Chunwijitra *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2016) 2016:15
DOI 10.1186/s13636-016-0093-x

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

A hybrid input-type recurrent neural network for LVCSR language modeling

Vataya Chunwijitra*, Ananlada Chotimongkol and Chai Wutiwiwatchai

Abstract

Substantial amounts of resources are usually required to robustly develop a language model for an open vocabulary speech recognition system as out-of-vocabulary (OOV) words can hurt recognition accuracy. In this work, we applied a hybrid lexicon of word and sub-word units to resolve the problem of OOV words in a resource-efficient way. As sub-lexical units can be combined to form new words, a compact set of hybrid vocabulary can be used while still maintaining a low OOV rate. For Thai, a syllable-based unit called pseudo-morpheme (PM) was chosen as a sub-word unit. To also benefit from different levels of linguistic information embedded in different input types, a hybrid recurrent neural network language model (RNNLM) framework is proposed. An RNNLM can model not only information from multiple-type input units through a hybrid input vector of words and PMs, but can also capture long context history through recurrent connections. Several hybrid input representations were also explored to optimize both recognition accuracy and computational time. The hybrid LM has shown to be both resource-efficient and well-performed on two Thai LVCSR tasks: broadcast news transcription and speech-to-speech translation. The proposed hybrid lexicon can constitute an open vocabulary for Thai LVCSR as it can greatly reduce the OOV rate to less than 1 % while using only 42 % of the vocabulary size of the word-based lexicon. In terms of recognition performance, the best proposed hybrid RNNLM, which uses a mixed word-PM input, obtained 1.54 % relative WER reduction when compared with a conventional word-based RNNLM. In terms of computational time, the best hybrid RNNLM has the lowest training and decoding time among all RNNLMs including the word-based RNNLM. The overall relative reduction on WER of the proposed hybrid RNNLM over a traditional n-gram model is 6.91 %.

Keywords: Recurrent neural network language model, Pseudo-morpheme, Hybrid language model, LVCSR

1 Introduction

The vocabulary of any active language continues to grow as new words, such as person names, place names, and new technical terms, are introduced everyday. This poses a challenge on building a language model (LM) for a large vocabulary continuous speech recognition (LVCSR) system. Substantial amount of resources, e.g., memory and computational time, for both training and decoding are required to handle an open vocabulary LM; otherwise, the performance of an LVCSR system could be hurt by a high out-of-vocabulary (OOV) rate. A hybrid LM of word and sub-word units has been shown to be resource-efficient for an LVSCR system in various languages [1–5] as sub-lexical units can be combined to form new words; thus, a compact set of vocabulary can be used while still maintaining a low OOV rate. On another aspect, different types of lexical units in a hybrid LM provide different levels of linguistic information which can be combined to better predict word probability. In [6, 7], characters were combined with words to add another type of constraint in Chinese hybrid LMs.

In this paper, we apply a hybrid LM of word and sub-word units to a Thai LVCSR system with two goals in mind, to alleviate the problem of OOV words and to improve recognition accuracy, both in a resource-efficient way. A suitable sub-lexical unit depends largely on the characteristic of each language. In Thai, since there is neither inflection nor derivative, a syllable-based unit called *pseudo-morpheme (PM)* is used as a sub-lexical unit in the proposed hybrid LM instead of a morpheme-based unit as in morphologically rich languages. According to

*Correspondence: vataya.chunwijitra@nectec.or.th
NECTEC, National Science and Technology Development Agency (NSTDA), 112 Pahonyothin Road, Pathumthani 12120, Thailand

Thai writing rules, PM is more deterministic when compared with word and has been shown to alleviate a word segmentation problem [8].

To benefit from different levels of linguistic information embedded in word and sub-word units, an approach which can naturally model multiple-type input units is considered. In this work, a recurrent neural network (RNN) is used to model a hybrid word-PM LM as there is no restriction on the RNN input types. Moreover, RNN is chosen from its ability to model longer history not only $n-1$ previous words through recurrent connections between its hidden layer and input layer as detailed in [9]. As a result, RNN can capture long context patterns instead of fixed-length contexts as in a traditional neural network (NN) LM [7].

In our proposed hybrid RNNLM, a hybrid vector of words and PMs is used as an input vector. Unlike the hybrid NNLM in [7], the output from our hybrid RNNLM can contain both words and PMs in order to handle OOV words. In the first-pass decoding, a hybrid n-gram LM similar to [4] is utilized to create a hybrid n-best list where OOV words could be recognized as a sequence of PMs. A hybrid RNNLM, which can consider information from different types of input units together, is then applied in the second-pass to re-score the hybrid n-best list for better recognition accuracy. Besides the application of RNN, we also explore several hybrid input representations to optimize both recognition accuracy and computational time. In addition to a full-hybrid RNNLM which takes both a word sequence and a PM sequence as its input, two variations of reduced-hybrid RNNLMs are proposed to decrease computational complexity. By using two types of units, the vocabulary size of the full-hybrid RNNLM could be twice the size of the word-based RNNLM. In the first reduced-hybrid RNNLM variation, the size of the hybrid vocabulary is reduced to be equal to the size of the word-based RNNLM vocabulary by including only frequent words and PMs. In the other variation, a mix of word and PM sequence is used as an input instead.

This paper is organized as follows: Section 2 explains the characteristics of Thai text together with a pseudo-morpheme (PM), a sub-lexical unit in Thai. Section 3 describes our proposed hybrid RNNLM framework for combining different input unit types. Section 4 describes the recognition process of the hybrid RNNLM. Recognition results on two Thai LVCSR tasks are then discussed in Section 5. We finally conclude our work and discuss future directions in Section 6.

2 Thai lexical

2.1 Lexical structure and vocabulary growth

Thai is a non-segmented script language, i.e., there is no boundary marker between words. Furthermore, there is no capital letter to indicate the beginning of a sentence or a proper noun. The definition of word unit is often ambiguous due to the presence of compound words. These characteristics become a challenge when processing Thai text.

The vocabulary growth of Thai text is illustrated by a type-token curve in Fig. 1. This curve is plotted from 5 million words randomly selected from three text and speech corpora: BEST [10], LOTUS-BN [11], and HIT-BTEC [12]. To balance the amount of data from different corpora and domains, 500K words are selected from each of the eight genres in BEST and 500 K words each are selected from LOTUS-BN and HIT-BTEC. The total becomes 5 millions words from ten different genres. We can see from the type-token curve that even with 5 million words, the vocabulary continues to grow. New names are the main cause of the vocabulary growth. In LOTUS-BN, where named entities were annotated with specific tags, the type-token ratio for named-entity alone is 0.357 while the type-token ratio for all words is 0.044. New words also arise from transliterated words and abbreviations. Since many named-entities are a compound word, this type of OOV word should be able to be modeled by sub-lexical units.

2.2 Pseudo-morpheme

The design of a sub-lexical unit depends largely on the characteristics of each language. In Thai, there is neither inflection nor derivative; hence, another type of sub-lexical unit should be used instead of morpheme. As a letter in Thai is a phonogram which roughly represents a phoneme or combination of phonemes, Thai word could be segmented into a sequence of syllable-like units.

A basic Thai textual syllable composes of four components, represented in the form of $\{C_i, V, C_f, T\}$, where C_i, V, C_f and T denote an initial consonant, a vowel, a final consonant, and a tone, respectively, as shown in Fig. 2. The corresponding phoneme (or phonemes) of each component in IPA is also illustrated.

The word /bâ:n - phrá:w/, which is a village name, in Fig. 2 consists of two syllables: /bâ:n/ and /phrá:w/. The

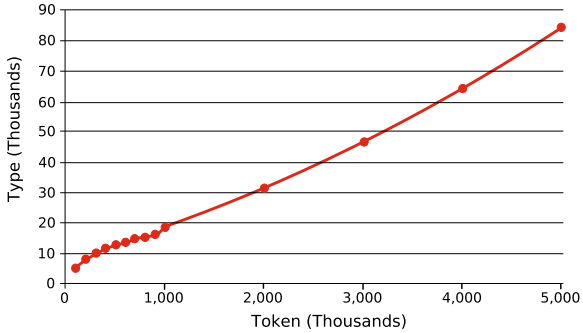

Fig. 1 Type-token. Type-token curve of Thai text

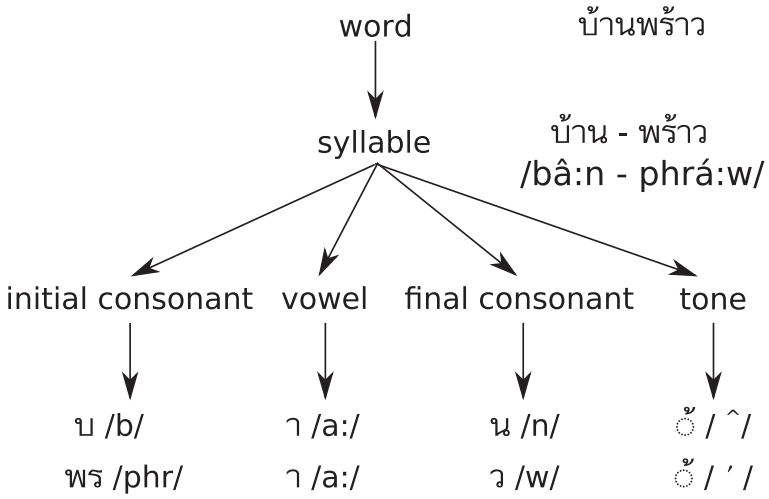

Fig. 2 Thai-syllable. An example of Thai word and sub-word units

first syllable /bâ:n/ has a basic syllable pattern with all four components { b, a:, n , ˆ }. Some textual syllables may have multiple initial consonants, vowel forms or final consonants while some syllables may have components omitted. The second syllable /phrá:w/ has two initial consonants; /ph/ and /r/. Nevertheless, patterns of syllables can be defined and are known to be finite. Thai writing rules can be used to identify syllables and their components in a given text.

A pseudo-morpheme (PM) defined as a syllable-like unit in a written form is used as a sub-lexical unit for Thai [8]. The example word in Fig. 2 consists of two PMs /bâ:n/ and /phrá:w/. According to Thai writing rules, PM is more deterministic when compared with word. Given a word or a string of text, PMs can be determined quite accurately with an automatic segmentation tool [13]. More examples of words and theirs corresponding PMs are given in Fig. 3 where PMs are separated by "|". A PM must not be confused with a phonetic syllable in word pronunciation as a PM may correspond to multiple phonetic syllables. For example the first word /júp - phá? | râ:t/ in Fig. 3, its first PM corresponds to two phonetic syllables /júp - phá?/. Phonetic syllables are separated by '-' in the third column. Words in Fig. 3 are examples of OOV words found in our test sets. The first and second words are proper names while the last word is a loan word from the word "poll result". Named-entities and loan words are known to be the main causes of OOV. By modeling an OOV word with a sequence of PMs, these OOV words could be correctly recognized by our hybrid LM.

3 Hybrid recurrent neural network language model

The purpose of a language model (LM) employed in an ASR system is to provide the probability of a word given the history of its preceding words. For an LM to efficiently predict the next word, it is well-known that long word history should be utilized to capture long context patterns, syntactic and semantic dependencies. A recurrent neural network (RNN) [9] can learn an effective representation of history from the training data through recurrent connections between a hidden layer and an input layer as shown in Fig. 4. With a recurrent connection through $s(t-1)$, the hidden layer or context layer s of RNNLM can capture longer word history than

Word	PM	Pronunciation
ยุพราช	ยุพ | ราช	[júp - phá? | râ:t]
ธาระรูป	ธาระ | รูป	[thā:n - rá? | rû:p]
ผลโพล	ผล | โพล	[phŏn | phō:l]

Fig. 3 Thai-PMs. Examples of words, their corresponding PMs, and their pronunciations

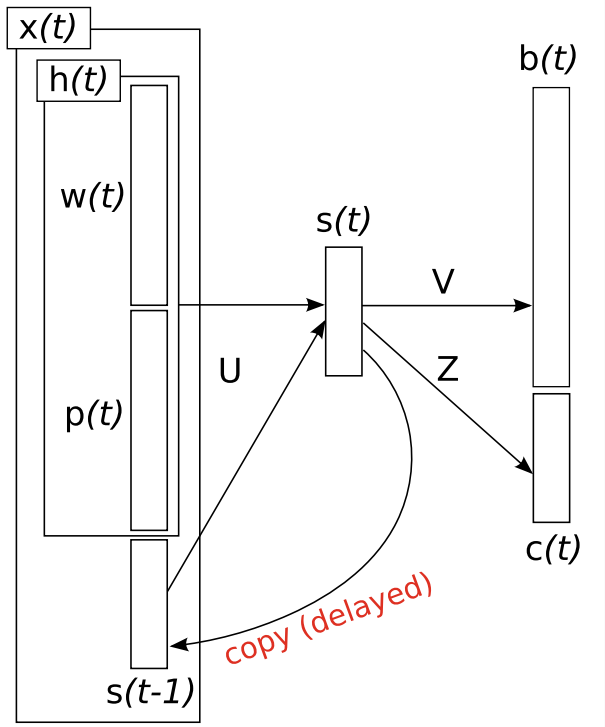

Fig. 4 Hybrid-RNNLM. Hybrid RNNLM architecture

fixed-length context in an NNLM or an n-gram LM. To be able to model multiple-type input units in the proposed hybrid RNNLM, some modifications are made to a conventional RNNLM framework. The model structure of the hybrid RNNLM is illustrated in Section 3.1. The input vector of our proposed hybrid RNNLM is a concatenated vector of word and PM vectors. A hybrid input representation and its variations are described in Section 3.2.

3.1 A model framework

The architecture of the proposed RNNLM is demonstrated in Fig. 4. The network is represented by three layers (input layer, hidden layer and output layer) and corresponding weight matrices (matrix **U** between the input layer and the hidden layer, and matrices **V** and **Z** between the hidden layer and the output layer). In this work, we employ a standard class-based RNNLM [14], where word classes are introduced in the output layer to reduce the computational bottleneck between the hidden layer and the output layer. Each word is assigned to exactly one class based on its frequency in training data. A class can be considered as a frequency bin.

Unlike a conventional word-based RNNLM, the input vector $\mathrm{x}(t)$ of the hybrid RNNLM is formed by concatenating a hybrid vector $\mathrm{h}(t)$, instead of a word vector $\mathrm{w}(t)$, with a vector $\mathrm{s}(t-1)$ as represented by the following equations:

$$\mathrm{x}(t) = \left[\mathrm{h}(t)^{\mathrm{T}}\ \mathrm{s}(t-1)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{1}$$

$$\mathrm{h}(t) = \left[\mathrm{w}(t)^{\mathrm{T}}\ \mathrm{p}(t)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{2}$$

where $\mathrm{h}(t)$ is a concatenated vector of a word vector $\mathrm{w}(t)$ and a PM vector $\mathrm{p}(t)$, and $\mathrm{s}(t-1)$ is the output from the hidden layer at time $t-1$. By using the hybrid vector, the hybrid RNNLM can simultaneously integrate multiple input types in its input layer.

The hidden layer compresses the information from two sources, $\mathrm{h}(t)$ and $\mathrm{s}(t-1)$ and computes a new context representation $\mathrm{s}(t)$ which will be an input of the next iteration through recurrent connections. The hidden layer employs a sigmoid activation function:

$$s_j(t) = f\left(\sum_i x_i(t)u_{ji}\right), \quad f(a) = \frac{1}{1+e^{-a}}, \tag{3}$$

where u_{ji} is an element in matrix **U**, i is an index to hybrid units in the the input vector $\mathrm{x}(t)$ and j is an index to hidden neurons in the hidden layers $\mathrm{s}(t)$.

In a class-based RNNLM, the output probability is factorized into two parts, the first part is the probability distribution over all classes, $\mathrm{c}(t)$, and the second part is the probability distribution over all hybrid units, $\mathrm{b}(t)$, in a single class, the one that contains the predicted hybrid unit:

$$c_m(t) = g\left(\sum_j s_j(t)z_{mj}\right) \tag{4}$$

$$b_k(t) = g\left(\sum_j s_j(t)v_{kj}\right), \tag{5}$$

where z_{mj} and v_{kj} are an element in matrix **Z** and **V**, respectively. To ensure that all output values are between 0 and 1, and their summation is equal to 1, the output layer employs a softmax activation function:

$$g(a_q) = \frac{e^{a_q}}{\sum_p e^{a_p}} \tag{6}$$

After the probability distribution over all classes and the probability distribution of hybrid units within the class are obtained, the probability of the predicted hybrid unit h_i is then computed as

$$P(h_i|history) = P(h_i|c_i, \mathrm{s}(t))\ P(c_i|\mathrm{s}(t)), \tag{7}$$

where i is an index of the predicted hybrid unit h_i and c_i is its class.

3.2 Input features for model training

In this section, we discuss a representation of RNNLM input features and a vocabulary list. As a hybrid RNNLM can take two types of input units, the training text consists of two sets: a word sequence set and a PM sequence set. Both data sets share the same content, but have different segmentations. Figure 5 illustrates various segmentation types of the same text utterance /phōn - thō: - sùʔ - wít - thā:n - ráʔ - rû:p/. The same text is segmented into a sequence of words in (a) and a sequence of PMs in (b).

After specifying training data, a vocabulary list is constructed. Typically, not every word found in the training data is included in the vocabulary list as low frequency words could be typos. In practice, only the top-N most frequent words are included. In a hybrid RNNLM, each input type has its own set of vocabulary. Let N be the size of word vocabulary and M be the size of PM vocabulary. The vocabulary size of a hybrid word-PM RNNLM is $N + M$ which could be twice the size of the word-based RNNLM as shown in Fig. 6a. The size of the vector h(t) in Eq. 1 is equal the vocabulary size. The hybrid RNNLM which uses a full vocabulary of both word and PM similar to [7], or a *full-hybrid* RNNLM, may suffer from the cost of computational complexity. Moreover, as our hybrid RNNLM also has a hybrid output, the output layer b(t) has the same dimensionality as h(t). As the output layer contains one neuron for each word or PM in the vocabulary, it may be infeasible to train the model with large vocabulary size.

To decrease computational complexity of the full-hybrid RNNLM (H-F), two variations of *reduced-hybrid* RNNLMs are proposed. In the first variation (H-R1), the vocabulary size is reduced to be equal to the size of the word-based RNNLM vocabulary (N) by including only frequent words and PMs as shown in Fig. 6a. Let N' and M' be the size of the top-N' most frequent words and the top-M' most frequent PMs, respectively, $N' + M' = N$ in H-R1. In the second variation (H-R2), a hybrid word-PM sequence is used as an input instead of two separate word and PM sequences as shown in Fig. 6b. Its vocabulary size is also limited to N. The difference from H-R1 is the composition of the vocabulary. Since the input text of H-R2 is a mix of words and PMs, the top-N'' most frequent words are determined first and kept as word units in the vocabulary. Next, the less frequent words are segmented into PMs. The top-M'' most frequent PMs from this list are then added into the vocabulary where $N'' + M'' = N$. Figure 5c illustrates a hybrid sequence where the first word /phōn - thō:/ (lieutenant general), which is a frequent word, is kept as a word while the second and third word /sùʔ - wít/ (first name) and /thā:n - ráʔ - rû:p/ (last name), which are infrequent words, are segmented into a set of PMs { /sùʔ/, /wít/ } and { /thā:n - ráʔ/, /rû:p/ }, respectively. If $N' = N''$ then word units in H-R1 and H-R2 are the same. However, PM units are different as the list of PMs and their frequency in H-R2 come from the less frequent words not all words in the training data as in H-R1. In both H-R1 and H-R2, the size of the vector h(t) is decreased. The reduction in computational time comparing with the full-hybrid RNNLM is discussed in Section 5.5.

4 Lattice decoding and re-scoring

To perform automatic speech recognition with the proposed hybrid RNNLM, we employ a two-pass decoding scheme. In the first-pass decoding, a hybrid n-gram LM similar to [4] is utilized to create a hybrid n-best list where OOV words could be recognized as a sequence of PMs. In the second-pass, a hybrid RNNLM which can consider different levels of linguistic information from different types of input units, i.e., word and PM, together is then applied to re-score the hybrid n-best list to improve recognition accuracy. The two recognition steps are discussed in detail below.

4.1 First-pass decoding with a hybrid n-gram LM

Conventionally, in the first pass, a decoder uses an acoustic model and a word-based n-gram language model to generate multiple recognition hypotheses which can be

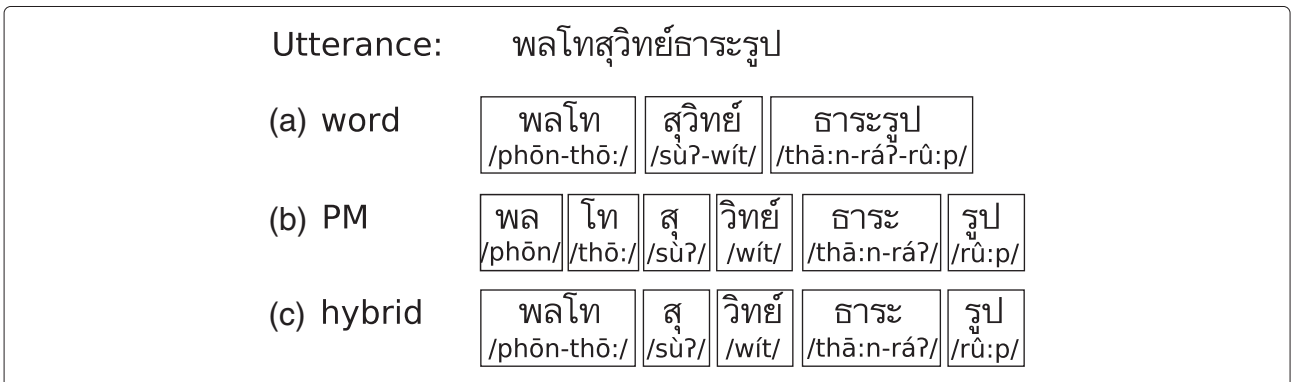

Fig. 5 Thai-segmentation. An illustration of different unit segmentations (**a**) word, (**b**) PM, and (**c**) hybrid

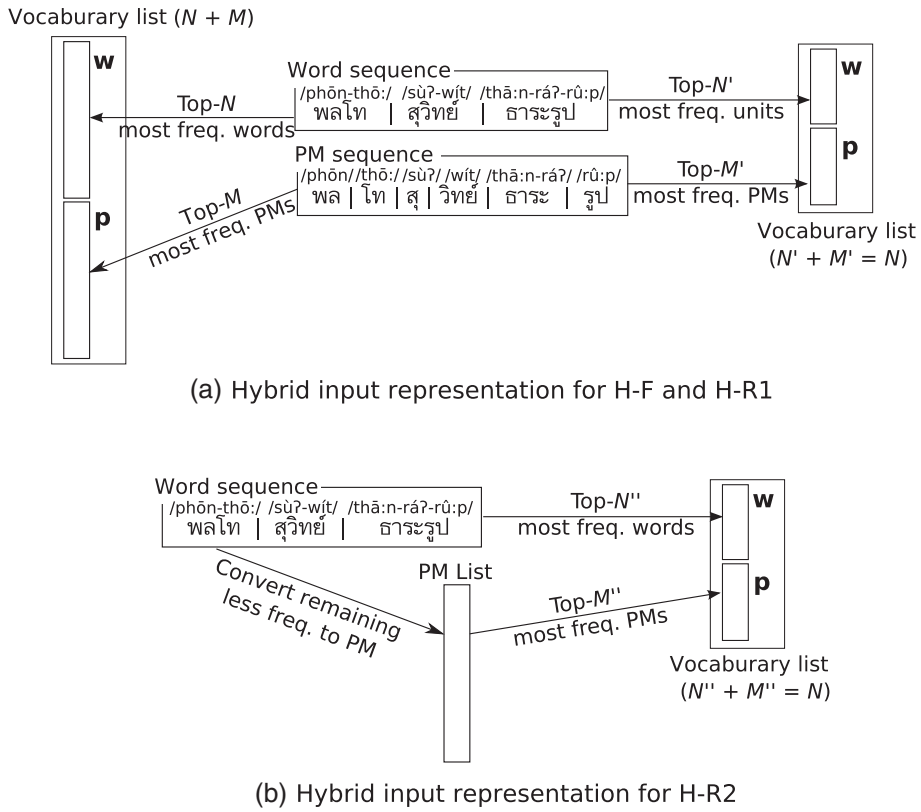

Fig. 6 Input-RNNLM. Input feature representations of 3 hybrid RNNLM variations (**a**) hybrid input representation for H-F and H-R1, and (**b**) hybrid input representation for H-R2

compactly represented in a data structure called word lattice as illustrated in Fig. 7a. In this paper, a hybrid n-gram LM is used instead in order to handle OOV words. To train a hybrid n-gram LM, a hybrid sequence of words and PMs similar to the one in Fig. 5c is used as LM training data. The vocabulary of our hybrid LM consists of frequent words and PMs from less frequent words. Words that occur more frequently than a threshold are kept as word units in the vocabulary while words that occur less often are segmented into PMs. All unique PMs are then added to the hybrid vocabulary. We note that some PMs could be similar to short words in the vocabulary. To avoid redundancy, these PMs are excluded.

In Fig. 7a, the utterance /phōn - thō: - sù? - wít - thā:n - rá? - rû:p/ cannot be recognized by the word-based lattice as the word /thā:n - rá? - rû:p/ is an OOV word. When a

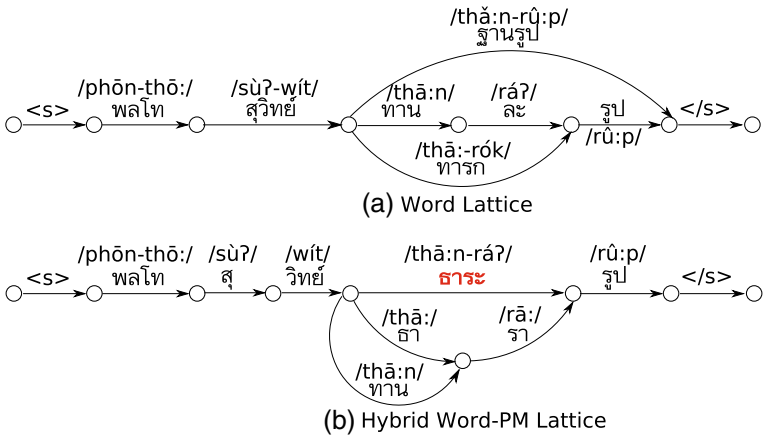

Fig. 7 Lattice. Word and hybrid lattices in the first-pass decoding (**a**) word lattice, and (**b**) hybrid word-PM lattice

hybrid n-gram LM is used to generate a hybrid lattice as illustrated in Fig. 7b, the OOV word /thāːn - ráʔ - rûːp/ can be recognized as a sequence of PMs /thāːn - ráʔ/ and /rûːp/.

4.2 Second-pass re-scoring with an RNNLM

In multi-pass decoding, an LM trained with higher order knowledge sources is used to re-score the lattice generated by an LM trained with simpler or lower order knowledge sources from the preceding pass. In this work, an RNNLM which is considered more complex is applied to re-score a hybrid lattice generated by the hybrid n-gram LM described in the previous section. Since the hybrid RNNLMs described in Section 3.2 use different input feature representations, the lattice generated from the first-pass decoding has to be converted accordingly. An n-best hypothesis list is first extracted from the lattice, then converted into a suitable representation for each hybrid RNNLM as shown in Fig. 8. A conventional n-best list extracted from a word-based lattice is also presented in Fig. 8a for comparison. For the full-hybrid RNNLM which takes both a word sequence and a PM sequence as its input, a PM sequence is obtained by splitting all words in the hypotheses into PMs as shown in Fig. 8b. The acoustic score and LM score of each PM are calculated by a uniform distribution of the corresponding word scores. When the hypotheses are extracted from a hybrid lattice, a word sequence may contain some PMs similar to a hybrid sequence in Fig. 8c. The reduced-hybrid H-R1 which takes the same input representation as H-F but use a reduced vocabulary set also uses the word-PM n-best list shown in Fig. 8b. For the reduced-hybrid H-R2 where frequent words are represented as word units while infrequent words are represented as PM units, a hybrid n-best list illustrated in Fig. 8c is used in the second-pass re-scoring. In this case, it can be seen that only an infrequent word /nâː - tìt - tāːm/ is segmented to PMs as /nâː/, /tìt/ and /tāːm/.

In the second-pass re-scoring, a new LM score is obtained by interpolating the probability from RNNLM with the first-pass n-gram LM score. Then, the hybrid n-best list with the new LM scores is reconstructed into a hybrid lattice. Finally, the new best hypothesis is chosen based on the re-scored score. For instance, after re-scoring with the hybrid RNNLM (H-R2), the 2-best hypothesis in Fig. 8c, which is the correct one, becomes the 1-best hypothesis and is chosen as a recognition result.

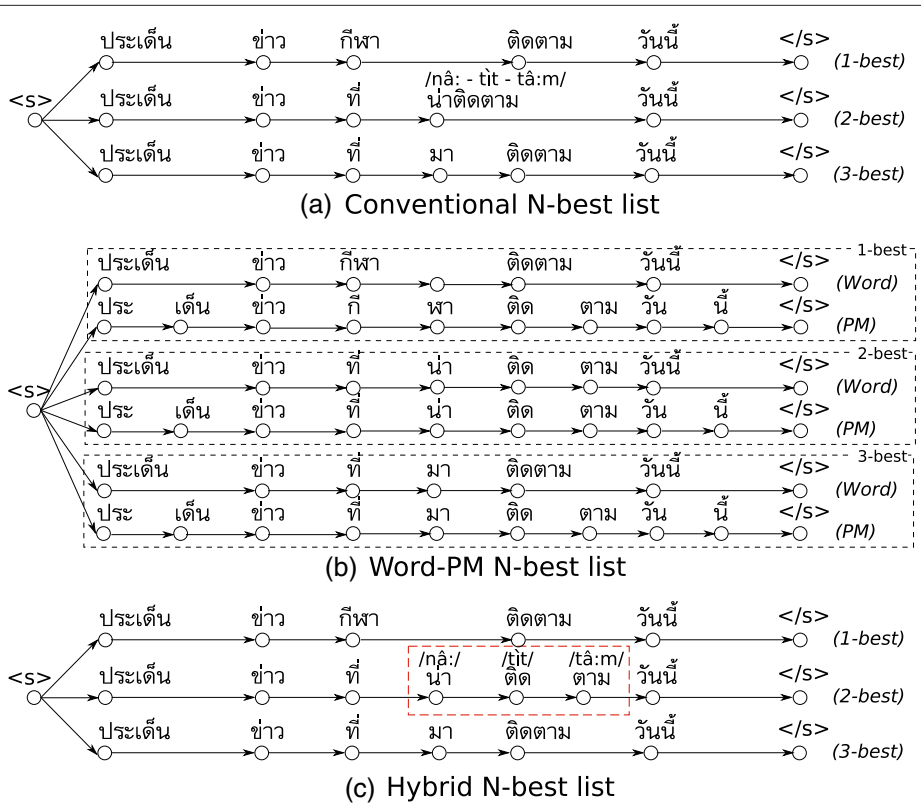

Fig. 8 N-best. N-best list representations in the second-pass re-scoring of different types (**a**) conventional N-best list, (**b**) word-PM N-best list, and (**c**) hybrid N-best list

5 Experiments

We evaluated the performance of the proposed hybrid RNNLMs on both recognition accuracy and computational efficiency. Training and test data along with the experimental conditions are described in Section 5.1. Recognition accuracy of the first-pass decoding, the second-pass re-scoring together with the experimental analysis are reported in Sections 5.2, 5.3, and 5.4, respectively, while the model run-time efficiency is discussed in Section 5.5.

5.1 Experimental conditions

We evaluated our approach with two different recognition tasks: broadcast news transcription and speech-to-speech translation. Acoustic model training data of our speech recognizer composes of 224 hours of speech from LOTUS [15], LOTUS-BN [11], and VoiceTra4U-M. VoiceTra4U-M is a speech translation application in sport and travel domains developed under the Universal Speech Translation Advanced Research (U-STAR) project (http://www.ustar-consortium.com/qws/slot/u50227/index.html). Twenty-two hours of speech were recorded on mobile devices in real environment. We used the Kaldi Speech Recognition Toolkit [16] to first train a conventional GMM-based acoustic model. We then applied the minimum phone error (MPE) discriminative training technique described in [17]. Each frame of speech data was converted into a sequence of 39 dimensional feature vectors of 12 MFCCs augmented with log energy, their first and second derivatives. We used a 25 millisecond frame length with 10 millisecond window shift each time. Features from a context window of 3 frames to the left and right were also included. A linear discriminate analysis (LDA) was also applied to the feature space to reduce feature dimensions to 40.

LM training data contain 9.4M words from three corpora, BEST [10], LOTUS-BN [11], and HIT-BTEC [12]. As these corpora cover variety of domains, e.g. law, news, and travel, with the vocabulary size of 121K, they are good resources for training a hybrid LM for an open-domain LVCSR system.

The test set of the broadcast news transcription task (BN) consists of 3140 utterances of two male speakers and one female speaker taken from the LOTUS-BN evaluation set. For the speech-to-speech translation task (VT), the test set consists of 1916 utterances of VoiceTra4U-M data not included in the training set.

5.2 Recognition performance of the first-pass decoding

Three 3-gram LMs were experimented to investigate the effect of different lexical units on the first-pass decoding performance of a Thai LVCSR system. A word-based LM (f-W) is a baseline LM which includes all word units in the training data in its lexicon. A PM-based LM (f-PM) uses PM as a lexicon unit instead of word. The training data were segmented into PMs and then used to train an n-gram model in the same way as a word-based LM. All PM units were included in the f-PM lexicon. A hybrid LM (f-H) uses a hybrid lexicon which includes only the words that occur more than three times in the training data and the PMs from less frequent words. The amounts of word and PM units used in each LM are shown in the second and third row, respectively, in Table 1. All 3-gram LMs were trained using Kaldi LM toolkit [16] with modified Kneser-Ney smoothing.

The performance of the 3-gram LMs in the first-pass decoding is measured in terms of accuracy and OOV rate. Since mixed types of units are used in the LMs, we report both *PM Error Rate (PER)* and *Word Error Rate (WER)* in all experiments. To recover word units from a hybrid output, we concatenate multiple-type output units into a string of text and then re-segment it into words using a word segmentation tool. OOV rate (OOV) is used to measure the coverage of a given lexicon over a test set. OOV reported in this section is an effective OOV rate which accounts for the percentage of words that are not in the full-word lexicon and cannot be constructed from the PMs.

From the results shown in Table 1, we can clearly see that the hybrid LM, f-H, can greatly reduce the OOV rates when compared with f-W in both test sets by using only 42 % of the vocabulary size (51 K vs. 121 K). In terms of PER and WER, among the 3 LMs, f-H achieved the best results on both test sets. On average, the hybrid LM can reduce PER by 0.21 % absolute and 1.02 % relative while reduce WER by 0.16 % absolute and 0.69 % relative when compared with the word-based LM, f-W. The PM-based, f-PM, has the lowest OOV rates on both test sets. However, it produced the highest error rates, both PER and WER. From the result, we can say that PM units are too small to capture context history and language dependencies especially with the fixed context length in an n-gram

Table 1 Recognition results of first-pass decoding

	LMs	f-W	f-PM	f-H
Tasks	#Word	121 K	0 K	30 K
	#PM	0 K	25 K	21 K
BN	PER (%)	20.64	23.76	20.54
	WER (%)	24.05	27.30	24.01
	OOV (%)	2.08	0.41	0.54
VT	PER (%)	19.71	20.99	19.40
	WER (%)	22.56	23.94	22.28
	OOV (%)	0.85	0.11	0.15
AVG	PER (%)	20.18	22.38	19.97
	WER (%)	23.31	25.62	23.15
	OOV (%)	1.96	0.39	0.51

model. This experiment shows that the hybrid lexicon of words and PMs is a resource-efficient representation that not only can greatly reduces the OOV rate but also improves recognition accuracy. Examples of correctly recognized OOV words, such as proper name and loan word, are shown in Fig. 3.

5.3 Recognition performance of the second-pass re-scoring

Five variations of RNNLMs were investigated: word-based (W), PM-based (PM), full-hybrid (H-F), and two variations of reduced-hybrid (H-R1 and H-R2). The PM RNNLM was trained in the same fashion as the conventional word RNNLM except that the input unit is PM instead of word. In all experiments, the class-based RNNLMs were trained by the RNNLM toolkit [18] with five iterations of Back-propagation through time (BPTT) [19], 400 hidden neurons, and 400 classes. In the training, 1 % or 10K words were excluded from the training set for validation testing. The size of the n-best list obtained from the first-pass decoding is set to be at most 100 hypotheses for each utterance. In the second-pass re-scoring, RNNLMs were interpolated with the 3-gram LM using a weight of 0.25 for RNNLMS.

We first examined the appropriate hybrid vocabulary for H-R1 and H-R2, namely the number of word units (N' and N'') and the number of PM units (M' and M''). In the first-pass decoding experiment discussed in Section 5.2, all PMs from less frequent words were included in the hybrid 3-gram model. Since an RNNLM requires much larger training resources, only frequent PMs should be included in the vocabulary. Three hybrid lexicons with different amounts of words and PMs were experimented. For fair comparison, the sizes of the lexicons are kept the same at 35 K units. The hybrid lexicons include words that occur more than three, four, and seven times and PMs that occur more than five, three, and two times, respectively. The amounts of corresponding words and PMs in each hybrid lexicon are shown in the second and third rows in Table 2. The effect of different word/PM ratios in the hybrid lexicons in terms of PER and WER in the second-pass re-scoring are reported.

For H-R1, on average, the best result was obtained from the vocabulary which contains 30 K words and 5 K PMs. For H-R2, the best result was obtained from the vocabulary which contains 25 K words and 10 K PMs. We note that both PER and WER do not change much with different word/PM ratios. Nevertheless, we chose the ratio that yields the best recognition performance for the next experiment. For comparison, the vocabulary size of the word-based RNNLM (N) is set to 35 K while the vocabulary size of the PM-based RNNLM (M) is set to 25 K, the amount of all PM units. For H-F, the vocabulary size is $N + M$ which is 60 K as detailed in Section 3.2.

Table 3 shows recognition results of the second-pass re-scoring using a hybrid 4-gram LM (s-H) and five variations of RNNLMs. As expected, all the second-pass re-scoring results were better than the first-pass result which used a hybrid 3-gram LM (f-H). When compared RNNLMs with a hybrid 4-gram LM in the second-pass re-scoring, all RNNLMs obtained better recognition results as they can capture longer context history than the 4-gram model. Among various RNNLMs, H-F achieved the lowest error rate in the BN test set while H-R2 gave the best recognition result in the VT test set and also on average. H-R2 is preferable as it is more computational efficient as discussed in Section 5.5. When compared with a conventional word-based RNNLM (W), the best proposed hybrid RNNLM, H-R2, obtained 2.41 % relative PER reduction and 1.54 % relative WER reduction on average. The performance improvements of the proposed hybrid input RNNLM over the traditional word-based RNNLM are statistically significant at the 0.01 (1 %) level for PER and at the 0.05 (5 %) level for WER.

From result analysis, we found that the word-based RNNLM sometimes made mistake by choosing a long word or a compound word when its can be acoustically confused with correct words in an input utterance. A hybrid word-PM RNNLM, on the other hand, has more

Table 2 Recognition performance with various hybrid lexicons

Tasks	LM	H-R1			H-R2		
	#Word	30 K	25 K	20 K	30 K	25 K	20 K
	#PM	5 K	10 K	15 K	5 K	10 K	15 K
BN	PER (%)	18.84	18.85	18.89	18.84	18.81	18.86
	WER (%)	22.16	22.19	22.24	22.15	22.13	22.18
VT	PER (%)	18.79	18.86	18.76	18.41	18.41	18.56
	WER (%)	21.57	21.66	21.57	21.27	21.27	21.47
AVG.	PER (%)	18.82	18.86	18.83	18.63	18.61	18.71
	WER (%)	21.87	21.93	21.91	21.71	21.70	21.83

Table 3 Recognition performance of second-pass re-scoring

Tasks	LM	4gr	RNNLMs				
		s-H	W	PM	H-F	H-R1	H-R2
	#Word	30 K	35 K	0 K	35 K	30 K	25 K
	#PM	21 K	0 K	25 K	25 K	5 K	10 K
BN	PER (%)	19.60	19.11	19.12	18.73	18.84	18.81
	WER (%)	22.92	22.22	22.23	22.05	22.16	22.13
VT	PER (%)	19.02	19.02	19.26	18.89	18.79	18.41
	WER (%)	21.75	21.85	22.17	21.69	21.57	21.27
AVG.	PER (%)	19.31	19.07	19.19	18.81	18.82	18.61
	WER (%)	22.34	22.04	22.20	21.87	21.87	21.70

Table 4 Positive and negative effects on recognition results

Tasks	LM	RNNLMs	
		W	H-R2
BN	COOV (%)	-	21.38
	MIV (%)	21.87	21.84
VT	COOV (%)	-	35.14
	MIV (%)	21.62	21.03
AVG.	COOV (%)	-	21.81
	MIV (%)	21.85	21.77

flexible unit choices as it can output both word and sub-word units, and thus can avoid this kind of mistakes. With the use of RNN for combining information from different types of units, 6.81 % relative improvement on PER and 6.26 % on WER can be obtained compared with the hybrid n-gram LM (f-H).

5.4 Experimental analysis

To further analyze the recognition improvement achieved by the proposed hybrid framework, we also reported *COOV* (correctly recognized OOVs) and *MIV* (misrecognized in-vocabulary words) in Table 4. The best proposed hybrid RNNLM (H-R2), reported in Section 5.3, is compared against the conventional word-based RNNLM (W). In this experiment, a word is considered an OOV word if it is not found in the 121K full-vocabulary of the training data. The OOV rates of the BN task and the VT task are 2.08 and 0.85 %, respectively, while the average OOV rate is 1.96 % as shown in Table 1. The COOVs in Table 4 show that the proposed hybrid framework of word and PM units can alleviate the problem of OOV words by correctly recognized 21.38 % of them in the BN task and 35.14 % of them in the VT task while the traditional word-based RNNLM cannot. We also show examples of recognition results of the word-based system against the hybrid system in Fig. 9. Words and PMs are separated by "-" and "|", respectively. Words in the Fig. 9 are OOV words found in the test sets. These words could correctly be recognized by the proposed hybrid framework, but could not be recognized by the word-based RNNLM framework and thus introduce recognition errors. The first and second rows are person names while the last row is a Thai transliterated word of the word "alliance". Named-entities and transliterated words are known to be the main causes of OOV. By modeling an OOV word with a sequence of PMs, these OOV words could be correctly recognized by our hybrid RNNLM.

When consider in-vocabulary words, we found that the MIV of our proposed hybrid RNNLM is not higher than that of the word-based one, thus there is no negative effect due to lexical confusion from including PMs in the language model. Furthermore, the misrecognition of in-vocabulary words was reduced by 0.35 % relatively in the hybrid framework (H-R2) over the word-based framework (W). The improvement in in-vocabulary recognition could come from different levels of linguistic information embedded in multiple-type input units which can be utilized by the proposed hybrid RNNLM framework. From the analysis of results shown in Table 4, we could say that the improvement in WER and PER of the hybrid RNNLM (H-R2) over the word-based RNNLM (W) comes from both better COOV and MIV.

5.5 Computational efficiency

In this section, we analyzed the computational time of the RNNLMs. Table 5 shows the amount of training time and the second-pass decoding time on a PC with 98 GB of memory and 24 cores 2.67 GHz CPU.

Since H-F and H-R1 used both word and PM sequences as an input when trained the models, they used much longer training time than other types of RNNLMs. Their decoding times are also almost twice when compared with other models. H-R2 which takes a single hybrid sequence of words and PMs as its input has the lowest training and decoding time among all RNNLM variations. When compared among three hybrid RNNLM variations, both the training and decoding time can be saved by more than half in H-R2 without affecting recognition accuracy. When compared with the 4-gram LM, which has about 6 min training time and 17 min decoding time, all RNNLMs have much longer training time but have faster decoding time while also achieve better recognition results. The 4-gram LM has longer decoding time due to its larger vocabulary size.

OOV words	Word-based RNNLM (W)	Hybrid RNNLM (H-R2)
จารี /c ā: r ī:/	จาก - ปี /c à: k - p ī:/	จา | รี /c ā: | r ī:/
เพียรจิตร /ph ī:a n c ì t/	เพียง - เจ็ด /ph ī:a ŋ - c è t/	เพียร | จิตร /ph ī:a n | c ì t/
อัลลายแอนซ์ /ʔ ā n l ā: j ʔ x̄ n/	อะไร - อัน /ʔ à ʔ r ā j - ʔ ā n/	อัล | ลาย | แอนซ์ /ʔ ā n | l ā: j | ʔ x̄ n/

Fig. 9 Recog Results. Recognized results samples of OOVs from word-based and hybrid RNNLM

Table 5 Training and decoding time (h=hour, m=minute, s=second)

RNNLMs	W	PM	H-F	H-R1	H-R2
#Word	35 K	0 K	35 K	30 K	25 K
#PM	0 K	25 K	25 K	5 K	10 K
Training	36 h 35 m	38 h 13 m	67 h 19 m	63 h 12 m	28 h 20 m
Decoding	08 m 25 s	10 m 22 s	17 m 47 s	15 m 48 s	08 m 15 s

6 Conclusions

We proposed a hybrid RNNLM framework for modeling multiple-type input units, namely word and sub-word. A hybrid lexicon was utilized to alleviate the problem of OOV words and to improve recognition accuracy through additional linguistic information from multiple unit types. Pseudo-morpheme (PM), a syllable-based unit, was chosen as an appropriate sub-word unit for Thai. A concatenated vector of word and PM vectors, or a hybrid vector, is used as an input vector instead of a word vector in the proposed hybrid RNNLM framework. Several hybrid input representations were also explored to optimize both recognition accuracy and computational time.

The hybrid LM has shown to be both resource-efficient and well-performed on two Thai LVCSR tasks: broadcast news transcription and speech-to-speech translation. The proposed hybrid lexicon can constitute an open vocabulary for Thai LVCSR as it can greatly reduce the OOV rate to less than 1 % while using only 42 % of the vocabulary size of the word-based lexicon. In terms of recognition performance, the best proposed hybrid RNNLM, a reduced-hybrid which uses a mixed input sequence of words and PMs, obtained 1.54 % relative WER reduction when compared with a conventional word-based RNNLM as hybrid input types provide more flexible unit choices for LM re-scoring. The improvement obtained with the proposed hybrid input RNNLM is statistically significant at the 0.05 level. Furthermore, the hybrid RNNLM framework has shown to be able to alleviate the problem of OOV words by correctly recognize 21.81 % of OOV words compared to the word-based system on the evaluation corpora. When only frequent words and PMs from less frequent words are used, the size of the input vector of the reduced-hybrid RNNLM can be reduced by half when compared with the full-hybrid RNNLM which takes two input streams, both word and PM sequences. The hybrid input representation can considerably save both training and decoding time while still achieving slightly better recognition accuracy. In the future, we plan to apply more complex lattice re-scoring algorithms, such as the one described in [20], to the hybrid RNNLM to further improve recognition performance. A cache-based RNNLM which can be used straightaway in the first-pass decoding [21] will also be considered.

Competing interests
The authors declare that they have no competing interests.

Received: 2 February 2016 Accepted: 19 July 2016
Published online: 08 August 2016

References

1. A El-Desoky, C Gollan, D Rybach, R Schlüter, H Ney, in *INTERSPEECH*. Investigating the use of morphological decomposition and diacritization for improving Arabic LVCSR, (2009), pp. 2679–2682. https://scholar.google.co.th/scholar_lookup?title=Investigating+the+use+of+morphological+decomposition+and+diacritization+for+improving+Arabic+LVCSR&btnG=.
2. MAB Shaik, AE-D Mousa, R Schlüter, H Ney, in *INTERSPEECH*. Hybrid language models using mixed types of sub-lexical units for open vocabulary German LVCSR, (2011), pp. 1441–1444. https://scholar.google.com/scholar_lookup?title=Hybrid+language+models+using+mixed+types+of+sublexical+units+for+open+vocabulary+German+LVCSR&btnG=&hl=th&as_sdt=0%2C5.
3. M Rondeau, R Rose, in *Information Science, Signal Processing and Their Applications (ISSPA), 2012 11th International Conference On*. Developing a hybrid language model for open vocabulary automatic speech recognition in a lecture speech task (IEEE, 2012), pp. 114–119. https://scholar.google.co.th/scholar_lookup?title=Developing+a+hybrid+language+model+for+open+vocabulary+automatic+speech+recognition+in+a+lecture+speech+task&btnG=&hl=th&as_sdt=0%2C5.
4. K Thangthai, A Chotimongkol, C Wutiwiwatchai, in *INTERSPEECH*. A hybrid language model for open-vocabulary Thai LVCSR, (2013), pp. 2207–2211. https://scholar.google.com/scholar_lookup?title=A+hybrid+language+model+for+openvocabulary+Thai+LVCSR&btnG=&hl=th&as_sdt=0%2C5.
5. Y He, B Hutchinson, P Baumann, M Ostendorf, E Fosler-Lussier, J Pierrehumbert, in *ICASSP*. Subword-based modeling for handling OOV words inkeyword spotting (IEEE, 2014), pp. 7864–7868. https://scholar.google.co.th/scholar_lookup?title=Subwordbased+modeling+for+handling+OOV+words+inkeyword+spotting&btnG=&hl=th&as_sdt=0%2C5.
6. X Liu, JL Hieronymus, MJ Gales, PC Woodland, Syllable language models for Mandarin speech recognition: Exploiting character language models. J. Acoust. Soc. Am. **133**(1), 519–528 (2013)
7. M Kang, T Ng, L Nguyen, in *INTERSPEECH*. Mandarin word-character hybrid-input neural network language model, (2011), pp. 625–628. https://scholar.google.com/scholar_lookup?title=Mandarin+word-character+hybridinput+neural+network+language+model&btnG=&hl=th&as_sdt=0%2C5.
8. M Jongtaveesataporn, I Thienlikit, C Wutiwiwatchai, S Furui, Lexical units for Thai LVCSR. Speech Comm. **51**(4), 379–389 (2009)
9. T Mikolov, M Karafiát, L Burget, J Cernocký, S Khudanpur, in *INTERSPEECH*. Recurrent neural network based language model, (2010), pp. 1045–1048. https://scholar.google.com/scholar_lookup?title=Recurrent+neural+network+based+language+model&btnG=&hl=th&as_sdt=0%2C5.
10. K Kosawat, M Boriboon, P Chootrakool, A Chotimongkol, S Klaithin, S Kongyoung, K Kriengket, S Phaholphinyo, S Purodakananda, T Thanakulwarapas, C Wutiwiwatchai, in *SNLP*. BEST 2009: Thai word segmentation software contest, (2009), pp. 83–88. https://scholar.google.com/scholar_lookup?title=BEST+2009%3A+Thai+word+segmentation+software+contest&btnG=&hl=th&as_sdt=0%2C5.
11. A Chotimongkol, K Saykhum, P Chootrakool, N Thatphithakkul, C Wutiwiwatchai, in *Oriental COCOSDA*. LOTUS-BN: A Thai broadcast news corpus and its research applications, (2009), pp. 44–50. https://scholar.google.com/scholar_lookup?title=LOTUS-BN%3A+A+Thai+broadcast+news+corpus+and+its+research+applications&btnG=&hl=th&as_sdt=0%2C5.
12. G Kikui, E Sumita, T Takezawa, S Yamamoto, in *INTERSPEECH*. Creating corpora for speech-to-speech translation, (2003). https://scholar.google.com/scholar_lookup?title=Creating+corpora+for+speech-tospeech+translation&btnG=&hl=th&as_sdt=0%2C5.
13. W Aroonmanakun, in *Oriental COCOSDA*. Collocation and Thai word segmentation, (2002), pp. 68–75. https://scholar.google.com/scholar_lookup?title=Collocation+and+Thai+word+segmentation&btnG=&hl=th&as_sdt=0%2C5.

14. T Mikolov, S Kombrink, L Burget, J Cernocký, S Khudanpur, in *ICASSP*. Extensions of recurrent neural network language model, (2011), pp. 5528–5531. https://scholar.google.com/scholar_lookup?title=Extensions+of+recurrent+neural+network+language+model&btnG=&hl=th&as_sdt=0%2C5.
15. S Kasuriya, V Sornlertlamvanich, P Cotsomrong, S Kanokphara, N Thatphithakkul, in *Oriental COCOSDA*. Thai speech corpus for Thai speech recognition, (2003), pp. 54–61. https://scholar.google.com/scholar_lookup?title=Thai+speech+corpus+for+Thai+speech+recognition&btnG=&hl=th&as_sdt=0%2C5.
16. D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlicek, Y Qian, P Schwarz, J Silovsky, G Stemmer, K Vesely, in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. The Kaldi speech recognition toolkit, (2011). https://scholar.google.com//scholar_lookup?title=The+Kaldi+speech+recognition+toolkit.&btnG=&hl=th&as_sdt=0%2C5.
17. D Povey, PC Woodland, in *ICASSP*. Minimum phone error and I-smoothing for improved discriminative training, vol. 1, (2002), pp. 105–108. https://scholar.google.com/scholar_lookup?title=Minimum+phone+error+and+Ismoothing+for+improved+discriminative+training&btnG=&hl=th&as_sdt=0%2C5.
18. T Mikolov, S Kombrink, A Deoras, L Burget, JH Cernocky, in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. RNNLM - recurrent neural network language modeling toolkit, (2011), pp. 1–4. https://scholar.google.com/scholar_lookup?title=RNNLM+-+recurrent+neural+network+language+modeling+toolkit&btnG=&hl=th&as_sdt=0%2C5.
19. DE Rumelhart, GE Hinton, RJ Williams, in *Neurocomputing: Foundations of Research*. Learning representations by back-propagating errors, (1988), pp. 696–699. https://scholar.google.com/scholar_lookup?title=Learning+representations+by+backpropagating+errors&btnG=&hl=th&as_sdt=0%2C5.
20. X Liu, Y Wang, X Chen, MJF Gales, PC Woodland, in *ICASSP*. Efficient lattice rescoring using recurrent neural network language models, (2014), pp. 4908–4912. https://scholar.google.com/scholar_lookup?title=Efficient+lattice+rescoring+using+recurrent+neural+network+language+models&btnG=.
21. Z Huang, G Zweig, B Dumoulin, in *ICASSP*. Cache based recurrent neural network language model inference for first pass speech recognition, (2014). https://scholar.google.com/scholar_lookup?title=Cache+based+recurrent+neural+network+language+model+inference+for+first+pass+speech+recognition&btnG=&hl=th&as_sdt=0%2C5&scioq=Learning+representations+by+backpropagating+errors.