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.
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
Insight?

She had a snake for a pet and an amulet and she was breeding a dwarf but she wasn’t done yet. 🎼

Frank Zappa

Kismet
Some history

• In 1791 Wolfgang von Kempelen produced an “acoustic-mechanical speech machine”.

• This machine used bellows and models of the tongue and lips, enabling to produce rudimentary vowels and some consonants.

• In the 1930s, the first electronic speech synthesizer, VOCODER, was produced by Bell Labs.
Modern TTS architectures

• **Formant synthesis**
  • An approach that synthesizes acoustics and formants based on rules and filters.

• **Concatenative synthesis**
  • The use of databases of stored speech to assemble new utterances.

• **Articulatory synthesis**
  • The modelling of the movements of the articulators and the acoustics of the vocal tract.
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.
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).
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)
Components of TTS systems

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

And now we define these terms...
Text analysis

How do we analyze the text the system is given to read?
Text analysis

• First we need to normalize the text. This involves splitting the text into sentences and word tokens and sometimes chunking tokens into reasonable sections.

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.”
Rules for sentence detection

- You’ve seen heuristics for this in assignment 1.
- You can also use ID3 or C4.5 for inducing decision trees automatically.
Identifying the types of tokens

• Pronunciation of a single word token can depend on its type or its 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>disCOUNTER</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 guess/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 as /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 }

• Other dictionaries, like CELEX, are sometimes used but are often too small, or too specific to one dialect.
Dictionaries are insufficient

- **Unknown words** (a.k.a. “out of vocabulary” (OOV)) increase with the square root of the number of words in a new, previously unseen text.
  - Of 39,923 tokens in a test of the Penn Treebank, 1775 tokens were OOV (4.4%, 943 unique types).
  - Of these, 1360 were names, and about 64 were typos.

- Commercial systems often use dictionaries, but back off to special name and acronym routines when necessary.
Names

• About 20% of tokens in a typical newswire are names.
  • Some are common and can be predicted (e.g., Obama, Putin). Others may become common only after a system is deployed.

• Given an unknown name, we can perform morphology according to prescribed rules (e.g., if you know ‘Walter’, you can infer ‘Walters’), or you can train statistical LTS systems on names.
Letter-to-sound rules

• Early algorithms used handwritten rules, e.g.,
  • ( <WORDSTART> [ch] <CONSONANT> ) = say /k/
    ( <WORDSTART> [ch] <VOWEL> ) = say /ch/
  • This correctly pronounces “Christmas” and “Choice”, but mispronounces “Chord”.
  • English is notoriously full of exceptions, and these handwritten rules don’t generalize to other languages.

• A modern approach is to learn LTS rules by automatic induction.
Induction of 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.,

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

- If you don’t have these alignments, they can be learned using expectation-maximization as we saw with, e.g., statistical machine translation.
Induction of 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.
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 used...*
Deciding on word emphasis

• Word emphasis depends on context. The ‘new’ information in the answer to a question is often emphasized.

• Q1: What types of foods are a good source of vitamins?
  • A1: **LEGUMES** are a good source of vitamins.

• Q2: Are legumes a bad source of vitamins?
  • A2: Legumes are a **GOOD** source of vitamins.

• Q3: What sorts of things do legumes give you, healthwise?
  • A3: Legumes are a good source of **VITAMINS**.
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
Pitch prosody for questions

\[ F_1 (\text{Hz}) \]

\[ F_0 (\text{Hz}) \]

\begin{align*}
\text{I'm} & \quad \text{sorry} \\
\text{I'm} & \quad \text{sorry}
\end{align*}
Waveform generation

How do we transform the analyzed text into sound?
Waveform synthesis

- Given a string of **phonemes** and a desired **prosody**, we need to generate a **waveform**.
- The three architectures do this in unique ways.
  - **Formant synthesis** produces waveforms by synthesizing the desired spectrograms directly.
  - **Concatenative synthesis** combines pre-recorded samples of human speech.
  - **Articulatory synthesis** produces waveforms with biologically-inspired models of the vocal tract.
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 beyond the scope of this course.
  • 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).
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.
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.
Duration modification

- Duration modification can be as simple as duplication or removal of short-term periodic sequences.

Phase vocoding is better
Pitch modification

- Duration modification can be as simple as ‘squishing’ or ‘stretching’ signals using **decimation** or **interpolation**.
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).

Recent TTS

• In order to model more complexity in the time domain, many new TTS systems use HMMs to generate synthetic speech.
  • Listeners seem to find that this approach produces natural-sounding speech.
TTS demos

• Festival

• Cepstral
  • [http://www.cepstral.com/cgi-bin/demos/general](http://www.cepstral.com/cgi-bin/demos/general)

• IBM