Speech Emotion Recognition via multi-task learning with speaker normalization

Seungwon Ha
Department of Computer Science
202032016 Team 31
Email: hapeter@korea.ac.kr

***Abstract*—With the recent successes of generative models, AI can better produce outputs given our careful prompting. While field studies focusing on prompts can be considered unilateral relationship, Speech Emotion Recognition (SER) is important as to providing the human computer interaction. This paper focuses to apply Speaker Normalization to Multi-Task learning (MTL) framework, which performs Automatic Speech Recognition and SER. Experiments on the IEMOCAP benchmark [1] show that applying Speaker Normalization achieves faster convergence and performance.**

Keywords: Speech emotion recognition, speaker normalization, self-supervised learning. https://github.com/haseungwon/Nlp

I. Introduction

Human emotion acts as an implicit context and acts as an important message along with explicit messages [2]. Understanding such emotional state helps better communication between participants. Speech Emotion Recognition (SER) automatically recognizing human emotion and affective states from speech signals [3], and in this paper we focus on datasets from EIMOCAP as inputs and classify them into 4 emotion class; Happy, Angry, Sad, Neutral.

Attempts on solving SER tasks with deep neural networks via end to end models were made, but some used speech input directly[4] [5] and some used hand-crafted acoustic features such as MFCC and etc [6]. Also different training methods have been applied such as multi-task learning [6], adversarial learning [7],

Also recent researches point out the lack that obtaining large SER datasets is hard and small datasets may contain biases [8].

In this paper we used the basic format of speech emotion recognition with multi-task learning (SER-MTL) model and applied speaker normalization technique and explored how the biases, due to lack of input data size, are affected. Due to the lack of computing resources, this experiment was conducted by comparing one-fold result from the SER-MTL paper.

II. Related Work

A. Multi-task learning

Multi-task learning reflects the learning process of humans as integrating knowledge across domains are what human intelligence conducts. Multi-task learning is a technique that uses the shared model to learn multiple tasks simultaneously. This method has the advantage of increasing data efficiency and performs faster learning speed. However choosing the appropriate optimization loss method is difficult. In this paper we simply used hyperparameter alpha & beta for each separate loss tasks and add them for joint optimization [9].

One research joined Automatic Speech Recongnition task (ASR) with SER [4]. They calculated ASR loss with Connectionist Temporal Classification (CTC), and for SER simple cross-entropy loss. Next constrained the loss of ASR by multiplying hyperparameter α to CTC loss and add with cross-entropy in order to compare the impact of the two losses. In this paper our model is based on [4] with additional downstream task, speaker noramlization.

B. Speaker Normalization

Computer vision domain contains huge and rich data, however when it comes to Speech domain has low resource smaples, and it is hard to obtain large speech dataset [7][8]. These may lead the model to learn shortcuts biases such as individual speakers' own characteristics to predict the affective states.

To addresss these issues, the network should learn the common representations where speaker identities can not be classified. In other words, normalizing speakers' characteristics is important and it is achieved by gradient reversal learning. In this paper use three tasks SER, ASR, and speaker normalization taks. We mostly compare the impact of speaker normalization to SER and ASR tasks.

C. Wav2Vec-2.0

Deep learning research was able to overcome problem of having specific individual networks by replacing them with a common pretrained model, which are task independent at pretraining phase and becomes task specific after fine-tuning.

Pretrained models are self-supervised and learn the good feature representation of the dataset during pretraining. Such self-supervising techniques include contrastive learning, masked-token prediction etc... In this paper we use pretrained Wave2vec-2.0 model. It adapts the self-supervising technique used to train Bert, predicting ground truth speech unit for masked parts of the audio[10]. Fine-tuning with Libris dataset [11], word2vec 2.0 showed good results in word-error-rate (WER). In this paper we use different dataset IEMOCAP and multiple downstream tasks to fine-tune word2vec 2.0; speaker normalization, speech emotion recognition and lastly automatic speech recognition using three different heads.

III. Proposed Method

In this section, we propose simple MTL with speaker normalization framework for SER. As shown in Figure 1, we have single input (raw waveform) fed to pretrained wave2vec 2.0 to extract features. These features get fed to three different heads and produce individual outputs (predicted speaker id, predicted emotion label, predicted characters). Let us denote pretrained wav2vec-2.0 as $f_\theta(\cdot)$. Now let input (raw waveform as $x \in \Re^L$ where L is the length of sample. Next we get the output of pretrained wave2vec-2.0 as $z \in \Re^{L\times d} = f_\theta(x)$, where d is the number of hidden dimension features.

A. Speech Emotion Recognition (SER)

We apply average pooling layer to $z \in \Re^{L\times d}$ into single vector $\hat{z} = \Re^d$. We feed this vector to fully-connected layer, mapping $\hat{z}$ to logits $c_e \in \Re^C$, where C is the number of classes in emotion in our case C is 4; $c_e = g_\Phi(\sum_{i=0}^{i=L-1} f_\theta(x))$. We apply cross-entropy loss for the back-propagation after we transform the logits c_e into probability $\hat{c_e}$.

$$L_{SER} = CrossEntropy(\hat{c_e}), \hat{c_e} = Softmax(c_e) \in \Re^C$$

B. Automatic Speech Recognition (ASR)

We apply simple linear projection from z to y, where $z \in \Re^{L\times d}$ and $y \in \Re^{L\times V}$ (V = 32, size of vocab); $y = h_w(f_\theta(x))$. Also after applying softmax to y, we use Connectionist Temporal Classification loss (CTC) [12]. CTC loss is popular for finding alignment between our predicted sequence to the ground truth sequence. It calculates the loss by summing the probability of possible alignments of input to target. This loss value is differentiable with respect to each input node and can be send for back propagation.

$$L_{ASR} = CTC(\hat{y}), \hat{y} = Softmax(y) \in \Re^{L\times V}$$

C. Speaker Normalization

Same as SER, we apply average pooling to $z \in \Re^{L\times d}$ into single vector $\hat{z} = \Re^d$. Next have a linear projection $\hat{z}$ to $s_{id} \in \Re^S$ where S is the number of speakers; $s_{id} = i_\omega(\sum_{i=0}^{i=L-1} f_\theta(x))$ We introduce gradient reversal layer [13], during forward propagation the layer passes data while reversing the gradient during back propagation. The model aims to maximize the speaker classification loss, and normalize speaker characteristics from the feature representations.

$$L_{sid} = CrossEntropy(\hat{s_{id}}), \hat{s_{id}} = Softmax(s_{id}) \in \Re^S$$

D. Joint Loss

Although there exists various different optimization techniques for calculating the loss [9], we simply use joint loss. It is possible to have various combination of hyperparameter alpha, beta to ASR loss and speaker normalization loss. In this paper we mainly focus the impact of speaker normalization, so we fix $\alpha = 0.1$ which was the best hyper-parameter in [4].

$$L = L_{SER} + \alpha L_{ASR} + \beta L_{sid}$$

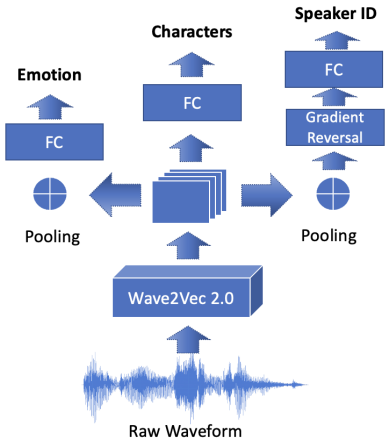

Fig. 1: Overall structure of the proposed framework.

IV. Experiments

In this section, we will discuss the details of our conducted experiment.

A. IEMOCAP Dataset

IEMOCAP dataset is the most widely used in Speech Emotion Recogntion [1]. It contains 5 sessions, each sessions consists of 2 speakers. Many research conducted either 5-fold, 4 sessions for training and 1 session for evaluation, or 10-fold 9 speakers for training and 1 speaker for evaluation. In this paper we used single V100 in colab. Due to the computing resource we couldn't conduct 5-fold or 10-fold, instead we compare only one out of the 10-folds; we trained on 9 speakers and evaluated on 1 speaker in total.

B. Hyperparameters & Evaluation metric

There exists two hyperparameters in calculating joint loss, we have fixed $\alpha = 0.1$ along with other parameters for direct comparison with the research conducted in [4]. We set $\beta = [0, 0.1, 1]$ to measure the performance difference. Details can be found in Table 1. It took a whole day to train on V100.

Although in this experiment we did not conduct 10-folds cross validation, we only conducted only one fold out of ten by evaluating on one speaker and training on other 9 speakers. We can calculate the Weighted Average (WA) by

$$wa = \frac{1}{N^{total}} \sum_{k=1}^{10} N_k^{correct}$$

where $N_k^{correct}$ represents the the number of correctly predicted emotions.In the near future we hope to experiment the whole 10 folds!

C. Experiment Results

In this subsection we show the results of our model compared to the baseline model [4].

We trained on 9 speakers and tested on 01F(session 1 female). Results can be found in table 2, our model has 1 percentage point increase than the original model. This

TABLE I: Hyperparmeters

	Hyperparameters
sample frequency	16k Hz
training epochs	100
optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^8$)
β	0, 0.1, 1
learning rate	5×10^{-5}
warm-up ratio	0.1 , linear warm-up
batch size	8 (accumulated batch size)

TABLE II: Speech emotion recognition (SER) results.

Method	**Accuracy**
SER with MTL [4]	0.746
Ours ($\beta = 1$)	0.751
Ours ($\beta = 0.1$)	0.751

suggests that applying speaker normalization for fine-tuning helps model predict emotion not using speaker specific characteristics. Also from fig. 2 we can find that applying speaker normalization through reverse gradient flow, showed slightly stable convergence earlier than the original model. Also $\beta = 1$ model's wer took longer to drop than the either two cases.

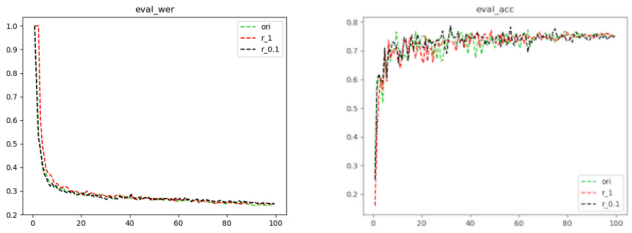

Fig. 2: Word Error Rate (WER) and accuracy of emotion recognition. ori refers to original model [4] Though dramatic difference is not observed, $\beta = 1$ there seems to be less fluctuation in the accuracy and . Also note there are no difference in WER. Also note that our models were trained with 100 Epoch, but the original paper [4] is 200 Epoch, although the paper stated that they experimented with 100... In the github page 200 training epoch was used for due to fluctuation

D. Ablation Study

It was interesting whether the gradient reversal layer had the impact on SER and ASR. In table 3, we measured wer, speaker emotion accuracy and speaker id prediction accuracy. Without gradient reversal layer, the model is trained to learn specific speaker characteristics during training. This is good for learning existing speaker identity prediction. For example, Forward(1) scored lower in speaker accuracy than corresponding Reverse (1). and the accuracy increases as the signal decreases i.e. gradient (β) decreases (Forward (0.1)).

However when it comes to speech emotion recognition, learning specific speaker characteristics during training is a problem. For example, the model may learn that a certain speaker's voice always sounds happy, even if they are actually feeling sad. This can lead to errors in speech emotion recognition. This is shwon in table 3, the speech emotion accuracy is generally lower than gradient reversal.

TABLE III: Ablation study of gradient reversal layer

Case	wer (ASR)	accuracy (SER)	speaker accuracy
Forward (1)	0.243	0.739	0.288
Forward (0.1)	0.239	0.749	0.321
Reverse (1)	0.244	0.751	0.292
Reverse (0.1)	0.247	0.751	0.298

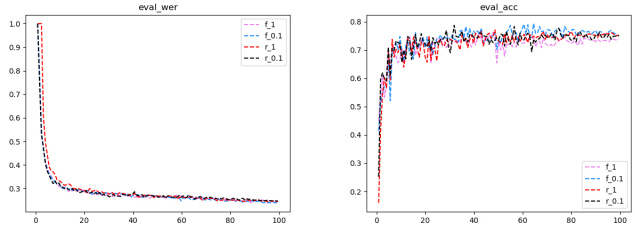

Fig. 3: Word Error Rate (WER) and accuracy of emotion recognition. f_1 means forward gradient flow with $\beta = 1$, $f\_0.1$ means $\beta = 0.1$. r_1 represents reverse gradient flow with $\beta = 1$ and $r\_0.1$ is $\beta = 0.1$. Interesting note applying weak forward gradient flow ($f\_0.1$) showed better performance in wer than reverse gradient flow. Normalizing speaker characteristics seems to undermine the ASR performance while increasing emotion recognition.

V. Conclusion & Analysis

This paper attempts to overcome the short-cut biases that may be due to the lack of decent amount of speech dataset. Although multi-task learning helps model to share imformation between tasks, these models may be prone to short-cuts. So appying additional task, speaker normalization via gradient reversal improved the performance. This hypothesis was tested on IEMOCAP dataset [1] and SER performance indeed increased. This suggests that the model previously attempted to learn individual speakers' characteristics from training and forcefully fit new speaker, which led to over-fitting. Additional ablation study showed that performance of ASR was decreased with speaker normalization, suggesting that there existed some common speaker characteristics that led guidance to better understanding speech recognition and applying speaker normalization undermined those feature characteristics.

Multi-model is a leading research area in many fields, combining the strengths of multiple models to achieve better performance than any single model could achieve on its own. Recent research joins image, video, speech to a common feature space for multi-modeling [14]. Multi-task learning can be used to improve the performance of each task by sharing information between the tasks. Multi-task learning can be approach for multi-models to learn generalized features.

References

[1] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, Dec. 2008.

[2] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[3] B. W. Schuller, "Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends," *Commun. ACM*, vol. 61, no. 5, p. 90–99, apr 2018. [Online]. Available: https://doi.org/10.1145/3129340

[4] S. Ghosh, S. Ramaneswaran, H. Srivastava, and S. Umesh, "Speech emotion recognition using multi-task learning and a multimodal dynamic fusion network," 2022.

[5] C. Etienne, G. Fidanza, A. Petrovskii, L. Devillers, and B. Schmauch, "Cnn+lstm architecture for speech emotion recognition with data augmentation," *Workshop on Speech, Music and Mind (SMM 2018)*, 2018.

[6] J. Ye, X. cheng Wen, Y. Wei, Y. Xu, K. Liu, and H. Shan, "Temporal modeling matters: A novel temporal emotional modeling approach for speech emotion recognition," 2023.

[7] Z. Lian, J. Tao, B. Liu, J. Huang, Z. Yang, and R. Li, "Context-Dependent Domain Adversarial Neural Network for Multimodal Emotion Recognition," in *Proc. Interspeech 2020*, 2020, pp. 394–398.

[8] I. Gat, H. Aronowitz, W. Zhu, E. Morais, and R. Hoory, "Speaker normalization for self-supervised speech emotion recognition," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7342–7346.

[9] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," 2020.

[10] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," 2019.

[11] 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)*, 2015, pp. 5206–5210.

[12] 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*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[13] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," 2016.

[14] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra, "Imagebind: One embedding space to bind them all," 2023.