Listen with Intent: Improving Speech Recognition with Audio-to-Intent Front-End

Swayambhu Nath Ray[2⋆], *Minhua Wu*[1⋆], *Anirudh Raju*[1⋆], *Pegah Ghahremani*[1⋆], *Raghavendra Bilgi*[2⋆], *Milind Rao*[1], *Harish Arsikere*[2], *Ariya Rastrow*[1], *Andreas Stolcke*[1], *Jasha Droppo*[1]

[1]Amazon Alexa, USA [2]Amazon Alexa, India

{swayar, wuminhua, pegahgh, ranirudh, rrbilgi}@amazon.com

Abstract

Comprehending the overall intent of an utterance helps a listener recognize the individual words spoken. Inspired by this fact, we perform a novel study of the impact of explicitly incorporating intent representations as additional information to improve a recurrent neural network-transducer (RNN-T) based automatic speech recognition (ASR) system. An audio-to-intent (A2I) model encodes the intent of the utterance in the form of embeddings or posteriors, and these are used as auxiliary inputs for RNN-T training and inference. Experimenting with a 50k-hour far-field English speech corpus, this study shows that when running the system in *non-streaming* mode, where intent representation is extracted from the entire utterance and then used to bias streaming RNN-T search from the start, it provides a 5.56% relative word error rate reduction (WERR). On the other hand, a *streaming* system using per-frame intent posteriors as extra inputs for the RNN-T ASR system yields a 3.33% relative WERR. A further detailed analysis of the streaming system indicates that our proposed method brings especially good gain on media-playing related intents (e.g. 9.12% relative WERR on PlayMusicIntent).

Index Terms: End-to-end speech recognition, RNN-T, audio-to-intent, spoken language understanding

1. Introduction

Spoken language understanding (SLU) systems are conventionally designed as a pipeline that includes an automatic speech recognition (ASR) system that converts speech to text, followed by a natural language understanding (NLU) system that extracts structured data such as domain, intent and slots.

For the ASR system, end-to-end models have gained popularity in recent years as they combine separate components of conventional DNN-HMM hybrid ASR systems [1] (acoustic, pronunciation and language models) into a single neural network. End-to-end models include connectionist temporal classification [2], recurrent neural network-transducer (RNN-T) [3], and attention-based sequence-to-sequence models [4–6] also known as LAS: Listen, Attend and Spell [7]. Among these three methods, RNN-T is replacing the traditional hybrid ASR models [8, 9] since it has good streaming capability which is challenging to LAS and does not have CTC's frame-independence assumption. Various directions have been explored to enhance RNN-T ASR performance. Depth-LSTM and layer normalization was tried in [10]. Using LAS as a second-pass rescorer by attending to both encoder features and n-best output from the RNN-T has been explored in [9], and an inter encoder-decoder attention mechanism was introduced in [11] to better align the encoder feature with the hypothesis. Other improvements include better initialization methods, training on TTS data, and use of lookahead encoders [12]. All these improvements on the RNN-T ASR system so far have focused on better acoustic or language modeling and rescoring extensions without adding any capability of understanding to the system.

When humans process speech signals, transcription and understanding happen simultaneously, and the capability of understanding enables human to usually provide better quality of transcription than machines. Training an end-to-end SLU system to predict intent and slot values directly from audio is therefore becoming a popular research area [13–19]. While such system may not readily outperform or replace large-scale conventional SLU systems with independently optimized ASR and NLU modules, it is indicated that semantic information such as intent and slots could potentially help improve an ASR system [16, 19]. However, these approaches were incorporating semantic information implicitly by using training intent and slot prediction as extra tasks. The technique to explicitly incorporate various contextual signals analogous to intent, such as dialog state and music play state, into an RNN-T based ASR system has been proposed in [20, 21], but most of these contextual signals are derived only after the first turn of the dialog and would only benefit subsequent turns. Therefore, extra studies are needed to investigate the impact of using semantic embeddings produced on-the-fly directly with input audio features on an ASR system. Focusing on intent-based semantic embeddings, our contributions in this study are as follows:

- We propose to incorporate intent embeddings of the audio into the RNN-T ASR system to improve its recognition accuracy through an auxiliary audio-to-intent (A2I) front-end.
- We run extensive experiments on a large corpus of 50k hours of far-field US English speech to demonstrate effectiveness of the proposed approach.

To the best of our knowledge, this is the first study to enhance RNN-T ASR performance using intent representations produced on-the-fly from an auxiliary intent prediction model.

2. Method

2.1. RNN-T ASR using intent representations

We train a new RNN-T ASR system where the input features for the encoder contain frames of original audio feature vectors along with intent representations obtained by feeding the audio frames into a pre-trained audio-to-intent (A2I) model which will be described with more details in Section 2.2. For results described in this paper, we incorporate intent representations only into the encoder portion of the RNN-T, since our preliminary experiments showed that feeding extra embeddings to the

* Equal contribution

prediction network is less effective. (This aligns with the findings in [22] that RNN-T encoder makes better use of contextual information than the prediction network).

Specifically, we concatenate each frame of input audio features (x_t) with its intent embeddings (e_t) (as illustrated in Figure 1) to train the proposed RNN-T ASR model. The A2I model is pre-trained from a much smaller intent-annotated corpus, and its parameters are not updated during RNN-T ASR model training. For a streaming ASR system, per-frame intent embeddings (e_t) are required to be concatenated with each frame of input audio features (x_t). In addition, we conduct an experiment where the whole-utterance intent embeddings from the final frame (e_T) are repeatedly concatenated with all the acoustic feature frames. This serves as an upper bound to the performance of a streaming RNN-T system using an auxiliary A2I model.

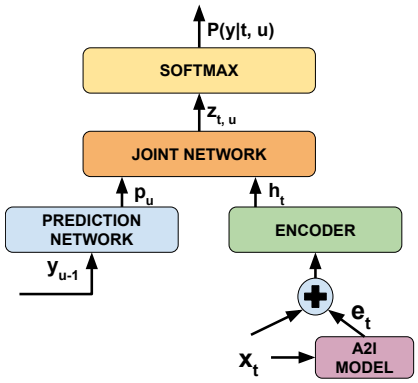


Figure 1: *Incorporate intent embeddings into RNN-T ASR in a streaming fashion. Audio features at each frame (x_t) are concatenated with per-frame intent embeddings (e_t) inferred from a pretrained A2I model.*

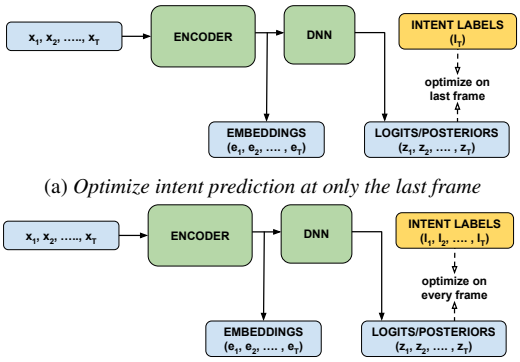
2.2. Audio-to-intent model

Using a separate intent-annotated corpus where each utterance is manually annotated with an intent label, an audio-to-intent (A2I) model can be pre-trained in a supervised manner to classify intent directly from audio input. We explored two A2I models in this work, architectures of which are illustrated in Figure 2. In model 2a, frames of audio features (x_t) are first encoded into frames of acoustic embeddings (e_t) using an LSTM encoder. They are subsequently passed to a dense layer, and intent prediction for the entire utterance is optimized based on posteriors at the last frame. The final-frame intent embeddings e_T can then be concatenated repeatedly with all acoustic feature frames as a non-streaming solution to incorporate intent representation for ASR model building. Model 2b is similar to 2a, except that intent prediction is optimized for each audio frame. The per-frame intent label is the same as the whole-utterance intent label when training model 2b. Per-frame intent embeddings e_t from this model can then be concatenated with each frame of input audio features in a streaming fashion to train our proposed RNN-T ASR model.

3. Data and experimental setup

3.1. Datasets

We use two far-field US English speech datasets in this work to train the A2I and ASR models respectively. The speech data used for training and evaluation are de-identified and based on queries to smart speakers.



(a) *Optimize intent prediction at only the last frame*

(b) *Optimize intent prediction at every frame; per-frame intent label is the same as the whole-utterance intent label.*

Figure 2: *Architectures of A2I models*

1. *16k-hour SLU dataset*: Human annotations of NLU intent are available for all utterances. In addition, a development set of 96 hours of data is available. We use this dataset to train the A2I model.
2. *50k-hour ASR dataset*: Human transcriptions are available for all utterances. In addition, a test set of approximately 200 hours is available. We use this dataset to train both the baseline and the proposed RNN-T ASR system. Note that 64% of this dataset contains human intent annotations, which we will use for the partial oracle experiment in Section 4.2.1.

3.2. Models

3.2.1. RNN-T ASR

The baseline RNN-T ASR model is trained on the 50k-hour ASR dataset. The encoder consists of 5 LSTM layers of 1024 hidden units, with a final layer output dimension of 4001. The prediction network has an input embedding layer of 512 units, 2 LSTM layers of 1024 units, and a final output dimension of 4001. The joint network adds the outputs from the encoder and prediction networks as in [3]. These outputs of size 4001 are softmax normalized and correspond to subword tokens of the same vocabulary size. The subword vocabulary was generated using the byte pair encoding algorithm [23].

3.2.2. A2I models

A2I models are trained on the 16k-hour SLU dataset with human-annotated intent labels. The model is trained to predict 64 intents, including the 63 most frequent intents and an "Other" class that covers all remaining intents. Our A2I models, illustrated in Figure 2, use an encoder of 2 LSTM layers with 512 units. The second LSTM layer has a projection layer generating an embedded representation of dimension 64. The projected representation then goes through an extra feed-forward layer of 64 units followed by softmax normalization for the final intent prediction.

3.2.3. RNN-T ASR using intent representations

In general, we concatenate every original frame of audio features with an intent representation as illustrated in Figure 1 to train our proposed RNN-T ASR model on the 50k-hour ASR dataset. The intent representations are inferred from the pretrained A2I model, and we freeze weights in the A2I model during the ASR model training. Note that we do not need NLU annotations for ASR training at this stage as we already have

a pretrained A2I model. Thus, our proposed method does not impose any restriction on RNN-T ASR training data.

Since 64% of the ASR dataset has human-annotated intent labels, we also conduct a partial oracle experiment (Section 4.2.1) by concatenating either a one-hot representation or an all-zero vector with each frame of input audio features, depending on whether human-annotated intent label for that audio is available.

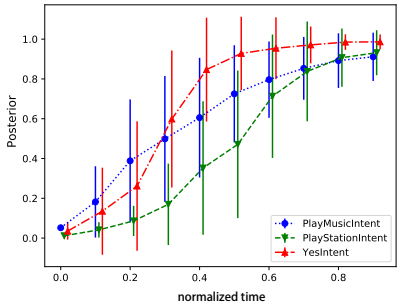
The audio features used by both A2I and ASR models are 64-dimensional log filter bank energy features computed over a 25ms window with 10ms shift. Each feature vector is stacked with 2 frames to the left and downsampled to a 30ms frame rate [24]. We use SpecAugment [25] during RNN-T training to improve model robustness. All models are trained using the Adam optimizer [26], with a learning rate schedule including an initial linear warm-up phase, a constant phase, and an exponential decay phase following [27]. These hyper-parameters are not specifically tuned for this work.

4. Results and discussion

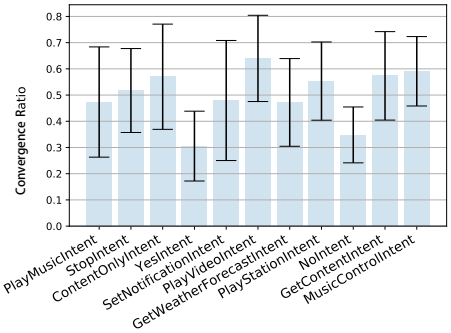
4.1. Audio-to-intent accuracy

Table 1: *Audio to intent prediction accuracy on 150k samples.*

A2I model optimized on	intent prediction accuracy (%)			
	train		dev	
	per-frame	per-utt	per-frame	per-utt
last frame	47.97	88.92	47.37	87.70
every frame	60.47	87.16	59.81	86.98



(a) *Mean and standard deviation of posterior over the normalized time for selected intents.*



(b) *Mean and standard deviation of convergence ratio for selected intents. The convergence ratio is defined as the normalized time when 80% of the final-frame intent posterior is achieved.*

Figure 3: *Per-intent posterior analysis for the A2I model optimized on every frame.*

Table 1 shows the intent prediction accuracy of the two A2I models we have trained. Per-utterance accuracy is calculated based on prediction at the last frame comparing with the whole-utterance NLU intent label from human annotator. Per-frame accuracy is computed based on prediction at every frame and uses the same NLU intent label for all frames. Per-frame prediction accuracy is relatively low because intent cannot be predicted reliably in the initial portion of an utterance, but integrating frame-wise intent representation into an ASR system makes the solution streamable.

With the A2I model optimized on every frame, Figure 3a shows mean and standard deviation of the posterior over the normalized time value (i.e., frame index to total length of utterance). For all utterances with the same predicted final intent, we compute mean and standard deviation of the posteriors at different normalized time stamps. Figure 3b shows the mean and standard deviation of the convergence ratio. This convergence ratio is defined as the ratio of the selected frame index to the total length of utterance, where the selected frame index is the first frame that achieves 80% of the final-frame intent posterior. These statistics are computed over all utterances with correct intent label prediction. The posterior values over time for all utterances are smoothed using a convolution filter with window size 4. These figures show that the confidence of the A2I model increases with more frames and the A2I model can generally predict the true intent from the initial 70% of the utterance.

4.2. RNNT-ASR using intent representations

4.2.1. Using human-annotated intent labels

We conduct a partial oracle experiment by training an RNN-T ASR model with the partially available human-annotated intents in the 50k-hour ASR training dataset. The one-hot intent representation for each frame is concatenated with the original LFBE features as new input to the encoder. We use all-zero vectors when human intent annotation is not available for a given utterance. We believe this acts as a regularizer helping the model to not depend entirely on intent information. However, results reported from this experiment will not reflect the lowest achievable WER since only 64% of the training dataset contains human intent annotation. Rather, they serve to verify the hypothesis that injecting intent information explicitly into an ASR system can help achieve lower WER. Column 3 of Table 2 reports relative word error rate reduction (WERR) for different intents by using human-annotated intent labels as extra input features. Note that, there is a "None" entry since not all utterances have human annotation in the training and test sets, and its relative WERR reduction is relatively small (1.46% in the table). This is because all-zero vectors are provided as auxiliary inputs for these utterances, which does not provide additional intent-based information for the ASR model to constrain the search space. For test utterances with human-annotated intents, we observe good relative WERRs, especially for those that have limited word choices in utterances (e.g. YesIntent, NoIntent, StopIntent).

4.2.2. Using Intent Representations from the A2I Model

For the real use case, we need an A2I model that infers intent representations from audio to help train the ASR model and run the inference, since human-annotated intent labels are not available when decoding input audio after model deployment.

Given the A2I model optimized with last-frame prediction (Figure 2a), we concatenate repeated final frame embeddings (e_T) with the original frames of acoustic features (x_t) as new inputs for the encoder of the proposed RNN-T ASR model (row 3 in Table 3). Since per-frame embeddings (e_t) are also available from this A2I model, we concatenate them with the original frames of acoustic features as an extra experiment (row 4 in

Table 2: *Per-intent relative WERR (%) by using different intent representations (compared to RNN-T ASR baseline).*

Intent (most frequent from test set)	# utt	RNN-T + human annotation (one-hot)	RNN-T + A2I (repeated final embeddings)	RNN-T + A2I (per-frame posteriors)
None (human annotation not available)	100.2k	1.46	4.88	3.61
PlayMusicIntent	14.0k	4.89	9.34	9.12
StopIntent	8.9k	23.69	4.85	2.52
ContentOnlyIntent	5.4k	14.85	6.94	3.45
YesIntent	3.6k	29.31	11.50	-4.63
SetNotificationIntent	3.5k	8.31	4.00	-3.69
PlayVideoIntent	3.1k	8.38	7.77	7.93
GetWeatherForecastIntent	2.8k	5.95	8.63	4.17
PlayStationIntent	2.7k	4.67	7.65	9.21
NoIntent	2.6k	36.88	13.66	3.26
GetContentIntent	2.4k	1.52	-0.25	-5.32
MusicControlIntent	2.2k	24.68	3.34	-7.97

Table 3: *Relative WERR (%) by using different intent representations (compared to RNN-T ASR baseline).*

No.	Exp.	A2I model optimized on	Intent representation	Streamable	relative WERR(%)	#params
1	RNN-T (baseline)	-	-	yes	-	63.5M
2	RNN-T (larger)	-	-	yes	1.91	67.3M
3	RNN-T + A2I	last frame (Fig. 2a)	repeated final-frame embeddings (e_T)	no	*5.56*	67.1M
4	RNN-T + A2I	last frame (Fig. 2a)	per-frame embeddings (e_t)	yes	*2.78*	67.1M
5	RNN-T + A2I	every frame (Fig. 2b)	per-frame embeddings (e_t)	yes	*2.89*	67.1M
6	RNN-T + A2I	every frame (Fig. 2b)	per-frame posteriors (z_t)	yes	*3.33*	67.1M

Table 3). Given the A2I model optimized with frame-wise prediction (Figure 2b), we concatenate frame-wise embeddings or posteriors (e_t or z_t) with the original frames of acoustic features (x_t) as inputs for the encoder of the RNN-T ASR model (rows 5,6 in Table 3). Using per-frame embeddings along with the audio feature vectors makes the resulting system fully streamable, while using repeated final embeddings indicates the largest gain achievable by using intent representation as auxiliary input.

Overall, incorporating repeated final-frame embeddings yields a 5.56% relative WERR while feeding per-frame posteriors in a streaming fashion provides a 3.33% relative WERR compared to the RNN-T ASR baseline (row 1 in Table 3). When comparing different intent representations, we see that using per-frame posteriors gives a slightly better relative WERR (3.33%) compared to using per-frame embeddings (2.89%). We have also trained a larger RNN-T ASR baseline (row 2 in Table 3) with its number of parameters comparable to our proposed RNN-T ASR system using an auxiliary A2I model. This larger baseline only yields a 1.91% relative WERR, which demonstrates that the gain of our proposed method by using the auxiliary A2I model does not come only from the enlarged parameter space.

For the case of using repeated final frame embeddings and per-frame posteriors, we also break down the relative WERR under different intents as shown in Table 2. Overall, we see gains on various intents. For the streaming system using per-frame posteriors, which could be applied in a real-time recognition system, we see good relative WERRs in media-playing related intents (9.12% relative WERR on PlayMusicIntent, 7.93% relative WERR on PlayVideoIntent and 9.21% relative WERR on PlayStationIntent). For the unannotated partition (intent "None"), we see that the RNN-T with A2I model outperforms the system using one-hot representations based on human annotation, which indicates that the A2I model can learn reasonable representations for these data while the one-hot model just treats this case as out-of-domain. This establishes the advantage of using the auxiliary A2I model since it does not need human intent annotation during ASR training and inference. Compared with the non-streamable RNN-T ASR system using repeated final-frame intent embeddings, it seems the streamable RNN-T ASR system using frame-wise posteriors preserves recognition accuracy on intents with longer utterances, such as PlayVideoIntent (3.13k ms on average) and PlayMusicIntent (3.07k ms on average), while its performance degrades on intents with relatively shorter utterances, such as YesIntent (1.37k ms on average) and StopIntent (1.30k ms on average).

Considering the WER gap between the non-streaming system and the streaming system, we perform a diagnostic experiment which feeds per-frame embeddings e_t for a certain percentage of frames in the utterance at the beginning and uses the final-frame embeddings e_T repeatedly afterwards. As shown in Table 4, feeding the final-frame intent embeddings after 70% of the audio only results in slight degradation compared to feeding final-frame intent embeddings from the start of an utterance. This indicates that intent information tends to improve the ASR system more towards the end of the utterance.

Table 4: *Relative WERR (%) by feeding per-frame embeddings e_t for a certain percentage of frames in the utterance at the beginning and using the final-frame embeddings e_T repeatedly afterwards.*

percent (%) of frames in the utterance after which e_T is fed	relative WERR (%)
0	*5.56*
50	*5.44*
70	*5.00*
100	*2.78*

5. Conclusions

We have demonstrated the benefit of incorporating intent representations for improving end-to-end ASR system with RNN-T. We showed that an audio-to-intent (A2I) auxiliary model is capable of predicting intent representations that substantially help lower ASR WER. Over all intent classes, incorporating repeated whole-utterance intent representation in a non-streaming fashion gives a 5.56% relative WERR, while feeding per-frame intent posteriors in a streaming fashion brings a 3.33% relative WERR. In the future, we hope to investigate the effectiveness of our approach as a function of NLU and ASR data size, as well as incorporating other types of auxiliary information.

6. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

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

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[5] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[6] S. N. Ray, S. S. Dasgupta, and P. Talukdar, "Ad3: Attentive deep document dater," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1871–1880.

[7] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[8] 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 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[9] 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.

[10] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 114–121.

[11] B. Wang, Y. Yin, and H. Lin, "Attention-based transducer for online speech recognition," *arXiv preprint arXiv:2005.08497*, 2020.

[12] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao *et al.*, "Developing RNN-T models surpassing high-performance hybrid models with customization capability," *arXiv preprint arXiv:2007.15188*, 2020.

[13] Y. Qian, R. Ubale, V. Ramanaryanan, P. Lange, D. Suendermann-Oeft, K. Evanini, and E. Tsuprun, "Exploring ASR-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 569–576.

[14] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5754–5758.

[15] Y.-P. Chen, R. Price, and S. Bangalore, "Spoken language understanding without speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6189–6193.

[16] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 720–726.

[17] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *arXiv preprint arXiv:1904.03670*, 2019.

[18] M. Rao, A. Raju, P. Dheram, B. Bui, and A. Rastrow, "Speech to semantics: Improve ASR and NLU jointly via all-neural interfaces," in *InterSpeech*. ISCA, 2020.

[19] M. Rao, P. Dheram, G. Tiwari, A. Raju, J. Droppo, A. Rastrow, and A. Stolcke, "Do as I mean, not as I say: Sequence loss training for spoken language understanding," *To appear in ICASSP, arXiv preprint arXiv:2102.06750*, 2021.

[20] Z. Wu, B. Li, Y. Zhang, P. S. Aleksic, and T. N. Sainath, "Multi-state encoding with end-to-end speech RNN transducer network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7819–7823.

[21] S. N. Ray, S. Mitra, R. Bilgi, and S. Garimella, "Improving RNN-T ASR performance with date-time and location awareness," *arXiv preprint arXiv:2106.06183*, 2021.

[22] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, "RNN-Transducer with stateless prediction network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7049–7053.

[23] Y. Shibata, T. Kida, S. Fukamachi, M. Takeda, A. Shinohara, T. Shinohara, and S. Arikawa, "Byte pair encoding: A text compression scheme that accelerates pattern matching," Technical Report DOI-TR-161, Department of Informatics, Kyushu University, Tech. Rep., 1999.

[24] G. Pundak and T. Sainath, "Lower frame rate neural network acoustic models," 2016.

[25] 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.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen *et al.*, "The best of both worlds: Combining recent advances in neural machine translation," *arXiv preprint arXiv:1804.09849*, 2018.