A GENERAL MULTI-TASK LEARNING FRAMEWORK TO LEVERAGE TEXT DATA FOR SPEECH TO TEXT TASKS

Yun Tang[1], Juan Pino[1], Changhan Wang[1], Xutai Ma[1,2], Dmitriy Genzel[1]

[1]Facebook AI, USA,   [2]Johns Hopkins University, USA
{yuntang,juancarabina,changhan,dgenzel}@fb.com,   xutai_ma@jhu.edu

ABSTRACT

Attention-based sequence-to-sequence modeling provides a powerful and elegant solution for applications that need to map one sequence to a different sequence. Its success heavily relies on the availability of large amounts of training data. This presents a challenge for speech applications where labelled speech data is very expensive to obtain, such as automatic speech recognition (ASR) and speech translation (ST). In this study, we propose a general multi-task learning framework to leverage text data for ASR and ST tasks. Two auxiliary tasks, a denoising autoencoder task and machine translation task, are proposed to be co-trained with ASR and ST tasks respectively. We demonstrate that representing text input as phoneme sequences can reduce the difference between speech and text inputs, and enhance the knowledge transfer from text corpora to the speech to text tasks. Our experiments show that the proposed method achieves a relative 10~15% word error rate reduction on the English LIBRISPEECH task compared with our baseline, and improves the speech translation quality on the MUST-C tasks by 3.6~9.2 BLEU.

Index Terms— Multi-task learning, speech recognition, speech translation

1. INTRODUCTION

Attention-based encoder-decoder modeling is a natural and powerful paradigm for speech to text tasks, such as automatic speech recognition (ASR) and speech translation (ST), and that has led to significant progress [1, 2, 3, 4, 5, 6]. However, it relies on large amounts of supervised speech data, which is expensive to transcribe and translate. In addition, the amount of speech transcripts and speech translation labels is dwarfed by the amount of text data available for language model (LM) and machine translation (MT) training. For example, the number of text tokens used for LM modeling is two orders of magnitude larger than the number of tokens from the corresponding speech corpus in the LIBRISPEECH data corpus [7], as shown in Table 1.

Attention-based encoder-decoder models are not designed to incorporate heterogeneous inputs and cannot benefit from large amounts of low cost text data directly in speech applications. As expected, performance gaps can still be observed between attention based encoder-decoder systems and conventional systems with multiple components [8, 9, 10]. In order to alleviate the data scarcity issue, different approaches have been studied, including acoustic [11, 12] and linguistic aspects [13, 14]. LM is the most commonly used method to integrate linguistic information into ASR. Prior work focuses on building LM with monolingual text data, and then integrate LM or transfer knowledge from it into the decoder [13, 15, 16]. [17] generate synthetic data from text to augment speech training corpus. Another direction is to leverage text data directly during training through multitask learning [18, 19, 20]. [19] use a common representation space to learn correspondences between different modalities for spoken language understanding. [20] propose multi-modal data augmentation to jointly train text and speech for ASR. [4, 21] are focused on ST tasks and trained with an ASR system together, where ASR is used as an auxiliary task. Hence, those methods cannot be applied back to ASR systems.

In this study, we focus on leveraging text data to improve linguistic modeling ability in speech to text tasks. We propose a general framework to leverage text data for ASR and ST tasks. A denoising autoencoder task [22, 23] is introduced to be jointly trained with the ASR task with monolingual data, while a machine translation task is co-trained with ST task with parallel data. Text input is represented as spoken form using phoneme sequence and it effectively reduces the difference between speech input and text input. We also carefully study different design choices for the joint training system, including strategies to share the text and speech encoders and comparing the joint training system with models initialized from pre-trained components. Our experiments show the proposed joint training systems can effectively reduce word error rate (WER) for the ASR task by 10% to 15% and improve BLEU score by 3.6~9.2 for ST tasks.

2. MODEL

The speech and text joint training framework is described in Fig. 1. It consists of three components: speech encoder, text encoder (within the dashed box) and decoder. Two types of
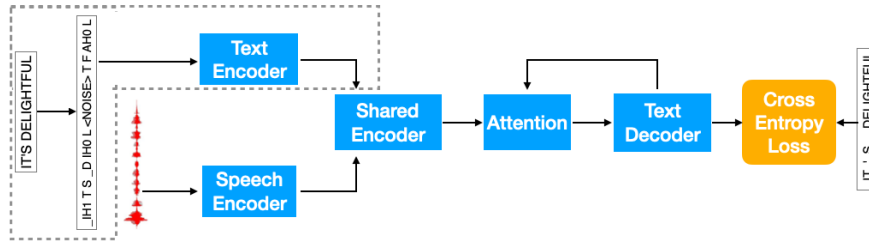
Fig. 1: Joint training framework. Text encoder (within dashed box) is dropped during inference.

input data are fed into the model: speech feature $X = (x_1, x_2 \cdots, x_N), x_n \in \mathcal{R}^{d_s}$ and text form tokens $Y = (y_1, y_2, \cdots, y_M)$. The model output is a token sequence $W = (w_1, w_2, \cdots, w_K)$, where d_s is the speech feature dimension, N, M and K are the number of speech feature frames, number of text form tokens, number of transcription or translation tokens. The text encoder is only used during training and dropped during inference. The training minimizes the loss L from two sub-tasks, e.g., the ST (or ASR) task and the MT (or denoising autoencoder) task, as below :

$$L = -\log P(W|X, \theta^s_{enc}, \theta_{dec}) - \log P(W|Y, \theta^t_{enc}, \theta_{dec}) \quad (1)$$

where $\theta^s_{enc}, \theta^t_{enc}$ and θ_{dec} are parameters for speech encoder, text encoder and decoder respectively.

2.1. Transformer Encoders and Decoder

Two encoders and decoder are all transformer based models [24]. Following [25], the speech encoder has an extra sub-sampling module, which consists of two 1-dimensional convolutional layers and each is with kernel width 3 and stride 2. The subsampling module outputs are combined with sinusoidal positional embeddings and fed into the following transformer layers. The text encoder and decoder consist of standard transformer layers. We use pre-layer normalization for more stable training [26].

2.2. Input Text Representation

The input text to the joint training model comes from two sources: the transcripts from the speech training corpora, and text data from non-speech corpora. The text input is presented as phoneme sequence instead of word tokens or subword tokens [20]. Phoneme sequence is the spoken form representation of the original text and is easier to be mapped to the corresponding speech input. The conversion from text to phoneme representation can be done through a grapheme to phoneme system or simply a dictionary lookup. The phoneme set used in this study is based on the Carnegie Mellon Pronouncing Dictionary with 39 phonemes. Vowels carry a lexical stress marker. We further extend the phoneme set by identifying the first phoneme in the word with additional "_" mark, which is similar to the notation in sentence piece process.

2.3. Auxiliary Text Tasks

The MT task is chosen as an auxiliary task to be jointly trained with the ST task. The input tokens are phoneme sequences converted from the corresponding source text and the target is represented as subword units derived from the corresponding translation labels.

In the ASR task, the auxiliary task maps the input phoneme sequence to the corresponding subword tokens derived from transcripts. However, the task is too simple since 76% of words in the Carnegie Mellon Pronouncing Dictionary can be recovered from a phoneme sequence deterministically. It brings little help to the ASR task as shown in the experiments in section 4. Hence, we modify this task to a denoising autoencoder task [22, 23] and part of the input phoneme tokens are masked as "⟨NOISE⟩". The decoder has to infer the target word based on the phonemes of this word as well as neighbouring context. Hence, the monolingual text data can be integrated into sequence to sequence ASR modeling naturally and effectively.

Fig. 1 shows the ASR input/output. The input text is "It's delightful" and the corresponding target subword sequence encoded with SentencePiece is "_IT ' S _DELIGHTFUL". The phoneme sequence corresponding to input text is "_IH1 T S _D IH0 L AY1 T F AH0 L". Phoneme "AY1" is randomly selected and masked with "⟨NOISE⟩" token, and the phoneme sequence fed to the model becomes "_IH1 T S _D IH0 L ⟨NOISE⟩ T F AH0 L".

3. EXPERIMENTAL SETTINGS

3.1. Data

We conduct experiments on two datasets, LIBRISPEECH [7] and MuST-C [27]. The detailed training data statistics are presented in Table 1. The second column is the total number of hours for the speech training data. The third and fifth columns are the number of (source) words.

ASR datasets: ASR is evaluated on the LIBRISPEECH dataset and the co-training text data is the language model training data coming with the LIBRISPEECH dataset.

ST datasets: ST is evaluated on three language pairs: English-German (EN-DE), English-Spanish (EN-ES) and

Speech	hours	#W(m)	Text	#W(m)
LibriSpeech	960	9.4	Gutenberg	803.3
MuST-C				
EN-DE	408	4.2	WMT17	105.2
EN-ES	504	5.2	WMT13	369.4
EN-FR	492	5.1	WMT14	1,018.5

Table 1: Data statistics for ST and ASR training sets. "#W(m)" stands for "number of words (million)".

Data set	#params (m)	Dev clean	Dev other	Test clean	Test other
LAS [12]	360	-	-	2.8	6.8
Transformer [25]	270	2.5	6.7	2.9	7.0
Transformer (M)	76	3.5	8.1	3.7	8.1
Joint Training (M)	76	3.0	7.4	3.3	7.6
Transformer (L)	161	3.3	7.9	3.6	8.0
Joint Training (L)	161	2.8	7.0	3.1	7.2

Table 2: WER results on Librispeech

Data corpus	#params(m)	EN-DE	EN-ES	EN-FR
Transformer [29]	30	17.7	20.9	26.5
Transformer [5]	-	22.9	28.0	32.7
Transformer [30]	435	25.2	-	34.5
Transformer	31	20.3	19.4	25.3
Joint Training	31	23.9	28.6	33.1

Table 3: BLEU results of three language pairs on the MuST-C tst-COMMON.

Data set	Librispeech Dev (WER) clean	Librispeech Dev (WER) other	MuST-C EN-DE (BLEU) tst-COMMON
None	3.5	8.1	20.3
Character	3.8	8.6	23.1
Phoneme	3.0	7.4	23.9

Table 4: Comparison of the input text representations.

English-French (EN-FR) on the tst-COMMON test set. WMT parallel data is used as text training corpus. Case-sensitive detokenized BLEU is reported by SACREBLEU.

Target subword units are learned from SentencePiece with vocabulary size 10k and full character coverage on all training text data. The grapheme to phoneme conversion for the input text is done through the "g2p_en" Python package [28]. The input phoneme vocabulary size is 134. Input speech is represented as 80D log mel-filterbank coefficients computed every 10ms with a 25ms window. Global channel mean and variance normalization is applied to the input speech features. SpecAugment [12] is employed to augment audio data for model training. LD and LB policies from [12] are applied to LIBRISPEECH and MuST-C tasks respectively.

3.2. Model Setup

In all experiments, the speech encoder has 12 transformer layers while the text encoder and decoder have 6 transformer layers. Three model configurations are examined: small, medium and large. The small configuration has a word embedding size of 256 and transformer middle layer dimension 2048; medium configuration has word embedding size equal to 512 and the middle layer dimension 2048; the large configuration sets word embedding size to 768 and middle layer dimension to 3072. The MuST-C models are using the small configuration, while the medium and large configurations are used for the LIBRISPEECH models. If not specifically mentioned, the medium configuration is used by default in LIBRISPEECH experiments. We use the Adam optimizer with a learning rate 0.001 in all experiments. Label smoothing and dropout rate are both set to 0.1. The LIBRISPEECH models

are trained with 240 epochs using 16 GPUs and the MuST-C models are trained with 200 epochs using 8 GPUs. The batch size is 40,000 frames for speech samples and 20,000 tokens for text samples per GPU. Speech input and text input are used to update the model alternatively. The models are trained with Fairseq [6]. The last 10 checkpoints are averaged for inference with beam size 5.

4. EXPERIMENTAL RESULTS

We present our main ASR results in Table 2. The top of the table shows results from literature. "Transformer" and "Joint Training" are results from the baseline and the corresponding jointly trained system. In the jointly trained system, 20% of input tokens are masked. Two configurations: medium (row 3-4) and large (row 5-6), are studied. For both configurations, the multi-task system outperforms the baseline system at all 4 test sets. The large configuration achieves the best results with relative WER reductions varied from 10% to 15%.

Table 3 demonstrates the main results on three ST tasks. The models are trained with the small configuration. The results from baseline (row 4) are comparable to results in [29], though all baselines are trained from scratch with random initialization. Row 5 demonstrates results from the jointly trained models with extra WMT parallel text. The performance is improved significantly that 3.6 to 9.2 BLEU score increases are observed in different language pairs.

Impact of Input Token Representations In Table 4, we compared the phoneme representation with character based representation on both ASR and ST tasks. "None" represents results from a system without text input. Results from the character based text representation are listed in row "Character". For the ASR task, the WERs are even higher than results from the baseline system trained with speech data only.

Data set	Librispeech Dev (WER)		MuST-C EN-DE (BLEU)
	clean	other	tst-COMMON
No Share	3.0	7.4	22.7
Share	3.0	7.4	23.9

Table 5: Comparison of encoder parameter sharing strategies

Data set	Librispeech Dev (WER)		MuST-C EN-DE (BLEU)
Masking ratio	clean	other	tst-COMMON
0.0	3.3	8.2	23.9
0.1	2.9	7.6	23.8
0.2	3.0	7.4	23.2
0.3	3.0	7.8	23.1

Table 6: Comparison of different masking ratios.

Though the ST result using character representation is better than the baseline, it is 0.8 BLEU lower than the phoneme representation based system. It is clear that organizing the input text in spoken form is critical for the speech-text multi-task learning, and it makes the knowledge transfer from text to speech more effective.

Encoder Parameter Sharing Strategies Speech to text tasks obtain linguistic information from text corpora via a shared decoder. It is natural to ask if sharing parameters between two encoders would improve performance further. We compare systems with shared encoder ("Share") and dedicated encoder ("No Share") in Table 5. Shared encoder means parameters in all 6 transformer layers in the text encoder are shared with the last 6 layers of the speech encoder. Sharing parameters between encoders is helpful for the MUST-C task. 1.2 BLEU decrease is observed without parameter sharing between two encoders. On the other hand, the ASR task is not sensitive to the parameter sharing that different strategies give comparable results.

Impact of Masking In Table 6, different input text masking ratios are examined for both ASR and ST tasks. Without masking (row 1), the jointly trained ASR model performs no better than the Transformer baseline (as shown in Table 4 row "None"). The best WER is achieved when 20% input tokens are masked. It confirms our assumption that mapping from phoneme sequence to the corresponding (sub)word sequence is too simple that the model can accomplish it without learning much of the language context. Masking some phonemes forces the model to learn better language context representation. For the ST task, masking degrades speech translation performance. We hypothesis that the English to German translation task is challenging, and the decoder needs to understand language context to associate the input phoneme sequence with the target subword sequence.

Comparison with Pre-Trained Models Compared to multi-

model	Encoder	Decoder	Joint Training	BLEU
I	✗	✗	✗	20.3
II	✓	✗	✗	22.7
III	✗	✓	✗	22.3
IV	✓	✓	✗	22.9
V	✗	✗	✓	23.9
VI	✓	✓	✓	24.0

Table 7: Comparison of models initialized with pre-trained model on the MuST-C EN-DE task.

task learning, another important direction is to learn different tasks individually, and then finetune the pre-trained model for downstream tasks. We compare these two approaches in Table 7. Two pre-trained components are used. The speech encoder is pre-trained on the ASR task and the decoder through a MT task. In row "II" and "III", ST models are initialized with pre-trained encoder and decoder respectively. In both cases, more than 2 BLEU score improvements are observed. Combining both pre-trained models, another 0.2 BLEU score gain is achieved in row "IV" compared with results in row "II". Row "V" is the result from the jointly trained model, which outperforms the ST model with both encoder and decoder initialized from pre-trained models by another 1.0 BLEU. In row "VI", the pre-trained model is finetuned using joint training, it is slightly better than the model trained from scratch in row "V". It shows that the jointly trained model could achieve good results even without pre-trained components.

5. CONCLUSION

In this study, we propose a general multi-task learning framework to leverage text data for ASR and ST tasks. The ASR task is co-trained with a denoising autoencoder task using monolingual text, while a MT task is jointly trained with the ST task with parallel data. Text input is represented as phoneme sequences to reduce the difference between speech input and text input. We examined different factors that impact the performance of the jointly trained system. Our experimental results show substantial WER reduction is achieved on the LIBRISPEECH dataset and large BLEU score gain is obtained in the MUST-C datasets.

6. REFERENCES

[1] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NIPS*, 2015.

[2] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[3] A. Berard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *NIPS*, 2016.

[4] R. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," in *INTERSPEECH*, 2017.

[5] H. Inaguma, S. Kiyono, K. Duh, S. Karita, N. Soplin, T. Hayashi, and S. Watanabe, "Espnet-st: All-in-one speech translation toolkit," in *ACL*, 2020.

[6] C. Wang, Y. Tang, X. Ma, A. Wu, D. Okhonko, and J. Pino, "fairseq s2t: Fast speech-to-text modeling with fairseq," in *AACL*, 2020.

[7] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[8] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "RWTH ASR systems for librispeech: Hybrid vs attention - w/o data augmentation," in *INTERSPEECH*, 2019.

[9] J. Niehues, R. Cattoni, S. Stüker, M. Negri, M. Turchi, E. Salesky, R. Sanabria, L. Barrault, L. Specia, and M. Federico, "The IWSLT 2019 evaluation campaign," 2019.

[10] Elizabeth Salesky and Alan W Black, "Phone features improve speech translation," in *ACL*, 2020.

[11] T. Ko, V. Peddinti, Daniel P., M. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," *ICASSP*, 2017.

[12] D. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. Cubuk, and Q. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Interspeech*, 2019.

[13] Jan Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *INTERSPEECH*, 2016.

[14] A. Rosenberg, Y. Zhang, B. Ramabhadran, Y. Jia, P. Moreno, Y. Wu, and Z. Wu, "Speech recognition with augmented synthesized speech," in *ASRU*, 2019.

[15] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," in *INTERSPEECH*, 2017.

[16] Y. Bai, J. Yi, J. Tao, Z. Tian, and Z. Wen, "Learn spelling from teachers: Transferring knowledge from language models to sequence-to-sequence speech recognition," *Interspeech*, 2019.

[17] Y. Jia, M. Johnson, W. Macherey, R. Weiss, Y. Cao, C. Chiu, N. Ari, S. Laurenzo, and Y. Wu, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," *ICASSP*, 2018.

[18] P. Bahar, T. Bieschke, and H. Ney, "A comparative study on end-to-end speech to text translation," in *ASRU*, 2019.

[19] Y. Huang, H. Kuo, S. Thomas, Z. Kons, K. Audhkhasi, B. Kingsbury, R. Hoory, and M. Picheny, "Leveraging unpaired text data for training end-to-end speech-to-intent systems," *ICASSP*, 2020.

[20] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multi-modal data augmentation for end-to-end asr," in *INTERSPEECH*, 2018.

[21] Antonios Anastasopoulos and David Chiang, "Tied multitask learning for neural speech translation," in *NAACL-HLT*, 2018.

[22] G. Lample, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," in *ICLR*, 2018.

[23] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *ACL*, 2020.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[25] G. Synnaeve, Q. Xu, J. Kahn, E. Grave, T. Likhomanenko, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," 2020.

[26] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," 2019.

[27] M. Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "MuST-C: a multilingual speech translation corpus," in *NAACL-HLT*, 2019.

[28] Y. Lee and T. Kim, "Learning pronunciation from a foreign language in speech synthesis networks," *ArXiv*, 2018.

[29] Mattia Antonino Di Gangi, Matteo Negri, and Marco Turchi, "One-to-many multilingual end-to-end speech translation," in *ASRU*, 2019.

[30] J. Pino, Q. Xu, X. Ma, M. Dousti, and Y. Tang, "Self-training for end-to-end speech translation," in *Interspeech*, 2020.