ICIC Express Letters ICIC International ©2023 ISSN 1881-803X
Volume **17**, Number **3**, March **2023** pp. **327–338**

NOVEL APPROACH TO PALI SAMAS SEGMENTATION USING BIDIRECTIONAL LONG SHORT-TERM MEMORY AND RULE-BASED ANALYSIS

Klangjai Tammanam, Sajjaporn Waijanya* and Nuttachot Promrit*

Center of Excellence in AI and NLP
Department of Computing
Faculty of Science
Silpakorn University
6, Rajamankha Nai Road, Muang District, Nakhon Pathom 73000, Thailand
tammanam_k@sikpakorn.edu; *Corresponding authors: { waijanya_s; promrit_n }@silpakorn.edu

Received June 2022; accepted August 2022

Abstract. *Pali Samas words cannot be found in any dictionary. They are created by placing Pali words that contain meaning after one another without changing their morphemes or with changes in morphemes and pronunciation. Thus, Pali Samas words need to be segmented back into their previously original words to obtain their meanings. This research presents a novel approach to Pali Samas segmentation using bidirectional long short-term memory to predict the splitting locations and applies the rules obtained from Samas word segmentation to achieving correct meanings. For the dataset used in this research, a total of 2,757 Thai Pali Samas words are used to further create 4,478 Samas words through text augmentation. The results from the Samas word segmentation indicate that the prediction for splitting locations has a weighted average of F1-score of 99.20%, with 81.91% of the original words derived from reverse segmentation based on the rules.*

Keywords: Bidirectional long short-term memory, Samas, Compound word, Pali Thai, Rule base, Compound splitting

1. **Introduction.** Pali and Sanskrit are inflected languages, so letters at the end of words indicate their functions and meanings. New Samas words are created by joining root words before adding a new suffix. They are created by placing Pali words that contain meaning after one another without changing their morphemes or with changes in morphemes and pronunciation. Another challenging point in creating Pali and Sanskrit Samas words is that they are not only generated by the conjugation of basic words but also derived from the conjugation between basic and Samas words or Samas and Samas words into a new word. The newly generated Samas words have not been listed in any Pali or Sanskrit dictionary.

This study points out a segmentation method of Thai Pali Samas words which can split and convert such words into root words using the bidirectional long short-term memory (BiLSTM) technique in conjunction with word transformation rules. Because no research has presented any machine learning technique for Thai Pali Samas segmentation, to the best of the authors' knowledge, this is the first study to introduce Thai Pali Samas segmentation. Although the structures and characteristics of Pali are very similar to those of Sanskrit, the research on Pali segmentation is rarely found. Sanskrit is still taught in India, whereas Pali is still only used in many Theravada countries to disseminate Buddhist teachings. Accordingly, Sanskrit research can develop in the Pali research field.

Regarding Pali and Sanskrit research, foreign researchers have developed Sanskrit language processes in various fields, such as sentence segmentation [1], database creation [2,3],

DOI: 10.24507/icicel.17.03.327

information retrieval [4], word form analysis [5], parts of speech categorization and function identification [6,7], and Sandhi word creation and segmentation [7-11]. Research on Samas word categorizations has been performed [12,13]. For instance, a research focused on detecting whether an input word is Samas or not [14]. Apart from Samas word detection and type classification in the Sanskrit language, Samas word segmentation [15,16] is used to split Samas words, predict their type, and find their meaning.

At present, neural networks are often applied in natural language processes, such as Sanskrit Sandhi segmentation models applying convolutional neural networks [10] and recurrent neural networks [7,8]. The studies on natural language processes with Thai language datasets are very challenging for researchers. Although there have been continuous attempts to study this field, works on Thai language processes still have low-resource datasets, especially those in special fields, such as Thai languages in prose poetry that requires BiLSTM to extract entities from Sunthon Phu's poem plays [17] or the use of the machine learning technique to analyze the melodiousness patterns of Phra Aphai Manee narrative poems [18]. Regarding other fields, such as the natural language process in the form of Thai Pali letters, some remarkable studies have been performed, such as the study on the machine translation system [19] and conversion of the Pali language into phonemes [20]. In particular, our former work focused on the Sandhi Samas segmentation model in the Thai Pali language [21] via BiLSTM for predicting the locations and patterns of word transformation and applying the analyzed rules to correct the wrong words from the Sandhi segmentation. However, it still does not cover Samas words in the Pali language because the segmentation and rule-based analysis in [21] can segment accurately and meaningfully words. Therefore, this study presents Thai Pali Samas word segmentation methods, which can segment and back-transform words into their original forms for a more perfect word segmentation process in the future.

2. Methodology.

This research uses 2,757 Thai Pali Samas words from eight Dhammapada Atthakatha books in conjunction with the dataset preparation and validation by an expert. All the aforementioned words are used as the database in this research for studying Thai Pali segmentation. The overview of the research methodologies is shown in Figure 1.

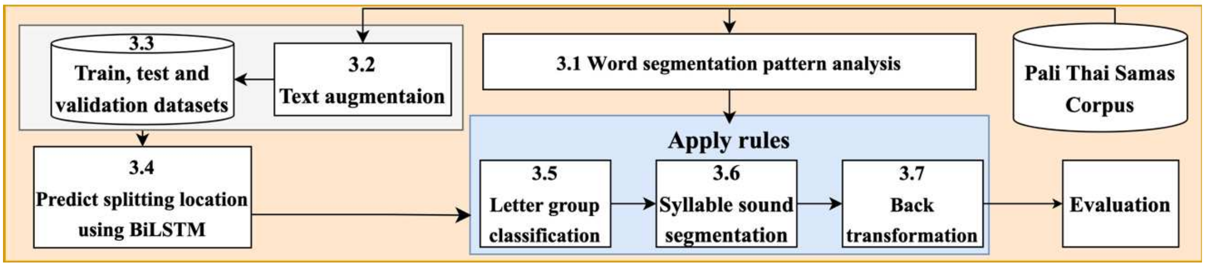

3.3
Train, test and validation datasets
3.2
Text augmentaion
3.1 Word segmentation pattern analysis
Pali Thai Samas Corpus
Apply rules
3.4
Predict splitting location using BiLSTM
3.5
Letter group classification
3.6
Syllable sound segmentation
3.7
Back transformation
Evaluation

FIGURE 1. Research overview

2.1. Samas word segmentation pattern analysis.

One of the single longest words in the dataset of 2,757 different Samas words compiled by the expert can be divided into seven words at maximum as shown in Table 1. Based on the analysis of Samas word segmentation patterns, there are four types of Samas words.

2.1.1. *Rule 1 pattern.* The Samas words are segmented into two roots, which are correct and have meaning and equal numbers of letters before and after segmentation, as shown in Figure 2.

2.1.2. *Rule 2 pattern.* The Samas words are segmented into two roots, which are correct and meaningful, but some letters are removed as shown in Figure 3.

Table 1. Example of Samas and target words

Samas word	Target words						
	1st	2nd	3rd	4th	5th	6th	7th
ชาติกุลปฺปเทสโภคยสวยสมฺปตฺติ (jātikulappadesabhogayasavayasampatti)	ชาติ jāti	กุล kula	ปเทส padesa	โภค bhoga	ยส yasa	วย vaya	สมฺปตฺติ sampatti
ธมฺมกถาทิ (dhammakathādi)	ธมฺม dhamma	กถา kathā	อาทิ ādi				

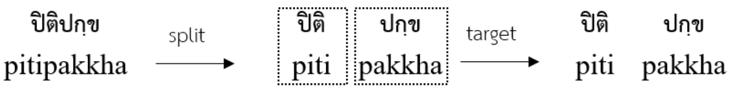


Figure 2. Rule 1 pattern

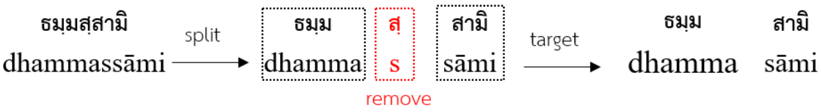


Figure 3. Rule 2 pattern

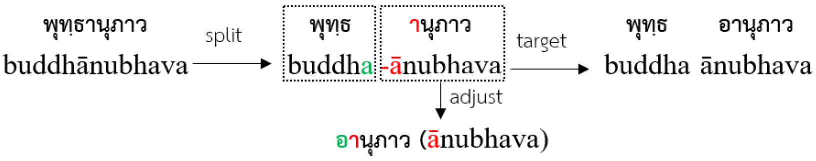


Figure 4. Rule 3 pattern

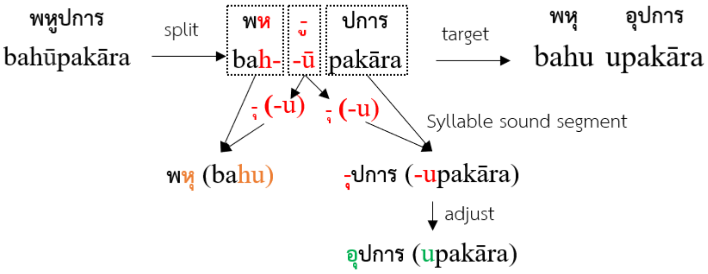


Figure 5. Rule 4 pattern

2.1.3. *Rule 3 pattern.* The Samas words are segmented into two roots. The first root is correct and meaningful, but the second root needs correction as shown in Figure 4.

2.1.4. *Rule 4 pattern.* The Samas words are segmented into more than two groups of letters, and none are correctly identified as the targeted results because there is a link between the vowel sound of the first root's last syllable and the first syllable of the latter root, according to the linking sound principle of the Pali language. Thus, the groups of letters can link the sound between the syllables and convert them back to their original forms as shown in Figure 5.

2.2. **Text augmentation.** The dataset of all Samas words compiled by the expert includes 2,757 words, which is quite sparse. Thus, there is a concept to create more Samas words. The syllable sound link between the last syllable of the former root and the first

TABLE 2. Link of the syllable sounds between two connected words into one syllable

		The vowel sound of the latter root's first syllable.			
		อ (a) (vowel forms)	อา (ā)	อุ (u)	โอ (o)
The vowel sound of the former root's last syllable.	อ (a) (consonant forms)	อา (ā)	อา (ā)	อุ (u), โอ (o)	โอ (o)
	อา (ā)	อา (ā)	อา (ā)	อา (ā), อู (ū)	-
	อุ (u)	อา (ā), อู (ū)	อา (ā), อู (ū)	อู (ū), โอ (o)	โอ (o)

syllable of the latter root, such as อ + อา (a + ā), will be converted into อา (ā) sound when a clash of the syllables' sounds occurs, as shown in Table 2.

This research creates extra Samas word data with the consideration of the possible sound link using 869 Samas words (words that change their letters after being joined by the root words) to create new Samas words. The method used is crossover augmentation [22].

Figure 6 illustrates examples of Samas word formation. The first Samas word, ราชาณา (rājāṇā), can be segmented into ราชา (rājā) and อาณา (āṇā), whereas the second Samas word, เทสนาวสาน (desanāvasāna), can be segmented into เทสนา (desanā) and อวสาน (avasāna). Accordingly, the two roots can generate new Samas words, namely, ราชาวสาน (rājāvasāna) and เทสนาณา (desanāṇā), respectively.

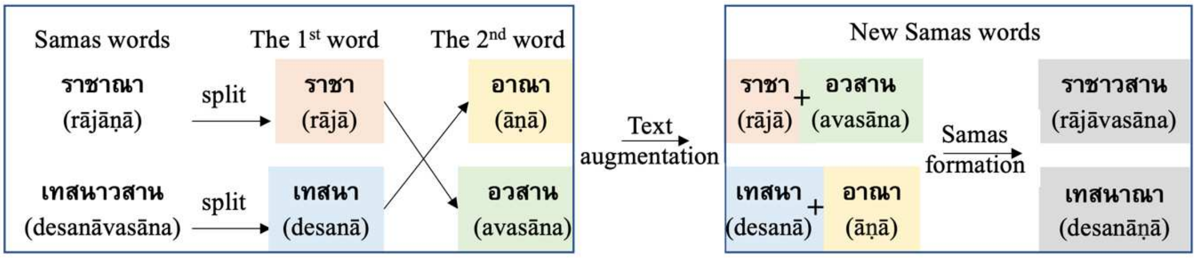
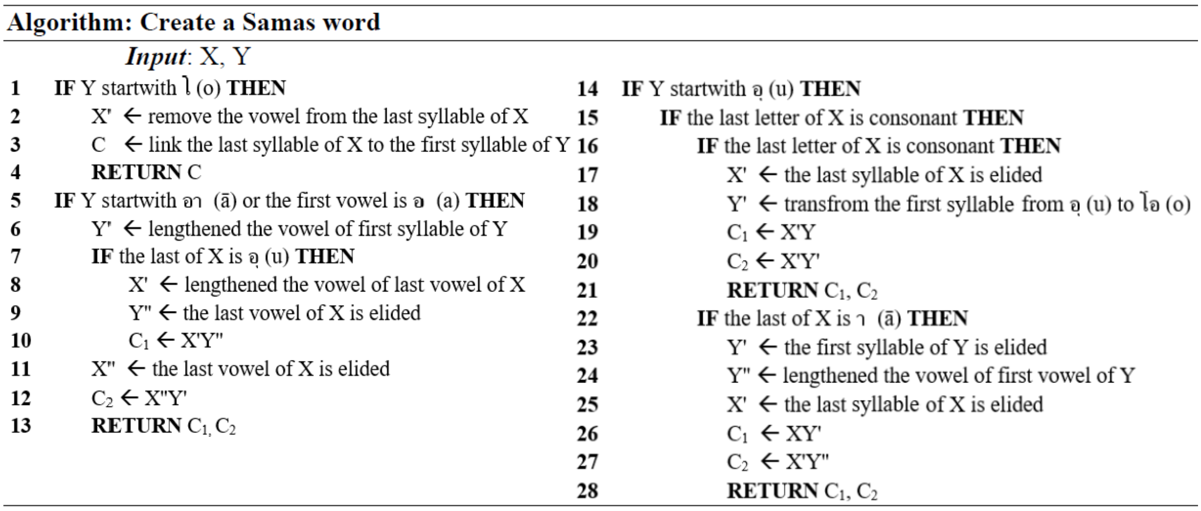


FIGURE 6. Example of a new Samas word creation

In this process, 869 Samas words changed the letters after being joined by the root words. Because these Samas words are divided into two words, they can form 1,720 new Samas words. The Samas text augmentation algorithm to create new words is shown below.



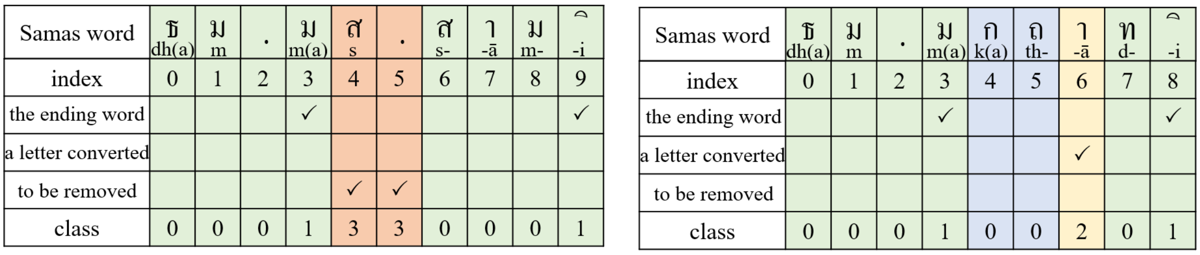
Algorithm: Create a Samas word

```
Input: X, Y
IF Y startwith โ (o) THEN
    X' ← remove the vowel from the last syllable of X
    C ← link the last syllable of X to the first syllable of Y
    RETURN C
IF Y startwith อา (ā) or the first vowel is อ (a) THEN
    Y' ← lengthened the vowel of first syllable of Y
    IF the last of X is อุ (u) THEN
        X' ← lengthened the vowel of last vowel of X
        Y" ← the last vowel of X is elided
        C1 ← X'Y"
    X" ← the last vowel of X is elided
    C2 ← X"Y'
    RETURN C1, C2
IF Y startwith อุ (u) THEN
    IF the last letter of X is consonant THEN
        IF the last letter of X is consonant THEN
            X' ← the last syllable of X is elided
            Y' ← transfrom the first syllable from อุ (u) to โอ (o)
            C1 ← X'Y
            C2 ← X'Y'
            RETURN C1, C2
        IF the last of X is า (ā) THEN
            Y' ← the first syllable of Y is elided
            Y" ← lengthened the vowel of first vowel of Y
            X' ← the last syllable of X is elided
            C1 ← XY'
            C2 ← X'Y"
            RETURN C1, C2
```

Regarding the creation of extra 1,720 Samas words as mentioned in this section, the data of class 0's segmentation location prediction increase by 12,029 from the original amount of 22,370 words. Hence, the total data in class 0 are 34,399 words. Class 1 increases by 2,171 words from 6,176 words, totaling 8,347 words. Class 2 increases by 2,674 words from 103 words, totaling 2,777 words. Class 3 increases by 0 word from 449 words, totaling to 449 words. Class 4 increases by 41,606 words from 64,674 words, totaling to 106,280 words.

2.3. **Samas segmentation location prediction.** A Samas word $\boldsymbol{S}$ has a total number of letters $|\boldsymbol{S}|$ and any letters are $\boldsymbol{S}_i$. There are five classes of the possible segmentation location, including class 0 = not a segmented location; class 1 = the location of the ending word, which can be a vowel or consonant; class 2 = the location of a letter converted back to its original form; class 3 = the letter that needs to be removed; and class 4 = the location for expansion.

Figure 7(a) portrays a sample data preparation for predicting the segmentation location of the Samas word ธมฺมสฺสามิ (dhammassāmi), which has letter สฺ (s) that must be removed. Figure 7(b) shows a sample data preparation for predicting the segmentation location of the Samas word ธมฺมกถาทิ (dhammakathādi).



Samas word	ธ dh(a)	ม m	ฺ	ม m(a)	ส s	ฺ	ส s-	า -ā	ม m-	ิ -i
index	0	1	2	3	4	5	6	7	8	9
the ending word				✓						✓
a letter converted										
to be removed					✓	✓				
class	0	0	0	1	3	3	0	0	0	1

(a) Letters needing to be removed

Samas word	ธ dh(a)	ม m	ฺ	ม m(a)	ก k(a)	ถ th-	า -ā	ท d-	ิ -i
index	0	1	2	3	4	5	6	7	8
the ending word				✓					✓
a letter converted							✓		
to be removed									
class	0	0	0	1	0	0	2	0	1

(b) Letters needing to be segmented in terms of the sound

FIGURE 7. Sample data preparation for the prediction

2.4. **Segmentation location prediction model.** After the data augmentation, the Samas words' length is expanded to 34 letters, which is equal to the longest Samas word. The words are encoded by converting them into integers, as mentioned in Section 2.3, containing 43 letters used in the Thai Pali language. The integers are then converted into a one-hot vector and inputted into a BiLSTM model to predict the segmentation locations. The details and prediction model structure are shown in Figure 8.

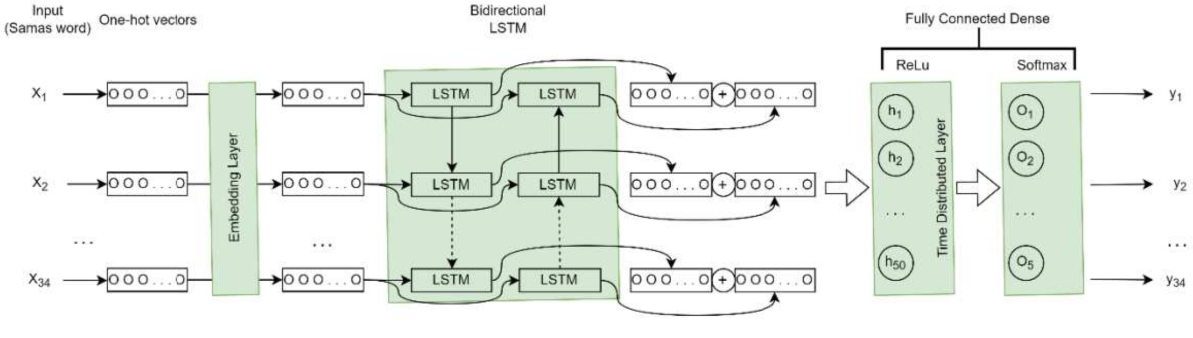


FIGURE 8. Samas word segmentation location prediction model

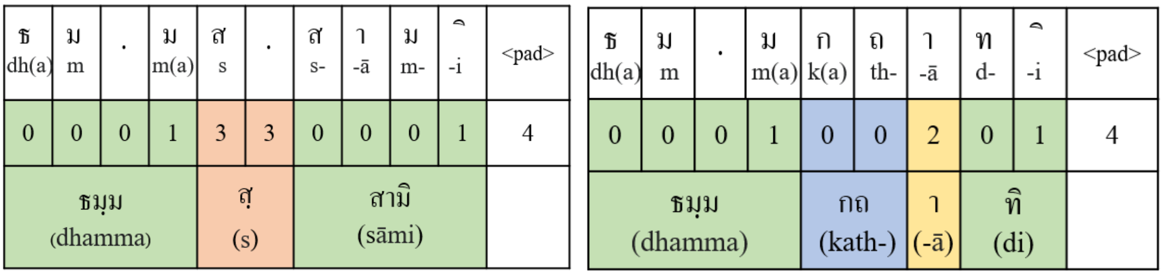
There are 34 steps, which is equal to the longest Samas words. The embedding size is 100, and there are 100 + 100 nodes in the BiLSTM. The hidden layer consists of 50 nodes and uses a recurrent dropout of 0.5 for the rectified linear unit function. Then the data are transferred to the output layer consisting of five nodes. The activation function is softmax, the model is trained by the Adam optimizer with 1,000 epochs, and the learning ratio starts from 0.001 with a batch size of 32.

2.5. **Letter group classification.** After Samas words are inputted and their letters are predicted into the segmentation locations by the model, they will be classified in groups to consider the spacing, displaying, and syllable segmentation based on the rules. Accordingly, there are four groups of letters presented as follows.

- *The group with ending letters and can be written by a regular expression, i.e.,* 0*1: There are letters predicted into Class 0 equal or up to 0, and they must be followed by the letter predicted as Class 1. Accordingly, these letters will be followed by the space mark when displayed.
- *The group without ending letters and can be written by a regular expression, i.e.,* 0^{+}[^23]: There is at least one letter predicted as Class 0, and the latter letters do not have the letters predicted as Class 2 or 3. In other words, this group of letters is only focused on the letters predicted as Class 0 – without the letters predicted as Class 2 or 3. These letters will not be followed by a space mark when displayed.
- *The group of letters that require sound segmentation and can be written by a regular expression, i.e.,* 2^{+}: There is at least one letter consecutively predicted as Class 2 twice. If the segmentation location prediction detects these Samas letters, the sound segmentation rules will be applied.
- *The group of letters that must be removed and can be written by a regular expression, i.e.,* 3^{+}: There is at least one letter consecutively predicted as Class 3 twice, and this group will not be displayed.

As shown in Figure 7(a), the Samas word "ธมฺมสฺสามิ" (dhammassāmi) that passes the segmentation location prediction will be classified into the groups of letters by regular expressions. They are classified into three groups, namely "ธมฺม สฺ สามิ" (dhamma s sāmi), but the "สฺ" (s) group needs to be removed. Therefore, the final result of the letter group classification process is "ธมฺม สามิ" (dhamma sāmi) as shown in Figure 9(a).

In Figure 7(b), the Samas word "ธมฺมกถาทิ" (dhammakathādi) that has been predicted is classified into four groups of letters, i.e., "ธมฺม กถ า ทิ" (dhamma kath-ā di), as shown in Figure 9(b). The third group only has one vowel "า" (-ā), which is predicted as a letter requiring the syllable sound segmentation rule for segmentation into two sounds (the last syllable of the letters "กถ" (kath-) and the first syllable of the letters "ทิ" (di)).



ธ dh(a)	ม m	.	ม m(a)	ส s	.	ส s-	า -ā	ม m-	ิ -i	<pad>
0	0	0	1	3	3	0	0	0	1	4
ธมฺม (dhamma)				สฺ (s)		สามิ (sāmi)				

(a) Letters that need to be removed

ธ dh(a)	ม m	.	ม m(a)	ก k(a)	ถ th-	า -ā	ท d-	ิ -i	<pad>
0	0	0	1	0	0	2	0	1	4
ธมฺม (dhamma)				กถ (kath-)		า (-ā)	ทิ (di)		

(b) Letters that require sound splitting

FIGURE 9. Example of letter group classification

2.6. **Syllable sound segmentation.** Syllable sound segmentation is applied to determining the sound of the link syllables. The sound of the two syllables is linked into one syllable. This process will be implemented when there is a group of letters' sounds that needs to be segmented, as shown in Figure 10.

To present the syllable sound segmentation rules of Figures 11 and 12 in the most understandable way, all groups of letters will be represented by colors: blue (dotted

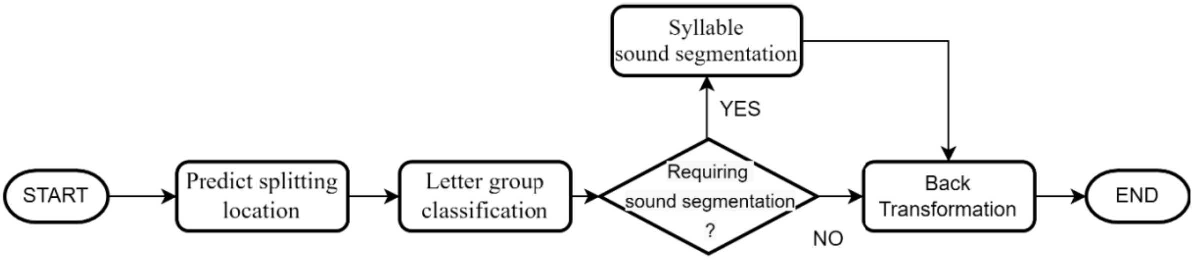

START
Predict splitting location
Letter group classification
Requiring sound segmentation ?
YES
Syllable sound segmentation
NO
Back Transformation
END

Figure 10. Samas word segmentation processes

border) represents the group without ending letters, yellow (no border outline) represents the group needing syllable sound segmentation, and green (bold border) represents the group with ending letters.

2.6.1. *1st rule for syllable sound segmentation.* The rules for the group of letters that consist of an ending letter at the beginning and the group of letters that consists of an ending letter at the beginning (Figure 11) have four criteria.

- If “โ-” (-o vowel) is in a vowel form and followed by a consonant (X represents a consonant), segment it into syllable sounds “โอ” (o vowel) consonant or “อุ” (u vowel) consonant by referencing from Thai Pali dictionaries.
- If it is “-ู” (-ū vowel), segment it into “า อุ” (-ā u).
- If it is “-ุ” (-u vowel), segment it into “า อุ” (-ā u).
- If it is “-า” (-ā vowel), segment it into the syllable sound “า อุ” (-ā u) or “า อา” (-ā ā) by referencing from Thai Pali dictionaries.

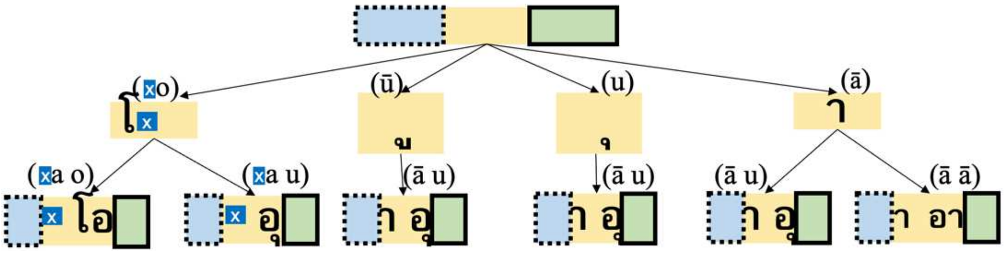


Figure 11. (color online) Rule for the group of syllable sound-segmented letters consisting of the ending letters at the beginning

2.6.2. *2nd rule for syllable sound segmentation.* The syllable sound segmentation rules for the group of letters beginning with the ending letters and the group of letters beginning with the ending letters are shown in Figure 12. To elaborate, the group segmentation syllable sound of “า” (-ā) vowel is segmented into the syllable sound “อ อ” (-a a) or “อ อา” (-a ā) by referring to the dictionaries. Because the syllable sound of the two conditions is อ (a) vowel sound, it does not have any initial consonant. When there is an initial consonant sound, a vowel form does not occur. Thus, this rule is determined to segment the “อ” (a) or “อา” (ā) sound according to the writing system.

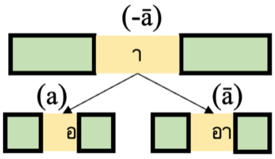


Figure 12. (color online) Rule for the group of letters beginning with the ending letters

2.7. **Back-transformation.** The letter group classification process considers that there is no group requiring sound segmentation or undergoing syllable sound segmentation for the letter groups needing to be segmented. Then, the groups of letters will be referred to the conditions presented in Table 3 to transform the words back to their original forms before changing them for the Samas language process.

TABLE 3. Back-transformation rules

Conditions	Processes
If มห (maha)	transform to มหนฺต (mahanta)
If อน (ana)	transform to น (na)
If อ (a)	transform to น (na)
If ส (sa)	transform to สห (saha)
Begin with นุ (nu) ภิ (bhi) ติ (ti) ป (pa) or น (na)	add อ (a) before the letters
Begin with vowel form except เ- (e-) or โ- (o-)	add อ (a) before the letters
The second letter is Bindu (a dot under a letter)	add อ (a) before the letters

Figure 13 presents the example of the Samas segmentation of "สปริวาร" (saparivāra) segmented into "ส ปริวาร" (sa parivāra), which is not part of the letter groups requiring syllable sound segmentation. Hence, each group is checked with the word transformation conditions, and "ส" (sa) should be corrected into "สห" (saha). Therefore, the final result is "สห ปริวาร" (saha parivāra).

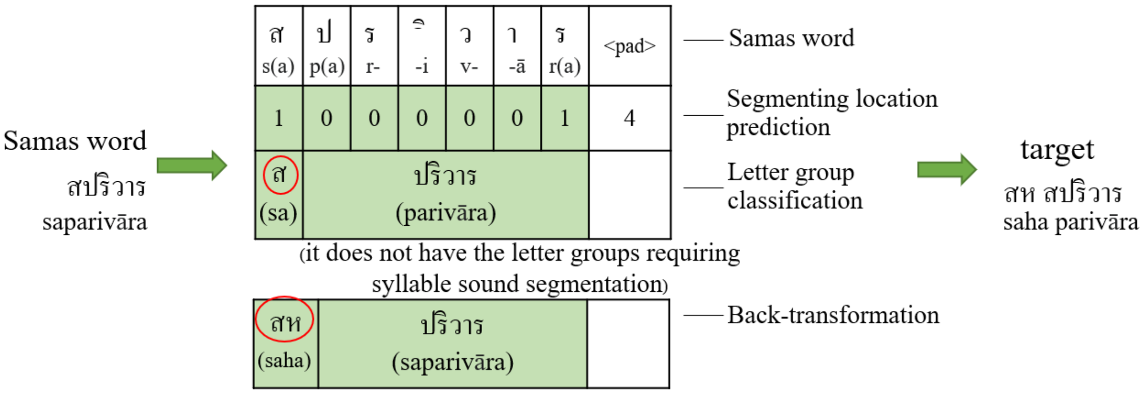
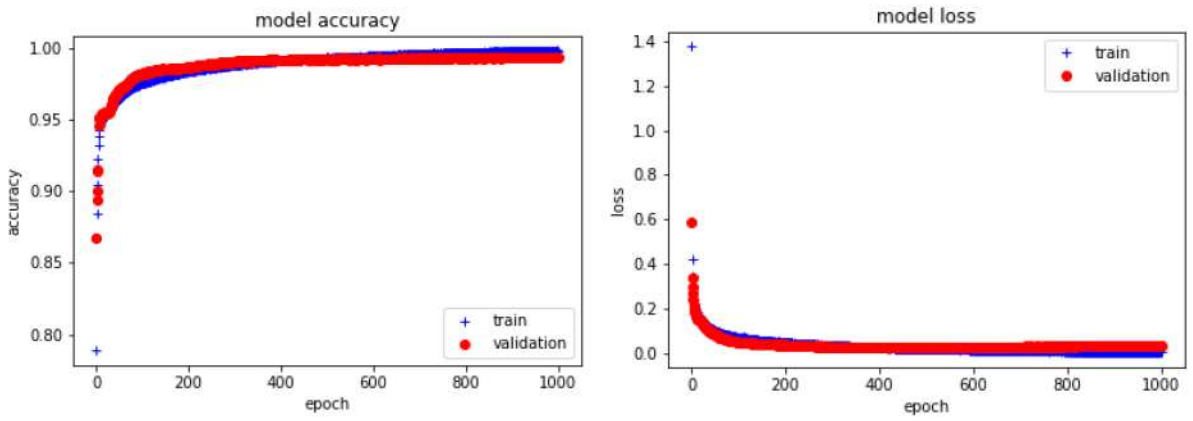


FIGURE 13. Example of Samas segmentation on the group of letters requiring back-transformation

3. Experimental Results.

The Pali Samas segmentation method presented in this research was first performed by finding the segmentation locations before classifying the words into groups of letters. If there are letters requiring syllable sound segmentation, they will be segmented in accordance with the rules. Then, all the letter groups will be referred to the back-transformation conditions to their original forms.

3.1. **Samas segmentation location prediction model.** There are 4,478 Samas words, divided into 3,223 words for the training dataset, 359 for the validation dataset, and 896 words for the test dataset. Moreover, Samas words are expanded to have the same length as the maximum length of the discovered word, which is 34 letters. Therefore, there are 3,223 × 34 = 109,582 locations for the training data, 359 × 34 = 12,206 locations for the validation dataset, and 896 × 34 = 30,464 locations for the test dataset.



(a) Accuracy (b) Loss

Figure 14. Results of the segmentation location prediction evaluation

The results of the model efficiency evaluation indicate that the accuracy values of the training and validation datasets are 0.9921 and 0.9879, respectively, as shown in Figure 14(a). The loss values of the training and validation datasets are 0.0079 and 0.0073, respectively, as shown in Figure 14(b).

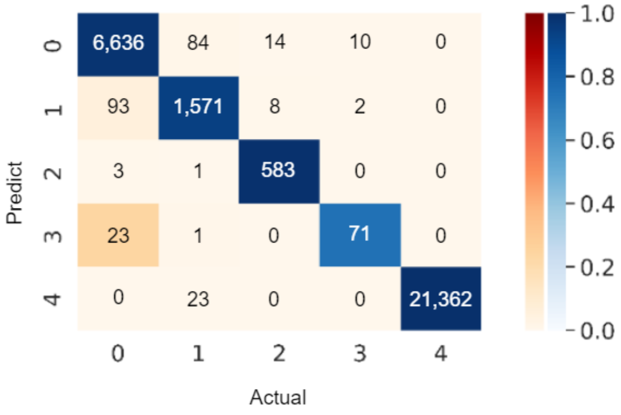
The graphs of Figures 14(a) and 14(b) illustrate few distances and good-fit learning potentials. This finding indicates that the model can learn well in the training process and accurately predict the validation dataset, which has not been attempted before.

3.2. **Samas word segmentation efficiency.** The efficiency of the Samas word segmentation method can be evaluated by two steps.

3.2.1. *Efficiency of the Samas word segmentation location prediction model.* The model can be evaluated through a confusion matrix, precision, recall, and F1-score using the test dataset of 896 Samas words. Each word is expanded to have 34 letters, so all the data for the test will be 896 × 34 = 30,464 locations.

Figure 15 presents the confusion matrix of the prediction model. The columns portray the real answers, and the rows portray the model's answers. The efficiency of the segmentation location prediction model can be evaluated by the precision recall and F1-score, as shown in Table 4.

In Table 4, there are 30,464 locations of Samas words used as the dataset of the segmentation locations, and the amount of data in Classes 0-4 are 6,744, 1,674, 587, 95,



Predict	0	1	2	3	4
0	6,636	84	14	10	0
1	93	1,571	8	2	0
2	3	1	583	0	0
3	23	1	0	71	0
4	0	23	0	0	21,362

Figure 15. (color online) Confusion matrix

Table 4. Precision, recall, and F1-score of the Samas words' segmentation location

Class	Precision	Recall	F1-score	The amount of data
0	0.9824	0.9840	0.9832	6,744
1	0.9470	0.9385	0.9427	1,674
2	0.9636	0.9932	0.9782	587
3	0.8554	0.7474	0.7978	95
4	1.0000	0.9999	1.0000	21,364
Micro-average	0.9497	0.9326	0.9404	30,464
Weighted-average	0.9920	0.9921	0.9920	30,464

and 21,364, respectively. Class 4 includes data the most because it is the post-padding stage that expands all Samas words' length equally. Class 1 has locations distancing from Classes 2 and 3 because it does not include the Samas segmentation letters. Moreover, the Samas segmentation letters are divided into Classes 1-3, which have ending letters that may be a consonant or vowel, letters that require syllable sound segmentation, and letters that need to be removed. Thus, the number of data is very different.

Table 5 presents a comparison between the prediction results of the model through the micro-average and weighted-average. The findings indicate that the F1-score before text augmentation provides less values, which means less accuracy in the segmentation location prediction.

Table 5. Comparison between the average values of the precision, recall, and F1-score before and after text augmentation

		Precision	Recall	F1-score
Before text augmentation	Micro-average	0.8509	0.7080	0.7270
	Weighted-average	0.9873	0.9876	0.9872
After text augmentation	**Micro-average**	**0.9497**	**0.9326**	**0.9404**
	Weighted-average	**0.9920**	**0.9921**	**0.9920**

3.2.2. *Thai Pali segmentation efficiency.* Regarding the evaluation presented in Section 3.2.1, the model can predict segmentation locations in the 896 words of the validation dataset, i.e., 896 × 34 = 30,464 locations and obtains high precision. However, the Samas words used for the test in this study generate more than two words after segmentation. A wrong prediction can affect the segmentation. Therefore, if there is a wrong location, the prediction will be considered failure. From the test, the model can correctly predict 749 of 896 Samas words and segment all of the 749 words correctly.

Table 6 points out a comparison between the splitting results of the purpose model, which indicates a percent of correctly predicted Samas segmentation locations before the text augmentation provides more values. This finding means a high accuracy of the proposed segmentation method.

3.3. **Segmentation evaluation by words that do not occur in the Thai Pali Samas corpus.** The Samas dataset is divided into the training, validation, and test datasets, which are used after the analysis of the Samas segmentation patterns, including the rules for adjusting the segmentation. We interview the expert about the rules for segmentation, questioning whether they are designed to adjust the Samas words in the dataset or not. Hence, we choose 200 new Samas words from the Magalatthadipani book and then select only the words that have never appeared in any of our dataset. Consequently, we obtain additional 124 words for the test with which the model can correctly segment 99 words (79.83%).

Table 6. Accuracy of the proposed segmentation method

	Correctly predicted Samas segmenting locations		Wrongly predicted segmenting locations
	Correct segmentation	Incorrect segmentation	
Before text augmentation	399 words (72.28%)	2 words (0.36%)	151 words (27.36%)
After text augmentation	**734 words (81.91%)**	**15 words (1.68%)**	**147 words (16.41%)**

The sample results of our purpose are as follows: the Samas word มหาสกุณสงฺฆ (mahāsakuna saṅgha) may be divided into two implications, including three words, มหา-สกุณ-สงฺฆ (mahā sakuna saṅgha), which means the flock of big birds' flock, or two words, including มหาสกุณ-สงฺฆ (mahāsakuna saṅgha), which means the flock of big birds. However, the objective of the segmentation proposed in this research is to segment Samas words that can be searched in the dictionary. Therefore, the presented segmentation result of the word มหาสกุณสงฺฆ (mahāsakunasaṅgha) as มหา-สกุณสงฺฆ (mahāsakuna saṅgha) is considered a wrong segmentation. The other Samas word ทฺวาทสหตฺถ (davādasahattha) has the correct segmentation location prediction, but its back-transformation is incorrect.

4. **Conclusions and Future Works.** This research presents a Thai Pali Samas word segmentation method by applying BiLSTM and the rules derived from the Samas segmentation pattern analysis. The dataset used includes 2,757 Samas words compiled by a Pali language expert. The researchers also created additional 4,478 Samas words using the text augmentation method mentioned in Section 2.2. The segmentation location prediction evaluation is implemented using the validation dataset with the precision, recall, and F1-score. The accuracy is 99.21%. However, considering the correct prediction, the correct words are 83.05%, and the percentage drops to 81.91% after applying this result with the rules. Ultimately, after we have implemented the further test, the complete segmentation on the entire words has an accuracy of 79.83%. For future work, the identification type of Thai Pali words as Samas, Sandhi or normal word will be researched and developed. The automatic identification of a word's type before passing it to the segmentation process will be useful for creating tools for Thai Pali translation. The innovation from the integration of computational linguistics and Buddhism will be an encouragement for understanding Buddhist principles.

Acknowledgment. We are wholeheartedly grateful for the support of Phramaha Chatchai Thawaro (Chatchai Apichon), an 8-level Pali scholar from Phra Du temple, and Samutsakorn province, who generously devoted his precious time to compiling the dataset and help in the validation.

REFERENCES

[1] O. Hellwig, Detecting sentence boundaries in Sanskrit texts, *The 26th International Conference on Computational Linguistics*, Osaka, Japan, 2016.

[2] S. Bhardwaj, N. Gantayat, N. Chaturvedi, R. Garg and S. Agarwal, SandhiKosh: A benchmark corpus for evaluating Sanskrit Sandhi tools, *Proc. of the 11th International Conference on Language Resources and Evaluation (LREC2018)*, 2018.

[3] T. Neill, LDA topic modeling for pramāṇa texts: A case study in Sanskrit NLP corpus building, *Association for Computational Linguistics*, 2019.

[4] M. Meyer, On Sanskrit and information retrieval, *Association for Computational Linguistics*, 2019.

[5] D. Alfter, *Morphological Analyzer and Generator for Pali*, Bachelor Thesis, University of Trier, 2014.

[6] A. Natarajan and E. Charniak, S^3 – Statistical Sandhi splitting, *Proc. of the 5th International Joint Conference on Natural Language Processing*, 2011.

[7] R. Aralikatte, N. Gantayat, N. Panwar, A. Sankaran and S. Mani, Sanskrit Sandhi splitting using $seq2(seq)^2$, *The 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.

[8] O. Hellwig, Using recurrent neural networks for joint compound splitting and Sandhi resolution in Sanskrit, *The 7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, Pozna, Poland, 2015.

[9] O. Hellwig, *Morphological Disambiguation of Classical Sanskrit*, Springer International Publishing, Cham, 2015.

[10] H. Oliver and N. Sebastian, Sanskrit word segmentation using character-level recurrent and convolutional neural networks, *The 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[11] S. Krishnan and A. Kulkarni, Sanskrit segmentation revisited, *arXiv.org*, arXiv: 2005.06383, 2020.

[12] A. Krishna, P. Satuluri, S. Sharma, A. Kumar and P. Goyal, Compound type identification in Sanskrit: What roles do the corpus and grammar play?, *Proc. of the 6th Workshop on South and Southeast Asian Natural Language Processing*, 2016.

[13] J. Sandhan, A. Krishna, P. Goyal and L. Behera, Revisiting the role of feature engineering for compound type identification in Sanskrit, *Proc. of the 6th International Sanskrit Computational Linguistics Symposium*, 2019.

[14] B. Premjith, C. Chandran, S. Bhat and S. Kp, A machine learning approach for identifying compound words from a Sanskrit text, *Proc. of the 6th International Sanskrit Computational Linguistics Symposium*, 2019.

[15] A. Kumar, V. Mittal and A. Kulkarni, *Sanskrit Compound Processor*, Springer, Berlin, Heidelberg, 2010.

[16] V. Mittal, Automatic Sanskrit segmentizer using finite state transducers, *Proc. of the ACL 2010 Student Research Workshop: Association for Computational Linguistics*, 2010.

[17] O. Khongtum, N. Promrit and S. Waijanya, *The Entity Recognition of Thai Poem Compose by Sunthorn Phu by Using the Bidirectional Long Short Term Memory Technique*, Springer International Publishing, Cham, 2019.

[18] P. Suksanguan, S. Waijanya and N. Promrit, The extraction of beautiful sound patterns from Sunthorn Phu's Poem using machine learning technique and internal rhyme rule, *International Journal of Advances in Intelligent Informatics*, vol.7, no.2, pp.198-210, 2021.

[19] N. Phonson, *The Rule-Based Machine Translation System from Pali to Thai*, Master Thesis, Mahidol University, Bangkok, 2001.

[20] W. Maleelai and P. Seresangtakul, Grapheme to phoneme Transcription for Pali-Thai, *KKU Science Journal*, vol.42, no.3, pp.636-645, 2022.

[21] K. Tammanam, N. Promrit and S. Waijanya, A hybrid approach to Pali Sandhi segmentation using BiLSTM and rule-based analysis, *Engineering and Applied Science Research*, vol.48, no.5, pp.614-626, 2021.

[22] F. M. Luque, Atalaya at TASS 2019: Data augmentation and robust embeddings for sentiment analysis, *arXiv.org*, arXiv: 1909.11241, 2019.