Vietnamese Automatic Speech Recognition with Transformer

Duong Trinh Anh, Sam Dang Van, Tuan Do Van and Vi Ngo Van

December 4, 2021

Vietnamese Automatic Speech Recognition with Transformer

VCCorp

Duong Trinh Anh, Sam Dang Van, Tuan Do Van, Vi Ngo Van

{duongtrinhanh, samdangvan, tuandovan}@tech.admicro.vn, vingovan@admicro.vn

Abstract

Recently, speech recognition using end-to-end models is gradually becoming a trend and has superior performance compared to traditional methods. The most frequently used methods are the combination of attention-based methods use an attention mechanism and connectionist temporal classification (CTC) for supervised Learning for Automatic Speech Recognition (ASR). In this paper, we propose a speech recognition model using the transformer architecture and achieved the top 3 in 2021 the Vietnamese Language and Speech Processing contest with 8.83% word error rate (WER) on private-test set.

Index Terms— end-to-end speech recognition, transformer, attention mechanism

1 Introduction

Although end-to-end automatic speech recognition (E2E ASR) has achieved great performance in tasks that have numerous paired data, it is still challenging to make E2E ASR robust against noisy and low-resource conditions. First, we proposed 5 methods of augmentation raw audio, that consists of change volume, change speed, change pitch, add background noise (environmental noise, music, human voice, ...) and Gaussian noise. Second, we implement SpecAugment [1] — which applied directly to the feature inputs of a neural network (i.e., filter bank coefficients, log mel spectrogram).

We use Transformer [2] architecture end-to-end model for automatic speech recognition (ASR). Models use hybrid CTC/attention [3] approach this combine Connectionist Temporal Classification (CTC) [4] loss and label smoothing loss to learn alignment the audio signal to linguistic units (character, phoneme or sub word,...)

2 Data

Audio: We use about 400 hours Vietnamese speech dataset with transcript for each audio. Data augmentation is a common strategy adopted to increase the quantity of training data. It is a key ingredient of the state of the art systems for image recognition and speech recognition. With the widespread adoption of neural networks in speech recognition systems which require a large speech database for training such a deep architecture, data augmentation is very useful for small data sets. Indeed, it is possible to augment speech databases and to use the augmented database to achieve improved accuracy.

- ***change speed***: speed up or down according ratio random from 0.8 to 1.2
- ***change volume***: increase or decrease dB audio random from -14 to 8
- ***change pitch***: change pitch according ratio random from -5 to 5
- ***add background noise***: mix noise audio (random from environmental noise, music, ...) with Signal-to-noise ratio (SNR) random from 0 to 15
- ***gaussian noise***: in signal processing, GN called **white noise** - is a random signal having equal intensity at different frequencies.

After augmentation data step, we have total about 4000 hours speech-text data for training Transformer ASR model. In addition, we propose SpecAugment [1], an augmentation method that operates on the mel spectrogram of the input audio, rather than the raw audio itself. This method is simple and computationally cheap to apply, as it directly acts on the mel spectrogram as if it were an image, and does not require any additional data. SpecAugment consists of three kinds of deformations of the log mel spectrogram. The first is time warping, a deformation of the time-series in the time direction. The other two augmentations, inspired by “Cutout”, proposed in computer vision, are time and frequency masking, where we mask a block of consecutive time steps or mel frequency channels.

Text pre-processing: We experiment with subword segmentation approaches that are widely used to address the open vocabulary problem in the context of end-to-end automatic speech recognition. In traditional ASR, these characteristics typically lead to large pronunciation lexicons and high out of vocabulary (OOV) rates. To be able to handle the out-of-vocabulary problem, it has become increasingly common to use a subword-level word representation for the language output sequence. We use 1000 byte-pair encoding (BPE) [8] by toolkit SentencePiece (https://github.com/google/sentencepiece).

3 Joint CTC-Attention Mechanism

Automatic speech recognition (ASR) is the task of transcribing a speech waveform into a text transcript. In a supervised setup, a dataset Dl = (Xi, Yi) N i=1 of N speech X and text Y pairs are provided. Specifically, speech is usually represented as a sequence of feature vectors $X = [x1, xT](Rm)$, where each feature frame x_t is an m-dimensional continuous vector such as mel-spectrogram. Text is usually represented as a sequence of discrete units Y = [y1, ···, yL] B , where B denotes the vocabulary of text sequences (e.g., bpe subword units). The goal is to build a model, typically a probabilistic one such as p(Y—X), to predict the transcription given speech audio. Such models are often classified into two categories: hybrid or end-to-end.

3.1 CTC

As mentioned above, the emission model in the hybrid system can be replaced with a neural network predicting the posterior probability over HMM states for each frame, but it requires a seed HMM-GMM model to derive the forced-alignment. Instead of maximizing the probability of the derived forced alignment to HMM states, CTC parameterizes the distribution over alignment to text sequences directly, and marginalizes over all possible alignments for the target text sequence during training to compute the posterior directly. Formally speaking, for an input speech sequence X of T frames, CTC predicts a distribution over $B^{'} = B \cup \{\epsilon\}$ for each input step, where B is the text alphabet and ϵ is a special blank symbol, representing empty output. The probability of an alignment $A = [a_1, ..., a_T]$ is defined as $\Pi_{t=1}^{T} p_\theta(a_t|X)$. Each alignment is mapped to a text sequence with a function g, which first removes consecutive repeating units and then removes all ϵ. For example, an alignment ”$c\epsilon aa\epsilon abb$” is mapped to a text sequence “caab”. The posterior probability of a text sequence Y given speech X of length T is therefore defined as:

$$p_\theta(Y|X) = \sum_{A:g(A)=Y,A(B')^T} \Pi_{t=1}^{T} p_\theta(a_t|X)$$

and the marginalization on the right hand side can be computed efficiently with dynamic programming. Training of CTC optimizes the likelihood of the posterior distribution:

$$p_\theta = argmax_{p_\theta} \ \Sigma_{i=1}^{N} log p_\theta(Y_i|X_i)$$

and decoding is approximated with finding the most probable alignment and mapping that alignment to a text sequence:

$$Y = g.(argmax_A \ \Pi_{t=1}^{T} \ p(a_t|X)$$

The CTC loss to be minimized is defined as the negative log likelihood of the ground truth label sequence y:

$$L_{CTC} = -lnP(y|x)$$

3.2 Attention

Unlike the CTC approach, the attention model directly predicts each target without requiring intermediate representation or any assumptions, improving CER as compared to CTC when no external language model is used. The model emits each label distribution at t conditioning on previous labels according to the following recursive equations:

$$P(y|x) = \Pi_t^T P(y_t|x, y_{1:t1})$$

The loss function of the attention model is computed as:

$$L_{Attention} = \Sigma_t^T lnP(y_t|x, y_{1:u1})$$

3.3 Hybrid join CTC/attention

Use CTC/attention architecture [3], which utilizes both benefits of CTC and attention during the training and decoding steps in ASR. The proposed training method uses a CTC objective function as an auxiliary task to train the attention model encoder, final objective function to optimize model is defined:

$$L = \lambda L_{CTC} + (1 - \lambda) L_{Attention}$$

with a tunable parameter $\lambda : 0 \leq \lambda \leq 1$ and L_{CTC} and $L_{Attention}$ are loss functions from the CTC and attention.

4 Model

4.1 Transformer

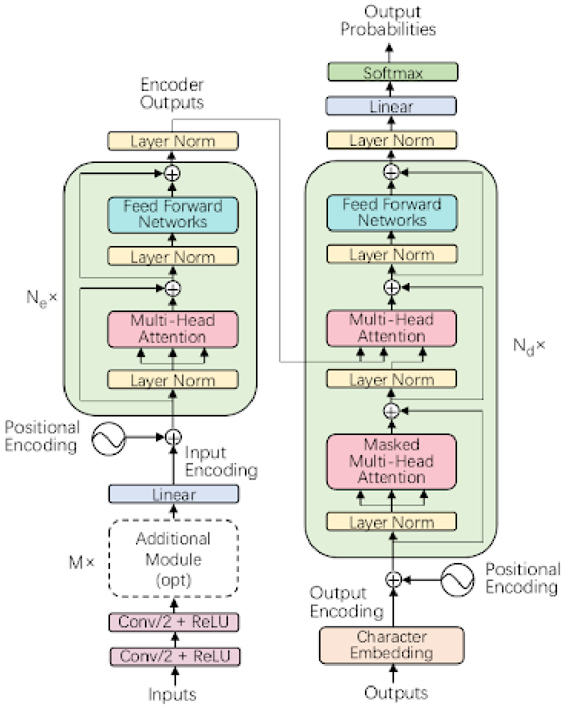
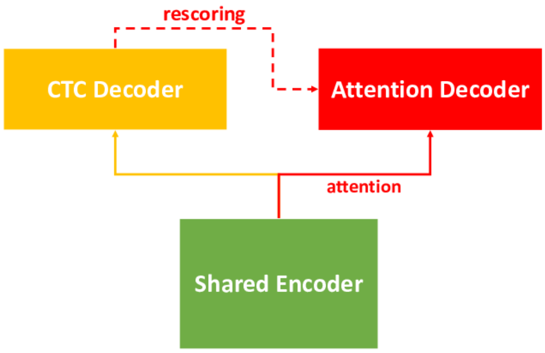


Fig. 2. Model architecture of the Speech-Transformer.

The Transformer architecture appeared in 2017 in the following paper [2] to solve the problem of machine translation. Later on, there was a boom in NLP, transformer architectures evolved, the range of tasks to be solved increased, the results of transformer-based solutions more and more went into the gap. Having taken over NLP, transformers have been introduced into other machine learning areas: speech recognition, speech synthesis, and computer vision, and so on.

The changes in the architecture are minimal — convolutional neural networks (CNN) layers have been added before submitting features to the input to the transformer. This makes it possible to reduce the difference in the dimensions of the input and output sequences (since the number of frames in audio is significantly higher than the number of tokens in the text), which has a beneficial effect on training. The hybrid Connectionist Temporal Classification (CTC)/attention E2E ASR architecture [3] has attracted lots of attention because it combines the advantages of CTC models and attention models. During training, the CTC objective is attached to the attention-based encoder-decoder model as an auxiliary task help the model converge faster.

5 Inference



CTC prefix beam search and decoder rescore: apply CTC prefix beam search on the CTC part of the model with an 5-gram language model is trained on text data collected from news and it is used in fusion with beam search decoding to find the n-best candidates with beam width n and then, use Transformer decoder part to get decoder score for each candidates and this score is combined with the score from the beam search decoding to produce the final score and ranking to give best candidate as transcript for audio input.

5.1 Post processing: text normalize

We use text normalization to convert spoken-domain automatic speech recognition (ASR) output into written-domain text, conversion for acronym cases like *phây búc-facebook, gu gô-google, phô tô-photo, xin x-sin x, cô xin y-cosin i, e n-n, e r-r, bê-b....* With the mapping improve WER about 1-2% compared with no use it.

6 Experiments

In our experiments, we use dataset in *Session 2*, during training, samples in the dataset that are smaller than 0.5 seconds or longer than 20 seconds are filtered out. Use 1000 BPE sub-word units [2] as the output of the model. The performance of the trained model is validated on the validate-set (splited from the training set) and vlsp2021-dev-set. Transformer architecture, the encoder context network for the Transformer model is composed of a convolutional subsampling, positional encoding, followed by a stack of 12 transformer layers with 8 heads. The hidden dimension is 512 and the feed-forward network dimension is 2048. Each transformer layer uses layer dropout with dropout probability 0.1.

Model	dev-set	private-test-set
Transformer	12.0%	8.83%

Table 1: WER(%) on test sets of our experiment

For the final result in **Table 1**, we used CTC beam search with beam width is 100, 5-gram language model with alpha, beta are 0.5, 1.5 and decoder rescore weight is 0.5 to get the best output from the model and finally we apply text normalization to normalize output.

7 References

[1]. Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, Quoc V. Le: *"SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition"*, 2019

[2]. Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, Łukasz and Polosukhin, Illia: *"Attention is all you need"*, 2017

[3]. Watanabe, S.; Hori, T.; Kim, S.; Hershey, J.R.; Hayashi, T: *"Hybrid CTC/Attention Architecture for End-to-End Speech Recognition"*, 2017

[4]. Alex Graves, Santiago Fern ́andez, Faustino Gomez, J ̈urgen Schmidhuber: *"Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks"*, 2006

[6]. Rico Sennrich, Barry Haddow, Alexandra Birch: *"Neural Machine Translation of Rare Words with Subword Units"*, 2016

[7]. L. Dong, S. Xu and B. Xu, "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5884-5888, doi: 10.1109/ICASSP.2018.8462506.