# FAIRSEQ S$^2$: A Scalable and Integrable Speech Synthesis Toolkit

**Changhan Wang*, Wei-Ning Hsu*, Yossi Adi, Adam Polyak, Ann Lee,**
**Peng-Jen Chen, Jiatao Gu, Juan Pino**
Facebook AI
{changhan,wnhsu,adiyoss,adampolyak,annl,
pipibjc,jgu,juancarabina}@fb.com

## Abstract

This paper presents FAIRSEQ S$^2$, a FAIRSEQ extension for speech synthesis. We implement a number of autoregressive (AR) and non-AR text-to-speech models, and their multi-speaker variants. To enable training speech synthesis models with less curated data, a number of preprocessing tools are built and their importance is shown empirically. To facilitate faster iteration of development and analysis, a suite of automatic metrics is included. Apart from the features added specifically for this extension, FAIRSEQ S$^2$ also benefits from the scalability offered by FAIRSEQ and can be easily integrated with other state-of-the-art systems provided in this framework. The code, documentation, and pre-trained models will be made available at https://github.com/pytorch/fairseq/tree/master/examples/speech_synthesis.

## 1 Introduction

Speech synthesis is the task of generating speech waveforms with desired characteristics, including but not limited to textual content (Hunt and Black, 1996; Zen et al., 2009; Shen et al., 2018; Ping et al., 2017; Li et al., 2019), speaker identity (Jia et al., 2018; Cooper et al., 2020), and speaking styles (Wang et al., 2018; Skerry-Ryan et al., 2018; Akuzawa et al., 2018; Hsu et al., 2018). It is also more often referred to as Text-to-Speech (TTS) when text is used as input to the system. Along with automatic speech recognition (ASR) and machine translation (MT), these language technologies have advanced rapidly over the past few years (Tan et al., 2021). Traditionally, these tasks may be used in conjunction to form a system (e.g., combining the three for speech-to-speech translation), but they rarely leverage each other during training. As a result, each application used to have its own dedicated open-source toolkit, for example, Kaldi (Povey et al., 2011) and HTK (Young et al., 2002) for ASR, HTS (Zen et al., 2007), Merlin (Wu et al., 2016), STRAIGHT (Kawahara et al., 1999), and WORLD (Morise et al., 2016) for speech synthesis, and Moses (Koehn et al., 2007) for MT.

Recently, there are growing interactions among these systems in the learning process. For example, Hayashi et al. (2018) and Rosenberg et al. (2019) propose to leverage speech synthesis systems to generate paired text and speech data for ASR training; Tjandra et al. (2017), Hori et al. (2019), and Baskar et al. (2019) chain ASR and TTS together to form a loop for semi-supervised learning with cycle-consistency loss; Weiss et al. (2017), Li et al. (2020), and Jia et al. (2019) demonstrate that it is possible to build an end-to-end system translating speech into text or speech in a target language.

Beyond text-based systems, there is also an emerging research topic that explores the use of units discovered from self-supervised speech representation learning (Oord et al., 2017; Baevski et al., 2019; Harwath et al., 2019; Hsu et al., 2021) to replace text for representing the lexical content in numerous applications, such as language modeling (Lakhotia et al., 2021), speech resynthesis (Polyak et al., 2021), image captioning (Hsu et al., 2020), and translation (Tjandra et al., 2020; Hayashi and Watanabe, 2020). This line of research bypasses the need for text and makes technologies applicable even to unwritten languages. However, to interpret the output of such systems - a sequence of learned units, a unit-to-speech model is required. This brings up the need of a framework for broader speech synthesis systems that can alternatively take learned units as input. These research directions can benefit from having a single toolkit with different state-of-the-art language technologies.

In this paper, we introduce FAIRSEQ S$^2$, a FAIRSEQ (Ott et al., 2019) extension for speech synthesis. FAIRSEQ is a popular open-source sequence modeling toolkit based on PyTorch (Paszke

[*] Equal contribution.

*Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 143–152
November 7–11, 2021. ©2021 Association for Computational Linguistics

et al., 2019) that allows researchers and developers to train custom models. It offers great support for training large models on large scale data, and provides a number of state-of-the-art models for language technologies. We extend FAIRSEQ to support speech synthesis in this work. In particular, we implement a number of popular text-to-spectrogram models, with interface to both signal processing-based and neural vocoders. Multi-speaker variants of those models are also implemented. While speech synthesis often relies on subjective metrics such as mean opinion scores for benchmarking, we implemented a suite of widely used automatic evaluation metrics to facilitate faster iteration on model development. Last but not least, we support a number of text and audio preprocessing modules, which allow developers to quickly build a new dataset from less curated in-the-wild data for speech synthesis.

The main contribution of this work is threefold. First, we implement a number of state-of-the-art models and provide pre-trained checkpoints and recipes, which can be used by researchers as baselines or as building blocks in applications such as text-to-speech translation. Second, we create pre-processing tools that enable developers to use customized data to build a TTS model, and demonstrate the effectiveness of these tools empirically. Lastly, as part of the FAIRSEQ codebase, this speech synthesis extension allows easy integration with numerous state-of-the-art MT, ASR, ST, LM, and self-supervised systems already built on FAIRSEQ. We provide an example by building a unit-to-speech system that can be used for text-free research.

The rest of the paper is organized as follows: Section 2 describes the features of FAIRSEQ $\mathrm{S}^2$. Experiments are presented in Section 3. Related work is discussed in Section 4, and we conclude this work in Section 5.

## 2 Features

**Fairseq Models** FAIRSEQ provides a collection of MT (Ng et al., 2019), ST (Wang et al., 2020), unsupervised speech pre-training and ASR (Baevski et al., 2020b; Hsu et al., 2021) models that demonstrate state-of-the-art performance on standard benchmarks. They are open-sourced with pre-trained checkpoints and can be integrated or extended easily for other tasks.

**Speech Synthesis Extension** FAIRSEQ $\mathrm{S}^2$ adds state-of-the-art text-to-spectrogram prediction models, Tacotron 2 (Shen et al., 2018) and Transformer (Li et al., 2019), which are AR with encoder-decoder model architecture. For the latest advancements on fast non-AR modeling, we provide FastSpeech 2 (Ren et al., 2019, 2020) as an example. All our models support the multi-speaker setting via pre-trained (Jia et al., 2018) or jointly trained speaker embeddings (Arik et al., 2017; Chen et al., 2020). Note that the former enables synthesizing speech for speakers unseen during training. For FastSpeech 2, pitch and speed are controllable during inference. For spectrogram-to-waveform conversion (vocoding), FAIRSEQ $\mathrm{S}^2$ has a built-in Griffin-Lim (Griffin and Lim, 1984) vocoder for fast model-free generation. It also provides examples for using external model-based vocoders, such as WaveGlow (Prenger et al., 2019) and HiFi-GAN (Kong et al., 2020).

**Speech Preprocessing.** Recent advances in neural generative models have demonstrated that neural-based TTS models, can synthesize high-quality, natural and intelligible speech. However, such models usually require high-quality, and clean speech data (Zhang et al., 2021). In order to enable leveraging noisy data for TTS training, we propose a speech preprocessing pipeline to enhance and filter data. The proposed pipeline is comprised of three main components: i) Background noise removal, ii) Voice Activity Detector (VAD), and iii) Outlier filtering using both Signal-to-Noise Ratio (SNR) and Character Error Rate (CER).

First, a speech enhancement model is applied over input recordings to remove background noise. We used the speech enhancement model proposed by (Defossez et al., 2020) where the $i_{th}$ convolutional layer has $2^{i-1} * 64$ output channels. As suggested by the authors, we additionally used a dry/wet knob, i.e. the final output is $dry \cdot \mathbf{x} + (1 - dry) \cdot \hat{\boldsymbol{y}}$, where $\mathbf{x}$ is the noisy input signal and $\hat{\boldsymbol{y}}$ is the output of the enhancement model. We experiment with $dry \in \{0.0, 0.01, 0.05, 0.1\}$ and find 0.01 to perform the best.

Next, we apply VAD to remove silence from the denoised utterances, as silence can vary in length significantly which causes increasing uncertainty and therefore degrades TTS performance. Silence regions at the beginning and end of the utterances are completely removed. In case we encounter a silence segment in the middle of the signal in where its length is greater than 300ms we replace it with a 300ms artificially generated silence (since completely removing silence regions produces unnatural speech). Silence regions of less than 300ms are left unchanged. We use the open-source implementation of the Google WebRTC VAD (Wiseman, 2016), of which four aggressiveness levels {0, 1, 2, 3} can be set. A higher aggressiveness level removes more silences but comes at the risk of removing partial speech. The aggressiveness level corresponds to the size of the processing window (a larger processing window will make the VAD work at a coarser level and remove silence frames more aggressively).

Lastly, we notice that in extremely noisy recordings (SNR close to zero), the generated denoised samples are often not intelligible enough to train a TTS or contain distortion artifacts. In addition, when setting the VAD aggressiveness level high, speech may be truncated along with silence. To remedy this, we proposed two outliers filtering methods. The first approach is based on SNR estimation. We approximate the noise by subtracting the output of the enhancement model from the input-noisy speech, then we compute the SNR between the two. The second approach is based on applying an Automatic Speech Recognition (ASR) over the denoised speech and compute the CER against the target transcription.

**Computation** FAIRSEQ is implemented in PyTorch (Paszke et al., 2019) and provides efficient batching, gradient accumulation, mixed precision training (Micikevicius et al., 2017), model parallelism, multi-GPU as well as multi-machine training for computational efficiency on large-scale experiments and enabling training gigantic models.

**Quantitative Metrics** We provide automatic metrics for fast evaluation in model development. Similarly to (Polyak et al., 2020), we report Gross Pitch Error (GPE) (Nakatani et al., 2008), Voicing Decision Error (VDE) (Nakatani et al., 2008), and F0 Frame Error (FFE) (Chu and Alwan, 2009) to evaluate F0 reconstructions of the generated speech. We additionally, report Mel Cepstral Distortion (MCD), Mel Spectral Distortion (MSD), and CER to evaluate both the overall similarity to the target speech and content intelligibility (Weiss et al., 2021).

**(i) GPE** GPE is an objective metric which measures the portion of voiced audio frames with a pitch error of more than 20%.

$$\mathrm{GPE}(\boldsymbol{p}, \hat{\boldsymbol{p}}, \boldsymbol{v}, \hat{\boldsymbol{v}}) = \frac{\sum_t \mathbb{1}[|\boldsymbol{p}_t - \hat{\boldsymbol{p}}_t| > 0.2 \cdot \boldsymbol{p}_t]\mathbb{1}[\boldsymbol{v}_t]\mathbb{1}[\hat{\boldsymbol{v}}_t]}{\sum_t \mathbb{1}[\boldsymbol{v}_t]\mathbb{1}[\boldsymbol{v}'_t]} \tag{1}$$

where $\boldsymbol{p}_t, \boldsymbol{p}'_t$ are the pitch frames from the target and generated signals, $\boldsymbol{v}_t, \hat{\boldsymbol{v}}_t$ are the voicing decisions from the target and generated signals, and $\mathbb{1}$ is the indicator function.

**(ii) VDE** VDE measures the portion of frames with voicing decision error,

$$\mathrm{VDE}(\boldsymbol{v}, \hat{\boldsymbol{v}}) = \frac{\sum_{t=1}^{T-1} \mathbb{1}[\boldsymbol{v}_t \neq \hat{\boldsymbol{v}}_t]}{T}, \tag{2}$$

where $T$ is the total number of frames.

**(iii) FFE** Combining GPE and VDE, FFE measures the percentage of frames that contain a deviation of more than 20% in pitch value or have a voicing decision error.

$$\mathrm{FFE}(\boldsymbol{p}, \hat{\boldsymbol{p}}, \boldsymbol{v}, \hat{\boldsymbol{v}}) = \mathrm{VDE}(\boldsymbol{v}, \hat{\boldsymbol{v}}) + \frac{\sum_{t=1}^{T-1} \mathbb{1}[|\boldsymbol{p}_t - \hat{\boldsymbol{p}}_t| > 0.2 \cdot \boldsymbol{p}_t]\mathbb{1}[\boldsymbol{v}_t]\mathbb{1}[\hat{\boldsymbol{v}}_t]}{T}. \tag{3}$$

**(iv) MCD/MSD** These are defined as the root mean squared error of the synthesized speech against the reference speech computed on the 13-dimensional MFCC features for MCD and log-mel spectral features for MSD. Since the reference and the synthesized speech may not be aligned frame-by-frame, instead of zero-padding the shorter one and assuming they are frame-wise aligned as done in Skerry-Ryan et al. (2018), we follow Weiss et al. (2021) and use dynamic time warping (Berndt and Clifford, 1994) to align the frames from the two sequences. The main difference between these two metrics lies in the features they compute distortion on: MFCC features aim to capture phonetic information while removing speaker information, while log-mel spectral features encode both, and hence MCD addresses phonetic similarity more.

**(v) CER** CER is computed between the transcription of the generated audio against the input text using an ASR system publicly available in FAIRSEQ.

**Visualization** FAIRSEQ integrates Tensorboard[1] for monitoring holistic metrics during model training. It also has VizSeq (Wang et al., 2019) integration for offline sequence-level error analysis,

[1] https://github.com/tensorflow/tensorboard

| | | MCD | CER (S/D/I) | MOS |
|---|---|---|---|---|
| Orig. Audio | | - | 3.3 (0.2/0.5/2.5) | 4.53±0.05 |
| TFM | Char | 4.2 | 6.0 (1.1/1.6/3.2) | 4.09±0.06 |
| | g2pE | 3.9 | 5.2 (1.1/1.2/2.9) | 4.18±0.06 |
| | espk | 4.4 | 5.3 (0.7/1.1/3.5) | 4.17±0.06 |
| | Unit | 3.4 | 6.2 (1.6/1.4/3.2) | 4.18±0.05 |
| FS2 | g2pE | 3.9 | 4.9 (1.2/0.8/2.9) | 4.15±0.09 |
| | Unit | 3.4 | 7.9 (2.7/2.2/3.1) | 3.99±0.05 |

Table 1: **Evaluation on LJSpeech.** We compare autoregressive model (“TFM") with non-autoregressive model (“FS2"), as well as 3 different types of inputs: characters (“char"), phonemes (“g2pE" and “espk") and HuBERT units (“unit").

| Audio Preprocessing | Hours | CER (dev) |
|---|---|---|
| Raw | 44 | 0.8 |
| DN+VAD-1 | 33 | 1.0 |
| DN+VAD-2 | 32 | 1.2 |
| DN+VAD-3 | 26 | 6.8 |
| DN+VAD-3 + FLT | 20 | 1.6 |

Table 2: **Audio preprocessing settings on VCTK.** FLT removes samples with CER $>$ 10%.

where transcript and target/predicted speech are visualized in Jupyter Notebook interface. FAIRSEQ $S^2$ further adds generated spectrogram and waveform samples to Tensorboard for model debugging.

# 3 Experiments

We evaluate our models in three settings: single-speaker synthesis, multi-speaker synthesis and multi-speaker synthesis using noisy data.

## 3.1 Experimental Setup

We use either characters, phonemes or discovered units as input representations. To convert texts into phonemes, we employ g2pE (Park, 2019) or Phonemizer (Bernard, 2015) with espeak-ng[1] backend. We use the Montreal Forced Aligner (McAuliffe et al., 2017) to obtain phonemes with frame durations for FastSpeech 2 training, which is based on the same pronunciation dictionary (CMUdict) as g2pE. For discovered units, we extract frame-level units using a Base HuBERT model trained on LibriSpeech[1] and collapse consecutive units of the same kind. We use the run length of identical units before collapsing as target duration for FastSpeech 2 training. We use a reduction factor (number of frames each decoder step predicts) of 4 for Transformer and 1 for FastSpeech 2 by default.

We resample audios to 22,050Hz and extract log-Mel spectrogram with FFT size 1024, window length 1024 and hop length 256. We optionally preprocess audios to improve model training: denoising (“DN"), level-2 or level-3 VAD (“VAD-2" or "VAD-3"), filtering by SNR$>$ 15 and CER$<$ 10% (“FLT") and volume normalization (“VN").

We use MCD and CER for automatic evaluation. MCD is computed on Griffin-Lim vocoded reference and model output spectrograms. We use vocoded references as opposed to the original ones to eliminate the error introduced by the vocoder and focus the evaluation on spectrogram prediction. HiFiGAN vocoders trained on each dataset are used to generate waveforms for CER evaluation. The large wav2vec 2.0 (Baevski et al., 2020a) ASR model, which achieves WERs of 1.8% and 3.3% on Librispeech test-clean and test-other, respectively and is provided in FAIRSEQ[1], is used both for CER filtering and evaluation. GPE, VDE, and FFE are not reported here, because these metrics are more meaningful when prosody modeling is taken into account (Polyak et al., 2020; Skerry-Ryan et al., 2018; Wang et al., 2018). For subjective evaluation, we conduct a Mean Opinion Score (MOS) test using the CrowdMOS package (Ribeiro et al., 2011) using the recommended recipes for detecting and discarding inaccurate scores. We randomly sample 100 speech utterances from the test set and collect manual scores using a crowd sourcing framework. The same samples are used across all tested methods. Each sample is rated by at least 10 raters on a scale from 1 to 5 with 1.0 point increments. Overall, scores for each tested method are averaged across more than 1000 manual annotations. We report both average MOS scores together with a 95% confidence interval (CI95).

## 3.2 Single-Speaker Synthesis on LJSpeech

LJSpeech (Ito and Johnson, 2017) is a single-speaker TTS corpus with 13,100 English speech samples (around 24 hours) from audiobooks. We follow the setting in Ren et al. (2020) to use 349 samples (with document title LJ003) for validation, 523 samples (with document title LJ001 and LJ002) for testing and the rest for training.

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

[1] https://dl.fbaipublicfiles.com/hubert/hubert_base_ls960.pt

[1] https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_vox_960h_pl.pt

| Audio | Spk. | Red. | MCD | CER | MOS |
|---|---|---|---|---|---|
| Original | - | - | - | 1.8 | 4.27±0.07 |
| Raw | LUT | 2 | 4.9 | 65.2 | 1.77±0.08 |
| | | 4 | 3.3 | 12.1 | 2.77±0.08 |
| DN+VAD-3 | LUT | 2 | 3.6 | 9.8 | 3.34±0.06 |
| | | 4 | 3.4 | 6.9 | 3.30±0.06 |
| DN+VAD-3 +FLT | LUT | 2 | 3.6 | 9.7 | 3.38±0.06 |
| | | 4 | 3.4 | 6.0 | 3.42±0.05 |
| | Emb | 2 | 3.6 | 7.6 | 3.38±0.06 |
| | | 4 | 3.5 | 5.8 | 3.25±0.08 |

Table 3: **Evaluation on VCTK.** We use Transformer with character inputs, and compare 3 audio preprocessing settings and 2 types of speaker embeddings.

| Audio | MCD | CER | MOS |
|---|---|---|---|
| Original | - | 3.0 | 4.0±0.057 |
| VN | 5.3 | 21.0 | 2.97±0.081 |
| DN+VAD-2+VN | 5.5 | 14.2 | 3.22±0.065 |
| DN+VAD-2+FLT+VN | 5.4 | 12.7 | 3.17±0.056 |

Table 4: **Evaluation on Common Voice English portion (top 200 speakers only).** We use Transformer model with phoneme (g2pE) inputs and compare 3 audio preprocessing settings.

On this de-facto standard benchmark, we compare autoregressive model (Transformer, “TFM") with non-autoregressive model (FastSpeech 2, “FS2"), as well as 3 different types of inputs: characters, phonemes (from g2pE or espeak-ng) and HuBERT units. We see from Table 1 that FastSpeech 2 performs comparably well to Transformer with phoneme inputs (g2pE), both achieving 4.2 MOS. However, the latter does not require input-output alignments for model training and supports more types of inputs—it achieves 4.1 MOS with characters (no need for phonemization), and 4.2 MOS with simpler phonemes (espeaker-ng). The task falls into the re-synthesis setting with unit inputs. We notice that FastSpeech 2 performs worse (4.0 vs. 4.2 on MOS) in this setting, likely due to the finer-grained inputs and its simplified attention mechanism.

### 3.3 Multi-Speaker Synthesis on VCTK

VCTK (Veaux et al., 2017) is a multi-speaker English TTS dataset that contains 44 hours of read speech from 109 speakers with various English accents[1]. We randomly sample 50 utterances for validation and 100 utterances for testing, and use the rest for training.

Speech recordings from VCTK include considerable amount of silence as shown in Figure 1 (raw); therefore, silence removal is considered a standard preprocessing step for VCTK (Jia et al., 2018; Cooper et al., 2020). Figure 1 shows silence-removed spectrograms with three VAD aggressiveness levels. We see that a higher aggressiveness level removes more silence, but may also truncate the speech. The dataset durations after silence removal and filtering with CER $< 10\%$ are listed in Table 2, along with the validation CER.

We use this dataset to study how audio-preprocessing and speaker representation affect the performance of TTS. We train a transformer TTS model with a reduction factor (i.e. how many frames each decoding step predicts) of 2 or 4 on three sets of audio: raw data (Raw), DN+VAD-3, and DN+VAD-3+FLT. A speaker embedding lookup table (LUT) is used by default. In addition, we train models on DN+VAD-3+FLT with a fixed embedding (Emb) for each speaker inferred from a pre-trained speaker verification model (Heigold et al., 2016), which would enable synthesizing the voice of an unseen speaker. Results in Table 3 show that increasing the reduction factor from 2 to 4 improves the performance consistently. Specifically, we found that without VAD, the model fails to train when using a reduction factor of 2. Finally, we found that using a pre-trained speaker embedder achieves similar performance than using a learnable lookup table, while enabling synthesizing speech for unseen speakers.

### 3.4 Multi-Speaker Synthesis using Noisy Data from Common Voice

Common Voice (Ardila et al., 2020) is a multi-speaker speech corpus with around 4.2K hours of read speech in 40 languages (version 4). It is crowd-sourced from around 78K voice contributors in various accents, age groups and genders. We use its English portion and select data from the top 200 speakers by duration (total 226 hours).

The audio data in this corpus is expectedly noisy given the lack of curated recording environments. We explore if speech processing can counteract the negative factors (background noise, long silence, variable volume across clips, etc.) during recordings and improve model training. Specifically, we examine 3 preprocessing settings with Transformer model and phoneme (g2pE) inputs:

[1] https://datashare.ed.ac.uk/handle/10283/3443

|  | Multi-Spk TTS | Non-AR TTS | ASR | MT | ST | Speech Pre-training | Audio Preprocess | Auto. Metrics |
|---|---|---|---|---|---|---|---|---|
| coqui TTS[1] | ✓ | ✓ |  |  |  |  |  |  |
| OpenSeq2seq[2†] |  |  | ✓ | ✓ |  |  |  |  |
| ESPnet-TTS[3] | ✓ | ✓ | ✓ | ✓ | ✓ |  | ✓[‡] | ✓[‡] |
| NeMo[4] | ✓ | ✓ | ✓ | ✓ |  |  |  |  |
| FAIRSEQ S$^2$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 5: **Comparison of FAIRSEQ S$^2$ with counterpart speech synthesis toolkits (as of June 2021).** [1] GitHub: coqui-ai/TTS. [2] Kuchaiev et al. (2018a). [3] Hayashi et al. (2020). [4] GitHub: NVIDIA/NeMo. [†] Archived and no longer updated. [‡] Supporting only VAD for audio preprocessing and MCD for automatic metric.

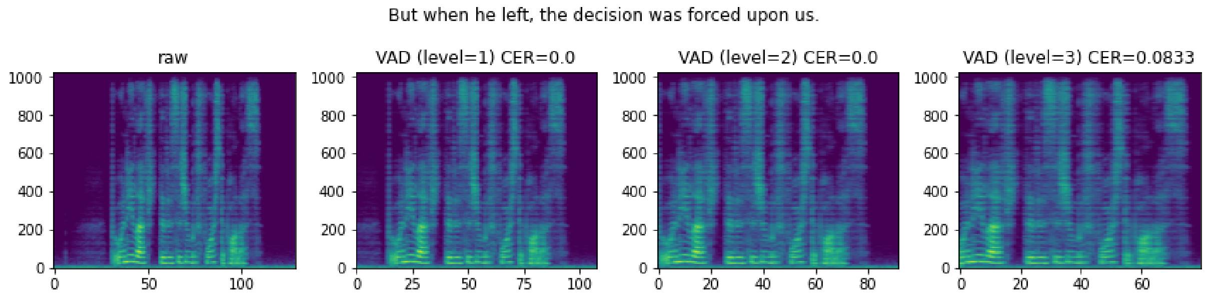

Figure 1: **A VCTK example.** With VAD level 3, the first word "But" is detected as silence and cut off.

VN, DN+VAD-2+VN and DN+VAD-2+FLT+VN. As shown in Table 4, the original audio has 0.3/0.5 lower MOS than the LJSpeech/VCTK one, confirming its relatively low recording quality. Noise and silence removal improve synthesis quality significantly by 0.2 MOS (DN+VAD-2+* vs. VN). Filtering by SNR and CER improves both model fitting (-0.1 MCD) and intelligibility (-1.5 CER) given the removal of difficult training examples.

## 4 Related Work

There are many existing open-source repositories for speech synthesis. The most prominent toolkits for conventional statistical parametric speech synthesis (SPSS) include HMM/DNN-based Speech Synthesis System (HTS) (Zen et al., 2007) and Merlin (Wu et al., 2016). These rely heavily on feature engineering and use signal processing-based vocoders like STRAIGHT (Kawahara et al., 1999) and WORLD (Morise et al., 2016) to synthesize waveforms from acoustic features (e.g., fundamental frequency, spectral envelope, and aperiodic information). Recently, end-to-end models that take minimally pre-processed features (characters and mel-spectrograms) have achieved superior performance compared to conventional systems (Shen et al., 2018), especially when paired with neural vocoders (Prenger et al., 2019; Kong et al., 2020). There are a number of open-source implementations available on Github [1], however, these repositories are solely for text-to-speech synthesis, and mostly support one model only.

ESPnet (Watanabe et al., 2018; Hayashi et al., 2020), NeMo, and OpenSeq2Seq (Kuchaiev et al., 2018b) are the most similar toolkits that also support multiple tasks. As listed in Table 5, FAIRSEQ S$^2$ provides more audio preprocessing tools and automatic metrics for building and evaluating speech synthesis models on custom datasets. As part of FAIRSEQ, it can also be easily integrated with numerous state-of-the-art models already provided in FAIRSEQ for exploring novel ideas. For example, we demonstrate that units discovered from a self-supervised speech pre-training model can be used to build a unit-to-speech system that converts output from systems like unit LM (Lakhotia et al., 2021) or image-to-unit (Hsu et al., 2020) to speech.

## 5 Conclusion

This paper introduces FAIRSEQ S$^2$, a FAIRSEQ extension for speech synthesis. We believe this extension will allow researchers and developers to more easily test novel ideas for language technologies by providing great support for scalability, integrability, and a wealth of tools for curating data as well as automatically evaluating trained systems.

[1] coqui-ai/TTS, Kyubyoung/tacotron, NVIDIA/tacotron2, Rayhane-mamah/Tacotron2, r9y9/deepvoice3_pytorch

## References

Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. 2018. Expressive speech synthesis via modeling expressions with variational autoencoder. *arXiv preprint arXiv:1804.02135*.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. 2017. Deep voice 2: Multi-speaker neural text-to-speech. *arXiv preprint arXiv:1705.08947*.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2019. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020a. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020b. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Murali Karthick Baskar, Shinji Watanabe, Ramon Astudillo, Takaaki Hori, Lukáš Burget, and Jan Černockỳ. 2019. Semi-supervised sequence-to-sequence asr using unpaired speech and text. *arXiv preprint arXiv:1905.01152*.

Mathieu Bernard. 2015. Phonemizer. `https://github.com/bootphon/phonemizer`.

Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:.

Mingjian Chen, Xu Tan, Yi Ren, Jin Xu, Hao Sun, Sheng Zhao, Tao Qin, and Tie-Yan Liu. 2020. Multispeech: Multi-speaker text to speech with transformer. *arXiv preprint arXiv:2006.04664*.

Wei Chu and Abeer Alwan. 2009. Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend. In *ICASSP*.

Erica Cooper, Cheng-I Lai, Yusuke Yasuda, Fuming Fang, Xin Wang, Nanxin Chen, and Junichi Yamagishi. 2020. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6184–6188. IEEE.

Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. 2020. Real time speech enhancement in the waveform domain. *INTERSPEECH*.

Daniel Griffin and Jae Lim. 1984. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243.

David Harwath, Wei-Ning Hsu, and James Glass. 2019. Learning hierarchical discrete linguistic units from visually-grounded speech. *arXiv preprint arXiv:1911.09602*.

Tomoki Hayashi and Shinji Watanabe. 2020. Discretalk: Text-to-speech as a machine translation problem. *arXiv preprint arXiv:2005.05525*.

Tomoki Hayashi, Shinji Watanabe, Yu Zhang, Tomoki Toda, Takaaki Hori, Ramon Astudillo, and Kazuya Takeda. 2018. Back-translation-style data augmentation for end-to-end asr. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 426–433. IEEE.

Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Yu Zhang, and Xu Tan. 2020. Espnet-tts: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7654–7658. IEEE.

Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. 2016. End-to-end text-dependent speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5115–5119. IEEE.

Takaaki Hori, Ramon Astudillo, Tomoki Hayashi, Yu Zhang, Shinji Watanabe, and Jonathan Le Roux. 2019. Cycle-consistency training for end-to-end speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6271–6275. IEEE.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *arXiv preprint arXiv:2106.07447*.

Wei-Ning Hsu, David Harwath, Christopher Song, and James Glass. 2020. Text-free image-to-speech synthesis using learned segmental units. *arXiv preprint arXiv:2012.15454*.

Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, et al. 2018. Hierarchical generative modeling for controllable speech synthesis. *arXiv preprint arXiv:1810.07217*.

Andrew J Hunt and Alan W Black. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset.

Ye Jia, Ron J Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. 2019. Direct speech-to-speech translation with a sequence-to-sequence model. *Proc. Interspeech 2019*, pages 1123–1127.

Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *arXiv preprint arXiv:1806.04558*.

Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain De Cheveigne. 1999. Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds. *Speech communication*, 27(3-4):187–207.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pages 17022–17033. Curran Associates, Inc.

Oleksii Kuchaiev, Boris Ginsburg, Igor Gitman, Vitaly Lavrukhin, Carl Case, and Paulius Micikevicius. 2018a. Openseq2seq: extensible toolkit for distributed and mixed precision training of sequence-to-sequence models. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 41–46.

Oleksii Kuchaiev, Boris Ginsburg, Igor Gitman, Vitaly Lavrukhin, Carl Case, and Paulius Micikevicius. 2018b. OpenSeq2Seq: Extensible toolkit for distributed and mixed precision training of sequence-to-sequence models. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 41–46, Melbourne, Australia. Association for Computational Linguistics.

Kushal Lakhotia, Evgeny Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Adelrahman Mohamed, et al. 2021. Generative spoken language modeling from raw audio. *arXiv preprint arXiv:2102.01192*.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6706–6713.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2020. Multilingual speech translation with efficient finetuning of pretrained models. *arXiv preprint arXiv:2010.12829*.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2017. Mixed precision training. *arXiv preprint arXiv:1710.03740*.

Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. 2016. World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884.

Tomohiro Nakatani et al. 2008. A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments. *Speech Communication*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Jongseok Park, Kyubyong & Kim. 2019. g2pe. https://github.com/Kyubyong/g2p.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. 2017. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv preprint arXiv:1710.07654*.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. Speech resynthesis from discrete disentangled self-supervised representations. *arXiv preprint arXiv:2104.00355*.

Adam Polyak, Lior Wolf, Yossi Adi, and Yaniv Taigman. 2020. Unsupervised cross-domain singing voice conversion. *arXiv preprint arXiv:2008.02830*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Flávio Ribeiro, Dinei Florêncio, Cha Zhang, and Michael Seltzer. 2011. Crowdmos: An approach for crowdsourcing mean opinion score studies. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2416–2419. IEEE.

Andrew Rosenberg, Yu Zhang, Bhuvana Ramabhadran, Ye Jia, Pedro Moreno, Yonghui Wu, and Zelin Wu. 2019. Speech recognition with augmented synthesized speech. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 996–1002. IEEE.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE.

RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. 2018. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR.

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. 2021. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2017. Listening while speaking: Speech chain by deep learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 301–308. IEEE.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2020. Speech-to-speech translation without text.

Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. 2017. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit.

Changhan Wang, Anirudh Jain, Danlu Chen, and Jiatao Gu. 2019. VizSeq: a visual analysis toolkit for text generation tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 253–258, Hong Kong, China. Association for Computational Linguistics.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. Fairseq s2t: Fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 33–39.

Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. 2018. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pages 5180–5189. PMLR.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. 2018. Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. *arXiv preprint arXiv:1703.08581*.

Ron J Weiss, RJ Skerry-Ryan, Eric Battenberg, Soroosh Mariooryad, and Diederik P Kingma. 2021. Wave-tacotron: Spectrogram-free end-to-end text-to-speech synthesis. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5679–5683. IEEE.

John Wiseman. 2016. Python interface to the webrtc voice activity detector. https://github.com/wiseman/py-webrtcvad.

Zhizheng Wu, Oliver Watts, and Simon King. 2016. Merlin: An open source neural network speech synthesis system. In *SSW*, pages 202–207.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. 2002. The htk book. *Cambridge university engineering department*, 3(175):12.

Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W Black, and Keiichi Tokuda. 2007. The hmm-based speech synthesis system (hts) version 2.0. In *SSW*, pages 294–299. Citeseer.

Heiga Zen, Keiichi Tokuda, and Alan W Black. 2009. Statistical parametric speech synthesis. *speech communication*, 51(11):1039–1064.

Chen Zhang, Yi Ren, Xu Tan, Jinglin Liu, Kejun Zhang, Tao Qin, Sheng Zhao, and Tie-Yan Liu. 2021. Denoispeech: Denoising text to speech with frame-level noise modeling. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7063–7067. IEEE.