The IMS Toucan system for the Blizzard Challenge 2021

Florian Lux, Julia Koch, Antje Schweitzer, Ngoc Thang Vu

Institute for Natural Language Processing, University of Stuttgart

[florian.lux, julia.koch, antje.schweitzer, thang.vu]@ims.uni-stuttgart.de

Abstract

For our contribution to the Blizzard Challenge 2021, we built a non-autoregressive speech synthesis system that transforms phoneme inputs from a phonemiser into spectrogram frames. A GAN based vocoder then transforms the spectrogram into a waveform. We handle code-switching by altering the phonemiser based on the output of a language identification system. Non-native phonemes are manually mapped to their closest native representation. An interactive demo is available[1].

Index Terms: Non-autoregressive synthesis, GAN vocoder, code-switching aware

1. Introduction

The Blizzard Challenge is a yearly occurring shared task in the text-to-speech synthesis (TTS) community, with extensive evaluation by human listeners. This year's challenge consists of a "hub" task and a "spoke" task. For both tasks, participants are required to build a voice for synthesising European Spanish texts. In the hub task, these texts contain exclusively Spanish words, whereas in the spoke task, the texts contain a small number of English words, i.e. there is *code-switching* in the Spanish texts. Participants are provided with approx. 5 hours of high-quality audio training data read by one female European Spanish speaker. These data do not contain code-switching; there are however some exceptions that we discuss below. In addition, ten utterances that contain more code-switching are provided as development data. Usage of additional audio data is explicitly allowed (for either task, but subject to the limitation that all in all no more than 100 hours of data are used for building the voice).

Our contribution to the challenge consists in a neural network (NN) system. NN systems have started to outperform concatenative systems in terms of intelligibility and even in terms of naturalness [1, 2, 3, 4, 5, 6]. One of the few disadvantages left are their computational cost, their tendency to sometimes skip or repeat parts of input words, and the lack of prosody control. DeepVoice [1] proposes modifications to speed up inference, however it only achieves close-to-human mean opinion scores (MOS) with a model that still runs slower than real time, and uses ground-truth F0 values and durations instead of predicted ones. Similarly TransformerTTS [3], FastSpeech [5] and FastSpeech 2 [6] propose new architectures to address this issue; however FastSpeech for instance does not predict optimal durations, and still sometimes skips or repeats parts of the utterance. Thus these issues are addressed again in the FastSpeech 2 paper [6]. Our ultimate research goal, irrespective of the challenge, was to create an easy-to-use toolkit for speech synthesis which can be used even without access to significant computational resources. Our contribution to the challenge as well as the competing candidate systems in our preliminary experiments are instances of models trained with said toolkit, IMS Toucan.

[1] https://github.com/DigitalPhonetics/IMS-Toucan#demonstration

We will describe the two architectures used in our candidate systems in more detail in section 2 below. We will then discuss aspects related to code-switching required for the spoke task in section 3, before we introduce the IMS Toucan Speech Synthesis Toolkit in section 4. Next we will give more details about the systems employed for the two challenge tasks in section 5. Finally we present the evaluation results from the challenge in section 6 before concluding in section 7.

2. Related work

As stated in the introduction, issues in NN synthesis are speed at inference time and efficiency at training time. In this vein, TransformerTTS [3] aims to improve inference speed by parallelising more calculations. It does so by using most of the Tacotron 2 [2] architecture, but replacing the recurrent NN components with the Transformer architecture, introduced in [4]. Like most other NN models, it generates spectrogram frames from phoneme embeddings. In contrast to FastSpeech (see below), it can train on text-speech pairs without explicit alignment information. It learns which part of the input belongs to which part of the output by using an attention mechanism, as introduced in [4]. The encoded space that the attention is applied to is subdivided into multiple attention heads, each of which calculates one routing of information from inputs to outputs. Using multiple heads improves the parallelisation capability of the system [3]. The self-attention allows modeling longer dependencies and is claimed to improve the adequacy of the overall sentence prosody. Having generated spectrogram frames in this way, TransformerTTS then employs WaveNet [7], an autoregressive NN vocoder, to generate the actual audio signal. WaveNet is well known for its superior audio quality, however due to its autoregressive nature it is relatively slow.

The issue of inference speed is also addressed by FastSpeech [5] and FastSpeech 2 [6]. Both FastSpeech systems convert their input to phoneme embeddings and produce mel spectrogram frames as output, which are then passed to another network (WaveGlow [8] for FastSpeech, Parallel WaveGAN [9] for FastSpeech 2) to generate actual audio signals. (FastSpeech 2 can also directly generate audio signals without generating mel spectrograms.) Both synthesise spectrograms from text in a non-autoregressive manner, i.e. without conditioning the prediction of spectrogram frames on previously generated frames. This is achieved by employing attention alignments from an autoregressive teacher model to predict the number of spectogram frames needed for each input embedding, thus allowing parallelisation of the generation process to speed up inference. The teacher model they propose is the TransformerTTS model [3] discussed above. In this model, at least one attention head learns the temporal alignment between input frames and output frames. The information on which input frame the head attends to for each output frame can be distilled into the duration information required for training the FastSpeech models.

In contrast to its predecessor, and also to other end-to-end

systems, FastSpeech 2 elegantly solves the mode collapse problem by predicting the pitch contour and the energy contour in dedicated submodules. While FastSpeech learns to predict durations for each phoneme embedding implicitly by means of the teacher model during training, FastSpeech 2 uses ground truth durations as determined by forced alignment to this end. Using a duration predictor enables frames to be computed in parallel, which is not only faster; but also eliminates the early stopping and repetition problems that many autoregressive systems struggle with.

For our contribution we considered both FastSpeech 2 [6] and TransformerTTS [3]. However we make use of the MelGAN architecture [10] for vocoding instead of WaveGlow [8], Parallel WaveGAN [9], or WaveNet [7]. As the name implies, MelGAN is a Generative Adversarial Network (GAN) [11] and consists of a fully convolutional generator that takes a full mel-scaled spectrogram as input. Since the amount of wave samples for a given amount of spectrogram frames can be calculated if the FFT window size and the hop size of the spectrogram are known, it is capable of calculating all of the wave samples in parallel, making it extremely quick. The spectrogram inversion has only one major problem, which is the reconstruction of the missing phase-shift information, which is very difficult to quantify in an objective function. To fix this, MelGAN employs an ensemble of discriminators, which operate on different scales. While the result is not perfectly natural, we find it offered the best quality/speed trade-off at the time of the implementation. At the time of writing this paper however, we find that the more recent HiFi-GAN [12] now outperforms MelGAN.

3. Code-switching

While modern TTS systems do impressively well in the language they are trained on, natural language often is not purely monolingual. In reality, sentences can contain a considerable amount of foreign words. This phenomenon is known as *code-switchig* and is addressed by this year's Blizzard Challenge in the spoke task. Code-switching is difficult for modern TTS because it usually requires synthesising phonemes that are not part of the phoneme inventory of the synthesised language, and therefore will not be present in the training data . One option is to employ additional data in the code-switched language that can be used to tune the NN model so it can synthesise speech in both languages. However, since the phoneme embeddings are a simple lookup table and the system would have no way to distinguish between phones in one language and phones in another language, the system would likely collapse the phonotactics of the two languages, hurting performance in the non-codeswitched language.

An additional consideration is that code-switching is also a challenge for humans, and that humans are rarely perfect in doing so. Even if readers are highly competent speakers of the code-switched language, they will almost never be native speakers of that language. For this scenario, the Perceptual Assimilation Model (PAM) [13, 14] predicts that speakers are more likely to correctly produce foreign phoneme categories with native-like proficiency if these categories are not too close to categories in their native language (L1), while they are expected to map phoneme categories of the code-switched language that are close to L1 categories onto these L1 categories.

The assumption above was corroborated by an analysis of the training and development data. We found that even the (theoretically all-Spanish) training data already contain a few instances of code-switching, underlining the ubiquity of this phenomenon. Confirming our expectations, the speaker rarely produces perfectly English pronunciation in these cases. Consider the examples listed in table 1. For utterance 2204 from the training data the speaker would be expected to code-switch for the English name *Grassville*, which should be pronounced with a [ɡ] at the beginning, and an [æ] vowel in the first syllable. However the speaker maps these to native Spanish [ɣ] and [a], respectively. For utterance 11429, the deviation from the correct English pronunciation is even more obvious: instead of something like [hɑmbl̪ʊɚ], the speaker produces [xomblaʊ̯er]. This shows that she is aiming for English pronunciation: instead of producing a silent /h/, as would be expected for native Spanish words with orthographic h, she replaces the English [h] by [x]. However, she assimilates [n] to [m] following Spanish phonological rules in the context of the upcoming labial [b], and she realises the single [ɚ] by the sequence [er]. Similarly the development data show that the speaker in code-switching cases usually aims for the English pronunciation (for instance, producing *Music* in utterance dev 14988 listed above as [mjusik] rather than Spanish [musik]), but that she also maps most English phoneme categories to Spanish ones (for instance, [z] to [s] in this example).

train 2204	Reflexiona y piensa que esto es todavía mejor que **Grassville**.
train 11429	¿Y si le pides a **Hornblower** que nos lleve allí?
dev 14988	**Mickey's Music Festival**, que estos días se representa en el Palacio de Congresos de Granada.

Table 1: *Examples of code-switching from training and development data.*

We take this observations as an indication that a TTS system does not necessarily have to synthesise code-switched phoneme categories with the perfection of a native speaker; instead it is desirable to achieve a convincing sounding, natural, possibly slightly accented version of the code-switched utterance parts, similar to what a trained reader with a reasonable knowledge of that language would produce. Our solution thus consists in mapping English phoneme categories to close categories that the native speaker employed in the training data.

4. The IMS Toucan toolkit

The IMS Speech Synthesis Toolkit (IMS Toucan), was developed as a tool specifically for teaching speech synthesis to newcomers to the field. It provides implementations of Tacotron 2 [2], TransformerTTS [3], FastSpeech 2 [6], MelGAN [10], and HiFi-GAN [12], which are packaged into easy to use interfaces.

Our PyTorch modules of Tacotron 2, TransformerTTS and FastSpeech 2 are closely derived from the ESPnet toolkit [15, 16]. The main difference between IMS Toucan and ESPnet is that in IMS Toucan all of the models and procedures are wrapped in pure python and are simplified as much as possible, whereas ESPnet contains unified interfaces for speech recognition, speech enhancement, speech translation and much more, which can be overwhelming. Furthermore the IMS Toucan training interfaces are modular, yet inference interfaces combine a synthesis and a vocoder model, to enable models to be used in other projects easily. The PyTorch modules of MelGAN and HiFi-GAN are

closely derived from another ESPnet related repository[2].

Our implementation of FastSpeech 2 contains two significant changes from the original paper. The first is the averaging of the pitch values and energy values for each phoneme according to its duration, as introduced by FastPitch [17], and as implemented in ESPnet [16]. This allows for great controllability, as the pitch and energy (and duration) values for each phoneme can be overwritten at inference time, effectively controlling most perceptual aspects of the produced speech at a very fine-grained level. While this is an added bonus to our system, we did not make use of it in the Blizzard Challenge. The second big change is the use of the convolution-augmented transformer (Conformer), introduced in [18], as the encoder and decoder, also as implemented in ESPnet [16]. This architecture was built with speech recognition in mind, however we find that it also performs very well for speech synthesis, as is also shown in [19].

Now that the toolkit is completed and thoroughly tested, it is freely available[3].

5. Candidate systems for the challenge

Our challenge entry does not use the currently released version of IMS Toucan, but an earlier development version. We built multiple models to compare in preliminary experiments, described in detail later in this section. We use a multi-stage system, starting with text processing, then predicting spectrogram frames using either a FastSpeech 2 model or a TransformerTTS model and finally transforming the spectrogram into a waveform.

Text processing For text processing, we use an open source phonemiser[4] with espeak-ng[5] as its backend. The phonemiser performs rudimentary text cleaning and then transforms the given input text into a sequence of IPA characters, including prosodic markers, such as lengthening and lexical stress. We chose to remove this suprasegmental information from the phoneme sequence and leave those properties to the two end-to-end approaches we employ. This is beneficial because the part of the networks concerned with the upsampling of the input frames to the output frames gets confused by non-monotonic alignments and overlaps in the durations of input frames.

FastSpeech 2 For the FastSpeech 2 based candidate, we performed forced alignment using the Aligner [20] with the calculated phoneme sequences, mapping the phoneme inventory of the phonemiser to that of the Aligner to get the duration information needed. We then use the Dio and Stonemask algorithms to extract pitch contour and energy contour [21]. Then we average the values for pitch and energy over all spectrogram frames which belong to one phoneme according to the durations that the Aligner produces. The phonemes are then transformed into embeddings with random initialisations, using a simple lookup-table based approach. The phoneme embeddings acquire their meaning through the gradient-descent based training over time. The phoneme embeddings are then fed into the FastSpeech 2 module of IMS Toucan to generate a spectrogram.

[2] https://github.com/kan-bayashi/ParallelWaveGAN

[3] https://github.com/DigitalPhonetics/IMS-Toucan

[4] https://github.com/bootphon/phonemiser

[5] https://github.com/espeak-ng/espeak-ng

TransformerTTS To train the TransformerTTS candidate, we apply the same phonemising step and the same lookup table for vectorising the phonemes. Since TransformerTTS does not require alignments, pitch and energy contour, we train the mapping from vectorized phoneme sequence to spectrogram frames directly using the TransformerTTS module of IMS Toucan.

Audio processing We use 16kHz as the audio sampling rate because we find it to be sufficient with respect to quality, faster with respect to inference speed, and it helps with model convergence during training. As our spectrogram representation, we use 80 mel-frequency buckets. We found that normalising the audios to always contain 250ms of silence in the beginning and end of an utterance, as well as adding a begin-of-sentence pseudo-token to the beginning and a silence marker to the end of the text during training helped the model produce more natural prosody. Furthermore we found that adding a silence marker to the end of any utterance during inference significantly improves the naturalness of the synthesized speech.

We trained both the FastSpeech 2 model as well as the TransformerTTS model for 32 hours on an NVIDIA RTX 3090. We trained the MelGAN model for 14 hours on an NVIDIA GTX TITAN X. Our FastSpeech2 based system achieves a Real-Time-Factor[6] of 0.5 on CPU (Core i7-4600U) and 0.05 on GPU (NVIDIA RTX 2070) to get from text to waveform. The same measurements for our TransformerTTS based candidate yield 1.8 on CPU and 0.2 on GPU.

Modifications for the spoke task Given our observations in section 3 of how the human speaker realises code-switching, we decided that due to the relatively pronounced Spanish accent of the speaker, it would be more appropriate and more convincing to define mappings of English to Spanish phonemes than to try and produce accent-free English code-switching. With this premise, we can avoid the issue of collapsing phonemes of both languages. By that, code-switching is reduced to a problem of G2P, which we solve by integrating a language identification (LID) system into our text-preprocessing frontend.

We use the Spanish-English LID module of the CodeSwitch[7] NLP tool, which is based on a multilingual BERT model. This tool takes a code-switched sentence as input and provides a token-wise annotation with language tags for Spanish (*spa*), or English (*en*). Additionally, there is a special tag *ne* denoting named entities. However, regarding *ne* tokens, we cannot differentiate between entities with Spanish (e.g. "Granada"), English (e.g. "New York"), or other (e.g. "Berlin", "Shiraito") pronunciation. We therefore added a hand-crafted lexicon of important English state and city names, which we annotate as *en*, and default all other *ne* to *spa*, mimicking the Spanish speaker's treatment of foreign names, who seems to realise names that are not obviously English following Spanish pronunciation rules.

We find that our LID tool overestimates the amount of code-switched tokens. Thus, we further filter *en*-tagged tokens and keep them only if they (a) occur within a sequence of at least two code-switched tokens, and/or (b) contain character combinations that do not occur in Spanish native words. We consider this an acceptable trade-off between accurately identifying English words and keeping our system comprehensible.

Based on the language annotation of each token, we switch our phonemiser to either Spanish or English. Since our model

[6] Defined as the amount of seconds it takes to produce one second of audio; smaller is better.

[7] https://github.com/sagorbrur/codeswitch

is trained on a Spanish phoneme set only, code-switched tokens still may contain unseen phonemes. Thus, following our analysis of how the human speaker produces code-switched segments, we map (sequences of) English phonemes to the most appropriate Spanish phoneme or sequence of phonemes. Our hand-crafted mapping rules can be seen in table 2. Furthermore we add a pause whenever a language switch occurs and the switched segment is longer than two words, because we found that this improves the naturalness of the code-switching significantly.

en	spa	en	spa	en	spa	en	spa
ɔɪ	oɪ	æ	a	dʒ	tʃ	ɹ	ɾ
oʊ	o	ʊ	u	g	ɣ	ɾ	t
ɚɹ	eɾ	ɔ	o	v	β	i	i
ɚ	eɾ	ɑ	o	z	s	ɐ	a
ɜ	ɛɾ	ʌ	a	ʒ	ʃ		
ə	e	ɪ	i	h	x		

Table 2: *Mappings from English to Spanish phonemes*

Preliminary experiments For the challenge contribution, we investigated several factors. We were interested whether the autoregressive TransformerTTS or the non-autoregressive FastSpeech 2-based model performed better, especially given the fact that 5 hours of training data is comparably small for NN TTS. Further we wanted to test whether the knowledge-distillation based durations of the TransfomerTTS teacher in FastSpeech 2 are sufficient, or if ground-truth durations from the Aligner are needed to get best results. And finally we wanted to see whether the 5 hours of training data were sufficient, or whether we can achieve improvements by using additional data. To that end, we used an additional 50h of freely available data from another speaker[8]. Since the data contained utterances that were cut slightly before word endings, we performed some cleaning based on the results of our forced alignment to eliminate these problematic utterances. We then used the data to pre-train our model and used the Blizzard training data for fine-tuning afterwards.

In order to decide which system to submit to the listening test, we built 6 models using all possible combinations of the factors listed above. Out of these we picked the one with the best robustness and naturalness according to our own subjective perception. Those six combinations are displayed in table 3.

Architecture	Pretraining	Duration Extraction
TransformerTTS	Yes/No	-
FastSpeech 2	Yes/No	Knowledge Distillation
FastSpeech 2	Yes/No	Aligner

Table 3: *Overview of candidate systems for the challenge.*

Surprisingly, we find that the performance given more data only increases in the case of TransformerTTS, because the model becomes a lot more robust, even if the naturalness suffers a bit. The performance of our FastSpeech 2 systems always decreases when pre-training on the larger dataset. This is likely due to the synthesis being trained as a single speaker system, yet the speakers of the two corpora are very different, both with respect to their voice, as well as their speaking style.

[8] https://www.kaggle.com/carlfm01/120h-spanish-speech

The TransformerTTS system in general produces natural and intelligible speech, however it makes more mistakes and suffers from unnatural prosody. The frequent repetitions in the phonotactics of Spanish tend to trigger the repeating word problem in the autoregressive system. We thus find the non-autoregressive TTS without pre-training to be the best in all of our pairwise comparisons. The impact of Aligner-based durations versus knowledge-distillation based durations is also very noticeable. The model trained on distilled duration information produces very unnatural prosody, likely due to mistakes in duration leading to follow-up mistakes in phoneme-averaged pitch and duration predictor. Furthermore, the system without proper durations sometimes leaves out phonemes entirely, which it likely picked up from the repeating-word confusion of the teacher-model, even when using teacher-forcing to generate the attention-based alignments. The FastSpeech 2 model trained with durations derived from the Aligner produces accurate pitch and energy contours, properly articulates all of the phonemes and even handles silences and pauses remarkably well. This is illustrated in figure 1. The lower part of the graphic shows the spectrogram that the FastSpeech 2 model produces for an utterance that contains severe code-switching (50% of the words are English). The spectrogram includes the phoneme boundaries as predicted by the duration model within FastSpeech 2. The corresponding phonemes for each segment are displayed below in IPA notation. The top part of the graphic shows the wave that the MelGAN model produces for this spectrogram. It can be seen that the predicted phoneme boundaries match very well with phoneme boundaries in the spectrogram. This was not the case when using durations produced by knowledge distillation. We conclude our preliminary experimentation with the finding that using homogeneous and correctly annotated but fewer data is better than using many hours of training data, i.e. it cannot always be assumed that the model generalises over imperfections given enough data.

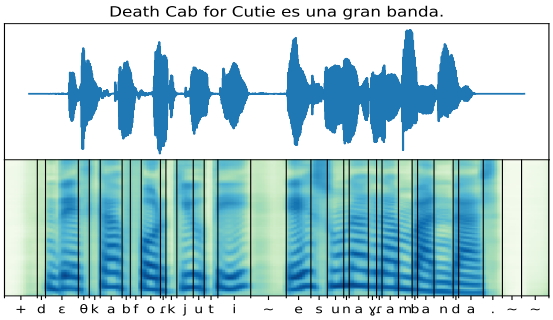

Figure 1: *Example output of the system selected for the challenge. ~ denotes silence, + denotes begin-of-sentence.*

6. Evaluation

This year 12 teams participated in the hub task, and 10 in the spoke task. Our system is identified as the letter *N*. The letter *R* denotes natural speech of the original speaker for reference.

Hub task All systems, including natural speech, were rated according to mean opinion scores (MOS) regarding naturalness and similarity to the original speaker. Please note that mean scores and their standard deviation may be misleading, as the

scores do not meet the normality requirements for parametric and descriptive statistics. The same is true for the evaluation of the spoke task. Further, intelligibility tests were performed where listeners were asked to type in what they heard. We report absolute scores for naturalness in figure 2, and word error rates (WER) for an intelligibility test are shown in figure 3. We attest our comparatively low absolute MOS scores to the low fidelity of 16kHz that we chose and the high pitched hum, which MelGAN sometimes produces during segments that should be silent. The artefacts that are introduced by MelGAN are most likely also the reason for the relatively high WER of our system.

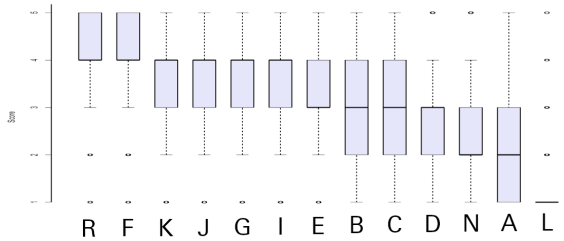

Figure 2: *Naturalness scores in the hub task*

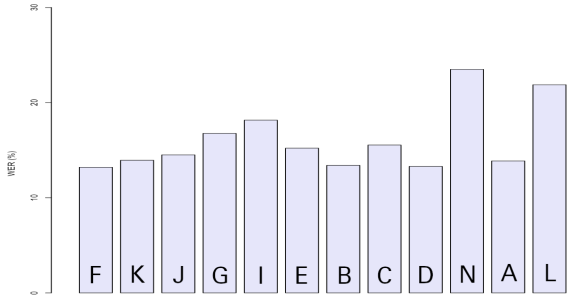

Figure 3: *Intelligibility on semantically unpredictable sentences*

A	B	C	D	E	F	G	I	J	K	L	R

Figure 4: *Similarity of other participating systems to original speaker, relative to our system. White: significantly better; Gray: equal; Black: significantly worse*

Regarding similarity to original speaker in the hub task, we report the scores for our system regarding statistically significant differences using Wilcoxon's signed rank tests in figure 4. Systems *B* and *C* are the only two other systems using a sampling rate of 16kHz. The results thus indicate that all systems which used a higher fidelity performed significantly better than ours (white cells), while we perform on par with the two other 16 kHz systems (gray cells), and outperform system L (black cell). We see these results as indicative of the fidelity having the most impact on the MOS of similarity and naturalness.

Spoke task For the spoke task, listeners additionally rated the acceptability of English words on a score from 1 to 5, shown in figure 5. Besides the aforementioned weaknesses of our system regarding sample rate and choice of vocoder, we attribute lower acceptability scores to failures in language identification. Figure 6 again shows the differences of our system to other systems in the spoke task according to Wilcoxon's signed rank tests. Our approach on code-switching is on par with or better than half of the other submitted systems.

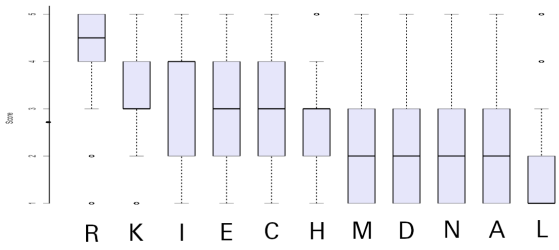

Figure 5: *Acceptability of English words*

Acc	A	C	D	E	H	I	K	L	M	R
Nat	A	C	D	E	H	I	K	L	M	R
Sim	A	C	D	E	H	I	K	L	M	R

Figure 6: *Acceptability, Naturalness and Similarity of other systems relative to ours. White: significantly better; Gray: equal; Black: significantly worse*

7. Conclusions and Outlook

Synthesising Spanish as well as code-switched Spanish-English utterances using IMS Toucan yields speech of good quality with extremely fast training and inference time compared to other NN systems even when using no more than five hours of training data. Reducing the problem of code-switching in synthesis to a problem of G2P seems to work well and is similar to how human speakers produce code-switched segments. Mapping non-native phones to their closest available counterparts is however quite labour intensive. We will address this aspect in future work.

Regarding naturalness and audio quality, we identify the use of a low fidelity and the MelGAN architecture for vocoding as key factors. Fidelity can easily be increased, however a higher fidelity also leads to longer runtime and increased hardware requirements. We consider keeping those two attributes of a system low to be equally crucial as a good naturalness. These properties are unfortunately not reflected in the scores of the challenge. To address the second factor, the high pitched hum of MelGAN, we find that the aforementioned HiFi-GAN with its multi periodicity discriminator does not have this problem, while maintaining the high inference speeds and low computational costs. At the time of writing this paper, we have already extended IMS Toucan with a HiFi-GAN vocoder.

Our results further confirm that non-autoregressive synthesis is superior to autoregressive synthesis with respect to robustness, speed and quality, as expected. The five hours of training data given are already sufficient to train a high quality non-autoregressive TTS. However, in our opinion it highly depends on the quality of the alignments and the uniformity of the data, if no further distinguishing features, such as embeddings of speaker and speaking style, are given.

8. Acknowledgements

We would like to thank the organisers of the challenge for providing the high quality data, holding the extensive evaluation and organising this enriching event.

9. References

[1] S. O. Arık, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep voice: Real-time neural text-to-speech," in *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," 2018.

[3] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[5] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: fast, robust and controllable text to speech," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 3171–3180.

[6] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations*, 2020.

[7] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016.

[8] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *ICASSP 2019 – 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[9] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel waveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[10] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[12] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[13] C. T. Best, "A direct realist view of cross-language speech perception," in *Speech Perception and Linguistic Experience: Issues in Cross Language Research*, W. Strange, Ed. Timonium, MD: York Press, 1995, pp. 171—-204.

[14] C. T. Best and M. D. Tyler, "Nonnative and second-language speech perception," in *Language Experience in Second Language Speech Learning*, M. J. Munro and O.-S. Bohn, Eds. Amsterdam: John Benjamins, 2007, pp. 13—–34.

[15] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[16] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7654–7658.

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

[18] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Proc. Interspeech 2020*, pp. 5036–5040, 2020.

[19] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi *et al.*, "Recent developments on ESPnet toolkit boosted by conformer," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5874–5878.

[20] S. Rapp, "Automatic phonemic transcription and linguistic annotation from known text with hidden markov models. an aligner for german," 1995.

[21] 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, no. 7, pp. 1877–1884, 2016.