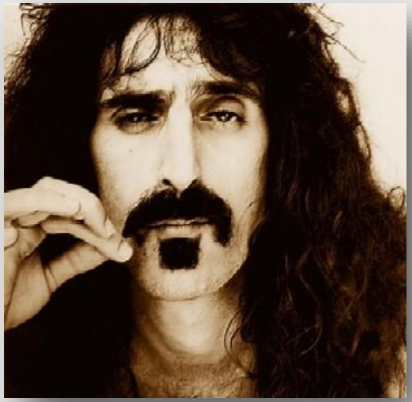
# speech synthesis

CSC401/2511 – Natural Language Computing – Spring 2021
Lecture 9
University of Toronto

# This lecture

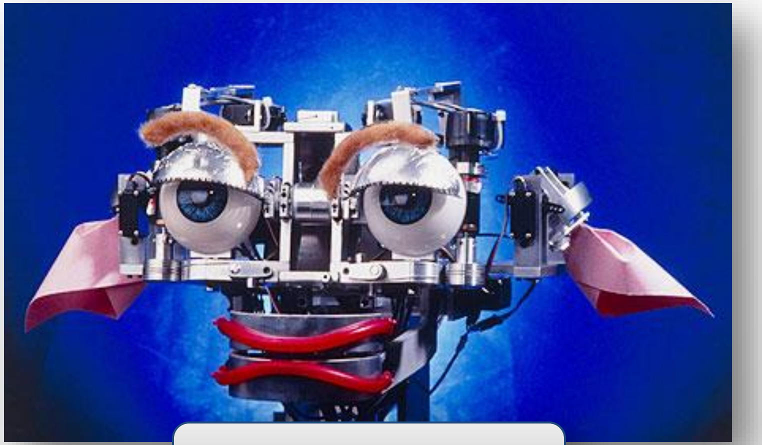
- Some text-to-speech architectures.
- Some text-to-speech components.
- **Text-to-speech**: *n.* the conversion of electronic text into equivalent, audible speech waveforms.

# Insight?



Frank Zappa

The computer can't tell you the emotional story. It can give you the exact mathematical design, but what's missing is the eyebrows.



Kismet

# Components of TTS systems

- Some components are common to all TTS systems, namely:
  1. **Text analysis**.
     - Text normalization
     - Homograph ("same spelling") disambiguation
     - Grapheme-to-phoneme (letter-to-sound)
     - Intonation (prosody)
  2. **Waveform generation**.
     - Unit and diphone selection.

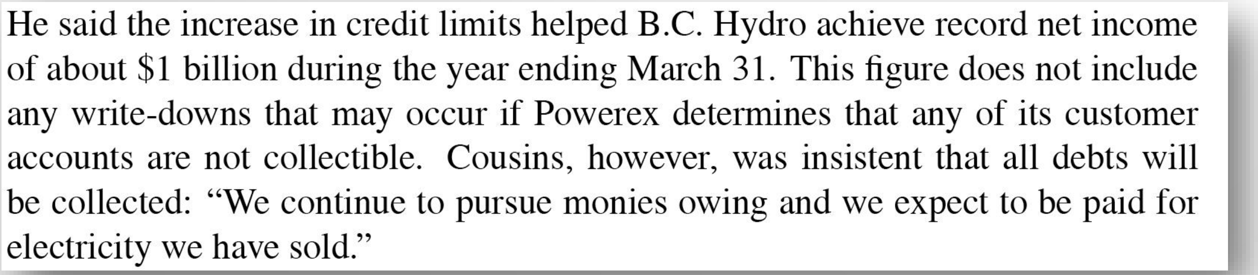
UNIVERSITY OF TORONTO

# Text analysis

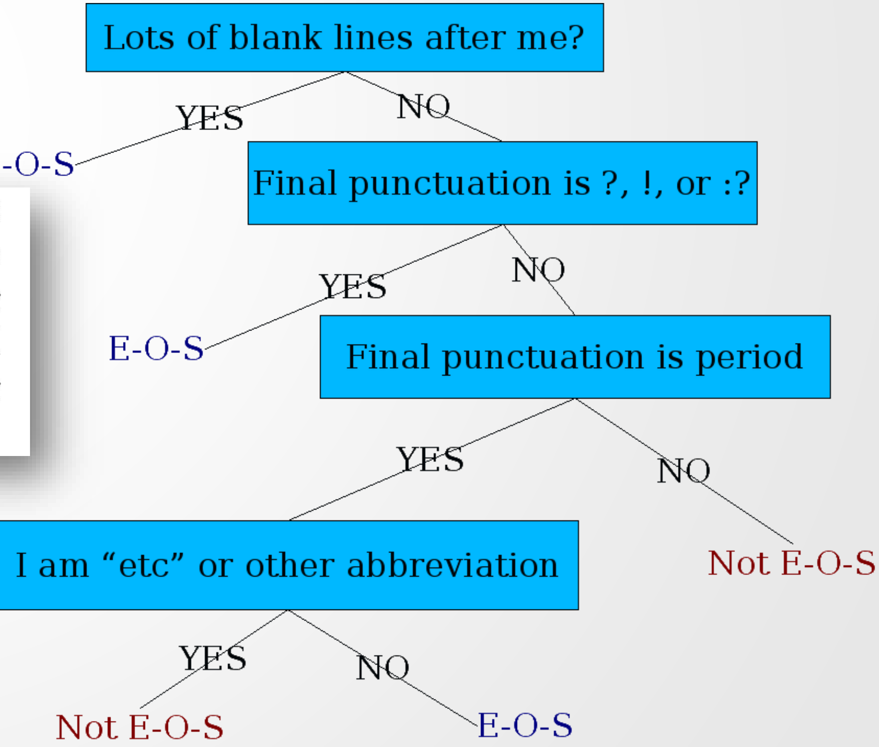
How do we analyze the text
we want to read?

# Text analysis

- First, we **normalize** the text. This involves splitting sentences, tokenizing, and sometimes chunking.
- You can also induce decision trees automatically.



He said the increase in credit limits helped B.C. Hydro achieve record net income of about $1 billion during the year ending March 31. This figure does not include any write-downs that may occur if Powerex determines that any of its customer accounts are not collectible. Cousins, however, was insistent that all debts will be collected: "We continue to pursue monies owing and we expect to be paid for electricity we have sold."


Lots of blank lines after me?
YES
NO
E-O-S
Final punctuation is ?, !, or :?
YES
NO
E-O-S
Final punctuation is period
YES
NO
I am "etc" or other abbreviation
Not E-O-S
YES
NO
Not E-O-S
E-O-S

# Identifying the types of tokens

- Pronunciation of each token can depend on its type or usage.
  - e.g., “1867” is
    - “*eighteen sixty seven*” if it’s a **year**,
    - “*one eight six seven*” if it’s in a **phone number**,
    - “*one thousand eight hundred and sixty seven*” if it’s a **quantifier**.
  - e.g., “25” is
    - “*twenty five*” if it’s an **age**,
    - “*twenty fifth*” if it’s a **day of the month**.

# Homograph disambiguation

- **Homograph**: *n.* a set of words that share the same spelling but have different meanings or pronunciations.
  - E.g.,
    - *“**close** the door! The monsters are getting **close**!”*
    - *“I **object** to that horrible **object**!”*
    - *“I **refuse** to take that **refuse**!”*
    - *“I’m **content** with the **content**.”*

- It’s important to pronounce these homographs correctly, or the meaning will be lost.

# Homograph disambiguation

- Homographs can often be distinguished by their part-of-speech.
  - E.g., “live” as a verb (/l ih v/) or an adjective (/l ay v/).

| Verb | Noun |
|---|---|
| Use /y uw z/ | Use /y uw s/ |
| House /h aw z/ | House /h aw s/ |
| reCORD | REcord |
| disCOUNT | DIScount |
| … | … |

# From words to phonemes

- There are at least two methods to convert words to sequences of phonemes:
  - Dictionary lookup.
  - Letter-to-sound (LTS) rules (if the word is not in the dictionary).

- Modern systems tend to use a **combination** of approaches, relying on **large dictionaries and samples** for common words, but using **rules** to assemble unknown words.

# Pronunciation dictionaries: CMU

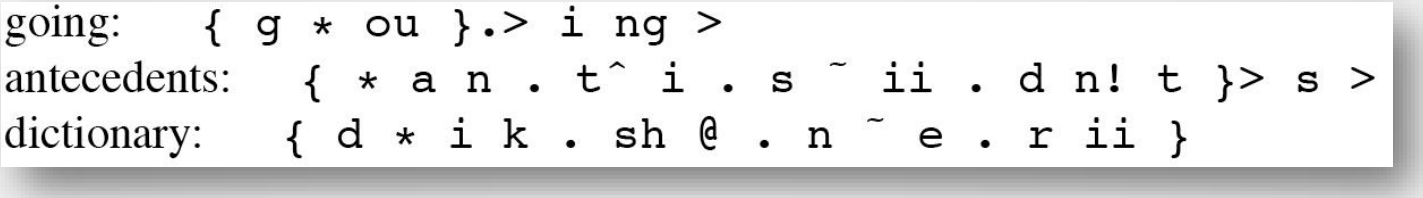
- The **CMU dictionary** has 127K words.

- Unfortunately,
  - It only contains *American* pronunciations,
  - It does **not** contain *syllable boundaries* (for timing),
  - It does **not** contain *parts-of-speech*

    (it contains no knowledge of homographs),
  - It does not distinguish case,
    - E.g. 'US' is transcribed as both /ah s/ and /y uw eh s/.

# Other pronunciation dictionaries

- The UNISYN dictionary has about 110K words, and includes syllabification, stress, and morphology.

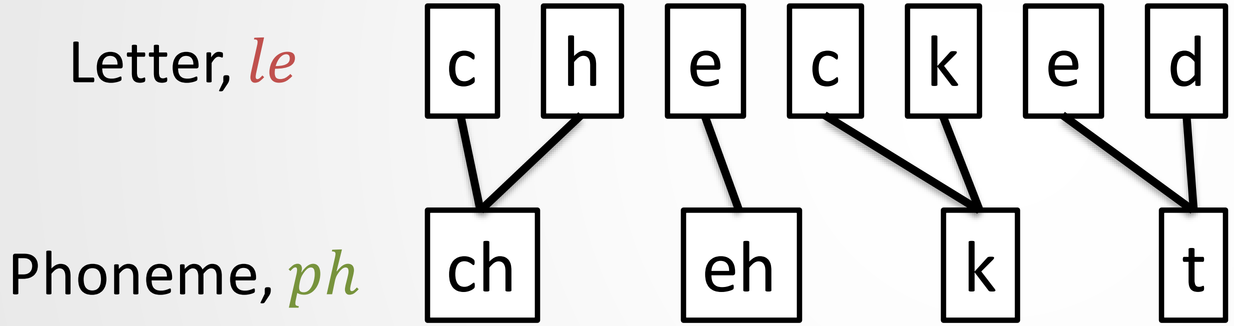


```
going:        { g * ou }.> i ng >
antecedents:    { * a n . t^ i . s ~ ii . d n! t }> s >
dictionary:    { d * i k . sh @ . n ~ e . r ii }
```

- **Unknown words** (a.k.a., “out of vocabulary” (OOV)) *typically* increase with the square root of the number of words in a new, previously unseen text.
- Commercial systems often use dictionaries, but back off to stochastic routines when necessary. *Such as…*

# Letter-to-sound rules

- First, we must align letters and phonemes,
- If you have access to these alignments, you can learn these with maximum likelihood estimation, e.g.,



$$P(ph|le) = \frac{Count(ph \wedge le)}{Count(le)}$$

- If you don't have these alignments, they can be learned using **expectation-maximization** as we saw with, e.g., statistical machine translation.

# Letter-to-sound rules

- Alignments can be improved by using hand-written rules that restrict the translation of letters to phonemes (e.g., C goes to /k, ch, s, sh/, or W goes to /w, v, f/).

- Some words have to be dealt with specifically, since their spelling is so different from their pronunciation.
  - E.g., abbreviations: "dept"→ /d ih p aa r t m ah n t/
    "wtf" →/w aw dh ae t s f ah n iy/

# Prosody

- Once you have a phoneme sequence, you may need to adjust other **acoustic** characteristics, based on the **semantic** context.

- Prosodic phrasing:
  - You need to mark **phrase boundaries**,
  - You need to **emphasize** certain syllables by modifying either F0, loudness, or the duration of some phonemes.

- In neural networks and HMMs, F0 can be learned (and hence sampled) simultaneously with phonemes.

# Three aspects for prosody in TTS

- **Prominence**: some syllables or words are more prominent than others, especially content words.
- **Structure**: Sentences have inherent prosodic structure. Some words group naturally together, others require a noticeable disjunction.
- **Tune**: To sound natural, one has to account for the intonational melody of an utterance.

*These are reasons to modify prosody, not the way prosody is modified…*

# Emphasis in noun phrases

- Proper names: the emphasis is often on the right-most word.
  - E.g., *New York CITY; Paris, FRANCE*

- Noun-noun compounds: emphasis is often on the left noun.
  - E.g., *TABLE lamp; DISK drive,*

- Adjective-noun compounds: stress on the noun
  - E.g., *large HOUSE; new CAR*

- Counterexamples exist, but with some predictability…
  - *MEDICAL building; cherry PIE*

# Waveform Generation

How do we actually produce the sounds, given the phonemes?

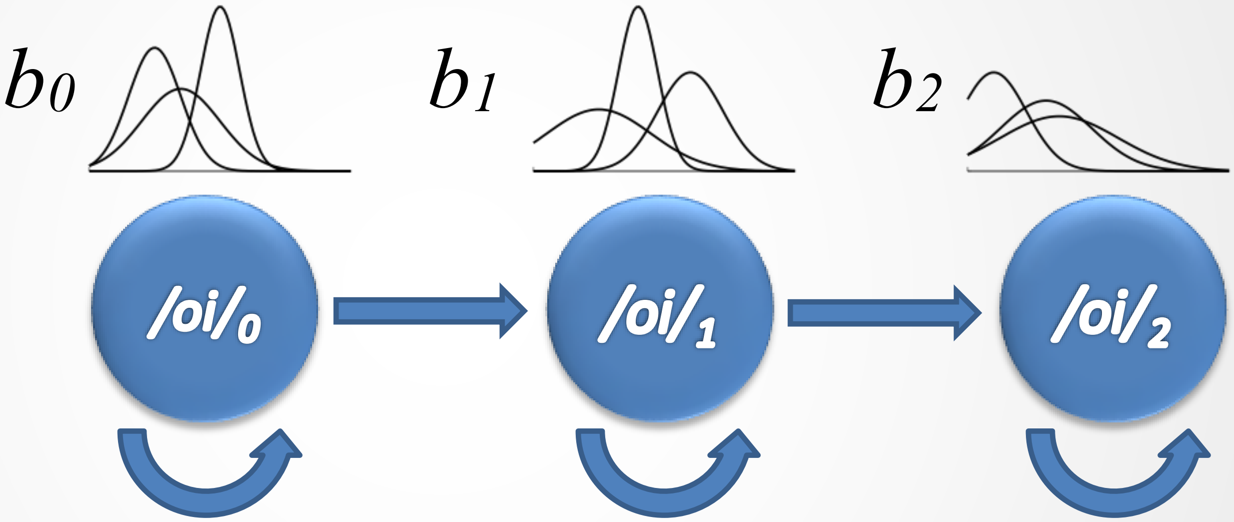
# Standard TTS architectures

1. **Formant synthesis**
   - Synthesizes acoustics based on rules and filters
2. **Concatenative synthesis**
   - Uses databases of stored speech to assemble audio.
3. **Articulatory synthesis**
   - Models the movements and acoustics of the vocal tract.
4. **Statistical model synthesis**
   - Samples from some stochastic model

Appendix

# 4. Synthesis from HMMs

- Use a trained HMM and sample from it.



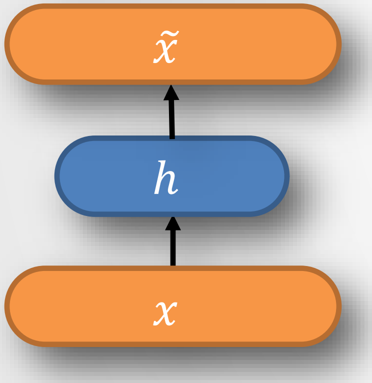
tristate phoneme model (e.g., */oi/*)

- Festival (http://www-2.cs.cmu.edu/~awb/festival_demos/index.html)

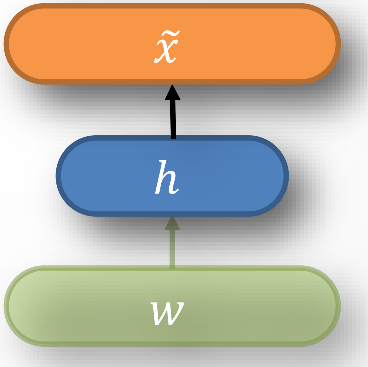
Y.-J. Wu and K. Tokuda (2008) Minimum generation error training with direct log spectral distortion on LSPs for HMM-based speech synthesis. In Proc. Interspeech, pages 577–580, 2008.

# 4. Synthesis from NNs

- RNNs can predict smoothly-changing acoustic features.
  - It can be difficult to learn high-dimensional acoustic features (e.g., MFCCs or raw spectra).
  - Solution? Learn better features using an **autoencoder**.



Train a NN that learns to recreate its own input audio signal $x$



And later use the resulting latent features to learn a mapping from words $w$

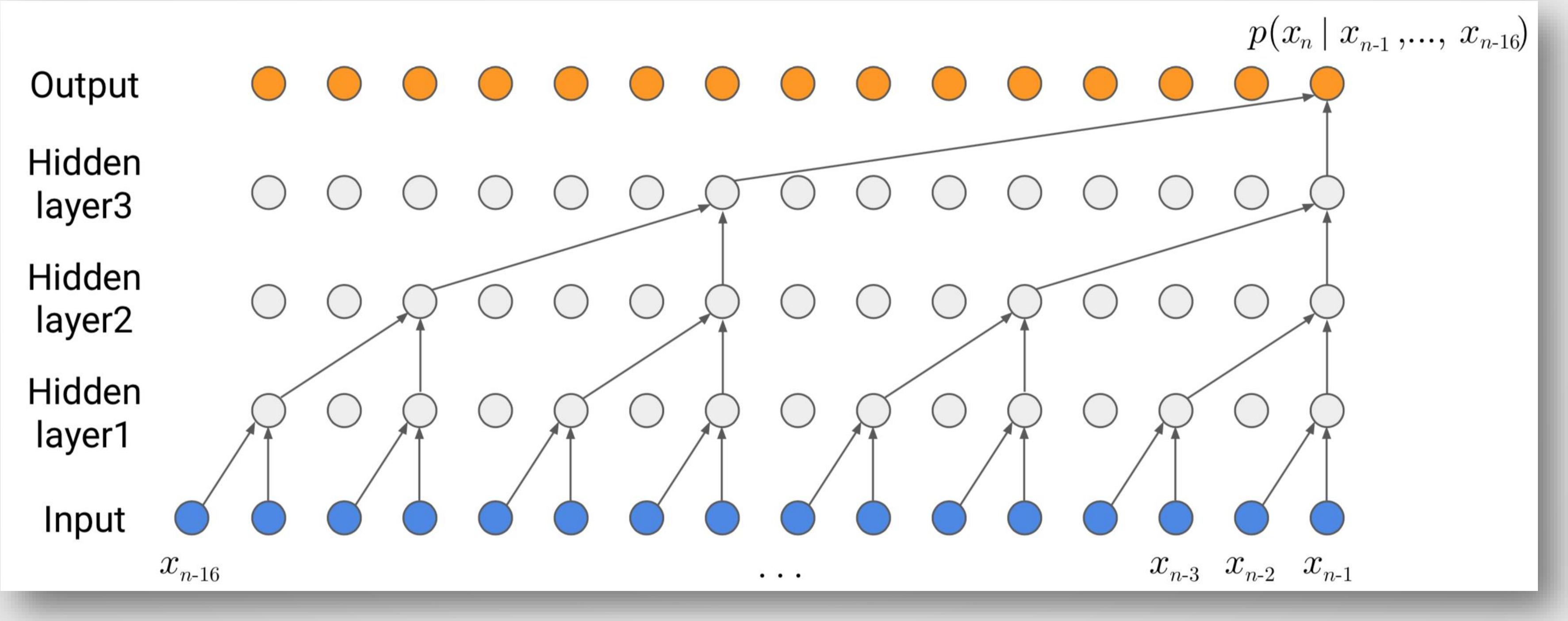
Y. Fan, Y. Qian, F.-L. Xie, and F. Soong. (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In Proc. Interspeech, pages 1964–1968.

H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak. (2016) Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices.In Proc. Interspeech.

S. Takaki and J. Yamagishi (2016) A deep auto-encoder based low-dimensional feature extraction from FFT spectral envelopes for statistical parametric speech synthesis. In Proc. ICASSP, pages 5535–5539.

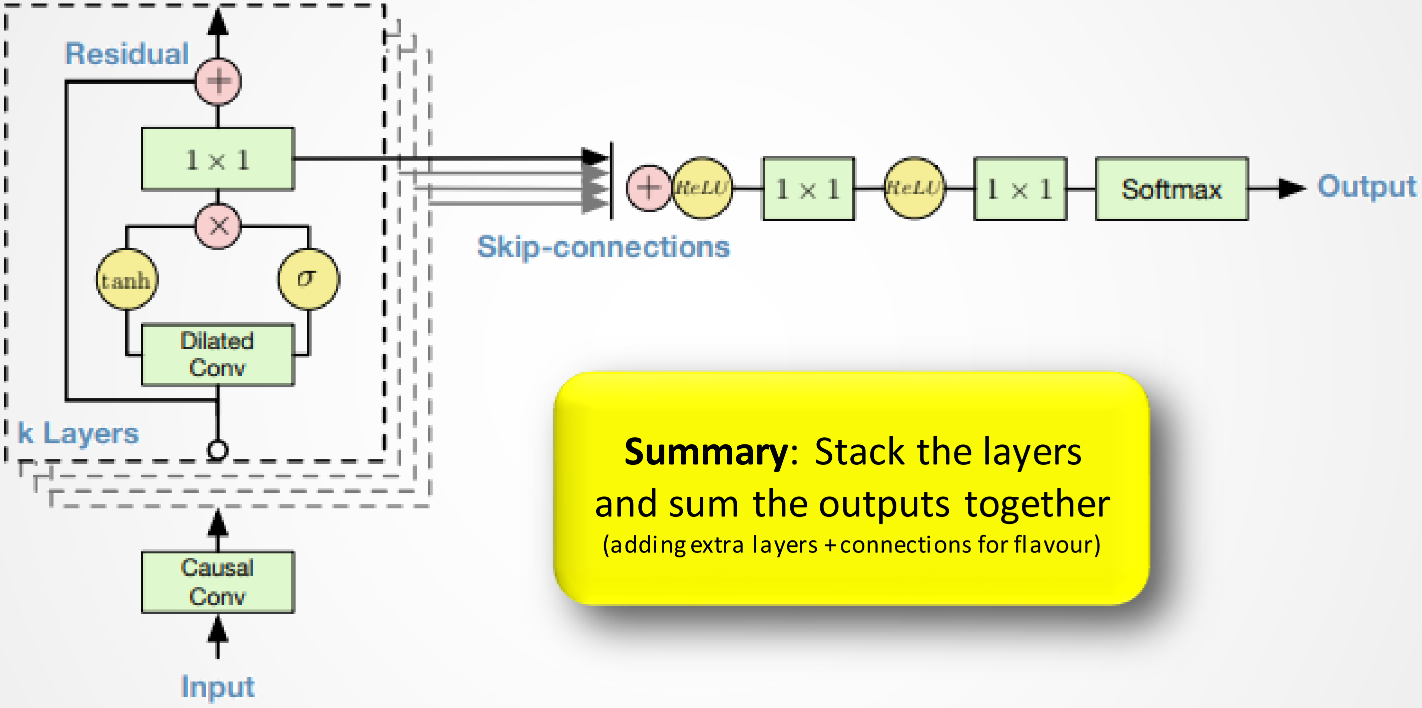
# 4. Synthesis from NNs

- If $x$ is raw audio, and we use a modest window (e.g., 100ms), your input can be a 1000+ dimensional dense vector, which can be too long for an RNN (or autoencoder).
  - Solution? Exponentially increase receptive field across layers.



Oord, A. et al. (2016). WaveNet: A Generative Model for Raw Audio, 1–15. http://arxiv.org/abs/1609.03499

# Aside – WaveNet Residual layers



**Summary**: Stack the layers and sum the outputs together
(adding extra layers + connections for flavour)

Figure 4: Overview of the residual block and the entire architecture.

Oord, A. et al. (2016). WaveNet: A Generative Model for Raw Audio, 1–15. http://arxiv.org/abs/1609.03499

# 4. Synthesis from NNs (TacoTron)

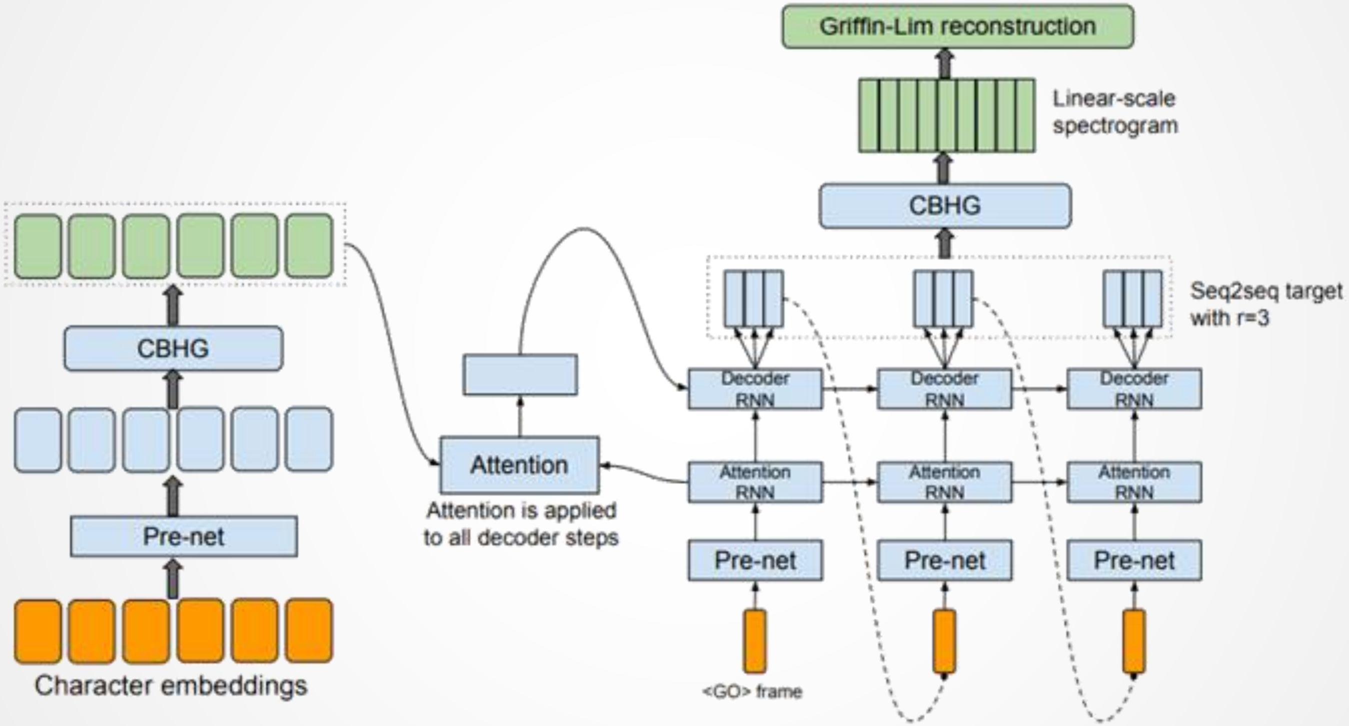
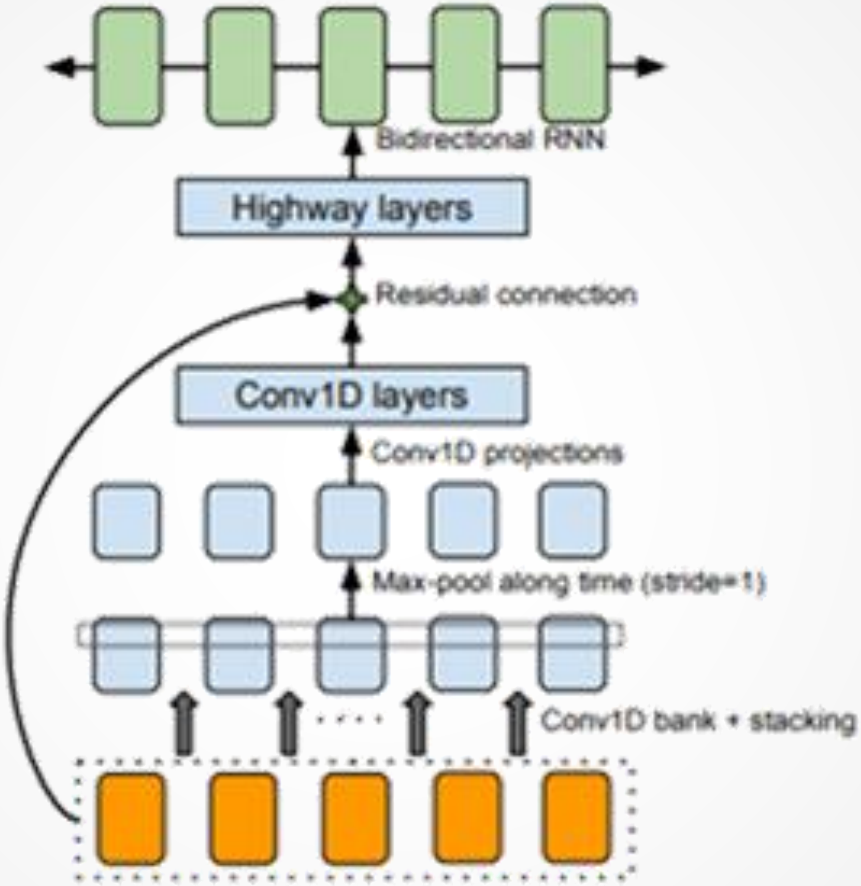


Figure 1: *Model architecture. The model takes characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesize speech.*

Wang, Y., et al (2017). Tacotron: Towards end-To-end speech synthesis. Proceedings of the INTERSPEECH, 2017-August, 4006–4010. https://doi.org/10.21437/Interspeech.2017-1452

# Aside – CBHG blocks (TacoTron)

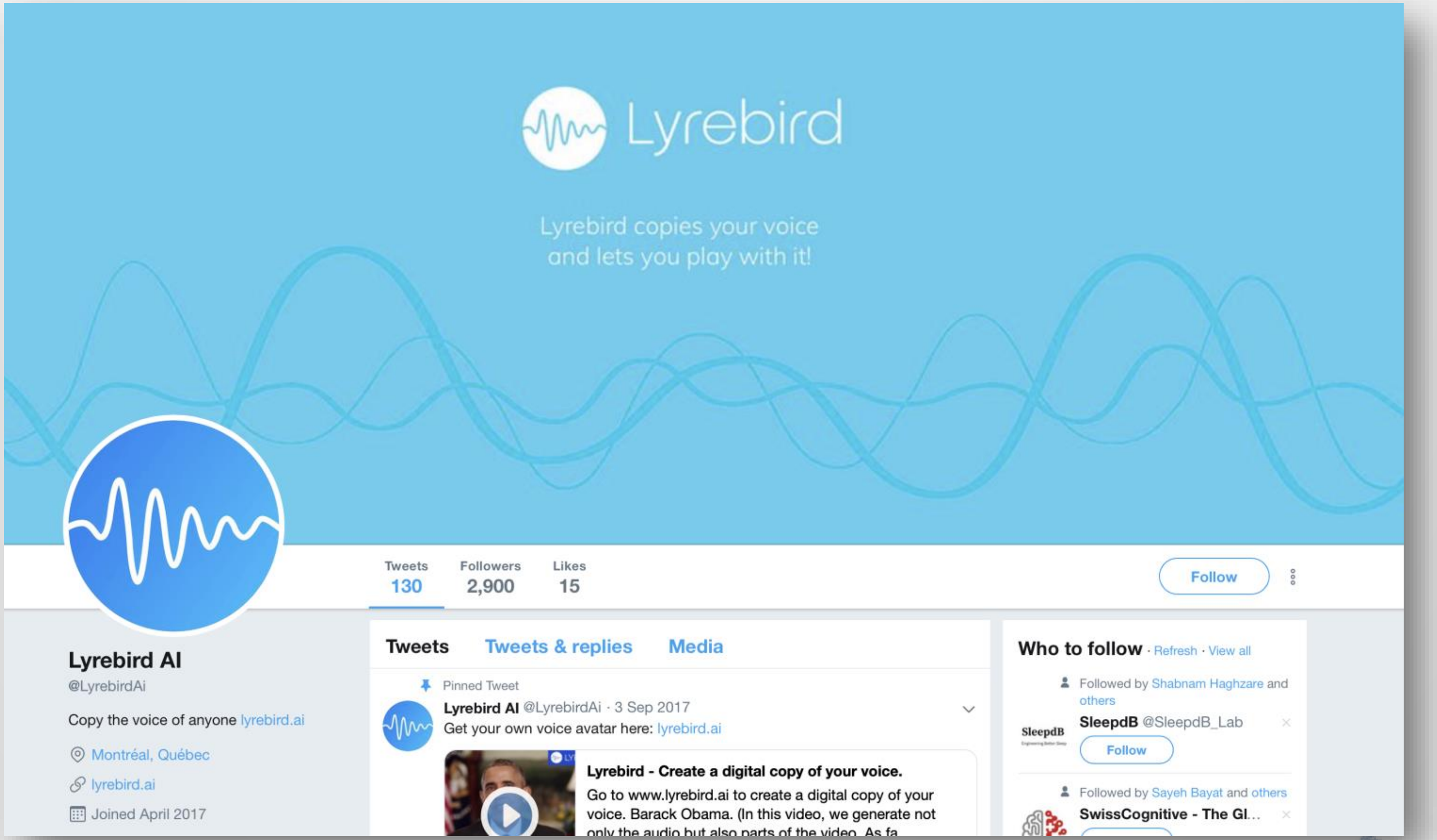


CBHG (1D convolution + highway + bidirectional GRU)

Wang, Y., et al (2017). Tacotron: Towards end-To-end speech synthesis. Proceedings of the INTERSPEECH, 2017-August, 4006–4010. https://doi.org/10.21437/Interspeech.2017-1452

# 4. Synthesis from NNs

- Typically, neural networks will be employed with other techniques to avoid unpredictability.
- They also open up the potential for dangerous **deep fakes**.



Lyrebird copies your voice and lets you play with it!

**Lyrebird AI**
@LyrebirdAi

Copy the voice of anyone lyrebird.ai

Montréal, Québec
lyrebird.ai
Joined April 2017

Pinned Tweet
**Lyrebird AI** @LyrebirdAi · 3 Sep 2017
Get your own voice avatar here: lyrebird.ai

**Lyrebird - Create a digital copy of your voice.**
Go to www.lyrebird.ai to create a digital copy of your voice. Barack Obama. (In this video, we generate not only the audio but also parts of the video. As fa

Followed by Shabnam Haghzare and others
**SleepdB** @SleepdB_Lab

Followed by Sayeh Bayat and others
**SwissCognitive - The Gl...**

# Evaluation of TTS

How do we declare victory?

# Evaluation of TTS

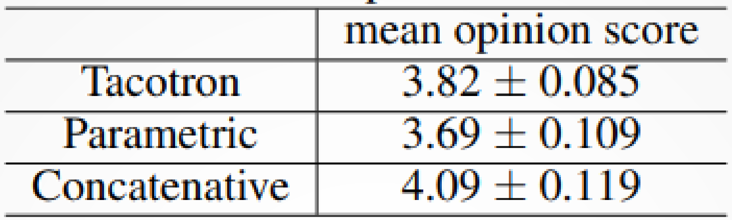
- Intelligibility tests.
  - E.g., the **diagnostic rhyme** test involves humans identifying synthetic speech from two word choices that differ by a single phonetic feature (e.g., voicing, nasality).
    - E.g., “**d**ense” vs. “**t**ense”, “ma**z**e” vs. “ma**c**e”

- Mean opinion score
  - Have listeners rate synthetic speech on a Likert-like scale (i.e., a goodness-badness scale).

http://www.synsig.org/index.php/Blizzard_Challenge_2013_Rules

# Evaluation of TTS

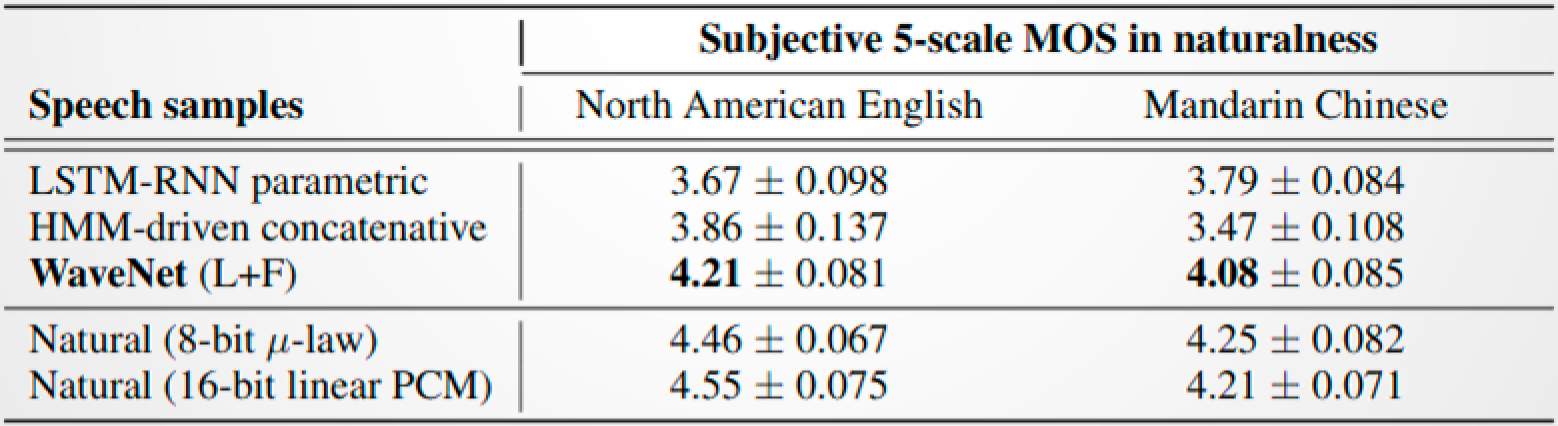
Table 2: 5-scale mean opinion score evaluation.



| | mean opinion score |
|---|---|
| Tacotron | 3.82 ± 0.085 |
| Parametric | 3.69 ± 0.109 |
| Concatenative | 4.09 ± 0.119 |

Mean opinion scores (1-5) from [unknown number of] ratings on 100 test sentences.

Wang, Y., et al (2017). Tacotron: Towards end-To-end speech synthesis. Proceedings of the INTERSPEECH, 2017-August, 4006–4010. https://doi.org/10.21437/Interspeech.2017-1452



| | **Subjective 5-scale MOS in naturalness** | |
|---|---|---|
| **Speech samples** | North American English | Mandarin Chinese |
| LSTM-RNN parametric | 3.67 ± 0.098 | 3.79 ± 0.084 |
| HMM-driven concatenative | 3.86 ± 0.137 | 3.47 ± 0.108 |
| **WaveNet** (L+F) | **4.21** ± 0.081 | **4.08** ± 0.085 |
| Natural (8-bit $\mu$-law) | 4.46 ± 0.067 | 4.25 ± 0.082 |
| Natural (16-bit linear PCM) | 4.55 ± 0.075 | 4.21 ± 0.071 |

Mean opinion scores (1-5) from [unknown number of] ratings on 100 test sentences.

Oord, A. et al. (2016). WaveNet: A Generative Model for Raw Audio, 1–15. http://arxiv.org/abs/1609.03499

# APPENDICES

## (EVERYTHING THAT FOLLOWS IS AN *ASIDE*. NOT ON THE EXAM.)

# APPENDIX: OTHER APPROACHES TO WAVEFORM GENERATION

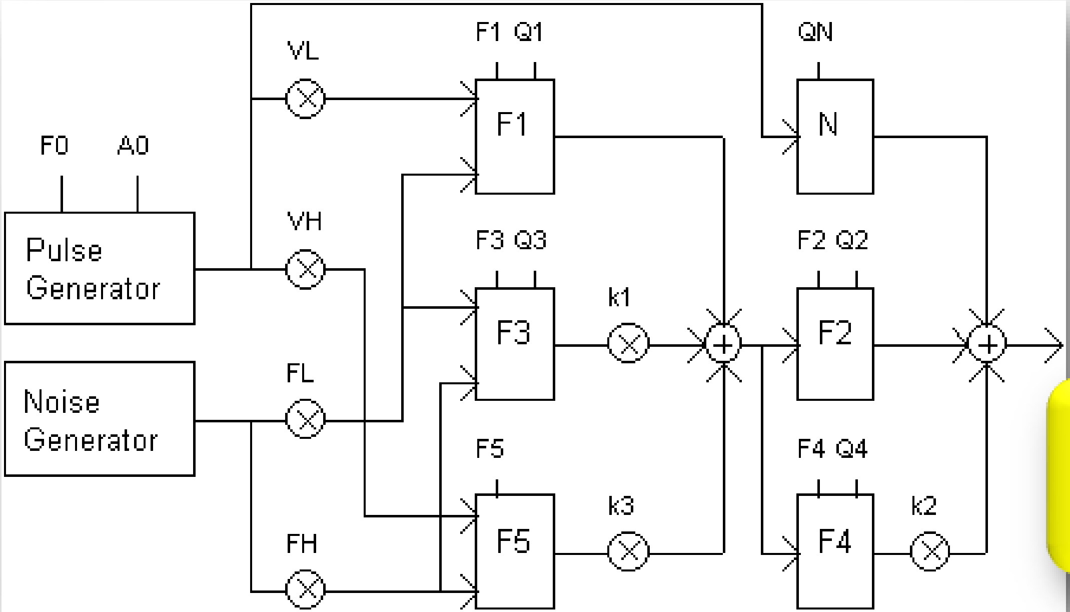
# 1. Formant synthesis

- Historically popular (MITalk in 1979, DECtalk in 1983).
- Stores a small number of parameters such as
  - **Formant frequencies** and **bandwidths** for vowels,
  - **Lengths** of sonorants in time,
  - Periodicity of the **fundamental frequency**.

- **Advantages**: This method can be very *intelligible*, avoids clipping artefacts between phonemes of other methods, and is *computationally inexpensive*.
- **Disadvantages**: This method tends to produce *unnatural* robotic-sounding speech.

# 1. Waveforms from formant synthesis

- The Klatt synthesizer produces either a **periodic pulse** (for sonorants like vowels) or **noise** (for fricatives) and passes these signals through **filters** – one for each formant.
  - These filters were parameterized by desired frequencies and bandwidths.



Don’t worry about the details

# Aside – linear predictive coding

- Formant synthesis is often performed by **linear predictive coding** (LPC), which is sometimes an alternative to MFCCs.
  - LPC is a very simple linear function which acts like a **moving average filter** over a signal $x$, e.g.,

$$y[n] = \sum_{i=-2}^{2} a_{n+i} x[n+i]$$

  - LPC results in very **smooth spectra**, which can result in **high intelligibility**, but **low naturalness** (real human spectra tend to be less smooth).

# 2. Concatenative synthesis

- Involves selecting short sections of **recorded** human speech and **concatenating** them together in time.

- **Advantages**: This method produces very human-like, *natural-sounding speech*. It is used in almost all modern commercial systems.
- **Disadvantages**: To be robust, this method requires a large (*computationally expensive*) database. Concatenating phones without appropriate blending can result in abrupt changes (*clipping glitches*).

# 2. Waveforms from concatenation

- **Diphone**: *n.* Middle of one phoneme to the middle of the next.

- Diphones are useful units because the middle of a phoneme is often in a **steady state** and recording diphones allows us to capture relevant **acoustic transitions** between phonemes.

- One speaker will record at least one version of each diphone, and in some cases whole (popular) words.

# 2. Waveforms from concatenation

- Given a **phoneme dictionary** of 50 phonemes, we might expect a (reduced) **diphone dictionary** of 1000 to 2000 diphones (multiplicatively more if we need to record diphones with/without stress, etc.)

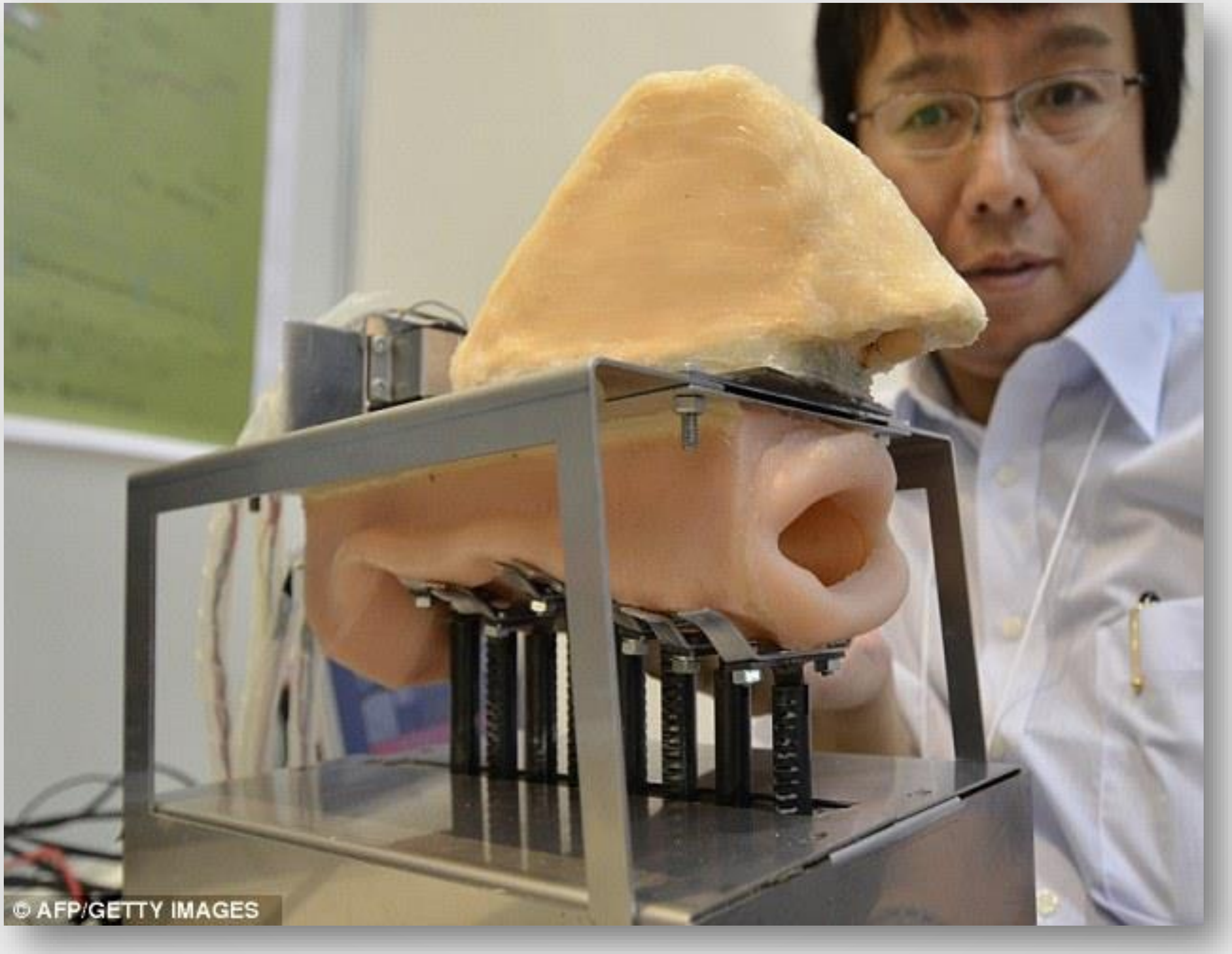
- When **synthesizing** an utterance, we extract relevant sequences of diphones, concatenate them together, and often perform some **acoustic post-processing** on the boundaries, or on the overall prosody of the utterance.

# 3. Articulatory synthesis

- Often involves the **uniform tube model** or some other biologically-inspired model of air propagation through the vocal tract.

- **Advantages**: This method is *computationally inexpensive* and allows us to *study speech production* scientifically, and to account for particular articulatory constraints.
- **Disadvantages**: The resulting speech is *not entirely natural*, and it can be *difficult to modify these systems* to imitate new synthetic speakers, or even complex articulations.

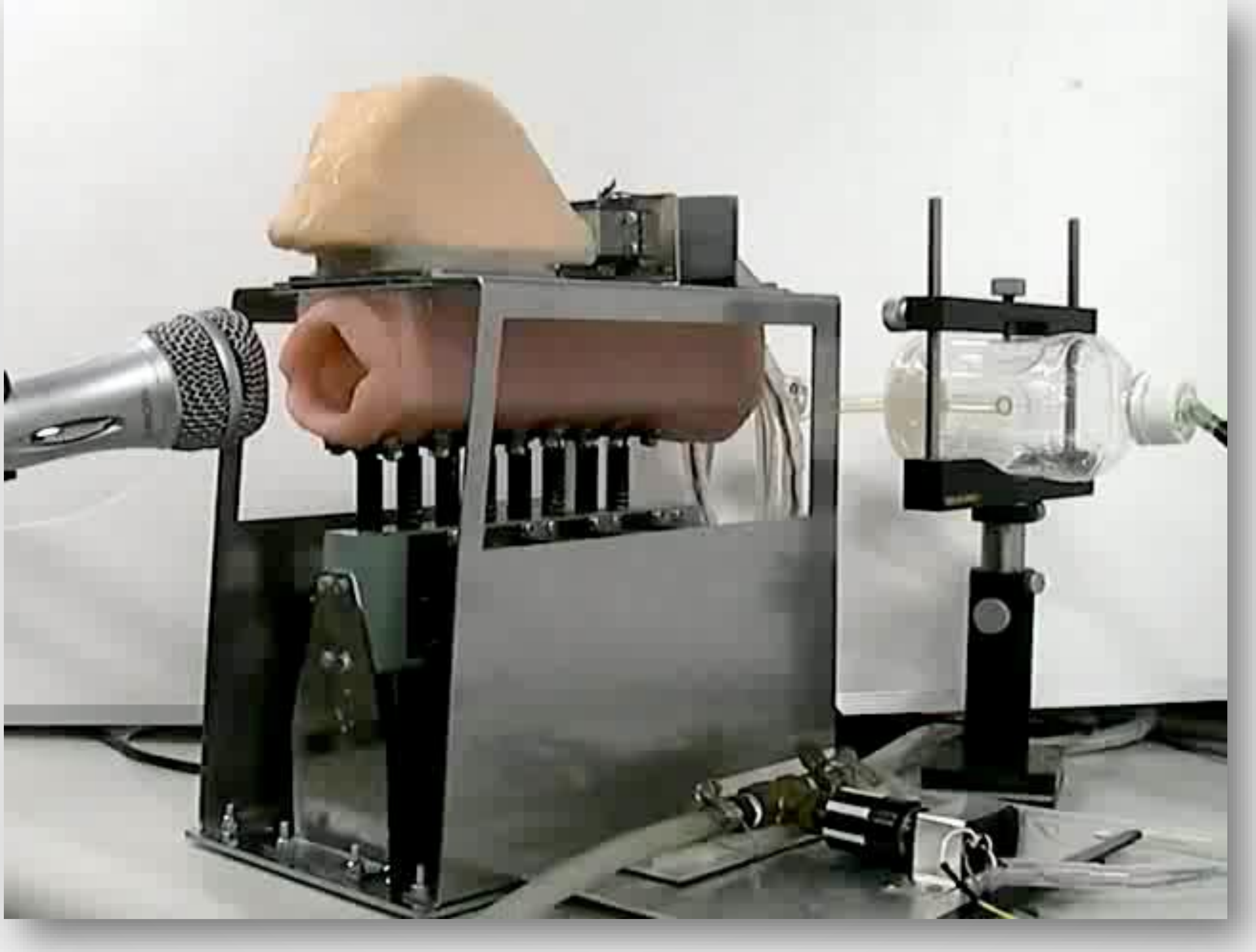
# 3. Articulatory synthesis



© AFP/GETTY IMAGES

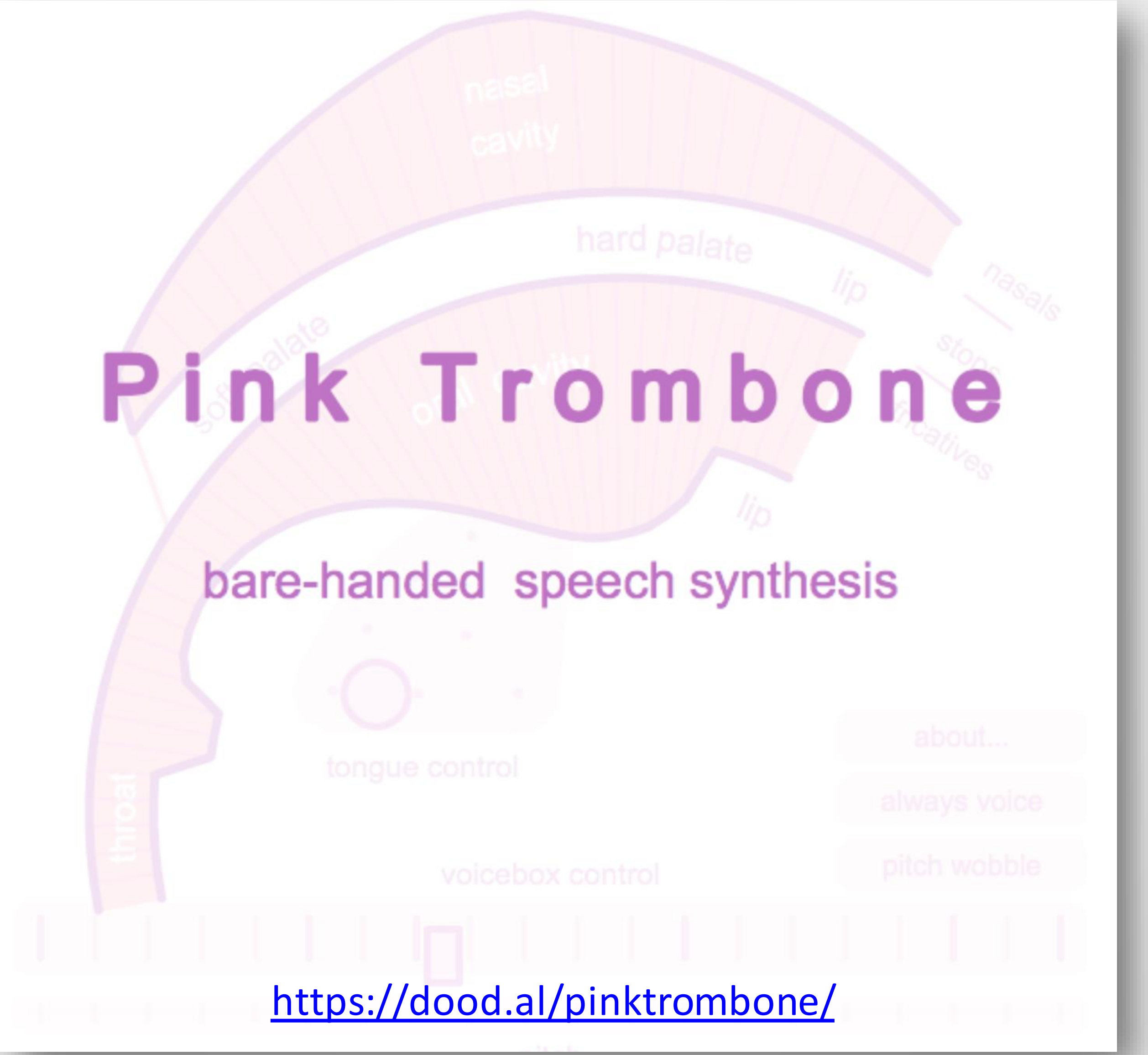
http://www.youtube.com/watch?v=Bht96voReEo

# 3. Articulatory synthesis



Note: this is singing, not speech (in case it's not obvious)

# 3. Articulatory synthesis



Pink Trombone

bare-handed speech synthesis

https://dood.al/pinktrombone/