Interspeech 2022
18-22 September 2022, Incheon, Korea

# K-Wav2vec 2.0: Automatic Speech Recognition based on Joint Decoding of Graphemes and Syllables

*Jounghee Kim, Pilsung Kang**

School of Industrial Management Engineering, Korea University, Seoul, Korea
jounghee_kim@korea.ac.kr, pilsung_kang@korea.ac.kr

## Abstract

Wav2vec 2.0 is an end-to-end framework of self-supervised learning for speech representation that is successful in automatic speech recognition (ASR), but most of the work has been developed with a single language: English. Therefore, it is unclear whether the self-supervised framework is effective in recognizing other languages with different writing systems, such as Korean. In this paper, we present K-Wav2Vec 2.0, which is a modified version of Wav2vec 2.0 designed for Korean ASR by exploring and optimizing various factors of the original Wav2vec 2.0. In fine-tuning, we propose a multi-task hierarchical architecture to reflect the Korean writing structure. Moreover, a joint decoder is applied to alleviate the out-of-vocabulary problem. In pre-training, we attempted the cross-lingual transfer of the pre-trained model by further pre-training the English Wav2vec 2.0 on a Korean dataset, considering limited resources. Our experimental results demonstrate that the proposed method efficiently yields robust and better performance on both Korean ASR datasets.

**Index Terms**: speech recognition, multi-task learning, joint decoding, cross-lingual transfer

## 1. Introduction

In recent years, self-supervised methodology has shown success in various fields [1, 2]. The Wav2vec 2.0 model [2] is an end-to-end framework of self-supervised learning for automatic speech recognition (ASR), and it has recently been presented as an effective pre-training method to learn speech representations. When followed by fine-tuning with small amounts of labeled data, the Wav2vec 2.0 model has shown remarkable performance in English ASR tasks. However, despite the model's great success, it is still an open question whether this method can be effective with other languages, because most experiments have been conducted with English datasets such as Librispeech [3] and TIMIT [4]. In this paper, we introduce the way to adapt the Wav2vec 2.0 model to Korean ASR by considering various language-specific features, incorporating an effective fine-tuning architecture and efficient pre-training method.

In the Korean writing system, letters are written in syllabic blocks, and these syllabic blocks are composed of 51 Korean grapheme units. This unique writing system allows us to build a Korean ASR model that is based on either graphemes [5, 6] or syllable blocks [7, 8, 9]. According to previous research that investigated modeling units in Korean ASR tasks, syllable-based models outperform grapheme-based models in most cases [10]. Because grapheme-based models require more combinations of input to predict, they generally underperform. However, syllable-based models also have data sparseness problem for infrequently used syllables and the out-of-vocabulary (OOV) problem when the training data is insufficient. In other languages which suffer from the same issues, previous research[11, 12, 13, 14] has identified methods to overcome these problems using a multi-task learning approach (MTL). Multi-task learning is a method for learning shared representations from different (but related) tasks using different modeling units together. By learning the shared representations between high-level and low-level modeling units, multi-task models alleviate data sparseness issues and achieve better performance [13]. Moreover, some research [14] has introduced recovering methods to substitute OOV words produced from high-level decoding with segments generated in low-level outputs.

Inspired by prior works conducted with other languages, we propose a multi-task hierarchical fine-tuning architecture of Wav2vec 2.0 to reflect the unique relationship that exists in Korean writing between syllables and graphemes. By learning useful intermediate representations, the proposed model can generate multi-level units without sacrificing performance. In the inference step, we used a joint decoding strategy that considers grapheme-level and syllable-level units to find the best sequence from a set of candidates instead of using additional language models. This decoding approach leads to better performance and robust results by alleviating the problem of OOV words in syllable-level outputs in various datasets.

In practice, collecting appropriate unlabeled data for stable training is expensive task. To consider the practical circumstance wherein only limited resources are available, we also experimented with the cross-lingual transfer of the English Wav2vec 2.0 model to Korean ASR tasks. Recently, several works [15, 16] have shown that the cross-lingual transfer methods, pre-training with English and fine-tuning with the other low-resource languages, can be effective in improving the performance of downstream tasks. We adopted the cross-lingual transfer method in ASR to improve the model's performance with a limited Korean dataset by further pre-training the English Wav2vec 2.0 model. Our further pre-training approach efficiently learns Korean speech representations, taking advantage of learned representations from another language.

## 2. Backgrounds

In this section, we briefly review the base architecture of Wav2vec 2.0 and its training process for ASR.

### 2.1. Pre-training Wav2vec 2.0

The base architecture of Wav2vec 2.0 consists of three networks: a feature encoder, a contextual transformer, and a quan-

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2022R1A2C2005455). This work was also supported by Institute of Information & communications Technology Planning Evaluation grant funded by the Korea government (MSIT) (No. 2021-0-00034, Clustering technologies of fragmented data for time-based data analysis).



10.21437/Interspeech.2022-547

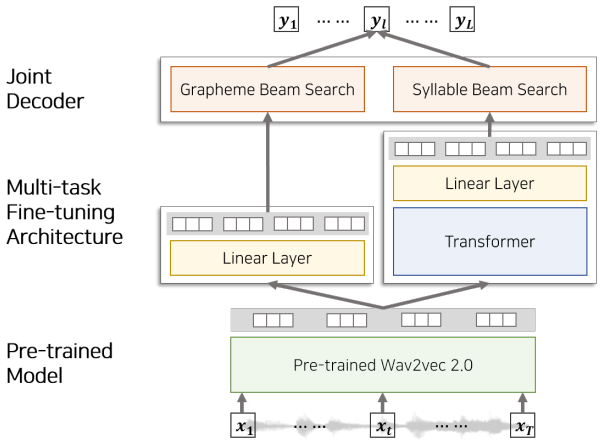


Figure 1: *Overview of our proposed ASR framework.*

tization module [2]. The feature encoder, composed of a multi-layer convolutional neural network, encodes raw audio $X$ and outputs the latent speech representations $Z$. The contextual transformer, a stack of transformer encoders, learns context representation $C$ by taking latent speech representations as input. The quantization module is used to map latent representations into discretized space $Q$, choosing discrete codebook entries in a fully differentiable way.

In pre-training, a certain portion of latent representations are randomly masked before feeding them into the contextual transformer. The model is trained by solving a contrastive task with masked representations, distinguishing the true quantized latent vector from those discrete latents vectors randomly sampled from other masked time steps. During the pre-training, the model learns contextualized representations only with unlabeled speech audio data.

### 2.2. Fine-tuning for ASR

For ASR task, a randomly initialized linear layer is added on the top of the pre-trained model. This linear layer takes the contextualized representations of the pre-trained model and generates most probable words. In fine-tuning, connectionist temporal classification (CTC) loss, an approach for sequence labeling without alignment information between output sequences and input audio, is used to train both the linear layer and the pre-trained model.

## 3. Method

Figure 1 shows the architecture of our Korean ASR system, K-Wav2vec 2.0. The proposed system is a stack of the pre-trained Wav2vec 2.0, the multi-task fine-tuning architecture, and the joint decoder.

### 3.1. Multi-task hierarchical architecture

The multi-task fine-tuning architecture, which consists of the grapheme encoder and the syllable encoder, is trained by taking contextualized representations learned from the pre-trained Wav2vec 2.0 model as an input. Given raw audio $X$, let $C = c_1, ..., c_F$ be the sequence of encoded audio features, i.e., the contextualized representations of the pre-trained model, where $F \in \mathbb{N}^+$ is the number of encoded audio frames. In the grapheme encoder, a linear layer is adopted to project the encoded features into a grapheme vocabulary $g \in G$, followed by the softmax function to produce the posterior probabilities of grapheme sequences $p(g_f|c_f)$ corresponding to each frame. In the syllable encoder, on the other hand, the encoded features are first fed to a stack of transformer encoders, which converts the encoded features $c_f$ to the sequence of hidden vectors $h_f$ to capture the relation between low-level and high-level information. Then, a linear projection layer and the softmax function are applied to produce the posterior probabilities of syllable sequences $p(s_f|h_f)$, where $s \in S$ is the syllable vocabulary. Under the conditional independence assumption, the posterior probability of a complete sequence is computed using the syllable outputs or grapheme outputs of each frame:

$$p_{syll.}(l = s_1, ...s_F|X) = \prod_{f=1}^{F} p(s_f|h_f) \quad (1)$$

$$p_{grap.}(l = g_1, ...g_F|X) = \prod_{f=1}^{F} p(g_f|c_f) \quad (2)$$

### 3.2. Multi-task fine-tuning

We used the CTC loss when training the model [17]. The CTC is an approach for sequence labeling, where the lengths of label sequence and output frames are different. Given a ground truth label sequence $Y = y_1, ...y_l$, the CTC process can expand $Y$ to a set of sequences $\Omega(Y)$ by adding a blank token between consecutive labels and allowing each label to be repeated, where the length of each sequence in the extended set $\Omega(Y)$ and output frames are same. Then the posterior conditional probability of label sequence with the CTC process is computed as follows:

$$p^{ctc}_{syll.}(Y|X) = \sum_{l \in \Omega(Y)} p_{syll.}(l|X) \quad (3)$$

The CTC loss is the log posterior conditional probability of label sequence with the CTC process. The objective function of the proposed architecture is the weighted sum of grapheme and syllable-level CTC losses to learn the two tasks simultaneously:

$$L_{MTL} = \lambda \log p^{ctc}_{syll.}(Y|X) + (1 - \lambda) \log p^{ctc}_{grap.}(Y|X) \quad (4)$$

where $\lambda : 0 \leq \lambda \leq 1$ is a hyper-parameter that controls the trade-off between the importance of syllable and grapheme.

### 3.3. Joint decoder

In the inference step, the joint decoder combines grapheme-level and syllable-level beam search results to find the most confident sequence within limited practical time. In contrast to prior works [18, 19], we only use outputs of acoustic model which allow us to build ASR system without external language models. To extract candidate sequences from outputs with the CTC process, we use the CTC beam search decoder described in [20]. The beam search decoding iteratively finds candidates over time-steps of CTC output and scores them with given probabilities of each time-step. Given syllable outputs, the beam search decoding returns several most likely sequences, syllable candidates $\hat{S}$, and their corresponding probabilities. The beam search results of grapheme outputs are grapheme candidates $\hat{G}$ and their probabilities in the same way. The beam search of each level is conducted separately to find their candidates. The objective of the joint decoder is to find the most probable se-

quence $\hat{Y}$ among them:

$$\hat{Y} = \underset{Y \in \hat{S} \cup \hat{G}}{\arg\max} \{\gamma p^{ctc}_{syll.}(Y|X) + (1-\gamma) p^{ctc}_{grap.}(Y|X)\} \quad (5)$$

where $\gamma : 0 \leq \gamma \leq 1$ is a weight that controls the contribution of results from two different levels to the final output. In the joint decoding process, the proposed model can be more robust than a single encoder (syllable-based decoder or grapheme-based decoder) by adjusting the probability of syllable and grapheme candidate where there is the same candidate in each beam search result $\hat{S} \cap \hat{G}$. The additional candidates to syllable output can be generated by using grapheme candidates, where non-overlapped candidates of grapheme beam search exist $\hat{G} - (\hat{S} \cap \hat{G})$. Because the grapheme modeling unit has less vocabulary but can express more syllables by combining graphemes, the additional candidates from the grapheme beam search result alleviate the OOV problem and the data sparseness problem pertaining to syllable output.

### 3.4. Cross-lingual transfer pre-training

To leverage the ASR performance with limited data, we explore a cross-lingual transfer by further pre-training the existing English model on Korean dataset. The motivation behind this approach is the early successes of further pre-training in natural language processing [21, 22, 23, 24]. They investigated the impact of additional pre-training with various language domains by further pre-training the already pre-trained models on a task-relevant corpus, followed by the classification task of the target domains. Recent research [21] has shown the benefit of further pre-training with the generalized models on target-specific data to specialize models in their target task. As long as speech representations of pre-trained models are known to share generalized features across languages [15], further pre-training on the target language can also benefit by taking advantage of shared information. In the proposed model, we further pre-train the English Wav2vec 2.0, which is the pre-trained on English dataset(960 hours), on Korean dataset (965 hours). When this additional pre-training is completed, the model is fine-tuned with Korean-labeled speech data for the downstream task.

Unlike other well-used cross-lingual pre-training methods in speech domain [15, 16, 25], which use a large amount of multilingual corpus composed of various languages from the initial stage of model pre-training, our proposed two-stage pre-training method utilizes the existing English model. Therefore, the proposed method does not have the disadvantage of putting the target language into the multilingual corpus first and developing it from scratch when the corresponding language is not included in the multilingual corpus.

## 4. Experiment setup

### 4.1. Datasets

We verify the proposed method with Korean speech datasets, including large-corpus dialog dataset, Kosponspeech [8], and call-based benchmark speech corpus, Clovacall [9]. The Ksponspeech consists of train, development, evaluation-clean, and evaluation-other, total of 1000 hours. The dual transcription, the phonetic script written with the original sound as possible and the orthographic script written with Korean standard orthographic rules by modifying numeric and abbreviation notation, is provided for downstream tasks in parallel. Following the preprocessing guideline [7], we split the dual transcription into the phonetic and orthographic scripts, and evaluate our method. The Clovacall has relatively short dialogues, containing 50 hours of label data for training and 1 hour for testing. We randomly sampled 10% of the training set for use as a development set for validation. The amount of Clovacall training data is relatively small to cover all vocabulary of the evaluation set; there is OOVs and data sparseness problem.

### 4.2. Pre-trained model

In our experiment, we only used Ksponspeech training data to pre-train the models. To build Korean adapted model called K-Wav2vec 2.0, we utilize the English Wav2vec 2.0 released by [2], which is pre-trained on 960 hours of English dataset, the Librispeech. We further pre-train the English model on 965 hours of Ksponspeech training data for 400k updates. We mostly followed the experimental settings of the English model to pre-train the models, including a learning rate of 5e-4, an Adam optimizer [26], and 32k warm-up steps. For stable pre-training, we used silence elimination similar to that in [7], excluding any ranges under 30 dB in raw audio. Because the dataset was recorded in a quiet environment, eliminating prolonged silence in conversation makes the model converge fast and is memory efficient. To compared our further pre-training approach, we pre-trained the base architecture from the initial state, and we call this model S-Wav2vec 2.0. We also compared our model to the English Wav2vec 2.0 (E-Wav2vec 2.0).

### 4.3. Fine-tuning strategies

After pre-training, we conducted multi-task fine-tuning, as shown in Figure 1. The graphene encoder is a single linear layer, and the syllable encoder is a stack of 2 transformer blocks, which contains model dimension 768 and 8 attention heads, and a linear layer. These two modules were fine-tuned using labeled training data with multi-task loss. We used a hyper-parameter, $\lambda$, of 0.5 for multi-task loss to reflect the grapheme and syllable CTC loss equally. We trained models with the Adam optimizer and a tri-state rate scheduler, where a learning rate of 1e-5 was applied to Ksponspeech and 3e-5 on Clovacall. In the low-resource experiment using Clovacall, we froze the parameters of pre-trained parts for the first 10k updates to train the multi-task architecture first. Then, all parameters, including the pre-trained network and the fine-tuning architecture, were optimized for 70k updates. With large-corpus data, we fine-tuned all parameters together for 320k updates without an initial freezing state. The modified SpecAugment [2], a strategy to make the model robust to noise by randomly masking embedding spans, was also used. For evaluation, we chose the checkpoint having the lowest word error rate (WER) on the development subset. To make the decoding process computationally efficient in the training phase, we selected the maximum probability syllables of each sequence and decoded it based on the inverse CTC process to collapse consecutive labels and delete the blank tokens. More details are available online[1].

## 5. Results

To investigate the effect of multi-task fine-tuning and decoding strategies, we built six K-Wav2vec 2.0 models as shown in Table 1 and 2. The first column represents the pre-training method with the fine-tuning structure, and the second column denotes the used decoding scheme. The transformer for fine-tuning architecture is identical to the syllable encoder of the multi-task

[1] https://github.com/JoungheeKim/K-wav2vec

Table 1: *Evaluation results on the Ksponspeech eval-clean and eval-other sets with dual transcription.*

| Model | Decoding | Eval-clean | | Eval-other | |
|---|---|---|---|---|---|
| | | CER | sWER | CER | sWER |
| **Ksponspeech - phonetic** | | | | | |
| Big Transf. | syllable | 7.77 | 12.87 | 8.26 | 14.40 |
| K-Wav2vec 2.0 | | | | | |
| + Linear | grapheme | 7.30 | 12.48 | 7.81 | 14.12 |
| + Linear | syllable | 6.96 | 12.01 | 7.48 | 13.68 |
| + Transf. | syllable | 6.98 | 12.02 | 7.53 | 13.84 |
| + Multi-task | grapheme | 7.29 | 12.47 | 7.83 | 14.20 |
| + Multi-task | syllable | 6.90 | 11.87 | 7.35 | 13.38 |
| + Multi-task | joint | **6.88** | **11.76** | **7.33** | **13.27** |
| S-Wav2vec 2.0 | | | | | |
| + Transf. | syllable | 7.16 | 12.40 | 7.64 | 14.01 |
| + Multi-task | joint | 7.08 | 12.16 | 7.70 | 14.13 |
| E-Wav2vec 2.0 | | | | | |
| + Transf. | syllable | 8.58 | 15.34 | 9.23 | 17.61 |
| + Multi-task | joint | 8.10 | 14.20 | 8.81 | 16.53 |
| **Ksponspeech - orthographic** | | | | | |
| Big Transf. | syllable | 8.46 | 14.02 | 9.23 | 16.31 |
| K-Wav2vec 2.0 | | | | | |
| + Linear | grapheme | 7.89 | 13.63 | 8.90 | 16.35 |
| + Linear | syllable | 7.49 | 13.03 | 8.43 | 15.69 |
| + Transf. | syllable | **7.48** | 13.03 | 8.45 | 15.67 |
| + Multi-task | grapheme | 8.05 | 13.96 | 9.09 | 16.75 |
| + Multi-task | syllable | 7.53 | 13.09 | **8.36** | 15.42 |
| + Multi-task | joint | 7.54 | **13.03** | 8.37 | **15.38** |

model, and the linear is the same as the grapheme encoder. For all models, we used the beam size of 100 for the decoding process, and a $\gamma$ of 0.5 was applied to the joint decoder. We evaluate the performance of each model in terms of character error rate (CER) and space-normalized WER (sWER). The sWER is modified version of word error rate (WER) to evaluated Korean ASR where space rules are flexible [8]. Note that the grapheme outputs are converted to syllables before evaluation.

### 5.1. High-resource evaluation

We first evaluated the proposed model in the Ksponspeech, which is a high-resource data, with two different transcriptions. As a baseline, we reproduced a big transformer of the previous work [8] with the syllable modeling unit. Table 1 show the performance of each model for evaluation-clean, and evaluation-other datasets. Based on the results, we can derive the following observations. First, the syllable-based models generally outperformed the grapheme-based models for both transcripts. Second, the multi-task fine-tuning seemed to slightly improve the Korean ASR performance, and joint decoding brings additional improvement. Third, our proposed model yielded the best performance for sWER on all evaluation sets, outperforming the baseline model in most cases. Based on the aforementioned observations, we can conclude that both components of our proposed model, i.e., multi-task learning and joint decoding, not only contribute to the performance improvement separately but also show a synergy effect when they are used together.

We also observe that our proposed pre-training models, K-Wavevc 2.0, clearly outperform S-Wav2vec 2.0 and E-Wav2vec 2.0. Compared to S-Wav2vec 2.0, further pre-training leads to large improvement without additional data by using pre-trained representation from another language. These results demonstrate that the further pre-training is an effective approach in cross-lingual transfer by language adaptation.

Table 2: *Evaluation results on the Clovacall with OOV syllables recovered by grapheme and joint decoding.*

| Model | Decoding | Evaluation | | # OOV |
|---|---|---|---|---|
| | | CER | sWER | Recovered |
| **Clovacall - Base** | | | | |
| Paper-DS2[9] | syllable | 11.4 | | |
| Paper-LAS[9] | syllable | 15.1 | | |
| Small Transf. | syllable | 11.23 | 19.81 | |
| K-Wav2vec 2.0 | | | | |
| + Linear | grapheme | 6.59 | 11.94 | 9(21.9%) |
| + Linear | syllable | 6.82 | 12.31 | |
| + Transf. | syllable | 13.43 | 24.46 | |
| + Multi-task | grapheme | 6.48 | 11.64 | 6(14.6%) |
| + Multi-task | syllable | 7.08 | 13.14 | |
| + Multi-task | joint | **6.41** | **11.49** | 6(14.6%) |

### 5.2. Low-resource evaluation

The second experiment was conducted on the Clovacall, where a total of OOV syllable occurrence is 41 in the testset. Table 2 shows that various configurations with further pre-trained models significantly outperformed the models introduced in [9] and the baseline model, small transformer, released by [27]. Since further pre-training with unlabeled data lead the model more adapt to Korean, fine-tuning with small amounts of labeled data is found to be effective. In this experiment, we also investigate whether decoding strategy alleviate OOV syllables, which can be restored with our grapheme dictionary. We observed that both grapheme and joint decoding could alleviate the OOV problem. The model with grapheme unit recovered 21.9% OOV syllables, and the multi-task model with joint decoding recovered 14.6% of OOV occurrences as much as grapheme decoding. This shows that the beam search result of grapheme utilized in the joint decoding process can alleviate the OOV problem when syllable decoding produces unconfident results for OOV.

In contrast to the high-resource experiment, an interesting observation is that the grapheme-based model outperformed the syllable-based model. Because of insufficient training data in the Clovalcall, syllable-based models do not have enough opportunities to train rare vocabularies and have OOV problems, resulting in an inaccurate vocabulary prediction. However, our proposed architecture with joint decoding gets a robust performance in both datasets. Considering the various conditions in the real world, the robustness of our proposed architecture can play an important role in a service-level ASR system.

## 6. Conclusion

In this work, we presented a multi-task fine-tuning architecture with a joint decoder and further pre-training approach for the Korean ASR. Our experiments show that the multi-task model can generate multi-level outputs without performance degeneration, and the joint decoder enhances the ASR performance by overcoming the drawbacks of each modeling unit. Our system achieved the best performance in terms of sWER and the most robust performance in both datasets. We also found the further pre-training approach effective using pre-trained representations of the English model. In the future, we will investigate decoding strategies using acoustic model and language model, which can relieve the OOV problem by additional vocabulary.

# 7. References


[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[3] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[4] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDROM," *Linguistic Data Consortium*, 1993.

[5] H. Park, S. Seo, D. J. Rim, C. Kim, H. Son, J.-S. Park, and J.-H. Kim, "Korean grapheme unit-based speech recognition using attention-ctc ensemble network," in *2019 International Symposium on Multimedia and Communication Technology (ISMAC)*. IEEE, 2019, pp. 1–5.

[6] M.-h. Lee and J.-H. Chang, "Korean speech recognition based on grapheme," *The Journal of the Acoustical Society of Korea*, vol. 38, no. 5, pp. 601–606, 2019.

[7] S. Kim, S. Bae, and C. Won, "Kospeech: Open-source toolkit for end-to-end korean speech recognition," *arXiv preprint arXiv:2009.03092*, 2020.

[8] J.-U. Bang, S. Yun, S.-H. Kim, M.-Y. Choi, M.-K. Lee, Y.-J. Kim, D.-H. Kim, J. Park, Y.-J. Lee, and S.-H. Kim, "Ksponspeech: Korean spontaneous speech corpus for automatic speech recognition," *Applied Sciences*, vol. 10, no. 19, p. 6936, 2020.

[9] J.-W. Ha, K. Nam, J. Kang, S.-W. Lee, S. Yang, H. Jung, H. Kim, E. Kim, S. Kim, H. A. Kim *et al.*, "Clovacall: Korean goal-oriented dialog speech corpus for automatic speech recognition of contact centers," *Proc. Interspeech 2020*, pp. 409–413, 2020.

[10] J. Wang, J. Kim, S. Kim, and Y. Lee, "Exploring lexicon-free modeling units for end-to-end korean and korean-english code-switching speech recognition," *Proc. Interspeech 2020*, pp. 1072–1075, 2020.

[11] R. Sanabria and F. Metze, "Hierarchical multitask learning with ctc," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 485–490.

[12] K. Krishna, S. Toshniwal, and K. Livescu, "Hierarchical multitask learning for ctc-based speech recognition," *arXiv preprint arXiv:1807.06234*, 2018.

[13] S. Chen, X. Hu, S. Li, and X. Xu, "An investigation of using hybrid modeling units for improving end-to-end speech recognition system," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6743–6747.

[14] S. Ueno, H. Inaguma, M. Mimura, and T. Kawahara, "Acoustic-to-word attention-based model complemented with character-level ctc-based model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5804–5808.

[15] M. Riviere, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7414–7418.

[16] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[17] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[18] T. Hori, S. Watanabe, and J. R. Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 287–293.

[19] T. Hori, S. Watanabe, and J. Hershey, "Joint ctc/attention decoding for end-to-end speech recognition," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 518–529.

[20] A. Hannun, "Sequence modeling with ctc," *Distill*, 2017, https://distill.pub/2017/ctc.

[21] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8342–8360.

[22] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *China National Conference on Chinese Computational Linguistics*. Springer, 2019, pp. 194–206.

[23] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3615–3620.

[24] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[25] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 58–68.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[27] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.-E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *Proc. Interspeech 2018*, pp. 2207–2211, 2018.