speech synthesis
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?

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

Frank Zappa

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.
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 use ID3 or C4.5 for inducing 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.”
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/).

<table>
<thead>
<tr>
<th>Verb</th>
<th>Noun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use /y uw z/</td>
<td>Use /y uw s/</td>
</tr>
<tr>
<td>House /h aw z/</td>
<td>House /h aw s/</td>
</tr>
<tr>
<td>reCORD</td>
<td>REcord</td>
</tr>
<tr>
<td>disCOUNT</td>
<td>DIScount</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
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.

```plaintext
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.,

\[
P(\text{ph}|\text{le}) = \frac{\text{Count(\text{ph} \land \text{le})}}{\text{Count(\text{le})}}
\]

- Letter, \textit{le}: c, h, e, c, k, e, d
- Phoneme, \textit{ph}: ch, eh, k, t

• If you don’t have these alignments, they can be learned using \textit{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/
Standard TTS approaches

How do we actually produce the sounds, given the phonemes?
Standard TTS architectures

• Formant synthesis
  • Synthesizes acoustics based on rules and filters.

• Concatenative synthesis
  • Uses databases of stored speech to assemble audio.

• Articulatory synthesis
  • Models the movements and acoustics of the vocal tract.

• Statistical model synthesis
  • Samples from some stochastic model
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 here
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.
Aside – TD-PSOLA

• Time-domain pitch synchronous overlap and add (TD-PSOLA) is a very efficient method for combining waveforms while preserving pitch.
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

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

https://dood.al/pinktrombone/
4. Synthesis from HMMs

• Use a trained HMM and sample from it.

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

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.


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.
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.
Waveform modification

How do we adjust the final audio?
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
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., “dense” vs. “tense”, “maze” vs. “mace”

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