IMPROVING OOV DETECTION AND RESOLUTION WITH EXTERNAL LANGUAGE MODELS IN ACOUSTIC-TO-WORD ASR

Hirofumi Inaguma[1]*, Masato Mimura*[1]*, Shinsuke Sakai*[1]*, Tatsuya Kawahara*[1]

[1]Graduate School of Informatics, Kyoto University, Japan

ABSTRACT

Acoustic-to-word (A2W) end-to-end automatic speech recognition (ASR) systems have attracted attention because of an extremely simplified architecture and fast decoding. To alleviate data sparseness issues due to infrequent words, the combination with an acoustic-to-character (A2C) model is investigated. Moreover, the A2C model can be used to recover out-of-vocabulary (OOV) words that are not covered by the A2W model, but this requires accurate detection of OOV words. A2W models learn contexts with both acoustic and transcripts; therefore they tend to falsely recognize OOV words as words in the vocabulary. In this paper, we tackle this problem by using external language models (LM), which are trained only with transcriptions and have better linguistic information to detect OOV words. The A2C model is used to resolve these OOV words. Experimental evaluations show that external LMs have the effects of not only reducing errors but also increasing the number of detected OOV words, and the proposed method significantly improves performances in English conversational and Japanese lecture corpora, especially for out-of-domain scenario. We also investigate the impact of the vocabulary size of A2W models and the data size for training LMs. Moreover, our approach can reduce the vocabulary size several times with marginal performance degradation.

Index Terms— End-to-end speech recognition, acoustic-to-word, attention-based encoder-decoder, multi-task learning, recurrent neural network language model, OOV

1. INTRODUCTION

The conventional HMM-based hybrid automatic speech recognition (ASR) systems are modularized into several components such as acoustic, subword, lexicon, pronunciation, and language model. While they are shown to achieve a human-level precognition performance [1, 2], they are separately optimized based on different objectives and training data, so the overall system is sub-optimal and needs a complicated decoding process to combine them.

In contrast, end-to-end systems, which optimize the direct mapping from acoustic features to transcriptions, have shown promising results using three end-to-end ASR models: connectionist temporal classification (CTC) [3–6], attention-based encoder-decoder models [7–9], and RNN-transducer models [10, 11]. In this paper, we focus on attention-based models because they show the most promising results among them [10–12]. There are some choices about output units, and a majority of the previous end-to-end systems is based on subword units such as characters [8] and word-piece units [13]. Recently, acoustic-to-word (A2W) models have received much attention because of their extremely simplified architecture and fast decoding process [12, 14–20]. When considering downstream processes of ASR such as machine translation, dialogue systems, and spoken term detection, word-level information is required, so it is a natural choice to adopt a word as the output unit. However, A2W models need a considerable amount of training data to match the performance of the state-of-the-art subword-based models [17]. Therefore, they encounter problems of data sparseness and over-fitting due to infrequent words [14]. Moreover, A2W models cannot recognize out-of-vocabulary (OOV) words.

To tackle these problems, some works explore model initialization [15] and multi-task learning (MTL) with low-level auxiliary tasks [12, 18, 20]. In the MTL approach, we jointly optimize an A2W model with an acoustic-to-character (A2C) model by sharing encoder parameters [12, 18, 20]. OOV words are resolved by referring to the corresponding partial hypothesis provided by the A2C model [12, 18]. It also has generalization effects by accelerating better parameter representations and leads to fast and stable convergence.

However, even with these regularization techniques, they are more likely to incorrectly recognize OOV words as other words in the predefined vocabulary. This is because A2W models learn contexts with both acoustic and word sequence, and attach too much importance to acoustic level. When recognizing speech corresponding to OOV words (these are often infrequent words), the A2W model assign high probabilities to other in-vocabulary words which have similar pronunciations. In this case, these words cannot be recovered by the A2C model.

External language models (LM) trained with a large text have better linguistic information and further improve the ASR performance. They are integrated to the attention-based models in the beam search decoding, which are referred to as *shallow fusion* [21]. Since external LMs are trained with a large text, they have better ability to detect OOV words accurately.

In this paper, we improve the OOV resolution for the attention-based A2W model by integrating the external LM. We combine *shallow fusion* with external LMs and the OOV resolution method by referring to the hypothesis of the A2C model. We show that external LMs not only reduce errors of the A2W models but also increase the number of OOV words in the hypothesis, which means that more OOV words are detected with external LMs. Then, recovering these OOV slots by the A2C model boosts the performance. In addition, we found that the effect of external LMs are enhanced with MTL without recovering OOV words.

We conduct experimental evaluations with English conversational and Japanese lecture corpora, and show the proposed method significantly improves the performances, especially for out-of-domain test sets. Moreover, we investigate the effects of the vocabulary size of the A2W models and the data size for training external LMs. By restricting the vocabulary to the frequent words and recovering many OOV words from the A2C model, the vocabulary size can be reduced several times from the best model with marginal performance degradation.

The remaining part of this paper is structured as follows. In Sec-

tion 2, we describe previous research on OOV problems of A2W models. In Section 3, we describe attention-based A2W models and the proposed method. Experimental evaluations are presented in Section 4. We conclude this paper in Section 5.

2. ACOUSTIC-TO-WORD END-TO-END SPEECH RECOGNITION WITHOUT OOV

Acoustic-to-word (A2W) end-to-end systems are attractive because they can directly optimize mapping from acoustic to word sequences with a single architecture, and realize fast decoding. Although A2W models have these advantages, they suffer from the data sparseness problems due to rare words and therefore require sufficient training data. Moreover, A2W models cannot recognize any OOV words. A practical solution is to adopt word-pieces as output units [13]. There are some solutions to make A2W models open-vocabulary. The first one is to decompose infrequent words to a sequence of subwords and encapsulate them into a single dictionary [16, 20]. Another solution is to recover OOV words by referring to the hypothesis of the A2C model, which is realized by joint training [12, 18]. Time alignments of the A2W and A2C models are synchronous because they share encoder parameters. This is similar to human perception, where frequent words are memorized but rare words such as named entities must be spelled out. In this paper, we focus on this modeling for the open-vocabulary A2W end-to-end ASR system.

3. MODEL DESCRIPTION

This section describes the baseline attention-based acoustic-to-word (A2W) model, the multi-task learning (MTL) framework with the acoustic-to-character (A2C) model, the recurrent neural network language model (RNNLM) integration to the attention-based models, and recovering OOV words with the A2C model. Let $\boldsymbol{x} = (x_1, \ldots, x_T)$ be the input speech frames of length T, $\boldsymbol{y}^w = (y_1^w, \ldots, y_N^w)$ be the corresponding word-level transcription of length N, and $\boldsymbol{y}^c = (y_1^c, \ldots, y_M^c)$ be the corresponding character-level transcription of length M.

3.1. Baseline attention-based A2W model

In this paper, A2W models are built based on the attention-based encoder-decoder model [7–9], which is an end-to-end sequence labeling model and can learn soft alignments between a variable-length input and a target sequence. Attention-based models incorporate contextual information from the target label sequence in the decoder part unlike the CTC models, which is another end-to-end sequence labeling model and learn frame-level contexts in the encoder part. We focus on the attention-based models as a baseline A2W model in this paper. The A2W model is composed of three components: encoder, word-level decoder, and attention layer.

The encoder network consists of the stacked multiple layers of bidirectional long-short term memory (BLSTM) [22] and transforms $\boldsymbol{x}$ into a distributed representation $\boldsymbol{h} = (h_1, \ldots, h_T)$.

The decoder network consists of a single LSTM layer and generates the probability distribution of the lexical entries conditioned over all the previous outputs. The decoder's hidden state $\boldsymbol{s}_n$ at each output timestep n is updated as a function of the context vector $\boldsymbol{c}_n$, and previously output word y_{n-1}^w (which is passed through the word embedding layer) recursively as follows:

$$\begin{aligned} y_n^w &\sim Generate(\boldsymbol{s}_{n-1}, \boldsymbol{c}_n) \\ \boldsymbol{s}_n &= Reccurency(\boldsymbol{s}_{n-1}, \boldsymbol{c}_n, y_n^w) \end{aligned} \tag{1}$$

The attention layer computes an attention distribution $\boldsymbol{\alpha}_n^w = (\alpha_{n,1}^w, \ldots, \alpha_{n,T}^w)$, which is a relevance score over the entire encoder's outputs $\boldsymbol{h}$, and computes the context vector $\boldsymbol{c}_n$ by summing over the encoder's outputs $\boldsymbol{h}$ as follows:

$$\begin{aligned} e_{n,t} &= \boldsymbol{v}^T tanh(\boldsymbol{W}\boldsymbol{s}_{n-1} + \boldsymbol{V}\boldsymbol{h}_t + \boldsymbol{U}\boldsymbol{f}_{n,t} + \boldsymbol{b}) \\ \boldsymbol{f}_n &= \boldsymbol{F} * \boldsymbol{\alpha}_{n-1}^w \\ \boldsymbol{\alpha}_n^w &= softmax(\boldsymbol{e}_n) \\ \boldsymbol{c}_n &= \sum_{t=1}^{T} \alpha_{n,t}^w \boldsymbol{h}_t \end{aligned}$$

where $\boldsymbol{F}$, $\boldsymbol{W}$, $\boldsymbol{V}$, $\boldsymbol{U}$, $\boldsymbol{v}$, and $\boldsymbol{b}$ are trainable parameters and $*$ denotes convolutional operation, and f_n is a convolutional feature from the previous attention distributions $\boldsymbol{\alpha}_{n-1}^w$.

The loss function is designed as the negative log-likelihood and used for parameter estimation:

$$L_w(\boldsymbol{x}, \boldsymbol{y}^w) = -\ln P(\boldsymbol{y}^w|\boldsymbol{x})$$

3.2. Multi-task learning with attention-based A2C model

To alleviate data sparseness issues of A2W models, the MTL with the A2C model is performed by sharing encoders' parameters. As with the previous section, the A2C model is also built based on the attention-based model and has different parameters concerning the decoder and attention layer. Although there is another choice of the CTC-based model for A2C the model as in [12], we adopt the attention-based model because character-level CTC models are more likely to misspell than attention-based models [10, 11]. The character-level decoder can be connected to the arbitrary intermediate layer [23, 24]. The overall loss function is the linear interpolation of the negative log-likelihood between the A2W and A2C models by a tunable parameter λ ($0 \leq \lambda \leq 1$):

$$L(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^c) = \lambda L_w(\boldsymbol{x}, \boldsymbol{y}^w) + (1-\lambda) L_c(\boldsymbol{x}, \boldsymbol{y}^c)$$

where L_c is the negative log-likelihood of the A2C model.

3.3. RNNLM integration

Although attention-based models explicitly model linguistic contexts in the decoder part, external LMs trained with a larger text corpus can provide a reliable probability distribution to the decoder. The left-to-right beam search decoding with an external language model, which is referred to as *shallow fusion* [21], is performed to find the most probable word sequence $\boldsymbol{y}^{w*}$ based on the following criterion:

$$\begin{aligned} \boldsymbol{y}^{\boldsymbol{w}*} = \arg\max_{\boldsymbol{y}^w} \{ &\log P_{a2w}(\boldsymbol{y}^w|\boldsymbol{x}) + \beta_w \log P_{wlm}(\boldsymbol{y}^w) \\ &+ \gamma_w \text{coverage} \} \end{aligned}$$

where β_w and γ_w are tunable parameters. To use a narrow beam width and keep the decoding efficiency of the A2W model, scores by the external LM is added in the loop of the decoder network, not in the rescoring step. The coverage terms are added to prevent long sequences composed of repeated tokens and calculated as follows:

$$\text{coverage} = \sum_{t=1,\ldots,T} [\sum_{n=1,\ldots,N} \alpha_{n,t}^w > \tau]$$

where τ is a threshold to receive a cumulative attention lager than its value. We set τ to 0, and this also purges short hypotheses from candidates in the beam.

Table 1: Recognition performances on Switchboard corpus (300h). SWB and CH represent Switchboard and CallHome subsets, respectively. #OOV represents the number of detected OOV words. Beam search decoding was performed with $beam\_size = 5$ in all experiments. The vocabulary size was about 11k. The OOV rates of SWB and CH test sets were 1.81 and 3.00, respectively.

Model	Resolving OOV	RNNLM	SWB	CH	Ave.
			WER (#OOV)	WER (#OOV)	WER
Word CTC	-	×	20.26 (240)	42.32 (358)	31.29
A2W (baseline)	-	×	18.99 (154)	38.46 (222)	28.73
	-	300h	18.45 (319)	38.49 (463)	28.47
	-	2000h	18.35 (322)	38.13 (490)	28.24
A2W+A2C	×	×	18.35 (183)	37.54 (267)	27.95
	✓	×	18.18 (〃)	37.40 (〃)	27.79
	×	300h	17.76 (349)	37.26 (513)	27.51
	✓	300h	17.43 (〃)	36.99 (〃)	27.21
	×	2000h	17.40 (346)	37.00 (546)	27.20
	✓	2000h	**17.11** (〃)	**36.71** (〃)	**26.91**

3.4. Resolving OOV words

In the MTL framework, the A2C model not only works as regularization effects but also provides additional information to the A2W model. As with [12, 18], in the inference stage, we refer to the A2C model's hypothesis $\hat{\boldsymbol{y}}^c = (\hat{y}^c_1, \cdots, \hat{y}^c_M)$ when outputting OOV words and replace them with the corresponding space-separated word including $\hat{y}^c_m$ by computing the index m where attention weights between the A2W and A2C models are most overlapped.

$$\{\overline{\boldsymbol{\alpha}^c_{m,i}}\}_i = \frac{\boldsymbol{\alpha}^c_{m,2i} + \boldsymbol{\alpha}^c_{m,2i+1}}{2}$$
$$m = \underset{m=1,\cdots,M}{\arg\max} (\boldsymbol{\alpha}^w_n \cdot \overline{\boldsymbol{\alpha}^c_m})$$

Since time resolutions of activations of the encoder attached to the A2W and A2C models are different due to the subsampling layers [8], frame-wise attention weight of A2C models $\boldsymbol{\alpha}^c$ is averaged between the adjacent two frames before multiplication. Note that character-level hypotheses are obtained by greedy decoding to keep the decoding speed of the A2W models.

4. EXPERIMENTAL EVALUATION

4.1. Switchboard corpus (300h)

4.1.1. System settings

We used the Switchboard corpus (LDC97S62) [25], which contains about 300-hour conversational English telephone speech, as the training set. Following data preparation in Kaldi recipe [26], we reserved the first 4k utterances as a validation set separately. Besides, we removed duplicated utterances ("yeah," "uh-huh" etc.) beyond a count threshold of 300. The final training set has about 192k utterances. For evaluation, we report results on Hub5 Eval2000 test set (LDC2002S09), which consists of two subsets, Switchboard (SWB) and CallHome (CH).

For A2W models, we restricted the vocabulary to words with at least five occurrences in the training set and replaced the rest to a single out-of-vocabulary (OOV) class. The resulting vocabulary size was roughly 11k words, and the OOV rates of SWB and CH test sets were 1.81 and 3.00, respectively. For A2C models, we used 47 character sets (26 alphabets, digits, hyphen, space, and end-of-sentence, etc.). The input features were static 80-channel log-mel filterbank outputs computed with a 25ms window and shifted every 10 ms. The features were normalized by the mean and the standard deviation on the speaker basis. None of speaker adaption techniques was used. The encoder consists of 5 stacked BLSTM layers with 320 memory cells per direction, and both word and character-level decoders consists of a single LSTM layer with 320 memory cells. Subsampling was performed in {1,2,4}-th layers of the encoder to approximately equate sequence lengths to the number of the corresponding tokens in transcriptions. The character-level decoder was attached to the 4-th layers of the encoder. This resulted in 4 and 8-fold reduction of the encoder activations in the A2C and A2W models, respectively [8, 27, 28]. Output words and characters were embedded to the fixed dimension of size 128 and 32, respectively. The dropout ratio 0.2 was applied to the encoder, decoders, and embedding layers. Training was performed on mini-batches of 50 or 60 utterances using Adam [29] with a learning rate of 1.0×10^{-3} followed by SGD [30] with a single GPU. For fast and stable training, all utterances in the training set were sorted in the ascending order by their lengths in all training stage [5, 6, 16]. All weights were initialized with random values drawn from a uniform distribution with a range $[-0.1, 0.1]$. We also clipped the norms of gradients so that they had maximum absolute values of 5 [31]. The probabilities of scheduled sampling [32] and label smoothing [28, 33] were 0.2 and 0.1, respectively. We empirically set $\lambda = 0.5$, $\beta_w = 0.2$, $\gamma_w = 0.4$ (w/o LM), and $\gamma_w = 0.6$ (w/ LM), respectively.

RNNLMs were composed of two layers of unidirectional LSTM with 512 memory cells and have residual connections between two LSTM layers [34]. Input and output embeddings were tied as in [35, 36]. RNNLMs were optimized using back-propagation through time (BPTT) with a sequence length of 100. We used the same transcriptions as the A2W models (300-hour) and also those appended with Fisher corpus (totally 2000-hour) for training RNNLMs. The beam width was set to 5 in all the experiments. All networks were implemented with a Pytorch framework [37].

4.1.2. Results

The results are shown in Table 1. The attention-based A2W model outperformed the word CTC model[1][2]. This is because the decoder part in the attention-based A2W model captured richer linguistic

[1] As in [15, 16], we also confirmed that the initialization of the BLSTM encoder with phone CTC improved the performances of both word CTC and attention-based A2W model. The resulting WERs w/o LMs were 19.78/39.97 and 18.86/37.81 (SWB/CH) with $beam\_width = 5$, respectively. However, the A2W model still outperformed the word CTC model.

[2] We did not use any speaker adaptation techniques such as i-vector based adaptation as in [15, 16]. We assume that this is the major cause of the performance gaps between our results and those in [15, 16] in CallHome subset.

Table 2: Recognition performances on CSJ (240h). *eval3* is the out-of-domain test set. #OOV represents the number of detected OOV words. Beam search decoding was performed with $beam\_size = 5$ in all experiments. The vocabulary size was about 12.5k. The OOV rates of *eval1*, *eval2*, and *eval3* were 1.24, 1.66, and 4.09, respectively.

Model	Resolving OOV	RNNLM	In-domain		Out-of-domain	Ave. WER
			eval1	*eval2*	*eval3*	
			WER (#OOV)	WER (#OOV)	WER (#OOV)	
Word CTC	-	×	12.79 (352)	11.12 (469)	20.28 (662)	14.73
A2C	-	×	12.82	9.98	18.98	13.93
A2W (baseline)	-	×	12.89 (265)	10.25 (299)	19.70 (498)	14.28
	-	240h	12.20 (437)	9.73 (531)	19.49 (761)	13.80
	-	600h	12.11 (443)	9.65 (516)	18.71 (759)	13.49
A2W+A2C	×	×	12.27 (252)	9.96 (334)	18.70 (521)	13.64
	✓	×	12.06 (〃)	9.67 (〃)	17.99 (〃)	13.24
	×	240h	11.71 (441)	9.40 (534)	18.21 (782)	13.11
	✓	240h	11.27 (〃)	8.85 (〃)	17.20 (〃)	12.44
	×	600h	11.70 (429)	9.29 (518)	17.54 (788)	12.85
	✓	600h	**11.27** (〃)	**8.77** (〃)	**16.57** (〃)	**12.21**

contexts as mentioned in Section 3.1. Both external RNNLMs trained with the original training data (300h) and that concatenated with the additional training data (2000h) improved the performances in Switchboard subset, and the latter improved the performances in CallHome subset. Moreover, the number of detected OOV words in the hypotheses was increased by *shallow fusion* with the external RNNLMs. This is because the vocabulary of RNNLMs was limited to that of the A2W models, and high probabilities were assigned to the OOV class. The MTL with the A2C model improved the baseline performance, and resolving OOV words boosted it as in our previous work [12]. External RNNLMs improved the performances of all models except for the one trained on 300h text data for the baseline A2W model. The MTL approach encouraged the effectiveness of *shallow fusion* thanks to alleviating data sparseness issues. In addition, *shallow fusion* enhanced the improvements by the OOV resolution. This suggests that the external LM helps detect OOV words more accurately. In summary, compared to the baseline A2W model with *shallow fusion*, the further combination with the A2C model and the OOV resolution method yielded absolute 1.24 (6.75% relative), and 1.42 (3.72% relative) gains in Switchboard (SWB) and CallHome (CH) subsets, respectively.

4.2. Corpus of spontaneous Japanese (CSJ)

4.2.1. System settings

The Corpus of Spontaneous Japanese (CSJ) [38] is one of the largest Japanese spontaneous speech corpora. The CSJ consists of about 600-hour spontaneous speech including academic and simulated lectures. In this paper, we focus on the academic lectures which have been the primary target of ASR research using this corpus, consisting of about 240 hours of training data in total. There are three evaluation sets (*eval1*, *eval2*, and *eval3*), each of which is composed of 10 lectures and the *eval3* set is regarded as an out-of-domain test set. We picked up the first 4k utterances from the training set as a validation set separately following Kaldi recipe [26]. The final training set has about 155k utterances.

For the A2W models, we restricted the vocabulary to words which occurred at least five times in the training set and replaced the rest to a single OOV class. The resulting vocabulary size was about 12.5k words, and the OOV rates of *eval1*, *eval2*, and *eval3* were 1.24, 1.66, and 4.09, respectively. For the A2C model, we used the 2820 kinds of standard Japanese characters including kanji, hiragana, and katakana characters, alphabets, digits, noise, space, and the end of sentence mark. Output words and characters were embedded in the fixed dimension of size 128 and 64, respectively. The topology and training scheme of RNNLMs were the same as in Section 4.1. The rest of the configurations was the same as in Section 4.1.

RNNLMs were composed of two layers of unidirectional LSTM with 512 memory cells and have residual connections between two LSTM layers [34]. Input and output embeddings were tied as in [35, 36]. RNNLMs were optimized using back-propagation through time (BPTT) with a sequence length of 100. We used the same transcriptions as ASR models (240-hour) and also those appended with additional text data of the simulated lectures (totally 600-hour) for training RNNLMs.

4.2.2. Results

The results are shown in Table 2. As in Section 4.1.2, the attention-based A2W model outperformed the word CTC model. In contrast, the A2C model outperformed the A2W model. This is because the length of characters per word in Japanese is shorter than that in English, so it is easy for the A2C model to capture linguistic dependencies from the target character sequence. The MTL of the A2W model with the A2C model improved the performance, and outperformed the A2C model. Resolving OOV words further improved the performance, which is consistent with results in Section 4.1.2.

Both external RNNLMs trained with the original training data (240h) and that concatenated with the additional training data (600h) improved performances in all test sets in proportion to the training data size for LMs. Adding the out-of-domain data to the training data for the external RNNLM alleviated the domain mismatch in *eval3* test set to some extent. The MTL alleviated data sparseness problems, and then the effect of *shallow fusion* was emphasized. *shallow fusion* also increased the number of detected OOV words in the hypotheses in this corpus, and this left room for the improvements by recovering OOV words by the A2C model. In summary, the combination of the MTL with the A2C model, resolving OOV words and *shallow fusion* with the external RNNLM showed the best WER in three test sets, especially for the out-of-domain scenario (*eval3*). Compared to the baseline A2W model with *shallow fusion*, our best model yielded absolute 0.84 (6.93% relative), 0.88 (9.11% relative) and 2.14 (11.43% relative) gains in each test subset, respectively.

Next, we changed the vocabulary size of the A2W models (see Figure 1 and 2). The OOV rates in each vocabulary are shown in Table 3. With the smaller vocabulary size, WER was drastically degraded due to the increase of the OOV rates in the test sets. However,

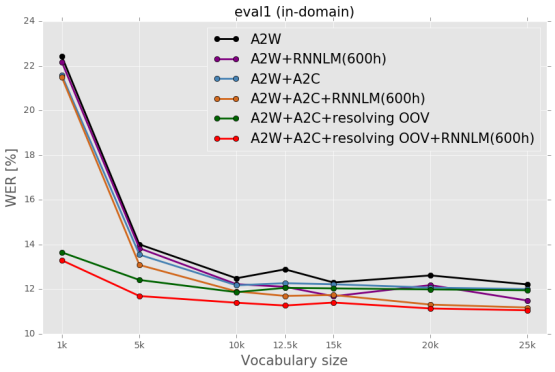

Fig. 1: Recognition performances with various vocabulary sizes in *eval1* test set. *eval1* is regarded as an in-domain test set.

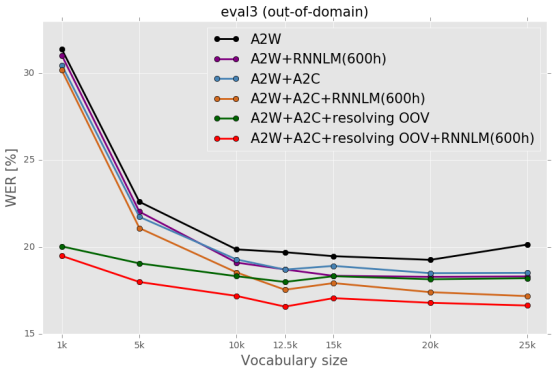

Fig. 2: Recognition performances with various vocabulary sizes in *eval3* test set. *eval3* is regarded as an out-of-domain test set.

the MTL approach with OOV resolution mitigated this problem and was robust to the vocabulary size. In the A2W models, the gain by the external RNNLMs was trivial with the small vocabulary. In contrast, external RNNLMs were always effective in case of using the OOV resolution even with the very small vocabulary such as 1k and 5k. The best results were obtained with vocabulary 15k, but the gaps of the performances between 5k and 15k were 0.30 and 0.94 in *eval1* and *eval3* test sets, respectively. Therefore, we can reduce the vocabulary size three times with the small performance degradation.

Finally, the real time factor (RTF) of the A2W models in *eval1* test set are shown in Figure 3. Decoding was performed with a single NVIDIA Titan GPU. Our attention-based models use the bidirectional encoders, but RTF is small enough for the real-time usage. When using a large vocabulary, there is few additional time for resolving OOV words with the external RNNLM because there are few OOV words. In contrast, when using a small vocabulary, the costs of the OOV resolution is more expensive than those of using external RNNLMs. However, note that all of the A2W models are always faster than the A2C model although some of them use the external RNNLM during decoding.

5. CONCLUSIONS

We have addressed an issue that the acoustic-to-word (A2W) model tends to incorrectly recognize OOV words as in-vocabulary words. Joint decoding with the external language model helps the A2W

Table 3: The OOV rates of in-domain (*eval1*) and out-of-domain (*eval3*) test sets in CSJ corpus (%).

Vocabulary size	*eval1*	*eval3*
1k	11.87	18.86
5k	3.14	8.20
10k	1.65	4.79
15k	1.30	3.64
20k	0.99	3.10
25k	0.92	2.69

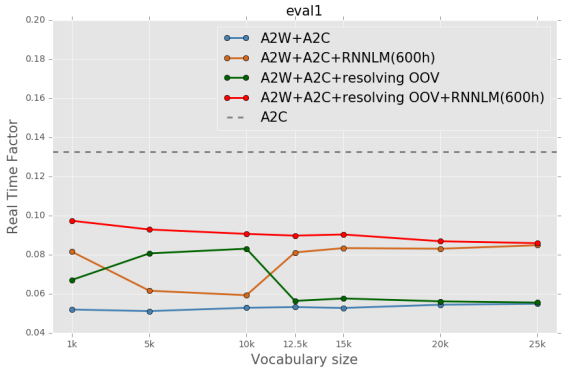

Fig. 3: Real time factor in *eval1* test set.

model detect OOV words more accurately because it has more reliable linguistic information. These OOV words can be recovered by the character-level decoder which attached to the same encoder as the A2W model in the multi-task learning (MTL) framework. We experimentally confirmed that external LMs encouraged the OOV prediction, and recovering OOV words further improved the performance. We also found that MTL alleviates data sparseness issues to some extent, and then the effectiveness of the LM integration is enhanced. In addition, by resorting the recognition of rare words to the character-level decoder, the A2W models can work with a small vocabulary size.

6. REFERENCES

[1] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, et al., "English conversational telephone speech recognition by humans and machines," in *Proc. of INTERSPEECH*, 2017, pp. 132–136.

[2] Andreas Stolcke and Jasha Droppo, "Comparing human and machine errors in conversational speech transcription," in *Proc. of INTERSPEECH*, 2017, pp. 137–141.

[3] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. of ICML*, 2006, pp. 369–376.

[4] Alex Graves, Abdelrahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. of ICASSP*, 2013, pp. 6645–6649.

[5] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "EESEN: End-to-end speech recognition using deep RNN

models and WFST-based decoding," in *Proc. of ASRU*, 2015, pp. 167–174.

[6] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[7] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Proc. of NIPS*, 2015, pp. 577–585.

[8] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. of ICASSP*, 2016, pp. 4960–4964.

[9] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. of ICASSP*, 2016, pp. 4945–4949.

[10] Rohit Prabhavalkar, Kanishka Rao, Tara N Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. of INTERSPEECH*, 2017, pp. 939–943.

[11] Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu, "Exploring neural transducers for end-to-end speech recognition," in *Proc. of ASRU*, 2017, pp. 206–213.

[12] Sei Ueno, Hirofumi Inaguma, Masato Mimura, and Tatsuya Kawahara, "Acoustic-to-word attention-based model complemented with character-level CTC-based model," in *Proc. of ICASSP*, 2018, pp. 5804–5808.

[13] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Katya Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. of ICASSP*, 2018, pp. 4774–4778.

[14] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Proc. of INTERSPEECH*, 2015, pp. 1468–1472.

[15] Kartik Audhkhasi, Bhuvana Ramabhadran, George Saon, Michael Picheny, and David Nahamoo, "Direct acoustics-to-word models for English conversational speech recognition," in *Proc. of INTERSPEECH*, 2017, pp. 959–963.

[16] Kartik Audhkhasi, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Michael Picheny, "Building competitive direct acoustics-to-word models for English conversational speech recognition," pp. 4759–4763, 2018.

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

[18] Jinyu Li, Guoli Ye, Rui Zhao, Jasha Droppo, and Yifan Gong, "Acoustic-to-word model without OOV," in *Proc. of ASRU*, 2017, pp. 111–117.

[19] Liang Lu, Xingxing Zhang, and Steve Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *Proc. of ICASSP*, 2016, pp. 5060–5064.

[20] Jinyu Li, Guoli Ye, Amit Das, Rui Zhao, and Yifan Gong, "Advancing acoustic-to-word CTC model," in *Proc. of ICASSP*, 2018, pp. 5794–5798.

[21] Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhifeng Chen, and Rohit Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *Proc. of ICASSP*, 2017, pp. 5824–5828.

[22] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu, "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," in *Proc. of INTERSPEECH*, 2017, pp. 3532–3536.

[24] Yonatan Belinkov and James Glass, "Analyzing hidden representations in end-to-end automatic speech recognition systems," in *Proc. of NIPS*, 2017, pp. 2438–2448.

[25] John J Godfrey, Edward C Holliman, and Jane McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. of ICASSP*, 1992, pp. 517–520.

[26] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *Proc. of ASRU*, 2011.

[27] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. of ICASSP*, 2017, pp. 4835–4839.

[28] Jan Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proc. of INTERSPEECH*, 2017, pp. 523–527.

[29] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR*, 2015.

[30] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[31] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *Proc. of ICML*, 2013, pp. 1310–1318.

[32] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Preceedings of NIPS*, 2015, pp. 1171–1179.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of CVPR*, 2016, pp. 2818–2826.

[34] Gakuto Kurata, Abhinav Sethy, Bhuvana Ramabhadran, and George Saon, "Empirical exploration of novel architectures and objectives for language models," in *Proc. of INTERSPEECH*, 2017, pp. 279–283.

[35] Hakan Inan, Khashayar Khosravi, and Richard Socher, "Tying word vectors and word classifiers: A loss framework for language modeling," *arXiv preprint arXiv:1611.01462*, 2016.

[36] Ofir Press and Lior Wolf, "Using the output embedding to improve language models," in *Procs. of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 157–163.

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[38] Kikuo Maekawa, "Corpus of spontaneous japanese: Its design and evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.