NPFL123 Dialogue Systems
11. Text-to-Speech Synthesis

https://ufal.cz/npfl123

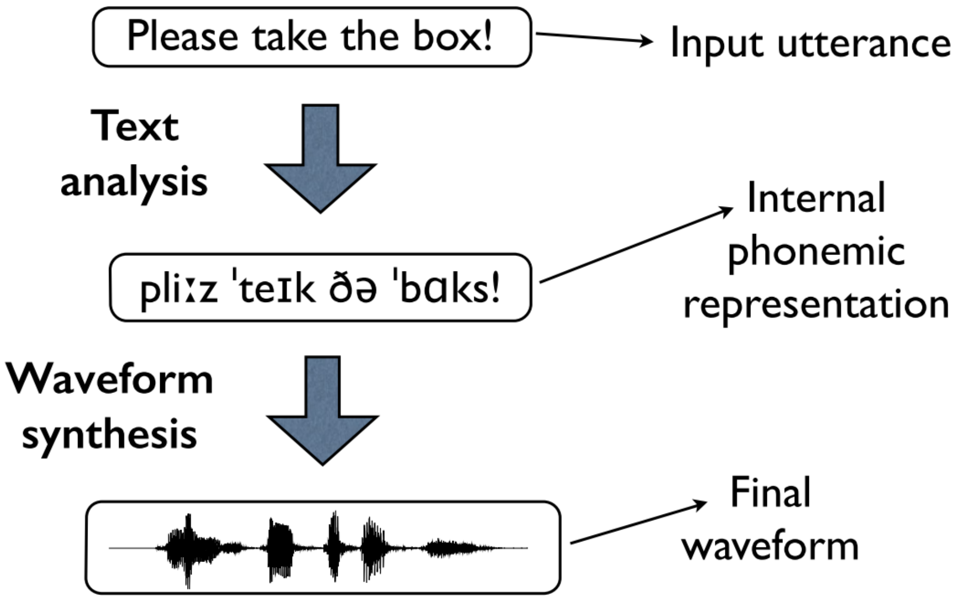
Ondřej Dušek, Simone Balloccu, Mateusz Lango, Kristýna Klesnilová & Jan Cuřín

15. 5. 2024

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Text-to-speech synthesis

- Last step in voice-based DS pipeline
  - from NLG (system utterance text)
  - to the user (audio waveform)
- Needed for all but the simplest DSs
- Sequence-to-sequence conversion
  - from discrete symbols (letters)
  - to continuous time series (audio waves)
    - regression problem
  - mimicking human articulation in some way
- Typically a 2-step pipeline:
  - **text analysis** (frontend) – converting written to phonetic representation
  - **waveform synthesis** (backend) – phonemes to audio

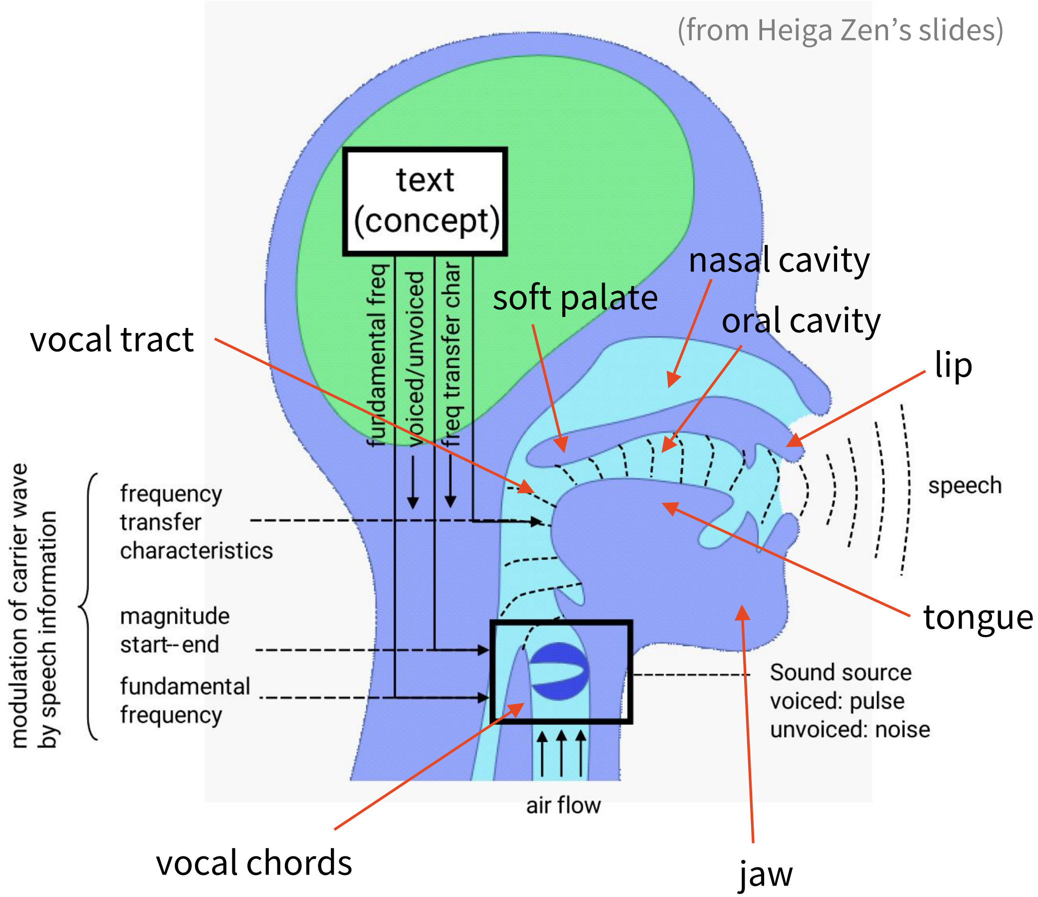


(from Pierre Lison's slides)

Human articulatory process

- text (concept) → movement of muscles → air movement (sound)
- source excitation signal = air flow from lungs
  - vocal cords resonation
    - base frequency (F0)
    - upper harmonic frequencies
  - turbulent noise
- frequency characteristics moderated by **vocal tract**
  - shape of vocal tract changes (tongue, soft palate, lip, jaw positions)
  - some frequencies resonate
  - some suppressed

(from Heiga Zen's slides)

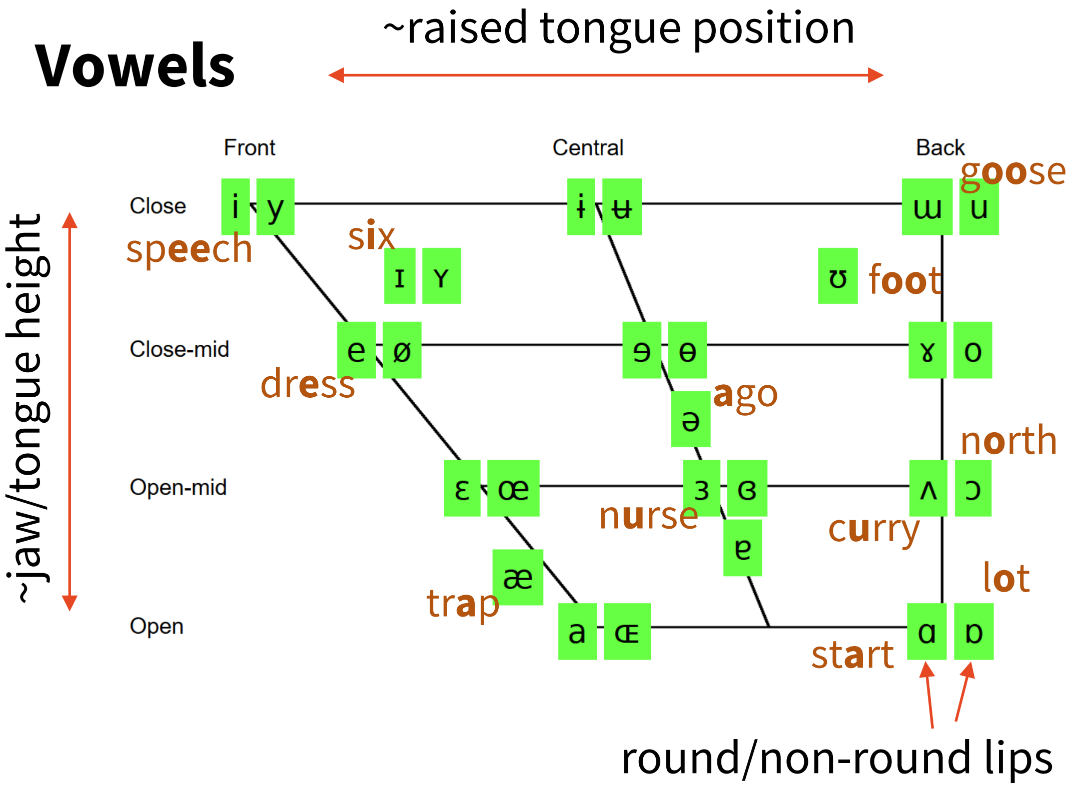


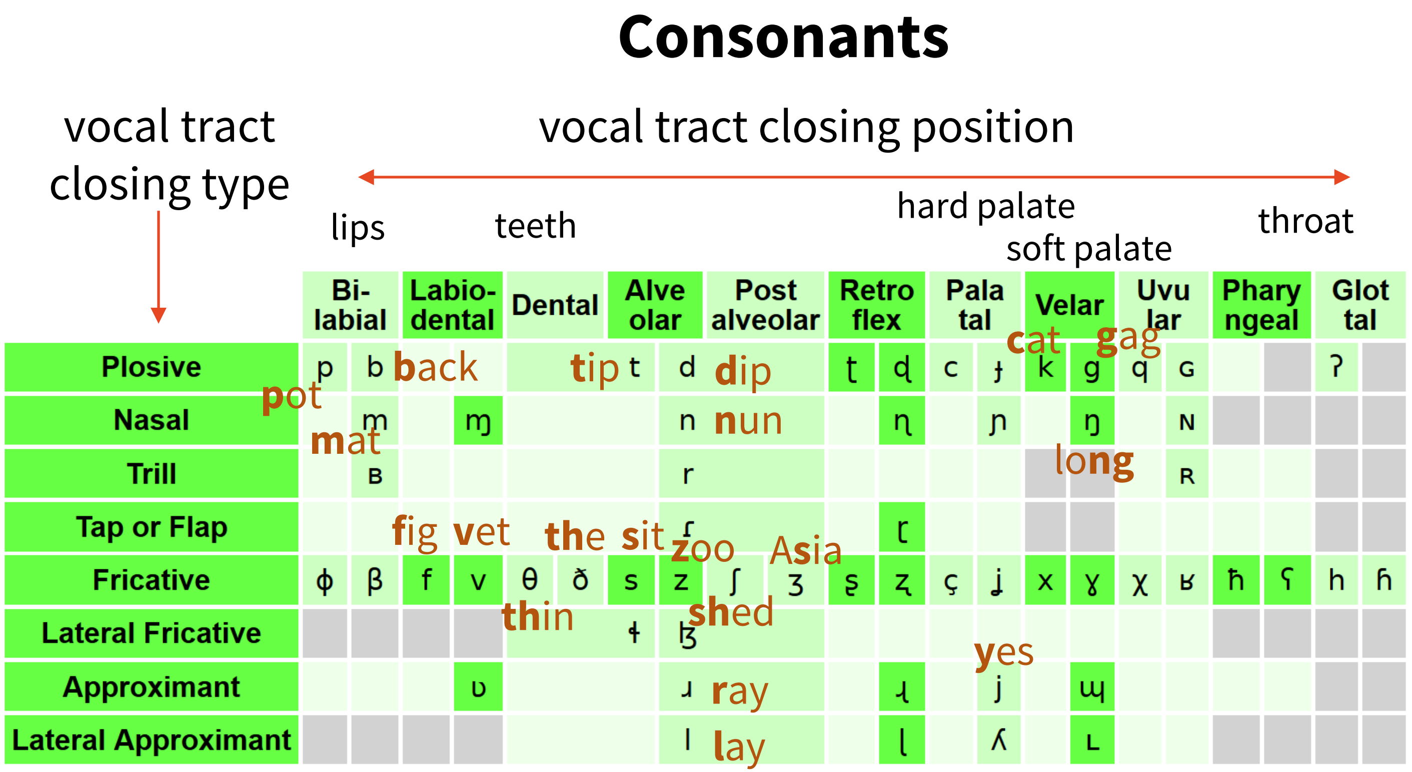
Sounds of Speech

- **phone/sound** –any distinct speech sound
- **phoneme** – sound that distinguishes meaning
  - changing it for another would change meaning (e.g. ***dog*** → ***fog***)
- **vowel** – sound produced with open vocal tract
  - typically **voiced** (=vocal chords vibrate)
  - quality of vowels depends mainly on vocal tract shape
- **consonant** – sound produced with (partially) closed vocal tract
  - voiced/voiceless (often come in pairs, e.g. [**p**] – [**b**])
  - quality also depends on type + position of closing
    - stops/plosives = total closing + “explosive” release ([**p**], [**d**], [**k**])
    - nasals = stops with open nasal cavity ([**n**], [**m**])
    - fricatives = partial closing (induces friction – hiss: [**f**], [**s**], [**z**] …)
    - approximants = movement towards partial closing & back, half-vowels ([**w**], [**j**] …)

Sounds of Speech

http://www.ipachart.com/
(clickable with sounds!)

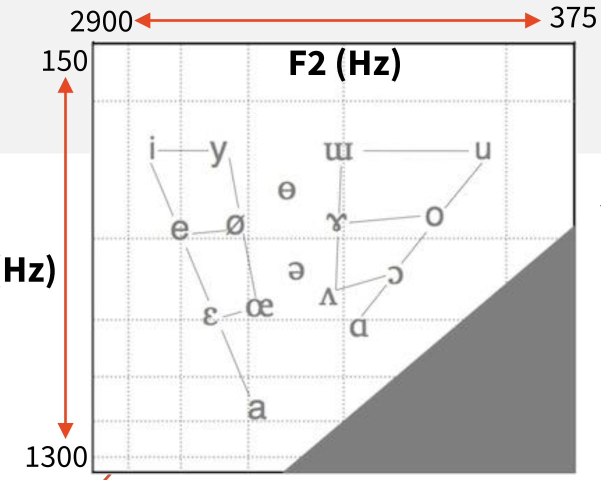




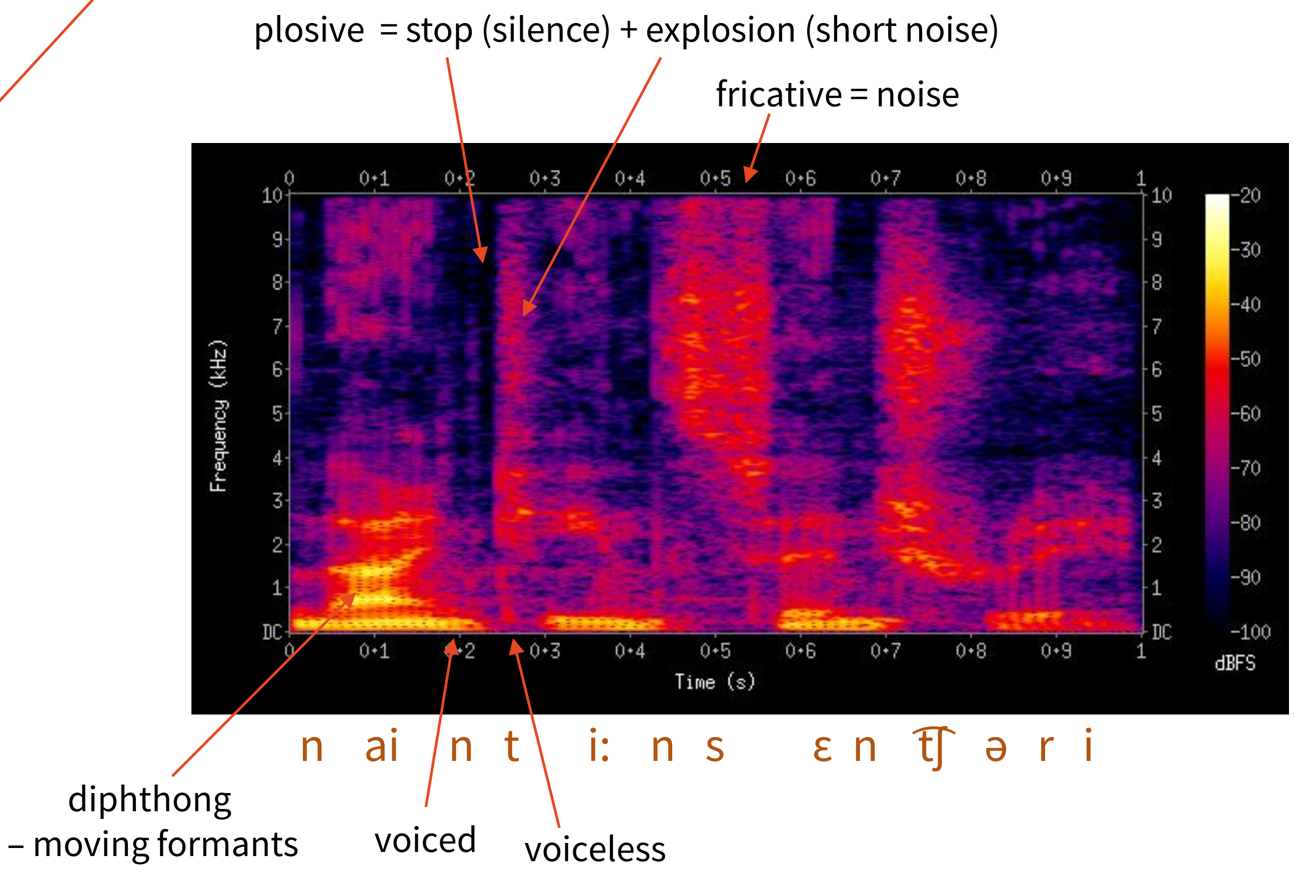
- Word examples according to Received Pronunciation ("Queen's English"), these vary across dialects
- More vowels: diphthongs (changing jaw/tongue position, e.g. [ei] *wait*, [əʊ] *show*)
- More consonants: affricates (plosive-fricative [tʃ] *chin*, [dʒ] *gin*), labio-velar approximant [w] *well*

Spectrum

- speech = compound wave
  - different frequencies (spectrum)
  - shows in a **spectrogram**
    - frequency – time – loudness
- base vocal cord frequency **F0**
  - present in voiced/vocals
  - absent in voiceless
- **formants** = loud upper harmonics
  - of base vocal cord frequency
  - **F1, F2** – 1st, 2nd formant
  - distinctive for vowels
- noise – broad spectrum
  - consonants (typical for fricatives)



vowel formants



https://www.englishspeechservices.com/blog/the-vowel-space/
https://youtu.be/FdldD0-kEcc (more insight on vowels & formants!)

https://en.wikipedia.org/wiki/Spectrogram

From sounds to utterances

- phones group into:
  - **syllables** – minimal pronounceable units
  - **stress units** (~ words) – group of syllables with 1 stressed
  - **prosodic/intonation units** (~ phrases)
    - independent prosody (single prosodic/pitch contour)
    - tend to be separated by pauses
  - utterances (~ sentences, but can be longer)
- neighbouring phones influence each other a lot!
- **stress** – changes in timing/F0 pitch/intensity (loudness)
- **prosody/melody** – F0 pitch
  - sentence meaning: question/statement
  - tonal languages: syllable melody distinguishes meaning

intonation unit

stress units

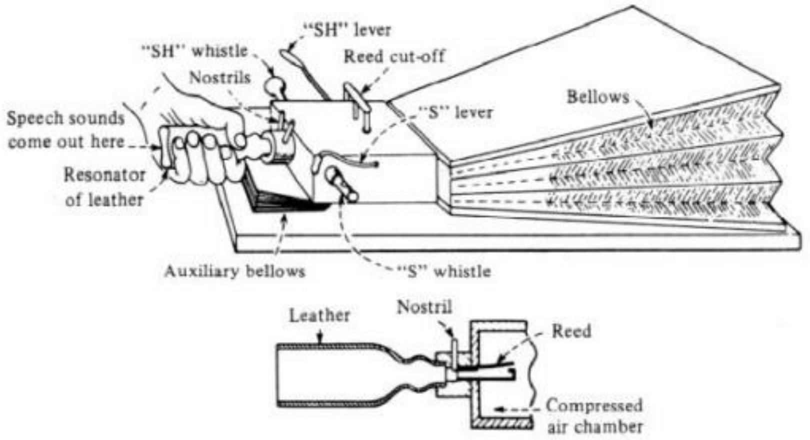
syllables

[ˈdʒæk | pɹəˌpɛəɹɪŋ ðə ˈweɪ | wɛnt ˈɒn ‖]

https://en.wikipedia.org/wiki/Prosodic_unit

TTS Prehistory

- 1st mechanical speech production system
  - Wolfgang von Kempelen's speaking machine (1790's)
  - model of vocal tract, manually operated
  - (partially) capable of monotonous speech
- 1st electric system – Voder
  - Bell labs 1930, operated by keyboard (very hard!)
  - pitch control
- 1st computer TTS systems – since 1960's
- Production systems – since 1980's (→)



(Lemmetty, 1999)
https://youtu.be/k_YUB_S6Gpo?t=67

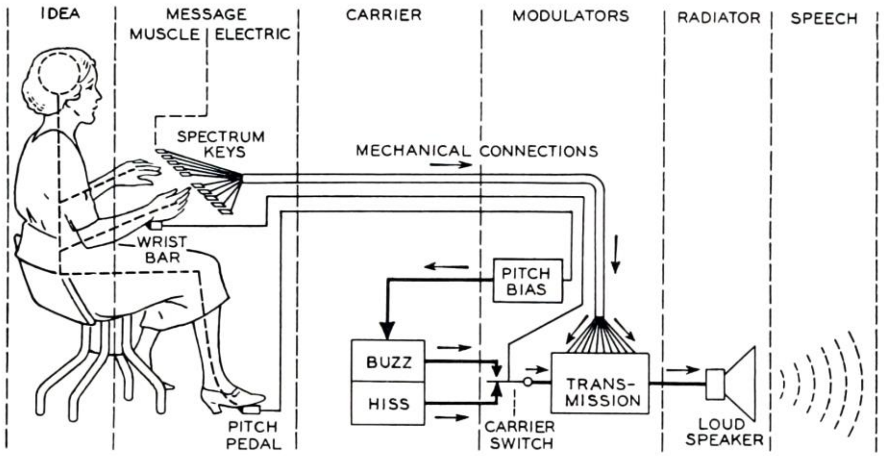
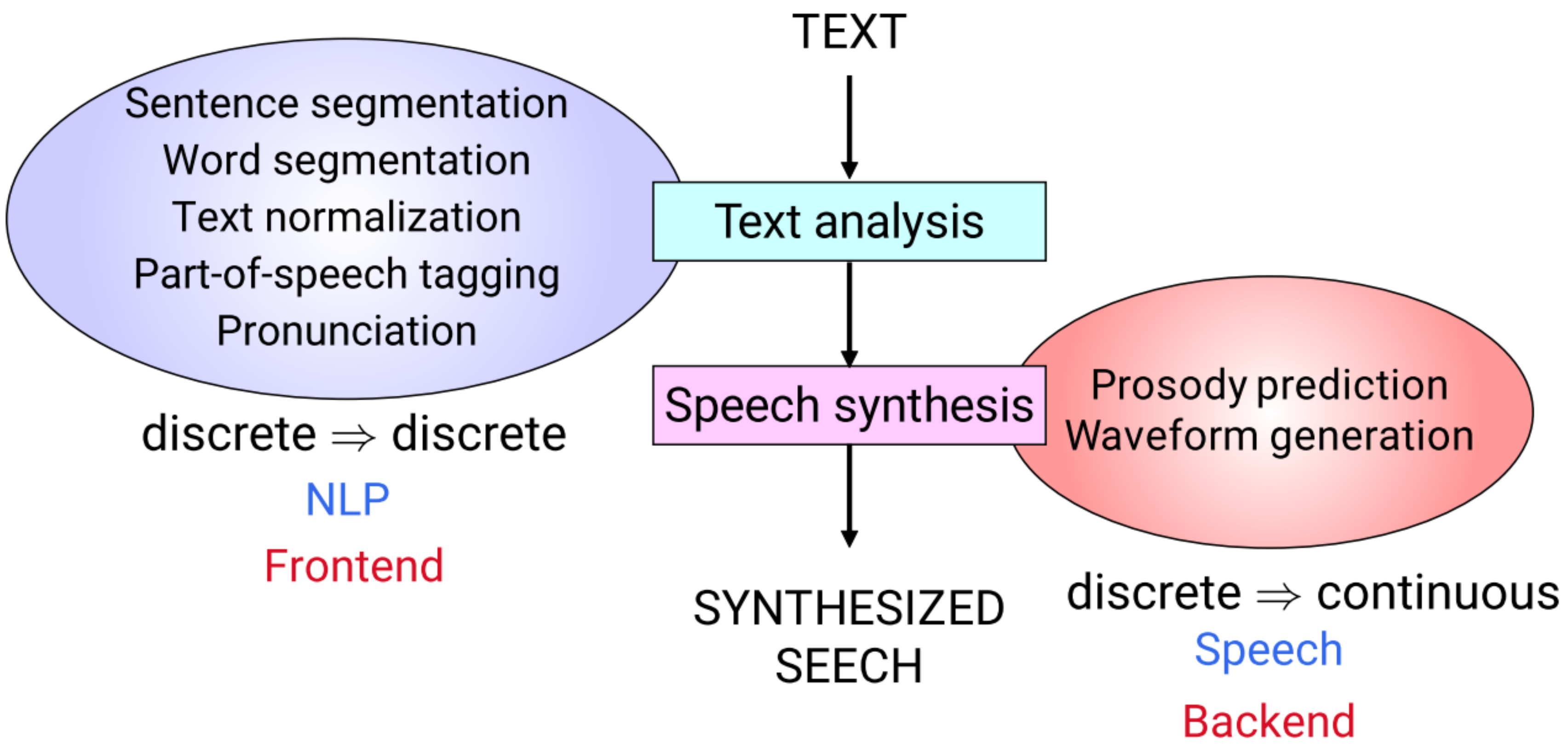


Fig. 8—Schematic circuit of the voder.

https://en.wikipedia.org/wiki/Voder
https://youtu.be/TsdOej_nC1M?t=36

TTS pipeline

- frontend & backend, frontend composed of more sub-steps
  - frontend typically language dependent, but independent of backend



(from Heiga Zen's slides)

Segmentation & normalization

- remove anything not to be synthesized
  - e.g. HTML markup, escape sequences, irregular characters
- segment sentences
- segment words (Chinese, Japanese, Korean scripts)
- spell out:
  - abbreviations (context sensitive!)
  - dates, times
  - numbers (ordinal vs. cardinal, postal codes, phone numbers…)
  - symbols (currency, math…)
- all typically rule-based

Tue Apr 5 → Tuesday April fifth
€ 520 → five hundred and twenty euros

432 Dr King Dr → four three two doctor king drive
1 oz → one ounce
16 oz → sixteen ounces

Grapheme-to-Phoneme Conversion

- main approaches: pronouncing dictionaries + rules
  - rules good for languages with regular orthography (Czech, German, Dutch)
  - dictionaries good for irregular/historical orthography (English, French)
  - typically it's a combination anyway
    - rules = fallback for out-of-vocabulary items
    - dictionary overrides for rules (e.g. foreign words)
  - can be a pain in a domain with a lot of foreign names
    - you might need to build your own dictionary (even with a 3rd-party TTS)
- phonemes typically coded using ASCII (SAMPA, ARPABET…)
- pronunciation is sometimes context dependent
  - part-of-speech tagging
  - contextual rules

phoneme
['fəʊniːm]
f@Uni:m
F OW N IY M

record (NN) = ['ɹɛkoːd]
record (VB) = [ɹɪ'koːd]

read (VB) = ['ɹiːd]
read (VBD) = ['ɹɛd]

the oak = [ð**iː**'əʊk]
the one = [ð**ə**'wʌn]

Intonation/stress generation

- rules/statistical
  - predicting intensity, F0 pitch, speed, pauses
    - stress units, prosody units
  - language dependent
  - traditionally: classification – bins/F0 change rules
- based on:
  - punctuation (e.g. “?”)
  - chunking (splitting into intonation units)
  - words (stressed syllables)
  - part-of-speech tags (some parts-of-speech more likely to be stressed)
  - syntactic parsing

SSML (Speech Synthesis Markup Language)

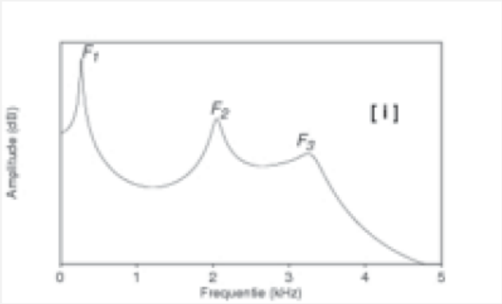
- manually controlling pronunciation/prosody for a TTS
  - must be supported by a particular TTS
  - e.g. Alexa supports this (a lot of other vendors, too)
- XML-based markup:
  - <break>
  - <emphasis level="strong">
  - <lang>
  - <phoneme alphabet="ipa" ph="ˈbɑ.təl">
  - <prosody rate="slow">, <prosody pitch="+15.4%">, <prosody volume="x-loud">
  - <say-as interpret-as="digits"> (date, fraction, address, interjection…)
  - _{subst} (abbreviations)
  - <voice>
  - <w role="amazon:VBD">read</w> (force part-of-speech)

https://developer.amazon.com/docs/custom-skills/speech-synthesis-markup-language-ssml-reference.html

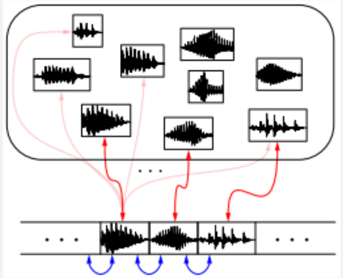
Waveform Synthesis

- many different methods possible
- **formant-based** (~1960-1980's)
  - rule-based production of formants & other components of the wave
- **concatenative** (~1960's-now)
  - copy & paste on human recordings
- **parametric** – model-based (2000's-now)
  - similar to formant-based, but learned from recordings
  - HMMs – dominant approach in the 2000's
  - NNs – can replace HMMs, more flexible
- NN-based **end-to-end methods**
  - now state-of-the-art

Rule-based, formant synthesis [1]



Sample-based, concatenative synthesis [2]



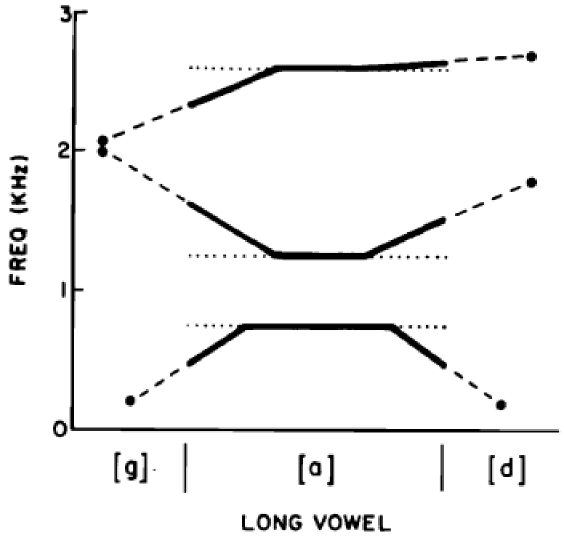
Model-based, generative synthesis

p(speech=  | text="Hello, my name is Heiga Zen.")

(from Heiga Zen's slides)

Formant-based Synthesis

- early systems
- based on careful handcrafted analysis of recordings
  - “manual” system training
  - very long evolution – DECtalk took ~20 years to production
  - barely intelligible at first
- rules for composing the output sound waves
  - based on formants resonators + additional components
  - rules for sound combinations (e.g. “b before back rounded vowels”)
  - rules for suprasegmentals – pitch, loudness etc.
- results not very natural, but very intelligible in the end
- very low hardware footprint



(Klatt, 1987)

Holmes et al., 1964

DECtalk, 1986

http://www.festvox.org/history/klatt.html (examples 17 & 35)

Concatenative Synthesis

http://www.festvox.org/history/klatt.html (examples 18 & 22)
https://www.ims.uni-stuttgart.de/institut/mitarbeiter/moehler/synthspeech/
(Festival English diphone example, MBROLA British English example)

- Cut & paste on recordings
  - can't use words or syllables – there are too many (100k's / 10k)
  - can't use phonemes (only ~50!) – too much variation
    - **coarticulation** – each sound is heavily influenced by its neighbourhood
- using **diphones** = 2nd half of one phoneme & 1st half of another
  - about 1,500 diphones in English – manageable
  - this eliminates the heaviest coarticulation problems (but not all)
  - still artefacts at diphone boundaries
- smoothing/overlay & F0 adjustments
  - over-smoothing makes the sound robotic
  - pitch adjustments limited – don't sound natural
- needs lots of recordings of a single person
- diphone representations: formants, LPC, waveform

Dixon & Maxey (1968)
formant diphones

Olive (1977)
LPC diphones

Festival (1997)
diphone synthesis
http://www.cstr.ed.ac.uk/projects/festival/

MBROLA (1996)
http://tcts.fpms.ac.be/synthesis/

Unit-selection Concatenative Synthesis

- using more instances of each diphone
  - minimize the smoothing & adjustments needed
- selecting units that best match the target position
  - match target pitch, loudness etc. (specification s_t) – **target cost** $T(u_t, s_t)$
  - match neighbouring units – **join cost** $J(u_t, u_{t+1})$
  - looking for best sequence $\widehat{U} = \{u_1, \dots, u_n\}$, so that:

$$\widehat{U} = \arg\min_{U} \sum_{t=1}^{n} T(u_t, s_t) + \sum_{t=1}^{n-1} J(u_t, u_{t+1})$$

  - solution: **Viterbi search**
  - leads to joins of stuff that was recorded together
- a lot of production systems use this
  - still state-of-the-art for some languages
  - but it's not very flexible, requires a lot of single-person data to sound good

Festival unit-selection

MARY TTS unit selection

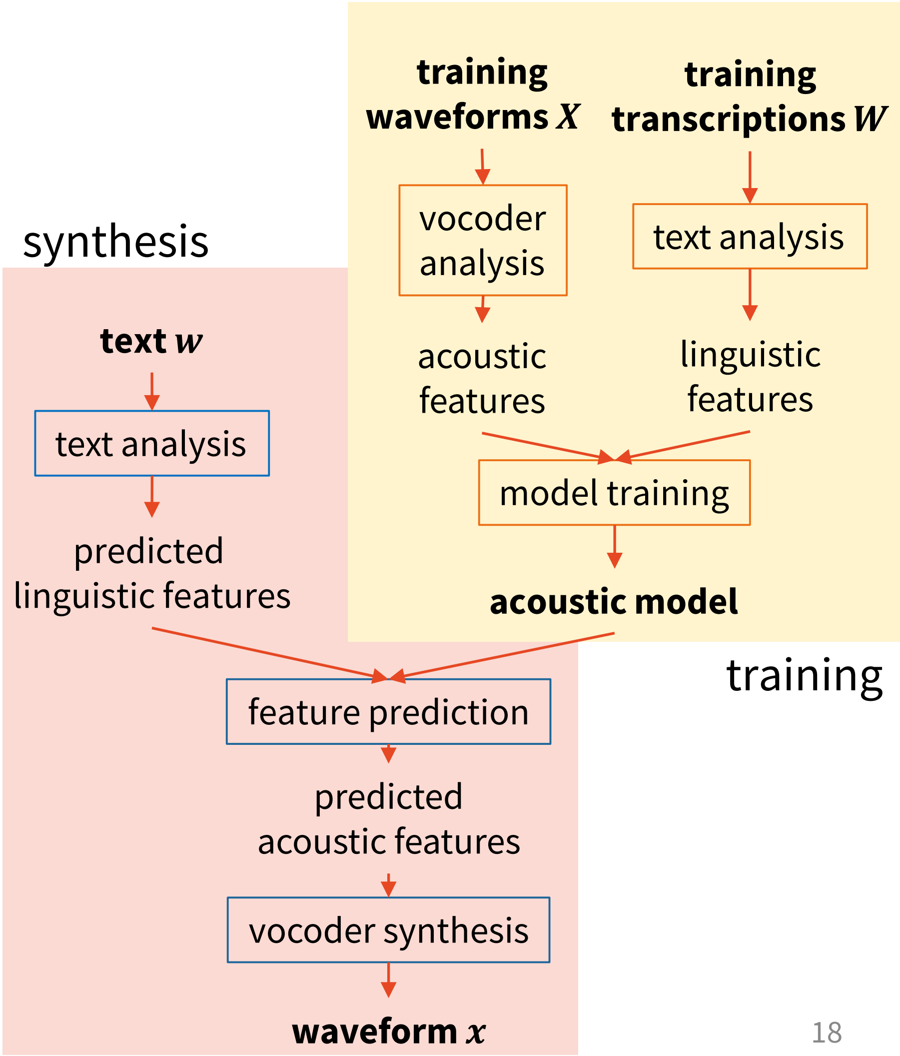
IBM Watson concatenative

Google concatenative

http://www.cs.cmu.edu/~awb/festival_demos/general.html
http://mary.dfki.de/
https://text-to-speech-demo.ng.bluemix.net/
https://deepmind.com/blog/wavenet-generative-model-raw-audio/

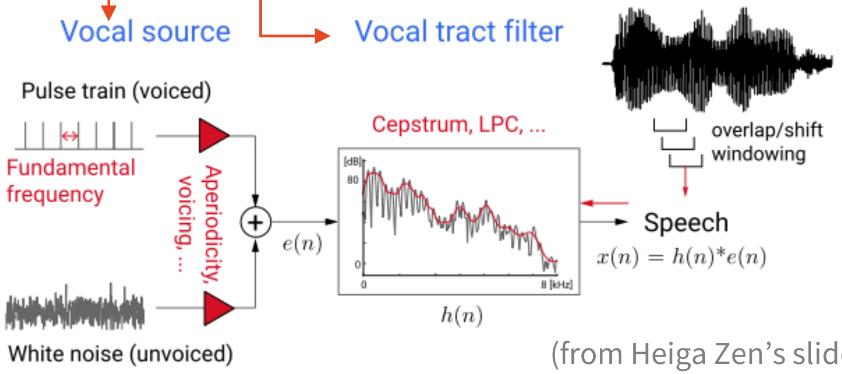
Model-based Parametric Synthesis

- trying to be more flexible, less resource-hungry than unit selection
- similar approach to formant-based – modelling
  - but this time learned statistically from a corpus
- inverse of model-based ASR (next lecture)
- ideal: model $p(x|w, X, W)$
  - auxiliary representations – features
  - approximate by step-by-step maximization:
    - extract features from corpus (acoustic, linguistic)
    - learn model based on features
    - predict features given text (linguistic, then acoustic)
    - synthesize given features

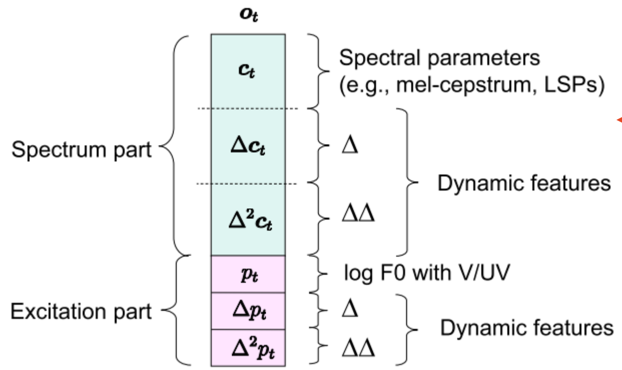


Features for model-based synthesis

- Acoustics: piecewise stationary source-filter model
  - spectrum (filter/resonance frequencies): typically MFCCs, Δ, ΔΔ
  - excitation (sound source): voiced/unvoiced, log F0, Δ, ΔΔ

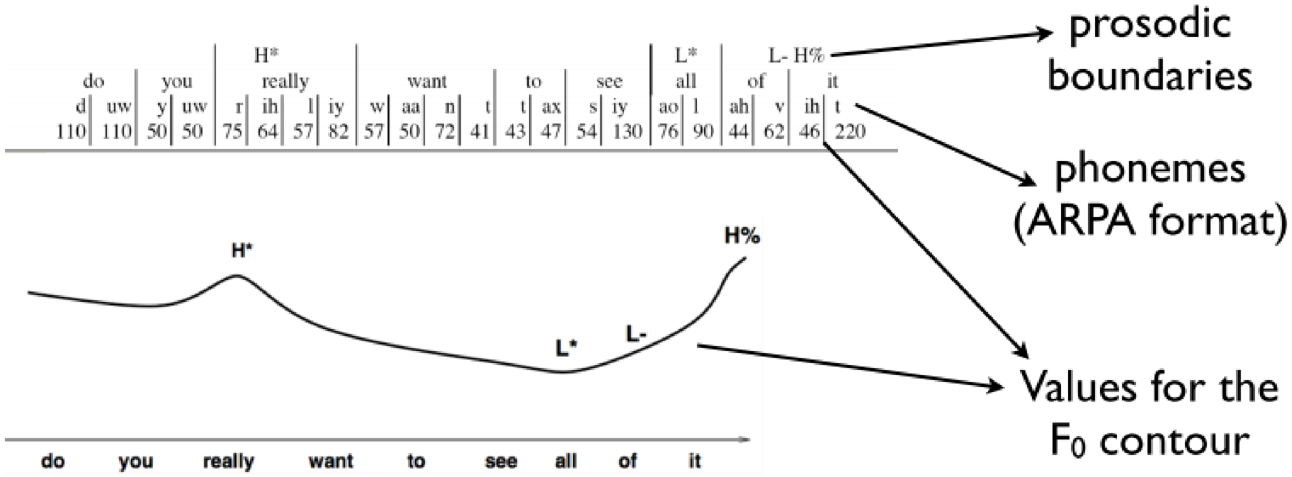


(from Heiga Zen's slides)



(Tokuda et al., 2013)

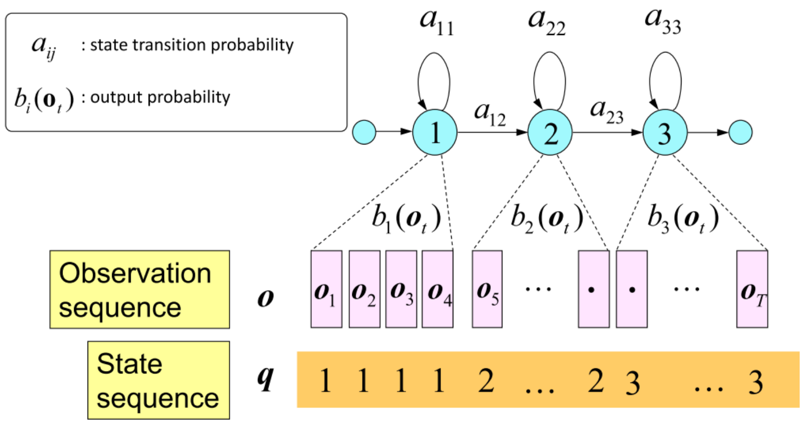
- Linguistics:
  - phonemes
  - stress
  - pitch



(from Pierre Lison's slides)

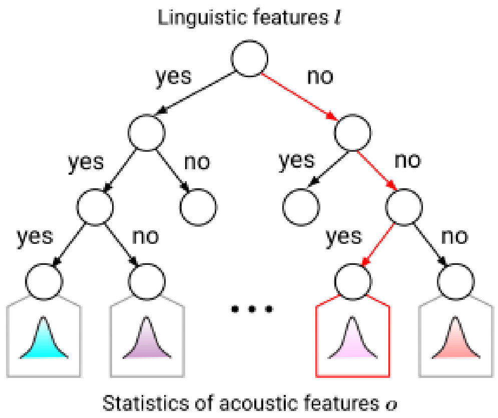
HMM-based Synthesis

- Using HMMs as the speech model
- Context-dependent phoneme-level HMMs
  - concatenated into a big utterance-level HMM
  - transition & emission probabilities
– multivariate Gaussian distributions
  - loops – handling different phoneme lengths



(Tokuda et al., 2013)

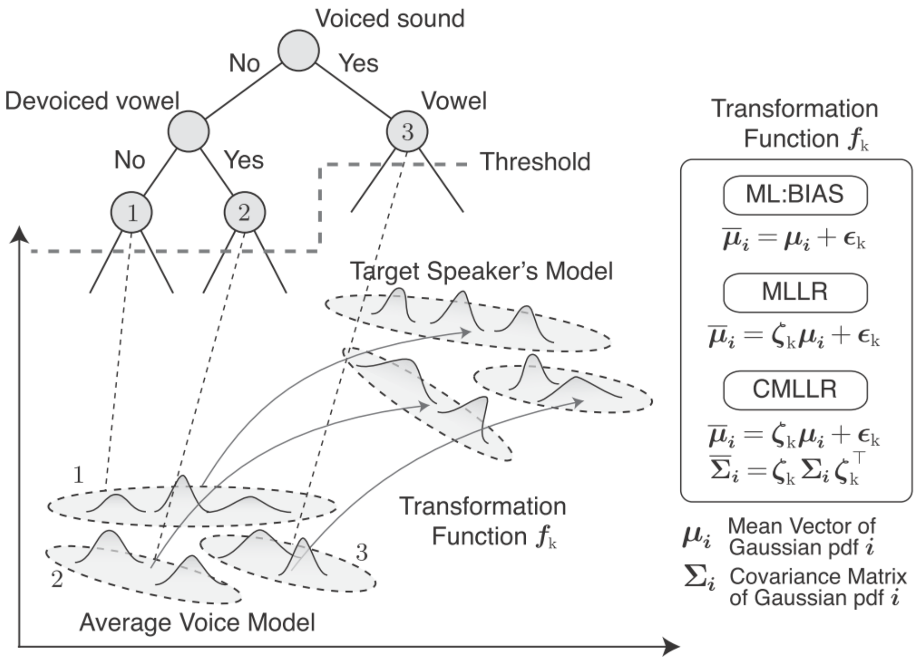
- Too many possible contexts → use decision-tree-based clustering
  - ~10M possible context combinations
  - regression trees (outputs = real-valued Gaussian parameters)
- Generating from this would result in step-wise sequence
  - sample from each Gaussian, wait a few ms, sample…
  - → this is where Δ, ΔΔ are used



(from Heiga Zen's slides)

HMM-based Synthesis

- Pros vs. concatenative:
  - small data footprint
  - robust to data sparsity
  - flexible – can change voice characteristics easily
- Con:
  - lowered segmental naturalness



(Tokuda et al., 2013)

FLite/HTS
(various settings)

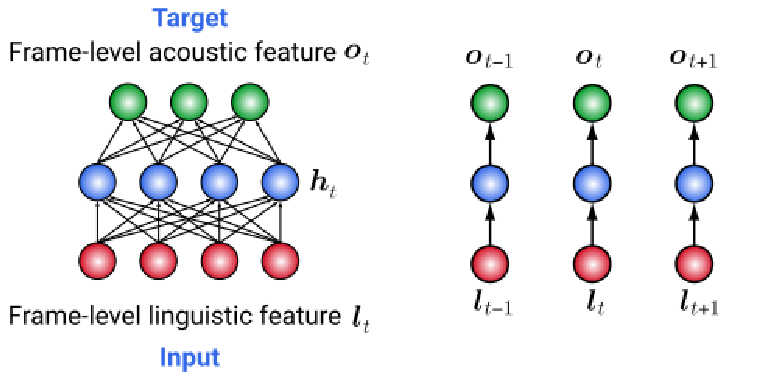
http://flite-hts-engine.sp.nitech.ac.jp/index.php

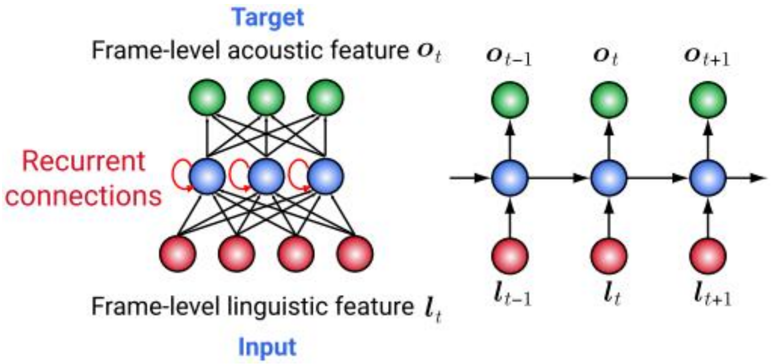
MARY TTS
HSMM-based

http://mary.dfki.de/

NN-based synthesis

- Replacing clunky HMMs and decision trees with NNs
- Basic – feed forward networks
  - predict conditional expectation of acoustic features given linguistic features at current frame
  - trained based on mean squared error
- Improvement – RNNs
  - same, but conditioned on current & previous frames
  - predicts smoother outputs (given temporal dependencies)
- NNs allow better features (e.g. raw spectrum)
  - more data-efficient than HMMs
- This is current production quality TTS





(from Heiga Zen's slides)

Google LSTM parametric

IBM Watson DNN

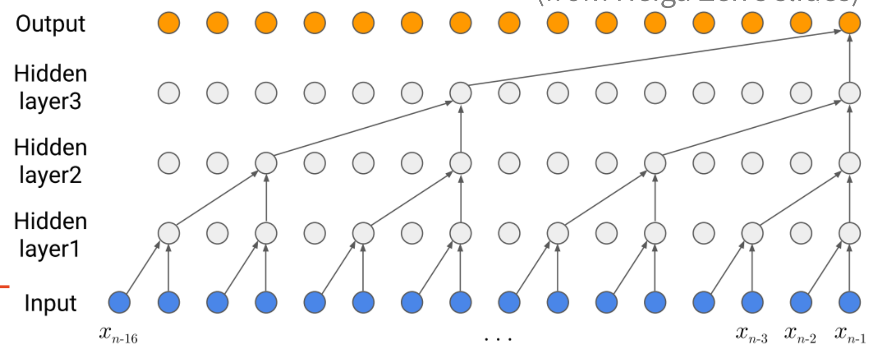
https://deepmind.com/blog/wavenet-generative-model-raw-audio/
https://text-to-speech-demo.ng.bluemix.net/

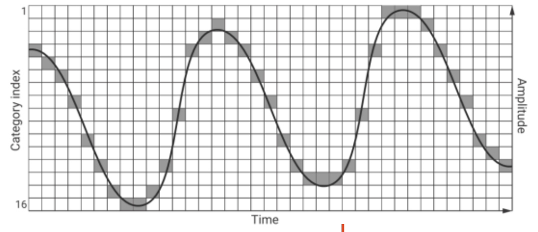
WaveNet

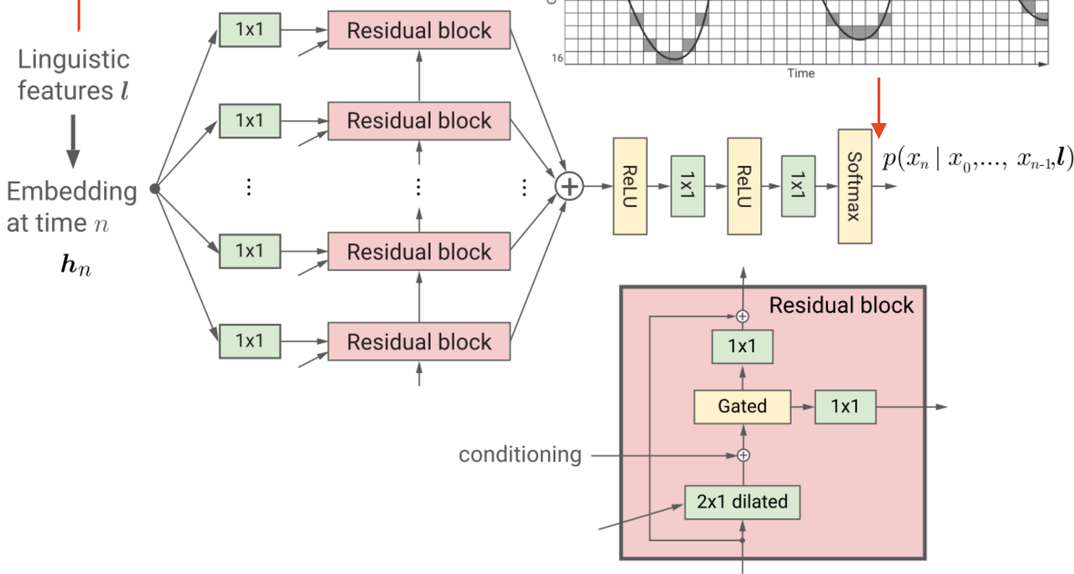
(van den Oord et al., 2016)
https://arxiv.org/abs/1609.03499
https://deepmind.com/blog/wavenet-generative-model-raw-audio/

- Removing acoustic features – direct waveform generation
  - no need for spectrum
- Based on convolutional NNs
  - 16k steps/sec → need very long dependencies
  - **dilated convolution** – skipping steps
  - exponential receptive field w.r.t. # of layers
  - conditioned on linguistic features
  - predicting quantized waves using softmax
- Not tied to ±stationary frames
  - can generate highly non-linear waves
- Very natural, Google's top offering now

(from Heiga Zen's slides)







Google WaveNet

Tacotron

(Wang et al., 2017)
https://arxiv.org/abs/1703.10135
https://google.github.io/tacotron/

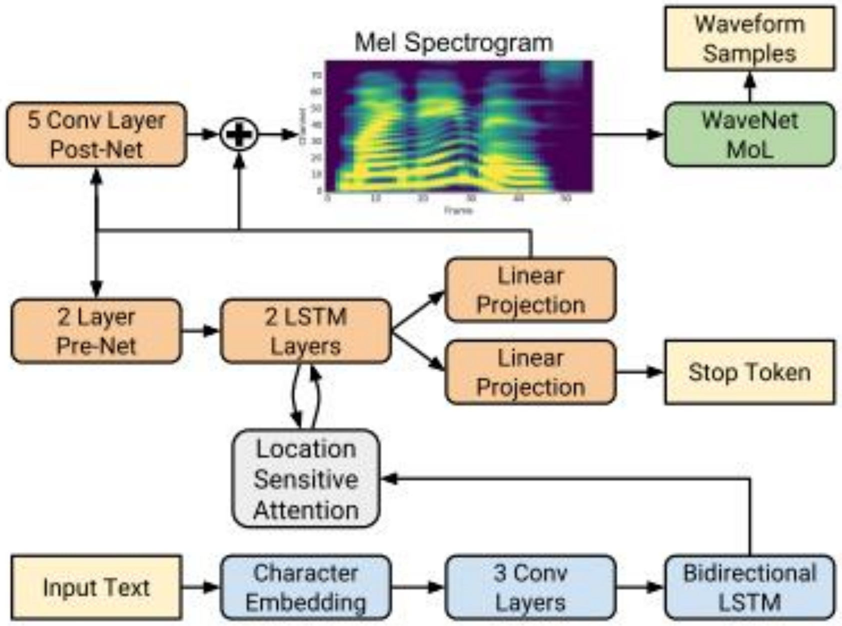
(Shen et al., 2018)
http://arxiv.org/abs/1712.05884

vanilla seq2seq

Tacotron 1

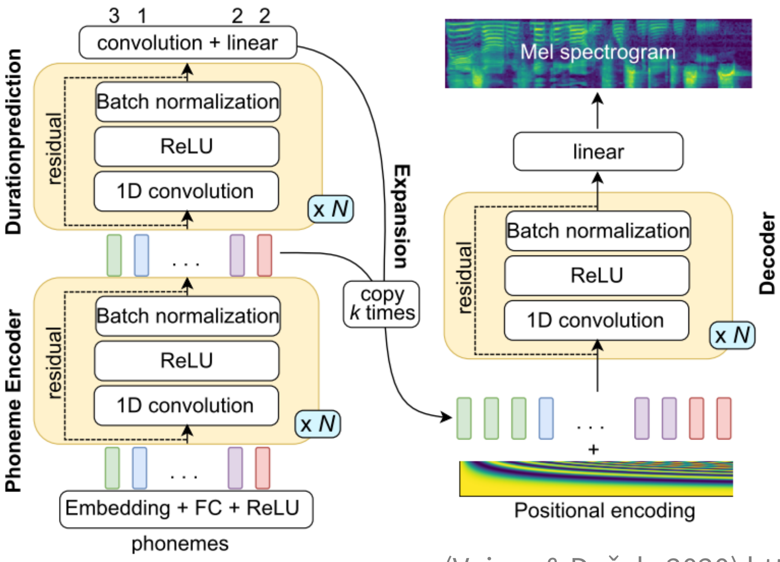
Tacotron 2

- Different approach: removing linguistic features
  - trained directly from pairs of waveforms & transcriptions
  - generates spectrograms (at frame level)
    - T1 – linear: Griffin-Lim conversion (estimate missing wave phase)
    - T2 – mel scale: needs something better, such as WaveNet, better quality
- Based on seq2seq models with attention
  - adapted – just LSTMs don't work well
  - T2 – encoder: convolutional + LSTM
  - T2 – decoder:
    - linear pre-net (scaling down previous spectrum)
    - LSTM + attention
    - stop classification
    - post-net – convolutions: produce spectrum
  - T1: similar, more complex (custom layers)



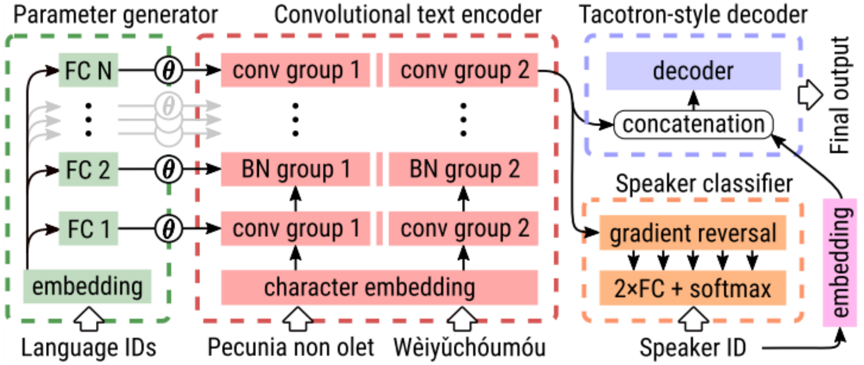
Extensions: Faster, Multilingual

- Faster: All convolutional (no RNNs)
  - predicting mel spectrograms
  - encode phonemes
  - predict duration (k frames)
  - copy encodings k times & decode
- Multilingual: Meta-learning
  - predict network parameters for each language with a smaller network
  - added speaker ID – multi-speaker
  - consistent voice with multiple languages



SpeedySpeech & MelGAN vocoder

(Vainer & Dušek, 2020) https://arxiv.org/abs/2008.03802
https://github.com/janvainer/speedyspeech



German & Chinese

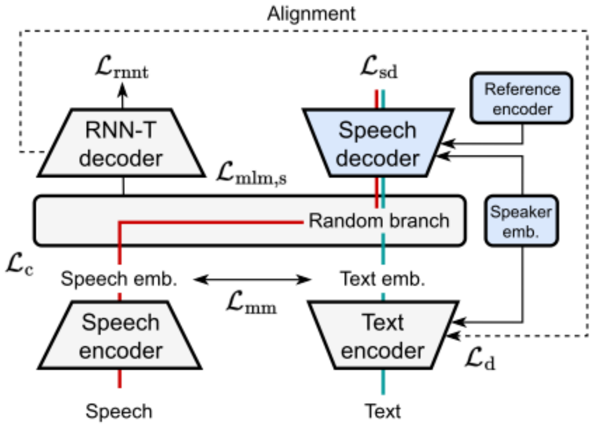
French & Russian

(Nekvinda & Dušek, 2020) https://arxiv.org/abs/2008.00768
https://github.com/Tomiinek/Multilingual_Text_to_Speech

Current works

(Tan et al., 2021) http://arxiv.org/abs/2106.15561

- Pretraining (voice & text) (Chung et al., 2019) https://arxiv.org/abs/1808.10128 (Jia et al., 2021) https://arxiv.org/abs/2103.15060
- Adversarial training (Kim et al., 2021) https://arxiv.org/abs/2106.06103
- Voice conversion (Biadsy et al., 2019) https://arxiv.org/abs/1904.04169 (Chou et al., 2019) https://arxiv.org/abs/1904.05742
- Unseen languages (Saeki et al., 2023) http://arxiv.org/abs/2301.12596
- Expressive synthesis (Valle et al., 2020) https://arxiv.org/abs/1910.11997
- Personality match (Gao et al., 2023) https://aclanthology.org/2023.sigdial-1.36
- Singing https://en.wikipedia.org/wiki/Suno_AI …
- Speech-to-speech, multimodal models https://ai.meta.com/blog/seamless-m4t/ https://openai.com/index/hello-gpt-4o/ …



(a) Speech-text pretraining on paired TTS data

(Saeki et al., 2023) http://arxiv.org/abs/2210.15447

Summary

- Speech production
  - “source-filter”: air + vocal cords vibration + resonation in vocal tract
  - sounds/phones, phonemes
  - consonants & vocals
  - spectrum, formants
  - pitch, stress
- Text-to-speech system architectures
  - rule/formant-based
  - concatenative – diphone, unit selection
  - model-based parametric: HMM, NNs
  - end-to-end neural: WaveNet, Tacotron (&similar)

Thanks

Labs in 10 mins

Contact us:

https://ufaldsg.slack.com/
{odusek,schmidtova,hudecek}@ufal.mff.cuni.cz
Skype/Meet/Zoom (by agreement)

Get these slides here:

http://ufal.cz/npfl123

References/Inspiration/Further:

- Heiga Zen's lecture (MIT 2017): https://ai.google/research/pubs/pub45882, https://youtu.be/nsrSrYtKkT8
- Tokuda et al. (2013): Speech synthesis based on Hidden Markov Models: http://ieeexplore.ieee.org/document/6495700/
- Pierre Lison's slides (Oslo University): https://www.uio.no/studier/emner/matnat/ifi/INF5820/h14/timeplan/index.html
- Dennis H. Klatt (1987): Review of text-to-speech conversion for English: http://asa.scitation.org/doi/10.1121/1.395275
- Heiga Zen's lecture (ASRU 2015): https://ai.google/research/pubs/pub44630
- Kathariina Makhonen's lecture notes (Tampere University): http://www.cs.tut.fi/kurssit/SGN-4010/puhesynteesi_en.pdf
- Raul Fernandez's lecture (2011): http://www.cs.columbia.edu/~ecooper/tts/SS_Lecture_CUNY_noaudio.pdf
- Sami Lemmetty's MSc. thesis (Helsinki Tech, 1999): http://research.spa.aalto.fi/publications/theses/lemmetty_mst/thesis.pdf
- BBC Radio 4 – Lucy Hawking on TTS history (2013): https://youtu.be/097K1uMIPyQ