Tampere University

Speech synthesis

SGN 14007

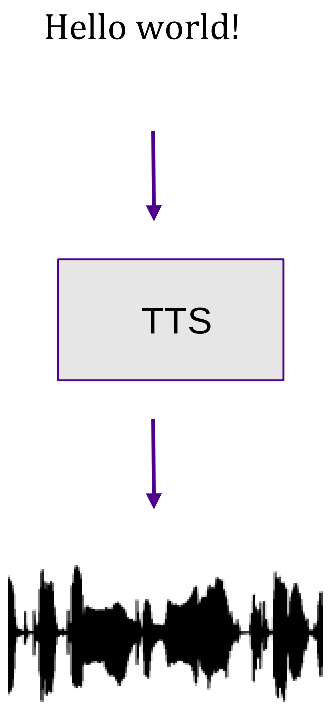
Lecture 10

Annamaria Mesaros

Text-to-speech (TTS)

- Text-to-speech (TTS) synthesis:
  - Generation of an acoustic speech signal based on ANY given text using a computer.

- ’Speech synthesis’ can also refer to other kind of speech waveform generation:
  - For example speech codec generates speech waveform based on speech parameters – this can be called speech synthesis (but not TTS).



Applications

- Assistive technology:
  - Screen readers for visually impaired
  - Google translate “listen” button
  - Voice output for people with speech impairment
- Entertainment (games and animation)
- Automatic announcements (e.g. in train)
- Human machine interaction when combined with speech recognition
  - Personal assistants (e.g. Siri)

Synthesis quality

- Quality of synthesized speech can be divided into:
  - Speech intelligibility (Can the listener understand what is spoken?)
  - Speech naturalness (Does the synthetic speech sound like human speech?)
- Different applications focus on different aspects:
  - E.g. intelligibility and high speaking rate are the key things in the screen readers for the visually impaired.
- The ideal speech synthesizer is both natural and intelligible. Speech synthesis systems usually try to maximize both characteristics.

Synthesis quality

“EASY” TASK

- **Intelligibility**: Can the listener decode the message from speech?
  - Relatively easy to solve: No big improvements since 1970’s (Taylor 2009).
  - Speech intelligibility can be evaluated by a listening test: How well listeners detect different phonemes (e.g. different consonants) in synthetic speech?
  - Longer-term intelligibility is evaluated by using synthetic utterances: Errors in single speech sounds do not necessarily affect the intelligibility.

DIFFICULT TASK

- **Naturalness:** Does the synthetic speech sound like human speech?
  - Non-human artifacts (pops, clicks, etc.) make synthesis unnatural.
  - Acceptable speech prosody and right variation in the realization of individual speech sounds are required.
  - Speaker identity: A synthesizer is required to sound like someone - a real or an artificial person.

Text-to-speech system

TEXT ANALYSIS
- Document structure detection
- Text normalization
- Linguistic analysis

PHONETIC ANALYSIS
- Homograph disambiguation
- Morphological analysis
- Grapheme-to-phoneme conversion

PROSODIC ANALYSIS
- Pitch and duration attachment

Frontend

SPEECH SYNTHESIS
- Voice rendering

Backend

Text analysis

- Let us have the following input text in our TTS system:

"THE LECTURE TAKES PLACE ON THU 28/11/2019. THE LECTURE STARTS AT 10:00 AM."

  - What should we do first?

  - Sentence 1: THE LECTURE TAKES PLACE ON THURSDAY TWENTY-EIGHTH OF NOVEMBER TWO-THOUSAND-NINETEEN.
  - Sentence 2: THE LECTURE STARTS AT TEN A-M.

- Required steps:
  - **Structure detection:** Sentence breaking and paragraph segmentation.
  - **Text normalization:** Processing of abbreviations, numbers, symbols etc.
  - **Linguistic analysis:** Finding larger syntactic units and semantics of words, phrases, clauses, and sentences.

Tampere University

Text analysis: Normalization

- Convert abbreviations, numbers, and symbols into text.

- Numbers, e.g:
  - 124 one hundred twenty-four
  - 1949 nineteen-forty-nine OR one thousand nine hundred forty-nine
  - 3/7 three-seventh OR March seventh
  - III (Chapter) three OR (Henry the) third

- Abbreviations, e.g: (of ambiguous abbreviations):
  - kg (one) kilogram OR (five) kilograms
  - St. saint OR street
  - Dr. doctor OR drive

- Symbols, e.g. :
  - & and
  - @ name@institute.com
  - $500 five hundred dollars

Text-to-speech system

TEXT ANALYSIS
- Document structure detection
- Text normalization
- Linguistic analysis

PHONETIC ANALYSIS
- Homograph disambiguation
- Morphological analysis
- Grapheme-to-phoneme conversion

PROSODIC ANALYSIS
- Pitch and duration attachment

SPEECH SYNTHESIS
- Voice rendering

Frontend

Backend

Phonetic analysis

- Phonetic analysis is prediction of pronunciation. It converts orthographic symbols (written text) to phonemes:
  - “HE MADE A RECORD. I WILL RECORD THE MINUTES OF THE MEETING.”
  - hiːmeɪd ə rɛkɔːd …

- Some phonetic alphabets:
  - IPA (non-ASCII)
  - SAMPA (Speech Assessment Methods – Phonetic Alphabet)
  - ASCII
  - Worldbet
  - ASCII encoding of IPA (with additional symbols)
  - Arpabet
  - ASCII encoding of American English phonemes

Phonetic analysis

"HE MADE A RECORD. I WILL RECORD THE MINUTES OF THE MEETING."

record[1] /'rekɔ:d˛ [am] 'rekərd/ *n*
1 *He has a long criminal record. the medical record[s]*
2 *There is no historical record of this event.*
…

record[2] /rɪ'kɔ:d/ *v tr*
1 *to record the minutes of a meeting, to record the decision in the minutes.*
2 *the only earthquake recorded in the area*
…

- Required steps:
  - **Homograph disambiguation**: Disambiguate similar-looking words with different meaning to determine the proper phonetic pronunciation.
  - **Morphological analysis**: Analysis of words to find inflected and derivational forms.
  - **Letter-to-sound:** Find the phonetic representation using a set of letter-to-sound rules and/or dictionary lookup.

Text-to-speech system

TEXT ANALYSIS
- Document structure detection
- Text normalization
- Linguistic analysis

PHONETIC ANALYSIS
- Homograph disambiguation
- Morphological analysis
- Grapheme-to-phoneme conversion

PROSODIC ANALYSIS
- Pitch and duration attachment

SPEECH SYNTHESIS
- Voice rendering

Frontend

Backend

Prosodic analysis

- Speech prosody refers to the longer-term properties of speech, such as:
  - **Pauses**: Indicate phrases,
  - **Pitch**: F0 as a function of time,
  - **Speaking rate**: Phoneme durations, timing, rhythm,
  - **Loudness**: Volume of speech.

“I WILL RECORD THE MINUTES OF THE MEETING.”

“COULD YOU RECORD THE MINUTES OF THE MEETING?”

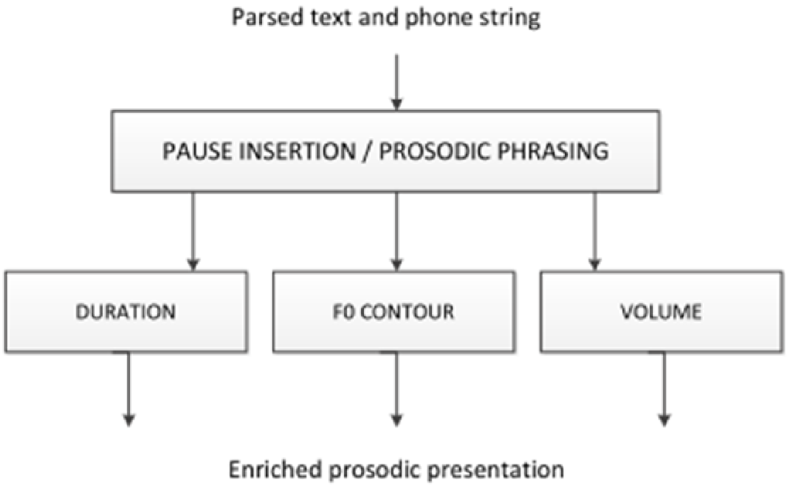
“I BUILT THE FIRST **BACKEND**.”

“I BUILT THE **FIRST** BACKEND.”

”**I** BUILT THE FIRST BACKEND”

Prosodic analysis

Elements of prosodic generation in TTS



Text-to-speech system

TEXT ANALYSIS
- Document structure detection
- Text normalization
- Linguistic analysis

PHONETIC ANALYSIS
- Homograph disambiguation
- Morphological analysis
- Grapheme-to-phoneme conversion

PROSODIC ANALYSIS
- Pitch and duration attachment

Frontend

SPEECH SYNTHESIS
- Voice rendering

Backend

Speech synthesis

Main approaches for signal generation in TTS:

1. Formant synthesis

2. Concatenative synthesis:
   - Diphone synthesis,
   - Unit selection synthesis.

3. Statistical (parametric) synthesis:
   - Hidden Markov model (HMM) based synthesis,
   - (Deep neural network-based synthesis).

4. Articulatory synthesis

UNIT SELECTION WIDELY USED IN COMMERCIAL APPLICATIONS

Formant synthesis

- Rules used to generate speech intonation (F0 curve) and formants.

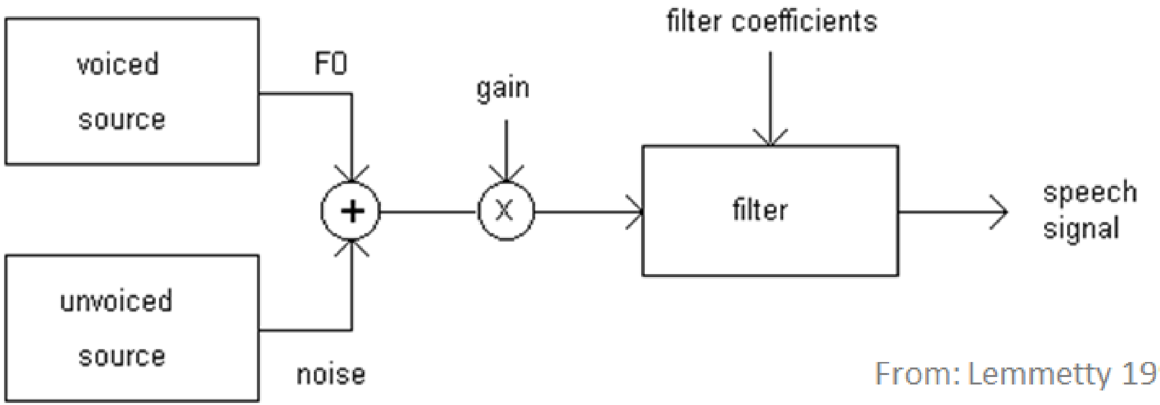
- Example rule for intonation:

”If the syllable is stressed, and it is not a function word, increase the level of f0.”

- Formants generated based on phone-specific target frequencies and bandwidths.
- Transitions between phonemes must be smooth.

Formant synthesis

- Formant synthesis based on the **source-filter model:**
  - Source signal is created based on the **intonation information**, the filter based on the **formant information.**



From: Lemmetty 1999

- At least the lowest three formants required.
- Model parameters updated every 5-10 ms.

Formant synthesis

+ Easy to modify:
  - Intonation (varying pitch over segment) and vocal tract model can be modified.

+ Relatively easy/efficient to implement:
  - Intelligible speech already with a low number of model parameters (e.g. 40).

+ Can produce any speech sounds:
  - Even combinations impossible for a human speaker.

- Simplifying models typically result in unnatural-sounding synthetic speech.
- Legacy technology, state-of-the-art until the mid 1980s

Thinking break (2 min)

Concatenative synthesis

- Concatenative speech synthesis is ‘**copy-paste**’ type of synthesis:
  - Recorded speech segments are joined together (=concatenated) to form synthetic speech.

- Speech database has an essential role:
  - Speech is recorded and waveforms are annotated either manually or by a computer.

- Since real speech segments are used to form waveforms, high intelligibility and naturalness can be achieved.

Concatenative synthesis

- Possible size of the database segments (i.e. “unit”):
  - Phone
  - Diphone
  - Triphone
  - Syllable
  - Word
  - Sentence
  - …

- By increasing the size of the segments we can decrease the amount of concatenation errors…

Concatenative synthesis

… but we increase the amount of possible segments! Example statistics for English:

- Phonemes: 42
- Syllables: 15 000
- Words: > 100 000; Oxford English dictionary: 250 000 (without inflections/new words)

- In addition, prosody is usually affected if long segments are concatenated.

- Typically **diphones** are used: mid-part of a speech sound is typically stable, hence easy concatenation:
  - Coarticulation taken into account automatically
  - Number of diphones in English: 1300

Concatenative synthesis: Diphone synthesis

- Concatenation of diphone-sized speech segments to form a synthetic speech signal.

- **Small database**: Each diphone of a language should be included.

- Prosody created by **modifying the database diphones:**
  - F0 and duration modified based on the prosody prediction; e.g. PSOLA (pitch synchronous overlap-add) used for modification.

- Modifications **decrease** the synthesis quality.

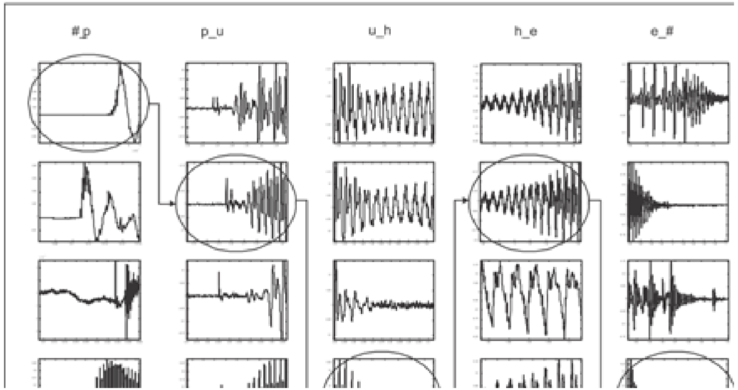
Concatenative synthesis: Unit selection

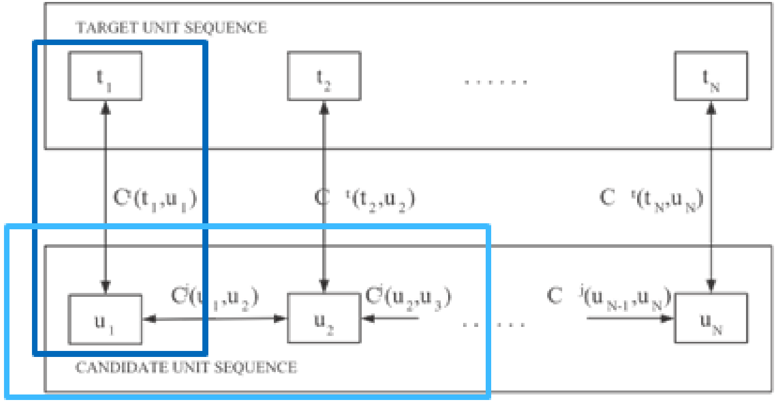
- Modification of speech segments in the diphone synthesis decreases the quality:
  - What if we built a larger database including multiple versions of each diphone (or whatever speech unit we decide to use) and did not modify the selected versions?

- In unit selection synthesis, a **large speech database** is recorded:
  - The database can contain even tens of hours of speech from one speaker.
  - Multiple instances for each diphone are included in the database and synthetic speech is formed by **selecting the best candidate sequence.**
  - In the best case, units can be used **without** any (or with a minimum amount of) **signal modifications.**

Concatenative synthesis: Unit selection

- Realization of a speech fragment depends on the context it appears in → Create acceptable prosody by selecting fragments u_i i=1,..,N from a suitable context to minimize a total cost.
- **Target cost** $C(t_i,u_i)$: how closely the linguistic context of each candidate unit from the database matches the linguistic specification of the sentence to be rendered
- **Joint cost** $C(u_i,u_{i+1})$: how well each possible sequence of candidate units will concatenate
- Smooth transitions between adjacent units without discontinuities in spectrum or fundamental frequency are desired.
- Concatenation of naturally adjacent speech fragments doesn't cause artifacts
- Joint cost can be measured e.g. with spectral discontinuity between last frame of u_i and first frame of the unit u_{i+1}.





Concatenative synthesis: Unit selection

- Synthesis example: ’puhe’:
  - Forming a target unit sequence:

    #-p p-u u-h h-e e-#

  - Analysis of the units and their contexts:

#-p:	position in a syllable:	1st units
	position in a phrase:	1st word
	previous phoneme:	NA
	following phoneme:	/u/
	etc.	
p-u:	position in a syllable:	2nd unit
	etc..	

Concatenative synthesis: Unit selection

- Identifying suitable candidates in the speech database:

#-p:	#-p(1),	#-p(2),	#-p(3),	#-p(4),	#-p(5)
p-u:	p-u(1),	p-u(2),	p-u(3),	p-u(4)	
u-h:	u-h(1),	u-h(2),	u-h(3)		
h-e:	h-e(1),	h-e(2),	h-e(3),	h-e(4),	h-e(5)
e-#:	e-#(1),	e-#(2),	e-#(3)		

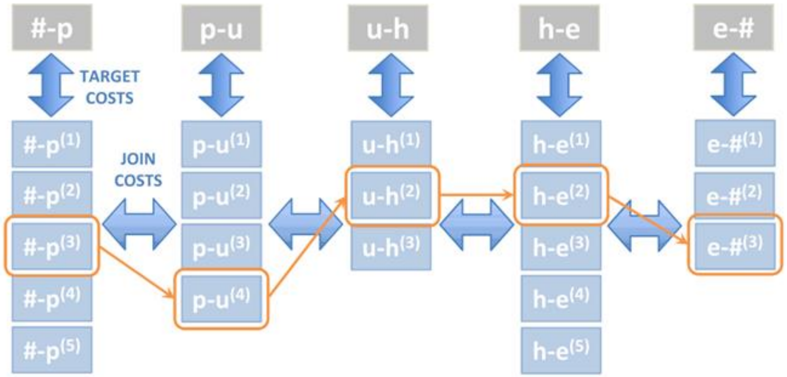
- Selection of a sequence of units that:
  1. Match with the properties of the target sequence. → Target cost: cost of selection
  2. Provide smooth concatenation for adjacent units. → Joint cost: cost of concatenation

- Formulating the selection procedure as a cost function minimization task.

$$d(U,T) = \Sigma_i C(u_i,u_{i+1}) + C(t_i,u_i)$$

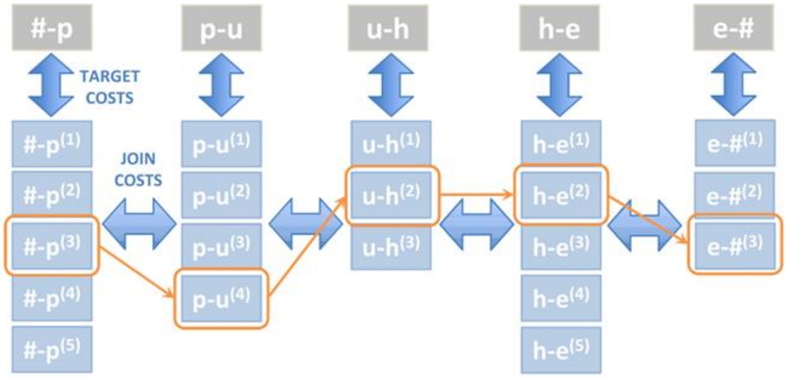
Concatenative synthesis: Unit selection

- Selection procedure as a cost function minimization task:
  - Find the **sequence of database units** that gives the **minimum total cost**.



Concatenative synthesis: Unit selection

- Selection procedure as a cost function minimization task:
  - Find the **sequence of database units** that gives the **minimum total cost**.



Baseline Proposed

Speech parameterization:

- STRAIGHT spectrum converted into 24th order Mel-cepstrum coefficients,

- STRAIGHT F0, and

- mean band aperiodicity of five frequency bands of the STRAIGHT aperiodicity

- An analysis update interval of 5ms is used for both approaches.

Using Robust Viterbi Algorithm and HMM-Modeling in Unit Selection TTS to Replace Units of Poor Quality, by Silén, H., Helander, E., Nurminen, J., Koppinen, K., and Gabbouj, M.

- Baseline: Traditional Viterbi algorithm to find the minimum cost sequence
- Proposed: Robust Viterbi algorithm to find the minimum cost sequence with unsuitable units ignored in the search and replaced afterwards using HMM-based

Conclusions: Concatenative synthesis

Diphone synthesis

\+ Diphone synthesis provides a light-weight approach for concatenative synthesis.

\- Modifications reduce quality.

Unit selection synthesis

\+ Minimum amount of modifications of recorded speech fragments -> possible to achieve natural-sounding speech.

   Widely used in the current commercial solutions.

\- Depends heavily on the database and requires a large amount of recorded speech.

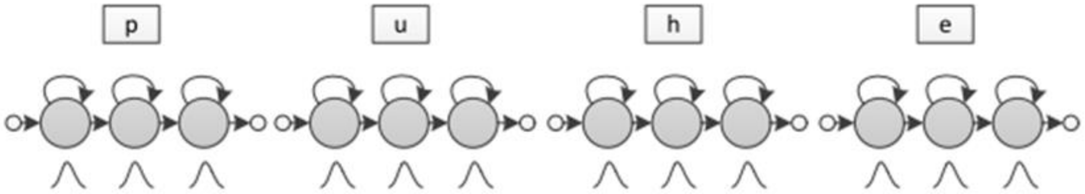
   A large speech database required, still there might be cases where no suitable units are available; hence might still result in poor quality.

   Collecting and annotation of a large database is time-consuming and expensive; for each new synthetic voice, a new database must be created.

Tampere University

Statistical speech synthesis

- Hidden Markov models are widely used in automatic speech recognition.



- HMM modeling in speech synthesis:
    - Parameters of the statistical model are learnt from training data
    - In synthesis, **phoneme sequence is known** and the task is to **generate a suitable sequence of speech features** (spectral features and F0) for **waveform generation.**
    - Allows synthesis with highly varying speaking styles (generalization) without the need for a large database (as would be the case in unit selection).

Statistical speech synthesis

- Speech **waveforms are parameterized** using a model:
  - This model contains the spectral part (e.g. MFCCs) and F0.
  - In the training, the aim is to learn a **context-dependent 3-state HMM for each phoneme** using a database with speech signals and phoneme labels.
    - Learns to model the spectral parameters based on phone + context.
    - Example of context-dependence: *'A model for spectral features and F0 for phoneme /e/ when the preceding phoneme is /h/ and the following phoneme is /l/ and the phone is the first one in the first syllable of a second word of a phrase and …'*

- In the synthesis stage, we select the sequence of correct context-dependent HMMs, and generate speech features based on them.
  - An external duration model provides information how many frames of spectral features are generated in each HMM's states.

- Waveform is then generated by a vocoder that maps generated parameters into acoustic a waveform (MFCCs+F0 into waveform).

Statistical speech synthesis

- Two separate phases:
- **Training phase**: The task is to learn context-dependent HMM for each phoneme of a language based on recorded and parameterized speech data.
- **Synthesis phase**: Identify the correct phoneme models for the given text, and

  (1) Generate speech features (often called speech parameters)

  (2) Synthesize the speech signal based on them.
- HMM-based speech synthesis is a corpus-based/data-driven approach, but not a concatenative approach!
- Deep neural networks are found more efficient to learn the non-linear mapping from frame-level linguistic information to spectral model parameters for synthetization, replacing HMMs. (This approach still depends on the front-end described, and a vocoder)

Conclusions: Statistical speech synthesis

+ Small footprint: After training, no need for the database anymore.
+ Easy adaptation to new synthesis voices and speaking styles with a small amount of new data.
+ Very stable quality without concatenation artifacts (compare with unit selection!)

- Effects on speech naturalness: Some of the small details are lost in simplifying modeling. Combined with the required speech parameterization, speech naturalness might be affected.

Articulatory synthesis

- Based on physical models of human speech production:
    - Physical models for articulator motion (e.g. piecewise constant area functions for vocal tract components such as lips, velum, glottal area etc.).
    - Model parameters are adjusted based on the text to be synthesized and speech is generated by the model.
    - The relationship between articulatory parameters and acoustic values is typically performed with non-linear methods such as neural-networks or codebooks.
- Parameters to be adjusted: e.g. lip and tongue movement, tension of the vocal cords etc.
    - Data from X-rays or MRI of real speech events.
- Example: VocalTractLab

\+ Only produces speech sound combinations that are possible for human speech production.
\+ Enables more precise generation of transient sounds compared to the alternative approaches.
\- Physical modeling is difficult.

Speech synthesis examples

- Examples of HMM-based synthesis:
  http://flite-hts-engine.sp.nitech.ac.jp/

- More examples: HMM, unit selection, diphone
  http://www.cstr.ed.ac.uk/projects/festival/morevoices.html

Break

Neural speech synthesis

Introduction

- Deep learning has allowed breakthroughs in several areas of research in recent years
  - Computer vision, audio processing, video processing
  - Neural networks with deep structures and millions of parameters

- Factors that have enabled their use:
  - New techniques allowing fast training of more deeper networks
  - Hardware development: Graphics processing units (GPUs)
  - Large amounts of data available

- Deep learning methods have state-of-the-art performance in several audio processing tasks:
  - Speech recognition
  - Synthesis
  - Enhancement
  - Separation
  - Sound event detection
  - Emotion recognition
  - etc

Tampere University

WaveNet (Sept 2016)

- Neural **autoregressive generative model**
  - **Autoregressive**: regression on past values to predict current value
  - Remember LPC from speech modeling, that used an AR-model (all-pole filter) to predict the next output sample
  - **Generative model**: the output is a probability density over all possible output values.

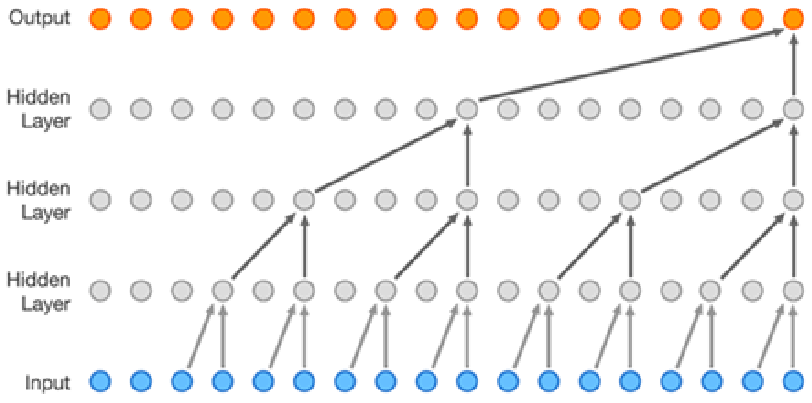
- Basic principle: given past output values, predict the (probability distribution) of the next sample value.
- A joint-probability model

$$p(\boldsymbol{x}) = \prod_{t=1}^{T} p(\underbrace{\boxed{x_t}}_{\text{Current output value}} \,|\, \underbrace{\boxed{x_1, x_2, \dots, x_{t-1}}}_{\text{Past output values}}), \quad \underbrace{\boxed{\boldsymbol{x}=[x_1, x_2, \dots, x_T]}}_{\text{Waveform samples}}$$

Tampere University

WaveNet

- WaveNet is a convolutional neural network
  - Dilated convolutions to increase the “receptive field” size
    - Skip between filter values increases by factor of 2 in each layer
    - Computationally efficient way of using thousands of samples to predict the current output (2 filter values / filter / layer)
  - Causal (doesn’t look at the future values, only past)
  - Can model any kind of audio (as we saw in the synthesis lecture)



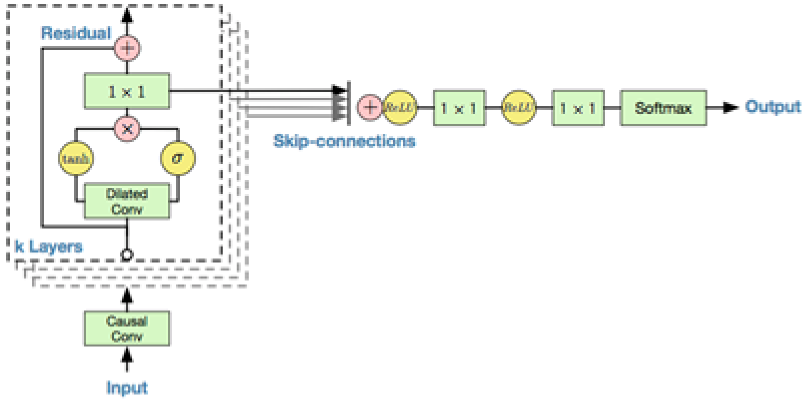


Figure: https://deepmind.com/blog/wavenet-launches-google-assistant/

WaveNet

- The model itself trained on speech data will be able to generate convincing sounding speech (even though gibberish)
- The model can be conditioned on **external input h:**
  - Here: information about the text to be synthesized with linguistic features (phone durations, F0 estimates)
- The conditional probability model

$$p(\boldsymbol{x}|\boldsymbol{h}) = \prod_{t=1}^{T} p(x_t|x_1, x_2, \ldots, x_{t-1}, \boldsymbol{h})$$

$$\boldsymbol{x}=[x_1, x_2, \ldots, xt]$$
$$\boldsymbol{h}=[h_1, h_2, \ldots, h_t]$$

- Linguistic feature are obtained from an external model
- Conditioning can also be done for different speakers

WaveNet

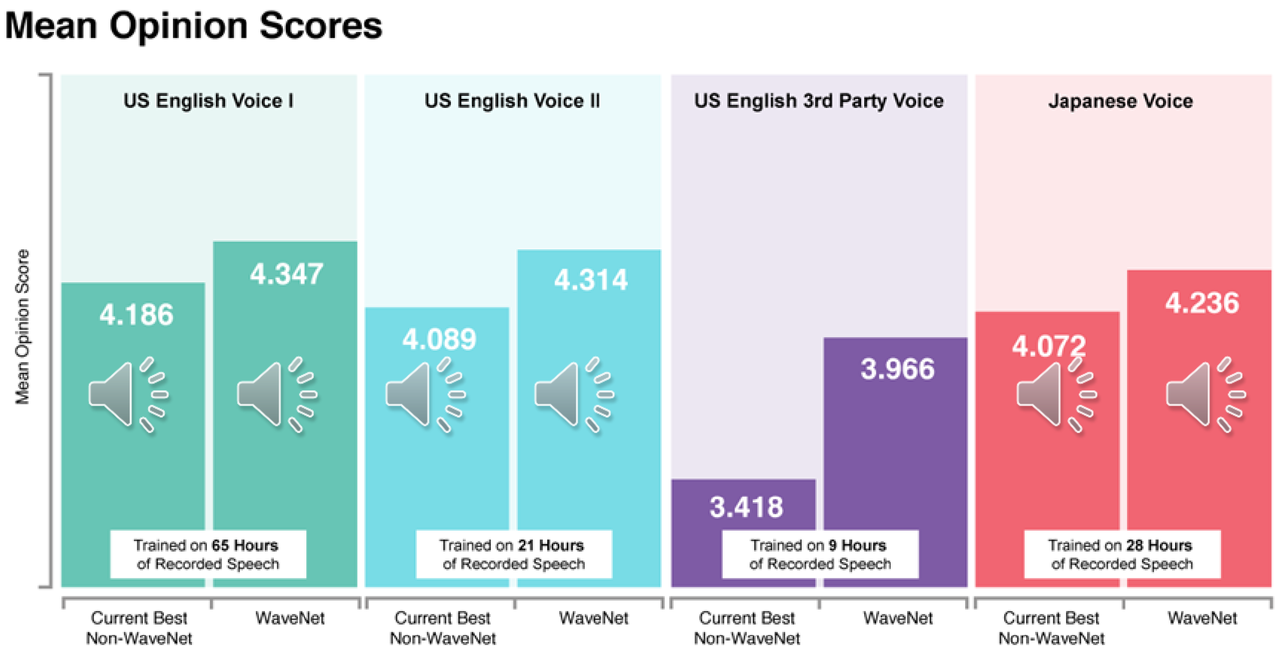
- Training (done once, takes a long time)
  1. Learn to predict next sample given past samples + conditioning data
  2. Update model parameters based on the prediction error

- Testing (synthesis)
  - Predict sample xt using T past predicted samples + conditioning data
  - Place predicted sample to xt+1

- Improvements announced in October/2017
  - 1000 x faster, 24 kHz, 16 bits/sample
  - Used in Google’s production system (US Eng, Japanese)

Tampere University

Results and samples



- Samples from https://deepmind.com/blog/wavenet-launches-google-assistant/
- For US English Voice I, the human speech is rated 4.667

WaveNet

- \+ Higher than traditional method’s TTS quality
- \+ Dilated convolution allows to see a long way into the past with less model parameters (efficient)
- \+ Produces direct waveform, no “synthesis method”or vocoder is required

- \- Use in TTS depends on external models for linguistic features

Tacotron and Tacotron2 (Dec 2017)

- Converts a sequence of characters into log-mel spectrogram using a neural network
  - Utilizes **attention mechanism**, introduced for machine translation
  - Attention mechanism allows efficient access to input sequence elements (i.e. the characters) while predicting the output spectral parameters

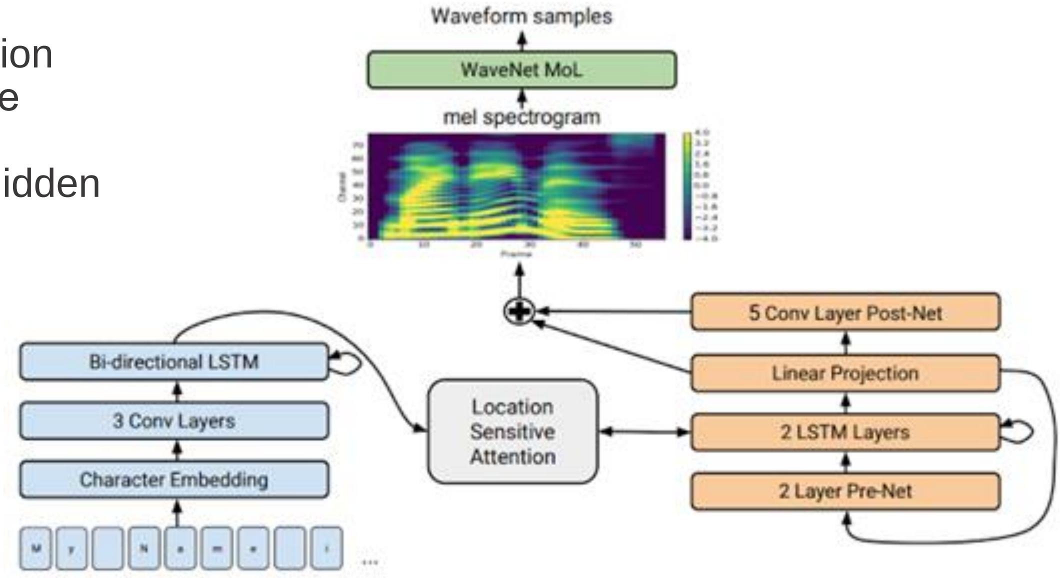
- Log-mel spectrogram needs to be converted to waveform
  - Tacotron: uses an external method for this
  - Tacotron2: uses wavenet to convert the 80 log-mel spectrogram values into waveform

- Training
  - Only requires <text,audio> pairs. No linguistic analysis, no phonetic information extraction. (Text normalization is done)

- MOS score: 4.53
  - Human professional recording 4.58
  - HMM-TTS 3.49
  - Concatenative 4.17

Tacotron2 block diagram

- Encoder-decoder structure with attention
- Encoder: characters → hidden feature representation
- Decoder: predicts spectrogram from hidden features (with attention)



Encoder

Decoder

Tacotron2

+ State-of-the-art performance, almost indistinguishable from real speech
+ Trained using only speech examples and corresponding text transcripts
+ Does not require linguistic and acoustic features as input

- Does not operate in real-time
- Injection of emotions (e.g. happy, sad) not controllable

Other approaches exist: Baidu’s deep voice 3 (ICLR 2018), MILA’s Char2Wav

Summary

- Traditional TTS systems comprise of the
  - Frontend (linguistic analysis)
  - Backend (synthesis method + optional vocoder)
    - Formant, concatenative, statistical, articulatory

- Modern deep learning based solutions
  - May still rely on traditional TTS components: frontend, vocoder
    - The goal is end2end
  - end2end approaches have appeared with SoTA performance
    - Still research required, except several improvements during 2018