Text-to-speech synthesis

Reminder

Our academic program coordinator/assistant program officer Nishi will be sending around a course evaluation. Please fill this out! We read every evaluation, and we do everything we can to be responsive to concerns raised therein.

Terminological note

This task is traditionally called *text-to-speech synthesis* (TTS) or simply *speech synthesis*. Much like *speech-to-text*, the term *text-to-speech* by itself is commonly used among non-specialists but is a shibboleth.

Downstream applications

- Voice user interfaces (virtual assistants, voice dialing, voice search, etc.)
- Audiobooks for the visually-impaired
- Public service announcement systems

General issues

- Intelligibility vs. naturalness
- High-quality vs. low-latency
- Domain-specific vs. general-purpose

Outline

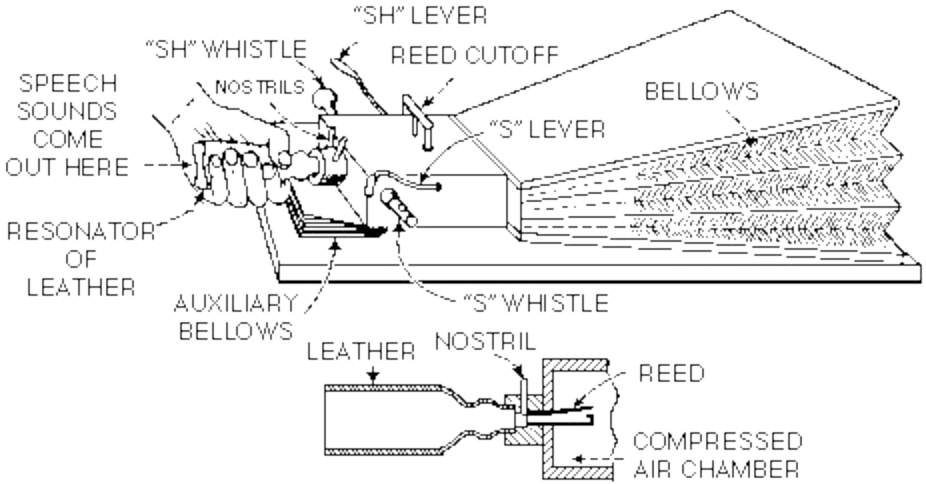
- A brief history
- Evaluation
- The frontend/backend dichotomy
- Parametric approaches
- Concatenative approaches
- Neural network-based approaches
- Software

A brief history of TTS

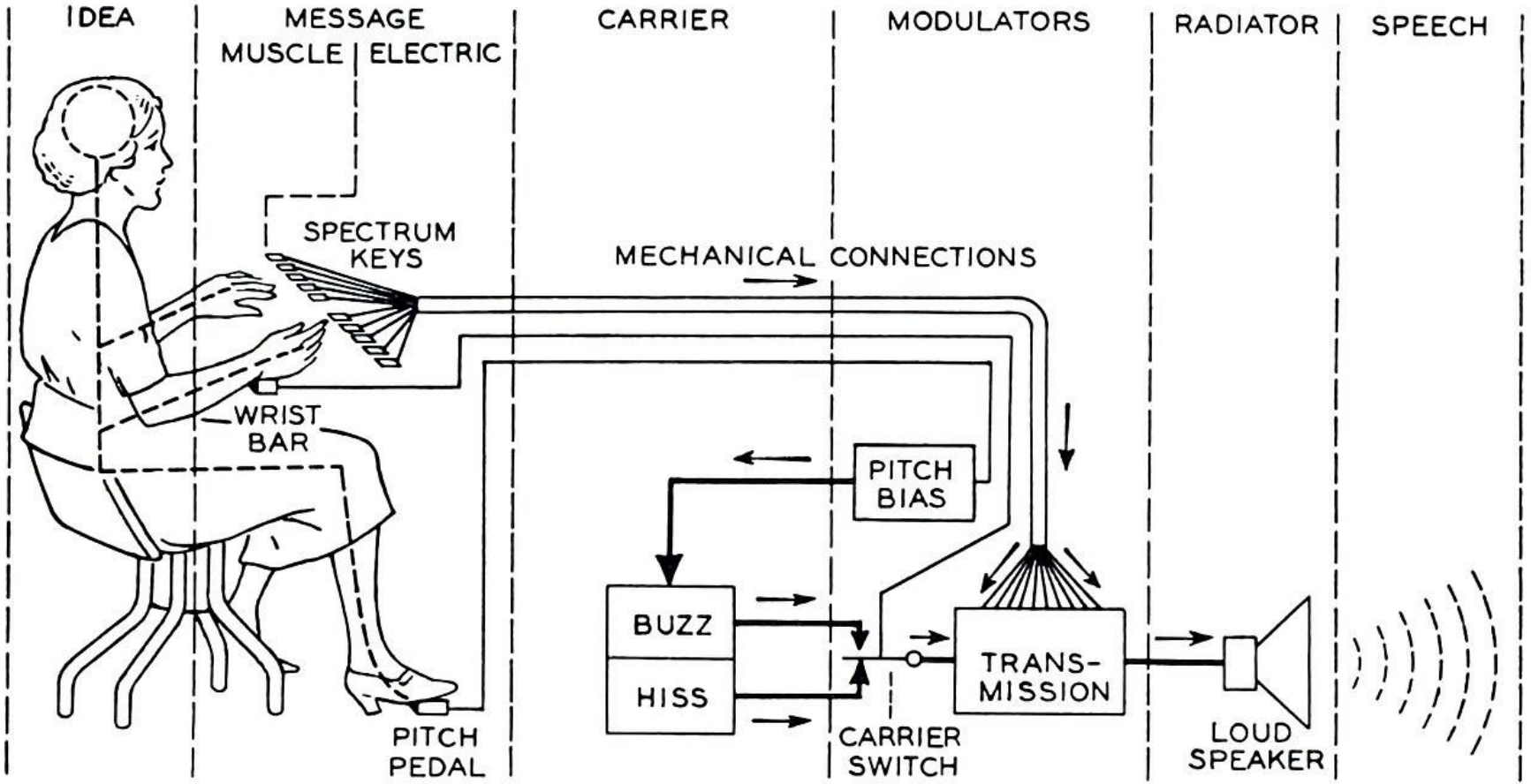
History of TTS

1791: Wolfgang von Kempelen's *speaking machine*.
1937: Homer Dudley's *vocoder* and *voder*.
1960s: Linear predictive coding.
1961: First computer "singing": "Daisy Bell (Bicycle Built For Two)", featured in *2001: A Space Odyssey* (1968).
1970s: Work begins on MITalk, later commercialized as DECtalk (1980s).
1978: Texas Instruments Speak & Spell, featured in Kraftwerk's "Nummern" (1981).
1980s: Parametric synthesis.
1996: Unit selection concatenative synthesis.
2000s: Hybrid concatenative/parametric synthesis.
2016: WaveNet.

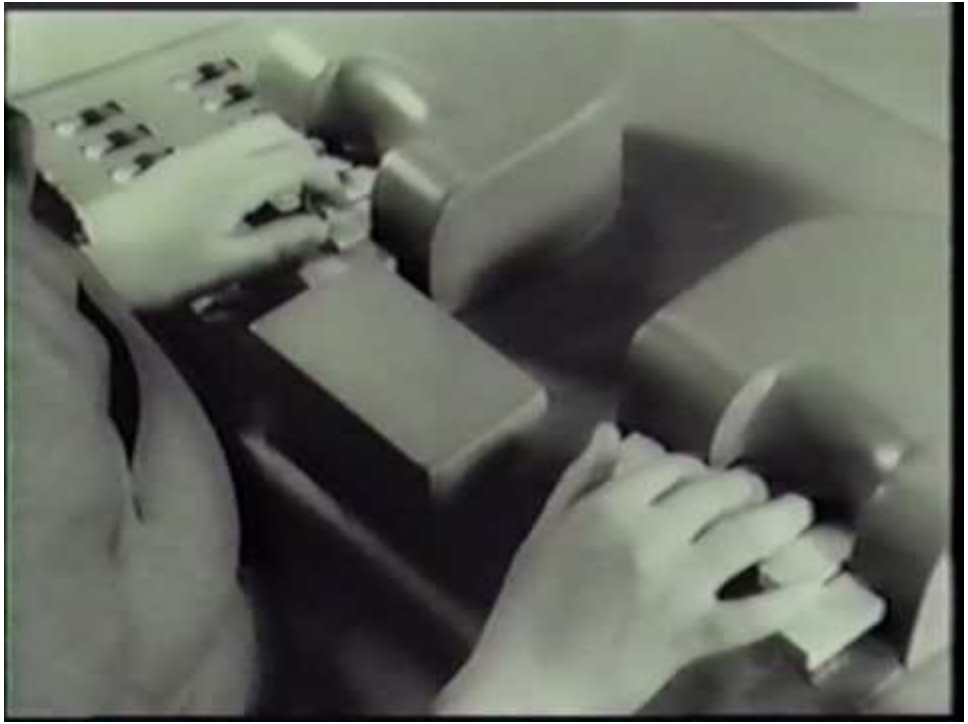
(Image credit: https://en.wikipedia.org/wiki/Wolfgang_von_Kempelen%27s_speaking_machine)



(Image credit: http://www.haskins.yale.edu/featured/heads/SIMULACRA/kempelen.html)



(Image credit: https://en.wikipedia.org/wiki/Voder)




Give me
your answ

(Image credit: https://en.wikipedia.org/wiki/Speak_%26_Spell_(toy))

Speak & Spell (US, 1979 Version): Texas Instruments






(Audio credits: https://deepmind.com/blog/article/wavenet-generative-model-raw-audio)

Evaluation

Intelligibility vs. naturalness

Intelligibility refers to the ease with which synthesized speech is understood by a speaker.

Naturalness refers to the degree to which synthesized speech resembles that of human speech.

These two features trade off because naturalness involves modeling of coarticulation and reduction, which arguably reduce intelligibility.

Evaluation

Mean opinion score (MOS): 5-point Likert scale measures of intelligibility and naturalness, averaged across subjects.

The Blizzard Challenge shared tasks run annual human quality evaluations including experts, paid undergrads, and volunteers.

Coverage of non-standard words, homographs, pronunciations, etc. are objective measurements of the frontend quality.

The frontend and backend

The backend

Narrowly, we can define speech synthesis as a problem of mapping from phone or phoneme sequences (with or without intonation) to waveforms. We refer to this system as the *backend*.

The backend also has to handle coarticulation, segment timing/duration, and pitch/intonation.

The frontend

The frontend is responsible for mapping from human-readable text to phone sequences (with or without intonation). This may include:

- Sentence boundary detection
- Text normalization, including:
  - Abbreviation expansion
  - Letter sequence prediction
- Homograph disambiguation
- Grapheme-to-phoneme conversion

Natural language systems used to generate sentences from templates (etc.) are usually not counted as part of the frontend.

Text normalization

- Currency expressions: $4.20 → *four dollars and twenty cents*
- Date expressions: 11/2 → *November second*
- Letter sequences: WinNT → *win n_letter t_letter*
- Numbers: 69 → *sixty nine*
- Measure expressions: 12kg → *twelve kilograms*

Following Taylor (2009), we refer to these categories as *semiotic classes*, and their conversion as *text normalization*.

Text normalization systems require some degree of linguistic sophistication.

A taxonomy of semiotic classes (Sproat et al. 2001)

	EXPN	abbreviation	*adv, N.Y, mph, gov't*
alpha	LSEQ	letter sequence	*CIA, D.C, CDs*
	ASWD	read as word	*CAT*, proper names
	MSPL	misspelling	*geogaphy*
	NUM	number (cardinal)	*12, 45, 1/2, 0·6*
	NORD	number (ordinal)	*May 7, 3rd, Bill Gates III*
	NTEL	telephone (or part of)	*212 555-4523*
	NDIG	number as digits	*Room 101*
N	NIDE	identifier	*747, 386, I5, pc110, 3A*
U	NADDR	number as street address	*5000 Pennsylvania, 4523 Forbes*
M	NZIP	zip code or PO Box	*91020*
B	NTIME	a (compound) time	*3·20, 11:45*
E	NDATE	a (compound) date	*2/2/99, 14/03/87* (or US) *03/14/87*
R	NYER	year(s)	*1998, 80s, 1900s, 2003*
S	MONEY	money (US or other)	*$3·45, HK$300, Y20,000, $200K*
	BMONEY	money tr/m/billions	*$3·45 billion*
	PRCT	percentage	*75%, 3·4%*
	SPLT	mixed or "split"	*WS99, x220, 2-car* (see also SLNT and PUNC examples)
	SLNT	not spoken,	word boundary or emphasis character:
M		word boundary	*M.bath, KENT*RLTY, _really_*
I	PUNC	not spoken,	non-standard punctuation: "***" in
S		phrase boundary	*$99,9K***Whites*, "…" in *DECIDE…Year*
C	FNSP	funny spelling	*sllooooooww, sh*t*
	URL	url, pathname or email	*http://apj.co.uk, /usr/local, phj@tpt.com*
	NONE	should be ignored	ascii art, formatting junk

Taxonomy of semiotic classes (Ebden & Sproat 2015)

- Cardinal: 69 → *sixty nine*
- Date: 11/2/1985 → *November second nineteen eighty five*
- Decimal: 23.3 → *twenty three point three*
- Electronic: kgorman@gc.cuny.edu → *k gorman at gc dot cuny dot edu*
- Fraction: 2/5 → *two fifths*
- Measure: 12kg → *twelve kilograms*
- Money: $5.96 → *five dollars and ninety six cents*
- Ordinal: 69th → *sixty ninth*
- Roman numeral: LIV → *fifty four*
- Telephone: 215-566-6123 → *two one five, five six six, six one two three*
- Time: 11:58 → *eleven fifty eight*

Wikipedia (“written” domain)

The giraffe has an extremely elongated neck, which can be up to 2 m (6 ft 7 in) in length, accounting for much of the animal's vertical height. Each cervical vertebra is over 28 cm (11 in) long. They comprise 52–54 percent of the length of the giraffe's vertebral column, compared with the 27–33 percent typical of similar large ungulates, including the giraffe’s closest living relative, the okapi.

Wikipedia (“spoken” domain)

The giraffe has an extremely elongated neck, which can be up to two meters (six feet seven inches) in length, accounting for much of the animal's vertical height. Each cervical vertebra is over twenty eight centimeters (eleven inches) long. They comprise fifty two to fifty four percent of the length of the giraffe's vertebral column, compared with the twenty seven to thirty three percent typical of similar large ungulates, including the giraffe’s closest living relative, the okapi.

"End-to-end" synthesis

Virtually all the neural network work on speech synthesis focuses on the backend (though see, e.g., Zhang et al. 2019 on text normalization).

Some newer neural network systems describe themselves as "end-to-end" because they eliminate certain modules from the frontend (e.g., Tacotron takes characters, not phoneme sequences, as inputs). However, these usually depend on other elements of the frontend (sentence boundary detection, tokenization, text normalization, homograph disambiguation, etc.) so the "end-to-end" term is somewhat undeserved.

Parametric synthesis

Articulatory synthesis

The earliest digital synthesizers tried to construct digital models of human speech articulation, performing synthesis by modifying the shape of the simulated "tongue", "lips", etc.

Parametric synthesis

In contrast, *parametric synthesis* mimics the physics of the vocal tract without directly modeling the articulators that control it.

Formant-based synthesis

Dennis Klatt's work on MITalk introduced a simple set of parameters based on excitations filtered by formants. Parameters include:

- Glottal noise source (impulses or white noise)
- Formant frequencies (and for nasals, nasal anti-frequencies)
- F0 and voicing amplitude

Limitations include:

- Need to estimate or manually tune parameters
- Poor handling of coarticulation

HMM-based synthesis

HMM-based synthesizers learn parameters of the model from data using the expectation maximization algorithm. Because of this, one can move from low dimensionality, human-interpretable parameterizations (like formants and excitations) to higher-dimensionality ones like MFCCs.

Concatenative synthesis

Concatenative synthesis

Concatenative synthesis systems concatenate pre-recorded phone-like units together, doing some simple signal processing at the "joins" (boundaries between units) and to create an intonational curve.

Traditionally, a database of $O(n^2)$ "diphones" are used. Thus *cat* in isolation might be:

```
/pau_k k_ae ae_t t_pau/
```

Unit-selection synthesis

Unit selection is a form of concatenative synthesis in which units larger than (di)phones may also be stored. The model that selects units to concatenate:

- prefers to use the longest possible stored units
- prefers to use units that minimize the local signal processing (the "join cost")

Since such systems prefer longer units, this suggests an easy way to improve quality: one has the voice talent record sentences that

- are frequently synthesized,
- contain frequent n-grams, or
- currently have poor MOS scores.

Parametric-concatenative hybridization

In a unit selection system, there may be some context where no stored unit can provide a "good join" with the adjacent units. Rather than using a unit that joins poorly, one can fall back on a parametric system to generate that unit.

A hybrid system of this form can be created using machine learning:

- a regression model estimates the cost of each join, and
- a classifier decides whether to use the lowest-cost stored unit or a parametric unit.

NN-based speech synthesis

WaveNet

WaveNet (van den Oord 2016) is a neural network-based synthesis model based on the PixelCNN architecture. It generates audio streams from phoneme sequences.

Two key ideas underlying WaveNet are *dilated convolution* and the *companding transformation*.

Dilated convolution

To work at the sample level one needs to be able to look quite far back in the signal to predict the next sample, accomplished using *causal dilated convolutional* layers.

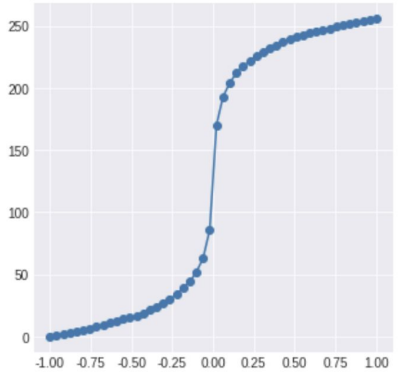
(Image credit: https://www.kdnuggets.com/2019/09/2019-guide-speech-synthesis-deep-learning.html)

Sample prediction and companding

Recall that "CD-quality" audio uses 44.01kHz 16-bit samples, so sample prediction is a massive 2^{16}-way (= 65,536) multinomial classification problem.

Instead, WaveNet does 256-way multinomial classification and uses a non-linear, sigmoid mapping back onto the full 16-bit sample-space.

$$f(x_t) = sign(x_t)\frac{\ln(1+\mu|X_t|)}{\ln(1+\mu)}, -1 < x_t < 1, \mu = 256$$



(Image credit: https://medium.com/@kion.kim/wavenet-a-network-good-to-know-7caaae735435)

Inputs to a WaveNet synthesizer

- Phonemic transcription
- F0 curve, predicted by an external model
- (For multispeaker tasks) a one-hot-encoding of the speaker ID

Software & Data

Free (*gratis*, at least) software

- Festival is a venerable open-source speech synthesis library featuring a number of pre-neural networks models.
- Mozilla Voice TTS is an open-source NN-based TTS engine implemented in TensorFlow, based on Google's Tacotron.
- Google Cloud Text-To-Speech provides a free tier with high-quality recognizers for over about 40 languages: one simply crafts a JSON query with the text and it replies with the audio.

Data

- The Blizzard Challenge datasets are commonly used by researchers.
- The Mozilla Common Voice project is crowdsourcing free multilingual data.

Unlike ASR, where systems are often built with hundreds of hours of speech, decent quality can be achieved in modern TTS systems with as few as a few hours. Commercial system developers spend a great deal of time recruiting, training, and pampering voice talent, like Susan Bennett, the voice behind Siri.

Outstanding problems

- High-quality understudies for rapid development
- Linguistically-informed models of intonation
- Reducing frontend errors
- Latency/quality tradeoffs
- Internationalization

Project ideas

- Implement a fragment of a TTS frontend.
- Implement a WaveNet synthesizer using your own voice.