VNU Journal of Science: Comp. Science & Com. Eng, Vol. 38, No. 1 (2022) 30-36

VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

ASR - VLSP 2021: Semi-supervised Ensemble Model for Vietnamese Automatic Speech Recognition

Pham Viet Thanh, Le Duc Cuong, Dao Dang Huy, Luu Duc Thanh, Nguyen Duc Tan, Dang Trung Duc Anh, Nguyen Thi Thu Trang*

Hanoi University of Science and Technology,
1 Dai Co Viet, Bach Khoa, Hai Ba Trung, Hanoi, Vietnam

Received 27 December 2021
Revised 5 April 2022; Accepted 5 May 2022

Abstract: Automatic speech recognition (ASR) is gaining huge advances with the arrival of End-to-End architectures. Semi-supervised learning methods, which can utilize unlabeled data, have largely contributed to the success of ASR systems, giving them the ability to surpass human performance. However, most of the researches focus on developing these techniques for English speech recognition, which raises concern about their performance in other languages, especially in low-resource scenarios. In this paper, we aim at proposing a Vietnamese ASR system for participating in the VLSP 2021 Automatic Speech Recognition Shared Task. The system is based on the Wav2vec 2.0 framework, along with the application of self-training and several data augmentation techniques. Experimental results show that on the ASR-T1 test set of the shared task, our proposed model achieved a remarkable result, ranked as the second place with a Syllable Error Rate (SyER) of 11.08%.

Keywords: Speech Recognition, Vietnamese, Semi-Supervised Learning, Self-Training

1. Introduction

Automatic speech recognition is a task which takes a speech segment as an input and generates the corresponding written format. In recent years, deep learning approaches have enabled ASR systems to gain huge advances and surpass human-level performance [1].

With the arrival of End-to-End architectures, ASR has become a much more active field. Semi-supervised learning methodology, which has the ability to utilize unlabeled data in training, is now one of the most popular methods in speech recognition. Pioneering works include Wav2vec 2.0 [2] and HuBERT [3]. These frameworks come with effective pre-training techniques which can learn speech representations from unlabeled speech, followed by fine-tuning the model with labeled data. Wav2vec 2.0 has shown superior

* Corresponding author.
E-mail address: trangntt@soict.hust.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.332

performance on Librispeech [4] compared to the previous state-of-the-art while using a much smaller amount of labeled data.

Although these systems achieve quite remarkable results on English datasets, it cannot be guaranteed that they can reach the equivalent performance on other languages. In some languages, there are only a few labeled datasets, and obtaining unlabeled data of the same domain can be very time-consuming. In the case of the Vietnamese language, there have been several public datasets released over the years, such as VLSP 2020 and VIVOS. However, these datasets are relatively small compared to English datasets, and thus semi-supervised learning may not be the best option.

In this paper, we propose an ASR system participating in the Automatic Speech Recognition Shared Task of VLSP 2021 [5]. The system is based on the Wav2vec 2.0 framework, along with the application of several data augmentation techniques. Additionally, self-training is used to utilize the unlabeled in-domain data. The final system is the combination of these techniques via an ensemble mechanism.

The remaining of this paper is organized as follows. The related works are described in Section 2, with our methodology given in Section 3. In Section 4, we discuss the experimental setup and Section 5 shows the results of the system. Finally, conclusions are drawn in Section 6.

2. Related works

Previous works have shown the efficiency of semi-supervised models on various tasks. Wav2vec 2.0 and HuBERT, with the ability to learn speech representations from unlabeled data, have been applied to achieve state-of-the-art results on several speech processing tasks other than speech recognition. The authors of [6] show that remarkable results can be obtained by fine-tuning Wav2vec 2.0 and HuBERT on downstream tasks, including speech emotion recognition, speaker verification and spoken language understanding. Other works include assessing the self-supervised architecture in French [7], or using Wav2vec 2.0 for End-to-End speech translation [8]. With these above-mentioned researches, we can be certain that speech representations learning is going to be much more developed and advanced over the next few years.

As being an approach to efficiently use unlabeled data, self-training has appeared frequently in speech recognition papers, along with its variants. [9] discusses adapting and improving noisy student training for ASR. In [10], the authors propose several methods for data-filtering for ASR self-training. The combination of self-training and unsupervised pre-training has proven to be effective for improving speech recognition performance, according to [11]. Overall, self-training is an efficient way for leveraging untranscribed data in the speech recognition task.

3. Methodology

3.1. Data Pre-processing

3.1.1. VLSP 2021 ASR Datasets

For the participation in the ASR-T1 task, we only use the datasets provided by the competition organizers. There are 3 training datasets, along with 1 development set. Table 1 shows the total duration of each dataset.

The general domain training set contributes 215.6 hours of speech. The two in-domain training sets consist of audios collected from online lectures, with one dataset having no transcriptions. Lastly, the in-domain development set contains only 2.5 hours of speech. This dataset is used for hyperparameter tuning.

Table 1. VLSP 2021 ASR Datasets:

Dataset	Hours
General domain training set	215.6
In-domain training set	23.0
Untranscribed in-domain training set	360.7
In-domain development set	2.5

3.1.2. Data pre-processing

As the datasets can contain some noises and errors, several pre-processing steps need to be done. The overall pipeline is illustrated in Figure 1. The first step is removing noisy audio samples from the data. For this step, we calculate the signal-to-noise ratio (SNR) for each audio using the WADA-SNR [12] algorithm. All audio samples having an SNR below 5.0 dB are removed from the training sets. The reason for choosing this value is because we found that in most audio samples with SNR below 5.0 dB, the volume of background noise starts to get higher than the volume of speech. Keeping these samples in the training set may result in lower model performance.

Another problem is that because the datasets are manually labeled, there can be errors in the transcriptions, such as typing errors or spelling mistakes. After looking through the transcriptions, we found that most errors come from the in-domain training set and in-domain development set. We then performed validation on the transcriptions of these two datasets. About over 5,000 samples with transcriptions mistakes are corrected manually.

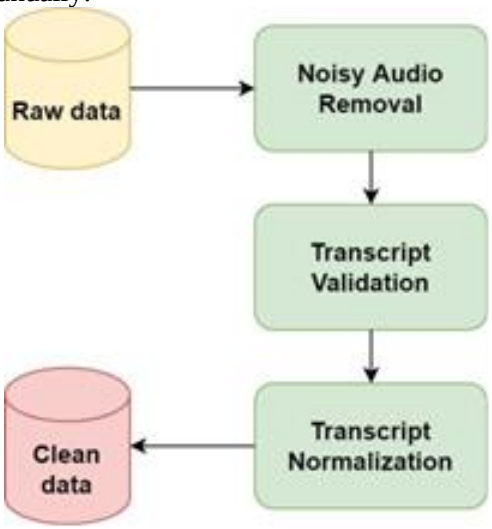

Figure 1. Overall data pre-processing pipeline.

Finally, after the validation step, the transcriptions are normalized. Numbers, abbreviations and loanwords are transformed into their spoken formats at this step. Numbers are normalized with a simple rule-based algorithm. In the case of loanwords and abbreviations, we use a dictionary which contains exactly one spoken format for each word.

3.2. Wav2vec 2.0

With a large amount of untranscribed training data, a semi-supervised architecture is preferable to a pure supervised one. In this section, we briefly discuss the architecture and the training phases of the chosen framework - Wav2vec 2.0 [2]. The architecture of Wav2vec 2.0 includes three main parts: the feature encoder, the context network and the quantization module. The feature encoder is a convolutional neural network (CNN), and its job is to take the raw waveform as input and output the latent speech representation. The context network then learns the contextualized representation from the latent speech representation. This context network follows the Transformer architecture [13]. Finally, the quantization module is used to generate quantized representation from the output of the feature encoder. The module is applied only in the pre-training phase.

3.2.1. Pre-training Phase

In the pre-training phase, the model learns contextualized representations of speech from unlabeled data. A portion of the feature encoder outputs is masked randomly before being fed to the context network. The final speech representation is learned by solving a contrastive task which is to identify the true quantized representation for a masked time step among a set of representations from the other masked time steps.

3.2.2. Fine-tuning Phase

For fine-tuning the model on downstream tasks, additional layers can be placed on top of the pre-trained model. In the case of ASR, a linear layer is added to map the contextualized speech representations learned from the pre-training process into a readable sentence.

Connectionist temporal classification (CTC) loss [14] is used to optimize the model during the fine-tuning phase.

3.3. Self-training Approach

To further utilize the untranscribed in-domain training set, we opt for a self-training strategy similar to [15]. The overall process is as follows:

- Training an acoustic model and a language model with the labeled training datasets.
- Using the trained acoustic model and language model to label the unlabeled data. This step is called pseudo-labeling.
- Filtering out bad transcriptions from pseudo-labeled data by using the perplexity values produced by the language model. Only the transcriptions with low perplexity are kept.
- Training a new acoustic model with the original training data and the pseudo-labeled training data.

Although iterative pseudo-labeling could be used for better modeling quality [16], we perform only 1 iteration to save computational cost.

3.4. Data Augmentation Techniques

We only apply 2 data augmentation techniques during the training process. The first one is time masking, which simply makes a random part of the audio silent. The second technique is manipulating the speed of the audio signal, called speed perturbation [17].

3.5. Loanwords and Abbreviations Handling

Table 2. Spoken formats of loanwords and abbreviations:

Original word	Spoken formats
adn	ây đi en, a đê nờ
keyword	ki guất, ki guật, ki guốt, ki guộc
alpha	an pha
cm	xen ti mét, xăng ti mét

As mentioned above, transcriptions are normalized before the training process. Thus, the spoken formats of the loanwords and abbreviations output by the system have to be converted to their original forms. We handle this by creating a list of spoken formats for loanwords and abbreviations and performing matching with the transcriptions. The original words can have multiple spoken.

4. Experimental Setup

4.1. Wav2vec 2.0 Pre-training

We chose to use the Wav2vec 2.0 Base architecture in all experiments. The Wav2vec 2.0 Base configuration includes 12 transformer blocks in the context network, with model dimension 768, inner dimension 3,072 and 8 attention heads. The feature encoder contains seven blocks of CNN and in each block the convolutional layers have 512 channels with strides (5,2,2,2,2,2,2) and kernel widths (10,3,3,3,3,2,2). Overall, the size of the model is about 95 million parameters.

In the pre-training phase, fairseq [18] model implementation is applied. We use all of the available training sets for pre-training, including the general domain training set, the in-domain training set and the untranscribed in-domain training set. The input to the model is raw waveform, with a sampling rate of 16,000 Hz. Adam is chosen as the optimization algorithm during the training process. We run the pre-training phase using an NVIDIA Tesla V100 for about 15 days, with a batch size of 4. An initial learning rate of $5 \times 10{-}4$ is used and decayed linearly.

4.2. Wav2vec 2.0 Fine-Tuning

To perform ensemble modeling for the final system, we do several fine-tuning experiments to obtain a set of base models. The fine-tuning process is implemented using the Huggingface's transformers library [19]. All the models are fine-tuned using the same pre-trained Wav2vec 2.0, and a randomly initialized linear layer is added to predict sequences of characters. Each model is fine-tuned for 25 epochs with Adam optimization. We use a batch size of 4 and an initial learning rate of $2 \times 10{-}4$ with linear decay. The only difference among the models is the training

data. Each model listed below takes about 3 days to converge.

Model 1. The first model is fine-tuned with the in-domain training set and general domain training set.

Model 2. Self-training is applied for the second model. Firstly, to pseudo-label the unlabeled in-domain training set, we combine Model 1 and a language model (LM) with the strategy described in Section 4.3. Then the samples with LM perplexity values higher than 70.0 are removed. Now we fine-tune Model 2 with the original labeled datasets and the filtered pseudo-labeled data.

Model 3. Before training Model 3, we first apply the above-mentioned data augmentation techniques to the in-domain training set. For time masking, a random segment of 200-300ms is masked in each audio sample. In the case of speed perturbation, 2 copies of the original data are generated with speed factors of 0.9 and 1.1, with the original speed factor being 1.0. Model 3 is now fine-tuned with the two original labeled training sets and the augmented in-domain training set.

4.3. Language Modeling

For the combination of the acoustic model and language model, we use a 6-gram language model trained on the transcriptions of the two labeled training datasets. The combination is done via a beam-search decoding algorithm [20] with a beam size of 200, language model weight of 0.5 and word insertion penalty of 5.0.

4.4. Final ASR system

The final system is an ensemble model of the three base models described previously in Section 4.2. First, we take an average of the outputs produced by the base models. This average output is then decoded with the language model described in Section 4.3. Finally, we process the loanwords and abbreviations with the above-mentioned technique to obtain the final output.

5. Evaluation Results

5.1. Evaluation protocol

In VLSP 2021 ASR Shared Tasks, the quality of the systems will be assessed using Syllable Error Rate (SyER). The evaluation metric is as follows:

$$\text{SyER} = \frac{S+D+I}{N} \quad (1)$$

In (1), S , D and I represent the number of substitutions, deletions and insertions, respectively. With C being the number of correct syllables, N is the number of syllables in the reference (N = S + D + C).

5.2. Experimental Results

5.2.1. Modeling Experiment

Table 3. Results of the fine-tuned models and the final system on the development set:

Model	SyER
No pre-training	16.95
Model 1	7.37
Model 2	7.87
Model 3	8.36
Final system	7.01

Table 3 shows the results of the experiments on the provided development set. Firstly, to assess the performance of the semi-supervised strategy, we train a model without the pre-training step. For the comparison, the labeled data used for this model will be the same as Model 1. As shown in the table, the SyER significantly improves when pre-training is applied. Secondly, to assess the performance of the ensemble mechanism, we compare the results of the base models with the final system. The results show that among the base models, Model 1 has the best performance while Model 3 does not perform as well as its counterparts. A possible reason is that the augmented data may harm the model performance by creating unrealistic speech. Compared to the base models, the ensemble model has the optimal result, achieving a SyER of 7.01%.

5.2.2. VLSP 2021 Experimental Results

The organizers of VLSP announce 2 private test sets: the ASR-T1 test set contains 993

audio segments of lecture speech, while the ASR-T2 test set includes 900 segments of spontaneous speech from different domains.

Table 4 describes the results of ASR systems on the ASR-T1 test set. The model we submitted was the ensemble model described above. Our system achieved a SyER of 11.08%, taking second place in the task as announced by the VLSP organizers. The results show that our proposed method only performs slightly better than that of the third team, while the first team achieves an outstanding result - SyER of 8.28%.

Table 4. Results of the systems on the ASR-T1 test set:

Team	SyER
Lightning	8.28
LAB-914-ASR (ours)	11.08
SMARTCALL	12.00
VC-Tus	12.41
VB_ASR	16.68

6. Conclusions

In this paper, we presented our solution for data pre-processing and validation for participating in VLSP 2021 ASR shared task. The overall process includes noisy audio removal, transcript validation and transcript normalization.

We have proposed an ASR system based on semi-supervised learning and ensemble model. The Wav2vec 2.0 framework is used in all experiments. With the pre-training phase, the model can learn speech representations, thus can make use of the provided unlabeled training data. While creating base models for ensemble modeling, 2 data augmentation methods were used, namely time masking and speed perturbation. To further utilize the unlabeled in-domain dataset, we apply a self-training method for generating additional labeled training data, which includes pseudo-labeling the unlabeled data and filtering out bad transcriptions. In our experiments, we found that with the use of the ensemble mechanism, the final system outperforms all the base models. In the ASR-T1 task evaluation, our system was ranked second place with a SyER of 11.08%.

Possible improvements for future works can be changing the base architecture and the ensemble mechanism. We will consider training with different architectures other than Wav2vec 2.0 in the condition of limited labeled data. We will also try to find a better ensemble approach, such as applying a new data sampling algorithm.

References

[1] T.-S. Nguyen, S. Stüker, A. Waibel, Super-Human Performance in Online Low-Latency Recognition of Conversational Speech, Interspeech 2021.

[2] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 12449–12460.

[3] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, A. Mohamed, Hubert: Self-supervised speech representation learning by masked prediction of hidden units, IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 29, 2021, pp. 3451–3460.

[4] V. Panayotov, G. Chen, D. Povey, 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.

[5] D. V. Hai, VLSP 2021 - ASR Challenge: Vietnamese Automatic Speech Recognition, VNU Journal of Science: Computer Science and Communication Engineering 38 (1).

[6] Y. Wang, A. Boumadane, A. Heba, A Fine-tuned Wav2vec 2.0/HuBERT Benchmark For Speech Emotion Recognition, Speaker Verification and Spoken Language Understanding, ArXiv abs/2111.02735.

[7] S. Evain, H. Nguyen, H. Le, M. Z. Boito, S. Mdhaffar, S. Alisamir, Z. Tong, Tomashenko, M. Dinarelli, T. Parcollet, Allauzen, Y. Estève, B. Lecouteux, F. Portet, Rossato, F. Ringeval, D. Schwab, L. Besacier,

LeBenchmark: A Reproducible Framework for Assessing Self-Supervised Representation Learning from Speech, Interspeech 2021.

[8] A. Wu, C. Wang, J. Pino, J. Gu, Self-Supervised Representations Improve End-to-End Speech Translation, Interspeech 2020.

[9] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, Li, Y. Wu, Q. V. Le, Improved Noisy Student Training for Automatic Speech Recognition, Interspeech 2020.

[10] A.-L. Georgescu, C. Manolache, D. Onea¸ta,Cucu, C. Burileanu, Data-Filtering Methods for Self-Training of Automatic Speech Recognition Systems, in: 2021 IEEE Spoken Language Technology Workshop (SLT), IEEE, 2021, pp. 1–7.

[11] Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau, R. Collobert, G. Synnaeve, M. Auli, Self-training and pre-training are complementary for speech recognition, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 3030–3034.

[12] C. Kim, R. M. Stern, Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis, in: Ninth Annual Conference of the International Speech Communication Association, 2008.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[14] A. Graves, S. Fernández, F. Gomez,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.

[15] J. Kahn, A. Lee, A. Hannun, Self-training for end-to-end speech recognition, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 7084–7088.

[16] Q. Xu, T. Likhomanenko, J. Kahn, A. Y. Hannun, G. Synnaeve, R. Collobert, Iterative Pseudo-Labeling for Speech Recognition, ArXiv abs/2005.09267.

[17] T. Ko, V. Peddinti, D. Povey, S. Khudanpur, Audio augmentation for speech recognition, in: Sixteenth annual conference of the international speech communication association, 2015.

[18] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, Ng, D. Grangier, M. Auli, fairseq: A Fast, Extensible Toolkit for Sequence Modeling, in: NAACL, 2019.

[19] T. Wolf, L. Debut, V. Sanh, J. Chaumond, Delangue, A. Moi, P. Cistac, T. Rault, Louf, M. Funtowicz, J. Brew, HuggingFace's Transformers: State-of-the-art Natural Language Processing, ArXiv abs/1910.03771.

[20] A. L. Maas, A. Y. Hannun, D. Jurafsky, A. Ng, First-Pass Large Vocabulary Continuous Speech Recognition using Bi-Directional Recurrent DNNs, ArXiv abs/1408.2873.