MINIMUM BAYES RISK TRAINING OF CTC ACOUSTIC MODELS IN MAXIMUM A POSTERIORI BASED DECODING FRAMEWORK

Naoyuki Kanda, Xugang Lu, Hisashi Kawai

National Institute of Information and Communications Technology, Japan

{naoyuki.kanda,xugang.lu,hisashi.kawai}@nict.go.jp

ABSTRACT

When using connectionist temporal classification (CTC) based acoustic models (AMs) for large vocabulary continuous speech recognition (LVCSR), most previous studies have used a naive interpolation of the CTC-AM score and an additional language model score, although there is no theoretical justification for such an approach. On the other hand, we recently proposed a theoretically more sound decoding framework for CTC-AM called maximum a posteriori (MAP)-based decoding. Although the superiority of the MAP-based decoding framework with CTC-AM has been demonstrated, the effect of additional minimum Bayes risk (MBR) training in the MAP-based decoding framework has not been investigated. In this paper, we report the results of various experiments that examine the effect of MBR training on CTC-AM by comparing two decoding frameworks. Our experiments with English and Japanese LVCSR tasks reveal that the MAP-based decoding framework is superior to the interpolation-based framework, even after the MBR training. In addition, by using about 600 h of training data, we show that the size of the training dataset is a critical factor in achieving good results under CTC-AM.

Index Terms— Connectionist temporal classification, deep neural network, acoustic model

1. INTRODUCTION

Acoustic models (AMs) that use connectionist temporal classification (CTC) [1, 2] have recently been proposed as an alternative to those based on hidden Markov models (HMMs) [3–5]. CTC aims to learn a mapping from an observation sequence $\mathbf{X}$ to a target symbol sequence $\mathbf{s}$ directly. CTC was initially proposed for phoneme recognition [1, 2] and recently successfully applied to large vocabulary continuous speech recognition (LVCSR) tasks [6–14].

Because the original training criterion of CTC-AM is based on maximizing the log posterior $logP(\mathbf{s}|\mathbf{X})$ of the target symbol sequence $\mathbf{s}$, it does not necessarily maximize the final recognition accuracy when decoding with an additional language model (LM). Thus, it would be better to use a training criterion that directly optimizes the decoding accuracy, as in the case of the sequence discriminative training of deep neural network (DNN)-HMM hybrids such as maximum mutual information (MMI) and state-level minimum Bayes risk (sMBR) training [15, 16].

One important point to note in defining such kind of training criterion of CTC is that there are two different decoding frameworks for CTC-AM: (i) interpolation-based decoding [6] and (ii) maximum a posteriori (MAP)-based decoding [17]. The interpolation based decoding framework is very simple – just use an interpolation score of CTC score $P(\mathbf{s}|\mathbf{X})$ and LM score $P(\mathbf{W})$ when searching for the best hypothesis. Although there is no theoretical justification for such an approach, this framework is widely used because of its simplicity [6–14]. The minimum Bayes risk (MBR) training of CTC-AM has already been investigated based on this interpolation-based framework, and showed large improvements in recognition accuracy [9, 10, 14].

On the other hand, we recently proposed a theoretically more sound decoding framework called MAP-based decoding [17]. The MAP-based decoding aims to maximize the word posterior $P(\mathbf{W}|\mathbf{X})$, as in the decoding framework for HMM-based AMs. In various LVCSR experiments, MAP-based decoding achieved consistent improvements over the conventional interpolation-based decoding [17]. However, previous investigations of MAP-based decoding were all based on normally trained (i.e., not MBR trained) CTC-AM. The effect of MBR training on CTC-AM in the MAP-based decoding framework has not been investigated.

In this paper, we investigate the MBR training of CTC-AM, comparing the interpolation-based and MAP-based decoding frameworks. The contributions of this paper are summarized as follows:

- We formulate the MBR training of CTC in the MAP-based decoding framework. It is important to note that previous investigations on the MBR training of CTC-AM [9, 10, 14] were all based on the interpolation-based framework.
- We present various experimental results for the MBR-trained CTC-AM on English and Japanese LVCSR tasks. Our experiments reveal that the MAP-based decoding framework is still better than the interpolation-based framework, even after MBR training. In addition, by using about 600 h of training data, we show that the size of the training dataset is a critical factor in achieving good results under CTC-AM, as is consistent with the prior reports [7, 13].

In the next section, we introduce CTC and its original training method. Then, we explain the interpolation-based and MAP-based decoding frameworks in Section 3. The MBR training of CTC (for both decoding frameworks) is formulated in Section 4. Finally, in Section 5, we present various experimental results from English and Japanese LVCSR tasks.

2. CTC-AM

2.1. Model structure

Given a frame-wise feature sequence $\mathbf{X}$ and a target subword sequence $\mathbf{s}$ (e.g., characters, phonemes), the goal of CTC-AM is to train a neural network that represents the posterior probability $P(\mathbf{s}|\mathbf{X})$. Toward this goal, an additional *blank* label $\emptyset$ is first introduced into the set of subword units (CTC-label) to compensate for the difference in length between $\mathbf{s}$ and $\mathbf{X}$. Then, the posterior probability of the CTC-label sequence $\mathbf{c} = \{c_1, ..., c_T\}$ for a given observation $\mathbf{X}$ is modeled by the frame-wise product of the neural

network's output as follows.

$$P(\mathbf{c}|\mathbf{X}) = \prod_{t=1}^{T} y_t(c_t). \quad (1)$$

Here, $y_t(c_t)$ is the output score of the neural network for CTC-label c_t at time frame t, where the output layer consists of the softmax activation function.

Next, a *collapsing function* $\Phi()$ is introduced to map the frame-wise CTC-label sequence $\mathbf{c}$ into the target subword sequence $\mathbf{s}$. This function converts the repetition of the CTC-label into one symbol, removing the blank label $\emptyset$. For example, the CTC-label sequences "$AA\emptyset B\emptyset CC\emptyset$" and "$\emptyset A\emptyset BB\emptyset C\emptyset$" are both mapped to the subword sequence "ABC" by applying Φ. Based on the collapsing function, the posterior probability of the subword sequence $\mathbf{s}$ given the observation $\mathbf{X}$ is finally modeled as

$$P(\mathbf{s}|\mathbf{X}) = \sum_{\mathbf{c}\in\Phi^{-1}(\mathbf{s})} P(\mathbf{c}|\mathbf{X}). \quad (2)$$

2.2. Maximum log-probability-based training of CTC

The conventional training criterion for CTC is defined as the log-probability over the entire set of training samples as follows:

$$\mathcal{F}^{CTC} = \sum_u \log P(\mathbf{s}_u|\mathbf{X}_u), \quad (3)$$

where u is an index of the training samples. The parameters are estimated so as to maximize $\mathcal{F}^{CTC}$ (or minimize $-\mathcal{F}^{CTC}$, also known as the CTC-loss).

The error signal w.r.t. the activation of the final softmax layer is calculated as follows:

$$\begin{aligned} e_u^{CTC}(c,t) &= \frac{\partial \mathcal{F}^{CTC}}{\partial a_t^u(c)} = \sum_{c'} \frac{\partial \mathcal{F}^{CTC}}{\partial y_t^u(c')} \frac{\partial y_t^u(c')}{\partial a_t^u(c)} \\ &= \frac{\sum_{\mathbf{c}\in\Phi^{-1}(\mathbf{s}_u)} \delta_{c_t,c} P(\mathbf{c}|\mathbf{X}_u)}{P(\mathbf{s}_u|\mathbf{X}_u)} - y_t^u(c), \end{aligned} \quad (4)$$

where $a_t^u(c)$ is the activation of the final softmax layer for CTC-label c at time frame t. The function $\delta_{c_t,c}$ is Kronecker's delta, which takes a value of one if the CTC-label at time frame t ($= c_t$) is c and zero otherwise. Equation (4) can be efficiently calculated using the forward-backward algorithm [1], and is used for the error backpropagation-based training of neural network parameters.

3. DECODING FRAMEWORK FOR CTC-AM

3.1. Interpolation-based decoding framework for CTC

Most previous studies have used a naive logarithmic interpolation of an LM score and a CTC-AM score [6–14]. In this type of framework, the word sequence $\mathbf{W}$ for a given observation $\mathbf{X}$ is estimated as

$$\tilde{\mathbf{W}} = \arg\max_{\mathbf{W}} \{ \max_{\mathbf{s}\in\Psi(\mathbf{W})} P(\mathbf{W}) P(\mathbf{s}|\mathbf{X})^{\alpha} \}, \quad (5)$$

where $\Psi()$ is a function that converts word sequence $\mathbf{W}$ into a set of possible subword sequences $\mathbf{s}$, $P(\mathbf{W})$ is a word-level LM (WLM) probability, and α is a scaling factor for the CTC AM. Practically, a word insertion penalty (denoted by $|\mathbf{W}|$ in [6]) is often used in combination with Eq. (5).

Importantly, there is no theoretical justification for such an interpolation, but this method is widely used because of its simplicity.

3.2. MAP-based decoding framework for CTC

We have recently proposed a more sound decoding framework for CTC-AM called MAP-based decoding [17]. In this framework, the speech recognition problem is defined as the problem of finding the word sequence $\mathbf{W}$ that maximizes the posterior probability $P(\mathbf{W}|\mathbf{X})$ for a given observation $\mathbf{X}$, and transformed as follows:

$$\tilde{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}) \quad (6)$$

$$= \arg\max_{\mathbf{W}} \sum_{\mathbf{s}\in\Psi(\mathbf{W})} P(\mathbf{W}|\mathbf{s}) P(\mathbf{s}|\mathbf{X})^{\alpha} \quad (7)$$

$$\simeq \arg\max_{\mathbf{W}} \{ \max_{\mathbf{s}\in\Psi(\mathbf{W})} P(\mathbf{W}|\mathbf{s}) P(\mathbf{s}|\mathbf{X})^{\alpha} \}, \quad (8)$$

where $P(\mathbf{W}|\mathbf{s})$ is calculated as

$$P(\mathbf{W}|\mathbf{s}) = \frac{P(\mathbf{s}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{s})^{\beta}}. \quad (9)$$

Here, $P(\mathbf{s})$ is a subword LM (SLM) probability and β is its scaling factor. The SLM probability $P(\mathbf{s})$ can be estimated from the training label for CTC-AM using conventional language modeling techniques such as the N-gram model. The term $P(\mathbf{s}|\mathbf{W})$ is a word–subword conversion probability, which can be modeled by a conventional word pronunciation dictionary.

Note that the final formula (Eq. (8)) is tailored for CTC-AM, whereas the starting point of the formula (Eq. (6)) is the same as the HMM-based decoding framework. We have shown that the MAP-based decoding framework can achieve consistently and substantially better results than the interpolation-based framework when decoding with normally trained CTC-AM [17].

4. MBR TRAINING OF CTC-AM

Maximizing the conventional CTC-training criterion $\mathcal{F}^{CTC}$ does not necessarily maximize the word recognition accuracy when decoding with LM. Thus, it would be better to use a criterion that directly optimizes the decoding accuracy. Using the analogy of sMBR training for DNN-HMM [15], the parameters in CTC-AM can be trained to maximize the expected recognition accuracy $\mathcal{F}^{MBR}$, which is defined as

$$\mathcal{F}^{MBR} = \sum_u \sum_{\mathbf{W}} P_*(\mathbf{W}|\mathbf{X}_u) \mathcal{A}(\mathbf{W}, \mathbf{W}_u). \quad (10)$$

Here, $P_*(\mathbf{W}|\mathbf{X}_u)$ is a posterior probability estimated by current models (defined later). The term $\mathcal{A}(\mathbf{W}, \mathbf{W}_u)$ is the accuracy of the hypothesis $\mathbf{W}$ compared with the reference label $\mathbf{W}_u$. In this paper, we count the frame-wise coincidence of the CTC-label between the hypothesis and reference as follows:

$$\mathcal{A}(\mathbf{W}, \mathbf{W}_u) := \sum_t \delta_{c_t, c_t^u}, \quad (11)$$

where c_t^u indicates the CTC-label in the reference alignment at time frame t.

The calculation of posterior probability $P_*(\mathbf{W}|\mathbf{X}_u)$ in Eq (10) must be strictly matched with the decoding framework that CTC-AM is used in. In the case of the conventional interpolation-based decoding framework, $P_*(\mathbf{W}|\mathbf{X}_u)$ should be calculated as

$$P_{int}(\mathbf{W}|\mathbf{X}_u) = \frac{1}{\mathcal{Z}_{int}^u} [\max_{\mathbf{s}\in\Psi(\mathbf{W})} P(\mathbf{W}) P(\mathbf{s}|\mathbf{X}_u)^{\alpha}], \quad (12)$$

where $\mathcal{Z}_{int}^u = \sum_{\mathbf{W}} \max_{\mathbf{s}\in\Psi(\mathbf{W})} P(\mathbf{W}) P(\mathbf{s}|\mathbf{X}_u)^{\alpha}$ is a normalization term to ensure $\sum_{\mathbf{W}} P_{int}(\mathbf{W}|\mathbf{X}_u) = 1$. On the other hand, in

the case of the MAP-based decoding framework, $P_*(\mathbf{W}|\mathbf{X}_u)$ should be calculated as

$$P_{map}(\mathbf{W}|\mathbf{X}_u) = \frac{1}{\mathcal{Z}^u_{map}}[\max_{\mathbf{s}\in\Psi(\mathbf{W})} P(\mathbf{W}|\mathbf{s})P(\mathbf{s}|\mathbf{X}_u)^\alpha], \quad (13)$$

where $\mathcal{Z}^u_{map} = \sum_{\mathbf{W}} \max_{\mathbf{s}\in\Psi(\mathbf{W})} P(\mathbf{W}|\mathbf{s})P(\mathbf{s}|\mathbf{X}_u)^\alpha$ is a normalization term to ensure $\sum_{\mathbf{W}} P_{map}(\mathbf{W}|\mathbf{X}_u) = 1$.

By differentiating $\mathcal{F}^{MBR}$ with respect to $y^u_t(c)$, we obtain

$$\frac{\partial \mathcal{F}^{MBR}}{\partial y^u_t(c)} = \frac{\alpha\gamma^u_t(c)}{y^u_t(c)}\{\bar{\mathcal{A}}_u(c) - \bar{\mathcal{A}}_u\}, \quad (14)$$

where

$$\gamma^u_t(c) = \sum_{\mathbf{W}} \delta_{c_t,c} P_*(\mathbf{W}|\mathbf{X}_u), \quad (15)$$

$$\bar{\mathcal{A}}_u(c) = \frac{\sum_{\mathbf{W}} \delta_{c_t,c} P_*(\mathbf{W}|\mathbf{X}_u)\mathcal{A}(\mathbf{W},\mathbf{W}_u)}{\sum_{\mathbf{W}} \delta_{c_t,c} P_*(\mathbf{W}|\mathbf{X}_u)}, \quad (16)$$

$$\bar{\mathcal{A}}_u = \sum_{\mathbf{W}} P_*(\mathbf{W}|\mathbf{X}_u)\mathcal{A}(\mathbf{W},\mathbf{W}_u). \quad (17)$$

Then, the error signal w.r.t. the activation $a^u_t(c)$ of the final softmax layer is calculated as

$$e^{MBR}_u(c,t) = \frac{\partial \mathcal{F}^{MBR}}{\partial a^u_t(c)} = \sum_{c'} \frac{\partial \mathcal{F}^{MBR}}{\partial y^u_t(c')} \cdot \frac{\partial y^u_t(c')}{\partial a^u_t(c)}$$
$$= \alpha\gamma^u_t(c)\{\bar{\mathcal{A}}_u(c) - \bar{\mathcal{A}}_u\} \quad (18)$$

Note that Eqs. (14)–(18) are applicable for both $P_{int}(\mathbf{W}|\mathbf{X}_u)$ and $P_{map}(\mathbf{W}|\mathbf{X}_u)$. Equation (18) can be efficiently calculated by the forward-backward algorithm over the generated lattices, similar to the sMBR training of DNN-HMM [15]. The only differences from the sMBR training for DNN-HMM are (1) the use of $P_{int}(\mathbf{W}|\mathbf{X}_u)$ or $P_{map}(\mathbf{W}|\mathbf{X}_u)$ instead of a DNN-HMM based posterior probability, and (2) the use of the CTC-label-based lattices instead of the hidden-state-based lattices for the error calculation.

5. EXPERIMENTS

5.1. WSJ experiment

5.1.1. Experimental settings

The first experiment was conducted on the Wall Street Journal (WSJ) corpus, known as LDC93S6B and LDC94S13B. We followed the experimental settings in [12] by using the EESEN software[1] developed by the authors of that paper.

The training data were prepared according to the recipe in EESEN, which gave us 77.5 h of training data with 3.8 h of cross-validation data. A phoneme-based bidirectional long-short-term memory (BLSTM) with four hidden layers, each comprising 320 nodes, was trained on the 120-dimensional filter-bank features (40 filter-bank features+Δ + $\Delta\Delta$) with mean and variance normalization (MVN). First, the BLSTM was trained from scratch based on $\mathcal{F}^{CTC}$. The initial learning rate and momentum parameter were set to 0.00004 and 0.95, respectively. After training the CTC-BLSTM AM, lattices were generated based on the AM using a 1-gram WLM trained from transcriptions in the training data with a scaling factor of $\alpha = 1.0$. In addition, when generating the lattices for the MAP-based framework, a bigram SLM that had been trained using the phoneme-converted transcription of the training data for AMs was used with $\beta = 0.5$. Finally, five epochs of MBR training were conducted with a fixed learning rate of 0.000001 and a momentum parameter of 0.9.

[1] https://github.com/srvk/eesen

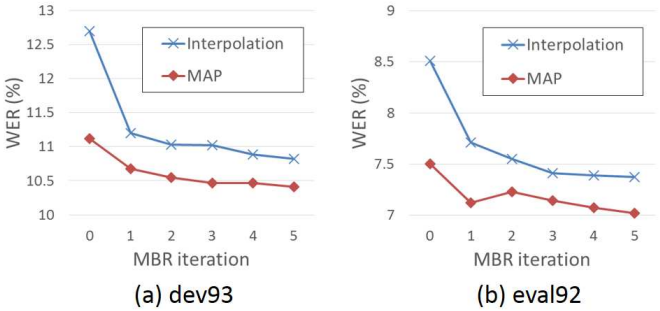

Fig. 1. *Effect of MBR training for WSJ test set.*

Table 1. *WER of various networks for WSJ eval92.*

AM	LM	Framework	WER (%)
phone-CTC	3-gram	Interpolation	8.5
		MAP	7.5
phone-MBR-CTC	3-gram	Interpolation	7.4
		MAP	**7.0**
Miao et al. [12]			
CE-DNN-HMM	3-gram	-	7.1
phn-CTC	Dictionary	Interpolation $^{(*)}$	26.9
phn-CTC	3-gram	Interpolation$^{(*)}$	7.9
char-CTC	3-gram	Interpolation$^{(*)}$	9.1
Bahdanau et al. [18]			
char-Enc-Dec	3-gram	Interpolation	10.8

$(*)$ A frame-wise prior was applied on the CTC score.

For the evaluation, the WSJ standard pruned trigram LM (20K vocabularies) was used for WLM. In addition, a bigram SLM was used for the MAP-based decoding in accordance with our previous investigation [17]. When decoding, the parameters (scaling factors α, β and word insertion penalty) were tuned by "dev93" and the best parameters were used to decode "eval92".

5.1.2. Results

The effects of MBR training with "dev93" and "eval92" are shown in Fig. 1. In this figure, the 0-iteration of MBR-training denotes the normal CTC-AM trained based on $\mathcal{F}^{CTC}$. When decoding with the normal CTC-AM, the MAP-based decoding framework achieved a much better word error rate (WER) (7.5%) than the interpolation-based decoding (8.5%). By applying the MBR training, the WERs were further improved in both conditions, and the MAP-based decoding still achieved a better WER, even after the MBR training. In this dataset, simply applying MAP-based decoding on normal CTC-AM achieved almost the same effect as the (more complicated) MBR training for interpolation-based decoding.

Compared with the interpolation-based decoding framework, the improvement offered by MBR training in the MAP-based decoding framework was smaller. This was within our expectation. Our interpretation is as follows – In the interpolation-based decoding framework, there was a huge mismatch between what CTC-AM calculated and how the hypothesis score was calculated. Applying MBR-training in the interpolation-based framework resolves this mismatch to some extent, as reflected in the large improvement in WERs. On the contrary, because MAP-based decoding already uses the CTC-AM score in a sound way, and because the training criterion $\mathcal{F}^{CTC}$ is already a "sequence discriminative" criterion, there is less room for improvement than in the interpolation-based framework.

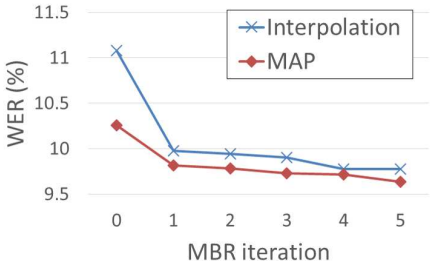

Fig. 2. *Effect of MBR training for CSJ test set (avg.).*

Table 2. *WER of CSJ (avg.) with various N-gram order of WLM in lattice generation for the MBR training.*

Framework	no-MBR	1-gram	2-gram	3-gram
Interpolation	11.08	10.18	9.77	9.83
MAP	10.26	10.02	**9.63**	9.66

In Table 1, we have listed the results for "eval92" against those of representative previous methods using end-to-end AMs [12, 18]. To the best of our knowledge, our WSJ result is the best among the literatures using end-to-end type AMs. Although the WER in the table is not competitive with state-of-the-art results using DNN-HMM, the next section shows that a large training data is essential to achieve good results with CTC-AM.

5.2. CSJ experiment

5.2.1. Experimental settings

We also conducted an evaluation using the "Corpus of Spontaneous Japanese" (CSJ) [19], which consists of over 600 h of lecture recordings. The corpus contains three official evaluation sets (E1, E2, and E3), each comprising 10 lecture recordings. We selected 10 lecture recordings as the development set to tune the system parameters, and used the rest of the data in CSJ (591 h of lecture recordings) for the training process.

As the baseline model, we trained a DNN-HMM with five hidden layers, each comprising 2,048 nodes. The output layer had 8,407 nodes, corresponding to the clustered context-dependent phoneme HMM states. As acoustic features, we used 72-dimensional filter-bank features (24 filter-bank features$+\Delta+\Delta\Delta$) with MVN applied to each speaker. The features of both the previous and subsequent seven frames were concatenated when input to the DNNs. The DNN was initialized using discriminative pre-training [20] and was fine-tuned using stochastic gradient descent based on the cross-entropy (CE) loss criterion. After training the CE-DNN, we additionally conducted five epochs of sMBR training [15].

The CTC-BLSTM was then trained based on the same 72-dimensional filter-bank features with no splicing. In this experiment, we used 263 Japanese syllables (or "kana") for the recognition unit. A BLSTM with five hidden layers, each comprising 320 nodes, was used. The BLSTM was first trained from scratch based on $\mathcal{F}^{CTC}$. The initial learning rate and momentum parameter were set to 0.00004 and 0.9, respectively. After training the CTC-BLSTM AM, we conducted MBR training starting from the AM. A similar recipe as for the WSJ experiment was used for MBR training, but with a 2-gram WLM used for lattice generation. The effect of the order of the WLM in lattice generation will be discussed later.

For the evaluation, we trained a 4-gram WLM from the transcription of the training data with Kneser–Ney smoothing [21]. The WLM had a vocabulary size of 98K words. We also trained a 2-gram SLM for the MAP-based decoding from the syllable-level transcription of the training data. When decoding, we tuned the scaling factors α, β, and the word insertion penalty using the development set. The best parameters were then used to decode the evaluation sets.

Table 3. *WERs for the CSJ test sets.*

AM	Decoding Framework	WER (%) E1	E2	E3	E (avg.)
CE-DNN-HMM	-	12.19	9.80	11.01	11.00
sMBR-DNN-HMM	-	11.17	9.05	9.92	10.05
CTC	Interpolation	12.81	9.75	10.67	11.08
	MAP	11.88	9.23	9.67	10.26
MBR-CTC	Interpolation	11.38	8.85	9.09	9.77
	MAP	**11.07**	**8.79**	**9.04**	**9.63**

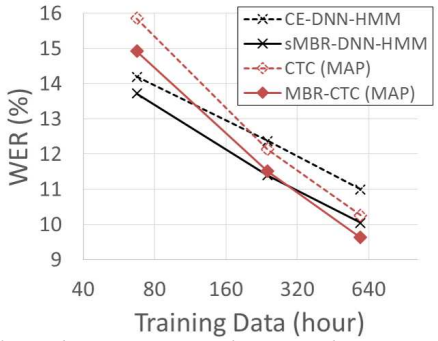

Fig. 3. *Relation between WER and training data size in CSJ (avg.).*

5.2.2. Results

The effect of the MBR training is shown in Fig. 2. As in the WSJ experiment, MBR training substantially improved the WER, and MAP-based decoding achieved better results than the interpolation-based decoding framework. Table 2 demonstrates the effect of the N-gram order of the WLM when generating the lattices for the MBR-training. In this dataset, 2-gram or 3-gram WLMs achieved slightly better results than the 1-gram model. A similar phenomenon was reported for the sequence discriminative training of DNN-HMM [22]. Note that the 2-gram WLM was also used for the sMBR training of DNN-HMM in our experiments.

Table 3 lists the detailed results for various AMs and decoding frameworks. Although the differences between the two decoding frameworks became small after the MBR training, MAP-based decoding consistently achieved better results for all evaluation sets. In addition, the MBR-CTC AM with the MAP-based decoding framework outperformed the sMBR-DNN-HMM.

Finally, to evaluate the effect of training data size, we conducted the same experiment with 67-h and 240-h subsets of the training data. The results are plotted in Fig. 3. Although CTC-AM was much worse than DNN-HMM when the data size was small, CTC-AM showed large reduction of WERs according to the increase of data, finally producing better results with 591 h of training data. This strongly suggests that CTC-AM will achieve much better results than DNN-HMM when more training data are added.

6. CONCLUSION

In this paper, we investigated the MBR-training of CTC-AM based on the MAP-based decoding framework. In our experiments, the MAP based decoding framework consistently achieved better results than the conventional interpolation-based framework, even after the MBR training. We also showed that the size of the training data is a critical factor in achieving good results under CTC-AM.

7. REFERENCES

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

[2] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[3] Frank Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks.," in *Proc. INTERSPEECH*, 2011, pp. 437–440.

[4] George E Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. SAP*, vol. 20, no. 1, pp. 30–42, 2012.

[5] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[6] Awni Y Hannun, Andrew L Maas, Daniel Jurafsky, and Andrew Y Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns," *arXiv preprint arXiv:1408.2873*, 2014.

[7] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., "Deepspeech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

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

[9] Hasim Sak, Andrew Senior, Kanishka Rao, Ozan Irsoy, Alex Graves, Françoise Beaufays, and Johan Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 4280–4284.

[10] Andrew Senior, Hasim Sak, Felix de Chaumont Quitry, Tara N. Sainath, and Kanishka Rao, "Acoustic modelling with CD-CTC-SMBR LSTM RNNs," in *Proc. ASRU*, 2015, pp. 604–609.

[11] Andrew L Maas, Ziang Xie, Dan Jurafsky, and Andrew Y Ng, "Lexicon-free conversational speech recognition with neural networks," in *Proc. NAACL HTL*, 2015.

[12] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. ASRU*, 2015, pp. 167–174.

[13] Yajie Miao, Mohammad Gowayyed, Xingyu Na, Tom Ko, Florian Metze, and Alexander Waibel, "An empirical exploration of CTC acoustic models," in *Proc. ICASSP*. IEEE, 2016, pp. 2623–2627.

[14] Ian McGraw, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif, Alexander Gruenstein, Carolina Parada, et al., "Personalized speech recognition on mobile devices," in *Proc. ICASSP*, 2016, pp. 5955–5959.

[15] Karel Veselỳ, Arnab Ghoshal, Lukás Burget, and Daniel Povey, "Sequence-discriminative training of deep neural networks.," in *Proc. INTERSPEECH*, 2013, pp. 2345–2349.

[16] Hang Su, Gang Li, Dong Yu, and Frank Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription.," in *Proc. ICASSP*, 2013, pp. 6664–6668.

[17] Naoyuki Kanda, Xugang Lu, and Hisashi Kawai, "Maximum a posteriori based decoding for CTC acoustic models," in *Proc. INTERSPEECH*, 2016, pp. 1868–1872.

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

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

[20] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011, pp. 24–29.

[21] Stanley F. Chen and Joshua Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–393, 1999.

[22] Hasim Sak, Oriol Vinyals, Georg Heigold, Andrew Senior, Erik McDermott, Rajat Monga, and Mark Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," *Proc. INTERSPEECH*, pp. 1209–1213, 2014.