11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Acquiring conversational speaking style from multi-speaker spontaneous dialog corpus for prosody-controllable sequence-to-sequence speech synthesis

Avrech Ben-David[1]*, *Slava Shechtman*[2]

[1]Technion – Israel Institute of Technology, Haifa – Israel
[2]IBM Haifa Research Lab, Haifa – Israel

avrech@campus.technion.ac.il, slava@il.ibm.com

Abstract

Sequence-to-Sequence Text-to-Speech (S2S TTS) architectures that directly generate low level acoustic features from phonetic sequence are known to produce natural and expressive speech, when provided with moderate-to-large amounts of high quality training data. When exposed to a sequence of coarse speaker-agnostic *prosodic descriptors*, such systems become prosody-controllable and can learn and transfer desired prosodic patterns (e.g. word-emphasis or speaking style) from one seen speaker to another (in multi-speaker settings).

But what if a high quality speech corpus for a desired speaking style is not available? In this work we explore the feasibility of teaching a neutral pre-trained prosody-controllable S2S TTS voice to speak with a conversational speaking style, as learnt from a low-quality multi-speaker spontaneous dialog corpus (originally intended for Automatic Speech Recognition). We have found that it is absolutely necessary to incorporate word semantics for that task. We fine-tune BERT network to predict the *prosodic descriptors* from the input text, based on that corpus, and apply them to the prosody-controllable S2S TTS at inference time. The subjective listening tests revealed that the learnt conversational style rated higher than baseline for 68% of the stimuli under test. The overall quality and naturalness rated higher than baseline in 64% of the stimuli under test. The improvement came mostly as a result of improving common conversational speech patterns, such as filler words and phrases. However, the overall MOS did not significantly improve due to less convincing realization of the rising intonation on declarative statements (*uptalk*).

Index Terms: expressive speech synthesis, sequence to sequence speech synthesis, conversational speech synthesis

1. Introduction

Sequence-to-Sequence Text-to-Speech (S2S TTS) architectures [1] [2] that directly generate low level acoustic features from phonetic sequence are known to produce natural and expressive speech, if provided with sufficient amount of high quality training data, covering a variety of speakers and speaking styles. Apparently, additional high quality expressive data is required for the S2S TTS to acquire a new speaking style for existing voices, to perform model retraining or adjustment.

But what if a high quality speech corpus for a desired speaking style is not available? Acquiring a new speaking style for existing voices in a pre-trained S2S TTS remains a hot research topic. Cross-speaker speaking style transplantation by means of style encoding of a single reference utterance, as proposed initially in [3], partially achieved that goal. However, it worked mostly when the text of the reference utterance closely matched the text to be synthesized [3], thus making this method less appropriate for standard TTS applications. Since then, several methods have been developed to apply various speaking styles, in unsupervised [4] or semi-supervised [5] configurations, when style encoding is jointly trained with S2S TTS. Such settings require high quality speech data to deduce speaking styles from the corpus. In this work, on the contrary, we explore the feasibility of acquiring an unseen speaking style from a readily available low-quality speech corpus, that seems unsuitable for high quality TTS purposes.

The style that we'd like to obtain is a conversational speaking style. Recently, a single-speaker conversational S2S TTS system has been proposed, trained on a proprietary high-quality spontaneous data corpus, recorded particularly for that purpose [6]. In this work, on the other hand, we'd like to explore a less expensive approach, making use of existing data. For that purpose we selected *Switchboard* [7], a well-known multi-speaker corpus of narrow-band spontaneous speech, originally intended for Automatic Speech Recognition (ASR) development purposes. It is a corpus of spontaneous conversations of telephone bandwidth speech. The corpus contains 2430 conversations averaging 6 minutes in length, spoken by over 500 US English speakers. The calls are manually transcribed and then submitted to an ASR system to establish approximate time alignments at the word level [7]. The recordings are truly spontaneous, with a lot of background noises, prolonged pauses, word repetitions, filler words, paralinguistics and burst-ins.

As such, this corpus cannot serve directly for high quality S2S TTS system training, but would rather be utilized for certain intermediate style representation. Considering the data set characteristics, this representation should be 1) purely prosodic, as *Switchboard*'s general spectral characteristics are very different from that of high quality wide band studio recordings used for S2S TTS voices, and 2) speaker- and gender-agnostic, as this multi-speaker dataset has only few stimuli per speaker available and we want to learn general conversational style aspects. Fortunately, our prosody-controllable S2S TTS architecture, originally proposed for unsupervised/weakly-supervised word-emphasis realization [8] is suitable for that purpose. In this architecture we condition the speech synthesis on an intermediate prosodic representation, *Hierarchical Prosodic Controls (HPC)*, comprising a sequence of hierarchical (word- and sentence- level) prosodic observations, designed to be gender- and speaker-agnostic.

In the current work we deploy an extended set of HPC parameters to better fit the desired speaking style application and design an HPC predictor model, trained on *Switchboard*. The predicted HPC sequences are utilized to condition the pre-trained prosody-controllable S2S TTS to convey the desired conversational speaking style. We explore various alternatives for conversational HPC prediction from phonetic and/or textual

*Work performed as an internship at IBM

10.21437/SSW.2021-12

input, incorporating LSTM [9] and/or BERT [10] networks. We introduce the system architecture in Sec. 2, detail on training procedure in Sec. 3 and present system evaluation in Sec. 4. Concluding remarks are provided in Sec. 5.

2. Architecture

2.1. Prosody-Controllable S2S TTS

The Sequence to Sequence Text To Speech model architecture, adopted in this work (Fig. 1), mostly follows the prosody-controllable S2S model originally proposed for unsupervised/weakly-supervised word-emphasis realization [8]. It is based on a Tacotron2 S2S acoustic model [2], augmented with Hierarchical Prosodic Controls [8]. The S2S acoustic model generates a sequence of acoustic feature vectors (composed of mel-cepstral and periodicity components [11, 12]), where each vector corresponds to a constant-length speech frame, that are then fed to an independently trained LPCNET-based neural vocoder [11] to generate high-quality samples in real time [12]. The inputs to the system are a set of symbolic sequences extracted from the input text by a rules-based TTS Front End module (adopted from a unit selection system [13]). All input sequences are aligned (by repetition) to contain the same number of symbols and are *one-hot* coded. The input sequences comprise:

- (A) phone identity (including silence phone) together with its lexical stress (primary, secondary or no stress)
- (B) phrase type (4-way: intermediate, declarative, interrogative, exclamation)

All the symbolic sequences are augmented with a special symbol for word boundary, inserted between the words with no silence between them. The *one-hot* coded input sequences are converted to a set of linear embeddings, concatenated together, and fed into Tacotron2 Encoder module (C), consisting of convolutional and bidirectional Long Short-Term Memory (Bi-LSTM) layers [2]. A global utterance-level speaker embedding (E), broadcast over the length of the sequence, is concatenated to the encoder output.

A set of Hierarchical Prosodic Controls, extended from the one introduced in [8] (and further elaborated in Sec. 2.2), is designed to enable both the sentence-level and the word-level modifications needed to realize the prosodic patterns associated with various speaking styles. They are designed to be speaker-agnostic to ease cross-speaker style transplantation. During training these prosodic controls are extracted from the target waveforms (E), while at inference time a separate predictive module (D) steps in to provide default predictions for the hierarchical prosodic trajectories.

The Decoder is an autoregressive network that largely follows the standard Tacotron2 architecture [2], but with modifications on the attention mechanism, choice of targets, and training losses. These are described in detail in [12] and briefly summarized as follows. The attention is an augmented two-stage attention where the hybrid content- and location-based attention mechanism of Tacotron2 [2] is followed by a structure-preserving mechanism encouraging monotonicity and unimodality in the alignment matrix [12]. The model is trained in a multi-task fashion to predict the end-of-sequence indicator and 80-dim mel cepstral features [2] in tandem with the parameters needed as inputs for an independently trained LPCNET neural vocoder [11]. For 22kHz signals, these features (which we denote as "LPC features") correspond to 256 waveform samples and consist of a 22-dim vector with 20 mel-cepstral coefficients, log f0 and f0 correlation. The predicted LPC features are also processed with two post-nets (one to refine the mel-cepstrum, and one to refine the pitch parameters). As opposed to [8, 12], the autoregressive feedback mechanism in the decoder is kept unmodified from the original Tacotron2 architecture.

Let y_t^M and y_t^L represent the target sequences for the mel and LPC tasks respectively, $\tilde{y}_t^M$ and $\tilde{y}_t^L$ their final predictions, and $\hat{y}_t^L$ the "intermediate" LPC-feature prediction (before the post-net). Then the combined acoustic loss function is used to train the system:

$$\mathcal{L} = MSE(\tilde{y}_t^M, y_t^M) + 0.8MSE(\hat{y}_t^L, y_t^L) + 0.4MSE(\tilde{y}_t^L, y_t^L) + 0.4MSE(\Delta\tilde{y}_t^L, \Delta y_t^L), \quad (1)$$

where the Δ operator applies the first difference in time to a sequence, and $MSE(,)$ is the mean-squared error. The above combined acoustic loss is added to the end of-sequence indicator cross-entropy loss [2] to yield the total training loss. For the sake of space, we omit some detail in this exposition, and refer the reader to [14, 12] for additional background and formulae.

The default prosodic-control predictor predicts the HPCs from the Front-End Encoder outputs of the S2S acoustic model (Fig. 1) and consists of stacked Bi-LSTMs (3x128), terminated with a linear layer. The predictor is trained separately (after the training of the main model has ended and all its weights are frozen) with *global MSE loss* of the combined HPC sequence (see section 2.2) and ADAM [15] optimizer. At inference time, the predictions of the prosodic-control subnet are rectified to be piecewise constant as the oracle values that the S2S system was trained with. To that end, a mean pooling function is applied to the prediction to be constant between the (known) sentence and word boundaries.

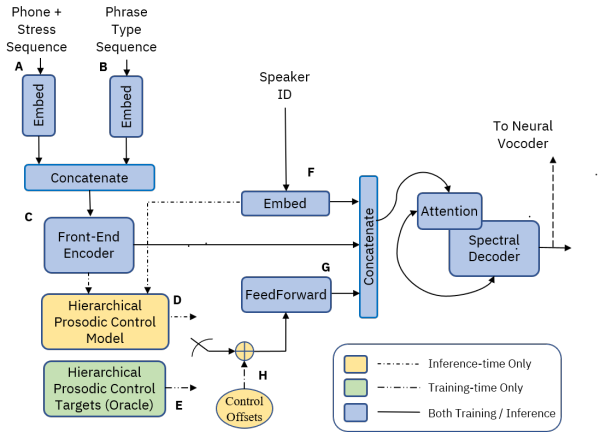

Figure 1: *Multi-speaker S2S synthesis acoustic model (phones to spectra) with Hierarchical Prosodic Controls.*

2.2. Hierarchical Prosodic-Control parameters

In this work we extend a set of four perceptually-interpretable prosodic measurements introduced in [8], evaluated over sentence and word intervals. The sentence-level components represent general speed and expressiveness [14], while the word-level components [8] represent fine-grained prosodic structure. In this work we extend the reported set of prosodic measurements [8] with two more pitch slope components to better suit

for speaking-style modeling. Altogether, we make use of the following statistics:

- S_{dur}: The log of the average per-phone durations, along a sentence (and excluding any silence).
- $S_{\Delta f_0}$: The f_0 dynamics (i.e., the difference between the 95- and 5-percentiles of log-f_0), along a sentence.
- $S_{\angle f_0}$: The log-f_0 linear regression slope along a sentence (excluding any silence).
- W_{dur}: The log of the average per-phone durations (as above), along each word.
- $W_{\Delta f_0}$: The f_0 dynamics (as above), along each word.
- $W_{\angle f_0}$: The log-f_0 linear regression slope (as above), along each word.

Note that the average per-phone durations in the above definitions are estimated as the duration (in seconds) of the relevant spans (word or sentence) divided by the number of phone symbols contained therein, and that therefore no fine-level phonetic alignment is required in the computation (only coarse word-level alignments and either phonetic transcriptions or a dictionary). The above sentence- and word-level properties are propagated down to the temporal granularity of the phonetic encoder outputs (i.e., phones) to form piecewise functions that are constant within a (sentence or word) unit. From this we define the following six-component prosodic-control target vector:

$$P = Norm_{\sigma}\{[S_{dur}, S_{\Delta f_0}, S_{\angle f_0}, W_{dur} - S_{dur}, W_{\Delta f_0} - S_{\Delta f_0}, W_{\angle f_0} - S_{\angle f_0}]\}, \quad (2)$$

where $Norm_{\sigma}\{\}$ is the linear map $[-3\sigma^2, 3\sigma^2] \rightarrow [-1, 1]$, and σ^2 is the global (multi-speaker corpus-wide) variance for each of the statistics in P. Note that all measurements in P are gender-agnostic.

2.3. Conversational HPC Prediction Model

For each utterance we predict an HPC sequence, comprising sentence- and word-level pitch- and duration-related features, as described in section 2.2. Silences are treated as special words for which only duration-related features are required. In order to adapt the prosodic controls to the conversational context, we consider combining various inputs: the input text, its phonetic sequence encoding, textual dialog context (casual) and prosodic dialog context, represented by HPCs, extracted from the past dialog audio. We use an encoder-decoder architecture, comprising the encoder (Fig. 2) that converts all the input streams into a sequence of context-aware word-embeddings, followed by a decoder (Fig.3) that contains three dedicated HPC decoder networks: word-level HPCs, sentence-level HPCs and a dedicated decoder for silence words. The encoder consists of three neural models: a BERT network for text processing and two distinct Bidirectional LSTM stacks for processing the phone-encoding and prosodic context input sequences.

2.3.1. Text encoding (with conversational context)

BERT is a widely used language model for language understanding tasks. The pre-trained model is commonly fine-tuned for a few epochs to extract high-quality task-specific word embedding [10]. BERT requires transforming the input text into tokens, e.g. with WordPiece tokenization [16]. In our work we deploy base (uncased) BERT model in its "question answering" configuration; we feed a window of the chat history, i.e. the textual *context*, into BERT's *sentence A* input, while the target utterance is fed as its *sentence B*. Attending to the conversational *context*, BERT produces token embeddings for the target utterance. We average the outputs of the 4 last hidden layers to produce token-representation, and represent each original word by its first token representation. We show in our experiments that all systems incorporating BERT text processing perform significantly better than the system that processes just phone-embeddings. We attribute this improvement to BERT's ability to capture semantic information, and to its robustness to text errors.

2.3.2. Phone sequence encoding

Following [14], we utilize the target utterance's phone sequence encoding generated by the pre-trained Tacotron2 Encoder module (Fig. 1, module C) from the phonetic sequence, to enrich the semantic word representation with its phonetic counterpart. We push the phone encoding sequence into a stack of three 32-dim Bi-LSTMs, and pool to word resolution by either averaging the output along word segments, or by taking the first output element corresponding to each word. The resulting word-level vectors are concatenated to the word-embedding produced by BERT. Note that the phone-sequence processing architecture resembles that of the baseline HPC prediction, as described in Section 2.1

2.3.3. Prosodic context encoding

We hypothesized that certain prosodic information extracted from the dialog history can help to attain better conversational prosody modeling. We extract the HPC sequence of the dialog context audio and feed it into additional stack of three 32-dim Bi-LSTMs. We average the whole output sequence into a single feature vector. This global feature vector is broadcast and concatenated to the target utterance word-vectors. We further apply a 128-dim linear layer followed by GELU [17] activation to each one of those word vectors, and output a 128-dim word embedding.

2.3.4. HPC decoder

The sentence-level, word-level and silence HPC features are decoded independently by three decoder modules, implemented as single linear layers (see Fig. 3). The decoders' outcomes are combined together to comprise the output HPC sequence. The sentence decoder is fed with sentence embeddings, where each sentence embedding is obtained by averaging over its corresponding word embeddings. Once a silence word needs to be inserted after a certain word $w1$, its embedding $we1$ (see Fig. 3) serves also as the embedding for the silence word and is fed to Silence Decoder (see Fig. 3) to obtain the silence HPC component (i.e. a silence duration estimate).

3. Conversational HPC Training

3.1. Data Preparation

The conversational HPC prediction model, excluding the pre-trained frozen parts of BERT and pre-trained S2S Front-End Encoder (see Fig. 2), is trained on Switchboard dataset that contains spontaneous conversations in a weakly controlled setup. As such, its data is challenging and noisy. Besides the speech, various paralinguistic and non-speech events occasionally happen, such as phone call quality distortions, external noise, coughing, laughter, stuttering, self-correction, un-

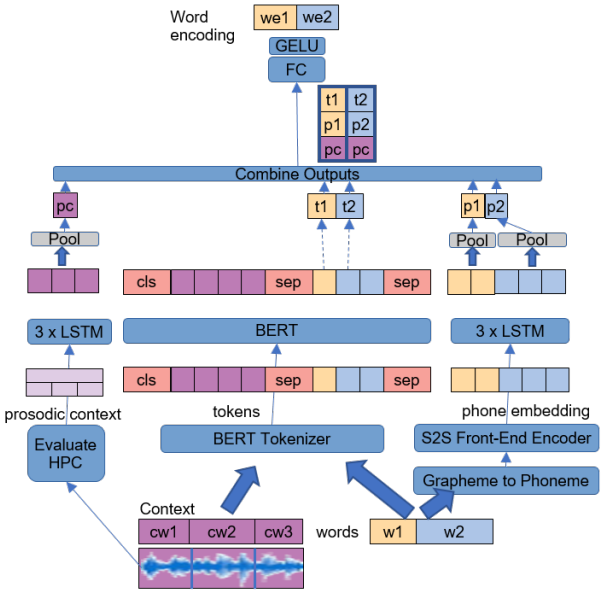

Figure 2: *Conversational HPC Predictor: Encoder. Outputs context-aware word-embedding, based on the word semantics and phonetics*

finished words, prolonged pauses, word and phrase fillers and para-linguistic particles. While the transcription is organized in turns, speakers can actually speak simultaneously. Inevitably, the transcribed text suffers from grammatical errors, incomplete sentences and weird phrasing. Fortunately, most of non-linguistic and para-linguistic events are consistently labeled in the manual transcription (with special signs). After removing all the special transcription words, we generated proper English utterances, inserted commas at each pause and obtained the phonetic sequences with word boundaries from each textual utterance, as described in Section 2.1. Then we compared the word counts in the *Switchboard* word-alignment files with the number of words in our generated phonetic sequence and left out all misaligned utterances. For each speaker turn we demanded to have a valid previous speaker turn, to be able to extract prosodic context by means of HPC. Eventually we retained ∼170K utterances for training, that served us to extract their phonetic sequences, target HPC sequences and prosodic (HPC-based) dialog context. We also held out a development set of 35 utterances for post-training adjustments (see Section 3.3) and a test set of 25 utterances for a listening test evaluation (see Section 4).

3.2. Dialog Context

One of the questions we wanted to explore was whether the dialog context is important for inferring the conversational style from the multi-speaker spontaneous conversational speech dataset (i.e. *Switchboard*).

3.2.1. Textual context

For the textual context (extracted from the past), we explored several context configurations:

- *NONE*: no context is used.
- *Last Turn (LT)*: The last turn and the target utterance are fed to BERT as *A*- and *B*-sequence correspondingly.

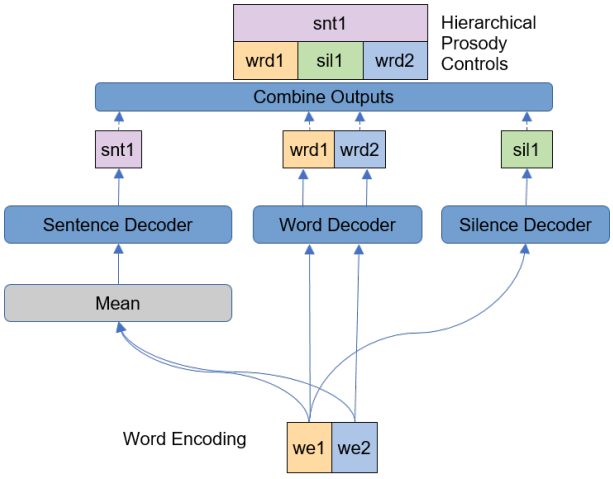

Figure 3: *Conversational HPC Predictor: Decoder*

- *Other Speaker Last Turn (OSLT)*: The context window goes back until the last turn of the other speaker. The text of the context turns is concatenated with period signs between turns, and is fed to BERT *sentence A* as in the LT case.
- *Other Speaker First Turn (OSFT)*: The context window is extended until covering the whole last burst of turns the other speaker said.

For training, we left the original text and punctuation as is, including the repeated words and the paralinguistics. The special event symbols were dropped.

3.2.2. Prosodic context

We concatenate the HPC sequences of the interlocutor's last burst of turns and set it as prosodic context. For the starting turn of each conversation we set the prosodic context to empty sequence. Examples whose prosodic context was missing due to pre-processing errors were discarded.

3.3. Training Procedure

We train our architecture with *MSE* regression loss, as follows:

$$\mathcal{L} = MSE_{word} + c_1 MSE_{sent} + c_2 MSE_{sil}, \qquad (3)$$

where the loss weights c_1, c_2 serve as hyper-parameters tuned to minimize *global MSE loss* of HPC sequence, as used in the baseline S2S TTS system, trained with the high quality speech corpora (see section 2.1).

Since BERT overfits small datasets very quickly, we freeze BERT in the first 5 training epochs while the rest of the architecture starts training. Then, we fine-tune the last two layers of BERT's encoder for 3 consequent epochs, and freeze BERT again for the rest of the training.

In addition, we used bucket batch sampler to equalize the LSTM's input sequence lengths, and let PyTorch-Lightning [18] automate the training process.

We used W&B sweeping tool [19] to apply massive Bayesian hyper parameter optimization. The optimized hyper parameters included the dialog context configuration, the sentence and silence loss coefficients, BERT fine-tuning parameters and others.

For each system we performed a separate hyper-parameter search to select several systems with the N-best losses. Among them we selected the best system by listening to 35 utterances of the held out development set.

We further post-process the predicted HPC features by adding a negative offset of -0.5 to the silence word duration HPC component to avoid too long silences that are common in the spontaneous speech, but disruptive when synthesized with neutral S2S TTS. This post-processing was also tuned on the held out development set. A list of the hyper parameters that was selected for each model under test is listed in Table 1. All the tested systems were trained with batch size of 128 and LSTM dropout of 0.3.

4. Evaluation

The training material for the pre-trained neutral multi-speaker S2S TTS system comprised corpora from three professional native speakers of US English, two females and one male, of 10-17K sentences each. A single female speaker was selected for the conversational evaluation, as it was found more suitable for the conversational speaking style than the others.

Several proposed variants of HPC models were trained separately on *Switchboard* to generate the HPC sequences from the input texts, with or without dialog contexts. Other parts of S2S TTS, besides the HPC model, where pre-trained and kept identical in all the systems.

In a subjective evaluation presented below we would like to assess how well the proposed conversational prosody models help to apply the conversational speaking style, while preserving a decent quality and naturalness of the synthesized speech. To that end, we consider the following systems:

1. **Base**: The baseline neutral S2S TTS system with the default HPC prediction, as learnt from the neutral voice corpora.
2. **Phn**: The neutral S2S TTS with the conversational HPC model, predicting HPC from phonetic sequence only (phones, lexical stress, phrase type, word boundaries).
3. **BERT-TC**: The neutral S2S TTS with the conversational HPC model based on a task-adjusted BERT, predicting HPC from textual input, enriched with textual dialog context.
4. **BERT-Phn**: The neutral S2S TTS with the conversational HPC model to predict HPC, based on phonetic and textual input, no dialog context is fed into BERT.
5. **BERT-Phn-TC**: The neutral S2S TTS with the conversational HPC model that combines that of item 2 and task-adjusted BERT, trained also with the textual dialog context. It predicts HPC, based on phonetic and textual input, enriched with textual dialog context.
6. **BERT-Phn-TPC**: The neutral S2S TTS with the conversational HPC model that combines that of item 2, but is conditioned also on the HPC dialog context, plus a task-adjusted BERT, trained with the input text and the textual dialog context. It predicts HPC, based on phonetic and textual input, enriched with both textual and prosodic dialog contexts.

To evaluate the systems defined above, we conducted a combined Mean Opinion Score (MOS) listening test for the six systems. No natural recordings were included in MOS tests, since no matched conversational utterances existed for the high quality voice. The test examples were taken from *Switchboard* with their original context.

We conducted a crowd-based evaluation (139 subjects) for a held-out test set of 25 sentences. The subjects were asked to rate 1) the overall quality and naturalness of an utterance and 2) *"how well the sound of the voice and the intonation convey the expressive character of a sentence in the context of the provided conversation"*. The subjects chose between five categorical answers (*1 - Poor, 2 - Bad, 3 - Fair, 4 - Good, 5 - Excellent*). The corresponding dialog context transcriptions were provided to the subjects so that they could assess how well the speaking style corresponds to the dialog context[1]. Each stimulus received 35 independent ratings. The raw ratings were subject to an outlier-removal procedure, after which each stimulus retained 31 independent votes on average.

Overall MOS results for 1) quality and naturalness and 2) conversational speaking style correspondence are provided in Table 2. As we requested a relatively large amount of independent votes per stimuli, it makes sense to present also a percentage of stimuli with higher than the **Base** model MOS scores, for each one of the five models under test (2-6).

5. Discussion and Conclusions

Overall MOS results have shown that the phone-only prosody prediction (**Phn**) fails to learn convincing speaking style, but rather significantly ($p < 0.01$) deteriorates the quality of the resultant speech. On the other hand, when considering the word semantics (using BERT), certain success in learning conversational style pattern from the noisy data is achieved. We observe that although the absolute MOS improvements for any of the BERT-containing models vs. the baseline in both the conversational style and the overall quality metrics are subtle (not statistically significant), the count of the better-scored stimuli is much higher for BERT (**BERT-TC**) model (68% for the conversational style and 64% for the overall quality metrics, correspondingly). The results revealed also that neither phonetic sequence, not prosodic dialog context contributed to better performance of the BERT-based HPC model (**BERT-TC**), probably due to the challenging spontaneous dataset used for training.

When exploring the best scored model's stimuli (**BERT-TC** vs. **Base**) and their corresponding MOS scores, we came to a conclusion that the perceived improvement came to some extent as a result of improving general expressiveness, but mostly due to better realization of common conversational speech patterns, such as filler words and phrases (e.g., "you know", "like", "well", etc.), that sound more fluent and natural in the proposed system. This observation aligns well with the fact that the conversational HPC models were learnt on a multi-speaker data set containing many speakers, conveying their own interpretations of a conversational speaking style, so just the most general conversational speech features could be acquired, as opposed to the previously reported setup where a large single speaker conversational data set is available [6]. This observation implies that most of the improvements came up at particular textual patterns, thus explaining why the word semantics (text-only input) seemed to be enough to gain those improvements.

Analyzing closely how the system scores change when adding the phonetic stream to the HPC predictor (e.g. **BERT-TC** vs. **BERT-Phn-TC**), we noted that more stimuli got worse perceptual prosody scores, due to some expressiveness deterio-

[1]Audio samples are available at http://ibm.biz/S2S-ConvStyle-SSW21.

System	Phn	BERT-TC	BERT-Phn-TC	BERT-Phn-TPC	BERT-Phn
bert context	-	LT	LT	OSLT	NONE
learning rate	2.892E-05	0.0002	0.0002	0.0002567	0.0002
max epochs	50	40	40	40	40
phone lstm pooling	starting phn	-	mean	mean	mean
sentence loss coef	11.929	13	13	13.002	13
silence loss coef	6.83	1	1	0.92	1
weight decay	0.0001067	0.0001	0.0001	0.0001218	0.0001

Table 1: *Hyper-parameters selected for each model.*

Table 2: *Mean opinion scores with 95% confidence interval (and percentage of stimuli with higher than **Base** MOS score) for the speaking style (Stl.) and overall quality and naturalness (Qual.)*

Cat.	Systems					
	Base	Phn	BERT-TC	BERT-Phn	BERT-Phn-TC	BERT-Phn-TPC
Stl.	3.81± 0.07 (-)	3.74± 0.07 (36%)	3.86± 0.06 (**68%**)	3.85± 0.06 (48%)	3.83± 0.06 (40%)	3.81± 0.07 (56%)
Qual.	3.82± 0.07 (-)	3.74± 0.07 (36%)	3.87± 0.06 (**64%**)	3.86± 0.07 (36%)	3.88± 0.06 (52%)	3.82± 0.07 (48%)

ration, thus obtaining lower count of stimuli with higher-than-baseline MOS scores. However, the terminal sentence prosody is improved, resulting in statistically similar overall MOS scoring.

Additional conversation style pattern, that is common in the spontaneous speech and acquired by the HPC model is *uptalk* [20], i.e. rising intonation on declarative statement end. However, we observed that our S2S TTS system (originally trained with neutral speech that had no uptalk examples) produced less convincing realizations of the uptalk pattern, that were consistently down-voted in the subjective evaluation.

Based on that findings, we are currently exploring gradual neutral and conversation HPC trajectory merging towards sentence ends, to eliminate the negative uptalk effect, while retaining other learnt conversational style effects.

6. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

[2] J. Shen, R. R. Pang, R. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *Proc. ICASSP*, Calgary, Canada, 2018, pp. 4779–4783.

[3] R. Skerry-Ryan, E. Battenberg, X. Y., Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron," *CoRR*, vol. abs/1803.09047, 2018. [Online]. Available: http://arxiv.org/abs/1803.09047

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

[5] P. Wu, Z. Ling, L. Liu, Y. Jiang, H. Wu, and L. Dai, "End-to-end emotional speech synthesis using style tokens and semi-supervised training," in *Proc. APSIPA*, Lanzhou, China, 2019, pp. 623–627.

[6] 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.

[7] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 1. IEEE Computer Society, 1992, pp. 517–520.

[8] S. Shechtman, R. Fernandez, and D. Haws, "Supervised and unsupervised approaches for controlling narrow lexical focus in sequence-to-sequence speech synthesis," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, Shenzhen, China, January 2021, pp. 431–437.

[9] F. A. Gers, J. Schmidhuber, and F. Cummings, "Learning to forget: Continual prediction with LSTM," *Neural Computaiton*, vol. 12, no. 10, pp. 2451–2471, 2000.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.

[11] J. M. Valin and J. Skoglund, "LPCNET: Improving neural speech synthesis through linear prediction," in *ICASSP*, Brighton, England, 2019, pp. 5891–5895.

[12] S. Shechtman, R. Rabinovitz, A. Sorin, Z. Kons, and R. Hoory, "Controllable sequence-to-sequence neural TTS with LPCNET backend for real-time speech synthesis on CPU," *CoRR*, 2020. [Online]. Available: http://arxiv.org/abs/2002.10708

[13] J. Pitrelli, R. Bakis, E. Eide, R. Fernandez, W. Hamza, and M. Picheny, "The IBM Expressive Text-to-Speech Synthesis System for American English." *IEEE Trans. Audio, Speech and Lang. Processing*, vol. 14, no. 4, pp. 1099–1108, July 2006.

[14] S. Shechtman and A. Sorin, "Sequence to Sequence Neural Speech Synthesis with Prosody Modification Capabilities," in *Proc. SSW10*, Vienna, Austria, 2019, pp. 275–280.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, May 2015. [Online]. Available: https://arxiv.org/abs/1412.6980

[16] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[17] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[18] e. a. Falcon, WA, "Pytorch lightning," *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, vol. 3, 2019.

[19] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/

[20] P. Warren, *Uptalk: The phenomenon of rising intonation*. Cambridge University Press, 2016.