INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Joint Maximization Decoder with Neural Converters for Fully Neural Network-based Japanese Speech Recognition

Takafumi Moriya[1], *Jian Wang*[1,2], *Tomohiro Tanaka*[1], *Ryo Masumura*[1], *Yusuke Shinohara*[1], *Yoshikazu Yamaguchi*[1], *Yushi Aono*[1]

[1]NTT Media Intelligence Laboratories, NTT Corporation, Japan
[2]Graduate School of Engineering, The University of Tokyo, Japan

takafumi.moriya.nd@hco.ntt.co.jp

Abstract

We present a novel fully neural network (FNN) -based automatic speech recognition (ASR) system that addresses the out-of-vocabulary (OOV) problem. The most common approach to the OOV problem is leveraging character/sub-word level units as output symbols. Unfortunately, this approach is not suitable for Japanese and Mandarin Chinese since they have many more grapheme sets than English. Our solution is to develop FNN-based ASR that uses a pronunciation-based unit set with dictionaries, i.e., word-to-pronunciation rules. A previous study proposed, for Mandarin Chinese, a greedy cascading decoder (GCD) that uses two neural converters, acoustic-to-pronunciation (A2P) and pronunciation-to-word (P2W) conversion models. However, to generate optimal word sequences, the previous work considered just optimal pronunciation sequences. In this paper, we propose a joint maximization decoder (JMD) that considers the joint probability of pronunciation and word in beam-search decoding. Moreover, we introduce a neural network based joint source channel model for improving A2P conversion performance. Experiments on Japanese ASR tasks demonstrate that JMD achieves better performance than GCD. Furthermore, we show the effectiveness of using just language resources to retrain the P2W conversion model.

Index Terms: ASR, neural network, encoder-decoder, joint source channel model, syllable based acoustic modeling

1. Introduction

Recent automatic speech recognition (ASR) systems can map acoustic features to word sequences directly; called acoustic-to-word (A2W) end-to-end ASR, the approach is based on a fully neural network (FNN) -based architecture [1–8]. Unfortunately, end-to-end ASR systems are not robust to out-of-vocabulary (OOV) words because the number of NN outputs, which correspond to word entries, is fixed.

The OOV problem degrades recognition performance and is particularly damaging to end-to-end ASR systems. Several studies have examined the OOV problem in end-to-end ASR systems. As one solution, English studies [9–12] use characters, word-piece and subword units [13, 14]. We note that multi-level long short-term memory-language model (LSTM-LM) [15] has the same motivation, i.e. robustness to OOVs and utilizing external language resources [16, 17]. The alphabet characters can cover every word in English because all words are various combinations of the 26 alphabet characters. However, these conventional approaches are difficult to apply to Japanese texts, because, as we know, Japanese texts use various mixed-character (MC) sets, e.g. $Hiragana$ (ひらがな), $Katakana$ (カタカナ), $Kanji$ (漢字), and $Roman$ alphabets in a mixed manner [13]. Of particular interest, $Kanji$-characters have over 50,000 entries (the number of often used $Kanji$-characters is approximately 2,000) [18]. Therefore, there are many more characters in Japanese texts than, for example, graphemes in English and OOVs exist at the character-level in external language resources. This means that end-to-end ASR systems need to be re-trained using new data pairs of speech and text every time a new word/character (OOV) is used [7, 8]. We want a framework that makes it easy to extend the vocabulary of FNN-based systems by using just text-based language resources.

To realize the above framework, we focus on a sequence-to-sequence (Seq2Seq) -based encoder-decoder model that outputs only $Katakana$ ($Kana$) -based syllables, as it is the basic Japanese pronunciation-based unit set. It has the advantage of avoiding the OOV problem because pronunciation, i.e. $Kana$, can express all vocabularies by using a dictionary, i.e., word-to-pronunciation rules. In addition, $Kana$-based units are longer linguistic units, which reduce the difficulty faced by encoder-decoder models in choosing among output unit candidates in the decoder. In [19], they tackle the same problem in Mandarin Chinese, which also has many graphemes. For generating word sequences from the acoustic features, they proposed the greedy cascading decoder (GCD) using two Seq2Seq models for the conversion of acoustic-to-pronunciation (A2P) and pronunciation-to-word (P2W); the first Seq2Seq model calculates the best pronunciation sequence from acoustic features, and then the other transforms the best pronunciation sequence into a word sequence. The first step, A2P, is a good solution for languages that have a lot of characters. However, the second Seq2Seq model considers only the best pronunciation sequence of the A2P outputs. The hypotheses that are pruned in the first decoding step should be considered if better word sequences are to be identified.

In this paper, we propose a joint maximization decoder (JMD) that considers both A2P and P2W conversion scores in beam search decoding. JMD can simultaneously output the pronunciation and word sequences while considering both of their hypotheses. That is, JMD must calculate the joint probability of a word sequence and its pronunciation. To model the conversion needed for our framework, we introduce the joint source channel model [20] extended by using the LSTM-LM. JMD utilizes the benefits of the unidirectional LSTM-LM, which can work left-to-right, for beam search decoding. We expect that the decoder can successively receive P2W scores as the feedback for modifying the best hypothesis of A2P; it deals with not only the best pronunciation sequence but also the other hypotheses. Moreover, LSTM-LM can be trained using just text data, which makes it easy to extend the vocabulary by using external language resources.

http://dx.doi.org/10.21437/Interspeech.2019-1558

We evaluate our proposal on a corpus of spontaneous Japanese (CSJ) that is split into two sub corpora; these are regarded as external language resources, both of which include OOVs. The results demonstrate that our proposal, JMD, achieves better ASR performance than GCD and the conventional FNN-based ASR schemes such as A2W encoder-decoder model. Moreover, we also show that retraining the P2W conversion model from just language resources is highly effective.

2. Attention-based encoder-decoder model for automatic speech recognition

For achieving end-to-end ASR systems, several modeling methods have been studied recently [21–31]. In this paper, we use the attention-based encoder-decoder model [23–25] for an end-to-end ASR system. Our implementation of the model is based on [4, 8], and summarized in this section. The encoder-decoder model, which has trainable parameters $\mathbf{\Theta}_{\text{EncDec}} = \{\boldsymbol{\theta}_{\text{Enc}}, \boldsymbol{\theta}_{\text{Dec}}\}$, can handle input and output labels of different lengths; $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$ and $\boldsymbol{Y} = (y_1, ..., y_N)$. These indicate the input acoustic feature sequence of length-T and target label sequence of length-N, respectively. The encoder which has bidirectional LSTM transforms $\boldsymbol{X}$ into intermediate representation vectors $\boldsymbol{H} = (\boldsymbol{h}_1, ..., \boldsymbol{h}_T)$ as follows:

$$\boldsymbol{H} = \texttt{Encoder}(\boldsymbol{X}, \boldsymbol{\theta}_{\text{Enc}}). \tag{1}$$

Next, the attention-based decoder which has unidirectional LSTM successively generates the n-th output y_n as:

$$y_n = \texttt{AttenDecoder}(y_{n-1}, \boldsymbol{H}, \boldsymbol{\theta}_{\text{Dec}}). \tag{2}$$

The objective function for training the attention-based model parameter $\mathbf{\Theta}_{\text{EncDec}}$ is optimized by the cross entropy loss calculated between the predicted- and the target correct-sequences. For the end-to-end ASR model with attention, we use two special labels to represent the start- and end-of-sentence, $\langle sos \rangle$ and $\langle eos \rangle$, respectively. The decoder completes the decoding an utterance when the end-of-sentence is emitted. It is possible to conduct beam search to further enhance the recognition performance. In this paper, we use the encoder-decoder model for the A2W conversion and the A2P conversion.

3. Pronunciation-to-Word model

In this section, we introduce a P2W neural converter that converts a pronunciation sequence into a word sequence. Our approach is inspired by the input method engine (IME) for Japanese, called *Kana-Kanji* conversion [32], and a joint source channel model used for machine transliteration [20, 33]. In the following, we first introduce the P2W neural converter using neural network-based joint source channel model, and then explain the Seq2Seq model for P2W conversion.

3.1. Joint source channel model with LSTM-LM

The conventional system maps acoustic feature sequence $\boldsymbol{X}$ to word sequence $\boldsymbol{W}$ directly. Our proposed system first maps $\boldsymbol{X}$ to pronunciation sequence $\boldsymbol{S}$ by A2P neural converter, i.e. the attention-based encoder-decoder, and then converts $\boldsymbol{S}$ into $\boldsymbol{W}$ by using the P2W neural converter. The joint source channel model-based P2W neural converter is defined as follows:

$$P(\boldsymbol{W}, \boldsymbol{S}) = \prod_{l=1}^{L} P(\langle w_l, s_l \rangle \mid \langle w_1, s_1 \rangle, \cdots, \langle w_{l-1}, s_{l-1} \rangle; \mathbf{\Theta}), \tag{3}$$

where $\boldsymbol{W}$ and $\boldsymbol{S}$ indicate word and pronunciation sequence, respectively; this data pair is taken as the input of the P2W neural converter. Therefore, sentence $\mathcal{S}$ containing L words is represented as $\mathcal{S} = (\langle w_1, s_1 \rangle, \cdots, \langle w_L, s_L \rangle)$, where s represents pronunciation-based word. The above relationship in Japanese is summarized in Fig. 1. $\mathbf{\Theta}$ is a trainable parameter of the P2W neural converter. In this paper, we use an LSTM-LM-based joint source channel model [20] as the P2W neural converter.

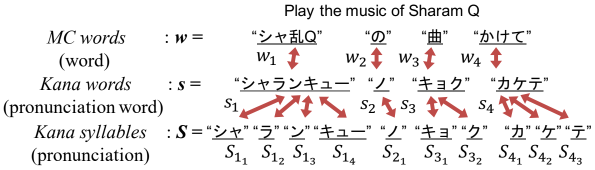
Play the music of Sharam Q
MC words : w =
(word)
"シャ乱Q" "の" "曲" "かけて"
w1 w2 w3 w4
Kana words : s =
(pronunciation word)
"シャランキュー" "ノ" "キョク" "カケテ"
s1 s2 s3 s4
Kana syllables : S = "シャ" "ラ" "ン" "キュー" "ノ" "キョ" "ク" "カ" "ケ" "テ"
(pronunciation)
S11 S12 S13 S14 S21 S31 S32 S41 S42 S43

Figure 1: *An example of Japanese sentence containing MC word, Kana word and Kana syllable sequence. MC word "シャ乱 Q" contains three mixed character sets (katakana, kanji and alphabet) in a single word.*

We apply LSTM-LM parameters to $\mathbf{\Theta}$ which is defined as $\mathbf{\Theta}_{\text{LSTM-LM}} = \{\boldsymbol{\theta}_{\text{Emb}}, \boldsymbol{\theta}_{\text{LSTM}}, \boldsymbol{\theta}_{\text{Out}}\}$. $\boldsymbol{\theta}_{\text{Emb}}$ represents the embedding layer parameter and is defined as follows:

$$\boldsymbol{e}_l = \texttt{Embed}(\langle w_l, s_l \rangle; \boldsymbol{\theta}_{\text{Emb}}). \tag{4}$$

$\boldsymbol{\theta}_{\text{Emb}}$ is used to convert $\langle w_l, s_l \rangle$ into embedded feature $\boldsymbol{e}_l$. A simple equation of LSTM with parameter $\boldsymbol{\theta}_{\text{LSTM}}$ is given as follows:

$$\boldsymbol{h}_l = \texttt{LSTMCell}(\boldsymbol{e}_l, \boldsymbol{h}_{l-1}, \boldsymbol{\theta}_{\text{LSTM}}), \tag{5}$$

where $\boldsymbol{h}_l$ and $\boldsymbol{h}_{l-1}$ indicate current- and previous-hidden state, respectively. Finally, we can obtain the following output values via a softmax function:

$$\boldsymbol{o}_l = \texttt{Softmax}(\boldsymbol{h}_l, \boldsymbol{\theta}_{\text{Out}}), \tag{6}$$

where $\boldsymbol{\theta}_{\text{Out}}$ is the output linear layer parameter, and $\boldsymbol{o}_l$ corresponds to the probability $P(\langle w_l, s_l \rangle \mid \langle w_1, s_1 \rangle, \cdots, \langle w_{l-1}, s_{l-1} \rangle; \mathbf{\Theta})$. The objective function for training the LSTM-LM-based joint source channel model is also optimized by the cross entropy loss calculated between the predicted- and the target correct-sequences.

For decoding, we deal with pronunciation-based character set as the input symbol which is defined as $(S_{l_1}, \cdots, S_{l_M}) \in \texttt{split}(s_l)$ where M is the number of pronunciation-based characters in the l-th pronunciation-based word S_l. Therefore, the pronunciation sequence $\boldsymbol{S}$ is written as $\boldsymbol{S} = (S_{1_1}, \cdots, S_{L_M})$, which can be conveted into lattices of possible sequences of words and pronunciation-based words by using a dictionary. The decoder can find the candidate that maximizes Eq.(3) with regard to $\boldsymbol{W}$. Unfortunately, the calculation costs are high, so the P2W neural converter conducts beam search to attain higher conversion efficiency.

3.2. Seq2Seq model for P2W conversion

We can use the attention-based encoder-decoder model for the Seq2Seq-based converter that transforms $\boldsymbol{S}$ into $\boldsymbol{W}$. In this paper, the model works in the same way as the attention-based encoder-decoder model explained in Section 2. We replace input sequence $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$ and output sequence $\boldsymbol{Y} = (y_1, ..., y_N)$ with $\texttt{EMBED}(\texttt{split}(s_1), \cdots, \texttt{split}(s_L); \boldsymbol{\theta}'_{\text{Emb}})$ and $(w_1, \cdots, w_L)$, respectively. $\boldsymbol{\theta}'_{\text{Emb}}$ is a trainable parameter. We note that input symbols, pronunciation-based word sequence $\boldsymbol{s}$, are split to yield pronunciation sequence $\boldsymbol{S}$.

4. Joint maximization decoder (JMD) with neural converters

In this paper, we use pronunciation units as the outputs of the attention-based encoder-decoder model from acoustic features. Therefore, the units are transformed into word sequences by using a dictionary. The definition of ASR that finds the best word sequence $\boldsymbol{W}$ from observation $\boldsymbol{X}$ is described as follows:

$$\begin{aligned} \hat{\boldsymbol{W}} &= \arg\max_{\boldsymbol{W}} P(\boldsymbol{W} \mid \boldsymbol{X}) \\ &= \arg\max_{\boldsymbol{W}} \sum_{\boldsymbol{S}} P(\boldsymbol{W} \mid \boldsymbol{S}) P(\boldsymbol{S} \mid \boldsymbol{X}) \\ &\approx \arg\max_{\boldsymbol{W}} P(\boldsymbol{W} \mid \boldsymbol{S}) P(\boldsymbol{S} \mid \boldsymbol{X}). \qquad (7) \end{aligned}$$

Eq.(7), which takes account of pronunciation sequence $\boldsymbol{S}$, is derived in [34]. In [19], they used GCD which maximizes $P(\boldsymbol{S} \mid \boldsymbol{X})$ and $P(\boldsymbol{W} \mid \boldsymbol{S})$ independently; it is described as follows:

$$\hat{\boldsymbol{S}} = \arg\max_{\boldsymbol{S}} P(\boldsymbol{S} \mid \boldsymbol{X}), \qquad (8)$$

$$\hat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}} P(\boldsymbol{W} \mid \hat{\boldsymbol{S}}). \qquad (9)$$

First, they obtained the best pronunciation sequence $\hat{\boldsymbol{S}}$ using an Seq2Seq model that converts $\boldsymbol{X}$ into $\boldsymbol{S}$ generated by using A2P neural converter, i.e. Eq.(1) and (2). Next, $\hat{\boldsymbol{S}}$ is transformed into the best word sequence $\hat{\boldsymbol{W}}$ by another Seq2Seq model with lexicon. This approach is a good solution for FNN-based ASR when there are symbols, e.g. Chinese and Japanese. However, we assume that the pronunciation sequence $\boldsymbol{S}$ and the word sequence $\boldsymbol{W}$ are to be jointly optimized, because the other hypotheses $\boldsymbol{S}$ other than $\hat{\boldsymbol{S}}$ may contain the correct sequence.

In this paper, we propose JMD; it considers both word and pronunciation sequences in the decoding step. The definition of our proposal which jointly finds $\boldsymbol{W}$ and $\boldsymbol{S}$ is described as follows:

$$\hat{\boldsymbol{W}} \approx \arg\max_{\boldsymbol{W}} P(\boldsymbol{W}, \boldsymbol{S})^{\lambda} P(\boldsymbol{S} \mid \boldsymbol{X}), \qquad (10)$$

where λ is a hyperparameter for biasing a joint probability term i.e. the outputs of LSTM-LM-based joint source channel model. This equation simultaneously finds $\boldsymbol{W}$ and $\boldsymbol{S}$ given observation $\boldsymbol{X}$. Eq.(10) is implemented in beam search decoding when converting $\boldsymbol{X}$ into $\boldsymbol{S}$. We utilize the ability of the P2W neural converter with joint source channel model to successively work left-to-right by unidirectional LSTM-LM for optimizing pronunciation hypotheses. Completion of A2P beam search yields the best sequences for $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{S}}$ at the same time.

5. Experiments

5.1. Data

We evaluated our proposal on two speech recognition tasks using a corpus of spontaneous Japanese (CSJ) [35]: the corpus was divided into two distinct sub-corpora, APS and SPS, which are regarded as external resources for each other. APS and SPS consist of 224-hours of academic public speeches and 251-hours of simulated public speeches, respectively. Japanese texts use the character sets of $Hiragana$, $Katakana$, $Kanji$, and $Roman$ alphabets in a mixed manner. Therefore, there are many more characters in Japanese than, for example, there are graphemes in English. In this paper, we used MC word and $Kana$ syllable unit sets as word and pronunciation respectively.

Table 1: *The number of MC-words and $Kana$-syllable symbols, i.e. word unit $\boldsymbol{W}$ and pronunciation unit $\boldsymbol{S}$, in each training set. The MC word vocabularies in each domain are registered without OOV, but we include OOV symbols in the evaluation set. The OOV rates of the word entry-level in each evaluation set of E1 and E3 are shown in 3rd row "E1/E3".*

Symbol	APS	SPS	APS+SPS
# MC words	32155	43510	57318
OOV rates (E1/E3)	7.1%/12.3%	10.6%/4.2%	5.2%/3.4%
# $Kana$ syllables	265(common)		

Table 1 shows the number of distinct units used for the models of each corpus. The corpora we used for model training have their own official evaluation sets 1 (E1) and 3 (E3) corresponding to the APS and SPS domain. E1 and E3 have durations of 1.9 and 1.3 hours, respectively.

5.2. System configuration

The input feature is a 40 dimensional log Mel-filterbank with non-overlapping frame stacking [36]; three frames were stacked and skipped to make each new super-frame. The acoustic encoder in the attention-based model consists of four layers of bidirectional LSTMs with 320 cells; the drop-out rate was 0.2 [37]. The attention-based decoder consists of one-layer LSTM with 320 cells, a hidden layer with 320 tanh nodes, and a softmax output layer for MC word entries in the conventional FNN-based ASR system, or $kana$ syllable entries in the A2P neural converter, i.e. the proposed FNN-based ASR system.

P2W neural converters were modeled by LSTM-LM or Seq2Seq model. The LSTM-LM architecture is composed of an embedding layer with 400 hidden units and one-layer LSTM with 400 cells. As the P2W neural converter, Seq2Seq model consists of an embedding layer with 400 hidden units and two-layers of bidirectional LSTMs with 400 cells. The attention-based decoder consists of one-layer LSTM with 400 cells, a hidden layer with 400 tanh nodes, Moreover, we also investigated an N-gram-based converter, which also uses the joint source channel model where $\boldsymbol{\Theta}_{\textbf{LSTM-LM}}$ is replaced with $\boldsymbol{\Theta}_{\textbf{N-gram-LM}}$ in Eq.(3). We used 3-gram LM[1]; it was built using the SRILM toolkit [38].

We used the Adam optimizer with the setting described in [39]. We also used gradient clipping with a threshold of 5.0. All network parameters were initialized with random values according to the setting in [40]. Since providing long input sequences can slow convergence at the beginning of the training for encoder-decoder model, the input data were sorted by increasing frame length before creating minibatches. We used label smoothing for improving the generalization performance of the encoder-decoder model as described in [41]. All NN-based models were trained by using the PyTorch toolkit [42]. In decoding with the A2W or A2P neural converters and the P2W neural converter, we used simple beam search with beam width of 4. The hyperparameter λ in Eq.(10) for JMD was set to 0.01. We evaluated performance in terms of word error rate (WER) and character error rate (CER).

5.3. Results

The IDs, "C*", in Table 2 show MC-based WERs and CERs for each converter system modeled by Seq2Seq, 3-gram LM and LSTM-LM. These systems were trained with APS+SPS text data, and converted correct $Kana$ sequences, i.e. Ora-

[1] We investigated several N-gram from 1 to 10. The best N was 3.

Table 2: *Several unit CERs and WERs of all systems. Each head alphabet in system ID corresponds to the experiment ID and all "Baseline" indicate the results of the direct A2W encoder-decoder model. Experiment "C*" was performed to compare P2W conversion system performance. "A*,S*" were performed for investigating the effectiveness of external language resource usage for the P2W neural converter. "A*,S*,AS*" were also conducted in order to confirm the impact of decoder type, i.e. GCD or JMD, on our proposal.*

Syst. ID	Training data		Converter type (P2W)	Decoder type	CER(***Kana***)		WER(MC)		CER(MC)	
	encoder-decoder (speech-text)	P2W converter (text)			E1	E3	E1	E3	E1	E3
C0	-	APS+SPS	Seq2Seq	-	0.0 (Oracle)		14.2	10.3	12.0	7.2
C1	-	APS+SPS	3-gram-LM	-			5.4	6.7	2.4	2.8
C2	-	APS+SPS	LSTM-LM	-			**5.1**	**5.2**	**2.1**	**2.2**
A0	Baseline(APS)	-	-	-	-	-	18.4	31.7	15.5	25.7
A1	APS	APS	LSTM-LM	GCD	9.1	14.7	19.7	32.1	14.1	23.3
A2	APS	APS+SPS		GCD	〃	〃	18.8	28.5	13.8	21.2
A3	APS	APS		JMD	8.4	13.5	17.8	29.6	**12.9**	21.4
A4	APS	APS+SPS		JMD	**8.3**	**12.9**	**17.7**	**24.9**	13.0	**18.4**
S0	Baseline(SPS)	SPS	-	-	-	-	31.7	15.6	28.2	12.1
S1	SPS	SPS	LSTM-LM	GCD	13.8	7.3	32.2	16.7	24.6	11.4
S2	SPS	APS+SPS		GCD	〃	〃	28.7	15.8	22.2	11.2
S3	SPS	SPS		JMD	13.6	**5.7**	29.9	**14.0**	23.2	**9.0**
S4	SPS	APS+SPS		JMD	**13.0**	5.8	**25.4**	**14.0**	**20.0**	**9.0**
AS0	Baseline(APS+SPS)	-	-	-	-	-	16.5	14.3	13.6	11.2
AS1	APS+SPS	APS+SPS	LSTM-LM	GCD	8.0	6.7	17.8	15.5	12.8	10.3
AS2	APS+SPS	APS+SPS		JMD	**7.2**	**5.4**	**15.7**	**13.2**	**11.3**	**8.5**

cle outputs of A2P neural converter, into MC word sequences. The best P2W system was the LSTM-LM-based neural converter. The joint source channel model approach, which is 3-gram LM and LSTM-LM, yielded much better performance than Seq2Seq. We assumed that simple converter models, i.e. joint source channel model, are sufficient for capturing the relationships between $Kana$ syllable and MC word i.e. pronunciation and word. We used the LSTM-LM-based neural converter in all subsequent experiments.

WERs(MC) and CERs(MC) in Table 2 show the A2W ASR performance or P2W conversion performance. The experiments with IDs of "A*" and "S*", in Table 2 were performed to investigate the effectiveness of vocabulary expansion. "Baseline" indicates the A2W encoder-decoder model, which directly maps acoustic features to MC word sequences. The column of CER ($Kana$) indicates A2P ASR performance, i.e. acoustic-to-$Kana$, for our system. The A2W encoder-decoder and both A2P and P2W neural converter of the systems, "A0,A1,A3" and "S0,S1,S3", were trained using only single domain data. Systems "A2,A4" and "S2,S4" were investigated to examine the effectiveness of vocabulary expansion assuming a P2W neural converter trained with only text data. The difference between "*1,*2" and "*3,*4" of "A*,S*" is decoder type, GCD or JMD, in our proposal. In the columns of WER(MC) and CER(MC), the rows of "A*,S*" show that using training data from the same domain yields better performance than data from out of the domain. We consider that this is reasonable because the system did not train acoustic and linguistic features in the target domain. We can see that our systems that did not use all text for training the P2W neural converter had worse WERs(MC) than the baseline. However, their CERs(MC) were much better than those of the baseline. The P2W neural converter can create substitution errors, but this merely results in the output of an MC word similar to the correct one. In contrast, the baseline system i.e. A2W encoder-decoder model, outputs gibberish when its result is wrong. We assume that because the proposed system uses aligned $Kana$ syllable sequences, it could capture more detail of the relationship between acoustic and linguistic features than the A2W encoder-decoder model. By extending the vocabulary through the use of external language resources, i.e. the systems "A2,A4" and "S2,S4", our proposed system could almost match or better the WERs of the baseline. This means that the performance of vocabulary extended systems can be further improved in out of domains. We argue that our proposed framework can raise recognition performance through the use of just text data. Moreover, the proposed decoder JMD was applied to the systems of "A3,A4" and "S3,S4". CER($Kana$), WER(MC) and CER(MC) were significantly improved with the joint maximization of both A2P and P2W scores. Considering MC sequences when decoding $Kana$ hypotheses was confirmed to be important and very effective.

Finally, we compared "AS*" with all other systems. All models were trained with APS+SPS speech and text. The results are shown in the columns of CER($Kana$), WER(MC) and CER(MC) in Table 2. The P2W neural converter of our proposal received $Kana$ syllable sequences, i.e. the A2P neural converter outputs. We can see that by improving the performance of the $Kana$ syllable-based encoder-decoder system, our systems, "AS1" and "AS2" achieved better WERs(MC) and CERs(MC) than "*0-2" or "*0-4" of "A*,S*". Moreover, the results of "AS1" and "AS2" confirm that the proposed decoder, JMD, was also more effective than conventional decoder GCD. We argue that our proposal is a key advance towards FNN-based ASR systems for languages that have a lot of units.

6. Conclusions

We have proposed JMD; it considers the joint probability of a pronunciation and a word in beam-search decoding. A joint source channel model using LSTM-LM was introduced for maximizing the joint probability. JMD utilizes the benefits of the unidirectional LSTM-LM which can work left-to-right for beam search decoding. Therefore, the decoder can successively receive P2W scores as the feedback for modifying A2P score, and handle not only the best pronunciation sequence, but also other hypotheses. Moreover, the P2W neural converter using LSTM-LM can be trained using just text that may contain MC-based OOVs. We showed the effectiveness of the proposal in Japanese ASR tasks where it achieved the best CER and WER.

7. References

[1] L. Lu, X. Zhang, and S. Renals, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *Proc. of ICASSP*, 2016, pp. 5060–5064.

[2] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Proc. of INTERSPEECH*, 2017, pp. 3707–3711.

[3] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," in *Proc. of ICASSP*, 2017, pp. 4759–4763.

[4] S. Ueno, H. Inaguma, M. Mimura, and T. Kawahara, "Acoustic-to-word attention-based model complemented with character-level CTC-based model," in *Proc. of ICASSP*, 2018, pp. 5804–5808.

[5] J. Li, G. Ye, A. Das, R. Zhao, and Y. Gong, "Advancing acoustic-to-word CTC model," in *Proc. of ICASSP*, 2018, pp. 5794–5798.

[6] S. Palaskar and F. Metze, "Acoustic-to-word recognition with sequence-to-sequence models," in *Proc. of SLT*, 2018, pp. 397–404.

[7] Z. Chen, Q. Liu, H. Li, and K. Yu, "On modular training of neural acoustics-to-word model for lvcsr," in *Proc. of ICASSP*, 2018, pp. 4754–4758.

[8] S. Ueno, T. Moriya, M. Mimura, S. Sakai, Y. Shinohara, Y. Yamaguchi, Y. Aono, and T. Kawahara, "Encoder transfer for attention-based acoustic-to-word speech recognition," in *Proc. of INTERSPEECH*, 2018, pp. 2424–2428.

[9] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," in *Proc. of ICASSP*, 2017, pp. 4805–4809.

[10] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. of ICASSP*, 2018, pp. 4774–4778.

[11] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Proc. of INTERSPEECH*, 2018, pp. 7–11.

[12] H. Xu, S. Ding, and S. Watanabe, "Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling," *CoRR*, vol. abs/1811.04284, 2018.

[13] M. Schuster and K. Nakajima, "Japanese and Korean voice search." in *Proc. of ICASSP*, 2012, pp. 5149–5152.

[14] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. of ACL*, 2016, pp. 1715–1725.

[15] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model." in *Proc. of INTERSPEECH*, 2010, pp. 1045–1048.

[16] T. Hori, S. Watanabe, and J. R. Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in *Proc. of ASRU*, 2017, pp. 287–293.

[17] T. Hori, J. Cho, and S. Watanabe, "End-to-end speech recognition with word-based rnn language models," in *Proc. of SLT*, 2018, pp. 389–396.

[18] A. Maciejewski and N. K. Leung, "The nihongo tutorial system an intelligent tutoring system for technical japanese language instruction," *CALICO Journal*, vol. 9, no. 3, pp. 5–25, 2003.

[19] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," in *Proc. of INTERSPEECH*, 2018, pp. 791–795.

[20] H. Li, M. Zhang, and J. Su, "A joint source-channel model for machine transliteration," in *Proc. of ACL*, 2004, pp. 159–166.

[21] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification : Labelling unsegmented sequence data with recurrent neural networks," in *Proc. of ICML*, 2006, pp. 369–376.

[22] A. Graves and N. Jaitly, "Towards End-To-End speech recognition with recurrent neural networks," in *Proc. of ICLR*, 2014, pp. 1764–1772.

[23] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: first results," in *Advances in NIPS*, 2014.

[24] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in NIPS*, 2015, pp. 577–585.

[25] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-End attention-based large vocabulary speech recognition," in *Proc. of ICASSP*, 2016, pp. 4945–4949.

[26] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. of ICASSP*, 2016, pp. 4960–4964.

[27] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," in *INTERSPEECH*, 2017, pp. 1298–1302.

[28] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," *Proc. of ICASSP*, pp. 4835–4839, 2017.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in NIPS*, 2017, pp. 5998–6008.

[30] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," *Proc. of ASRU*, pp. 193–199, 2017.

[31] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. of ICASSP*, 2018, pp. 5884–5888.

[32] S. Mori, M. Tsuchiya, O. Yamaji, and M. Nagao, "Kana-kanji conversion by a stochastic model," in *Transactions of IPSJ*, 1999, pp. 2946—-2953.

[33] Y. Okuno and S. Mori, "An ensemble model of word-based and character-based models for japanese and chinese input method," in *Proc. of WTIM2*, 2012, pp. 15–28.

[34] N. Kanda, X. Lu, and H. Kawai, "Maximum a posteriori based decoding for ctc acoustic models," pp. 1868–1872, 2016.

[35] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of japanese," in *Proc. of LREC'00*, 2000.

[36] H. Sak, A. W. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Proc. of INTERSPEECH*, 2015, pp. 1468–1472.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, pp. 1929–1958, 2014.

[38] A. Stolcke, "SRILM - an extensible language modeling toolkit." in *Proc. of INTERSPEECH*, 2002.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR*, 2015.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. of ICCV*, 2015, pp. 1026–1034.

[41] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, "Regularizing neural networks by penalizing confident output distributions," in *Proc. of ICLR*, 2017.

[42] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.