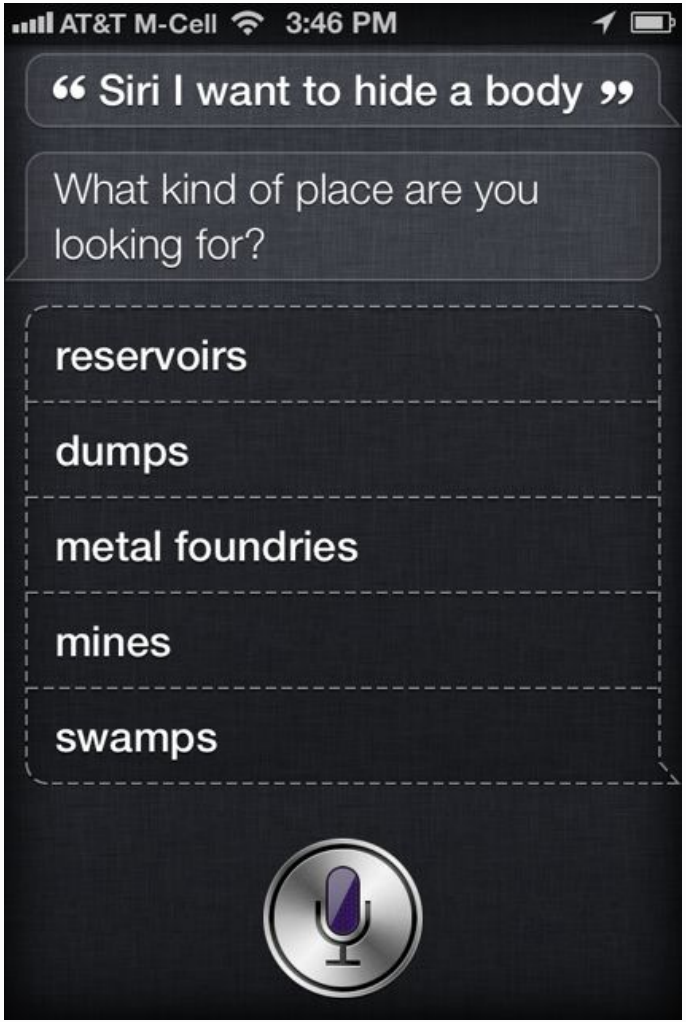


Image credit: PCWorld

CS 2731 Introduction to Natural Language Processing

Session 24: Speech technologies, ASR, TTS

Michael Miller Yoder

November 27, 2023

School of Computing and Information

Course logistics

- Project, project, project (due 12-14)
- Project peer review form due this Thu 11-16
  - Rate yourself and other group members
  - For identifying issues in workload distribution among group members
  - Not used for final project grading
- Feedback on basic working systems this week

Course logistics

- Course evaluations (OMETs) are open
- Will close Dec 10

https://go.blueja.io/BEBlAj4xFEydvsaSR780YA

Overview: Speech technologies, ASR, TTS

- Automatic speech recognition (ASR)
  - ASR feature extraction
  - ASR system architecture
  - ASR evaluation
- Speech datasets
- Text-to-speech (TTS)

Automatic speech recognition (ASR)

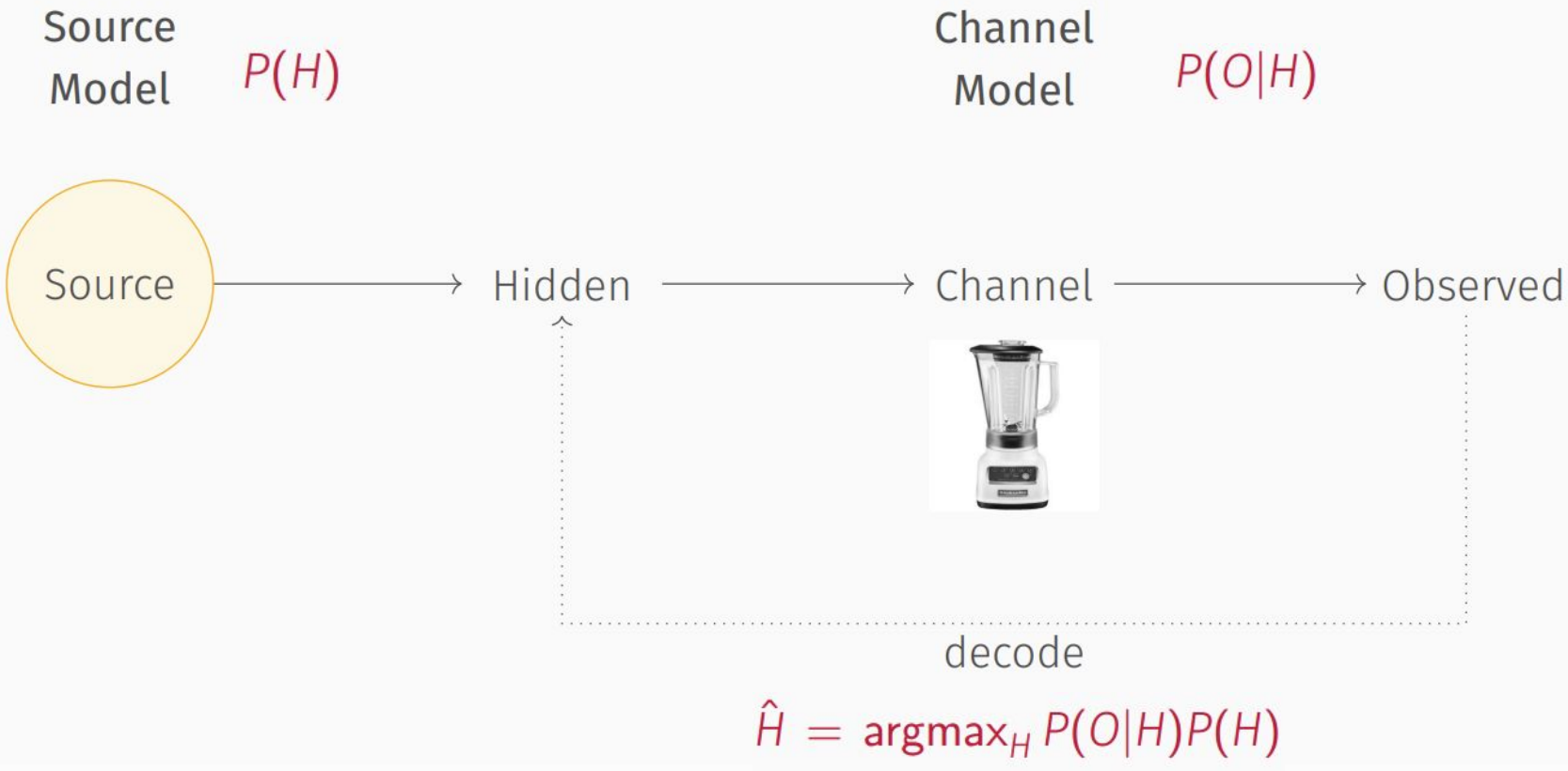
The task of automatic speech recognition (ASR)

input: an acoustic signal containing spoken utterances

output: an orthographic representation of the utterances

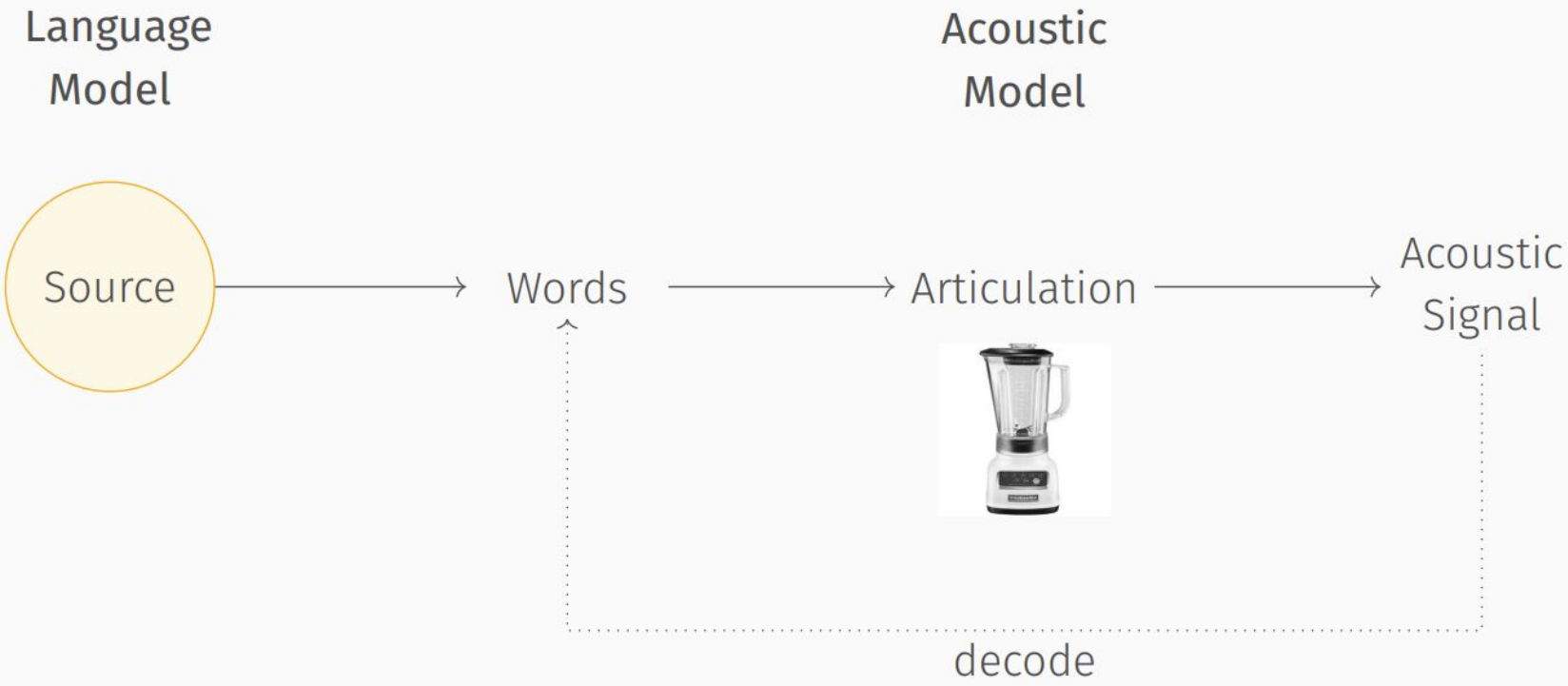
Slide credit: David Mortensen

Decoding with the Noisy Channel Model



Slide credit: David Mortensen

ASR and the Noisy Channel



Slide credit: David Mortensen

An informal example

Ping has a hearing loss in both ears. This means that his ACOUSTIC MODEL is weak. Out of context, he has trouble recognizing sounds or sequences of sounds.

In some domains, though, Ping has a strong LANGUAGE MODEL. When people are talking about subjects he has discussed a lot, he can identify which of the sequences of sounds that are probable under his acoustic model are likely utterances in that domain. He can identify words fairly accurately then.

If he is talking to people about different subjects (for which he doesn't have a strong language model) his weak acoustic model shows. He hears all sorts of sequences that were not intended by the speakers.

Ping is not an ASR system, but he isn't completely different from one.

Slide adapted from David Mortensen

ASR feature extraction

From Soundwaves to Integers

Slide credit: David Mortensen

Sampling

- A sound wave is continuous
- It is transduced into a continuously variable electric current by a microphone
- To discretize this signal, it must be SAMPLED at fixed intervals
- This sampling rate is typically measured in Hz (cycles per second) or KHz (thosands of cycles per second)
- Two common sampling rates are 8 KHz and 16 KHz.
- The maximum frequency that a sampled sound can represent is $\frac{\text{sample rate}}{2}$ (the Nyquist Frequency)

Slide credit: David Mortensen

Quantization

- To obtain a usable waveform, it is necessary to quantize it (convert all of the floating-point values to integers)
- The integers are usually 8 bit (−128–127) or 16 bit (−32768–32767)
- Values that are closer together than the quantum size are represented identically
- We represent the nth quantized sample as $x[n]$

Slide credit: David Mortensen

Windowing

- Spectral features have to be extracted from a WINDOW (a subsequence of samples)
- The window has a width (e.g., 25 ms) and process has a STRIDE (the among by which the window is shifted at each iteration (e.g., 10 ms)

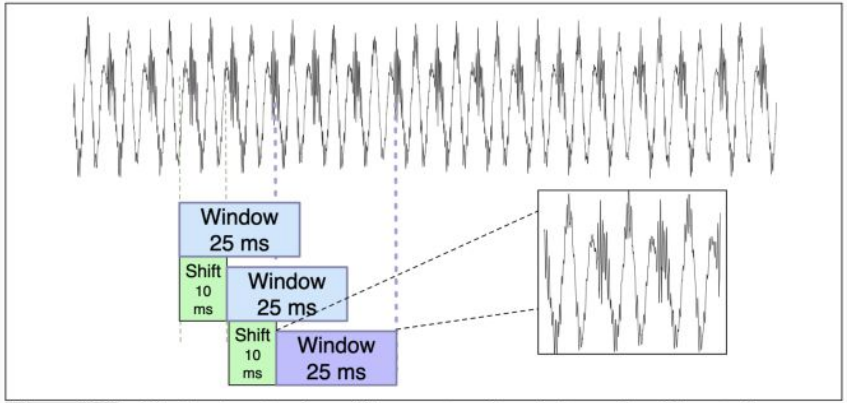


Figure 26.2 Windowing, showing a 25 ms rectangular window with a 10ms stride.

Slide credit: David Mortensen

Frames

The signal extracted at each window is calld a FRAME

To reiterate, the frame has a:

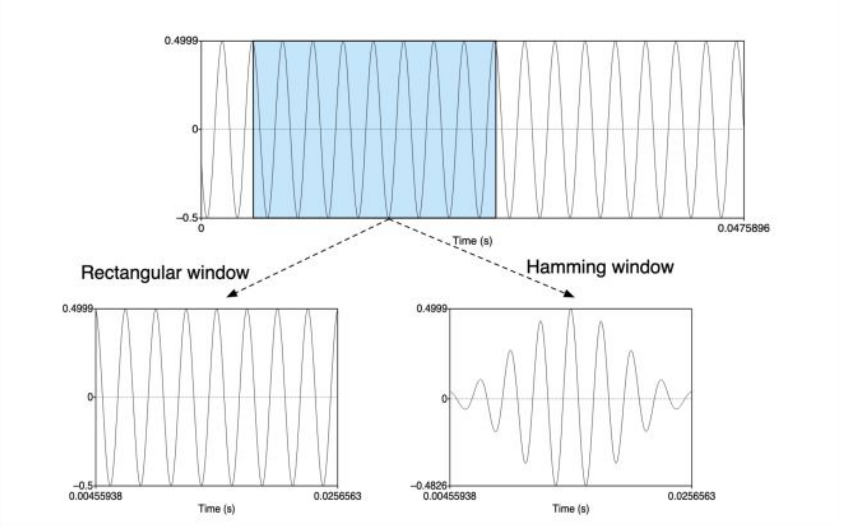
- WINDOW SIZE or FRAME SIZE
- FRAME STRIDE
- FRAME SHAPE (this is new)

For many applications, we will use a HAMMING WINDOW rather than a RECTANGULAR WINDOW

Slide credit: David Mortensen

Hamming Windows

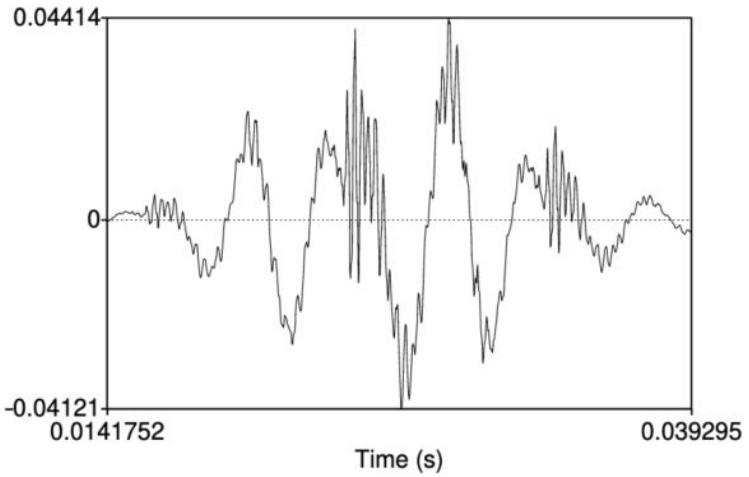
The window on Slide 19 was a rectangular window: the slice of samples with no additional processing. This is not good for Fourier analysis, so we typically use Hamming Windows instead:



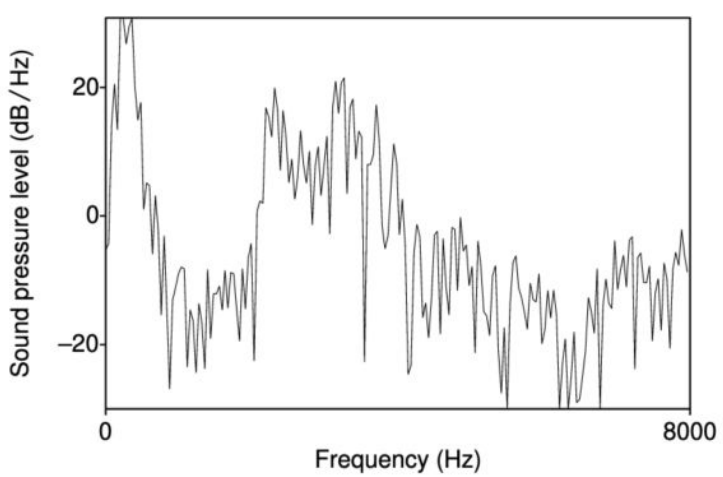
Slide credit: David Mortensen

The Discrete Fourier Transform

The input to a DFT (DISCRETE FOURIER TRANSFORM) is a hamming-windowed portion of a waveform (a). The output is a the corresponding spectrum (b):



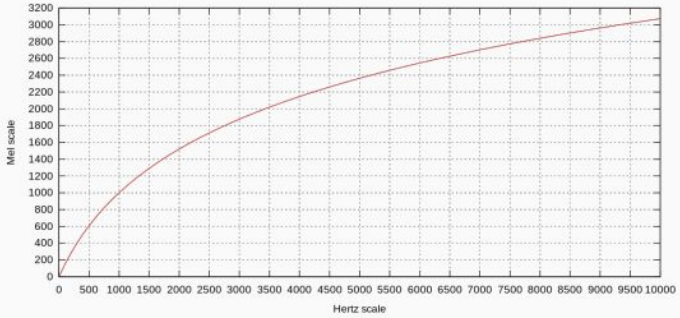
(a)



(b)

Slide credit: David Mortensen, Jurafksy & Martin

The Mel is a Psychoacoustic Unit of Pitch



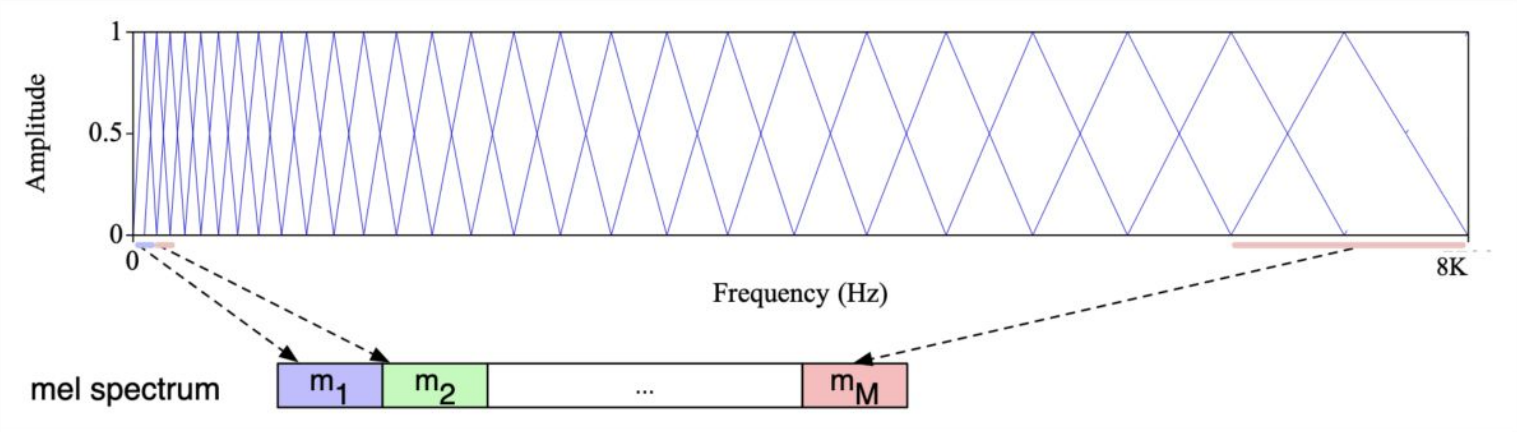
- Mimics frequency response of human ear
- At the bottom of the scale, small differences in Hz correlate with large differences in Mels
- At the top of the scale, large difference in Hz correlate with small differences in Mels

$$\text{mel}(f) = 1127 \ln(1 + \frac{f}{700}) \quad (1)$$

Slide credit: David Mortensen

The Mel Filter Bank

We can create a bank of filters that collect energy from each feature band where bands are spread logarithmically (high resolution at low frequencies; low resolution at high frequencies):



Slide credit: David Mortensen

We Take the Log of the Mel Spectrum Values

Once we have all of the spectrum values from the Mel Fiter Bank, we take the log of each spectrum value (since the human response to signal level—like the human response to frequency—is logarithmic).

This also makes the spectrum less sensitive to incidental variation in signal strength (like the speaker moving closer to or farther from the mic).

Slide credit: David Mortensen

We now have the acoustic features that we need to train and employ a modern ASR system.

Slide credit: David Mortensen

Architecture of an ASR system

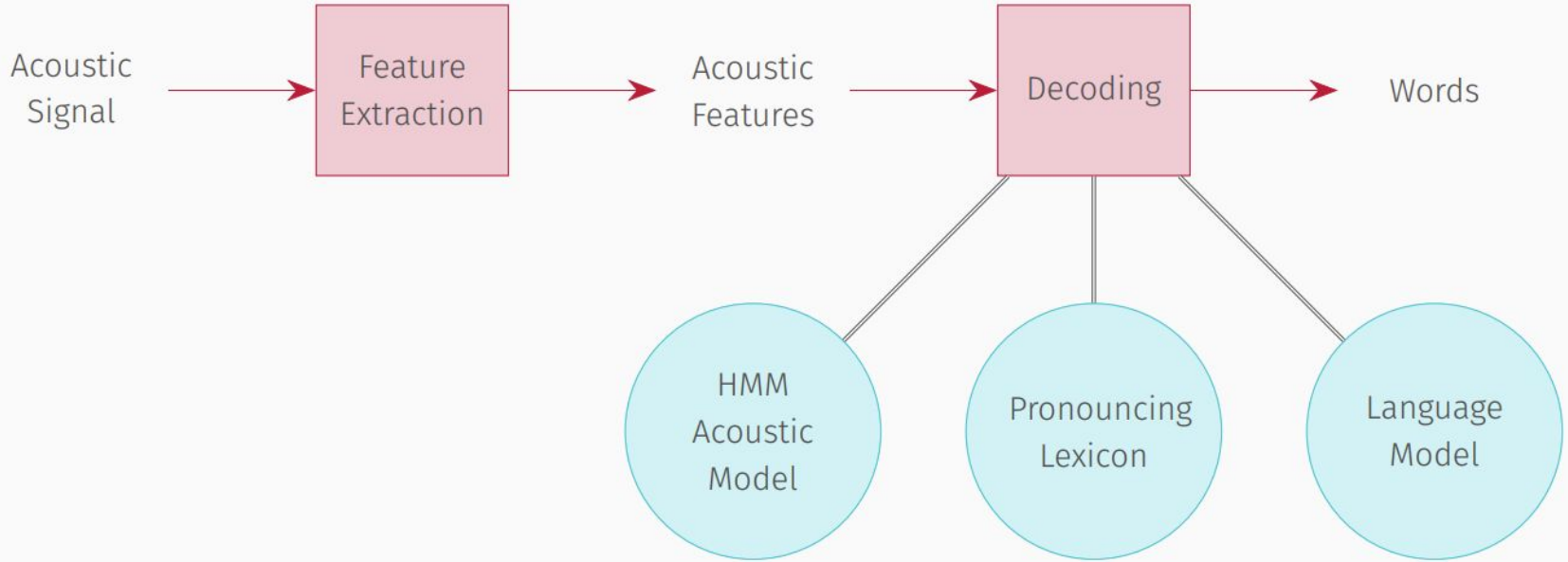
The Traditional ASR Architecture Was Simple in Essence

To calculate $P(O|W)P(W)$ (where O are observations and W are words):

Model $P(W)$	ngram language model
Model $P(O|W)$	HMM (one hidden state per phoneme or word)

Slide credit: David Mortensen

The Traditional ASR Architecture Was Quite Complicated



Slide credit: David Mortensen

The Modern Architecture is Often Variant of Encoder-Decoder

- Can use either RNNs or Transformers
- Is basically the same as the architecture that we introduced in Lecture 15
- **Input:** log mel spectral features
- **Output:** letters (or subwords like BPE or wordpiece units)

Slide credit: David Mortensen

Why Attention-Based Encoder-Decoder (AED)?

In ASR, input and output sequences have dramatically different lengths.

- The input consists of a sequence of frames (on the order of 16K per second, before subsampling)
- The output consists of a few words per second, each of which consists of a few letters or wordpieces

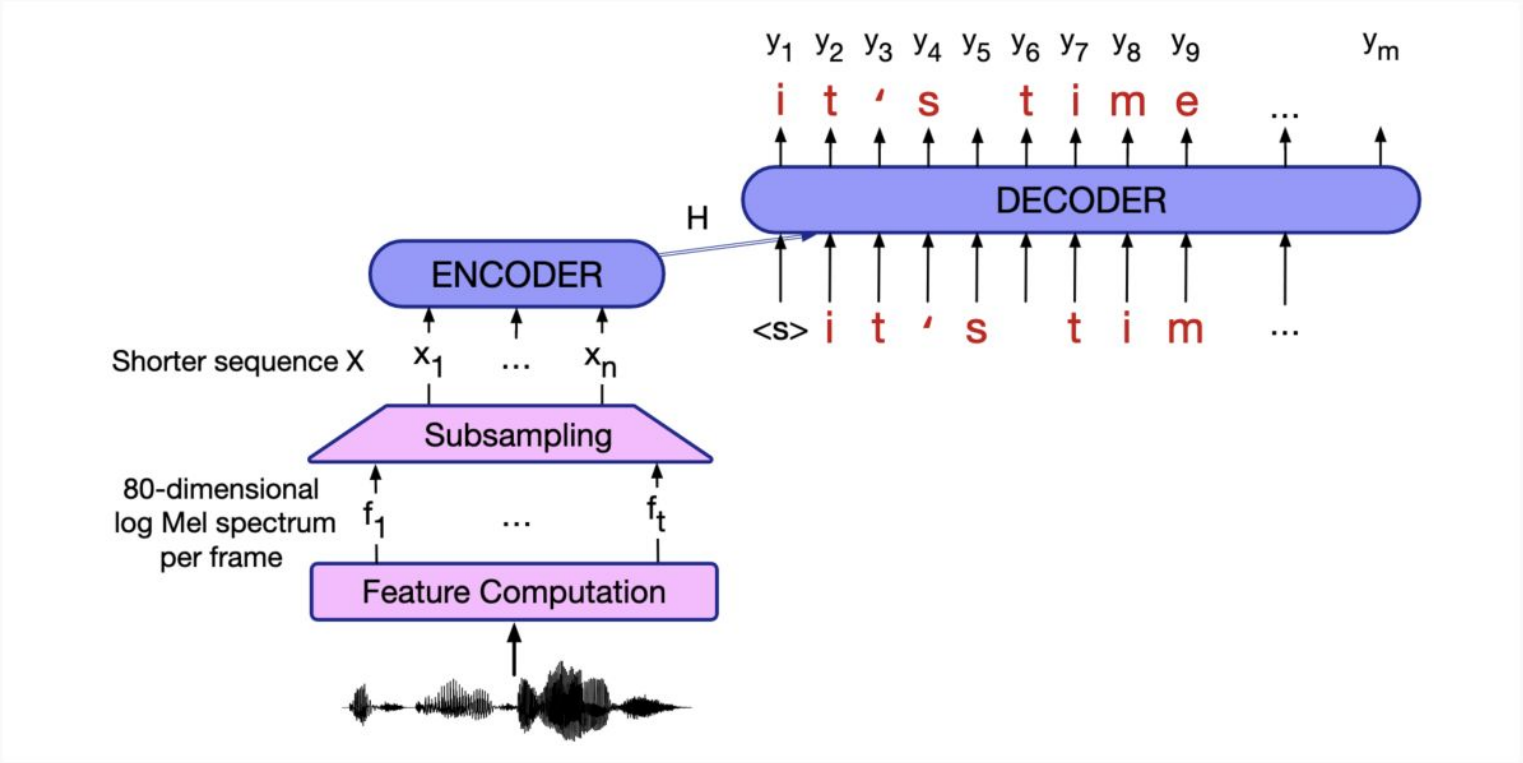
This is an especially good match for Encoder-Decoder

- The acoustic signal for a time interval is encoded as a hidden representation *H*
- This representation is then decoded as a sequence of letters (or wordpieces)

This is much simpler than the old way, at some level.

Slide credit: David Mortensen

What a Schematic Architecture for an Encoder-Decoder ASR System Looks Like



Slide credit: David Mortensen, Jurafsky & Martin

Subsampling Mitigates the Fact that the Inputs Are so Long Relative to the Outputs

- You need some kind of compression to shorten the acoustic feature sequence
- (Or a loss function like CTC, that deals well with compression).
- The simplest algorithm for subsampling is LOW FRAME RATE
  - Concatentate acoustic feature vector f_i with f_{i-1} and f_{i-2}, yielding a feature vector three times as long as f_i
  - Delete f_{i-1} and f_{i-2}
  - Instead of a 40-dimensional vector every 10 ms, we have a 120-dimensional vector every 30 ms
  - Sequence length is one third as long

After subsampling, ASR looks just like MT.

Slide credit: David Mortensen

Sometimes We Add a Language Model to AED

- An enoder-decoder model is essentially a (conditional) language model
- However, sometimes one needs a better language model than can be trained on the transcriptions of speech available in one's training data
- A simple way of incorporating an LM is to use BEAM SEARCH to obtain and N-BEST LIST of outputs and then RESCORE them using the LM

Slide credit: David Mortensen

Adding a Language Model for Rescoring an n-Best List

- Beam search explores a space by expanding the most promising node in a limited set (with only a predetermined number of best partial solutions—the beam—being kept as candidates)
- This can be used to find the n best candidate outputs from the encoder-decoder
- These can be rescored by interpolating the LM score and the encoder-decoder score (with a weight λ tuned on a held-out set).
- A length factor enters in as well.

$$\text{score}(Y|X) = \frac{1}{|Y|_c} \log P(Y|X) + \lambda \log P_{LM}(Y)$$

where X is the input, Y is the hypothesis, and $|Y|_c$ is the length of the hypothesis in characters

Slide credit: David Mortensen

Training AED with Cross-Entropy Loss

AED is usually trained with normal cross-entropy loss. For a single letter y_i, that is

$$L_{CE} = -\log p(y_i|y_1, \ldots, y_{i-1}, X)$$

and for a whole sequence, it is the sum of these losses:

$$L_{CE} = -\sum_{i=1}^{m} \log p(y_i|y_1, \ldots, y_{i-1}, X)$$

This loss is backpropagated through the entire model (decoder and encoder) to train it. Normally, teacher forcing is used (the decoder history is forced to be the correct gold y_i all or some of the time).

Slide credit: David Mortensen

Evaluating ASR with Word Error Rate (WER) and Character Error Rate (CER)

Word Error Rate

WORD ERROR RATE is, in essense, Levenshtein distance over words, normalized by the length of the reference (in words). It is the most common metric for evaluating ASR.

After two strings of words are aligned, you can observe three kinds of edits:

insertion	present in hypothesis but not in reference
deletion	present in reference but not in hypothesis
substitution	corresponding words in hypothesis and reference are different

REF:	i	***	**	UM	the	PHONE	IS	i	LEFT	THE	portable	****	PHONE	UPSTAIRS	last	night
HYP:	i	GOT	IT	TO	the	*****	FULLEST	i	LOVE	TO	portable	FORM	OF	STORES	last	night
Eval:		I	I	S		D	S		S	S		I	S	S		

WER is the sum of these edits, in the test set, over the length of the test set (in words). A useful tool is NISTs `sclite`.

Slide credit: David Mortensen

Character Error Rate

CHARACTER ERROR RATE (CER) is also used to evaluate ASR systems, particularly those that recognize subword units.

It is similar to WER, but is computed over characters instead of words.

Slide credit: David Mortensen

Speech datasets

Overview of Speech Datasets

Name	Hours	KHz	Notes
LibriSpeech	> 1000	16 KHz	Read speech (books from the LibraVox project)
Switchboard	240	8 KHz	Prompted telephone conversations between strangers
CALLHOME	60	8 KHz	Unscripted telephone conversations between close friends and family
SBCSAE			Large corpus of varied, natural conversations
CORALL			150 interviews with speakers of Black English/African American Language
CHiME Challenge			Series of difficult ASR shared tasks
HKUST Mandarin Telephone Speech Corpus	> 200		ten-minute telephone conversations between speakers of Mandarin
AISHELL-1	170		Mandarin read speech in various domains

Slide credit: David Mortensen

The LibriSpeech Dataset Consists of Read Audiobooks

- The LibriSpeech corpus consists of read speech from books
- Based on the LibriVox project
- A variety of genres
- More than 1000 hours of audio at 16 KHz

Slide credit: David Mortensen

The CALLHOME Corpus Consists of Unscripted Phone Conversations between Friends and Family

- 120 conversations, each 30 minutes long (=60 hours)
- unscripted, unprompted telephone conversations between close friends and family members
- Native speakers of English
- Collected in the late 1990s

Slide credit: David Mortensen

HKUST MTSC and AISHELL-1 Are Speech Corpora for Mandarin

- The HKUST Mandarin Telephone Speech Corpus
  - Telephone conversations from speakers of Mandarin throughout China
  - 1206 ten-minute telephone conversations
  - Some conversations between friends and others between strangers
- AISHELL-1
  - Read speech from various domains
  - Mostly speakers from Northern China
  - 170 hours

Slide credit: David Mortensen

State-of-the-Art Results on Several Datasets

English Tasks	**WER%**
LibriSpeech audiobooks 960hour clean	1.4
LibriSpeech audiobooks 960hour other	2.6
Switchboard telephone conversations between strangers	5.8
CALLHOME telephone conversations between family	11.0
Sociolinguistic interviews, CORAAL (AAL)	27.0
CHiMe5 dinner parties with body-worn microphones	47.9
CHiMe5 dinner parties with distant microphones	81.3
Chinese (Mandarin) Tasks	**CER%**
AISHELL-1 Mandarin read speech corpus	6.7
HKUST Mandarin Chinese telephone conversations	23.5

Slide credit: David Mortensen, Jurafksy & Martin

Text-to-speech (TTS)

The Text-to-Speech Task

input An orthographic representation of an utterance

output An acoustic signal representing the same utterance

Slide credit: David Mortensen

Modern TTS Usually Uses Encoder-Decoder Architectures

- Text-to-speech is the task of converting a string of words into a corresponding acoustic signal
- It is ASR in reverse
- Like ASR, TTS is basically a sequence-to-sequence task
- As with ASR, modern approaches to TTS use encoder-decoder models

Slide credit: David Mortensen

The Two Steps of TTS

spectrogram prediction: map strings of letters to mel spectrograms
(sequences of mel spectral values over time)

vocoding: map mel spectrograms to waveforms

Slide adapted from David Mortensen

One challenging aspect of TTS is text normalization

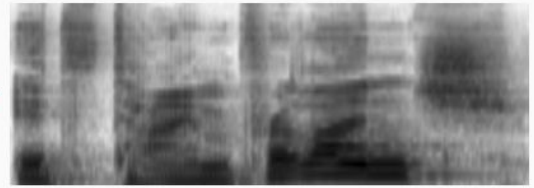
Abbreviations, acronyms, numbers, dates, and so on must be converted to a canonical form that can be verbalized:

semiotic class	examples	verbalization
abbreviations	**gov't**, *N.Y., mph*	government
acronyms read as letters	**GPU**, *D.C., PC, UN, IBM*	G P U
cardinal numbers	**12**, *45, 1/2, 0.6*	twelve
ordinal numbers	*May* **7**, *3rd, Bill Gates III*	seventh
numbers read as digits	*Room* **101**	one oh one
times	*3.20,* **11:45**	eleven forty five
dates	**28/02** *(or in US, 2/28)*	February twenty eighth
years	**1999**, *80s, 1900s, 2045*	nineteen ninety nine
money	**$3.45**, *€250, $200K*	three dollars forty five
money in tr/m/billions	**$3.45 billion**	three point four five billion dollars
percentage	**75%** *3.4%*	seventy five percent

Slide credit: David Mortensen

Spectrogram Prediction

It’s time for lunch!



Carried out by an encoder and decoder.

Slide credit: David Mortensen

Converting a Spectrum to a Waveform

Carried out by a specialized component like WaveNet (an autoregressive network)

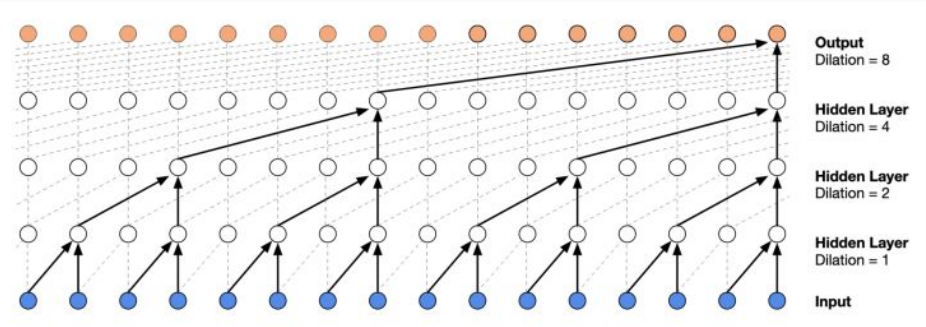
Slide credit: David Mortensen

WaveNet in More Detail

- Probability of a waveform, a sequence of 8-bit μ-law values $Y = y_1, \ldots, y_t$ given an intermediate input mel spectrogram h can be computed as

$$p(Y) = \prod_{t=1}^{t} P(y_t | y_1, \ldots, y_{t-1}, h_1, \ldots, h_t)$$

- This can be modeled with a stack of special convolutional layers (with dilated convolutions) and a special non-linearity function
- Intuition: dilated convolutions dilate backward (increase in dilation) as you travel upwards through layers.



Slide credit: David Mortensen

Evaluating TTS

The only reliable way of evaluating a TTS system right now is with human listeners.

MOS (MEAN OPINION SCORE) play synthesized utterances to listeners and ask them to rate how good the utterances are (scale of 1–5)

AB tests play the synthesized output given the same input sentence to human listeners and ask them which they like better

Slide credit: David Mortensen

Wrapping up

- Automatic speech recognition (ASR) converts speech to text
- ASR involves lots of feature extraction: sampling, but also conversion to sequences of mel spectogram values
- Contemporary approaches to ASR use an encoder-decoder approach
- ASR is evaluated with word or character error rate
- Speech datasets include read speech like LibriSpeech and difficult conversational speech in noisy environments like CHiME
- Text-to-speech systems also use encoder-decoder approaches

Questions?