Text data-augmentation using Text Similarity with Manhattan Siamese long short-term memory for Thai language

Thananya Phreeraphattanakarn* and Boonserm Kijsirikul

1Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand 10330

*E-mail: 6170930921@student.chula.ac.th

Abstract. In this paper, we address the issue of using small text datasets for learning of neural networks. We explore the method that is used with image and sound datasets to augment data for increasing the performance of models. We then leverage this data augmentation technique to expand the training set of textual data. A great challenge in our dataset is that the amount of data is insufficient for training models. For this reason, we propose a method for augmenting text data specifically for Thai language which is based on Text Similarity and using the model to determine the semantic relationship between two sentences. The experimental results indicated that our proposed method is able to improve the performance of text classification.

1. Introduction
Nowadays, the usage of social media plays an important role in communication which is currently growing and becoming a popular trend. Therefore, many corporations and traders tend to conduct their businesses further on social media. The survey of Facebook [1] conducted by 8,000 Facebook users found that 69% felt that communicating by sending their messages directly to the company gave them more confidence in the product. This one shows that customers prefer to use social media channels to contact the company rather than to call for information via the call center which takes a long time.

A chatbot is a popular software application used to conduct an online conversation via text or text to speech with a live human [2]. The research found that 70% of the question from the customer is frequently asked questions (FAQ). Likewise, chatbots will search the answer from the most similarity questions and provide that answer for the customer. The popular method to find the similarity is Long-Short Term Memory (LSTM) together with Siamese Neural Network which consists of two main parts: (1) calculating the similarity of the question (2) answering the question with the answer of the most similarity question [3]. To develop the chatbot system, it is necessary to have sufficient information for training and testing; the amount of variety of data directly affects the result of the learned model.

This paper aims to present the method called Text Data-Augmentation for increasing the textual data and using Manhattan Siamese Long Short-Term Memory as a model for the evaluation. Our approach creates a new question by using cosine similarity for finding a similar word and replacing it in the same sequence. The method will accrue a variety of vocabularies in the dataset which will be
used as training data in machine learning. The next section presents related work. Section 3 presents the model followed by section 4 for the method and section 5 for the results. Finally, section 6 presents the conclusion of the paper.

2. Related work
This section is divided into two parts: Understanding Manhattan Siamese Long Short-Term Memory (MaLSTM) and Text-Data Augmentation.

2.1. The Siamese long short-term memory and distance similarity approach
J. Mueller and A. Thyagarajan. [3] presented the model using a dual encoder network, a Siamese-like neural network architecture for labeled data. This research is the state of the art for measuring the similarity between paired sentences. The input data was a word-embedding vector with labeled binary data. For the evaluation, Jonas used Manhattan Distance to measure the similarity of a representative sentence.

N. Othman, et al. [4] used Manhattan LSTM (MaLSTM) to analyze the entire question based on its words and its local contexts and predict the similarity between questions. This research used the Manhattan distance to measure the similarity between questions. The biggest challenge in the question retrieval task is the shortness of the community questions, which leads to the word mismatch problem. This research also used Question Expansion by adding terms to the community questions vectors to deal with short question sentences. The additional words are those having similar embedding vectors. For example, the original question is “Do chocolate really kill my dog?” containing 3 distinct words and the number of the additional similar words is set to 3. The expanded query after preprocessing will have 12 words as follows: chocolate kill dog eat death bitch candy toxic puppy food sick animal.

2.2. The text data-augmentation approach
A. Mosolova, et al. [5] presented a method for text augmentation, which consisted of using synonymous words. This research used a concept of data augmentation method which was already known as machine learning techniques for augmenting data for image or sound datasets. They are a common way to increase the performance of the model and avoid overfitting. However, this method is not suitable for text because of the danger of losing the sense of a sentence. The first step excluded all pronouns, conjunctions, prepositions, and articles so that they remain intact. The second step randomly replaced some words in the sentence with the synonyms. For example, if a sentence consists of 10 words, and augmentation is set for 25%, the algorithm will substitute two words in the sentence.

J. Wei and K. Zou. [6] presented a data augmentation technique for boosting performance for both convolutional and recurrent neural networks and demonstrated particularly strong results for smaller datasets on five text classification tasks. The technique in this research consists of four operations 1) synonym replacement 2) random insertion 3) random swap, and 4) random deletion. The result of this technique demonstrates the improvement of accuracy by using only 50% of training data together with the augmented dataset to achieve the same accuracy as normal training with all available data.

3. Our model
Our method applies the Manhattan LSTM model [3] as a model for assessing semantic similarity between sentences and enhances the model by using data created from our method as training data. The architecture of Manhattan LSTM model is shown in figure 1. There are two networks, i.e. LSTMa and LSTMb whose weights are tied. Initially, LSTMa and LSTMb are used to read the input word-vectors. Secondly, the final hidden layer corresponds to a vector representation for each sentence. Thirdly, the similarity between these vector representations from LSTMa and LSTMb is calculated by using Manhattan distance as a similarity function. Finally, the distance which is calculated by the similarity function is used for predicting the similarity between the input sentences of LSTMa and LSTMb.
4. Method
The purpose of the Text Data-Augmentation for the Thai language in this paper is to augment a training data. This technique can be useful when only small text datasets are available for neural networks, which can increase the performance of the model. Our method is summarized in figure 2.

4.1. Algorithm
This work applies the synonym word replacement technique [5][6]. However, instead of using dictionary-based synonym words, we consider word embeddings which are close to each others as synonym words. The main reason that we use the word embedding is due to the issue of low-language resource of Thai language. Thus, we propose an approach to augment text data that is based on using word embedding and the cosine similarity for finding a similar word in a sentence and using these words instead of the original ones.

The pipeline architecture of the method can be divided into three parts in figure 3: 1) data preprocessing 2) finding a similar word and creating the list of similar words 3) word replacement.

The first step is data preprocessing that consists of 1) data cleaning to remove symbolic data and numerical data, 2) word segmentation by using dictionary as shown in table 1, 3) stop-word removal to remove all common words from a sentence, and 4) word embedding by using a pre-trained word embedding, named Thai2Fit[1], which supports Thai language.

| Table 1. An example of the word Segmentation. |
|---------------------------------------------|

| Sentence | Word Segmentation |
|----------|-------------------|
| ขอโทษไม่พูดภาษาไทยของเรา | ขอ,โทษ,ไม่,พูด,ภาษา,ไทย,ของเรา |

| Table 2. An example of the list of similar words by using the cosine similarity. |
|-----------------------------------------------|

| Word | Similar Word |
|------|--------------|
| ขอโทษ | ขอ,โทษ |
| ไม่ | ไม่ |
| พูด | 说法 |
| ภาษา | 说法 |
| ไทย | 说法 |
| เรา | 说法 |

The second step is to find a similar word for each word in the sentence by using the cosine similarity with a threshold value of similarity score equal 0.97, selecting the highest similarity score to be a similar word, and adding it to the list of similar words as shown in table 2.
The third step is word replacement by using a similar word for each word from the list to substitute the word at the same position in the sentence. An example of the sentence after the three-step processing is shown in table 3.

The last step is word embedding by using a pre-trained word embedding to encode all the sentences after the three-step processing to vectors of real numbers for using with our model.

Table 3. An example of a sentence after Text Data-Augmentation.

| Sentence | Augmented Data |
|----------|----------------|
| example | similar word |
| sentence | replaced with |
| word | similar word |

4.2. Dataset
We used 1,169 sentences for our dataset and paired sentences according to their semantics, each of which was assigned 2 class labels (1: similar, 0: dissimilar). An example of our dataset is presented in table 4.

Table 4. An example of the Paired Sentence Dataset.

| Sentence 1 | Sentence 2 | Label |
|------------|------------|-------|
| My apartment is no person. | My apartment is no person. | 1 |
| My apartment is no person. | My apartment is no person. | 0 |
| My apartment is no person. | My apartment is no person. | 1 |
| My apartment is no person. | My apartment is no person. | 0 |

The number of paired sentences is 339,985 pairs that are used as an original dataset. We then expand this dataset to create an augmented dataset by using the proposed algorithm in section 4.1. As a result, the number of paired sentences in the augmented dataset will expand from 339,985 to 1,329,197 pairs.

5. Result
5.1. Datasets
Two datasets are used in our experiments:
1. An Original Dataset consists of 339,985 paired sentences. A set of 310,866 pairs was used for training, a set of 13,366 pairs was used for validation and a set of 15,753 pairs for testing.

2. An Augmented Dataset with Text Data-Augmentation consists of 1,329,197 paired sentences. A set of 1,300,078 pairs was used for the Text Data-Augmentation and a set of 13,366 pairs was used for validation and a set of 15,753 pairs for testing.

5.2. Evaluation

The goal of this data augmentation focuses on the improvement of using small data to train Siamese Long Short-Term Memory model. To test the performance of a model, the similarity of sentences is predicted using the model. The evaluation metrics are precision, recall and F1-score.

5.3. Experiments

To evaluate the performance of the Text Data-Augmentation, we compare against our paired data by using an enhancement result of the Manhattan LSTM model. We split the dataset into 70%, 15% and 15% for training, validation, and testing, respectively. For the splitting dataset, we use a blind test method for training, validating, and testing to prevent duplicated sentences between all training dataset. Thence, we carry out training for 50 epochs with early stopping rate by monitoring a minimum change of validation loss = 0.0002 and using the number of epochs with no improvement as 3. We then choose the model with the best validation performance to be evaluated on the test set. The results in table 5 show that the Manhattan LSTM model using the Augmented Data as the dataset is more effective than that of using the Original Data.

Table 5. The Manhattan LSTM model performance comparison of different datasets.

| Dataset         | Precision | Recall  | F1-score | Similarity Threshold |
|-----------------|-----------|---------|----------|----------------------|
| Original Data   | 0.061     | 0.086   | 0.071    | 0.094                |
| Augmented Data  | 0.098     | 0.079   | 0.088    | 0.141                |

We present the values of precision, recall, and F1-score using of the similarity threshold for original data and augmented data in figure 4 and figure 5, sequentially. Considering the F1-score as our reference metric, we observe from figure 4 and figure 5 that the similarity threshold values of 0.094 and 0.141 produces the best F1-score for the original data and the augmented data, respectively. Figure 6 displays the precision-recall curves and we investigate that the augmented data has an area under curve = 0.056 and the original data has an area under curve = 0.040. As the area under curve indicates that the augmented data has better precision value and recall value which are a consequence that F1-score of the augmented data performs better than the original data.

Figure 4. Accuracy, recall, precision and F1-score varying the similarity threshold of original Data.

Figure 5. Accuracy, recall, precision and F1-score varying the similarity threshold of Augmented Data.

Figure 6. Precision-recall curves.

Figure 7. Receiver operating characteristic curves.
In section 4.2, our data was assigned 2 class labels (1: similar, 0: dissimilar). The result in table 6 show that the Manhattan LSTM model predicted the positive classes and negative classes between test dataset of the original data and the augmented data by using the threshold values as 0.094 and 0.141 for original data and augmented data, respectively.

**Table 6. Confusion matrix of the classification for the semantic sentence.**

| Dataset          | True positive | True negative | False positive | False negative |
|------------------|---------------|---------------|----------------|----------------|
| Original Data    | 39            | 14694         | 605            | 415            |
| Augmented Data   | 36            | 14966         | 333            | 418            |

In figure 7, we report the receiver operating characteristic curves (ROC) of the original data and the augmented data for evaluation to see if the augmented dataset increases the performance of the Manhattan LSTM model. We investigate the area under curve of the original data and the augmented data which are 0.517 and 0.575, respectively. Therefore, the area under curve in figure 7 shows that the Manhattan LSTM model which uses our proposed method as a training data performs better than that of the original data.

6. Conclusion
We proposed an algorithm for the Text Data-Augmentation that uses word similarity instead of the synonym word replacement. As a result, we have the augmented dataset for training that has diversify of words which are close to the original words. In the evaluation, the Text Data-Augmentation has been shown to be an enhancement on text classification tasks for the Manhattan LSTM model when training on the augmented dataset. For future work, we plan to develop our augmentation model and add the possibility to implement this method for Question Retrieval application like Chatbot by using a small dataset.

Acknowledgments
The authors gratefully thank Thanatwit Angsarawanee, Ramon Robloke and Natthika Srikiattiwong for supporting and insightful feedback.

References
[1] *twilio*, How Consumers Use Messaging Today, https://www.twilio.com/learn/commerce-communications/how-consumers-use-messaging
[2] Wikipedia, Chatbot, https://en.wikipedia.org/wiki/Chatbot
[3] J. Mueller and A. Thyagarajan. 2016. *Siamese recurrent architectures for learning sentence similarity*. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, AAAI Press: 2786–92.
[4] N. Othman, et al. 2019. *Manhattan Siamese LSTM for Question Retrieval in Community Question Answering*. Cham, Springer International Publishing.
[5] A. Mosolova, V. Fomin, and I. Bondarenko. 2018. *Text Augmentation for Neural Networks*, In Supplementary Proceedings of the Seventh International Conference on Analysis of Images, Social Networks and Texts, Moscow, pp. 104-9
[6] J. Wei and K. Zou. 2019. *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks*, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, pp. 6382–88.