Multi-Encoder Learning and Stream Fusion for Transformer-Based End-to-End Automatic Speech Recognition

Timo Lohrenz, Zhengyang Li, Tim Fingscheidt

Technische Universität Braunschweig
Institute for Communications Technology
Schleinitzstr. 22, 38106 Braunschweig, Germany

t.lohrenz@tu-bs.de, zhengyang.li@tu-bs.de, t.fingscheidt@tu-bs.de

Abstract

Stream fusion, also known as system combination, is a common technique in automatic speech recognition for traditional hybrid hidden Markov model approaches, yet mostly unexplored for modern deep neural network end-to-end model architectures. Here, we investigate various fusion techniques for the all-attention-based encoder-decoder architecture known as the transformer, striving to achieve optimal fusion by investigating different fusion levels in an example single-microphone setting with fusion of standard magnitude and phase features. We introduce a novel multi-encoder learning method that performs a weighted combination of two encoder-decoder multi-head attention outputs *only* during training. Employing then only the magnitude feature encoder in inference, we are able to show consistent improvement on Wall Street Journal (WSJ) with language model and on Librispeech, without increase in runtime or parameters. Combining two such multi-encoder trained models by a simple late fusion in inference, we achieve state-of-the-art performance for transformer-based models on WSJ with a significant WER reduction of 19% relative compared to the current benchmark approach.

Index Terms: End-to-end speech recognition, information fusion, multi-encoder learning, transformer, phase features

1. Introduction

In recent years a paradigm shift in automatic speech recognition (ASR) research is seen towards the replacement of established hybrid hidden Markov model (HMM) based approaches [1] by end-to-end trained neural networks making pronunciation dictionaries and phonetic modeling techniques obsolete [2]. Proposed methods for end-to-end training include connectionist temporal classification (CTC) [3], recurrent neural network transducers (RNN-T) [4] and recent attention-based encoder-decoder (AED) models, namely the listen-attend-and-spell (LAS) [5] and the transformer model [6]. While the LAS models employ recurrent connections in the typical encoder-decoder structure of the end-to-end models, transformer models rely entirely on the attention mechanism to capture temporarily relevant information in speech [7]. On large datasets such as Librispeech [8], transformer models outperform hybrid speech recognition already by a large margin [9].

For the well-established hybrid speech recognition stream fusion approaches can be classified into three categories based on which stage in the system fusion is performed: early fusion—combination in the input feature domain [10, 11, 12], middle fusion—combination of an intermediate information representation [13, 14, 15] (e.g., state likelihoods), or late fusion—combination of system outputs (e.g., word hypotheses [16], output posteriors, confusion networks [17], or lat-

tices [18]). A prominent task for fusion is audiovisual automatic speech recognition (AV-ASR) [19, 20], employing additional visual sensors to increase robustness in noisy conditions. In single-channel settings, usually different feature representations are used for fusion (e.g., filterbank and fMLLR features [12], different short-time Fourier transform window sizes [15], or as here, standard magnitude features with phase features [21]), or multiple acoustic models [22, 23].

Concerning the fusion of additional information into end-to-end transformer models, the few existing approaches stem from audiovisual automatic speech recognition [24, 25] and neural machine translation [26], where additional encoders are used to gather visual speech information or contextual information, respectively. Recent successful non-fusion techniques for end-to-end models are multi-task learning, e.g., by using a combination of CTC and attention-based losses [27, 28], and augmentation techniques such as spectral augmentation [29]. Those methods improve neural networks by adding more variety to the trained models either by composite losses or by randomly withholding information in the input features. An unexplored approach to add such variety, strongly related to fusion, is to use multiple encoders during training of the transformer model. Some context-aware approaches in neural machine translation use additional context encoders [30] to incorporate previous context of a sentence to achieve a better translation, while in [26] it has been found that results improve even if such context is ignored during inference.

In this paper we adopt and modify fusion techniques from hybrid ASR to transformer-based end-to-end speech recognition on an exemplary audio-only fusion task by combining the common magnitude-based feature representations with additional phase-based features. To elaborate the best possible fusion method we apply simple input feature and output posterior combination methods as well as middle fusion schemes, that use two different encoders to perform fusion of the respective encoder-decoder multi-head attention outputs. For this middle fusion approach we investigate several variants comprising an optional sharing of the encoder-decoder attention parameters as well as different paradigms combining the outputs thereof. In addition we explore a novel method which we dub *multi-encoder learning* (MEL) that uses both individual encoders *only* during training, thereby increasing robustness of the standard non-fusion transformer even during single encoder inference.

The paper is structured as follows: In Section 2, we introduce known and novel fusion and learning approaches to end-to-end model architectures. Section 3 describes the fusion experiment setup on Wall Street Journal (WSJ) and Librispeech, while corresponding results are reported and discussed in Section 4. The paper is concluded in Section 5.
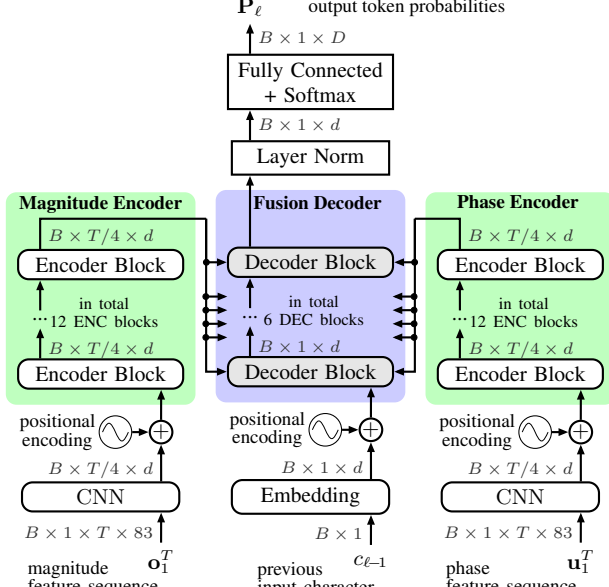
Figure 1: *Transformer architecture for the middle fusion approaches (Fusion-Mid) for training and inference, and for multi-encoder learning (MEL) during training. Two individual stream encoders are employed for each feature sequence; details of the decoder block are shown in Figure 2.*

2. Fusion Methods for End-to-End ASR

2.1. Early Fusion

When it comes to fusion in end-to-end systems, the simplest approach is *early fusion* as it is often applied in hybrid systems by stacking the individual feature vectors $\mathbf{o}_t$ and $\mathbf{u}_t$ to a joint feature representation $\mathbf{x}_t = (\mathbf{o}_t^\mathsf{T}, \mathbf{u}_t^\mathsf{T})^\mathsf{T}$, with $(\ )^\mathsf{T}$ being the transposed. When using filterbank features, it has become a common technique to use convolutional neural networks (CNNs) [31] in the input layer. In our Fusion-Early approach the additional feature stream is treated as an additional input channel, yielding an input tensor to the CNN block of size $B \times (2 \cdot C) \times T \times F$ with B, C, T, F being the batch size, channel depth, feature sequence length, and feature dimension, respectively. After the input layer, the processing follows the standard transformer model architecture using a single attention-based encoder and a single decoder as in [6].

2.2. Middle Fusion

For the *middle fusion* approaches we use two individual stream encoders for each feature sequence as shown as the green boxes in Figure 1. Based on the previous output token $c_{\ell-1} \in \mathcal{C} = \{c^{(1)}, c^{(2)}, \ldots, c^{(D)}\}$ and the entire feature sequences $\mathbf{o}_1^T$ and $\mathbf{u}_1^T$, where $t \in \{1, \ldots, T\}$ and $\ell \in \{1, \ldots, L\}$ are time instants of the input feature vector and output token sequences, respectively, the transformer outputs a vector $\mathbf{P}_\ell$ with output token probabilities for the current sequence time instant ℓ. Each of the stream encoders comprises a total of 12 identical encoder blocks, each consisting of the multi-head self-attention mechanism and position-wise fully connected layers as in [6]. The output of the last encoder block is then passed on to each of the in total 6 decoder blocks, which are detailed in Figure 2. We investigate two different strategies for middle fusion, both using two separate encoder-decoder multi-head attention blocks (shown in yellow) for each stream encoder. Fusion is then
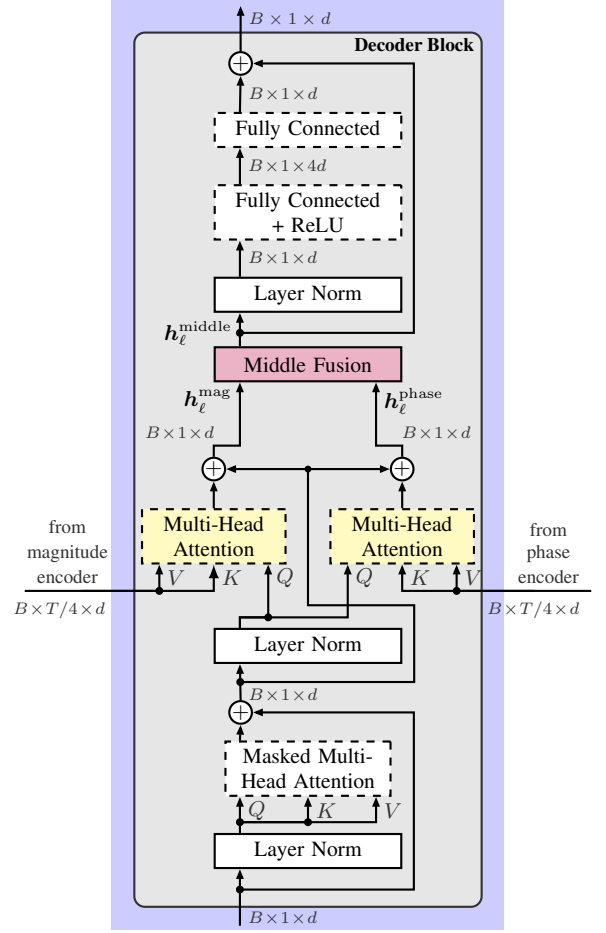

Figure 2: *Single decoder block (cf. Figure 1) for the middle fusion approach of both encoder-decoder multi-head attention block outputs. The same setup is used for the Multi-Encoder Learning approaches (MEL-t-mag and MEL-t-phase) only during training, while only one encoder is active in inference. Dropout layers [32] are in dashed line boxes.*

applied to the hidden entities $\boldsymbol{h}_\ell^{\mathrm{mag}}$ and $\boldsymbol{h}_\ell^{\mathrm{phase}}$ after the two encoder-decoder multi-head attention blocks for each stream, shown as red block in Figure 2, yielding $\boldsymbol{h}_\ell^{\mathrm{middle}}$. First is the weighted sum approach, dubbed Fusion-Mid-WS, employing a simple linear combination

$$\mathbf{h}_\ell^{\mathrm{middle}} = \alpha \boldsymbol{h}_\ell^{\mathrm{mag}} + (1 - \alpha)\boldsymbol{h}_\ell^{\mathrm{phase}} \tag{1}$$

with $\alpha \in [0, 1]$ being a fusion weight to balance the influence of each of the encoder-decoder multi-head attention blocks. In addition, for the Fusion-t-Mid-WS approach, we tied ("-t-") the parameters of both involved encoder-decoder multi-head attention blocks.

The second variant dubbed Fusion-Mid-CC is the straightforward concatenation of both entities according to $\mathbf{h}_\ell^{\mathrm{middle}} = ((\boldsymbol{h}_\ell^{\mathrm{mag}})^\mathsf{T}, (\boldsymbol{h}_\ell^{\mathrm{phase}})^\mathsf{T})^\mathsf{T}$ as it has been used for audiovisual speech recognition in [25]. To still allow residual connections, in this case it becomes necessary to halve the dimension of both encoder-decoder multi-head attention block outputs to $d/2$ and add the residual from the self-attention after the concatenation, where the previous model dimension is restored.

2.3. Late Fusion

As *late fusion* we investigate the fusion of output token probability vectors $\mathbf{P}_\ell^{\text{mag}}$ and $\mathbf{P}_\ell^{\text{phase}}$ stemming from separately trained transformer networks for each feature stream $\mathbf{o}_1^T$ and $\mathbf{u}_1^T$. The final output token probability in the log domain for each time instant ℓ is then computed as (Fusion-Late)

$$\log \mathbf{P}_\ell^{\text{late}} = \beta \log \mathbf{P}_\ell^{\text{mag}} + (1 - \beta) \log \mathbf{P}_\ell^{\text{phase}} \qquad (2)$$

with $\beta \in [0, 1]$ being a posterior fusion weight and $\log(\ )$ operating element-wise. One major advantage of the late fusion approach is that it uses independently trained models, and the balancing hyperparameter β can be easily set during inference time if one feature stream deteriorates.

2.4. Novel Multi-Encoder Learning (MEL)

In addition to the previous fusion paradigms, we employ a novel yet simple multi-encoder learning (MEL) approach to investigate if the additional information during training helps to increase robustness without using any additional parameters in inference. For this method, we train the middle fusion transformer model exactly as for the Fusion-t-Mid-WS approach using both encoders, but tie ("-t-") all parameters of both multi-head attention blocks (shown as yellow blocks in Figure 1). During inference, however, only *one* of the encoders is active and the decoder uses one instance of the jointly trained multi-head attention. For the MEL-t-mag and MEL-t-phase approaches, only the magnitude or the phase encoder is active during inference, respectively, while the fusion weight $\alpha = 0.9$ *during training* is biased towards the inference encoder. Both models trained with the MEL method can also be subject to late fusion, dubbed MEL-t-Fusion-Late in Tables 1 and 3.

2.5. Language Model and Decoding

For all investigated approaches including non-fusion baselines Baseline-mag and Baseline-phase, we use beam-search decoding during inference and slightly deviate from the standard transformer architecture in [6] by using layer normalization before each attention or stack of fully connected layers according to the implementation in [33]. During decoding, the final output $\mathbf{P}_\ell^{\text{final}}$ of *all* approaches can optionally be computed as

$$\log \mathbf{P}_\ell^{\text{final}} = \log \mathbf{P}_\ell + \lambda \log \mathbf{P}_\ell^{\text{LM}}, \qquad (3)$$

adding logarithmic character probabilities from the language model $\mathbf{P}_\ell^{\text{LM}}$, with the standard language model weight λ chosen according to the shallow integration technique [34]. For experiments on the Wall Street Journal task we report all results both without and with additional language model in Table 1.

3. Experimental Setup

3.1. Databases

We evaluate our fusion approaches on the 81-hour Wall Street Journal (WSJ) dataset [35] using the `dev93` and `eval92` splits to evaluate system performance in terms of word error rate WER $= 1 - \frac{N-D-I-S}{N}$, as well as w.r.t. character error rate (CER), where the number of units N, deletions D, insertions I, and substitutions S are calculated on character-level instead of on word-level as for the WER. To investigate our approaches also on a large-scale dataset, all experiments are repeated on Librispeech [8] using the 960 h training set along with the `clean` and `other` portions of the dev and test datasets. All used speech signals are sampled at 16 kHz and analyzed with a 25 ms window and a frame shift of 10 ms.

3.2. Acoustic Frontends

For the middle fusion approaches, each of the encoders receives a sequence of T feature vectors of dimension $F = 83$. As magnitude features $\mathbf{o}_1^T$ we use standard 80-dimensional filterbank features extended with 3-dimensional pitch features extracted with the `Kaldi` toolkit [36]. For the phase features $\mathbf{u}_1^T$ we follow the processing of [37, 38] and use the group-delay representation extracted from an all-pole model, and also apply an 80-dimensional mel-filterbank. For details on the processing, please refer to [38]. The convolutional neural networks (CNNs) at the input layers, shown as CNN blocks in Figure 1, consist of a total of four convolutional layers each using 3×3 filter kernels. The second and forth convolutional layer use a stride of 2 in both temporal and frequency direction thus compressing the input sequence length to $T/4$. We note that it might be beneficial to apply separate convolutions to the pitch features but follow [39, 28] for comparability.

3.3. Acoustic and Language Model Configuration

As shown in Figure 1, the used transformer architecture for the acoustic model follows the standard architecture from [6] employing a total of 12 encoder blocks for each used encoder, while the decoder stacks 6 decoder blocks. For WSJ, the model dimension is set to $d = 256$ and multi-head attention blocks use 4 attention heads, while for Librispeech we use a larger model, where both values are doubled to $d = 512$ and 8 attention heads. Transformer models were trained using the Adam optimizer with label-smoothed cross-entropy loss [40]. We follow [7] for learning rate scheduling. For Librispeech experiments we additionally used spectral augmentation [29].

For language modeling in WSJ experiments, we apply a 3-layer LSTM network with a size of 1200 each, which is trained on word-level but yields character-level probabilities $\mathbf{P}_\ell^{\text{LM}}$ for a total of $D = 52$ characters, using the lookahead method proposed in [41]. As language model weight we follow [39] and choose $\lambda = 0.9$. For Librispeech we use `SentencePiece` for word tokenization with an output token dimension size of $D = 5000$ embeddings [42] and use a 4-layer LSTM as token-based LM with each layer having a size of 1024. The language model weight for Librispeech is set to $\lambda = 0.4$ following [39].

For the Fusion-Late approach, the posterior fusion weight β (only applied during inference) has been optimized on the respective development sets. For all middle fusion approaches with weighted sum we set the fusion weight $\alpha = 0.9$ without further tuning. The same value was used for the MEL approaches during training for the respective primary encoder (magnitude encoder for MEL-t-mag and phase encoder for MEL-t-phase).

All models were trained using the `espresso` and `fairseq` toolkits based on `PyTorch` [39, 43, 44]. WSJ models were trained on a single GTX1080Ti GPU, while Librispeech models used 4 Tesla P100 GPUs. All experiments use the same random seed.

4. Recognition Results and Discussion

Results of all approaches on the WSJ task are shown in Table 1. For the single-encoder approaches, we note that the Baseline-mag transformer performs slightly better than the Baseline-phase approach (about 1% absolute in terms of WER on the `eval92` set with language model). Comparing results *without* and *with* language model (LM) for both Baseline approaches, we note that with LM the error rates are significantly reduced (especially for word errors), showing the effectiveness of the word-based lookahead LM [41].

Table 1: *Transformer-based approaches on the WSJ task. Best results are bold, second best results are underlined.*

Approach	Inference complexity		*Without* language model				*With* language model			
	# of parameters	relative runtime	dev93		eval92		dev93		eval92	
			WER	CER	WER	CER	WER	CER	WER	CER
Baseline-mag	16.8M	1.0	14.62	5.28	11.66	4.03	6.51	3.64	4.43	2.37
Baseline-phase			15.75	5.79	12.90	4.33	7.32	4.23	5.48	3.17
Fusion-Early	16.8M	1.02	14.46	5.07	10.83	3.70	6.57	3.39	4.41	2.27
Fusion-Mid-CC	28.4M	1.36	17.11	6.17	11.31	3.81	6.58	3.71	4.20	2.23
Fusion-Mid-WS	28.8M	1.37	16.82	5.75	13.43	4.11	6.40	3.55	4.38	2.46
Fusion-t-Mid-WS	27.2M	1.33	15.89	5.58	12.28	4.07	6.23	3.57	_4.09_	_2.10_
Fusion-Late	33.5M	1.81	**13.38**	**4.79**	_10.65_	_3.53_	_5.79_	3.08	4.31	2.32
MEL-t-mag	16.8M	1.0	15.22	5.36	11.73	3.95	6.29	3.43	4.31	2.50
MEL-t-phase	16.8M		16.01	5.85	12.09	4.25	7.00	4.04	4.68	2.74
MEL-t-Fusion-Late	33.5M	1.81	_14.04_	_4.90_	**10.12**	**3.38**	**5.50**	**3.01**	**3.40**	**1.95**

Table 2: *WER comparison of recent transformer-based end-to-end ASR approaches with language model on WSJ.*

Approach	dev93	eval92
Tsunoo et al. 2019 [45]	-	5.00
Karita et al. 2019 [46]	7.70	4.50
Karita et al. 2019 [9]	6.80	4.40
Moriya et al. 2020 [28]	6.90	4.20
Ours	**5.50**	**3.40**

Considering all Fusion-X approaches, an overall improvement compared to Baseline-Y methods is visible in most cases, suggesting that the phase-based speech representation indeed yields complementary information that can improve recognition. Interestingly, none of the Fusion-Mid approaches is able to provide consistent improvements. While only slightly increasing the size and complexity of the transformer, the common Fusion-Early approach is not able to decrease the WER with language model on the dev93 set. Among the Fusion-Mid approaches with LM, the Fusion-t-Mid-WS variant performs best by achieving 4.09% WER on eval92 while also adding the least complexity to the model during inference. The fully modular Fusion-Late approach yields the highest computational complexity as both baseline transformers have to be inferred. On the other hand, it offers an inference-time parameter β to balance the fusion and performs remarkably well *without* LM. *With* LM, however, Fusion-Late is not able to generalize the superior dev93 performance towards the eval92 set.

Our novel MEL-t-mag and MEL-t-phase approaches with LM achieve a WER reduction on eval92 of 0.12% and even 0.8% *absolute* compared to their respective Baseline approaches without adding any additional complexity during inference. This suggests that the parameter tying of both encoder-decoder multi-head attention blocks in the course of our multi-encoder learning (MEL) strongly improves robustness and generalization, especially for the mid-size WSJ training set. Combining both improved MEL-based models in the MEL-t-Fusion-Late approach yields the lowest WER of 3.40% on eval92, corresponding to a WER reduction of 0.91% absolute w.r.t. the normal Fusion-Late approach (4.31%), and a remarkable reduction of up to 19% relative compared to the best recently published transformer-based approach by Moriya et al. [28], as shown in Table 2.

All approaches are also evaluated on the Librispeech task with results being reported in Table 3. Among the fusion approaches only Fusion-t-Mid-WS and Fusion-Late yield consistent improvement over all data splits compared to both base-

Table 3: *Transformer-based approaches on Librispeech.*

| Approach | # of inference param. | WER | | | |
		dev clean	dev other	test clean	test other
Baseline-mag	69.8M	3.44	7.80	4.05	8.14
Baseline-phase	69.8M	3.80	9.00	4.43	9.62
Fusion-Early	69.8M	3.35	7.82	3.78	8.14
Fusion-Mid-CC	114.0M	3.70	7.98	4.08	8.25
Fusion-Mid-WS	115.6M	3.43	8.02	3.96	8.34
Fusion-t-Mid-WS	109.3M	3.35	7.26	3.77	7.68
Fusion-Late	139.6M	**2.99**	6.91	3.63	7.34
MEL-t-mag	69.8M	3.37	7.68	3.87	7.90
MEL-t-phase	69.8M	3.68	8.66	4.05	9.03
MEL-t-Fusion-Late	139.6M	3.05	**6.63**	**3.34**	**7.15**

lines, while Fusion-Late is the better yet more costly one. As for the WSJ task, also on Librispeech our MEL-t-X approaches consistently outperform the respective Baseline approaches (4.4% and 8.6% relative improvement on test clean, respectively), while having equal inference complexity. With late fusion of both MEL-enhanced transformer models (MEL-t-Fusion-Late), we achieve our best results on Librispeech with a remarkable WER reduction of 17.5% and 12.2% relative on the clean and other portions of the test set, respectively, compared to the standard transformer approach (Baseline-mag).

5. Conclusion

In this contribution we introduced several fusion mechanisms to transformer-based end-to-end speech recognition. In addition, we apply a novel multi-encoder learning method (MEL), that uses the additional information from a second encoder only during training, while just a single encoder is used during inference. Compared to standard transformer approaches our novel MEL achieves a consistent WER reduction on all investigated tasks at the same runtime and number of parameters. By performing additional fusion, we achieve a WER reduction of 19% relative on the Wall Street Journal task compared to state of the art, thereby defining a new benchmark for transformer-based ASR on that task.

6. Acknowledgements

7. References

[1] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Springer Science and Business Media, 1994.

[2] D. Bahdanau *et al.*, "End-to-End Attention-Based Large Vocabulary Speech Recognition," in *Proc. of ICASSP*, Shanghai,China, Mar. 2016, pp. 4945–4949.

[3] A. Graves and N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," in *Proc. of ICML*, Beijing, China, Jun. 2014, pp. 1764–1772.

[4] A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in *Proc. of ICASSP*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[5] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition," in *Proc. of ICASSP*, Shanghai,China, Mar. 2016, pp. 4960–4964.

[6] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv:1706.03762*, Dec. 2017.

[7] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition," in *Proc. of ICASSP*, Calgary, AB, Canada, Apr. 2018, pp. 5884–5888.

[8] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR Corpus Based on Public Domain Audio Books," in *Proc. of ICASSP*, South Brisbane, QLD, Australia, Apr. 2015, pp. 5206–5210.

[9] S. Karita *et al.*, "A Comparative Study on Transformer vs RNN in Speech Applications," in *Proc. of ASRU*, Singapore, Singapore, Dec. 2019, pp. 449–456.

[10] G. Potamianos, J. Luettin, and C. Neti, "Hierarchical Discriminant Features for Audio-Visual LVCSR," in *Proc. of ICASSP*, Salt Lake City, UT, USA, May 2001, pp. 165–168.

[11] R. Schlüter, A. Zolnay, and H. Ney, " Feature Combination using Linear Discriminant Analysis and its Pitfalls," in *Proc. of INTERSPEECH*, Pittsburgh, PA, USA, Sep. 2006, pp. 345–348.

[12] M. Ravanelli, T. Parcollet, and Y. Bengio, "The Pytorch-Kaldi Speech Recognition Toolkit," in *Proc. of ICASSP*, Brighton, United Kingdom, May 2019, pp. 6465–6469.

[13] J. Luettin, G. Potamianos, and C. Neti, "Asynchronous Stream Modeling for Large Vocabulary Audio-Visual Speech Recognition," in *Proc. of ICASSP*, Salt Lake City, UT, USA, May 2001, pp. 169–172.

[14] H. Misra, H. Bourlard, and V. Tyagi, "New Entropy Based Combination Rules in HMM/ANN Multi-Stream ASR," in *Proc. of ICASSP*, vol. 2, Hong Kong, China, Apr. 2003, pp. 741–744.

[15] T. Lohrenz and T. Fingscheidt, "BLSTM-Driven Stream Fusion for Automatic Speech Recognition: Novel Methods and a Multi-Size Window Fusion Example," in *Proc. of INTERSPEECH*, Shanghai, China, Oct. 2020, pp. 26–30.

[16] J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proc. of ASRU*, Santa Barbara, CA, USA, Dec. 1997, pp. 347–352.

[17] B. Hoffmeister, T. Klein, R. Schlüter, and H. Ney, "Frame Based System Combination and a Comparison With Weighted ROVER and CNC," in *Proc. of INTERSPEECH*, Pittsburgh, PA, USA, Sep. 2006, pp. 537–540.

[18] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum Bayes Risk Decoding and System Combination Based on a Recursion for Edit Distance," *Computer Speech and Language*, vol. 25, no. 4, pp. 802–828, Oct. 2011.

[19] G. Potamianos *et al.*, "Recent Advances in the Automatic Recognition of Audiovisual Speech," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306–1326, Sep. 2003.

[20] S. Receveur, R. Weiss, and T. Fingscheidt, "Turbo Automatic Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 846–862, May 2016.

[21] T. Lohrenz and T. Fingscheidt, "Turbo Fusion of Magnitude and Phase Information for DNN-Based Phoneme Recognition," in *Proc. of ASRU*, Okinawa, Japan, Dec. 2017, pp. 118–125.

[22] J. Du *et al.*, "The USTC-iFlytek Systems for CHiME-5 Challenge," in *CHiME 2018 Workshop on Speech Processing in Everyday Environments*, Hyderabad, India, Sep. 2018, pp. 11–15.

[23] W. Xiong *et al.*, "The Microsoft 2017 Conversational Speech Recognition System," *arXiv:1708.06073*, Aug. 2017.

[24] S. Petridis *et al.*, "Audio-Visual Speech Recognition with a Hybrid CTC/Attention Architecture," in *Proc. of SLT*, Athens, Greece, Dec. 2018, pp. 513–520.

[25] T. Afouras *et al.*, "Deep Audio-Visual Speech Recognition," *arXiv:1809.02108*, Dec. 2018.

[26] B. Li *et al.*, "Does Multi-Encoder Help? A Case Study on Context-Aware Neural Machine Translation," in *Proc. of ACL*, online, Jul. 2020, pp. 3512–3518.

[27] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention Based End-to-End Speech Recognition Using Multi-task Learning," in *Proc. of ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 4835–4839.

[28] T. Moriya *et al.*, "Self-Distillation for Improving CTC-Transformer-Based ASR Systems," in *Proc. of INTERSPEECH*, Shanghai, China, Oct. 2020, pp. 546–550.

[29] D. S. Park *et al.*, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. of INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 2613–2617.

[30] J. Zhang *et al.*, "Improving the Transformer Translation Model with Document-Level Context," *arXiv:1810.03581*, Jul. 2018.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[32] N. Srivastava *et al.*, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jun. 2014.

[33] A. Vaswani *et al.*, "Tensor2Tensor for Neural Machine Translation," *arXiv:1803.07416*, Dec. 2018.

[34] C. Gulcehre *et al.*, "On Using Monolingual Corpora in Neural Machine Translation," *arXiv:1503.03535*, Mar. 2015.

[35] D. B. Paul and J. M. Baker, "The Design for the Wall Street Journal-Based CSR Corpus," in *Proc. of 5th DARPA Speech and Natural Language Workshop*, Stroudsburg, PA, USA, Feb. 1992, pp. 357–362.

[36] D. Povey *et al.*, "The Kaldi Speech Recognition Toolkit," in *Proc. of ASRU*, Waikoloa, HI, USA, Dec. 2011, pp. 1–4.

[37] E. Loweimi, S. M. Ahadi, and T. Drugman, "A New Phase-Based Feature Representation for Robust Speech Recognition," in *Proc. of ICASSP*, Vancouver, BC, Canada, Sep. 2013, pp. 7155–7159.

[38] T. Lohrenz, W. Li, and T. Fingscheidt, "A New TIMIT Benchmark for Context-Independent Phone Recognition Using Turbo Fusion," in *Proc. of SLT*, Athens, Greece, Dec. 2018, pp. 498–505.

[39] Y. Wang *et al.*, "Espresso: A Fast End-to-End Neural Speech Recognition Toolkit," in *Proc. of ASRU*, Singapore, Singapore, Dec. 2019, pp. 136–143.

[40] R. Müller, S. Kornblith, and G. Hinton, "When Does Label Smoothing Help?" *arXiv:1906.02629*, Jun. 2020.

[41] T. Hori, J. Cho, and S. Watanabe, "End-to-End Speech Recognition With Word-Based RNN Language Models," in *Proc. of SLT*, Athens, Greece, Dec. 2018, pp. 389–396.

[42] T. Kudo and J. Richardson, "SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing," *arXiv:1808.06226*, Aug. 2018.

[43] M. Ott *et al.*, "fairseq: A Fast, Extensible Toolkit for Sequence Modeling," in *Proc. of NAACL-HLT 2019: Demonstrations*, Minneapolis, Minnesota, Jun. 2019, pp. 48–53.

[44] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proc. of NeurIPS*, Vancouver, BC, Canada, Dec. 2019, pp. 8024–8035.

[45] E. Tsunoo, Y. Kashiwagi, T. Kumakura, and S. Watanabe, "Transformer ASR with Contextual Block Processing," in *Proc. of ASRU*, Singapore, Singapore, Dec. 2019, pp. 427–433.

[46] S. Karita *et al.*, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Proc. of INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 1408–1412.