TOWARDS END-TO-END INTEGRATION OF DIALOG HISTORY FOR IMPROVED SPOKEN LANGUAGE UNDERSTANDING

Vishal Sunder[*1], *Samuel Thomas*[2], *Hong-Kwang J. Kuo*[2], *Jatin Ganhotra*[2],
Brian Kingsbury[2], *Eric Fosler-Lussier*[1]

[1] The Ohio State University, Columbus, OH, USA
[2] IBM Research AI, Yorktown Heights, NY, USA

ABSTRACT

Dialog history plays an important role in spoken language understanding (SLU) performance in a dialog system. For end-to-end (E2E) SLU, previous work has used dialog history in text form, which makes the model dependent on a cascaded automatic speech recognizer (ASR). This rescinds the benefits of an E2E system which is intended to be compact and robust to ASR errors. In this paper, we propose a hierarchical conversation model that is capable of directly using dialog history in speech form, making it fully E2E. We also distill semantic knowledge from the available gold conversation transcripts by jointly training a similar text-based conversation model with an explicit tying of acoustic and semantic embeddings. We also propose a novel technique that we call DropFrame to deal with the long training time incurred by adding dialog history in an E2E manner. On the HarperValleyBank dialog dataset, our E2E history integration outperforms a history independent baseline by 7.7% absolute F1 score on the task of dialog action recognition. Our model performs competitively with the state-of-the-art history based cascaded baseline, but uses 48% fewer parameters. In the absence of gold transcripts to fine-tune an ASR model, our model outperforms this baseline by a significant margin of 10% absolute F1 score.

Index Terms— spoken dialog system, spoken language understanding, end-to-end systems

1. INTRODUCTION

Traditionally, spoken dialog systems (SDS) comprise an automatic speech recognition (ASR) component followed by a natural language understanding (NLU) component [1, 2, 3, 4]. In recent times, end-to-end (E2E) spoken language understanding (SLU) has gained popularity for SDS [5, 6, 7, 8]. The reason is that state-of-the-art models for ASR and NLU can be inherently large and cascading them leads to an even larger overall model size, which may make them difficult to deploy, especially for on-device applications. Also, ASR errors can degrade the performance of a cascaded SLU system. In contrast, E2E SLU models are compact and more robust to ASR errors as they process speech directly.

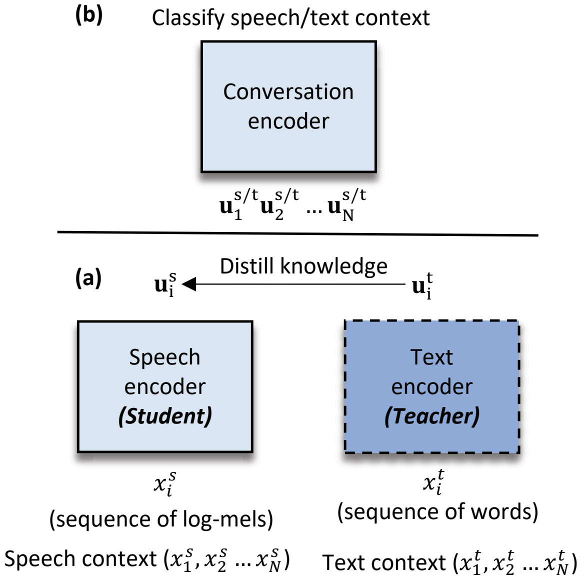
(b)
Classify speech/text context
Conversation encoder
$\mathbf{u}_1^{s/t}\mathbf{u}_2^{s/t} \dots \mathbf{u}_N^{s/t}$
(a)
$\mathbf{u}_i^s$
Distill knowledge
$\mathbf{u}_i^t$
Speech encoder ***(Student)***
Text encoder ***(Teacher)***
x_i^s
(sequence of log-mels)
x_i^t
(sequence of words)
Speech context $(x_1^s, x_2^s \dots x_N^s)$
Text context $(x_1^t, x_2^t \dots x_N^t)$

Fig. 1. Model overview. **(a)** Modality specific utterance encoders; **(b)** Modality agnostic conversation encoder. The text encoder acts as a teacher for the speech encoder during training and is discarded at test time.

It is also well known that a dialog system's NLU performance benefits from using the entire dialog history instead of just the utterance to be labeled. Dialog history helps particularly by resolving ambiguities and co-references. This idea has been explored extensively in written/typed dialog systems [9, 10, 11]. Some recent work has explored ways of incorporating dialog history for intent identification in E2E SLU. However, a limitation here is that the features for the dialog history are extracted from a text encoder which makes the model dependent on a cascaded ASR.

In particular, Tomashenko et al. [12] use text from the previous system prompt as history. In an extension to this,

[*] Work done during an internship at IBM

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

Ganhotra et al. [13] use the entire conversation as history which is a transcript decoded from an ASR, making it prone to ASR errors and also not fully E2E. ASR performance has also been shown to improve by integrating dialog history [14] and intent representations [15].

In this paper, we propose a conversation model with a full E2E integration of dialog history that is significantly more compact than its cascaded counterpart. Our conversation model is hierarchical, where we have an utterance level speech encoder at the lower level and a conversation encoder at the top level.

To further improve the performance, we use a teacher-student training framework to distill knowledge from the semantically richer BERT encoder using available transcriptions. We jointly train a BERT based conversation model along with the speech based model with weight sharing at the conversation level between modalities (see Figure 1). Further, speech embeddings are tied with BERT embeddings via Euclidean loss [16] and a novel use of Contrastive loss [17].

Using dialog history in an E2E manner can lead to an increased training time. We propose a technique called DropFrame to deal with this issue. Similar to dropout [18], this technique drops out random speech frames during training. In addition to a reduced training time, this technique also helps to improve SLU performance.

We perform extensive experiments on the recently released HarperValleyBank corpus [19] which is a task oriented dialog dataset. We focus on the multi-label classification task of dialog action recognition. When gold transcripts are available for fine-tuning an ASR, our model performs competitively with a state-of-the-art cascaded baseline while using 48% fewer parameters. In the absence of gold transcripts, our model significantly outperforms this baseline.

2. HIERARCHICAL CONVERSATION MODEL

We treat the SLU problem as a context labeling task, where instead of labeling the current utterance, the model labels the entire context (dialog history + current utterance). Let the speech based context, $C^s = (x_1^s, x_2^s, ..., x_N^s)$ where x_N^s is the current utterance and the rest is dialog history. Text based context, $C^t = (x_1^t, x_2^t, ..., x_N^t)$ which are gold transcripts of C^s. The model overview shown in Figure 1 is described below.

Speech encoder (Student): The speech signal, x_i^s from the i^{th} utterance in the context is encoded using a neural network.

$$\mathbf{u}_i^s = f^s(x_i^s; \boldsymbol{\theta}^s)$$

$f^s(.; \boldsymbol{\theta}^s)$ is the speech encoder parameterized by $\boldsymbol{\theta}^s$. The encoded speech context is denoted as, $\mathbf{U}^s = \{\mathbf{u}_i^s\}_{i=1}^N$.

We use the transcription network of a pretrained RNN-T based ASR model [20] as the speech encoder which encodes the speech signal directly to give a vector representation. This model is first pre-trained for ASR using 40-dimensional global mean and variance normalized log-mel filterbank features, extracted every 10 ms. These features are further augmented with Δ and $\Delta\Delta$ coefficients, every two consecutive frames are stacked, and every second frame is skipped, resulting in 240-dimensional vectors every 20 ms. Thus, x_i^s is a sequence of 240-dimensional log-mel features. The encoder, $f^s(.; \boldsymbol{\theta}^s)$ is a 6-layer bidirectional LSTM with 640 cells per layer per direction. We concatenate vectors from the forward and backward direction of the LSTM's last layer for the last time step to get a 1280-dimensional vector. This vector goes through a linear layer which shrinks its dimensionality to 768. Thus, $\mathbf{u}_i^s$ is a 768-dimensional representation of a speech utterance.

Text encoder (Teacher): The i^{th} text utterance, x_i^t is a sequence of word tokens encoded using WordPiece embeddings. x_i^t is encoded as,

$$\mathbf{u}_i^t = f^t(x_i^t; \boldsymbol{\theta}^t)$$

$f^t(.; \boldsymbol{\theta}^t)$ is a text encoder parameterized by $\boldsymbol{\theta}^t$. The encoded text context is denoted as, $\mathbf{U}^t = \{\mathbf{u}_i^t\}_{i=1}^N$.

We use a pretrained BERT model[1] [21] as the text encoder, $f^t(.; \boldsymbol{\theta}^t)$. The 768-dimensional [CLS] token output from this model is treated as the text representation, $\mathbf{u}_i^t$. The text encoder is only used during training to transfer semantic knowledge to the speech encoder and is discarded at test time.

Conversation encoder (Modality agnostic): Once we get representations for the utterances in a context for the speech and text modalities, we use a higher level sequence encoder, $g(.; \boldsymbol{\phi})$ to encode the context into a single representation.

$$\mathbf{c}^s = g(\mathbf{U}^s; \boldsymbol{\phi})$$
$$\mathbf{c}^t = g(\mathbf{U}^t; \boldsymbol{\phi})$$

Here, $\mathbf{c}^s$ and $\mathbf{c}^t$ are representations for the speech and text context respectively. Notice that the sequence encoder is shared between the speech and text modalities. This helps to transfer knowledge from text to speech during training [5].

We model $g(.; \boldsymbol{\phi})$ as a 6-layer 1-head transformer (with a similar architecture as Vaswani et al. [22]). We add absolute positional embeddings to the sequence and pass it to the transformer. The output of the last step in the sequence output of the transformer is used as $\mathbf{c}^s$ or $\mathbf{c}^t$. We also explore an LSTM architecture for $g(.; \boldsymbol{\phi})$ and show additional results.

3. TRAINING PROCEDURE

The representations $\mathbf{c}^s$ and $\mathbf{c}^t$ from $g(.; \boldsymbol{\phi})$ are passed into a classification layer to get predictions. The binary cross-entropy loss is then computed for the speech (denoted as $L_{BCE}(\boldsymbol{\theta}^s, \boldsymbol{\phi})$) and the text modalities (denoted as $L_{BCE}(\boldsymbol{\theta}^t, \boldsymbol{\phi})$). The proposed model can be used in three ways:

[1] https://huggingface.co/bert-base-uncased

1. **HIER-ST**: Co-trained with speech and text using $L_{BCE}(\boldsymbol{\theta}^s, \boldsymbol{\phi}) + L_{BCE}(\boldsymbol{\theta}^t, \boldsymbol{\phi})$.
2. **HIER-S**: Trained with speech only using $L_{BCE}(\boldsymbol{\theta}^s, \boldsymbol{\phi})$.
3. **HIER-T**: Trained with text only using $L_{BCE}(\boldsymbol{\theta}^t, \boldsymbol{\phi})$.

3.1. Cross-modal loss functions

In general, BERT embeddings are well suited for NLU tasks as they are pretrained on large text corpora. The semantic knowledge from BERT can be transferred to speech embeddings through cross-modal loss functions. We investigate two losses for this purpose, Euclidean loss and Contrastive loss.

Euclidean loss (L_{EUC}): This is computed as the $L2$ distance between speech and text representations, [16]. Formally,

$$L_{EUC}(\boldsymbol{\theta}^s) = \frac{1}{|B|}\sum_{i=1}^{|B|} \left\| \mathbf{u}_N^s[i] - \mathbf{u}_N^t[i] \right\|_2$$

where $|B|$ is batch size; $\mathbf{u}_N^s[i]$ and $\mathbf{u}_N^t[i]$ are respective outputs from $f^s(.;\boldsymbol{\theta}^s)$ and $f^t(.;\boldsymbol{\theta}^t)$ of the i^{th} utterance in the batch.

Contrastive loss (L_{CON}): We define the similarity between the i^{th} speech and the j^{th} text utterance in a training batch as:

$$s_{ij} = cos(\mathbf{u}_N^s[i], \mathbf{u}_N^t[j])/\tau$$

where $cos(.,.)$ is the cosine similarity and τ is a temperature hyperparameter. Then, we define the contrastive loss as:

$$L_{CON}(\boldsymbol{\theta}^s) =$$
$$-\frac{1}{2|B|}\sum_{i=1}^{|B|}\big(\log \frac{\exp(s_{ii})}{\sum_{j=1}^{|B|}\exp(s_{ij})} + \log \frac{\exp(s_{ii})}{\sum_{j=1}^{|B|}\exp(s_{ji})}\big)$$

The first term in the above sum is the log-likelihood of a text utterance given the corresponding speech and the second term, vice-versa. By minimizing the above loss, the similarity between same utterances in different modalities is maximized and that between different utterances is minimized.

The gradients from these losses only update the speech branch so that the BERT representations are not affected.

All models were trained in PyTorch on a A100 GPU using the Adam optimizer at a learning rate of $2e-5$. We use 10% dropout and early stopping with a patience of 10 epochs. The maximum context length is set to 10 utterances. For L_{CON}, we set $\tau = 0.07$ after tuning it on the development set.

3.2. DropFrame

Compared to a sequence of text tokens, we noticed that the corresponding sequence of speech frames can be five times longer. When dialog history is used in speech form, this sequence can become very long which results in an increased training time for E2E models.

# Frames (l)	Macro-F1	Train time
64	56.5	4
256	61.7	8
1024	60.2	25
All	59.9	27

Table 1. Effect of DropFrame. Train time is in minutes/epoch.

To solve this problem, we propose a simple yet effective novel technique that we call DropFrame. If the length of the sequence, x_i^s is longer than a hyperparameter l, we randomly drop out some of the frames in x_i^s such that its length becomes l. In our experiments, we found that $l = 256$ works best. At test time, we switch off DropFrame. This technique not only decreases the training time but also improves performance.

4. EXPERIMENTS

4.1. Dataset

We conduct experiments on the HarperValleyBank (HVB) corpus [19] which is a public dataset of simulated spoken conversations between bank employees and customers. It consists of 1446 dyadic human-human conversations with audio and human annotated transcripts. There are 59 unique speakers involved and the data is annotated for speaker identity, caller intent, dialog actions and emotional valence.

In this work, we focus on the dialog action recognition task which is a multilabel classification problem. There are 16 dialog actions and each utterance can be labelled with one or more of these. We split the conversations into train, development and test sets. The train set contains 1174 conversations (10 hours of audio) and the test set has 199 conversations (1.8 hours hours of audio), the rest make up the development set[2].

4.2. Effect of DropFrame

Table 1 shows the effect of our proposed DropFrame approach (Section 3.2). We fine-tuned the frame length l with several values on the development set, but show results for only four of these values on the test set. As we can see, the training time is reduced significantly when frames are dropped. Also note that when we increase the value of l beyond 256, the performance starts to drop. We hypothesize that dropping out frames works as an effective regularizer which helps the model generalize better by aggregating over different sequences of frames for the same utterance [23, 24]. A more thorough analysis of this is left for future work.

4.3. SLU results

Table 2 shows experiments assuming gold transcripts are available during training. **(1T-3T)** shows the oracle performance

[2] To get a conversation level data split, it was necessary to accept speaker overlap between the train, development, and test sets.

	Model	Macro-F1	# Params
(1T)	BERT (on utterance)	56.1	168M
(2T)	BERT (on context)	**63.5**	168M
(3T)	HIER-T	63.3	200M
(1C)	ASR $\longrightarrow$ BERT (on context)	**62.2**	168M
(2C)	ASR $\longrightarrow$ HIER-T	61.3	200M
(1E)	LSTM (on utterance)	54.0	54M
(2E)	HIER-S	58.3	88M
(3E)	HIER-ST	59.0	88M
(4E)	HIER-ST + L_{EUC}	60.3	88M
(5E)	HIER-ST + L_{CON}	**61.7**	88M
(6E)	HIER-ST + L_{EUC} + L_{CON}	60.9	88M
(7E)	HIER-ST + L_{CON} ($g(.;\phi)$ = LSTM)	61.3	**62M**

Table 2. Performances when gold transcripts are available with model size (in million parameters). **(T)** denotes text-based models; **(C)** denotes ASR-NLU cascaded models; **(E)** denotes E2E models. In **(C)**, the ASR model is adapted on in-domain data. In **(E)**, the speech encoder is the transcription network from an adapted RNN-T ASR. Note that we did not observe significant variance in results between runs.

	Model	Macro-F1
(3C)	ASR $\longrightarrow$ BERT (on context)	50.3
(8E)	HIER-S	57.7
(9E)	HIER-ST + L_{CON} (w/ ASR text)	**60.3**
(10E)	HIER-ST + L_{CON} (w/ Gold text)	**61.7**

Table 3. Performances when gold transcripts are unavailable (except **(10E)**). In **(C)**, we use an off-the-shelf ASR. In **(E)**, the speech encoder is from an off-the-shelf RNN-T ASR.

on clean text data. In **(1T)**, only the current utterance is fed to BERT whereas in **(2T)**, the entire context (dialog history + current utterance) is given to BERT. **(3T)** is our hierarchical model with only the text branch. Clearly, using dialog history on text makes a big difference in performance. In the cascaded ASR-NLU settings, **(1C-2C)**, we see a drop in performance compared to **(2T-3T)**, because of ASR errors.

(1E-7E) represent E2E models using a pretrained ASR based speech encoder. This encoder corresponds to a 6-layer LSTM transcription network of an RNN-T model adapted on the HVB dataset. **(1E)** uses this LSTM on the current utterance, whereas all the others use dialog history in speech form through our hierarchical setup. Note that **(2E)** already gives a 4.3% improvement over **(1E)** which shows that our hierarchical model effectively uses history in an E2E manner. **(3E-7E)** use the available text transcriptions to train the full model, although the text branch is discarded at test time. We see that co-training with text helps improve performance. By adding the different cross-modal losses defined in Section 3.1, we see further performance improvements. Note that our best performing E2E model, **(5E)** uses the proposed contrastive loss. This model shows comparable performance to **(1C)** which has a distinct advantage that BERT can attend to the entire text conversation in a more fine-grained manner at the word level. This is not feasible in a speech based conversation which can be intractably longer compared to text. Hence, we have proposed a tractable hierarchical setup. Notably, **(5E)** gives a better performance compared to a similar hierarchical cascaded setup, **(2C)** which shows its robustness to ASR errors.

Note that all of the proposed end-to-end models are significantly smaller in size compared to the cascaded models. To mimic a more deployable scenario, we replaced the transformer-based conversation encoder, $g(.;\phi)$ with a 1-layer bidirectional LSTM and add the forward and backward representations of the last time step to get the context representations. The result is shown in row **(7E)**. We see that the F1 is only 0.9% less than the cascaded baseline **(1C)**, but the size of the former is 64% smaller and hence easily deployable. This also shows that the conversation encoder is agnostic to the underlying neural architecture.

For the next set of experiments in Table 3, we assume a constrained SLU setting. Here, gold transcripts are not available to adapt the ASR model. Instead, only the SLU labels are available. This practical setting is common as it reduces the transcription cost when creating an SLU dataset. The cascaded baseline in this case **(3C)** has a much worse performance compared to **(1C)** in Table 2. As the ASR used in **(3C)** has not been adapted to the in-domain data, its output can be noisy (WER of 11.3% compared to the adapted case with a WER of 1.7%). We also observe that in a lot of cases the unadapted RNN-T based ASR model does not output a transcript because of acoustic mismatch. This can be detrimental for the cascaded SLU system. In the absence of human transcriptions, we trained our best performing HIER-ST model from Table 2 (with L_{CON}) with the ASR hypotheses from the off-the-shelf ASR. The result is shown in row **(9E)** and we observe a significant improvement of 10% absolute F1 over **(3C)**. This is because our model is fully E2E and robust to ASR errors.

We also conduct an additional experiment where although human transcriptions are available for the spoken data, they are not used for adapting the ASR; rather, they are used to co-train the HIER-ST model only **(10E)**. In this setup, we see a similar performance as **(5E)** in Table 2. This shows that our model's speech encoder does not need to be adapted to every new domain using the ASR objective and can be used off-the-shelf. This can save a lot of time in new domain deployments.

5. CONCLUSION

We believe that this work is the first step towards building a fully E2E SLU system using dialog history in speech form. We propose a hierarchical conversation model that can encode speech utterances at the lower level and the entire conversation at the higher level. We use text transcriptions to co-train a similar text-based system which acts as a teacher for our E2E system. We also incorporate cross-modal loss functions which further enhance the performance of the E2E system. Future work should explore a computationally efficient encoding of speech-based conversations using a single BERT-like model.

6. REFERENCES

[1] Gokhan Tur, Jerry Wright, Allen Gorin, Giuseppe Riccardi, and Dilek Hakkani-Tür, "Improving spoken language understanding using word confusion networks," in *Proc. ICSLP*, 2002.

[2] Dilek Hakkani-Tür, Frédéric Béchet, Giuseppe Riccardi, and Gokhan Tur, "Beyond ASR 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, 2006.

[3] Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young, "Discriminative spoken language understanding using word confusion networks," in *Proc. SLT.*, 2012.

[4] Chao-Wei Huang and Yun-Nung Chen, "Adapting pretrained transformer to lattices for spoken language understanding," in *Proc. ASRU.*, 2019.

[5] Subendhu Rongali, Beiye Liu, Liwei Cai, Konstantine Arkoudas, Chengwei Su, and Wael Hamza, "Exploring transfer learning for end-to-end spoken language understanding," *arXiv preprint arXiv:2012.08549*, 2020.

[6] Yu-An Chung, Chenguang Zhu, and Michael Zeng, "Splat: Speech-language joint pre-training for spoken language understanding," in *NAACL*, 2021.

[7] Edmilson Morais, Hong-Kwang J Kuo, Samuel Thomas, Zoltán Tüske, and Brian Kingsbury, "End-to-end spoken language understanding using transformer networks and self-supervised pre-trained features," in *ICASSP*, 2021.

[8] Pavel Denisov and Ngoc Thang Vu, "Pretrained semantic speech embeddings for end-to-end spoken language understanding via cross-modal teacher-student learning," 2020.

[9] Chandrakant Bothe, Cornelius Weber, Sven Magg, and Stefan Wermter, "A context-based approach for dialogue act recognition using simple recurrent neural networks," *arXiv preprint arXiv:1805.06280*, 2018.

[10] Vipul Raheja and Joel Tetreault, "Dialogue act classification with context-aware self-attention," *arXiv preprint arXiv:1904.02594*, 2019.

[11] Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloe Clavel, "Guiding attention in sequence-to-sequence models for dialogue act prediction," in *Proc. AAAI*, 2020.

[12] Natalia Tomashenko, Christian Raymond, Antoine Caubrière, Renato De Mori, and Yannick Estève, "Dialogue history integration into end-to-end signal-to-concept spoken language understanding systems," in *Proc. ICASSP*, 2020.

[13] Jatin Ganhotra, Samuel Thomas, Hong-Kwang J Kuo, Sachindra Joshi, George Saon, Zoltán Tüske, and Brian Kingsbury, "Integrating dialog history into end-to-end spoken language understanding systems," *Proc. Interspeech*, 2021.

[14] Suyoun Kim, Siddharth Dalmia, and Florian Metze, "Gated embeddings in end-to-end speech recognition for conversational-context fusion," *arXiv preprint arXiv:1906.11604*, 2019.

[15] Swayambhu Nath Ray, Minhua Wu, Anirudh Raju, Pegah Ghahremani, Raghavendra Bilgi, Milind Rao, Harish Arsikere, Ariya Rastrow, Andreas Stolcke, and Jasha Droppo, "Listen with intent: Improving speech recognition with audio-to-intent front-end," *arXiv preprint arXiv:2105.07071*, 2021.

[16] Yinghui Huang, Hong-Kwang Kuo, Samuel Thomas, Zvi Kons, Kartik Audhkhasi, Brian Kingsbury, Ron Hoory, and Michael Picheny, "Leveraging unpaired text data for training end-to-end speech-to-intent systems," in *Proc. ICASSP*, 2020.

[17] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang, "Cross-modal contrastive learning for text-to-image generation," in *Proc. CVPR*, 2021.

[18] Nitish Srivastava, "Improving neural networks with dropout," *University of Toronto*, vol. 182, 2013.

[19] Mike Wu, Jonathan Nafziger, Anthony Scodary, and Andrew Maas, "HarperValleyBank: a domain-specific spoken dialog corpus," *arXiv:2010.13929*, 2020.

[20] George Saon, Zoltán Tüske, Daniel Bolanos, and Brian Kingsbury, "Advancing RNN transducer technology for speech recognition," in *Proc. ICASSP*, 2021.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017.

[23] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," 2019.

[24] Arjit Jain, Pranay Reddy Samala, Deepak Mittal, Preethi Jyoti, and Maneesh Singh, "Spliceout: A simple and efficient audio augmentation method," *arXiv preprint arXiv:2110.00046*, 2021.