


Article

An Efficient Deep Learning for Thai Sentiment Analysis

Nattawat Khamphakdee and Pusadee Seresangtakul * 

Natural Language and Speech Processing Research Group, Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen 40002, Thailand; k.nattawat@kkumail.com

* Correspondence: pusadee@kku.ac.th

Abstract: The number of reviews from customers on travel websites and platforms is quickly increasing. They provide people with the ability to write reviews about their experience with respect to service quality, location, room, and cleanliness, thereby helping others before booking hotels. Many people fail to consider hotel bookings because the numerous reviews take a long time to read, and many are in a non-native language. Thus, hotel businesses need an efficient process to analyze and categorize the polarity of reviews as positive, negative, or neutral. In particular, low-resource languages such as Thai have greater limitations in terms of resources to classify sentiment polarity. In this paper, a sentiment analysis method is proposed for Thai sentiment classification in the hotel domain. Firstly, the Word2Vec technique (the continuous bag-of-words (CBOW) and skip-gram approaches) was applied to create word embeddings of different vector dimensions. Secondly, each word embedding model was combined with deep learning (DL) models to observe the impact of each word vector dimension result. We compared the performance of nine DL models (CNN, LSTM, Bi-LSTM, GRU, Bi-GRU, CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU) with different numbers of layers to evaluate their performance in polarity classification. The dataset was classified using the FastText and BERT pre-trained models to carry out the sentiment polarity classification. Finally, our experimental results show that the WangchanBERTa model slightly improved the accuracy, producing a value of 0.9225, and the skip-gram and CNN model combination outperformed other DL models, reaching an accuracy of 0.9170. From the experiments, we found that the word vector dimensions, hyperparameter values, and the number of layers of the DL models affected the performance of sentiment classification. Our research provides guidance for setting suitable hyperparameter values to improve the accuracy of sentiment classification for the Thai language in the hotel domain.

Keywords: sentiment analysis; word embedding; Word2Vec; deep learning; natural language processing



Citation: Khamphakdee, N.; Seresangtakul, P. An Efficient Deep Learning for Thai Sentiment Analysis.

Data **2023**, *8*, 90. <https://doi.org/10.3390/data8050090>

Received: 2 April 2023

Revised: 5 May 2023

Accepted: 5 May 2023

Published: 13 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In December 2019, a novel coronavirus (COVID-19) disease was discovered, which was later declared a pandemic by the World Health Organization (WHO). The COVID-19 outbreak severely impacted global economies and the financial markets, especially the tourism industry [1]. The tourism industry was one of the main economic sectors in Thailand affected by the COVID-19 pandemic. In the pre-COVID-19 era, Thailand was one of the most popular tourist destinations for travelers from around the world. In the post-COVID-19 pandemic, the hotel industry needs to prepare, transform, and propose new services for customers.

Currently, data represents one of the most important assets of an organization. For example, Agoda.com and Booking.com are website travel agents that provide a platform for customers to share their experiences and provide feedback on the service quality, location, room, and cleanliness of hotels. The customers can write text reviews on the platform without any length limitations. Hotel companies can use customer reviews to improve

their products, business, and services. However, text reviews are in the format of unstructured data, and the amount of them is quickly increasing. This makes it difficult to analyze them manually [2], as this process requires extensive resources and time [3]. Therefore, the sentiment analysis technique was applied to process text reviews for polarity classification.

Sentiment analysis or opinion mining is one of the most important approaches in natural language processing (NLP), which refers to the task of extracting, detecting, classifying, and identifying people's opinions [4,5]. The main goal of sentiment analysis is the polarity classification of text reviews into positive, negative, or neutral [6]. Several business domains such as product, film, travel, hotel, marketing, and news industries implement sentiment analysis to obtain useful information from customer text reviews to improve their product quality or services. Machine learning algorithms and DL models are two NLP methods used for text review classification [7]. Traditional ML algorithms have been widely utilized to perform sentiment classification in various domains [8–10], obtaining greater accuracy than lexicon-based methods [11]. However, traditional ML algorithms struggle with complex text reviews and long text sequences, which can lead to less accurate results [12,13]. Recently, DL models have been applied in several NLP tasks, including sentiment analysis, machine translation, speech-to-text, and keyword extraction. In several studies [14–18], DL models were found to significantly outperform lexicon-based and traditional ML algorithms in classifying polarity. The main categories of the DL models that are widely used in sentiment analysis are convolution neural networks (CNNs) and recurrent neural networks (RNNs) [15].

Sentiment analysis is one of the most researched areas in NLP, covering a wide range of applications such as social media monitoring, product analysis, customer support insights, and employee sentiment evaluation. Numerous sentiment analysis studies using DL models in English and other European languages can be found, which can achieve great predictive accuracy; however, they use richly developed resources and tools to construct the corpus. The Thai language, on the other hand, is a low-resource language lacking the available datasets for training and testing sentiment analysis using AI systems [19]. Moreover, sentiment analysis studies based on the Thai language are comparatively very scarce. Therefore, a suitable DL model needs to be investigated for Thai sentiment analysis.

The main contributions of this paper can be summarized as follows:

1. We collected data and constructed a Thai sentiment corpus in the hotel domain;
2. We focused on and applied deep learning models to discover a suitable architecture for Thai hotel sentiment classification;
3. We applied the Word2Vec model with the CBOW and skip-gram techniques to build a word embedding model with different vector dimensions, highlighting their effect on the accuracy of sentiment classification in the Thai language. We then compared the Word2Vec, FastText, and BERT pre-trained models;
4. We also evaluated the classification accuracy of deep learning models using Word2Vec and term frequency-inverse document frequency (TF-IDF) models, comparing their performance with various traditional machine learning models.

The remainder of this paper is organized as follows: Section 2 briefly outlines the various sentiment classification techniques for different languages by applying feature extraction using ML algorithms and deep learning models. Section 3 presents the research background. In Section 4, the proposed methodology is explained. The experimental results are presented and discussed in Section 5. Lastly, we provide the conclusion and future perspectives in Section 6.

2. Related Works

Many techniques have been applied to sentiment analysis to classify text reviews as positive, negative, or neutral. Piyaphakdeesakun et al. [20] proposed an approach to sentiment classification in the Thai language using deep learning techniques. CNN and RNN models were compared to find an appropriate approach for the sentiment classification of Thai online documents. The pre-trained ULMFiT Thai language model was utilized for

text classification. The research result found that the BGRU model with an attention mechanism had the best performance. Ayutthaya et al. [21] incorporated two-feature extraction methods for accurate sentiment classification using deep learning techniques. The speech feature was utilized to identify types of words and the sentic features were utilized to identify the emotion of certain words in the reviews. The Bi-LSTM and CNN models were combined for the sentiment classification of 40 Thai children's stories. The proposed approach obtained the best results. A comparative study was also presented in [22] to gauge the performance of various deep learning techniques, including the CNN, LSTM, and Bi-LSTM models, by extracting several features. The results showed that the combination of CNN with three feature extraction methods (word embedding, POS tagging, and sentic vectors) achieved the highest accuracy. A framework for Thai sentiment analysis was also proposed in [23], which includes data pre-processing, feature extraction, and DL model construction to classify sentiment. The three datasets in the Thai language (WiseSight, ThaiEconTwitter, and TaiTales) were utilized for the evaluation of the performance of DL models. The results indicated that the combination of feature and hybrid DL models can increase the performance of classification. In their model, however, they utilized the CNN and LSTM model combination and thai2vec word embedding for Thai sentiment classification. They did not test with another DL model or other word embedding algorithms. Leelawat et al. [24] utilized ML algorithms for sentiment polarity classification of sentiment and intention classes. The dataset was collected from the Twitter social media platform with the application programming interface (API) along with Thailand tourism data. This research used the TF-IDF to represent textual documents as vectors, which required ML algorithms. The experimental result found that SVM reached the best result for sentiment analysis. The random forest algorithm achieved the best result from the intention analysis. However, deep learning models were compared for sentiment polarity classification in this research. Bowornlertsutee and Pireekreng [25] proposed the technique of building a model for the sentiment classification of online shopping reviews in the Thai language. This research compared the accuracy of polarity classification in terms of positive, neutral, and negative using the DL model (long short-term memory) and three ML models (SGD, LR, and SVM). The experimental results show that the LSTM model provided the highest accuracy. However, other word embedding approaches, such as TF-IDF or word2vec, should be applied to the model to compare the performance of sentiment classification.

Pugsee et al. [26] applied various deep learning techniques for sentiment classification in the Thai language on the TripAdvisor website. The dataset was divided into three classes: positive, negative, and neutral. The CNN and LSTM models were combined to build a classification model and measure the sentiment of text reviews. The proposed classification model achieved greater accuracy in the sentiment classification task. Vateekul et al. [27] applied deep learning techniques to classify sentiment polarity in the Thai language using a Twitter dataset. An appropriate data pre-processing approach was also proposed to deal with noisy data. Two deep learning techniques were applied to evaluate their performance in accurately classifying positive and negative polarities. The best model for sentiment classification was the DCNN model, producing a higher accuracy than the LSTM and traditional machine learning algorithms, such as NB and SVM. Thiengburanatham and Charoenkwan [28] compared traditional ML algorithms, deep learning models, and pre-trained bi-directional encoder representations for transformers (BERT) to predict toxic comments in Thai tweets. The bag-of-words (BOW) and term frequency-inverse document frequency (TF-IDF) methods were utilized to extract features and transform each word in the sentence into a number. The proposed approach showed that the extra trees algorithm and BOW combination outperformed deep learning and BERT, producing the highest accuracy. Khamphakdee and Seresangtakul [29] compared nine ML algorithms for text classification in the Thai language in the hotel domain. The different techniques used for feature extraction consisted of Delta TF-IDF, TF-IDF, N-Gram, and Word2Vec to classify sentiment polarity. The SVM algorithm with Delta TF-IDF combination archived the best classification results. Li et al. [30] applied DL models for sentiment analysis in the restaurant review

domain combined with Word2Vec, Bi-GRU, and Attention. A dataset from Dianping.com was used to test and validate the sentiment analysis model. The proposed model achieved good results, which were superior to the ML models used.

Lai et al. [31] applied ML algorithms and deep learning approaches for fake news classification. TF-IDF was combined with ML algorithms, while Word2Vec was applied with deep learning models for classification. The experiment found that the CNN and LSTM models outperformed the traditional ML algorithms. Kim and Jeong [18] proposed convolutional neural networks for sentiment analysis using three datasets consisting of Amazon customer review data, Stanford sentiment treebank data, and movie reviews for polarity classification. The proposed CNN model obtained the highest accuracy for binary and ternary classification. Xu et al. [2] proposed a sentiment analysis method using the Bi-LSTM model for the binary classification of hotel comments. The Word2Vec embedding model was used to obtain a representation of distributed words. A new representation method of word vectors was also proposed to improve the term weight computation. The proposed method was compared with many sentiment analysis methods including the RNN, CNN, LSTM, and NB models. The proposed method achieved the highest accuracy. Muhammad et al. [32] integrated the Word2Vec model and LSTM model to analyze sentiments in Indonesian hotel reviews. The LSTM model was also combined with the Word2Vec model (skip-gram and CBOW) to compare differences in sentiment classification performance. The skip-gram method was applied with a vector dimension value of 300. The LSTM model had a dropout value and learning rate of 0.2 and 0.001, respectively. The proposed approach can solve the problem of sentiment classification. Naqvi et al. [33] proposed a framework for text sentiment analysis in Urdu using deep learning approaches. This research utilized different word embedding methods combined with deep learning models to classify sentiment. The experiment showed that Bi-LSTM-ATT was the best approach for sentiment classification, obtaining the highest performance among the approaches assessed.

Fayyumi et al. [34] proposed two models: the traditional Arabic language (TAL) model and semantic partitioning Arabic language (SAP) model, to compare the polarity categorization of Jordanian opinions collected from tweets. This study utilized traditional ML algorithms (support vector machine (SVM), naïve Bayes (NB), J48, multi-layer perceptron (MPL), and logistic regression (LR)) to measure the performances of sentiment analysis in terms of positive and negative polarity. The SAP model outperformed the TAL model. Ay Karakuş et al. [35] used a movie review dataset in the Turkish language to evaluate various deep learning techniques. This research also compared the accuracy and time computation performance of sentiment classification using different deep learning techniques. The Word2Vec model with the skip-gram method was applied to build a pre-trained word embedding model from the dataset. The experimental results showed that the combination of three models (CNN, LSTM, and the pre-trained word embedding model) outperformed all other models, including CNN, Bi-LSTM, and LSTM. The one-layer CNN model and CNN-LSTM also exhibited the best performance in terms of overall running time. Rehman et al. [36] proposed a hybrid model using a combination of the CNN and LSTM models, which outperformed traditional models in sentiment analysis. The dropout technology, normalization, and a rectified linear unit were also applied to boost accuracy. The Word2Vec embedding model was utilized for the transformation of text reviews into numerical vectors. The proposed hybrid model outperformed the traditional deep learning and machine learning algorithms in terms of precision, recall, F1-score, and accuracy. Feizollah et al. [37] focused their sentiment analysis on tweets referring to two halal topics: tourism and cosmetics. The Word2Vec and Word2Seq embedding methods were applied to transform the tweets into vectors, and then each word embedding method was combined with the CNN and LSTM models to analyze the tweet sentiments. The experimental results showed that the combination of the Word2Vec embedding method with the CNN and LSTM models achieved better results. Dang et al. [38] compared different deep learning architectures to solve the sentiment analysis problem on different datasets.

Two popular word embedding models (TF-IDF and Word2Vec) were applied to transform words into vectors. Each word embedding method was combined with DNN, CNN, and RNN models to compare their accuracy in sentiment classification. The combination of Word2Vec and CNN outperformed the other models in terms of accuracy and CPU runtime, while the RNN model obtained a higher accuracy on most datasets at the cost of a longer computation time. Tashtoursh et al. [39] evaluated the performance of DL models and a hybrid model to compare polarity classification using the COVID-19 fake news dataset. The pre-trained GloVe was applied to convert text into vectors to represent words. The highest accuracy score was achieved by the CNN model.

3. Background

This section provides details of the word embedding techniques and DL models.

3.1. Word2Vec

There are several techniques to convert words into vectors to represent words. Although TF-IDF (term frequency-inverse document frequency) [40] is widely used in sentiment analysis to classify polarity along with ML algorithms and DL models, it does not consider the semantic context between words in sentences, while also generating high-dimensional sparse vectors. In 2013, Word2Vec was published by Mikolov, T. et al. [41], after which it became one of the most popular techniques for learning the vector representation of words. The Word2Vec technique can be used to create word embeddings by mapping words to numerical vectors using neural networks. A comparison of TF-IDF and Word2Vec also revealed that the Word2Vec technique was shown to achieve a higher accuracy than TF-IDF in sentiment classification [42]. The Word2Vec technique produces numerical vector representations of words through a training sentiment corpus. Researchers can define the size parameter of word embedding to produce a suitable model. There are two different architectures used in the Word2Vec technique to create word embedding representations: continuous bag-of-words (CBOW) and skip-gram. In the CBOW architecture, the context word is used as input to predict the central word. On the other hand, the skip-gram architecture uses the central word as the input to predict the context word [43]. The CBOW architecture has a better learning rate than the skip-gram architecture but at the cost of greater computation time. On the other hand, the skip-gram architecture exhibits a higher accuracy than the CBOW architecture if the dataset is small and contains many word variations [44]. To obtain the best word embedding model in this research, the CBOW and skip-gram architectures were applied to generate different vector dimensions to analyze their impact on polarity classification using DL models.

3.2. FastText Pre-Training Model

Word embedding models have become one of the important parts of natural language processing due to their increase in accuracy. In 2016, a Facebook research team proposed the word embedding process called the FastText embedding model [45]. The main application of this model is the sentiment classification task. This model is an extension of the continuous skip-gram model [41], which improves the processing speed and performance of classification. The FastText embedding model splits words into sub-words and then uses the n-gram technique to build word representations. Therefore, the FastText embedding model can build word representations as numeric vectors of words that do not appear in the corpus. The FastText embedding model is an open-source and efficient model. There are pre-trained word embeddings for 157 languages in addition to the Thai language that can be downloaded at <https://fasttext.cc/docs/en/crawl-vectors.html> (accessed on 22 April 2023). The pre-trained models were trained using the CBOW approach with hyperparameters defined as follows: a dimension of 300, length of n-grams of 5, and window sizes of 5 and 10.

3.3. BERT Pre-Training Model

In 2018, the Google AI team introduced BERT (bi-directional encoder representations from transformers) [46], which became the state-of-the-art framework for several NLP tasks, such as question answering and sentence pair classification. The two steps of BERT are pre-training and fine-tuning. BERT is a language processing pre-trained model that uses a large dataset with unlabeled data from the BooksCorpus and English Wikipedia. It can then be fine-tuned for downstream tasks with labeled data. BERT was pre-trained in two tasks. The first task is masked language modeling (MLM), in which 15% of the tokens in a sentence fed into the model are randomly masked. After that, the model predicts those hidden words at the output layer. The second task is next sentence prediction (NSP), in which the model trains a pair of sentences to understand the relationships between words in a sentence and then predicts whether the second sentence is related or not (e.g., question answering and natural language inference). There are two different BERT architectures: BERT_{base} and BERT_{large}, with 12 and 24 encoder layers, respectively. The total number of parameters of BERT_{base} is 110 M and BERT_{large} is 340 M.

Currently, there are many BERT models available, which have been presented in different domains. Some BERT models were pre-trained for multi-lingual language processing, such as multi-lingual BERT (M-BERT) [47] with 104 languages, and XML-RoBERTa [48], with 100 languages. However, those language models produce a low performance on downstream tasks for the Thai language. To address these problems, WangchanBERTa [49] was proposed by Lowphansirikul et al. in 2021. Specifically, it is a mono-lingual language model for the Thai language that contains a large dataset (78 GB) of many domains, including social media posts, news articles, and other public adverts. The WangchanBERTa language model was pre-trained based on the RoBERTa architecture and WanghanBERTa model and can be downloaded at <https://huggingface.co/airesearch/wangchanberta-base-att-spm-uncased> (accessed on 22 April 2023). for fine-tuning to different downstream tasks.

3.4. Deep Learning

Deep learning models are gaining increased popularity to solve several tasks, such as NLP, image processing, bioinformatics, and medical problems. Deep learning is a type of machine learning with multiple hidden layers in the neural network. Deep learning models achieve better accuracy and performance than machine learning algorithms because they can automatically learn and extract features from very complex patterns of large datasets [35]. Deep learning models can compute a huge amount of unstructured data to extract important information. In the NLP task, deep learning models can solve most language problems while achieving state-of-the-art results [50].

3.4.1. Convolutional Neural Network (CNN)

The CNN model is a type of deep neural network architecture that is mostly used in image processing, object detection, image segmentation, and face detection. Moreover, the CNN model can be applied for sentiment classification, achieving superior results to traditional ML algorithms. It can detect the complex features of data while reducing the execution time. There are three major layers in the CNN model, including the convolution layer, pooling layer, and fully connected layer [38,51].

The word embedding results are used as the input to the convolution layer to extract features using filters to produce a feature map as the output. Several techniques can be used to construct the word vector matrix, such as Word2Vec, FastText, and GloVe. To apply a CNN, the words N in a sentence S are transformed into an embedding vector of size s_i . Then, the sentence is represented as a matrix M .

$$M = [s_1, s_2, s_3, \dots, s_i, \dots, s_{n-1}, s_n] \quad (1)$$

To perform convolution, let x be the input data and k the number of filters in the convolutional layers. Convolution can be performed using the following [52]:

$$y_i = \sum_j f(M * w_j + b_j), j = 1, 2, \dots, k \quad (2)$$

where y_i is the matrix after the convolution operation, $*$ is the convolution operation, w_j and b_j are the weight and bias, respectively, and $f(\cdot)$ is an activation function.

The pooling layer reduces the dimensions of features by combining the outputs, thereby reducing the number of parameters for computation while retaining the most important information. Two methods are commonly used for pooling: max pooling and average pooling. The operation of average pooling can be calculated as follows:

$$z = \frac{1}{N} \sum_{(i,j) \in s} x_{ij}, i, j = 1, 2, \dots, p \quad (3)$$

where x_{ij} is the activation value at (i, j) .

Finally, the fully connected layer produces the result of sentiment classification from the output of the previous layers.

$$Y = \sum_i f(wz + b) \quad (4)$$

where Y and z are denoted the output vector and input features, respectively, and w and b represent the weight and bias of the fully connected layer, respectively.

3.4.2. Recurrent Neural Network (RNN)

The RNN model is a type of deep learning network structure designed to deal with special sequence data, such as text reviews, sensor data, and stock prices. The RNN model has gained increased popularity in the NLP task in recent years. Unlike traditional neural networks, the RNN model can process sequence data by retaining the output of previous states before feeding it as an input of the next state for a better prediction. The most used RNN models are long short-term memory (LSTM) and gated recurrent unit (GRU).

- Long short-term memory (LSTM) and bi-directional long short-term memory (Bi-LSTM).

The LSTM model is a special type of recurrent neural network (RNN) that is applied in several areas to process the long-term dependencies of input sequence data [53]. The LSTM model was proposed by Hochreiter and Schmidhuber [54] to address the vanishing gradient and exploding gradient problems of RNNs by adding a memory cell and gate units, thereby reducing the complexity in training and fine-tuning parameters. The LSTM model consists of two main state vectors: hidden state h_i and cell state C_i . In addition, the three main gates of the LSTM model are the input gate i_t , output gate o_t , and forget gate f_t [55]. Each state is calculated as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (5)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (6)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (7)$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (8)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (9)$$

$$h_t = O_t \times \tanh(C_t) \quad (10)$$

The bi-directional long short-term memory (Bi-LSTM) model improves on the disadvantages of the LSTM model, which only processes sequential information in a forward-to-backward direction. The Bi-LSTM model can instead process sequential information in both directions, thus better learning and capturing the context of the sentence. The Bi-LSTM model is often used to solve NLP tasks, exhibiting better performance than the LSTM model [6].

- Gated recurrent unit (GRU) and bi-directional gated recurrent unit (Bi-GRU).

The GRU model was proposed by Chung et al. [56] to address the issues with traditional recurrent neural networks such as the LSTM model. The GRU model has a similar structure to the LSTM model, with gating structures for processing sequence information. It contains two main gate structures: the reset gate r_t and the update gate z_t . The reset gate determines the information to forget, which allows one to control the information to preserve memory. The GRU model also solves the disadvantages of the LSTM model, such as memory consumption and slow processing time. The GRU model has been used in the task of sentiment classification, showing a better performance than the LSTM model [57–60]. Each gate and state of the GRU model is calculated as follows [55]:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (11)$$

$$r_t = \sigma(W_i \times [h_{t-1}, x_t] + b_r) \quad (12)$$

$$\tilde{h}_t = \tanh(W_c \times [r_t \times h_{t-1}, x_t] + b_c) \quad (13)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (14)$$

Although the GRU model can automatically learn and extract useful information better than the LSTM model, it also learns sequence information in the forward-to-backward direction only. Therefore, useful information can be easily lost in sentiment analysis tasks. Accordingly, the Bi-GRU model can solve this problem, constituting a forward output GRU model and reverse output GRU model to learn sequence information [61]. Several studies have demonstrated the better performance of the Bi-GRU model compared to the GRU, LSTM, and Bi-LSTM models in the sentiment classification task [62–64].

A bi-directional RNN (BRNN) can outperform a uni-directional RNN because the bi-directional model can learn the context of reviews in both the past and future. The BRNN is computed as follows [55]:

$$\vec{h} = \sigma \left(x_t U + \vec{h}_{t-1} + w + b_t \right) \quad (15)$$

$$\overleftarrow{h} = \sigma \left(x_t U + \overleftarrow{h}_{t-1} + w + b_t \right) \quad (16)$$

where \vec{h} and \overleftarrow{h} are the forward hidden state and backward hidden state, respectively, to obtain the hidden state at time t .

4. Methodology

In this research, we propose a framework for the sentiment analysis of Thai hotel reviews using the Word2Vec technique, applying DL models for polarity classification. Figure 1 depicts the proposed framework including data collection, corpus construction, building word embedding, DL model design and evaluation, and experimental results.

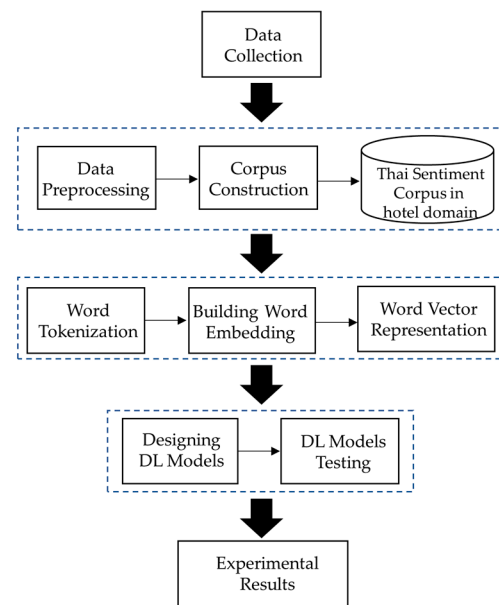


Figure 1. Framework of sentiment analysis using a combination of the Word2Vec and DL models.

4.1. Data Collection

There are limited datasets in the Thai language to study sentiment analysis tasks. Therefore, we collected and constructed a corpus of customer reviews from two popular travel websites (Agoda.com and Booking.com) used for hotel booking. A total of 25,398 unlabeled customer reviews were collected from January 2019 to March 2020. An example of the collected unlabeled dataset is illustrated in Table 1.

Table 1. Examples of unlabeled reviews.

No	Reviews
1	ห้องน่ารัก สะอาด ไม่ใหญ่มาก เดินทางไปไหนสะดวก ... พนักงานผู้ชายไม่น่ารักค่ะ ตอนเช็คอินไม่สวยดี ไม่อธิบายอะไรเลย ... แต่ตอนเช็คเอาท์พนักงานผู้หญิงน่ารักดีค่ะ Lovely room, clean, not very big, easy to travel anywhere ... Male employees are not cute. Check-in is not good. Doesn't explain anything ... But when checking out, the female staff are nice.
2	อุปกรณ์เครื่องใช้ภายในชำรุดเช่น ที่ทำน้ำอุ่นไม่ทำงาน/สายชำระไม่มี/ผ้าปูที่นอนสกปรก/ของใช้เก่ามากผ้าเช็ดตัวและชุดเครื่องนอนเก่าดำไม่สมราคา Internal equipment is damaged, such as the water heater does not work/there is no payment line/the curtains are dirty/the items are very old, the towels and bedding are old, black, not worth the price.
3	โรงแรมเก่า ผ้าปูที่นอนยับ เก้าอี้ในห้องเบาะขาดและจะหักแล้ว ถ้าพักแบบไม่คิดอะไรก็ได้ Old hotel, wrinkled sheets, the chair in the cushion room is torn and will be broken. If you can rest without thinking about anything.
4	ขนาดห้องก็กว้างมีกาแฟและที่ต้มน้ำส่วนห้องอาบน้ำนั้นน้ำไม่อุ่นพอดีไปช่วงอากาศเย็นและเวลาอาบน้ำระบายน้ำไม่ค่อยได้ดีเท่าที่ควร The size of the room is large, there is coffee and a water boiler, and the shower room is not warm enough in the cold weather and the water drainage is not as good as it should be.

4.2. Thai Sentiment Corpus Construction in the Hotel Domain

We utilized a framework [65] consisting of three main modules (data pre-processing, cosine similarity, and polarity labeling) to construct the Thai sentiment corpus in the hotel domain.

4.2.1. Data Pre-Processing

In order to construct the Thai sentiment corpus, a data pre-processing step was applied to transform the raw text reviews into an appropriate data format to build a sentiment corpus using a cosine similarity method. Unlike the English language, text review

pre-processing for the Thai language consists of many steps to obtain a useful and understandable format, because text reviews contain spelling errors and are written without spaces between the words. Moreover, the text reviews do not contain punctuation marks that identify where one sentence ends and another sentence begins. We utilized the Python@ 3.8 version and the newmm engine of the PyThaiNLP library to develop each data pre-processing step. The following data pre-processing steps were applied:

- Symbol removal: the regular expression is applied to remove a symbol, such as "<, >, () {}, =, +, @", and punctuation is also removed, such as ";, ,, ?, !, -, .";
- Number removal: Numbers do not convey the writer's feelings and they are useless for sentiment analysis. Thus, all numbers are removed from the text review;
- English word removal: English words are not considered in the text pre-processing, and they also affect the word tokenization step;
- Emoji and emoticon removal: Emojis and emoticons are a short form to convey the writer's feelings using keyboard characters. However, there are many emojis and emoticons that do not give information about the feeling of the writer, such as 🦇 (Bat), 🐻 (Bear), 🥳 (Cheer), \o/ (Cheer), @; (Rose), > < > (Fish);
- Text normalization: This process aims to improve the quality of the input text. This step transforms the mistyped word into a correct form. For example, the sentence "ห้องเก่า บริการแย พนักงานไม่สุภาพ" will be normalized as "ห้องเก่า บริการแย พนักงานไม่สุภาพ" (old room, poor service, impolite staff), which is the correct form of the Thai text. We can see that the word "บริการ" and "พนักงาน" have been transformed into the "บริการ" (service) and "พนักงาน" (employee). However, the text normalization step cannot transform the word into a complex misspelled word (i.e., "บริการแย่มาก", "ไม่สุภาพมาก");
- Word tokenization: The Thai writing system has no spaces between words. Instead, a space is utilized to identify the end of a sentence. In Thai text reviews, the expression of feelings is written in free form and contains many sentences. This makes the process difficult if the sentence contains complex words and misspellings. Thus, word tokenization is a crucial part of Thai sentiment analysis. For example, a sentence "ห้องเก่า บริการแย่มาก พนักงานไม่สุภาพ" will be tokenized into an individual word as {"ห้อง", "เก่า", " ", " ", "บริ", "การ", "แย", "มาก", " ", " ", "พนักงาน", "ไม่", "สุภาพ"}. We can see that the words "บริ", "การ", "แย", "มาก", "ไม่", and "สุภาพ" were tokenized incorrectly. Hence, the database was created to store custom words (i.e., the words "บริการ", "แย่มาก", and "ไม่สุภาพ") and to refine words in the sentences for word tokenization. Thus, the output of word tokenization is split into individual words, such as {"ห้อง", "เก่า", " ", "บริการ", "แย่มาก", " ", "พนักงาน", "ไม่สุภาพ"}. However, the words "บริการ" and "แย่มาก" are misspelled mistakes in the Thai text. They are converted into the correct form in the checking spelling errors step.
- Whitespace and tap removal: After the sentences are tokenized into individual words, there are whitespaces, blanks, and taps that are not useful for text analysis. These are removed, and the output, such as {"ห้อง", "เก่า", "บริการ", "แย่มาก", "พนักงาน", "ไม่สุภาพ"}, is produced;
- Single character removal: Single characters often appear after the word tokenization step. They have no meaning in the review;
- Converting abbreviations: "กม." and "จ." are examples of abbreviations. They are converted into "กิโลเมตร" (kilometer) and "จังหวัด" (province);
- Checking spelling errors: The text reviews contain misspelled words. These lead to incorrect tokenization. For example, the words "บริการ" (service) and "แย่มาก" (very bad) are spelled incorrectly. They are converted into "บริการ" and "แย่มาก";
- Stop-word removal: Stop-words are commonly used words in the Thai language, and they are useless for sentiment analysis. Examples of stop-words are "คือ" (is), "หรือ" (or), "มัน" (it), "ฉัน" (I), and "อื่นๆ" (other). These stop-words must be removed from reviews.

4.2.2. Cosine Similarity

To construct the Thai sentiment corpus in the hotel domain, the cosine similarity technique was applied for a similarity measurement of the sentiment training corpus and text reviews. Initially, we randomly selected 1000 reviews from the collected dataset to label as 1 (positive) or 0 (negative) and the initial sentiment training corpus was built by five experts in text sentiment analysis. The rest of the text reviews were used as testing data. Next, both the initial sentiment training corpus and text review were transformed into numerical vectors using the TF-IDF technique. Then, the TF-IDF vector of the testing data was compared to the TF-IDF vectors of the initial sentiment corpus to produce similarity scores, with a score value from 0 to 1. A score value close to 1 indicated that the testing data had a greater similarity to the initial sentiment training corpus of the positive or negative polarity. Otherwise, a score value close to zero indicated that the testing dataset was dissimilar to the initial sentiment training corpus. However, the result was also reviewed by the experts because the initial sentiment corpus was small. Lastly, the correct results of the similarity measurements were increased in the initial sentiment training corpus, whereas the incorrect results were repeated for the similarity measurement. Table 2 shows some text reviews of Thai hotels with specified polarity classes. We obtained a sentiment corpus of 22,018 reviews, which were classified into 11,086 positive and 10,932 negative reviews.

Table 2. Examples of Thai hotel reviews with positive and negative polarities.

No	Reviews	Class
1	ห้องไม่สะอาด ห้องน้ำสกปรก ผนังขึ้นรา ควรปรับปรุงนะคับ The room is not clean, the bathroom is dirty, the walls are moldy, should be improved.	0 (negative)
2	สภาพห้องเป็นห้องเก่าๆ ห้องน้ำเหม็นมาก ไม่มีแชมพู ไม่มีตู้เย็น พนักงานบริการไม่ดี The room is old. The bathroom is very smelly, no shampoo, no refrigerator, bad service from staff.	0 (negative)
3	บริการด้วยรอยยิ้ม อยู่ใจกลางเมือง ดันหลังมีผับ ใกล้ๆก็มีร้านขายของกินเพียบเลย Service with a smile, located in the center of the city. There is a pub in the back. There are many food shops nearby.	1 (positive)
4	ชอบมาก เตียงใหญ่นุ่ม สะอาด สมาย น้ำก็แรง เครื่องทำน้ำอุ่นก็ดีมาก อาบสบายสุดๆ ชอบค่ะ I like it very much; the bed is soft, clean, and comfortable; the water pressure is strong; the water heater is very good; the bath is very comfortable; I like it.	1 (positive)

4.3. Building Word Embedding

The dataset was pre-processed before being fed into the DL models. Text data were converted into numeric data for computation with the ML algorithms or DL models. This was typically conducted using a one-hot encoding method. However, this approach is unsuitable for a large number of unique words because this method generates a sparse vector matrix, where zero values increase the computation cost. Therefore, this research utilized a word embedding method to solve the above problem, i.e., the Word2Vec technique. This technique uses a neural network model to learn word embedding from a large corpus of text to produce dense word vectors as the output. The Word2Vec technique has several advantages over one-hot encoding such as a small size of word embedding, less memory use, and faster processing. We generated word embedding dimensions of different sizes to evaluate their performance in line with [66]. Table 3 depicts the hyperparameter values for generating word embedding. The CBOW and skip-gram architectures were utilized for the parameter evaluation of different vector dimensions.

Table 3. Hyperparameter for training Word2Vec.

Embedding Hyperparameters	Values
Dimensions	50, 100, 150, 200, 250, 300
Architectures	CBOW, skip-gram
Window size	2
Min_count	1
Workers	2
Sample	1×10^3

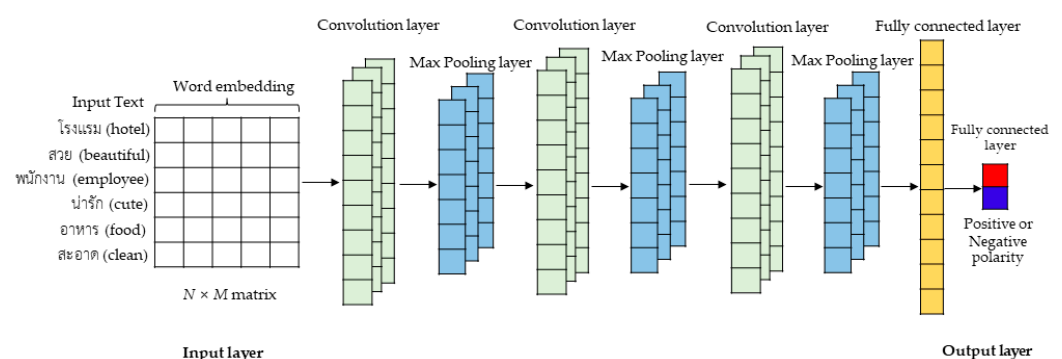
4.4. DL Model Design for Evaluation

The main aim of our research was to build a suitable DL model for Thai sentiment classification with more design options. The performance of various DL models, namely, CNN, LSTM, GRU, Bi-LSTM, Bi-GRU, CNN-LSTM, CNN-GRU, CNN-BiLSTM, and CNN-BiGRU, were compared in sentiment classification. The hyperparameter values of the DL models were determined according to [66].

To compare the performance of the CNN models, the hyperparameter values in Table 4 were introduced. We applied CNN models with 3–5 convolution layers. Figure 2 shows the CNN model with three convolution layers. Each convolution layer used the same number of units with a kernel size of 2. There were two max-pooling layers, and the activation function was set to “ReLU”. The two dense fully connected layers were used for classification based on the output of the convolution layers. In the first layer, the activation function was set to “ReLU”. We trained each CNN model at a learning rate of 0.0001, with the batch size set to 128 and the dropout rate set to 0.2. The dataset was trained for 30 epochs. The optimizer and loss functions were set to “adam” and “binary_crossentropy”. The final dense layer used the “sigmoid” activation function. The results were obtained in terms of accuracy, F1-score, recall, and precision. The process was repeated for CNN models with a different number of units.

Table 4. Hyperparameter values for CNN model configuration.

Embedding Hyperparameters	Values
Number of convolution layers	3, 4, 5
Number of units	8, 16, 32, 64, 128
Batch size	128
Learning rate	0.0001
Dropout rate	0.2
Kernel size	2
Epochs	30

**Figure 2.** Example of CNN model using three convolution layers for sentiment classification.

To evaluate the performance of the RNN and Bi-RNN models (LSTM, Bi-LSTM, GRU, and Bi-GRU), we evaluated the performance of 3–5 layers in sentiment classification. The specific hyperparameter values of the sentiment analysis models are shown in Table 5. For

example, Figures 3 and 4 depict the flow of the three-layer RNN and Bi-RNN models for sentiment classification, respectively. In the experimental step, the word embedding results were fed into the developed models (RNN and Bi-RNN). Each layer of the developed models had the same number of units, and the “return_sequences” parameter was set to true, while the dropout layer was set to 0.2. The global max pooling layer was applied to reduce the feature size according to the output of the previous layer.

Table 5. Hyperparameter values for RNN and Bi-RNN model configuration.

Embedding Hyperparameters	Values
Number of layers	3, 4, 5
Number of units	8, 16, 32, 64, 128
Batch size	128
Learning rate	0.0001
Dropout	0.2
Epochs	30

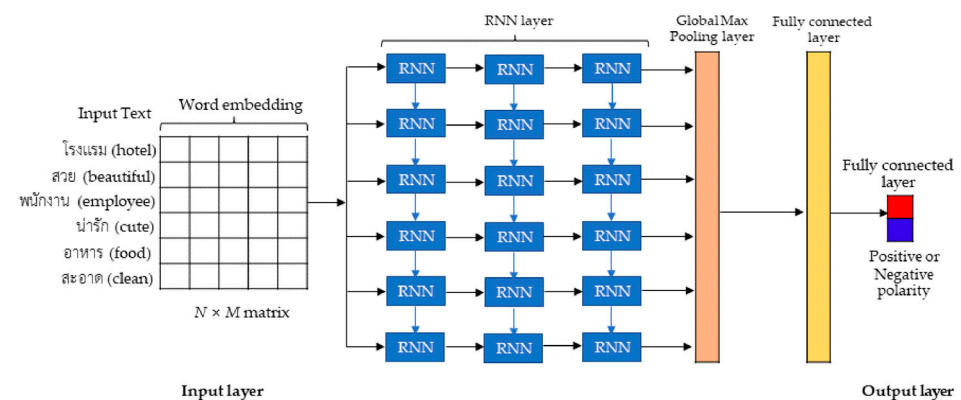


Figure 3. Example of RNN model using three layers for sentiment classification.

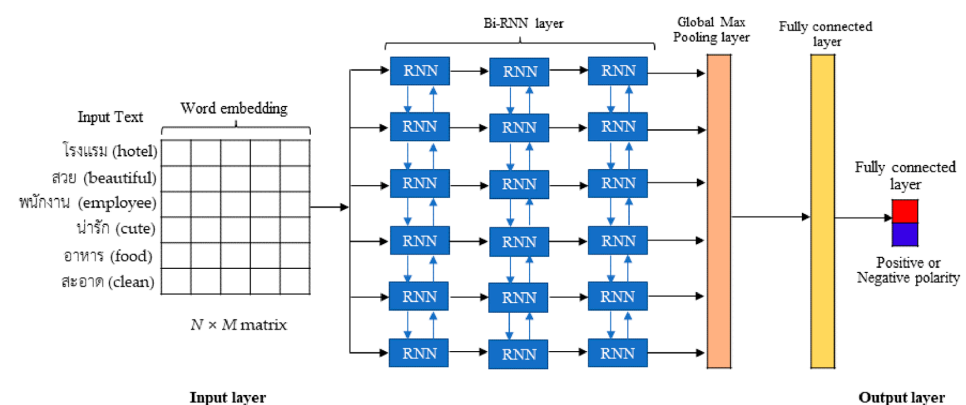


Figure 4. Example of Bi-RNN model using three layers for sentiment classification.

Lastly, the two dense fully connected layers were configured with the activation function “ReLU”, and the last dense layer was set to “sigmoid” to predict the result of positive or negative polarity. We trained each developed model with a learning rate of 0.0001 and a batch size of 128 over 30 epochs. The optimizer and loss functions were set to “adam” and “binary_crossentropy”. The final dense layer used a “sigmoid” activation function. The results were obtained in terms of accuracy, F1-score, recall, and precision. The process was repeated for the RNN and Bi-RNN models with a different number of units.

In this research, we also developed hybrid DL models by combining CNN and RNN models (e.g., CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU) to evaluate their performance in Thai sentiment analysis. Table 6 shows the hyperparameter values for the

hybrid models. Figure 5 shows the overall structure of an example of a developed hybrid model combining CNN and LSTM with five main layers. The first layer was the input layer of word embedding generated with different vector dimensions. The second layer was the CNN model with 3–5 convolution layers, each assigned with the same number of units, a kernel size of 2, and the “ReLU” activation function, followed by two max-pooling layers with a dropout rate of 0.2. The third layer was the LSTM model applied to filter information from the CNN output, with a dropout rate of 0.2. Finally, two dense fully connected layers were applied for the product output in terms of sentiment polarity using the “ReLU” and “sigmoid” activation functions. We used the “adam” optimizer and “binary_crossentropy” loss function considering their suitability for binary classification.

Table 6. Hyperparameter values for hybrid model configuration.

Embedding Hyperparameters	Values
Number of convolution layers	3, 4, 5
RNN layer	LSTM, BiLSTM, GRU, BiGRU
Number of units	8, 16, 32, 64, 128
Batch size	128
Learning rate	0.0001
Dropout	0.2
Epochs	30

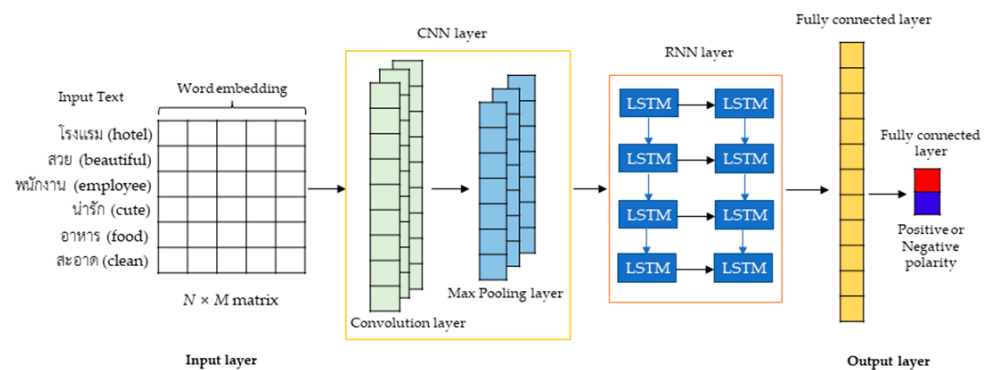


Figure 5. Example of hybrid CNN-LSTM model for sentiment classification.

5. Experimental Results

To evaluate the performance of the DL models for the text classification of Thai hotel reviews, the data collection, data pre-processing, experimental setup, and performance metrics are described below.

5.1. Experimental Setup

To perform the experiment, we used an NVIDIA GeForce RTX 3060 12 GB GPU, Intel(R) Core(TM) i9-9900 k 3.60 GHz CPU, 64 GB of RAM, and the Windows 10 Education operating system. The Keras [67] and Tensorflow [68] libraries were utilized to develop the nine DL models for sentiment classification of the Thai hotel dataset. Other libraries such as pandas [69], scikiter [70], and matplotlib [71] were also used for the investigation of the dataset and visualization of the confusion metrics. All DL models were developed using the Python3.8 programming language. In our experiments, we used 70% of the dataset for training, while the remaining data were used for testing (15%) and performance validation (15%) of the trained classifiers.

5.2. Evaluation Metrics

To evaluate the performance of each DL model in binary classification, we utilized a confusion matrix to report the results of the classification problem as true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These terms were then

used to calculate the following performance metrics: accuracy, recall, precision, and F1-score using Equations (17)–(20) [30,33,72], respectively.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (18)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (19)$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (20)$$

5.3. Results Comparison and Analysis

We performed word embedding with various vector dimensions using Word2Vec (CBOW and skip-gram) and compared their performance. All developed DL models (CNN, LSTM, Bi-LSTM, GRU, Bi-GRU, CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU) were employed to solve the sentiment analysis problem of the Thai hotel domain into binary classes. Table 7 shows the experimental results of the CBOW architecture with various vector dimensions. The results of the best DL model are reported on the basis of the number of layers with the highest accuracy. The overall experimental results revealed the highest accuracy of the CNN model with four convolution layers and 64 units (0.9146), outperforming the other DL models with 100-word embedding dimensions. In the case of 50-word embedding dimensions, the CNN-BiLSTM model with four convolution layers and 64 units achieved the highest accuracy of 0.9119. With 150- and 250-word embedding dimensions, the GRU model with four convolution layers performed best in sentiment classification, with 16 and 8 units achieving accuracies of 0.9107 and 0.9113, respectively. With 200-word embedding dimensions, the CNN-GRU model with three convolution layers and 32 units reached the highest accuracy of 0.9137, whereas the highest accuracy of 0.9113 was achieved by the CNN-BiLSTM model with three convolutional layers and four units for 300-word embedding dimensions.

We applied the same investigation for sentiment classification with Word2Vec using skip-gram. Table 8 summarizes the results in terms of accuracy, precision, recall, and F1-score. For 100-word embedding dimensions, the highest accuracy was achieved by the CNN model (0.9170) with four convolution layers and 64 units. For 50-word embedding dimensions, the CNN-LSTM model with four convolution layers and 128 units achieved better results than the other models with an accuracy of 0.9143. For 150-word embedding dimensions, the CNN-BiLSTM model with four convolution layers and 64 units achieved the best accuracy of 0.9149. For 200-word embedding dimensions, the CNN and CNN-GRU models achieved equal results in terms of accuracy (0.9146); the CNN model used three convolution layers and 32 units, while the CNN-GRU model used five convolution layers and 64 units. For 250- and 300-word embedding dimensions, the CNN model with five convolution layers achieved the best results with an accuracy of 0.9128, with 64 and 32 units, respectively.

Table 7. Performance comparison of CBOW technique with different vector dimensions.

Vector Dimensions	DL Models	Layers	Units	Matrix			
				Accuracy	Precision	Recall	F1-Score
50	CNN-LSTM	3	64	0.9098	0.9204	0.8995	0.9099
	CNN-BiLSTM	4	64	0.9119	0.9127	0.9133	0.9130
	LSTM	5	16	0.9107	0.9185	0.9037	0.9111
100	CNN-LSTM	3	32	0.9077	0.8855	0.9390	0.9115
	CNN	4	64	0.9146	0.9167	0.9145	0.9156
	LSTM	5	16	0.9128	0.9134	0.9134	0.9134
150	GRU	3	16	0.9107	0.9232	0.8983	0.9106
	GRU	4	16	0.9098	0.9124	0.9091	0.9107
	CNN-BiGRU	5	64	0.9095	0.9214	0.8977	0.9094
200	CNN-GRU	3	32	0.9137	0.9170	0.9121	0.9145
	GRU	4	16	0.9119	0.9127	0.9133	0.9130
	CNN-BiLSTM	5	32	0.9122	0.9032	0.9258	0.9144
250	CNN-LSTM	3	64	0.9104	0.8954	0.9318	0.9132
	GRU	4	8	0.9113	0.9171	0.9067	0.9119
	BiGRU	5	8	0.9101	0.9273	0.8933	0.9095
300	CNN-BiLSTM	3	32	0.9113	0.9212	0.9019	0.9115
	CNN-GRU	4	32	0.9095	0.9163	0.9037	0.9100
	GRU	5	16	0.9122	0.9098	0.9175	0.9136

Table 8. Performance comparison of skip-gram technique with different vector dimensions.

Vector Dimensions	DL Models	Layers	Units	Matrix			
				Accuracy	Precision	Recall	F1-Score
50	CNN-BiGRU	3	64	0.9116	0.9088	0.9175	0.9131
	CNN-LSTM	4	128	0.9143	0.9136	0.9175	0.9155
	CNN-BiGRU	5	128	0.9113	0.9102	0.9151	0.9126
100	CNN-LSTM	3	128	0.9140	0.9201	0.9091	0.9146
	CNN	4	64	0.9170	0.9294	0.9094	0.9170
	CNN	5	64	0.9113	0.8924	0.9378	0.9146
150	CNN-BiGRU	3	32	0.9143	0.9088	0.9234	0.9160
	CNN-BiLSTM	4	64	0.9149	0.9243	0.9061	0.9151
	CNN	5	32	0.9137	0.9160	0.9133	0.9146
200	CNN	3	32	0.9146	0.9117	0.9205	0.9161
	CNN-GRU	4	64	0.9119	0.9177	0.9073	0.9125
	CNN-GRU	5	64	0.9146	0.9098	0.9228	0.9163
250	CNN-BiGRU	3	64	0.9128	0.9134	0.9145	0.9139
	CNN	4	32	0.9125	0.9046	0.9246	0.9145
	CNN	5	64	0.9128	0.9042	0.9258	0.9149
300	CNN-BiLSTM	3	32	0.9101	0.9104	0.9121	0.9113
	CNN	4	32	0.9116	0.9197	0.9043	0.9119
	CNN	5	32	0.9128	0.9184	0.9085	0.9134

According to the above results, we can see that the skip-gram architecture and CNN model combination achieved better results than all CBOW architecture and model combinations, with an accuracy of 0.9170, a precision of 0.9294, a recall of 0.9094, and an F1-score of 0.9170 for the sentiment classification of the Thai hotel dataset.

Table 9 presents the results of different DL models combined with the Delta TF-IDF technique to classify sentiments in the hotel reviews dataset. Each DL model was defined with 3–5 layers and 8, 16, 32, 64, or 128 units to compare their performance, resulting in slightly different overall accuracies. The LSTM model with five layers and 128 units outper-

formed the other DL models with an accuracy of 0.9091. On the contrary, the combination of the CNN and LSTM models with five layers and 128 units produced the lowest result (accuracy of 0.7581). The sentiment analysis of Thai hotel reviews did not achieve effective results using a combination of hybrid models and the Delta TF-IDF method. The CNN-BiLSTM produced an accuracy of 0.8485 only, while the accuracies of the CNN-GRU and CNN-BiGRU models were only 0.7992 and 0.7793, respectively. Similarly, the accuracies of the CNN, Bi-LSTM, GRU, and Bi-GRU models were only 0.8883, 0.8874, 0.8868, and 0.8880, respectively. Thus, these models could not capture the semantic meaning of words from the text reviews, producing lower accuracies than the combination of DL models and the Word2Vec method.

Table 9. Performance comparison of Delta TF-IDF technique with different DL models.

DL Models	Layers	Units	Matrix			
			Accuracy	Precision	Recall	F1-Score
CNN	3	16	0.8871	0.8648	0.9077	0.8857
	4	32	0.8883	0.8815	0.8960	0.8887
	5	128	0.8880	0.8923	0.8870	0.8896
LSTM	3	32	0.8886	0.8839	0.8946	0.8892
	4	64	0.8831	0.9126	0.8640	0.8877
	5	128	0.9091	0.8983	0.9254	0.9116
Bi-LSTM	3	16	0.8874	0.9019	0.8787	0.8901
	4	64	0.8804	0.8839	0.8802	0.8821
	5	16	0.8843	0.8977	0.8767	0.8870
GRU	3	8	0.8856	0.8857	0.8878	0.8868
	4	8	0.8816	0.9001	0.8704	0.8850
	5	16	0.8868	0.8869	0.8890	0.8880
Bi-GRU	3	16	0.8847	0.8983	0.8768	0.8874
	4	32	0.8780	0.8863	0.8743	0.8802
	5	8	0.8880	0.8659	0.9083	0.8866
CNN-LSTM	3	16	0.7172	0.8013	0.6899	0.7414
	4	128	0.7374	0.7971	0.7141	0.7533
	5	128	0.7581	0.8306	0.7290	0.7765
CNN-BiLSTM	3	64	0.7299	0.8019	0.7049	0.7503
	4	16	0.8485	0.7606	0.6961	0.7269
	5	32	0.7944	0.8612	0.7630	0.8091
CNN-GRU	3	64	0.6897	0.7606	0.6704	0.7126
	4	64	0.7992	0.9081	0.7230	0.8050
	5	128	0.7520	0.7923	0.7372	0.7638
CNN-BiGRU	3	8	0.7060	0.7151	0.7071	0.7111
	4	128	0.7605	0.8013	0.7447	0.7720
	5	64	0.7793	0.8671	0.7408	0.7990

The performance obtained from different DL models, the Word2Vec and FastText combination, and different BERT models are shown in Table 10. We chose the best DL model from the combinations with the Word2Vec model and then applied it with FastText. The results show that the WangchanBERTa pre-trained model outperformed the other models with an optimum accuracy of 0.9225. In addition, it performed better than the DL models and Word2Vec model combination in classifying sentiment in the Thai language, in which the BERT model learned the contextual meaning of each word using a bi-directional strategy. The overall results of the DL model and FastText model provided accuracies lower than those of the DL models and Word2Vec combination because Word2Vec was trained in a specific domain. Similarly, the pre-trained M-BERT model exhibited a poor performance for sentiment classification in the Thai language as a result of the limited support for non-English languages.

Table 10. Model performance comparison for sentiment polarity classification.

ML Models	Matrix			
	Accuracy	Precision	Recall	F1-Score
CNN + FastText	0.9028	0.9132	0.8954	0.9042
LSTM + FastText	0.8925	0.8631	0.9391	0.8995
CNN-LSTM + FastText	0.9037	0.9013	0.9119	0.9066
CNN + Word2Vec (skip-gram)	0.9170	0.9294	0.9094	0.9170
CNN + Word2Vec (CBOW)	0.9146	0.9167	0.9145	0.9156
WangchanBERTa	0.9225	0.9204	0.9291	0.9247
XML-RoBERTa	0.9195	0.9201	0.9195	0.9194
M-BERT	0.7545	0.6914	0.6914	0.7969

Table 11 indicates the experimental results of the Delta TF-IDF technique combined with traditional ML models, i.e., stochastic gradient descent (SGD), logistic regression (LS), Bernoulli naïve Bayes (BNB), support vector machine (SVM), and ridge regression (RR), obtained from the scikit-learn library. Among the traditional ML models, the SVM model produced the best performance with an accuracy of 0.8966 and an F1-score of 0.8968.

Table 11. Experimental results of Delta TF-IDF technique with different ML models.

ML Models	Matrix			
	Accuracy	Precision	Recall	F1-Score
SGD	0.8962	0.8951	0.8983	0.8967
LR	0.8965	0.8900	0.9030	0.8964
BNB	0.8789	0.8704	0.8869	0.8786
SVM	0.8966	0.8921	0.9015	0.8968
RR	0.8924	0.8821	0.9019	0.8919

Statistical evaluation of the performance of each model pair using the Z-test analysis [73] for each of the classification results, including accuracy, precision, and F1-Score. This is utilized to check whether the performance of a model that obtains the highest score is significantly different from the others or not. We used a Z-test with a 95% confidence level of significance: $Z < -1.645$. Therefore, if the Z-test score is less than -1.645 for each model pair, there is a significant difference between the classification result of the model pairs. For example, as can be seen in Table 12, the WangchanBERTa model obtained a significantly higher accuracy than the CNN + FastText, LSTM + FastText, CNN-LSTM + FastText, M-BERT, and SVM models at $Z < -1.645$. Although the WangchanBERTa model achieved higher accuracy than the CNN + Word2Vec and XML-RoBERTa models, it was not significant at $Z > -1.645$.

Table 12. The Z-test results of model pairs.

Models	Z-Test			
	Accuracy	Precision	Recall	F1-Score
CNN + FastText–WangchanBERTa	−2.836	−1.059	−4.843	−2.979
LSTM + FastText–WangchanBERTa	−4.208	−7.495	1.639	−3.616
CNN-LSTM + FastText–WangchanBERTa	−2.712	−2.724	−2.584	−2.645
CNN + Word2Vec (skip-gram)–WangchanBERTa	−0.823	1.388	−2.940	−1.159
CNN + Word2Vec (CBOW)–WangchanBERTa	−1.174	−0.550	−2.211	−1.364
XML-RoBERTa–WangchanBERTa	−0.452	−0.045	−1.475	−0.803
M-BERT–WangchanBERTa	−18.552	−23.532	−24.640	−15.001
SVM–WangchanBERTa	−4.001	−4.593	−3.347	−3.976

6. Conclusions and Future Work

This research proposed various DL models for the sentiment classification of Thai reviews in the hotel domain. The Word2Vec model (CBOW and skip-gram) was utilized to build different word embedding dimensions. Delta TF-IDF was also utilized to extract features from text reviews. Nine DL models were evaluated to compare the binary sentiment classification performance (positive and negative). In this experiment, a crucial step was to tune the hyperparameter values of each DL model to verify their effect on sentiment analysis. The results revealed the superior performance of 100-word embedding dimensions using the Thai hotel reviews dataset to extract features. The CNN model with four convolution layers and 64 units achieved better results than the other models developed on the dataset. The combination of the Word2Vec method (skip-gram) and DL models achieved better results than the Delta TF-IDF + DL model and Delta TF-IDF + ML model combinations. Moreover, we also evaluated the performance of sentiment classification using a combination of the FastText pre-trained model, DL models, and the BERT pre-trained model. The WangchanBERTa pre-trained model exhibited the best performance among the models tested. However, this research only considered binary sentiment classification (positive and negative), and all of the models were evaluated on a small dataset.

In future work, we will extend the dataset for multi-class classification to verify the performance of the developed models, and we will continue to design better DL architectures and BERT models for sentiment classification in the Thai language on other tasks, such as aspect-based sentiment analysis, fake news, and so on. We believe that this study can provide researchers with a more comprehensive idea of current practices in this domain.

Author Contributions: Conceptualization, N.K. and P.S.; methodology, N.K. and P.S.; software, N.K.; investigation, N.K. and P.S.; resources, N.K.; data curation, N.K.; writing—original draft preparation, N.K. and P.S.; writing—review and editing, P.S.; supervision, P.S.; project administration, P.S.; funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a Computer and Information Science Interdisciplinary research grant from the Department of Computer Science, College of Computing, Khon Kaen University, Thailand.

Institutional Review Board Statement: Not applicable to this study.

Informed Consent Statement: Not applicable to this study.

Data Availability Statement: Data available on request from the authors.

Acknowledgments: This work was supported through a Computer and Information Science Interdisciplinary research grant from the Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen, Thailand. The authors would like to thank Wichuda Chaisiwamongkol for her suggestion on statistical analysis.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Orden-Mejía, M.; Carvache-Franco, M.; Huertas, A.; Carvache-Franco, W.; Landeta-Bejarano, N.; Carvache-Franco, O. Post-COVID-19 Tourists' Preferences, Attitudes and Travel Expectations: A Study in Guayaquil, Ecuador. *Int. J. Environ. Res. Public Health* **2022**, *19*, 4822. [\[CrossRef\]](#)
2. Xu, G.; Meng, Y.; Qiu, X.; Yu, Z.; Wu, X. Sentiment Analysis of Comment Texts Based on BiLSTM. *IEEE Access* **2019**, *7*, 51522–51532. [\[CrossRef\]](#)
3. Ombabi, A.H.; Ouarda, W.; Alimi, A.M. Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks. *Soc. Netw. Anal. Min.* **2020**, *10*, 53. [\[CrossRef\]](#)
4. Razali, N.A.M.; Malizan, N.A.; Hasbullah, N.A.; Wook, M.; Zainuddin, N.M.; Ishak, K.K.; Ramli, S.; Sukardi, S. Opinion mining for national security: Techniques, domain applications, challenges and research opportunities. *J. Big Data* **2021**, *8*, 150. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Manalu, B.U.; Tulus; Efendi, S. Deep Learning Performance in Sentiment Analysis. In Proceedings of the 4rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM), Medan, Indonesia, 3–4 September 2020; pp. 97–102.
6. Yue, W.; Li, L. Sentiment Analysis using Word2vec-CNN-BiLSTM Classification. In Proceedings of the Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), Paris, France, 14–16 December 2020; pp. 1–5.
7. Zhou, Y. A Review of Text Classification Based on Deep Learning. In Proceedings of the 3rd International Conference on Geoinformatics and Data Analysis, Marseille, France, 15–17 April 2020; ACM: Marseille, France, 2020; pp. 132–136.
8. Regina, I.A.; Sengottuvelan, P. Analysis of Sentiments in Movie Reviews using Supervised Machine Learning Technique. In Proceedings of the 4th International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 16–17 December 2021; pp. 242–246.
9. Tusar, T.H.K.; Islam, T. A Comparative Study of Sentiment Analysis Using NLP and Different Machine Learning Techniques on US Airline Twitter Data. *arXiv* **2021**, arXiv:2110.00859.
10. Mandloi, L.; Patel, R. Twitter Sentiments Analysis Using Machine Learning Methods. In Proceedings of the International Conference for Emerging Technology (INCET), Belgaum, India, 26–28 May 2020; pp. 1–5.
11. Kusriani; Mashuri, M. Sentiment Analysis in Twitter Using Lexicon Based and Polarity Multiplication. In Proceedings of the International Conference of Artificial Intelligence and Information Technology (ICAIIIT), Yogyakarta, Indonesia, 13–15 March 2019; pp. 365–368.
12. Alshammari, N.F.; AlMansour, A.A. State-of-the-art review on Twitter Sentiment Analysis. In Proceedings of the 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 1–3 May 2019; pp. 1–8.
13. Pandya, V.; Somthankar, A.; Shrivastava, S.S.; Patil, M. Twitter Sentiment Analysis using Machine Learning and Deep Learning Techniques. In Proceedings of the 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4), Bangalore, India, 16–17 December 2021; pp. 1–5.
14. Zhou, J.; Lu, Y.; Dai, H.-N.; Wang, H.; Xiao, H. Sentiment Analysis of Chinese Microblog Based on Stacked Bidirectional LSTM. *IEEE Access* **2019**, *7*, 38856–38866. [\[CrossRef\]](#)
15. Mohbey, K.K. Sentiment analysis for product rating using a deep learning approach. In Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; pp. 121–126.
16. Demirci, G.M.; Keskin, S.R.; Dogan, G. Sentiment Analysis in Turkish with Deep Learning. In Proceedings of the IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2215–2221.
17. Xiang, S. Deep Learning Framework Study for Twitter Sentiment Analysis. In Proceedings of the 2nd International Conference on Information Science and Education (ICISE-IE), Chongqing, China, 26–28 November 2021; pp. 517–520.
18. Kim, H.; Jeong, Y.-S. Sentiment Classification Using Convolutional Neural Networks. *Appl. Sci.* **2019**, *9*, 2347. [\[CrossRef\]](#)
19. Poncelas, A.; Pidchamook, W.; Liu, C.-H.; Hadley, J.; Way, A. Multiple Segmentations of Thai Sentences for Neural Machine Translation. *arXiv* **2020**, arXiv:2004.11472.
20. Piyaphakdeesakun, C.; Facundes, N.; Polvichai, J. Thai Comments Sentiment Analysis on Social Networks with Deep Learning Approach. In Proceedings of the International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Jeju Island, Republic of Korea, 23–26 June 2019; pp. 1–4.
21. Ayuthaya, T.S.N.; Pasupa, K. Thai Sentiment Analysis via Bidirectional LSTM-CNN Model with Embedding Vectors and Sentic Features. In Proceedings of the International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Pattaya, Thailand, 15–17 November 2018; pp. 1–6.
22. Pasupa, K.; Seneewong Na Ayuthaya, T. Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features. *Sustain. Cities Soc.* **2019**, *50*, 101615. [\[CrossRef\]](#)

23. Pasupa, K.; Seneewong Na Ayutthaya, T. Hybrid Deep Learning Models for Thai Sentiment Analysis. *Cogn Comput.* **2022**, *14*, 167–193. [\[CrossRef\]](#)
24. Leelawat, N.; Jariyapongpaiboon, S.; Promjun, A.; Boonyarak, S.; Saengtabtim, K.; Laosunthara, A.; Yudha, A.K.; Tang, J. Twitter Data Sentiment Analysis of Tourism in Thailand during the COVID-19 Pandemic Using Machine Learning. *Heliyon* **2022**, *8*, e10894. [\[CrossRef\]](#)
25. Bowornlertsutee, P.; Paireekreng, W. The Model of Sentiment Analysis for Classifying the Online Shopping Reviews. *J. Eng. Digit. Technol.* **2022**, *10*, 71–79.
26. Pugsee, P.; Ongsirimongkol, N. A Classification Model for Thai Statement Sentiments by Deep Learning Techniques. In Proceedings of the 2nd International Conference on Computational Intelligence and Intelligent Systems, Bangkok Thailand, 23–25 November 2019; ACM: New York, NY, USA, 2019; pp. 22–27.
27. Vateekul, P.; Koomsubha, T. A study of sentiment analysis using deep learning techniques on Thai Twitter data. In Proceedings of the 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), Khon Kaen, Thailand, 13–15 July 2016; pp. 1–6.
28. Thiengburanathum, P.; Charoenkwan, P. A Performance Comparison of Supervised Classifiers and Deep-learning Approaches for Predicting Toxicity in Thai Tweets. In Proceedings of the Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, Cha-am, Thailand, 3–6 March 2021; pp. 238–242.
29. Khamphakdee, N.; Seresangtakul, P. Sentiment Analysis for Thai Language in Hotel Domain Using Machine Learning Algorithms. *Acta Inform. Pragensia* **2021**, *10*, 155–171. [\[CrossRef\]](#)
30. Li, L.; Yang, L.; Zeng, Y. Improving Sentiment Classification of Restaurant Reviews with Attention-Based Bi-GRU Neural Network. *Symmetry* **2021**, *13*, 1517. [\[CrossRef\]](#)
31. Lai, C.-M.; Chen, M.-H.; Kristiani, E.; Verma, V.K.; Yang, C.-T. Fake News Classification Based on Content Level Features. *Appl. Sci.* **2022**, *12*, 1116. [\[CrossRef\]](#)
32. Muhammad, P.F.; Kusumaningrum, R.; Wibowo, A. Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews. *Procedia Comput. Sci.* **2021**, *179*, 728–735. [\[CrossRef\]](#)
33. Naqvi, U.; Majid, A.; Abbas, S.A. UTSA: Urdu Text Sentiment Analysis Using Deep Learning Methods. *IEEE Access* **2021**, *9*, 114085–114094. [\[CrossRef\]](#)
34. Fayyumi, E.; Idwan, S. Semantic Partitioning and Machine Learning in Sentiment Analysis. *Data* **2021**, *6*, 67. [\[CrossRef\]](#)
35. Ay Karakuş, B.; Talo, M.; Hallaç, İ.R.; Aydın, G. Evaluating deep learning models for sentiment classification. *Concurr. Comput. Pr. Exper.* **2018**, *30*, e4783. [\[CrossRef\]](#)
36. Rehman, A.U.; Malik, A.K.; Raza, B.; Ali, W. A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. *Multimed. Tools Appl.* **2019**, *78*, 26597–26613. [\[CrossRef\]](#)
37. Feizollah, A.; Ainin, S.; Anuar, N.B.; Abdullah, N.A.B.; Hazim, M. Halal Products on Twitter: Data Extraction and Sentiment Analysis Using Stack of Deep Learning Algorithms. *IEEE Access* **2019**, *7*, 83354–83362. [\[CrossRef\]](#)
38. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F. Sentiment Analysis Based on Deep Learning: A Comparative Study. *Electronics* **2020**, *9*, 483. [\[CrossRef\]](#)
39. Tashtoush, Y.; Alrababash, B.; Darwish, O.; Maabreh, M.; Alsaedi, N. A Deep Learning Framework for Detection of COVID-19 Fake News on Social Media Platforms. *Data* **2022**, *7*, 65. [\[CrossRef\]](#)
40. Mishra, R.K.; Urolagin, S.; Jothi, J.A.A. A Sentiment analysis-based hotel recommendation using TF-IDF Approach. In Proceedings of the International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, 11–12 December 2019; pp. 811–815.
41. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
42. Sohrabi, M.K.; Hemmatian, F. An efficient preprocessing method for supervised sentiment analysis by converting sentences to numerical vectors: A twitter case study. *Multimed. Tools Appl.* **2019**, *78*, 24863–24882. [\[CrossRef\]](#)
43. Onishi, T.; Shiina, H. Distributed Representation Computation Using CBOW Model and Skip-gram Model. In Proceedings of the 9th International Congress on Advanced Applied Informatics (IIAI-AAI), Kitakyushu, Japan, 1–15 September 2020; pp. 845–846.
44. Styawati, S.; Nurkholis, A.; Aldino, A.A.; Samsugi, S.; Suryati, E.; Cahyono, R.P. Sentiment Analysis on Online Transportation Reviews Using Word2Vec Text Embedding Model Feature Extraction and Support Vector Machine (SVM) Algorithm. In Proceedings of the International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), Jakarta, Indonesia, 29–30 January 2022; pp. 163–167.
45. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguistics* **2017**, *5*, 135–146. [\[CrossRef\]](#)
46. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
47. Pires, T.; Schlinger, E.; Garrette, D. How Multilingual Is Multilingual BERT? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Cedarville, OH, USA, 2019; pp. 4996–5001.

48. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-Lingual Representation Learning at Scale. *arXiv* **2019**, arXiv:1911.02116.
49. Lowphansirikul, L.; Polpanumas, C.; Jantrakulchai, N.; Nutanong, S. WangchanBERTa: Pretraining Transformer-Based Thai Language Models. *arXiv* **2021**, arXiv:2101.09635.
50. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *arXiv* **2018**, arXiv:1708.02709.
51. Tam, S.; Said, R.B.; Tanriover, O.O. A ConvBiLSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification. *IEEE Access* **2021**, *9*, 41283–41293. [[CrossRef](#)]
52. Nosratabadi, S.; Mosavi, A.; Duan, P.; Ghamisi, P.; Filip, F.; Band, S.; Reuter, U.; Gama, J.; Gandomi, A. Data Science in Economics: Comprehensive Review of Advanced Machine Learning and Deep Learning Methods. *Mathematics* **2020**, *8*, 1799. [[Cross-Ref](#)]
53. Van Houdt, G.; Mosquera, C.; Nápoles, G. A review on the long short-term memory model. *Artif. Intell. Rev.* **2020**, *53*, 5929–5955. [[CrossRef](#)]
54. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
55. Seo, S.; Kim, C.; Kim, H.; Mo, K.; Kang, P. Comparative study of Deep Learning-based Sentiment classification. *IEEE Access* **2020**, *8*, 6861–6875. [[CrossRef](#)]
56. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
57. Raza, M.R.; Hussain, W.; Merigo, J.M. Cloud Sentiment Accuracy Comparison using RNN, LSTM and GRU. In Proceedings of the Innovations in Intelligent Systems and Applications Conference (ASYU), Elazig, Turkey, 6–8 October 2021; pp. 1–5.
58. Santur, Y. Sentiment Analysis Based on Gated Recurrent Unit. In Proceedings of the International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 21–22 September 2019; pp. 1–5.
59. Dehkordi, P.E.; Asadpour, M.; Razavi, S.N. Sentiment Classification of reviews with RNNMS and GRU Architecture Approach Based on online customers rating. In Proceedings of the 28th Iranian Conference on Electrical Engineering (ICEE), Tabriz, Iran, 4–6 August 2020; pp. 1–7.
60. Shrestha, N.; Nasoz, F. Deep Learning Sentiment Analysis of Amazon.Com Reviews and Ratings. *Int. J. Soft Comput. Artif. Intell. Appl.* **2019**, *8*, 1–15. [[CrossRef](#)]
61. Gao, Z.; Li, Z.; Luo, J.; Li, X. Short Text Aspect-Based Sentiment Analysis Based on CNN + BiGRU. *Appl. Sci.* **2022**, *12*, 2707. [[CrossRef](#)]
62. Fu, Y.; Liu, Y.; Wang, Y.; Cui, Y.; Zhang, Z. Mixed Word Representation and Minimal Bi-GRU Model for Sentiment Analysis. In Proceedings of the Twelfth International Conference on Ubi-Media Computing (Ubi-Media), Bali, Indonesia, 5–8 August 2019; pp. 30–35.
63. Saeed, H.H.; Shahzad, K.; Kamiran, F. Overlapping Toxic Sentiment Classification Using Deep Neural Architectures. In Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 1361–1366.
64. Pan, Y.; Liang, M. Chinese Text Sentiment Analysis Based on BI-GRU and Self-attention. In Proceedings of the IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; pp. 1983–1988.
65. Khamphakdee, N.; Seresangtakul, P. A Framework for Constructing Thai Sentiment Corpus using the Cosine Similarity Technique. In Proceedings of the 13th International Conference on Knowledge and Smart Technology (KST-2021), Chonburi, Thailand, 21–24 January 2021.
66. Step 5: Tune Hyperparameters | Text Classification Guide | Google Developers. Available online: <https://developers.google.com/machine-learning/guides/text-classification/step-5> (accessed on 23 November 2021).
67. Keras Layers API. Available online: <https://keras.io/api/layers/> (accessed on 17 November 2021).
68. TensorFlow. Available online: <https://www.tensorflow.org/> (accessed on 17 November 2021).
69. Pandas—Python Data Analysis Library. Available online: <https://pandas.pydata.org/> (accessed on 17 November 2021).
70. Scikit-Learn: Machine Learning in Python—Scikit-Learn 1.0.2 Documentation. Available online: <https://scikit-learn.org/stable/> (accessed on 17 November 2021).
71. Matplotlib—Visualization with Python. Available online: <https://matplotlib.org/> (accessed on 17 November 2021).
72. Salur, M.U.; Aydin, I. A Novel Hybrid Deep Learning Model for Sentiment Classification. *IEEE Access* **2020**, *8*, 58080–58093. [[CrossRef](#)]
73. Isaac, E.R. *Test of Hypothesis-Concise Formula Summary*; Anna University: Tamil Nadu, India, 2015; pp. 1–5.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.