Second Midterm Exam

601.467/667 Introduction to Human Language Technology

Fall 2022
Johns Hopkins University
Co-ordinator: Philipp Koehn

10 November 2022

Complete all questions.

Use additional paper if needed.

Time: 75 minutes.

Name of student: _____

Q1. Auditory System *15 points*

1. What is a critical band? (5pt)
   A critical band is a frequency region over which energy is integrated Idea is: If two frequencies are close enough to each (within a critical band), they activate same region in the ear; so the sound is not perceived as loud. If they are far apart (larger than a critical band), a new region in the ear is activated making the sound seem louder

2. Which cues does the auditory system extract from our two ears to allow us to localize sounds in space? Do these cues work effectively for any audio signal regardless of frequency? (5pt)
   The brain extracts Interaural timing differences (ITD) and interaural level -or intensity- differences (ILD) by comparing the timing of arrival and level of the signal arriving in both ears.

   - *ILD*
     - *Head-size dependent: larger heads create bigger ILDs for the same frequency*
     - *Very-frequency dependent – larger effect at higher frequencies*
   - *ITD*
     - *Head-size dependent: larger heads create bigger range of ITDs*
     - *Less-frequency dependent – works over large freq range, though more effective for lower frequencies*

3. MP3 coding uses a very basic concept of auditory masking to achieve great compression rates on audio signals. How does this masking work? (5pt)
   Simultaneous or frequency masking occurs when the presence of a strong audio signal makes weaker audio signals in the proximity imperceptible. It effectively raises the threshold of hearing (the level below which a sound is not heard) for a masked sounds hence allowing it to be ignored by the compression scheme since it won't be perceived.

Q2. Speech Production *15 points*

1. Describe the differences between articulation place and manner classes. Describe at least two examples of each (5pt)
   Manner classes are ways in which we articulate that produce consonant sounds. For instance, plosives are manner classes. In plosives we close completely the vocal tract and release the air stream to generate the sounds. Fricatives are also manner classes. In fricatives, we constrict the vocal tract partially, which creates turbulences that generate the fricative sounds. In contrast, articulation places are points or areas in the vocal tract where there is a constriction (with or without contact) which has the most relevance in the generated sound. Articulation points can be alveolar, glottal, labiodental, etc. In alveolar points, the tongue generates a constriction (total or partial) in the front of the hard palate. In bilabial articulation points, the lips generate the constriction.

2. Respond to the following questions (you might need to use Figure 1) (5pt):
   a) Which is the main articulator involved in changing the frequency formants that define the vowels?
   The tongue
   b) Are nasals sonorant consonants? In other words, do the vocal folds vibrate while pronouncing most nasal sounds?
   Yes, nasals are sonorant consonants
   c) Are plosives sonorant consonants? In other words, do the vocal folds vibrate while pronouncing most plosive sounds?
   No, most plosives are not sonorant as the vocal folds do not vibrate while producing them (touch your throat while pronouncing a plosive to check it).
   d) According to the IPA table, can fricatives be labiodental? If affirmative, please, provide two examples of labiodental fricative sounds.
   Yes, they can. Examples: f and v.
   e) According to the IPA table, can nasals be glottal? If affirmative, please, provide some examples of glottal nasal sounds.
   No, they cannot be glottal.

THE INTERNATIONAL PHONETIC ALPHABET (revised to 2015)

CONSONANTS (PULMONIC) © 2015 IPA

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.

Figure 1: IPA Table

3. What are the differences between phones and phonemes (2.5pt)
   A phoneme is a perceptually distinct speech element that could distinguish one word from another. A phone, is a segment that includes a distinct sound, that can correspond to a phoneme or not. A phone can be part of a phoneme or a phoneme.

4. Provide an example of a case in which changing a sound changes the meaning of a word, and a case in which it does not (2.5pt)
   For instance, the word "cake": if you change the first sound and turn it into an "l" it will be "lake". However, in the same wordk "cake" people from two different regions (USA and UK, for instance) will pronounce the middle "a" differently, but that does not change the meaning of the word. It's just a change of dialect.

Q3. Hidden Markov Models in ASR *20 points*

Several "objects" were introduced to illustrate various ideas, methods and algorithms used in automatic speech recognition, such as

(a) A word-list with phonemic pronunciation(s) for each word

(b) Speech waveforms

(c) A word-level transcript of the speech

(d) A transcript with (word- or phoneme-level) time-marks, a.k.a. time-alignments

(e) A text corpus in the language (possibly unrelated to the speech)

(f) A set of HMMs, one per phoneme (with unspecified parameters)

(g) A set of HMMs, one per phoneme, with known parameters

(h) A language model (with unspecified parameters)

(i) A language model with known parameters

Mathematical notation was introduced to discuss these concepts, such as

$$
\begin{aligned}
P(\mathbf{A}|\mathbf{W}) &: \text{The likelihood of a speech waveform } \mathbf{A} \text{ given a transcript } \mathbf{W} \\
P(\mathbf{W}) &: \text{The likelihood of a transcript } \mathbf{W} \\
P(\mathbf{S}|\mathbf{A},\mathbf{W}) &: \text{The likelihood of a time-alignment } \mathbf{S} \text{ of } \mathbf{A} \text{ with } \mathbf{W} \\
P_{\mathbf{\Theta}}(\mathbf{A},\mathbf{S}) &= \prod_t P_{\mathbf{\Theta}}(a_t|s_t) P_{\mathbf{\Theta}}(s_t|s_{t-1}) \quad : \quad \text{An HMM with parameters } \mathbf{\Theta}
\end{aligned}
$$

With these in mind, imagine that you now have to code various training and inference algorithms for HMM-based ASR, and answer the following questions.

1. What mathematical quantity does the Forward-Backward algorithm compute?
   It computes $P_{\mathbf{\Theta}}(\mathbf{A}|\mathbf{W})$ by summing over all possible alignments $\mathbf{S}$.

2. Which objects from the list (a)-(i) above, if any, are inputs to the Forward-Backward algorithm and which objects are outputs of the algorithm?
   Inputs: (a), (b), (c) and (g); Outputs: none.

3. What mathematical quantity does the Baum-Welch algorithm compute?
   It computes $\widehat{\mathbf{\Theta}}$, an estimate of the parameters $\mathbf{\Theta}$ of the HMM.

4. Which objects from the list (a)-(i) above, if any, are inputs to the Baum-Welch algorithm and which objects are outputs of the algorithm?
   Inputs: (a), (b), (c) and (f); Outputs: (g).

5. What mathematical quantity do unigram-, bigram- and trigram-models compute?
   They are differnt ways of computing $P(\mathbf{W})$.

6. Which objects from the list (a)-(i) above, if any, are inputs for training a unigram-, bigram- or trigram model and which objects are the outputs?
   Inputs: (c), (e) and (h); Outputs: (i).

7. What mathematical quantity does the Viterbi decoding algorithm compute?
   It computes $\widehat{\mathbf{W}} = \arg\max_{\mathbf{W}} \max_{\mathbf{S}(\mathbf{W})} P_{\boldsymbol{\Theta}}(\mathbf{S}(\mathbf{W})|\mathbf{A})$, *the most likely time-aligned word sequence* $\widehat{\mathbf{W}}$ *given the speech* $\mathbf{A}$.

8. Which objects from the list (a)-(i) above, if any, are inputs to the Viterbi decoding algorithm and which objects are outputs of the algorithm?
   Inputs: (a), (b), (g), (i); Outputs: (c). It also produces (d) as a by-product.

Q4. Speaker Recognition *16 points*

1. Name at least three applications of speaker recognition and name an example of each one of them. (5pt)
   (1.66pt/application)

   - *Law Enforcement and Forensics, e.g.:*
     - *Search for criminals in telephone conversations.*
     - *Detect known telephone fraudsters in black-lists.*
     - *Identify people making telephone threats.*
     - *Convict or discharge a defendant in court.*
   - *Identity authentication and access control, e.g.:*
     - *Access high security physical facilities.*
     - *Access computer networks and web services.*
     - *Telephonic banking.*
     - *Password reset.*
   - *Enriching meeting transcription, e.g.:*
     - *Knowing who spoke when, who said what.*
   - *Audio-Visual Media Indexing, e.g.:*
     - *Add metadata of who is speaking to Broadcast TV or youtube videos.*
     - *Improve indexing and searching documents.*
   - *Personalization, e.g.:*
     - *Voice assistants adapt to the user: play user's music, e-mails, parental control.*

2. Draw the block diagram of a speaker verification system, indicating the three different operation phases. Briefly explain what happens in each phase and the function of each block. (10pt)
   (2.5pt) (2.5pt) In the training phase, we take a large database with speech of lots of people, we extract acoustic features from the audio and we train some statistical models on those features.
   Once we have done that; we can put the speaker verification system into production. In production we have 2 phases, the enrollment phase and test phase.
   (2.5pt) In the enrollment phase, we register new speakers into our system. For example we take Alice's voice, we compute acoustic features from it, and then we compute a speaker embedding for Alice. The speaker embedding is just a vector that compresses the speaker identity information. This is typically calculated using a neural network. Finally, the speaker embedding is stored into the system's database.
   (2.5pt) Finally, in test phase, I get a recording from somebody that says to be Alice, then I

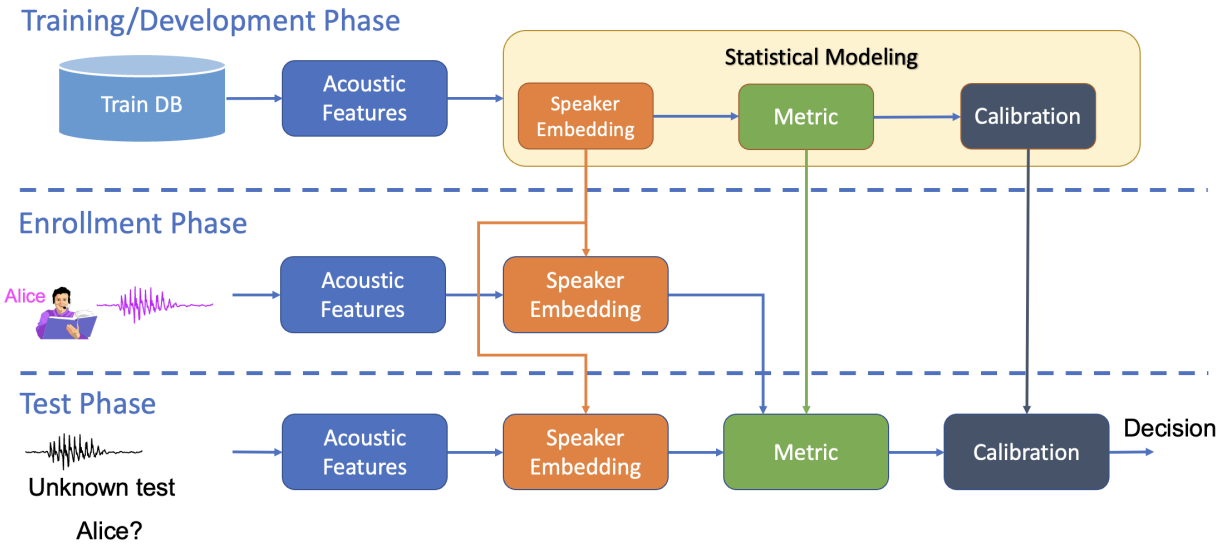

Figure 2: speaker verification pipeline

extract an embedding for that recording. Then, we put both Alice embedding and the test embedding into the metric block. This block, compares both embeddings and produce a score. if the score is high that means that the test speaker is Alice, if the score is low that means that it is not Alice., Sometimes, we have an extract block that calibrates the score into a well calibrated probability. And finally, we put a threshold on the score to take the decision of accept or reject.

3. The speaker verification system produces a score proportional to probability that the test audio belongs to the target speaker. We put a threshold to this score to decide if we accept or reject the trial. When doing so we can make errors. Name the types of errors that we can make, what are the error rates corresponding to them, how do you calculate the equal error rate metric? (5pt)
   (2.5pt) There are two types of errors:

   - *Misses or False rejection: when a target speaker is classified as an impostor, it is measured by the probability of miss.*
   - *False alarm: when an impostor is classified as the target speaker, it is measured by the probability of false alarm.*

   (2.5pt) Miss and false alarm probabilities depend on the decision threshold. EER obtained by finding the threshold at which the Miss rate is equal to the false alarm rate and them EER=$P_{miss} = P_{fa}$.

Q5. Neural Networks in ASR *20 points*

Consider a fully-neural ASR system of the kind discussed in class: either the early architecture with CNNs and BLSTMs used in Baidu's Deep Speech systems, or the encoder-decoder architecture with attention that was proposed later.

Such networks are presented with a speech segment of some length, say N, and produce a sequence of posterior probabilities of some (shorter) length, say K, on some output alphabet. E.g., in class, we imagined them being presented with MFCC vectors at a 10 ms frame-rate, and producing posterior probabilities on phonemes in the set $\{\text{ae}, \text{d}, \text{h}, \text{k}, \text{n}, \text{t}\}$ for the running example word sequence `cat and hat`.

Next, recall that under the CTC criterion, many K length character sequences are equivalent to a specific word sequence. E.g. The word sequence

`cat and hat`

may be realized using graphemic sub-word symbols as

ϕ c c ϕ ϕ a t ϕ a n ϕ d h ϕ ϕ ϕ a t ϕ.

This is just one possible realization among many.

For this problem, consider *graphemic* sub-word units, i.e. assume that at each of the K frames, the neural network outputs a posterior probability vector on the set of all characters, plus the "blank" symbol used in CTC and denoted by β or ϕ.

1. Draw the finite state transducer (FST) that accepts all the CTC sequences corresponding to the word `bee`.
   Recall that such an FST topology was created in the hands-on tutorial for `HELLO`.

   Alternatively:

   Draw a hidden Markov model corresponding to set of CTC sequences for the word `bee` using the one-state HMM topology for each letter and β, as discussed in class.
   Recall that you may use *null* arcs *where appropriate* to connect the letter HMMs.

   The figure below shows the HMM version of the CTC topology. Note that null arcs are used to connect the β models before the first `e` *and after the second* `e`*, but a* non-null *arc is used in the model for β between the repeated* `e`*'s.*
   A similar diagram for the FSA version should also get full credit.

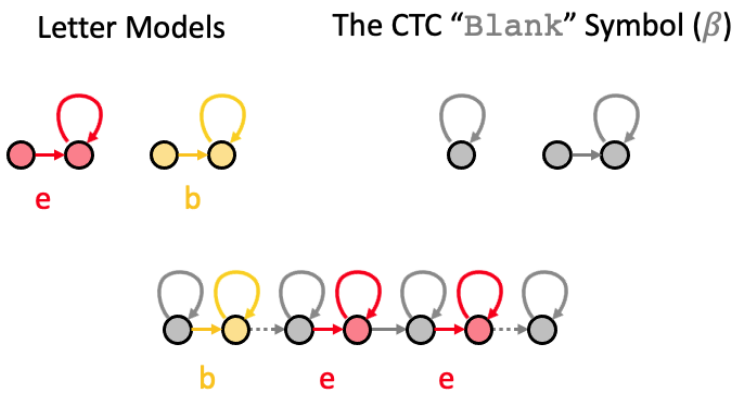

Figure 3: HMM representing all possible CTC letter sequences for the word bee.

2. Enumerate all letter sequences of length 5 that correspond to the word bee.

   Since 4 emitting arcs must be traversed to go from the leftmost to the rightmost state in the HMM drawn above, the only way one can get from the beginning to the end in 5 steps is to take exactly one self-loop, no more and no less. Since there are 7 self loops, there are 7 sequences of length 5 for bee*, one corresponding to each choice of the self-loop.*

   ϕ b e ϕ e
   b b e ϕ e
   b ϕ e ϕ e
   b e e ϕ e
   b e ϕ ϕ e
   b e ϕ e e
   b e ϕ e ϕ

3. Imagine a neural network that produces posterior probabilities on the set {b,e,ϕ}, and consider the case when its output matrix for a speech input of length 5 is

$$\begin{matrix} P(\mathtt{b}|\mathbf{A}) \\ P(\mathtt{e}|\mathbf{A}) \\ P(\phi|\mathbf{A}) \end{matrix} \equiv \begin{bmatrix} 0.3 & 0.6 & 0.2 & 0.0 & 0.0 \\ 0.1 & 0.2 & 0.5 & 0.6 & 0.7 \\ 0.6 & 0.2 & 0.3 & 0.4 & 0.3 \end{bmatrix}$$

   Compute the posterior probability of each letter sequence you listed above. Then compute the total (posterior) probability, i.e. the CTC objective function value, of the word bee.
   Answer will be provided later.

4. Note that the number of letter sequences grows exponentially with K, so calculating the posterior probability of each sequence is not tractable in general.

Which algorithm may be used to efficiently compute the total posterior probability, i.e. the CTC objective function?
The Forward-Backward algorithm for hidden Markov models is applicable "as is" for efficiently computing the CTC objective function.

5. What independence assumptions are made, if any, in calculating the total posterior probability of the set of all letter sequences corresponding to a specific word sequence?
Successive output characters are assumed to be conditionally independent given the entire speech input. E.g. $P(\phi\,\mathsf{b}\,\mathsf{e}\,\phi\,\mathsf{e}|\mathbf{A}) = P_1(\phi|\mathbf{A}) \times P_2(\mathsf{b}|\mathbf{A}) \times P_3(\mathsf{e}|\mathbf{A}) \times P_4(\phi|\mathbf{A}) \times P_5(\mathsf{e}|\mathbf{A})$*, where the subscripts on the right hand side represent the temporal order (column index) for the per-frame posterior probabilities.*

Q6. Diarization *14 points*

1. Draw the block diagram for the general cascaded/modular diarization approach. (5pt)
   The pink boxes are the essential modules of the diarization. Each of them can be optimized

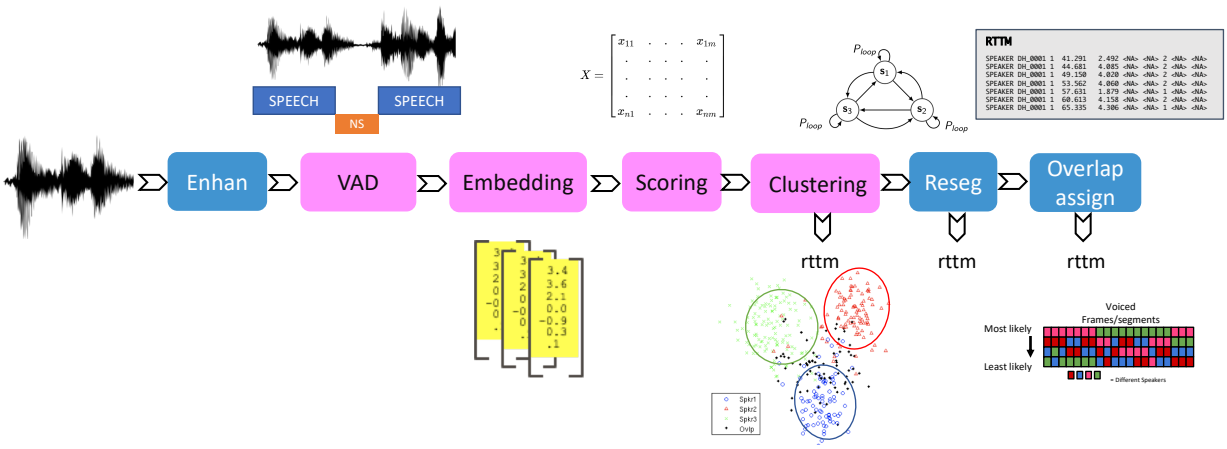

Figure 4: speaker verification pipeline

independently. The blue boxes are modules that boost the performance but that are optional.

- *First, the speech signal is assumed to be noisy, so an enhancement module reduces the effects of the noise.*
- *Voice activity detection (VAD), which removes the noises from the speech signal*
- *Embedding extraction, computes the mathematical representations of those chunks.*
- *Scoring computes a similarity matrix using either cosine scoring or probabilistic linear discriminant analysis (PLDA).*
- *Clustering, grouping together segments based on the scores that belong to the same speaker.*
- *If we want to refine the speaker segments, we do a resegmentation, based on speaker attribution at a frame level.*
- *There are ways to tackle the overlap; if we can detect the chunks of audio that contain overlapping speakers, we can assign two or three speakers according to some posterior probability.*

2. What are the main differences between cascaded/modular diarization and end-to-end diarization? (5pt)
   Answer: For the cascaded system each of the modules is optimized independently. In general, it needs an extra stage to deal with overlapping speakers. On the other hand, the end-to-end

diarization deals with the overlapping speakers in a natural way as it does a multi-label classification that involves predicting zero or one or two classes. Moreover, the end-to-end approach is a single deep neural network that is optimized to minimize a diarization loss based on binary-cross-entropy.

3. What are the two main metrics used for diarization? Give two examples of challenging scenarios, where diarization performance is poor. (4pt)

   - *DER* $= \frac{FA+MISS+SC}{total}$
     The most common metric is the DER (diarization error rate), composed of summing up false alarms (FA), missed detection (MISS) and speaker confusion(SC) divided by the total time of the recording. The false is the duration of non-speech mistakenly classified as speech. The missed detection is the duration of speech incorrectly classified as non-speech and the confusion is the duration of speakers classified as another speaker.

   - *JER* $= \frac{FA_{s_i}+MISS_{s_i}}{total_{s_i}}$
     Jaccard error rate is defined per reference speaker s_i*. The computation matches the reference speaker with the hypothesized speaker (the reference speaker matches at most one hypothesized speaker and viceversa). The optimal mapping is solved using the Hungarian algorithm. The FA is only for that reference speaker as it is the missed detection; the total is the union of the duration of the reference and the hypothesized speakers. Then summing up all the Jaccard error rates to calculate a global computation.*

4. Given a modular/cascaded system, how is the overlapping speaker problem addressed (bonus)?
   There are two different ways to address the problem:

   - *The first one is by using a technique called Variational Bayes HMM clustering (VBx) whose output is a speaker attribution matrix. We take the voice activity detection, a binary vector specifying speech and non-speech frames, and we get a hypothesis for a speaker in all voiced frames, overlapped or not. Next, consider a similar binary vector for overlap as well as the second most likely speaker from the attribution matrix. We intersect the overlap and the second most likely speaker, resulting in a secondary track in the final hypothesis.*
   - *The second way is by using EEND as post-processing. The idea is to extract the two longest sequences of speakers and apply EEND. This approach computes the overlapping speakers and solves the permutation ambiguity using previous knowledge of the speakers (if any). It assigns labels to the given segments. Then we get the next pair, excluding the parts already used. Then the system applies EEND, and once again solves for permutation ambiguity. The algorithm iterates until there are no pairs left.*

Extra Space