% when more than 4,000 characteristics are employed [27]. In this paper, he combine the advantages of convolutional neural networks and long short-term memory (LSTM). Then a convolution kernel of varying sizes is used to extract higher level semantic information from the text. The accuracy percentage for model testing has reached 98.03%. The outcomes indicate that this model has the best performance. [28]. In this paper, a text classification approach based on deep learning is proposed. The method is founded on the combinatorial word vectors obtained from the word2vec model's training. CNN and Attention classifiers utilize combinatorial word vectors as inputs. After a series of comparative trials, the proposed method for text categorization on a company's complaint text increased the accuracy rate significantly. The accuracy rate is superior to other algorithms, exceeding 90% [29]. In this study, he demonstrated the efficacy of long short-term memory (LSTM) models for text data based on deep learning. Vocabulary storage is essential for sentiment analysis, and LSTM is one model that considerably improves the accuracy of text classification over prior models. The results of the experiments indicate that the proposed LSTM model outperformed existing models [30]. This research introduces a multilayer text classification system based on the Virtual Category tree (VC tree). The objective of developing the classifier in a bottom-up manner was to minimize the repetition and time of sample learning. The testing results revealed that the proposed method outperformed classifications based on support vector machines (SVM) [31]. Using the Bi-LSTM-CNN approach, this study studies the application issue of NLP in text classification with the goal of enhancing text classification accuracy. The experiment demonstrates that the model presented in this research is superior for classifying news texts [32]. This project is a collection of Wongnai's datasets, which are mostly in Thai. It is hoped that these datasets will advance research in natural language processing (NLP), especially in Thai [33].

Many research efforts have primarily utilized Natural Language Processing (NLP) and ranking algorithms as core methodologies, demonstrating their effectiveness in text ranking tasks [34]. However, despite the extensive exploration of methodologies for text classification, a conspicuous dearth of studies exists that specifically address the harnessing of pre-trained language models for this particular purpose. The prevailing research landscape overwhelmingly focuses on text classification methodologies, primarily within the realm of English texts. This concentration prompts a discernible need for robust and effective approaches tailored to this domain. Consequently, there emerges a distinct gap in the literature that necessitates attention and investigation. A comprehensive exploration of leveraging pretrained language models for text classification is essential to 3ed classification techniques. Notably, the study emphasizes the utilization of single label classifiers, challenging the conventional approach. It underscores that the performance of these classifiers can be significantly enhanced by employing a relevant feature set. To substantiate these findings, the research employs Keras to conduct fundamental machine learning system benchmarks. The Naive Bayes Classifier for multinomial models, Linear Support Vector Machine, Logistic Regression, and Bag of Words (BoW) serve as the focal points of investigation. This endeavor sheds light on the potential improvements achievable in the performance of single label classifiers, offering valuable insights into the landscape of text classification, particularly in the context of Thai language attraction reviews from Wongnai.

3. Materials and Methods

In this section, the authors intricately outline the essential steps required for the seamless implementation of Thai text classification. The process unfolds through meticulously defined stages, encompassing data collection, text preparation, the formulation of text classification algorithms, and the judicious selection of model word embeddings to enhance the efficacy of text classifiers. The journey commences with a thorough exploration of data collection methodologies, emphasizing the significance of acquiring relevant and representative datasets to ensure the robustness of the classification models. Subsequently, the text preparation phase is elucidated, encompassing critical tasks such as data cleaning, stop-word removal, and tokenization, laying the foundation for the subsequent analytical steps. The construction of text classification algorithms takes center stage, with the authors delving into the intricacies of method selection and the fine-tuning of parameters. The importance of aligning these algorithms with the unique nuances of the Thai language is underscored, showcasing a nuanced understanding of linguistic complexities. A pivotal aspect of the section involves the authors' guidance on the selection of model word embeddings, a crucial determinant of a classifier's performance. The authors navigate through the landscape of available embeddings, offering insights into their strengths and applicability in the context of Thai text classification.

The culmination of this meticulous process is the performance evaluation, the ultimate litmus test for the efficacy of the implemented classification models. The authors employ rigorous evaluation metrics to objectively assess the models' accuracy and effectiveness, providing a comprehensive understanding of their real world applicability. To enhance comprehension, Fig. 1 visually encapsulates the model structure, offering a succinct yet comprehensive overview of the intricate architecture developed in the pursuit of effective Thai text classification. This section, woven with clarity and precision, serves as a valuable guide for researchers and practitioners navigating the nuanced realm of Thai text classification.

3.1 Dataset

In this section, the research methodology involves the meticulous selection of data sourced from the prominent 'Wongnai' website [35], recognized as a formidable repository for classifying online articles' text. With a substantial dataset at our disposal, comprising a total of 40,000 articles in each category, this selection stands as a robust foundation for the text classification endeavors undertaken in this study. The dataset, comprised of textual information, was systematically discovered online on November 1, 2021. To provide a visual representation of the dataset's composition, Fig. 2 elucidates the distribution of the total number of articles across various categories. This insightful visualization aids in comprehending the scale and diversity inherent in the collected data.

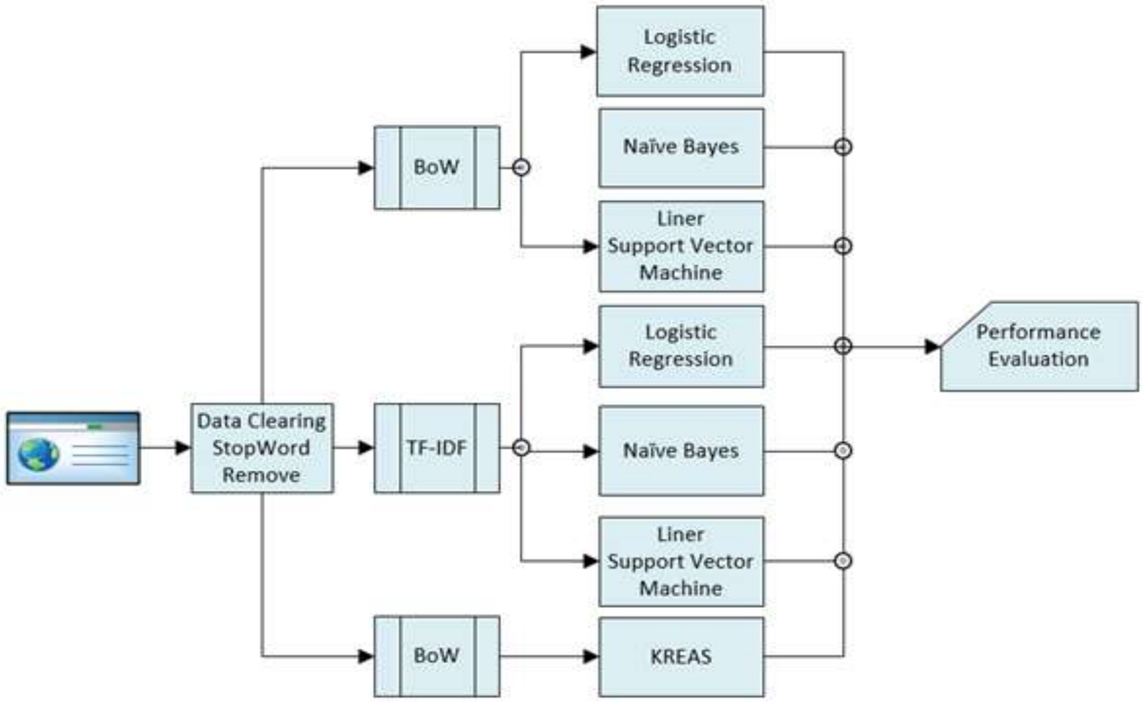


Fig. 1 Workflow of the proposed model with performance evaluation.

However, before embarking on the text classification task, the gathered dataset undergoes a rigorous procedure of data preprocessing. This critical step ensures that the dataset is refined and optimized for the subsequent analytical processes, enhancing the overall reliability and effectiveness of the text classification models developed in the study. A distinctive aspect of this research lies in its analysis of sentiment statements, specifically focusing on the positive and negative polarities within the review texts. The determination of sentiment polarity is anchored in the evaluation scores provided by the authors of the respective reviews. This nuanced approach adds depth to the text classification task, enriching the study's insights into the sentiment nuances embedded within the articles sourced from 'Wongnai.'

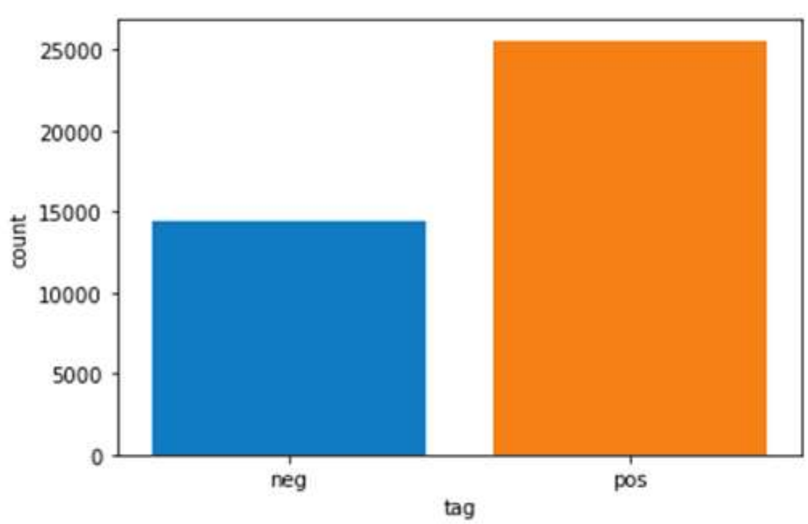


Fig. 2 Displays the total of articles dataset.

The foundational assumption guiding our approach is rooted in the belief that review ratings serve as a reflective gauge of the sentiments articulated within the content of the reviews. In this conceptual framework, negative reviews naturally garner lower ratings, while positive reviews command higher scores. To harmonize the scoring system across diverse domains, the entire score for each category is uniformly adjusted to a standard scale, with an overarching score of 5.

In pursuit of a nuanced understanding of sentiment, researchers establish a discerning criterion for composite scores. In this paradigm, a review is deemed positive if its compound score equals or exceeds 4. Conversely, if the compound score falls below (or equals) 3, the review is categorized as negative. This meticulous scoring framework ensures a comprehensive and standardized assessment of sentiment, enriching the research's ability to decipher the nuanced spectrum of opinions embedded within the diverse landscape of review content.

3.2 Pre-processing

Undoubtedly, preprocessing stands as a pivotal phase in the realm of sentiment analysis, wielding the power to significantly influence the accuracy and efficacy of text classifiers. To streamline the dataset and enhance its readiness for categorization, a meticulous process is initiated, aimed at mitigating complexity and ensuring optimal data quality. The initial step in this transformative journey involves the removal of inconsequential, or unimportant, data, strategically trimming the volume of material that bears significance for the analysis. This not only expedites processing but also acts as a catalyst in elevating the precision of subsequent text classifiers. Subsequently, the dataset undergoes a refinement process targeting words that contribute minimal value to the text processing endeavor. Words such as "มี, เฉยๆ, เช่นใด, เพียงแต่, น้อยๆ, ข้างเคียง", deemed as devoid of meaningful processing relevance, are judiciously excised. This selective elimination ensures that the dataset is purified of elements that could potentially introduce noise or ambiguity to the sentiment analysis task. With the dataset now refined, the next stride involves tokenization, a pivotal procedure that dissects the text into discrete tokens or units. Following this, a crucial step in linguistic streamlining transpires as the tokens undergo stemming. This meticulous technique, employed to reduce each word to its root form, serves to curtail unnecessary linguistic variations, streamlining the document and effectively decluttering it from superfluous details.

In essence, this preprocessing ballet not only simplifies the dataset but also lays the groundwork for a more accurate and nuanced sentiment analysis, epitomizing the beauty of meticulous data refinement in the pursuit of insightful text categorization.

3.3 Text classification

Text categorization stands as a quintessential task in the realm of Natural Language Processing (NLP), offering a diverse array of presentation types. Among these, key methodologies include the application of Naïve Bayes [9, 12, 23, 24], support vector machines (SVM) [7, 9, 12, 23, 25, 27, 31], and logistic regression [12, 23]. In this pivotal phase, each review article's text from various attractions is transformed into a feature vector, laying the groundwork for subsequent analysis. The process unfolds by deriving a vector of new features through the utilization of the provided dataset. Leveraging the TF-IDF (Term Frequency-Inverse Document Frequency) methodology, the authors selectively sample pertinent features from the dataset, strategically enhancing the dataset's representation for emotion classification in text.

As feature vectors are obtained, the intricate task of text categorization commences, employing advanced algorithms to distill insights from the transformed data. This study distinguishes itself by contrasting the efficacy of the SVM and logistic regression classification algorithms with the Multinomial Naive Bayes (MNB) algorithm, which has been a staple in previous research endeavors. This comparative analysis not only contributes to the ongoing discourse in text categorization but also offers a nuanced understanding of the strengths and limitations of different classification methodologies in the context of attraction review articles.

3.4 Implementation algorithms

Bag of Words (BoW): The Bag of Words model (BoW model) represents a streamlined abstraction of written documents, meticulously crafted by selecting sections based on specific criteria, most notably the frequency of term usage. This model finds application across diverse domains, including computer vision, natural language processing, Bayesian spam filters, document categorization, and machine learning-based information retrieval. At its core, the BoW model encapsulates a powerful methodology for distilling textual information into essential components, paving the way for a myriad of applications. In the realm of BoW, a body of text, be it a document or a sentence, is metaphorically referred to as a "bag of words." This evocative term captures the essence of the BoW process, where word lists are systematically generated. While the conventional nuances of grammar and the sequential arrangement of words are deliberately disregarded, the emphasis lies on quantifying the occurrence of words. The resultant numerical representation, devoid of syntactical intricacies, serves as a foundational element for discerning the essence and primary themes embedded within the documents.

By employing the BoW methodology, the generated word lists not only unveil the content but also encapsulate the contextual significance of the words within the given text. Despite the apparent simplicity in its treatment of language, the BoW model's numerical representation proves instrumental in deciphering the core themes and salient points of diverse documents. This methodical approach, while eschewing grammatical intricacies, serves as a testament to the beauty of simplicity in uncovering the inherent meaning and essence embedded within textual data.

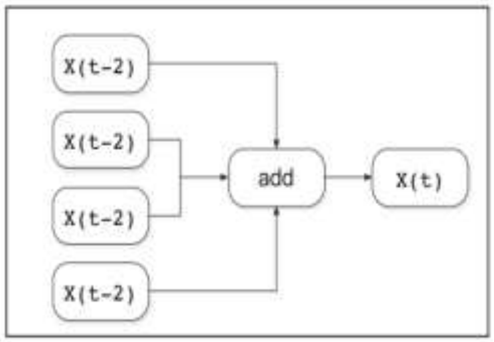


Fig. 3 The model of Bag-of-word [32].

KREAS: Keras emerges as a beacon in the realm of transfer learning frameworks, seamlessly facilitating the creation and training of models with ease. Its user-friendly interface empowers practitioners to conduct model testing with minimal complexity, requiring only essential details, the specification of training epochs, and the choice of metrics to monitor. The beauty of Keras lies in its ability to streamline the utilization of a myriad of deep learning models, significantly reducing the code footprint necessary for implementation. Notably, Keras serves as a catalyst for efficiency, allowing consultants to redirect their efforts away from intricate technological execution and towards more paramount endeavors, such as enhancing the performance of deep learning algorithms. This strategic shift in focus amplifies productivity, enabling professionals to delve deeper into refining and optimizing the intricate details of their models.

Beyond its simplicity, Keras offers a versatile toolkit for crafting intricate and sophisticated models through its API, Model, and Layer classes. These elements can be seamlessly adjusted to cater to a diverse array of needs, showcasing Keras as a dynamic and adaptable framework. In essence, Keras not only simplifies the implementation of deep learning models but also empowers practitioners to explore the depths of complexity, embodying the beauty of efficiency and flexibility in algorithmic development.

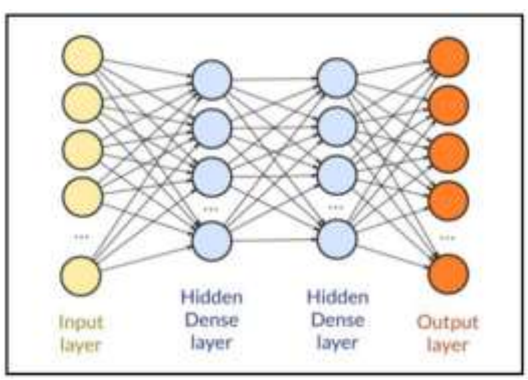


Fig. 4 Model Deep Learning in Keras [5].

Logistic Regression: Linear regression is a sort of regression function that models a dependent variable with one of two possible values using the logistic function. The outcome of a standard linear regression is a straight line, whereas the result of a logistic regression is an S-shaped curve. The name for this curve is sigmoidal. The main thing that makes this method different from linear regression is that it uses maximum likelihood instead of least squares to estimate. The likelihood is written as *E(Y|x)*, which stands for the expected value of Y if the variable x is known. The equation for logistic regression is represented as eq. (1);

$$E(Y|x) == \pi(x) = \frac{e^{\alpha+\beta X}}{1+ e^{\alpha+\beta X}} \tag{1}$$

The exponential value is used to make sure the expected value doesn't go above 1 or below 0. Using natural logarithms to transform eq. (2) will result in a new equation. This process is called "logit transformation." This step aims to clarify the content analyzed with eq. (2);

$$logit(Y) = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 X \tag{2}$$

Most of the time, logistic regression is used to predict cases in which the answer will be either "yes" or "no." For example, using a set of data of tourist attraction, Logistic Regression can predict which Users express their positive or negative. The positive is shown by the number *1*, and the negative is shown by the number *0*. This makes the condition binary [23].

Naïve Bayes: The Naive Bayes Classifier is an algorithm that uses the Naive Bayes theorem to take a probabilistic approach. Naive Bayes assign that variables *A1* through *An* in a given category *C* are conditionally independent with each other given *C*. *C* acts as the root node, and variables *A1 through An* are child nodes of it. With *A1 and An, which* are conditionally independent of each other for *C,*acquired *P(A_i-C,A_j)=P(A_i-c)*. To explain the content analyzed using the eq. (3) [27].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3}$$

Liner Support Vector Machine: One important ML algorithm that is similar to SVM is linear support vector classification. One of the most important things about this algorithm is that it lets you choose and lose functions. It is also used to fig out how many samples are used. According to studies, SVM is based on a "one against one" strategy, whereas LSVC is based on a "one against the rest" strategy. It is utilized in a variety of contexts, but its primary application is in the classification of natural languages. For example, the training dataset of n points is given by the eq. (3) to explain the content analyzed.

$$(x_1, y_1), \dots, (x_n, y_n), \tag{4}$$

Where the are either 1 or -1, each indicating the class to which the point belongs. Each is a *p*- dimensional real vector that is defined so that the distance between the hyperplane and the nearest point from either group is maximized [23].

3.5 Representing text

Before subjecting each text to the classification process, a meticulous application of the bag of words technique was undertaken, transforming every text in the dataset into a numerical vector. This method, specifically term frequency-inverse document frequency (tf-idf), operates with a strategic disregard for word order, instead focusing on the quantification of word occurrences. In the rich linguistic landscape of the Thai language, stop words played a crucial role, selectively removing less significant words and thereby elevating the precision of the subsequent analysis. Following the judicious elimination of less meaningful words, each category was enriched with a unigram, a bigram, and a trigram. A unigram encapsulates the word most intricately linked with a given category, while a bigram weaves together two such words, and a trigram artfully combines three. The corpus was represented as a feature matrix derived from unigrams, bigrams, and trigrams, producing a dimensional structure of 28,000 × 26,778.

Subsequently, each classifier underwent training, leveraging the transformative power of the corpus vector. The narrative seamlessly transitions to the next section, where the focus shifts to the utilization of Feature Extraction Techniques text. This exploration delves into the seamless integration of these techniques into the classifiers, offering a captivating insight into the intricate process of harmonizing textual data with the nuances of classification algorithms. In this symphony of data transformation, precision, and linguistic richness, the beauty lies in the elegant orchestration of intricate details.

Feature Extraction Techniques: Term weighting was used as a feature extraction strategy in this investigation. The frequency of a term is the number of occurrences in a text as well as in all papers in the corpus. The TF-IDF is a calculation that determines the term's importance in a document and corpus, allowing the relevance of a document to all other papers in the corpus to be determined. The extraction and selection of attributes aim to select a subset of words that occur only in practice and when using sets. Selecting an attribute using the information obtained as a criterion to assess the importance of the attribute. Engineering Features the Term Frequency-Inverse Document Frequency (TF-IDF) method can be used to choose attributes based on frequency in both training and testing datasets. The dataset and the weighting of infrequent terms were computed using the TF-IDF method, as defined by eq. (5). The following section provides an explanation of the analytical process based on this equation.

$$TF\text{ - }IDF = TF_{t,d} * IDF_t \tag{5}$$

Where t is a term (attribute) in a given document (example) and d is the document (example) in which t appears. Term Frequency (TF) is the ratio of the number of occurrences of a term in a document to the total number of terms in the document (n). To explain the content analyzed using the eq. (6);

$$TF - IDF = \frac{n_t}{n} * log_2 \frac{N_d}{N_t} \tag{6}$$

Inverse Document Frequency is the ratio between the total number of documents in the corpus (Nd) and the number of documents that contain the term t (Nt) [3].

Performance Evaluation: Researchers use the assessment procedure to determine our performance. In that case, accuracy is the percentage of correctly classified instances. To explain the content analyzed using the eq. (7);

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{7}$$

Recall measures how effectively the model identifies true positive cases. It was calculated using eq. (8), and the subsequent explanation describes the analysis based on this equation.

$$Recall = \frac{TP}{(TP+FN)} \tag{8}$$

Precision is the rate of positive identifications that were actually corrected. It was calculated using the formula. To explain the content analyzed using the eq. (9);

$$Precision = \frac{TP}{(TP+FP)} \tag{9}$$

Where, TP : True Positive, sentiments that are positive and actually classified as 1.
TN : True Negative, sentiments that are negative and actually classified as 0.
FP : False Positive, sentiments that are negative and classified as 1.
FN : False Negative, sentiments that are positive and classified as 0.

The F1 score is the harmonic mean of precision and recall. It is given by the formula [7]. To explain the content analyzed using the eq. (10);

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{10}$$

4. Results and Discussions

4.1 Evaluation metrics for classification

In the pursuit of assessing the efficacy of classification algorithms, the authors conducted simulations, culminating in the determination of precision, recall, and F1 values, thoughtfully compiled in Table 1. The evaluations were performed using vectors derived from both the term frequency-inverse document frequency (TF-IDF) and Bag of Words (BoW) processes. These tests aimed to gauge the accuracy of each classifier post-training, focusing on three fundamental classifiers: Naive Bayes, Linear Support Vector Machines, and Logistic Regression. Table 1 serves as a numerical testament to the performance of the tested algorithms. Notably, the observations reveal that, when utilizing TF-IDF, Logistic Regression attains superior accuracy and a higher F1-score in comparison to Naive Bayes and Support Vector Machines for the selected datasets. However, when employing Bag of Words, Naive Bayes emerges as the frontrunner, outperforming SVM and LR in both accuracy and F1 scores. This juxtaposition highlights the nuanced challenges inherent in text classification when employing text mining approaches. The findings underscore the importance of the data gathering process, recognizing it as a pivotal precursor to subsequent stages such as attribute abstraction and text mining. Through this journey, the authors gleaned insights into the significance of text preprocessing, recognizing that the nature of data molds itself through various preprocessing techniques, each yielding distinct investigation results. All models exhibit weaknesses in handling context-dependent texts, particularly those involving negation or sarcasm, with Naïve Bayes most affected due to its independence assumption, while SVM and Logistic Regression perform comparatively better. Logistic Regression achieves the strongest results with TF-IDF (F1 = 0.69), slightly outperforming SVM and Naïve Bayes, whereas Bag-of-Words yields more balanced performance across models (≈0.70). These outcomes highlight the crucial role of feature representation in shaping algorithm effectiveness. In contrast, state-of-the-art deep learning approaches such as CNNs, RNNs, and BERT typically achieve F1-scores above 0.85 by capturing contextual and semantic nuances beyond the reach of traditional frequency-based methods. The observed performance differences reflect underlying algorithmic assumptions: Naïve Bayes is constrained by independence, SVM leverages margin maximization in high-dimensional spaces, and Logistic Regression adapts effectively under TF-IDF. Overall, Logistic Regression with TF-IDF emerges as the most effective among the tested methods, though all remain clearly outperformed by modern neural models, underscoring the need to align feature representation with algorithm selection in sentiment analysis.

To enhance usability, the authors emphasize the need for organized datasets, advocating for key preprocessing steps such as tokenization, the removal of stop words, stemming, and the construction of a vector space document. This intricate dance of data preparation, as revealed through experience, accentuates the beauty of meticulous organization and thoughtful preprocessing in the pursuit of effective text mining and classification.

Table 1 Comparative analysis of the machine learning algorithm.

	Comparison Algorithm with TF-IDF				Comparison Algorithm with BoW			
	Precision	**Recall**	**F1-core**	**Accuracy**	**Precision**	**Recall**	**F1-core**	**Accuracy**
SVM	0.74	0.67	0.70	0.67	0.70	0.70	0.70	0.72
NB	0.73	0.66	0.69	0.66	0.71	0.71	0.71	0.74
LR	0.69	0.70	0.69	0.70	0.70	0.71	0.70	0.73

The overarching accuracy, reflective of the trained models, served as a holistic measure to estimate comprehensive performance across the datasets. To further illustrate this, Fig. 5 presents a comparative analysis of three algorithms Naïve-Bayes, Linear SVM, and Logistic Regression—under two feature representation methods: Term Frequency–Inverse Document Frequency (TF-IDF) and Bag of Words (BoW). The results demonstrate notable differences in performance between the two representations. Under TF-IDF, Linear SVM and Logistic Regression consistently outperformed Naive Bayes in terms of Accuracy, F1-score, Recall, and Precision, with Precision scores approaching 0.75. This suggests that TF-IDF captures more discriminative features, enabling the models to better distinguish between classes. Conversely, when using BoW, the performance gap among the three algorithms narrowed, with all models achieving relatively balanced scores across the four metrics. Notably, Naive Bayes exhibited higher accuracy under BoW compared to TF-IDF, indicating its strength in handling simpler frequency-based representations. These findings highlight two important observations. First, the choice of text representation significantly impacts classification performance, with TF-IDF generally providing richer feature representations for linear classifiers. Second, while Logistic Regression and Linear SVM remain robust across both representations, Naive

Bayes benefits more from BoW, suggesting its suitability for simpler feature spaces. Collectively, the visualization in Fig. 5 reinforces the necessity of aligning model selection with feature representation strategies to optimize Thai text classification outcomes.

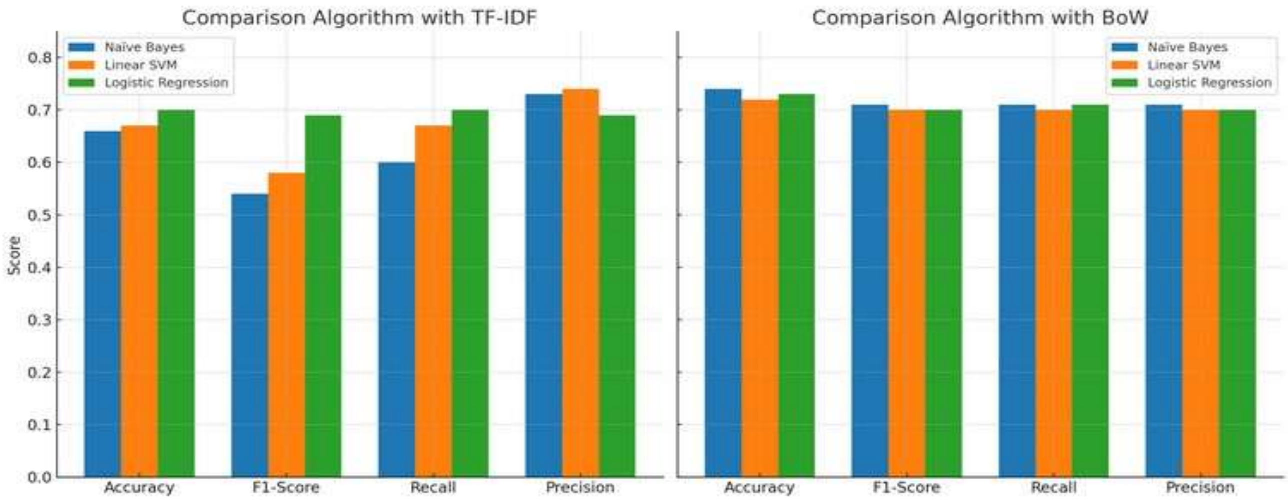
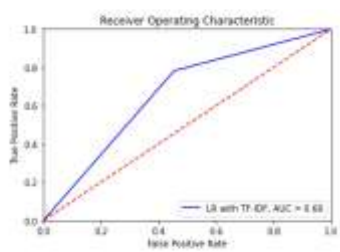
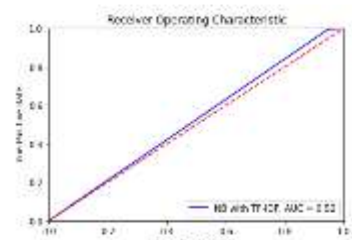


Fig. 5 Plot showing the test accuracies.

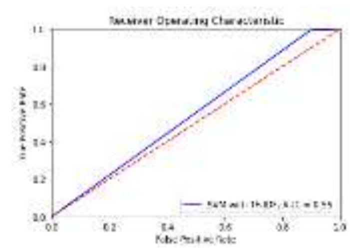
To further assess the discriminative power of the models, multiclass one-vs-all Receiver Operating Characteristic (ROC) curves were plotted (Fig. 6). The TF-IDF representation, as explained in Fig. 6 (a) – (c), shows that Logistic Regression achieved the highest Area Under the Curve (AUC = 0.66), outperforming Naive Bayes (AUC = 0.52) and SVM (AUC = 0.55). The relatively low AUC for Naive Bayes and SVM suggests weaker predictive ability, particularly in minority classes with fewer instances. Conversely, under the BoW representation (Fig. 6 (d) – (f)), all three classifiers showed improved and more comparable results. Naive Bayes achieved the highest AUC (0.71), while both Logistic Regression and SVM scored closely at 0.70. These findings indicate that BoW provides a more stable feature space across classifiers, reducing variability in prediction quality.



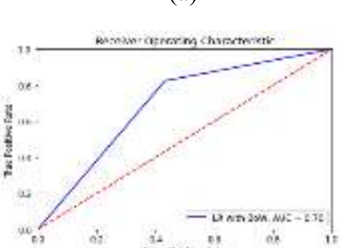
(a)



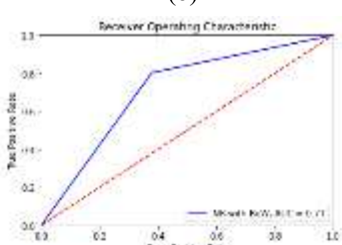
(b)



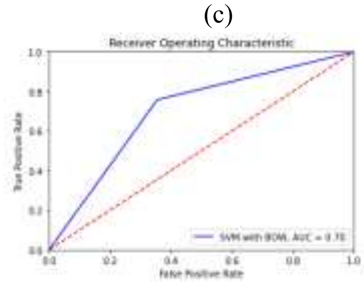
(c)



(d)



(e)



(f)

Fig. 6 ROC Curves for Comparative analysis.

The comparative analysis of Fig. 5 and Fig. 6 highlights two central insights. First, the choice of text representation substantially affects classification outcomes: TF-IDF enhances performance for linear models such as Logistic Regression and SVM, while BoW benefits Naive Bayes by leveraging simpler frequency-based features. Second, the consistency of results across evaluation metrics and ROC curves underscores the necessity of aligning algorithm selection with feature representation strategies to optimize Thai text classification.

In our empirical study, several challenges affecting the validity of text classification through text mining were identified. Data gathering emerged as a critical foundation influencing preprocessing, feature abstraction, and classification outcomes. To address methodological limitations, we refined the KERAS pipeline through iterative adjustments in preprocessing and feature representation. This process underscored the role of experiential learning, where each refinement enhanced model robustness. Ultimately, the study highlights the necessity of adaptability and continuous improvement in developing reliable text mining methodologies.

The central objective of this paper is to provide valuable insights for researchers and practitioners by determining the most accurate and resource-efficient algorithm for text analysis. While previous studies [12, 23] have yielded clear conclusions, this research distinguishes itself by adopting a comprehensive approach of comparing multiple algorithms. However, a noticeable gap exists in the pretraining phase of models. To address this gap and contribute to the existing body of knowledge, the authors introduced the KREAS process before training the three models. This addition serves as a crucial step to evaluate and enhance the performance of the target algorithm in comparison to prior experiments. The integration of the KREAS process not only enriches the study's methodology but also signifies a commitment to meticulous experimentation and improvement.

In the pursuit of methodological rigor, this research endeavors to not only identify the superior algorithm but also to shed light on the preparatory steps that can influence model performance. The beauty of this approach lies in its holistic examination, encompassing both algorithmic comparisons and the enhancement of pretraining processes through the KREAS methodology.

	loss	accuracy	val_loss	val_accuracy
0	0.562374	0.716032	0.519133	0.737143
1	0.503562	0.758532	0.521191	0.733571
2	0.449480	0.792302	0.537451	0.733214
3	0.371197	0.838294	0.557221	0.726429
4	0.269656	0.893571	0.595703	0.720000
5	0.189213	0.930040	0.661266	0.722857
6	0.136462	0.950397	0.726480	0.715357
7	0.107329	0.961865	0.789978	0.716071
8	0.087205	0.969484	0.855372	0.695357
9	0.079778	0.971587	0.883943	0.713571

Fig. 7 Training cycles in Keras**.**

Fig. 7 presents the training and validation performance of a Keras-based Bag of Words (BoW) model across ten epochs. The results demonstrate a steady improvement in training accuracy, rising from 71.60 % in the first epoch to 97.20 % by the tenth epoch, accompanied by a corresponding reduction in training loss. Validation accuracy, however, remains relatively stable between 69% and 74%, with validation loss gradually increasing after the fifth epoch. This divergence indicates potential overfitting, as the model continues to learn from the training data without achieving similar gains on validation data. The peak training accuracy of 97% highlights the model's strong capacity to fit the dataset, while the plateau in validation accuracy suggests limited generalizability. These findings emphasize the need for regularization or alternative feature representations to improve validation performance. Overall, the fig provides critical evidence of both the strengths and limitations of employing Keras with BoW in text classification tasks.

Fig. 8 illustrates the progression of training and validation loss across ten epochs using Keras. The training loss decreases consistently, reaching near-zero by the final epochs, which reflects effective model fitting on the training data. In contrast, the validation loss initially decreases slightly but begins to rise sharply after the first epoch, indicating the onset of overfitting. This divergence between training and validation curves suggests that while the model continues to improve its performance on the training set, it fails to generalize effectively to unseen data. The trend highlights a common limitation in deep learning models, where excessive optimization on the training set compromises external validity. These results illustrate the value of implementing regularization techniques, early stopping, or alternative feature representations to mitigate overfitting and improve model generalization.

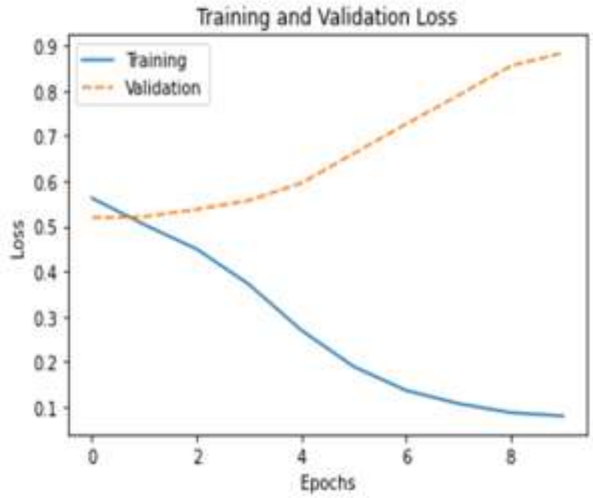


Fig. 8 Training and Validation Loss in Keras.

Despite the continued decrease in training loss, which approaches near zero by epoch 8, the validation loss exhibits a steady increase after the initial epoch. This trend reflects the model's growing tendency to overfit, as it becomes increasingly specialized in fitting the training data while failing to generalize to unseen data. The decline in training loss is an expected aspect of model optimization, demonstrating the network's capacity to learn representations of the training set with high precision. However, the divergence between the training and validation loss curves underscores the critical challenge of overfitting and highlights the importance of employing regularization techniques or early stopping criteria to maintain a balance between model learning and generalization performance.

4.2 Discussion

In the pursuit of optimal algorithm performance, it is evident that no universal approach to algorithm tuning exists. Instead, selecting algorithms that align with model specifications is crucial. This study highlights key recommendations, including the use of regularization to mitigate overfitting in models such as Logistic Regression (LR), Naive Bayes (NB), and Support Vector Machines (SVM). For classifiers employing TF-IDF, Bag of Words (BoW), and Keras, strategically augmenting or constraining weak learners can further control overfitting. Nevertheless, the study faces limitations that should be considered in future research. The optimization strategy, which involves testing every possible combination of algorithms, presents challenges for researchers with limited computational resources and time, suggesting the need to explore more efficient optimization techniques. Another constraint arises from the high dimensionality of tokenized text, which increases time and memory requirements and reduces model effectiveness in high-dimensional feature spaces. Dimensionality reduction techniques, such as principal component analysis, provide a potential solution by enabling faster computation while improving model performance. Overall, this discussion underscores the complex interplay between model selection, computational efficiency, and the challenges of high-dimensional text analysis, while emphasizing the importance of adaptive strategies to achieve balanced and effective outcomes

4. Conclusion

The primary objective of this study was to identify the most effective algorithm for text analysis in the context of Thai language documents. To this end, Logistic Regression (LR), Naïve Bayes (NB), and Support Vector Machines (SVM) were evaluated using both Term Frequency–Inverse Document Frequency (TF-IDF) and Bag of Words (BoW) representations, with Keras employed to integrate BoW into neural network training. Experimental results indicate that LR combined with TF-IDF consistently outperformed other traditional models, while NB paired with BoW achieved higher accuracy than its counterparts.

Most notably, the integration of BoW with neural networks in Keras achieved an accuracy of 97%, highlighting the ability of deep learning to surpass conventional machine learning approaches.

These findings underscore the importance of aligning algorithm selection with feature representation and task requirements. Traditional models provide reliable baselines and demonstrate effectiveness in certain conditions; however, neural network architectures are better equipped to capture semantic and contextual nuances, thereby offering superior predictive performance. From a practical perspective, this suggests that for sentiment analysis of Thai texts, employing BoW with neural networks constitutes a highly accurate and reliable framework. Future research should extend these findings by exploring dimensionality reduction techniques and optimization strategies to further enhance computational efficiency and generalizability across diverse text datasets.

References

[1] A. Yusuf, K. Allix, L. Veiber, C. Lothritz, T.F. Bissyandé, J. Klein, A. Goujon, A comparison of pre-trained language models for multi-class text classification in the financial domain, WWW '21: Companion Proceedings of the Web Conference 2021, Machinery, New York, NY, United States, 19 – 23 April 2021, 260 – 268.

[2] A.M.F. Al-Sbou, A survey of Arabic text classification models, IJ-ICT. 8(1) (2019) 25–28.

[3] M. Bounabi, E.M. Karim, S. Khalid, Text classification using fuzzy TF-IDF and machine learning models, BDIoT '19: Proceedings of the 4th International Conference on Big Data and Internet of Things, Machinery, New York, NY, United States, 23 – 24 October 2019, 1 – 6.

[4] R.B. Bhavana, P. Jeyanthi, A multilabel classifier for text classification and enhanced BERT system, Rev. Intell. Artif. 35(2) (2021) 167–176.

[5] T.C. Bahzad, B.S. Amira, A comprehensive survey of deep learning models based on Keras framework, J. Soft Comput. Data Min. 2(2) (2021) 49 – 62.

[6] J. Deng, L. Chen, Z. Wang, Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification, Comput. Speech Lang. 68 (2021) 101182.

[7] E.C. Denis, P. Irene, Performance comparison of TF-IDF and Word2Vec models for emotion text classification, Bull. Electr. Eng. Inform. 10(5) (2021) 2780 – 2788.

[8] D. Robert, S. Dmitrij, Text classification using different feature extraction approaches, 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), IEEE, Vilnius, Lithuania, 5 – 25 April 2019, 1 – 4.

[9] F. Fanny, Y. Mulyana, T. Fidelson, A comparison of text classification methods k-NN, Naïve Bayes, and support vector machine for news classification, JPIT. 3(2) (2018) 157– 160.

[10] J. Gong, Z. Teng, Q. Teng, H. Zhang, L. Du, S. Chen, M.Z.A. Bhuiyan, J. Li, M. Liu, H. Ma, Hierarchical graph transformer-based deep learning model for large-scale multi-label text classification, IEEE Access. 8 (2020) 30885 – 30896.

[11] L. Huang, D. Ma, S. Li, X. Zhang, H. Wang, Text level graph neural network for text classification, arXiv preprint arXiv:1910.02356, 2019.

[12] B.-M. Hsu, Comparison of supervised classification models on textual data, Mathematics. 8 (2020) 851.

[13] K. Jamal, R. Kurniawan, A. S. Batubara, M.Z.A. Nazri, F. Lestari, P. Papilo, Text classification on Islamic jurisprudence using machine learning techniques, J. Phys. Conf. Ser. 1566(1) (2020) 12066.

[14] M. Jiang, Y. Liang, X. Feng, X. Fan, Z. Pei, Y. Xue, R. Guan, Text classification based on deep belief network and softmax regression, Neural Comput. Appl. 29(1) (2018) 61–70.

[15] R.S. Kathuria, S. Gautam, A. Singh, S. Khatri, N. Yadav, Real-time sentiment analysis on Twitter data using deep learning (Keras), 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, Noida, India, 18 – 19 October 2019, 69–73.

[16] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, Text classification algorithms: A survey, Information. 10(4) (2019) 150.

[17] J. Kim, S. Jang, E. Park, S. Choi, Text classification using capsules, Neurocomputing. 376 (2020) 214 – 221.

[18] Y. Li, X. Wang, P. Xu, Chinese text classification model based on deep learning, Future Internet. 10(11) (2018) 113.

[19] P. Jagadeesh, B. Jasmine, T.K.R. Tarun, M.R.C. Manish, Sentiment analysis of product reviews using deep learning, 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, Bangalore, India, 19 – 22 September 2018, 2408 – 2410.

[20] I.M. Rabbimov, S.S. Kobilov, Multi-class text classification of Uzbek news articles using machine learning, J. Phys. Conf. Ser. 1546 (2020) 012097.

[21] A. Rusli, A. Suryadibrata, S. Nusantara, J. Young, A comparison of traditional machine learning approaches for supervised feedback classification in Bahasa Indonesia, IJNMT. 7(1) (2020) 28 – 32.

[22] X. She, Z. Di, Text classification based on hybrid CNN–LSTM hybrid model, 2018 11th International Symposium on Computational Intelligence and Design (ISCID), IEEE, Hangzhou, China, 08 – 09 December 2018, 185 – 189.

[23] P.A. Telnoni, R. Budiawan, M. Qana'a, Comparison of machine learning classification method on text-based case in Twitter, 2019 International Conference on ICT for Smart Society (ICISS), IEEE, Bandung, Indonesia, 19 – 20 November 2019, 1 – 5.

[24] K. Thirumoorthy, K. Muneeswaran, Feature selection for text classification using machine learning approaches, Natl. Acad. Sci. Lett. 45(1) (2022) 51 – 56.

[25] G. Wang, S.Y. Shin, An improved text classification method for sentiment classification, J. Inf. Commun. Converg. Eng. 17(1) (2019) 41–48.

[26] Y. Wu, J. Li, V. Chen, J. Chang, Z Ding, Z. Wang, Text classification using triplet capsule networks, in: International Joint Conference on Neural Networks (2020), IEEE, Glasgow, UK, 19 – 24 July 2020, 1–7.

[27] X. Lu, Efficient English text classification using selected machine learning techniques, Alex. Eng. J. 60(3) (2021) 3401 – 3409.

[28] X. Liu, H. Ni, Chinese text classification based on hybrid model of CNN and LSTM, DSIT 2020: 2020 3rd International Conference on Data Science and Information Technology, Machinery, New York, NY, United States, 26 August 2020, 129 – 134.

[29] Y. Yan, W. Li, G. Chen, W. Li, An improved text classification method based on convolutional neural networks, CCRIS '20: Proceedings of the 2020 1st International Conference on Control, Robotics and Intelligent System, Machinery, New York, NY, United States, 27 – 29 October 2020, 185 – 190.

[30] P.K. Yechuri, S. Ramadass, Classification of image and text data using deep learning-based LSTM model, Trait. Signal. 38(6) (2021) 1809 – 1817.

[31] W. Zhao, G.Y. Wang, B. Peng, Knowledge text classification based on virtual category tree, Rev. Intell. Artif. 33(1) (2019) 15 – 19.

[32] C. Liu, G. Zhang, Z. Li, News text classification based on improved Bi-LSTM-CNN, 2018 9th International Conference on Information Technology in Medicine and Education (ITME), IEEE, Hangzhou, China,19 – 21 October 2018, 890 – 893.

[33] P. Pasupa, T. Seneewong-Na-Ayudhaya, Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features, Sustain. Cities Soc. 50 (2019) 101615.

[34] P. Sroison, J.H. Chan, Resume parser with natural language processing, TechRxiv. 29 December 2021.

[35] E. Charoenphakdee, Wongnai data services, https://github.com/wongnai/wongnai-corpus/tree/master/review, Accessed 1 November 2021.

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

[37] L. Ashbaugh, Y. Zhang, A comparative study of sentiment analysis on customer reviews using machine learning and deep learning, Computers. 13(12) (2024).