Asia Pacific Journal of Information Systems
Vol. 33 No. 1 (March 2023), 69-82

ISSN 2288-5404 (Print) / ISSN 2288-6818 (Online)
https://doi.org/10.14329/apjis.2023.33.1.69

COVID-19 Fake News Detection with Deep Learning

Rutchaneewan Kowirat[a,*], Laor Boongasame[b]

[a] *Master's Degree Student, Department of Mathematics, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand*
[b] *Lecturer, Department of Mathematics, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand*

A B S T R A C T

Social media has become one of the most popular channels to keep updated with daily news because it can quickly and easily access information. This advantage is used by malicious people to spread fake news widely. Since the COVID-19 pandemic, fake news has become a huge social problem, causing people to panic and misunderstand how to cure or protect themselves from the virus. So, the goal of this research is to use deep learning as the Recurrent Neural Network (RNN) model to find fake news about COVID-19 in the Thai language on social media and help filter information by classifying real and fake news.

Keywords: Fake News, COVID-19, Deep Learning, Recurrent Neural Network (RNN) Model, Social Media

Ⅰ. Introduction

Computer networking and the Internet are examples of rapidly growing computer technologies in an era characterized by continuous technological advancement. As a result, human data communication has become increasingly Internet-based. Moreover, social media is becoming one of the media with a considerable impact on everyday communication. Social media is a form of Internet communication that enables individuals to interact in a virtual community via the use of media platforms such as Twitter, Facebook, YouTube, and TikTok. Social media is incredibly efficient in sending and receiving data. It also enables everyone to be a communicator, sharing their expertise and telling their experiences using letters, images, and videos (Noosom, 2018). It is also a two-way communication method, allowing the receiver to engage with the sender and express their opinions. As a result, individuals choose to follow the news via social media as opposed to traditional media such as newspapers or radio, since it is convenient, quick, and makes information more accessible.

*Corresponding Author. E-mail: rutchaneewan.kowi@gmail.com

Due to the prominence of social media as a source of news. As a result, malicious individuals use this medium to propagate fake news for a variety of purposes, including assaulting business competitors, obtaining information through malware, advancing political objectives, and for entertainment (Aldwairi and Alwahedi, 2018; Kowirat and Boongasame, 2021). Due to the great efficiency of data transmission and reception, misleading news can spread quickly. As a result, fake news has become a major social issue. When the world is confronted with a pandemic, such as the coronavirus or COVID-19 (World Health Organization, n.d.), individuals actively seek out additional information and check social media for news. For instance, how to treat and prevent infection, locations at risk for infection, government legislation and even admission processes. It allows more malicious individuals to distribute misleading information.

Therefore, the purpose of this research is to detect fake news related to COVID-19 on social media with deep learning in order to help filter information and reduce the dissemination of fake news. From the study of previous research, this research will extract data directly from Twitter in the Thai language. There are a few studies that retrieved COVID-19-related data directly from Twitter and extracted Thai-language data. For instance, Kowirat and Boongasame (2021) and Patwa et al. (2021) collected data straight from Twitter in English for their studies.

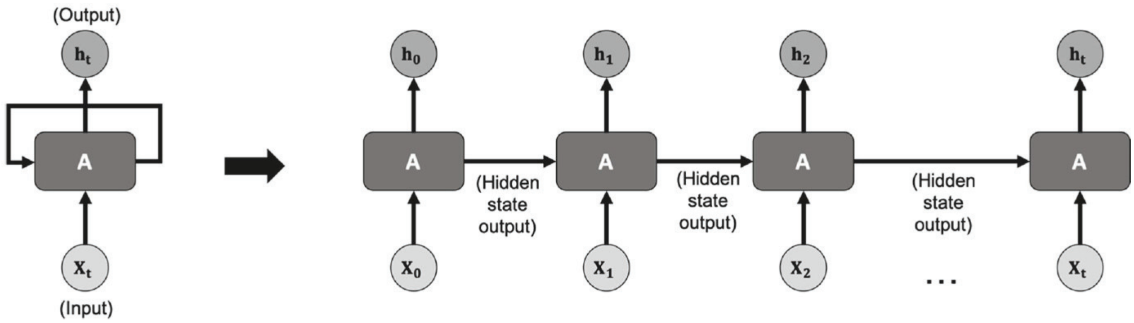
Furthermore, the Convolutional Neural Network (CNN) model and the Recurrent Neural Network (RNN) model were found to be the two most popular models from the analysis of models utilized in previous research that detect fake news (Bahad et al., 2019; Choudhary et al., 2021; Nasir et al., 2021; Samadi et al., 2021; Sastrawan et al., 2021). In order to reduce the chance that information will loss when adjusting weight, this research will focus on Recurrent Neural Network (RNN) models that are appropriate for receiving sequential data and using Bidirectional RNN Cell. However, the previous research such as the research of Sastrawan et al. (2021) and Bahad et al. (2019) were based on RNN Cell using Long Short Term Memory (LSTM) solely. The LSTM has the advantage of considering the context of words well. In the RNN, in addition to the LSTM, there is the Gated Recurrent Unit (GRU), which has the advantage of quick processing. As a result, the focus of this research will be on Recurrent Neural Network (RNN) with Bidirectional Long Short Term Memory (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU). Moreover, the models will be applied to Word2Vec and tuning the hyperparameters to find the most effective performing model.

Ⅱ. Literature Review

2.1. Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) (Olah, 2015; Promrit, 2020; Sherstinsky, 2020; Tangruamsub, 2017) is an artificial neural network that receives a dataset or a sequence of data, such as video (a sequence of images) or text, through a time series input node (a sequence of images). RNN is the ideal model for classification, word segmentation, named entity recognition (NER), and sequence tagging, as well as information that needs to be included in traditional data for analysis.

As shown in <Figure 1>, where A is the RNN cell, X(t) is input data at time t, and h(t) is output data at time t, it can be seen that when input data enters the RNN cell, it not only sends output data as a vector, but it also loops the RNN cell's hidden state output around that cycle to be the input state



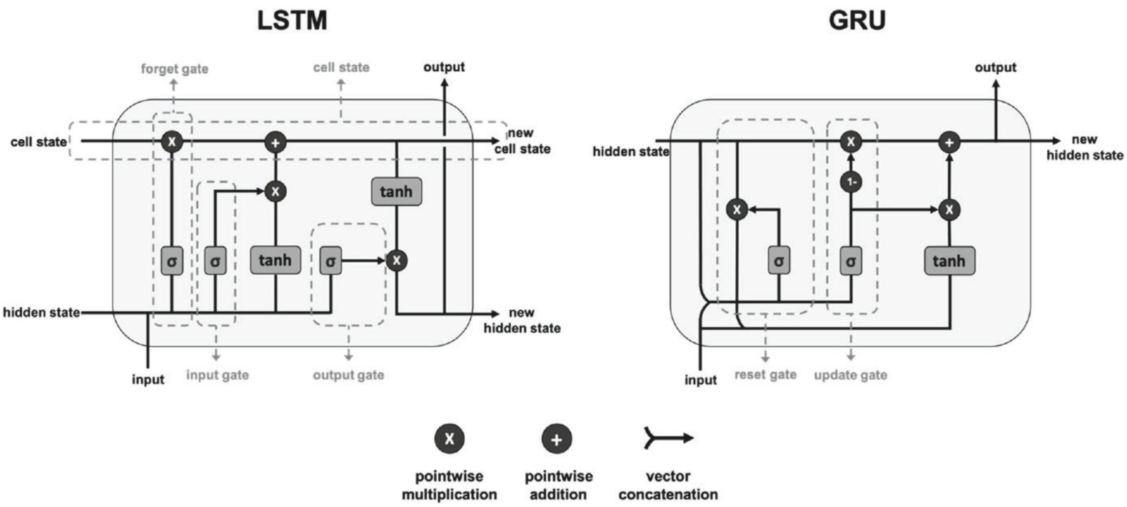
Note: Adapted from Olah (2015)

<Figure 1> RNN Operation

in the next cycle. This feature supports memorizing word context during training, such as first and last name recognition. If the preceding word is most likely a name, the following word will be categorized as a surname.

The RNN, on the other hand, has a weakness in the weight adjustment process. It can be classified into two parts. The first is the exploding gradient, which increases the weight with each adjustment. Another problem is the vanishing gradient, in which the weight decreases and approaches zero, making weight adjustments ineffective. Problems can be solved using various methods, such as the Long Short-Term Memory (LSTM), the Gated Recurrent Unit (GRU), Batch Normalization, or ReLU as an activation function.

In addition, there is a circuit within the RNN Cell for processing input data to weight in various forms. The Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU) are two popular circuits nowadays.



Note: Adapted from Olah (2015); Phi (2018)

<Figure 2> LSTM and GRU Circuits

The LSTM circuit consists of an input gate, an output gate, a forget gate, and a cell state and has two vector outputs, while the GRU circuit consists of a reset gate, an update gate, and only has one vector output, as shown in <Figure 2>.

The LSTM is suitable for applications that require sequential data, while the GRU is suitable for word-component models. Furthermore, the LSTM and GRU can perform bidirectionally (Bahad et al., 2019; Sastrawan et al., 2021). In other words, rather than depending on historical data, It could also utilize future data to assist in decision-making. This is also a solution if you lost data when reversing weight adjustments and the data you put in was very long.

2.2. Long Short-Term Memory (LSTM)

The LSTM (Hochreiter and Schmidhuber, 1997; Phi, 2018; Singkhornart, 2021) is designed to solve the RNN problem of adjusting the weights of very long data. As shown in <Figure 2>, the circuit utilizes sigmoid and tanh to multiply the values in order to adjust the weights. Because the sigmoid compresses weight values between 0 and 1, while tanh compresses weight values between -1 and 1. As a result, the problem of exploding and disappearing gradients is reduced.

Furthermore, the sigmoid not only assists in the compression of weight values but also assists the gates in deciding what data is important by weighting the data. If the weight value is very close to 1, the data is important and should be sent to be updated. If the weight value is very close to 0, the data is not important enough to store and should be removed.

The LSTM circuit is presented in <Figure 2>. The gates used to control data flow are the input gate, output gate, and forget gate. The current data and data from the previous hidden state are passed into the circuit to start the workflow. The data is entered into the forget gate to decide its importance. Important data will be passed to the input gate, which will update the data in the cell state, while unimportant data will be removed. The input data is multiplied by sigmoid and tanh before multiplying the tanh output with the sigmoid output. The sigmoid assists in determining which tanh output data should be updated in the cell state. In the final gate, the output gate multiplies the input by sigmoid and the cell state by tanh before multiplying the tanh output with the sigmoid output to identify important data and send it out as output data to be used as the next input data. As a result, the LSTM performs well with sequential data.

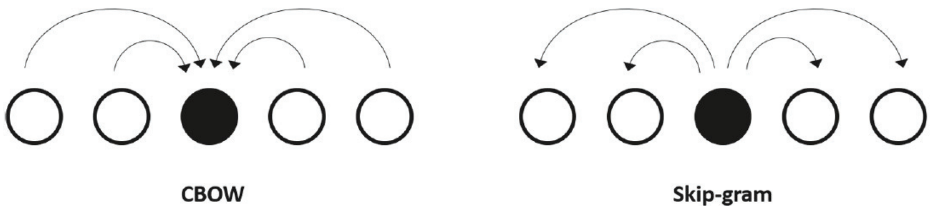
2.3. Gated Recurrent Unit (GRU)

The GRU (Chung et al., 2014; Phi, 2018; Singkhornart, 2021) is similar to the LSTM in terms of management, but the gate is different, as shown in <Figure 2>. The gate has a reset and an update gate. As a result, GRU works faster because its gates are smaller than LSTM. The workflows that begin at the reset gate will determine the importance of data being removed or stored. In the next stage, the update gate will determine what data should be sent as output data.

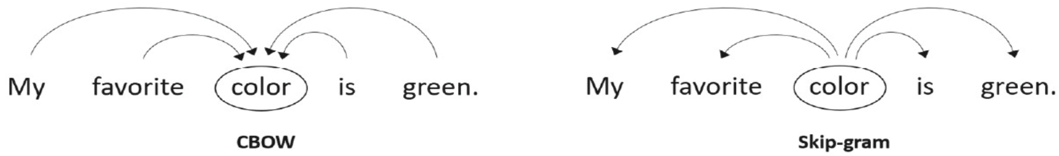
2.4. Word2Vec

Word2Vec (Mikolov et al., 2013; Sastrawan et al., 2021) is a Word Embedding model that converts words to numbers in a Word Vector format while keeping correlation to the words in the surrounding phrase. Word2Vec uses Continuous Skip-gram and Continuous Bag of Words (CBOW) to make Word Embedding (TensorFlow, 2022).

As shown in <Figure 3>, the CBOW function



<Figure 3> The Operation of CBOW and Skip-gram



<Figure 4> An Example of CBOW with Skip-gram

uses the word vector of the words surrounding the input word to predict the word vector, but the Skip-gram will define one word as the center word and then use that word to predict the surrounding words. The n-gram could be used to determine the length of a suitable context. The n-words in front of and behind the center word are taken into consideration.

For example, as shown in <Figure 4>, the CBOW section will refer to the surrounding words when predicting the word vector of the word color, while the Skip-gram would select the color as the center word first and then use it to predict the words that will appear surrounding it. An example is a 2-gram, which covers two words on the front and back of the center word.

2.5. Related Works

This research specifically looked into a variety of various research studies, which can be summarized in a preliminary table as shown in <Table 1>.

<Table 1> shows that different datasets and algorithms were used in different studies to find fake news, and model performance was also different.

Based on the preliminary analysis, the research of Kowirat and Boongasame (2021) and Patwa et al. (2021) focused on data collecting. In the research of Kowirat and Boongasame (2021) data will be collected directly from Twitter via the Twitter API. However, the text of the tweet will not be the main focus of this research because only quantitative data such as the number of followers and the number of following will be input into the model. As a result, this research cannot detect fake news by considering the context of the word. While the research of Patwa et al. (2021) focuses on collecting the text of the tweet data from various social media platforms in order to detect fake news by considering the context of the word. Additionally, the data is cleaned by removing the links, non-alphanumeric characters, and English stop words.

Additionally, it was found that the previous research of Samadi et al. (2021), Nasir et al. (2021), Choudhary et al. (2021), Sastrawan et al. (2021), and Bahad et al. (2019) focused mainly on using deep

<Table 1> Related Works

Research	Data	Model	Performance
Fake News Detection on Social Media: Case Study of 2019 Novel Coronavirus (Kowirat and Boongasame, 2021).	Extracted directly from Twitter.	Decision Tree	Accuracy = 99.92%
Fighting an Infodemic : COVID-19 Fake News Dataset (Patwa et al., 2021).	Extracted directly from Social media such as Twitter and Facebook.	1) Decision Tree 2) Logistic Regression 3) Support Vector Machine (SVM) with linear Kernel 4) Gradient Boost	SVM is the best model, getting 93.32% on the F1-score.
Deep contextualized text representation and learning for fake news detection (Samadi et al., 2021).	1) LIAR (Wang, 2017) 2) ISOT (Ahmed et al., 2017) 3) COVID-19 (Patwa et al., 2021)	1) Single-Layer Perceptron (SLP) 2) Multi-Layer Perceptron (MLP) 3) Convolutional Neural Network (CNN)	The performance is quite versatile.
BerConvoNet: A deep learning framework for fake news classification (Nasir et al., 2021).	1) George McIntire Dataset 2) Kaggle 3) Gossipcop from FakeNewsNet Repository. 4) Politifact from FakeNewsNet Repository.	1) Random Embedding with CNN 2) Static & Dynamic GloVe with CNN 3) Random Embedding with LSTM 4) ELMo with Neural Networks 5) BERT with CNN	The performance is quite versatile.
Fake news detection: A hybrid CNN-RNN based deep learning approach (Choudhary et al., 2021).	Many data set such as LIAR (Wang, 2017) and ISOT (Ahmed et al., 2017).	1) Convolutional Neural Network (CNN) 2) Recurrent Neural Network (RNN) 3) Hybrid CNN-RNN	The performance is quite versatile.
Detection of fake news using deep learning CNN-RNN based methods (Sastrawan et al., 2021).	1) ISOT (Ahmed et al., 2017) 2) Fake news dataset (UTK Machine Learning Club, 2018) 3) Fake or real news dataset (McIntire, 2017) 4) Fake news detection dataset (Jruvika, 2017)	1) Convolutional Neural Network (CNN) 2) Bidirectional LSTM 3) Residual Network (ResNet)	The performance is quite versatile.
Fake News Detection using Bi-directional LSTM-Recurrent Neural Network (Bahad et al., 2019)	1) Fake news detection dataset (Jruvika, 2017) 2) real_or_fake dataset (Ralucachitic, 2018)	1) Convolutional Neural Network (CNN) 2) Vanilla RNN 3) Unidirectional LSTM-RNN 4) Bidirectional LSTM-RNN	The performance is quite versatile.

learning to detect fake news. As shown in <Table 1>, CNN and RNN models were the most popular deep learning models used in previous research. However, some previous research may also use additional models, such as SLP, MLP (Samadi et al., 2021), ResNet (Sastrawan et al., 2021), or neural networks (Nasir et al., 2021). Models based on CNN and RNN models, on the other hand, outperform other models.

<Table 2> Attributes of Dataset

Attributes	Definition
Tweet	Text of the tweet.
Label	Identify the data as real or fake news.

Word embedding was used in a number of studies with models like Word2Vec, GloVe, BERT, and others to turn words into numbers in word vector format.

Additionally, the RNN model-based previous research prefers using only RNN cells as LSTM. The ability to recognize word context is an advantage of the LSTM. But some past research (Bahad et al., 2019; Sastrawan et al., 2021) used RNN Cell as Bidirectional LSTM to solve the problem of data loss when adjusting the weight.

Ⅲ. Method

3.1. Data Pre-processing

This research extracted data related to COVID-19 in the Thai language directly from Twitter using a python library like Tweepy and others. The concept of collecting real and fake news data is based on the research of Patwa et al. (2021) and the research of Kowirat and Boongasame (2021). The real news data is obtained from verified sources like the accounts of news agencies in Thailand, whereas the fake news is obtained from the Thai keyword “fake news COVID.” Moreover, the fake news will be filtered from only the accounts of news agencies in Thailand in order to ensure the accuracy of the data. The dataset has 1,018 rows and consists of 2 attributes, namely tweet and label, as shown in <Table 2>. In addition, some examples of real and fake news from the dataset are shown in <Table 3>.

The next step will randomize data to have the same size based on the number of lower label sizes to decrease the risk of data bias when inputting data into the model. The dataset for this research is divided into 600 rows of real news and 418 rows of fake news. As a result, two datasets were generated at random as fake news and real news, with 418 rows of each label.

When all of the labels are the same size to help reduce text size and complexity, the text will be cleaned up by removing non-alphanumeric characters or text that is unlikely to affect model performance, such as URLs and account names. However, full stops do exist in the text as they are only used for abbreviations in Thai, which differs from English in that full stops are also used to terminate sentences. The word segmentation will be more accurate if full stops are included in the text rather than removed. Due to the fact that separated letters used in abbreviations will provide noisy data, For example, the abbreviation “postscript (p.s.)” will include p and s with periods after each letter. In the word segmentation, the p and s will be separated if the dot is removed, as

<Table 3> Examples of Real and Fake News from the Dataset

Label	Text
Fake	กินอาหารที่เป็นด่างสามารถฆ่าเชื้อ COVID-19 ได้
Real	“Long COVID” เป็นอาการที่ผู้ติดเชื้อโควิดมักจะเป็นต่อหลังสิ้นสุดการติดเชื้อ สามารถพบได้มากกว่า 200 อาการ อาการที่พบส่วนใหญ่คือ อ่อนเพลีย หายใจไม่อิ่ม ภาวะสมองเสื่อม

<Table 4> The Layers of the Model

Layer	Definition
1) Input Layer	The layer of input data.
2) Embedding Layer	This layer use pre-trained word embeddings.
3) Bidirectional Layer	In the case of GRU uses Activation as a ReLU whereas LSTM uses a parameter called merge_mode with the value concat to define how vectors are concatenated.
4) Dense Layer	Dimensions are defined as 128 and activation is defined as ReLUs.
5) Dropout Layer	This layer removes some nodes at random to reduce the data's complexity.
6) Dense Layer	Dimensions are defined as 128 and activation is defined as ReLUs.
7) Dropout Layer	Remove some Nodes at random again.
8) BatchNormalization Layer	The data is normalized in this layer.
9) Dense Layer	Set Activation as Softmax to get a probability value as output data.

opposed to keeping the original form.

Text adjustments have been performed to fix typos and modify transliteration into the original language in order to improve word segmentation performance. In addition, the data in the fake label is extracted from Twitter with the keyword "fake news COVID." This keyword of news agencies is a fake news alert that makes the data real. As a result, the word "fake news" in the fake label as well as any words identifying the content as a fake news alert, such as "don't share" will be filtered out. Only genuine fake news messages will be left in the data at the conclusion.

After word segmentation using the PyThaiNLP library, a Keras Tokenizer is created to generate a Bag of words that represents the total number of words. The next steps are to determine the maximum length of the words in the sentence for padding, convert each word in the sentence to a number, and add a zero to make all the sentences the same length. In addition, the data in the label section is encrypted One Hot in the form of a number.

3.2. Model

The RNN model will be utilized in this research to detect fake news. However, the RNN model has a problem of data loss when adjusting the weight. As a result, this research will use a bidirectional RNN cell as a BiLSTM and a BiGRU to solve that problem. The two models are integrated with Word2Vec to compare and measure the performance of the appropriate models. Furthermore, the data is separated into train and test groups before being entered into the RNN model.

The model must be defined in sequential mode and use an Adam optimizer. The layers of the model are shown in <Table 4>.

As shown in <Table 4>, Bidirectional Layer is used to avoid the problem of data loss during weight adjustments. ReLU is also used in the BatchNormalization Layer and the Activation Function to fix problems with gradients that explode or go away.

In addition, in the case of an application with Word2Vec, the Word2Vec model's parameters are set with min_count = 1, size = 128, workers = 6, sg = 1 (Skip-gram), and iter = 1000. Then, create an embedding matrix and deploy it in the Embedding Layer.

For the training model, the hyperparameters of the initial model will be shown in <Table 5>. The

<Table 5> The Hyperparameters of Initial Model

Hyperparameters	Value
Loss Function	Cross-Entropy
Optimizer	Adam
Learning Rate	0.001
Dropout	0.10
Epochs	10
Batch Size	32

next step is hyperparameter tuning to compare and analyze model performance in order to obtain the best model.

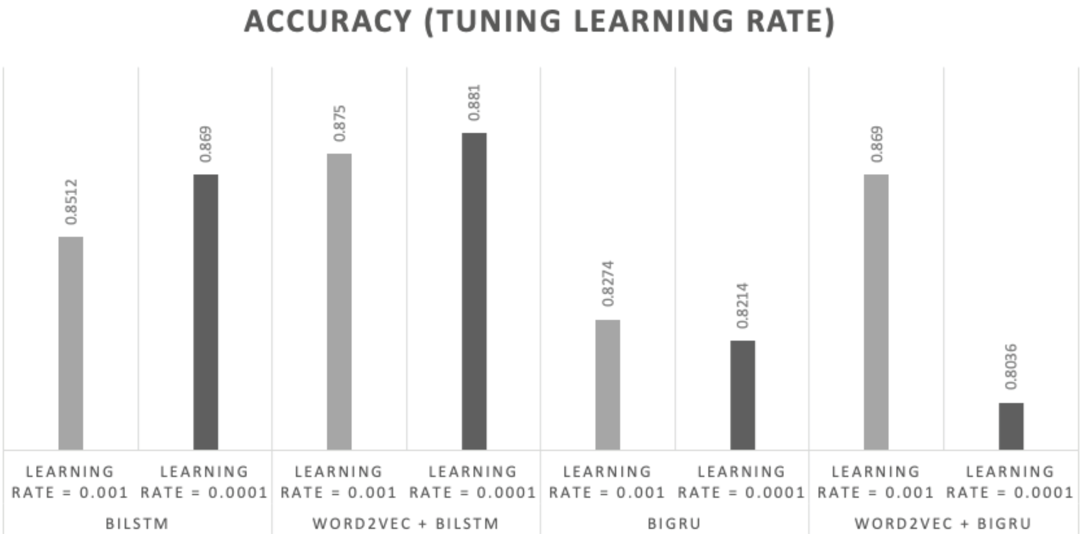
3.3. Result

This section discusses the research results after individual hyperparameter tuning, starting with learning rate, dropout, epochs, and batch size in that order, as follows:

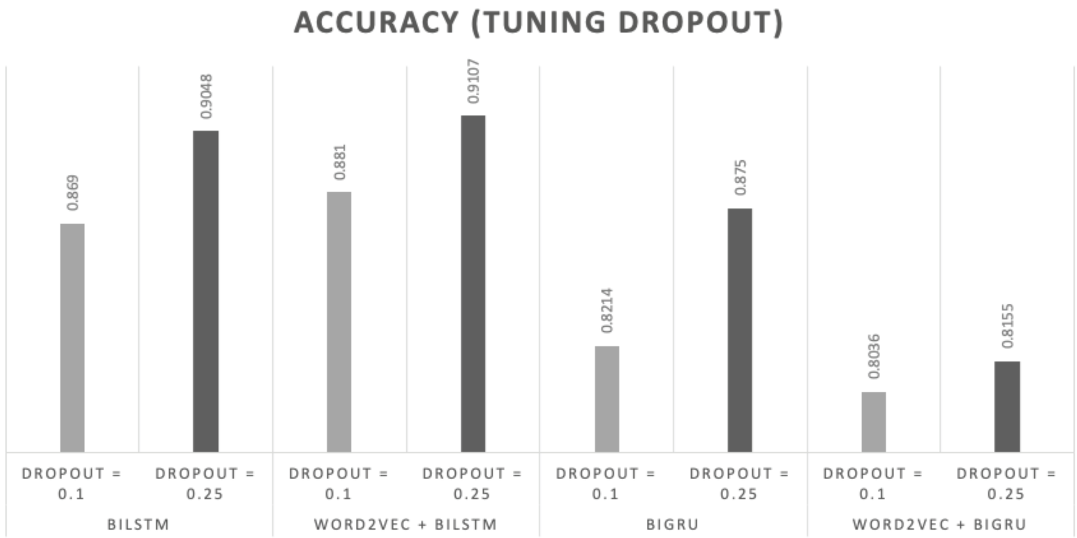
3.3.1. Tuning Learning Rate

The learning rate was adjusted from 0.001 to 0.0001 while the other hyperparameters remained constant in order to find the best learning rate to utilize in the models because the learning rate determines the model's weight. The model's weight will change less if the learning rate is too low, which will slow down the convergence of the answer and increase the training process. Conversely, if the learning rate is too high, it may fail to converge to the answer. Furthermore, The performance of the models is shown in <Figure 5>.

From <Figure 5>, we see that the models on the BiLSTM and BiGRU are very different. Based on the model with the highest accuracy, the Word2Vec + BiLSTM model has an accuracy of 88.10% when a learning rate of 0.0001 is applied. As a result, this learning rate is chosen for the next hyperparameter tuning because it is considered suitable for the dataset.



<Figure 5> Model Performance While Tuning Learning Rate

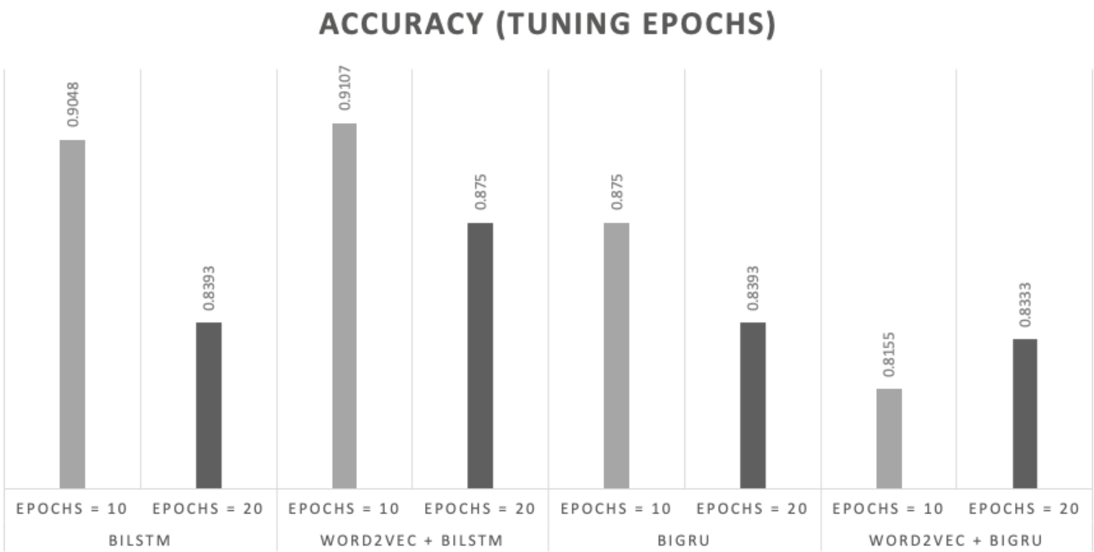


<Figure 6> Model Performance While Tuning Dropout

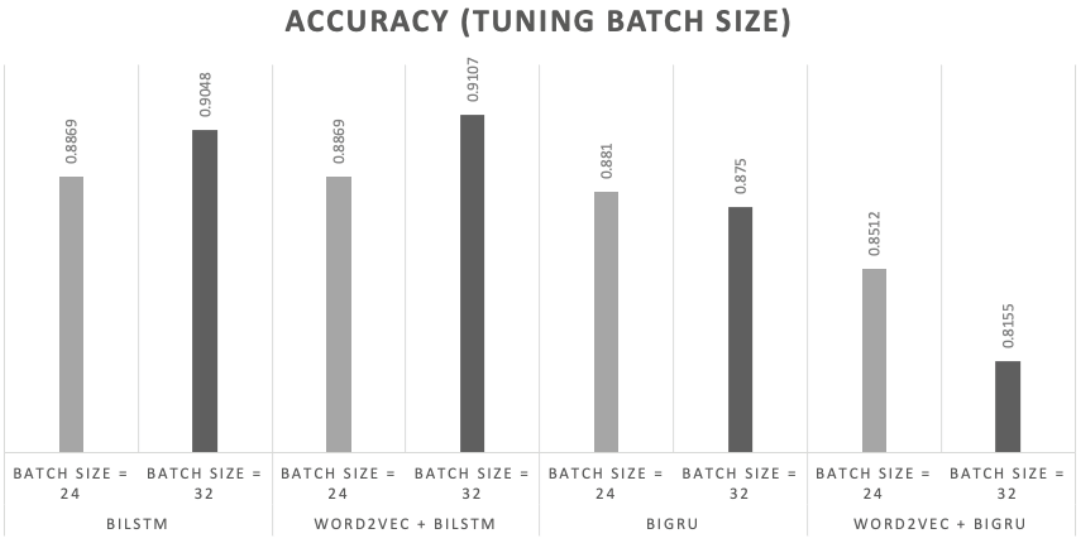
3.3.2. Tuning Dropout

The next step is tuning the dropout from 0.10 to 0.25 to determine how simplifying the model by increasing the random dropout from 10% to 25% will affect the model. The performance of the models is shown in <Figure 6>.

From <Figure 6>, the accuracy of the whole model increased when a node closure was added. It is possible that the dataset is still complex. As a result, dropout at 0.25 is considered to be the best value for the dataset and should be used for the next hyper-



<Figure 7> Model Performance While Tuning Epochs



<Figure 8> Model Performance While Tuning Batch Size

parameter tuning.

3.3.3. Tuning Epochs

The next step will be to increase the number of model training cycles by tuning epochs from 10 to 20 in order to determine how many cycles are sufficient for the model's training. The performance of the models is shown in <Figure 7>.

As shown in <Figure 7>, almost all models performed better when the epochs were 10. As a result, epochs at 10 are considered to be the best value for the dataset and should be used for the next hyperparameter tuning.

3.3.4. Tuning Batch Size

The final step is tuning the batch size from 32 to 24 in order to find out how many samples should be processed before the model is updated. The performance of the models is shown in <Figure 8>.

From <Figure 8>, we see that the models on the BiLSTM and BiGRU are very different. When the batch size is set to 32, the Word2Vec + BiLSTM model has the highest accuracy. As a result, this batch size is considered suitable for the dataset.

After finishing the hyperparameter tuning, it was determined that learning rate at 0.0001, dropout at 0.25, epochs at 10, and batch size at 32 are the best hyperparameters for the dataset. The Word2Vec + BiLSTM model is the best model because it has the highest accuracy at 91.07%.

<Figure 8> demonstrates that BiLSTM outperforms BiGRU, perhaps because BiLSTM continues to perform effectively within Twitter's character limit (Developer Platform, n.d.). According to the comparison data, Word2Vec + BiLSTM performed slightly better than BiLSTM. This is likely due to the fact that the majority of datasets obtained from news providers consist primarily of tweets with brief titles and links to additional news content. This reduces the total number of words in a sentence.

Ⅳ. Conclusion

This research is able to detect and classify fake news in the Thai language on social media by considering the context of words in order to help filter information and reduce the dissemination of fake news. However, since the dataset was collected in Thai, cleaning the data and word segmentation will be challenging. It is significantly challenging to separate compound words in Thai since words are frequently typed next to one another without gaps like in English. As a result, there may still be some noise in the dataset, and the performance model may be impacted. Furthermore, the dataset of real and fake news only comes from verified sources, such as the accounts of news agencies in Thailand. The dataset was still not collected from normal users. Due to the fact that anyone can publish content on social media, this is considered another research limitation.

In the future, the researcher will train additional word sets to improve word segmentation, add data from normal users, find popular keywords used in real and fake news, as well as search for an algorithm that can be used to determine the optimal values for hyperparameter tuning in order to identify the model that is most suitable for the dataset.

Acknowledgments

This work is supported by King Mongkut's Institute of Technology Ladkrabang [KREF016310]

<References>

[1] Ahmed, H., Traore, L., and Saad, S. (2017). *Detection of online fake news using n-gram analysis and machine learning techniques*. International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments (ISDDC 2017), 127-138.

[2] Aldwairi, M., and Alwahedi, A. (2018). *Detecting Fake News in Social Media Networks*. The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018), 215-220.

[3] Bahad, P., Saxena, P., and Kamal, R. (2019). *Fake News Detection using Bi-directional LSTM-Recurrent Neural Network* International Conference on Recent Trends in Advanced Computing 2019 (ICRTAC 2019), *165*, 74-82.

[4] Choudhary, M., Chouhan, S. S., Pilli, E. S. and Vipparthi, S. K. (2021). BerConvoNet: A deep learning framework for fake news classification. *Applied Soft Computing*, *110*, 107614.

[5] Chung, J., Gulcehre, C., Cho, K. H., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS 2014 Workshop on Deep Learning*. https://doi.org/10.48550/arXiv.1412.3555

[6] Developer Platform. (n.d.). *Counting characters when composing Tweets*. [Online]. Retrieved from https://developer.twitter.com/en/docs/counting-characters

[7] Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[8] Jruvika. (2017). *Fake news detection*. Retrieved from https://www.kaggle.com/jruvika/fake-news-detection/version/1

[9] Kowirat, R., and Boongasame, L. (2021). *Fake News Detection on Social Media: Case Study of 2019 Novel Coronavirus*. 2021 3rd International Conference on E-Business and E-commerce Engineering (EBEE), 298-302.

[10] McIntire, G. (2017). *Fake or real news*. Retrieved from https://github.com/joolsa/fake_real_news_dataset

[11] Mikolov, T., Chen, K., Corrado, G., and Dean, J.

(2013). Efficient estimation of word representations in vector space. *International Conference on Learning Representations*. https://doi.org/10. 48550/arXiv.1301. 3781

[12] Nasir, J. A., Khan, O. S., and Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights*, *1*(1), 100007.

[13] Noosom, N. (2018). *The Analysis of Fake News and The Level of Media Literacy of Users in Bangkok*. Master of Communication Arts, Bangkok University.

[14] Olah, C. (2015). Understanding LSTM Networks. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[15] Patwa, P., Sharma, S., PYKL, S., Guptha, V., Kumari, G., Akhtar, M. S., Ekbal, A., Das, A., and Chakraborty, T. (2021). Fighting an infodemic: COVID-19 fake news dataset. *Combating Online Hostile Posts in Regional Languages during Emergency Situation*. https://doi.org/10.48550/arXiv.20 11.03327

[16] Phi, M. (2018). Illustrated Guide to LSTM's and GRU's: A step by step explanation. Retrieved from https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[17] Promrit, N. (2020). Sentiment Analysis 101. Retrieved from https://blog.pjjop.org/sentiment-analysis-101/?fbclid=IwAR1YLY8egVdu04K1B3wM7WRdka5358qOU6H-Ev0mPJDeOA_5AiCE_qR1l9A

[18] Ralucachitic. (2018). real_or_fake. Retrieved from https://www.kaggle.com/datasets/rchitic17/real-or-fake

[19] Samadi, M., Mousavian, M., and Momtazi, S. (2021). Deep contextualized text representation and learning for fake news detection. *Information Processing and Management*, *58*(6), 102723.

[20] Sastrawan, I. K., Bayupati, I. P. A., and Arsa, D. M. S. (2021). Detection of fake news using deep learning CNN-RNN based methods. *The Korean Institute of Communications and Information Sciences (KICS)*, *8*(3), 396-408.

[21] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

[22] Singkhornart, T. (2021). *Deep Learning for Video Subtitle Detection and Recognition*. Master of Science, Mahasarakham University.

[23] Tangruamsub, S. (2017). Long Short-Term Memory (LSTM). Retrieved from https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6

[24] TensorFlow. (2022). word2vec. [Online]. Retrieved from https://www.tensorflow.org/tutorials/text/word2vec

[25] UTK Machine Learning Club. (2018). *Fake news*. Retrieved from https://www.kaggle.com/c/fake-news/overview

[26] Wang, W. Y. (2017). "Liar, liar pants on fire": A new benchmark dataset for fake news detection. *In Proceedings of the 55th annual meeting of the Association for Computational Linguistics*, *2*, 422-426.

[27] World Health Organization. (n.d.). Coronavirus disease (COVID-19). [Online]. Retrieved from https://www.who.int/health-topics/coronavirus#tab=tab_1

◆ About the Authors ◆

Rutchaneewan Kowirat

Rutchaneewan Kowirat is a master's degree student in the KMITL Digital Analytics and Intelligence Center from the Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. She has a bachelor's degree in computer engineering from Bangkok University, Thailand. Her research interests include deep learning and detecting fake news.

Laor Boongasame

Laor Boongasame is a lecturer in the School of Science, King Mongku's Institute of Technology Ladkrabang, Bangkok, Thailand. Laor is the head of the Business innovation and investment Laboratory: B2I-Lab, School of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. She obtained her Ph.D. in Computer Engineering from King Mongkut's University of Technology Thonburi, Thailand. Her research interests involve buyer coalitions, n-person game theory, and investment. She has published several research papers in internationally refereed journals and has presented several papers at several international conferences. She can be reached at laor.bo@kmitl.ac.th

Submitted: June 7, 2022; 1st Revision: September 10, 2022; 2nd Revision: October 7, 2022; Accepted: October 18, 2022