EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

A hybrid approach to Pali Sandhi segmentation using BiLSTM and rule-based analysis

Klangjai Tammanam, Nuttachot Promrit* and Sajjaporn Waijanya*

Center of Excellence in AI and NLP, Department of Computing, Faculty of Science, Silpakorn University, Nakhon Pathom 73000, Thailand

Received 8 January 2021
Revised 2 March 2021
Accepted 8 March 2021

Abstract

Pali Sandhi is a phonetic transformation from two words into a new word. The phonemes of the neighbouring words are changed and merged. Pali Sandhi word segmentation is more challenging than Thai word segmentation because Pali is a highly inflected language. This study proposes a novel approach that predicts splitting locations by classifying the sample Sandhi words into five classes with a bidirectional long short-term memory model. We applied the classified rules to rectify the words from the splitting locations. We identified 6,345 Pali Sandhi words from Dhammapada Atthakatha. We evaluated the performance of our proposed model on the basis of the accuracy of the splitting locations and compared the results with the dataset. Results showed that 92.20% of the splitting locations were correct, 1.10% of the Pali Sandhi words were predicted as non-splitting location words and 5.83% were not matched with the answers (incomplete segmentation).

Keywords: BiLSTM, Pali Sandhi, Thai Pali, Rule base, Pali Sandhi splitting

1. Introduction

Pali is an important language to Buddhists. It has been used to record the teachings of Buddha and Buddhist scriptures. Pali language recording was done using the specific characters of each country because they do not have their own alphabets. Nevertheless, with its grammar rules, Pali allows the meaning to be fully maintained. Therefore, the study of the Pali language can be a tool to help gain insight into Buddha's teachings. Table 1 shows the important teachings called Ovada Patimokkha in both Thai and Roman Pali scripts along with their meanings.

Table 1 Ovada Patimokkha in both Thai and Roman Pali scripts along with their meanings

Thai Pali Script	Thai Meaning	Roman Pali Script	Roman Meaning
สพฺพปาปสฺส อกรณํ	การไม่ทำความชั่วทั้งปวง	sabbapāpassa akarāṇaṃ	Doing no evil,
กุสลสฺสูปสมฺปทา	การทำกุศลให้ถึงพร้อม	kusalassūpasaṃpadā	Engaging in what's skillful,
สจิตฺตปริโยทปนํ	การทำจิตให้ผ่องใส	sacittapariyodapanaṃ	Doing no evil,
เอตํ พุทฺธาน สาสนํ.	นี่เป็นคำสอนของพระพุทฺธเจ้า	etaṃ buddhāna sāsanaṃ	This is the teaching of the buddhas.

The order of words in Pali sentences has spaces between words, making them easy to notice and split. Vocabulary can be conjugated in various forms according to the functions and numbers. Figure 1 shows the conjugation of ปาป (pāpa) to ปาปสฺส (pāpasa), which is the singular form acting as an indirect object or the possessive form. Pali grammar can create new words by combining two or more words. When the new words are transformed into compound words, they are called Samas. These words do not appear in the dictionary because they can always be rebuilt. Samas is a complete word and has the same meaning as the word สพฺพปาปสฺส (sabbapāpassa), which is shown in Table 1. The word สพฺพปาป (sabbapāpa) resulting from the combination of the words สพฺพ (sabba) and ปาป (pāpa) is inflected into สพฺพปาปสฺส (sabbapāpassa), as shown in Figure 2.

In addition to creating Samas words, Pali grammar can build new words by linking two or more consecutive words. The syllable sound between the last syllable of the first word and the first syllable of the second word are linked as if they are pronounced as the same syllable to create a melodiousness and harmonious sound. This process of building a new word into a portmanteau word is called Sandhi. For example, the words กุสลสฺส (kusalassa) and อุปสมฺปทา (upasaṃpadā) are combined into the Pali Sandhi word กุสลสฺสูปสมฺปทา (kusalassūpasaṃpadā). As shown in Figure 3, this Pali Sandhi word is not considered to be meaningful. Thus, to understand its meaning, the word must be reverted to its original form.

*Corresponding author.
Email address: promrit_n@silpakorn.edu; waijanya_s@silpakorn.edu
doi: 10.14456/easr.2021.63

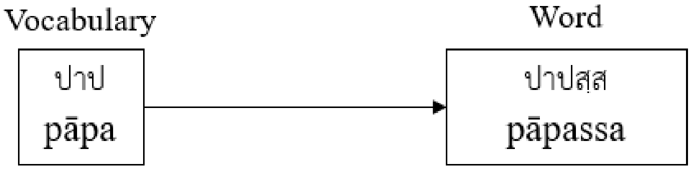


Figure 1 Inflecting a vocabulary into a new word

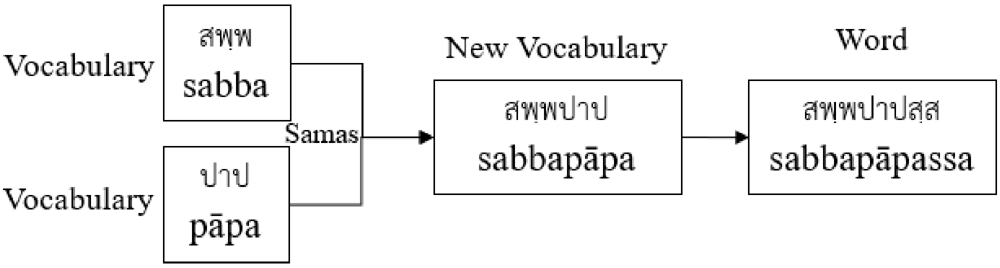


Figure 2 Building a new vocabulary by Samas processing and inflecting it into a new word

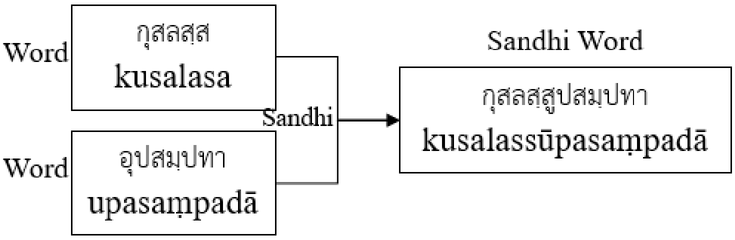


Figure 3 Combining two words by Pali Sandhi processing

The challenges of segmenting Sandhi words (i.e. Pali Sandhi words) are as follows:

- The difficulty in identifying whether it is a Sandhi word or not. For example, ปสฺสทานิ (passadāni) is a Sandhi word that is created from the combination of the words ปสฺส (passa) and อิทานิ (idāni).
- A Sandhi word cannot be fully segmented. For example, ปสฺสทานิสฺส (passadānissa) must be segmented into ปสฺส อิทานิ อสฺส (passa idāni assa). In particular, the word ปสฺสทานิ (passadāni) is a word that is difficult to identify whether it is a Sandhi word or not.
- The gender of the word is absent after Sandhi processing. For example, the word อถสฺสาหํ (athasāhaṃ) can be segmented into two forms, i.e. อถ อสฺส อหํ (atha assa ahaṃ) or อถ อสฺสา อหํ (atha assā ahaṃ). Both อสฺส (assa) and อสฺสา (assā) are the singular forms acting as an indirect object or the possessive forms. The difference is that อสฺส (assa) is a masculine word and อสฺสา (assā) is a feminine word.
- The number of words is lost. For example, the word มหาราชาติ (mahārājāti) can be segmented into มหาราชา อิติ (mahārājā iti) or มหาราช อิติ (mahārāja iti).
- The difficulty in identifying whether the end of the word has a short or long vowel. Because the vowel sound has been deleted or changed due to the linking of words, such as วณฺณทาสีติ (vaṇṇadāsīti), which must be segmented into วณฺณทาสี อิติ (vaṇṇadāsī iti) (the word วณฺณทาสี (vaṇṇadāsī) means a female slave caste, whereas the word วณฺณทาโส (vaṇṇadāso) means a male slave caste), the word คจฺฉสีติ (gacchasīti) must be segmented into คจฺฉสิ อิติ (gacchasi iti), whereas the word ธญฺญราสีติ (dhaññrāsīti) can be segmented into ธญฺญราสิ อิติ (dhaññrāsi iti) or ธญฺญราสี อิติ (dhaññrāsī iti). The words ธญฺญราสิ (dhaññrāsi) and ธญฺญราสี (dhaññrāsī) are singular and plural forms, respectively.

Normally, to segment the Pali Sandhi words back to their original forms, the vocabularies must be thoroughly understood so that they can be segmented correctly and accurately. Moreover, the surrounding context and the preceding text must be considered so that the Sandhi words can be segmented as accurately and relevantly as possible. The word segmentation process of the Thai Pali alphabets is even more challenging. The Thai Pali alphabets are incompatible with their pronunciations because the front vowels are เ- and โ- .

The Pali language is still used in many Theravada countries in Southeast Asia. Particularly in Thailand, Thai people learn Pali via chanting since primary school. Therefore, understanding Pali is the significant key that leads to access to Buddhist scriptures and chants. Nevertheless, there has never been any research on Thai Pali Sandhi segmentation.

This study proposed Thai Pali Sandhi segmentation techniques that use the bidirectional long short-term memory (BiLSTM) model to predict the words' splitting locations and select the classes of the applicable rules. Then, subrules from the selected applicable rules were used to improve accuracy and identify the meanings of words. Sandhi words used in this research were segmented more than once. The structure of this paper includes the literature review in Section 2, the methodology in Section 3, the experiments and results in Section 4 and the conclusion and future work in Section 5.

2. Literature review

Currently, Thai natural language processing has many challenges, such as word segmentation, sentence boundary detection and part-of-speech (POS) tagging. Thailand's language has a wide variety of word choices. Words can be formulated into new words using words from Pali and Sanskrit. A wide variety of these words appear in prosody poetry. Researchers have proposed the application of natural language processing of Thai poetry to syllable or word segmentation in prosodies, generation of Thai love quotes generation [1] and entity recognition of Thai poems [2], making it easier to find meanings.

Pali and Sanskrit are similar and related to each other because of their origins. Sanskrit is currently not used in daily communication but is still taught in a few places in India and the Western world probably because Sanskrit only appears in scriptures, ancient books or inscriptions and has similar roots to English and most of the other European languages. By contrast, the Pali language is still used in many Theravada countries in Southeast Asia to disseminate Buddhist teachings in the form of chants and books. Furthermore, in Thailand and some other countries, Pali has been taught to Buddhist monks, as well as the common people, in schools and universities.

Thai and Myanmar researchers focused on the application of natural language processing and computational linguistics to Pali and Sanskrit processing. Considerable research on the machine translation of Pali, i.e. Thai Pali to Thai [3], Thai Pali to English [4] and Myanmar Pali to Myanmar [5], has been conducted. Moreover, research on the detection of Thai Pali loanwords [6] and the identification of adopted Pali words in Myanmar text [7] has been conducted. Not only Thai and Myanmar researchers but also Indian researchers have developed a text-to-speech synthesiser for the Devanagari script in the Pali language [8]. Furthermore, POS tagger algorithms for Pali have been developed [9]. The SeNeReKo project, a joint research project of the Center for Religious Studies at Ruhr University Bochum and the Trier Center for Digital Humanities, published the Dictionary [10] of Pali and Preparation Buddhist Corpus in the Middle Indo-Aryan Pali Language by content-based analysis, followed by tokenisation (including Sandhi resolution by rule base), lemmatisation and POS tagging [11].

As previously mentioned, understanding Pali is important for gaining access to Buddhist scriptures and chants. To achieve the most accurate result, the functions of the other words in the sentence and the contexts of the previous sentences must be correctly clarified. The splitting of the constituent words of Pali Sandhi will improve content reading. However, only a few studies of natural language processing for splitting Pali Sandhi have been conducted. Alfter [12] proposed a morphological analyser and generator for Pali using the rule-based approach. Alfter considered Pali Sandhi in his work but did not report the accuracy of the proposed morphological analyser and generator. By contrast, Basapur et al. [13] proposed computational rules for the Pali Sandhi joiner on the basis of Pāṇinian grammar rules [14].

Currently, many researchers are more interested in Sanskrit language processing than Pali language processing because Sanskrit has similar roots to English and most of the other European languages. Researchers particularly focus on morphological analysis [15], word annotation [16], sentence boundary detection [17] and segmentation of Samas and Sandhi words.

The segmentation of Sandhi words in Sanskrit can be done using the convolutional neural network (CNN) [18] and recurrent neural network (RNN). Sandhi segmentation can also be done by applying the double decoder technique combined with the location decoder, which predicts the splitting locations using the bidirectional RNN. Then, the character decoder is applied to segment Sandhi words from the words in the splitting locations [19]. Research on similar processes, which are divided into two parts, has also been conducted. The first part is to predict the splitting locations called the Sandhi window. The second part is to split the words in the Sandhi window into two parts, segment the two parts using the Sandhi method and replace the results in the Sandhi window [20]. Two similar studies applied the RNN method to predict the splitting locations [19, 20]. However, Aralikatte et al. [19] converted the text in a single splitting location, whereas Dave et al. [20] split the words in the splitting location into two parts. Moreover, research applying statistical methods has been conducted [21]. Sandhi segmentation is an interesting and challenging work for computer linguists. Bhardwaj et al. [22] compared the performance of the (1) Sanskrit Sandhi Recogniser and Analyser developed at the Jawaharlal Nehru University (JNU), (2) Sanskrit Computational Toolkit developed at the University of Hyderabad (UoH) and (3) Sanskrit Reader Companion (INRIA) [23].

The BiLSTM is an extension of the traditional long short-term memory [24]. Both techniques are based on RNN (or sequential neural network). The BiLSTM has been applied to Chinese sentence detection [25], Thai sentence segmentation [25] and Sanskrit word segmentation [26]. The characteristics of Thai Pali Sandhi word segmentation are different from those of other word segmentation models. Thus, this study will use the BiLSTM to predict the words' splitting locations and the rule base to complete word segmentation.

However, no research on Thai Pali Sandhi segmentation or even Thai Sandhi segmentation has been conducted. We are quite convinced that the models for Thai word segmentation cannot segment the Pali Sandhi words because they are compound words formed by morphological transformation under the influence of consecutive words. The constituent words might transform after being combined with Sandhi words. To confirm this assumption, we will compare our proposed approach with well-known Thai word segmentation models, i.e. DeepCut [27] and AttaCut [28]. Both Thai word segmentation models use the CNN deep learning technique and publish libraries.

3. Methodology

This research used 6,345 Thai Pali Sandhi words from 8 volumes of dharma books, i.e. Dhammapada Atthakatha, about the origins and meanings of the words in Dharma topics that interest the people who have been affected by the presentation of the sermon. The topics are currently used as a course for learning Thai Pali. Many Pali language experts prepare and segment Sandhi words. The research processes are as follows: (3.1) data preparation, (3.2) analysis of the segmentation patterns, (3.3) categorisation of the splitting locations, (3.4) prediction of the splitting locations and (3.5) rule-based analysis of Pali constituent words, as shown in Figure 4.

In Sandhi word segmentation, first, the text is expanded to be as long as the longest Sandhi word. Then, the words are encoded as integer vectors before inputting them to predict the splitting locations. The five classes of splitting locations are Class 0, i.e. no splitting location, and Classes 1 to 4, i.e. splitting locations according to the first to fourth rule types, respectively. Then, the splitting locations of the Sandhi words are identified and the applicable rules are selected, as shown in Figures 5 and 6, respectively.

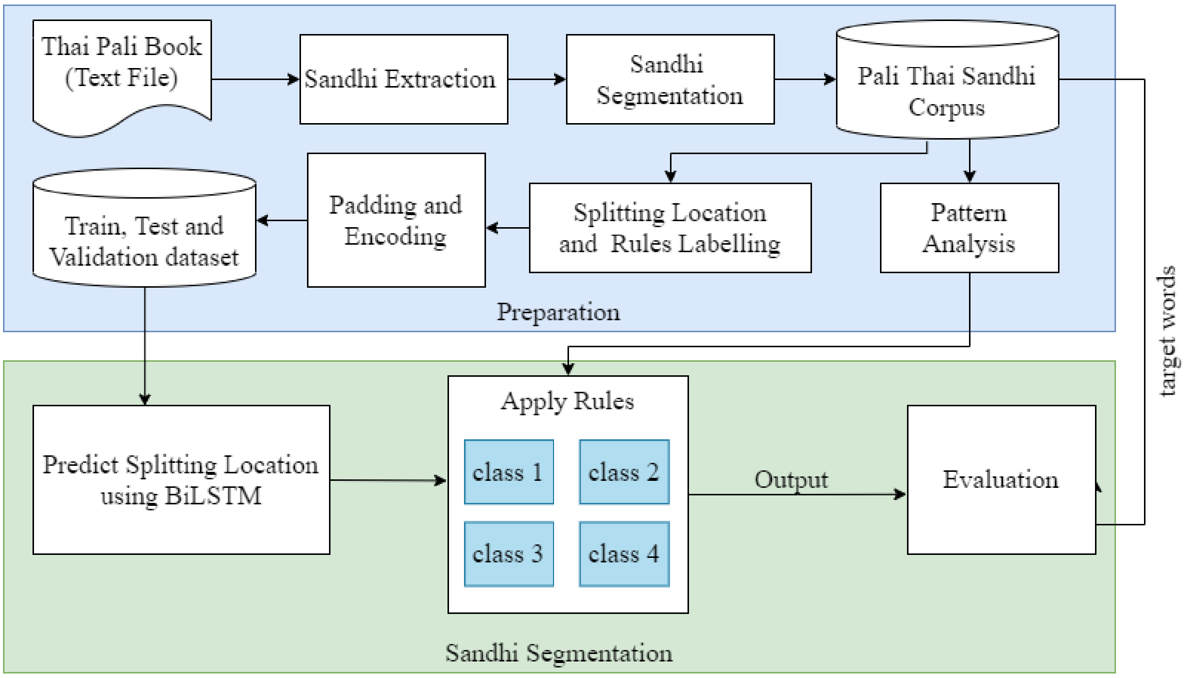


Figure 4 Overview of all research processes

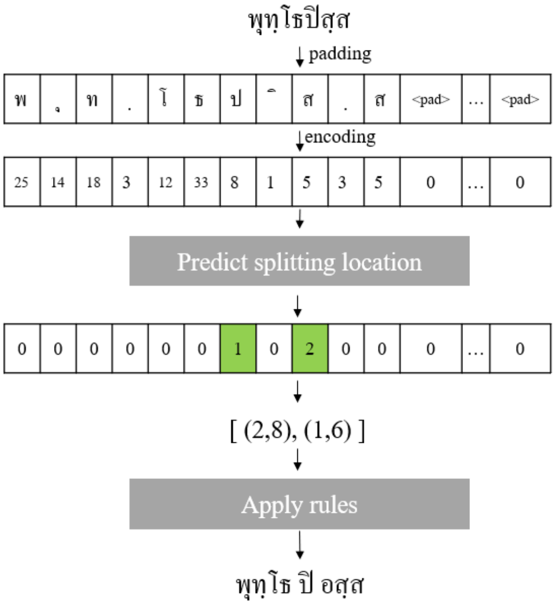


Figure 5 Example of Sandhi segmentation

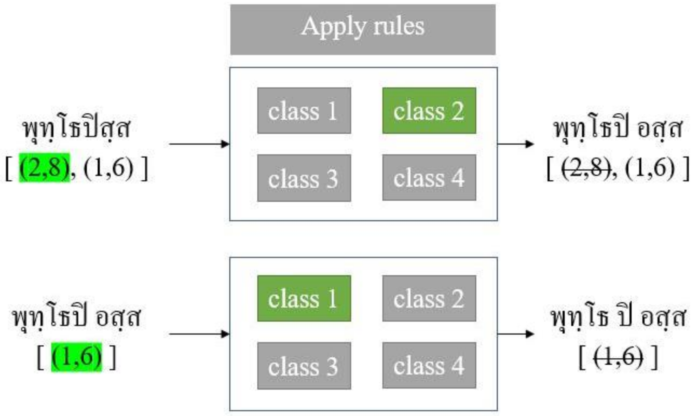


Figure 6 Selection and use of the applicable rules for the Sandhi words

3.1 Data preparation

The Tripitaka, Dhammapada and Pali language books were created and compiled into text files for use in websites and application by those interested in research. The text files of Dhammapada are stored with texts the same as that in the actual book, including the dash symbol (-) used to separate words that need to be split to other lines because of insufficient spaces, element tags added to the lines and footnotes to control the display on the website.

Therefore, the data listed below had to be cleaned up before they were sent to specialists who search for and split Sandhi words, as shown in Table 2. The first column shows the Sandhi words, whereas the second to fifth columns show the target words of Sandhi word segmentation. For example, the Sandhi word อถสฺสาหํ (athassāhaṃ) must be segmented into three words, i.e. อถ (atha), อสฺสา (assā) and อหํ (ahaṃ).

- Remove any element tag that appears in the file, such as <span> </span>, <sup> </sup> and <sub> </sub>.
- Convert the alphabets and as they appear in Pali into the Thai alphabets ญ and ฐ because these two characters contain unused character codes that cannot be displayed.
- Fix the incomplete words with a dash symbol that appears at the end of a line because it is split to a new line with \n added after the dash symbol (-). To illustrate, ปฏิ-\nสนฺธิ is rectified as ปฏิสนฺธิ because it is an incorrect segmentation.

Table 2 Data of the Sandhi and target words

Sandhi words	target words				
	1st word	**2nd word**	**3rd word**	**4th word**	**5th word**
มนุสฺสาปิ (manussāpi)	มนุสฺสา	ปิ			
อถสฺสาหํ (athassāhaṃ)	อถ	อสฺสา	อหํ		
อิทญฺจิทญฺจาติ (idañcidañcati)	อิทํ	จ	อิทํ	จ	อิติ

3.2 Analysis of the segmentation patterns

The following differences of Pali writing systems between Thai and Roman scripts result in more complex rules.

- Roman Pali script has only one vowel form (a, ā, i, ī, u, ū, e, o). Meanwhile, in Thai Pali script, the alphabet อ is used to mark the vowels in the syllable without the initial consonant (อ, อา, อิ, อี, อุ, อู, เอ and โอ). If there is an initial consonant sound, then the form of that consonant will be used instead of the alphabet อ.
- Roman Pali has a writing style that is in accordance with the following order of sound transcriptions: initial consonant, vowel and final consonant. Meanwhile, Thai Pali has a writing style that does not match the transcriptions because of the front vowels เ- and โ- .

From the analysis of the segmentation patterns, the applicable rules can be categorised into the following four types.

3.2.1 Pattern of the first rule type

In the first rule type, a splitting location can be separated into two words, and both can result in correct words and meanings, as shown in Figure 7.

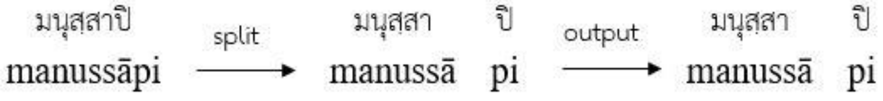


Figure 5 Pattern of the first rule type

3.2.2 Pattern of the second rule type

In the second rule type, a splitting location can be split into two words. The first word is correct and meaningful, whereas the second word must be rectified according to the subrules, as shown in Figure 8.

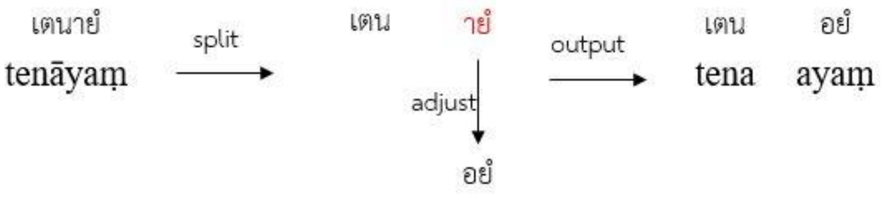


Figure 6 Pattern of the second rule type

3.2.3 Pattern of the third rule type

In the third rule type, a splitting location can split into two words. The second word is correct and meaningful, whereas the first word must be modified according to the subrules, as shown in Figure 9.

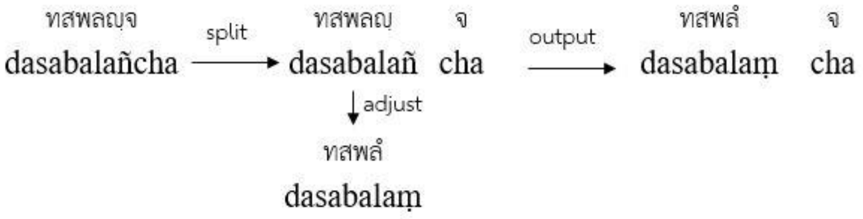


Figure 7 Pattern of the third rule type

3.2.4 Pattern of the fourth rule type

In the fourth rule type, no splitting location can be split into meaningful words. Therefore, two splitting locations are determined, and the words within the splitting locations are adjusted according to the subrules, as shown in Figure 10.

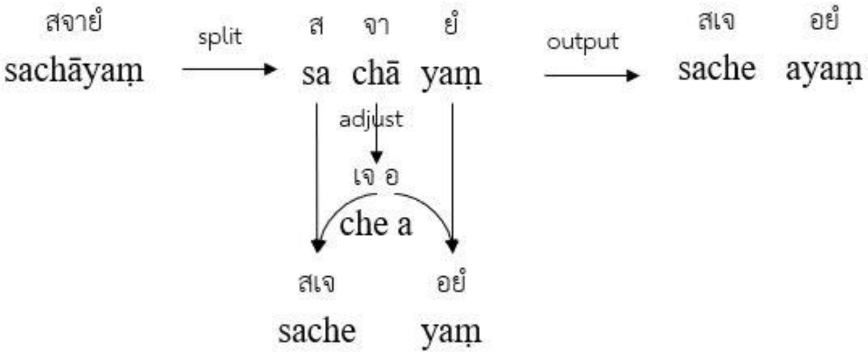


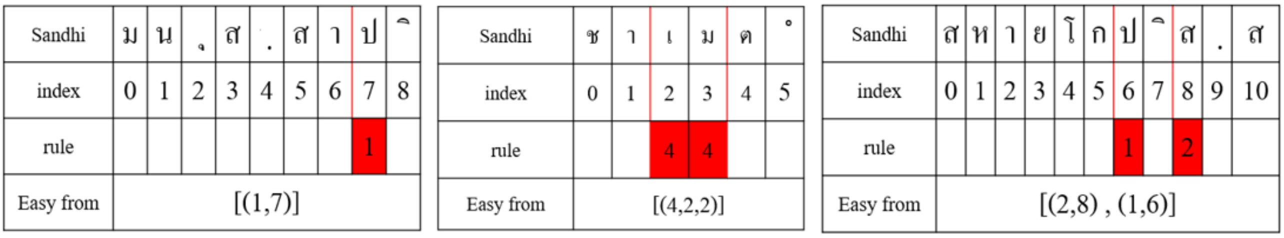
Figure 8 Pattern of the fourth rule type

3.3 Categorisation of the splitting locations

As discussed in Section 3.2, the first to third rule types have only one splitting location, whereas the fourth rule type has two splitting locations. Thus, $i-$th represents the segmentation of Sandhi word S by $(r,p,l)_i$, r represents the type of the rule, p represents the splitting location and l represents the length of the word in the splitting location of the fourth rule type. Nevertheless, the first to third rule types do not need to specify these values.

Sandhi words can be segmented more than once; thus, the splitting locations solution will be $[(r,p,l)_n, \ldots, (r,p,l)_1]$. Figures 11(a) and 11(b) illustrate the examples of Sandhi words that have been segmented once according to the first and fourth rule types, respectively. Figure 11(c) illustrates the examples of Sandhi words that have been segmented multiple times.

The input data of splitting location prediction are Sandhi word S and the solutions O, $|S| = |O|$ and $O_i \epsilon$ [0,4], where $O_i = 0$ denotes no splitting location, $O_i = 1$ denotes the splitting location according to the first rule type, $O_i = 2$ denotes the splitting location according to the second rule type, $O_i = 3$ denotes the splitting location according to the third rule type and $O_i = 4$ denotes the splitting location according to the fourth rule type.



Sandhi	ม	น	ุ	ส	ฺ	ส	า	ป	ิ
index	0	1	2	3	4	5	6	7	8
rule								1	
Easy from	[(1,7)]								

(a) Position where the Class 1 rule of segmentation is applied

Sandhi	ช	า	เ	ม	ต	ํ
index	0	1	2	3	4	5
rule			4	4		
Easy from	[(4,2,2)]					

(b) Position where the Class 4 rule of segmentation is applied

Sandhi	ส	ห	า	ย	โ	ก	ป	ิ	ส	ฺ	ส
index	0	1	2	3	4	5	6	7	8	9	10
rule							1		2		
Easy from	[(2,8) , (1,6)]										

(c) Sandhi word that has more than one splitting location

Figure 9 Example of the preparation of the splitting location data with specific rule classes

Table 3 shows the Sandhi words and the results of the splitting locations, where the rule types are presented as the vectors of the ordinal pairs and in the form of integer vectors. For example, the word มนุสฺสาปิ (manussāpi) has seven splitting locations according to the first rule type, i.e. segmented once and denoted as [(1,7)], and when written as an integer vector, it creates a zero vector with a length of 57 and converts the number in the splitting locations into rule types. To exemplify, the number 0 in the zero vector of position 7 is converted into number 1 with the position starting from zero.

Table 3 Data and solutions for splitting location prediction

Sandhi words	Results	Results (length = 57)
มนุสฺสาปิ (manussāpi)	[(1, 7)]	[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, …, 0]
อถสฺสาหํ (athassāhaṃ)	[(2, 5), (2, 2)]	[0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, …, 0]
อิทญฺจิทญฺจ (idañcidañca)	[(3, 10), (2, 6), (3, 5)]	[0, 0, 0, 0, 0, 3, 2, 0, 0, 0, 3, 0, 0, 0, …, 0]
อิทญฺจิทญฺจาติ (idañcidañcāti)	[(2, 11), (3, 10) , (2, 6), (3, 5)]	[0, 0, 0, 0, 0, 3, 2, 0, 0, 0, 3, 2, 0, 0, …, 0]
พุทฺโธปิสฺส (buddhopissa)	[(2, 8), (1, 6)]	[0, 0, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, …, 0]

3.4 Prediction of the splitting locations

From the dataset prepared to train the splitting location prediction model, first, the Sandhi word is extracted to expand the edge of the length till it is as long as the longest Sandhi word (i.e. padding). Then, encode the Sandhi word in text form into a vector of integers, as shown in Figure 12. The result is expanded by 0 (i.e. zero padding) and adjusted to O_i from scalar to one-hot vector, as shown in Figure 13.

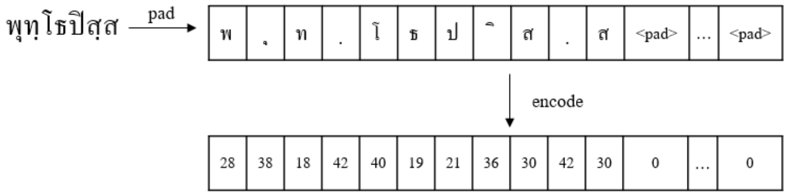


Figure 10 Encoding the Sandhi word before interpolating in the splitting location prediction model

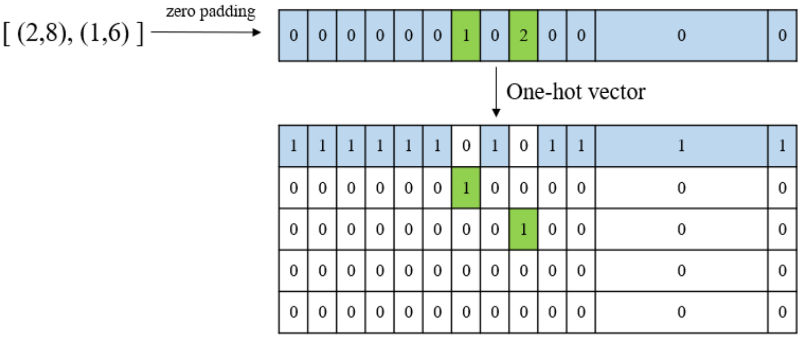


Figure 13 Converting the solution into a one-hot vector

The 33 characters of the Thai Pali alphabets were used in this research, together with 1 Bindu (ฺ) and 8 vowels (8 vowels with 7 forms because there is no form for the ‘a’ vowel sound, but the อ form is used instead of vowels without initial consonants) and the ี form, which was used instead of writing ิ + Nikhit (ํ). Thus, 33 + 1 + 8 + 1 = 43 characters were used.

This research attempted to segment the Thai Pali scripts at the character level. The functional structure of the splitting location prediction model is shown in Figure 14.

Figure 14 shows that the longest Sandhi word has 57 characters. The embedding size is equal to 100 for the embedding layer. The BiLSTM layer with 100 + 100 nodes, recurrent dropout value of 0.5 and rectified linear unit (ReLU) activation function were used before the data were sent to the last hidden layer containing 50 nodes with the recurrent dropout value of 0.5 and ReLU activation function before sending the data to the output layer with 5 nodes. The activation function is set to SoftMax, and the model is trained using the Adam optimiser with the number of epochs of 1,000, initial learning rate of 0.0001 and batch size of 32.

3.5 Rule-based analysis of Pali constituent words

From the model shown in Figure 14, the Sandhi word or message has been identified as the splitting locations. Then, the subrules from the selected applicable rules were used to improve accuracy and identify the meanings of the words.

3.5.1 First rule type

The first rule type has no subrules because words can be segmented accurately and meaningfully.

3.5.2 Second rule type

In the second rule type, the second word is modified using one of the four subrules, as follows:

- Rectify ติ or าติ into อิติ and ทานิ into อิทานิ.
- If the word begins with a vowel without อ vowel, then add อ.
- Rectify เทว or เยว into เอว.
- If the word starts with ค, ท or ห, then add อา in front of the word.

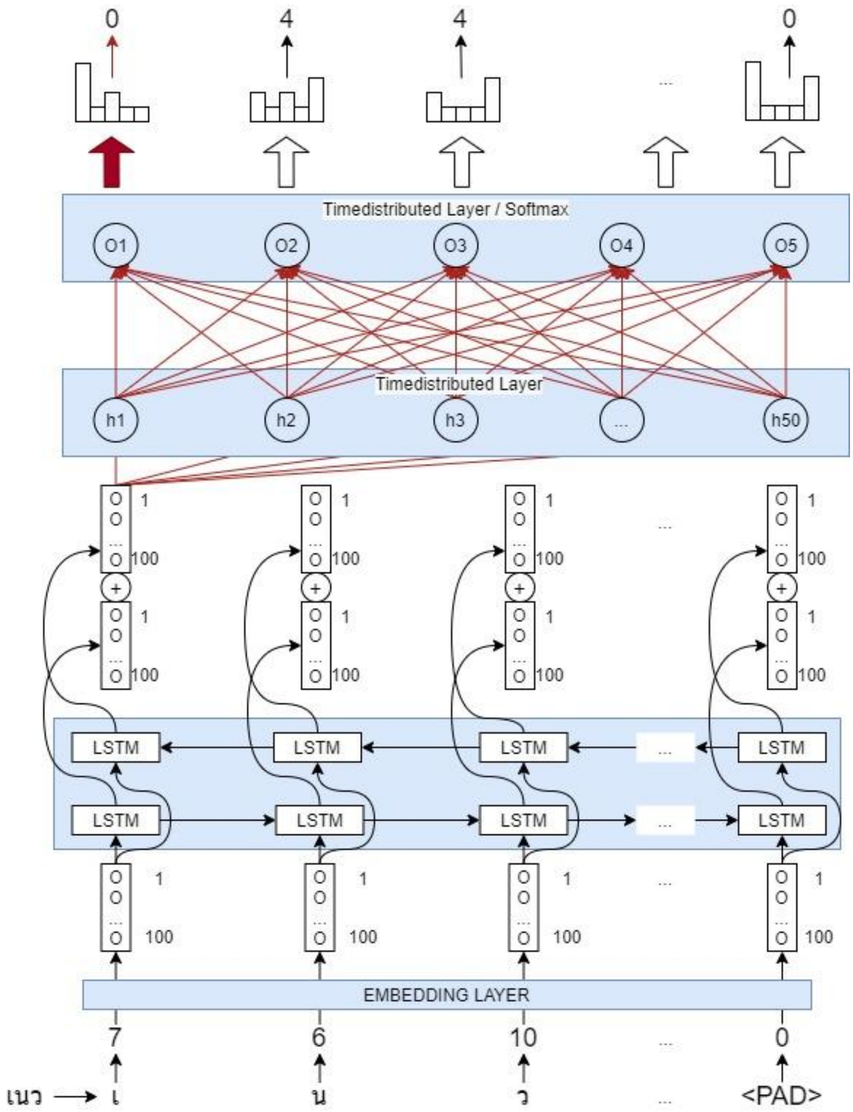


Figure 14 Splitting location prediction model

3.5.3 Third rule type

In the third rule type, the first word is modified one of the two subrules, as follows:

- Convert งฺ, ญฺ, ณฺ, นฺ or มฺ into Nikhit (◌ํ).
- If the word begins with the i vowel (i.e. the form of the ◌ิ vowel without อ preceding) and followed by งฺ, ญฺ, ณฺ, นฺ or มฺ, then all three characters must be replaced with Nikhit (◌ํ).

3.5.4 Fourth rule type

No splitting location can be split into meaningful words. Therefore, two splitting locations are determined, and the words within the splitting locations are adjusted using one of the three subrules, as follows:

- Convert จา into เจ อ.
- Convert เตฺวา into อิติ เอว.
- Convert เสฺว into โส เอ.

4. Experiments and results

This study proposes the Thai Pali Sandhi segmentation using a hybrid approach combining deep learning and rule base. The experiments and results are explained in the (4.1) splitting location prediction model, (4.2) performance evaluation of the model, (4.3) comparison with a well-known Thai word segmentation model and (4.4) comparison with Sanskrit Sandhi segmentation subsections.

4.1 Splitting location prediction model

To predict the splitting location of Thai Pali Sandhi words, this experiment's dataset consists of 6,345 Thai Pali Sandhi words. The dataset is divided into 4,568 words for training, 508 words for validation and 1,269 words for testing. The Sandhi words are post-padded to have 57 characters, which is the same length as the longest Sandhi word. Because the splitting locations in this study means the position of characters in each word, the model's dataset is the number of the spot of all of the words' locations multiplied by the longest word's length (i.e. 57). Thus, the training data have 4,568 × 57 = 260,376 spots, the validation data have 508 × 57 = 28,956 spots and the test data have 1,269 × 57 = 72,333 spots.

For our proposed model shown in Figure 14, we set the embedding size of 100 for the embedding layer, the recurrent dropout value of 0.5 for the BiLSTM layer and the recurrent dropout value of 0.5 for the last hidden layer. The activation functions are ReLU and SoftMax. The model is trained using the Adam optimiser with the number of epochs of 1,000, initial learning rate of 0.0001 and batch size of 32.

The splitting location prediction model had five possible prediction results, i.e. Class 0 to Class 4. Class 0 means no splitting location, Class 1 means segmenting the word according to the first rule type, Class 2 means segmenting the word according to the second rule type, Class 3 means segmenting the word according to the third rule type and Class 4 means segmenting the word according to the fourth rule type.

The accuracy and loss of the splitting location prediction model using BiLSTM are shown in the accuracy and loss graphs. Figure 15(a) shows that the accuracy of the training data is 0.9996 and that of the validation data is 0.9981. Figure 15(b) shows that the loss value of the training data is 0.0001 and that of the validation data is 0.0138. The narrow gap between training and validation indicates the good fit of the proposed model.

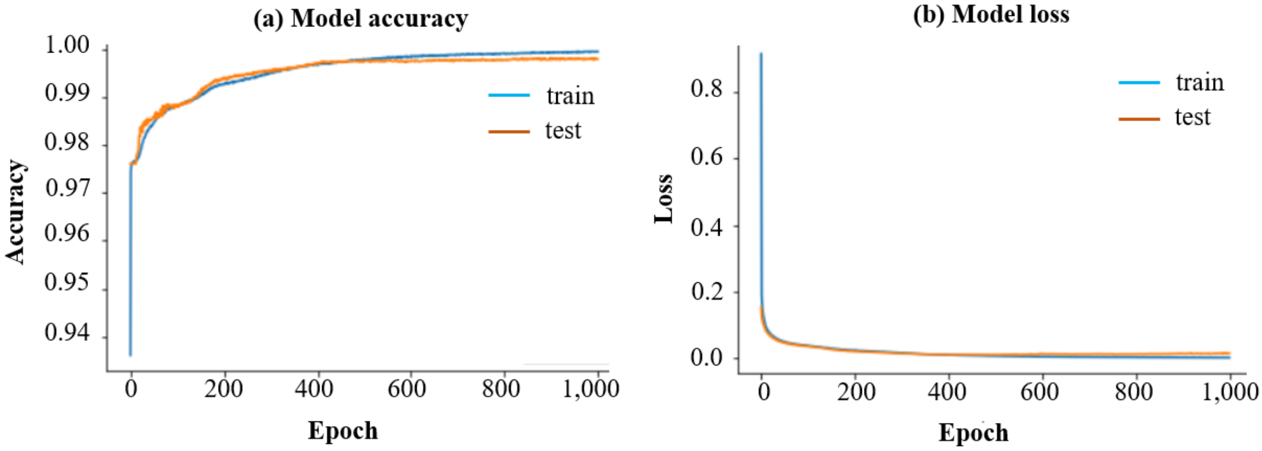


Figure 11 Accuracy and loss of the splitting location prediction model

4.2 Performance evaluation of the model

4.2.1 Performance of the splitting location prediction model

To evaluate the performance of the model, we use the confusion matrix, precision, recall and F1 score. The test data are 1,269 Thai Pali Sandhi words with 72,333 locations. Referencing the analysis to find the pattern of Pali segmentation previously mentioned in our methodology, there are five classes of segmentation pattern prediction. The test data are imbalanced in each class because of the different number of words, i.e. some classes with only a few words are those consisting of rarely used words. Therefore, the numbers of the locations for predicting whether the words can be split or not, or which splitting patterns they are, are as follows: Class 0 = 70,662 locations, Class 1 = 278 locations, Class 2 = 474 locations, Class 3 = 98 locations and Class 4 = 804 locations.

Table 4 shows the confusion matrix of the splitting location prediction model of all of the five classes. The columns represent the actual classes, whereas the rows represent the prediction results.

Table 4 Confusion matrix

		Actual				
		0	**1**	**2**	**3**	**4**
Predict	0	70,622	4	32	1	31
	1	2	274	0	0	0
	2	32	0	439	0	5
	3	4	0	0	97	0
	4	19	0	3	0	768

Table 4 shows that Class 0 correctly answered 70,582 locations, whereas Classes 1, 2, 3 and 4 incorrectly answered 2, 32, 4 and 19 locations, respectively. Class 1 correctly answered 274 locations and incorrectly answered 4 locations in Class 0. Class 2 correctly answered 439 locations and incorrectly answered 32 and 3 locations in Classes 0 and 4, respectively. Class 3 correctly answered 97 locations and incorrectly answered 1 location in Class 0. Class 4 correctly answered 768 locations and incorrectly answered 31 and 5 locations in Classes 0 and 2, respectively.

We measure the model’s performance using precision, recall and F1 score. Table 5 illustrates the performance of each class, both macro-average and weighted average. For the macro-average, the precision is 0.9693, the recall is 0.9712 and the F1 score is 0.9702. For the weighted average, the precision is 0.9982, the recall is 0.9982 and the F1 score is 0.9982.

Table 5 Precision, recall and F1 score of the splitting location prediction model

Class	Precision	Recall	F_1-Score	Support
0	0.9990	0.9992	0.9991	70,679
1	0.9928	0.9865	0.9892	278
2	0.9223	0.9262	0.9242	474
3	0.9604	0.9898	0.9749	98
4	0.9722	0.9552	0.9636	804
Macro - average	0.9693	0.9712	0.9702	72,333
Weighted - average	0.9982	0.9982	0.9982	72,333

The performance evaluation values of Class 2 are slightly lower than the other classes. From the examination, the incorrect results derived from the words should be split according to Class 2, but the splitting location of those words are the vowel alphabets (i.e. า, ◌ิ , ◌ี, ◌ุ, ◌ู, เ and โ). An example of an incorrect output is เทมาติ (demāti). The label of the splitting location is [0 0 0 $\underline{2}$ 0 0 …], whereas that of the model’s prediction result is [0 0 0 $\underline{0}$ 0 0 …].

Table 6 shows other examples of incorrect outputs. The table presents the input (i.e. Sandhi), the target and the output. The target column shows the expected splitting location and words. The output column shows the predicted splitting locations from the splitting location model and the results after splitting.

Table 6 Examples of incorrect outputs

Sandhi	The target		The output		Description
	Splitting location	Words	Splitting location	Words	
เทมาติ (demāti)	0 0 0 $\underline{2}$ 0 0 0 0 0 0 0 0 0	เทม อิติ	0 0 0 $\underline{\mathbf{0}}$ 0 0 0 0 0 0 0 0 0	เทมาติ	No splitting location
วตฺวาติ (vatvāti)	0 0 0 0 $\underline{0}$ $\underline{2}$ 0 0 0 0 0 0 0	วตฺวา อิติ	0 0 0 0 $\underline{\mathbf{2}}$ $\underline{2}$ 0 0 0 0 0 0 0		Excessive splitting location
อิทญฺจิทญฺจ(idañcidañca)	0 0 0 0 0 3 $\underline{2}$ 0 0 0 3 0	อิทํ จ อิทํ จ	0 0 0 0 0 3 $\underline{\mathbf{0}}$ 0 0 0 3 0	อิทํ จิทํ จ	Splitting location are not enough
สยถาติ (sayathāti)	0 0 0 $\underline{2}$ $\underline{0}$ 0 0 0 0 0 0 0 0	สยถ อิติ	0 0 0 $\underline{\mathbf{0}}$ $\underline{\mathbf{2}}$ 0 0 0 0 0 0 0 0	สยถา อิติ	Incorrect splitting location

4.2.2 Thai Pali Sandhi segmentation evaluation

A total of 72,333 splitting locations of 1,269 Thai Pali Sandhi words were predicted with high accuracy. The final result of this research proposes the Pali constituents from the Thai Pali Sandhi words. Therefore, we need to investigate the words that have more than one splitting location individually because they consist of more than two constituent words.

After we determined the Sandhi words’ splitting locations using BiLSTM, we used the subrules from the selected applicable rules to rectify or transform the words. The segmentation correctness is evaluated by searching the Pali constituent words from the datasets of experts who are Levels 5 and 8 Pali scholars. The results of the evaluation are the (1) complete segmentation with all correct constituent words, (2) complete segmentation with some correct constituent words and (3) incomplete segmentation, as shown in Figure 16.

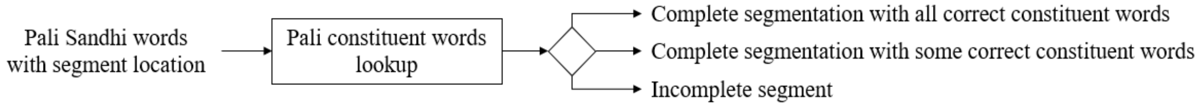

Pali Sandhi words
with segment location
Pali constituent words
lookup
Complete segmentation with all correct constituent words
Complete segmentation with some correct constituent words
Incomplete segment

Figure 12 Evaluation of our Sandhi segmentation

In Figure 16, complete segmentation means the Thai Pali Sandhi words with correct splitting locations. Incomplete segmentation means the Thai Pali Sandhi words with incorrect splitting locations and no splitting location. After investigating the words individually, the accuracy of our proposed model is evaluated and the results are shown in Table 7.

Table 7 Accuracy of our proposed model (number of words)

	Completely segmented		Incompletely segmented	
	Correct all constituent words	Correct some constituent words	Incorrect splitting location	No splitting location
Thai Pali Sandhi Words	1,170 (92.20%)	11 (0.87%)	74 (5.83%)	14 (1.10%)

As shown in Table 7, 1,170 (92.20%) completely segmented Sandhi words have correct constituent words according to the answers. By contrast, 11 (0.87%) completely segmented Sandhi words have incorrect constituent words according to the answers. The incompletely segmented Sandhi words are 74 words (5.83%) and the Sandhi words with no splitting location are 14 words (1.10%).

The results of the Thai Pali Sandhi splitting model show that 74 incompletely segmented words consist of 88 words with incorrect splitting locations. Meanwhile, 14 words did not have splitting locations. Of the 74 incompletely segmented Sandhi words with incorrect splitting locations, 7 constituent words could be read according to grammar rules. However, their contexts did not match those of the book because some of the Sandhi words could be segmented into more than one pattern. There were nine incompletely segmented Sandhi words because their predicted splitting locations were switched. Moreover, 38 words did not match the number of splitting locations because the predicted splitting locations were more or less than the answers.

4.3 Comparison with a well-known Thai word segmentation model

In this section, we compare our proposed model with two well-known Thai word segmentation models, i.e. DeepCut and AttaCut. DeepCut and AttaCut are Thai word segmentation libraries that use CNN. In our assumption, both libraries might not be able to segment Thai Pali words. The models may be trained by various Thai words and learn some Thai Pali Sandi words, particularly the Thai Pali loanwords. However, Thai Pali Sandhi words are compound words formed by morphological transformation under the influence of consecutive words. Hence, the constituent words might transform after being combined with Sandhi words.

In the experiment, we use 1,269 Thai Pali Sandhi words. The expected output of the comparison is the accuracy of the words that each model can segment. Table 8 illustrates the accuracy of our proposed model and two well-known Thai word segmentation models. Our proposed model can obtain correct results for 1,170 words (92.20%), whereas the two other models are unsuitable for Sandhi splitting because they have low accuracy (i.e. DeepCut = 2.04% and Attacut = 0.97%).

Table 8 Comparison with Thai word segmentation

Our purpose	Thai word segmentation	
	Deep cut	**Atta cut**
1,170 (92. 20%)	25 (2.04%)	10 (0.97%)

Table 9 shows the examples of the results of the comparison between Thai word segmentation models. The first column shows Sandhi words, and the second column shows the target words. The output column consists of the results of Sandhi segmentation of our proposed model, DeepCut and AttaCut.

Table 9 Examples of the results of the comparison between Thai word segmentation models

Sandhi	Target	The output		
		Our purpose	**DeepCut**	**AttaCut**
มาตาปิตูนํปิ (mātāpitūnaṃpi)	มาตาปิตูนํ ปิ	✓ มาตาปิตูนํ ปิ	× มาตาปิตูน ◌ํ ปิ	× มาตาปิตูนํปิ
ปูชํปิ (pūjaṃpi)	ปูชํ ปิ	✓ ปูชํ ปิ	× ปูชํปิ	× ปู ชํปิ
กาญฺจนรูปกโตปิ (kāncanarūpakatopi)	กาญฺจนรูปกโต ปิ	✓ กาญฺจนรูปกโต ปิ	× กาญฺจนรูปกโตปิ	× กาญฺจน รูป กโตปิ
วีสติปิ (vīsatipi)	วีสติ ปิ	✓ วีสติ ปิ	× วีสติปิ	✓ วีสติ ปิ
วิชฺชมานาปิ (vijjamānāpi)	วิชฺชมานา ปิ	✓ วิชฺชมานา ปิ	✓ วิชฺชมานา ปิ	× วิชฺชมา นาปิ

4.4 Comparison with Sanskrit sandhi segmentation

Previous studies of Pali Sandhi word segmentation did not consider the accuracy of their models, and their numbers are still few. Therefore, to compare our proposed model with the models proposed in other works, Sanskrit Sandhi word segmentation studies are selected according to the similarity of their origins. The results of the comparison are shown in Table 10.

Table 10 Comparison with Sanskrit Sandhi segmentation

Research Article	Scope of work	Language	Corpus	Words	Method	Accuracy
SandhiKosh: A Benchmark Corpus for Evaluating Sanskrit Sandhi Tools [22]	Create corpus + survey Sandhi splitting tools	Devanagari Sanskrit	UoH + SandhiKosh [22]	13,648 (100% test)	JNU Tools, UoH Tools, INRIA Tools	7.64% of JNU, 53.69% of UoH, and 58.18% of INRIA
Sanskrit Sandhi Splitting using seq2(seq)^2 [19]	Sandhi segmentation + Create model	Devanagari Sanskrit	UoH + SandhiKosh [22]	71,747 (80% train, 20% test)	Double Decoder RNN	95.0% of location prediction and 79.5% of split prediction
Neural Compound-Word (Sandhi) Generation and Splitting in Sanskrit Language [20]	Sandhi segmentation + Create model	Devanagari Sanskrit	UoH + SandhiKosh [22]	77,842 (80% train, 20% test)	RNN for location predion and BiLSTM for split prediction	92.3% of location prediction and 86.8% of split prediction
Our purpose	Sandhi segmentation + Create model	Thai Pali	8 Dhammapada books	6,345 (80% train, 20% test)	BiLSTM + Rule Base	99.82% of location prediction and 92.20% of split prediction

Table 10 summarises the performance of the model proposed in each research, as follows: Bhardwaj et al. [22] assessed the performance of three Sanskrit Sandhi segmentation tools. The dataset is the SandhiKosh corpus with 13,848 words. The segmentation accuracy of JNU is 7.64%, UoH is 53.69% and INRIA is 58.18%.

Aralikatte et al. [19] used the Sanskrit corpus with 71,747 words from the UoH and SandhiKosh corpora to train and test the model with 14,349 words. The accuracy of splitting location prediction is 95.0% and that of Sanskrit Sandhi segmentation is 79.5%.

Dave et al. [20] used the Sanskrit corpus with 71,842 words from the UoH and SandhiKosh corpora to train and test the model with 15,569 words. The accuracy of splitting location prediction is 92.3% and that of Sanskrit Sandhi segmentation is 86.8%.

We used 6,435 unique Thai Pali Sandhi words from 8 Dhammapada books to develop and train our proposed model. The accuracy of splitting location prediction is 99.82% and that of Thai Pali Sandhi segmentation is 92.20%.

Currently, we still have not found any research on natural language processing and computational linguistics for Thai Pali Sandhi word segmentation. Furthermore, the Thai Pali Sandhi word corpus has never been published before. Thus, this study is the first approach to such linguistic fields by all means.

5. Conclusion and future work

This research presents the segmentation of Thai Pali Sandhi words using deep learning BiLSTM to predict splitting locations and applicable rules to rectify the incorrect words.

This research's input data are the sample Sandhi words. To correctly segment Sandhi words so that the segmented words are related to the context, other words in the sentences must also be considered. In addition to Sandhi words, which are known as words that are not included in dictionaries or lexicons and have no meaning, other words, which are derived from the combination of roots into new words, that are not included in dictionaries but have meanings are called Samas. Splitting Samas words will enable better language processing because the split words can be traced back to the original meanings before they are combined.

The processing could be more conveniently completed by inputting the data that help detect Samas and Sandhi words. Then, segmentation could be easily done, resulting in easy word forms with correct meanings. Therefore, this research could be used as a reference and a primary analysis tool for subsequent Thai Pali language processing.

6. References

[1] Khongtum O, Promrit N, Waijanya S. Text-based LSTM networks for automatic Thai love quotes generation on twitter. Inform Tech J. 2019;14(2):1-8.

[2] Khongtum O, Promrit N, Waijanya S. The entity recognition of Thai poem compose by Sunthorn Phu by using the bidirectional long short term memory technique. In: Chamchong R, Wong K, editors. International conference on multi-disciplinary trends in artificial intelligence; 2019 Nov 17-19; Kuala Lumpur, Malaysia. Berlin: Springer; 2019. p. 97-108.

[3] Phonson N. The rule-based machine translation system from Pali to Thai [thesis]. Bangkok: Mahidol University; 2001.

[4] Kornwirat B. A program for the machine translation of Pali into English (Pali MT) [thesis]. Bangkok: Mahidol University; 2003.

[5] Khaing PP, Thwe KZ. Proposed framework for Pali words to Myanmar text translation. Int Conf Comput Appl. 2015:90-5.

[6] Wanglem B, Tongtep N. Pattern-sensitive loanword estimation for Thai text clustering. Walailak J Sci Tech. 2017;14(10): 813-23.

[7] Maung ZM. Identification of adopted Pali words in Myanmar text. Int J Comput Sci Issues. 2012;9(6):128-36.

[8] Mache S, Mahender C. Development of text-to-speech synthesizer for Pali language. J Comput Eng. 2016:18(3):35-42.

[9] Haribhakta Y, Nadageri L. Parts of speech tagger for Pali language. International J Sci Res Comput Sci, Eng Informat Tech. 2018:2(4):845-53.

[10] Knauth J, Alfter D. A dictionary data processing environment and its application in algorithmic processing of Pali dictionary data for future NLP tasks. In: Boitet C, Malik MGA, editors. Proceedings of the fifth workshop on south and Southeast Asian natural language processing; 2014 Aug 23; Dublin, Ireland. Dublin: Association for Computational Linguistics and Dublin City University; 2014. p. 65-73.

[11] Elwert F, Sellmer S, Wortmann S, Pachurka M, Knauth J, Alfter D. Toiling with the Pali Canon. In: Mambrini F, Passarotti M, Sporleder C, editors. Proceedings of the workshop on corpus-based research in the humanities; 2015 Dec 10; Warsaw, Poland. 2015. p. 39-48.

[12] Alfter D. Morphological analyzer and generator for Pali [Bachelor thesis]. Trier: University of Trier; 2015.

[13] Basapur S, Shivani V, Nair SS. Pali Sandhi - a computational approach. In: Goyal P, editor. Proceedings of the 6th International Sanskrit Computational Linguistics Symposium; 2019 Oct 23-25; West Bengal, India. Stroudsburg: Association for Computational Linguistics; 2019. p. 182-193.

[14] Scharf PM. Modeling Paṇinian grammar. In: Huet G, Kulkarni A, Scharf P, editors. International Sanskrit computational linguistics symposium; 2008 May 15-17; France. Berlin: Springer; 2009. p. 95-126.

[15] Hellwig O. Morphological disambiguation of classical Sanskrit. In: Mahlow C, Piotrowski M, editors. Systems and frameworks for computational morphology; 2015 Sep 17-18; Stuttgart, Germany. Berlin: Springer; 2015. p. 41-59.

[16] Hellwig O, Hettrich H, Modi A, Pinkal M. Multi-layer annotation of the Rigveda. Proceedings of the eleventh international conference on language resources and evaluation (LREC); 2018 May 7-12; Miyazaki, Japan. France: European Language Resources Association; 2018.

[17] Hellwig O. Detecting sentence boundaries in Sanskrit texts. In: Matsumoto Y, Prasad R, editors. Proceedings of COLING 2016, the 26th international conference on computational linguistics; 2016 Dec 11-16; Osaka, Japan. Japan: The COLING 2016 Organizing Committee; 2016. p. 288-297.

[18] Hellwig O, Nehrdich S. Sanskrit word segmentation using character-level recurrent and convolutional neural networks. In: Riloff E, Chiang D, Hockenmaier J, Tsujii J, editors. Proceedings of the 2018 conference on empirical methods in natural language processing; 2018 Oct 31-Nov 4; Brussels: Belgium. Stroudsburg: Association for Computational Linguistics; 2018. p. 2754-63.

[19] Aralikatte R, Gantayat N, Panwar N, Sankaran A, Mani S. Sanskrit sandhi splitting using seq2(seq)2. In: Riloff E, Chiang D, Hockenmaier J, Tsujii J, editors. Proceedings of the 2018 conference on empirical methods in natural language processing; 2018 Oct 31-Nov 4; Brussels: Belgium. Stroudsburg: Association for Computational Linguistics; 2018. p. 4909-14.

[20] Dave S, Singh AK, PA P, Lall B. Neural compound-word (Sandhi) generation and splitting in Sanskrit language. In: Haritsa J, Roy S, Gupta M, Mehrotra S, Srinivasan BV, Simmhan Y, editors. CODS COMAD 2021: 8th ACM IKDD CODS and 26th COMAD; 2021 Jan 2-4; Bangalore, India. New York: Association for Computing Machinery; 2021. p. 171-7.

[21] Natarajan A, Charniak E. S³ - Statistical sandhi splitting. In: Wang H, Yarowsky D, editors. Proceedings of 5th international joint conference on natural language processing; 2011 Nov 8-13; Chiang Mai, Thailand. Hong Kong: Asian Federation of Natural Language Processing; 2011. p. 301-8.

[22] Bhardwaj S, Gantayat N, Chaturvedi N, Garg R, Agarwal S. Sandhikosh: a benchmark corpus for evaluating Sanskrit Sandhi tools. In: Calzolari N, Choukri K, C Cieri C, Declerck T, Goggi S, Hasida K, et al, editors. Proceedings of the eleventh international conference on language resources and evaluation (LREC); 2018 May 7-12; Miyazaki, Japan. France: European Language Resources Association; 2018.

[23] Goyal P, Huet G. Completeness analysis of a Sanskrit reader. 5th International Symposium on SansSkrit Computational Linguistics; 2013 Jan 4-6; Mumbai, India.

[24] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735-80.

[25] Xu H, Hongsu W, Sanqian Z, Qunchao F, Jun L. Sentence segmentation for classical Chinese based on LSTM with radical embedding. J China Univ Post telecomm. 2019;26(2):1-8.

[26] Hellwig O. 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; 2015 Nov 27-29; Poznan, Poland. p. 289-93.

[27] Kittinaradorn R, Achakulvisut T, Chaovavanich K, Srithaworn K, Chormai P, Kaewkasi C, et al. DeepCut: a Thai word tokenization library using deep neural network [computer program]. Version 1.0. Zenodo; 2019.

[28] Chormai P, Prasertsom P, Rutherford A. AttaCut: A fast and accurate neural Thai word Segmenter. arXiv:1911.07056. 2019:1-13.