Thai Conversational Chatbot Classification Using BiLSTM and Data Augmentation

Nunthawat Lhasiw[1], Tanatorn Tanantong[1](✉), and Nuttapong Sanglerdsinlapachai[2]

[1] Thammasat Research Unit in Data Innovation and Artificial Intelligence, Department of Computer Science, Faculty of Science and Technology, Thammasat University, Pathum Thani, Thailand
nunthawat.lha@dome.tu.ac.th, tanatorn@sci.tu.ac.th

[2] Strategic Analytics Networks with Machine Learning and AI Research Team, National Electronics and Computer Technology Center, Pathum Thani, Thailand
nuttapong.sanglerdsinlapachai@nectec.or.th

Abstract. Chatbot platforms, e.g., Facebook and Line, have revolutionized human interaction in the digital age. In order to develop an automatic chatbot classification, there are several challenges especially for Thai chat messages. Conversational messages are usually short and ambiguous. Therefore, it is difficult to find a dataset for constructing an effective classification model. To address the limited size of the dataset, data augmentation techniques can be possibly applied. Data augmentation involves generating synthetic messages by applying various transformations to existing data samples while preserving their original meaning. In this study, the size and diversity of the dataset is increased by two methods, i.e., text augmentation using word2vec from Thai2Fit and English-Thai machine translation models proposed by VISTEC. Based on the augmented messages, a Deep Learning technique, BiLSTM, is used to construct a chatbot classification model. The experimental obtained results demonstrate that data augmentation can help to increase the classification performance.

Keywords: Chatbot Classification · Data Augmentation · Deep Learning · BiLSTM

1 Introduction

As the usage of social media platforms continues to rapidly increase [1], people can freely share their opinions and thoughts [2]. Online platforms e.g., Facebook and Line offer convenient and rapid channels for people to engage in conversations and share information. These platforms have the way people to interact, providing features such as automated question-answering systems and interactive exchanges. Users can communicate seamlessly, regardless of location or time constraints [3]. Conversational messages are typically brief and ambiguous, making it challenging to find a suitable dataset for constructing an effective classification model. One technique that can address the issue of insufficient data is data augmentation, which has been applied and presented in

C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 127–141, 2023.
https://doi.org/10.1007/978-981-99-7969-1_10

several studies [4–7]. Data augmentation serves to enhance both the size and quality of the training data, thereby improving the model's generalization capabilities. Some of the techniques involved in data augmentation include back translation and word embedding. More specifically, back translation relies on a transformer architecture, encompassing an encoder-decoder model. The paraphrasing approach utilizes machine translation models that translate text into an intermediate language and then back into the original language. In our approach, the original text is input into a Thai-English translation model, followed by an English-Thai mixture of experts translation model [2]. Another robust augmentation method is Word2vec, which employs a word embedding model trained on a public dataset to identify the most similar words for a given input word. This technique is referred to as Word2vec-based (learned semantic similarity) augmentation [10].

Moreover, deep learning techniques have shown great potential to understand the meaning of the sentence which is written in different forms [11]. One such technique is the Bidirectional Long Short-Term Memory (BiLSTM) model [8, 12], which extends the capabilities of the traditional Recurrent Neural Network (RNN) [9]. BiLSTM excels in capturing and understanding the meaning of both short and long text sequences, making it highly suitable for tasks such as text classification. By leveraging the power of BiLSTM, chatbots can better analyze and interpret user inputs, delivering more accurate and contextually relevant responses [10]. In this study, we have conducted a study on Thai Conversational Chatbot Classification but we have the small sample of chat message. To address this problem, we propose a method for Thai Conversational Chatbot Classification by employing the potential of data augmentation techniques to solve the small sample of chat message and deep learning models; namely, BiLSTM.

2 Related Works

2.1 Conversational Chatbot Classification

Many researchers have proposed deep learning models for classifying textual data. Among the related works, Maktapwong, P. and colleagues [11] introduced a chatbot application for breast cancer patients in Thailand. This application aims to increase communication channels and address issues related to a shortage of medical staff. In their study, they employed the deep learning model BiLSTM to construct chatbot conversation messages from breast cancer patients. They chose BiLSTM because it can effectively capture contextual information in chatbot conversations and demonstrated efficiency in text processing. The reported accuracy of their experiments was 86.90%.

Anki, P. and colleagues [9] proposed LSTM and BiLSTM models for classifying chatbot messages, implemented using the Python programming language. They subsequently evaluated the performance of both LSTM and BiLSTM models. Furthermore, they discussed the attributes of LSTM, which can learn data over long distances. They highlighted that combining LSTM models can lead to enhanced learning, as it enables data processing in two directions, encompassing both past and future information. Based on the experimental results, the deep learning BiLSTM model demonstrated outstanding performance in classifying chatbot messages, achieving an accuracy rate of 99.52%.

L. Shi and their team [8] introduced a BiLSTM model designed for understanding the context of dialogues in online chatbots. They collected the dataset for their study using the Scrapy tool, which extracted data from open-source projects like AngularJS, Bootstrap, and Chromium, yielding a total of 65,428 dialogues. The model evaluation demonstrated that the approach employed in the study successfully met its objectives, achieving an average precision, recall, and F1-score of 88.52%, 88.50%, and 88.51%, respectively.

R. Anhar and their colleagues [12] conducted a study using a BiLSTM model for question classification. Question classification plays a crucial role in question-answer systems as it directly impacts the accuracy of generated answers. Traditional approaches, such as Support Vector Machine (SVM), pattern matching, naive Bayes classification, and Latent Dirichlet Allocation (LDA), have been utilized for question classification. However, they are constrained by specific sentence patterns. To overcome these limitations, deep learning methods, including Bidirectional Long Short Term Memory (BiLSTM), have been proposed. In this study, the effectiveness of BiLSTM in question classification was evaluated, achieving an accuracy of 90.90% with a loss of 31.60%. Based on Literature reviews above, a Bi-LSTM model is potential effective model for classifying texts, particularly in the context of chat messages.

2.2 Text Classification Using Data Augmentation

There are various approaches to working with language in computers, such as employing back translation, substituting words with synonyms, or utilizing word embeddings. However, data augmentation in text-based tasks and natural language processing (NLP) remains a challenge. In related research, Beddiar, D.R and their team [2] proposed a data augmentation method to address issues related to insufficient datasets. Their method primarily utilized back translation, which was based on a transformer architecture incorporating an encoder-decoder model. The original text was processed through an English-French translation model, followed by a French–English mixture of experts' translation model. The study involved five original datasets, which included AskFM, Formspring, Olid, Warner and Waseem, and Wikipedia Toxic. The results were reported in terms of accuracy, F1 score, precision, and recall. Notably, they achieved a high recall score of 99.7% and a precision of 99.6% for the expanded version of the Warner and Waseem dataset. In summary, the best accuracy and F1 score, at 99.4%, were recorded for the expanded Wikipedia toxic comments dataset.

Phreeraphattanakarn, T. and their colleagues [4] encountered limitations in the available data, as well as unequal data group sizes within an automatic chatbot dataset. Through their study, the researchers discovered that data augmentation techniques can significantly enhance model performance. They employed this data augmentation technique to expand the training dataset, which primarily consisted of text data. Initially, the dataset comprised 339,985 pairs of sentences. The researchers applied data augmentation by leveraging word embeddings from the pre-trained Thai2fit model and cosine similarity to identify similar words within sentences. This approach resulted in a substantial increase in the number of sentence pairs, totaling 1,329,197. The results of the model's performance evaluation on the testing dataset and the augmented dataset yielded F1 scores of 0.088 and 0.071, respectively.

Ma, J. and their colleagues [5] employed data augmentation techniques to address limitations in their dataset by applying Back Translation and EDA (Easy Data Augmentation) to Chinese language data. The Chinese dataset comprised toxic comments obtained from the Kaggle platform. In their approach, the researchers translated the Chinese texts into English and then back-translated them to the original language. For this study, deep learning models, specifically LSTM and CNN, were utilized to classify toxic comments. The models' performance was assessed based on their accuracy, and the experimental results clearly demonstrated that the back translation technique significantly improved the effectiveness of the models.

Rizos, G., and their colleagues [6] aimed to tackle the challenge of augmenting text data for hate speech classification by introducing three text-based data augmentation techniques tailored specifically for text. These techniques encompass synonym replacement based on word embedding vector similarity. Deep learning approaches, particularly those employing word embedding techniques, have proven to be more effective in hate speech classification. The proposed framework exhibits a substantial improvement in hate speech detection, surpassing the baseline performance in the largest online hate speech database by an absolute 5.7% increase in Macro-F1 score and a 30% boost in hate speech class recall.

Fadaee et al. [7] have demonstrated that Back Translation (BT) can serve as a valuable method for expanding datasets without necessitating modifications to the training process of language translation models. The paper presents experimental results for the WMT news translation task, with a specific focus on German-English and English-German translation pairs. Furthermore, the study uncovered that the inclusion of synthetic data proves advantageous, particularly for words exhibiting high prediction loss during training.

3 Methodology

This section presents the process of data collection and preparation. Following that, we expanded the original dataset using two methods: text augmentation using word2vec from Thai2Fit (Data Augmentation by PyThaiNLP) and English-Thai machine translation models proposed by VISTEC (Data Augmentation by VISTEC). Next, we applied feature extraction. Lastly, we utilized the processed data to construct the classification model. Figure 1. Demonstrates a framework for Thai Conversational classification.

3.1 Data Collection and Preparation

This research collected chat text data from the office of the registrar, Thammasat University, with a specific focus on chat messages related to topics such as 'Document Request,' 'Registration,' 'Payment/Invoice,' 'Graduate Registration,' 'Grade,' and 'Student Profile.' The data collection period extended from March to April and from August to October 2021, resulting in a total of 3,609 chat messages. Each message was meticulously categorized by experts. Table 1 presents an example of messages and their corresponding classes.

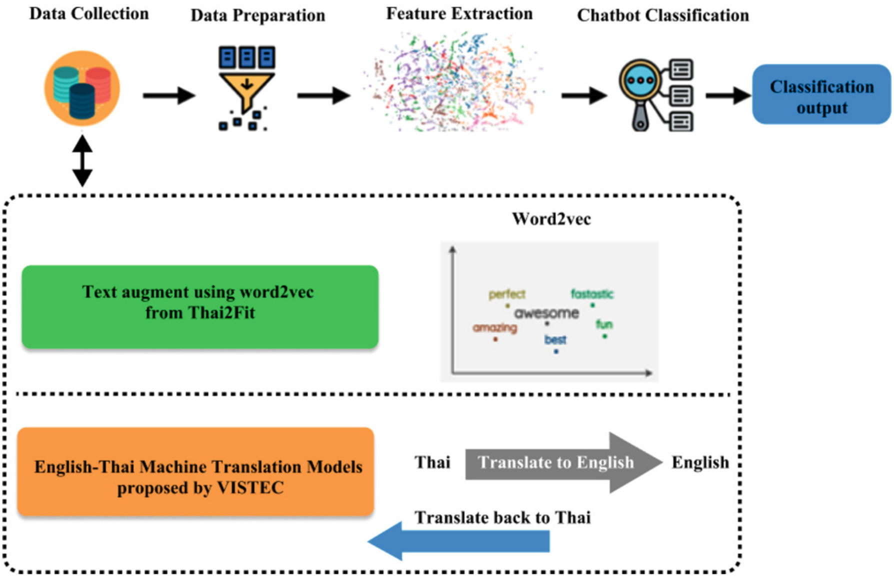


Fig. 1. Thai Conversational Chatbot Classification Framework.

Furthermore, data transformation is an important step in preparing the data for building a neural network for text classification in chatbot messages. In this study, data processing involves two main processes: 1) Data Transformation on a Message and 2) Data Transformation on a Class Label.

Regarding Data Transformation on a Message, encoding the data requires several preprocessing steps to clean the text and make it more suitable for subsequent analysis. These processes include:

1. Removal of special characters from a message: This step involves eliminating special characters such as ‘?’, ‘#’, and ‘,’ from the text.
2. Word Tokenization: The text is segmented into individual words to facilitate further processing.
3. Stop words removal: In this step, insignificant words, known as stop words, are removed from the text.
4. Indexing: The text is indexed, enabling efficient data retrieval and analysis.
5. Zero padding: All input sequences are adjusted to match the length of the longest text by replacing shorter sequences with zeros.

In terms of Data Transformation on a Class Label, Anhar [12] stated that the conversion of text categories into binary vectors using the one-hot encoding principle facilitates text categorization. This transformation represents each text category as a binary list, where the value 1 indicates membership in that specific category. The size of the text vector corresponds to the total number of categories. An example of data transformation on a class label is provided in Table 2.

Table 1. Chat messages annotated by experts.

Message	Class
อาจารย์คะพอจะมีไฟล์ใบขึ้นทะเบียนนักศึกษามีไหมคะ (Teacher, do you have the student registration form file?)	‘Document Request’
ต้องลงทะเบียนวันทีลงล่าช้าหรอครับ (Do I need to register on the late date?)	‘Registration’
แล้วใบเสร็จก็คือเอาไปเบิกของค่าราชการได้เลนมั้ยคะ (Can be used the receipt to withdraw government expenses?)	‘Payment/Invoice ‘
จบแล้วค่ะ แต่ไม่มีปุ่มขึ้นบัณฑิตซักทีเลยค่า (There is no graduate button.)	‘Graduate registration’
โดยปกติต้องมีเวลากำหนดการส่งเกรดมายังสำนักงานทะเบียนใช่ไหมครับ เพราะหากเด็กลงซัมเมอร์คะแนนจะต้องออกก่อนลงทะเบียนเรียน (Is there time to submit grades? Because, they must know the grade if a student wants registering on summer.)	‘Grade’
นักศึกษาใหม่กรอกประวัติผิด (New students filled in wrong information)	‘Student Profile’
นักศึกษารหัส 64 อยากทราบว่าใช้เวลานานมั้ยคะ กว่าสถานะรอขึ้นทะเบียน จะเปลียนเป็นปกติ พอดีมีความจำเป็นต้องยืมไอแพด และหนังสือจากห้องสมุดเร่งด่วนค่ะ (Student code 64, would like to know when the status is pending registration will change to normal. It is necessary to borrow an iPad and books from the library)	‘Student Status’

Table 2. Example data transformation with annotated label.

Class	Label encoding						
‘Document Request’	1	0	0	0	0	0	0
‘Registration’	0	1	0	0	0	0	0
‘Payment/Invoice’	0	0	1	0	0	0	0
‘Graduate registration’	0	0	0	1	0	0	0
‘Grade’	0	0	0	0	1	0	0
‘Student Profile’	0	0	0	0	0	1	0
‘Student Status’	0	0	0	0	0	0	1

3.2 Data Augmentation

We performed data augmentation on the original dataset using 2 methods: 1. text augmentation using word2vec from thai2fit (data augmentation by pythainlp) [13] and 2. english-thai machine translation models proposed by vistec (data augmentation by VISTEC) [14].

3.2.1 Text augmentation using word2vec from Thai2Fit (Data Augmentation by PyThaiNLP)

Text augmentation using word2vec from Thai2Fit (Data Augmentation by PyThaiNLP) [13] involved the application of the word2vec principle. Word2vec is a word embedding technique or vectorization method that assigns vectors to words, aiding in the search for words with similar meanings. An illustrative example of data augmentation using the Thai2Fit method is provided in Table 3.

Table 3. An example of increasing the amount of data using Data Augmentation by PyThaiNLP.

Class	Original Messages	Augmented Messages
'Document Request'	อาจารย์คะพอจะมีไฟล์ใบขึ้นทะเบียนนักศึกษามีไหมคะ (Teacher, do you have the student registration form file?)	อาจารย์คะพอจะมีไฟล์ใบขึ้นทะเบียนนักศึกษามีไหมคะ (Dear teacher, may I inquire if you have the file for the student registration form?)
'Registration'	ต้องลงทะเบียนวันทีลงล่าช้าหรอครับ (Do I need to register on the late date?)	ต้องลงทะเบียนวันทีลงล่าช้าหลิมครับ (You must register on the date of late entry.)
'Payment/Invoice '	แล้วใบเสร็จก็คือเอาไปเบิกของค่าราชการได้เลยมั้ยคะ (Can be used the receipt to withdraw government expenses?)	แล้วrocksก็คือเอาไปเบิกของค่าราชการได้เลนเดี๋ยวคะ (Then the rocks are that you can take it to pay for government expenses. You can wait.)

3.2.2 English-Thai Machine Translation Models Proposed by VISTEC (Data Augmentation by VISTEC)

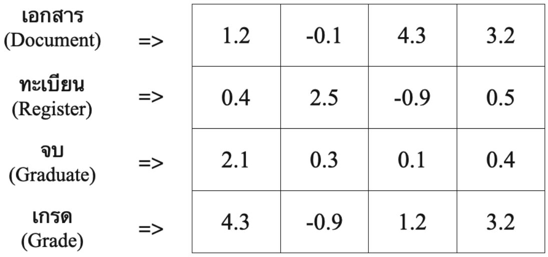
English-Thai machine translation models proposed by VISTEC (data augmentation by VISTEC) [14] is a backtranslation method. The backtranslation method is the process of translating the original language into English and then translating it back into the original language to generate new sentences. An example of original messages and augmented messages generated by data augmentation by VISTEC can be seen in Table 4.

Table 4. Original messages and augmented messages generated using data augmentation by VISTEC.

Class	Original messages	Augmented messages
‘Document Request’	อาจารย์คะพอจะมีไฟล์ใบขึ้นทะเบียนนักศึกษามีไหมคะ (Teacher, do you have the student registration form file?)	คุณมีใบอนุญาตสำหรับอาจารย์ของคุณหรือไม่? (Will you have to make your teacher?)
‘Registration’	ต้องลงทะเบียนวันทีลงล่าช้าหรอครับ (Do I need to register on the late date?)	ฉันต้องลงทะเบียนสำหรับวันทีล่าช้าหรือไม่? (Do I have to register for the late date?)
‘Payment/Invoice ‘	แล้วใบเสร็จก็คือเอาไปเบิกของค่าราชการได้เลนมั้ยคะ (Can the receipt be used to withdraw from the government?)	แล้วใบเสร็จจะถูกนำไปหักเป็นค่าบริการอย่างเป็นทางการมั้ยคะ (And will the receipt be deducted as an official service fee?)

3.3 Feature Extraction

Feature extraction holds a crucial role in text classification because it directly influences classification accuracy [15]. Word Embedding is a technique used to represent words in a distributed manner, improving the accuracy of natural language processing models. It involves extracting features from a specific type of word to enhance the efficiency of neural networks in classifying diverse data categories [16]. Figure 2 shows the example of feature extraction using word embedding principle.



เอกสาร (Document)	=>	1.2	-0.1	4.3	3.2
ทะเบียน (Register)	=>	0.4	2.5	-0.9	0.5
จบ (Graduate)	=>	2.1	0.3	0.1	0.4
เกรด (Grade)	=>	4.3	-0.9	1.2	3.2

Fig. 2. Features extracted by word embedding principle.

3.4 Chatbot Classification

Figure 3 illustrates the process of classifying Thai conversational chatbot messages. The initial step involves preparing the Thai Conversational Chatbot Messages. Data augmentation is performed using two distinct methods: one proposed by VISTEC and another by Py-ThaiNLP. The method by VISTEC utilizes the back-translation concept,

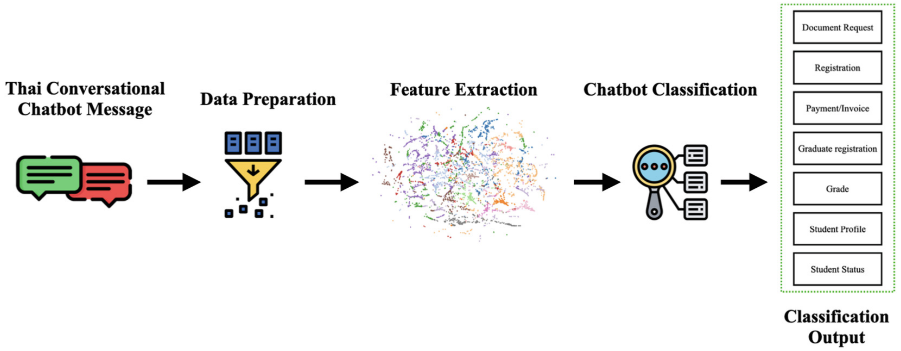


Fig. 3. Thai conversational chatbot messages classification processes.

employing the Transformer model for this purpose. The transformer functions as an encoder-decoder network. In this setup, the encoder captures pertinent information from an input sentence, while the decoder uses this information to generate an output sentence. This architecture is typically used for tasks like translation. On the other hand, the method by PyThaiNLP employs the word2vec concept, which assigns vector representations to words, facilitating the identification of words with similar meanings.

In the data preparation step, Thai conversational chatbot messages are converted into vectors. We then extract features from these vectors using the word embedding principle. The extracted features are utilized as input for our deep neural model, which aims to classify Thai conversational chatbot messages. In the chatbot classification phase, we employ BiLSTM. LSTM (Long Short-Term Memory) follows the basic architecture of RNN (Recurrent Neural Network) but effectively addresses one of the drawbacks of Simple RNN by learning long-term dependencies in text datasets [16]. BiLSTM shares the same underlying concept as LSTM [17] but adds bidirectional propagation. This means that the BiLSTM architecture learns from both past to future directions. The backward propagation layer essentially functions as the reverse of the forward LSTM, making the architecture more stable and capable as it operates from both ends. Finally, the following section will calculate and analyze the obtained results. You can observe the model architecture of BiLSTM for text classification in Fig. 4.

4 Experimental Settings and Results

4.1 Experimental Settings

In this study, the classification models are trained from 7 datasets, i.e., 1. Original dataset, 2. VISTEC dataset (Augmenting Original dataset with data augmentation by VISTEC), 3. VISTEC to VISTEC dataset (Augmenting VISTEC dataset with data augmentation by VISTEC), 4. VISTEC to Thai2fit dataset (Augmenting VISTEC dataset with data augmentation by PyThaiNLP), 5. Thai2Fit (Augmenting Original dataset with data augmentation by PyThaiNLP) and, 6. Thai2Fit to Thai2Fit dataset (Augmenting the Thai2Fit

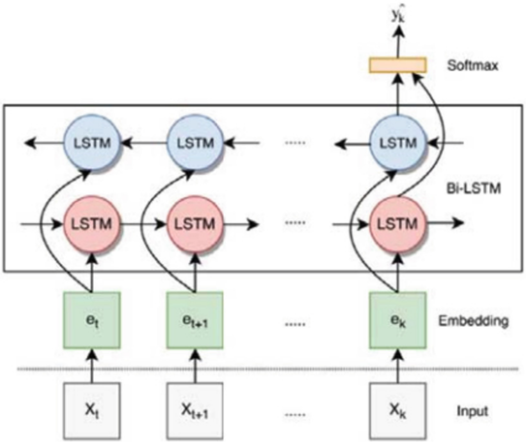


Fig. 4. The network structure of BiLSTM.

dataset with Data Augmentation by PyThaiNLP), and 7. Thai2Fit to VISTEC (Augmenting Thai2Fit dataset with data augmentation by VISTEC). The performance of the classification models is evaluated using well-established metrics, including precision, recall, F-value, and accuracy. Our results encompass outcomes from two key aspects: the 10-fold cross-validation and testing on an independent test dataset. The dataset used in this research comprises conversation texts exchanged between students and university staff. These conversations span a range of topics, including 'Document Request,' 'Registration,' 'Payment/Invoice,' 'Graduate registration,' 'Grade,' 'Student Profile,' and 'Student Status.' For more detailed information, please refer to Table 5, which also illustrates the extent of dataset expansion achieved through data augmentation by PyThaiNLP and VISTEC.

This research study has divided the data into 3 parts. The first two parts were used for model training, with 80% of the dataset. The remaining part was reserved for model testing, utilizing 20% of the dataset. In this study, we investigate the parameter settings of a classification model, specifically examining the values used for various parameters. These crucial parameters include the number of nodes in the BiLSTM layer, epoch, batch_size, dropout rate, loss function, and optimizer. The choice of these settings significantly influences the performance and effectiveness of the classification model. Our goal is to gain insights into how the values assigned to each parameter impact the overall performance of the model. You can find the specific values of each parameter in Table 6. Referring to Table 6, the number of nodes in the BiLSTM layer represents the number of output nodes within the layer. In our model, we selected 100 nodes for this purpose. It's worth noting that many studies utilize specific hyperparameter settings for building textual classification models. In our case, we adopted hyperparameter values from these studies as well. For instance, Maktapwong and colleagues [11] used 20 epochs to train their model. An epoch represents an iteration over the entire set of feature and label data.

Table 5. Size of datasets before and after expansion using data augmentation by PyThaiNLP, data augmentation by VISTEC, and a combination of original dataset with both techniques.

Class	Original dataset	VISTEC dataset	VISTEC to VISTEC dataset	VISTEC to Thai2fit dataset	Thai2Fit dataset	Thai2Fit to Thai2fit dataset	Thai2Fit to VISTEC dataset
Document Request	951	2222	8129	10181	34962	5330	17142
Registration	146	1904	5837	9814	32966	4569	16226
Payment/Invoice	713	1426	4424	9646	32426	3422	15661
Graduate registration	129	888	3164	8912	30536	2075	15157
Grade	1107	250	720	8509	29075	701	14328
Student Profile	444	292	1094	8473	28609	619	14117
Student Status	119	258	565	8511	28664	535	14179
Total	3609	7240	23933	64046	217238	17251	106810

Table 6. Hyperparameter setting value of the classification model.

Hyperparameter	Original dataset	VISTEC dataset	VISTEC to VISTEC dataset	VISTEC to Thai2fit dataset	Thai2Fit dataset	Thai2Fit to Thai2fit dataset	Thai2Fit to VISTEC dataset
number of nodes in BiLSTM layer	100						
epoch	20						
batch_size	128						
dropout rate	0.5						
loss function	categorical_crossentropy						
optimizer	Adam						

Anki and colleagues [9] employed a batch_size of 128 in their model. Batch size represents the number of training data instances used in a single iteration. Additionally, Ma and colleagues [5] utilized a dropout rate of 0.5. The dropout rate is a crucial parameter set to prevent overfitting in neural networks. They also employed the categorical_crossentropy loss function in combination with the Adam optimizer to construct their text classification model.

4.2 Experimental Results

In this section, we present the comprehensive results of our experiments, including two key components: the outcomes of 10-fold cross-validation and the results obtained from testing on the independent test dataset. The experiment results of the 10-fold cross-validation encompass the following 7 experiments, as follows: 1. Original dataset, 2. Original dataset + VISTEC dataset (augment1 dataset), 3. Original dataset + VISTEC to VISTEC dataset (augment2 dataset), 4. Original dataset + VISTEC to Thai2fit dataset (augment3 dataset), 5. Original dataset + Thai2Fit (augment4 dataset), 6. Original dataset + Thai2Fit to Thai2Fit dataset (augment5 dataset), and 7. Original dataset + Thai2Fit to VISTEC (augment6 dataset). The experiment results for the test dataset, which was evaluated using 7 test datasets, are presented in Table 8, as follow: 1. Original dataset, 2. VISTEC dataset (augment1 dataset), 3. VISTEC to VISTEC dataset (augment2 dataset), 4. VISTEC to Thai2fit dataset (augment3 dataset), 5. Thai2Fit dataset (augment4 dataset), 6. Thai2Fit to Thai2fit dataset (augment5 dataset) and 7. Thai2Fit to VISTEC dataset (augment6 dataset). Additionally, precision, recall, F-value, and accuracy metrics are utilized to evaluate the performance of chat message classification model.

From Table 7, a model trained with the original dataset exhibits excellent performance across multiple metrics, with precision at 97.01%, recall at 96.91%, F-score at 96.91%, and accuracy at 96.92%. To address the limitations of the original dataset's size, we increased its size by concatenating it with augmented datasets. When we combined the original data with the augment1 dataset, we observed a slight decrease in performance. However, merging the original dataset with the augment2 dataset resulted in a slight improvement, with precision at 94.51%, recall at 94.45%, F-score at 94.45%, and accuracy at 94.46%. Further enhancements in performance were achieved when joining the original dataset with the augment3 dataset. This combination yielded the highest precision (95.67%), recall (95.64%), F-score (95.64%), and accuracy (95.63%). Additionally, combining the original dataset with the augment5 dataset increased the volume and led to improved performance, with precision at 97.65%, recall at 97.63%, F-score at 97.63%, and accuracy at 98.38%. However, there was a significant drop in performance when integrating the original dataset with the augment6 dataset. Clearly, the highest performance was achieved when combining the original dataset with the augment4 dataset. This combination resulted in significantly improved precision (98.40%), recall (98.38%), F-score (98.38%), and accuracy (98.38%). For further performance evaluation, we also reported the results of using a test dataset to assess the model, as shown in Table 8.

According to Table 8, we evaluated the model using the test dataset for each variant. The original dataset exhibited moderate performance, achieving precision at 74.57%, recall at 72.57%, F-score at 73.00%, and accuracy at 80.00%. When we used the augment1 dataset for testing with the model trained from the original dataset and the augment1 dataset, there was an improvement in performance. Precision reached 79.00%, recall stood at 78.29%, F-score reached 78.71%, and accuracy reached 83.00%. Testing the model trained from the original dataset and the augment2 dataset with the augment2 dataset further enhanced the results. Precision increased to 82.14%, recall improved to 78.00%, F-score rose to 79.43%, and accuracy reached 84.00%. Evaluating the model trained with the original dataset and the augment3 dataset with the augment3 dataset yielded a precision of 74.00%, recall of 84.00%, F-score of 79.00%, and accuracy of

Table 7. Experiment results of 10 folds cross-validation.

Models trained with each dataset	Evaluation Matrices			
	Precision	Recall	F-score	Accuracy
original dataset	97.01%	96.91%	96.91%	96.92%
original dataset + augment1 dataset	94.50%	94.28%	94.28%	94.23%
original dataset + augment2 dataset	94.51%	94.45%	94.45%	94.46%
original dataset + augment3 dataset	95.67%	95.64%	95.64%	95.63%
original dataset + augment4 dataset	**98.40%**	**98.38%**	**98.38%**	**98.38%**
original dataset + augment5 dataset	97.65%	97.63%	97.63%	97.63%
original dataset + augment6 dataset	97.08%	97.06%	97.06%	97.06%

Table 8. Experiment results of the test dataset.

Models trained with each dataset	Evaluation Matrices			
	Precision	Recall	F-score	Accuracy
original dataset	74.57%	72.57%	73.00%	80.00%
augment1 dataset	79.00%	78.29%	78.71%	83.00%
augment2 dataset	82.14%	78.00%	79.43%	84.00%
augment3 dataset	74.00%	84.00%	79.00%	80.00%
augment4 dataset	**87.29%**	**86.71%**	**86.57%**	**91.00%**
augment5 dataset	86.14%	85.43%	85.57%	89.00%
augment6 dataset	76.86%	75.57%	76.00%	80.00%

80.00%. The highest performance was observed when testing the model from the original dataset and the augment4 dataset with the augment4 dataset. It achieved precision at 87.29%, recall at 86.71%, F-score at 86.57%, and accuracy at 91.00%. Similarly, the model trained from the original dataset and the augment5 dataset maintained consistently high performance, with precision at 86.14%, recall at 85.43%, F-score at 85.57%, and accuracy at 89.00%. Finally, testing the model from the original dataset and the augment6 dataset with the augment6 dataset resulted in precision at 76.86%, recall at 75.57%, F-score at 76.00%, and accuracy at 80.00%.

5 Conclusion

In this paper, we conducted a study on Thai conversational chatbot classification, despite having a small sample of chat messages. To address this limitation, we employed data augmentation methods, specifically Data Augmentation by PyThaiNLP and Data Augmentation by VISTEC, to expand the size of the datasets. We utilized the deep learning model, BiLSTM, for chat message classification. The experimental results, derived from the original dataset augmented with the Thai2Fit dataset using 10-fold cross-validation, showcased exceptional performance with a precision of 98.40%, recall of 98.38%, F-score of 98.38%, and accuracy of 98.38%. Moreover, the results from testing on the test dataset using 10-fold cross-validation yielded a precision of 87.29%, recall of 86.71%, F-score of 86.57%, and accuracy of 91.00%. Based on the experimental findings for Thai Conversational Chatbot Classification, it is evident that data augmentation by PyThaiNLP consistently outperforms data augmentation by VISTEC. For future work, we plan to explore different datasets and data augmentation methods to further enhance the performance of chatbot classification models.

References

Tanantong, T., Parnkow, M.: A survey of automatic text classification based on Thai social media data (2022). https://doi.org/10.4018/IJKSS.312578

Beddiar, D.R., Jahan, M.S., Oussalah, M.: Data expansion using back translation and paraphrasing for hate speech detection. 53 (2021). https://doi.org/10.1016/j.osnem.2021.1001

Lhasiw, N., Sanglerdsinlapachai, N., Tanantong, T.: A bidirectional LSTM model for classifying chatbot messages 173 (2021). https://doi.org/10.1109/iSAI-NLP54397.2021.9678

Phreeraphattanakarn, T., Kijsirikul, B.: Text data-augmentation using text similarity with manhattan siamese long short-term memory for Thai language (2021). https://doi.org/10.1088/1742-6596/1780/1/012018

Ma, J., Li, L.: Data augmentation for Chinese text classification using back-translation (2020). https://doi.org/10.1088/1742-6596/1651/1/012039

Rizos, G., Hemker, K., Schuller, B.: Augment to prevent (2019). https://doi.org/10.1145/3357384.3358040

Fadaee, M., Monz, C.: Back-translation sampling by targeting difficult words in neural machine translation (2018). https://doi.org/10.18653/v1/D18-1040

Shi, L., Xing, M., Li, M., Wang, Y., Li, S., Wang, Q.: Detection of hidden feature requests from massive chat messages via deep siamese network (2020). https://doi.org/10.1145/3377811.3380356

Anki, P., Bustamam, A.: Measuring the accuracy of LSTM and BiLSTM models in the application of artificial intelligence by applying chatbot programme (2021). https://doi.org/10.11591/ijeecs.v23.i1

Gong, X., Ying, W., Zhong, S., Gong, S.: Text sentiment analysis based on transformer and augmentation. https://doi.org/10.3389/fpsyg.2022.906061 (2022)

Maktapwong, P., Siriphornphokha, P., Tubglam, S., Imsombut, A.: message classification for breast cancer chatbot using bidirectional LSTM (2022). https://doi.org/10.1109/ITC-CSCC55581.2022.9895035

Anhar, R., Adji, T.B., Akhmad Setiawan, N.: question classification on question-answer system using bidirectional-LSTM (2019). https://doi.org/10.1109/ICST47872.2019.9166190

Source code for pythainlp.augment.word2vec.thai2fit. pythainlp.augment. word2vec.thai2fit - PyThaiNLP 4.0.2 documentation. https://pythainlp.github.io/dev-docs/_modules/pythainlp/augment/word2vec/thai2fit.html

Data Augmentation in NLP in a world that craves data, with Back Translation

Dzisevic, R., Sesok, D.: text classification using different feature extraction approaches (2019). https://doi.org/10.1109/eStream.2019.8732167

Pham, T.-H., Le-Hong, P.: End-to-end Recurrent Neural Network Models for Vietnamese Named Entity Recognition: Word-level vs. Character-level (2017)

Recurrent neural networks for prediction - lagout.org. https://doc.lagout.org/science/0_Computer%20Science/3_Theory/Neural%20Networks/Recurrent%20Neural%20Networks%20for%20Prediction.pdf

Wang, Y., Huang, M., zhu, xiaoyan, Zhao, L.: Attention-based LSTM for aspect-level sentiment classification (2016). https://doi.org/10.18653/v1/D16-1058