Learning Methods and Features for
Corpus-based Phrase Break Prediction on Thai

Chatchawarn Hansakunbuntheung, Ausdang Thangthai, Chai Wutiwiwatchai, and Rungkarn Siricharoenchai

Speech Technology Section, Information R&D Division
National Electronics and Computer Technology Center (NECTEC), Thailand
{chatchawarnh, ausdang.thangthai, chai, rungkarn}@nectec.or.th

Abstract

This paper presents applications of five famous learning methods for Thai phrase break prediction. Phrase break prediction is particularly important for our Thai text-to-speech synthesizer (TTS), where input Thai text has no word and sentence boundary. The learning methods include a POS sequence model, CART, RIPPER, SLIPPER and neural network. Features proposed for the learning machines can be extracted directly from the input text during real processing. The best method based on the CART model gives 80.14\% correct-break, 94.40\% juncture-correct, and 2.37\% false-break scores. Comparing to our previous models based on C4.5 and RIPPER, the new optimized method achieves almost the best performance.

1. Introduction

On recent research of text-to-speech synthesis (TTS), the most significant goal is naturalness of synthesized speech. The naturalness comprises of factors in both suprasegmental and segmental levels. In this paper, we focus on phrase break position, which is a significant factor on the suprasegmental level. It functions on segmenting utterance into meaningful sub-utterances. Thus, placing phrase break at different positions in a sentence can generate different meanings or, even, incorrect senses. It also effects highly on fluency of generated speech.

The early research on phrase break prediction was based on syntactic analysis and rule-based methods. These kinds of methods have been commonly known that they can generate unreliable results when dealing with complicated or unseen text. Therefore, data-driven approaches were widely proposed such as a probabilistic n-gram model [1], a tree-based model [2], a neural network [3], a boosting technique [4] or hybrid models [5][6]. No unique method has been claimed as a state-of-the-art algorithm for phrase break prediction since there is no a complete comparison among the methods and no standard evaluation data. The phrase break prediction is also a language-dependent task, where features and a prediction algorithm need to be optimally selected for a particular language.

In case of Thai, Thai is another language that has its own characters and writing system. The characteristics of Thai writing system are as the followings: 1) Thai writing has no word segmentation. When writing generally, Thai characters are written consecutively without any space or special symbol to hint about word boundaries and, furthermore, each word can comprised of one character up to any number of characters without any explicit rule. 2) On sentence and phrase level, Thai has no phrase and sentence boundary mark. Spaces and punctuation marks do not clearly indicate the boundaries. Furthermore, phrase break can occur at any places, even between any words, spaces or symbols in the text.

At present, there has been only a few research sites investigating on Thai phrase break prediction. Methods proposed in the past included a rule-driven method [7], a decision tree, and a boosting technique [4]. Although [4] successfully applied RIPPER, a kind of boosting technique, to this task, some features used in [4] could not be implemented in a real application as one of the features, a number of syllables between the current juncture and the next phrase break, was unable to be extracted during real processing. The cause of incident is that the next phrase break is a target of the prediction. Therefore, we need to find a new set of features, which can produces a good result and can be used in real-time TTS.

In this paper, five candidates of machine learning methods are evaluated with our proposed features for predicting prosodic phrase breaks. The best method will be used in our Thai TTS system [8]. Analyzed features are carefully selected so that they can be extracted from input text in the real-time TTS. The rest of paper is organized as follows: the next section mentions about the architecture of our proposed Thai phrase break predicting system. Section 3 describes the features proposed for phrase break prediction. Section 4 gives details of our experimental data, which is a large Thai speech corpus with prosody tags for the data-driven TTS. Section 5 explains five machine learning methods investigated in our experiment. Section 6 shows evaluation criteria. Section 7 presents experimental results with discussions in Sect. 8. The final section concludes the paper.

2. The Architecture of Our Thai Phrase Break Prediction System

As mentioned about the characteristics of Thai writing system in the previous section, we designed the structure of phrase break analysis module, which considered the constraints of Thai writing characteristics. The system is illustrated in Fig. 1.

First, the input Thai text is tokenized into several segments by their types e.g. Thai character, digit or symbol sequences. Next the Thai character sequences are segmented into word-units and tagged with their corresponding POSs. Then, the POS-tagged word sequences are processed by probabilistic sentence segmentation. Finally, the word sequences are analyzed by the phrase break prediction.
3. Features

As described in the introduction, the aim of useful features is that they can be computed given only input text and uncomplicated features are preferable so that they can be used in real-time TTS. The proposed features are shown in Table 1. Part-of-speech (POS) used here consists of 47 types as defined in ORCHID [9]. The real-value features are normalized by their corresponding means and standard deviations.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSL2, POSL1, POSR1, POSR2</td>
<td>Part-of-speech context (POS) consisting of two preceding and succeeding POSs of words around the current juncture</td>
</tr>
<tr>
<td>NSyl</td>
<td>Total number of syllables in the current sentence</td>
</tr>
<tr>
<td>NWrdl</td>
<td>Total number of words in the current sentence</td>
</tr>
<tr>
<td>NSyl_PrevPB</td>
<td>Number of syllables between the previous phrase break and the current juncture</td>
</tr>
<tr>
<td>NSyl_ToEnd</td>
<td>Number of syllables between the current juncture and the end of current sentence</td>
</tr>
<tr>
<td>NWrd_PrevPB</td>
<td>Number of words between the previous phrase break and the current juncture</td>
</tr>
<tr>
<td>NWrd_ToEnd</td>
<td>Number of words between the current juncture and the end of current sentence</td>
</tr>
<tr>
<td>SylPosRatio</td>
<td>The ratio of NSyl_PrevPB and NSyl</td>
</tr>
<tr>
<td>WrdPosRatio</td>
<td>The ratio of NWrd_PrevPB and NWrd</td>
</tr>
</tbody>
</table>

4. Experimental Data

In this paper, two sets of experimental data are used. The first set is a large data set used for evaluating learning methods. The data set are selected from a Thai speech corpus for TTS development (TSynC) [10], which contains read speech tagged with POSs, hand-labeled prosodic phrase breaks and other information. The experimental data used here consists of 5,066 sentences (102,271 words and 15,716 phrase breaks). We divided the data set into a training subset and a test subset. The training subset contains 4,560 sentences (92,044 words and contains 14,115 phrase breaks). The test subset contains 506 sentences (10,227 words and 1,601 phrase breaks).

The second set is designed as that used in [4]. The aim of this set is to compare our proposed model with the engine implemented in [4]. This data set is selected from the ORCHID corpus [9]. It contains totally 1,169 sentences (12,538 words and 1,960 phrase breaks). The POSs and phrase breaks have been tagged by linguists. This set is also divided into two parts. The first part is a training subset containing 10,059 words and 1,596 phrase breaks. The second part is a test subset consisting of 2,479 words and 364 phrase breaks.

5. Learning Methods

Five learning methods are compared in our experiment. Some of them have been successfully applied for this task.

5.1. POS sequence model

A POS sequence model is a widely-used probabilistic model as expressed in Eq. 1.

\[
\tilde{P}(C|j_i) = \frac{\text{count}(C|j_i)}{\text{count}(j_i)}
\]  

where \( C \) is the given POS sequence and \( j_i \) is the considered juncture type.

This model estimates from a training set probabilities of target junctures using their neighboring POSs. Many research works in the past showed that this model gave considerable results on this task. Thus, this model is used as a baseline system in our experiment.

5.2. CART

A CART learning machine is based on a binary decision tree. This method has been applied in many research works on this task as mentioned above. Its advantages are that it can manipulate both symbolic and real-value features, and the model can be efficiently trained even the training data is sparse. Since some of our proposed features are originally not quantitative and the training set is probably sparse, we include this method in our experiments.

5.3. RIPPER

RIPPER [11] is a well-known rule learning system, which produces a compact set of readable rules. In [4], they used this system to detect phrase break in Thai. Therefore, we also need to evaluate this method.

5.4. SLIPPER

SLIPPER [12] is another famous rule learning system, which has been proven to be superior to the RIPPER in many tasks [12]. Thus we compare this system in our experiment.
5.5. Neural Network

A neural network (NN) is one of famous non-linear classification and regression machines. It generates favor results in various kinds of research. Some research works on phrase break prediction used the NN in a hybrid model [6]. In this paper, we apply a multilayer perceptron type of NN with a single hidden-layer for phrase break/non-phrase break classification.

6. Evaluation Criteria

Evaluation criteria used in our experiment indicate performance of the investigated methods and the proposed features in several aspects including a break-correct score (BC), a juncture-correct score (JC) and a false-break score (FB).

6.1. Break-correct (BC)

This score determines the correctness of predicted phrase-break junctures using a ratio between the number of correctly predicted break-junctures and total number of actual break-junctures as shown in Eq. 2.

\[
\text{Break-correct (BC)} = \frac{CB}{B} \times 100\%
\]

6.2. Juncture-correct (JC)

This score indicates the overall correctness of juncture prediction both on break and non-break junctures. It can be computed by Eq. 3.

\[
\text{Juncture-correct (JC)} = \frac{CB + CNB}{B + NB} \times 100\%
\]

6.3. False-break (FB)

To prevent over break-insertion, we take this score into account. This score shows the error caused by incorrect break insertion. It is calculated by Eq. 4.

\[
\text{False-break (FB)} = \frac{NB - CNB}{NB} \times 100\%
\]

where

- \(CB\) is the total number of correct-predicted breaks,
- \(CNB\) is the total number of correct-predicted non-breaks,
- \(B\) is the total number of actual breaks, and
- \(NB\) is the total number of actual non-breaks.

7. Experiments and Results

In the first experiment, we compare the five learning methods mentioned in the Sect. 5 when dealing with the proposed features as described in the Sect. 3. The training and test set are those extracted from the TSynC corpus as mentioned in the Sect. 4.

In the POS sequence model, a window of four neighboring POSs, two preceding and two succeeding POSs around the juncture, is used. Probabilities in the POS sequence model are calculated as shown in the Eq. 1. We use this model as a baseline system. Prediction results are shown in Table 2.

In a training phase, learning parameters of each method are optimized so that the best results are obtained. In CART training, the maximum number of data in leaf nodes and the number of partitions for real-value inputs are finely tuned to get the best result. In neural network training, the number of hidden nodes is empirically adjusted.

After optimizing each training method, their results when evaluating with the test set are shown in Table 2. According to the Table 2, the baseline method produces quite fair scores but its scores are the lowest when comparing to the other methods. The results report that the best method is the neural network and the second one is the CART model. Comparing to the baseline model, both methods give higher scores on all measurements i.e. break-correct, juncture-correct and false-break scores. Therefore, we choose these two methods as candidates for the next experiment.

Table 2: Comparative results between the systems used in the past research [4] and the proposed models.

<table>
<thead>
<tr>
<th>Method</th>
<th>Features</th>
<th>BC</th>
<th>JC</th>
<th>FB</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS Seq.</td>
<td>This paper</td>
<td>81.31</td>
<td>94.64</td>
<td>2.83</td>
</tr>
<tr>
<td>NN</td>
<td>This paper</td>
<td>74.49</td>
<td>93.67</td>
<td>2.69</td>
</tr>
<tr>
<td>C4.5 Coll.</td>
<td>POS</td>
<td>69.75</td>
<td>89.04</td>
<td>7.97</td>
</tr>
<tr>
<td>C4.5 Coll.+Syl</td>
<td>POS</td>
<td>72.53</td>
<td>95.44</td>
<td>4.03</td>
</tr>
<tr>
<td>SLIPPER</td>
<td>Coll.</td>
<td>74.18</td>
<td>93.79</td>
<td>3.83</td>
</tr>
<tr>
<td>RIPPER</td>
<td>Coll.</td>
<td>80.21</td>
<td>93.67</td>
<td>2.44</td>
</tr>
<tr>
<td>RIPPER</td>
<td>Coll.+Syl</td>
<td>84.60</td>
<td>94.15</td>
<td>2.46</td>
</tr>
</tbody>
</table>

In the next experiment, we compare the performance of the top two methods from the first experiment with the systems proposed in the previous Thai phrase-break prediction research [4]. Data sets in comparison are designed as those used in [4]. In [4], C4.5 and RIPPER were used to learn rules using features extracted from the training set. The features consist of “collocation” features, which are n-contiguous words and POSs, and “syllable” features, which are the numbers of syllables between the target juncture and the previous and the next phrase break. We found that the last feature, the number of syllable from a juncture to the next phrase break, cannot be extracted in real-use since the next phrase break is the result of the prediction. Furthermore, the n-word patterns strongly depend on the training data and its domain. The comparative results are shown in Table 3. The table shows that the CART method, instead, gives better results than the neural network. Comparing with the past research [4], the CART one gives almost the best results on this experiment.

Table 3: Comparative results between the systems used in the past research [4] and the proposed models.

<table>
<thead>
<tr>
<th>Method</th>
<th>Features</th>
<th>BC</th>
<th>JC</th>
<th>FB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CART</td>
<td>This paper</td>
<td>81.31</td>
<td>94.64</td>
<td>2.83</td>
</tr>
<tr>
<td>NN</td>
<td>This paper</td>
<td>74.49</td>
<td>93.67</td>
<td>2.69</td>
</tr>
<tr>
<td>POS Seq.</td>
<td>Coll.</td>
<td>69.75</td>
<td>89.04</td>
<td>7.97</td>
</tr>
<tr>
<td>C4.5</td>
<td>Coll.+Syl</td>
<td>72.53</td>
<td>95.44</td>
<td>4.03</td>
</tr>
<tr>
<td>SLIPPER</td>
<td>Coll.</td>
<td>74.18</td>
<td>93.79</td>
<td>3.83</td>
</tr>
<tr>
<td>RIPPER</td>
<td>Coll.</td>
<td>80.21</td>
<td>93.67</td>
<td>2.44</td>
</tr>
<tr>
<td>RIPPER</td>
<td>Coll.+Syl</td>
<td>84.60</td>
<td>94.15</td>
<td>2.46</td>
</tr>
</tbody>
</table>

8. Discussion

In the first experiment, as expected, the POS sequence model gives the lowest break-correct and juncture-correct scores. Due to the limited range of contextual information presented in the POS sequence model, it cannot capture far contextual
clues in the text. A solution is to extend the range of surrounding context. However, this solution is impractical. The total number of POSs is 48. To constitute all possible four-POS sequences around a juncture, the estimated number of required patterns is about 10,616,832 sequences (48 POSs * 48 POSs * 2 Juncture types *48 POSs *48 POSs), while the number of junctures with contextual POSs in our corpus is only 102,271. It means that a vast number of POS sequences are not constructed and an optimal smoothing technique is required. When implementing the POS sequence model in the second experiment, which uses a relatively small size of training data, an influence of data sparseness becomes more obvious. Similarly to the first experiment, the POS sequence model gives the lowest scores among all methods due to the limitation of training data size.

Table 2 shows that the neural network gives the best break-correct score with a slightly better juncture-correct and false-break scores provided by the CART method. During neural network training, the optimal number of the hidden nodes was 200 nodes. This number of the hidden nodes requires high computation especially when the prediction needs to be performed at every juncture of the input text. While the CART method gives a lower break-correct score, it achieves better juncture-correct and false-break scores and it needs very low computation. Furthermore, the performance of neural network decreases when training by a small data set. Although, we have reduced the number of hidden nodes to several values, the best one still gives worse results than the CART model. To include the phrase break prediction in a real-time TTS system, the CART method is more suitable in a sense of high correctness, low computation, fast processing and ease of rule interpretation.

Besides the several learning methods described in the Sect. 5, two kinds of well-known probabilistic models, including support vector machine (SVM) and hidden Markov models (HMM), have also been investigated. However, based on the proposed features, we have not yet obtained considerable results from these two models. SVM kernels, probability distribution functions, a suitable set of features as well as the topology of HMM are all needed to be optimized in order to get acceptable results.

To optimize the proposed prediction model in the future work, we need to intensively select the input features and the learning methods. In the case of features, the current number of features is 12. Only the most significant subset of these 12 features might be optimal for a particular learning method. A suitable way to select a subset of significant features such as a minimum classification error (MCE) algorithm [13] can be applied. Apart from the current features, one of interesting features is the window size of POS sequences as it has been proven to highly affect the performance of phrase break prediction for English [1]. If the window size of POS sequences is suitable, the learning method can better gather crucial information from the input text.

9. Conclusions

This paper presented the learning methods and the proposed features for Thai phrase break prediction to be used in real Thai TTS. The result of the experiment showed that the neural networks give the best results i.e. 83.89% of break-corrects, 94.42%, and 2.52% of false-break. However, the CART method gives preferable result and need lower computation than the neural network methods. Furthermore, It shows more reliability on both the larger data set and the smaller one. Thus the CART method is more preferable in this task. The best scores of this method are 80.14% of break-correct, 94.90% of juncture-correct, and 2.37% of false-break. In addition, the CART methods and the proposed features give almost highest performance and are more practical when comparing with the past research.

10. References