Homework 1: Speech Systems and Phonetics
Due: 12th April 2017 at 11:59 PM PST

CS224S/LINGUIST285

Prof. Andrew Maas

Please read this entire homework handout before beginning. We advise you to *start early* and to make use of the TAs by coming to office hours and asking questions! For collaboration and the late day policy, please refer to the course website.

1 Turning in the assignment

Students will be turning in their assignment on Gradescope. You can use the code **9NDYG9** to sign up for the class using your Stanford id. Each student **must** submit their writeup individually.

2 Questions

1. For the first three problems, you are going to investigate the performance of and errors in speech-enabled personal assistant apps. You can use Siri, Google Now, or any kind of similar kind of speech-based personal assistant on a smartphone or other mobile device (or use multiple devices and compare!). **Please provide screenshot(s) that captures the interaction.**

   (a) First, write a couple of texts or emails using the microphone button on your mobile keyboard. What is the rough speech recognition word error rate (the number of incorrect words; more technically this would be the edit distance, the number of substitutions + deletions + insertions between the transcribed sentence and the correct word string)? Can you characterize what's going on with the errors? Try to "barge in" (i.e. talk while the system is talking to you). Does the system allow barge-in? Try to break the ASR with accent/far microphone and report your results.

   (b) Make and cancel some calendar appointments. Also try setting up alarms using just voice commands. Again, analyze any errors: did they fail because of speech recognition or the natural language or

dialog components? If the natural language or dialog, what went wrong?

(c) Try to find a business (a restaurant or etc.). Again, analyze any errors: did they fail because of speech recognition or NL/dialog? If NL/dialog, what went wrong?

(d) Go to Google webpage on your desktop/laptop and use the Google search microphone button to do a simple web search about Stanford Computer Science program. Does it work well? Is this an interface style that you would use daily? Why?

(e) Finally let's compare the results from part(a) and part(d). Pick a long search query (containing at least 15 words) of your choice and try both the inbuilt keyboard microphone button and the google search button to search for that query. Do they produce the same result? Also, do they produce similar pattern of errors?

2. Now check out some TTS systems. For example, on an Android, you can download the Google Text to Speech App. Or on an iOS device, go to the Settings page, select General → Accessibility, and turn on Speak Selection. Now when you highlight any text it will give you a Speak option that you can click. On a Mac, in the terminal window you can use the "say" command on the unix command line.

   Test out the TTS by choosing 4 different sentences. Try to be creative, including questions, exclamations, or whatever. Write down at least 5 errors that you hear; note whether these errors are due to wrong phones, due to incorrect stress, or due to a problem with the intonation/prosody.

   Finally compare the unit selection and HMM system here. How easy is it to understand the system? How human/natural does the system sound? How does it convey prosody (pitch that follows what a natural speaker would)

3. Find and correct the mistakes in the ARPAbet transcriptions of the following words:

   (a) three [dh r i]
   (b) sing [s ih n g]
   (c) eyes [ay s]
   (d) study [s t uh d i]
   (e) though [th ow]
   (f) planning [p pl aa n ih ng]
   (g) slight [s l iy t]

4. Transcribe the following words into the ARPAbet.

   (a) red

(b) blue

(c) green

(d) yellow

(e) black

(f) white

(g) orange

(h) purple

(i) dark

(j) suit

(k) greasy

(l) wash

(m) water

5. Transcribe the following two wavefiles at the word level (that is, write down the words that occur in the utterance). Make sure to listen to them carefully and more than one time. If you have trouble listening to them, let us know **immediately**. Try to transcribe any word fragments you hear. E.g. if the word should be "having" and the speaker drops the final syllable you might transcribe it as "hav–"

   (a) Utterance from Boston Radio News corpus

   (b) Utterance from Switchboard corpus

6. Now open both files in Praat, the speech analysis program. Transcribe both files into the ARPAbet, using Praat to help you play pieces of each wavfile, and to look at the wavefile and the spectrogram. (In fact, you can use Praat to play the files for the previous exercise as well).

   Turn in the ASCII ARPAbet sequences for the two files (just type it into your homework answers). For the Switchboard file, also label the start and end and identity of each phone using the "Annotate" -¿ "To Text Grid", with just one tier for "phone" (you don't need to use a word tier). Include a picture of this Praat labeled file using the "Draw" window (select the Sound and TextGrid, then click Draw, then save as EPS. Convert the file PDF before you attach it.) (If the file is too long to read the fonts clearly in the "draw" window, just break it into 2 or 3 parts and attach separate pictures). This is very hard, so I don't expect you to be perfect, I just want to you try to listen carefully for what's happening in each file.

7. Get the minimum and maximum pitch for the two files. Record the pitch range (range = max - min).

8. What are some differences between the Boston News file and the Switchboard file, in terms of transcription differences, pitch range, or other things

you noticed. Switchboard is human-human speech; Boston News is broadcast speech, which resembles human-machine speech. Could this play a causal role in the differences you found? How?

You may use an on-line ARPAbet dictionary to help you. Here is the CMU dictionary. But many or most words in the above sentences will not be the same as they are in the dictionary! So be careful not to just copy the pronunciation from the dictionary (CMU uses a slightly different version of the ARPAbet than the one in the slides from lecture 1, but you can use any ARPAbet version you want, including the version on Wikipedia).

Getting Praat: Praat itself is here, and is free and very simple to download, just grab the executable. It runs on most popular platforms.

A quick Praat intro, written by Edward Flemming, is here.

A longer and more comprehensive Praat tutorial is here.