INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Automatic Evaluation of Turn-taking Cues in Conversational Speech Synthesis

Erik Ekstedt, Siyang Wang, Éva Székely, Joakim Gustafson, Gabriel Skantze

KTH, Royal Institute of Technology, Stockholm, Sweden

erikekst@kth.se, siyangw@kth.se, szekely@kth.se, jkgu@kth.se, skantze@kth.se

Abstract

Turn-taking is a fundamental aspect of human communication where speakers convey their intention to either hold, or yield, their turn through prosodic cues. Using the recently proposed Voice Activity Projection model, we propose an automatic evaluation approach to measure these aspects for conversational speech synthesis. We investigate the ability of three commercial, and two open-source, Text-To-Speech (TTS) systems ability to generate turn-taking cues over simulated turns. By varying the stimuli, or controlling the prosody, we analyze the models performances. We show that while commercial TTS largely provide appropriate cues, they often produce ambiguous signals, and that further improvements are possible. TTS, trained on read or spontaneous speech, produce strong turn-hold but weak turn-yield cues. We argue that this approach, that focus on functional aspects of interaction, provides a useful addition to other important speech metrics, such as intelligibility and naturalness.

Index Terms: text-to-speech, turn-taking, human-computer interaction

1. Introduction

In recent years, there has a been an increased interest in developing conversational Text-to-Speech (TTS) [1, 2, 3, 4]. Whereas earlier conversational systems had to rely on TTS built for read speech, conversational TTS will allow systems to interact in a more natural and fluid way, more closely resembling a human-human conversation. Many commercial vendors are now also offering conversational variants of their voices.

Until recently, the evaluation of conversational TTS has been primarily centered around the voice's perceived naturalness and intelligibility, but this approach is being progressively challenged [5]. While these aspects are important, we think there are other, more functional, aspects that also should be considered, especially in a conversational setting. One such aspect is the way in which the voice may help to coordinate the interaction between the participants, more specifically turn-taking. It is important to model turn-taking accurately in spoken dialog systems (SDS), in order to avoid long response delays or inadvertent interruptions (from both the user and the system). The failure of appropriate timing can substantially deteriorate the quality of the interaction [6]. It is well known that humans send and receive various turn-yielding and turn-holding cues to coordinate turn-taking [7, 8, 6]. For example, a syntactically or semantically incomplete phrase may signal a turn-hold, whereas a complete phrase may be turn-yielding [9]. A filled pause is a strong cue to turn-hold [10, 11]. When syntax and semantics is ambiguous, prosody, gestures, or gaze can also be informative [8]. With regards to prosody, flat pitch, higher intensity and longer duration are associated with turn-hold, whereas a rising or falling pitch, lower intensity and shorter duration are associated with yielding the turn. These cues allow humans to take turns with very small gaps (around 200ms), while avoiding large overlaps [12].

The modeling of turn-taking in conversational systems has so far mostly focused on detecting turn-taking cues in the user's voice, in order to allow the system to take turns or give backchannels at appropriate places [6]. However, it is equally important that the system's voice also exhibits accurate cues, so that the user knows when to take the turn and when to allow the system to finish its turn. This is problematic in current TTS systems, as there is typically no control over these cues. Thus, there is a risk that the user might interrupt the system at the wrong places (unintended barge-in), or that the user in other ways gets confused over the allocation of the floor.

To evaluate turn-taking cues in TTS, one option could be to ask human raters to listen to TTS samples and ask them to press a button when they expect a turn-shift, similar to psycholinguistic experiments on understanding turn-taking cues in human speech [13, 14]. However, this is clearly a costly method and not feasible for large-scale evaluations. Thus, an automatic method would be desirable, that could complement other automatic evaluation metrics, such as MOSNet [15] and ASR [16].

In this paper, we introduce an automatic method for evaluating turn-taking cues in conversational TTS, based on Voice Activity Projection (VAP), a turn-taking model proposed by [17]. The model has been shown to outperform prior work [18] that demonstrated that computational turn-taking models trained in a self-supervised fashion perform better than humans on predicting the next speaker on recorded data. Prior work has also shown that the VAP model is sensitive to prosodic cues in synthesized speech [19], as well as filled pauses [20]. Here, we use the model to assess the likelihood that a user would take the turn at each frame in a synthesized spoken utterance. This way, we can make sure that this likelihood is as low as possible while the system is supposed to have the turn (i.e., it has more things to say before yielding), while it should be as high as possible towards the end of the turn. While we only focus on offline evaluation here, such a model could potentially also be used as a training objective when developing the TTS. The VAP model[1] and listening samples[2] are publically available.

2. Automatic Evaluation Method

Voice Activity Projection [17] (VAP) is a training objective where the voice activity (VA) of two speakers are predicted in-

[1] https://github.com/ErikEkstedt/VoiceActivityProjection

[2] https://erikekstedt.github.io/vap_tts

10.21437/Interspeech.2023-2064

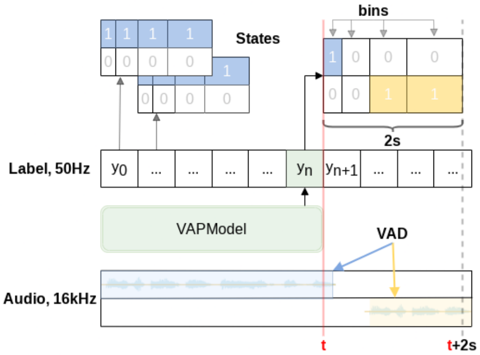

Figure 1: *The VAP model receives a stereo channel audio input and predicts a discrete label, y, at frame, n, corresponding to time, t. Each label represents a state that consists of 8 binary bins, 4 for each speaker, spanning the next 2s of dialog.*

crementally (left-to-right) over the course of a dialog. The VA is defined in binary terms (speech/no-speech), and the two speakers' future activities are jointly encoded into a discrete state that represents the upcoming 2s of dialog. The states are defined by discretizing the 2s windows of activity into eight smaller sub-state-bins, four for each speaker, of increasing duration (0.2s, 0.4s, 0.6s, 0.8s) and are considered active if they contain a majority of VA. This discretization step result in $2^8 = 256$ possible discrete states (labels) to predict during training, see Figure 1.

The VAP model used in this experiment is a stereo version of the original VAP model that operates on two separate waveforms (one for each speaker) and is trained on the Fisher part 1 and the Switchboard corpus [21, 22]. The model consists of a CPC-encoder [23, 24] that extracts framewise representations from the raw audio followed by a 4-layer transformer [25] decoder with cross-attention between the two speaker channels (5.79M parameters).

During inference we scale each sub-state-bin, with their associated label probability, and combine all contributions to a single aggregate state representation. We define two separate probabilities representing the next speaker predictions inside the most immediate region, P_{now} (0-600 ms), and the more distant future, P_{fut} (600-2000 ms), i.e. the first and last four sub-state-bins, respectively. The probabilities are normalized across speakers to produce a final value between 0 and 1 that represent the prediction probability of speaker A being active in the corresponding region.

For our automatic evaluation method, we use the VAP-model to assess these predictions during pauses, when the system should hold the turn, and towards the end, when the system should yield the turn. In this way, the VAP model works as as a user model. In the more specific scenario we investigate in this paper, we focus on turns containing two sentences – a statement and a question – which will naturally include a pause that can be mistaken for a turn-shift and can cause the user to “interrupt” the system, if the TTS generates ambiguous turn-yielding signals. Furthermore, we are interested in whether the user would not just be able to detect that the turn has been yielded, but also *predict* that the turn is about to be yielded (see Figure 2 for an example of this).

Given the $P_{now/fut}$ values over the generated turns we define four metrics covering different regions. First, we focus on the pause between the sentences (red vertical lines) and consider it a **Weak Hold** if the long term prediction P_{fut} favors the agent. It's considered weak because it allows for the user to be the most likely speaker in the short term (P_{now}), corresponding to an invitation of a quick back-and-forth, like a user backchannel or acknowledgment [26]. Second, we define a **Strong Hold** to be the subset where both the P_{now} and P_{fut} values favors the agent. Third, we focus on the last 600 ms of speech (between the orange and the first green line) and define it as an **Early Yield** if P_{fut} favors the user. Lastly, the silence after the turn (green lines) is a **Late Yield** if both P_{now} and P_{fut} predicts user activity. Arguably, the desired outcome for a conversational system planning to say two sentences would be to send clear turn-holding signals before the pause, while signaling that the turn is about to be yielded towards the end.

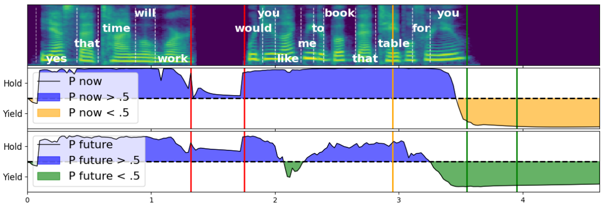

Figure 2: *Example (Amazon TTS): ”Yes that time will work. Would you like me to book it for you?”. From the top, we show the mel-spectrogram, the P_{now}, and P_{fut} values. The dashed black line signifies equal speaker probability, with blue areas (hold) above and yellow or green (yield) below. Vertical lines mark the pause (red), Early Yield (orange, green), and Late Yield (green) zones. The model labels the pause as a Weak and Strong Hold and the end as an Early and Late Yield.*

3. Experiment

We consider a hypothetical spoken dialog system (SDS) setup where a TTS model is used to generate speech in a task-oriented setting. To simulate this setup we extract text data from the MultiWoZ corpus [27]. The corpus contains over 10,000 annotated written dialogs covering 8 different domains where two humans have a fictional interaction where one has the role of a “user” and the other a ”clerk“, here referred to as the agent. The users have goals and several sub-goals to complete over the course of a dialog, such as booking a hotel, a taxi, a train or a restaurant, among others.

We extract agent turns that consist of a sentence pair where a statement, **SD**, (ending with a period) is followed by a question, **Q**, (ending with a question mark). For simplicity, and to avoid unexpected behavior, we omit sentence pairs that includes commas, digits, contains less than 5 words per sentence, and where the total amount of characters are less than 50 or more than 250. Finally, we only keep sentence pairs where both sentences end with a word containing a single syllable, in order to simplify the prosody manipulation in section 3.2. In total, we extract 1482 sentence pairs, such as the one shown in Figure 2.

3.1. Text-To-Speech

We utilize three popular ASR services, namely Amazon[3], Google[4] and Microsoft[5] and use their most natural and conversational american english neural TTS engines: “Joanna-

[3] https://aws.amazon.com/polly/

[4] https://cloud.google.com/text-to-speech

[5] https://azure.microsoft.com/en-us/products/cognitive-services/text-to-speech

neural-en-US" (Amazon), "en-US-Neural2-C" (Google) and "en-US-JennyNeural" (Microsoft). Additionally, we include the open-source FastPitch (**FP**) model[6] [28] (46.27M parameters), trained on the LJSpeech[7] corpus, using the pre-trained checkpoints available to the speech community at large. We also trained a conversational TTS on the widely known Tacotron 2 (**TT2**) architecture [29] (28.21M parameters), with modifications from [30] that allow for control of duration and pitch at the phoneme and word level. The voice is trained on the ThinkComputers Corpus [4], a corpus created from the recordings of a podcast which is made available in the public domain[8]. For the FastPitch and modified Tacotron 2 systems, the speech signal is decoded using the neural vocoder HiFi-GAN [31] (13.93M parameters).

3.2. Results

We generate speech for the extracted turns and normalize the pause (and the Late Yield) duration to be 400 ms, using forced alignment [32], to focus on the prosodic signals of the speech rather than varying lengths of the pauses. After the silence normalization we apply the VAP model and extract the proposed metrics over the regions of interest. Additionally, we approximate the intelligibility and conceived naturalness of the generated speech using the open-source ASR model Whisper[9] (large) to extract the word error rate, WER, and the automatic MOS predictor [33] that output a score between 1 and 5.

All systems signal a Weak Hold over the vast majority of pauses, where TT2 achieves the highest score of 97% and Google the lowest score of 79%, see Table 1. A lower Weak Hold score raises the risk that a system prematurely signals the end of its turn. For the Strong Hold metric, TT2 outperforms the other systems, producing 93% Strong Holds, while Amazon and FP produce around 30%, and Google and Microsoft achieve around 20%. All systems except TT2 have a high risk of inviting user activity inside the pause, which, if not handled correctly by the SDS, could result in a dialog breakdown. As discussed earlier, a Weak Hold might be acceptable (or even desirable, if the user is invited to give a backchannel) for an interactive conversational system. This means that context of the SDS interaction have to be taken into consideration [5] and factored in to the analysis of the importance of the Strong Hold metric in general.

Table 1: *The aggregate metrics for the ORIGINAL speech. All values are percentages (%) except for the MOS.*

Metric	**AMZN**	**GGL**	**MSFT**	**FP**	**TT2**
Weak Hold↑	87	79	86	84	**97**
Strong Hold↑	31	21	20	30	**93**
Early Yield↑	**44**	28	32	5	4
Late Yield ↑	**95**	**95**	90	39	14
MOS↑	4.2	4.3	**4.8**	4.4	3.9
WER ↓	2.7	2.7	**2.3**	5.6	5.2

Towards the end of the turns, Amazon outperforms all other systems and provides Early Yield signals for 44% of the samples. During an interaction this could allow for better conversational flow by preparing the users for their expected upcoming turn. After the end of the generated speech, all commercial systems signal a Late Yield for 90-95% of the samples. This means that while the systems may not always provide early prosodic turn-yield signals, they do convey their intention to yield the turn upon its completion. However, both TT2 and FP heavily under-perform on this metric and provide 14% and 39% Late Yields and only 4% and 5% Early Yields, respectively. This may be explained by the fact that FP is trained on read speech, and while TT2 is trained on conversational speech the content is largely monologic, where a single speaker talks at length about a topic with minimal input from the interlocutor.

According to the naturalness scores, Microsoft outperforms all systems averaging a MOS of 4.8, while the least natural voice was TT2 with a score of 3.9. Microsoft also provides the most intelligible speech with a WER of 2.3%, Amazon and Google are slightly worse with 2.7% WER, while FP and TT2 are less intelligible, averaging 5.6% and 5.2%.

Text Manipulation: Because TTS models are commonly trained on text that includes punctuation, we provide two different input permutations of each agent turn that can induce different prosodic realizations at the end of the SD sentence. Whereas a period denotes the end of a sentence, commas are used to separates parts within them, and can potentially condition the TTS systems to generate stronger hold cues. Filled pauses (or *fillers* for short), such as "um", are also well known to be a strong cue to turn hold [10, 11]. To study if such manipulations would indeed have a turn-holding effect, we experiment by replacing the ending period with either a comma or with the filler "um," (also ending with a comma).

The comma prompt produced an increase of Weak Holds, for all models (except TT2), and raised the Weak Hold frequency to over 90% for each system except Google (83%), as seen in Table 2. The FP model showed the largest relative improvement, classifying 56% of the pauses as a Strong Hold. Microsoft had the second largest gain, improving its Strong Hold classification from 20% to 33%, outperforming Google (29%), but falls short of Amazon with 48%. The large improvement for FP could be due to it being trained on read speech, where the source text is grammatical and contains a lot of punctuation, making the prompt conditioning prominent. However, TT2 was largely unaffected by the change of prompt which, similarly, could be because it is trained on spontaneous speech, that naturally contains less punctuation, making it less sensitive to punctuation in general. The results indicate that while the effect of changing periods to commas provides stronger hold signals, the strength varies depending on the systems. Furthermore, given the simplicity and the straight-forward meaning of punctuation control it could be beneficial for spontaneous TTS to include such examples in their training data.

Interestingly, while the manipulation only changes the punctuation prior to the pause, the later yield signals are affected as well. The cause of this effect can be because of differences in the generated speech at the end of the questions or how the entire turn is perceived. Because the VAP model is trained on 20 s segments of continuous dialog, it can learn to utilize context over multiple sentences to infer the yield probabilities. However, the effect is small across all systems and does not change the general outcome in a significant way. Lastly, we note that both the intelligibility and naturalness metrics are roughly unaffected by the manipulation.

While changing the punctuation does not introduce any ad-

[6] https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/SpeechSynthesis/FastPitch

[7] https://keithito.com/LJ-Speech-Dataset/

[8] https://archive.org/details/podcasts_miscellaneous Creator: ThinkComputers

[9] https://github.com/openai/whisper

Table 2: *The aggregate metrics for the COMMA permutation. All values are percentages (%) except for the MOS.*

Metric	AMZN	GGL	MSFT	FP	TT2
Weak Hold↑	92	83	90	90	**97**
Strong Hold↑	48	29	33	54	**92**
Early Yield↑	**39**	28	28	6	5
Late Yield↑	92	**94**	89	43	18
MOS↑	4.2	4.3	**4.8**	4.4	3.8
WER ↓	2.8	2.8	**2.2**	5.4	6.1

ditional words or letters to the prompt, inserting a filler does. Whereas a filler can be considered to function as a word [10], it is not guaranteed that all TTS systems include them in their training, and the realization of their characteristic hesitation-like properties [34] can vary between systems. Table 3 shows that both Microsoft and TT2 achieve 100% Strong Hold when a filler is included. While TT2 shows strong hold cues in both the original and comma permutations, the filler drastically strengthens the hold for the Microsoft voice. From listening to the generated filler samples we note that both Microsoft and TT2 have learned to generate fillers in a prosodically relevant way. This is reasonable considering the spontaneous nature of the TT2 training data and distinguishes Microsoft from the other commercial services. These results show that if a TTS has the ability to produce fillers, it can be a very efficient way to avoid user barge-ins.

Table 3: *The aggregate metrics for the FILLER permutation. All values are percentages (%) except for the MOS.*

Metric	AMZN	GGL	MSFT	FP	TT2
Weak Hold↑	93	77	**100**	92	**100**
Strong Hold↑	57	26	**100**	69	**100**
Early Yield↑	**38**	27	23	6	5
Late Yield↑	92	**95**	88	38	21
MOS↑	4.2	4.2	**4.7**	4.3	3.8
WER ↓	6.5	4.9	**3.1**	7.5	7.2

Prosody Manipulation: As an alternative to text manipulation, we experiment with direct manipulation of the prosody prior to the pause, to achieve prosodic signals that are associated with turn-hold, according to the literature [6, 35]. We exclusively target the last syllable of the SD sentence, corresponding to the last word given the text extraction step, and raise the intensity, make the duration longer and flatten the intonation. We consider two approaches where we (1) apply post-processing on the original generated speech, using Praat [36] and torchaudio, and (2) use the FP and TT2 innate ability to control for these properties explicitly in the generation step.

The post-processing produces a substantial increase of Strong Holds for both Microsoft (81%) and Amazon (70%), while Google (53%) is less affected, see Table 4. This indicates that Google provides more ambiguous signals earlier on in the statement and requires greater intervention to produce consistent turn-holding aspects over the course of the turn. While TT2 consistently produces strong hold cues, FP shows a relative improvement similar to Microsoft and increases its Strong Hold performance to 85%. Overall, the results highlight the possibility for TTS systems to improve their generative capabilities w.r.t. the functional turn-taking aspects of conversational speech.

Table 4: *The aggregate post-processing metrics. All values are percentages (%) except for the MOS.*

Metric	AMZN	GGL	MSFT	FP	TT2
Weak Hold↑	**97**	92	96	97	**99**
Strong HOLD↑	70	53	81	85	**98**
Early Yield↑	**46**	26	29	6	4
Late Yield↑	95	**97**	95	42	18
MOS↑	3.8	3.9	**4.6**	4.0	3.6
WER ↓	2.7	2.5	**2.4**	3.9	5.2

Post-processing inadvertently introduces artifacts that affects the perceived naturalness, reflected by the consistent decrease of the MOS across all TTS systems. The alternative approach, controlling the prosody directly in the generation process, alleviates this negative impact, where FP produces MOS of 4.3 instead of 4.0 and TT2, 3.7 instead 3.6, as compared to the post-processing. However, the hold signals are less prominent and the Strong Hold score for TT2 (93%) is unaffected (as compared to the original), while FP produces the same score as for the simpler comma prompt (54%).

4. Conclusion

We introduced a new automatic evaluation method that can measure a TTS model's ability to produce turn-taking cues. We show that although commercial TTS systems often do provide appropriate turn-taking cues, they still produce ambiguous, or opposite, cues for up to 21% of pauses (Google) and up to 10% of yields (Microsoft). Furthermore, we show that TTS models trained on read or spontaneous speech are generally good at producing turn-holding cues but show low performance w.r.t. turn-yielding cues. Replacing a period with a comma is a simple approach to condition TTS models to provide stronger turn-holding cues. If a TTS model has the ability to correctly generate fillers, they can be added to a prompt to strongly convey the agent's intention to continue its turn. By directly manipulating the prosody, we show the possibility of improvement for conversational TTS to produce turn-taking cues independent of the provided input prompt.

While controllable TTS has mainly focused on prosodic control [28], or style control (such as emotion or speaker) [37], we argue that controlling for turn-taking cues could provide added benefits useful for conversational speech. A VAP model could even provide turn-taking signals to be optimized during training, that could enable the ability to generate appropriate turn-taking cues without relying on additional datasets.

This work is breaking down barriers by providing a free, open-source, model that can be used to evaluate the generative turn-taking capabilities of conversational TTS systems. It enables researchers with limited resources to conduct fast model iteration, without relying on expensive human evaluations, and is a step towards making speech research more equal in general.

5. Acknowledgements

This work was supported by the Riksbankens Jubileumsfond (RJ) project (P20-0484), the Swedish Research Council projects (2020-03812) and (VR-2019-05003), and finally the Digital Futures project, *AAIS*.

6. References

[1] J. O'Mahony, C. Lai, and S. King, "Combining conversational speech with read speech to improve prosody in text-to-speech synthesis," in *Proc. Interspeech*, 2022.

[2] H. Guo, S. Zhang, F. K. Soong, L. He, and L. Xie, "Conversational end-to-end tts for voice agents," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 403–409.

[3] T. Raitio, R. Rasipuram, and D. Castellani, "Controllable neural text-to-speech synthesis using intuitive prosodic features," in *Proc. Interspeech*, 2020, pp. 4432–4436.

[4] É. Székely, G. E. Henter, J. Beskow, and J. Gustafson, "Spontaneous conversational speech synthesis from found data." in *Proc. Interspeech*, 2019, pp. 4435–4439.

[5] P. Wagner, J. Beskow, S. Betz, J. Edlund, J. Gustafson, G. Eje Henter, S. Le Maguer, Z. Malisz, É. Székely, C. Tånnander *et al.*, "Speech synthesis evaluation—state-of-the-art assessment and suggestion for a novel research program," in *Proc. SSW10)*, 2019.

[6] G. Skantze, "Turn-taking in Conversational Systems and Human-Robot Interaction : A Review," *Computer Speech & Language*, vol. 67, p. 101178, 2021, publisher: Elsevier Ltd.

[7] H. Sacks, E. Schegloff, and G. Jefferson, "A simplest systematics for the organization of turn-taking for conversation," *Language*, vol. 50, pp. 696–735, dec 1974.

[8] S. Duncan, "Some Signals and Rules for Taking Speaking Turns in Conversations," *Journal of Personality and Social Psychology*, vol. 23, no. 2, pp. 283–292, 1972.

[9] C. Ford and S. Thompson, "Interactional units in conversation: syntactic, intonational, and pragmatic resources for the management of turns," in *Interaction and grammar*, ser. Studies in interactional sociolinguistics 13, E. Ochs, E. Schegloff, and A. Thompson, Eds. Cambridge: Cambridge University Press, 1996, ch. 3, pp. 134–184.

[10] H. Clark and J. E. Fox Tree, "Using uh and um in spontaneous speaking," *Cognition*, vol. 84, no. 1, pp. 73–111, 2002.

[11] P. Ball, "Listeners' responses to filled pauses in relation to floor apportionment," *British Journal of Social and Clinical Psychology*, vol. 14, no. 4, pp. 423–424, 1975.

[12] S. C. Levinson and F. Torreira, "Timing in turn-taking and its implications for processing models of language," *Frontiers in Psychology*, vol. 6, p. 731, 2015.

[13] J. P. De Ruiter, H. Mitterer, and N. J. Enfield, "Projecting the end of a speaker's turn: A cognitive cornerstone of conversation," *Language*, vol. 82, pp. 515 – 535, 2006.

[14] S. Bögels and F. Torreira, "Listeners use intonational phrase boundaries to project turn ends in spoken interaction," *Journal of Phonetics*, vol. 52, pp. 46–57, 2015.

[15] J. Williams, J. Rownicka, P. O. Gallegos, and S. King, "Comparison of speech representations for automatic quality estimation in multi-speaker text-to-speech synthesis," in *Odyssey 2020 The Speaker and Language Recognition Workshop*. ISCA, 2020, pp. 222–229.

[16] J. Taylor and K. Richmond, "Confidence intervals for asr-based tts evaluation," in *Proc. Interspeech*, 2021, pp. 2791–2795.

[17] E. Ekstedt and G. Skantze, "Voice Activity Projection: Self-supervised Learning of Turn-taking Events," in *Proc. Interspeech 2022*, 2022, pp. 5190–5194.

[18] G. 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*. Saarbrücken, Germany: Association for Computational Linguistics, 2017, pp. 220–230.

[19] E. Ekstedt and G. Skantze, "How Much Does Prosody Help Turn-taking? Investigations using Voice Activity Projection Models," in *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Edinburgh, UK: Association for Computational Linguistics, 2022, pp. 541–551.

[20] B. Jiang, E. Ekstedt, and G. Skantze, "What makes a good pause? investigating the turn-holding effects of fillers," in *Proc. ICPhS*, 2023.

[21] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing*. USA: IEEE Computer Society, 1992, p. 517–520.

[22] C. Cieri, D. Miller, and K. Walker, "The Fisher Corpus: a resource for the next generations of speech-to-text," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal: European Language Resources Association (ELRA), May 2004.

[23] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[24] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," 2020.

[25] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5998–6008.

[26] V. H. Yngve, "On getting a word in edgewise," *Chicago Linguistics Society, 6th Meeting, 1970*, pp. 567–578, 1970.

[27] X. Zang, A. Rastogi, S. Sunkara, R. Gupta, J. Zhang, and J. Chen, "MultiWOZ 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines," in *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, ACL 2020*, 2020, pp. 109–117.

[28] A. Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6588–6592.

[29] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[30] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *Proc. ICASSP*, 2020, pp. 6189–6193.

[31] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[32] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," in *Proc. Interspeech 2017*, 2017, pp. 498–502.

[33] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, "Generalization ability of mos prediction networks," in *Proc. ICASSP*, 2022, pp. 8442–8446.

[34] J. E. Fox Tree, "Interpreting Pauses and Ums at Turn Exchanges," *Discourse Processes*, vol. 34, no. 1, pp. 37–55, Jul. 2002.

[35] A. Hjalmarsson, "The additive effect of turn-taking cues in human and synthetic voice," *Speech Commun.*, vol. 53, pp. 23–35, 2011.

[36] P. Boersma and D. Weenink, "Praat: Doing phonetics by computer," 2022.

[37] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.