Conversational UIs

Spoken Language Processing

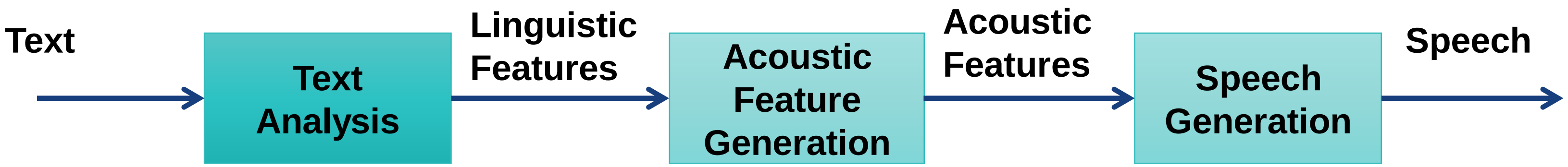
Topic 2: Speech Synthesis

Dr. Dong Minghui
Institute for Infocomm Research
Agency for Science, Technology and Research (A-Star)
Email: mhdong@i2r.a-star.edu.sg

What is Speech Synthesis

- To convert text into speech. Also referred to Text-to-Speech (TTS).
- To make machine speak like human.
- To convert discrete symbols into continuous signals.

Speech Synthesis Process



Text Analysis

- To convert raw text into linguistic features

Acoustic feature generation

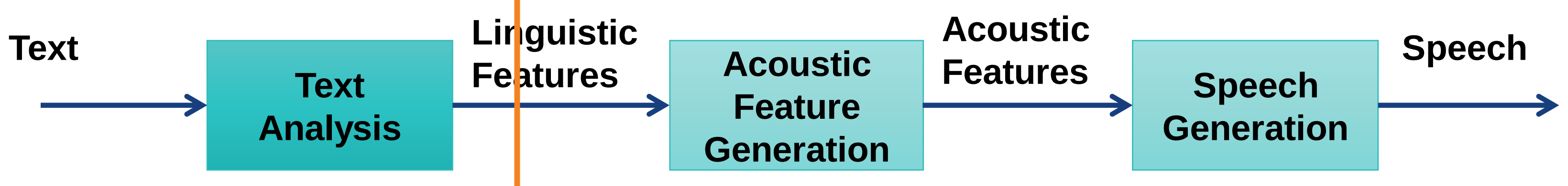
- To convert linguistic features into acoustic and prosodic information

Speech Generation

- To convert acoustic and prosodic information into speech signal.

PART 1. TEXT ANALYSIS

Speech Synthesis Process



Text Analysis

- To convert raw text into linguistic features

Acoustic feature generation

- To convert linguistic features into acoustic and prosodic information

Speech Generation

- To convert acoustic and prosodic information into speech signal.

Text Analysis - Normalization

- **Identify sentence end**
  - Sentence end is not straight forward as a dot can be a period or it can be an abbreviation.
  - e.g. My place on Forest Ave. is around the corner. (Is dot the end of sentence?)
- **Convert all symbols into words**
  - 10%, $120, #10, etc
  - 11:10 (time or a score for a badminton game ?)
- **Convert all abbreviation to words**
  - 10 Jan. (10th of January)
  - It’s 13 St. Andrews St. (Saint or Street ?)
- **Convert digits to words**
  - 1830 (Year or number?)
  - IV (four or I.V.?)

Text Analysis - Pronunciation

- **Word pronunciations**
  - Lexicon is used for pronunciations
  - Example: PRONUNCIATION: P R OW N AH N S IY EY SH AH N .
  - How to deal with words with multiple pronunciations? Like “record” as a noun or verb.
  - How to deal with out-of-vocabulary words?
- **G2P model**
  - For words are not in lexicon, machine learning programs are used to train a grapheme to phoneme model.

Text Analysis – POS tagging

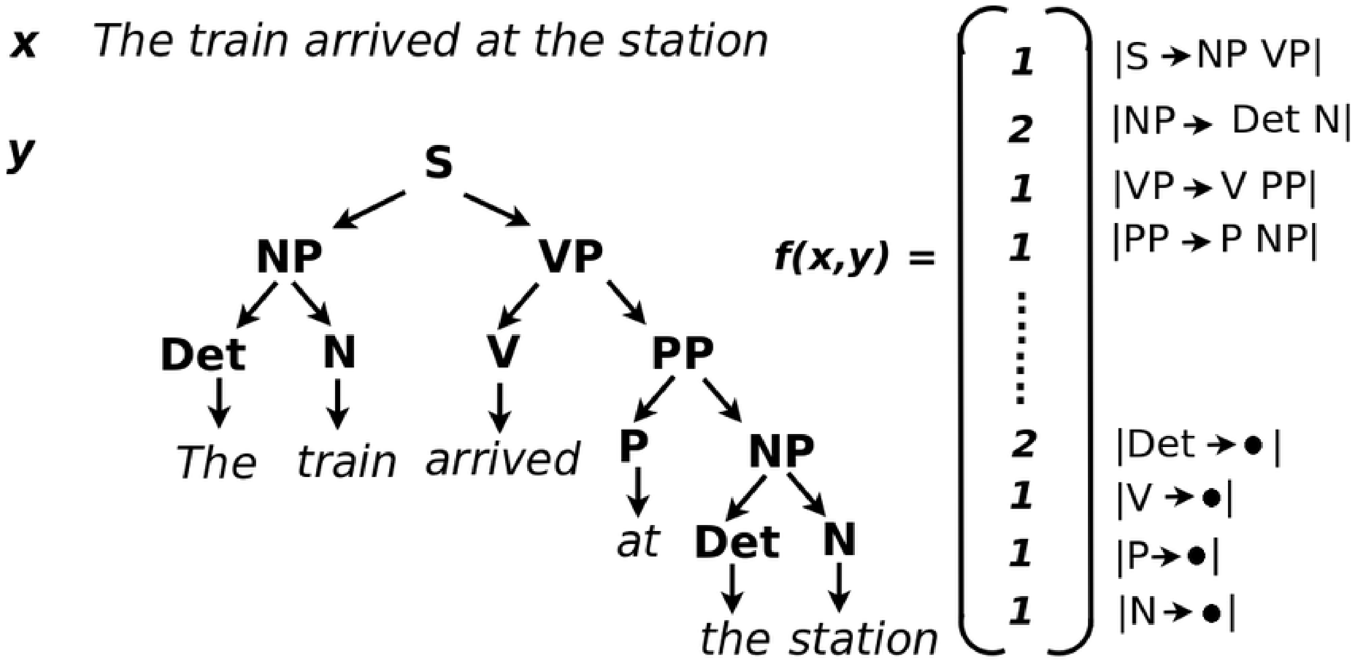
Part-of-Speech Tagging

- To know a word is a noun, verb, adjective or adverb, pronoun, etc.
- Different types of word play different functions in sentence
- POSs describe the relationship between words.
- Used to determine pronunciation based on POS. e.g. the words record is read differently as a noun or a verb.

```
[('They', 'PRP'), ('refuse', 'VBP'), ('to', 'TO'), ('permit', 'VB'), ('us',
'PRP'), ('to', 'TO'), ('obtain', 'VB'), ('the', 'DT'), ('refuse', 'NN'),
('permit', 'NN')]
```

Text Analysis - Parsing

- **Parsing**
  - To understand the structure of sentence.
  - Parsing may also be applied to help the prosodic phrase prediction.



Text Analysis- Prosodic Phrase

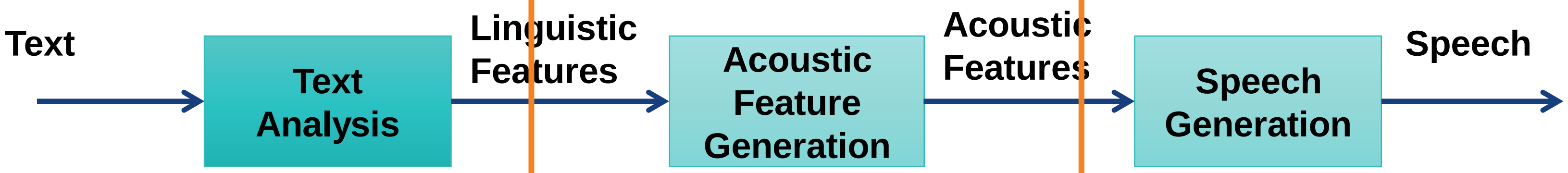
- **Prosodic phrase boundary prediction**
  - Long sentences are normally broken into phases when reading.
  - Prosodic phrase prediction is to predict where the pauses will be.
  - Use statistical based machine learning models to predict.
  - Use POS and Parsing result as features for prediction.

Example:

A sentence is a group of words | that are put together | to mean something.

PART 2. ACOUSTIC FEATURE PREDICTION

Speech Synthesis Process



Text Analysis

- To convert raw text into linguistic features

Acoustic feature generation

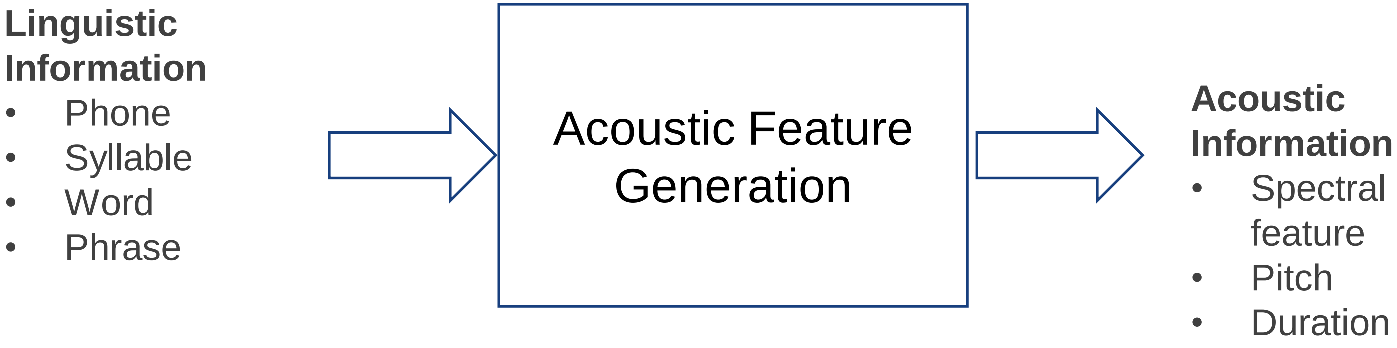
- To convert linguistic features into acoustic and prosodic information

Speech Generation

- To convert acoustic and prosodic information into speech signal.

Acoustic Feature Generation

- **To convert linguistic (text) information into acoustic (speech) information.**



Linguistic Information

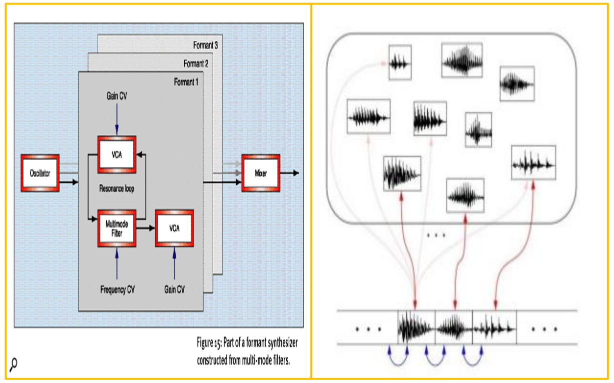
- Phone
- Syllable
- Word
- Phrase

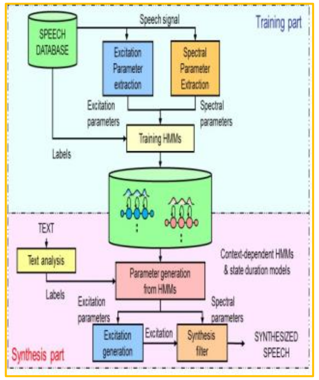
Acoustic Feature Generation

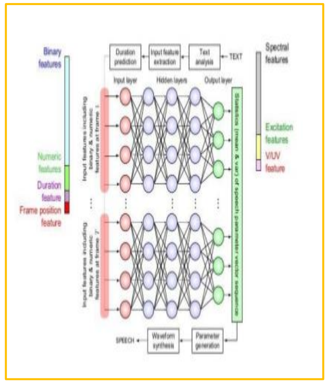
Acoustic Information

- Spectral feature
- Pitch
- Duration

Development of TTS methods







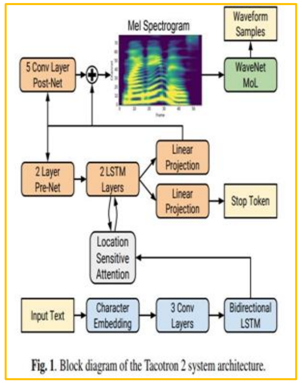


Fig. 1. Block diagram of the Tacotron 2 system architecture.

Early methods

- 1970s,1980s
- Formant
- PSOLA

Concatenation methods

- Since 1990s
- Unit selection

HMM methods

- Since 2000
- Parameters+Vocoder

Deep Learning

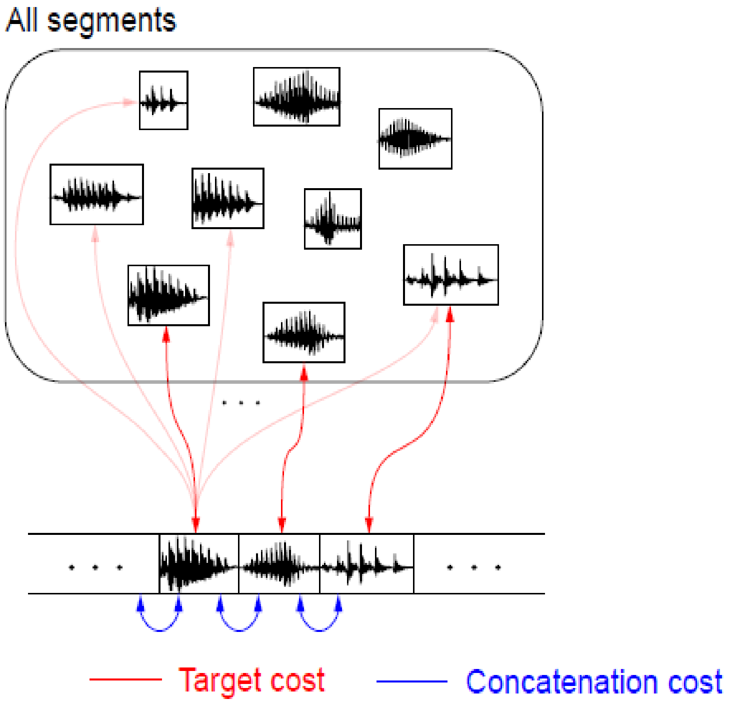
- Since 2012
- DNN for acoustic modelling

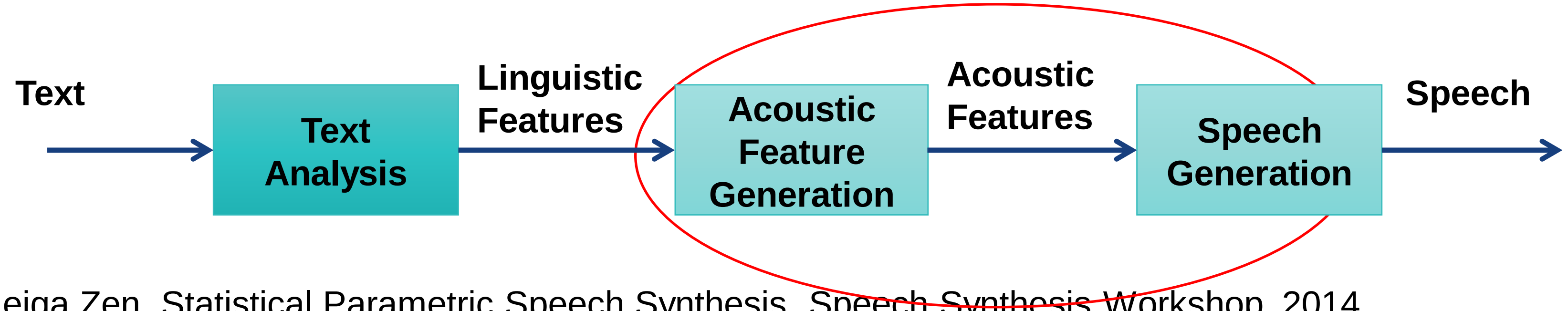
End-to-end

- Since 2017
- End2end model
- Neural vocoder

Concatenation Method (Unit selection)

- Directly copy speech segments and concatenate them.
- Big speech database as unit inventory.
- Target cost to control the prosodic and acoustic appropriateness.
- Concatenation cost to control the smoothness
- Viterbi algorithm to selected best unit sequence.

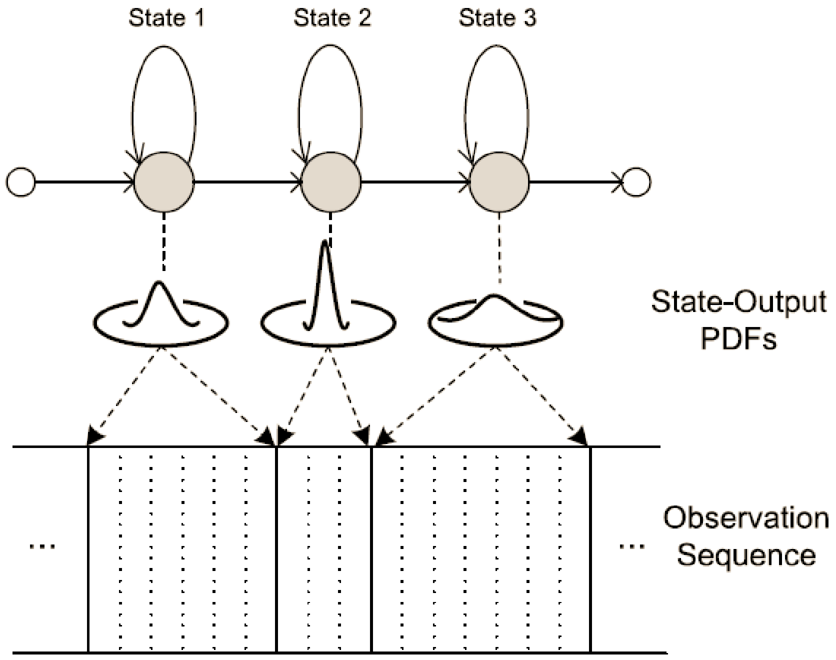


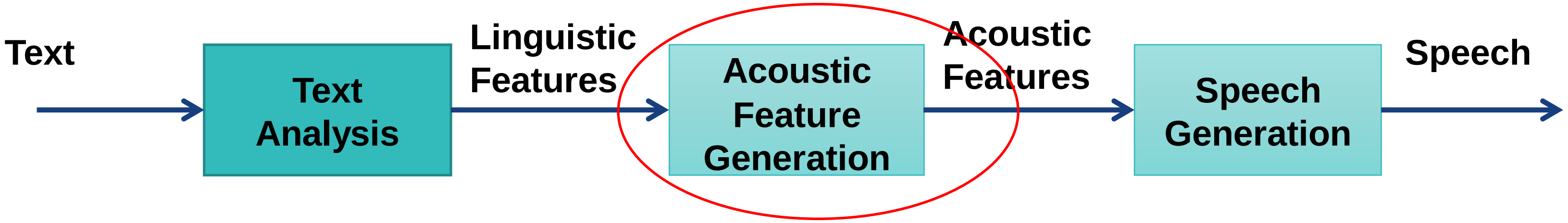


Heiga Zen, Statistical Parametric Speech Synthesis, Speech Synthesis Workshop, 2014.

HMM Method

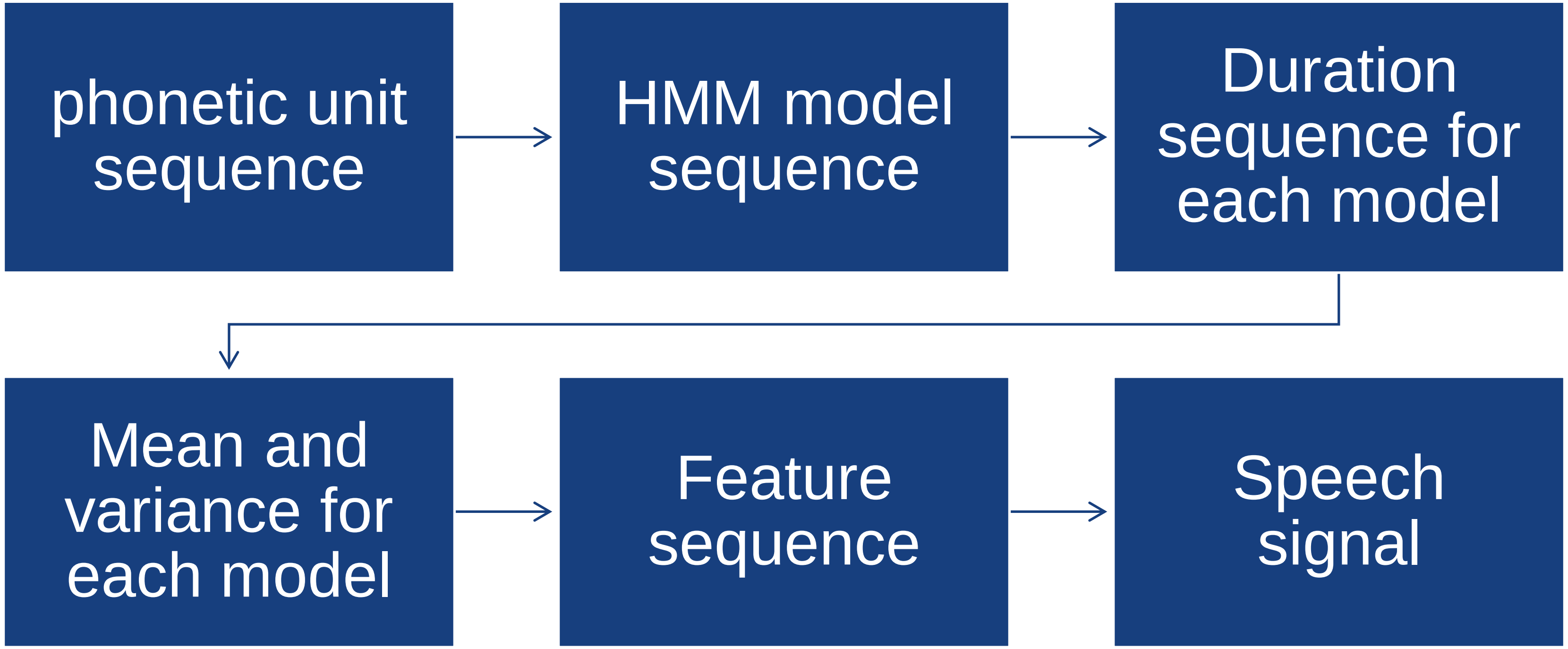
- Inspired by speech recognition process.
- Speech frames are mapped to HMM states.
- Train model with speech database
- Generate frame sequence from hidden state sequence
- Convert frame feature sequence into speech





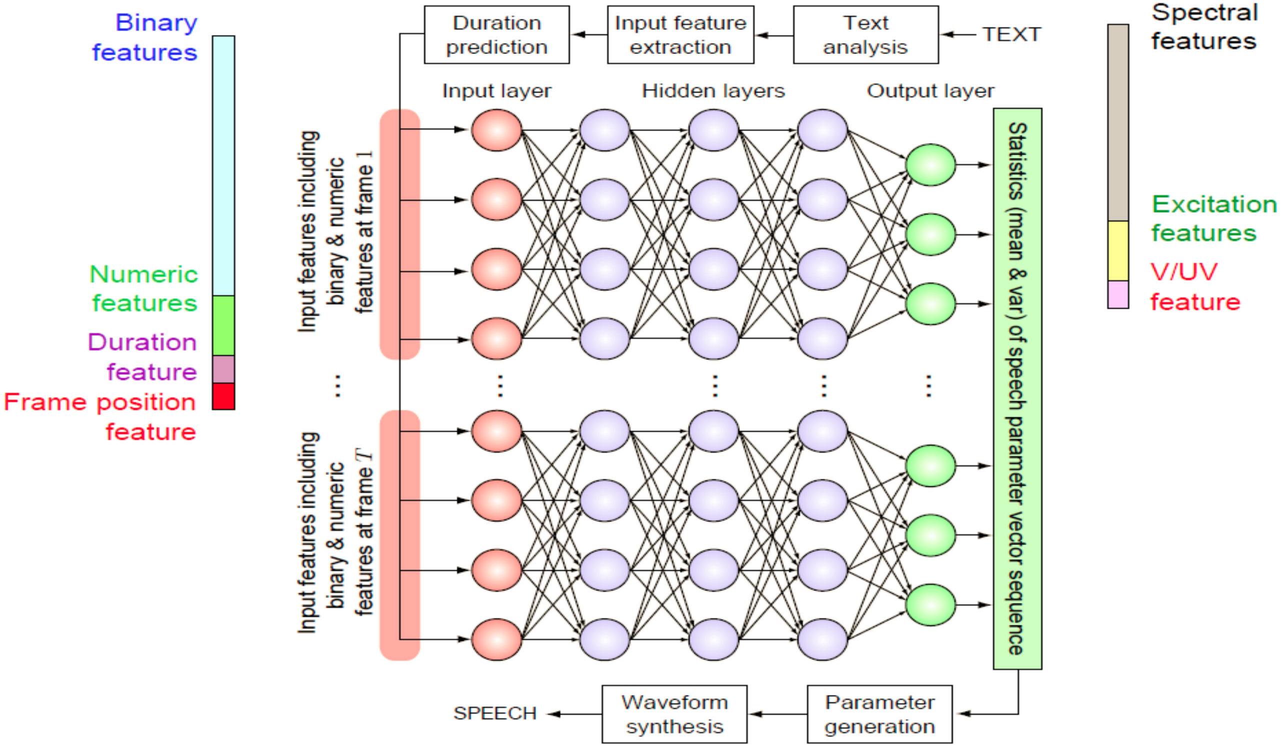
Heiga Zen, Statistical Parametric Speech Synthesis, Speech Synthesis Workshop, 2014.

HMM Method

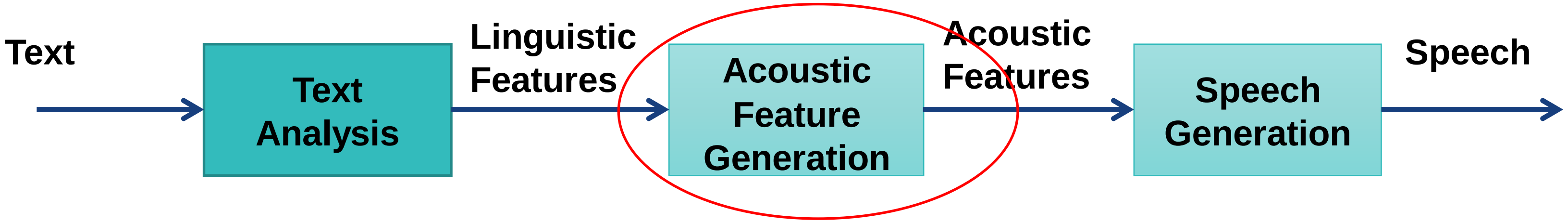


phonetic unit sequence → HMM model sequence → Duration sequence for each model → Mean and variance for each model → Feature sequence → Speech signal

Deep Learning Method - DNN

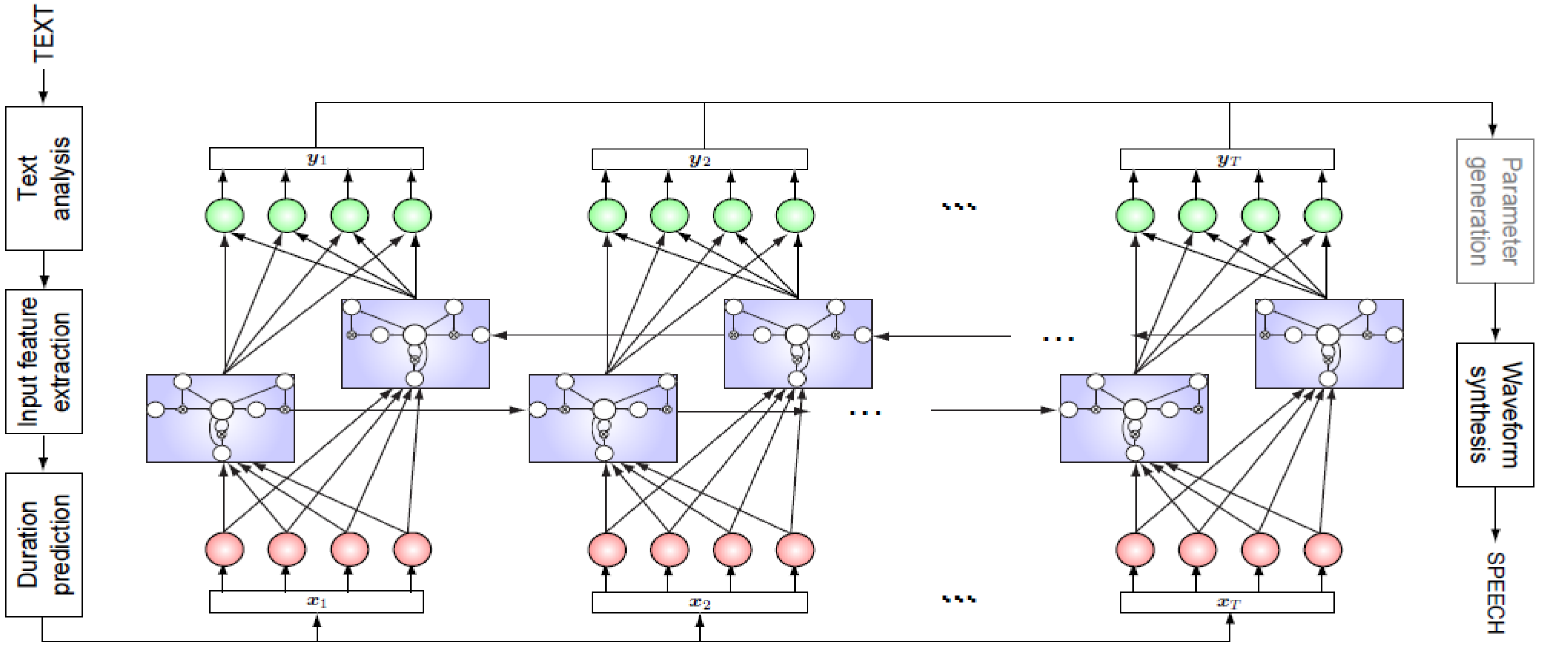


This is a replacement of HMM methods (same kinds of features used)

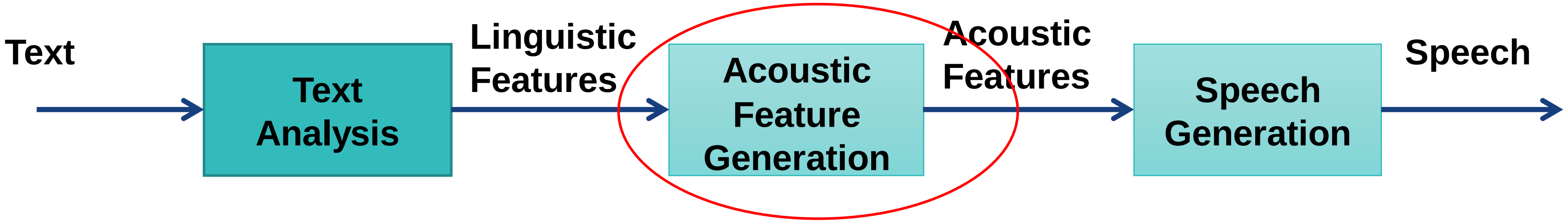


Heiga Zen, Statistical Parametric Speech Synthesis, Speech Synthesis Workshop, 2014.

Deep Learning Method - LSTM



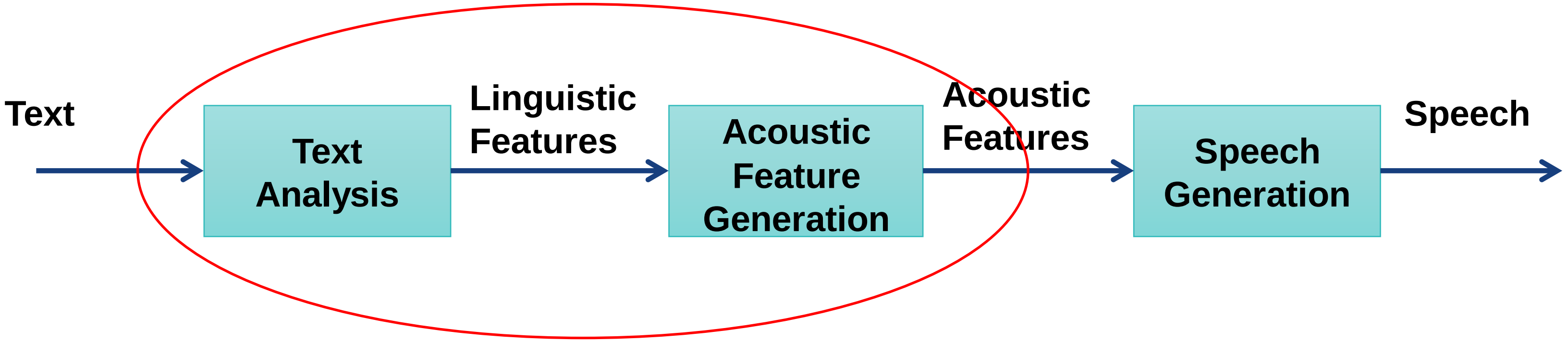
Past and future information in the time series have been considered.


Text
Text Analysis
Linguistic Features
Acoustic Feature Generation
Acoustic Features
Speech Generation
Speech

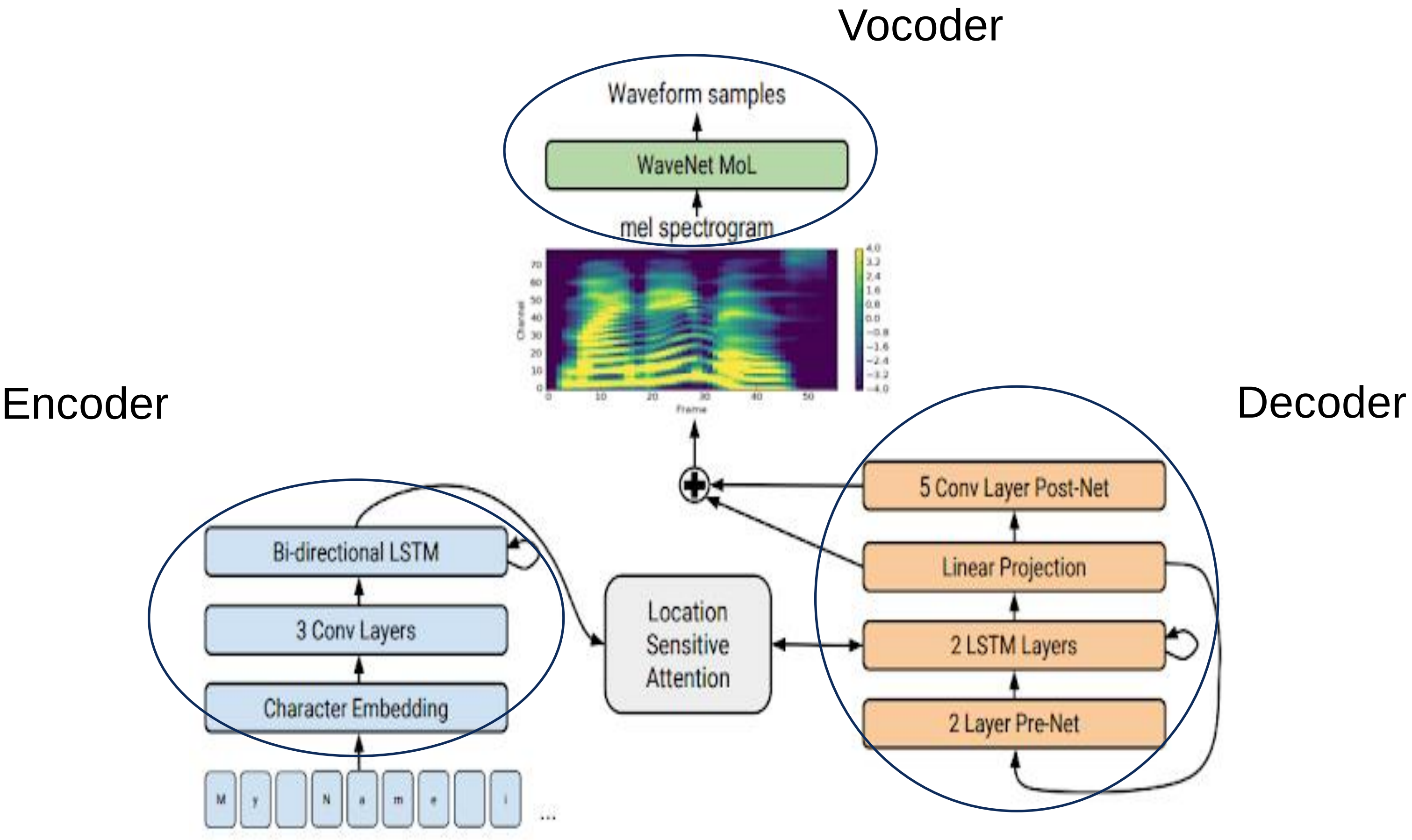
Heiga Zen, Statistical Parametric Speech Synthesis, Speech Synthesis Workshop, 2014.

End-to-End Synthesis (Tacotron 2)

- **Tacotron**
  - Tacotron 2 was published by Google in Dec 2017.
  - Solution: Sequence-to-Sequence model
  - An encoder, attention-based decoder, post-processing
- **Advantages:**
  - Character as input, no feature engineering needed
  - More robust than multi-stage model where errors can compound.
  - No need of phoneme-level alignment.



End-to-end Synthesis (Tacotron 2)



J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," Dec 2017

End-to-end Synthesis (Tacotron 2)

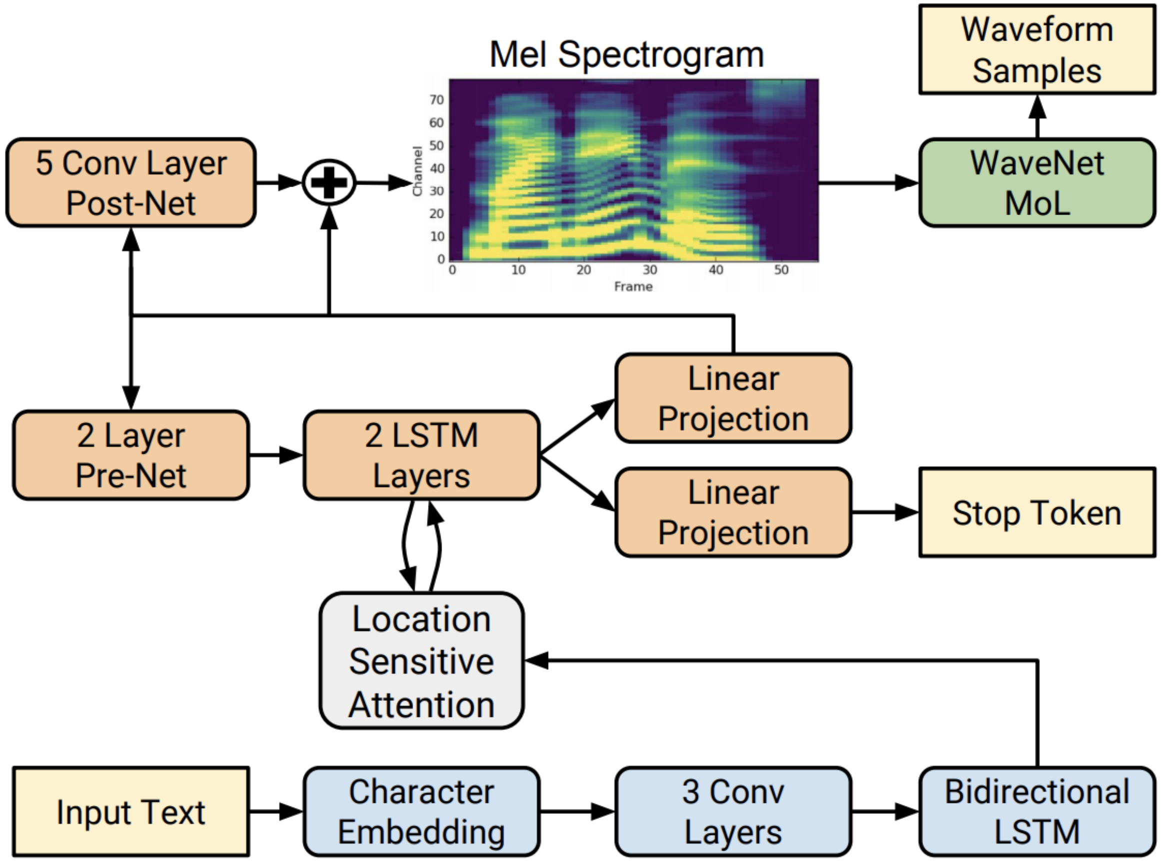
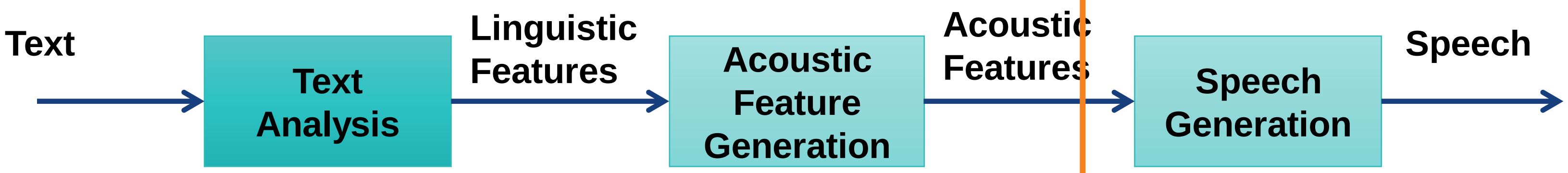


Fig. 1. Block diagram of the Tacotron 2 system architecture.

J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," Dec 2017

PART 3. VOCODER

Speech Synthesis Process



Text Analysis

- To convert raw text into linguistic features

Acoustic feature generation

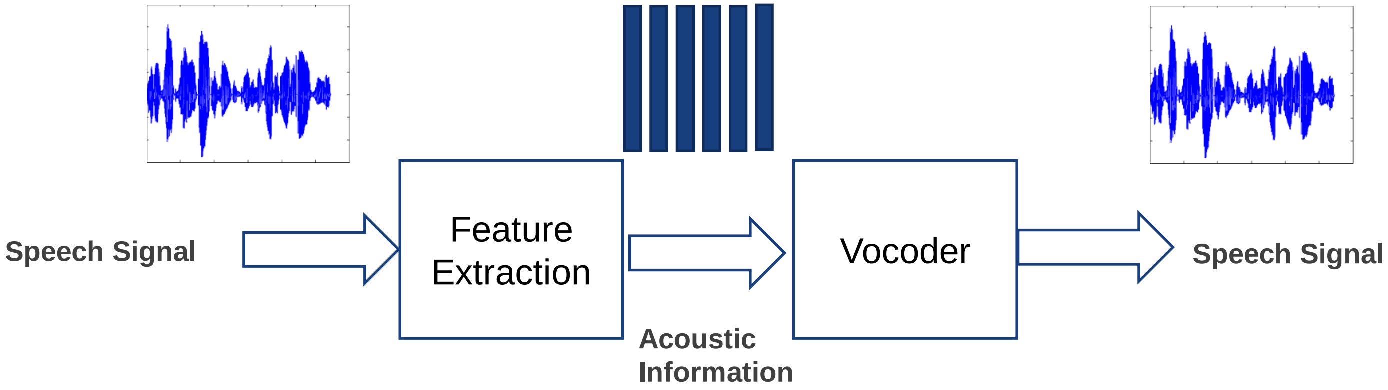
- To convert linguistic features into acoustic and prosodic information

Speech Generation

- To convert acoustic and prosodic information into speech signal.

Speech Signal Generation:Vocoder

Vocoder



- **Traditional vocoder**
  - Based on signal processing techniques
  - Not able to recover speech details from predicted features
  - Machine like voice quality

Neural vocoder

- Based on deep learning technology
- Generate signal sample by sample
- Close to human voice quality

Neural Vocoder

- **Wavenet**
  - Proposed by DeepMind in Sep 2016.
  - Directly modelling the raw waveform of the audio signal, one sample at a time.
  - Every sample is influenced by all previous ones.

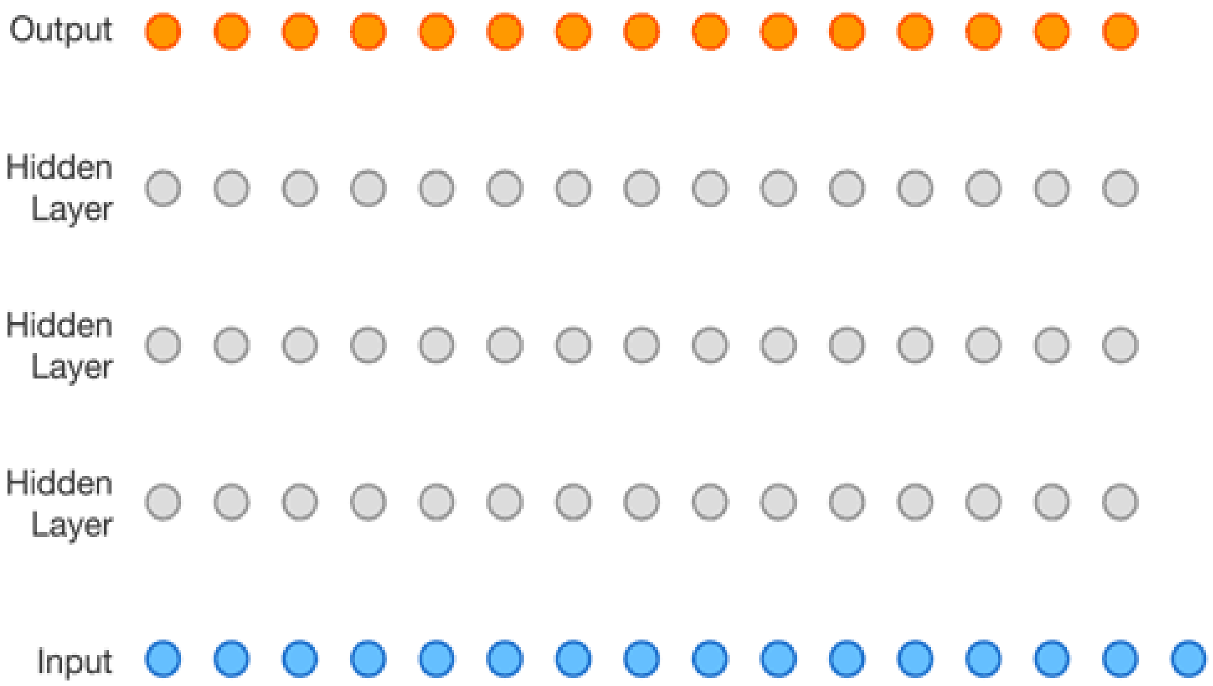
$$p\left(\mathbf{x} \mid \mathbf{h}\right)=\prod_{t=1}^{T} p\left(x_t \mid x_1, \ldots, x_{t-1}, \mathbf{h}\right).$$

  - Computational expensive method – needs GPU support
  - Slow – take long time to generate voice.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, Sep 2016

Neural Vocoder

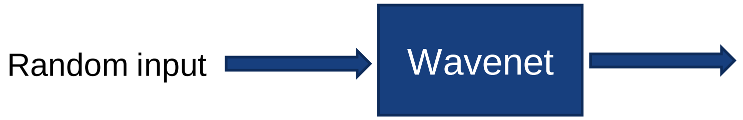
- **Wavenet**



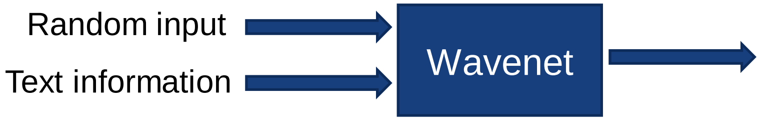
Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, Sep 2016

Neural Vocoder

- **Wavenet**
  - Random Generation



  - Conditional Generation



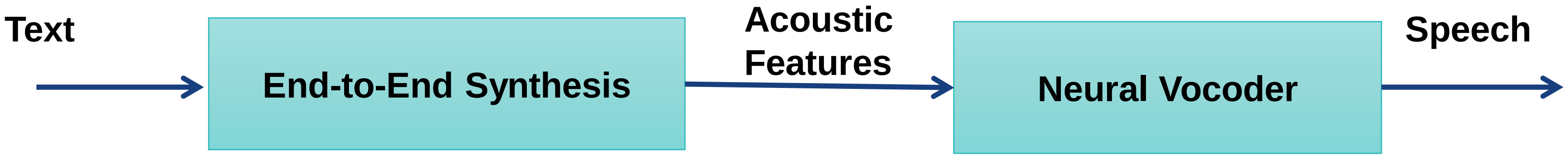
Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, Sep 2016

Other Vocoder

- **Improved neural vocoders**
  - Parallel Wavenet (Parallel processing)
  - Clarinet (parallel + improved modelling method)
  - WaveGlow/ FloWavnet (parallel modelling)
  - WaveRNN (Optimized matrix computation)
  - LPCNet (Neural network + signal processing)
  - WaveGAN (Appying GANs to unsupervised synthesis of raw-waveform audio).

PART 4. TTS SYSTEMS

An Example of TTS Engine



- **Tactoron 2 + Flowavenet**
- **Database**
  - 20-40 hours recording of clean speech from professional speaker.
- **Training Vocoder**
  - A neural vocoder needs about 1 to 3 weeks to train on a Nvidia TITAN V GPU card.
- **Training Acoustic Model**
  - End-to-end neural Text-to-Speech model takes about 30-40 hours on TITAN V GPU card.

End-to-end Synthesis (Sample)

Welcome to the A*STAR Procurement Governance e-Learning Module. This module introduces you to the Principles of Procurement, Procurement Procedures and Good Practices. Throughout this module, you will also be learning the importance of Procurement Governance.

By the end of this module you will be able to:

- Understand the Principles of Procurement.
- Identify the 8 stages of the Procurement Life Cycle.
- Follow the proper procurement procedures and good practices.

This module should take you approximately 60 minutes to complete.

Click the Next button below to begin.

End-to-end Synthesis (sample)

A team led by Stanford University researchers has demonstrated a way of separating hydrogen and oxygen gas from seawater via electricity. Current methods of splitting water rely largely on purified water, which is a scarce resource. The scientists layered nickel-iron hydroxide and nickel sulfide on top of a nickel foam core to allow it to withstand seawater corrosion.

Ongoing Research Efforts

- **Low Resource Language Speech Synthesis**
  - Build TTS system with very small dataset (e.g. less than 1 hour or even a few minutes)
  - Build speech synthesis for languages without text (e.g. dialects)
- **Personalized Speech Synthesis**
  - Generate a person's voice with small amount samples.
- **Emotion and Subtle Expression Generation**
  - Generate different emotions
  - Prosody details for expressing different intensions.

Open Source Packages

- **Traditional Systems:**
  - Festival: Traditional speech synthesis toolkit. Good examples for all the processing steps.
  - HTS: Traditional parametric system. Good example for parametric methods.
  - Merlin: Pipeline deep learning system,
- **End-to-End Systems:**
  - Tacotron: State-of-the-art acoustic modelling
  - FloWavenet: State-of-the-art vocoder modelling.
  - FastSpeech: State-of-the-art acoustic modelling.
  - EPSNet: End-to-end speech processing toolkit.

PART 5. TTS PROGRAMMING

Hands-on Session

- **Open source pytts3**
  - pyttsx3 is a text-to-speech conversion library in Python.
  - Supported synthesizers
    - SAPI5 on Windows XP and Windows Vista and Windows 8,8.1 , 10
    - NSSpeechSynthesizer on Mac OS X 10.5 (Leopard) and 10.6 (Snow Leopard)
    - espeak on Ubuntu Desktop Edition 8.10 (Intrepid), 9.04 (Jaunty), and 9.10 (Karmic)

Hands-on Session

Sample Program

```
import pyttsx3

engine = pyttsx3.init() # object creation


# Speaking rate

engine.setProperty('rate', 200)     # setting up new voice rate

engine.runAndWait()

rate = engine.getProperty('rate')   # getting details of current speaking rate


engine.say("Hello World!")

engine.say('My current speaking rate is ' + str(rate))

engine.runAndWait()

engine.stop()
```

Hands-on Session

Sample Program

```
#Listening for events

import pyttsx3

def onStart(name):

   print ('starting', name)

def onWord(name, location, length):

print ('word', name, message[location:location+length+1])

def onEnd(name, completed):

   print ('finishing', name, completed)

message = 'The quick brown fox jumped over the lazy dog.'
```

Hands-on Session

Sample Program

```
engine = pyttsx3.init()

dict1 = engine.connect('started-utterance', onStart)

dict2 = engine.connect('started-word', onWord)

dict3 = engine.connect('finished-utterance', onEnd)

engine.say(message)

engine.runAndWait()

engine.disconnect(dict1)

engine.disconnect(dict2)

engine.disconnect(dict3)
```

Thank you!

mhdong@i2r.a-star.edu.sg