Korean Automatic Speech Recognition based on HuBERT

Korea University COSE461 Final Project

Zhongsong Gao
Department of Computer Science and Engineering
Team 23
2018320136

Eunsang Lee
Department of Mechanical and Engineering
Team 23
2018170603

Abstract

In this paper we train a HuBERT-based Korean ASR model, which is then compared and optimized with an existing newer model (K-wav2vec). Due to constraints, we only used the Korean dataset provided by AI hub for pre-training and fine-tuning optimization. The accuracy of HuBERT is slightly better than the K-wav2vec model in general.[4][5]

1 Introduction

Through the reading of related papers, we know that in the pre-training process of wav2vec2.0, the transformer context and vector quantization(VQ)[3][5] modules are joint training. Here, VQ is the target for learning while training. In the pre-training process of HuBERT, VQ is trained and can directly provide the target. Because the VQ module is randomly initialized and the quality of the target provided at the beginning of training is not high, many updates at the beginning of training are not so effective. In the training process of HUBERT, because VQ is a model trained in advance, it can provide a better target in the early stage of pre-training, so that the model can learn faster.[2][3][4][5]

Because the VQ module is randomly initialized, the quality of the targets provided in the pre-training stage is not high, and many updates in the pre-training stage are not so effective. Then, coupled with the quality of the training dataset and the complexity of the Korean grammar on the target, provides an inaccurate or incorrect target, resulting in poor performance of the final Korean ASR model.[3][4]

Therefore, we believe that HuBERT with pre-trained VQ to provide targets can avoid the performance degradation caused by initially providing low-quality

targets and the impact of complex Korean grammar and training dataset quality.[1][3]

Link to our code:https://github.com/dennisg21/NLP-COSE461-final-project

2 Related Work

2.1 K-wav2vec

K-Wav2Vec 2.0 is a modified version of Wav2vec 2.0 designed for Korean automatic speech recognition by exploring and optimizing various factors of the original Wav2vec 2.0. This model used as a comparison.[6]

2.2 HuBERT

The Hidden-Unit BERT (HuBERT) approach for self-supervised speech representation learning, which utilizes an offline clustering step to provide aligned target labels for a BERT-like prediction loss. It is base model we used.[3]

2.3 HuBERT-EE

An early exit scheme for ASR, namely HuBERT-EE[8], that allows the model to stop the inference dynamically. Our model is optimized with reference to it.

2.4 KoSpeech

an open-source software, is modular and extensible end-to-end Korean automatic speech recognition (ASR) toolkit based on the deep learning library PyTorch[7]. We refer to his data processing method when we process data.

2.5 connectionist tem- poral classification(CTC) loss

It calculates a loss between a continuous (unsegmented) time series and a target sequence. It does this by summing over the probability of possible alignments of input to target, producing a loss value which is differentiable with respect to each input node.

3 Approach

We use HuBERT and wav2vec 2.0 as the base model for training with the Korean data sets. Use the method of null control variables to verify the influence of different VQ(vector quantization) provision methods on the Korean ASR model, and to verify the conjecture that HuBERT is more suitable for Korean than wav2vec 2.0. In order to improve the training efficiency under limited hardware, an early exit module is added in our HuBERT model, model's structure likes figure 1[8].

3.1 Model Training Method

Due to hardware limitations and time limitations. We utilized English wav2vec 2.0 model which is pretrained on 960h of Librispeech without fine-tuning and English

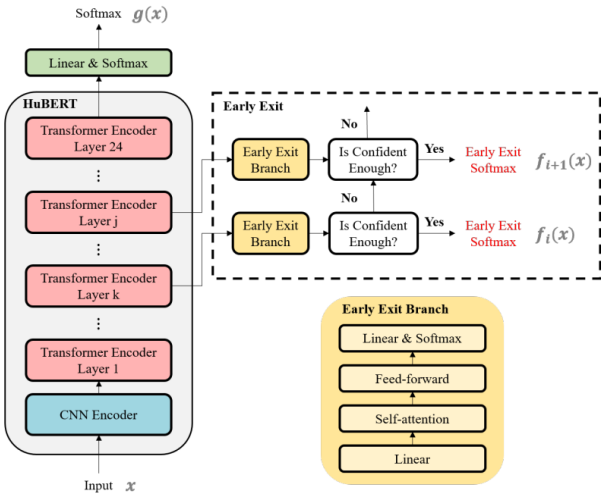

Figure 1: Model structure referenced to [HuBERT-EE: Early Exiting HuBERT for Efficient Speech Recognition]

HuBERT model which is pretianed on 960h of Librispeech without fine-tuning. We use these existing English models, train or fine-tune them using the Korean datasets.

3.2 Data Preprocess Method

We refer to the data preprocessing method in kospeech[7], and delete all symbols except Korean fonts in the Korean data provided by AI HUB.

Data Preprocess Example	
-original	o/ 나도 몰라. 나 그/ (3G)/(쓰리 쥐)* 하나도 안 봤음. 어.
-after	나도 몰라 나 그쓰리 쥐 하나도 안 봤음어

3.3 Impact verification of vector quantization(VQ)

We use the unpreprocessed Korean dataset to train HuBERT and wav2vec2.0, to verify that wav2vec 2.0 provides the target VQ while training, and it is not more accurate than the pre-trained, target VQ in HuBERT high degree. The quality of the model is reduced, and it will be more affected by the quality of the dataset[9].

For the sake of rigor, in order to prevent the quality of the model from being degraded due to the quality of the data set rather than the VQ problem, we also used the preprocessed data to train to verify the above conclusions.

3.4 Early Exiting Module

Referring to the early exiting[9] method proposed in the HuBERT - EE paper[8], it is applied to our trained HuBERT model.

When calculating confidence, the average confidence score of $f_i(x)$ is given as[6][7]:

$$Confidence = \frac{1}{T}\sum_{T}\max_{cc} f(x)^{(c)} \tag{1}$$

When calculating entropy, the entropy of the i th early exit branch's output $f_i(x)$ can be computed as[6][7]:

$$Entropy = \frac{1}{T * C} \sum_{T} \sum_{C} f_i(x) * log f_i(x) \tag{2}$$

The structure of the early exiting module is shown in Figure 2[8].

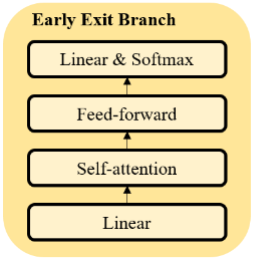

Figure 2: Early Exiting module structure

4 Experiments

4.1 Data

We used the korean speech dataset provided by AI HUB[6][7], which includes daily conversations, shopping conversations, political conversations, economic conversations, hobby conversations, AI assistants, simultaneous interpretation, emotional conversations, voice intelligence services, and other types of data about 140GB in size. Due to the limitation of hardware performance, we cannot train all the data, so only the amount of data we can train is selected as the training set

4.2 Evaluation method

We used the character error rate (CER), WER all of which are commonly used in Korean ASR as performance evaluation metrics.

D : Number of syllables incorrectly deleted in speech recognized text
S : Number of syllables incorrectly replaced in speech recognized text
I : Number of syllables incorrectly added to speech recognized text
N: number of syllables in the correct text.

$$CER = \frac{S + D + I}{N} \qquad WER = \frac{S + D + I}{N} \tag{3}$$

4.3 Experimental details

The model is trained and fine-tuned by using PyTorch. Change the model structure in the source file and add a module to build the model we need to use in Project.We did not use all the data of AI HUB for training, and selected about one thousandth of the data for training, which took about 5 to 6h. Training and fine-tuning are done in about 12 times in total.

4.4 Results

4.4.1 Unprocessed datasets

Model	Decoding	E-Clean	E-Other
wav2vec 2.0	Fairseq-LM	CER=40.656 WER=60.778	CER=49.233 WER=66.932
HuBERT	Fairseq-LM	CER=36.586 WER=55.331	CER=47.887 WER=62.556

4.4.2 Preprocessed datasets

Model	Decoding	E-Clean	E-Other
wav2vec 2.0	Fairseq-LM	CER=37.112 WER=56.998	CER=46.676 WER=64.114
HuBERT	Fairseq-LM	CER=33.452 WER=51.117	CER=43.913 WER=57.776
HuBERT(EEM)	Fairseq-LM	CER=35.936 WER=55.110	CER=44.617 WER=60.937

- First of all, due to hardware limitations, we did not train all the training data, we selected a small part of it for training, So from the results, the performance of the model will not be particularly good.

- Second, the results also verify that VQ has an impact on model performance. Regardless of the quality of the dataset, providing the VQ of the target at the same time as pre-training will result in a low quality of the provided target due to the ineffectiveness of the previous training update, which will eventually lead to a decrease in the overall model performance.

- We know that early exiting works well here, it saves training time, and the gap between performance and fully trained model is not very large.

5 Analysis

Since we use incomplete training data due to hardware performance limitations, there may be some special data that may be very suitable for a certain model. As a result, although the result verifies the conjecture, the result is not rigorous because the training data is very suitable for a certain model. So in the next work, out of rigor, we should use the complete training data to complete our experiments again.

6 Conclusion

We found that the VQ that provides the target at the same time of pre-training is easily affected by the data set. The initial VQ module itself is randomly initialized, the quality of the target provided in the early stage of training is not high, and the quality of data memory will magnify this shortcoming, making the VQ provided by The target is inaccurate, which ultimately leads to a degraded model performance. Then, we also found that the targets provided by the pretrained and fine-tuned VQ are more accurate. Since the training of the VQ and the training of the model are

separated, yes it is more conducive to fine-tuning. We think this approach is more beneficial for grammatically complex languages. While improving performance, reducing the size of the model can be described as killing two birds with one stone, achieving both.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[4] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

[5] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

[6] Jounghee Kim and Pilsung Kang. K-wav2vec 2.0: Automatic speech recognition based on joint decoding of graphemes and syllables. *arXiv preprint arXiv:2110.05172*, 2021.

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

[8] Ji Won Yoon, Beom Jun Woo, and Nam Soo Kim. Hubert-ee: Early exiting hubert for efficient speech recognition. *arXiv preprint arXiv:2204.06328*, 2022.

[9] Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. Deebert: Dynamic early exiting for accelerating bert inference. *arXiv preprint arXiv:2004.12993*, 2020.

[10] Yingzhi Wang, Abdelmoumene Boumadane, and Abdelwahab Heba. A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding. *arXiv preprint arXiv:2111.02735*, 2021.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10]

A Appendix: Team contributions

We did all the experimental and analytical work together in our spare time. Almost all work is done together for better understanding and learning