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

Comparing acoustic and textual representations of previous linguistic context for improving Text-to-Speech

Pilar Oplustil-Gallegos, Johannah O'Mahony, Simon King

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
p.s.oplustil-gallegos@sms.ed.ac.uk

Abstract

Text alone does not contain sufficient information to predict the spoken form. Using additional information, such as the linguistic context, should improve Text-to-Speech naturalness in general, and prosody in particular. Most recent research on using context is limited to using textual features of adjacent utterances, extracted with large pre-trained language models such as BERT.

In this paper, we compare multiple representations of linguistic context by conditioning a Text-to-Speech model on features of the preceding utterance. We experiment with three design choices: (1) acoustic vs. textual representations; (2) features extracted with large pre-trained models vs. features learnt jointly during training; and (3) representing context at the utterance level vs. word level.

Our results show that appropriate representations of either text or acoustic context alone yield significantly better naturalness than a baseline that does not use context. Combining an utterance-level acoustic representation with a word-level textual representation gave the best results overall.

Index Terms: Text-to-Speech, speech synthesis, context, prosody

1. Introduction and Related Work

Although text alone is not sufficient to predict prosody accurately, Text-to-Speech (TTS) systems are generally trained to generate spoken utterances given textual input only, and utterances are assumed to be independent from one another. While this might be true for certain types of text, utterances in monologues, conversation, audio-books or from any other long-form discourse are not isolated, but influenced by context [1, 2, 3]. Utterances are organized into a discourse structure in which neighbouring utterances are part of the linguistic context [4]. Context can have a global effect on the average and range of F_0 and speech rate, or a localized one such as the absence or presence of prominence.

In this paper we study how linguistic context, specifically the previous utterance, can be exploited to improve TTS. Our proposed method conditions the generation of an utterance on the acoustic and/or textual properties of the immediately preceding one. We experiment with different design choices to answer the general research question: how should linguistic context be represented?

Augmenting TTS model inputs with linguistic context information has been proposed by several authors, including the use of position of sentence inside a larger unit such as a paragraph [1, 5], explicit discourse features such as discourse relations [6] or topic structure [7]. While discourse features can improve synthetic speech, feature extraction relies on models that require supervised training on appropriately-labelled data.

Other approaches include directly labelling emphasis [8, 9, 10] or phrase breaks [11, 12]. Direct labelling can be useful for controllability, but accurately predicting labels only from text is hard.

In order to avoid the need for labelled data, unsupervised approaches can be used to learn contextual representations from acoustic features using encoders, which are later driven by traditional textual features [13, 14, 15]. However, these models still generally use within-sentence textual input features, which are insufficient to accurately predict prosody. [16] takes a different approach by using linguistic features and acoustic distance from the previous utterance to sample from a variational autoencoder of prosody, which synthesizes the current sentence. However, their method is applied at inference time only.

Another approach, closer to what we propose here, uses textual context to enhance a TTS baseline, conditioning mel spectrogram prediction directly on a representation of context [17, 18] in which BERT-derived features represent neighbouring (both preceding and following) sentences. Although neither method uses explicit prosodic features or learns prosodic representations, it was observed that the use of context significantly improves the prosody of the synthesized speech.

How the different features of context are captured is an important design choice. While [17] and [18] only capture textual features, in previous work [19] we saw that acoustic features can also lead to significant improvement. That approach makes use of a prosody transfer module, Global Style Tokens [20], to extract a prosodic representation from the mel spectrogram of the context. That representation is then used to condition the model, in a similar fashion to [17] and [18].

Our previous work was limited to represent acoustic features of the context at the utterance level using mel spectrograms. Here, we substantially expand the scope of our work to consider additional design choices, and to compare against methods proposed by others.

Therefore, the current goal is to experiment with three design choices regarding how to represent context: (1) textual vs. acoustic features; (2) representations extracted with large pre-trained models vs. representations learnt jointly with the TTS training; and (3) context at the utterance-level or at the word-level.

We will show that: either textual or acoustic representations of context can significantly improve speech naturalness, and a combination of both yields the best results; representations extracted with large pre-trained models outperform representations extracted using jointly-trained model components; and, word-level representations seem to be better matched to textual features, while an utterance-level representation is better for acoustic features.

10.21437/SSW.2021-36

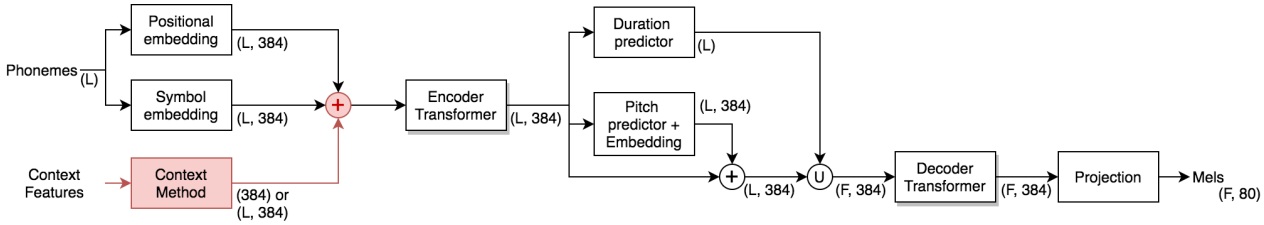


Figure 1: *System diagram. The baseline architecture is FastPitch [21] which we augment with a Context Method (in red) whose output is summed to the embeddings at the encoder input. The Context Method is one of the 4 possible models shown in Figures 2 and 3. L is the length of the current sentence (in phones). F the duration of the output (in frames). U denotes upsampling from phones to frames.*

2. Experimental design

2.1. Baseline

Our baseline model is FastPitch [21], which comprises Transformer-based encoder and decoder, with explicit duration and F_0 predictors. Input symbols (phonemes in the current work) and their positional encoding are embedded, summed, and input to the encoder. F_0 is embedded and summed to the encoder output before going into the decoder during training. The duration of each input symbol determines the upsampling between encoder output and decoder input. F_0 is modelled per-input symbol, with per-speaker mean/variance normalisation. During training, ground-truth values of F_0 and duration are used, whilst a predictor is trained for each of them. For inference, predicted values are used.

FastPitch was selected because it is fast and stable in both training and inference, and has an open source implementation from the original author [22]. All models in the current work were trained from scratch for ∼77k iterations. To vocode the generated mel spectrograms to waveforms, we used the WaveGlow [23] checkpoint included with the FastPitch implementation, which has been trained on the LJSpeech corpus [24].

To condition FastPitch on previous sentence context, we add a module that provides a representation that is summed to the encoder inputs, labelled as *Context Method* in Figure 1. This location was selected as the best place to inject context into the model in prototyping experiments.

2.2. Context features and representations

We compare acoustic vs. textual features, each of which can be input to either a large pre-trained model, or a model jointly trained with the TTS model, to create a Context Representation.

Acoustic: we use the same mel spectrograms extracted for training. As in FastPitch [21], these are 80-band mel spectrograms extracted with a window length of 1024 samples 256 hop size. For the **jointly-learnt** condition, a Context Representation is learnt from the mel spectrograms as described in Section 2.3.

For the **pre-trained** condition, the mel spectrogram is used to obtain a Context Representation from a large pre-trained model. We use the Deep Spectrum [25, 26], which was found in our previous work to be capable of encoding global acoustic characteristics [27]. It extracts a fixed-dimension vector by treating the mel spectrogram as an image and inputting it to a large-scale image classification model. We use the implementation from the original authors [28], using layer *fc2* of the VGG-19 model to obtain a 4096-dim vector. One vector can be obtained for the whole utterance, or for each of a sequence of fixed windows (which, in our experiments, will depend on the word-level or utterance-level condition, see Section 2.3).

Text: we use two types of features derived from the text: phonetic transcriptions for the jointly-trained condition and word tokens for the pre-trained one. To **jointly-learn** a Context Representation, we use the phonetic transcription of the previous sentence. Phonetic transcriptions are obtained as for all the training data for the models (Section 2.4), and use 47 symbols including phones and punctuation. Word or syllable boundaries are not included in the transcription.

To obtain a context representation from a **pre-trained** model, we use BERT, and therefore, the context features used as input correspond to text words (or tokens). BERT embeddings are extracted using an off-the-shelf model in the transformers Python library [29]. 768-dim vectors at the utterance-level are obtained by averaging the activations of second to last hidden layer, or at the word-level by summing the activations of the last four layers of the model [30].

We decided to use a phonetic transcription for the jointly-learnt condition rather than textual words or tokens as it seemed unlike that the Context Method would be able to learn a relationship over sparse combinations of words for our training data (which is why large models as BERT are required to encode such relationships).

2.3. Context methods

The third design choice we are interested in is whether to represent context at utterance- or word-level. We anticipate that the model will learn global prosodic effects from utterance-level representations, and local effects from word-level representations. The utterance-level representations are a fixed-length vector that is constant for every encoder step. In contrast, the word-level method outputs a representation that potentially varies for every encoder step.

Whilst it is desirable to maintain the most similar model architecture for all combinations of design choices, the differences in resolution and nature of the representations do entail some differences, illustrated in Figures 2 and 3. In both figures, Context Features are always extracted from the previous sentence. The resulting Processed Context Representation is the one finally added to the encoder inputs in Figure 1, conditioning the current sentence.

2.3.1. Using an utterance-level representation

Utterance-level Context Methods make use of Global Style Tokens [20] which, as we have already shown [19], can be used to represent context at the utterance-level and have been used in TTS for diverse tasks [31, 32, 33]. GSTs are a set of randomly initialized tokens (vectors). Multi-head attention is used to learn the relevance of each token for every training utterance. Since the tokens are constant, they can be thought of as labels,

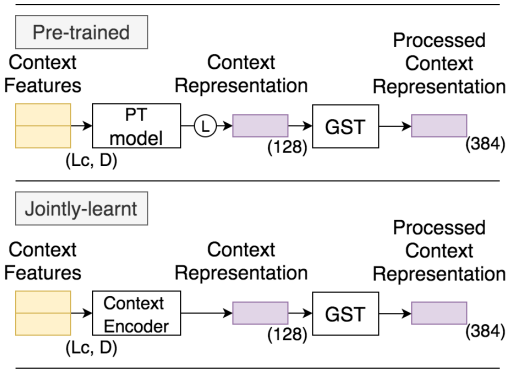


Figure 2: *Context Methods for extracting an utterance-level representation of context. As described in Section 2.2, for the pre-trained condition, Context Features correspond either to mel spectrograms (Lc = frames, D = 80) input to a Deep Spectrum pre-trained model, or to text word tokens (Lc = word tokens, D = embedding dim) input to BERT. Because pre-trained models output Context Representations with a different dimension, a linear layer (circle-L) is used to reduce dimensionality. For the jointly-learnt condition, mel spectrogram (Lc = frames, D = 80) or phonetic transcription (Lc = phones, D = embedding dim) are the Context Features. While pre-trained models (PT model) extract a single vector Context Representation already, for the jointly-learnt condition a single vector Context Representation is obtained through a Context Encoder. Finally, for both conditions, a Processed Context Representation is obtained by applying GST.*

with attention 'labelling' the data in unsupervised fashion.

GST takes as input a fixed-dimension vector. The representations obtained from pre-trained models (Deep Spectrum or BERT) can be obtained at the utterance-level, and therefore are simply reduced in dimensionality before GST. In contrast, the representations obtained from jointly-trained models must be summarised into a single vector. We use a Context Encoder (lower part of Figure 2) with the same architecture as the reference encoder in [20]. We train GST with 10 tokens and 8 heads to output a 384-dim vector. We use the implementation provided by [34].

2.3.2. *Using a word-level representation*

Figure 3 explains how the word-level Context Methods create a Context Representation from the previous sentence, for each word in the Current Sentence, which has the potential to encode local prosodic phenomena.

Pre-trained models output Context Representations at the word-level (or pseudo-word-level for Deep Spectrum) already. For the jointly-learnt condition, the Context Features are first processed by a block of convolutional layers with the same architecture as the transformer (1D conv > Relu > 1D conv > summed to the residual > layer norm). Then, word-level resolution is obtained by averaging frames or phones within word boundaries.

Once the Context Representation is obtained, attention is used to gather elements of it and potentially re-order them in a way that is relevant for the Current Sentence. Finally, the new Processed Context Representation is simply added to the encoder inputs without further processing.

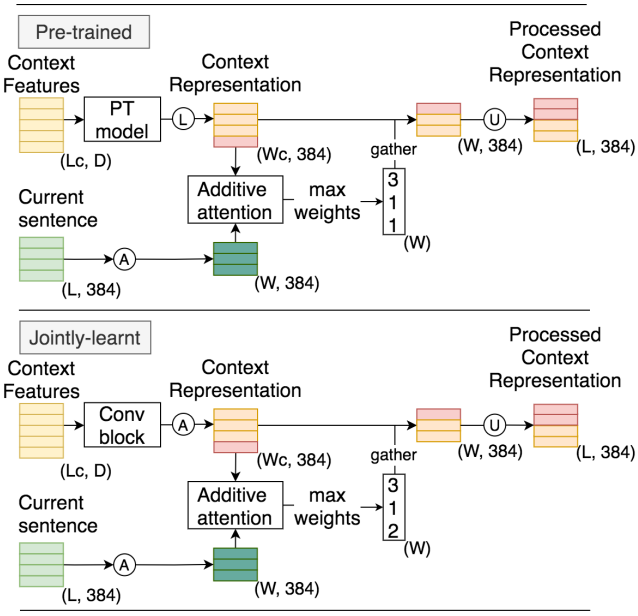
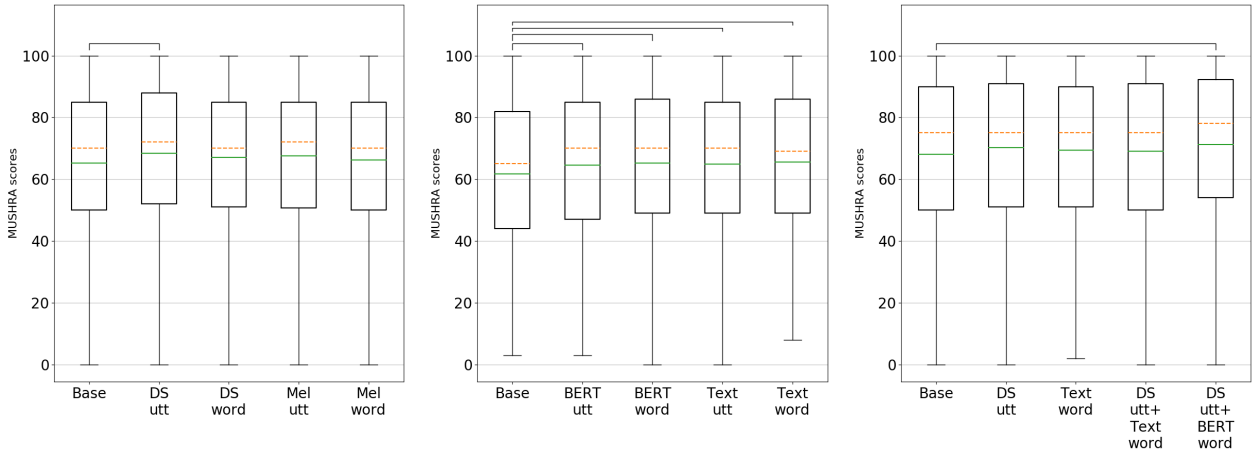


Figure 3: *Context Methods for extracting a word-level representation of context. Context Features correspond to those described in Figure 3. Context Representations are now obtained to match word-like resolution (Wc). For the pre-trained condition, BERT embeddings are obtained for every word token, while for Deep Spectrum, mel spectrograms are divided into one second segments (without overlap). As before, these are reduced in dimensionality by a linear layer (circle-L) to obtain the Context Representation. For the jointly-learnt condition, a block of convolutions is first applied to learn a Context Representation, however this module does not affect the resolution of the features (Lc = frames, for mel spectrograms, Lc = phones, for phonetic transcription). To obtain a word-level representation (Wc), we average (circle-A) using word boundaries. In parallel, word-level representations for the Current Sentence phones are obtained averaging. Next, the attention mechanism calculates how relevant each word in the Context Representation is to each word in the Current Sentence. The maximum attention weight for each word in the Current Sentence is used to identify the most relevant word in the Context Representation; the Context Representation of that word is gathered into a sequence of length W. The resulting Processed Context Representation is upsampled (circle-U) to match the length required to sum it to the encoder inputs.*

2.4. Data and pre-processing

All models used phonetized inputs obtained while force-aligning the data with the Montreal Forced Aligner [35] to extract the ground-truth durations required to train the duration predictor and upsample phones to frames, and to obtain the word boundary information needed for word-level representations. Out-of-vocabulary words were transcribed using G2P [36] and punctuation was restored. We obtained F_0 contours using Praat for Python [37] as in FastPitch [21].

We trained and tested all models using LJSpeech [24], with 12443 training sentences and 525 test sentences. We follow the data naming structure to obtain previous-current sentence pairs.



(a) (b) (c)

Figure 4: *Listening test results (a) acoustic context alone; (b) text context alone; (c) best models compared with acoustic+text combinations. Horizontal bars connect pairs of systems that are significantly different.*

Model name	**Context Feature**	**Context Representation**	**Context Method**
DS-utt	Acoustic	Deep Spectrum	Utterance
DS-word	Acoustic	Deep Spectrum	Word
mel-utt	Acoustic	Learnt from mels	Utterance
mel-word	Acoustic	Learnt from mels	Word
BERT-utt	Text	BERT	Utterance
BERT-word	Text	BERT	Word
Text-utt	Text	Learnt from phones	Utterance
Text-word	Text	Learnt from phones	Word

Table 1: *Summary of models compared in experiments.*

3. Evaluation and Results

Testing all combinations of our three design choices resulted in the 8 models summarised in Table 1. We predicted that text vs acoustic features, and utterance-level vs word-level representation, would be complementary, so we also tested some combinations. To make evaluation feasible, the listening test was conducted in three parts: (1) compare the 4 acoustic feature systems; (2) compare the 4 text feature systems; (3) compare the best acoustic system, best text system, and two systems that combine both.

We did not know whether acoustic or text context would be most informative. However, we did hypothesise that acoustic context would be best represented at the utterance level, and that text context would be best represented at the word level.

Each of the three listening tests used a MUSHRA-like design[1]. and compared 4 models, plus the baseline and the hidden reference (vocoded natural speech). The same 25 sentences were used for all listening tests. Each MUSHRA screen presented the reference audio, then the 6 samples to be rated, without text. Participants were instructed to rate the naturalness of the synthetic speech. For the acoustic systems, features were extracted from a natural rendering of the context utterance: Section 4 comments on the possible effects of using synthetic speech instead.

We implemented the test online using Qualtrics and recruited participants who self-identified as native speakers of English and US citizens, using Prolific Academic. Results from participants who rated any reference sample lower than 50, or were too fast to complete the task, were discarded. For each test, the first 20 participants who passed these checks were used to calculate the results. Each test used different participants.

Statistical significance was determined using the Wilcoxon signed-rank test with Bonferroni correction. Figure 4 shows the results for the three tests.

3.1. First listening test: acoustic context

Results for acoustic context are in Figure 4(a) for the systems listed in the upper 4 rows of Table 1. Only Deep Spectrum features at the utterance level were significantly better than baseline. Although not significant, all other acoustic contexts resulted in slightly higher scores than baseline, with utterance-level representation tending to be better than word-based.

3.2. Second listening test: text context

Results for the text context are in Figure 4(b) for the systems listed in the lower 4 rows of Table 1. All models using text context were significantly more natural than baseline. Although not significantly different between each other, word-level representation tended to lead to slightly higher naturalness than utterance-level.

3.3. Third listening test: best models and combinations

We compared the most effective way to use acoustic context (DS-utt), the most effective way to use text context (Text-word, which had the most significant difference to the baseline), and two combinations of acoustic and text context.

[1]Samples:
`https://pilarog.github.io/ssw2021/index.html`

We trained the combinations: DS-utt + Text-word and DS-utt + BERT-word. Deep Spectrum was clearly the most effective acoustic feature. Since there was no significant difference between the models using text context, we included both Text-word and BERT-word. Results are shown in Figure 4(c). The system using Deep Spectrum features to derive an utterance-level representation of acoustic context, with BERT features to derive a word-level representation of text context, was significantly better than baseline.

It is not surprising that neither DS-utt or Text-word were significantly better than baseline here, even though they were in the preceding listening tests. MUSHRA ratings are relative, with an element of ranking, so a different set of systems under comparison (especially a change in the least natural system; there is no anchor in our tests) will lead to a different rating space.

3.4. Listening test results analysis

Our results illustrate the benefit of using both acoustic and text features of the context utterance, individually or in combination. In every listening test, the baseline was outperformed by at least one model employing context. DS-utt + BERT-word was the best combined system, which supports our hypothesis that acoustic features are most useful when represented at the utterance level, with text features at the word level. Pre-trained models generally outperformed jointly-trained ones.

Informally, we observed that the use of context affected the speech in different ways: in prosody, pauses, and pronunciation, with the most apparent changes being prosodic in nature. Although we did not ask participants to directly judge prosody, it seems likely that they are implicitly doing so, given some of their comments. At the end of each listening test, we included an optional comment box. Several participants mentioned how it was "interesting" or "challenging" to distinguish the different "inflections" in the samples.

3.5. Qualitative analysis

Our results indicate that context is informative. To confirm this and to further analyse its effect, we examined differences in the output when synthesizing the same sentence with different contexts. This differs from what was evaluated in the listening tests of the previous section. Here, we confirm that changes in context produce changes in the output.

Although pronunciation can also be affected by the context, most of the variation we observed was prosodic. Figure 5 provides some example F_0 contours. In (a), using acoustic context represented at utterance level, the overall F_0 pattern tends to stay the same, and changing context has a global effect, shifting F_0 or affecting speech rate. In contrast, (b) shows that representing text context at the word level can modify the position and strength of prominence. Finally, combining acoustic and text context in (c) illustrates both effects.

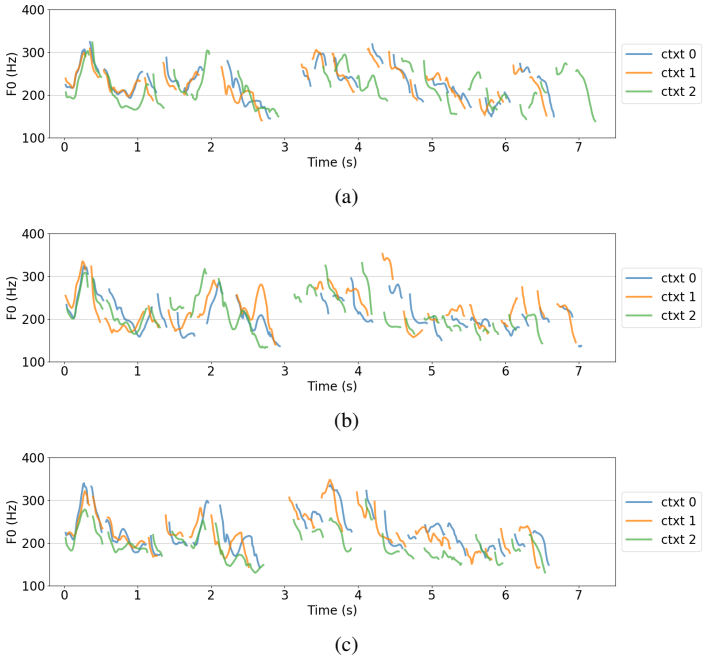


Figure 5: *Illustration of the effect of context for a single sentence synthesized with (a) DS-utt; (b) BERT-word; (c) a combination of both. In each plot, the three F_0 contours are the result of using three different context utterances (the same three across all plots).*

4. Conclusion and future work

Our results provide further evidence that additional context can improve TTS naturalness, and that the way in which context is represented matters. Even if context is not used explicitly to improve prosody, this seems to be the aspect that is affected the most.

We have shown that both acoustic and text context, when suitably represented, can significantly improve naturalness, and that the best results are obtained by combining them. In a real use-case (e.g., long-form synthesis), acoustic context would need to be extracted from the previous *synthesized* utterance. Although we did not test this condition here, we provide samples on the companion web page for DS-utt using features extract from synthesized speech context: degradation appears to be minimal. Text features have the notable advantage of being available for future context, although this was not tried here.

Our results indicate that features extracted using large pre-trained models are more effective than using jointly-trained models, especially for acoustic features. It could be that the acoustic relationships between context and current sentence is very sparse. In contrast, using text features with a jointly-trained model was comparable (in the second listening test) to BERT. Very recent work proposes using BERT on phonetic transcriptions [38], which would be worth trying.

To obtain the best results from a jointly-trained model for extracting a Context Representation, it might be necessary to incorporate an extra loss, as in our preliminary work [19]. We did not include this condition here as the focus was on how best to represent context rather than on the model itself.

There is also evidence that acoustic features give best results when represented at utterance level, and text features when represented at word level. The qualitative analysis in Section 5 suggests that these are associated with producing global and local prosodic effects respectively, without having to model these in an explicit way or through very specific features.

In future work, choice of data and speaker is important [27]. We aim to use more expressive or spontaneous data to better evaluate the effect of using context.

The listening test in this paper was restricted to measuring the naturalness of isolated sentences, which were not presented in context. This was a deliberate choice, but in-context eval-

uation will be a fundamental part of future work. Pioneering work [39] has tested such an evaluation paradigm, but we believe that it still needs to be further developed before we can apply it to our systems, and therefore we are also working on suitable evaluation methods for speech in context [40].

Acknowledgements: This work was supported in part by: ANID, Becas Chile, nº 72190135. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement Nº 859588.

5. References

[1] M. Farrús, C. Lai, and J. D. Moore, "Paragraph-based prosodic cues for speech synthesis applications," *Speech Prosody*, 2016.

[2] G. M. Ayers, "Discourse functions of pitch range in spontaneous and read speech," *Papers in Linguistics*, 1994.

[3] J. Hirschberg, "Communication and prosody: Functional aspects of prosody," *Speech Communication*, vol. 36, no. 1-2, pp. 31–43, 2002.

[4] H. Bunt, "Dialogue pragmatics and context specification," *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics.*, pp. 81–150, 2000.

[5] À. Peiró Lilja and M. Farrús, "Paragraph prosodic patterns to enhance text-to-speech naturalness," in *Speech Prosody*, 2018.

[6] A. Aubin, A. Cervone, O. Watts, and S. King, "Improving speech synthesis with discourse relations." in *Interspeech*, 2019.

[7] J. Hirschberg, "Accent and discourse context: Assigning pitch accent in synthetic speech." in *AAAI*, vol. 90, 1990, pp. 952–957.

[8] Z. Malisz, H. Berthelsen, J. Beskow, and J. Gustafson, "Controlling prominence realisation in parametric dnn-based speech synthesis." in *Interspeech*, 2017.

[9] A. Suni, S. Kakouros, M. Vainio, and J. Šimko, "Prosodic prominence and boundaries in sequence-to-sequence speech synthesis," *arXiv preprint arXiv:2006.15967*, 2020.

[10] S. Shechtman, R. Fernandez, and D. Haws, "Supervised and unsupervised approaches for controlling narrow lexical focus in sequence-to-sequence speech synthesis," in *SLT*, 2021.

[11] A. Rendel, R. Fernandez, Z. Kons, A. Rosenberg, R. Hoory, and B. Ramabhadran, "Weakly-supervised phrase assignment from text in a speech-synthesis system using noisy labels." in *Interspeech*, 2017.

[12] V. Klimkov, A. Nadolski, A. Moinet, B. Putrycz, R. Barra-Chicote, T. Merritt, and T. Drugman, "Phrase break prediction for long-form reading tts: Exploiting text structure information." in *Interspeech*, 2017.

[13] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," in *SLT*, 2018.

[14] S. Karlapati, A. Abbas, Z. Hodari, A. Moinet, A. Joly, P. Karanasou, and T. Drugman, "Prosodic representation learning and contextual sampling for neural text-to-speech," *ICASSP*, 2021.

[15] Z. Hodari, A. Moinet, S. Karlapati, J. Lorenzo-Trueba, T. Merritt, A. Joly, A. Abbas, P. Karanasou, and T. Drugman, "Camp: a two-stage approach to modelling prosody in context," *ICASSP*, 2021.

[16] S. Tyagi, M. Nicolis, J. Rohnke, T. Drugman, and J. Lorenzo-Trueba, "Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection," *Interspeech*, 2020.

[17] H. Guo, S. Zhang, F. K. Soong, L. He, and L. Xie, "Conversational end-to-end tts for voice agents," in *SLT*, 2021.

[18] G. Xu, W. Song, Z. Zhang, C. Zhang, X. He, and B. Zhou, "Improving prosody modelling with cross-utterance bert embeddings for end-to-end speech synthesis," *arXiv preprint arXiv:2011.05161*, 2020.

[19] P. Oplustil-Gallegos and S. King, "Using previous acoustic context to improve text-to-speech synthesis," *arXiv preprint arXiv:2012.03763*, 2020.

[20] 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*, 2018.

[21] A. Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," *ICASSP*, 2021.

[22] NVIDIA, "Fastpitch 1.0 for pytorch," https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/SpeechSynthesis/FastPitch, 2021.

[23] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP*, 2019.

[24] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[25] N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl, and B. W. Schuller, "An Image-based Deep Spectrum Feature Representation for the Recognition of Emotional Speech," in *ACM*, 2017.

[26] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, A. Baird, and B. Schuller, "Snore Sound Classification Using Image-Based Deep Spectrum Features," in *Interspeech*, 2017.

[27] P. Oplustil-Gallegos, J. Williams, J. Rownicka, and S. King, "An unsupervised method to select a speaker subset from large multi-speaker speech synthesis datasets," *Interspeech*, 2020.

[28] S. Amiriparian, M. Gerczuk, S. Ottl, and B. Schuller, "Deep spectrum repository," https://github.com/DeepSpectrum/DeepSpectrum, 2021.

[29] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[30] C. McCormick and N. Ryan, "Bert word embeddings tutorial." http://www.mccormickml.com, 2021.

[31] 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 *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2019.

[32] H. Li and J. Yamagishi, "Noise tokens: Learning neural noise templates for environment-aware speech enhancement," *Interspeech*, 2020.

[33] S. Kato, Y. Yasuda, X. Wang, E. Cooper, S. Takaki, and J. Yamagishi, "Rakugo speech synthesis using segment-to-segment neural transduction and style tokens—toward speech synthesis for entertaining audiences," in *SSW*, 2019.

[34] NVIDIA, "Mellotron repository," https://github.com/NVIDIA/mellotron, 2021.

[35] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, 2017.

[36] K. Park and J. Kim, "g2pe," https://github.com/Kyubyong/g2p, 2019.

[37] Y. Jadoul, B. Thompson, and B. de Boer, "Introducing Parselmouth: A Python interface to Praat," *Journal of Phonetics*, vol. 71, pp. 1–15, 2018.

[38] Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, "Png bert: Augmented bert on phonemes and graphemes for neural tts," *Interspeech*, 2021.

[39] R. Clark, H. Silen, T. Kenter, and R. Leith, "Evaluating long-form text-to-speech: Comparing the ratings of sentences and paragraphs," *SSW*, 2019.

[40] J. O'Mahony, P. Oplustil-Gallegos, C. Lai, and S. King, "Factors affecting the evaluation of synthetic speech in context," in *SSW (submitted)*, 2021.