INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Modality Confidence Aware Training for Robust End-to-End Spoken Language Understanding

Suyoun Kim[1], *Akshat Shrivastava*[1], *Duc Le*[2], *Ju Lin*[1], *Ozlem Kalinli*[1], *Michael L. Seltzer*[1]

[1]Meta, USA
[2]TikTok, USA

suyounkim@meta.com

Abstract

End-to-end (E2E) spoken language understanding (SLU) systems that generate a semantic parse from speech have become more promising recently. This approach uses a single model that utilizes audio and text representations from pre-trained speech recognition models (ASR), and outperforms traditional pipeline SLU systems in on-device streaming scenarios. However, E2E SLU systems still show weakness when text representation quality is low due to ASR transcription errors. To overcome this issue, we propose a novel E2E SLU system that enhances robustness to ASR errors by fusing audio and text representations based on the estimated modality confidence of ASR hypotheses. We introduce two novel techniques: 1) an effective method to encode the quality of ASR hypotheses and 2) an effective approach to integrate them into E2E SLU models. We show accuracy improvements on STOP dataset and share the analysis to demonstrate the effectiveness of our approach.
Index Terms: speech recognition, spoken language understanding

1. Introduction

As voice-driven device interfaces continue to gain popularity in conversational AI, spoken language understanding (SLU) systems that generate semantic parse from speech are becoming increasingly important. Traditional SLU systems are typically composed of two separate components, including automatic speech recognition (ASR) and natural language understanding (NLU). The ASR component generates transcriptions from speech, while the NLU component generates semantic parse from the ASR's output hypotheses. While the pipeline approach allows for individual development of ASR and NLU components, it also has limitations such as being vulnerable to error propagation from ASR to NLU, having limited acoustic information which reduces NLU accuracy, and a lack of parameter sharing that hinders on-device deployment of SLU.

An alternative End-to-End (E2E) approach [1, 2, 3, 4, 5, 6, 7, 8, 9] that directly converts speech into semantics has been recently introduced to address these limitations of pipeline approach. The E2E approach has shown improved performance in domain or intent prediction and slot tagging tasks. However, there are only a few studies on E2E SLU in resource-constrained environments [8, 10, 11].

More recently, the paper [8, 10] proposed a deliberation-based approach to E2E SLU inspired by two-pass E2E ASR [12, 13, 14, 15]. Particularly, [10] shows promising results in resource-constrained on-device environments. However, we found that the model can still experience limitations in cases where the quality of text representation is compromised due to transcription errors from the ASR component. This is especially problematic in on-device streaming use cases where ASR performance is often poor due to limited resources and contextual information.

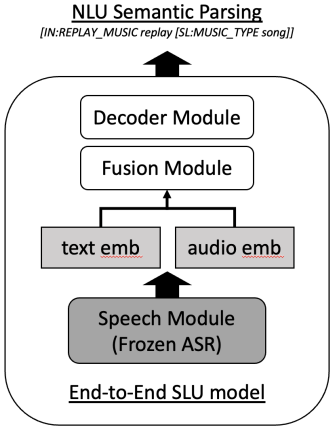

Figure 1: *The overall architecture of End-to-End Spoken Language Understanding model.*

In this work, we propose a new E2E SLU system that improves its robustness to ASR errors by fusing audio and text representations wisely, taking into account the estimated confidence scores of each modality representation. Our approach distinguishes itself from previous studies [10] by its ability to dynamically shift focus towards the audio representation in cases where the ASR output hypotheses contain errors. We present two novel techniques to improve E2E SLU models: 1) a method to encode ASR hypothesis quality and 2) an effective method to integrate these quality information into E2E SLU models. We show accuracy improvements on STOP dataset [16] in the on-device streaming scenario and share the analysis to demonstrate the effectiveness of our approach.

2. Modality Confidence Aware Training

2.1. Deliberation NLU component in E2E SLU

In our proposed Modality Confidence Aware Training (MCAT), we extend the deliberation model [10], which is one of the latest on-device streaming E2E SLU techniques. Unlike pipeline SLU systems with separate ASR and NLU models, the deliberation-based E2E SLU models optimize ASR and NLU components jointly. For on-device streaming, the Recurrent Neural Network Transducer (RNNT) [17, 18, 19, 20, 21, 22] is used as

The work is performed during Duc Le is at Meta.

10.21437/Interspeech.2023-2575

the ASR component. The ASR component is trained separately and then frozen to maintain transcription accuracy. This is because fine-tuning the ASR model on SLU data (which is relatively smaller than ASR training data) may negatively impact its performance on out-of-domain test cases, such as long-form transcription tasks.

The vanilla deliberation-based NLU component consists of two main modules: (1) `Fusion`, and (2) `Decoder` module. Rather than using the RNNT's final hypotheses directly, the `Fusion` module in NLU component takes the intermediate audio and text representation, such as text embeddings $e_{\text{txt}}^{1:U}$ from the `Predictor` of the frozen RNNT and audio embeddings $e_{\text{aud}}^{1:T}$ from the `Encoder` of the frozen RNNT. By using Multi-Head Attention($\texttt{MHA}_{\texttt{fusion}}$), the `Fusion` module generates the fused feature $e_{\text{fused}}^{1:U}$ as follows:

$$e_{\text{fused}} = \texttt{MHA}_{\texttt{fusion}}(e_{\text{txt}}, e_{\text{aud}}, e_{\text{aud}}) \quad (1)$$

Note that we used e_{txt} as a query, e_{aud} as a key and value in `MHA`.

Given the fused feature e_{fused} from the `Fusion` module, the `Decoder` module generates the final target semantic token distributions ($o^{1:V}$) by using transformer-based decoder with *pointer-generator* technique [23, 24]. At each output time step v, (1) the probability of generating a new semantic token (g_v) and (2) the probability of copying a token (c_v) are computed and combined together based on a mixing probability P_{copy}, then the final output token distribution (o_v) is computed. The copying probability (c_v) is computed from the decoder state, and the generating probability (g_v) is computed from the $\texttt{MHA}_{\texttt{dec}}$.

$$g_v = \texttt{Softmax}(\texttt{Linear}(d_v)) \quad (2)$$

$$c_v, a_v = \texttt{MHA}_{\texttt{dec}}(d_v, e_{\text{fused}}, e_{\text{fused}}) \quad (3)$$

$$P_{\text{copy}} = \sigma(\texttt{Linear}([d_v, c_v])) \quad (4)$$

$$o_v = \texttt{Softmax}(P_{\text{copy}} \cdot c_v + (1 - P_{\text{copy}}) \cdot g_v) \quad (5)$$

Figure 1 describes the overall architecture of the deliberation-based E2E SLU systems [10]. While the deliberation E2E SLU models has been shown to mitigate ASR error propagation by leveraging both audio and text representations, we observed that relying solely on audio representation can yield better results when the ASR hypothesis contains errors. This observation suggests that effectively incorporating both audio and text modalities could further improve NLU performance, especially in cases where the text representation is unreliable due to ASR hypothesis errors.

2.2. Integration of Modality Confidence Score

The main idea of our proposed approach, MCAT, is to build the E2E SLU model that can wisely combine audio and text representation based on the confidence level of each modality. For example, when the output of the ASR component is accurate, the NLU component will rely more on the text representation, while the output of the ASR component contains errors, the NLU component will rely more on the audio representation in order to correct those errors.

In this section, we describe our proposed methods to integrate the modality confidence information into the deliberation NLU component described in Section 2.1. We assume we have the modality confidence score (`score`) ranges between 0 to 1 ($0 \leq \texttt{score} \leq 1$). A score closer to 1 indicates that the text modality input is highly reliable, while a score closer to 0 indicates that the audio modality input is highly reliable. We

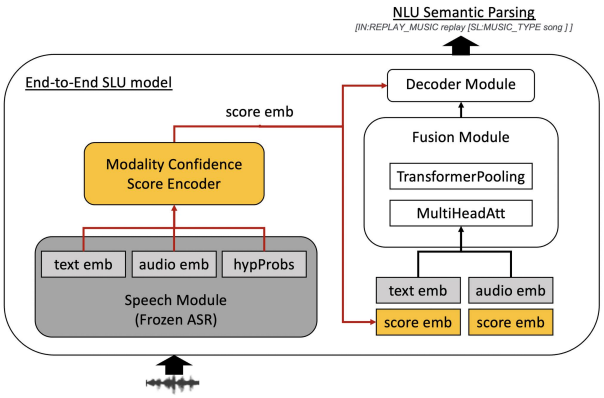

Figure 2: *The overall architecture of our proposed Modality Confidence Aware Training (MCAT).*

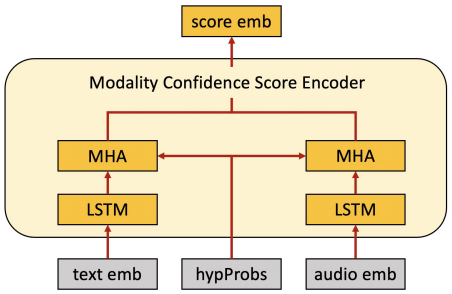

Figure 3: *The overall architecture of our Modality Confidence Score Encoder.*

will describe how to obtain and encode this modality confidence score in Section 2.3.

We explore three different methods to integrate `score` into E2E NLU component in this study. The first method is multiplication. Given the modality confidence score, we multiply it to text embedding directly, and ($1 - \texttt{score}$) to audio embeddings and then forward to the `Fusion` module. The second method is appending the score to text and audio embeddings then forward to the `Fusion` module as follows:

$$e_{\text{txt}}^{\text{MCAT}} = [e_{\text{txt}}, \texttt{score}] \quad (6)$$

$$e_{\text{aud}}^{\text{MCAT}} = [e_{\text{aud}}, \texttt{score}] \quad (7)$$

$$e_{\text{fused}}^{\text{MCAT}} = \texttt{MHA}_{\texttt{fusion}}(e_{\text{txt}}^{\text{MCAT}}, e_{\text{aud}}^{\text{MCAT}}, e_{\text{aud}}^{\text{MCAT}}) \quad (8)$$

Additionally, we can use the score as an additional feature for computing P_{copy} in the `Decoder` module as follows:

$$P_{\text{copy}}^{\text{MCAT}} = \sigma(\texttt{Linear}([d_v, c_v, \texttt{score}])) \quad (9)$$

In Section 4.1, we will present the results on the different score integration methods.

2.3. Modality Confidence Score Encoder

To obtain the modality confidence score (`score`), we build a Score Encoder (`ScoreEncoder`) to generate a single score embedding by utilizing all the available information from the frozen RNNT model. This includes the log probability of the ASR hypothesis (`hypProbs`), text embeddings, and audio embeddings.

$$\texttt{score} = \texttt{ScoreEncoder}(\texttt{hypProbs}, e_{\text{txt}}, e_{\text{aud}}) \quad (10)$$

Since these inputs have different lengths (1, 1:T, 1:U, respectively), we use a simple LSTM and MHA to map those features into a single score embedding. The LSTM for each modality first takes audio or text embeddings to model sequential information and the LSTM output of each modality is combined with hypProbs using a Multi-Head Attention (MHA). Finally, the MHA output of each modality are concatenated to generate a single score embedding.

$$s_{\text{txt}} = \texttt{MHA}(\texttt{hypProbs}, \texttt{LSTM}_{\text{txt}}(e_{\text{txt}}), \texttt{LSTM}_{\text{txt}}(e_{\text{txt}}))$$
$$s_{\text{aud}} = \texttt{MHA}(\texttt{hypProbs}, \texttt{LSTM}_{\text{aud}}(e_{\text{aud}}), \texttt{LSTM}_{\text{aud}}(e_{\text{aud}}))$$
$$\texttt{score} = [s_{\text{aud}}, s_{\text{txt}}]$$

Our ScoreEncoder is trained using error information from hypotheses generated by our frozen RNNT model on the STOP training dataset. We use a simple binary target scheme, where 1 represents a correct ASR hypothesis and 0 represents an ASR hypothesis with an error. We first train the ScoreEncoder, followed by joint training of the entire NLU component with the frozen ScoreEncoder and RNNT. We will present the results of different training methods in Section 4.2. The overall ScoreEncoder architecture is illustrated in Figure 3. We limit the number of parameters in the ScoreEncoder to 0.3M parameters to ensure it is suitable for on-device streaming scenarios.

3. Experiments

3.1. Data

3.1.1. STOP dataset

We used the largest public SLU dataset, STOP (Spoken Task Oriented Semantic Parsing) [16] to evaluate our proposed approach. The STOP dataset is based on Task-Oriented Semantic Parsing (TOPv2) [25], a well-known NLU benchmark, that covers 8 different domains including alarm, messaging, music, navigation, timer, weather, reminder, and event. The spoken data was collected by Amazon Mechanical Turk (MTurk). The dataset split into three subsets: 120k training data, 33k validation data, and 76k evaluation data.

3.2. Models

3.2.1. E2E SLU: Frozen ASR Component

In this study, which targets on-device streaming use cases, we used three different ASR models with relatively smaller sizes: 10M, 15M, and 25M parameters. All of these models were variants of RNNT, a widely-used architecture in streaming use cases. These models have a 1-layer LSTM predictor, a conformer encoder [26, 27] with varying numbers of layers (3L, 6L, 13L for 10M, 15M, 25M models, respectively), and a 1-layer of Joiner. We used a 4-stride, 40ms lookahead, and 120ms segment size audio input features, and 4k of sentence piece targets [28]. Using the Alignment Restricted RNN-T loss [29] and SpecAugment techniques [30], the model was trained on 145k hours of in-house speech data with the same recipe in [31]. The Word Error Rates (WER) for each split of the STOP datasets, using ASRs of three different sizes (10M, 15M, and 25M), are presented in Table 1.

Note that the ASRs were kept frozen and were not fine-tuned with the NLU component for focusing on improving NLU component while maintaining the table transcription performance of the ASRs.

Table 1: *The WER results for each split of the STOP datasets, using the frozen RNNT ASR of three different sizes (10M, 15M, and 25M).*

Frozen ASR (RNNT)	WER train	WER valid	WER test
10M	7.68	6.99	6.54
15M	5.36	4.88	4.59
25M	4.19	3.83	3.54

3.2.2. E2E SLU: NLU Component

For NLU component for on-device streaming use cases, we used 5M parameters of deliberation-based NLU architecture as described in Section 2.1. From the frozen ASR, 256 dimensional audio and text embeddings were passed into the Fusion module. The Fusion module is Multi-Head Attention (MHA) with 8 attention heads and the fused features are then passed into the Pooling module consists of 2 transformer [32] encoder layers with 8 attention heads. The output feature of Pooling is 224 dimensional feature and then finally passed into the Decoder module consistent of a single transformer decoder layer with 2 attention heads with a pointer-generator network with 1 attention head [24]. We used 586 ontology tokens including semantic parse in addition to 4k sentence pieces that used in ASR. The baseline NLU component was trained on STOP dataset with using union strategy, a combination of reference text and hypothesis from ASR with the same recipe in [31].

For our ScoreEncoder, we used a single layer of LSTM with 128 cells for each modality. Initially, we trained the model and subsequently kept it frozen without fine-tuning it with the NLU component.

Our E2E SLU models were evaluated using Exact Match (EM) [25], which measures the accuracy of the model's hypothesis by comparing it to the reference annotation using a string match, while ignoring punctuation and casing. Both the parse structure and slot content transcription need to be matched to be considered correct.

4. Results and Analysis

4.1. How to Incorporate the Modality Confidence Information?

We first investigated the effective way to integrate the confidence information into the deliberation-based NLU component. We compared three different methods to integrate the confidence information: (1) multiplication in the fusion module, (2) appending in the fusion module, and (3) appending in both fusion and decoder module (described in Section 2.2). In this experiment, we used oracle modality confidence score that we defined based on WER as follows:

$$\texttt{score}^{oracle} = 1.0 - min(1.0, \texttt{WER}) \tag{11}$$

Note that $\texttt{score}^{oracle}$ ranges between 0 to 1, and 1 represents a correct ASR hypothesis and 0 represents a hypothesis with an ASR error. As shown in Table 2, all three methods show higher EMs for both the w/ ASR error and w/o ASR error cases compared to the baseline E2E SLU. We also observed similar gains in EM when using either appending or multiplication method in fusion module. However, we found a significant increase in EM

by appending method in decoder module, particularly in ASR error cases.

Table 2: *The EM results for the baseline E2E SLU and three different integration methods of the modality confidence information: (1) MCAT with multiplication method in the fusion module (MUL FUSION), (2) MCAT with appending method in the fusion module (APPEND FUSION), and (3) MCAT with appending method in both fusion (APPEND FUSION) and decoding modules (APPEND DEC).*

Integration Methods	**ASR Error**	
	Yes (61k utters)	No (14k utters)
Our Baseline	84.4	32.3
+ MUL FUSION	84.7	35.5
+ APPEND FUSION	**84.9**	35.4
+ APPEND DEC	83.7	**41.8**

4.2. Building a Modality Confidence Score Encoder

We next investigated what is the best way to encode the modality confidence information. We experiment different strategies to build Modality Confidence Score Encoder. We first observed that using three input resources performed the best (1) text embedding from ASR RNNT predictor, (2) audio embedding from ASR RNNT encoder, and (3) ASR hypothesis probability (`HypProb`). For the objective function to train the Modality Confidence Score Encoder, we tried several options such as classification with weighted class, regression, and focal loss [33]. We found that the binary weighted classification performed the best results. In our experiments, we assigned a class weight of 0.3 to the "1" label, which represents a correct ASR hypothesis, and a class weight of 0.7 to the "0" label.

One of our main challenges in building the Modality Confidence Score Encoder was dealing with unbalanced training data, as our ASR system achieved the high accuracy on the STOP dataset (as shown in Table 1). For example, the target of training data was heavily skewed, the majority of examples having a score of 1. To resolve this problem, we addressed the class imbalance in the training dataset by augmenting the examples with ASR errors ("0" label). We added noise to the audio of the original STOP training data by using Noise Injection technique [34] and generated examples with ASR errors intentionally.

4.3. How good should Modality Confidence Score Encoder perform to improve EM?

We analyzed the minimum performance requirement for the Score Encoder to improve EM. During training the NLU module, we used the binary oracle score and then intentionally introduced errors during decoding by randomly flipping the oracle score ($1 \Longleftrightarrow 0$) in order to simulate estimation from the imperfect Score Encoder. Figure 4 shows he EM results of the baseline and our proposed approach MCAT with different flipping ratios (ranging from 0 to 100%) in corrupted scores. The results showed that the Score Encoder needs to achieve at least 87% accuracy to improve EM compared to the baseline. Interestingly, we observed that when the score was flipped 100% of the time, the model performed significantly worse than the baseline. These result indicates that our MCAT approach functions as intended and its effectiveness is not due to randomness.

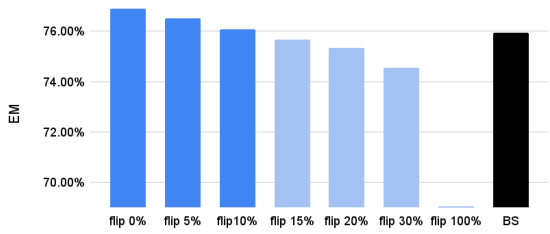

Figure 4: *The EM results of the baseline (BS) and our proposed approach MCAT with different flipping ratios (ranging from 0% to 100%) in corrupted scores. "flip 0%" indicates no flipping, while "flip 100%" indicates 100% flipping.*

4.4. Semantic Parsing Results (EM)

Table 3 shows the EM of (1) the baseline E2E SLU, (2) E2E SLU with Oracle Confidence Score, and (3) our proposed approach, Modality Confidence Aware Training (MCAT) on STOP dataset with varying sizes of the frozen ASR RNNT models (10M, 15M, and 25M parameters). We observed an absolute EM improvement of 0.41, 0.35, and 0.11 with our proposed method when using ASR models of size 10M, 15M, and 25M parameters, respectively. These results suggest that our MCAT incorporating modality confidence information may be particularly beneficial when the ASR model is less accurate, such as on-device streaming use cases. On the other hand, as the ASR performance improves, the potential benefits of using our MCAT may be diminished.

Table 3: *The EM results for three different models: (1) Baseline E2E SLU, (2) Baseline with Oracle Confidence Info, and (3) our proposed model, MCAT, on the STOP dataset with varying sizes of ASR RNNT models (10M, 15M, and 25M parameters), along with the same size of the NLU component (5M parameters).*

	NLU	**Frozen ASR (RNNT)**		
	5M	10M	15M	25M
Our baseline E2E SLU	68.37	71.97	74.05	
w/ oracle confidence Info	69.66	72.95	74.90	
w/ our MCAT	68.78	72.32	74.16	

5. Conclusions

We have introduced our novel approach for building robust end-to-end (E2E) spoken language understanding (SLU) models leverages modality confidence information to intelligently fuse audio and text input representations in the NLU component. The model is designed to prioritize the audio representation when the quality of the text representation is poor due to ASR hypothesis errors. In our experiments on the public STOP dataset with an on-device streaming scenario, our approach MCAT outperformed strong E2E models. Going forward, we plan to extend our method by incorporating token-based modality confidence information and exploring its effectiveness on other datasets with different modalities and scenarios.

6. References

[1] P. Haghani, A. Narayanan, M. A. U. Bacchiani, G. Chuang, N. Gaur, P. J. M. Mengibar, D. Qu, R. Prabhavalkar, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *Proc. SLT*, 2018.

[2] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," 2018.

[3] N. Potdar, A. R. Avila, C. Xing, D. Wang, Y. Cao, and X. Chen, "A streaming end-to-end framework for spoken language understanding," in *IJCAI*, 2021.

[4] M. Radfar, A. Mouchtaris, S. Kunzmann, and A. Rastrow, "Fans: Fusing asr and nlu for on-device slu," in *Interspeech*, 2021.

[5] P. Wang, X. Ye, X. Zhou, J. Xie, and H. Wang, "Speech2slot: An end-to-end knowledge-based slot filling from speech," *ArXiv*, vol. abs/2105.04719, 2021.

[6] M. Rao, A. Raju, P. Dheram, B. Bui, and A. Rastrow, "Speech to semantics: Improve asr and nlu jointly via all-neural interfaces," in *Proc. INTERSPEECH*, 2020.

[7] A. Raju, G. Tiwari, M. Rao, P. Dheram, B. Anderson, Z. Zhang, B. Bui, and A. Rastrow, "End-to-end spoken language understanding using rnn-transducer asr," *arXiv preprint arXiv:2106.15919*, 2021.

[8] S. Arora, S. Dalmia, X. Chang, B. Yan, A. Black, and S. Watanabe, "Two-pass low latency end-to-end spoken language understanding," *Interspeech*, 2022.

[9] D. Xu, S. Dong, C. Wang, S. Kim, Z. Lin, A. Shrivastava, S.-W. Li, L.-H. Tseng, A. Baevski, G.-T. Lin *et al.*, "Introducing semantics into speech encoders," *arXiv preprint arXiv:2211.08402*, 2022.

[10] D. Le, A. Shrivastava, P. Tomasello, S. Kim, A. Livshits, O. Kalinli, and M. L. Seltzer, "Deliberation model for on-device spoken language understanding," *Interspeech*, 2022.

[11] T. Desot, F. Portet, and M. Vacher, "End-to-end spoken language understanding: Performance analyses of a voice command task in a low resource setting," *Computer Speech & Language*, vol. 75, p. 101369, 2022.

[12] T. N. Sainath, R. Pang, D. Rybach, Y. He, R. Prabhavalkar, W. Li, M. Visontai, Q. Liang, T. Strohman, Y. Wu *et al.*, "Two-pass end-to-end speech recognition," *arXiv preprint arXiv:1908.10992*, 2019.

[13] W. Li, J. Qin, C.-C. Chiu, R. Pang, and Y. He, "Parallel rescoring with transformer for streaming on-device speech recognition," *arXiv preprint arXiv:2008.13093*, 2020.

[14] L. Xu, Y. Gu, J. Kolehmainen, H. Khan, A. Gandhe, A. Rastrow, A. Stolcke, and I. Bulyko, "Rescorebert: Discriminative speech recognition rescoring with bert," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6117–6121.

[15] S. Kim, K. Li, L. Kabela, R. Huang, J. Zhu, O. Kalinli, and D. Le, "Joint audio/text training for transformer rescorer of streaming speech recognition," *EMNLP*, 2022.

[16] P. Tomasello, A. Shrivastava, D. Lazar, P.-C. Hsu, D. Le, A. Sagar, A. Elkahky, J. Copet, W.-N. Hsu, Y. Mordechay, R. Algayres, T. A. Nguyen, E. Dupoux, L. Zettlemoyer, and A. Mohamed, "STOP: A dataset for Spoken Task Oriented Semantic Parsing," in *CoRR*.

[17] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[18] R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson, and N. Jaitly, "A Comparison of Sequence-to-Sequence Models for Speech Recognition," in *Interspeech*, 2017, pp. 939–943.

[19] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *ASRU*. IEEE, 2017, pp. 206–213.

[20] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*. IEEE, 2019, pp. 6381–6385.

[21] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving rnn transducer modeling for end-to-end speech recognition," in *ASRU*. IEEE, 2019, pp. 114–121.

[22] S. Kim, Y. Shangguan, J. Mahadeokar, A. Bruguier, C. Fuegen, M. L. Seltzer, and D. Le, "Improved neural language model fusion for streaming recurrent neural network transducer," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7333–7337.

[23] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[24] A. Aghajanyan, J. Maillard, A. Shrivastava, K. Diedrick, M. Haeger, H. Li, Y. Mehdad, V. Stoyanov, A. Kumar, M. Lewis, and S. Gupta, "Conversational Semantic Parsing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2020.

[25] X. Chen, A. Ghoshal, Y. Mehdad, L. Zettlemoyer, and S. Gupta, "Low-Resource Domain Adaptation for Compositional Task-Oriented Semantic Parsing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[26] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[27] Y. Shi, C. Wu, D. Wang, A. Xiao, J. Mahadeokar, X. Zhang, C. Liu, K. Li, Y. Shangguan, V. Nagaraja *et al.*, "Streaming transformer transducer based speech recognition using non-causal convolution," *Proc. ICASSP*, 2022.

[28] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing," in *Proc. EMNLP: System Demonstrations*, 2018.

[29] J. Mahadeokar, Y. Shangguan, D. Le, G. Keren, H. Su, T. Le, C. Yeh, C. Fuegen, and M. L. Seltzer, "Alignment Restricted Streaming Recurrent Neural Network Transducer," in *Proc. SLT*, 2021.

[30] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[31] D. Le, M. Jain, G. Keren, S. Kim, Y. Shi, J. Mahadeokar, J. Chan, Y. Shangguan, C. Fuegen, O. Kalinli, Y. Saraf, and M. L. Seltzer, "Contextualized Streaming End-to-End Speech Recognition with Trie-Based Deep Biasing and Shallow Fusion," in *Proc. Interspeech*, 2021, pp. 1772–1776.

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

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[34] C. K. Reddy, H. Dubey, K. Koishida, A. Nair, V. Gopal, R. Cutler, S. Braun, H. Gamper, R. Aichner, and S. Srinivasan, "Interspeech 2021 deep noise suppression challenge," *arXiv preprint arXiv:2101.01902*, 2021.