PROMPTFORMER: PROMPTED CONFORMER TRANSDUCER FOR ASR

Sergio Duarte-Torres[1], *Arunasish Sen*[2], *Aman Rana*[1]
Lukas Drude[1], *Alejandro Gomez-Alanis*[1], *Andreas Schwarz*[1], *Leif Rädel*[1], *Volker Leutnant*[1]

[1]Amazon AGI, Aachen, Germany; [2]Amazon AGI, Cambridge, UK

ABSTRACT

Context cues carry information which can improve multi-turn interactions in automatic speech recognition (ASR) systems. In this paper, we introduce a novel mechanism inspired by hyper-prompting to fuse textual context with acoustic representations in the attention mechanism. Results on a test set with multi-turn interactions show that our method achieves 5.9% relative word error rate reduction (rWERR) over a strong baseline. We show that our method does not degrade in the absence of context and leads to improvements even if the model is trained without context. We further show that leveraging a pre-trained sentence-piece model for context embedding generation can outperform an external BERT model.

Index Terms— Conformer transducer, text context modeling, prompting, Automatic Speech Recognition

1. INTRODUCTION

Cross-utterance context has been shown to be beneficial for improving the accuracy of long-form [5] and dialog-oriented ASR systems [22, 2]. This context can refer to previous ASR recognitions, system responses, dialog acts [2, 3, 20] or other information up to the current point of the speech session [23, 21]. In this work, we integrate context into the attention mechanism of the encoder of conformer-transducer ASR models. Our goal is to improve ASR tasks involving multi-turn interactions. A typical multi-turn interaction example can arise when an user follows up a query like *who is the singer of* ***x****?* with *when was* ***x*** *released?* or *can you play a song from* ***x****?*. We propose to account for such textual context like ***x*** across turns by concatenating textual and acoustic representations in the keys and values of the Multi-Head Attention (MHA) mechanism of the conformer encoder. This approach draws inspiration from hyper-prompting [9]. Our approach differs from traditional multi-headed cross-attention methods [11] in that it does not require separate kernels to create textual query, key and values projections. In our method, acoustic and textual kernels are shared, which leads to better fused multimodal representations, as demonstrated by our experiments. Alternatively, [23] concatenates context representations in the feature dimension of the input. This method is prohibitive when using a large number of context embeddings and under-performs attention-based methods. Context has also been integrated by expanding the architecture of end-to-end (E2E) ASR systems [11, 19, 16, 23, 10, 15, 4]. [23] includes context encoders and a context combiner module, which are trained with a combined loss to predict intents and transcriptions. [11] adds an embedding extractor to represent sentence pieces of context tokens and an attention module to attend on each token. [18] introduces a dedicated biasing and attending mechanism module to the listen-attend-spell (LAS) architecture to bias context words. [10] proposes an expanded Transformer ASR with a conformer encoder that uses multiple consecutive utterances as context for monologue and dialogues. Similarly, [15] proposes a token and sentence hierarchical transformer text encoder, which is trained by distilling from a pre-trained large context language model. Although these extensions have proven effective to leverage contextual information, they come at the expense of an increase in model parameters. They also add complexity to the training regime, leading to increases in latency and cost. In [5], it is proposed to fuse context, text and acoustic representations in the joint network using context representations obtained with a Bidirectional Encoder Representations from Transformers (BERT) model. These are summarized with self-attentive pooling. In this work, we compare context representations obtained with a pretrained BERT model against an encoder derived from the embedding layer of the prediction network. We show that the latter outperforms the representations from the former, with the added advantage of requiring less OPS. Our method also deviates from [5] in that it does not require to train all model parameters to obtain significant improvements.

In section 2, we describe our conformer model architecture as well as the methods we consider for context generation and context consumption. In section 3, we describe our experimental setup, and discuss experimental results in section 4.

2. METHOD

2.1. Streaming Conformer Transducer

In this work, the Recurrent Neural Network Transducer (RNN-T) [7] model structure is used for ASR. The neural

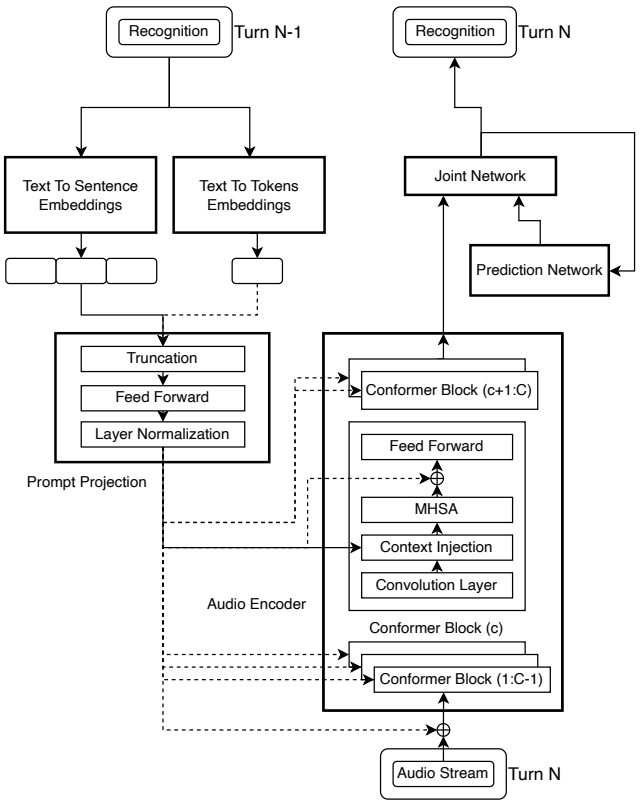

Fig. 1: *Proposed system for generating and consuming textual context prompts. Dotted lines represent the various ablative configurations described in our result section.*

transducer consists of three modules: (i) audio encoder, (ii) prediction network, and (iii) joint network. Our audio encoder is based on the conformer architecture [8]. To use the audio encoder in a streaming setup, we use a causal masking with temporally shifted sliding window masks for each frame of audio.

2.2. Context Generation

We use ASR recognitions from the previous turns $\mathbf{text}_{rec}^{1:N}$ (which we denote as $\mathbf{text}$ for simplicity, as we use $\mathrm{N} = \mathbf{1}$ for all our experiments) as context source. We use two approaches to convert the $\mathbf{text}$ into a token sequence $\mathbf{T}_{1:S}$ of length S: by (a) using an external pre-trained BERT model, and (b) using our internal pre-trained sentence piece model (SPM). For the former, we use a frozen BERT-Tiny model [6] having 2 transformer layers with 128 head size (4.4M parameters) to generate either sentence or token level embeddings. For the sentence level embeddings, we obtain the CLS token output representation, denoted as $\mathbf{Enc}_{bert}^{CLS}$. For the token level embeddings, $\mathbf{Enc}_{bert}$, we use the corresponding outputs of individual subword units and discard the CLS token output. For the internal pre-trained SPM approach, we utilize a learnable embedding layer but initialize the embedding in two different methods: (1) random initialization $\mathbf{Emb}_{ri}$, and (2) initialization from the pre-trained RNN-T prediction network embedding layer $\mathbf{Emb}_{pn}$.

The encoded token sequence $\mathbf{T}_{1:S}$ is projected into the encoder's hidden dimension via a series of dense projections $\mathbf{W}_p$ and $\mathbf{tanh}$ activations, which has shown beneficial in prompt fine-tuning in NLP tasks [14]. Finally, we apply a Layer Normalization (LN) to obtain a prompt embedding $\mathbf{P}_{1:S}$. For summarization techniques like $\mathbf{Enc}_{bert}^{CLS}$, the sequence length is 1. For other cases, we perform a truncation of the token sequence $\mathbf{P}_{1:S}$ to obtain $\mathbf{P}_{1:TW}$ with fixed token window size $\mathrm{TW} = \mathbf{30}$ to cater towards runtime latency requirements of our streaming setup. Our final context representation is therefore given by

$$\begin{aligned} \mathbf{P}_{1:S} &= \mathbf{tanh}_{pn}(...\mathbf{tanh}_{p1}(\mathbf{T}_{1:S}\mathbf{W}_{p1})...\mathbf{W}_{pn}), \\ \mathbf{P}_{1:TW} &= \mathbf{Trunc}_{1:TW}(\mathrm{LN}(\mathbf{P}_{1:S})) \end{aligned} \tag{1}$$

The effective number of trainable weights introduced in both setups (a) and (b) is $\|\mathbf{W}_p\|$. The overall number of ops is significantly larger in-case of (a) due to the extra BERT encoder.

2.3. Context Consumption

Our first context consumption baseline is based on simple concatenation approach, where the context embedding is concatenated to the features of each acoustic frame along the feature dimension, as described in Equation 2. This is only applicable when we use a summary embedding like $\mathbf{Enc}_{bert}^{CLS}$.

$$\hat{\mathbf{A}}_{input} = (\mathbf{P}_{1:1} \oplus \mathbf{A}_{input}) \tag{2}$$

Our second context consumption baseline is inspired from [11] and [18]. In this case, masked cross-attention is used as a separate biasing module. We use the output of Multi-Head Self-Attention (MHSA) module at conformer block c, denoted as $\mathbf{MHSA}_{output}^{c}$, and $\mathbf{P}_{1:TW}$ to produce the queries, keys and values using separate Cross-Attention (CA) kernels $\mathbf{W}_{qca}$, $\mathbf{W}_{kca}$, $\mathbf{W}_{vca}$ respectively. These are used in a MHA module to produce $\mathbf{MHCA}_{output}^{c}$ for block c. This is then combined back with $\mathbf{MHSA}_{output}^{c}$ and passed to the corresponding feed forward layer. The entire operation is described in Equation 3.

$$\begin{aligned} \mathbf{kca}^c &= (\mathbf{P}_{1:TW})\mathbf{W}_{kca}, \\ \mathbf{vca}^c &= (\mathbf{P}_{1:TW})\mathbf{W}_{vca}, \\ \mathbf{qca}^c &= (\mathbf{MHSA}_{output}^{c})\mathbf{W}_{qca}, \\ \mathbf{MHCA}_{output}^{c} &= \mathbf{MHA}(\mathbf{qca}^c, \mathbf{kca}^c, \mathbf{vca}^c), \\ \mathbf{CA}_{out}^{c} &= \mathbf{MHSA}_{output}^{c} + \mathbf{MHCA}_{output}^{c} \end{aligned} \tag{3}$$

Our proposed context consumption approach described in Fig. 1 is very similar to hyper-prompting [9]. However, unlike [9], we only have a single global prompt. The number of global prompts can be extended towards various task specific fine-tuning use-cases and is left for future work. We concatenate $\mathbf{P}_{1:TW}$ to the acoustic representations $\mathbf{A}_{-CW:i}^{c}$ along

the temporal axis. Next we project the concatenated tensor with key and value kernels to obtain augmented keys $\hat{\mathbf{k}}^c$ and values $\hat{\mathbf{v}}^c$. These are used in a MHA operation along with original queries $\mathbf{q}^c$ obtained from $\mathbf{A}^c_{-CW:i}$ only. In this way we do not increase the effective acoustic temporal sequence length at the output of MHA. We describe the entire operation in Equation 4.

$$\begin{aligned} \hat{\mathbf{k}}^c &= (\mathbf{P}_{1:TW} \oplus \mathbf{A}^c_{-CW:i})\mathbf{W}_k, \\ \hat{\mathbf{v}}^c &= (\mathbf{P}_{1:TW} \oplus \mathbf{A}^c_{-CW:i})\mathbf{W}_v, \\ \mathbf{q}^c &= \mathbf{A}^c_{-CW:i}\mathbf{W}_q, \\ \mathbf{MHSA}^c_{output} &= \mathbf{MHA}(\mathbf{q}^c, \hat{\mathbf{k}}^c, \hat{\mathbf{v}}^c) \end{aligned} \tag{4}$$

We believe our method has the following advantages over Multi-Head Cross Attention (MHCA): (i) The textual and acoustic kernels for query, key and value projections are shared, forcing the kernels to produce fused multimodal representations. (ii) The softmax is shared between textual context and regular acoustic context frames, thereby leading to better equalization of the final output representation of a particular acoustic frame.

3. EXPERIMENTAL SETUP

3.1. Model configuration

As a seed model for all experiments we use a RNN-T conformer transducer with 91.7 M parameters and 12 conformer blocks. The encoder dimension is 512 and each conformer block has 8 self-attention heads with head size 64. The encoder has in total 60.6 M parameters. We use $\mathbf{ContextWindow(CW) = 40}$ for all our experiments. The prediction network consist of 2 x 1280 long short-term memory (LSTM) layers and it consists of 23.9 M parameters. The joint network is a feed-forward layer with 512 units and tanh activation. A softmax layer is used on top with a word-piece vocabulary size of 4K. The audio is processed into 64-dimensional Log-Mel-Frequency features per frame with frame shift of 10 ms downsampled by a factor of 3. SpecAugment was used for feature-based augmentation during model training [17]. The seed model was trained using Adam optimizer with a linear warm-up period (5K steps) and exponential decay thereafter [12]. The seed model was trained without context. We evaluate a baseline and our proposed methods for context modeling by fine-tuning the seed model without and with the presence of context [10, 1]. Fine-tuning is conducted for 250K steps and the final model is obtained by averaging the last 10 checkpoints (5K steps each) [24].

3.2. Datasets

Models are trained with a de-identified in-house English voice assistant dataset consisting of audio and text transcriptions. A fine-tuning training set of 140K hours of audio with context is used. Context is the recognition associated to the previous stream within a session. Sessions are defined by grouping streams from the same voice assistant device and using a sliding window of 90 seconds between two consecutive streams. We estimated that up to 70% of the utterances have non-empty context ASR recognition. We report rWERR with respect to the baseline (fine-tuning without context). We show results on two in-house test sets: average traffic and multi-turn. The former contains 29.1K utterances, and 71% of the utterances have non-empty context. The second dataset has 7K utterances with non-empty context for each utterance. As reference, the fine-tuned baseline model improves by 11% and 3.3% rWERR over the seed model in the multi-turn and average traffic test sets, respectively. Performance of the seed model is below 10% WER absolute.

Table 1: Comparison of relative word error rate reduction (rWERR) achieved by different context consumption and context generation approaches. *CA* refers to Cross attention, *CP* refers to *copying* joint network params for first projection layer initialization.

Context cons.	Context gen.	Multi-Turn.	Avg. Traffic.	Params
None (Baseline)	None	0.0%	0.0%	+0.0%
CA	BERT/Sent.	3.2%	2.4%	+15.1%
CA + CP	BERT/Tok.	4.1%	3.3%	+18.7%
Feature concat.	BERT/Sent.	2.3%	1.4%	+3.8%
Prompts	BERT/Sent.	3.4%	2.0%	+0.5%
Prompts	BERT/Tok.	5.0%	2.9%	+0.5%
Prompts	SPM/Tok.	4.6%	3.7%	+3.7%
Prompts + CP	SPM/Tok.	5.9%	3.4%	+3.7%

4. EXPERIMENTAL RESULTS

4.1. Context integration approaches

Table 1 compares the integration approaches explored in this paper. Copying (CP) refers to copying the joint network parameters in first projection layer as initialization. All models with context provide significant gains in respect to the baseline. Cross Attention (CA) and prompting variants using sentence embeddings outperform Feature Concatenation (FC) by up to +1.8% and +1.1% rWERR respectively. This can be due to the lack of a mapping mechanism between textual and acoustic space in the FC method. A similar trend was observed in the average traffic test set. All prompt token models consistently outperformed the equivalent CA variant in the multi-turn test set, despite the significantly smaller increase in model parameters. For instance, prompting using BERT and SPM tokens achieves 5.0% and 5.9% rWERR respectively. The CA model (with SPM tokens) achieves 4.1% rWERR. This model adds +18.7% parameters, while the prompting alternatives introduce only up to 3.7% more.

Table 2: Relative word error rate reduction (rWERR) on Average Traffic test set for selected models categorized by the presence or absence of context. All models use prompting for context consumption.

Context gen.	All	Context	No Context
None (Baseline)	0.0%	0.0%	0.0%
BERT/Sent.	2.0%	1.9%	0.9%
BERT/Toks.	2.9%	2.9%	1.6%
SPM/Toks.	3.7%	4.1%	1.5%
Dataset Size	29.1K	20.7K	8.4K

4.2. Textual Context Representation

We observed token embeddings lead to better WER compared to sentence embeddings. For example, using BERT tokens lead to an improvement of +1.6% and +0.9% rWERR in the multi-turn and average traffic test sets over the sentence embedding counterpart. This indicates that the acoustic encoder benefits from the finer level of granularity provided by distinct sub-words representations. Similar trends have been observed for textual encoders [13]. We also found BERT marginally outperforms SPM embeddings (+0.4% rWERR in the multi-turn test set) despite the smaller amount of added parameters to the model (+0.5% vs +3.7%). This may be the result of the BERT model having seen more textual data and providing better textual representations than the trained SPM projection layers. On the other hand, results were comparable in the average traffic test set. The best results were obtained with the SPM encoder when the first projection layer is initialized with the weights of the prediction network embedding layer (i.e. *copying*). This initialization enables encoding text with the same word piece vocabulary of the base model, avoiding a vocabulary mismatch. This approach is advantageous because it reduces the need to use an external encoder. Despite the slight increase in encoder parameters compared to the BERT variant, the total number of OPs is smaller since we do not need to run a BERT forward pass when encoding the text, reducing overall inference latency.

Table 2 shows rWERR results for a breakdown of the average traffic test set based on whether there is or not context available. Values reported are in respect to the word error rate (WER) of the baseline on each subset. We observed that the majority of the gains are obtained on the subset of utterances having context available. For instance, the model trained with the SPM encoder improves over the baseline by 3.1% rWERR on the subset of utterances having context. Candidates also improve up to 1.6% on the subset of utterances without context, which shows our approach is robust for the cases when context is not available.

4.3. Training regime

Table 3 provides a comparison when not all components of the RNN-T model are fine-tuned. Results are shown for the model trained with SPM encoder with *copying*. We found that by fine-tuning only the MHA modules and projection layers, almost the same performance can be achieved, with only negligible changes of -0.1% and -0.4% rWERR (in respect to fine-tuning all parameters) in the multi-turn and average traffic test set. The larger gap in the latter test set can be explained by the fact that fine-tuning other RNN-T modules contributes to overall performance gains. This result also suggests that adapting the attention mechanism is sufficient for the model to incorporate context effectively. This training strategy provides faster training speed given only 11.4% of the model parameters are fine-tuned. We also experimented with fine-tuning only the projection layers. We found significant gains are still obtained in the multi-turn test set, when fine-tuning for 50k steps (3.9% word error rate reduction (WERR)) and for 25K steps (3.5% WERR). However, there was a loss of performance of -1.9% WERR in respect to the best performing model in the multi-turn test set. We noticed training starts to overfit after 50K. This result demonstrates our approach is also effective incorporating context for the scenario when modifying the base model parameters is not feasible. Under this regime the training time is reduced significantly since only up to 2.2M parameters need to be trained for a limited number of steps.

Table 3: Relative word error rate reduction (rWERR) when fine-tuning different model components. We utilized the model using SPM embeddings and copying. *All* refers to fine-tuning the full conformer-transducer model. Fine tuning was carried out for 250K steps unless a different value is specified. WERs are relative to fine-tuning without context.

Model setup	Multi-turn	Avg. Traffic	Train. params.
Baseline	0.0%	0.0%	91.7M
All	5.9%	3.4%	103.5M
MHAs and projs.	5.8%	3.1%	11.7M (11.4%)
Only projs.	3.5%	2.5%	2.3M (2.2%)
Only projs. (25K)	3.7%	2.3%	2.3M (2.2%)
Only projs. (50K)	3.9%	2.3%	2.3M (2.2%)

5. CONCLUSIONS

In this work, we proposed to integrate context in the RNN-T model by prompting textual representations to obtain fused multimodal representations. We demonstrated this approach consistently outperforms cross-attention and feature concatenation, while being more cost-effective. We proposed a simple, yet effective, mechanism that re-uses the internal RNN-T embedding projection network to create text representations. This method outperforms representations obtained with a BERT model, and has the additional benefit of incurring in less ops. Our best candidate achieves 5.9% rWERR in a multi-turn test set, without incurring degradations on non-contextual use cases. We also showed our method can achieve comparable results when only the MHSA and projections are fined-tuned.

6. REFERENCES

[1] A. G. Alanis, L. Drude, A. Schwarz, R. V. Swaminathan, and S. Wiesler. Contextual-utterance training for automatic speech recognition. In *iberSPEECH*, 2022.

[2] S. Arora, H. Futami, E. Tsunoo, B. Yan, and S. Watanabe. Joint modelling of spoken language understanding tasks with integrated dialog history. In *ICASSP*, 2023.

[3] F.-J. Chang, T. Muniyappa, K. M. Sathyendra, K. Wei, G. P. Strimel, and R. McGowan. Dialog act guided contextual adapter for personalized speech recognition. In *ICASSP*, 2023.

[4] F.-J. C. Chang, J. Liu, M. Radfar, A. Mouchtaris, M. Omologo, A. Rastrow, and S. Kunzmann. Context-aware transformer transducer for speech recognition. In *ASRU*, 2021.

[5] S.-Y. Chang, C. Zhang, T. N. Sainath, B. Li, and T. Strohman. Context-aware end-to-end asr using self-attentive embedding and tensor fusion. In *ICASSP*, 2023.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, 2018.

[7] A. Graves. Sequence transduction with recurrent neural networks. In *ICML*, Edinburgh, Scotland, 2012.

[8] A. Gulati, C.-C. Chiu, J. Qin, J. Yu, N. Parmar, et al. Conformer: Convolution-augmented transformer for speech recognition. *ArXiv*, 2005.08100, 2020.

[9] Y. He, S. Zheng, Y. Tay, J. Gupta, Y. Du, V. Aribandi, Z. Zhao, Y. Li, Z. Chen, D. Metzler, et al. Hyperprompt: Prompt-based task-conditioning of transformers. In *PMLR*, 2022.

[10] T. Hori, N. Moritz, C. Hori, and J. Le Roux. Advanced long-context end-to-end speech recognition using context-expanded transformers. In *Interspeech*, 2021.

[11] M. Jain, G. Keren, J. Mahadeokar, G. Zweig, F. Metze, and Y. Saraf. Contextual rnn-t for open domain asr. *arXiv preprint arXiv:2006.03411*, 2020.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[13] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li. On the sentence embeddings from pre-trained language models. In *EMNLP*. Association for Computational Linguistics, 2020.

[14] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, and J. Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *ACL*, Dublin, Ireland, 2022.

[15] R. Masumura, N. Makishima, M. Ihori, A. Takashima, T. Tanaka, and S. Orihashi. Hierarchical transformer-based large-context end-to-end asr with large-context knowledge distillation. In *ICASSP*, 2021.

[16] R. Pandey, R. Ren, Q. Luo, J. Liu, A. Rastrow, A. Gandhe, D. Filimonov, G. Strimel, A. Stolcke, and I. Bulyko. Procter: Pronunciation-aware contextual adapter for personalized speech recognition in neural transducers. In *ICASSP*, 2023.

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*. ISCA, 2019.

[18] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao. Deep context: end-to-end contextual speech recognition. In *IEEE SLT*, 2018.

[19] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann. Contextual adapters for personalized speech recognition in neural transducers. In *ICASSP*, 2022.

[20] V. Sunder, S. Thomas, H.-K. J. Kuo, J. Ganhotra, B. Kingsbury, and E. Fosler-Lussier. Towards end-to-end integration of dialog history for improved spoken language understanding. In *ICASSP*.

[21] T. Tran, K. Wei, W. Ruan, R. McGowan, N. Susanj, and G. P. Strimel. Adaptive global-local context fusion for multi-turn spoken language understanding. *AAAI*, 2022.

[22] K. Wei, P. Guo, and N. Jiang. Improving transformer-based conversational asr by inter-sentential attention mechanism. *arXiv preprint arXiv:2207.00883*, 2022.

[23] K. Wei, T. Tran, F.-J. Chang, K. M. Sathyendra, T. Muniyappa, J. Liu, A. Raju, R. McGowan, N. Susanj, A. Rastrow, et al. Attentive contextual carryover for multi-turn end-to-end spoken language understanding. In *ASRU*, 2021.

[24] Y. Zhang, J. Qin, D. S. Park, W. Han, C. Chiu, R. Pang, Q. V. Le, and Y. Wu. Pushing the limits of semi-supervised learning for automatic speech recognition. *CoRR*, 2020.