INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Naturalness Enhancement with Linguistic Information in End-to-End TTS Using Unsupervised Parallel Encoding

Alex Peiró-Lilja, Mireia Farrús

TALN Research Group, Universitat Pompeu Fabra, Barcelona, Spain
alex.peiro@upf.edu, mireia.farrus@upf.edu

Abstract

State-of-the-art end-to-end speech synthesis models have reached levels of quality close to human capabilities. However, there is still room for improvement in terms of naturalness, related to prosody, which is essential for human-machine interaction. Therefore, part of current research has shift its focus on improving this aspect with many solutions, which mainly involve prosody adaptability or control. In this work, we explored a way to include linguistic features into the sequence-to-sequence Tacotron2 system to improve the naturalness of the generated voice. That is, making the prosody of the synthesis looking more like the real human speaker. Specifically we embedded with an additional encoder part-of-speech tags and punctuation mark locations of the input text to condition Tacotron2 generation. We propose two different architectures for this parallel encoder: one based on a stack of convolutional plus recurrent layers, and another formed by a stack of bidirectional recurrent plus linear layers. To evaluate the similarity between real read-speech and synthesis, we carried out an objective test using signal processing metrics and a perceptual test. The presented results show that we achieved an improvement in naturalness.

Index Terms: naturalness, prosody adaptation, end-to-end, sequence-to-sequence, text-to-speech.

1. Introduction

Parametric speech synthesis has reached human-like attributes, which are essential for many applications such as automatic dialogue or storytelling, through the use of current deep learning architectures and techniques. Most of these models are based on a combination of two architectures: a text-to-feature sequence-to-sequence (seq2seq), and a feature-to-wave model. The text-to-feature seq2seq maps input text to acoustic features [1, 2, 3, 4], while the feature-to-wave model —also known as neural vocoder [5, 6, 7]—, generates audio waveform from predicted features. A concatenation of a feature mapping and a neural vocoder, or also systems that englobe both of them [8], are commonly called end-to-end text-to-speech (E2E-TTS) systems.

Although E2E-TTS systems become more and more lighter and with faster synthesis response, there is still room for improvement in terms of generating natural prosody variability —conveyed through intonation, rhythm and stress—. During decades, natural prosody variation has been studied in parametric speech synthesis, using mainly rule or explicit labeled-based systems [9, 10]. More recent works have performed rule-based modifications to enhance TTS naturalness [11]. Nowadays, current research is focused on finding ways to compact prosody information. In one of the first works on prosody style embedding, the authors compressed acoustic features in the training stage to condition decoder output states [12]. Later, another related work was presented, where the use of global style tokens (GST) allowed the creation of *soft* labels that modify output prosody style [13]. And even more recently, an approach using variational auto-encoders (VAE), permits controlling specific latent prosody attributes found in acoustic features [14]. After that, some other proposed methods are even able to adapt prosody attributes at phoneme level [15]. The main disadvantage in most of these approaches, such as in VAE, is that the user has to select a specific acoustic embedding in order to obtain the desired prosody variability in the output.

Besides, prediction of prosody variability from linguistic features is also gaining interest. Recent works, such as [16], restate the tight relationship between syntax and prosody. Actually, the use of linguistics also helps to solve the mentioned disadvantage of acoustic embedding selection approaches [17]. In this paper, we present another approach to encode linguistic information in parallel with character sequence in an E2E-TTS system. More concretely, we extracted part-of-speech (POS) and, instead of keeping punctuation marks in the input sequence like in the baseline model —that is, period, comma, semicolon, exclamation, question mark, etc.—, we removed them but stored their locations. A similar objective is previously shown in [18], where the authors used syntactic word relationships as part of the input features to improve naturalness. With the same goal, paragraph cues and discourse relations have been explored with the premise that they are also strongly correlated with prosody attributes [19, 20, 21]. Other works, such as in [22], present models with hierarchical architectures to represent word, syllable, phone and frame-level.

The authors of [18] concatenated syntactic features with embedded sequence after passing them through a pre-net linear layer, whereas our approach is more focused on locality using more basic sentence structure features. Specifically, we gave POS labels and punctuation marks location a binary matrix shape, where each column is associated to a character of the input sequence and a row represents a POS or a punctuation mark category. Thus, an activation (value 1) indicates that a character belongs to a POS category and whether it has or not a specific punctuation mark next to it. We also propose two different architectures to process that binary matrix in order to condition the input of the Tacotron2 [4] decoder, which is the sequence of embedded characters. Therefore, we expected the model to take additional effort to focus on the sentence structure to provide more natural speech.

The structure of this paper unfolds as follows. Section 2 describes the extraction and representation of linguistic features. Details about proposed parallel encoder architectures and the baseline E2E-TTS model are detailed in Section 3. Objective and perceptual evaluations as well as discussion of the results can be found in Section 4; and finally, the conclusions and future work are drawn in Section 5.

Copyright © 2020 ISCA

http://dx.doi.org/10.21437/Interspeech.2020-1788

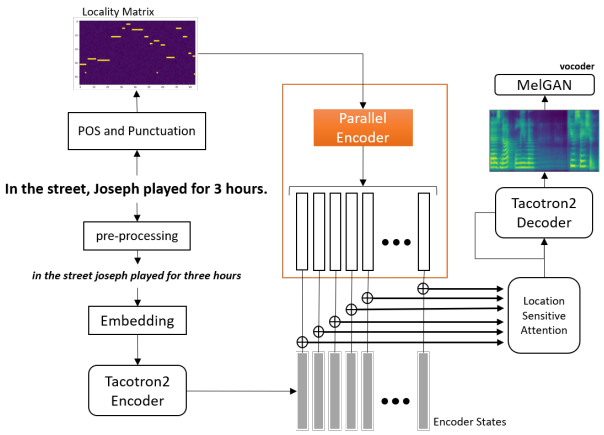

Figure 1: *Complete outline of the model.*

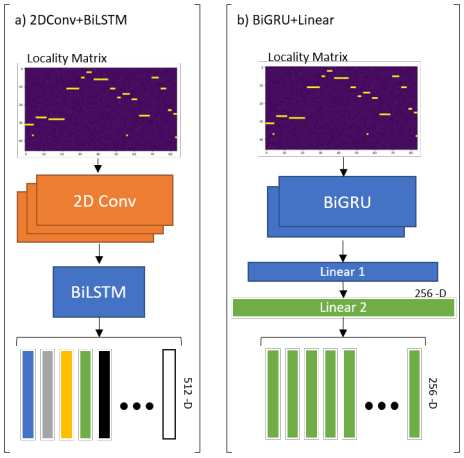

Figure 2: *Parallel encoder architectures explored in this work.*

2. POS and punctuation information

It has been proven since decades that there is correlation between prosody and the structure of the sentence. Prominence, pauses and pitch range are factors that depend on word relationships and punctuation. That is why we implemented an additional pre-processing stage to extract mark delimiters and part-of-speech (POS) tags, which we think they are structure information basics. We worked only with the punctuation mark set found in the database we used for training (details about database and training in the following subsections). English POS tagger was taken from NLTK library. Recent works on deep syntactic parsers have explored POS tags as part of the input features [23]. Thus, we expected that the implemented parallel encoder could find some implicit relationships between words and punctuation.

2.1. Database

We used the public domain LJ Speech Dataset [1] from LibriVox project. It consists of 13,000 short audio clips of a single speaker reading passages from 7 non-fiction books with transcriptions. Clips vary in length from 1 to 10 seconds and have a total length of approximately 24 hours. From the total number of utterances, we used 12,500 to train the model and 100 for the validation set, which was performed every 1,000 training steps. Due to variable-length sequences, we padded with zeroes text and spectrogram sequences.

2.2. Binary location matrix

We propose a binary matrix, which we will call location matrix, as a way to insert POS tags and punctuation marks. The reason is that we can represent locality by setting up as many columns as number of input characters or phonemes, while each row represents a POS tag or a punctuation category. Then, we can active with value 1 the matrix cell associated to the character or characters that belong to a POS tag and/or to any mark delimiter that exists next to it (see Figure 3). The NLTK library[2] was used to extract POS from input sequences with a tagset of 35 different labels. As POS taggs are word level categories, all columns that belong to a same tag (i.e. characters of the same word) are 1. After studying LJSpeech text dataset we detected 11 different punctuation mark categories:

- Endings: '.', '?', '!'
- Separations: ',', ';', ':'
- Word containers: '( )', '{ } '
- Statements: '-'
- Dialogue: ' " '
- Others: ' \'

Marks such as endings and separations, which are placed next to a character, were located in the column associated to that closest character. However, activation of word containers and dialogues were upsampled to all characters placed between the pair of symbols, as we consider a change of prosody when it comes to clarification or a dialogue inside a container.

In total, we collected 46 categories. In order to avoid information redundancy, apart from the conventional text normalization pre-processing, we removed any punctuation category from input sequence before entering the model. Hence, for a sentence like

In the street, Joseph played for 3 hours.

the actual input to the model was:

in the street joseph played for three hours

We presume that encoding of syntax and punctuation in parallel could be beneficial for Tacotron2 baseline model because, then, it could be focused on the generic articulation of phonemes, while our parallel encoder could look more for the specific structure of the sentence and condition the decoding towards a more natural speech.

3. Model

3.1. E2E-TTS

Our baseline E2E-TTS is the concatenation of the seq2seq Tacotron2 [4] together with the neural vocoder MelGAN [7]. The latter was already trained with the same dataset we used in this work. We chose Tacotron2 architecture because overcomes its antecesor in terms of quality and complexity. Moreover, we have also adapted and test it in other languages [24]. Furthermore, MelGAN is one of the latest vocoder models based on generative adversarial networks, much more lighter and faster than previous models such as WaveNet and WaveGlow, while

[1] https://keithito.com/LJ-Speech-Dataset/
[2] https://www.nltk.org/

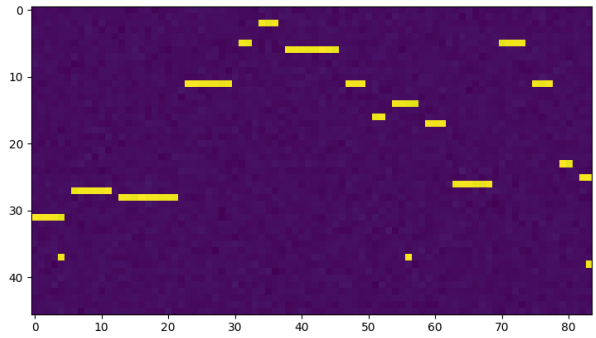

Figure 3: *Location matrix: POS tags and punctuation marks locations, at character-level, represented in a binary matrix.*

reaching a close level of quality. Essentially, we attached to Tacotron2 model a structure that encodes the location matrix in parallel. The output of this parallel encoder was summed to the output hidden states of the Tacotron2 encoder, thus, we could say they were biased (Figure 1). We were inspired by the approach in [13], where the authors tried different ways to introduce the style embedding resulting from global style tokens to the Tacotron model, getting the best results by summing it to the encoder outputs.

3.2. Parallel unsupervised encoder

We proposed two different architectures (Figure 2) to carry out our research (Figure 1). One was inspired by Tacotron2 encoder architecture, which plays the role of the language model of a TTS. Its structure basically consists of a stack of convolutional plus a bidirectional long short-term memory (BiLSTM) layers. Our goal was to see the location matrix as a sparse spectrogram representation, such as a MIDI piano notes representation. Moreover, state-of-the-art signal processing works have presented CNN+RNN architectures for spectrogram processing [25]. Therefore, we implemented a stack of 3 2D convolutionals (2DConv) with kernel size of 3, 7 and 11 by 3 and 8, 16 and 16 channels, respectively. We included batch normalization, dropout of 50% and ReLU activation function on every convolutional layer. Outputs of the BiLSTM were 512D matrix shape, resulting the same dimensionality as Tacotron2 encoder output. Hence, both matrices could be summed before passing to the decoder. In addition, we put some background normal noise to the location matrix.

In the second approach we aimed at performing a more compact model by reducing Tacotron2's character and encoder embedding to 256. The second architecture proposed is formed by two bi-directional gated recurrent unit (BiGRU) plus two fully connected linear layers, returning only a single vector of 256 dimensions. Thus, we upsampled the latter to sum it to all character embeddings, just as in [13]. The stack of BiGRU returns a 92D state, the first linear layer upgrades the vector to 128, and finally, the second linear layer outputs a 256 vector. We used tanh as activation function of linear layers. No background noise was added for this architecture, and we considered location matrix as a concatenation of categorical vectors.

4. Implementation and evaluation

In the following subsections we describe the details of the training phase, evaluation of the system performed through both objective metrics and a perception test, and the discussion of the evaluation results.

4.1. Training

In total, we trained three different versions of Tacotron2 model: Tacotron2 baseline (BS), Tacotron2 with the first parallel encoder version (2DConv+BiLSTM) attached and Tacotron2 with the second parallel encoder version (BiGRU+Linear). All the models were trained using a batch size of 32 and a learning rate of 0.001. The training set had 12500 samples, and validation was processed every 1000 steps with 100 samples. Only in the BiGRU+Linear version the word embedding was reduced to 256 dimensions. The parallel encoder was not pre-trained, but randomly initialized. We did not modified the Tacotron2's objective function, thus parallel encoders were trained unsupervisedly.

4.2. Objective test

We performed an objective test with the aim to quantify the similarity between target and generated speech. To do so, we took four signal processing metrics to compare F0 behavior and spectrogram. These were already used in previous works such as [12], where authors compared the pitch contour from generated samples with a reference one.

4.2.1. Similarity metrics

The similarity metrics used in the objective test are the following: (1) Voicing Decision Error (VDE), (2) Gross Pitch Error (GPE), (3) F0 Frame Error (FFE), and (4) Mel Cepstral Distortion (MCD).

The VDE computes the error when choosing voiced or unvoiced frame as follows:

$$VDE = \frac{N_{VoiceUnvoice} + N_{UnvoiceVoice}}{N} \times 100\% \quad (1)$$

The GPE computes the difference of pitch values in matched voiced frames:

$$GPE = \frac{N_{F0Error}}{N_{VoiceVoice}} \times 100\% \quad (2)$$

where $N_{F0Error}$ is the number of frames for which

$$\left| \frac{F0_{i,estimated}}{F0_{i,reference}} - 1 \right| > \delta\% \quad (3)$$

where i is the frame number, and δ is a threshold which is typically 20.

The FFE is the combination of these two metrics into one, proposed by [26]:

$$FFE = \frac{N_{VU} + N_{UV} + N_{F0Err}}{N} \times 100\% \quad (4)$$

Finally, we also computed the mel cepstral distortion (MCD) of the first 20 coefficients to compare the overall spectrogram. In order to align generated and target pitch curves, both were forced to start with their first non-zero value, and the shorter one was filled with zeros until reaching the same length. We were aware that time differences would penalize synthesis; for this reason, we computed MCD using dynamic time warping

Table 1: *Table of comparisons between Tacotron2 baseline and our two approaches with parallel encoders.*

	VDE (%)	**GPE** (%)	**FFE** (%)	**DTW-MCD** (linear)	**Memory size** (MB)	**Execution time** (char/sec)
Baseline – TS (76k)	3.59	4.81	5.70	$\mathbf{2.82k}$	330.46	**0.06**
Baseline – TS (64k)	3.60	4.88	5.71	$2.96k$		
2Dconv(3)+BiLSTM(1) – TS (70k)	3.62	5.32	5.87	$2.97k$	354.25	0.08
2Dconv(3)+BiLSTM(1) – TS (58k)	3.83	**4.63**	**5.67**	$3.13k$		
BiGRU(2)+Linear(2) – TS (56k)	3.58	4.82	**5.67**	$3.30k$	**257.02**	**0.06**
BiGRU(2)+Linear(2) – TS (50k)	**3.45**	5.33	5.68	$6.66k$		

(DTW). Hence, we had metrics to evaluate the capability of the model to generate natural prosody variability such as the target with time penalization (VDE, GPE, FFE) and without (MCD).

4.2.2. Results

We applied the metrics on the LJSpeech test samples up to 50 characters. Longer ones were skipped due to the high level of distortion that we observed in the synthesis. Consequently, we evaluated 45 sentences in total. Moreover, this evaluation was performed in every training checkpoint from around 45 thousand training steps (TS) to 70 thousand in steps of 2000. We obtained averaged metric values of every TS version evaluated, which allowed us to choose the best version of each model in terms of MCD and also in terms of VDE, GPE and FFE. The results in Table 1 show two versions of each approach. The one that provided, in average, the lowest DTW-MCD, and the version that provided lowest value in one or more pitch tracking metrics.

4.3. Perception test

Besides, we performed a MOS perception test over 14 listeners to evaluate the naturalness of our parallel encoders. That is, how natural the speaker sounds human-like according to the sentence, which could be read by the listener. We did not evaluate emotions or expressiveness, neither did the audio quality or intelligibility. To do so, we collected a total of 16 samples from the LJSpeech Database test set and we divided them into four groups: human speaker, Tacotron2 baseline, Tacotron2 with 2DConv+BiLSTM and Tacotron2 with BiGRU+Linear, obtaining 4 samples per group. For this test, we chose the best of the two versions of each approach shown in Table 1. As we mentioned in the previous section, we evaluated sentences close to 50 characters, so the chosen samples were lower or around that number. The resulting scores can be seen in Table 2.

Table 2: *MOS scores obtained from perceputal test.*

	MOS
Baseline – TS(76k)	3.47
2DConv+BiLSTM –TS(58k)	**3.98**
BiGRU+Linear – TS(56k)	3.34
Human	**4.48**

4.4. Discussion

Objective and perceptual results are consistent with each other in saying that Tacotron2 with 2dConv+BiLSTM parallel encoder performs the best in terms of naturalness. According to the metrics, pitch contours seem closer to be like human reader. However, there is a slight increase of the MCD. On the other hand, although the model with BiGRU+Linear architecture looks like almost equivalent in the objective metrics, perceptual test shows that it performed the worst. This may be due to reduction of model complexity (almost 100MB less than other versions) and its higher level of cepstral distortion. However, we should not discard a lighter architecture for applications with a necessity of fast answer.

5. Conclusions and future work

In this paper, we have explored another way to enhance naturalness of E2E-TTS synthesis with linguistic features. We have proposed an approach in which we divided input into two: a sequence of characters without punctuation marks, and POS tags together with punctuation marks locations from the same input. This information was processed by an encoder in parallel. The output of this encoder was summed to the hidden outputs of the sequence-to-sequence Tacotron2 encoder. Thus, we expected text structure information to condition the synthesis. We proposed two different architectures for this task: one based on a stack of 2D convolutionals and a bidirectional LSTM, and another one consisting of two bidirectional GRUs plus two linear layers. The former returned as many embeddings as the number of sequence characters, and the latter only one, which was upsampled. We compared both versions and, although both parallel encoders objectively seem to perform better than Tacotron2 baseline model, perceptual tests shown that, with 2DConv+BiLSTM parallel encoder, the model performed the best. Therefore, convolutional layers seem to fit better for this problem. However, we will further investigate BiGRU+Linear architecture, and perhaps we can improve its results with a better feature representation. In further experiments we plan to explore syntactic relationships and discourse features, and to use a bigger database to extend our evaluation.

6. Acknowledgements

This work is a part of the INGENIOUS project, funded by the European Union's Horizon 2020 Research and Innovation Programme and the Korean Government under Grant Agreement No 833435. The second author has been funded by the Agencia Estatal de Investigación (AEI), Ministerio de Ciencia, Innovación y Universidades and the Fondo Social Europeo (FSE) under grant RYC-2015-17239 (AEI/FSE, UE). This work has been carried out using an NVIDIA GPU Titan Xp generously provided by NVIDIA Company.

7. References

[1] J. Sotelo, S. Mehri, K. Kumar, J. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," *Iclr*, no. October, pp. 44–51, 2017.

[2] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voice synthesis for in-the-wild speakers via a phonological loop," *CoRR*, vol. abs/1707.06588, 2017.

[3] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, and Q. Le, "Tacotron: Towards end-to-end speech synthesis," in *Proceedings of the INTERSPEECH*, 2017, pp. 4006–4010.

[4] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017.

[5] M. Morise, F. Yokomori, and K. Ozawa, "World: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99.D, pp. 1877–1884, 2016.

[6] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[7] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019, pp. 14 910–14 921.

[8] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," *CoRR*, vol. abs/1612.07837, 2016.

[9] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. W. Wightman, P. Price, J. B. Pierrehumbert, and J. Hirschberg, "ToBI: A standard for labeling English prosody," in *Proceedings of the ICSLP*, 1992.

[10] P. Taylor, "Analysis and synthesis of intonation using the Tilt model," *Journal of the Acoustical Society of America*, vol. 107, pp. 1697–1714, 1998.

[11] A. Peiró-Lilja and M. Farrús, "Paragraph prosodic patterns to enhance text-to-speech naturalness," in *Proceedings of the 9th International Conference on Speech Prosody*, Poznań, Poland, 2018, pp. 512–516.

[12] R. J. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *CoRR*, vol. abs/1803.09047, 2018.

[13] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *CoRR*, vol. abs/1803.09017, 2018.

[14] W. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, "Hierarchical generative modeling for controllable speech synthesis," *CoRR*, vol. abs/1810.07217, 2018.

[15] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," *CoRR*, vol. abs/1811.02122, 2018. [Online]. Available: http://arxiv.org/abs/1811.02122

[16] A. Köhn, T. Baumann, and O. Dörfler, "An empirical analysis of the correlation of syntax and prosody," *arXiv preprint arXiv:1806.05900*, 06 2018.

[17] S. Tyagi, M. Nicolis, J. Rohnke, T. Drugman, and J. Lorenzo-Trueba, "Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection," *arXiv preprint arXiv:1912.00955*, 2019.

[18] H. Guo, F. K. Soong, L. He, and L. Xie, "Exploiting syntactic features in a parsed tree to improve end-to-end TTS," *CoRR*, vol. abs/1904.04764, 2019. [Online]. Available: http://arxiv.org/abs/1904.04764

[19] M. Farrús, C. Lai, and J. D. Moore, "Paragraph-based prosodic cues for speech synthesis applications," in *Proceedings of the 8th Speech Prosody Conference*, Boston, MA, 2016.

[20] J. Kleinhans, M. Farrús, A. Gravano, J. M. Pérez, C. Lai, and L. Wanner, "Using prosody to classify discourse relations," in *Proceedings of the INTERSPEECH*, Stockholm, Sweden, 2017, pp. 778–781.

[21] A. Aubin, A. Cervone, and O. Watts, "Improving speech synthesis with discourse relations," in *Proceedings of the INTERSPEECH*, Graz, Austria, 2019, pp. 4470–4474.

[22] V. Wan, C. Chan, T. Kenter, J. Vit, and R. Clark, "CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," *CoRR*, vol. abs/1905.07195, 2019. [Online]. Available: http://arxiv.org/abs/1905.07195

[23] J. Legrand and R. Collobert, "Deep neural networks for syntactic parsing of morphologically rich languages," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 573–578.

[24] B. Külebi, A. Öktem, A. Peiró-Lilja, S. Pascual, and M. Farrús, "Catotron – A neural text-to-speech system in Catalan," in *Proceedings of the INTERSPEECH*, Shanghai, China, 2020.

[25] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, and T. N. Sainath, "Deep learning for audio signal processing," *CoRR*, vol. abs/1905.00078, 2019.

[26] Wei Chu and A. Alwan, "Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 3969–3972.