Intelligibility Prediction for Pronunciation Remediation

These slides:
http://j.mp/irslides

February 2017
James Salsman

jim@talknicer.com

(27 slides)
Abstract

Speech recognition assesses English learners' pronunciation using authentic intelligibility to predict whether transcriptionists can type what they are supposed to say using Python's Scikit-learn support vector machine classifier (SVC.) Carnegie Mellon PocketSphinx is used in alignment mode with many recognition passes to find substitution and deletion of expected expected phonemes and insertion of unexpected phonemes along with differences of phonetic place, closure, roundedness, voicing, and the proportion of physiologically neighboring phonemes less likely in PocketSphinx's n-best results. SVC models achieve 82% agreement with the accuracy of crowdworkers' transcriptions, up from 75% reported by the inventor of the technique and Educational Testing Service. After asking learners to pronounce a word, if the SVC model suggests it was not pronounced intelligibly, we then ask them to repeat it after offering audio with the word pronounced correctly and their worst phoneme amplified and extended in duration.

For more information: https://arxiv.org/abs/1709.01713

Again, these slides are at: j.mp/irslides
Terms

Speech recognition

Pronunciation assessment

Pronunciation prediction (requires accent, dialect adaptation)

Intelligibility prediction (Nakagawa et al., 2011)

Remediation

  Feedback

  Interactions

  Sequencing
Motivation

Intelligibility assessment

- Word phonemes: W, IH, DH
- IPA: w, i, ã
- Audio utterance
- Acoustic score: -377, -155, -335
- Durations: 40 ms, 60 ms, 150 ms
- Recognized neighbor phonemes: y, eh, v

Text-independent scoring
- Learner recording exemplars
- Native speaker exemplars
- Authentic intelligibility remediation

(A) worst
(B) poor
(C) better
(D) best
Why accent adaptation?

14 accent loci in UK and Ireland,
3-4 in the USA and Canada,
Australia, South Africa, New Zealand, and
English as a Foreign Language everywhere
Motivation

English language instruction

Figure 1. Mobile Learning Value Added Services and Income per Person in the Developing World

Income per person (log scale; inflation- and purchasing power parity-adjusted) 1860-2011
Market
English language instruction
Intelligibility remediation

Goal

The goal is to ask people to try to pronounce words, and some day phrases, in a way that speech recognition features predict will be correctly transcribed by those who hear the audio utterance.

This technique can correctly adapt to spoken accents like vowel shifts, but not dialect.
Programs
Web server

```python
#!/usr/bin/env python2
#
# -*- coding: utf-8 -*-
#
Created on Thu Sep 14, 2017

@author: jsalsman
#

from __future__ import print_function

from os import chdir, system, mkdir, listdir
from shutil import move, rmtree
from sys import stderr, platform
from numpy import asarray
from flask import Flask, request, send_from_directory, jsonify, redirect
from textwrap import dedent
from urllib2 import urlopen
from datetime import datetime
from sklearn.svm import SVC
from random import choice

if platform.startswith("darwin"):
    chdir("/Users/james/Desktop/recdem")
else:
    chdir("/home/centos/htmlserve/")
```
If you have feature extraction sufficiently parsimonious with the dependent variable(s), you can use fast, small, and relatively simple support vector machine classifiers (SVC) instead of deep learning.
Intelligibility remediation
Data flow diagram

1. **Prompting phrase or word**

2. **Multiple audio recordings of learners' attempts at correct pronunciation**
   - Transcriptions from native and nonnative language transcriptionists
   - Amazon Mechanical Turk

3. **Booleans of each word's transcription accuracy (phrase's word-wise difference)**

4. **Six layer DNN: input features with softmax activation; 4 x 32; 2 category outputs**
   - The DNN maps each phoneme's duration, acoustic score from PocketSphinx alignment, substitution score, and insertion and deletion score into a probability that each word is intelligible.

5. **Learner's spoken attempt at correct pronunciation**

6. **Intelligibility assessment**

7. **Interactive remediation**

---

**SVM classifier training**

**SVM classifier prediction**
Intelligibility remediation
Scoring: 4 features/phoneme
Intelligibility remediation
Scoring: 9 features/phoneme

<table>
<thead>
<tr>
<th>Phoneme (CMUBET)</th>
<th>IPA (International Phonetic Alphabet)</th>
<th>Place: 1.0-5.0</th>
<th>Closedness: 1.0-7.0</th>
<th>Roundedness: no=1.0, yes=1.0</th>
<th>Voiced: unvoiced consonant=-1.0, voiced consonant=+0.5, vowel=+1.0</th>
<th>PNLL (Proportion Neighbors Less Likely)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>a</td>
<td>5</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>AE</td>
<td>æ</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>AH</td>
<td>ʌ</td>
<td>5</td>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>These</td>
</tr>
<tr>
<td>AO</td>
<td>ɔ</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>are</td>
</tr>
<tr>
<td>AW</td>
<td>ɑʊ</td>
<td>4.5</td>
<td>3.5</td>
<td>-0.5</td>
<td>1</td>
<td>are calculated</td>
</tr>
<tr>
<td>AY</td>
<td>ɑɹ</td>
<td>3.5</td>
<td>3.5</td>
<td>-1</td>
<td>1</td>
<td>from</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>0.5</td>
<td>the</td>
</tr>
<tr>
<td>CH</td>
<td>ʧ</td>
<td>3.5</td>
<td>4.0</td>
<td>-0.5</td>
<td>-1</td>
<td>calculated</td>
</tr>
<tr>
<td>D</td>
<td>d</td>
<td>3</td>
<td>4.0</td>
<td>-0.5</td>
<td>0.5</td>
<td>recognition</td>
</tr>
<tr>
<td>DH</td>
<td>ð</td>
<td>2.5</td>
<td>4.0</td>
<td>0</td>
<td>0.5</td>
<td>results.</td>
</tr>
<tr>
<td>EH</td>
<td>ɛ</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>ER</td>
<td>ɜɹ</td>
<td>3</td>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>
Intelligibility remediation

Feature extraction

featex-vta.c (feature extraction with vocal tract articulation)

```c
// keep track of relative tongue position.
// expected is (mdef->ciname[align[i].cipid])
// or numerically, align[i].cipid; observed is p
for (k=0; k<4; k++) {
    cips[k] += cipd[cip][k];
    cipm[k]++;

    // DEBUG:
    fprintf(stderr, "DEBUG: phoneme %s (%d) has %.2f in"
            " cipd[%d] against %.2f\n", s, cip,
            cipd[cip][k], k, cipd[align[i].cipid][k]);
}

// PNLL (Proportion Neighbors Less Likely)
for (k=0; k<6; k++) {
    if (cip == cin[k]) nh++;
}
```

For the final PNLL (proportion of neighbors less likely) feature, unlike the four articulation features, 1.0 instead of zero is best
Scoring
SVM classifier

Read data

Parse word descriptions

Fit model

Initialize model

Parse feature vectors
Intelligibility remediation
Demo: http://j.mp/intremed
Phoneme with most room for improvement ("worst" phoneme)

The additional four vocal tract articulation features are all set to zero, and the ninth, proportion of neighbors less likely, is set to 1.0 for this step.
Web server

Phoneme emphasis

```python
if pron != '':
    rawpron = list(pron)
    pron[phonpos] = '<b><em>' + pron[phonpos] + '</b></em>'
    retstr += ''.join(pron[1:]) + '</p>

if p_i < 0.5:
    # pick an exemplar file from those available
    examps = []
    for fn in listdir('exemplars/'):  
        if fn.startswith(word + '-'):  
            examps.append(fn)
    exemp = choice(examps)
    # get start and dur indexes for the phoneme
    with open('exemplars/' + exemp, 'r') as fi:
        for el in fi.readlines():
            tokens = el.strip().split()
            if tokens[0] == phonpos + 1: # 1-based but SIL padded
                start = float(tokens[2])
                dur = float(tokens[3])
                break
    bfn = int(exemp.replace(word + '-', '').replace('.txt', ''))
    bfl = listdir('exemplars/') + word
    bfl.sort()
    basefile = 'exemplars/' + word + '/' + bfl[bfn - 1] # 0-based
    outfile = 'uploads/' + nowstamp + '-example.wav'
    #
    # basefile: path, start: fract_seconds, dur: fract_secs, outfile: path
    system("PATH=/usr/local/bin:$PATH sox -m -t wav
    + "sox -V1 " + basefile + " -t wav - fade t 0"
    + str(start + 0.03) + " -t wav "sox -V1 " + basefile
    + " -t wav - trim " + str(start - 0.03) + " fade t 0.06"
    + str(dur + 0.06) + " 0.06 gain 5 pad"
    + str(start / 2.0 - 0.03) + " tempo 0.5"
    + " -t wav 'sox -V1 " + basefile + " -t wav - trim"
    + str(start + dur - 0.03) + " fade t 0.06 0 0 pad"
    + str(start + (dur * 2.0 - 0.03) + " " + outfile
    + " gain 3")"
    #
    # construct an audio URL for feedback if everything worked
    feedback = '/' + outfile
    retstr = dedent(''"
        <audio controls="")<source type="audio/wav" src="/uploads/' + nowstamp + '" + outfile + '.wav"></audio>"
    
```

Stored alignment data comes from featex-vta's stderr log

Single "worst" phoneme re-mixed with sox louder and longer but same pitch.
Data collection
Words, speech and transcripts

Prior to an accuracy assessment bug fix, the data collection parameters were:

- 700 words, one phrase (for comparison the Cambridge/EC English Profile has 6,500 words in levels A1-C2);
- About 30 recordings per word;
- About 4 transcripts per recording; and
- 4 numeric features per phoneme, upgraded to 9.

After the test-to-train data leak bug in Keras model.fit() was isolated, evidence suggests that large accuracy gains can be obtained both by collecting additional student attempted and exemplary recordings, and collecting additional transcripts of them.
Data collection
Sufficiency

How much is gained by increasing the number of recordings at each number of transcripts:

<table>
<thead>
<tr>
<th>records</th>
<th>transcripts</th>
<th>gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20</td>
<td>-1.34</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>-1.85</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>0.907</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>0.224</td>
</tr>
<tr>
<td>2</td>
<td>28</td>
<td>1.29</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>-1.52</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>0.249</td>
</tr>
<tr>
<td>3</td>
<td>24</td>
<td>-0.397</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>1.39</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>-0.775</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>-0.478</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>0.277</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>-0.278</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>1.08</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>5.52</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>0.458</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>0.289</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>2.35</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>1.07</td>
</tr>
<tr>
<td>2</td>
<td>28</td>
<td>2.22</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>-0.399</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>0.646</td>
</tr>
<tr>
<td>3</td>
<td>24</td>
<td>0.792</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>0.678</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>6.67</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>0.170</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>-0.399</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>0.646</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>0.792</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>0.468</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>6.67</td>
</tr>
</tbody>
</table>
# Data collection

## Balancing

<table>
<thead>
<tr>
<th></th>
<th>Current</th>
<th>Goal</th>
<th>Balancing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prompts (word or phrase)</strong></td>
<td>700</td>
<td>7000</td>
<td>Vocabulary grade level (eg. A1, A2, B...)</td>
</tr>
<tr>
<td><strong>Recordings</strong></td>
<td>30 per prompt</td>
<td>60 per prompt</td>
<td>Requires both good, completely wrong, and marginal</td>
</tr>
<tr>
<td><strong>Transcripts</strong></td>
<td>4 per recording</td>
<td>8 per recording</td>
<td>Beware of corruption from lazy and other defectors</td>
</tr>
<tr>
<td><strong>Exemplary recordings</strong></td>
<td>15 per prompt (40 words)</td>
<td>s*4 per prompt? (2 gender x 2 age)</td>
<td>s needs to be large enough for balancing recordings</td>
</tr>
</tbody>
</table>
Intelligibility remediation
Manifest and plans

Speech collection
Transcript collection
Transcript integration
Balancing
Sufficiency
Scoring
  Feature extraction
  SVM classifier
    Phoneme with-most-room-for-improvement isolation
ID (email) -- adaptivity and payment processing integration TBD
Exemplary flag for data collection
  -- collecting transcripts from learners TBD
Multiple choice support tool done but not used yet
Web server API
http://j.mp/intremed

Testing pronunciation intelligibility assessment

Please try the live microphone demo!

Audio utterance URL:

Format: ○ MP3, ○ WAV at 16,000 per second of 16 bit signed little endian monophonic samples, or ○ raw samples in that format, or ○ M4A.

Evaluate Note: audio will be stored.

Word: ○ airport, ○ argue, ○ ask, ○ baby, ○ banana, ○ bowl, ○ branch, ○ breakfast, ○ brother, ○ bus, ○ chair, ○ coach, ○ contest, ○ december, ○ eighth, ○ enjoy, ○ enough, ○ expensive, ○ family, ○ father, ○ feature, ○ flower, ○ forget, ○ fourth, ○ future, ○ grandma, ○ hall, ○ happy, ○ hill, ○ holiday, ○ homework, ○ hospital, ○ idea, ○ inside, ○ invent, ○ july, ○ lesson, ○ library, ○ litter, ○ live.

For JSON check here: □

<form>

聘用type= multipart/form-data
Web pages
API utilization in Javascript

```javascript
mp3blob = new Blob(buffer, {type: 'audio/mp3'});

(window.XMLHttpRequest) ? req = new XMLHttpRequest() :
  (window.ActiveXObject) ? req = new ActiveXObject("Microsoft.XMLHTTP") :
    req = false; // cross platform for IE7 or something

req.open("post", "/");
formdata = new FormData();
formdata.append('url', '');
formdata.append('filetype', 'mp3');
formdata.append('word', document.getElementById('word').innerHTML);
formdata.append('email', document.getElementById('email').value);
formdata.append('exemp', document.getElementById('exemp').checked);
formdata.append('audio', mp3blob);
formdata.append('json', 'checked')

req.addEventListener('load', function(event) {
  document.getElementById('upload').disabled = true;
  var resp = JSON.parse(req.responseText);
  var fUrl = '/rec/index.html?email='
    + escape(document.getElementById('email').value);
  if (resp.prob_good < 50 && resp.feedback != "") {
    fUrl += '&word=' + escape(word) + '&feedback=' + escape(resp.feedback);
  }
  alert('Data sent, parsed response:
        Visiting: ' + fUrl);
  document.location = fUrl;
});
req.addEventListener('error', function(event) {
  alert('Unable to upload.');
});

req.send(formdata);
```
Programs

Debugging

Python:

* web debugger: shows backtraces; optional PIN environment variable for examining variables, etc.:
  $ WERKZEUG_DEBUG_PIN=12345 nohup python valserve-40-test.py &
  http://werkzeug.pocoo.org/docs/0.14/debug/

* localhost stand-alone: Spyder and PyCharm both have good debuggers

C: gdb; PocketSphinx API documentation

HTML with getUserMedia/WebRTC: browsers
Transcript collection
Students instead of Turkers

Having students do the transcription task may produce data similar in quality to crowdworkers, at least in greater quantity, and at far less cost.

If you want to help with a Google Adwords campaign for a minimum viable freemium pronunciation tutor and data collection system, please get in touch.
Intelligibility remediation
Questions? Email:

jim@talknicer.com

These slides:
http://j.mp/irslides