HIERARCHICAL CONDITIONAL END-TO-END ASR WITH CTC AND MULTI-GRANULAR SUBWORD UNITS

Yosuke Higuchi, Keita Karube, Tetsuji Ogawa, Tetsunori Kobayashi

Department of Communications and Computer Engineering, Waseda University, Tokyo, Japan

ABSTRACT

In end-to-end automatic speech recognition (ASR), a model is expected to implicitly learn representations suitable for recognizing a word-level sequence. However, the huge abstraction gap between input acoustic signals and output linguistic tokens makes it challenging for a model to learn the representations. In this work, to promote the word-level representation learning in end-to-end ASR, we propose a *hierarchical conditional* model that is based on connectionist temporal classification (CTC). Our model is trained by auxiliary CTC losses applied to intermediate layers, where the vocabulary size of each target subword sequence is gradually increased as the layer becomes close to the word-level output. Here, we make each level of sequence prediction explicitly conditioned on the previous sequences predicted at lower levels. With the proposed approach, we expect the proposed model to learn the word-level representations effectively by exploiting a hierarchy of linguistic structures. Experimental results on LibriSpeech-{100h, 960h} and TEDLIUM2 demonstrate that the proposed model improves over a standard CTC-based model and other competitive models from prior work. We further analyze the results to confirm the effectiveness of the intended representation learning with our model.

***Index Terms*—** hierarchical conditional model, connectionist temporal classification, acoustic-to-word, end-to-end ASR

1. INTRODUCTION

End-to-end automatic speech recognition (ASR) aims to model direct speech-to-text conversion [1–3], which substantially simplifies the training and inference processes without external knowledge (e.g., a pronunciation lexicon). With well-established sequence-to-sequence modeling techniques [4–7] and more sophisticated neural network architectures [8–10], end-to-end ASR models have shown promising performance on various benchmarks [11–13].

Contrary to carefully designed feature extraction in the traditional pipeline framework, end-to-end models are generally expected to implicitly learn representations suitable for solving a specific task. For example, the learned representations have been shown to represent shape features for image classification [14] and syntactic structures for language modeling [15]. However, in ASR, it can be more challenging for an end-to-end model to learn representations automatically. Having no access to segmentation or alignment information, end-to-end ASR models are required to predict word-level linguistic tokens from frame-level acoustic signals. This input-output gap in the level of abstraction makes it difficult to optimize end-to-end ASR, unless a large amount of data or a strong language model is accessible during training or inference [16, 17].

To promote word-level representation learning in end-to-end ASR, we believe that a model should be trained to gradually increase the abstraction level of linguistic information, as it has long been considered reasonable for recognizing speech (i.e., speech → phonemes → words → text) [18]. By exploiting lower levels of abstractions to conditionally compose the higher-level linguistic information, an end-to-end ASR model should be able to handle the sparsity problem of words [19] and extract effective representations.

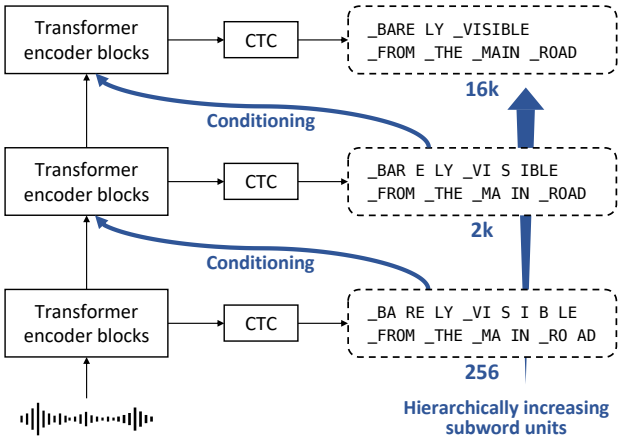

Fig. 1. Proposed hierarchical conditional model of end-to-end ASR.

To achieve such progressive representation learning for ASR, we propose *hierarchical conditional* modeling of end-to-end ASR (Figure 1). Our model consists of multiple connectionist temporal classification (CTC) [4] losses hierarchically applied to the intermediate and last layers, inspired by previous studies [20–26]. Each loss calculation targets sequences with a different granularity of linguistic information: sequences with lower abstraction levels are predicted from the intermediate layers, and a word-level sequence is predicted from the last layer. Specifically, we focus on subwords (n-gram characters) and increase the vocabulary size to word-level as the model layer becomes close to the output (e.g., 256 → 2k → 16k). In addition to this hierarchical structure, we design the model to predict each sequence at an abstraction level by explicitly conditioning on the previously predicted sequences at lower levels, which is crucial for maintaining subwords attributed to composing the higher-level sequence. The proposed model should capture a hierarchy of linguistic structures and yield representations suitable for modeling words.

The key contributions of this work are summarized as follows. 1) We show that the proposed approach enables a CTC-based system to learn accurate word-level ASR, mitigating the data-sparsity issue by gradually increasing the abstraction level of intermediate predictions. 2) Based on experiments conducted on LibriSpeech and TEDLIUM2, we demonstrate the effectiveness of our model independently of variations in the amount of data and speaking styles. All the implementations are made publicly available on our ESPnet fork (`https://github.com/YosukeHiguchi/espnet/tree/hierctc`). 3) We carefully compare our model with other CTC-based models and further analyze the results, which provides in-depth insights into the advantage of the proposed modeling.

2. HIERARCHICAL CONDITIONAL END-TO-END ASR

2.1. Baseline architecture of end-to-end ASR

End-to-end ASR is formulated as a sequence-mapping problem between a T-length input sequence $X = (\mathbf{x}_t \in \mathbb{R}^D | t = 1, \dots, T)$ and L-length output sequence $Y = (y_l \in \mathcal{V} | l = 1, \dots, L)$. Here, $\mathbf{x}_t$ is a D-dimensional acoustic feature at frame t, y_l is an output token at position l, and $\mathcal{V}$ is a vocabulary. As a baseline, we focus on a Transformer-based model [27] optimized by CTC [4] with intermediate loss calculation [25, 26].

Transformer encoder: For encoding an audio sequence X into latent representations, we construct the Transformer encoder [27] consisting of a stack of E self-attention layers. The i-th layer outputs a sequence of d_{model}-dimensional latent representations $X^{(i)} = (\mathbf{x}_t^{(i)} \in \mathbb{R}^{d_{\mathsf{model}}} | t = 1, ..., T)$ as

$$\tilde{X}^{(i)} = X^{(i-1)} + \mathrm{SelfAttention}(X^{(i-1)}), \tag{1}$$

$$X^{(i)} = \tilde{X}^{(i)} + \mathrm{FeedForward}(\tilde{X}^{(i)}), \tag{2}$$

where $i \in \{1, ..., E\}$, and $X^{(0)}$ is obtained by adding positional encodings to X. In Eqs. (1) and (2), layer normalization is applied to each input of the self-attention mechanism $\mathrm{SelfAttention}(\cdot)$ and feedforward network $\mathrm{FeedForward}(\cdot)$. We also train a model with the Conformer encoder [9], which introduces a convolution neural network (CNN) into the Transformer encoder, i.e., a convolution module is added between Eqs. (1) and (2).

Connectionist temporal classification: CTC [4] optimizes the model to predict a monotonic alignment between the encoded input $X^{(E)}$ and output Y. To align the sequences in frame-level, the output sequence Y is augmented with a unique blank token ϵ, which results in a latent token sequence $Z = (z_t \in \mathcal{V} \cup \{\epsilon\} | t = 1, \dots, T)$. On the basis of the conditional independence assumption per token-frame prediction, CTC models the conditional probability $P_{\mathsf{ctc}}(Y|X^{(E)})$ by marginalizing over latent token sequences as

$$P_{\mathsf{ctc}}(Y|X^{(E)}) \approx \sum_{Z \in \mathcal{B}^{-1}(Y)} \prod_{t=1}^{T} P(z_t|X^{(E)}), \tag{3}$$

where $\mathcal{B}^{-1}(Y)$ returns all possible latent sequences compatible with Y. The CTC loss is defined as the negative log-likelihood of Eq. (3):

$$\mathcal{L}_{\mathsf{ctc}}(Y|X^{(E)}) = -\log P_{\mathsf{ctc}}(Y|X^{(E)}). \tag{4}$$

Intermediate CTC: In addition to the standard CTC loss calculated from the model output, auxiliary CTC losses can be iteratively applied to intermediate layers [25, 26]. Such intermediate losses effectively regularize the model training and lead to improved ASR performance. We consider training the model with a total of K CTC losses applied to the output and intermediate layers:

$$\mathcal{L}_{\mathsf{sc\text{-}ctc}} = \frac{1}{K} \left\{ \mathcal{L}_{\mathsf{ctc}}(Y|X^{(E)}) + \sum_{k=1}^{K-1} \mathcal{L}_{\mathsf{ctc}}(Y|X^{(\lfloor \frac{kE}{K} \rfloor)}) \right\}, \tag{5}$$

where $1 < K \leq E$, and we equally distribute the weight across the losses [28]. In Eq. (5), we adopt the self-conditioning mechanism [29], which improves a CTC-based model by relaxing the conditional independence assumption. For the intermediate layer, from which a CTC loss is calculated, we modify Eq. (2) as

$$\tilde{\tilde{X}}^{(i)} = \tilde{X}^{(i)} + \mathrm{FeedForward}(\tilde{X}^{(i)}), \tag{6}$$

$$X^{(i)} = \tilde{\tilde{X}}^{(i)} + \mathrm{Linear}(A^{(i)}), \tag{7}$$

where $i \in \{\lfloor kE/K \rfloor\}_{k=1}^{K-1}$, and $A^{(i)} = \mathrm{softmax}(\tilde{\tilde{X}}^{(i)})$ is a sequence of the posterior distributions w.r.t latent tokens computed by CTC.

2.2. Subword segmentation

For tokenizing target sequences, subword segmentation is a widely used approach for alleviating the out-of-vocabulary problem [30], where words in a sentence are split into subword units (or n-gram characters). In the general algorithm for building a subword vocabulary, pairs of subword units are repeatedly merged on the basis of the frequency appearing in a text corpus. The iteration stops when the vocabulary reaches an arbitrary size.

We adopt subwords for tokenizing ASR transcriptions. As opposed to characters, subwords can provide the model with shorter output sequences, thus reduce the difficulty of modeling the dependency between outputs. This can be especially important for CTC-based modeling with the conditional independence assumption. However, it should be noted that increasing the subword vocabulary size makes a sequence close to word-level and potentially lead to the data-sparsity problem [19].

2.3. Proposed hierarchical conditional model

Figure 1 represents an overview of the proposed hierarchical conditional model of end-to-end ASR. It is similar to the intermediate CTC training, but the granularity of subword units is gradually increased to word-level as the sequence transduction proceeds in the self-attention layers. Let $Y^{(k)} = (y_l^{(k)} \in \mathcal{V}^{(k)} | l = 1, \dots, L^{(k)})$ be an $L^{(k)}$-length target subword sequence of the k-th CTC loss, which is generated by the corresponding subword segmenter with a vocabulary of $\mathcal{V}^{(k)}$. We hierarchically increase the vocabulary size, as the position of the CTC loss becomes close to the output layer (i.e., $|\mathcal{V}^{(<K)}| < |\mathcal{V}^{(K)}|$). Given the target sequences with different units, the objective of the proposed model is defined by modifying Eq. (5) as follows:

$$\mathcal{L}_{\mathsf{hc\text{-}ctc}} = \frac{1}{K} \left\{ \mathcal{L}_{\mathsf{ctc}}(Y^{(K)}|X^{(E)}) + \sum_{k=1}^{K-1} \mathcal{L}_{\mathsf{ctc}}(Y^{(k)}|X^{(\lfloor \frac{kE}{K} \rfloor)}) \right\}. \tag{8}$$

If the vocabulary size of each target sequence is the same, Eq. (8) is equal to Eq. (5). With the conditioning mechanism realized by Eq. (7), each CTC loss calculation in Eq. (8) is conditioned on the previously predicted sequences with lower levels of subword units:

$$\mathcal{L}_{\mathsf{ctc}}(Y^{(k)}|X^{(k)}) = -\log P_{\mathsf{ctc}}(Y^{(k)}|\hat{Y}^{(1)}, ..., \hat{Y}^{(k-1)}, X^{(k)}), \tag{9}$$

where $\hat{Y}^{(k)}$ denotes a sequence predicted by the k-th CTC, which is implicitly represented by the posterior distributions of latent tokens.

In the proposed hierarchical conditional model, we break down the word-level recognition into a process of progressively integrating subwords in a fine-to-coarse manner. By making the shallower layers predict frequent subwords with small units and the deeper layers predict sparse subwords with large units, we expect the model to use a hierarchy of linguistic structures and yield word-level representations effectively.

2.4. Applying CTC losses in parallel

To verify the effectiveness of the proposed model with the hierarchical structure, we also consider training a model with CTC losses applied in parallel to the final layer, which has been shown effective in several studies [23, 31–33]. The objective for the parallel CTC losses is defined by modifying Eq. (8) as

$$\mathcal{L}_{\mathsf{paractc}} = \frac{1}{K} \left\{ \mathcal{L}_{\mathsf{ctc}}(Y^{(K)}|X^{(E)}) + \sum_{k=1}^{K-1} \mathcal{L}_{\mathsf{ctc}}(Y^{(k)}|X^{(E)}) \right\}. \tag{10}$$

We apply a single linear layer to $X^{(E)}$ for adapting features to each CTC loss with a different granularity of subword units.

Table 1. Word error rate (WER) [%] on LibriSpeech-{100h, 960h} and TEDLIUM2. Output subword vocabulary size was set to 16k for LibriSpeech-100h and TEDLIUM2, and 32k for LibriSpeech-960h. We did not use language model or beam-search during decoding.

Model		LibriSpeech-100h				LibriSpeech-960h				TEDLIUM2	
		Dev WER		Test WER		Dev WER		Test WER		Dev WER	Test WER
		clean	other	clean	other	clean	other	clean	other		
Transformer	CTC	11.5	24.8	11.8	25.5	4.2	10.0	4.5	9.9	11.8	10.7
	SC-CTC	8.9	21.0	9.1	21.7	3.2	8.2	3.5	8.2	9.4	**8.6**
	HC-CTC	**8.2**	**19.9**	**8.4**	**20.6**	**3.1**	**8.0**	**3.4**	**8.0**	**9.1**	**8.6**
	ParaCTC	10.4	24.0	10.9	24.3	4.6	10.3	4.8	10.3	10.9	10.2
Conformer	SC-CTC	7.1	17.7	7.7	18.3	**2.8**	**6.7**	**3.0**	6.9	8.5	7.8
	HC-CTC	**6.9**	**17.1**	**7.1**	**17.8**	**2.8**	6.9	**3.0**	**6.8**	**8.0**	**7.6**

The parallel CTC training treats the predictions of multi-granular sequences equally, where finer subword predictions provide an inductive bias to promote coarse word-level modeling [32].

3. RELATIONSHIP TO PRIOR WORK

Several studies have explored introducing auxiliary CTC losses to intermediate model layers and demonstrated its effectiveness for improving various end-to-end ASR systems, based on attention-based sequence-to-sequence [34,35], recurrent neural network transducer [36], and CTC [25, 26, 37, 38]. For the CTC-based system, hierarchically applying low-level supervision (e.g., phonemes) to the intermediate CTC losses has shown to improve a primary CTC loss with higher-level recognition [20–24]. The proposed model can be considered an extension of these hierarchical CTC-based models. However, our work differs from prior work in the following perspectives. 1) Each CTC loss is explicitly conditioned on the sequences predicted previously at lower abstraction levels. We expect the model to maintain subwords that contribute to composing a word-level sequence and promote the CTC training with conditional dependencies [29]. 2) Given that, in recent studies [25, 26], the intermediate CTC losses are effective even without the hierarchical supervision, we carefully conduct a comparative experiment and further analyze the effectiveness of hierarchical modeling. 3) We only use subwords for target sequences, which does not require additional labeling effort and is easy to control the granularity of target sequences. 4) We evaluate models using the recent state-of-the-art architectures (i.e., Transformer [27] and Conformer [9]).

4. EXPERIMENTS

4.1. Experimental setup

Data: The experiments were carried out using the LibriSpeech (LS) [39] and TEDLIUM2 (TED2) [40] datasets. LS consists of utterances from read English audio books. We trained the models using the 100-hour subset (LS-100) or the 960-hour full set (LS-960). TED2 consists of utterances from English Ted Talks and contains 210 hours of training data. For each dataset, we used the standard development and test sets. As input speech features, we extracted 80 mel-scale filterbank coefficients with three-dimensional pitch features using Kaldi [41], which were augmented by speed perturbation and SpecAugment [42]. We used SentencePiece [43] to construct subword vocabularies for each dataset.

Evaluated models: **CTC** denotes a standard CTC-based model trained with $\mathcal{L}_{\text{ctc}}$ from Eq. (4) [1]. **SC-CTC** is a conventional model trained with the intermediate CTC losses [25, 26] and the self-conditioning mechanism [29] defined by $\mathcal{L}_{\text{sc-ctc}}$ in Eq. (5). **HC-CTC** is the proposed hierarchical conditional model trained with $\mathcal{L}_{\text{hc-ctc}}$ from Eq. (8). **ParaCTC** is a conventional model trained with the parallel CTC losses defined by $\mathcal{L}_{\text{paractc}}$ in Eq. (10) [23, 31, 32].

Training and decoding configurations: All experiments were conducted using ESPnet [44]. We used the Transformer [27] architecture to train the above models, which consisted of two CNN layers followed by a stack of 18 self-attention layers. The number of heads d_{h}, dimension of a self-attention layer d_{model}, and dimension of a feed-forward network d_{ff} were set to 4, 256, and 2048, respectively. We also trained the models using the Conformer architecture [9], which had a kernel size of 15 and the same configurations as the Transformer-based models, except d_{ff} was set to 1024. The models were trained up to 100 epochs. For models with multiple CTC losses (i.e., SC-CTC, HC-CTC, and ParaCTC), we set the total number of losses to 3 ($K = 3$). The output vocabulary sizes for LS-100, LS-960, and TED2 were set to 16384, 32768, and 16384, respectively. Each vocabulary size was determined on the basis of the maximum number we could set using SentencePiece, which is large enough to be considered as word-level. SC-CTC had intermediate losses with the same vocabulary size as the output's. For HC-CTC and ParaCTC, we set ($|\mathcal{V}^{(1)}|$, $|\mathcal{V}^{(2)}|$, $|\mathcal{V}^{(3)}|$) to (256, 2048, 16384) for LS-100 and TED2, and (512, 4096, 32768) for LS-960. After training, a final model was obtained by averaging model parameters over 10 to 20 checkpoints with the best validation performance. During decoding, we did not use any language model and carried out the best path decoding of CTC [4]. Our implementations are publicly available to ensure reproducibility (see Sec. 1).

4.2. Main results

Table 1 lists the results on LS-100, LS-960, and TED2 in terms of the word error rate (WER). Looking at the Transformer results, all the models trained with multiple CTC losses led to an improvement over the standard CTC-based model. Especially, SC-CTC and HC-CTC significantly reduced the WER on all of the tasks. On LS-100, HC-CTC showed a clear improvement over SC-CTC, indicating the effectiveness of hierarchically increasing subword units. In contrast, on LS-960 and TED2 with more data, the performance gap was reduced, and HC-CTC performed slightly better than SC-CTC. Therefore, it can be concluded that our model is particularly effective for smaller-scale data, where the word-level units are likely to become sparser. SC-CTC was capable of handling word-level units when there is a sufficient amount of data. However, the large vocabulary-sized softmax calculation (in Eq. (7)) led to a severe slow-down of the SC-CTC training and inference processes. HC-CTC, on the other hand, was able to perform faster training and inference, using finer units for the losses from intermediate layers. Due to the same reason regarding the softmax calculation, the model size of HC-CTC was much smaller than that of SC-CTC (e.g., 36.4M vs. 67.6M on LS-960). By comparing HC-CTC with ParaCTC, HC-CTC achieved much lower WERs on all tasks, demonstrating the effectiveness of applying CTC losses to intermediate layers as well as gradually in-

Table 2. WER [%] on LS-100 dev. sets for Transformer-based models trained with different combinations of subword vocabulary sizes.

Model	$\lvert\mathcal{V}^{(1)}\rvert$-$\lvert\mathcal{V}^{(2)}\rvert$-$\lvert\mathcal{V}^{(3)}\rvert$	dev-clean	dev-other
SC-CTC	256 - 256 - 256	8.4	22.8
SC-CTC	2k - 2k - 2k	8.5	22.0
SC-CTC	16k - 16k - 16k	8.9	21.0
HC-CTC	256 - 256 - 16k	**8.2**	20.2
HC-CTC	2k - 2k - 16k	8.4	20.2
HC-CTC	256 - 2k - 16k	**8.2**	**19.9**

creasing the subword units in a hierarchical manner.

Using Conformer further improved the performance of SC-CTC and HC-CTC, and HC-CTC again achieved more favorable performance than SC-CTC with faster training and inference. Our Conformer results are comparable with other strong CTC-based models of the same size [10, 45, 46], even without exhaustive tuning.

4.3. Analysis on subword vocabulary size

While using sparse word-level units can make training of an ASR model challenging [19], we observed that the standard CTC-based model, with the Transformer-based architecture, benefits from training with a large subword vocabulary size. By increasing the output vocabulary size from 256 to 16k, the WERs for dev. sets changed from 11.1/28.1% to 11.5/24.8% on LS-360, and 12.3% to 11.8% on TED2. Similarly, the performance on LS-960 changed from 4.6/12.1% to 4.4/10.5% by changing the vocabulary size from 2k to 32k. These decent improvements from increasing the subword vocabulary size can be attributed to compensating for the CTC's incapability of modeling output dependencies (cf. Eq.(3)).

Considering the above observation, we evaluated SC-CTC and HC-CTC with different combinations of vocabulary sizes, focusing on Transformer-based models trained on LS-100. From the results for SC-CTC in Table 2, the performance on the dev-other set improved by increasing the vocabulary size, benefiting from the CTC training with large subword units. HC-CTC performed better than the 16k result of SC-CTC, indicating HC-CTC was more effective at modeling word-level recognition besides the advantage of CTC training with a large vocabulary size. While the SC-CTC performance on the dev-clean set degraded by increasing the vocabulary size, HC-CTC succeeded in learning robust word-level representations and achieved the lowest WER with the 16k-vocabulary size. Comparing the HC-CTC results, hierarchically increasing the subword units resulted in better performance than using the same vocabulary size for intermediate losses, suggesting the importance of gradually increasing the abstraction level for learning word-level representations effectively.

4.4. Importance of conditioning

We studied the effectiveness of the conditioning mechanism, which is one of the important components of the proposed model (cf. Eq. (9)). The Transformer-based HC-CTC was trained on LS-100 without conditioning each CTC loss (i.e., Eqs. (1) and (2) were used for all the intermediate layers). Note that this model is similar to those from previous studies [20–24]. Without the conditioning mechanism, HC-CTC achieved WERs of 8.7/20.7% and 9.0/21.3% on dev. sets and test sets, respectively. While these results are better than those obtained from CTC, SC-CTC, and ParaCTC in Table 1, HC-CTC with the conditioning mechanism achieved much lower WERs. Overall, we can conclude that 1) hierarchical modeling based on multi-granular subword units as well as 2) the conditioning mechanism for explicitly maintaining lower levels of predictions are effective for learning word-level representations.

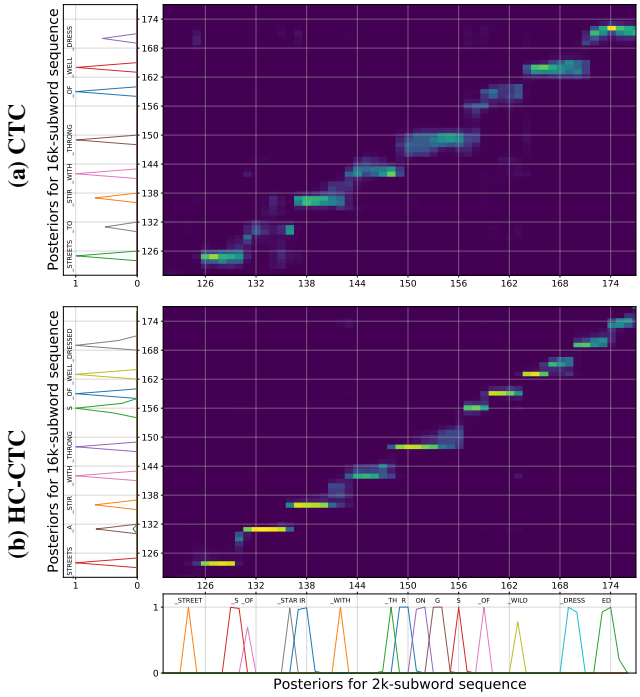

Fig. 2. Attention visualization of (a) CTC and (b) HC-CTC trained on LS-100 from Table 1. We manually chose partial utterance from dev-other set (116-288045-0000), transcription of which is *"STREETS ASTIR WITH THRONGS OF WELL DRESSED"*.

4.5. Attention visualization

Figure 2 visualizes attention weights between a source (x-axis) and target (y-axis) sequences, comparing Transformer-based (a) CTC and (b) HC-CTC trained on LS-100 from Table 1. We focused on weights that seemed to contribute to predicting a 16k-subword sequence in the final CTC (from the 18-th layer). For HC-CTC, we show the CTC posteriors (from the 12-th layer) for predicting a 2k-subword sequence in advance to see the relationship to the 16k prediction. Comparing the overall weights, HC-CTC learned more solid and confident weights than CTC. HC-CTC seemed to exploit the lower-level 2k predictions to detect important frames for predicting each token, effectively composing complex word-level tokens using the lower-level tokens. For example, HC-CTC successfully recognized the words "THRONGS" and "DRESSED" with proper conjunctions, while CTC failed to handle these infrequent words.

5. CONCLUSIONS

We proposed a hierarchical conditional model of CTC-based end-to-end ASR. We trained the model by gradually increasing the subword units for CTC losses applied to intermediate layers. Each CTC loss was conditioned on the sequences with lower abstraction to compose higher-level prediction. Experimental results and in-depth analysis demonstrated that our model effectively learned word-level representations for improving ASR performance. Future work includes introducing an additional decoder network [47] and using acoustic-based subword unit for lower-level predictions [48, 49].

6. ACKNOWLEDGEMENT

This work was supported in part by JST ACT-X (JPMJAX210J).

7. REFERENCES

[1] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014, pp. 1764–1772.

[2] Jan K Chorowski et al., "Attention-based models for speech recognition," in *Proc. NeurIPS*, 2015, pp. 577–585.

[3] William Chan et al., "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[4] Alex Graves et al., "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[5] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[6] Ilya Sutskever et al., "Sequence to sequence learning with neural networks," in *Proc. NeurIPS*, 2014, pp. 3104–3112.

[7] Dzmitry Bahdanau et al., "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2014.

[8] Linhao Dong et al., "Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*, 2018, pp. 5884–5888.

[9] Anmol Gulati et al., "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[10] Somshubra Majumdar et al., "Citrinet: Closing the gap between non-autoregressive and autoregressive end-to-end models for automatic speech recognition," *arXiv preprint arXiv:2104.01721*, 2021.

[11] Chung-Cheng Chiu et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018, pp. 4774–4778.

[12] Christoph Lüscher et al., "RWTH ASR systems for LibriSpeech: Hybrid vs attention," in *Proc. Interspeech*, 2019, pp. 231–235.

[13] Shigeki Karita et al., "A comparative study on Transformer vs RNN in speech applications," in *Proc. ASRU*, 2019, pp. 449–456.

[14] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014, pp. 818–833.

[15] Matthew E Peters et al., "Deep contextualized word representations," in *Proc. NAACL-HLT*, 2018, pp. 2227–2237.

[16] Yu Zhang et al., "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[17] Kazuki Irie et al., "Language modeling with deep Transformers," in *Proc. Interspeech*, 2019, pp. 3905–3909.

[18] Frederick Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[19] Hagen Soltau et al., "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[20] Santiago Fernández et al., "Sequence labelling in structured domains with hierarchical recurrent neural networks," in *Proc. IJCAI*, 2007, pp. 774–779.

[21] Kanishka Rao and Haşim Sak, "Multi-accent speech recognition with hierarchical grapheme based models," in *Proc. ICASSP*, 2017, pp. 4815–4819.

[22] Shubham Toshniwal et al., "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," *arXiv preprint arXiv:1704.01631*, 2017.

[23] Ramon Sanabria and Florian Metze, "Hierarchical multitask learning with CTC," in *Proc. SLT*, 2018, pp. 485–490.

[24] Kalpesh Krishna et al., "Hierarchical multitask learning for CTC-based speech recognition," *arXiv preprint arXiv:1807.06234*, 2018.

[25] Andros Tjandra et al., "Deja-vu: Double feature presentation and iterated loss in deep Transformer networks," in *Proc. ICASSP*, 2020, pp. 6899–6903.

[26] Jaesong Lee and Shinji Watanabe, "Intermediate loss regularization for CTC-based speech recognition," in *Proc. ICASSP*, 2021, pp. 6224–6228.

[27] Ashish Vaswani et al., "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.

[28] Jaesong Lee et al., "Layer pruning on demand with intermediate CTC," in *Proc. Interspeech*, 2021, pp. 3745–3749.

[29] Jumon Nozaki and Tatsuya Komatsu, "Relaxing the conditional independence assumption of CTC-based ASR by conditioning on intermediate predictions," in *Proc. Interspeech*, 2021, pp. 3735–3739.

[30] Rico Sennrich et al., "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016, pp. 1715–1725.

[31] Jinyu Li et al., "Acoustic-to-word model without OOV," in *Proc. ASRU*, 2017, pp. 111–117.

[32] Jan Kremer et al., "On the inductive bias of word-character-level multi-task learning for speech recognition," *arXiv preprint arXiv:1812.02308*, 2018.

[33] Abdelwahab Heba et al., "Char+CV-CTC: Combining graphemes and consonant/vowel units for CTC-based ASR using multitask learning," in *Proc. Interspeech*, 2019, pp. 1611–1615.

[34] Suyoun Kim et al., "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*, 2017, pp. 4835–4839.

[35] Takafumi Moriya et al., "Multi-task learning with augmentation strategy for acoustic-to-word attention-based encoder-decoder speech recognition.," in *Proc. Interspeech*, 2018, pp. 2399–2403.

[36] Jae-Jin Jeon and Eesung Kim, "Multitask learning and joint optimization for Transformer-RNN-Transducer speech recognition," in *Proc. ICASSP*, 2021, pp. 6793–6797.

[37] Geoffrey Zweig et al., "Advances in all-neural speech recognition," in *Proc. ICASSP*, 2017, pp. 4805–4809.

[38] Ethan A Chi et al., "Align-Refine: Non-autoregressive speech recognition via iterative realignment," in *Proc. NAACL-HLT*, 2021, pp. 1920–1927.

[39] Vassil Panayotov et al., "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[40] Anthony Rousseau et al., "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *Porc. LREC*, 2014, pp. 3935–3939.

[41] Daniel Povey et al., "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[42] Daniel S Park et al., "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[43] Taku Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proc. ACL*, 2018, pp. 66–75.

[44] Shinji Watanabe et al., "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[45] Edwin G. Ng et al., "Pushing the limits of non-autoregressive speech recognition," in *Proc. Interspeech*, 2021, pp. 3725–3729.

[46] Yosuke Higuchi et al., "A comparative study on non-autoregressive modelings for speech-to-text generation," in *Proc. ASRU*, 2021, pp. 47–54.

[47] Yosuke Higuchi et al., "Mask CTC: Non-autoregressive end-to-end ASR with CTC and mask predict," in *Proc. Interspeech*, 2020, pp. 3655–3659.

[48] Hainan Xu et al., "Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling," in *Proc. ICASSP*, 2019, pp. 7110–7114.

[49] Wei Zhou et al., "Acoustic data-driven subword modeling for end-to-end speech recognition," in *Proc. Interspeech*, 2021, pp. 2886–2890.