Parrotron: An End-to-End Speech-to-Speech Conversion Model and its Applications to Hearing-Impaired Speech and Speech Separation

Fadi Biadsy, Ron J. Weiss, Pedro J. Moreno, Dimitri Kanvesky, Ye Jia
Google
{biadsy, ronw, pedro, dkanevsky, jiaye}@google.com

Abstract

We describe Parrotron, an end-to-end-trained speech-to-speech conversion model that maps an input spectrogram directly to another spectrogram, without utilizing any intermediate discrete representation. The network is composed of an encoder, spectrogram and phoneme decoders, followed by a vocoder to synthesize a time-domain waveform. We demonstrate that this model can be trained to normalize speech from any speaker regardless of accent, prosody, and background noise, into the voice of a single canonical target speaker with a fixed accent and consistent articulation and prosody. We further show that this normalization model can be adapted to normalize highly atypical speech from a deaf speaker, resulting in significant improvements in intelligibility and naturalness, measured via a speech recognizer and listening tests. Finally, demonstrating the utility of this model on other speech tasks, we show that the same model architecture can be trained to perform a speech separation task.

Index Terms: speech normalization, voice conversion, atypical speech, speech synthesis, sequence-to-sequence model

1. Introduction

Encoder-decoder models with attention have recently shown considerable success in modeling a variety of complex sequence-to-sequence problems. These models have been successfully adopted to tackle a diverse set of tasks in speech and natural language processing, such as machine translation [1], speech recognition [2], and even combined speech translation [3]. They have also achieved state-of-the-art results in end-to-end Text-To-Speech (TTS) synthesis [4] and Automatic Speech Recognition (ASR) [5], using a single neural network that directly generates the target sequences, given virtually raw inputs.

In this paper, we combine attention-based speech recognition and synthesis models to build a direct end-to-end speech-to-speech sequence transducer. This model generates a speech spectrogram as a function of a different input spectrogram, with no intermediate discrete representation.

We test whether such a unified model is powerful enough to normalize arbitrary speech from multiple accents, imperfections, potentially including background noise, and generate the same content in the voice of a single predefined target speaker. The task is to project away all non-linguistic information, including speaker characteristics, and to retain only what is being said, not who, where or how it is said. This amounts to a text-independent, many-to-one voice conversion task [6]. We evaluate the model on this voice normalization task using ASR and listening studies, verifying that it is able to preserve the underlying speech content and project away other information, as intended.

We demonstrated that the pretrained normalization model can be adapted to perform a more challenging task of converting highly atypical speech from a deaf speaker into fluent speech, significantly improving intelligibility and naturalness. Finally, we evaluate whether the same network is capable of performing a speech separation task. Readers are encouraged to listen to sound examples on the companion website.

A variety of techniques have been proposed for voice conversion, including mapping code books [7], neural networks [8,9], dynamic frequency warping [10], and Gaussian mixture models [11,12]. Some recent work has also addressed accent conversion [14,15].

In this paper, we propose an end-to-end architecture that directly generates the target signal, synthesizing it from scratch, unlike filtering-based approaches. It is most similar to recent work on sequence-to-sequence voice conversion such as [16,17,18]. [10] uses a similar end-to-end model, conditioned on speaker identities, to transform word segments from multiple speakers into multiple target voices. Unlike [17], which trained separate models for each source-target speaker pair, we focus on many-to-one conversion. Our model is trained on source-target spectrogram pairs, without augmenting inputs with bottleneck features from a pretrained speech recognizer to more explicitly capture phonemic information in the source speech [17]. However, we do find it helpful to multilabel train the model to predict source speech phonemes. Finally, in contrast to [18], we are able to train the model without auxiliary alignment nor auto-encoding losses.

Similar voice conversion techniques have also been applied to improving intelligibility for speakers with vocal disabilities [19,20], and hearing-impaired speakers in particular [21]. We apply more modern machine learning techniques to this problem, and demonstrate that, given sufficient training data, an end-to-end trained one-to-one conversion model can dramatically improve intelligibility and naturalness of a deaf speaker.

2. Model Architecture

We use an end-to-end sequence-to-sequence model architecture that takes an input source speech and generates/synthesizes target speech as output. The only training requirement of such a model is a parallel corpus of paired input-output speech utterances. We refer to this speech-to-speech model as Parrotron.

As shown in Figure 1, the network is composed of an encoder and a decoder with attention, followed by a vocoder to synthesize a time-domain waveform. The encoder converts a sequence of acoustic frames into a hidden feature representation which the decoder consumes to predict a spectrogram. The core architecture is based on recent attention-based end-to-end ASR models [22] and TTS models such as Tacotron [23].

2.1. Spectrogram encoder

The base encoder configuration is similar to the encoder in [24], and some variations are evaluated in Section 3.1. From the input

[https://google.github.io/tacotron/publications/parrotron]}
We accomplish this by adding an auxiliary ASR decoder to pre-
with the same framing as the input features, and a 2048-point
The decoder targets are 1025-dim STFT magnitudes, computed
followed by batch normalization and tanh activation.
of emitting each grapheme in the output vocabulary.
512-dim attention context. This is passed into a 256 unit LSTM
toward a representation of the same underlying speech content.
Since the goal of this work is to only generate speech sounds
2.3. Multitask training with an ASR decoder
Since the goal of this work is to only generate speech sounds
and not arbitrary audio, jointly training the encoder network to
simultaneously learn a high level representation of the underly-
ing language serves to bias the spectrogram decoder predictions
toward a representation of the same underlying speech content.
We accomplish this by adding an auxiliary ASR decoder to pre-
dict the (grapheme or phoneme) transcript of the output speech,
conditioned on the encoder latent representation. Such a mul-
titask trained encoder can be thought of as learning a latent
representation of the input that maintains information about the
underlying transcript, i.e. one that is closer to the latent represen-
tation of the original input signal after normalization, we report
3. Applications
3.1. Voice normalization
We address the task of normalizing speech from an arbitrary
speaker to the voice of a predefined canonical speaker. As dis-
cussed in Section 2, to make use of Parrotron, we require a
parallel corpus of utterances spanning a variety of speakers and
recording conditions, each mapped to speech from a canonical
speaker. Since it is impractical to have single speaker record
many hours of speech in clean acoustic environment, we use
Google’s Parallel WaveNet-based TTS [31] system to generate
training targets from a large hand-transcribed speech corpus. Es-
sentially this reduces the task to reproducing any input speech in
the voice of a single-speaker TTS system. Using TTS to generate
this parallel corpus ensures that: (1) the target is always spoken
with a consistent predefined speaker and accent; (2) without any
background noise or disfluencies. (3) Finally, we can synthesize
as much data as necessary to scale to very large corpora.
3.1.1. Experiments
We train the model on a ~30,000 hour training set consisting
of about 24 million English utterances which are anonymized
and manually transcribed, and are representative of Google’s US
English voice search traffic. Using this corpus, we run a TTS
system to generate target utterances in a synthetic female voice.
To evaluate whether Parrotron preserves the linguistic con-
tent of the original input signal after normalization, we report
word error rates (WERs) using a state-of-the-art ASR engine on
the Parrotron output as a measure of speech intelligibility. Note
that the ASR engine is not trained on synthesized speech, a do-
main mismatch leading to higher WER. Table 1 compares differ-
ent model architectures and loss configurations. We run all mod-
els on a hand-transcribed held-out test set of 10K anonymized
utterances sampled from the same distribution as the train set.
The WER on the original speech (i.e., matching condition)
on this set is 8.3%, which can be viewed as our upper bound.
The top row of Table 1 shows performance using the base model architecture described in Section 2 using a spectrogram decoder employing additive attention [1] without an auxiliary ASR loss. Adding a parallel decoder to predict graphemes leads to a significant improvement, reducing the WER from 27.1% to 19.9%. Extending the additive attention with a location sensitive term [32] further improves results. This improves outputs on long utterances where additive attention sometimes failed.

Since orthography in English does not uniquely predict pronunciation, we hypothesize that using phoneme targets for the ASR decoder (obtained from forced alignment to the reference transcript) may reduce noise propagated back to the encoder. Indeed we find that this also shows consistent improvements.

Turning our attention to the encoder architecture, we found that reducing the number of parameters by removing the CLSTM significantly hurts performance. However, using 2 extra BLSTM layers instead of the CLSTM slightly improves results, while simultaneously simplifying the model. Hyperparameter tuning revealed that simply using slower learning rate decay (ending at 0.16 instead of 0.0) on our best model yields 17.6% WER.

Figure 1 for an example model output.

Table 2: Performance of Parrotron models on real speech.

| Model                  | MOS  | WER  |
|------------------------|------|------|
| Real speech            | 4.04 ± 0.19 | 34.2 |
| Parrotron (female)     | 3.81 ± 0.16 | 39.8 |
| Parrotron (male)       | 3.77 ± 0.16 | 37.5 |

Table 3: Subjective evaluation of Parrotron output quality.

| Survey question                              | Avg. score / agreement |
|----------------------------------------------|------------------------|
| How similar is the Parrotron voice to the    |                        |
| TTS voice on the 5 point Likert scale?       | 4.6                    |
| Does the output speech                       |                        |
| use a standard American English accent?      | 94.4%                  |
| contain any background noise?                | 0.0%                   |
| contain any disfluencies?                    | 0.0%                   |
| use consistent articulation, standard intonation and prosody? | 83.3% |

Synthesizing the reference transcripts and transcribing them using our ASR engine obtains a WER of 7.4%.

We analyze the types of phoneme errors Parrotron makes after normalization. We first obtain the true phonemes by force aligning each manual transcript with the corresponding real speech signal. Using this alignment, we compute two confusion matrices on the test set: (A) one computed by aligning the true phonemes with the hypothesized phonemes from the original speech, i.e. the Parrotron input; (B) another computed by aligning the true phonemes to the hypothesized phonemes from the normalized speech. We subtract A from B and rank the phoneme confusions to identify confusions which occur more frequently in Parrotron output than in real speech. Since we have 40 phonemes (+ epsilon), we have 1681 phoneme confusion pairs. In the top 5% of confusions, we observe that 26% of them are plosives (\(h/\), \(n/\), \(d/\), \(g/\), and \(b/\)) which are mostly dropped. The average rank of plosive confusions is 244/1681, suggesting that the model does not accurately model these short phonemes. We also observe another 12% correspond to vowel exchanges. This is not surprising since the model attempts to normalize multiple accents to that of the target TTS speaker.

Errors in plosive and other short phonemes are not surprising since the model uses an L2 reconstruction loss. Under this loss, a frame containing a vowel contributes the same amount as a frame containing \(h/\). Since there are significantly more vowel frames than plosives in the training data, this echoes the original speech content with an American accent, in the target voice. Such behavior is qualitatively different from what one would obtain by simply running an ASR followed by a TTS for example. A careful listening study is needed to further validate these results.

3.1.2. Error analysis

We focus on one case study of a profoundly deaf subject who was born in Russia to normal-hearing parents, and learned English as a teenager. The subject used Russian phonetic representation of English words and learned to speak them using Russian letters (e.g., cat \(\rightarrow\) k a T). Using a live (human in the loop) transcription service and ASR systems for multiple years helped improve their articulation. See [33] for more details.

We experiment with adapting the best model from Section 3.1 using a dataset of 15.4 hours of speech, corresponding to reading movie quotes. We use 90% of the data for adaptation (KADPT), and hold out the remainder: 5% (about 45 minutes) for dev and 5% for test (KTEST). This data was challenging; we learned that some prompts were difficult to pronounce by

Table 4: Performance on speech from a deaf speaker.

| Model                  | MOS  | WER  |
|------------------------|------|------|
| Real speech            | 2.08 ± 0.22 | 89.2 |
| Parrotron (male)       | 2.58 ± 0.20 | 109.3|
| Parrotron (male) finetuned | 3.52 ± 0.14 | 32.7 |

3.2. Normalization of hearing-impaired speech

Addressing a more challenging accessibility application, we investigate whether the normalization model can be used to convert atypical speech from a deaf speaker into fluent speech. This could be used to improve the vocal communication of people with such conditions or other speech disorders, or as a front-end to voice-enabled systems.

We observe that feeding Arabic and Spanish utterances into the US-English Parrotron model often results in output which echoes the original speech content with an American accent, in the target voice. Such behavior is qualitatively different from what one would obtain by simply running an ASR followed by a TTS for example. A careful listening study is needed to further validate these results.

Table 5: Performance on speech from a deaf speaker.

| Model                  | MOS  | WER  |
|------------------------|------|------|
| Real speech            | 2.08 ± 0.22 | 89.2 |
| Parrotron (male)       | 2.58 ± 0.20 | 109.3|
| Parrotron (male) finetuned | 3.52 ± 0.14 | 32.7 |

3.3. Normalization of hearing-impaired speech

We analyze the types of phoneme errors Parrotron makes after normalization. We first obtain the true phonemes by force aligning each manual transcript with the corresponding real speech signal. Using this alignment, we compute two confusion matrices on the test set: (A) one computed by aligning the true phonemes with the hypothesized phonemes from the original speech, i.e. the Parrotron input; (B) another computed by aligning the true phonemes to the hypothesized phonemes from the normalized speech. We subtract A from B and rank the phoneme confusions to identify confusions which occur more frequently in Parrotron output than in real speech. Since we have 40 phonemes (+ epsilon), we have 1681 phoneme confusion pairs. In the top 5% of confusions, we observe that 26% of them are plosives (\(h/\), \(n/\), \(d/\), \(g/\), and \(b/\)) which are mostly dropped. The average rank of plosive confusions is 244/1681, suggesting that the model does not accurately model these short phonemes. We also observe another 12% correspond to vowel exchanges. This is not surprising since the model attempts to normalize multiple accents to that of the target TTS speaker.

Errors in plosive and other short phonemes are not surprising since the model uses an L2 reconstruction loss. Under this loss, a frame containing a vowel contributes the same amount as a frame containing \(h/\). Since there are significantly more vowel frames than plosives in the training data, this echoes the original speech content with an American accent, in the target voice. Such behavior is qualitatively different from what one would obtain by simply running an ASR followed by a TTS for example. A careful listening study is needed to further validate these results.

The top row of Table 1 shows performance using the base model architecture described in Section 2 using a spectrogram decoder employing additive attention [1] without an auxiliary ASR loss. Adding a parallel decoder to predict graphemes leads to a significant improvement, reducing the WER from 27.1% to 19.9%. Extending the additive attention with a location sensitive term [32] further improves results. This improves outputs on long utterances where additive attention sometimes failed.

Since orthography in English does not uniquely predict pronunciation, we hypothesize that using phoneme targets for the ASR decoder (obtained from forced alignment to the reference transcript) may reduce noise propagated back to the encoder. Indeed we find that this also shows consistent improvements.

Turning our attention to the encoder architecture, we found that reducing the number of parameters by removing the CLSTM significantly hurts performance. However, using 2 extra BLSTM layers instead of the CLSTM slightly improves results, while simultaneously simplifying the model. Hyperparameter tuning revealed that simply using slower learning rate decay (ending at 0.16 instead of 0.0) on our best model yields 17.6% WER.

See Figure 1 for an example model output.

Using the best-performing Parrotron model, we conducted listening studies on a more challenging test set, which contains heavily accented speech plus background noise. As shown in Table 2, we verify that under these conditions Parrotron still preserves the linguistic content, since its WER is comparable to that of real speech. The naturalness MOS score decreases slightly with Parrotron when compared to that of real speech. Recall that the objective in this work is to perform many-to-one translation, with such conditions or other speech disorders, or as a front-end to voice-enabled systems.

We observe that feeding Arabic and Spanish utterances into the US-English Parrotron model often results in output which echoes the original speech content with an American accent, in the target voice. Such behavior is qualitatively different from what one would obtain by simply running an ASR followed by a TTS for example. A careful listening study is needed to further validate these results.

We focus on one case study of a profoundly deaf subject who was born in Russia to normal-hearing parents, and learned English as a teenager. The subject used Russian phonetic representation of English words and learned to speak them using Russian letters (e.g., cat \(\rightarrow\) k a T). Using a live (human in the loop) transcription service and ASR systems for multiple years helped improve their articulation. See [33] for more details.

We experiment with adapting the best model from Section 3.1 using a dataset of 15.4 hours of speech, corresponding to reading movie quotes. We use 90% of the data for adaptation (KADPT), and hold out the remainder: 5% (about 45 minutes) for dev and 5% for test (KTEST). This data was challenging; we learned that some prompts were difficult to pronounce by

Table 4: Performance on speech from a deaf speaker.

| Model                  | MOS  | WER  |
|------------------------|------|------|
| Real speech            | 2.08 ± 0.22 | 89.2 |
| Parrotron (male)       | 2.58 ± 0.20 | 109.3|
| Parrotron (male) finetuned | 3.52 ± 0.14 | 32.7 |
Experiments

Our first experiment is to test the performance of Google’s state-of-the-art ASR system on kTEST. As shown in Table 2, we find that the ASR system performs very poorly on this speech, obtaining 89.2% WER on the test set. The MOS score on kTEST is 2.08, rated by subjects unfamiliar with the subject’s speech.

We then test whether our best out-of-the-box Parrotron trained for the normalization task, shown in Section 3.1, can successfully normalize this type of speech. The only difference here is that Parrotron is trained on a male TTS speech, obtained from our production WaveNet-based TTS. Testing on kTEST, we find that the output of this model was rated as natural as the original speech, but our ASR engine performs even more poorly on the converted speech than the original speech. In other words, Parrotron normalization system trained on standard speech fails completely to normalize this type of speech. We have also manually inspected the output of this Parrotron and found that the model produces speech-like sounds but nonsense words.

Now, we test whether utilizing KADPT would have any impact on Parrotron performance. We first take the fully converged male Parrotron normalization model and conduct multiple fine-tuning experiments using KADPT. With a constant learning rate of 0.1, we (1) adapt all parameters on the fully converged model; (2) adapt all parameters except freezing the spectrogram decoder parameters; (3) freeze both spectrogram decoder and phoneme decoder parameters while fine-tuning only the encoder.

We find that all fine-tuning strategies lead to intelligible and significantly more natural speech. The best fine-tuning strategy was adapting all parameters, which increased the MOS naturalness score by over 1.4 points compared to the original speech, and dramatically reduced the WER from 89.2% to 32.7%. Fine-tuning strategy (2) obtains 34.1% WER and adapting only encoder parameters (strategy (3)), obtains 38.6% WER.

Note that one advantage of directly converting speech to speech over cascading a finetuned ASR engine with TTS is as follows. Synthesizing the output of an ASR engine may generate speech far from intended, due to unavoidable ASR errors. A speech-to-speech model, however, is likely to produce sounds closer to the original speech. We have seen significant evidence to support this hypothesis, but leave it to future work to quantify.

3.3. Speech separation

Finally, to illustrate that the Parrotron architecture can be used in a variety of speech applications, we evaluate it on a speech separation task of reconstructing the signal from the loudest speaker within a mixture of overlapping speech. We focus on instantaneous mixtures of up to 8 different speakers.

It is important to stress that our intent in this section is not to propose a state of the art separation system, but rather to demonstrate that the proposed architecture may apply to different speech applications. More importantly, in contrast to previous applications which made use of synthetic training targets, we evaluate whether Parrotron is able to generate speech from an open set of speakers, generalizing beyond the training set. Furthermore, unlike state-of-the-art speech separation techniques, Parrotron generates the signal from scratch as opposed to using a masking-based filtering approach and it able to rely on an implicit phoneme language model. We use the same voice-search data described in Section 3.1 to artificially construct instantaneous mixtures of speech signals. For each target utterance in the training data, we randomly select a set of 1 to 7 utterances to mix together as the background noise. The number of background utterances is also randomly selected. Before mixing, we normalize all utterances to have similar gains.

We mix target utterances with the background noise by simply averaging the two signals with a randomly sampled weight \( w \in [0.1, 0.5] \) for the background and \( 1 - w \) for the target utterance. This results in an average SNR across all artificially constructed utterances of 12.15 dB, with a standard deviation of 4.7. 188K utterances from this corpus are held out for testing. While we do not explicitly incorporate reverberation or non-speech noise, the underlying utterances come from a variety of recording environments with their own background noise.

To evaluate whether Parrotron can perform this separation task, we train a model to the best performing architecture as in Section 3.1. We feed as inputs our mixed utterances and train the model to generate corresponding original clean utterances.

We evaluate the impact of this separation model using Google’s ASR system. We compare WERs on three sets of 188K held-out utterances: (1) the original clean speech before adding background speech; (2) the noisy set after mixing background speech; (3) the cleaned output generated by running Parrotron on the noisy set. As shown in Table 5, we observe significant WER reduction after running Parrotron on the noisy set, demonstrating that the model can preserve speech from the target speaker and separate them from other speakers. Parrotron significantly reduces insertions, which correspond to words spoken by background speakers, but suffers from increased deletions, which is likely due to early end of utterance prediction.

4. Conclusion

We described Parrotron, an end-to-end speech-to-speech model that converts an input spectrogram directly to another spectrogram, without intermediate symbolic representation. We find that the model can be trained to normalize speech from different speakers into speech of a single target speaker’s voice while preserving the linguistic content and projecting away non-linguistic content. We then showed that this model can successfully be adapted to improve WER and naturalness of speech from a deaf speaker. We finally demonstrate that the same model can be trained to successfully identify, separate and reconstruct the loudest speaker in a mixture of overlapping speech, improving ASR performance. The Parrotron system has other potential applications, e.g. improving intelligibility by converting heavily accented or otherwise atypical speech into standard speech. In the future, we plan to test it on other speech disorders, and adopt techniques from [16,30] to preserve the speaker identity.

5. Acknowledgments

We thank Franoise Beaufays, Michael Brenner, Diamantino Casero, Zhifeng Chen, Mohamed Elhokky, Patrick Nguyen, Bhuvana Ramabhadran, Andrew Rosenberg, Jason Pelaezios, Johan Schalkwyk, Yonghui Wu, and Zelin Wu for useful feedback.
