TURN-TAKING AND BACKCHANNEL PREDICTION WITH ACOUSTIC AND LARGE LANGUAGE MODEL FUSION

Jinhan Wang[1†] *Long Chen*[2†] *Aparna Khare*[2] *Anirudh Raju*[2] *Pranav Dheram*[2]
Di He[2] *Minhua Wu*[2] *Andreas Stolcke*[2] *Venkatesh Ravichandran*[2]

[1]University of California, Los Angeles, USA [2]Amazon Alexa AI, USA

ABSTRACT

We propose an approach for continuous prediction of turn-taking and backchanneling locations in spoken dialogue by fusing a neural acoustic model with a large language model (LLM). Experiments on the Switchboard human-human conversation dataset demonstrate that our approach consistently outperforms the baseline models with single modality. We also develop a novel multi-task instruction fine-tuning strategy to further benefit from LLM-encoded knowledge for understanding the tasks and conversational contexts, leading to additional improvements. Our approach demonstrates the potential of combined LLMs and acoustic models for a more natural and conversational interaction between humans and speech-enabled AI agents.

***Index Terms*—** turn-taking, backchannel, large language model, model fusion, instruction tuning.

1. INTRODUCTION

AI voice assistants are becoming increasingly multi-functional and important in people's daily lives [1, 2, 3]. However, conventional voice assistant systems are mostly designed for query-based use cases. Towards the goal of more effective and effortless interaction between humans and AI, voice assistants that can solve tasks in a more natural manner and human-human-like conversational experience would be very desirable [4]. As one of its most basic capabilities, the system should be able to determine when to take turns naturally and with minimal latency in a dialogue with the user, and without the need for push-to-talk or wakewords. One common solution for turn-taking is to trigger the system's response after a period of silence based on a predefined threshold [5, 6, 7, 8]. However, this threshold-based method may result in a suboptimal user experience due to lack of naturalness [9, 10]. Another behavior that is important for managing human-human conversations that are a challenge for present-day conversational systems is backchanneling [11, 12]. Backchannels are defined as short utterances expressing acknowledgment or reactions on the part of the listener, without signaling an intent to take a turn, such as "*uh-huh*", "*oh no*" and "*right*". They typically occur during the current speaker's turn and do not necessarily trigger turn-taking [13, 14, 11].

Going back to conversation analysis in linguistic pragmatics [15], there is a long history of descriptive and computational research trying to capture turn-taking and backchanneling cues in multiple modalities. In the acoustic domain, prosodic features such as duration, pitch, voice quality and intensity have been shown to have high correlation with turn-taking and backchannel locations [16, 11, 17, 9]. In [6], turn-taking prediction has been embedded as an auxiliary task in automatic speech recognition (ASR), based on acoustic encoder features. Aside from acoustic features, linguistic features have also been investigated. Given context or predicted transcription from an ASR system, word embeddings like Word2Vec [5] and encoded hidden states from transformer networks [18] or recurrent neural networks (RNNs) [19, 20] have been used as linguistic representations for prediction. Furthermore, multi-modal fusion or joint modeling have been explored in earlier work, using RNN-based text and acoustic encoders [21, 17].

However, in these earlier works that use linguistic modeling, features and representations are relatively simple and only approximate the full range of linguistic cues used by humans in daily conversation [22]. Large language models (LLMs) promise to better capture the formal dependencies and meaning relations in language [23, 24, 25]. Ekstedt et al. [22] proposed TurnGPT to leverage LLMs (in the form of GPT2 [26]) for turn-taking prediction, showing superior performance compared to conventional modeling techniques. However, that work is still limited to turn-taking prediction and uses only lexical (text) information.

In this work, we propose a novel approach for turn-taking and backchannel location prediction in spoken dialogue, with a fusion of LLM and acoustic models. We adopt two LLMs, GPT2 [26] and RedPajama [27] for modeling linguistic cues, and we use HuBERT [28] for modeling acoustic cues, to leverage both representations and prior knowledge learned during pretraining. Two fusion methods are explored by manipulating the LLM branch to better understand the role of the different modalities in joint modeling. Furthermore, inspired by the success of instruction fine-tuning of LLMs for other tasks [29, 30], a novel multi-task instruction fine-tuning is proposed to further utilize the ability of LLMs to understand task descriptions and dialogue history, and direct the joint model to focus on different tasks with task-specific submodules triggered by corresponding instructions. Our main contributions are thus (1) extending the turn-taking model to include backchanneling, (2) use of LLMs with acoustic fusion for these tasks, and (3) exploration of LLMs for instruction-tuning rather than simple token encoding and prediction.

2. PROPOSED METHOD

2.1. Problem setup

For a more natural human-agent interaction, the task of interest here is to predict the proper turn-related behavior with respect to the user's input to a voice assistant system during conversation. Three distinct behaviors are considered: 1) **Continuing Speech**: the currently active speaker is predicted to continue speaking (the other party keeps listening); 2) **Backchannel**: the listening party (system or user) should generate a brief utterance as a sign of acknowledgment, understanding or assessment without an intention to take the turn [13]; 3) **Turn-taking**: the current speaker is predicted to be

[†] Equal contribution. This first author was an intern at Amazon.

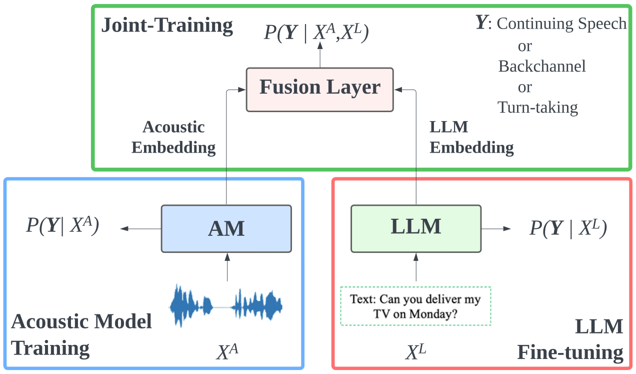

Fig. 1: Schematic of combined acoustic and LLM modeling for turn-taking, backchannel and continuing speech prediction.

done talking and the nonspeaking party should take over the conversation and provide a response. More formally, given a (partial) utterance with acoustic features X^A and text features X^L, the goal is to predict the class/behavior posteriors $P(Y|X^A, X^L)$, where Y is from the label set consisting of "Continuing Speech", "Backchannel" and "Turn-taking". The framework is depicted in Figure 1.

2.2. Acoustic and language modeling

The acoustic model is shown as the bottom left module in Figure 1. In this work, HuBERT [28] is used to encode speech signals of the (partial) utterances, as well as to serve as the base acoustic model (AM) for prediction with single modality. To manipulate the architecture for classification, the average-pooled 768-dimensional HuBERT embedding across all time steps emitted from the base model is fed into a projection layer to obtain a 256-dimensional vector. Then a linear classifier maps the projection to three classes. During model training, the acoustic base model is frozen.

The linguistic modeling is done by LLM fine-tuning as shown at the bottom right in Figure 1. Here, either GPT2 or RedPajama are used to encode the text of the (partial) utterances, producing embeddings of 768 and 2560 dimensions, respectively. Unlike in acoustic modeling, LLM fine-tuning uses the embedding of the last token. Then, the embedding is fed into a linear layer of dimension 3 for classification. Depending on the base LLM being used, different fine-tuning strategies are applied, as discussed further in Section 3.3.

2.3. Fusion or joint training

A late fusion mechanism is used where the final embeddings emitted from the AM and LLM are concatenated and fed into a single linear classification layer with dimension 3 to predict $P(Y|X^A, X^L)$, as shown in the top module of Figure 1. Two different fusion setups are investigated. In Option 1 (**Opt1**) both AM and LLM are loaded from the pretrained library [31] without fine-tuning. Then, both the fusion layer and the LLM base model undergo domain adaptation and downstream task training. In Option 2 (**Opt2**), aside from loading the pretrained AM as in Opt1, the LLM is loaded after standalone fine-tuning as described in Section 2.2. Then the LLM branch is also frozen and only the fusion layer is trained. The key difference between **Opt1** and **Opt2** is whether LLM has been fine-tuned for the downstream task and frozen. Though more sophisticated architectures could be helpful, we will demonstrate the effectiveness of combining AM and LLM for turn-taking and backchannel prediction tasks even with the two simple fusion options considered here.

2.4. Multi-task instruction fine-tuning

Besides serving as an advanced text encoder, LLMs have also demonstrated the ability to understand narrative instructions in natural language. Instruction fine-tuning [29] has been used to teach LLMs this behavior. Thus, we reformulate our framework as a multi-task training scenario with instructions specific to our tasks. Rather than setting up a three-way classification, each class is handled as a separate binary classification task. This will later allows us to evaluate performance as three separate detection tasks. Figure 2 shows the diagram of this multi-task instruction fine-tuning process, where Sample 0, 1 and 2 are considered as the samples with corresponding ground-truth labels of "Continuing Speech", "Backchannel" and "Turn-taking", respectively. During training, each sample will be augmented three times, with the following respective instructions: 1) **Inst 0**: "Identify if the current speaker will continue to speak at the end of the sentence."; 2) **Inst 1**: "Identify if another speaker will backchannel at the end of the sentence."; 3) **Inst 2**: "Identify if another speaker will take the turn at the end of the sentence."

For each generated sample, if the prepended instruction corresponds to the ground-truth label, i.e. {*inst0, sample0*}, {*inst1, sample1*} and {*inst2, sample2*}, then the corresponding binary label will be assigned as 1, otherwise 0. Each classifier is only in charge of one corresponding instruction and updates only its parameters, without being affected by samples augmented by the other two instructions. Let X denote a batch of samples $X = [x_1, x_2, ..., x_n]$, with corresponding ground-truth labels $Y = [y_1, y_2, ..., y_n]$, and denote by s the instruction index. The workflow can be written as follows:

$$\begin{aligned} X_s &= \{inst_s, X\} \\ &= (inst_s, x_1), (inst_s, x_2) ... (inst_s, x_n) \;\; s = 0, 1, 2 \end{aligned} \tag{1}$$

$$Y_s = [y_{s,1}, y_{s,2}, ..., y_{s,n}], \quad y_{s,i} = \begin{cases} 1 & s = y_i \\ 0 & s \neq y_i \end{cases} \tag{2}$$

$$\hat{Y}_s = \mathit{Classifier}_s(\mathit{Model}(X_s)) \tag{3}$$

$$L_s = \mathit{BCELoss}(\hat{Y}_s, Y_s), \quad L = \sum_{s=0}^{2} L_s \tag{4}$$

Compared to Section 2.2, the LLM is used not only as a text encoder, but also for instruction understanding, leveraging pretrained knowledge about the tasks. Furthermore, having independent task-specific binary classifiers enables scaling to additional speaker activity classes or multi-label tasks. The training setup fully utilizes all original samples for each task to update the corresponding classifier by prepending the appropriate instruction.

We also explore a variant of instruction fine-tuning with added dialogue history to contextualize the model's interpretation. Here, two sentences preceding the target partial utterance, with speaker changes marked, are appended to the task-specific instruction, using the following format: "Identify <instruction text>: <history with speaker token>. <target sample with speaker token>."

3. EXPERIMENTS

3.1. Dataset

We use the Switchboard corpus [32] (*Switchboard-1 Telephone Speech Corpus: Release 2*). It is comprised of 2438 dyadic conversational dialogues involving 543 male and female speakers, who were connected by phone to converse about one of around 70 topics. The

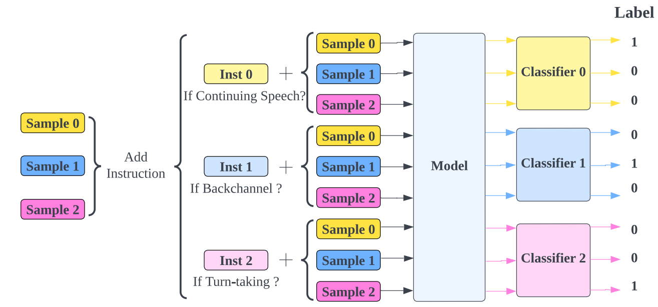

Fig. 2: LLM-based multi-task instruction fine-tuning for turn-taking, backchannel, and continuing speech prediction.

dataset consists of around 260 hours of audio with ground-truth transcripts comprising around 3 million words; word-level time alignments are also available. We use these symmetrical human-human dialogues to model appropriate system behavior for a conversational voice assistant when receiving user input. Though all data consists of human-human dialogue, we treat the currently active user's speech as if input to a dialogue system, and the other (listening) speaker's behavior as a model for how the system should behave. As discussed, the possible behaviors are "let current speaker continue", "produce backchannel", or "take a turn". To utilize the data fully, user and system identities are swapped at each speaker change, i.e., when speaker A is active, A is the user and the system will try to behave as speaker B, and then vice versa for the next turn.

Given ground-truth speaker-wise dialog transcripts and word alignments, we prepare the data in each session as follows: 1) Extract dialog sentences from each speaker, while simultaneously normalizing special annotations, including [silence]/[noise] removal, partial word completion, and mispronunciation correction, as in [22]. 2) Mark isolated one-word or two-word phrases as backchannel candidates. Backchannels are considered to be the 20 most frequent one and two-word phrases, such as "yeah", "mmhmm" and "oh okay", as summarized in [22]. 3) Combine the two speakers' dialog sentences in start-time ascending order and break sentences into words, for the purpose of word-level labeling in the following steps. 4) Remove words marked as backchannels and save them in a candidate list along with their speaker, start-time and end-time attributes. 5) Mark all speaker changes as "Turn-taking" at last word spoken by a speaker. 6) Insert backchannel candidates back into the original dialogue according to their start-times and mark the word spoken by the other speaker where backchanneling occurs as "Backchannel". If a word is marked by none of these two labels, the default label of "Continuing Speech" is assigned.

Note that overlapping speech is only recorded for backchannel utterances, but not for regular turns, which are serialized, in a way compatible with TurnGPT processing [22]. We leave the prediction of turn-taking with overlap [33] for future work. (While overlapping turns are not uncommon for human-human dialog, a polite AI agent might refrain from producing them.) After this preparation, each sample is a (partial) utterance (audio, text, or both) spoken by a single speaker, with the class label given by the last word's label within the utterance. The data is split by session with train:validation:test ratio of 2000:300:138 [22].

3.2. Training and evaluation scenarios

During training, since "Continuing Speech" is by far the majority class, a downsampling procedure is applied to samples of that class, such that that the label frequency equals the average number of "Backchannel" and "Turn-taking" samples. The resulting subset has "Continuing Speech", "Backchannel", and "Turn-Taking" occurring with counts 71k vs. 56k vs. 86k in training, and 6k vs. 5k vs. 7k for validation, respectively. During evaluation, all samples are used without downsampling, i.e., samples of each class in the test session will be decoded regardless of the class imbalance. The test set has samples with "Continuing Speech", "Backchannel", and "Turn-taking" with counts 123k, 2.3k and 3.2k, respectively.

In TurnGPT [22], the balanced accuracy (bAcc) over true and false turn-shifts is used as the evaluation metric. Here, since we have formulated three binary detection tasks, we prefer performance metrics that are independent of class priors and operating points (thresholds), namely, area-under-the-curve (AUC) and equal error rate (EER), evaluated for each class separately and in average. The metrics are based on decision scores that are given by the logits for the targeted class, after softmax normalization.

3.3. Experimental details

All frameworks are implemented using the Huggingface Transformers Library [31] on 8 NVIDIA V100 GPUs. To validate the generalization of the proposed methods, two pretrained LLMs of different sizes are investigated, namely GPT2 (124M parameters) [26] and RedPajama (3B parameters) [27]. For GPT2, the entire model is unfrozen for LLM fine-tuning and fusion Option 1. For RedPajama, a parameter-efficient fine-tuning approach, LoRA [34] with a rank of 32, is applied in LLM fine-tuning and fusion Option 1, resulting in around 0.4% (≈10M) trainable parameters. All models are trained with learning rate 5×10^{-5}, number of epochs 5, and batch size 4. All other hyperparameters are set to the default values provided by the Transformer library [31].

4. RESULTS

4.1. Single modality versus fusion

Experimental results for single modalities and the two fusion approaches are reported in Table 1. First, with single modalities, language models yield much better performances than acoustic models. Second, by comparing the text-only models based on LLMs, it turns out that even though fine-tuning RedPajama results in significantly fewer trainable parameter than fully fine-tuning GPT2, RedPajama still achieves comparable performance, with an average AUC of 0.8351, as compared to 0.8292 for GPT2. This result indicates that RedPajama, as a larger LLM, has higher efficiency and greater potential for modeling conversational dialogue, and benefits more from approaches that exploit the model's language understanding capabilities, such as the proposed instruction fine-tuning.

Under the same LLM setup, both fusion approaches achieve significant improvements over models with single modality, as shown in Table 1. For instance, the fusion model with RedPajama + HuBERT + Opt1 achieves the best performance for all three classes with an average AUC of 0.8657, leading to relative improvements of 22.6% and 3.67% over the best acoustic and text single modality models, respectively. Moreover, by comparing the three classes, predicting "Continuing Speech" and "Turn-taking" benefits most from the fusion, while "Backchannel" only shows a small improvement. This result aligns with known properties of these turn-management events, where "Turn-taking" and "Continuing speech" are strongly cued by intonation and duration features, whereas "Backchannel" is possibly more related to syntactic and semantic information. In

Table 1: Results for single modality and fusion models.

Method	AUC(Cont)	AUC(Back)	AUC(Turn)	AUC(avg)	EER(avg)
HuBERT	0.7323	0.6455	0.7401	0.7060	34.87
GPT2	0.8510	0.7744	0.8623	0.8292	24.47
+ HuBERT Opt1	0.8783	0.7798	0.884	0.8474	22.63
+ HuBERT Opt2	0.8778	**0.7862**	0.8859	0.8500	22.77
RedPajama	0.8629	0.7739	0.8685	0.8351	23.60
+ HuBERT Opt1	**0.8992**	**0.7862**	**0.9116**	**0.8657**	**20.33**
+ HuBERT Opt2	0.8982	0.7743	0.9006	0.8577	21.57

Table 2: Results with multi-task instruction fine-tuning.

Method	AUC(Cont)	AUC(Back)	AUC(Turn)	AUC(avg)	EER(avg)
GPT2	0.8416	0.7863	0.8582	0.8287	24.13
+ HuBERT Opt1	0.8726	0.7901	0.8766	0.8464	22.50
+ HuBERT Opt2	0.8806	0.7838	0.8890	0.8511	22.23
RedPajama	0.8668	0.8097	0.8796	0.8520	21.80
+ HuBERT Opt1	0.9000	**0.8229**	0.9127	0.8785	19.50
+ HuBERT Opt2	0.8980	0.8182	0.9129	0.8764	19.60
RedPajama + History	0.8747	0.8074	0.8912	0.8578	21.63
+ HuBERT Opt1	**0.9029**	0.8184	**0.9197**	**0.8803**	**19.30**

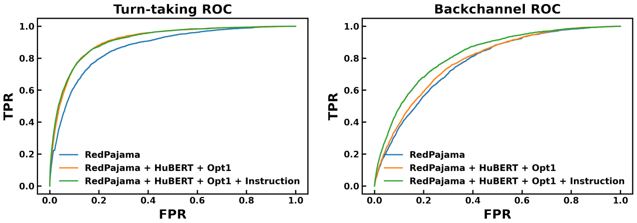

Fig. 3: ROC plots for turn-taking (left) and backchannel (right).

addition, the difference between Opt1 and Opt2 does not share the same pattern for GPT2 and RedPajama. RedPajama works better with Opt1. We suspect that GPT2 has relatively limited modeling capacity. Unfreezing the larger RedPajama model will learn information that better complements the acoustic information.

We aimed to compare our model with TurnGPT [22], where bAcc of 0.789 and 0.823 were reported for the "Spoken" dataset (of which Switchboard makes up the majority) with training on "Assistant" and "Full" datasets, respectively. We obtained a similar bAcc of 0.8002 for the GPT2-only model when focusing only on "Turn-taking" and "Continuing Speech" samples. Though this is not an apples-to-apples comparison, as the evaluation samples could be different, it shows that our LLM is comparable to TurnGPT when we leave out backchanneling. In this focused two-class evaluation, we obtain an improved bAcc of 0.8578 for the RedPajama + HuBERT + Opt1 fusion model, confirming the benefit of complementing lexical with acoustic information.

4.2. Multi-task instruction fine-tuning

Results for multi-task instruction fine-tuning are reported in Table 2. For GPT2, it shows that applying instruction fine-tuning only results in a very minor differences to Table 1. However, when replacing GPT2 with RedPajama, significant improvements on average AUC and EER are observed for all three modeling approaches, with relative improvements of 2.02%, 1.5% and 2.16% on average AUC for RedPajama, RedPajama + HuBERT Opt1 and Opt2, respectively. Moreover, RedPajama + HuBERT + Opt1 with multi-task instruction fine-tuning achieves the best performance for all the cases without the dialogue history, with an average AUC of 0.8785.

More interestingly, comparing to the results without the instruction fine-tuning in Table 1, "Backchannel" prediction sees the highest AUC gain from applying the multi-task instruction fine-tuning, compared to other classes. Figure 3 shows the ROC curves for Turn-taking and Backchannel. It is clear here that Turn-taking benefits remarkably from the fusion, but benefits minimally from the instruction fine-tuning, while Backchannel shows the opposite trend. We conducted a further analysis by calculating the Backchannel score distribution of the samples. As shown in the left plot in Figure 4, applying instruction fine-tuning helps to push the score distribution of the backchannel (positive) samples and non-backchannel (negative) samples higher and lower, respectively. The right portion of Figure 4 shows an example, with a transcript of "It gets very uncomfortable on hills, most the teachers have taking[1] to wearing uh track shoes", the model with instruction fine-tuning correctly predicts the backchannel behavior after "hills", while the model without predicts a backchannel after "uncomfortable". This observation also supports our earlier conjecture that backchanneling is a speech activity best predicted by syntactic/semantic context. These results demonstrate that RedPajama benefits from multi-task instruction fine-tuning for better task understanding and more accurate backchannel prediction.

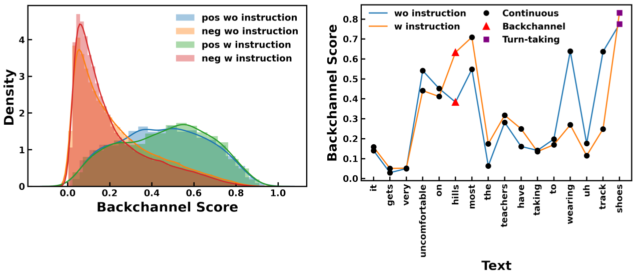

Fig. 4: Left: backchannel score distribution for the positive and negative samples. Right: a sentence example with token-level backchannel score. The markers represent the ground-truth token labels.

4.3. Instruction fine-tuning with dialogue history

The last section of Table 2 shows the results for multi-task instruction fine-tuning with added dialogue history. As described earlier, a dialogue history of two sentences is included in the instruction, prepended to each sample utterance. Compared to the instruction fine-tuning results without history, average AUC and EER improve with history. However, when looking at each class individually, both "Turn-taking" and "Continuing Speech" classes are predicted better with history information, while the "Backchannel" class sees a slight degradation. This could be because backchanneling is largely a locally-cued behavior and affected little by long-term context.

5. CONCLUSIONS

We have proposed a fusion model for turn-taking and backchannel prediction in spoken dialogue, combining both LLM and acoustic modeling. We experimented with LLMs of various sizes (GPT2 and RedPajama) and used HuBERT for modeling acoustic cues, to leverage both representations and prior knowledge learned from pretraining. Experiments demonstrate that our fusion approach consistently outperforms the baseline models with single modality, which indicates that joint modeling is effective at exploiting the complementarity of the modalities. Moreover, the proposed multi-task instruction fine-tuning strategy leverages LLMs for better task understanding and further improvements. Our approach provides a solution for more accurate causal turn-taking and backchannel prediction, ultimately enabling more natural and conversational human-agent interactions. In future work, it will be worth investigating how the use of automatic instead of ground-truth transcriptions would affect results, as required for inference in real-time applications.

[1] The provided corpus transcription. The actual word spoken is "taken".

6. REFERENCES

[1] Matthew B. Hoy, "Alexa, Siri, Cortana, and more: an introduction to voice assistants," *Medical reference services quarterly*, vol. 37, no. 1, pp. 81–88, 2018.

[2] Ramin Yaghoubzadeh, Marcel Kramer, Karola Pitsch, and Stefan Kopp, "Virtual agents as daily assistants for elderly or cognitively impaired people: Studies on acceptance and interaction feasibility," in *Intelligent Virtual Agents*, 2013, pp. 79–91.

[3] Pranav Rane, Varun Mhatre, and Lakshmi Kurup, "Study of a home robot: Jibo," *International journal of engineering research and technology*, vol. 3, no. 10, pp. 490–493, 2014.

[4] Nigel G. Ward and David DeVault, "Ten challenges in highly-interactive dialog system.," in *AAAI Spring Symposium*, 2015.

[5] Divesh Lala, Koji Inoue, and Tatsuya Kawahara, "Evaluation of real-time deep learning turn-taking models for multiple dialogue scenarios," in *Proc. ACM ICMI*, 2018, pp. 78–86.

[6] Shuo-yiin Chang, Bo Li, Tara N Sainath, Chao Zhang, Trevor Strohman, Qiao Liang, and Yanzhang He, "Turn-taking prediction for natural conversational speech," in *Proc. Interspeech*, 2022, pp. 1821–1825.

[7] Gabriel Skantze, Anna Hjalmarsson, and Catharine Oertel, "Turn-taking, feedback and joint attention in situated human-robot interaction," *Speech Communication*, vol. 65, pp. 50–66, 2014.

[8] Nigel G. Ward, Olac Fuentes, and Alejandro Vega, "Dialog prediction for a general model of turn-taking," in *Proc. Interspeech*, 2010, pp. 2662–2665.

[9] Zakaria Aldeneh, Dimitrios Dimitriadis, and Emily Mower Provost, "Improving end-of-turn detection in spoken dialogues by detecting speaker intentions as a secondary task," in *Proc. ICASSP*, 2018, pp. 6159–6163.

[10] Erik Ekstedt and Gabriel Skantze, "Voice activity projection: Self-supervised learning of turn-taking events," in *Proc. Interspeech*, 2022, pp. 5190–5194.

[11] Gabriel Skantze, "Turn-taking in conversational systems and human-robot interaction: a review," *Computer Speech & Language*, vol. 67, pp. 101178, 2021.

[12] Markus Mueller, David Leuschner, Lars Briem, Maria Schmidt, Kevin Kilgour, Sebastian Stueker, and Alex Waibel, "Using neural networks for data-driven backchannel prediction: A survey on input features and training techniques," in *Human-Computer Interaction: Interaction Technologies*, 2015, pp. 329–340.

[13] Daniel Ortega, Chia-Yu Li, and Ngoc Thang Vu, "Oh, jeez! or uh-huh? a listener-aware backchannel predictor on ASR transcriptions," in *Proc. IEEE ICASSP*, 2020, pp. 8064–8068.

[14] Tatsuya Kawahara, Takashi Yamaguchi, Koji Inoue, Katsuya Takanashi, and Nigel G Ward, "Prediction and generation of backchannel form for attentive listening systems.," in *Interspeech*, 2016, pp. 2890–2894.

[15] Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson, "A simplest systematics for the organization of turn taking for conversation," in *Studies in the organization of conversational interaction*, pp. 7–55. Elsevier, 1978.

[16] Gabriel Skantze, "Towards a general, continuous model of turn-taking in spoken dialogue using LSTM recurrent neural networks," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2017, pp. 220–230.

[17] Chaoran Liu, Carlos Toshinori Ishi, and Hiroshi Ishiguro, "Turn-taking estimation model based on joint embedding of lexical and prosodic contents.," in *Proc. Interspeech*, 2017, pp. 1686–1690.

[18] Jiudong Yang, Peiying Wang, Yi Zhu, Mingchao Feng, Meng Chen, and Xiaodong He, "Gated multimodal fusion with contrastive learning for turn-taking prediction in human-robot dialogue," in *Proc. ICASSP*, 2022, pp. 7747–7751.

[19] Ryo Masumura, Tomohiro Tanaka, Atsushi Ando, Ryo Ishii, Ryuichiro Higashinaka, and Yushi Aono, "Neural dialogue context online end-of-turn detection," in *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 2018, pp. 224–228.

[20] Ryo Masumura, Taichi Asami, Hirokazu Masataki, Ryo Ishii, and Ryuichiro Higashinaka, "Online end-of-turn detection from speech based on stacked time-asynchronous sequential networks.," in *Proc. Interspeech*, 2017, pp. 1661–1665.

[21] Matthew Roddy, Gabriel Skantze, and Naomi Harte, "Multimodal continuous turn-taking prediction using multiscale RNNs," in *Proc. ACM ICMI*, 2018, pp. 186–190.

[22] Erik Ekstedt and Gabriel Skantze, "TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog," in *Proc. EMNLP*, 2020, pp. 2981–2990.

[23] Luciano Floridi and Massimo Chiriatti, "GPT-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, pp. 681–694, 2020.

[24] Naman Jain, Skanda Vaidyanath, Arun Iyer, Nagarajan Natarajan, Suresh Parthasarathy, Sriram Rajamani, and Rahul Sharma, "Jigsaw: Large language models meet program synthesis," in *Proc. ACM ICSE*, 2022, pp. 1219–1231.

[25] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al., "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and individual differences*, vol. 103, pp. 102274, 2023.

[26] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019.

[27] Together Computer, "RedPajama: An open source recipe to reproduce LLaMA training dataset," https://github.com/togethercomputer/RedPajama-Data, Apr. 2023.

[28] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[29] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le, "Finetuned language models are zero-shot learners," in *Proc. ICLR*, 2022.

[30] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, 2023.

[31] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush, "Transformers: State-of-the-art natural language processing," in *Proc. EMNLP*, 2020, pp. 38–45.

[32] John J Godfrey, Edward C. Holliman, and Jane McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. IEEE ICASSP*, 1992, vol. 1, pp. 517–520.

[33] Elizabeth Shriberg, Andreas Stolcke, and Don Baron, "Observations on overlap: findings and implications for automatic processing of multi-party conversation," in *Proc. Interspeech*, 2001, pp. 1359–1362.

[34] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen, "LoRA: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.