Lecture 1: Introduction, ARPAbet, Articulatory Phonetics
April 3, Week 1

- Course introduction
- Course topics overview
  - Speech recognition
  - Dialog / conversational agents
  - Speech synthesis (Text to speech)
  - Affect extraction
- Very brief history
- Articulatory Phonetics
- Course Logistics
- ARPAbet transcription
An exciting time for spoken language processing

- Amazon Echo 2015
- Google Home 2016
- Facebook M 2015
- Apple Siri 2011
- Google Assistant 2016
- Microsoft Cortana 2014
- Anki Cozmo 2016
- Slack Bot API 2015
LVCSR

• Large Vocabulary Continuous Speech Recognition
  • ~64,000 words
  • Speaker independent (vs. speaker-dependent)
  • Continuous speech (vs isolated-word)
Current error rates

Ballpark numbers; exact numbers depend very much on the specific corpus

<table>
<thead>
<tr>
<th>Task</th>
<th>Vocabulary</th>
<th>Word Error Rate %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digits</td>
<td>11</td>
<td>0.5</td>
</tr>
<tr>
<td>WSJ read speech</td>
<td>5K</td>
<td>1</td>
</tr>
<tr>
<td>WSJ read speech</td>
<td>20K</td>
<td>1</td>
</tr>
<tr>
<td>Broadcast news</td>
<td>64,000+</td>
<td>4</td>
</tr>
<tr>
<td>Conversational Telephone</td>
<td>64,000+</td>
<td>6</td>
</tr>
</tbody>
</table>
Why is conversational speech harder?

• A piece of an utterance without context

• The same utterance with more context
## HSR versus ASR

<table>
<thead>
<tr>
<th>Task</th>
<th>Vocab</th>
<th>ASR</th>
<th>Hum SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous digits</td>
<td>11</td>
<td>.5</td>
<td>.009</td>
</tr>
<tr>
<td>WSJ 1995 clean</td>
<td>5K</td>
<td>3</td>
<td>0.9</td>
</tr>
<tr>
<td>WSJ 1995 w/noise</td>
<td>5K</td>
<td>9</td>
<td>1.1</td>
</tr>
<tr>
<td>SWBD 2004</td>
<td>65K</td>
<td>~6</td>
<td>3-4?</td>
</tr>
</tbody>
</table>

- **Conclusions:**
  - Gap increases with noisy speech
  - These numbers are rough, take with grain of salt
  - We are overfitting to the benchmark datasets
HSR versus ASR

<table>
<thead>
<tr>
<th>Deletions</th>
<th>SWB</th>
<th>Human</th>
<th>ASR</th>
<th>CH</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>30: it</td>
<td>19: i</td>
<td>46: i</td>
<td>20: i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20: i</td>
<td>17: it</td>
<td>46: it</td>
<td>18: and</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17: that</td>
<td>16: and</td>
<td>39: and</td>
<td>15: it</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16: a</td>
<td>14: that</td>
<td>32: is</td>
<td>15: the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14: and</td>
<td>14: you</td>
<td>26: oh</td>
<td>14: is</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14: oh</td>
<td>12: is</td>
<td>25: a</td>
<td>13: not</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14: you</td>
<td>12: the</td>
<td>20: to</td>
<td>10: a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12: %bcack</td>
<td>11: a</td>
<td>19: that</td>
<td>10: in</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12: the</td>
<td>10: of</td>
<td>19: the</td>
<td>10: that</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11: to</td>
<td>9: have</td>
<td>18: %bcack</td>
<td>10: to</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Insertions</th>
<th>SWB</th>
<th>Human</th>
<th>ASR</th>
<th>CH</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>13: i</td>
<td>16: is</td>
<td>23: a</td>
<td>17: is</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10: a</td>
<td>14: %hes</td>
<td>14: is</td>
<td>17: it</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7: and</td>
<td>12: i</td>
<td>11: i</td>
<td>16: and</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7: of</td>
<td>11: and</td>
<td>10: are</td>
<td>14: have</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6: you</td>
<td>9: it</td>
<td>10: you</td>
<td>13: a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5: do</td>
<td>6: do</td>
<td>9: the</td>
<td>13: that</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5: the</td>
<td>5: have</td>
<td>8: have</td>
<td>12: i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5: yeah</td>
<td>5: yeah</td>
<td>8: that</td>
<td>11: %hes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4: air</td>
<td>5: you</td>
<td>7: and</td>
<td>10: not</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4: in</td>
<td>4: are</td>
<td>7: it</td>
<td>9: oh</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Most frequent deletion and insertion errors for humans and ASR system on SWB and CH.

<table>
<thead>
<tr>
<th>SWB</th>
<th>Human</th>
<th>ASR</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>11: and / in</td>
<td>16: (%hes) / oh</td>
<td>21: was / is</td>
<td>28: (%hes) / oh</td>
</tr>
<tr>
<td>9: was / is</td>
<td>12: was / is</td>
<td>16: him / them</td>
<td>22: was / is</td>
</tr>
<tr>
<td>7: it / that</td>
<td>7: (i-) / %hes</td>
<td>15: in / and</td>
<td>11: (%hes) / %bcack</td>
</tr>
<tr>
<td>6: (%hes) / oh</td>
<td>5: (%hes) / a</td>
<td>8: a / the</td>
<td>10: bentsy / benji</td>
</tr>
<tr>
<td>6: him / them</td>
<td>5: (%hes) / hmm</td>
<td>8: and / in</td>
<td>10: yeah / yep</td>
</tr>
<tr>
<td>6: too / to</td>
<td>5: (a-) / %hes</td>
<td>8: is / was</td>
<td>9: a / the</td>
</tr>
<tr>
<td>5: (%hes) / i</td>
<td>5: could / can</td>
<td>8: two / to</td>
<td>8: is / was</td>
</tr>
<tr>
<td>5: then / and</td>
<td>5: that / it</td>
<td>7: the / a</td>
<td>7: (%hes) / a</td>
</tr>
<tr>
<td>4: (%hes) / %bcack</td>
<td>4: %bcack / oh</td>
<td>7: too / to</td>
<td>7: the / a</td>
</tr>
<tr>
<td>4: (%hes) / am</td>
<td>4: and / in</td>
<td>6: (%hes) / a</td>
<td>7: well / oh</td>
</tr>
</tbody>
</table>

Table 2: Most frequent substitution errors for humans and ASR system on SWB and CH.

(Saon et al, 2017)
Why accents are hard

• A word by itself

• The word in context
So is speech recognition solved? Why study it vs just use some API?

- In the last ~5 years
  - Dramatic reduction in LVCSR error rates (16% to 6%)
  - Human level LVCSR performance on Switchboard
  - New class of recognizers (end to end neural network)
- Understanding how ASR works enables better ASR-enabled systems
  - What types of errors are easy to correct?
  - How can a downstream system make use of uncertain outputs?
  - How much would building our own improve on an API?
- Next generation of ASR challenges as systems go live on phones and in homes
Speech Recognition Design

Intuition

- Build a statistical model of the speech-to-words process
- Collect lots and lots of speech, and transcribe all the words.
- Train the model on the labeled speech
- Paradigm: Supervised Machine Learning + Search
Dialogue (= Conversational Agents)

- Personal Assistants
  - Apple Siri
  - Microsoft Cortana
  - Google Assistant

- Design considerations
  - Synchronous or asynchronous tasks
  - Pure speech, pure text, UI hybrids
  - Functionality versus personality
Paradigms for Dialogue

- **POMDP**
  - Partially-Observed Markov Decision Processes
  - Reinforcement Learning to learn what action to take
  - Asking a question or answering one are just actions
    - “Speech acts”

- **Simple regular expressions and slot filling**
  - Pre-built frames
    - Calendar
      - Who
      - When
      - Where
  - Filled by hand-built rules
    - (“on (Mon|Tue|Wed... )”)
Paradigms for Dialogue

- **POMDP**
  - Exciting Research
  - Implemented in no commercial systems

- **Simple regular expressions and slot filling**
  - State of the art used most systems

- **Reusing new search engine technology**
  - Intent recognition / semantic parsing

- **Neural network chatbots**
  - Recent research, not really dialog yet
Extraction of Social Meaning from Speech

- Detection of student uncertainty in tutoring
  - Forbes-Riley et al. (2008)
- Emotion detection (annoyance)
  - Ang et al. (2002)
- Detection of deception
  - Newman et al. (2003)
- Detection of charisma
  - Rosenberg and Hirschberg (2005)
- Speaker stress, trauma
Conversational style

- Given speech and text from a conversation
- Can we tell if a speaker is
  - Awkward?
  - Flirtatious?
  - Friendly?
- Dataset:
  - 1000 4-minute “speed-dates”
  - Each subject rated their partner for these styles
  - The following segment has been lightly signal-processed:
Speaker Recognition tasks

- Speaker Recognition
  - Speaker Verification (Speaker Detection)
    - Is this speech sample from a particular speaker
      - Is that Jane?
  - Speaker Identification
    - Which of these speakers does this sample come from?
      - Who is that?
    - Related tasks: Gender ID, Language ID
      - Is this a woman or a man?
- Speaker Diarization
  - Segmenting a dialogue or multiparty conversation
    - Who spoke when?
Applications of Speaker Recognition

- Speaker Recognition:
  - Speaker verification (binary decision)
    - Voice password
    - Telephone assistant
  - Speaker identification (one of N)
    - Criminal investigation
- Diarization
  - Transcribing meetings
TTS (= Text-to-Speech) (= Speech Synthesis)

- Produce speech from a text input
- Applications:
  - Personal Assistants
    - Apple SIRI
    - Microsoft Cortana
    - Google Assistant
  - Games
  - Airport Announcements
TTS Overview

- Main Commercial Algorithm
  - Google TTS
- Collect lots of speech (5-50 hours) from one speaker, transcribe very carefully, all the syllables and phones and whatnot
- To synthesize a sentence, patch together syllables and phones from the training data.
- Parametric synthesis shows recent gains
- First end to end neural systems in 2016
History: foundational insights 1900s-1950s

- **Automaton:**
  - Markov 1911
  - Turing 1936
  - McCulloch-Pitts neuron (1943)
  - Shannon (1948) link between automata and Markov models

- **Human speech processing**
  - Fletcher at Bell Labs (1920’s)

- **Probabilistic/Information-theoretic models**
  - Shannon (1948)
Speech synthesis is old!

• Pictures and some text from Hartmut Traunmüller’s web site:
  • http://www.ling.su.se/staff/hartmut/kemplne.htm
• Von Kempeln 1780 b. Bratislava 1734 d. Vienna 1804
• Leather resonator manipulated by the operator to try and copy vocal tract configuration during sonorants (vowels, glides, nasals)
• Bellows provided air stream, counterweight provided inhalation
• Vibrating reed produced periodic pressure wave
Von Kempelen:

- Small whistles controlled consonants
- Rubber mouth and nose; nose had to be covered with two fingers for non-nasals
- Unvoiced sounds: mouth covered, auxiliary bellows driven by string provides puff of air

*From Traunmüller's web site*
History: Early Recognition

- 1920’s Radio Rex
  - Celluloid dog with iron base held within house by electromagnet against force of spring
  - Current to magnet flowed through bridge which was sensitive to energy at 500 Hz
  - 500 Hz energy caused bridge to vibrate, interrupting current, making dog spring forward
  - The sound “e” (ARPAbet [eh]) in Rex has 500 Hz component
History: early ASR systems

• 1950’s: Early Speech recognizers
  • 1952: Bell Labs single-speaker digit recognizer
    • Measured energy from two bands (formants)
    • Built with analog electrical components
    • 2% error rate for single speaker, isolated digits
  • 1958: Dudley built classifier that used continuous spectrum rather than just formants
  • 1959: Denes ASR combining grammar and acoustic probability
History: early ASR systems

- 1960’s
  - FFT - Fast Fourier transform (Cooley and Tukey 1965)
  - LPC - linear prediction (1968)
  - 1969 John Pierce letter “Whither Speech Recognition?”
    - Random tuning of parameters,
    - Lack of scientific rigor, no evaluation metrics
    - Need to rely on higher level knowledge
ASR: 1970’s and 1980’s

• Hidden Markov Model 1972
  • Independent application of Baker (CMU) and Jelinek/Bahl/Mercer lab (IBM) following work of Baum and colleagues at IDA

• ARPA project 1971-1976
  • 5-year speech understanding project: 1000 word vocab, continuous speech, multi-speaker
  • SDC, CMU, BBN
  • Only 1 CMU system achieved goal

• 1980’s+
  • Annual ARPA “Bakeoffs”
  • Large corpus collection
    • TIMIT
    • Resource Management
    • Wall Street Journal
Course Logistics
Course Logistics

- http://www.stanford.edu/class/cs224s

- Homeworks released and due on Wednesdays

- Gradescope for homework submission

- Piazza for questions. Email staff only for personal/confidential questions

- Project poster session tentatively June 7 (during class time)
Admin: Requirements and Grading

• Readings:
  • Selected chapters from
    • Jurafsky & Martin. Speech and Language Processing.
    • Will mix chapters from 2\textsuperscript{nd} and in progress 3\textsuperscript{rd} editions
    • A few conference and journal papers

• Grading
  • Homework: 40%
    • 4 assignments. Will use Python, Tensorflow, and command line tools
  • Course Project: 50%
    • Group projects of 3 people
  • Participation: 10%
Necessary Background

- Foundations of machine learning and natural language processing
  - CS 124, CS 224N, CS 229, or equivalent experience
- Mathematical foundations of neural networks
  - Understand forward and back propagation in terms of equations
- Proficiency in Python
  - Programming heavy homeworks will use Python and Tensorflow