INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Improving the response timing estimation for spoken dialogue systems by reducing the effect of speech recognition delay

Jin Sakuma[1], *Shinya Fujie*[2,1], *Huaibo Zhao*[1], *Tetsunori Kobayashi*[1]

[1]Waseda University, Japan
[2]Chiba Institute of Technology, Japan

jsakuma@pcl.cs.waseda.ac.jp, zhao@pcl.cs.waseda.ac.jp, fujie@pcl.cs.waseda.ac.jp, koba@pcl.cs.waseda.ac.jp

Abstract

In conversational systems, the proper timing of the system's response is critical to maintaining a comfortable conversation. To achieve appropriate timing estimation, it is important to know what the users have said, including their most recent words, but ASR delay usually prevents the use of full user utterance. In this paper, we attempted to employ an extremely low latency ASR model called Multi-Look-Ahead ASR by Zhao et al. to enable near full utterance for response timing estimation. Additionally, we examined the effectiveness of using low latency ASR in combination with a parameter called Estimates of Syntactic Completeness (ESC), which indicates how soon the user's speech is completed. We evaluated on a Japanese simulated dialog database of a restaurant information center. The results confirmed that reducing ASR delay improves the accuracy of response timing estimation. This effect also appeared when the method using ESC is combined with the use of low latency ASR.

Index Terms: spoken dialog systems, turn-taking, response timing, streaming ASR

1. Introduction

By introducing an extremely low-latency ASR and combining it with the prediction of speech completion, we attempt to improve the accuracy of the response timing estimation where quick responses are required in conversation.

In conversational systems, it is important to determine the appropriate timing of the system's response. In particular, if the response timing is slower than expected, it may not only stall the conversation or lead to collisions of utterances, but may even convey an unwilling, hesitant, backward message.

As already pointed out in much of the literature [1, 2, 3, 4, 5], the appropriate response timing is context-dependent. Therefore, appropriate timing estimation needs past ASR results. Depending on what this process is synchronized with, response timing estimation methods can be classified into two types. One is the end of utterance (EoU) synchronous timing detection (ES-TD), and the other is the frame synchronous one (FS-TD).

Many conversational systems [6, 7, 8, 9, 10, 11, 12, 13] employ ES-TD for ease of implementation. In this approach, after detecting an EoU, ASR is performed. Based on the recognition results, the system decides whether the user keeps or releases the turn, and if it decides the turn is released, it starts speaking after a certain time. In many cases, EoU detection itself is performed in a context-independent manner. Therefore, accurate EoU detection requires waiting for a stable pause, and a certain amount of time is necessary to output ASR results after this detection. In the data we have, half of the human response timing is less than 350 ms, but EoU detection and ASR generally require more than 350 ms. ES-TD based methods, which wait for EoU detection to determine keeping/releasing, is unlikely to achieve comfortable turn-taking as humans do.

On the other hand, FS-TD framework [14, 15, 16, 17, 18, 19, 20, 21] synchronizes with analysis frames at regular intervals and decides whether or not to start speech. The decision is made based on a combination of the linguistic features, instantaneous word sequence output by the streaming ASR, and the acoustic features such as prosody-related information. This approach does not explicitly use EoU detection, so they enable quicker responses than ES-TD. In addition, more fine-grained timing controls are possible depending on the context. Thus, this approach is thought to be more promising at this stage.

In previous work, we have introduced the language-model-based following word-sequence prediction in the FS-TD framework [21]. This method adds a new feature, named estimates of syntactic completeness (ESC), to the response timing estimation in addition to the commonly used linguistic and prosodic features. ESC is a feature obtained by estimating how many more words are needed to syntactically complete an utterance. Humans also time their utterances while predicting whether or not the conversational partner is about to finish speaking. This method incorporates this in a natural way into FS-TD framework.

Our previous method successfully improved the accuracy of response timing estimation, however in his method, and also in FS-TD in general, the delay in ASR causes several problems. The streaming ASR model generally needs to look ahead to a segment of speech in order to improve accuracy. In other words, to obtain the recognition result at a certain time point, it is necessary to wait for the input of the segment of speech to be looked ahead. This means that there is a speech segment that has already been spoken but is not reflected in the linguistic feature, the recognition result. In a general FS-TD framework, the ASR delay prevents the use of language information that should be available. Furthermore, in our previous method, the problem of shortening the available ASR results due to this delay may also affect the estimation of ESC.

In this study, we attempt to increase the linguistic information available for FS-TD framework by introducing a low latency ASR. In addition, we aim to improve the accuracy of ESC estimation [21] and further improve the accuracy of response timing estimation. As a low-latency ASR model, we employ the Multi-Look-Ahead ASR [22] which consists of an encoder with a long look-ahead to improve accuracy and an encoder with a zero look-ahead to reduce delay. Thanks to the part of the zero-look-ahead encoder, the ASR delay can be dramatically reduced without performance degradation.

The rest of the paper is organized as follows. Section 2

10.21437/Interspeech.2023-1618

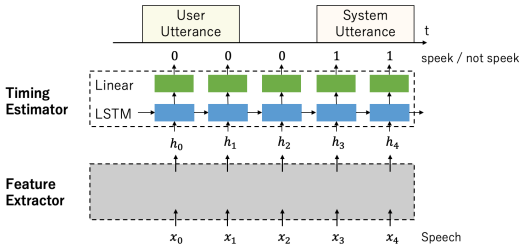

Figure 1: *Overall architecture of our proposed system.*

introduces our proposed response timing estimation model. In Section 3, we describe architecture of the ASR model and its decoding algorithm. In Section 4, we examine the effectiveness of the low latency ASR model through response timing estimation experiments and analyze the results. Finally, Section 5 concludes this paper.

2. Response Timing Estimation Model

Figure 1 shows an overview of our proposed system. It consists of feature extractors and a timing estimator.

2.1. Feature Extractor

The system has feature extractors that extract prosodic, temporal, linguistic features and also features related to syntactic completeness.

Prosodic features. To extract prosodic features, we use the method of Yokoyama et. al. [23]. Their method reconstructs narrowband spectrograms of 100ms speech signals by CNN-Autoencoder and extracts 128-dimensional bottleneck feature vectors as features representing the rising and falling patterns of the fundamental frequency. We feed them to the LSTM [24] and linear projection layers to obtain prosodic features.

Temporal features. We use the elapsed time after detecting the EoU as temporal features. We estimate whether the user is speaking or not using a LSTM-based pre-trained voice activity detection (VAD) model. As the input features, we use the bottleneck feature vectors of the CNN-Autoencoder. The cumulative sum of the probabilities of not speaking at each frame is calculated and used as the information representing the elapsed time after detecting the EoU. This value is reset to 0 when the user starts speaking. We also use the length of the previous utterance as temporal information. We apply a linear projection layer to these features and obtain temporal feature.

Linguistic features. First, the text of the utterance is obtained by streaming ASR. The proposed system uses MLA CBS-T, while a SLA CBS-T is used in the baseline system for comparison. The details of ASR are explained in Section 3. As the input of ASR, we use 80-dimensional log-mel spectrogram features and 3-dimensional pitch features. Then, an Transformer [25] encoder is applied to encode the text and obtain linguistic features.

Estimates of syntactic completeness (ESC) [21]**.**

Using results from the streaming ASR, the language model calculates the probability of EoU appearing $K(1, \ldots, M)$ tokens ahead. For each tokens, it generates N candidates. A feature representing ESC is a vector obtained by arranging these K values. ESC is effective especially to estimate quick response timings as it represents whether the current utterance will be finished in the near future.

Algorithm 1 Beam search for Multi-look ahead ASR

$T^p = T_{N_l} + T_{N_c} + T_{N_r}$ ▷ primary encoder block
$T^a = T_{N_l} + T_{N_c}$ ▷ auxiliary encoder block
$\mathbf{B}^p \leftarrow \emptyset, \mathbf{B}^a \leftarrow \emptyset$
$(\mathbf{H}^p, \mathbf{H}^a) \leftarrow$ Init() ▷ model state
for t = T_B^p to T by T_B^p **do**
  $\mathbf{E}^p \leftarrow$ PrimaryEncoder($X[t - T^p, t]$)
  $(\mathbf{B}^p, \mathbf{H}^p) \leftarrow$ BeamSearch($\mathbf{B}^p, \mathbf{E}^p, \mathbf{H}^p$)
  $\mathbf{E}^a \leftarrow$ AuxiliaryEncoder($X[t - T^a, t]$)
  $(\mathbf{B}^a, \mathbf{H}^a) \leftarrow$ BeamSearch($\mathbf{B}^p, \mathbf{E}^a, \mathbf{H}^p$)
  $\mathbf{y} = \mathrm{Max}(\log Pr(\frac{\mathbf{y}}{|\mathbf{y}|}))$ in $\mathbf{B}^a$ ▷ streaming result
  if $\langle eou \rangle$ in $\mathbf{y}$ **then** ▷ ending point
    break

2.2. Timing Estimator

Timing estimator consists of LSTM and linear layers. It runs incrementally and classifies whether the system should speak or not for each frame. The first frame classified as "system should speak" is determined as the response timing of spoken dialog system. The features extracted by feature extractor are concatenated and used as inputs. The prosodic and temporal features are input in every frame (50ms). The linguistic features and ESC are updated when the streaming ASR results are updated, otherwise the same values as at the previous time are input to the timing estimator. During training, binary cross entropy loss between the ground truth and output y is minimized. The optimization is applied from the start of the user utterance to 1 second after the start of the system utterance.

3. Multi-Look-Ahead ASR

Transformer-based end-to-end ASR models achieve high performance with the self-attention mechanism, but it needs to process the entire input sequence. Attention masks enable streaming ASR, but require a certain amount of look-ahead frames to minimize the performance degradation in comparison to processing the entire sequence. These look-ahead frames cause the delay. We use Multi-Look-Ahead streaming ASR proposed by Zhao et al. [22]. They introduced contextual block streaming ASR technique to RNN-Transducer, and also integrated multiple encoders with different delays.

3.1. Contextual Block Streaming Transducer (CBS-T)

The streaming ASR models in this paper are the Contextual Block Streaming Transducers(CBS-T), based on RNN-T [26]. RNN-T consists of three modules: acoustic encoder, label encoder, and joint network. CBS-T uses an encoder of Contextual Block Streaming ASR [27] (CBS-Encoder) for acoustic encoder. CBS-Encoder consists of 6 conformer [28] layers. The input speech is divided into blocks with three parts representing history, target, and look-ahead, whose numbers of frames are N_l, N_c, and N_r, respectively. The b-th block Z_b is processed as

$$H_b, \boldsymbol{c}_b = \mathrm{CBSEncoder}(Z_b, \boldsymbol{c}_{b-1}), \quad (1)$$

where H_b and $\boldsymbol{c}_b$ denote encoder output and contextual vector when the b-th block is input, respectively. The contextual vector passes global context information from past blocks to the current block.

3.2. Multi-Look-Ahead CBS-T

While CBS-Encoder enables streaming ASR, it still has a certain delay because it requires look-ahead frames. To reduce such delays while maintaining accuracy, Zhao et al. [22] proposed CBS-T that integrates multiple encoders with different delays. The encoders consist of a primary encoder which ensures high accuracy by encoding with a sufficient amount of look-ahead, and auxiliary encoders which give low latency results by encoding inputs of the look-ahead part of the primary encoder with a shorter look-ahead. While they proposed several architectures combining these encoders, in this study, we use one where the parameters are shared among the encoders and the auxiliary encoders receive the shifted input of the primary encoder to output the result corresponding to the look-ahead frames. This architecture achieves low latency and high accuracy while reducing model size by sharing parameters. We call this ASR system MLA CBS-T (Multi-Look-Ahead CBS-T). On the other hand, we refer to the single-encoder CBS-T used in the baseline model for comparison as SLA CBS-T.

Algorithm 1 shows the beam search algorithm in MLA CBS-T when $N_r/N_c = 1$. MLA CBS-T has N_r/N_c auxiliary encoders.The primary encoder outputs results up to time $t - T_{N_r}$ corresponding to the target frame at time t, where T_{N_r} denotes the time for N_r frames. Therefore, ASR is delayed by T_{N_r}. The input speech of ith auxiliary encoder is shifted $i \times N_c$ frames from the primary encoder's input. The numbers of the history frames and the target frames are N_l and N_c, the same as the primary encoder, and the number of the look-ahead frames is $N_r - i \times N_c$. The auxiliary encoder outputs results up to time $t - T_{N_r} + T_{i \times N_c}$, which is not included in the output of the primary encoder, to reduce delay. The beam search is first performed using the output of the primary encoder to update the search space $\mathbf{B}^p$. Next, beam search is performed using the output of the auxiliary encoder. At this time, beam search is performed based on the primary encoder's search space $\mathbf{B}^p$, and the search space is updated to $\mathbf{B}^a$. The next time, the search begins from the search space $\mathbf{B}^p$ before reflecting the update with the auxiliary encoder. Therefore, the ASR results reflect the results of the auxiliary encoder only at the time corresponding to the latest look-ahead frames, and the results of the primary encoder, which has a larger look-ahead size and higher reliability, are reflected for other frames including the target frame in latest block.

4. Experiments

We conducted experiments to verify whether a low latency ASR model is effective in response timing estimation and whether it is also effective when combined with ESC. We compare the proposed model with the baseline model, which utilizes SLA CBS-T without ESC, the model that uses MLA CBS-T, and the model that uses ESC.

4.1. Experimental Setting

We use the system described in Section 2, using SLA CBS-T as ASR and excluding the use of ESC as the baseline. For both the prosodic feature extractor and the VAD model, single layer LSTM which has 512 hidden dimensions is applied. Parameters M and N in syntactic completeness prediction are set to 5 and 3, respectively. The number of the layers, attention head, and hidden dimension of the Transformer encoder for linguistic feature extraction are set to 3, 2, and 300 respectively. The dimension of the prosodic feature h^p, the temporal feature h^t, and the linguistic feature h^l are all set to 128.

Table 1: *Character Error Rate and delay on our dataset.*

Method	**Delay** [ms]	**CER** [%]
SLA CBS-T	320	15.9
MLA CBS-T	64	16.2

The token units for ASR and linguistic feature extraction are Japanese characters. All encoders in both SLA CBS-T and MLA CBS-T have a common structure. We used 6 Conformer layers for the acoustic encoder and one layer of LSTM for the label encoder. The number of frames for history (N_l), target (N_c), and look-ahead (N_r) in the primary encoder of CBS-T are set to 8, 4, and 8, respectively. Every ASR model was pretrained with the core data of the CSJ corpus and then fine-tuned to our datasets. For decoding, we use the modified adaptive expansion beam search [29] with beam size of 5. In this experiment, for linguistic feature extraction, ASR is run in advance and its results are input to the Transformer encoder, taking into account the ASR delay. We assumed a 100 ms processing delay.

As described in Section 1, response timing estimation approaches can be classified into ES-TD and FS-TD, and the proposed method is FS-TD. As an ES-TD approach for comparison, we use a model based on GMF [13]. We replace the feature extraction part of GMF with that of our proposed model. The fixed pause was set to 350 ms, which is the median of the response timing distribution in the validation set.

Every model was trained for 30 epochs with AdamW [30]. The learning rates are set to 0.001 for our model and 0.0001 for GMF. The ASR model, the language model for syntactic completeness prediction, and the VAD model for temporal feature extraction are trained separately on the training set.

4.2. Data

We conducted our experiments on Japanese simulated spoken dialog data. This dataset contains a simulated conversation between two people playing the roles of user and agent of a restaurant information center. In the experiments, the agent's response timings were estimated as those of the spoken dialog systems. The dataset contains 90 dialogs, each averaging about 10 minutes, for a total of 14 hours. We used 72 dialogs for training, and 18 dialogs for test and evaluated the results by conducting a five-fold cross-validation. Unnatural data, e.g. long pause caused when the agent searches for a restaurant by the user's request, were excluded from the dataset. The final dataset used in the experiment included 4,927 pauses during user utterance and 1,231 speaker changes from user to agent.

4.3. ASR Results

Table 1 shows the character error rate (CER) and delay of encoder in SLA CBS-T and MLA CBS-T. The delay is calculated assuming that the reference token is in the center of the input block. MLA CBS-T degraded CER by 0.3% but reduced the 256 ms delay caused by look-ahead frames.

4.4. Response Timing Estimation Results

Table 2 shows the experimental results of response timing estimation. To evaluate whether the system was able to stay silent in a situation without a speaker change, start speaking in a sit-

Table 2: *Experimental results. P_x, R_x, and F_x denote precision, recall, and F_1 score, respectively. Each score is calculated assuming x seconds is the acceptable absolute error.*

Method	$P_{0.25}$	$R_{0.25}$	$F_{0.25}$	$P_{0.50}$	$R_{0.50}$	$F_{0.50}$	$P_{1.00}$	$R_{1.00}$	$F_{1.00}$
GMF	0.270	0.233	0.248	0.475	0.410	0.437	0.582	0.502	0.535
Baseline	0.340	0.380	0.357	0.534	0.587	0.557	0.700	0.765	0.729
+ MLA	0.361	0.389	0.375	0.550	0.590	0.569	**0.720**	0.771	0.744
+ ESC	0.364	0.397	0.380	0.546	0.596	0.570	0.707	0.771	0.737
+ MLA + ESC	**0.373**	**0.415**	**0.393**	**0.552**	**0.615**	**0.581**	0.707	**0.788**	**0.745**

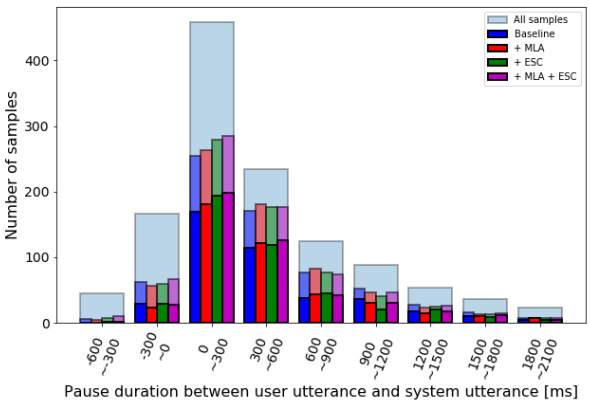

Figure 2: *Number of the correct answers per pause duration between user utterance and system utterance. The first and second rows of the bar graph show the cases of 250 ms and 500 ms acceptable absolute error, respectively.*

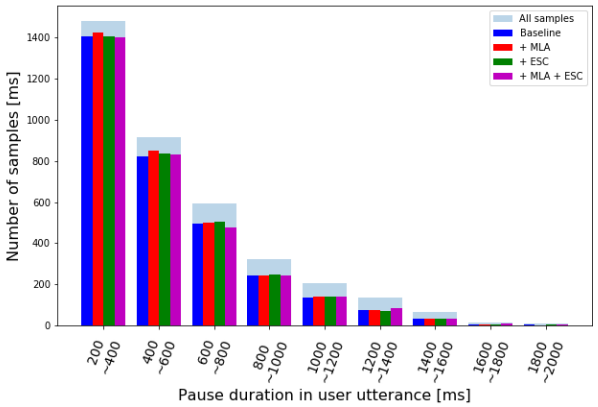

Figure 3: *Number of the correct answers per pause duration in user utterance.*

uation with a speaker change, and if so, whether it was able to speak at the appropriate timing, we use the precision, recall, and F1 score calculated with various acceptable absolute errors between the estimated and ground truth timings. If the estimated and ground truth timings are within the acceptable error, they are treated as true positives. Recall indicates whether the responses were correctly estimated in the situation where the system should speak within the absolute errors on the occasions when the system should speak. Precision indicates how many of the system's inferred utterances were correct.

Our models outperform the approach using GMF in all metrics. Unlike the ES-TD based approach including GMF, which runs only at the EoUs, the proposed approach estimates sequentially, and thus is able to use more temporal information.

The model with MLA CBS-T outperforms the baseline model in all metrics. In addition, the model with both ESC and MLA CBS-T outperforms the model with ESC and SLA CBS-T in all metrics. These results indicate that low-latency decoding by MLA CBS-T improves the performance of response timing estimation and it is also effective when combined with ESC.

Figure 2 shows the number of samples that the system should take a turn for each ground truth timing, and the number of samples that each model correctly predicted within the acceptable absolute errors. We can see the increase in the number of correct answers by introducing MLA CBS-T and ESC in the range of response timing from 0 ms to 300 ms. In other words, as we expected, the two approaches improve the performance of response timing estimation for quick response.

Figure 3 shows the number of samples that the system should not take a turn for each length of pause during user, and the number of samples that each model correctly predicted that the system should not speak. The differences among the models are small compared to Figure 2, but the number of correct answers is slightly lower when ESC is used. We analyzed that the model using ESC incorrectly predicted turn-taking because completion of utterance was predicted when an expression appeared in the middle of a user utterance that actually continued but could have ended the utterance.

5. Conclusions

Although human conversation often involves quick responses, conventional response timing estimation models have difficulty predicting quick responses due to ASR delays. The ASR delay prevents the use of linguistic information near the end of speech which is crucial especially when estimating quick responses. We introduced MLA CBS-T to reduce delay while minimizing the degradation of ASR accuracy. We conducted experiments to confirm the effectiveness of reducing ASR delay in response timing estimation. We also verified whether it is effective when combined with a method from a previous study that uses a feature called ESC. Experimental results showed that introducing MLA CBS-T is effective in improving the performance of ESC estimation and response timing estimation for quick response.

6. Acknowledgements

This research is supported by NII CRIS collaborative research program operated by NII CRIS and LINE Corporation.

7. References

[1] M. Bull and M. P. Aylett, "An analysis of the timing of turn-taking in a corpus of goal-oriented dialogue," *5th International Conference on Spoken Language Processing (ICSLP 1998)*, 1998.

[2] S. G. Roberts, F. Torreira, and S. C. Levinson, "The effects of processing and sequence organization on the timing of turn taking: a corpus study," *Frontiers in Psychology*, vol. 6, p. 509, 2015.

[3] T. Stivers, N. J. Enfield, P. E. Brown, C. Englert, M. Hayashi, T. Heinemann, G. Hoymann, F. Rossano, J. P. de Ruiter, K.-E. Yoon, and S. C. Levinson, "Universals and cultural variation in turn-taking in conversation," *Proceedings of the National Academy of Sciences*, vol. 106, pp. 10 587 – 10 592, 2009.

[4] L. ten Bosch, N. Oostdijk, and J. P. de Ruiter, "Durational aspects of turn-taking in spontaneous face-to-face and telephone dialogues," in *International Conference on Text, Speech and Dialogue*, 2004, pp. 563–570.

[5] P. A. Heeman and R. Lunsford, "Turn-taking offsets and dialogue context," in *Proceedings of 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017)*, 2017, pp. 1671–1675.

[6] T. Kawahara, T. Iwatate, and K. Takanashi, "Prediction of turn-taking by combining prosodic and eye-gaze information in poster conversations," in *Interspeech*, 2012, pp. 726–729.

[7] A. Maier, J. Hough, and D. Schlangen, "Towards deep end-of-turn prediction for situated spoken dialogue systems," in *Interspeech*, 2017, pp. 1676–1680.

[8] R. Masumura, T. Tanaka, A. Ando, R. Ishii, R. Higashinaka, and Y. Aono, "Neural dialogue context online end-of-turn detection," in *SIGDIAL Conference*, 2018, pp. 224–228.

[9] K. Hara, K. Inoue, K. Takanashi, and T. Kawahara, "Prediction of turn-taking using multitask learning with prediction of backchannels and fillers," in *Proceedings of 19th Annual Conference of the International Speech Communication Association (INTERSPEECH 2018)*, 2018, pp. 991–995.

[10] D. Lala, K. Inoue, and T. Kawahara, "Evaluation of real-time deep learning turn-taking models for multiple dialogue scenarios," *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pp. 78–86, 2018.

[11] Z. Aldeneh, D. Dimitriadis, and E. M. Provost, "Improving end-of-turn detection in spoken dialogues by detecting speaker intentions as a secondary task," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6159–6163, 2018.

[12] E. Ekstedt and G. Skantze, "TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2981–2990.

[13] J. Yang, P.-H. Wang, Y. Zhu, M. Feng, M. Chen, and X. He, "Gated multimodal fusion with contrastive learning for turn-taking prediction in human-robot dialogue," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7747–7751, 2022.

[14] M. Takeuchi, N. Kitaoka, and S. Nakagawa, "Generation of natural response timing using decision tree based on prosodic and linguistic information," *8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, pp. 609–612, 2003.

[15] G. Skantze, "Towards a general, continuous model of turn-taking in spoken dialogue using lstm recurrent neural networks," in *SIGDIAL Conference*, 2017, pp. 220–230.

[16] M. Roddy, G. Skantze, and N. Harte, "Multimodal continuous turn-taking prediction using multiscale rnns," *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pp. 2442–2452, 2018.

[17] M. Roddy and N. Harte, "Neural generation of dialogue response timings," pp. 2442–2452, 2020.

[18] S. Fujie, H. Katayama, J. Sakuma, and T. Kobayashi, "Timing generating networks: Neural network based precise turn-taking timing prediction in multiparty conversation," in *Proceedings of 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH 2021)*, 2021, pp. 3226–3230.

[19] E. Ekstedt and G. Skantze, "Voice activity projection: Self-supervised learning of turn-taking events," in *Proceedings of 23rd Annual Conference of the International Speech Communication Association (INTERSPEECH 2022)*, 2022, pp. 5190–5194.

[20] J. Sakuma, S. Fujie, and T. Kobayashi, "Response timing estimation for spoken dialog system using dialog act estimation," in *Proceedings of 23rd Annual Conference of the International Speech Communication Association (INTERSPEECH 2022)*, 2022, pp. 4486–4490.

[21] ——, "Response timing estimation for spoken dialog systems based on syntactic completeness prediction," *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 369–374, 2023.

[22] H. Zhao, S. Fujie, T. Ogawa, J. Sakuma, Y. Kida, and T. Kobayashi, "Conversation-oriented asr with multi-look-ahead cbs architecture," *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[23] K. Yokoyama, H. Takatsu, H. Honda, S. Fujie, and T. Kobayashi, "Investigation of users' short responses in actual conversation system and automatic recognition of their intentions," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 934–940, 2018.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017, pp. 5998–6008.

[26] A. Graves, "Sequence transduction with recurrent neural networks," *ArXiv*, vol. abs/1211.3711, 2012.

[27] E. Tsunoo, Y. Kashiwagi, T. Kumakura, and S. Watanabe, "Transformer asr with contextual block processing," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 427–433, 2019.

[28] 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," in *Proceedings of 21st Annual Conference of the International Speech Communication Association (INTERSPEECH 2020)*, 2020, pp. 5036–5040.

[29] J. Kim, Y. Lee, and E. Kim, "Accelerating rnn transducer inference via adaptive expansion search," *IEEE Signal Processing Letters*, vol. 27, pp. 2019–2023, 2020.

[30] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.