Text-to-Speech (TTS) Synthesis

Anusha Prakash
PhD Scholar
Guides: Prof. S. Umesh, Prof. Hema A. Murthy

IIT Madras

Outline

- What is TTS?
- Brief history
- TTS frameworks
- Evaluation of TTS systems
- Potential research areas

Text

How are you

Characters

a, b, c ... z

अ आ इ ई

Speech

phonemes ⇒ basic units

↳ spoken language

phonemes { /iy/ → ~~ई~~ ई
/k/ क

ASR

table text

Text → Phonemes

Lexicon / Dictionary

आम	/ah/	/m/	
good	/g/	/uh/	/d/

Text

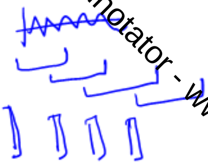
↓

$\boxed{\text{Lexicon}}$

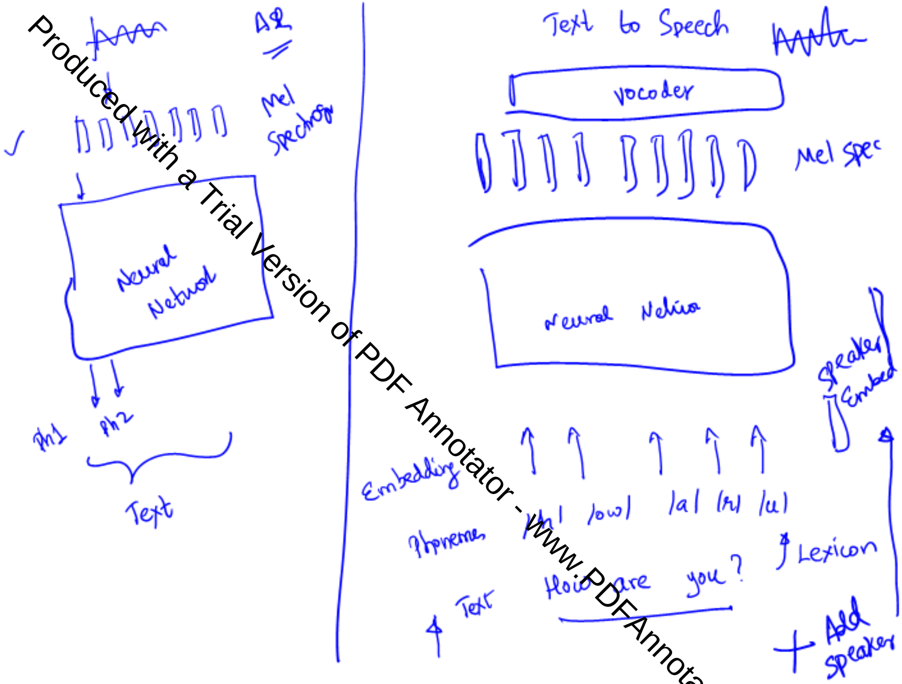
↓

sequence of phonemes.

ASR



Mel Filter Bank

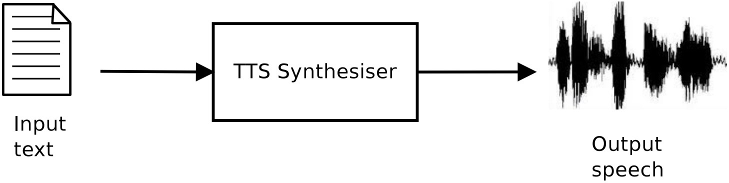

ASR
Mel
Spectrog
Neural
Network
Ph1
Ph2
Text
Text to Speech
vocoder
Mel spec
Neural
Speaker
Embed
Embedding
Phonemes
/h/ /ow/ /a/ /r/ /u/
Lexicon
Text
How are you?
Add
Speaker

Text-to-speech (TTS) synthesis

- Speech – important mode of communication

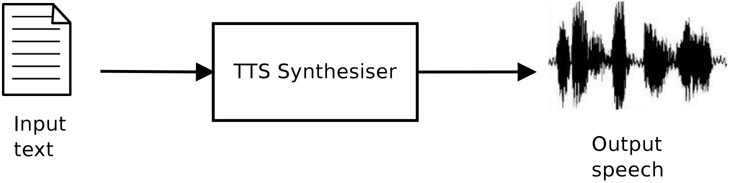
Text-to-speech (TTS) synthesis

- Speech – important mode of communication



Text-to-speech (TTS) synthesis

- Speech – important mode of communication

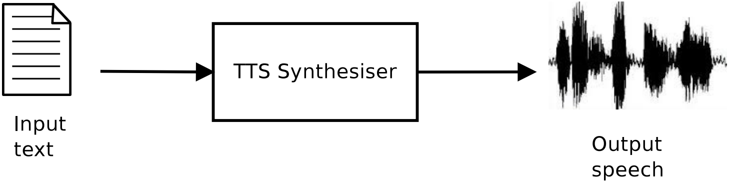


TTS system

- Automatically generate speech corresponding to given text

Text-to-speech (TTS) synthesis

- Speech – important mode of communication



TTS system

- Automatically generate speech corresponding to given text
- Synthesised audio – natural and intelligible speech

Applications

- Speech based technologies
- Helps people with literacy difficulties
- Aid the visually challenged

History

- Late 18^{th} century: mechanical models of the vocal tract, acoustic-mechanical speech machine

[1] www.youtube.com/watch?v=0rAyrmm7vv0

[2] www.nytimes.com

[3] www.youtube.com/watch?v=gSOtG9zrUVE&t=275s

History

- Late 18^{th} century: mechanical models of the vocal tract, acoustic-mechanical speech machine
- 1930s: Vocoder, Voder (Bell labs) Keyboard operated speech synthesiser [1]

[1] www.youtube.com/watch?v=0rAyrmm7vv0

[2] www.nytimes.com

[3] www.youtube.com/watch?v=gSOtG9zrUVE&t=275s

History

- Late 18^{th} century: mechanical models of the vocal tract, acoustic-mechanical speech machine
- 1930s: Vocoder, Voder (Bell labs) Keyboard operated speech synthesiser [1]
- 1961: computer generated voice (Bell Labs)– song "Daisy Bell"

[1] www.youtube.com/watch?v=0rAyrmm7vv0

[2] www.nytimes.com

[3] www.youtube.com/watch?v=gSOtG9zrUVE&t=275s

History

- Late 18^{th} century: mechanical models of the vocal tract, acoustic-mechanical speech machine
- 1930s: Vocoder, Voder (Bell labs) Keyboard operated speech synthesiser [1]
- 1961: computer generated voice (Bell Labs)– song "Daisy Bell"
- 1968: first general English TTS (Electrotechnical Laboratory, Japan)

[1] www.youtube.com/watch?v=0rAyrmm7vv0

[2] www.nytimes.com

[3] www.youtube.com/watch?v=gSOtG9zrUVE&t=275s

History

- Late 18^{th} century: mechanical models of the vocal tract, acoustic-mechanical speech machine
- 1930s: Vocoder, Voder (Bell labs) Keyboard operated speech synthesiser [1]
- 1961: computer generated voice (Bell Labs)– song "Daisy Bell"
- 1968: first general English TTS (Electrotechnical Laboratory, Japan)
- Linear prediction coding (LPC), formant synthesis, articulatory synthesis

[1] www.youtube.com/watch?v=0rAyrmm7vv0

[2] www.nytimes.com

[3] www.youtube.com/watch?v=gSOtG9zrUVE&t=275s

History

- Late 18^{th} century: mechanical models of the vocal tract, acoustic-mechanical speech machine
- 1930s: Vocoder, Voder (Bell labs) Keyboard operated speech synthesiser [1]
- 1961: computer generated voice (Bell Labs)– song "Daisy Bell"
- 1968: first general English TTS (Electrotechnical Laboratory, Japan)
- Linear prediction coding (LPC), formant synthesis, articulatory synthesis
- 1990s: first successful female voice synthesiser (AT&T Bell Laboratories) Audio clip [2]

[1] www.youtube.com/watch?v=0rAyrmm7vv0

[2] www.nytimes.com

[3] www.youtube.com/watch?v=gSOtG9zrUVE&t=275s

History

- Late 18^{th} century: mechanical models of the vocal tract, acoustic-mechanical speech machine
- 1930s: Vocoder, Voder (Bell labs) Keyboard operated speech synthesiser [1]
- 1961: computer generated voice (Bell Labs)– song "Daisy Bell"
- 1968: first general English TTS (Electrotechnical Laboratory, Japan)
- Linear prediction coding (LPC), formant synthesis, articulatory synthesis
- 1990s: first successful female voice synthesiser (AT&T Bell Laboratories) Audio clip [2]
- Concatenative speech synthesis

---

[1] www.youtube.com/watch?v=0rAyrmm7vv0

[2] www.nytimes.com

[3] www.youtube.com/watch?v=gS0tG9zrUVE&t=275s

History

- Late 18^{th} century: mechanical models of the vocal tract, acoustic-mechanical speech machine
- 1930s: Vocoder, Voder (Bell labs) Keyboard operated speech synthesiser [1]
- 1961: computer generated voice (Bell Labs)– song "Daisy Bell"
- 1968: first general English TTS (Electrotechnical Laboratory, Japan)
- Linear prediction coding (LPC), formant synthesis, articulatory synthesis
- 1990s: first successful female voice synthesiser (AT&T Bell Laboratories) Audio clip [2]
- Concatenative speech synthesis
- Railway announcement [3]

[1] www.youtube.com/watch?v=0rAyrmm7vv0

[2] www.nytimes.com

[3] www.youtube.com/watch?v=gS0tG9zrUVE&t=275s

History

- Late 18^{th} century: mechanical models of the vocal tract, acoustic-mechanical speech machine
- 1930s: Vocoder, Voder (Bell labs) Keyboard operated speech synthesiser [1]
- 1961: computer generated voice (Bell Labs)– song "Daisy Bell"
- 1968: first general English TTS (Electrotechnical Laboratory, Japan)
- Linear prediction coding (LPC), formant synthesis, articulatory synthesis
- 1990s: first successful female voice synthesiser (AT&T Bell Laboratories) Audio clip [2]
- Concatenative speech synthesis
- Railway announcement [3] – restricted domain synthesis

[1] www.youtube.com/watch?v=0rAyrmm7vv0

[2] www.nytimes.com

[3] www.youtube.com/watch?v=gS0tG9zrUVE&t=275s

TTS systems

Frameworks:

1. Unit selection synthesis (USS)
2. Hidden Markov model based (HTS)
3. Neural network based (conventional)
4. End-to-end (E2E)

TTS systems

Frameworks:

1. Unit selection synthesis (USS)
2. Hidden Markov model based (HTS)
3. Neural network based (conventional)
4. End-to-end (E2E)

Phases:

1. Training phase
2. Synthesis/testing phase

TTS systems

Frameworks:

1. Unit selection synthesis (USS)
2. Hidden Markov model based (HTS)
3. Neural network based (conventional)
4. End-to-end (E2E)

Phases:

1. Training phase
2. Synthesis/testing phase

Training data/ Speech database

<text, audio> pairs – continuous speech

1. Unit selection synthesis (USS)[4]

- Select and concatenate units from large speech database
- Choice of units (C: consonant, V: vowel):
    - Phone (C or V)
    - Akshara (C^*V)
    - Diphone (two adjacent half-phones – captures transition)
    - Syllable (C^*VC^*)
    - Word

[4] Andrew J. Hunt and Alan W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database", ICASSP, 1996, pp. 373-376.

Unit selection synthesis (USS)

- Idea: select and concatenate units from large speech database
- Choice of units (C: consonant, V: vowel) [Example: How are you]
  - Phone (C or V) [h, a, w, aa, r, y, uu (7)]
  - Akshara (C^*V) [ha, w, aa, r, yuu (3 aksharas + 2 phones)]
  - Diphone [h-a, a-w, w-aa, aa-r, r-y, y-uu (6)]
  - Syllable (C^*VC^*) [haw, aar, yuu (3)]
  - Word [how, are, you (3)]

Andrew J. Hunt and Alan W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database", ICASSP, 1996, pp. 373-376.

Unit selection synthesis (USS)

- Training:
  - Splice the speech database at the unit level
  - Database should cover multiple realisations of units in different contexts
  - Fallback mechanism: Words $\rightarrow$ Syllables $\rightarrow$ Aksharas/Diphones $\rightarrow$ Phones
- Synthesis
  - Appropriate units selected and concatenated
  - Units chosen to minimise target and concatenation costs
- Synthesised examples: English USS Hindi USS

Unit selection synthesis (USS)

Advantage

- Natural sounding speech

Disadvantages

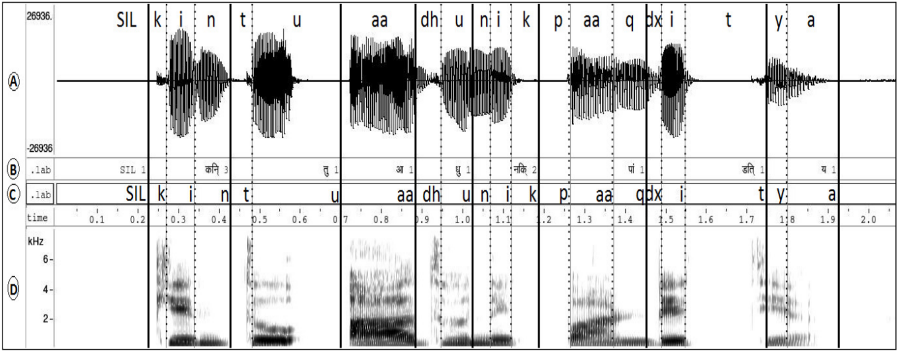
- Requires very large database
- Discontinuities perceived at concatenation points

TTS system: modules

- Lexicon/ LTS/ Grapheme-to-Phoneme (G2P):
  AGRICULTURE $\rightarrow$ AE G R IH K AH L CH ER (CMU representation)

TTS system: modules

- Lexicon/ LTS/ Grapheme-to-Phoneme (G2P):
  AGRICULTURE → AE G R IH K AH L CH ER (CMU representation)
- Alignment:



2. Hidden Markov model (HMM) based speech synthesis system (HTS)[5]

- Instead of storing actual waveform units, models of units stored
- Based on source-filter model of speech
- Extraction of features — source (log $f0$), system (MFCC)
- Statistical parametric modelling:
  - Parametric: speech is described using parameters
  - Statistical: parameters are described using statistics (mean, variance of probability density functions)

[5] H Zen, K Tokuda and A W Black, "Statistical parametric speech synthesis", Speech Communication, vol. 51, no. 3, pp. 1039-1064, November 2009.

HTS

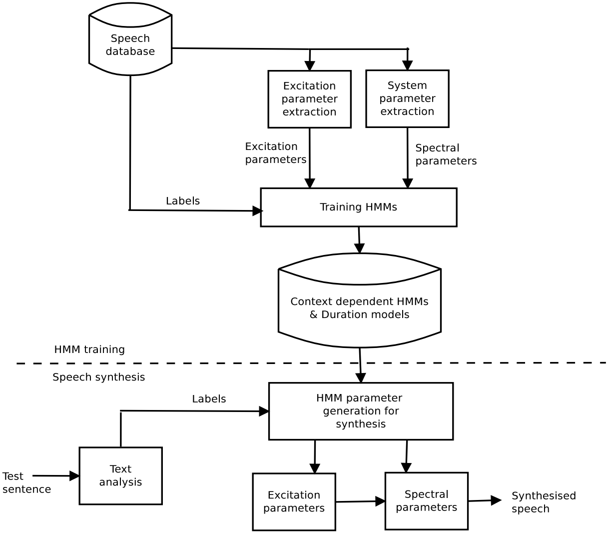


Figure: Training and synthesis phases of HTS

HTS

Training

- Feature extraction from aligned <text, speech> data
- Every phone is modeled by a 3-state HMM: Each state has a GMM

HTS

Training

- Feature extraction from aligned <text, speech> data
- Every phone is modeled by a 3-state HMM: Each state has a GMM

Synthesis

Text for synthesis
→ broken into constituent phones
→ corresponding HMMs selected and concatenated– sentence HMM
→ Generates acoustic features which is fed to a vocoder for synthesis

HTS

Training

- Feature extraction from aligned <text, speech> data
- Every phone is modeled by a 3-state HMM: Each state has a GMM

Synthesis

Text for synthesis
→ broken into constituent phones
→ corresponding HMMs selected and concatenated– sentence HMM
→ Generates acoustic features which is fed to a vocoder for synthesis

Models trained

1. Duration model: predicts how many frames to be assigned for every phone (*Remember the self-transition of HMMs*)
2. Acoustic model: predicts acoustic features for required number of frames

Context-dependent model

- Basic unit in HTS: context-dependent pentaphone
- Model for monophone in pentaphone context

Context-dependent model

- Basic unit in HTS: context-dependent pentaphone
- Model for monophone in pentaphone context
- Text: **It is a** lovely day

Monophones	Pentaphone context
i	x-x-**i**-tx-i
tx	x-i-**tx**-i-s
i	i-tx-**i**-s-a
s	tx-i-**s**-a-l
a	i-s-**a**-l-a

- Other contexts- position of current phoneme in current syllable, position of current syllable in current word, etc.

Context-dependent model

- Basic unit in HTS: context-dependent pentaphone
- Model for monophone in pentaphone context
- Text: **It is a** lovely day

Monophones	Pentaphone context
i	x-x-**i**-tx-i
tx	x-i-**tx**-i-s
i	i-tx-**i**-s-a
s	tx-i-**s**-a-l
a	i-s-**a**-l-a

- Other contexts- position of current phoneme in current syllable, position of current syllable in current word, etc.
- If a language has 50 phones
  - No. of monophone models: 50
  - No. of pentaphone-context models: 50^5
  - Including contexts: large number of combinations

Handling context

Challenges with handling context

- Some combinations not available in the training data
- Unseen combinations present in the test sentence

Handling context

Challenges with handling context

- Some combinations not available in the training data
- Unseen combinations present in the test sentence

Decision tree-based context clustering

- Binary tree- HMMs split into two sub-categories based on certain yes/no questions
- Cluster HMM states- model parameters shared among states in each leaf node
- Question set

HTS

- Synthesised examples: English HTS Hindi HTS Malayalam HTS

HTS

- Synthesised examples: English HTS Hindi HTS Malayalam HTS

Advantages

- Low amount of training data
- Small footprint size (few MBs)
- Fast synthesis
- Flexible – tune HMM parameters to vary speaking style, emotion

HTS

- Synthesised examples: English HTS Hindi HTS Malayalam HTS

Advantages

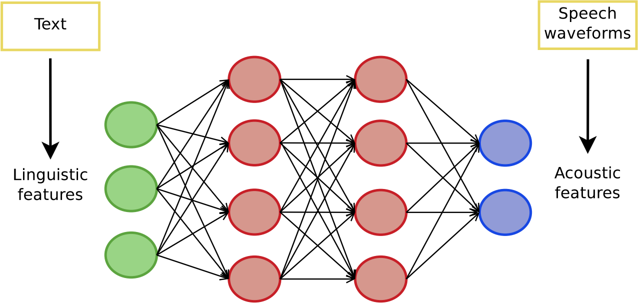
- Low amount of training data
- Small footprint size (few MBs)
- Fast synthesis
- Flexible – tune HMM parameters to vary speaking style, emotion

Disadvantage

- Voice quality relatively poor – "muffling" due to state-tying in the decision tree

3. Neural network based TTS (conventional)

- Learn the mapping between linguistic and acoustic feature vectors using neural network



Y. Qian, Y. Fan, W. Hu, and F. Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis", ICASSP, 2014.

Y. Fan, Y. Qian, and F. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks", Interspeech, 2014.

Neural network based TTS (conventional)

- Types:
  - Feed forward neural network (DNN)
  - Long short-term memory (LSTM) based recurrent neural network (RNN)
  - Bidirectional LSTM (BLSTM)
- Train duration and acoustic models
- Synthesis:
  - Text $\rightarrow$ linguistic features $\rightarrow$ generate acoustic features $\rightarrow$ vocoder for speech reconstruction

Neural network based TTS (conventional)

- Synthesised examples: English Hindi

Neural network based TTS (conventional)

- Synthesised examples: English Hindi

Advantages

- Better synthesis quality compared to HTS

Neural network based TTS (conventional)

- Synthesised examples: English Hindi

Advantages

- Better synthesis quality compared to HTS

Disadvantage

- Requires more training data to produce good quality speech

TTS: “Decompressing”

- Text: 20 phones
- Number of frames: 200

TTS: "Decompressing"

- Text: 20 phones
- Number of frames: 200
- Sampling rate: 16 KHz
- Frame shift: 10 msec (160 samples)

TTS: "Decompressing"

- Text: 20 phones
- Number of frames: 200
- Sampling rate: 16 KHz
- Frame shift: 10 msec (160 samples)
- No. of samples in the output: 32,000

Decompressing the text

20 phones $\rightarrow$ 32,000 values

TTS: "Decompressing"

- Text: 20 phones
- Number of frames: 200
- Sampling rate: 16 KHz
- Frame shift: 10 msec (160 samples)
- No. of samples in the output: 32,000

Decompressing the text

20 phones $\rightarrow$ 32,000 values

Modeling raw audio is challenging

WaveNet

- Raw waveform of audio directly modeled one sample at a time
- Predictive distribution for audio sample conditioned on previous samples
- Joint probability of waveform $\mathbf{x} = x_1, \ldots, x_T$:

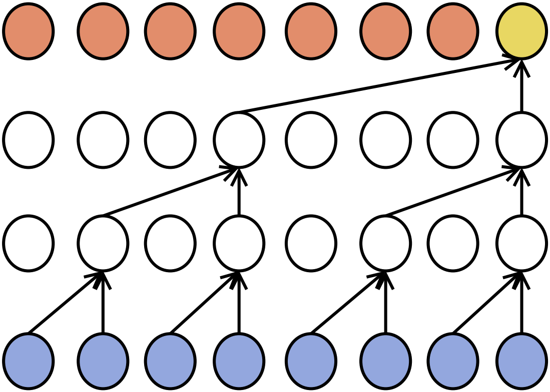
$$
\begin{aligned}
p(\mathbf{x}) &= \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1}) \\
&= p(x_t | x_1, \ldots, x_{t-1}) p(x_{t-1} | x_1, \ldots, x_{t-2}) \ldots p(x_2 | x_1) p(x_1)
\end{aligned}
$$

- Auto-regressive model

Aaron van den Oord et. al., "WaveNet: A Generative Model for Raw Audio", ISCA Speech Synthesis Workshop (SSW9), September 2016, USA.

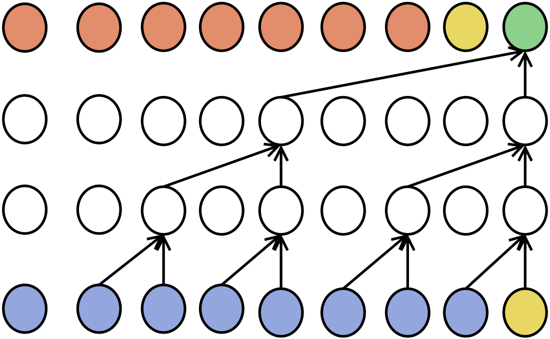
WaveNet

- Conditional probability modeled by stack of convolution layers



WaveNet

- Conditional probability modeled by stack of convolution layers



WaveNet

- Causal convolutions
- Training: predictions for all timesteps made in parallel $\rightarrow$ faster training due to no recurrent connections
- Synthesis: sequential prediction
- To capture more context:
    - $\uparrow$ number of layers
    - dilated convolution

Conditional WaveNet

- Network trained without text sequence: babbling Audio

Conditional WaveNet

- Network trained without text sequence: babbling Audio
- TTS – additionally condition on linguistic features $(\mathbf{h})$:

$$p(\mathbf{x}|\mathbf{h}) = \prod_{t=1}^{T} p(x_t|x_1, \ldots, x_{t-1}, \mathbf{h})$$

- Synthesised examples: HTS USS WaveNet

Conditional WaveNet

- Network trained without text sequence: babbling Audio
- TTS – additionally condition on linguistic features ($\mathbf{h}$):

$$p(\mathbf{x}|\mathbf{h}) = \prod_{t=1}^{T} p(x_t|x_1, \ldots, x_{t-1}, \mathbf{h})$$

- Synthesised examples: HTS USS WaveNet
- Also condition on speaker identity: Speaker 1 Speaker 2

Conditional WaveNet

- Network trained without text sequence: babbling Audio
- TTS – additionally condition on linguistic features ($\mathbf{h}$):

$$p(\mathbf{x}|\mathbf{h}) = \prod_{t=1}^{T} p(x_t|x_1, \ldots, x_{t-1}, \mathbf{h})$$

- Synthesised examples: HTS USS WaveNet
- Also condition on speaker identity: Speaker 1 Speaker 2

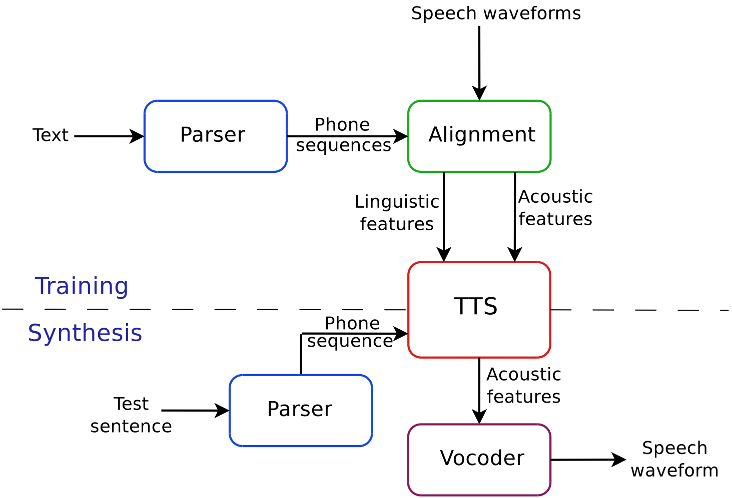
Advantages

- Produce good quality speech

Disadvantages

- Require a lot of data to produce good quality speech
- Computationally intensive

TTS system: modules

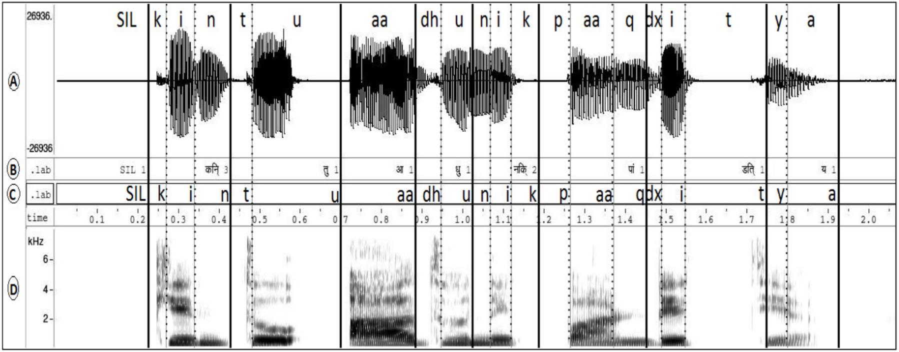


TTS system: modules

- Lexicon/ LTS/ Grapheme-to-Phoneme (G2P):
  AGRICULTURE $\rightarrow$ AE G R IH K AH L CH ER (CMU representation)

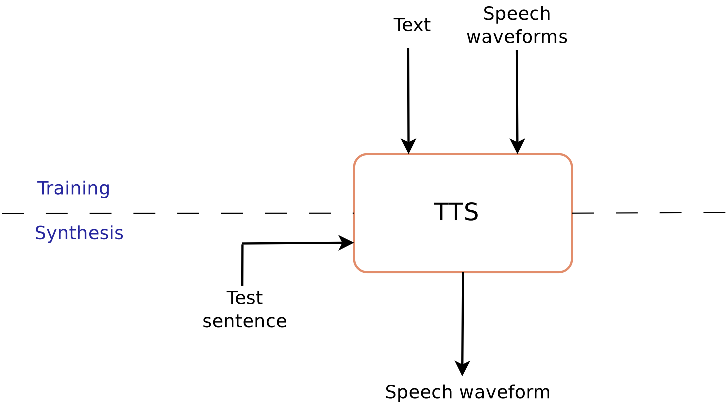
TTS system: modules

- Lexicon/ LTS/ Grapheme-to-Phoneme (G2P):
  AGRICULTURE → AE G R IH K AH L CH ER (CMU representation)
- Alignment:



Towards end-to-end speech synthesis

- Speech directly synthesised from characters



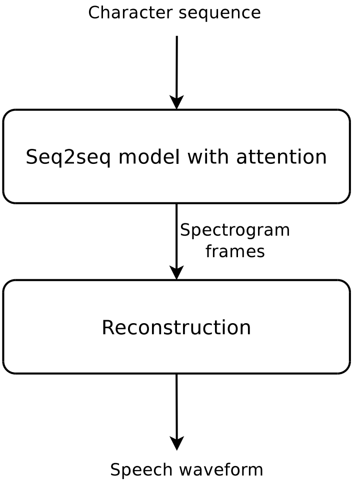
End-to-end speech synthesis

- No need of developing separate modules (parsing, alignment)
- Allows rich conditioning on speaker, language, high-level features
- Single model likely more robust than multi-stage model

Types of neural speech synthesisers that will be discussed

1. Tacotron2
2. Fastspeech
3. Fastspeech2

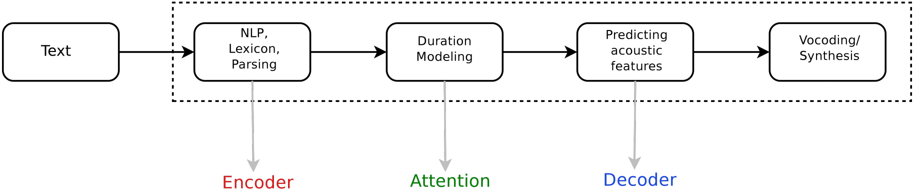
Tacotron2



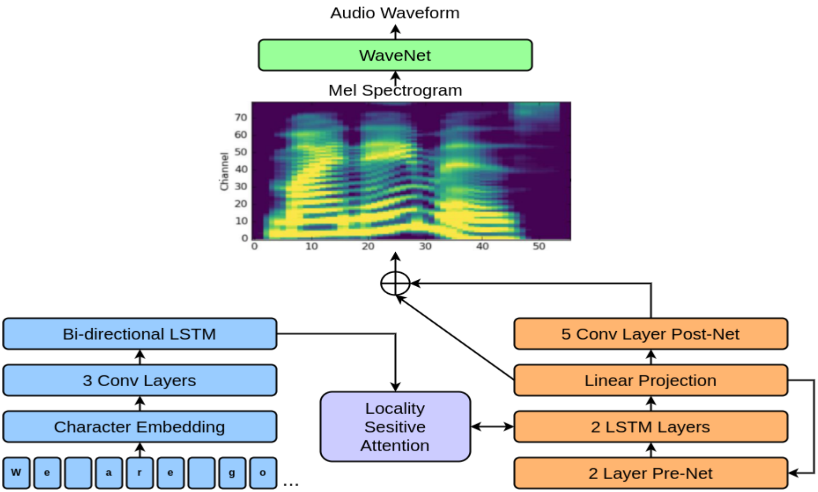
1. Encoder and attention-based decoder
2. Vocoder

J. Shen et al., "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions", ICASSP, 2018, pp. 4779–4783.

Comparison with traditional systems



Tacotron2



Audio Waveform
WaveNet
Mel Spectrogram
Channel
70
60
50
40
30
20
10
0
0
10
20
30
40
50
Bi-directional LSTM
3 Conv Layers
Character Embedding
w
e
a
r
e
g
o
...
Locality
Sesitive
Attention
5 Conv Layer Post-Net
Linear Projection
2 LSTM Layers
2 Layer Pre-Net

Tacotron2: encoder-decoder architecture

Encoder

- Encodes the input text
- Extract character embedding from each sentence (similar to word embeddings in word2vec)
- Convolution layers to capture context information
- BLSTM layer to capture sequence information

Tacotron2: encoder-decoder architecture

Encoder

- Encodes the input text
- Extract character embedding from each sentence (similar to word embeddings in word2vec)
- Convolution layers to capture context information
- BLSTM layer to capture sequence information

Attention

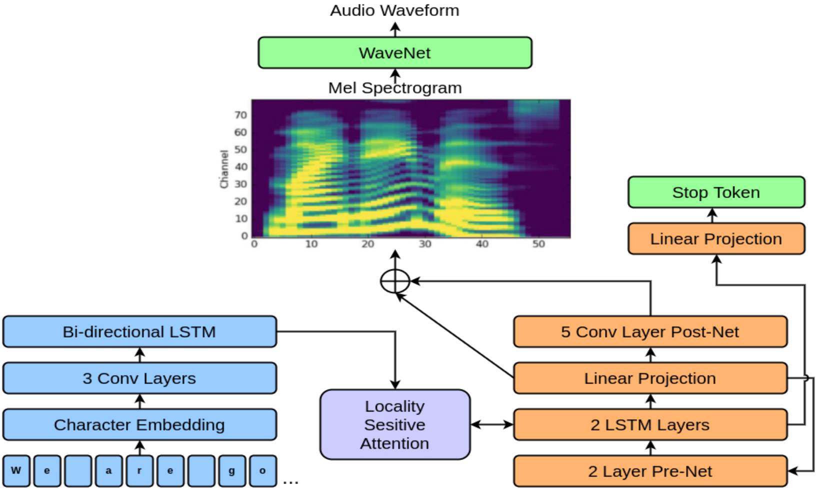
- Tells us which encoded features are more relevant at each time step
- Attention generates context vector at each decoder step

Tacotron2: encoder-decoder architecture

Decoder

- Auto-regressive RNN: Predicts mel-spectrogram frame(s) at each time step from encoded features
- Pre-net (2 fully connected layers) to learn attention
- Output of pre-net and context vector concatenated $\rightarrow$ passed through 2 LSTM layers
- LSTM output and context vector concatenated $\rightarrow$ projected through linear layer to predict target mel-spectrogram
- Linear layer: 80 neurons – dimensionality of mel-spectrogram
- Postnet for improved quality

Tacotron2: when to stop generating?



Tacotron2: "Stop token" prediction

- Tells decoder when to stop generating mel-spectrogram frames
- Decoder LSTM output and context vector concatenated $\rightarrow$ projected to scalar value $\rightarrow$ sigmoid activation function: probability that generation is completed
- Value exceeds set threshold $\rightarrow$ generation stops

Tacotron2: Synthesised examples

Sample 1

The quick brown fox jumps over the lazy dog. Audio

Sample 2

Peter Piper picked a peck of pickled peppers. How many pickled peppers did Peter Piper pick? Audio

Sample 3

To deliver interfaces that are significantly better suited to create and process RFC eight twenty one, RFC eight twenty two, RFC nine seventy seven, and MIME content. Audio

Issues with auto-regressive based network

- Synthesised speech is not robust– has repetitions and word skips (error propagation, wrong attention alignments between text and speech)
- Slow inference speed
- Lack of control over speed and prosody

Fastspeech

Non-robust speech generation

$\rightarrow$ provide hard alignments between phonemes and mel-spectrograms using a duration predictor

Ren et. al., "FastSpeech: Fast, Robust and Controllable Text to Speech", International Conference on Neural Information Processing Systems, 2019, pp. 3171–3180.

Fastspeech

Non-robust speech generation

→ provide hard alignments between phonemes and mel-spectrograms using a duration predictor

Slow inference speed

→ parallel mel-spectrogram generation using feed-forward transformer

Ren et. al., "FastSpeech: Fast, Robust and Controllable Text to Speech", International Conference on Neural Information Processing Systems, 2019, pp. 3171–3180.

Fastspeech

Non-robust speech generation

→ provide hard alignments between phonemes and mel-spectrograms using a duration predictor

Slow inference speed

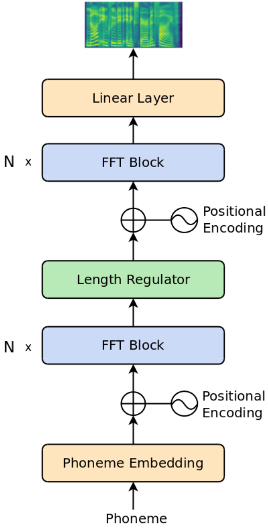
→ parallel mel-spectrogram generation using feed-forward transformer

Lack of control over speed and prosody

→ adjust voice speed by lengthening or shortening phoneme duration
→ add breaks between adjacent phonemes for better prosody

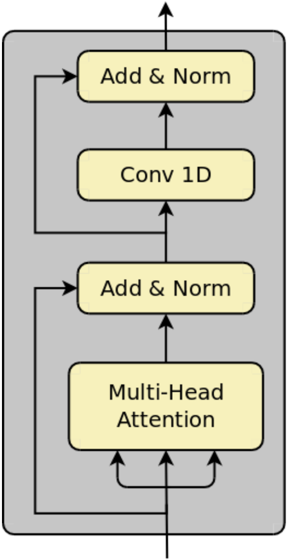
Ren et. al., "FastSpeech: Fast, Robust and Controllable Text to Speech", International Conference on Neural Information Processing Systems, 2019, pp. 3171–3180.

Fastspeech



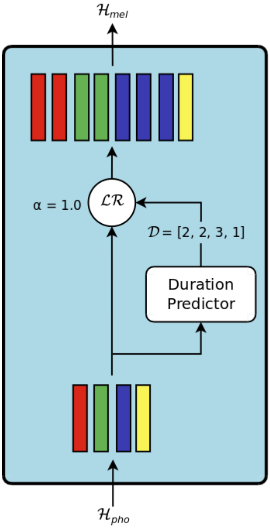
- Architecture based on Feed-forward Transformer (FFT)
- N-FFT blocks on input and output sides
- Length regulator: expands phoneme sequence to match number of mel-spectrogram frames

Fastspeech: Feed-forward transformer (FFT)



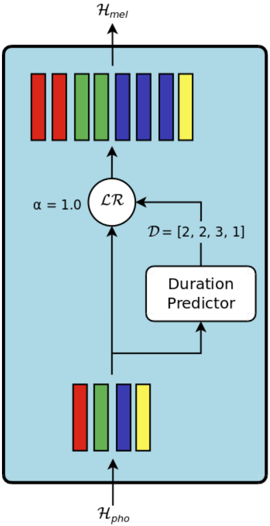
1. Self-attention: multi-head attention to extract cross-position information
2. 1-D convolution layers (instead of dense networks in transformer network): as adjacent hidden states of phoneme or mel-spectrogram sequences are more closely related

Fastspeech: Length regulator



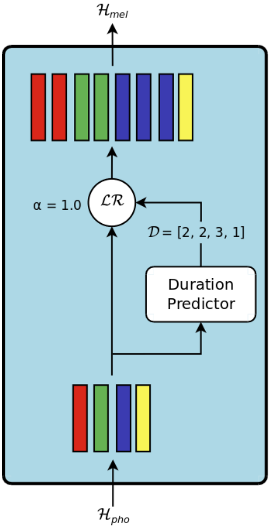
- Length of phoneme sequence expanded to length of mel-spectrogram sequence
- Speed up or slow down generated speech during synthesis

Fastspeech: Length regulator



$$\mathcal{H}_{mel} = \mathcal{LR}(\mathcal{H}_{pho}, D, \alpha)$$

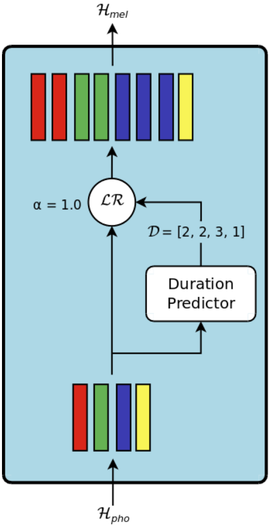
Fastspeech: Length regulator



$$\mathcal{H}_{mel} = \mathcal{LR}(\mathcal{H}_{pho}, D, \alpha)$$

Let $\mathcal{H}_{pho} = [h_1, h_2, ..., h_n]$
n: length of input sequence

Fastspeech: Length regulator

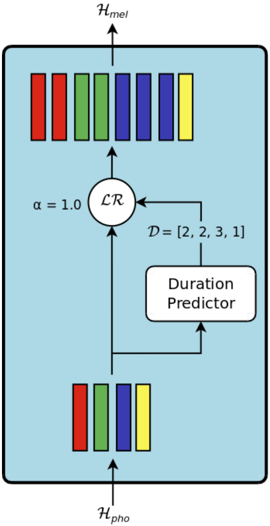


$$\mathcal{H}_{mel} = \mathcal{LR}(\mathcal{H}_{pho}, D, \alpha)$$

Let $\mathcal{H}_{pho} = [h_1, h_2, ..., h_n]$
n: length of input sequence

Phoneme duration sequence:
$\mathcal{D} = [d_1, d_2, ..., d_n]$, where d_1 is the no. of frames corresponding to h_1

Fastspeech: Length regulator



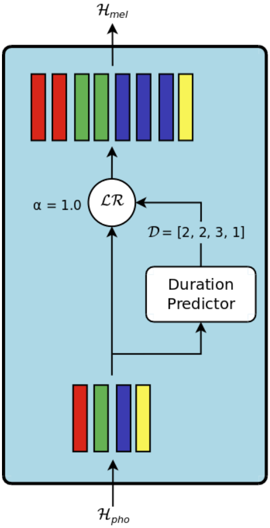
$$\mathcal{H}_{mel} = \mathcal{LR}(\mathcal{H}_{pho}, D, \alpha)$$

Let $\mathcal{H}_{pho} = [h_1, h_2, ..., h_n]$
n: length of input sequence

Phoneme duration sequence:
$\mathcal{D} = [d_1, d_2, ..., d_n]$, where d_1 is the no. of frames corresponding to h_1

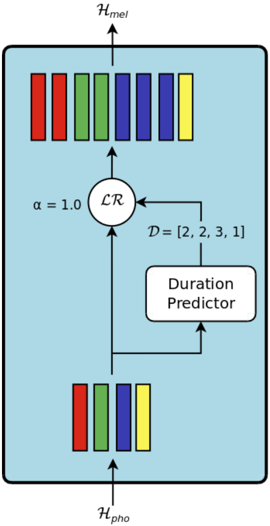
α: hyperparameter for speed control

Fastspeech: Length regulator



If $\alpha = 0.5$ (fast speed),
$\mathcal{D}_{\alpha=0.5} = [1, 1, 1.5, 0.5]$

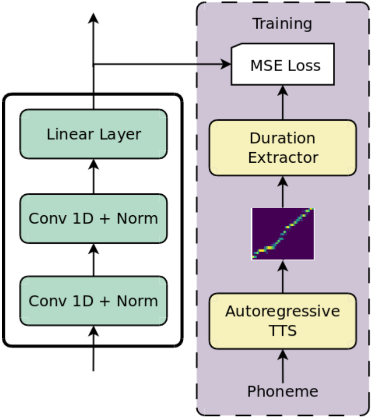
Fastspeech: Length regulator



If $\alpha = 0.5$ (fast speed),
$\mathcal{D}_{\alpha=0.5} = [1, 1, 1.5, 0.5]$

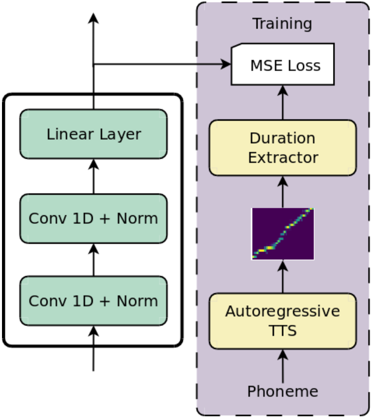
$\mathcal{D}_{\alpha=0.5} \approx [1, 1, 2, 1]$

Fastspeech: Duration predictor



- 2 layers of 1D convolution network with layer normalization
- Linear layer at output: predicts phoneme duration
- Duration predictor used only during synthesis

Fastspeech: Knowledge distillation



- Duration information for training data: attention alignments generated from teacher network (Tacotron2, transformer)
- Fastspeech model trained with training text and mel-spectrograms generated by teacher network

Fastspeech: Synthesised examples

To deliver interfaces that are significantly better suited to create and process RFC eight twenty one, RFC eight twenty two, RFC nine seventy seven, and MIME content. Auto-regressive Fastspeech

From Fastspeech to Fastspeech2

Teacher-student distillation

- Complicated training of two-stage teacher-student model
- Information loss of target mel-spectrogram
- Duration information extracted from teacher model may not be accurate

Ren et. al., "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech", International Conference on Learning Representations (ICLR), 2021.

From Fastspeech to Fastspeech2

Teacher-student distillation

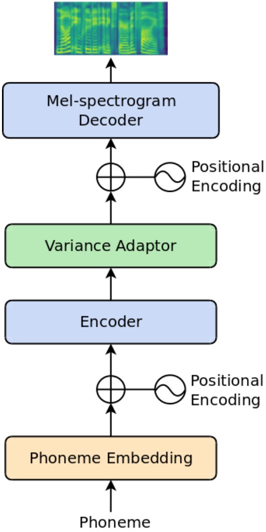
- Complicated training of two-stage teacher-student model
- Information loss of target mel-spectrogram
- Duration information extracted from teacher model may not be accurate

Fastspeech2

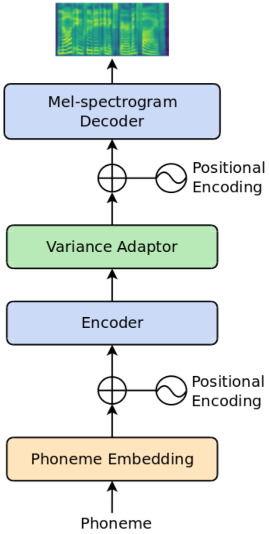
- Ground truth mel-spectrograms used instead of generated mel-spectrograms from teacher model
- Pitch and energy also included for more variation

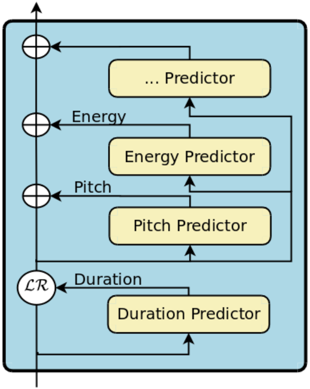
Ren et. al., "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech", International Conference on Learning Representations (ICLR), 2021.

Fastspeech2

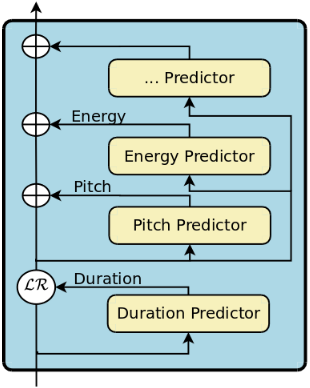


Fastspeech2





Fastspeech2



- Training: alignments (duration information) obtained from external aligner
- Training: Ground truth values of pitch and energy
- Synthesis: duration, pitch and energy predictors used
- Synthesised examples: Fastspeech Fastspeech2

Neural Vocoders

- WaveNet
- WaveGlow
- GAN based: Parallel WaveGAN, MelGAN, Multi-band MelGAN, HiFiGAN, StyleMelGAN

Re-defining end-to-end TTS

- Get rid of mel-spectrograms as intermediate representation
- Waveform contains more information (Ex: phase) than mel-spectrograms $\rightarrow$ Information gap between text and waveform larger compared to text and mel-spectrogram
- Fastspeech2s [6], VITS [7], JETS [8]

[6] Ren et. al., "FastSpeech: Fast, Robust and Controllable Text to Speech", International Conference on Neural Information Processing Systems, 2019, pp. 3171–3180.

[7] Jaehyeon Kim, Jungil Kong, and Juhee Son, "Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech", ICML 2021

[8] Dan Lim, Sunghee Jung, Eesung Kim, "JETS: Jointly Training FastSpeech2 and HiFi-GAN for End to End Text to Speech", INTERSPEECH 2022.

With additional embeddings

- Speaker embedding (multispeaker training)
- Language embedding (multilingual training)
- Global style token (GST) embedding

End-to-end TTS

Advantages

- High quality speech
- Easy to train

End-to-end TTS

Advantages

- High quality speech
- Easy to train

Disadvantages

- Requires a huge amount of training data
- Computationally intensive

Evaluation of TTS systems

- Objective measures:
  - Mel-cepstral distortion scores
- Subjective measures:
  - Mean opinion score (MOS)
  - Degradation mean opinion score (DMOS)
  - Pairwise comparison (PC) test

Research areas

- Multilingual aspects– bilingual, code-mixing
- Prosody- expressive voice, emotional TTS
- Voice conversion
- Conversational speech

Resources

- Festival [9], Festvox [10]
- HTK [11], HTS [12]
- Merlin toolkit [13]
- ESPnet [14]
- Indic TTS website [15]

---

[9] www.cstr.ed.ac.uk/projects/festival/
[10] http://festvox.org/
[11] www.danielpovey.com/files/htkbook.pdf
[12] hts.sp.nitech.ac.jp
[13] github.com/CSTR-Edinburgh/merlin
[14] github.com/espnet/espnet
[15] www.iitm.ac.in/donlab/tts/

Additional References

- Alan W. Black and Paul A. Taylor, "The Festival Speech Synthesis System: System documentation". Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, Scotland, UK, 1997. Avaliable at `www.cstr.ed.ac.uk/projects/festival.html`.
- Steve Young et. al., (2015). "The HTK Book" (version 3.5a).
- Zhizheng Wu, Oliver Watts, Simon King, "Merlin: An Open Source Neural Network Speech Synthesis System", ISCA Speech Synthesis Workshop (SSW9), September 2016, USA.
- S. Watanabe et al., "Espnet: End-to-end speech processing toolkit", INTERSPEECH, 2018, pp. 2207–2211.
- Tomoki Hayashi et. al., "Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit", INTERSPEECH, 2020, pp. 7654–7658.
- Xu Tan et. al., "A Survey on Neural Speech Synthesis", CoRR abs/2106.15561 (2021).
- R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A Flow-based Generative Network for Speech Synthesis", ICASSP, 2019, pp. 3617–3621.
- Ryuichi Yamamoto, Eunwoo Song, Jae-Min Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram", ICASSP, 2020, pp. 6199–6203.

Additional References

- Kundan Kumar et. al., "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis", NeurIPS, 2019, pp. 14910—14921.
- Jungil Kong, Jaehyeon Kim, Jaekyoung Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis", NeurIPS, 2020, pp. 17022-–17033.
- Ahmed Mustafa, Nicola Pia, Guillaume Fuchs, "StyleMelGAN: An Efficient High-Fidelity Adversarial Vocoder with Temporal Adaptive Normalization", ICASSP, 2021, pp. 6034–6038.