CROSS-MODAL LEARNING FOR CTC-BASED ASR: LEVERAGING CTC-BERTSCORE AND SEQUENCE-LEVEL TRAINING

Mun-Hak Lee[1]*, Sang-Eon Lee*[1]*, Ji-Eun Choi*[2] *and Joon-Hyuk Chang*[1]

Hanyang University,
Department of Electronics Engineering[1],
Department of Artificial Intelligence[2],
Seoul, Republic of Korea

ABSTRACT

Due to the nature of neural networks that easily overfit the training set, neural network-based speech recognition models are vulnerable to prior shifts in data distribution or unseen words. Therefore, studies have been conducted to overcome this problem by using language models trained with a relatively easy-to-obtain unpaired corpus. In this paper, we present a new training method that uses BERT to improve the performance of a connectionist temporal classification (CTC)-based ASR model. The proposed method follows a cross-modal learning scenario and induces the CTC model to better embed contextual information by utilizing an auxiliary objective function operating at the sequence level. We applied the proposed method to fine-tune the pre-trained wav2vec 2.0 model with CTC loss and confirmed that the proposed method improves the generalization performance of the ASR model.

Index Terms— Speech recognition, Connectionist temporal classification, BERT, Cross-modal learning

1. INTRODUCTION

Automatic speech recognition (ASR) aims to determine the most probable token sequence $\hat{Y}$ inherent to given speech X. Using a probabilistic expression, we can express the decoding process of ASR models as:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}} P(Y|X), \tag{1}$$

where $P(Y|X)$ denotes the conditional probability distribution. To build a high-performance ASR model, $P(Y|X)$ must be approximated accurately. Neural network-based models perform this task well. Representative examples include deep neural network-hidden Markov model (DNN-HMM) based hybrid ASR models, attention-based encoder-decoder (AED) models, connectionist temporal classification (CTC) models, and recurrent neural network transducers (RNNT) [1, 2, 3, 4, 5]. The CTC model is an end-to-end ASR model with a simple structure and intuitive training method. Indeed, CTC is trained to maximize the likelihood of all possible alignments a between token sequence Y and acoustic model (AM) posterior.

$$P(Y|X) = \sum_{a \in \beta^{-1}(Y)} P(a|X), \tag{2}$$

where β denotes a function that removes silence and repetitions. Based on the assumption of conditional independence, $P(a|X)$ can be approximated as follows:

$$P(a|X) = \prod_{t=1}^{|a|} P(a_t|X, a_{0:t-1}) P(a_0) \approx \prod_{t=1}^{|a|} P(a_t|X), \tag{3}$$

where a_t denotes the token of time step t in alignment a, $a_{0:t-1}$ denotes all previous tokens before time t, a_0 denotes the start of the sentence token, and $|a|$ denotes the length of alignment a. The conditional independence assumption allows the modeling of $P(a|X)$ with a limited amount of computation; however, it is an incomplete assumption in that it does not model the correlation between tokens [4, 5, 6]. Language models (LMs) can be used to overcome this limitation. Using the Bayes rule, the decoding process of the ASR system in Eq. (1) can be converted as follows:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}} \frac{P(X|Y)P(Y)}{P(X)} \tag{4}$$

An external LM trained using a large corpus approximates $P(Y)$ well, and the performance of the CTC model can be effectively improved through a decoding method. In particular, because masked language models (MLM) and large LMs have shown remarkable potential in natural language processing tasks, the need for research on effective methods for applying them to ASR models has been emphasized. As a representative example, a method using an LM in the first-pass decoding process (prefix beam search) or second-pass rescoring methods has been studied [7, 8, 9, 10, 11].

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00302424). ※ MSIT: Ministry of Science and ICT

979-8-3503-0689-7/23/$31.00 ©2023 IEEE

In addition to using an external LM in the decoding process, training methods that induce the CTC model to better embed long-range semantic information by itself using a pre-trained LM have been studied. [12, 13, 14, 15] proposed methods in which the CTC and BERT models are jointly trained, and [16, 17] suggest methods of training the CTC model in the direction of imitating the representation generated by BERT along the cross-modal knowledge distillation scenario. These methods alleviate the limitations of the CTC model caused by the conditional independence assumption and improve its generalization performance. The proposed CTC training method shares the same motivation as the cross-modal knowledge distillation method or the LM fusion method and follows the training process of cross-modal learning. In the following sections, we introduce existing studies on which our research is based.

2. BACKGROUND

2.1. Cross-Modal Learning Paradigm

Given two instances (x and y) instantiated in two different modalities, cross-modal learning aims to learn the correlations between instances with different modalities using a neural network model. Common cross-modal learning methods consist of the following four steps [18].

1. A representation vector $h^X = g_X(f_X(x))$ is generated on a fine-grained latent space with dimension d_H using a modality-specific encoder $f_X : X \mapsto \mathbb{R}^{d_X}$ and mapping function $g_X : \mathbb{R}^{d_X} \mapsto \mathbb{R}^{d_H}$.

2. $h^Y = g_Y(f_Y(y))$ is generated in the same way using a modality-specific encoder $f_Y : Y \mapsto \mathbb{R}^{d_Y}$ and a mapping function $g_Y : \mathbb{R}^{d_Y} \mapsto \mathbb{R}^{d_H}$.

3. The similarity function $S(\cdot, \cdot)$ is used to estimate the similarity between the two representation vectors ($h^X, h^Y \in \mathbb{R}^{d_H}$).

4. The neural network-based model is trained to increase the similarity between positive pairs and reduce the similarity between negative pairs.

Cross-modal learning methods have shown excellent performance in tasks such as image retrieval [19], video-caption retrieval [20], video retrieval [21], and image-text pair pretraining [22]. We intend to improve the performance of the CTC-based ASR models by applying cross-modal learning. To this end, we first determine a *task-specific encoder* specialized for speech and text encoding and then define a *similarity measure* between two sequences of different lengths. For the last, we design an appropriate *objective function* to update the CTC model parameters. In the following sections, we introduce the original ideas of the proposed similarity measure and objective function.

2.2. BERT and BERTScore

BERT is an LM with a transformer encoder structure trained using a masked word prediction method [23, 24]. Due to the characteristics of the transformer-based model structure and masked word prediction method, BERT exhibits excellent performance in embedding long-range semantic information. [25] proposed a method of using the BERTScore to evaluate the quality of candidate sentences $\hat{y}$ generated by the conditional language generation model by making full use of the characteristics of BERT. The BERTScore method evaluates the quality of $\hat{y}$ based on the similarity between the hidden space representations obtained by forwarding a tokenized reference sentence $y = [y_1, ..., y_u]$ and a tokenized candidate sentence $\hat{y} = [\hat{y}_1, ..., \hat{y}_{\hat{u}}]$ to the BERT. While the existing N-gram matching approach and edit-distance-based metrics only estimate surface-form similarity, the BERTScore method has the advantage of estimating semantic similarity, which is not revealed externally.

Definition 2.1 (BERTScore). For the hidden-space embedding of BERT $z = f_{\text{BERT}}(y)$ and $\hat{z} = f_{\text{BERT}}(\hat{y})$, three-types of BERTScores (recall, precision and F1) are defined as follows:

$$R_{\text{BERT}} = \frac{1}{u} \sum_{z_i \in z} \max_{\hat{z}_j \in \hat{z}} \Phi_{ij}, \tag{5}$$

$$P_{\text{BERT}} = \frac{1}{\hat{u}} \sum_{\hat{z}_j \in \hat{z}} \max_{z_i \in z} \Phi_{ij}, \tag{6}$$

$$F_{\text{BERT}} = 2 \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}, \tag{7}$$

where $\Phi_{ij} = \frac{z_i^\top \hat{z}_j}{||z_i|| \, ||\hat{z}_j||}$, u denotes the length of y and $\hat{u}$ denotes the length of $\hat{y}$.

2.3. Sequence-Level Objective Function

The direct modeling of $P(Y|X)$ is intractable because there are infinite token sequences Y that can be generated. To overcome this problem, a method of modeling by factorizing it was used, as shown in Eq. (3). ASR models are trained to maximize the likelihood estimated at the frame or token levels. Although ASR models are trained at the frame level, they are evaluated at the sequence level. To reduce this discrepancy, several studies have investigated methods for training ASR models using sequence-level objective functions [26, 27, 11]. In this section, two sequence-level training methods for end-to-end ASR models are highlighted.

Definition 2.2 (MWER). Let $(x, y) \sim P(X, Y)$ be a paired speech and sentence sample. We denote M hypotheses corresponding to x by $\dot{\mathbf{y}} = [\dot{y}_1, \ldots, \dot{y}_M]$. We denote the edit distance from reference y to hypothesis $\dot{y}_m$ by d_m. We can

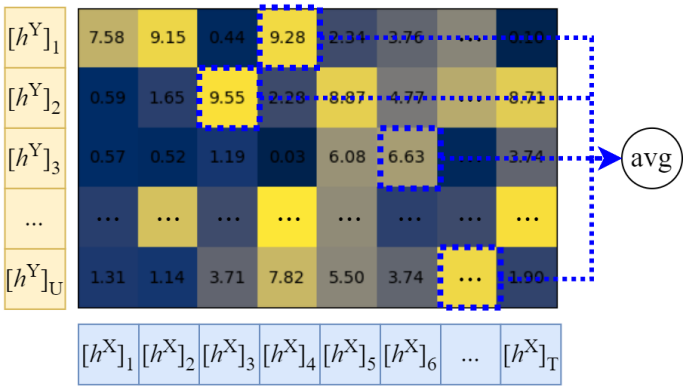

Fig. 1. Schematic diagram of the CTC-BERTScore ($P_{\text{C.B}}$) measurement process. h^X and h^Y denote the speech and text representation, respectively.

define the minimum word error rate (MWER) loss as follows:

$$\mathcal{L}_{\text{MWER}} = \sum_{m=1}^{M} p_m \cdot (d_m - \bar{d}), \tag{8}$$

$$p_m = \frac{e^{-s_m}}{\sum_{i=1}^{M} e^{-s_i}}, \ \bar{d} = \frac{1}{M}\sum_{m=1}^{M} d_m, \tag{9}$$

where $s_m = f_{\text{PLL}}(x, \dot{y}_m; \theta_{\text{ASR}})$ is the sequence-level score estimated by the ASR models (e.g, pseudo-log-likelihood; lower is better). The MWER is an objective function designed to minimize the expectation of the edit distance for a set of hypotheses. However, the MWER loss induces a high probability of mass being assigned only to the correct answer sequence, which can cause an overestimation problem. [11] proposed a new sequence-level objective function, matching word error distribution (MWED), which alleviates this problem.

Definition 2.3 (MWED). Using the same notation as in Definition 2.2, the MWED loss is defined as below:

$$\mathcal{L}_{\text{MWED}} = -\sum_{m=1}^{M} p_m^d \log p_m^s, \tag{10}$$

$$p_m^d = \frac{e^{d_m}}{\sum_{j=1}^{M} e^{d_j}}, \tag{11}$$

$$p_m^s = \frac{e^{s_i/\tau}}{\sum_{j=1}^{M} e^{s_j/\tau}}, \tag{12}$$

where $\tau > 0$ (temperature) is a tuning factor. As can be seen from the above formula, the MWED loss trains the model to minimize the cross-entropy between the target probability distribution that converts the edit distance for each hypothesis and the sequence-level score distribution estimated by the neural network. Unlike the MWER, MWED evenly distributes the probability mass estimated by the neural network in proportion to the accuracy of the hypotheses.

Reference (y)	Hypotheses ($\dot{y}_m$)	d_m	ψ_m	p_m^{ψ}
I love a dog	I love a dog	0	1	0.6103
	I love a a a a a dog	4	0.1353	0.0826
	I a dog	1	0.3679	0.2245
	I love dog a	2	0.1353	0.0826

Fig. 2. Example of sentence-level similarity distribution generation process based on edit-distance using text data augmentation.

3. PROPOSED METHODS

3.1. On the Conditional Independence Assumption of the CTC Models

Unlike the CTC model, which calculates $P(Y|X)$ based on the assumption of conditional independence, some ASR models have structural designs that can directly model the correlations between output tokens. The DNN-HMM-based ASR model models the transition probability between states based on the Markov assumption, whereas the AED and RNNT models mitigate the loss of contextual information through an auto-regressive structure. Recently, however, some studies have argued against the need for such structural designs to model the correlation between output tokens. [28] showed that the learned transition probability of the HMM model does not significantly affect the performance of the AM. [29] revealed that removing the recurrent unit of the prediction network in the RNNT model or providing only limited contextual information does not significantly affect model performance. The most noteworthy hypothesis explaining the cause of these phenomena is that transformer or RNN-based neural network models learn the correlation between tokens well on their own without an external structure for modeling the transition probabilities [28]. Encouraged by these recent findings, we assumed that a transformer-based CTC model trained with an appropriate objective function can effectively learn the correlation between output tokens by itself. Based on these assumptions, we propose a new training method based on cross-modal learning that induces transformer models to better embed contextual information.

3.2. CTC-BERTScore

As summarized in Section 2.1, to train a model using cross-modal learning strategies, we must find an appropriate similarity function $S(\cdot, \cdot)$ that operates in shared latent space H. We measured the similarity between speech samples and sentences using CTC-BERTScore, which is a modified version of the BERTScore [25].

Definition 3.1 (CTC-BERTScore). For a paired speech and text sample $(x, y) \sim P(X, Y)$, let $h^X = g_X(f_{\text{CTC}}(x))$ and $h^Y = g_Y(f_{\text{BERT}}(y; l_{\text{BERT}}))$. Where $f_{\text{CTC}} : X \mapsto$

$\mathbb{R}^{T\times d_{\mathrm{CTC}}}$ denotes the CTC model, $f_{\mathrm{BERT}}: Y \mapsto \mathbb{R}^{U\times d_{\mathrm{BERT}}}$ denotes the BERT, $g_X : \mathbb{R}^{T\times d_{\mathrm{CTC}}} \mapsto \mathbb{R}^{T\times d_H}$ and $g_Y : \mathbb{R}^{U\times d_{\mathrm{BERT}}} \mapsto \mathbb{R}^{U\times d_H}$ denote trainable mapping functions, T indicates the length of speech and U indicates the length of the token sequence. The CTC-BERTScore (recall and precision) is defined as follows:

$$R_{\mathrm{C.B}}(x,y) = \frac{1}{T}\sum_{[h^X]_i\in h^X}\max_{[h^Y]_j\in h^Y}\Phi_{ij}, \tag{13}$$

$$P_{\mathrm{C.B}}(x,y) = \frac{1}{U}\sum_{[h^Y]_j\in h^Y}\max_{[h^X]_i\in h^X}\Phi_{ij}, \tag{14}$$

where $\Phi_{ij} = \frac{[h^X]_i^{\mathsf{T}}[h^Y]_j}{||[h^X]_i||\,||[h^Y]_j||}$ and $[h^X]_i$ denotes a representation vector of the ith time step in h^X. $f_{\mathrm{CTC}}(\cdot)$ receives speech as an input and generates a d_{CTC}-dimensional penultimate layer output, and $f_{\mathrm{BERT}}(\cdot; l_{\mathrm{BERT}})$ is a function that receives a token sequence as an input and generates the representation vector of the l_{BERT}th layer of BERT. The mapping functions g_X and g_Y serve to match the dimensions of the representation vectors created by CTC and BERT and are composed of one layer of trainable fully connected layers. The process for calculating CTC-BERTScore is shown in Fig. 1.

3.3. Auxiliary Sequence-Level Training Loss for CTC

In this section, we propose an appropriate objective function that enables the neural network to learn the correlation between positive and negative pairs based on a CTC-BERTScore. The loss function for training an ASR model can be classified into two types. The first involves training a model to maximize the likelihood of a linear chain factorized in units of tokens or frames, and the second involves training a model using scores calculated at the sentence level [26, 28, 27, 11]. CTC-BERTScore has the characteristic of estimating the similarity between speech and text at the sequence level; therefore, it is more appropriate to use a sequence-level objective function for model training. We propose an objective function that modifies the MWED loss defined in Section 2.3 that makes the positive pair have a higher CTC-BERTScore than the negative pair.

Estimating the likelihood at the sequence level is generally intractable because there are an infinite number of token sequences. Several studies have solved this problem by approximating the denominator using a finite set of M hypotheses. We generated a set of hypotheses in two ways. The first method is to acquire M-best hypotheses corresponding to speech x by decoding using the ASR model. The second method obtains M augmented texts by applying the stochastic text data augmentation method to reference sentence y. In Section 4.3, we discuss the detailed setup of our stochastic text-data augmentation method. The edit-distance-based similarity between the ith sentence $\dot{y}_i$ of the hypothesis set $\dot{\mathbf{y}}$ and the reference sentence y can be defined as follows:

$$\psi_i = \exp(-\frac{d_i}{\tau\max(|y|,|\dot{y}_i|)}), \tag{15}$$

where τ denotes temperature and we used $\frac{1}{M}$ as the temperature value. We expect the estimated CTC-BERTScore between speech x and hypothesis set $\dot{\mathbf{y}}$ to match the distribution of ψ. We propose the following cross-modal matching word error distribution (CMWED) loss.

Definition 3.2 (CMWED). Given the edit-distance-based sequence-level similarity ψ_m and CTC-BERTScore ($P_{\mathrm{C.B}}(x,\dot{y}_m)$ or $R_{\mathrm{C.B}}(x,\dot{y}_m)$) for the mth hypothesis $\dot{y}_m$ in the hypothesis set, the CMWED loss is defined as follows:

$$\mathcal{L}_{\mathrm{CMWED}} = \sum_{m=1}^{M} -\,p_m^{\psi}\log p_m^{P_{\mathrm{C.B}}}, \tag{16}$$

$$p_m^{\psi} = \frac{\psi_m}{\sum_{i=1}^{M}\psi_i}, \tag{17}$$

$$p_m^{P_{\mathrm{C.B}}} = \frac{P_{\mathrm{C.B}}(x,\dot{y}_m)}{\sum_{i=1}^{M}P_{\mathrm{C.B}}(x,\dot{y}_i)}, \tag{18}$$

The CMWED loss matches the similarity between the hidden space representation vectors of speech and hypothesis texts to an edit-distance-based similarity distribution. We only updated the parameters of the CTC and mapping function during the training process and did not update the parameters of the pre-trained BERT. The model was trained using the gradient descent method by summing the CMWED and CTC losses ($\mathcal{L}_{\mathrm{CTC}}$). The final loss function is expressed as follows:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{CTC}} + \alpha\mathcal{L}_{\mathrm{CMWED}}, \tag{19}$$

where $\alpha \geq 0$ is a hyperparameter. In our experiments, the CTC model produced two types of outputs. The first output estimates the probability distribution of letter existence per frame and is trained with the CTC loss targeting the reference token sequence. The second output is generated by forwarding the penultimate layer output of the CTC model to $f_X(\cdot)$ and is trained with the proposed CMWED loss ($\mathcal{L}_{\mathrm{CMWED}}$). Therefore, the CTC model is trained as a multi-task learning scenario using $\mathcal{L}_{\mathrm{total}}$. The advantages of the proposed method are summarized as follows:

1. For cross-modal knowledge distillation from the LM to the ASR model, the process of finding a one-to-one correspondence between speech and transcriptions of different lengths must be performed [16, 17, 30, 31]. The proposed method uses the CTC-BERTScore to estimate the similarity between sequences with different lengths; therefore, finding an alignment is unnecessary.

2. The general ASR model performs transcription in a left-to-right manner. Because of this characteristic, it is difficult and inefficient to use an MLM such as BERT

Table 1. WER results for CTC models trained with the LibriSpeech100h/960h dataset and CMWED loss. l_{BERT} denotes the number of transformer layers used to create the BERT representation vector, and $S(\cdot,\cdot)$ denotes the similarity measure and $\dot{y}$ generation refers to the method used to create the hypothesis set.

l_{BERT}	α	$S(\cdot,\cdot)$	$\dot{y}$ generation	Trainset (h)	WER (%, w/o LM)				WER (%, 4-gram LM)			
					test		dev		test		dev	
					clean	other	clean	other	clean	other	clean	other
-	0	-	-	100	6.13	12.95	6.06	13.43	3.44	8.24	2.79	8.07
12	$1/U$	$P_{\mathrm{C.B}}$	Aug	100	5.68	13.02	5.61	13.38	3.32	8.18	2.76	8.02
12	$1/T$	$R_{\mathrm{C.B}}$	Aug	100	5.67	**12.76**	5.66	13.47	3.31	**8.05**	2.71	8.08
12	$1/T$	$R_{\mathrm{C.B}}$	M-best	100	**5.52**	13.00	**5.55**	13.51	**3.25**	8.26	**2.66**	**7.99**
9	$1/T$	$R_{\mathrm{C.B}}$	Aug	100	5.82	12.85	5.72	13.47	3.32	8.17	2.73	8.01
7	$1/T$	$R_{\mathrm{C.B}}$	Aug	100	5.65	12.98	5.66	13.50	3.37	8.32	2.70	8.01
3	$1/T$	$R_{\mathrm{C.B}}$	Aug	100	5.71	12.83	5.64	**13.34**	3.31	8.17	2.68	8.07
1	$1/T$	$R_{\mathrm{C.B}}$	Aug	100	5.79	12.90	5.65	**13.34**	3.38	8.23	2.81	8.15
12	$0.1/T$	$R_{\mathrm{C.B}}$	Aug	100	5.63	13.04	5.72	13.49	3.29	8.21	2.72	8.15
12	$10/T$	$R_{\mathrm{C.B}}$	Aug	100	5.71	13.22	5.71	13.74	3.34	8.35	2.76	8.33
-	0	-	-	960	3.34	8.61	3.14	8.82	2.65	6.15	1.91	5.81
12	$1/T$	$R_{\mathrm{C.B}}$	M-best	960	3.20	8.45	3.13	8.59	2.56	6.02	1.88	5.66

in the decoding process of an ASR model [8, 9, 10, 11]. We propose a method for utilizing BERT in the training process of the ASR model. The proposed method has a complementary relationship with the prefix beam search decoding method and leads to additional performance enhancement when applied together.

3. Although the CTC model does not structurally consider the relationship between tokens, the BERT model exhibits excellent performance in embedding long-range semantic information. The proposed method induced the CTC model to learn the correlation between token sequences intrinsically by training the CTC model to maximize the CTC-BERTScore.

4. EXPERIMENTS

4.1. Datasets

We trained and evaluated the ASR model using the LibriSpeech dataset [32]. The LibriSpeech training set consists of 960 hours of speech and comprises three parts: Libri-clean-100, Libri-clean-360, and Libri-other-500. The LibriSpeech evaluation set consisted of four parts: dev-clean, dev-other, test-clean, and test-other. Dev-other, test-other, and Libri-other-500 were composed of relatively challenging utterances and exhibited low performance in the experimental results. The LibriCorpus dataset (40 million sentences) was used to train the LMs.

4.2. Models

All the experiments were conducted using the FairSeq framework [33]. For the CTC model, we used the wav2vec 2.0 base model, which was pre-trained with self-supervised learning, and released [34]. The wav2vec 2.0 model was pre-trained with speech data in the 960-hour LibriSpeech trainset, and in the fine-tuning step, supervised learning was performed targeting the reference token sequence in letter units. We utilized two types of LMs in our experiments. The first LM was a 4-gram-based statistical LM, and we performed decoding using a beam search (1,500 beam widths) [7]. The second LM was an MLM-based model, and we used the RoBERTa model from the FairSeq framework [35, 33]. The RoBERTA model consists of 12 transformer encoder structures and was trained and inferred using the same letter unit tokenization method as the CTC (we trained it from scratch). Regarding the hyperparameters related to decoding or training that have not been mentioned, we conducted experiments with the same settings as [34, 35], and more detailed information can be found in these studies.

4.3. Hypotheses Generation Methods

The process of generating hypotheses to compose a denominator must precede sequence-level training. We used two methods for generating hypotheses. The first method generates M best hypotheses by decoding the ASR model. We generated $M = 20$ hypotheses with Viterbi decoding using the wav2vec2.0 base model pre-trained with Libri960h and fine-tuned with Libri100h. During the training process, we randomly extracted and used four of the 20 hypotheses at every iteration. The second method is text data augmentation, in which we generated M hypotheses by applying a stochastic perturbation to the reference sentence. We used three types of text-data augmentation methods. The first is a random swap. We randomly determined a consecutive section to be less than

1/2 of the length of the reference sentence and used a method of randomly shuffling the order of the token sequence within it. The second method is the stochastic deletion method, in which consecutive sections are randomly selected to be less than 1/2 of the length of the reference sentence, and a hypothesis sentence is generated by removing the selected section. The last method is the stochastic insertion method, in which we randomly select one token from the reference token sequence and repeat the token for a random length up to the maximum reference token sequence length. Examples of these data augmentation methods are shown in Fig. 2.

5. RESULTS

We applied the proposed method to a pre-trained wav2vec2.0-based CTC model showing state-of-the-art performance and evaluated the generalization performance based on the word error rate (WER). Table 1 shows the performance of the model trained using only CTC loss and the models trained with auxiliary sequence-level loss. As a result of the experiments, the proposed method showed a relative word error rate reduction (RERR) of 5-10$\%$ for the clean set (4th line in Table 1). In experiments using Libri100, the proposed method showed performance improvement in the clean set, but no performance gain was obtained in the other set. We speculate that these results were because all the Libri100h datasets were made of clean sets.

Next, we measured the change in performance based on the difference in the similarity measure used to calculate the sequence-level auxiliary loss ($P_{\mathrm{C.B}}, R_{\mathrm{C.B}}$). From the experimental results of the second and third lines of Table 1, it can be confirmed that $R_{\mathrm{C.B}}$ exhibits slightly better performance. We observed a change in model performance according to the depth of the BERT used to extract the hidden space representation. We obtained the highest performance gain when using the third-layer output; however, we could not confirm a meaningful trend according to the depth (5-8th lines in Table 1). We also measured the performance of the model according to the hypothesis generation methods and obtained a slightly better performance when the M-best decoding result was used compared to when the stochastic text data augmentation result was used (3rd and 4th lines in Table 1). Finally, we measured the model performance by varying the weight of the sequence-level training loss α from $0.1/T$ to $10/T$. The experiment with alpha set to $1/T$ achieved the highest average performance.

6. RELATED WORKS

Various methods have been investigated to improve the performance of ASR systems using external LMs. These methods can be classified into two types: methods using an external LM in the decoding process and methods using an external LM in the ASR model training process. [7] proposes a prefix beam search decoding method that utilizes an external LM in the beam search decoding process of the CTC model. First, this method successfully compensates for the disadvantages of the CTC model, which cannot consider contextual information due to the conditional independence condition; second, it allows the prior knowledge of the language learned through the large corpus to be used in the decoding process of the ASR model. Another method leveraging the LM in the ASR system decoding process is the rescoring method [8, 9, 10, 11]. The rescoring-based decoding method consists of two parts: a first-pass decoding process to find hypothesis sets, and a second-pass rescoring process to re-score hypothesis sets with an external scoring module (generally a large LM). Through the second-pass rescoring process, massive LMs can be used efficiently, and MLMs that are difficult to utilize in the first-pass decoding process can be used [8, 9, 10, 11]. The proposed method is different in that it allows the CTC model to model contextual information well without the help of an external LM in the decoding process and operates orthogonally with the prefix beam search method to achieve additional performance gain.

The cross-modal knowledge transfer method uses an external LM during the training process of the CTC model. The cross-modal knowledge transfer method assumes a source modality with a rich dataset, and a target modality with a relatively small dataset. Based on this assumption, the goal of the cross-modal knowledge transfer task is to transfer the performance of the teacher model trained on the source modality dataset to the student model trained on the target modality. In [16, 17], an LM trained with a large corpus is used as the teacher model, the CTC model is used as the student model, and knowledge distillation is performed by finding an alignment between speech and text of different lengths. The proposed method is different in that it uses sequence-level objectives without any frame or token-level knowledge distillation. Therefore, the proposed method does not require an alignment-finding process and is free from the performance degradation that can occur when using incorrect alignment.

7. CONCLUSION

The human brain is good at combining the information received by multiple sensory organs. These abilities help humans make more efficient and accurate judgments. Similar to the human brain, further research on cross-modal learning needs to be conducted to build a machine-learning algorithm that works effectively. In this study, we proposed a method for improving the performance of a CTC model using a cross-modal learning paradigm. We believe that there will be more efficient and effective *modality-specific encoders*, *similarity functions*, and *objective functions* for cross-modal learning of ASR models. In the future, we plan to conduct additional research to identify these factors and apply them effectively to ASR models.

8. REFERENCES

[1] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in *Proc. of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 6645–6649.

[2] 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 2016 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 4960–4964.

[3] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based Models for Speech Recognition," in *Proc. of Advances in Neural Information Processing Systems*, 2015, vol. 28.

[4] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proc. of 2006 International Conference on Machine Learning* (*ICML*), pp. 369–376.

[5] Alex Graves, "Sequence Transduction with Recurrent Neural Networks," *arXiv preprint arXiv:1211.3711*, 2012.

[6] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi, "Calibrating Sequence Likelihood Improves Conditional Language Generation," in *Proc. of International Conference on Learning Representations* (*ICLR*), 2023.

[7] Andrew L. Maas, Awni Y. Hannun, Dan Jurafsky, and A. Ng, "First-Pass Large Vocabulary Continuous Speech Recognition using Bi-Directional Recurrent DNNs," *ArXiv*, vol. abs/1408.2873, 2014.

[8] Joonbo Shin, Yoonhyung Lee, and Kyomin Jung, "Effective Sentence Scoring Method Using Bert for Speech Recognition," in *Proc. of Asian Conference on Machine Learning*. PMLR, 2019, pp. 1081–1093.

[9] Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff, "Masked Language Model Scoring," in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2699–2712.

[10] Shih-Hsuan Chiu and Berlin Chen, "Innovative BERT-based Reranking Language Models for Speech Recognition," in *Proc. of the 2021 IEEE Spoken Language Technology Workshop* (*SLT*), pp. 266–271.

[11] Liyan Xu, Yile Gu, Jari Kolehmainen, Haidar Khan, Ankur Gandhe, Ariya Rastrow, Andreas Stolcke, and Ivan Bulyko, "Rescorebert: Discriminative Speech Recognition Rescoring with BERT," in *Proc. of 2022 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 6117–6121.

[12] Yosuke Higuchi, Brian Yan, Siddhant Arora, Tetsuji Ogawa, Tetsunori Kobayashi, and Shinji Watanabe, "BERT Meets CTC: New Formulation of End-to-End Speech Recognition with Pre-trained Masked Language Model," *arXiv preprint arXiv:2210.16663*, 2022.

[13] Zhong Meng, Yashesh Gaur, Naoyuki Kanda, Jinyu Li, Xie Chen, Yu Wu, and Yifan Gong, "Internal Language Model Adaptation with Text-Only Data for End-to-End Speech Recognition," *arXiv preprint arXiv:2110.05354*, 2021.

[14] Zhong Meng, Naoyuki Kanda, Yashesh Gaur, Sarangarajan Parthasarathy, Eric Sun, Liang Lu, Xie Chen, Jinyu Li, and Yifan Gong, "Internal Language Model Training for Domain-Adaptive End-to-End Speech Recognition," in *Proc. of 2021 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 7338–7342.

[15] Xie Chen, Zhong Meng, Sarangarajan Parthasarathy, and Jinyu Li, "Factorized Neural Transducer for Efficient Language Model Adaptation," in *Proc. of 2022 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 8132–8136.

[16] Hayato Futami, Hirofumi Inaguma, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara, "Distilling the Knowledge of BERT for CTC-based ASR," *arXiv preprint arXiv:2209.02030*, 2022.

[17] Keqi Deng, Songjun Cao, Yike Zhang, Long Ma, Gaofeng Cheng, Ji Xu, and Pengyuan Zhang, "Improving CTC-based Speech Recognition Via Knowledge Transferring from Pre-trained Language models," in *Proc. of 2022 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 8517–8521.

[18] Alex Liu, SouYoung Jin, Cheng-I Lai, Andrew Rouditchenko, Aude Oliva, and James Glass, "Cross-Modal Discrete Representation Learning," in *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 3013–3035.

[19] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li, "CLIP4Clip: An Empirical Study of CLIP for End-to-End Video Clip Retrieval and Captioning," *Neurocomputing*, vol. 508, pp. 293–304, 2022.

[20] Mathew Monfort, SouYoung Jin, Alexander Liu, David Harwath, Rogerio Feris, James Glass, and Aude Oliva, "Spoken Moments: Learning Joint Audio-Visual Representations from Video Descriptions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14871–14881.

[21] Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, et al., "AVLnet: Learning Audio-Visual Language Representations from Instructional Videos," in *Proc. of the INTERSPEECH 2021*.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., "Learning Transferable Visual Models from Natural Language Supervision," in *Proc. of 2021 International Conference on Machine Learning* (*ICML*), pp. 8748–8763.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is All You Need," in *Proc. of Advances in Neural Information Processing Systems*, 2017, vol. 30.

[24] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. of NAACL-HLT*, 2019, pp. 4171–4186.

[25] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi, "BERTScore: Evaluating Text Generation with BERT," in *Proc. of International Conference on Learning Representations* (*ICLR*), 2020.

[26] Karel Veselỳ, Arnab Ghoshal, Lukás Burget, and Daniel Povey, "Sequence-Discriminative Training of Deep Neural Networks.," in *Proc. of the INTERSPEECH 2013*, pp. 2345–2349.

[27] Rohit Prabhavalkar, Tara N Sainath, Yonghui Wu, Patrick Nguyen, Zhifeng Chen, Chung-Cheng Chiu, and Anjuli Kannan, "Minimum Word Error Rate Training for Attention-based Sequence-to-Sequence Models," in *Proc. of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 4839–4843.

[28] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely Sequence-Trained Neural Networks for ASR based on Lattice-Free MMI.," in *Proc. of the INTERSPEECH 2016*, pp. 2751–2755.

[29] Mohammadreza Ghodsi, Xiaofeng Liu, James Apfel, Rodrigo Cabrera, and Eugene Weinstein, "RNN-Transducer with Stateless Prediction Network," in *Proc. of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 7049–7053.

[30] Mun-Hak Lee and Joon-Hyuk Chang, "Knowledge Distillation from Language Model to Acoustic Model: A Hierarchical Multi-Task Learning Approach," in *Proc. of 2022 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 8392–8396.

[31] Kwanghee Choi and Hyung-Min Park, "Distilling a Pretrained Language Model to a Multilingual ASR Model," *arXiv e-prints*, pp. arXiv–2206, 2022.

[32] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an ASR Corpus Based on Public Domain Audio Books," in *Proc. of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 5206–5210.

[33] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, "Fairseq: A Fast, Extensible Toolkit for Sequence Modeling," in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics* (*Demonstrations*), pp. 48–53.

[34] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *Proc. of Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 12449–12460.

[35] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv e-prints*, p. arXiv:1907.11692, July 2019.