NUS National University of Singapore | ISS

Conversational UIs

Spoken Language Processing

Dr. Gary Leung

Email: cc.leung@nus.edu.sg

Agenda

- Day 3
  - 1: Speech processing basics
  - 2: Speech recognition (Speech-to-text)
  - 3: Speaker diarization
- Day 4
  - 4: Speech synthesis (Text-to-speech)
  - 5: Voice conversion and generation
  - 6: Spoken dialogue system (Spoken chatbot)

Copyright National University of Singapore

Topic 4: Speech Synthesis

Speech Synthesis Process

Text Analysis	Acoustic feature generation	Speech Generation
• To convert raw text into linguistic features	• To convert linguistic features into acoustic and prosodic information	• To convert acoustic and prosodic information into speech signal.

- **To convert text into speech, or Text-to-Speech (TTS).**

Copyright National University of Singapore

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.

Copyright National University of Singapore

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.?)

Copyright National University of Singapore

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.

Copyright National University of Singapore

Text Analysis – POS tagging

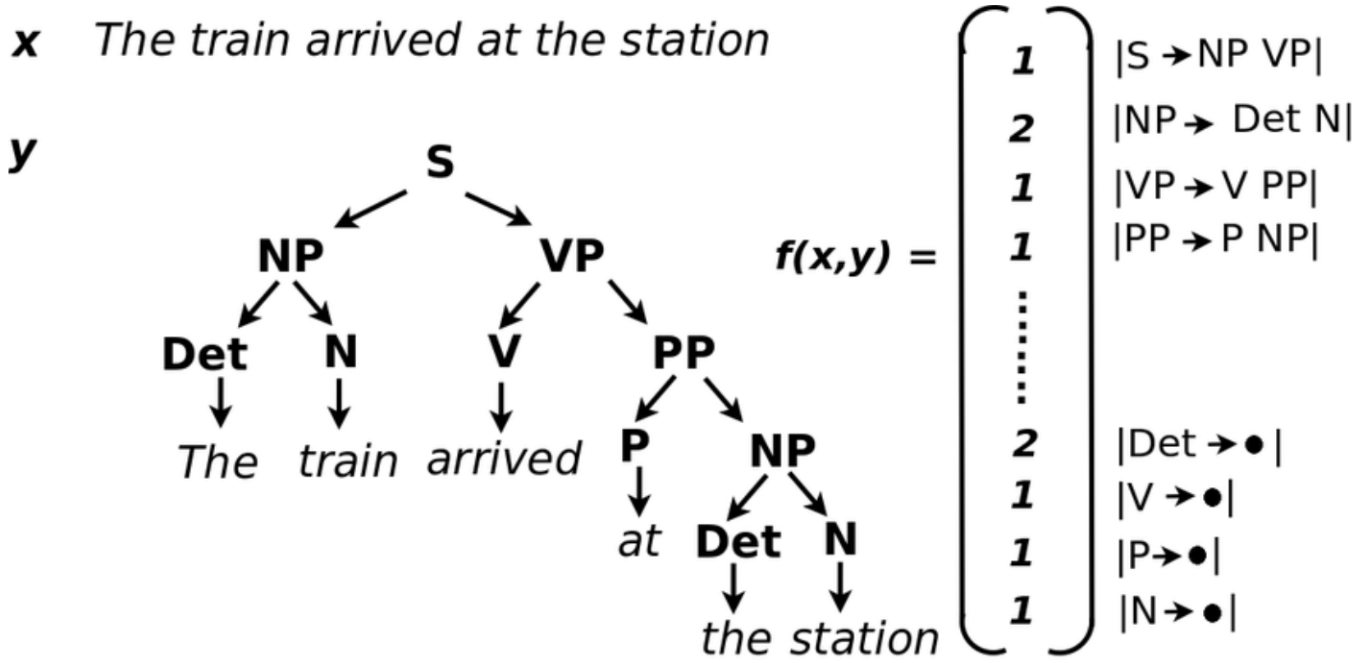
- **Part-of-Speech Tagging**
  - To know a word is a noun, verb, adjective or adverb, pronoun, etc.
  - Different types of word play different functions in sentence
  - POSs describe the relationship between words.
  - Used to determine pronunciation based on POS. e.g. the words record is read differently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


Copyright National University of Singapore

Text Analysis - Parsing

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



Copyright National University of Singapore

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.

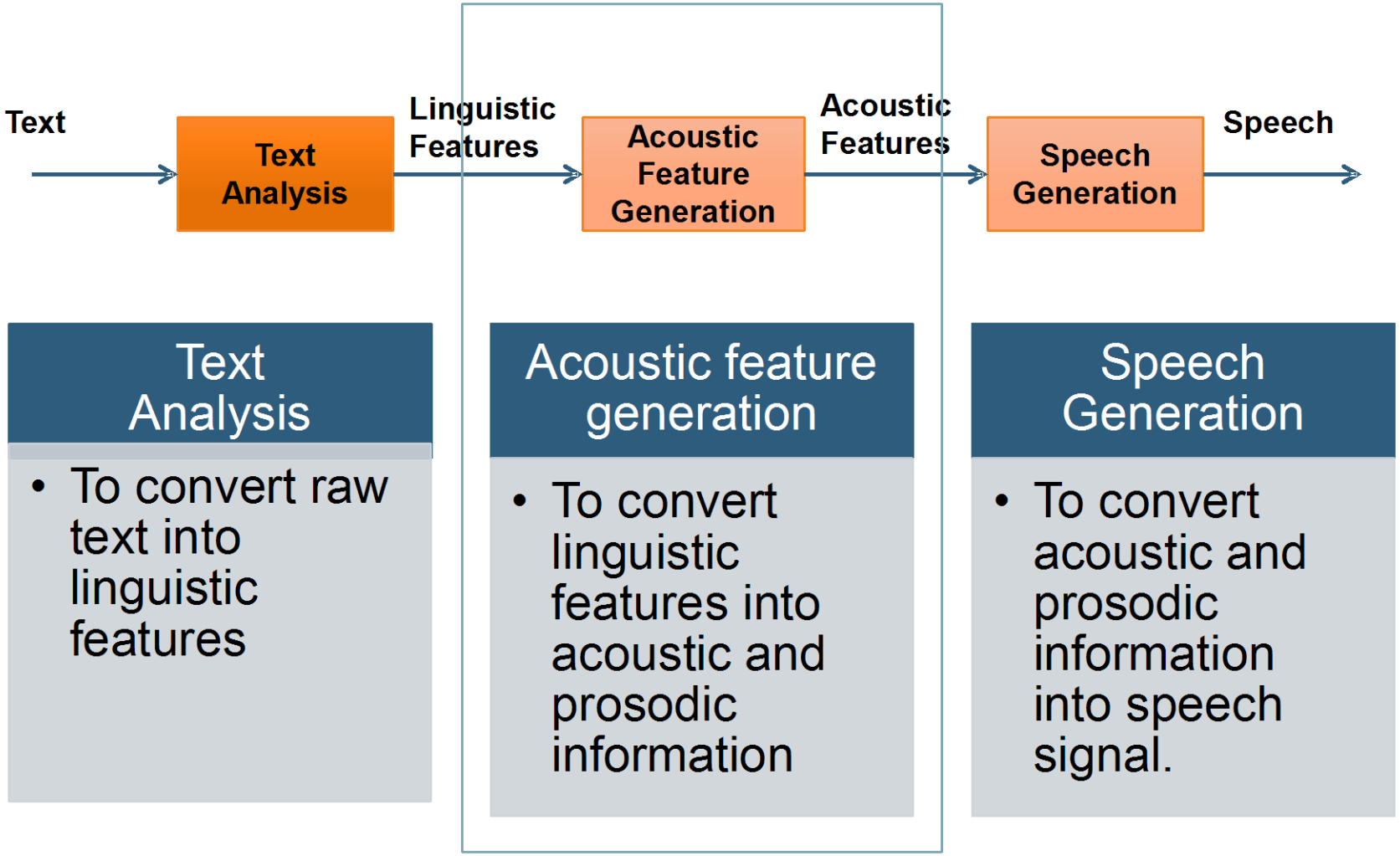
Example:

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

Copyright National University of Singapore

PART 2. ACOUSTIC FEATURE PREDICTION

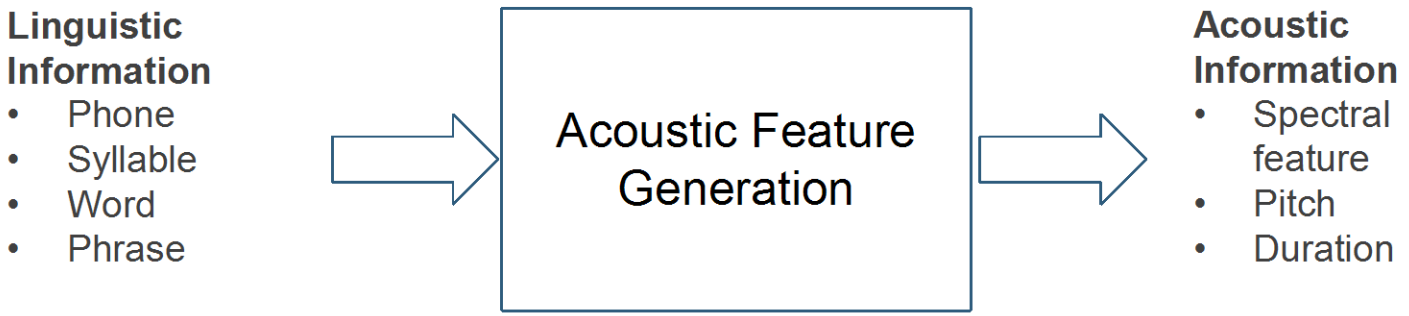
Speech Synthesis Process



Copyright National University of Singapore

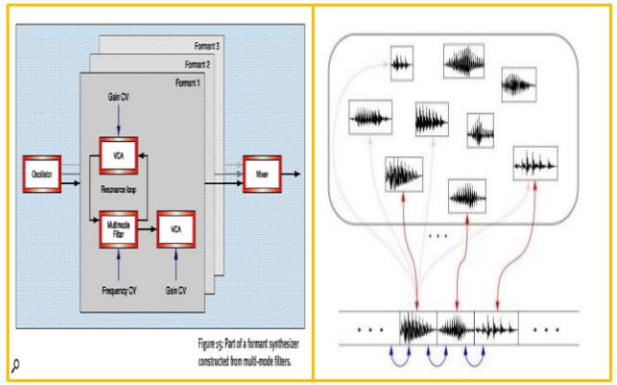
Acoustic Feature Generation

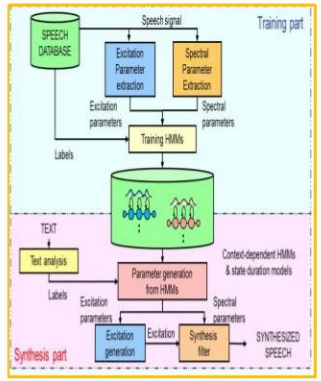
- **To convert linguistic (text) information into acoustic (speech) information.**

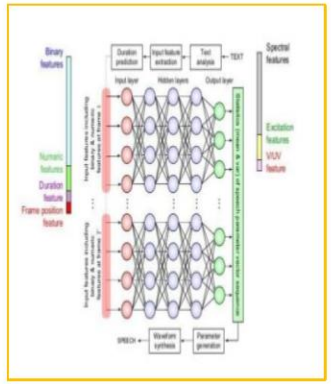


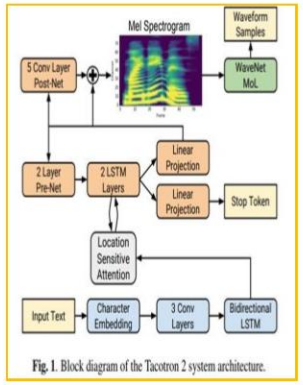
Copyright National University of Singapore

Development of TTS methods







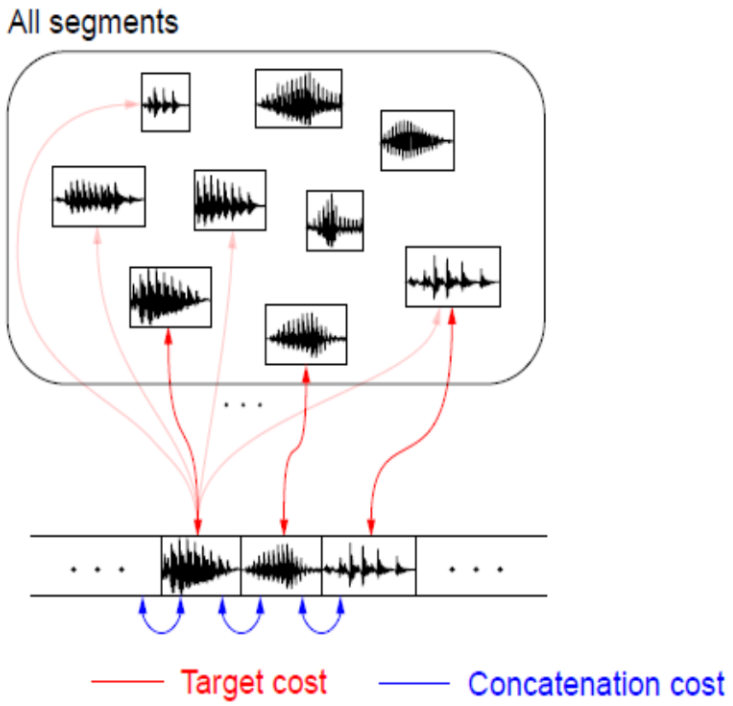


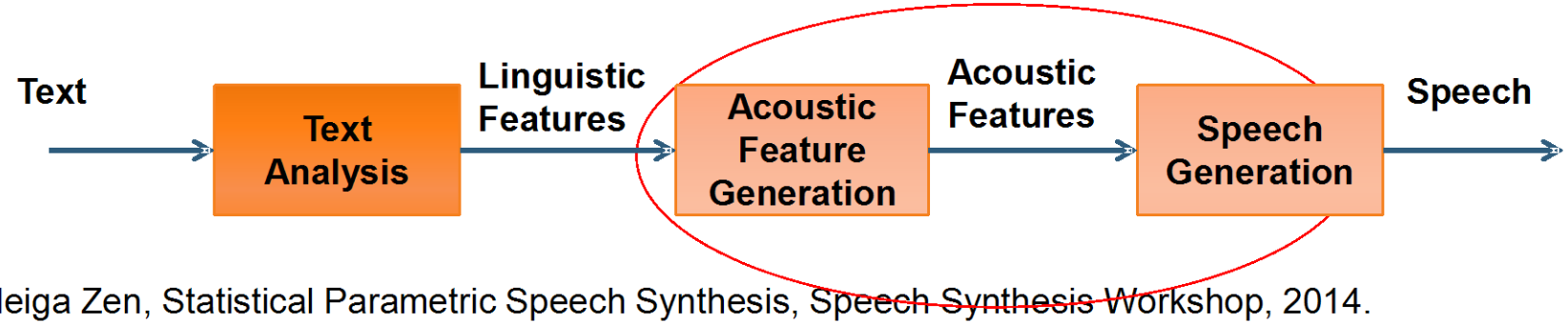
Early methods	Concatenation methods	HMM methods	Deep Learning	End-to-end
• 1970s,1980s • Formant • PSOLA	• Since 1990s • Unit selection	• Since 2000 • Parameters+Vocoder	• Since 2012 • DNN for acoustic modelling	• Since 2017 • End2end model • Neural vocoder

Copyright National University of Singapore

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.



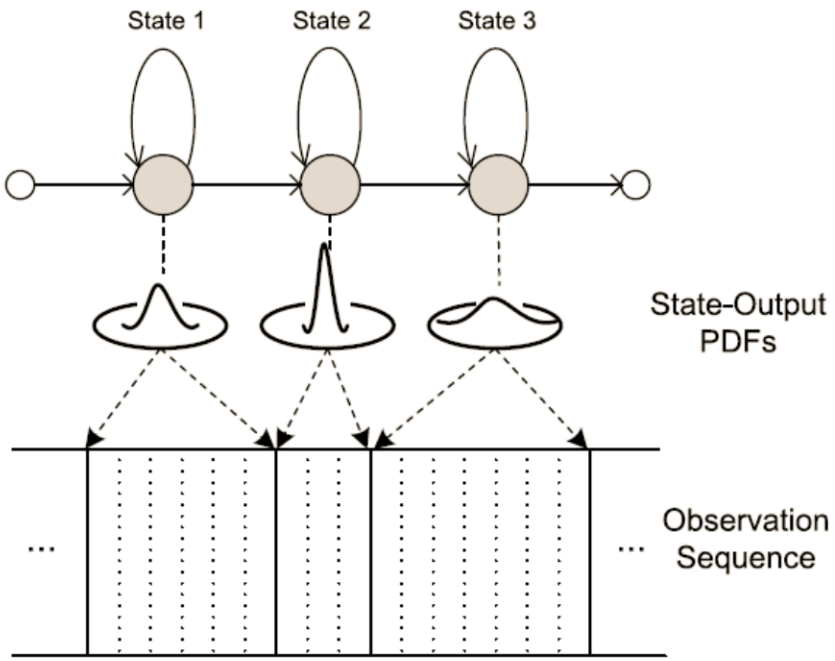


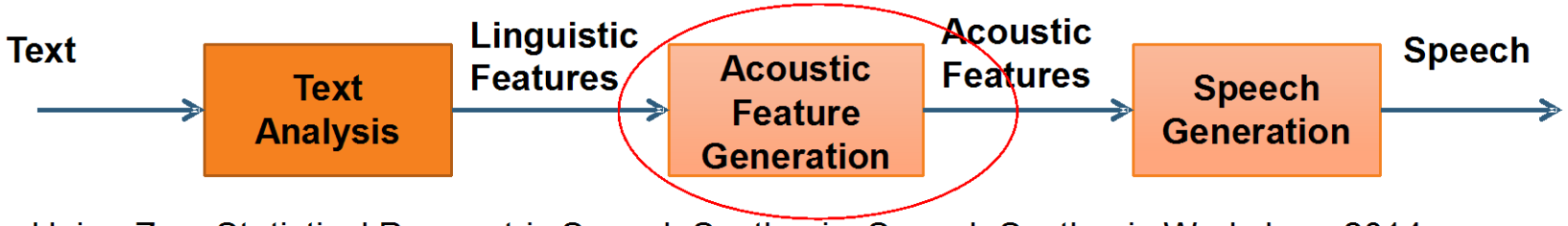
Heiga Zen, Statistical Parametric Speech Synthesis, Speech Synthesis Workshop, 2014.

Copyright National University of Singapore

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

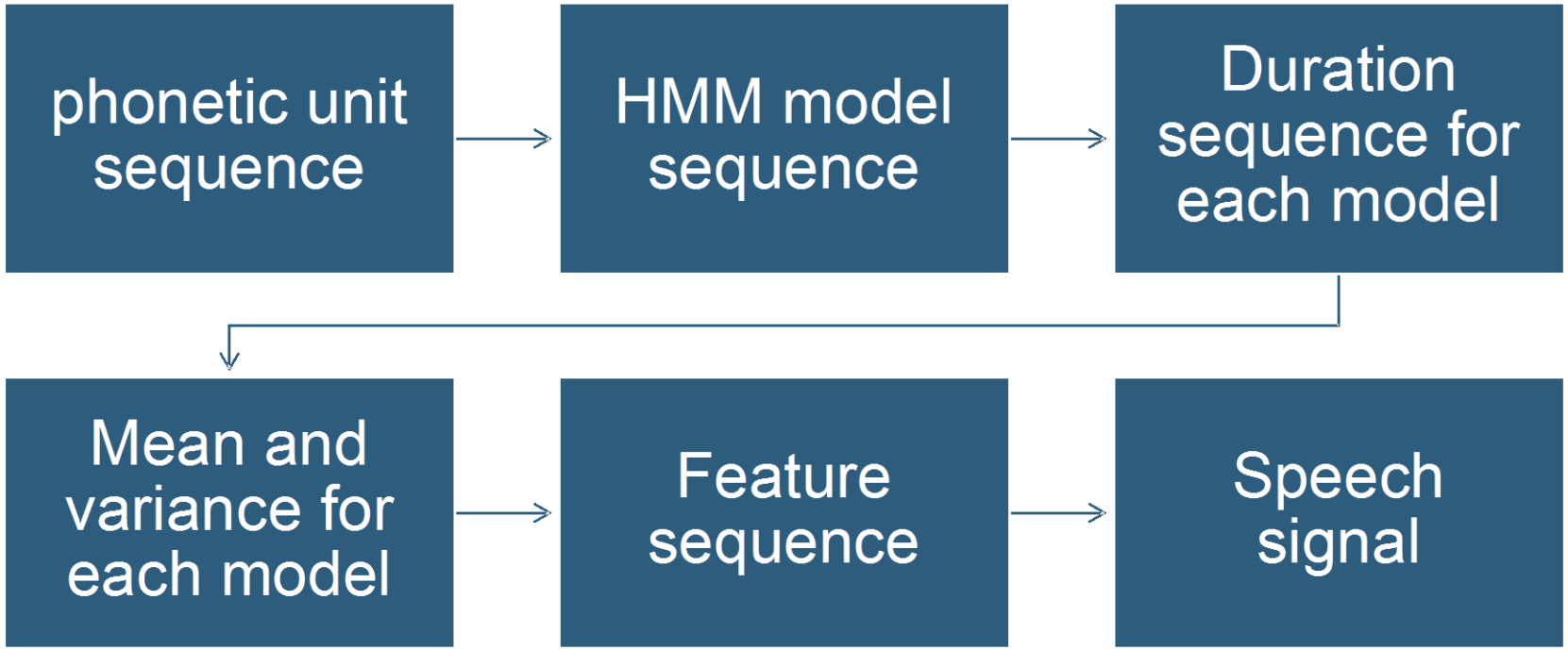




Heiga Zen, Statistical Parametric Speech Synthesis, Speech Synthesis Workshop, 2014.

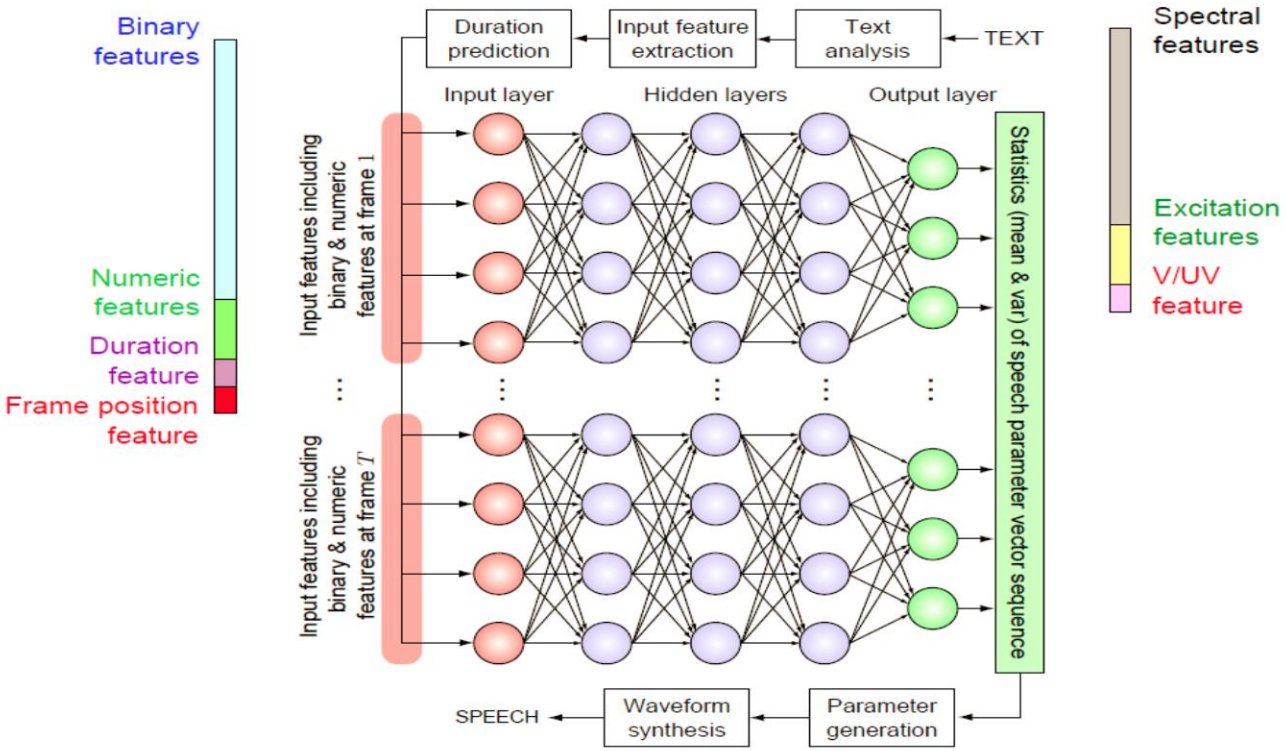
Copyright National University of Singapore

HMM Method

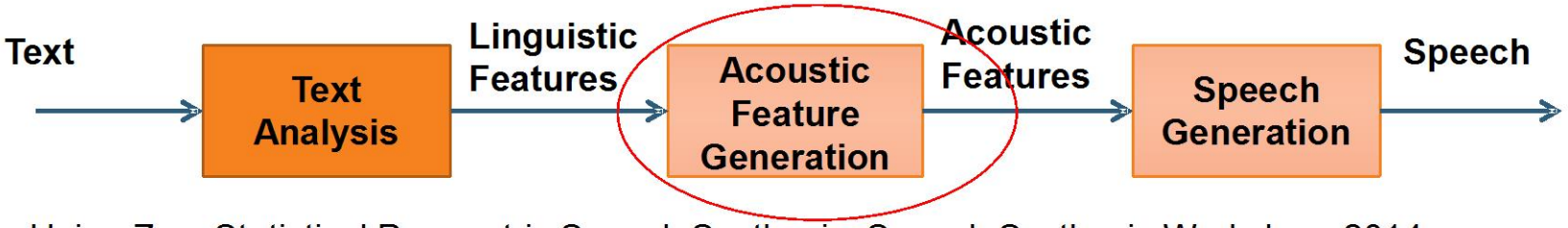


Copyright National University of Singapore

Deep Learning Method - DNN



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

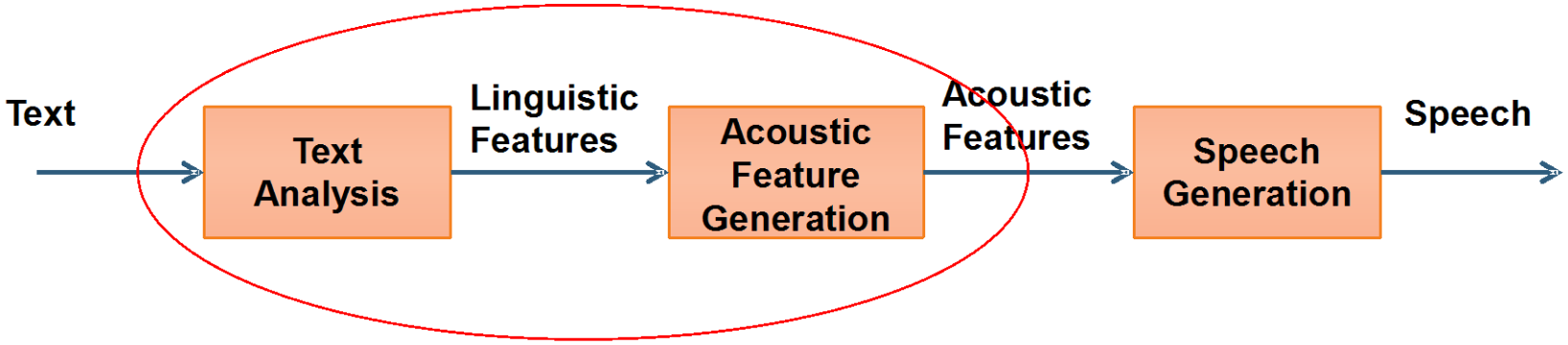


Heiga Zen, Statistical Parametric Speech Synthesis, Speech Synthesis Workshop, 2014.

Copyright National University of Singapore

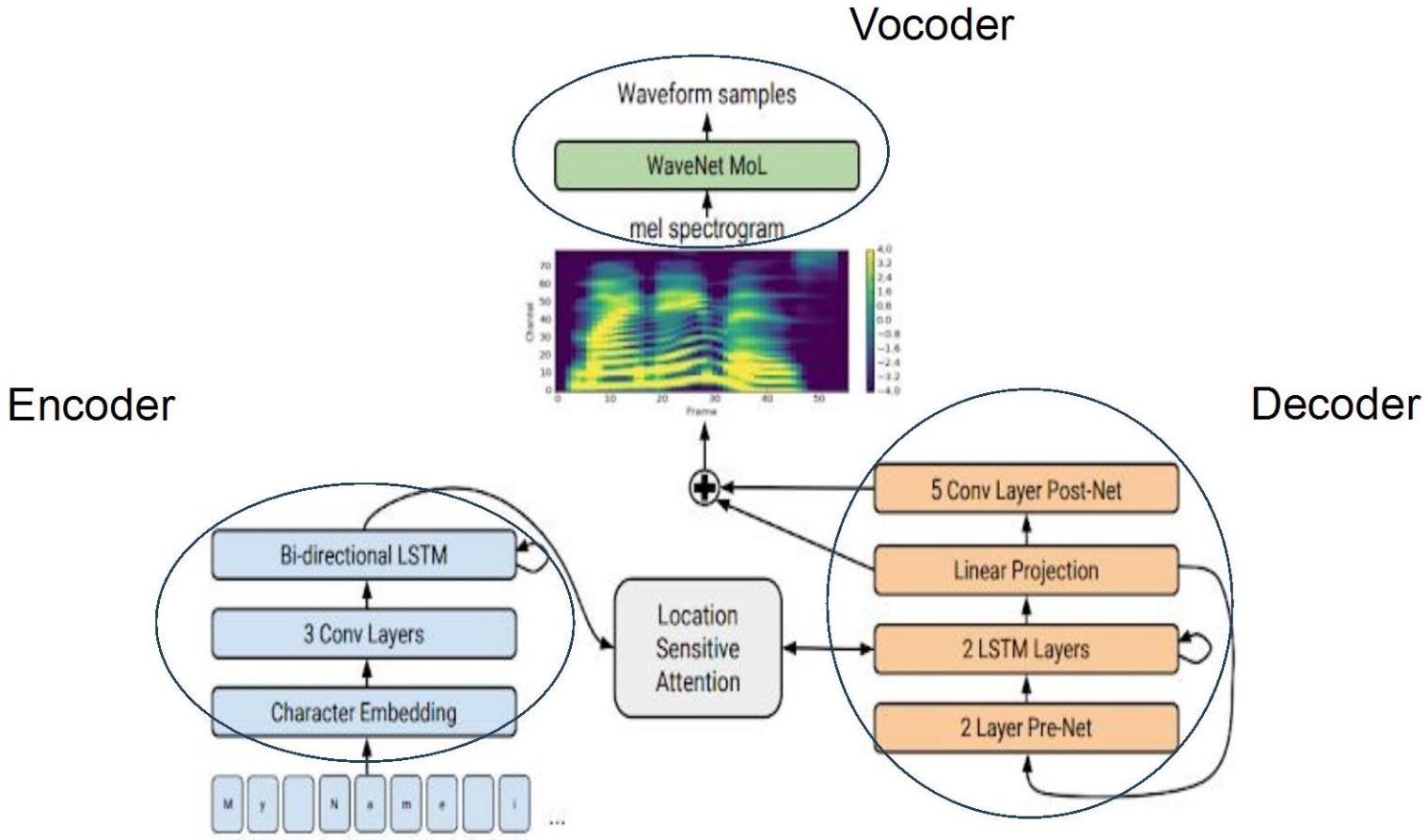
End-to-End Synthesis (e.g. Tacotron 2)

- **Tacotron 2**
  - 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.



Copyright National University of Singapore

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

Copyright National University of Singapore

End-to-end Synthesis (Tacotron 2)

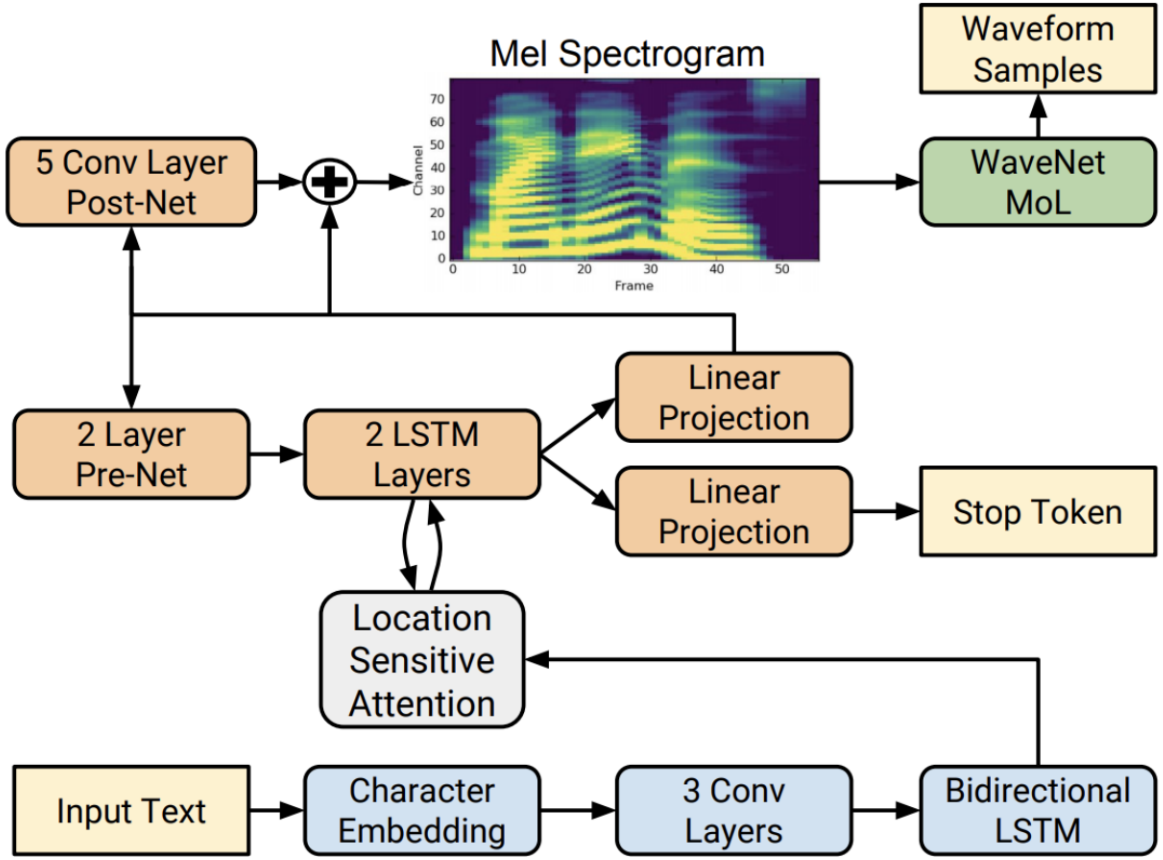


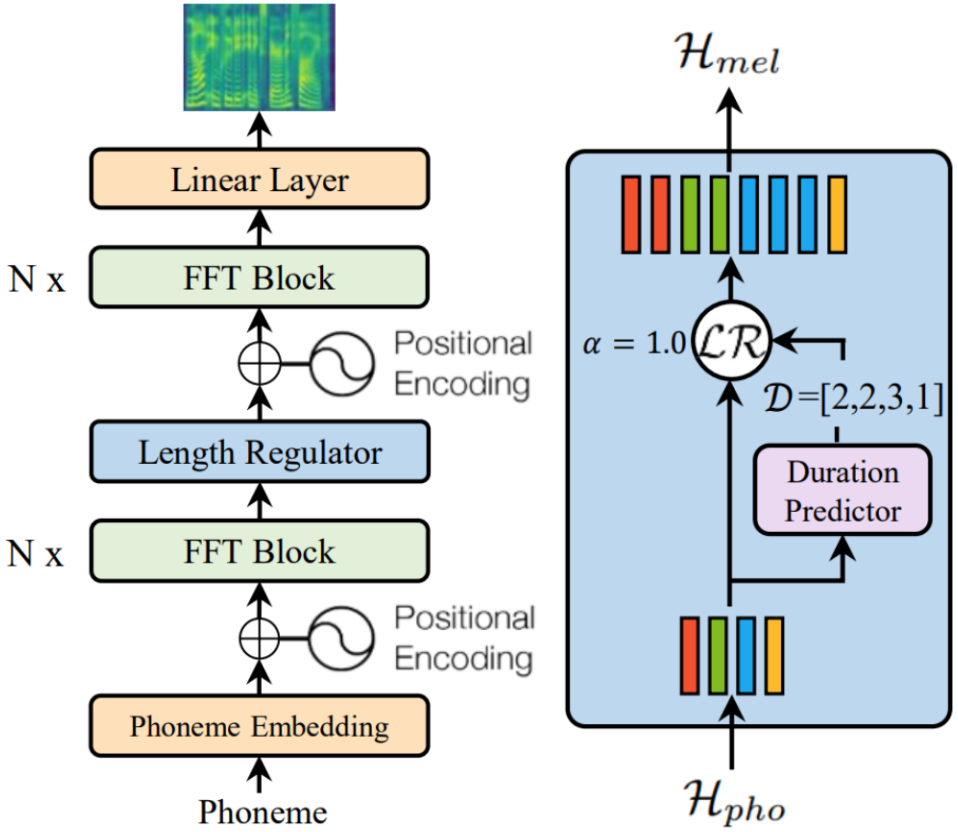
Fig. 1. Block diagram of the Tacotron 2 system architecture.

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

Copyright National University of Singapore

Transformer based

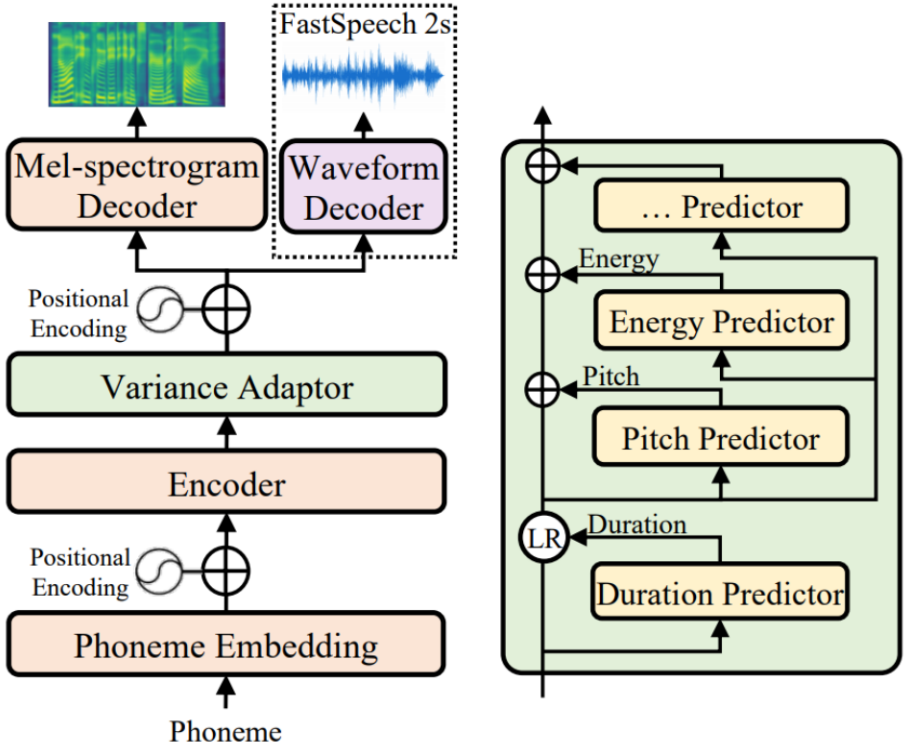
- **FastSpeech**
  - Generate mel-spectrogram in parallel (for speedup)
  - Remove the text-speech attention mechanism (for robustness)
  - Feed-forward transformer with length regulator (for controllability)



Copyright National University of Singapore

Transformer based

- **FastSpeech 2**
  - **Non-Autoregressive Approach**: faster and more efficient, suitable for real-time applications.
  - **Improved Prosody**: The model introduces mechanisms to better capture the prosody
  - **Variance Adaptor**: A novel component that predicts the variance of speech features, more expressive
  - **Multiple Heads Self Attention (MHSA):** capture relationships between different phonemes, leading to more accurate and natural-sounding speech.
  - **Improved Training Stability**: offers more stable training dynamics, ensuring consistent performance

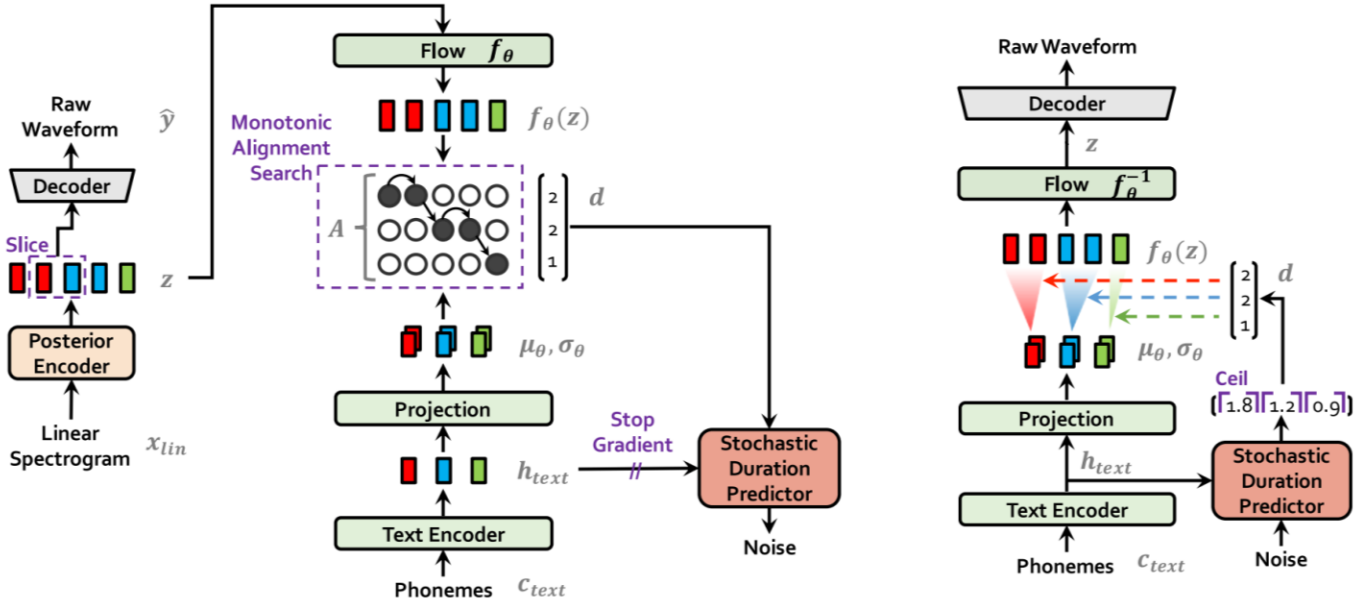


(a) FastSpeech 2 (b) Variance adaptor

Copyright National University of Singapore

Fully End-to-End TTS: VITS

- A combination of acoustic model and vocoder
- VAE: input mel, output waveform
- Flow for VAE prior
- GAN for waveform generation
- Monotonic alignment search



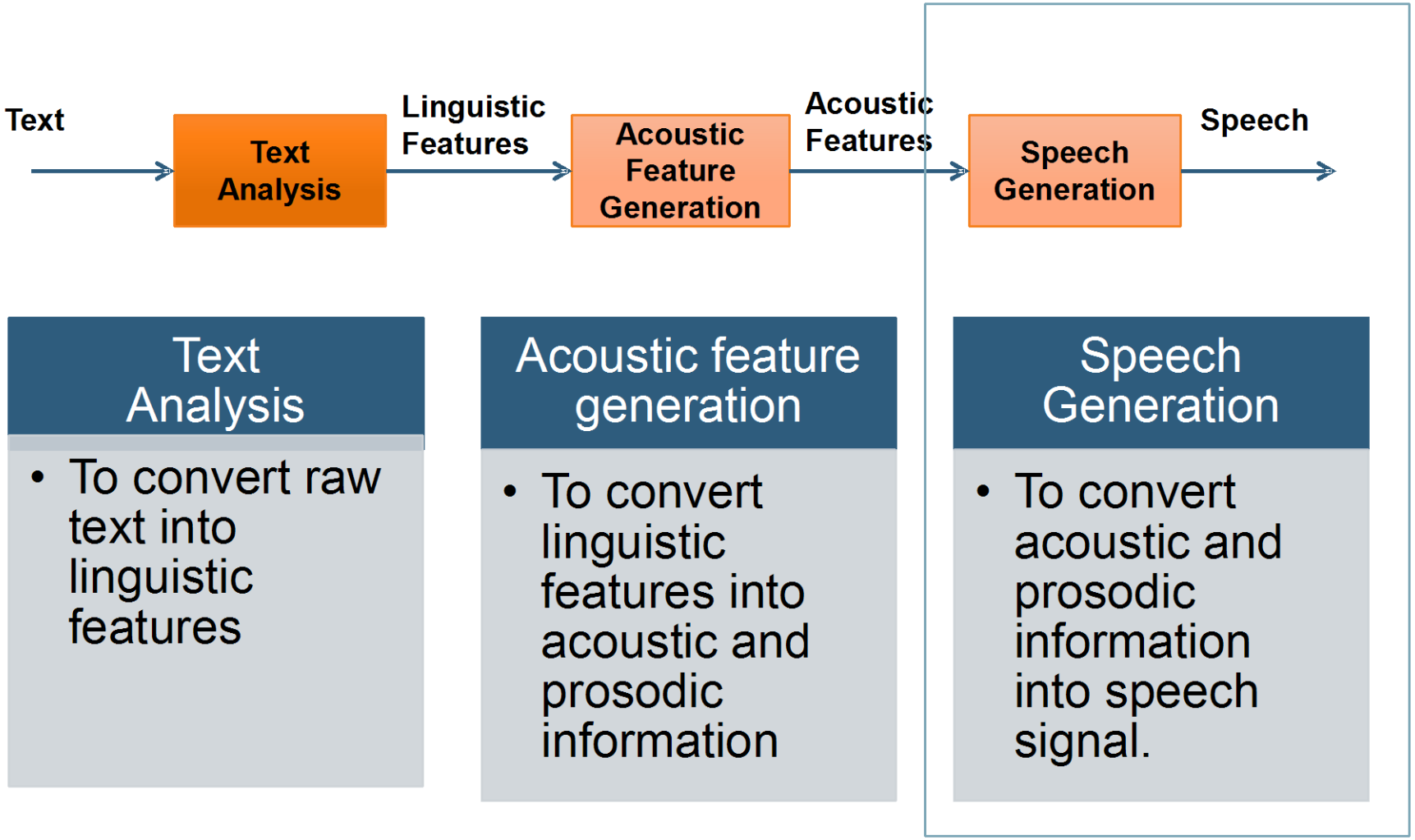
(a) Training procedure

(b) Inference procedure

Copyright National University of Singapore

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

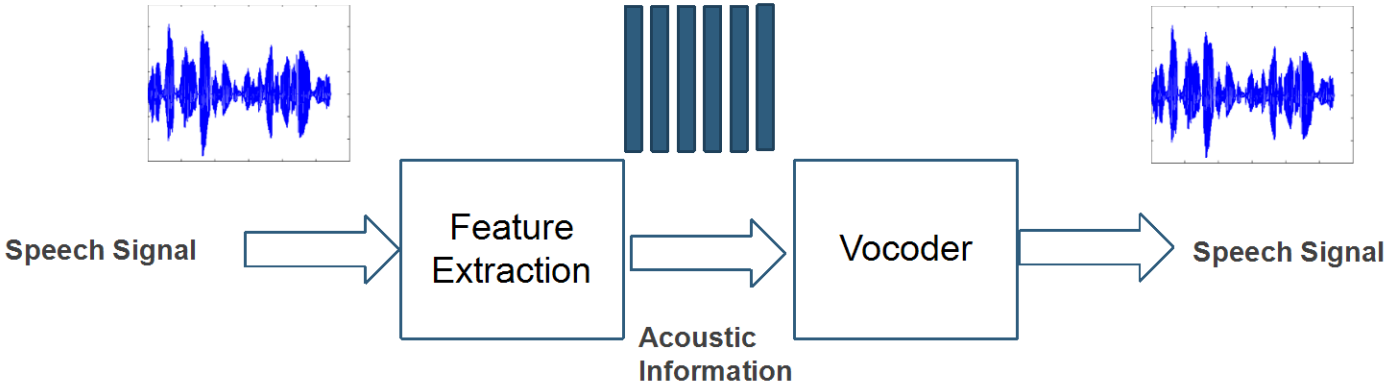
Speech Generation

- To convert acoustic and prosodic information into speech signal.

Copyright National University of Singapore

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

Copyright National University of Singapore

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.

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

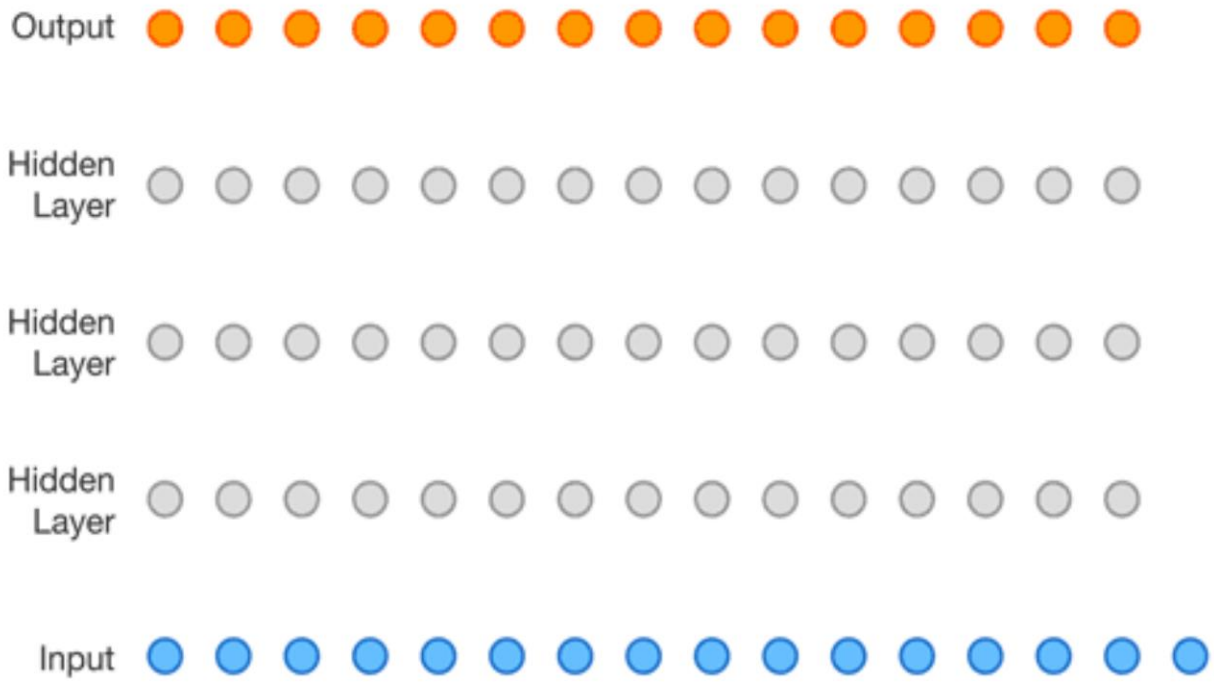
  - Computational expensive method – needs GPU support
  - Slow – take long time to generate voice.

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

Copyright National University of Singapore

Neural Vocoder

- **Wavenet**

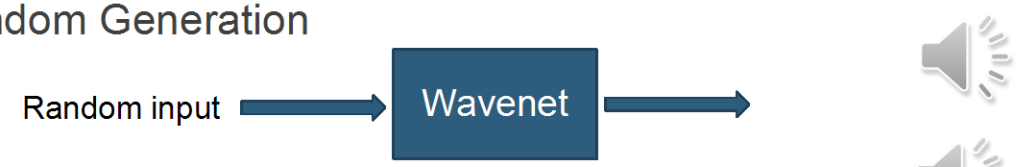


Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, Sep 2016

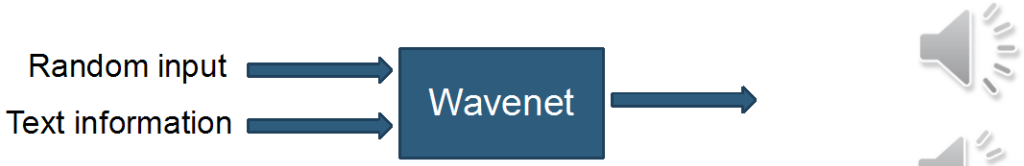
Copyright National University of Singapore

Neural Vocoder

- **Wavenet**
  - Random Generation



  - Conditional Generation



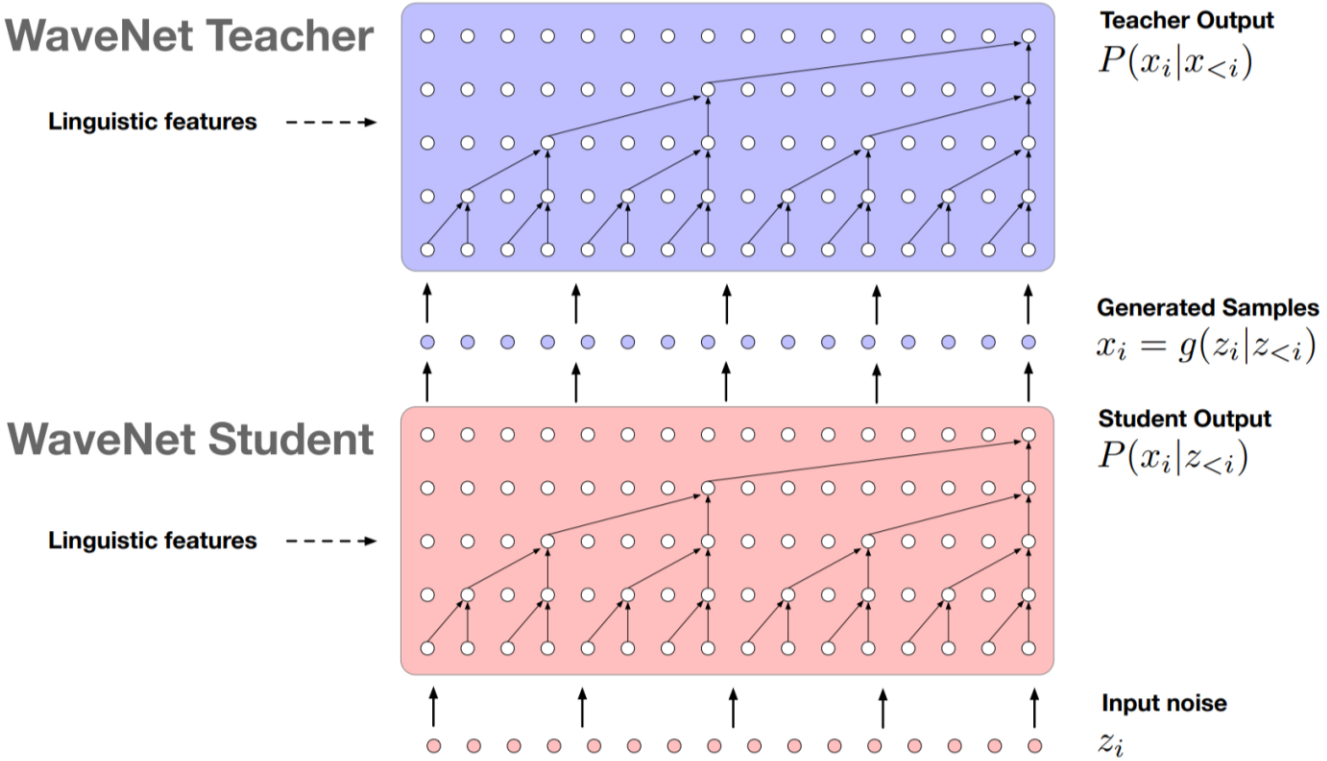
Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, Sep 2016

Copyright National University of Singapore

Vocoder—Flow: Parallel WaveNet

- **Flow:**
  - Map between data distribution x and standard (normalizing) prior distribution z

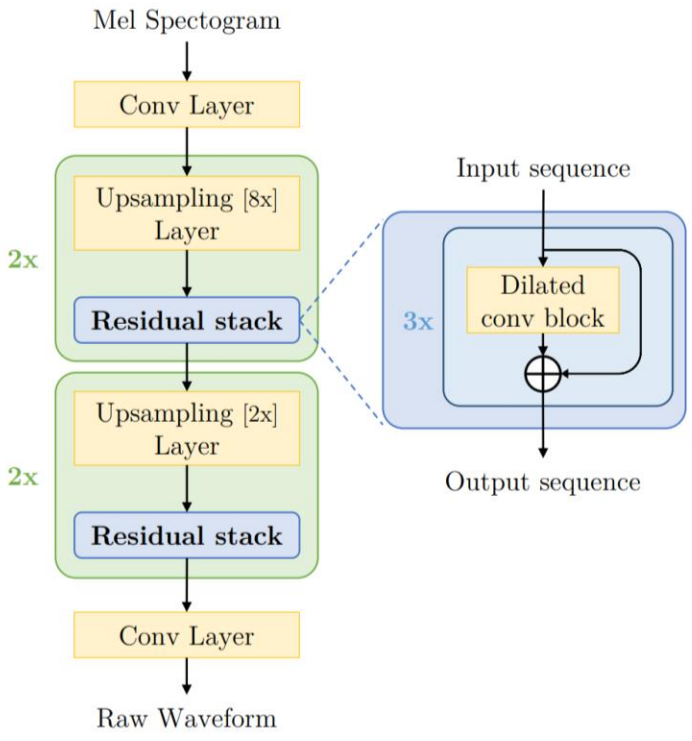
    Evaluation $z = f^{-1}(x)$ Synthesis $x = f(z)$
- **Knowledge distillation:**
  - First train a teacher model (autoregressive flow)
  - Train the student model (inverse autoregressive flow)) with the help of the teacher model
- **Combine the best of both worlds**
  - Parallel inference of IAF student
  - Parallel training of AF teacher



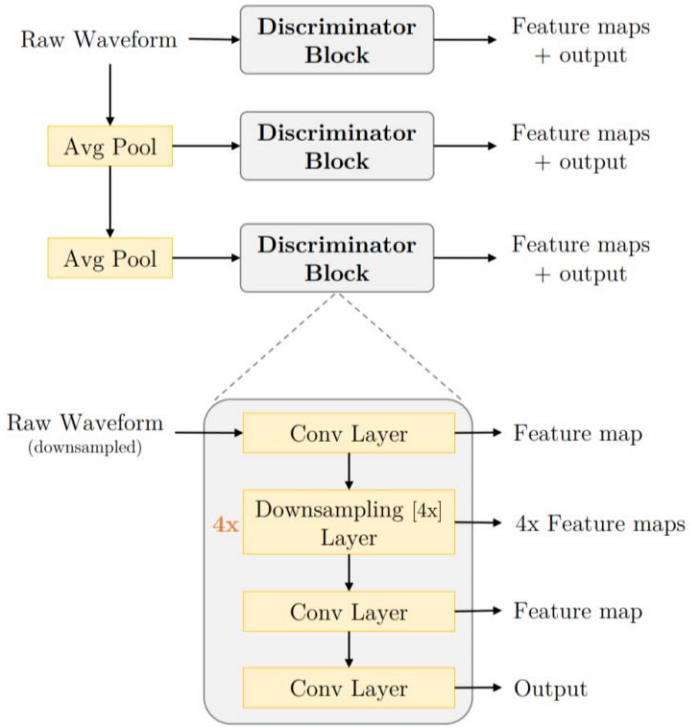
Copyright National University of Singapore

Vocoder——MelGAN

- **Generator:**
  - Transposed conv for upsampling
  - Dilated conv to increase receptive field
- **Discriminator:**
  - Multi-scale discrimination



(a) Generator



(b) Discriminator

Copyright National University of Singapore

Other Vocoders

- **Improvement on 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).

Copyright National University of Singapore

PART 4. ADVANCED TTS RESEARCH

Advanced Research Efforts

- Personalized Speech Synthesis (Voice Cloning)
- Low Resource Speech Synthesis
- Expressive Speech Generation
- Conversational Speech Generation
- Zero-shot Speech Synthesis
- Text Prompt Controllable Speech Synthesis
- Edit Speech via Text
- Speech Driven Talking Face Generation

Copyright National University of Singapore

Personalized Speech Synthesis (Voice Cloning)

- **To generate speech of a given speaker**
- **Sample speech data from the speaker are required.**
- **Solutions:**
  - Transfer learning (model adaptation)
  - Voice conversion
  - Few-shot and Zero-shot learning

Copyright National University of Singapore

Low-resource TTS

- **Demands:**
  - There are about 7,000 languages in the world, best commercialized speech services only support dozens of languages.
- **Challenge:**
  - Lack of data in low-resource languages and the data collection cost is very high.
- **Solutions:**
  - Self-supervised training: Text pre-training, speech pre-training, discrete token quantization
  - Cross-lingual transfer: Languages share similarity, phoneme mapping/re-initialization/IPA/byte
  - Cross-speaker transfer: Voice conversion, voice adaptation
  - Speech chain/back transformation: TTS ← ASR
  - Dataset mining in the wild: Speech enhancement, denoising, disentangling

Copyright National University of Singapore

Expressive TTS

- **Expressiveness**
  - Characterized by content (what to say),
  - Speaker/timbre (who to say),
  - Prosody/emotion/style (how to say),
  - Noisy environment (where to say), etc
- **Emotions:**
  - Happy, Sad, Angry, etc.

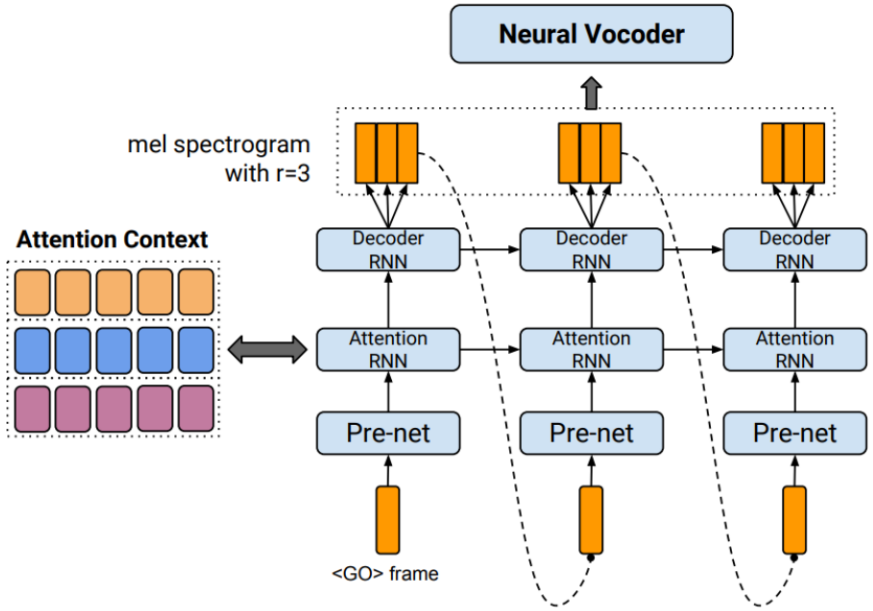
Text

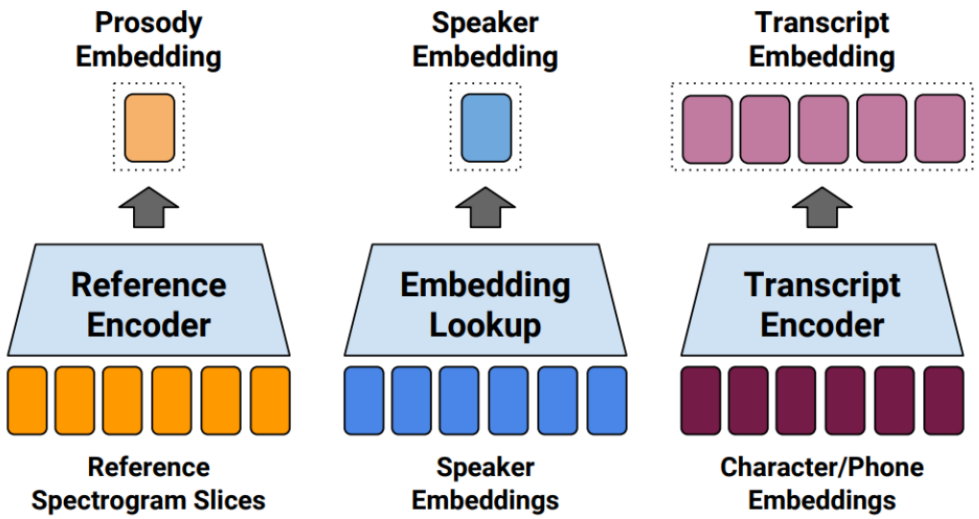
multiple speech variations
(duration, pitch, sound volume, speaker, style, emotion, etc)

Copyright National University of Singapore

Expressive TTS——Reference encoder

- **Prosody embedding from reference audio**

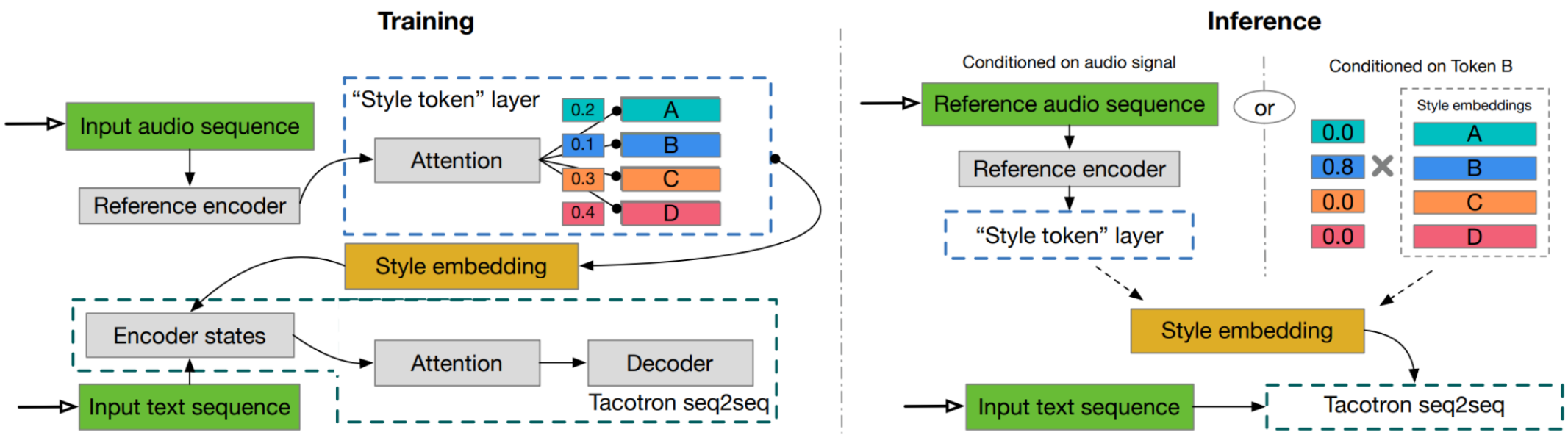




Copyright National University of Singapore

Expressive TTS——Reference encoder

- **Style tokens**
  - Training: attend to style tokens
  - Inference: attend to style tokens or simply pick style tokens



Copyright National University of Singapore

Conversational Speech Generation

Introducing super realistic AI voices optimized for conversations

By Melinda Ma

Published Sep 21 2023 01:48 AM 2,551 Views

Now, in human-bot conversational interactions, AI can produce more natural, fluent, and high-quality responses than ever before, thanks to the power of Large Language Models (LLMs) such as Azure OpenAI GPT. Consequently, when engaging in verbal conversations, the demand for naturalness and expressiveness in Text-to-Speech (TTS) voices is higher than ever. We are introducing these new voices specifically designed for conversational scenarios. Whether you are creating a speech-based chatbot, a voice assistant, or a conversational agent, these new voices will ensure your interactions are more realistic, lifelike, and engaging.

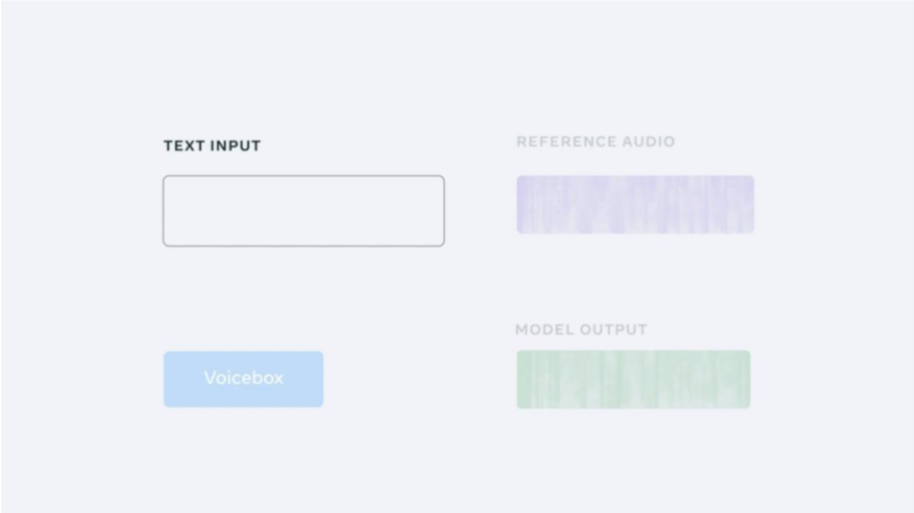
https://techcommunity.microsoft.com/t5/azure-ai-services-blog/introducing-super-realistic-ai-voices-optimized-for/ba-p/3933744

- Delightful TTS 2
  - Vector quantized auto-encoders with adversarial training (VQ-GAN)
  - Jointly optimize the acoustic model and the vocoder
- Mulan TTS
  - Contextual and emotion encoders to enrich beyond sentences for long-form prosody and dialogue expressiveness.
  - Apply denoise algorithms and long audio processing corpora
- Conversational Data

Copyright National University of Singapore

Zero-Shot Speech Synthesis

- **Input:**
  - Text to synthesize
  - Reference speech any audio style
- **Output:**
  - speech that sounds coherent to the reference in every aspects, including voice, background noise, and speaking style.



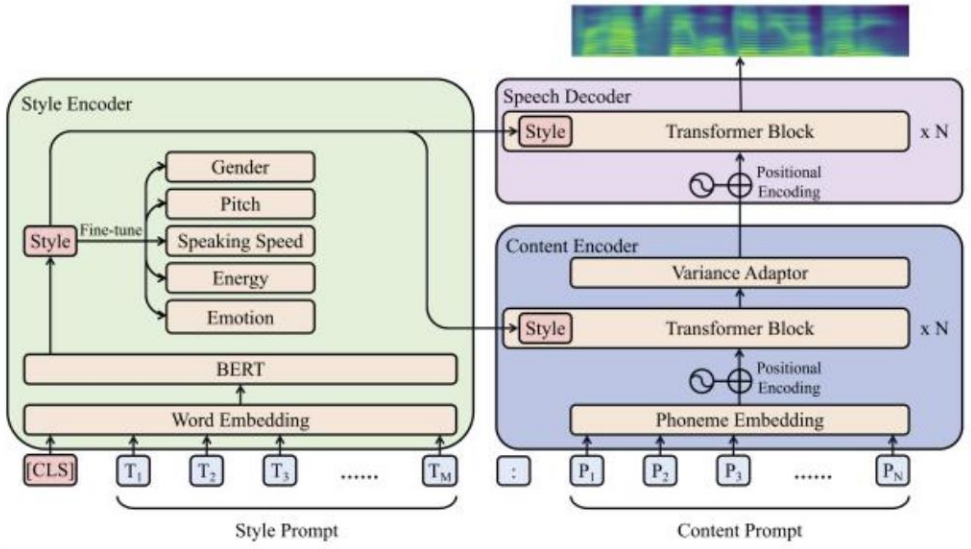
https://voicebox.metademolab.com/zs_tts.html

Copyright National University of Singapore

Text Prompt Controllable TTS

- **Use text prompt with both style and content descriptions as input to synthesize the corresponding speech.**

Style Prompt	Content Prompt
I need a man with a deep and loud voice to talk cheerfully:	No important letter come in a parcel, is there?
Please ask a girl to shout loudly with a sharp voice:	But folks hereabouts don't like him.
A lady whispers to her friend slowly:	Everything will go fine, right?

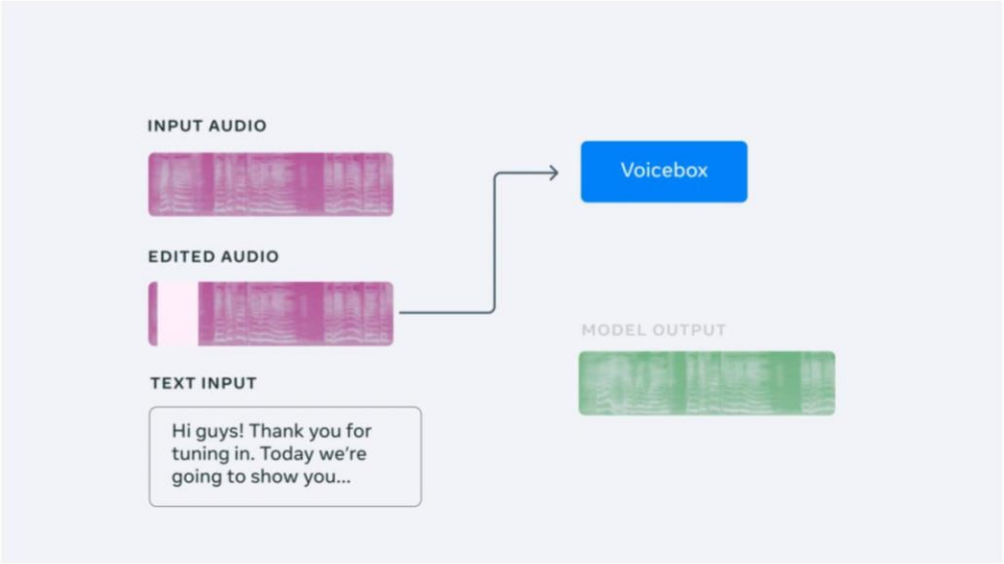


https://speechresearch.github.io/promptttts/

Copyright National University of Singapore

Editing Speech via Text

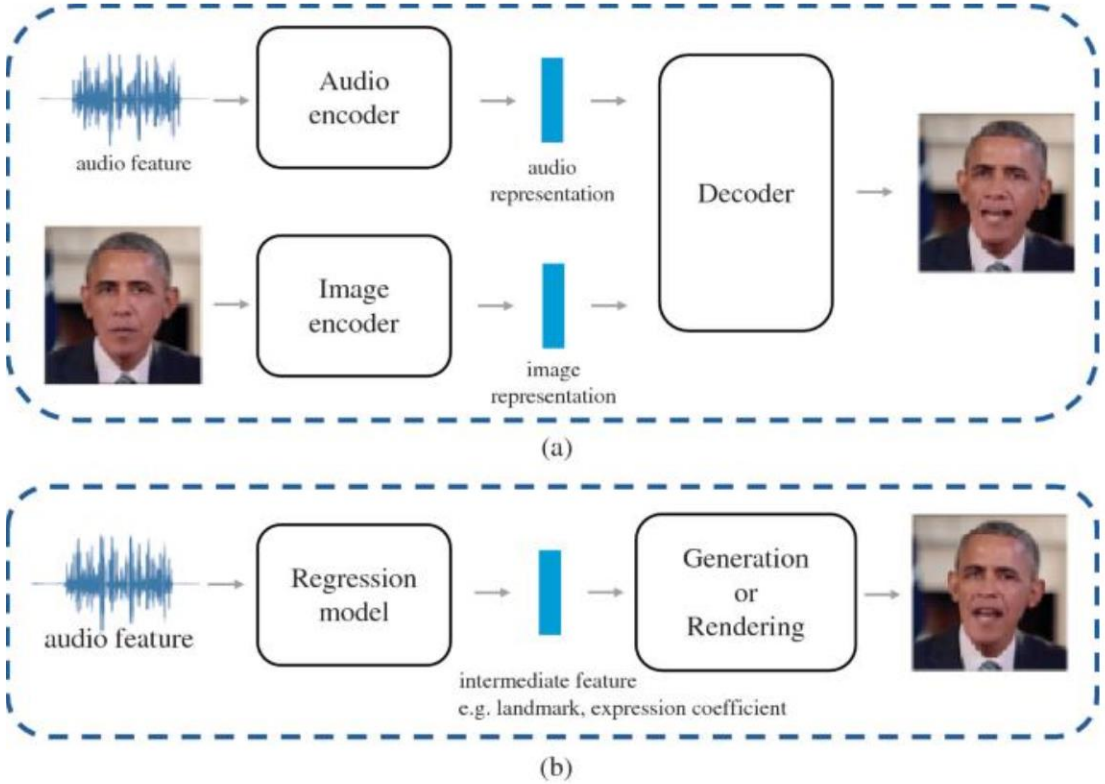
- **Speech recording may contain errors.**
  - Use speech recognition to match speech and text
  - Edit text
  - Regenerate the new text with the same voice.



https://voicebox.metademolab.com/edit.html

Copyright National University of Singapore

Speech Driven Talking Face



Talking Faces: Audio-to-Video Face Generation, Yuxin Wang, Linsen Song, Wayne Wu, Chen Qian, Ran He & Chen Change Loy, Handbook of Digital Face Manipulation and Detection, Springer 2022.

Copyright National University of Singapore

PART 5. RESOURCE AND PROGRAMMING

Open Source Packages

- **Traditional Systems:**
  - Festival: Traditional speech synthesis toolkit. Good examples for all the processing steps.
  - HTS: Traditional parametric system. A good example of parametric methods.
  - Merlin: Pipeline deep learning system
- **End-to-End Systems:**
  - Coqui TTS: TTS package supporting the latest acoustic models and vocoders; support multiple speakers and multiple languages, etc
  - TensorflowTTS: Tensorflow-based TTS package supporting acoustic models and vocoders.

Copyright National University of Singapore

Commercial TTS systems

- **Google Cloud**
  - https://cloud.google.com/text-to-speech/
- **Microsoft Azure**
  - https://azure.microsoft.com/en-us/products/cognitive-services/text-to-speech
- **AWS, Baidu, etc.**

Copyright National University of Singapore

Hands-on Session

- **Open source pytts3**
  - pyttsx3 is a text-to-speech conversion library in Python.
  - https://pypi.org/project/pyttsx3/
  - Install: pip install pyttsx3
  - 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)

Copyright National University of Singapore

Sample Program (property and synthesis)

```
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()
```


Copyright National University of Singapore

Hands-on Session

Sample Program (Handling events)

```
#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.'
```


Copyright National University of Singapore

Hands-on Session

Sample Program (Handling events)

```
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)
```


Copyright National University of Singapore

Topic 5: Voice Conversion and Audio Generation

Voice Conversion

- Purpose:
  - Primarily aims to modify an existing voice to resemble another while retaining the original content and linguistic features.
- Input and Output:
  - Takes a source voice as input
  - Produces an output that mimics a target voice.
- Use Cases:
  - Speaker imitation, accent conversion, emotion/style transfer.

.

Copyright National University of Singapore

Approaches for Voice Conversion

- CycleGAN-VC:
  - Utilizes adversarial training to learn mappings between different voice domains without paired data
  - Effectively translates voice characteristics from one speaker to another.
  - Audio samples at https://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/cyclegan-vc/

- StarGAN-VC:
  - An extension of StarGAN that enables multi-speaker voice conversion.
  - Operates in a single model, allowing conversions between multiple target voices based on a single set of training data.
  - Audio sample at https://www.kecl.ntt.co.jp/people/kameoka.hirokazu/Demos/stargan-vc/

Copyright National University of Singapore

Audio Generation

- Purpose:
  - Focuses on creating audio from scratch, whether it's speech, music, or sound effects, often WITHOUT a predefined input voice.
- Input and Output:
  - May take various forms of input (e.g., text, noise)
  - Entirely new audio signals as output.
- Use Cases:
  - TTS, music composition, sound effect generation, speech enhancement

Copyright National University of Singapore

State-of-the-Art Approaches

- AudioLDM2
  - Generating high-quality audio based on text prompts or other audio inputs.
  - Gradually transforms noise into coherent audio by iteratively refining the signal (diffusion-based approach)
  - Audio samples at https://audioldm.github.io/audioldm2/

Copyright National University of Singapore

Summary of Voice Conversion and Audio Generation

	Voice Conversion	Audio Generation
Goal	Transform a voice to sound like a different speaker.	Generate new audio (speech, music, sound effects) from input data.
Input/Output	Input: Existing speech signal. Output: Modified speech with a different voice but same content.	Input: Text, musical notes, sound description, etc. Output: Newly generated audio content.
Content Generation	Does not generate new content (content-preserving).	Generates new content (speech, music, or other audio types).
Applications	Speaker imitation, accent conversion, emotion/style transfer.	TTS, music composition, sound effect generation, speech enhancement.

Copyright National University of Singapore

Topic 6: Spoken Dialogue Processing

PART 1. INTRODUCTION

Examples of Spoken Dialogue

- Speech Assistant:
  - Amazon Echo
  - Google Home
  - Siri

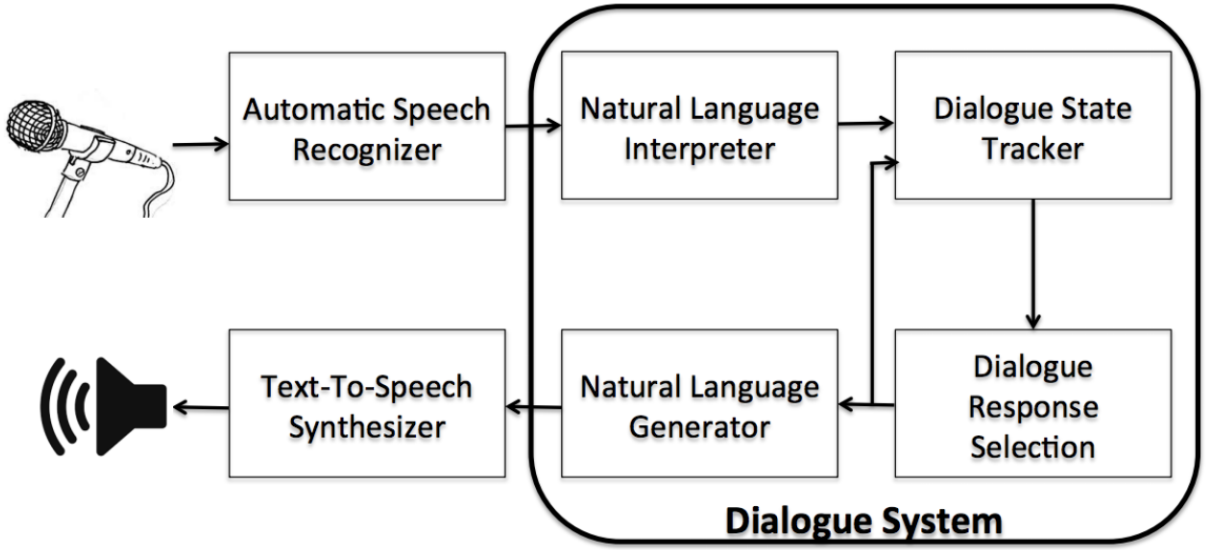
- Robots
  - Social robot

- In Car
  - Navigation and information

Copyright National University of Singapore

Spoken Dialogue System (SDS)

- Speech recognizer: to listen to human voice
- Speech synthesizer: to speak to human
- Dialogue system: to understand the language and respond



Copyright National University of Singapore

PART 2. CONSIDERATIONS IN BUILDING SDS

Purpose and Scope

- **Domain:**
  - Is your system domain-specific (e.g., booking flights) or open-domain (e.g., general conversation)?
- **Functionality:**
  - What tasks should the system perform? E.g., information retrieval, transaction processing, or simple chit-chat.
- **User Base:**
  - Who are the target users? This can influence the language, tone, and complexity of the dialogue.

Copyright National University of Singapore

ChatGPT vs DialogFlow

- ChatGPT (Generative Chatbot)
  - Produce conversational response based on a large language model.
  - Flexibility: Capable of handling a wide variety of topics and conversational contexts due to its ability to generate novel sentences.
  - Contextual Understanding: Can maintain context over multiple turns in a conversation, providing more coherent and relevant responses.

- DialogFlow (Task-oriented Chatbot)
  - Detect **intents** and extract **entities** from user queries to map them to pre-defined responses or backend integrations. Reference: BERT for Joint Intent Classification and Slot Filling by Chen et al. (2019).
  - Predefined Paths: Conversations follow specific pathways based on user inputs, ensuring that the dialogue adheres to a planned structure.
  - Task-Oriented: Often designed for specific applications, like customer support or booking systems, where the flow can be clearly defined.
  - State Management: Maintains context and state through the conversation, allowing it to guide users through a defined process.

Copyright National University of Singapore

ChatGPT vs DialogFlow

Aspect	ChatGPT	Dialogflow
Type of AI System	Generative Conversational AI / LLM	Task-Oriented Conversational AI / NLU Framework
Response Generation	Generative, open-ended responses based on context and large-scale data	Intent-driven, rule-based responses mapped to predefined structures
Conversational Flexibility	Flexible, suitable for open-domain and multi-turn conversations	Structured, suitable for task-specific and guided dialogues
Use Cases	General-purpose chat, creative writing, Q&A, coding help, open-domain AI	Customer service bots, booking systems, voice assistants, FAQ bots
Integration & Deployment	API-based integration, broad applications across domains	Integrates with Google Cloud, deployable on various channels

Copyright National University of Singapore

Speech Recognition

- **Automatic Speech Recognition (ASR):**
  - This converts spoken language into text. Consider factors like accuracy, real-time processing, and noise handling.
- **Acoustic Models:**
  - These are trained on large datasets to recognize phonetic patterns in speech. Models are trained on data similar to the target applications.
- **Language Models:**
  - These predict the likelihood of a sequence of words. Similar language context.
- **Preprocessing component**
  - Make sure preprocessing component delivers correct speech signal.

Copyright National University of Singapore

Speech Synthesis

- **Text-to-Speech (TTS):**
  - Convert system responses back into spoken language. Consider the naturalness and intelligibility of the synthesized voice.
- **Voice Selection:**
  - Choose a voice that fits the application's context and user base.

Copyright National University of Singapore

User Experience (UX) and Design

- **Prompt Design:**
  - How the system initiates and guides the conversation.
- **Error Handling:**
  - How the system responds to unrecognized or misunderstood inputs.
- **Turn-taking:**
  - Managing interruptions and overlaps in speech.
- **Feedback:**
  - Providing auditory cues (like a beep) to indicate the system is listening or processing.

Copyright National University of Singapore

Evaluation and Testing

- **Usability Testing:**
  - Engage real users to test the system's functionality and UX.
- **Performance Metrics:**
  - Measure ASR accuracy, NLU accuracy, response time, etc.
- **Continuous Improvement:**
  - Regularly update the system based on user feedback and error analysis.

Copyright National University of Singapore

Scalability and Deployment

- **Infrastructure:**
  - Ensure the system can handle the expected number of users and scale if needed.
- **Integration:**
  - How will the SDS integrate with other systems or databases?

Copyright National University of Singapore

Multimodality and Fallback

- **Multimodality**
  - Consider if the system will integrate other modes of interaction, like touch or visual displays.
- **Fallback Strategies**
  - Have a strategy for when the system can't understand or process a user's request.
  - This might involve redirecting to a human operator or asking the user to rephrase.

Copyright National University of Singapore

PART 3. DIALOG FLOW

Dialogflow

- **What is Dialogflow**
  - A Google Cloud product.
  - A natural language understating platform to build conversational user interface.
  - Both input and output can be text or speech.
- **Google Cloud**
  - It contains quite a lot of products such as NLP, Computer Vision, Machine Learning, Big data, etc.

Copyright National University of Singapore

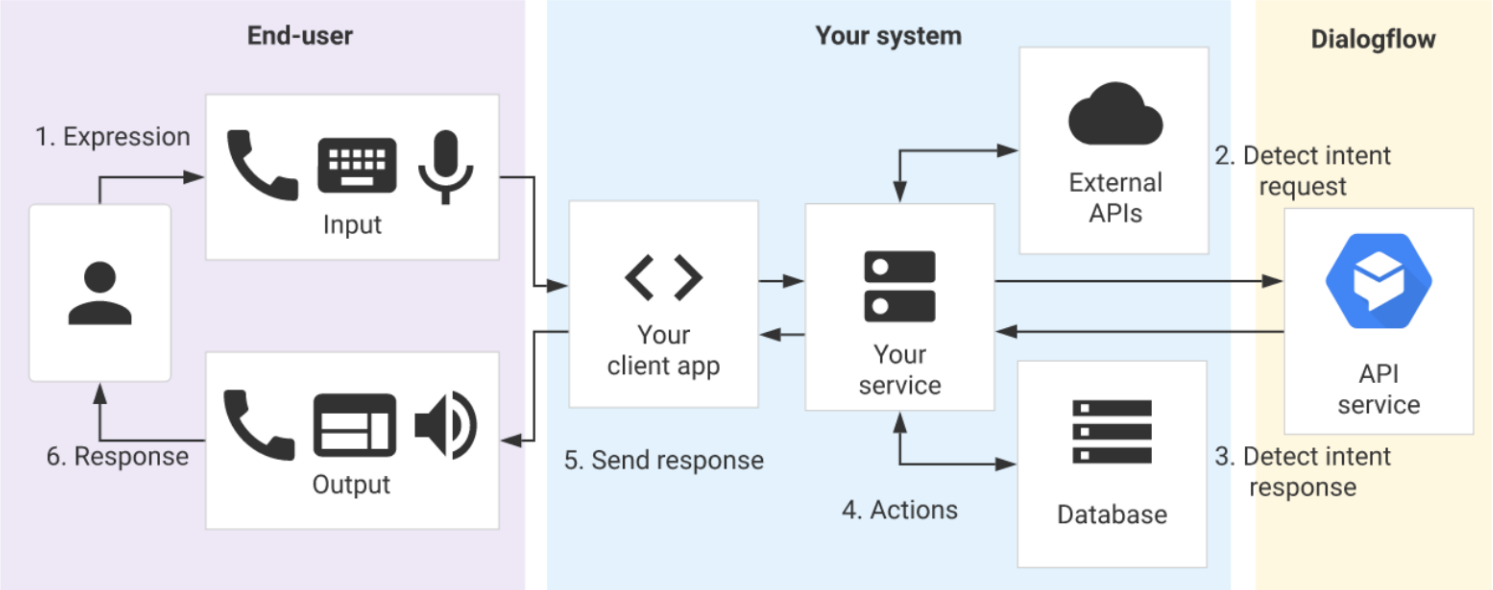
Dialogflow concepts

- **Agent:**
  - a module to handle the conversation with end user.
- **Intent detection:**
  - to categorize end-user intentions for each conversation turn
- **Entity recognition (Slot filling):**
  - to identify and extract specific data from end-user expressions. E.g. dates, times, colors, email addresses.

Copyright National University of Singapore

Dialogflow with Speech

- **Dialogflow supports speech input and output. Both input and output can be configured to use text or speech.**
- **DialogFlow support APIs.**
- **Two ways for speech input:**
  - Input file: pass the full recording to speech engine
  - Streaming: pass the recording little by little.



Copyright National University of Singapore

PART 4. WORK WITH DIALOGFLOW

Tasks

- **Create a dialog:**
  - To build a dialog demo that support booking a meeting room, the required information includes: Date, Time, Duration, Location, Room Name (A, B, C)
  - We will load the predefine dialogue from file.
- **Build spoken dialog with APIs.**
  - Using APIs to support speech recognition and speech synthesis.

Copyright National University of Singapore

Dialogflow with Speech API

- **Create Agent**
  - dialogflow.cloud.google.com
  - You need to have a google account
  - Go to Dialogflow Console
  - Create an agent (Name, Language, Timezone)
  - Project automatically created. Keep a record of the Project ID
  - Import an existing project (RoomReservation.zip)

Copyright National University of Singapore

Create Intent

- **Create Intent**
  - If we would like to create an intent from scratch (instead of importing)
  - Please refer to https://cloud.google.com/dialogflow/cx/docs/quick/build-agent

Copyright National University of Singapore

Dialogflow with Speech API

- **Create Service Account Key**
  - https://console.cloud.google.com/apis/credentials/serviceaccountkey
  - Select the project
  - From the **Service account** list, select **Create service account.**
  - In the **Service account name** field, enter a name, click **Create and continue**
  - From the **Role** list, select **Project** > **Owner**, click **Continue**, then click **Done**
  - Click three dots, select **manage keys**.
  - Select **Add key**, select **Create new key,** Key type: select **JSON**
  - Click **Create** and save the generated file.

Copyright National University of Singapore

Dialogflow with Speech API

- **To use with API:**
  - Install package on computer: pip install dialogflow
- **Samples:**
  - Speech input from file: test-asr.py

    https://cloud.google.com/dialogflow/es/docs/how/detect-intent-audio
  - Speech output to file: test_tts.py

    https://cloud.google.com/dialogflow/es/docs/how/detect-intent-tts
  - Speech input by Streaming: test-steam.py

    https://cloud.google.com/dialogflow/es/docs/how/detect-intent-stream

Copyright National University of Singapore

Further Reading

- Dialogflow
  - https://dialogflow.cloud.google.com/
  - https://chatimize.com/dialogflow-tutorial/
- Google Dialogflow Place calls on your chatbot | IVR
  - https://www.youtube.com/watch?v=eyRXtsEEDJY
- Google Cloud Tech: Integrate Dialogflow with Telephony Gateway
  - https://www.youtube.com/watch?v=whZPw0HFnTE
- Voximplant: Convert your Dialogflow Chatbot into a Voicebot (can use 3rd party TTS)
  - https://www.youtube.com/watch?v=2kp5haQ7zNQ
- Alexa Skills Kit
  - https://developer.amazon.com/en-US/alexa/alexa-skills-kit
  - https://www.youtube.com/watch?v=EqawCvlmnH4

Copyright National University of Singapore

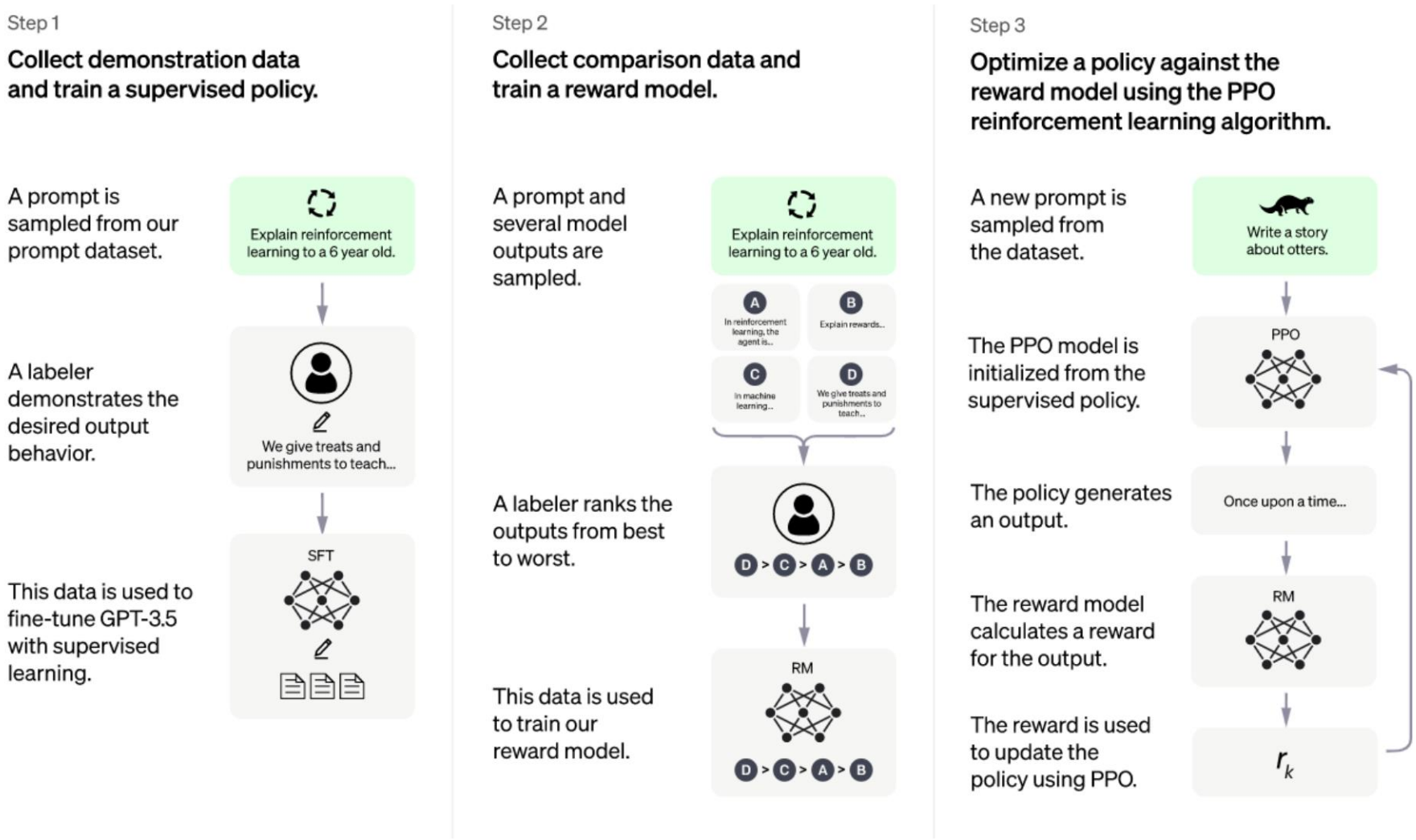
PART 5. CHATGPT AND OPENAI API

ChatGPT

- Launched by OpenAI on 30 Nov 2022.
- Can perform many text generation tasks through multi-round conversations.
- Generates high-quality answers.
- ChatGPT API launched on 1 March 2023

Copyright National University of Singapore

ChatGPT – Training process



Copyright National University of Singapore

ChatGPT's Voice and Image Capability

ChatGPT can now see, hear, and speak

We are beginning to roll out new voice and image capabilities in ChatGPT. They offer a new, more intuitive type of interface by allowing you to have a voice conversation or show ChatGPT what you're talking about.

OpenAI announced on 25 Sep 2023 (Monday)

Copyright National University of Singapore

ChatGPT's new speech ability

- **Speak with ChatGPT and have it talk back**
  - https://openai.com/blog/chatgpt-can-now-see-hear-and-speak

- **Text-to-Speech**
  - The new voice technology—capable of crafting realistic synthetic voices from just a few seconds of real speech—opens doors to many creative and accessibility-focused applications.

Copyright National University of Singapore

OpenAI APIs

- **Key Concepts:**
  - Prompt: User input
  - Completion: text that matches the user input.
  - Token: words or chunks of characters.
- **Prompt design:**
  - Give accurate instructions,
  - Give good examples,
  - Or a combination of the two.

Copyright National University of Singapore

OpenAI APIs

- Text Completion: multiple tasks
- Code Completion: code generation
- Chat Completion: multiple tasks (better support for chat)
- Image Generation:
- Fine Tuning: model tuning
- Embeddings: to convert text into a vector representation
- Speech-to-Text: speech recognition and translation
- Moderation: detect whether text may be sensitive or unsafe

Copyright National University of Singapore

Program with OpenAI APIs

1. Obtain an API key:

1. Log into http://openai.com
2. Click on “Personal”, then click “API keys”
3. Click “Create new secret key”, and save the generated key to a file.

2. Install the SDK:

1. pip install openai

Copyright National University of Singapore

Text Completion

- **Features of Text Completion API:**
  - Generation: story ideas, business plans, marketing slogans, etc.
  - Conversation: To chat with humans.
  - Transformation: Translation, conversions, summarization, completion
  - Factual responses: respond to factual questions.
  - Inserting text: providing both prefix and suffix.
  - Editing text: give input, and instruction, it will edit text.

Copyright National University of Singapore

OpenAI API - PlayGround

- It's simply a text box where you can submit a prompt to generate a completion.
- https://platform.openai.com/playground/
- To see the meanings of each parameters.
- To test APIs by changing parameters.

Copyright National University of Singapore

Text Completion API

- **Text Completion Example**

```
# Now, let's generate some text
import openai
openai.api_key = “xxxxx”

prompt = (f"Write an article on ChatGPT API")

completions = openai.Completion.create(
engine="text-davinci-003",
prompt=prompt,
max_tokens=1024,
n=1,
stop=None,
temperature=0.5,
)

message = completions.choices[0].text
print(message)
```


Copyright National University of Singapore

Chat Completions (ChatGPT)

- **ChatGPT is powered by gpt-3.5-turbo**
- **Supported tasks:**
  - Draft an email or other piece of writing
  - Write Python code
  - Answer questions about a set of documents
  - Create conversational agents
  - Give your software a natural language interface
  - Tutor in a range of subjects
  - Translate languages
  - Simulate characters for video games and much more

Copyright National University of Singapore

Chat Completions (ChatGPT)

- **Example of Chat Completion**

```
import openai

openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "Who won the world series in 2020?"},
        {"role": "assistant", "content": "The Los Angeles Dodgers won the World Series in 2020."},
        {"role": "user", "content": "Where was it played?"}
    ]
)
```


Copyright National University of Singapore

Speech-to-Text Processing

- **Speech Recogntion**

```
import openai
audio_file= open("/path/to/file/audio.mp3", "rb")
transcript = openai.Audio.transcribe("whisper-1", audio_file)
```

- **Speech Translation**

```
import openai
audio_file= open("/path/to/file/german.mp3", "rb")
transcript = openai.Audio.translate("whisper-1", audio_file)
```


Copyright National University of Singapore

NUS National University of Singapore | ISS

facebook.com/iss.nus

instagram.com/iss.nus

linkedin.com/company/iss_nus

youtube.com/@nus-iss

Thank you

www.iss.nus.edu.sg