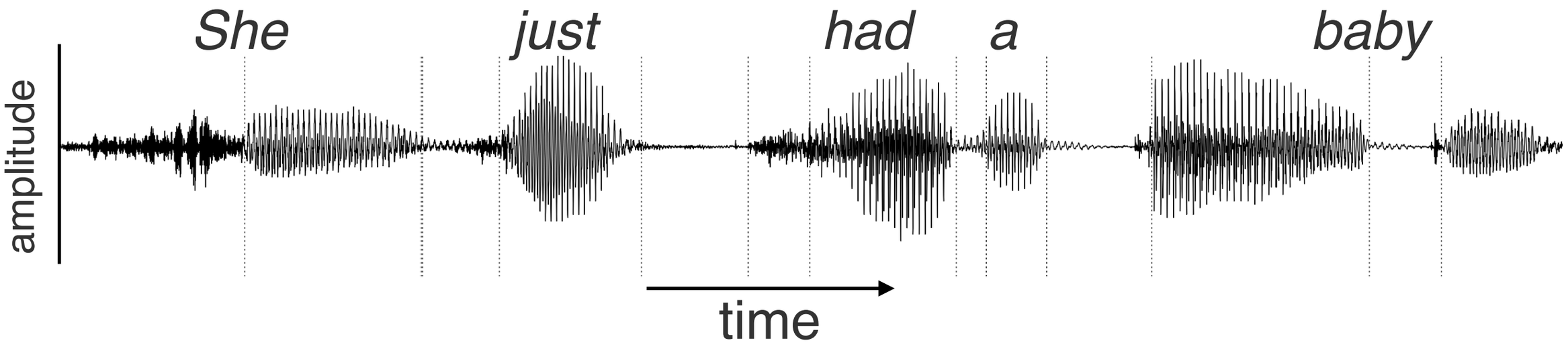
Speech Preliminaries + Noisy Channel Model

EECS 183/283a: Natural Language Processing

Recap:Speech Representations



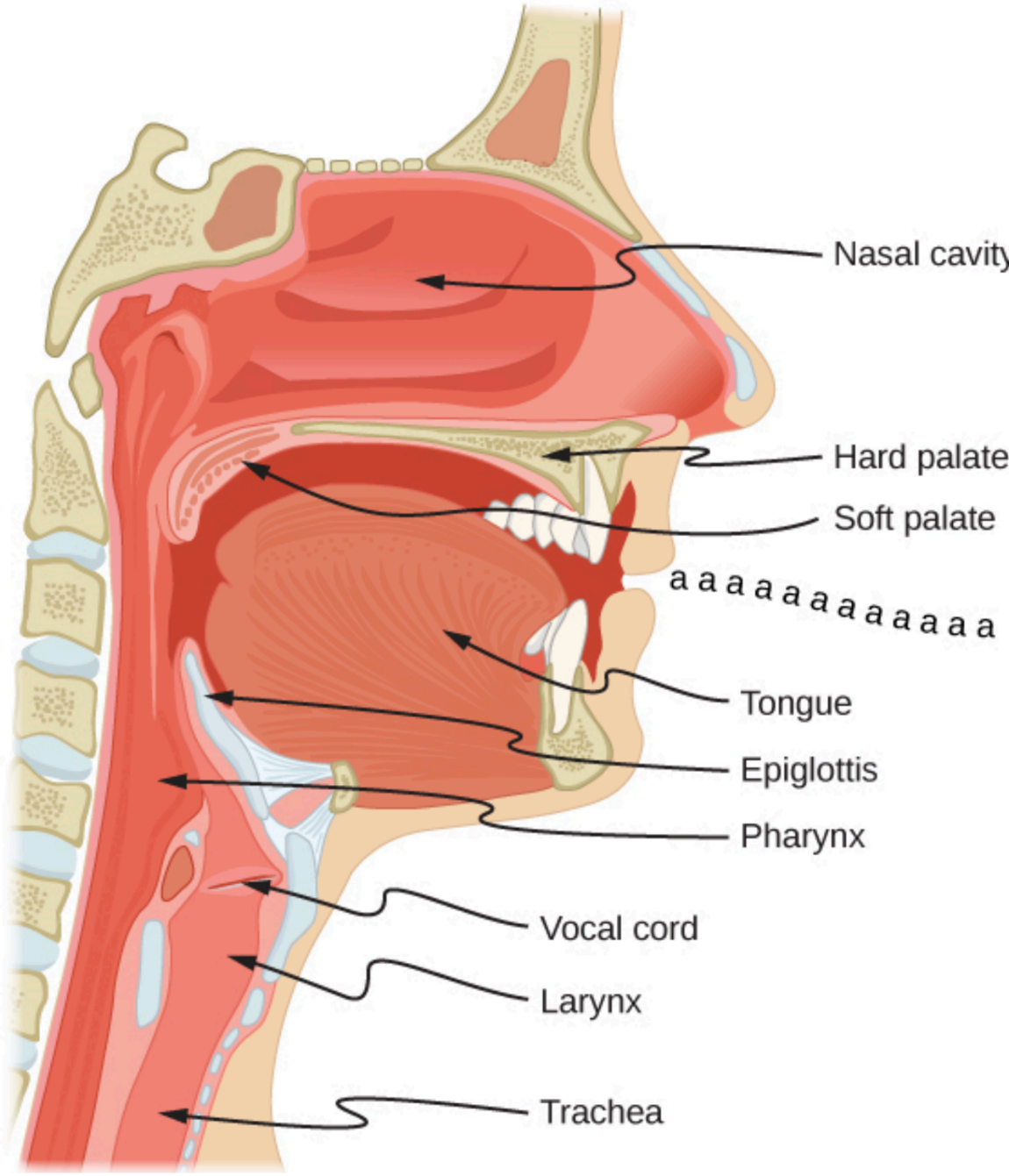
* The result of the vocal articulation is an acoustic pressure wave

* Speech can thus be represented as an acoustic waveform

* Waveforms are continuous time series cannot be easily analyzed or interpreted, or computed with

* Signal processing can give more interpretable information

Speech Production



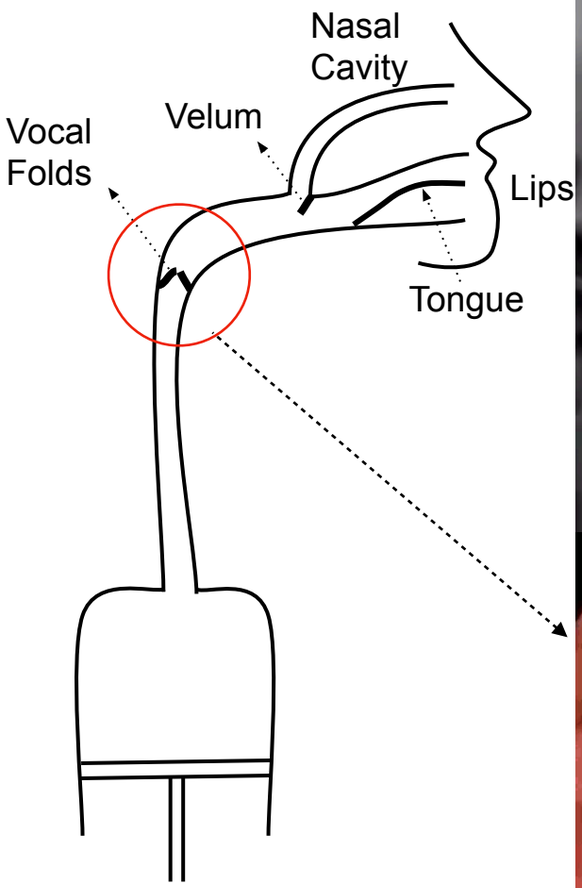
Vocal articulators that produce speech

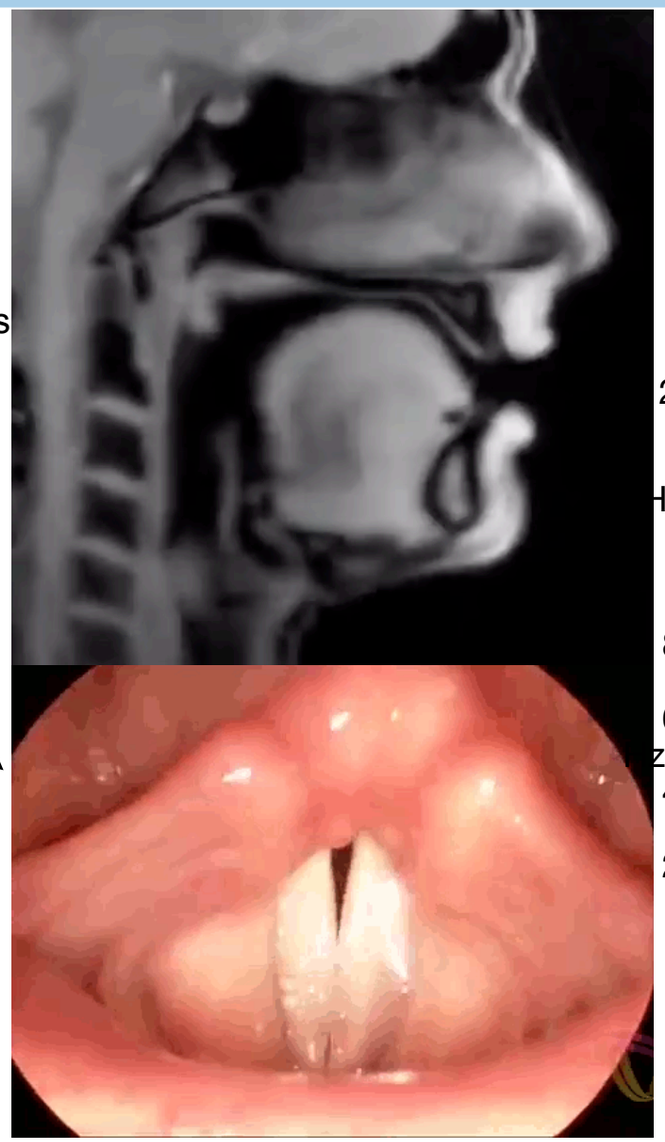
* Air passing through vocal articulators produces speech

* Vocal folds, tongue, jaw, lips, velum are both independently and jointly controlled to produce different sounds
  * eg. Vocal fold vibration causes voicing.

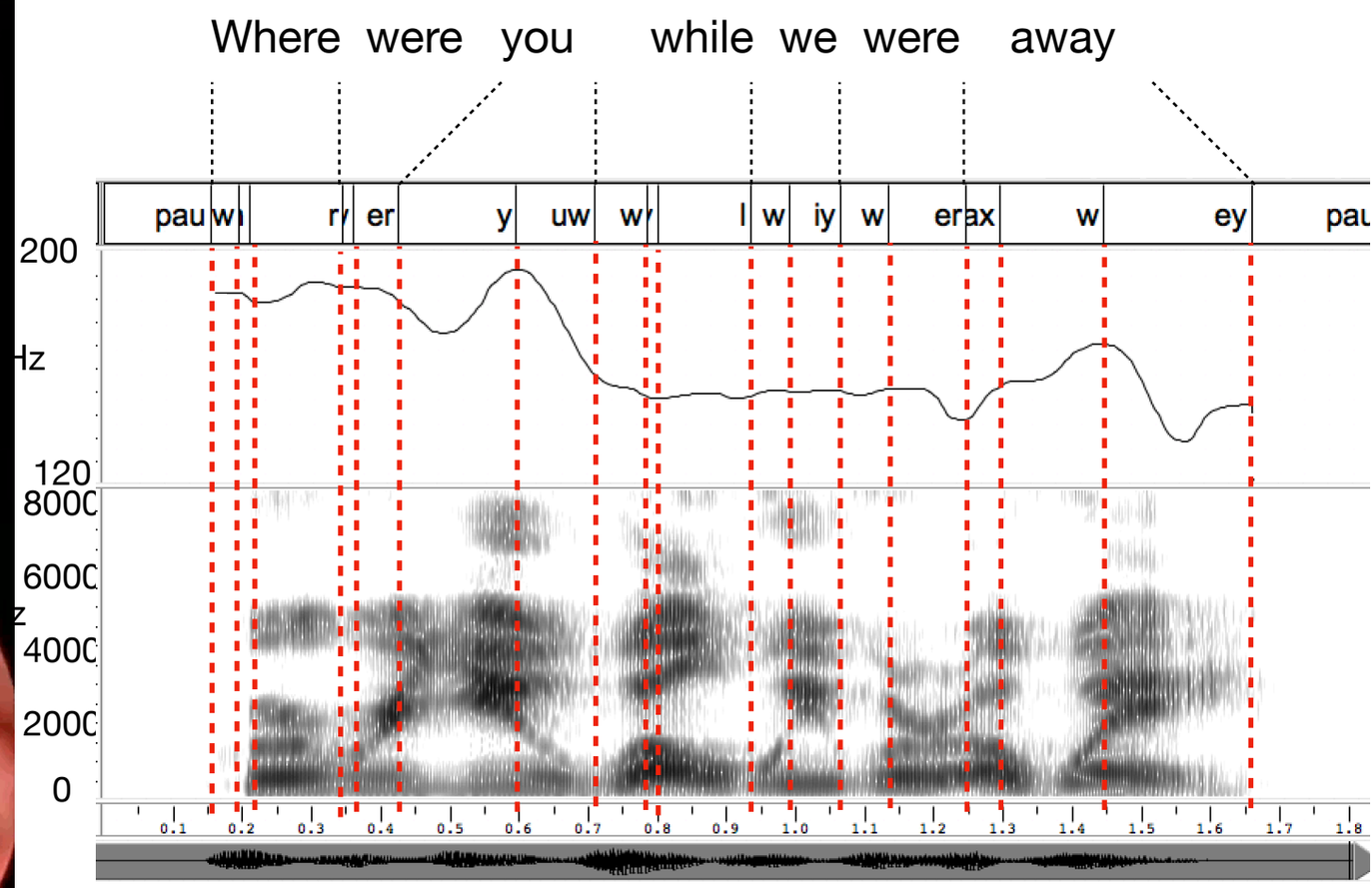
* The output of vocal articulation is an acoustic pressure wave

The most complex action we do is speaking





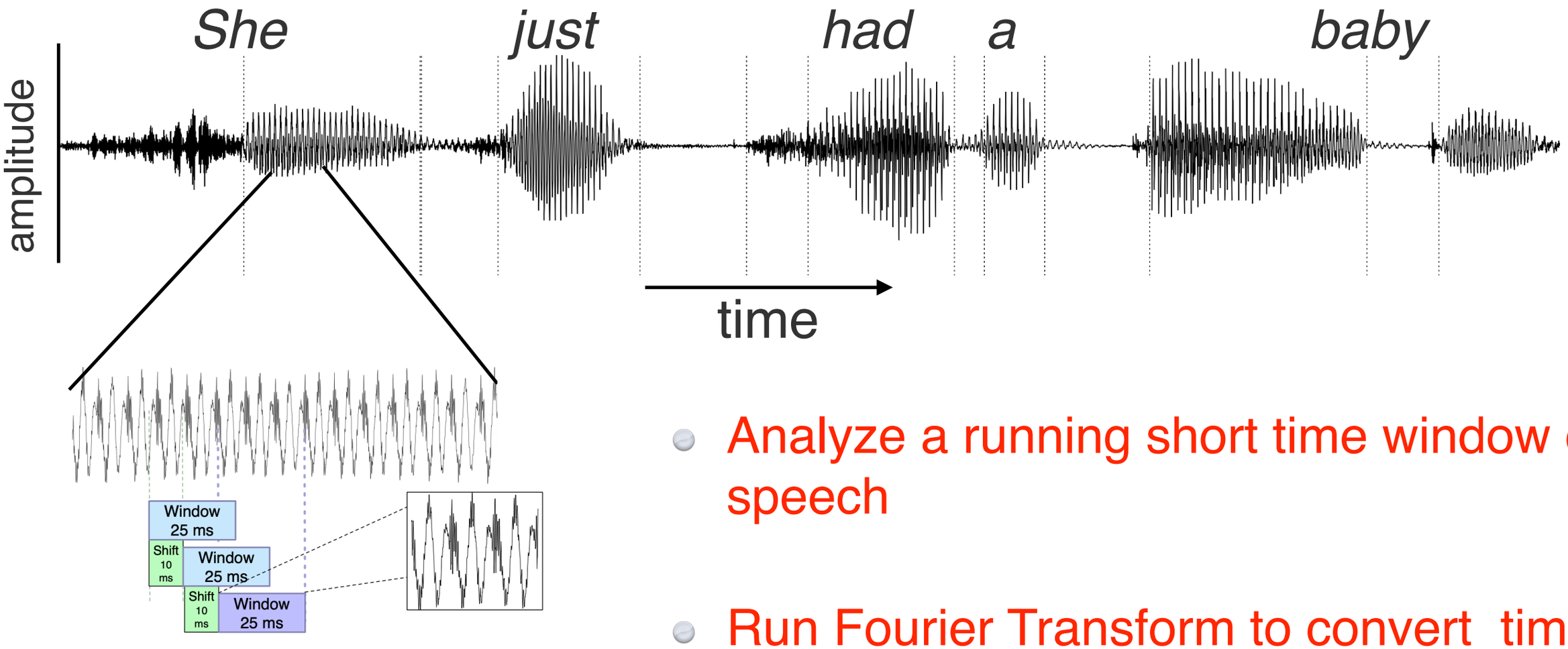
Sound filter



Sound source

Time

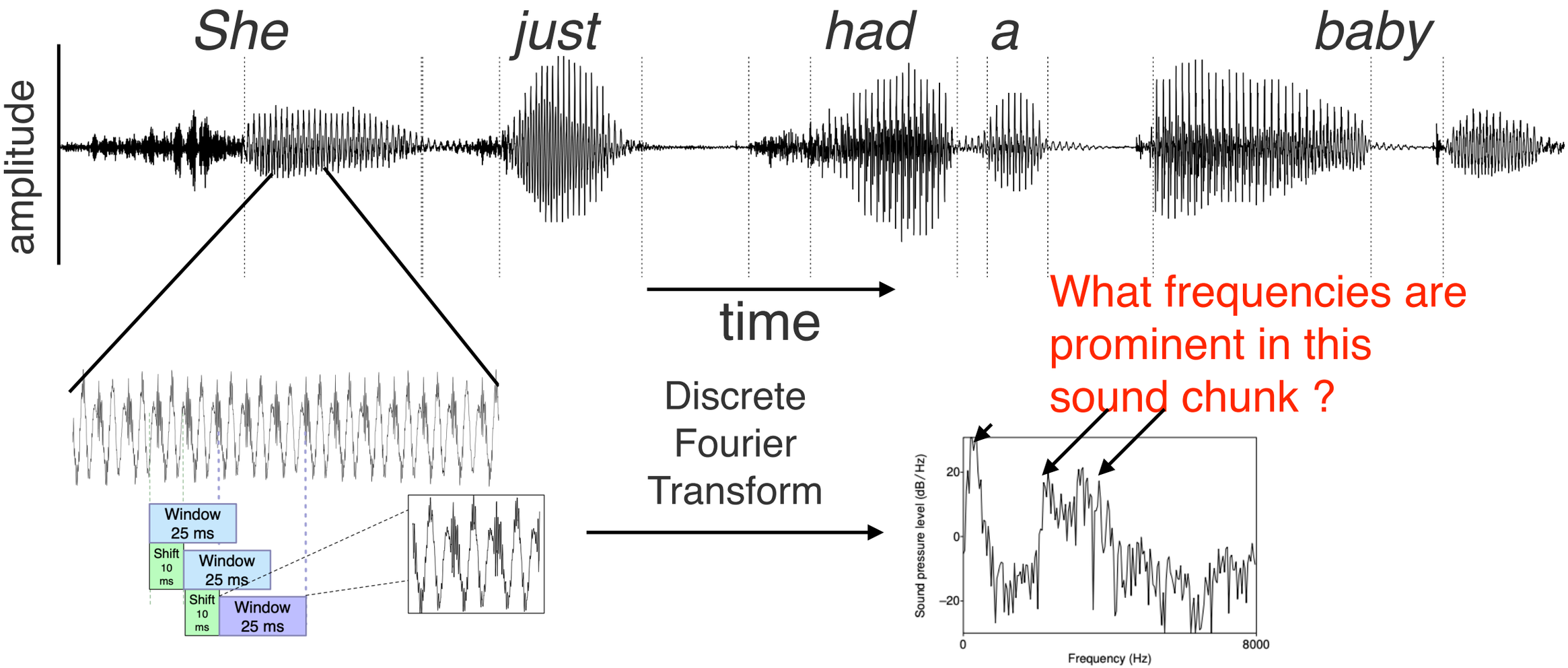
Speech Waveform



- Analyze a running short time window of speech

- Run Fourier Transform to convert time to Time-frequency representation

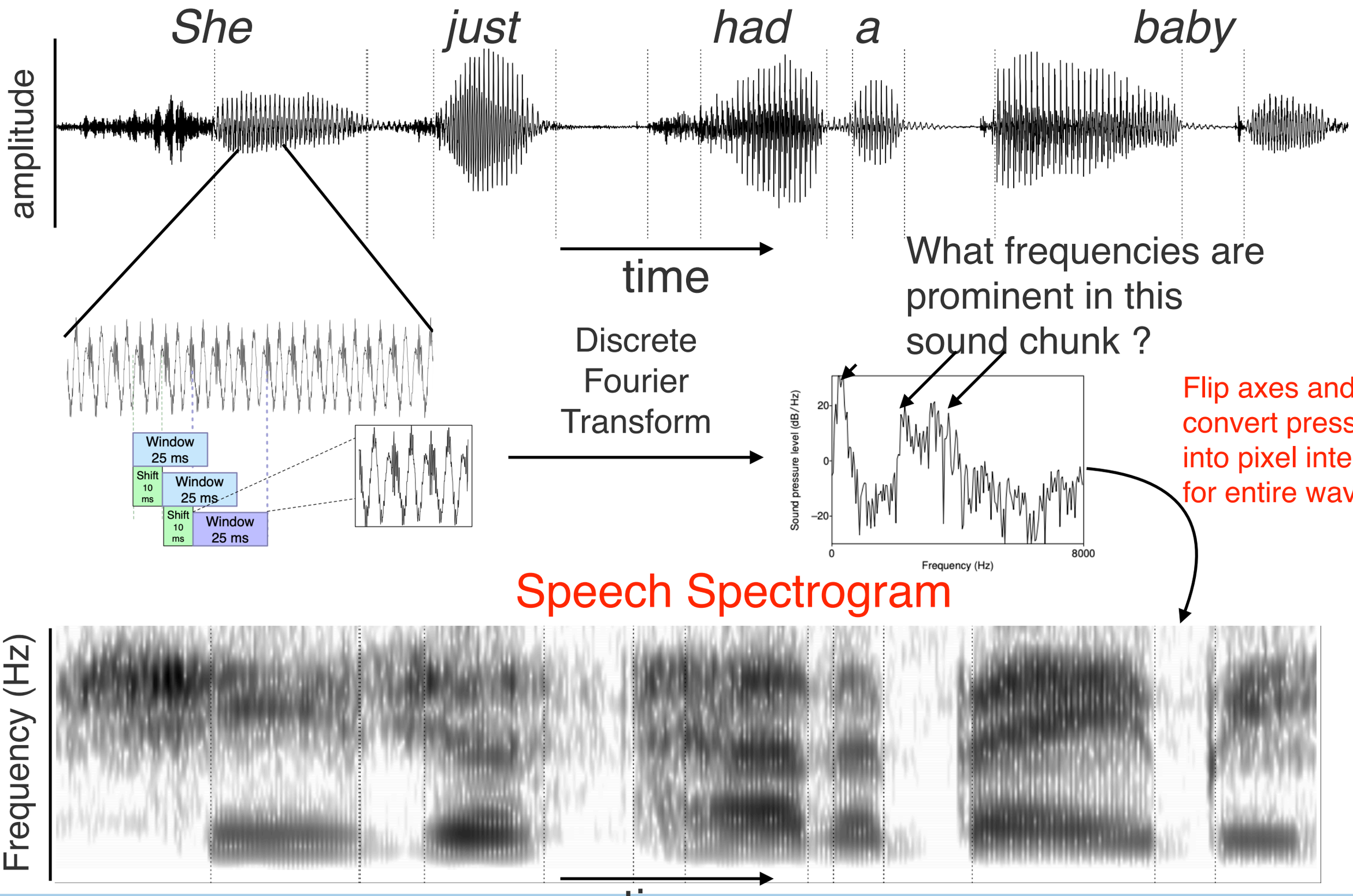
Speech Waveform



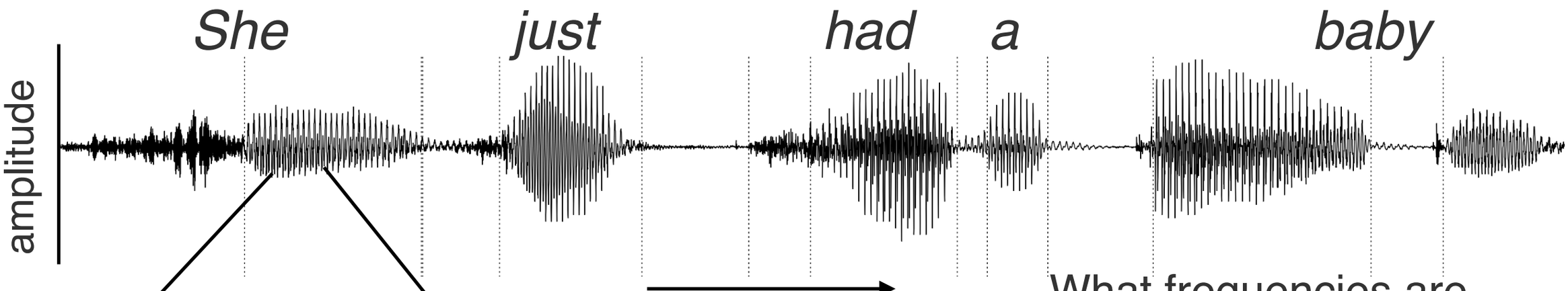
Discrete
Fourier
Transform

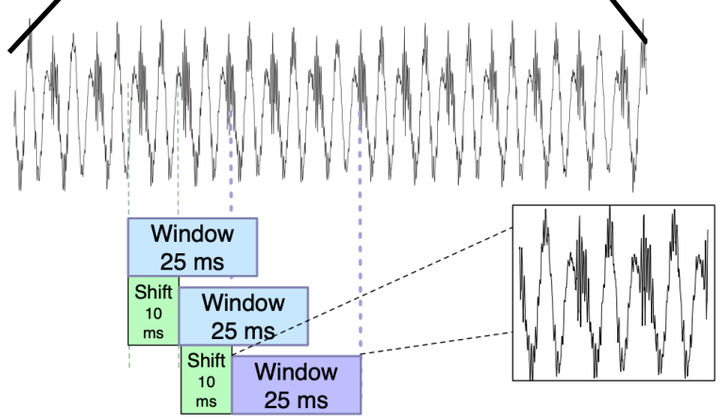
What frequencies are prominent in this sound chunk ?

Speech Spectrogram



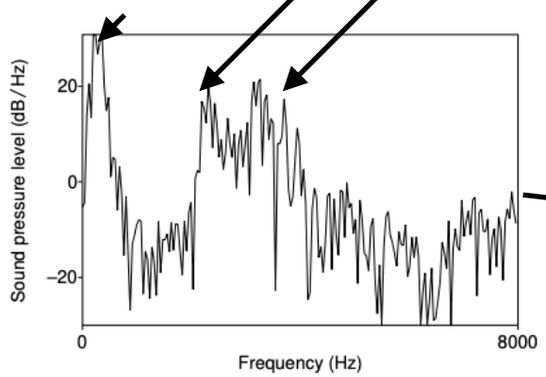
Speech Spectrogram





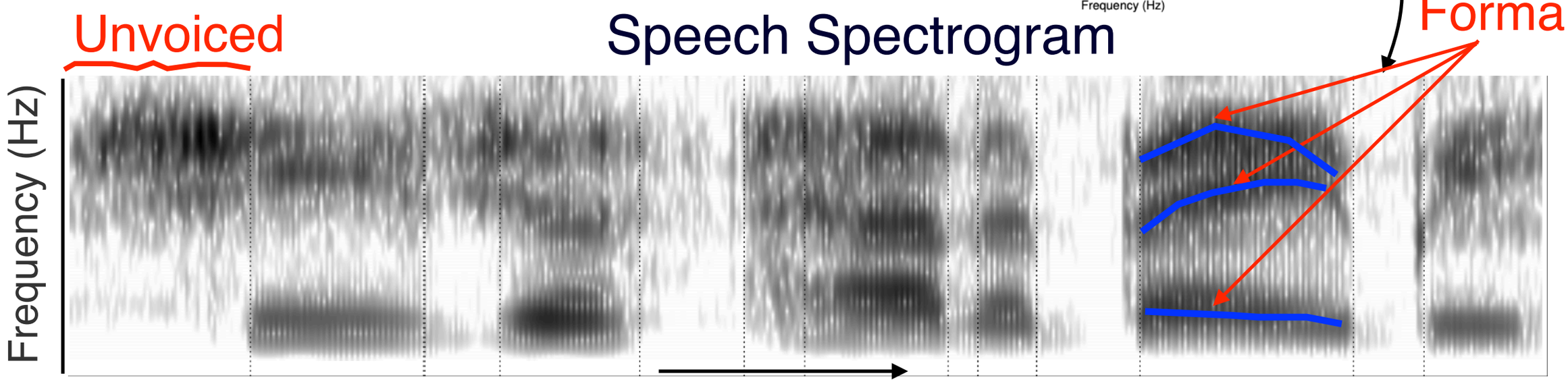
Discrete Fourier Transform

What frequencies are prominent in this sound chunk ?



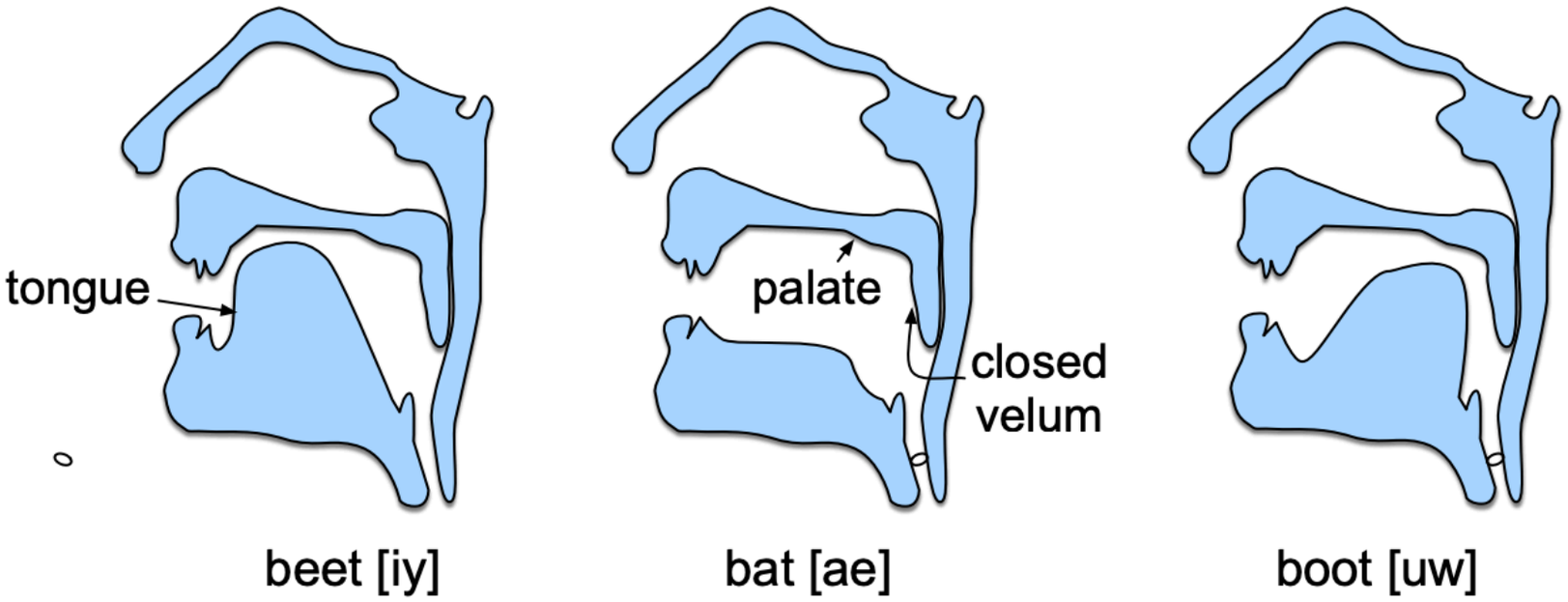
Flip axes and convert pressure into pixel intensity for entire waveform

Speech Spectrogram

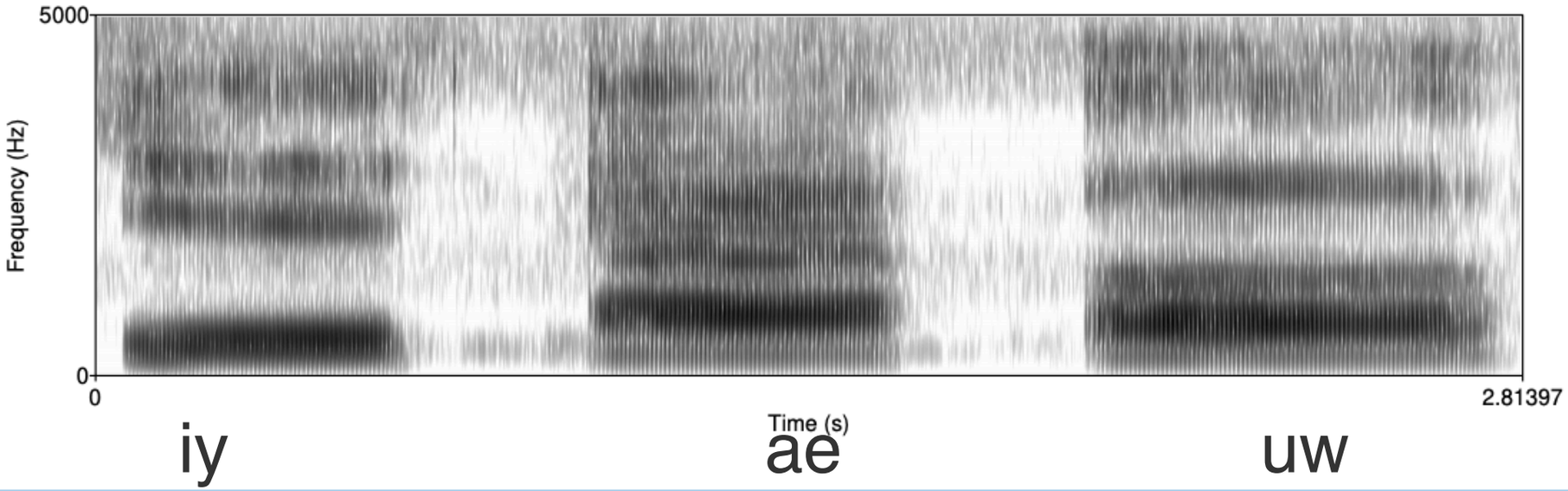


Phonetics

Study of speech sounds — their physical production, spectral and perceptual properties

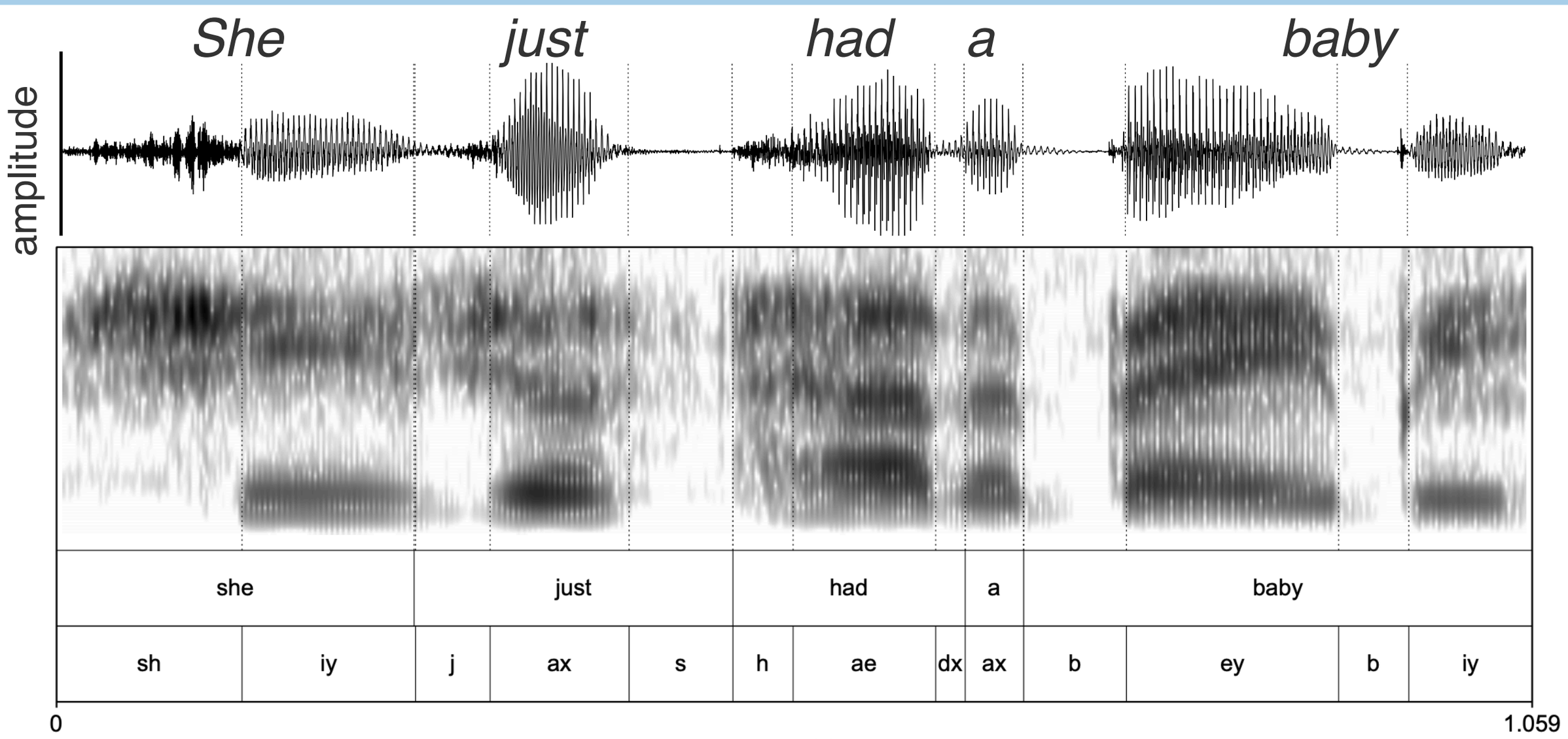


Articulatory Phonetics



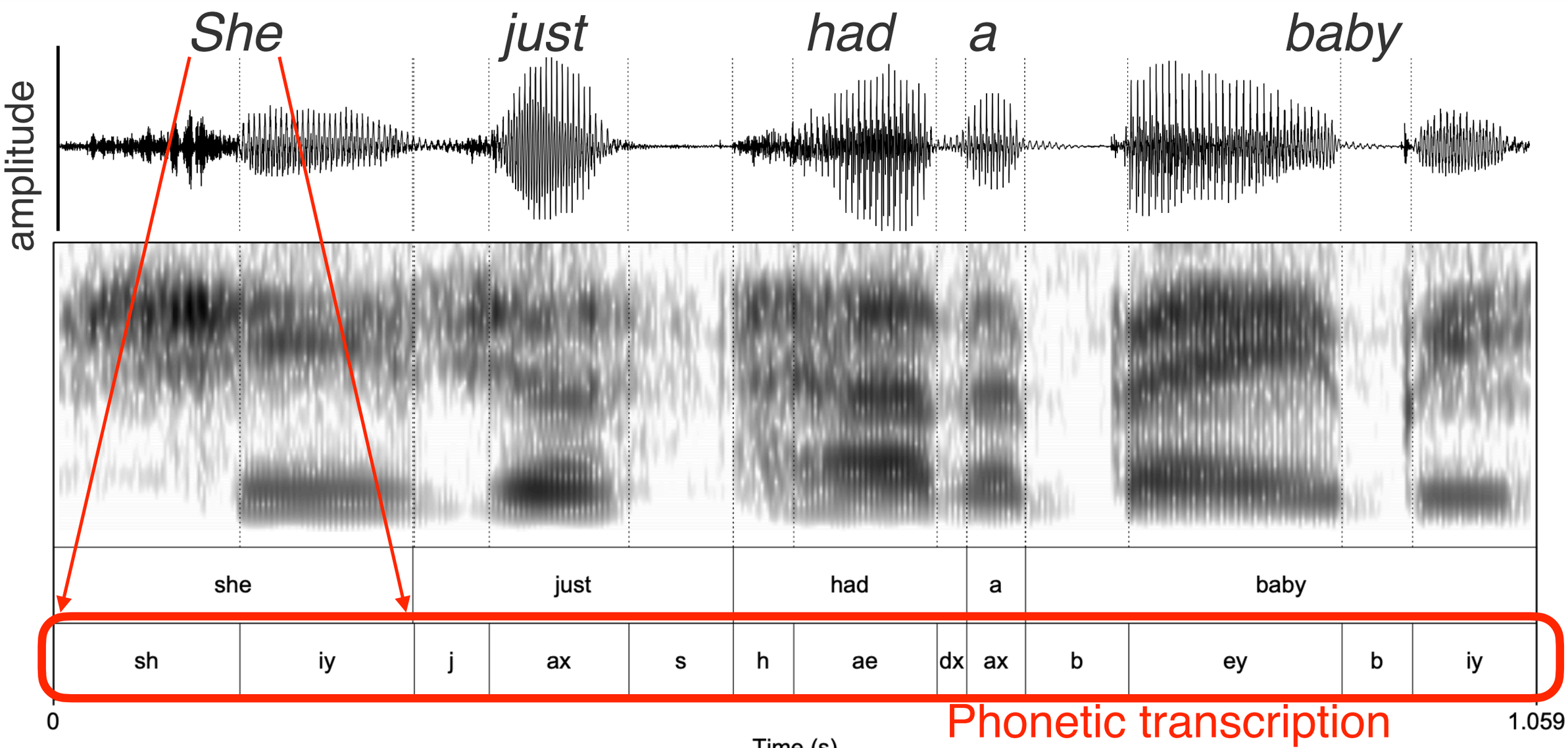
Acoustic Phonetics

Acoustic Phonetics



- Spectrogram reveals some segmental structure with distinct properties
- These are Phonemes — perceptually distinct speech sounds

Acoustic Phonetics



- Spectrogram reveals some segmental structure with distinct properties

- These are phonemes — perceptually distinct speech sounds

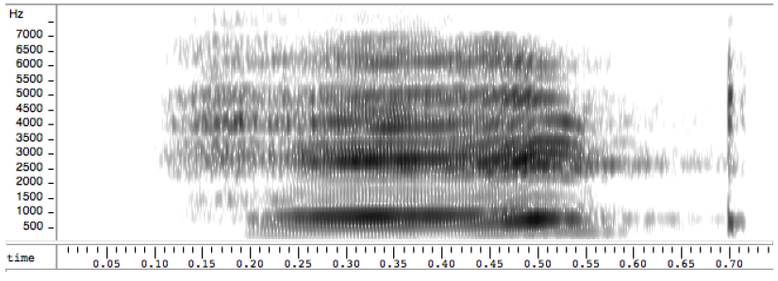
But, Speech is not just the phonemes

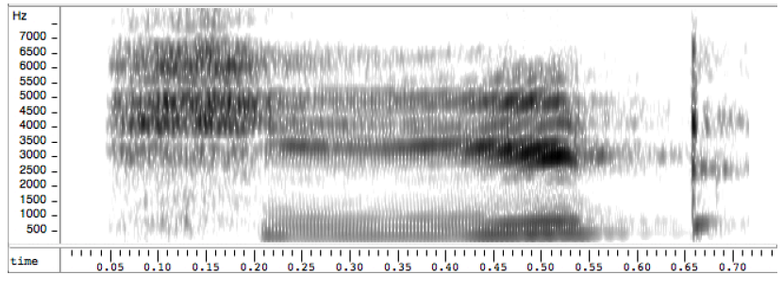
This is a short example

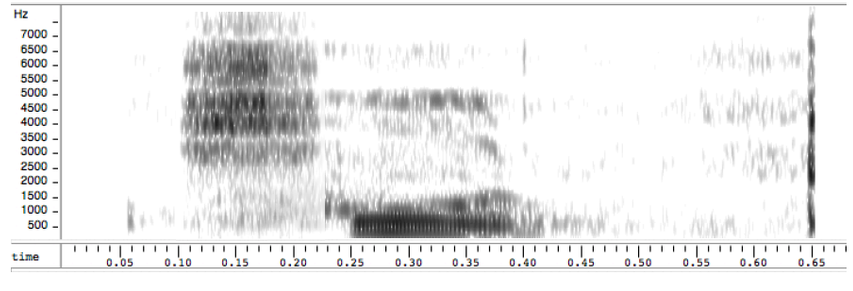
Coarticulation: speech is not segmental or linear

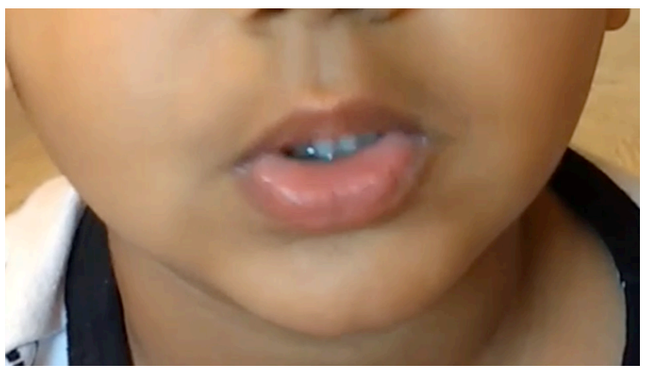
Head **Heed** **Hood**

/h eh d/ **/h iy d/** **/h uh d/**





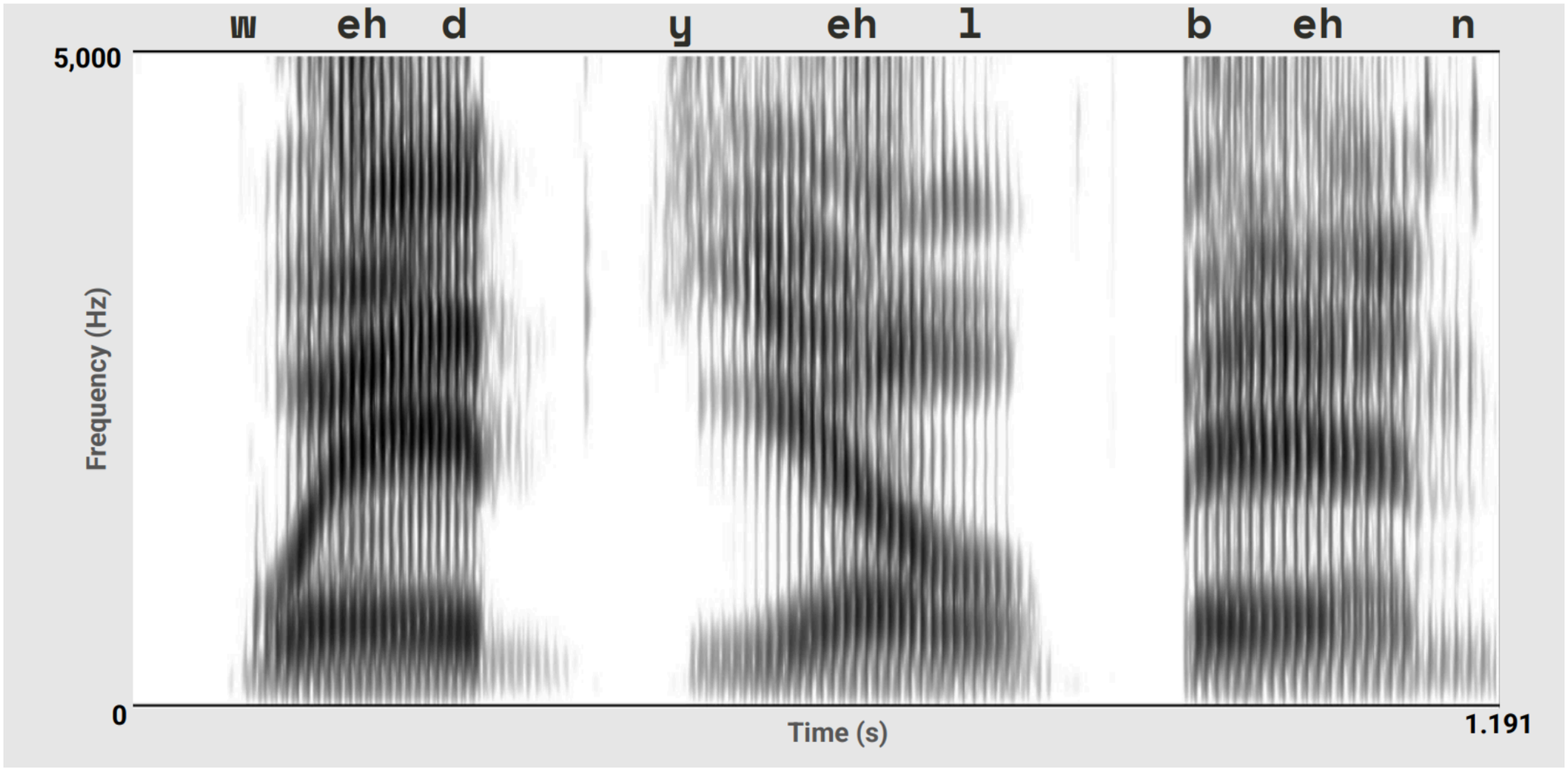




Co-articulation: An elegant phonological-articulatory transformation to facilitate rapid communication

Coarticulation = Bad for pattern matching

3 different “eh” sounds



International Phonetic Alphabet (IPA)

- Phoneticians compiled a common set of sounds used to codify different speech sounds (across languages)
- English spelling is not phonetic
  - It has about 40 distinct phonemes represented by 26 graphemes
  - About 16 vowel sounds
  - About 24 consonant sounds

THE INTERNATIONAL PHONETIC ALPHABET (revised to 2020)

CONSONANTS (PULMONIC) — © 2020 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.

CONSONANTS (NON-PULMONIC)

Clicks	Voiced implosives	Ejectives
ʘ Bilabial	ɓ Bilabial	’ Examples:
ǀ Dental	ɗ Dental/alveolar	pʼ Bilabial
ǃ (Post)alveolar	ʄ Palatal	tʼ Dental/alveolar
ǂ Palatoalveolar	ɠ Velar	kʼ Velar
ǁ Alveolar lateral	ʛ Uvular	sʼ Alveolar fricative

OTHER SYMBOLS

- ʍ Voiceless labial-velar fricative
- w Voiced labial-velar approximant
- ɥ Voiced labial-palatal approximant
- ʜ Voiceless epiglottal fricative
- ʢ Voiced epiglottal fricative
- ʡ Epiglottal plosive
- ɕ ʑ Alveolo-palatal fricatives
- ɺ Voiced alveolar lateral flap
- ɧ Simultaneous ʃ and x

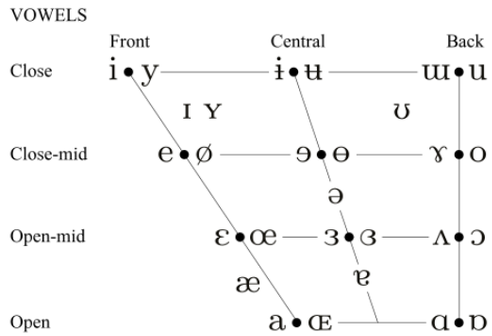
Affricates and double articulations can be represented by two symbols joined by a tie bar if necessary. t͡s k͡p

DIACRITICS

◌̥ Voiceless	n̥ d̥	◌̤ Breathy voiced	b̤ a̤	◌̪ Dental	t̪ d̪
◌̬ Voiced	s̬ t̬	◌̰ Creaky voiced	b̰ a̰	◌̺ Apical	t̺ d̺
◌ʰ Aspirated	tʰ dʰ	◌̼ Linguolabial	t̼ d̼	◌̻ Laminal	t̻ d̻
◌̹ More rounded	ɔ̹	◌ʷ Labialized	tʷ dʷ	◌̃ Nasalized	ẽ
◌̜ Less rounded	ɔ̜	◌ʲ Palatalized	tʲ dʲ	◌ⁿ Nasal release	dⁿ
◌̟ Advanced	u̟	◌ˠ Velarized	tˠ dˠ	◌ˡ Lateral release	dˡ
◌̠ Retracted	e̠	◌ˤ Pharyngealized	tˤ dˤ	◌̚ No audible release	d̚
◌̈ Centralized	ë	~ Velarized or pharyngealized	ɫ		
◌̽ Mid-centralized	e̽	◌̝ Raised	e̝ (ɹ̝ = voiced alveolar fricative)		
◌̩ Syllabic	n̩	◌̞ Lowered	e̞ (β̞ = voiced bilabial approximant)		
◌̯ Non-syllabic	e̯	◌̘ Advanced Tongue Root	e̘		
◌˞ Rhoticity	ɚ a˞	◌̙ Retracted Tongue Root	e̙		

Some diacritics may be placed above a symbol with a descender, e.g. n̠̊ʲ

VOWELS



Where symbols appear in pairs, the one to the right represents a rounded vowel.

SUPRASEGMENTALS

ˈ Primary stress	ˌfoʊnəˈtɪʃən
ˌ Secondary stress	
ː Long	eː
ˑ Half-long	eˑ
◌̆ Extra-short	ĕ
‖ Minor (foot) group	
‖ Major (intonation) group	
. Syllable break	ɹi.ækt
‿ Linking (absence of a break)	

TONES AND WORD ACCENTS

LEVEL		CONTOUR	
e̋ or ˥	Extra high	e᷄ or ˦˥	Rising
é ˦	High	ê ˥˦	Falling
ē ˧	Mid	e᷅ ˧˥	High rising
è ˨	Low	e᷄ ˨˧	Low rising
ȅ ˩	Extra low	e᷈ ˩˨	Rising-falling
↓	Downstep	↗	Global rise
↑	Upstep	↘	Global fall

International Phonetic Alphabet (IPA)

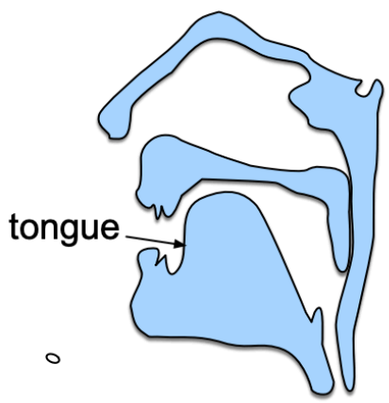
- Vowels are characterized by jaw position and tongue shape
- Some vowels also use lips (eg. sound *uw* in *cool*)

Tongue frontness

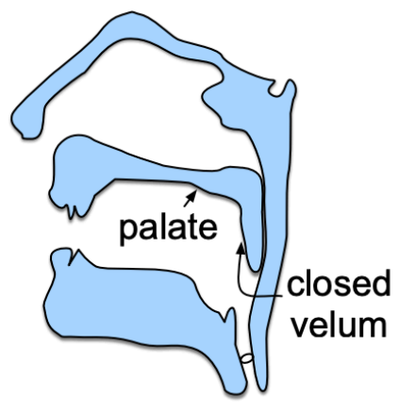
VOWELS

Jaw aperture	Front	Central	Back
Close	i • y	ɨ • ʉ	ɯ • u
	ɪ ʏ		ʊ
Close-mid	e • ø	ɘ • ɵ	ɤ • o
		ə	
Open-mid	ɛ • œ	ɜ • ɞ	ʌ • ɔ
	æ	ɐ	
Open	a • Œ	ɑ • ɒ	

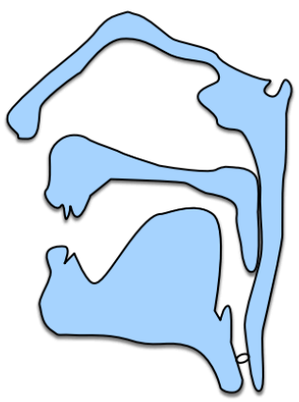
Where symbols appear in pairs, the one to the right represents a rounded vowel.



beet [iy]



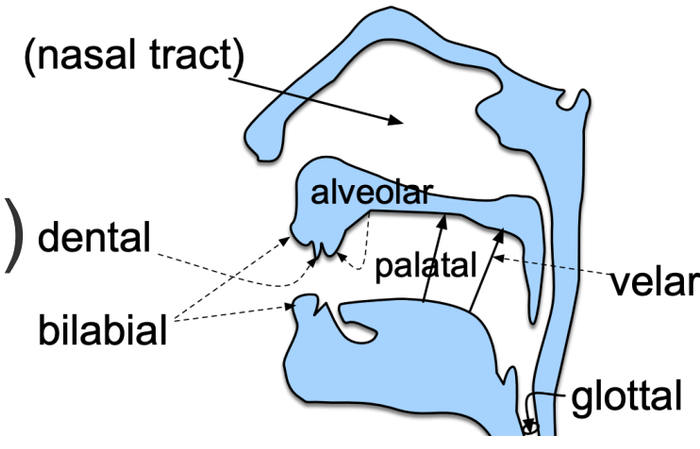
bat [ae]



boot [uw]

International Phonetic Alphabet (IPA)

- Consonants are characterized by place and manner of articulation
  - /p/ is caused by constriction at lips (labial)
  - /p/ is caused by sudden release of air (plosive)



Place of articulation

Manner of articulation

CONSONANTS (PULMONIC) ©⓪◎ 2020 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.

What is phone and phoneme???
GO TO: “g oʊ t u” or “G OW T UW”

- Phone: g oʊ t u
  - Devised by International Phonetic Association
  - Physical categorization of speech sound
  - Not applicable to all languages, needs special characters, too many variations

- Phoneme: one of the units that distinguish one word from another in a **particular language**
  - /r/ and /l/ are degenerated in some languages (e.g., “rice” and “lice” sounds same for me!). Then, we don’t have to distinguish them.
  - ARPAbet: `G OW T UW`
    - Proposed by ARPA for the development of speech recognition of only “American English”
    - Represented by ASCII characters

Pronunciation dictionary

- CMU dictionary
  - http://www.speech.cs.cmu.edu/cgi-bin/cmudict

”I want to go to campus”

→AY W AA N T T UW G OW T UW K AE M P AH S

- Powerful, but limited
- Out of vocabulary issue, especially new word

→ Grapheme2Phoneme mapping based on machine learning

From letters to sounds

- Pronunciation dictionaries (often made by linguists) give the syllables and phonemes within each word in vocabulary
  - CMU Phonetic Dictionary gives the syllabic and phonetic spellings for >110K words in English
  - ML based phonetizers are built on such phonetic dictionaries

Graphemes	She	just	had	a	baby
IPA	ʃiː	ʤʌst	hæd	ə	ˈbeɪbi
Arpabet	sh iy	jh ah s t	h ae d	ah	b ey b iy

Arpabet is an ASCII friendly representation of IPA

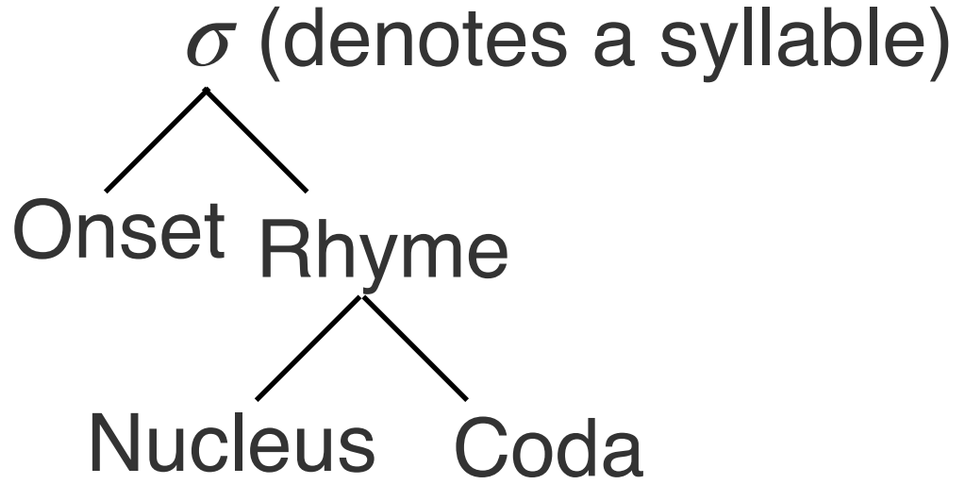
Phonology

- Phonology are the grammatical rules that phonemes of a language follow
- Lexical Phonology: Study of rules that govern the organization of sounds in a language (Phonemes —> Syllables —> Words)
  - Lexical Stress (project (noun) vs project (verb))
  - Allophones
    - (*r* and *l* in Japanese; *p* and *b* in Arabic; *t* and *k* in Hawaiian)
  - Phonological changes in continuous speech
    - Westside vs Westend
- Intonational Phonology: Study of the Fundamental Frequency (F0) in relation to the intended meaning of an utterance
  - *I never said she stole my money* (Emphasize each word for different meanings)

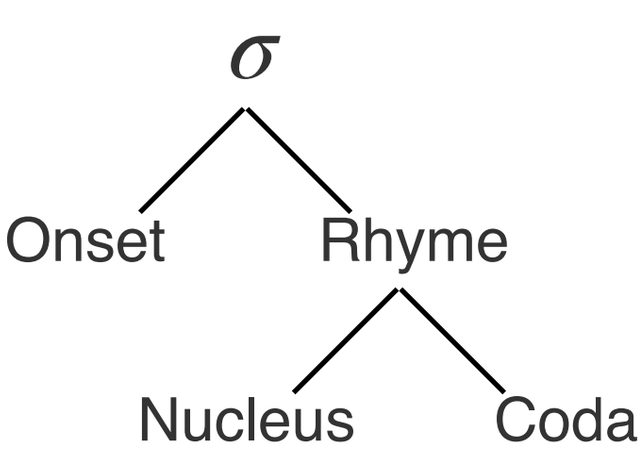
Lexical Phonology

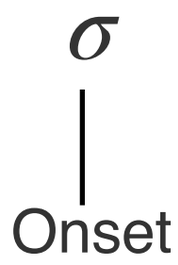
- Phonology is the study of rules that govern the organization of sounds in a language (Phonemes —> Syllables —> Words)

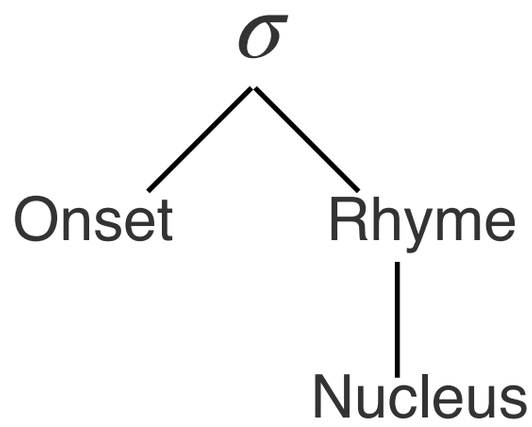
- Syllabic constituency

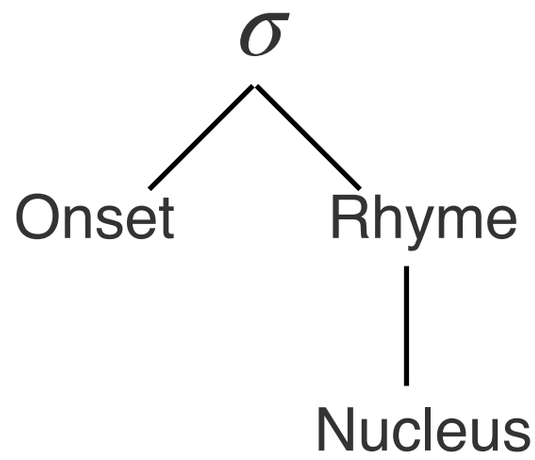


- How is "*had a baby*" composed?









Speech in the Wild

- Audio is neither clean nor just restricted to spoken language

- What technologies fall under spoken language research ?

- Speech recognition
- Speech synthesis
- Voice conversion
- Speaker recognition
- Language recognition
- Speech emotion recognition
- Speaker diarization
- Speech coding
- Speech perception

- Speech enhancement
- Microphone array processing
- Audio event classification and detection
- Speech separation
- Spoken language understanding
- Spoken dialogue systems
- Speech translation
- Multimodal processing

Speech in the Wild

- Audio is neither clean nor just restricted to spoken language

- What technologies fall under spoken language research ?

- Speech recognition
- Speech synthesis
- Voice conversion
- Speaker recognition
- Language recognition
- Speech emotion recognition
- Speaker diarization
- Speech coding
- Speech perception

- Speech enhancement
- Microphone array processing
- Audio event classification and detection
- Speech separation
- Spoken language understanding
- Spoken dialogue systems
- Speech translation
- Multimodal processing

Speech Recognition

aka
ASR: Automatic Speech Recognition

Speech Perception: Transforming Sound to Meaning

sentence

Have you got enough blankets?

meaning

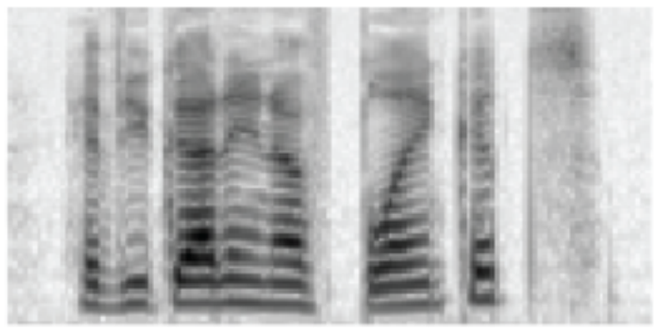
word

hɪv|y|gɑr|ɪf|ʌf|blæŋ|kɬts

syllable

hɪvygɑrɪfʌfblæŋkɬts

phoneme



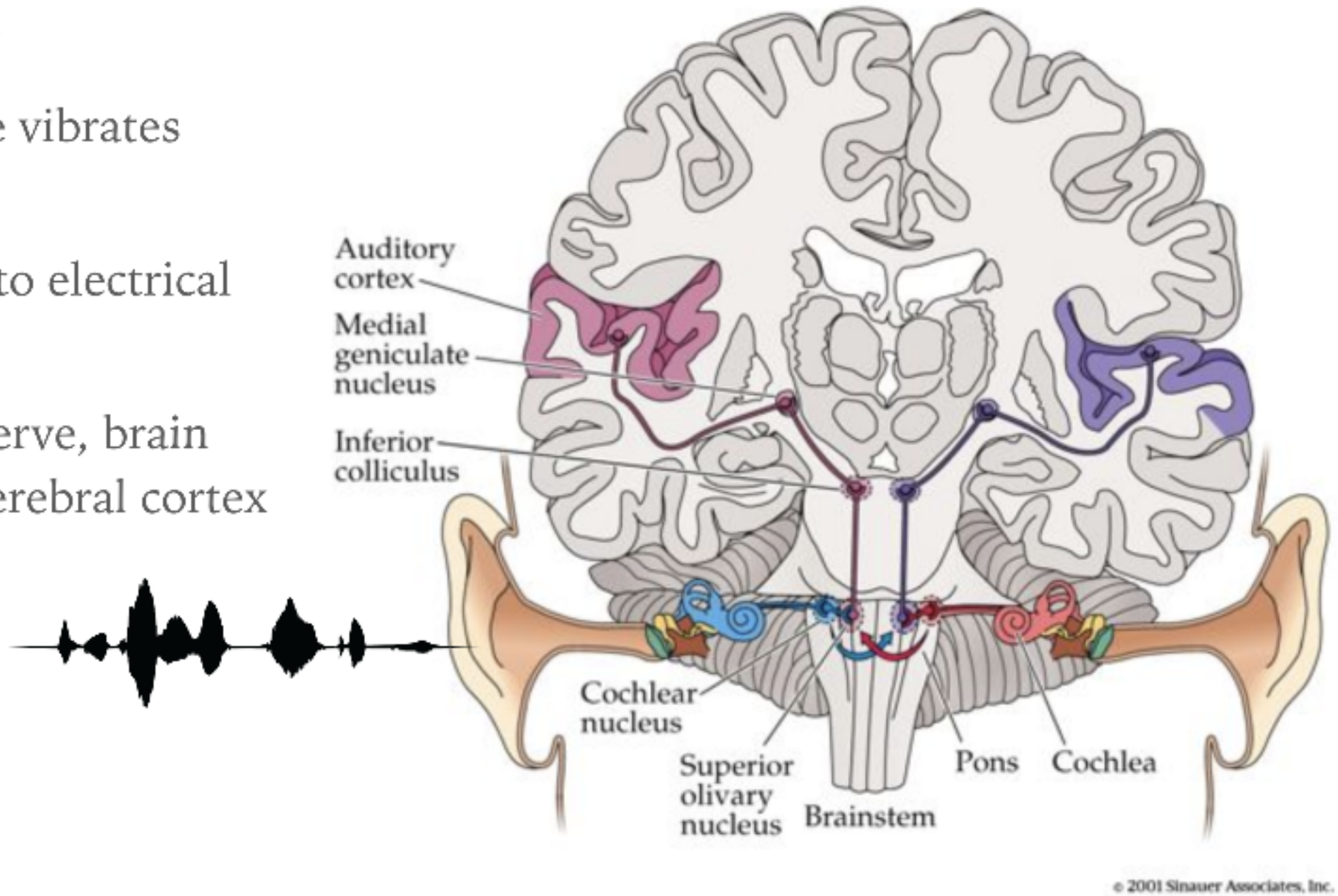
spectrotemporal

(frequency decomposition)

acoustic vibration

Auditory Pathway

- Sounds reach the ear
- Sound pressure wave vibrates tympanic membrane
- Vibration converted to electrical signal
  - sent to auditory nerve, brain stem, thalamus, cerebral cortex



Computations along the auditory pathway

Anatomy	Proposed Computation
Cochlea	Frequency decomposition of sounds (FFT)
Brain stem	Sound localization (azimuth and elevation)
Cerebral cortex	Acoustic to phonetic transformation, meaning of words (semantics), higher cognitive functions (decision making)

The ear

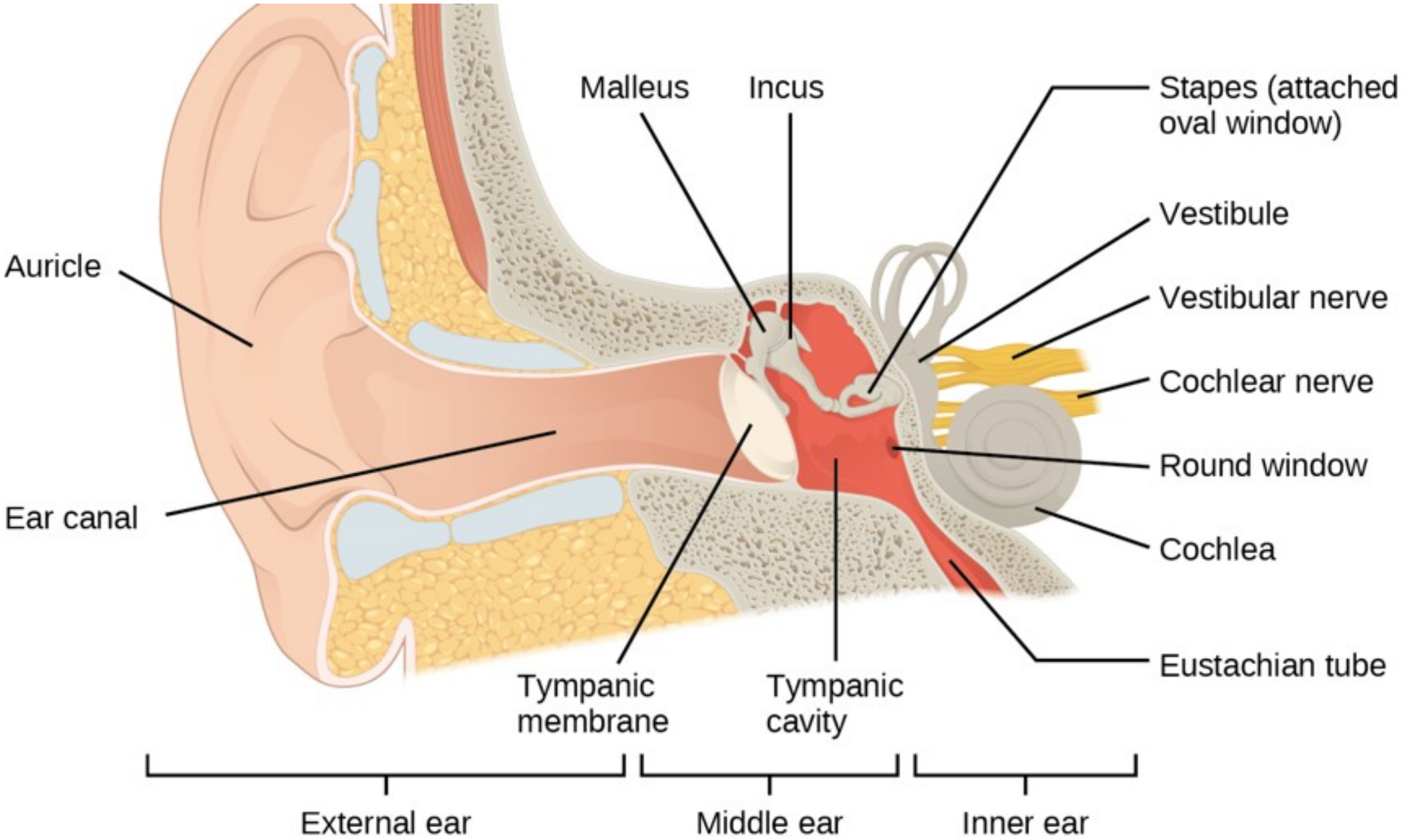
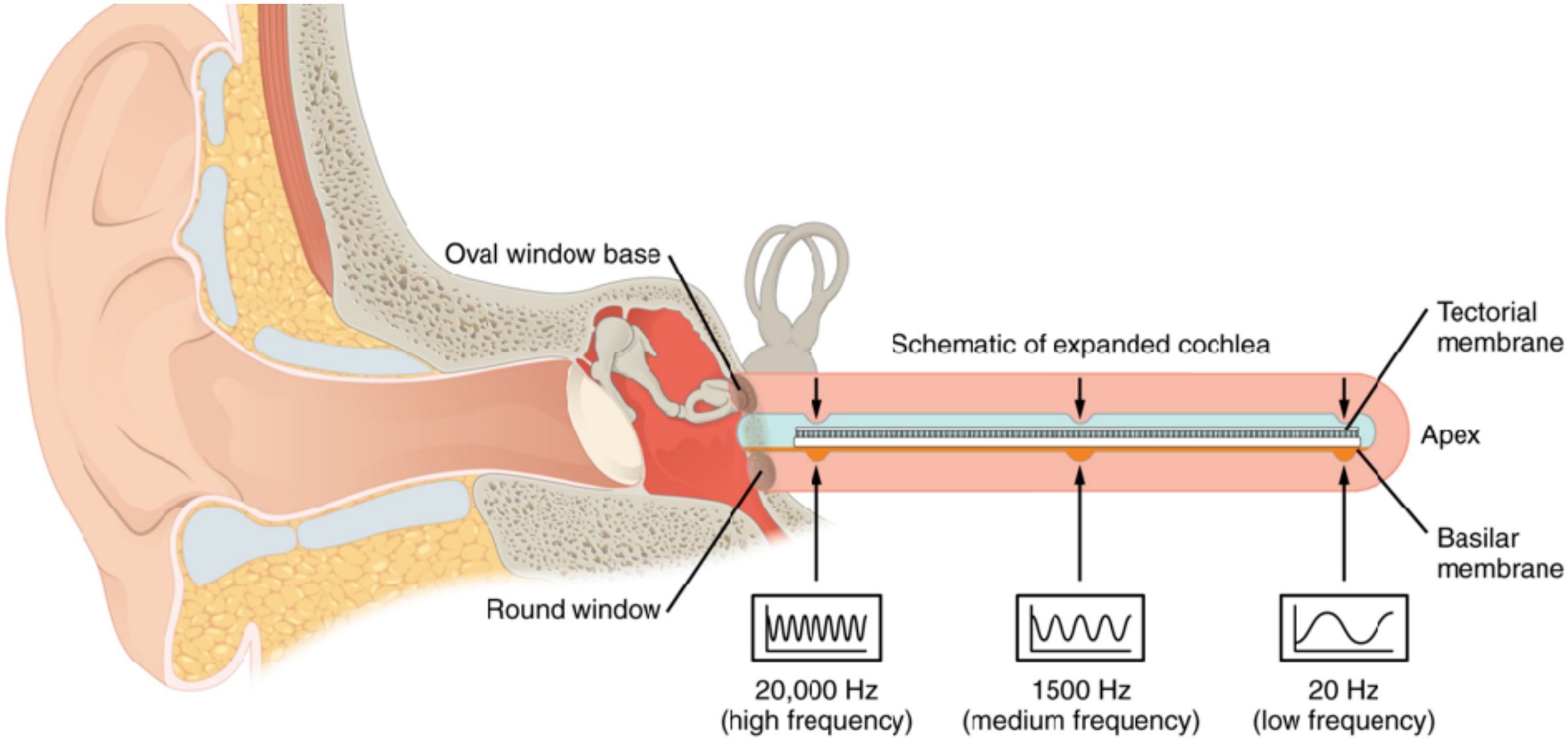


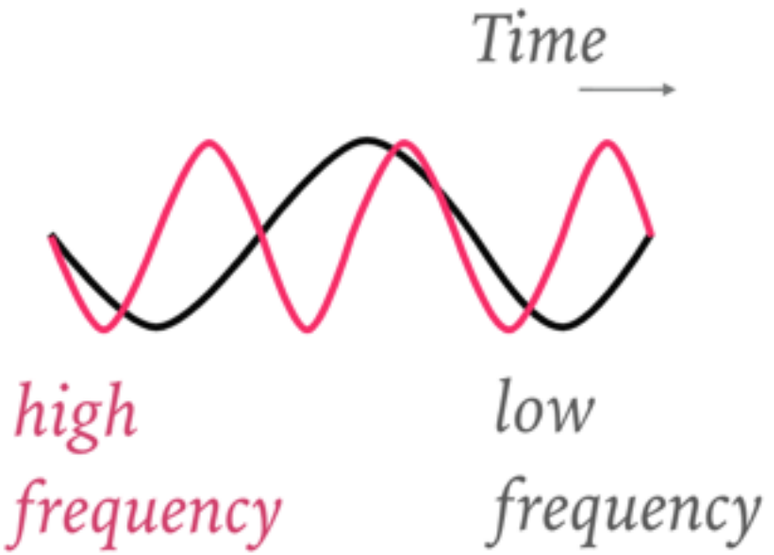
Image credit: OpenStax CC-BY-4.0 - Access for free at https://openstax.org/books/biology-2e/pages/1-introduction

FFT in the ear

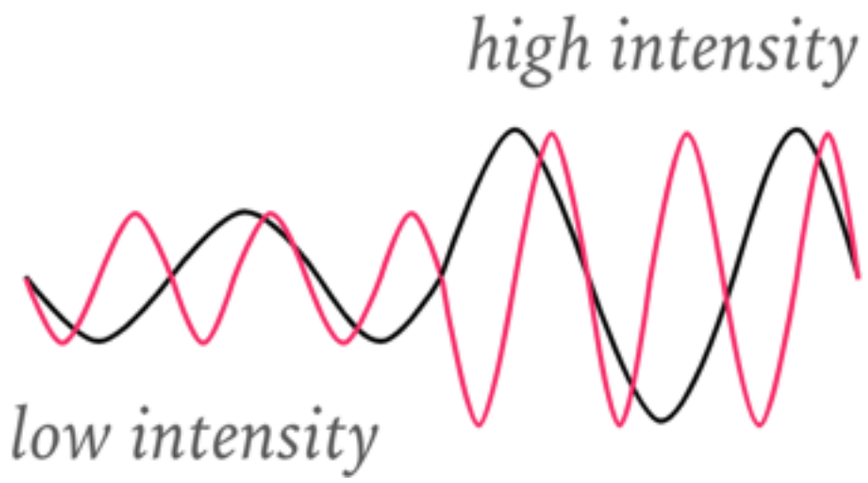


Important parameters of sound waves

- **Frequency**: # of waves that pass any point in one second. Measured in cycles/sec (Hz)
  - subjective correlate: pitch



- **Intensity/amplitude**: Magnitude of the movements produced (measured in dB)
  - subjective correlate: loudness

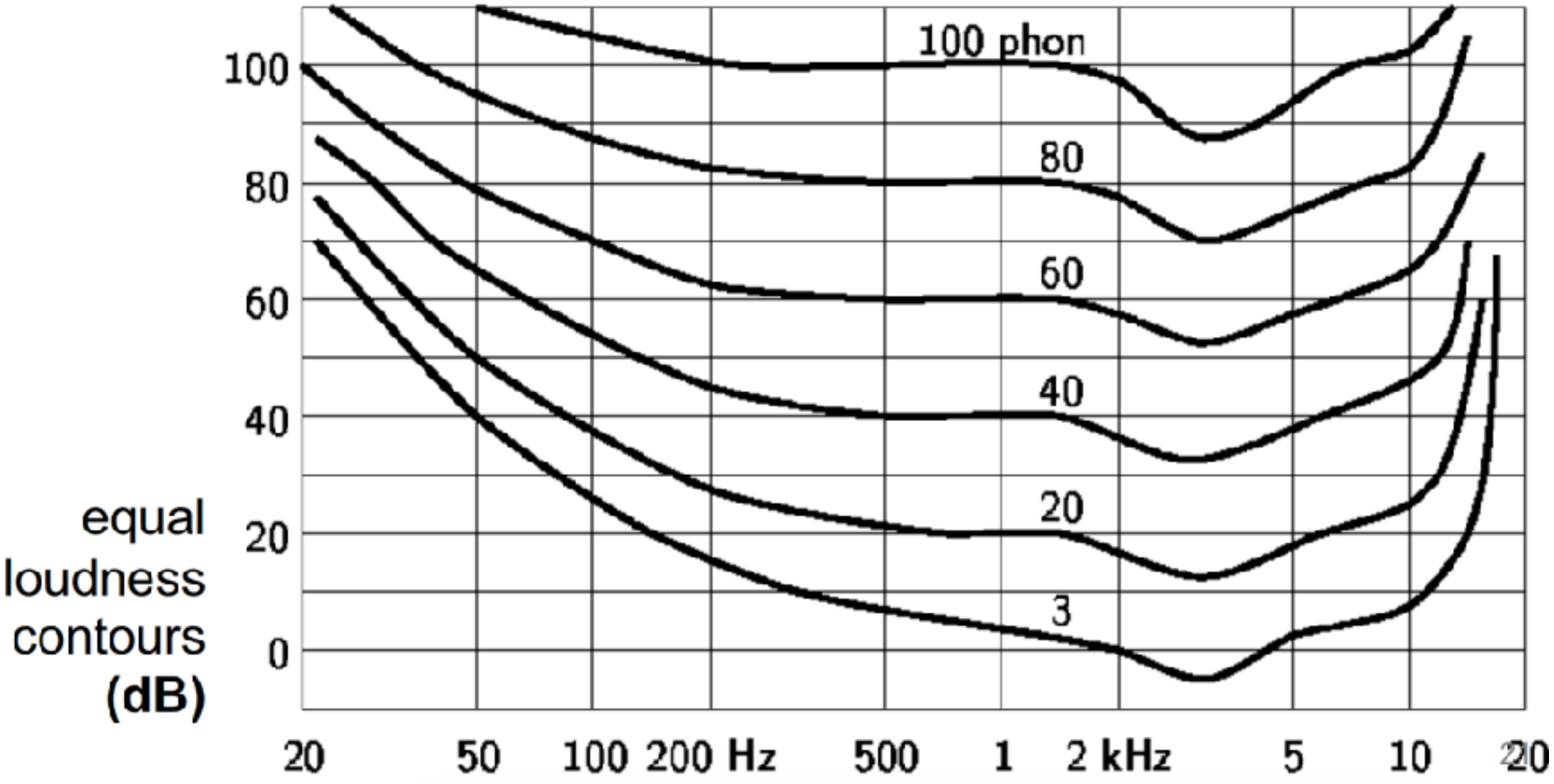


Perception of Sound

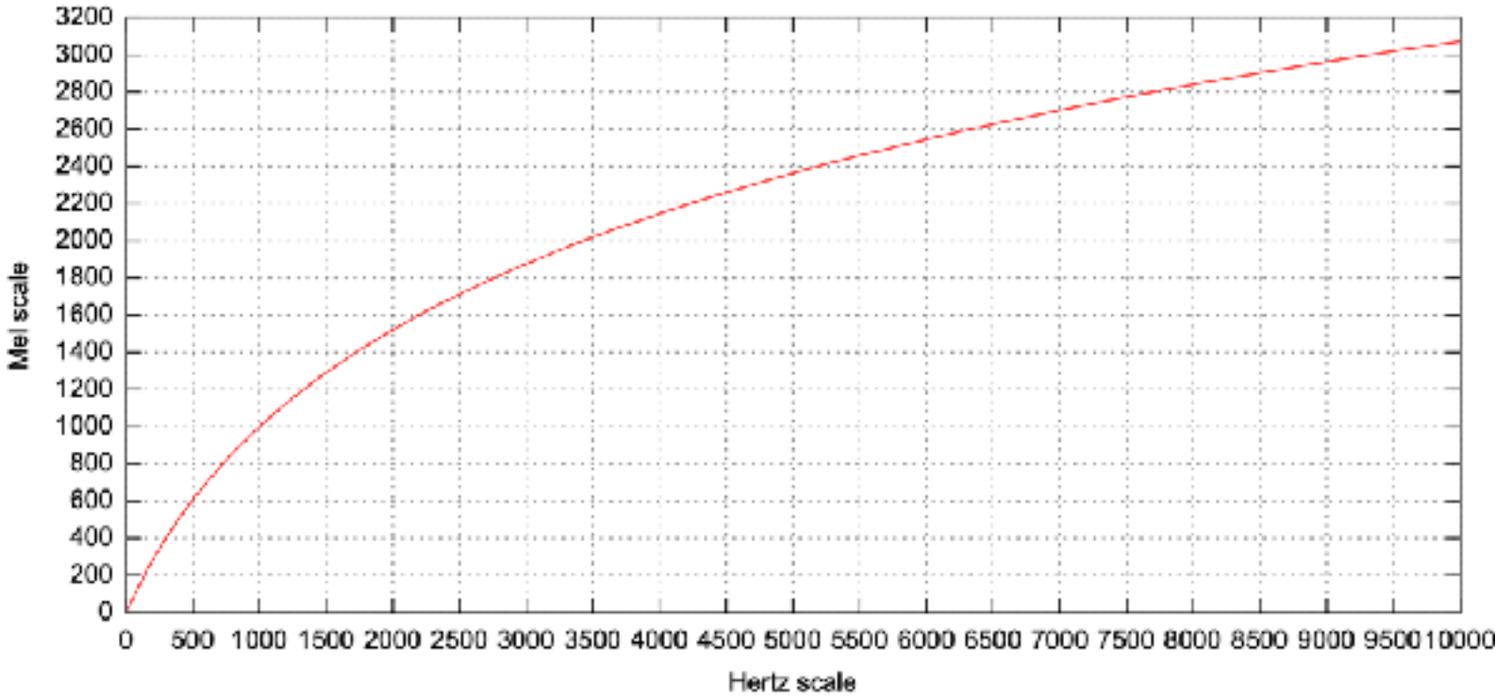
Physical Quantity	Perceptual Quality
Intensity	Loudness
Fundamental frequency (f_0)	Pitch
Spectral envelope (formants)	Timbre
Onset/offset time	Timing
Phase difference in the two ears	Location

Perception of Sound

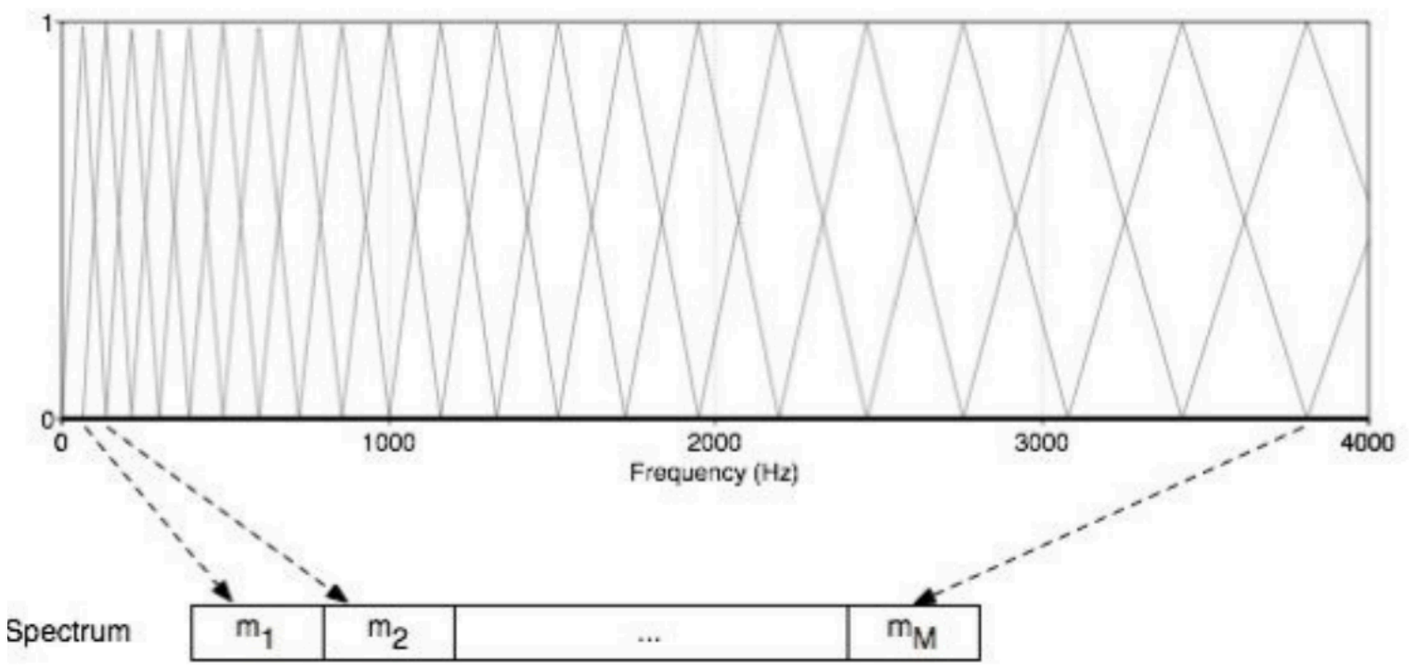
- One divergence between perceptual and physical properties is the **non-uniform equal loudness contours.** The ears are most sensitive to sounds with a frequency of 3 ~ 4 kHz.



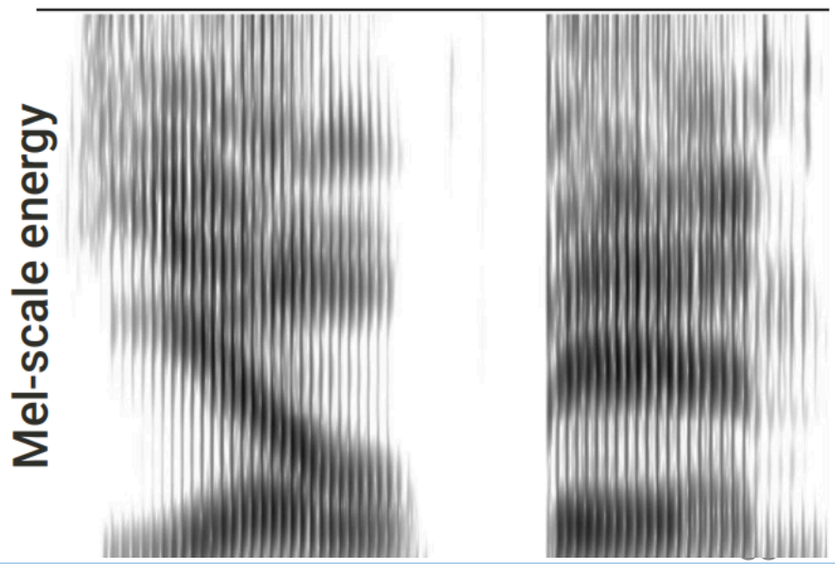
Mel scale: A Logarithmic filter bank for Perception



Mel Filter bank for MFCC

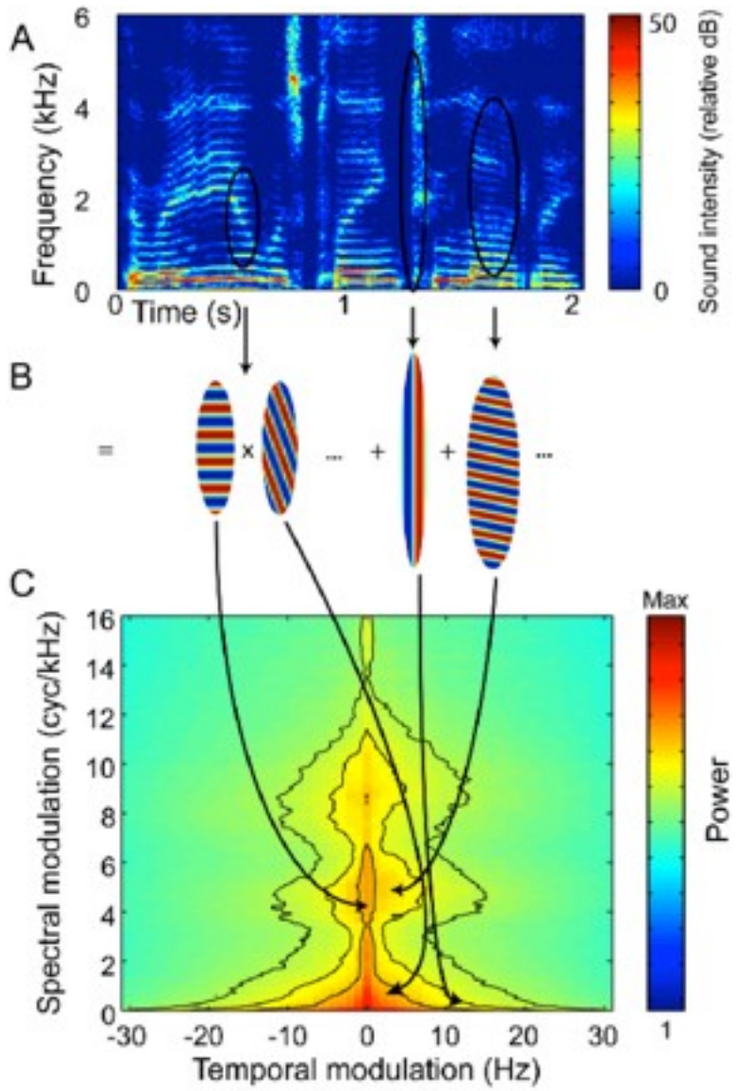


Mel Frequency Spectrogram



Speech is a small part of all possible sound

- Sounds can be decomposed into spectral and temporal modulations
- Natural stimuli (including speech) can be made up of many types
  - If high on one dimension, usually low on the other



Elliott & Theunissen PLoS Computational Biology 2011

Speech Recognition

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

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 (AAVE)	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

Figure 27.1 Rough Word Error Rates (WER = % of words misrecognized) reported around 2020 for ASR on various American English recognition tasks, and character error rates (CER) for two Chinese recognition tasks.

Conversational Speech

- Utterance without Context

- With context

HSR versus ASR

Deletions				Insertions			
SWB		CH		SWB		CH	
ASR	Human	ASR	Human	ASR	Human	ASR	Human
30: it	19: i	46: i	20: i	13: i	16: is	23: a	17: is
20: i	17: it	46: it	18: and	10: a	14: %hes	14: is	17: it
17: that	16: and	39: and	15: it	7: and	12: i	11: i	16: and
16: a	14: that	32: is	15: the	7: of	11: and	10: are	14: have
14: and	14: you	26: oh	14: is	6: you	9: it	10: you	13: a
14: oh	12: is	25: a	13: not	5: do	6: do	9: the	13: that
14: you	12: the	20: to	10: a	5: the	5: have	8: have	12: i
12: %bcack	11: a	19: that	10: in	5: yeah	5: yeah	8: that	11: %hes
12: the	10: of	19: the	10: that	4: air	5: you	7: and	10: not
11: to	9: have	18: %bcack	10: to	4: in	4: are	7: it	9: oh

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

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

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

(Saon et al, 2017)

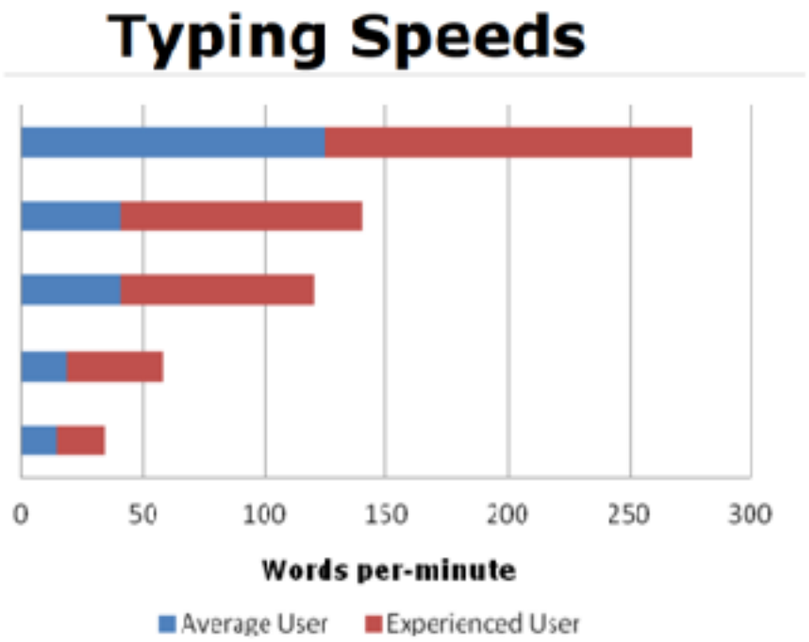
Why Study ASR ?

- In the last ~5 years
  - Dramatic reduction in LVCSR error rates (16% to <3%)
  - 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

Why Study ASR ?



With speech, interaction becomes independent of screen estate

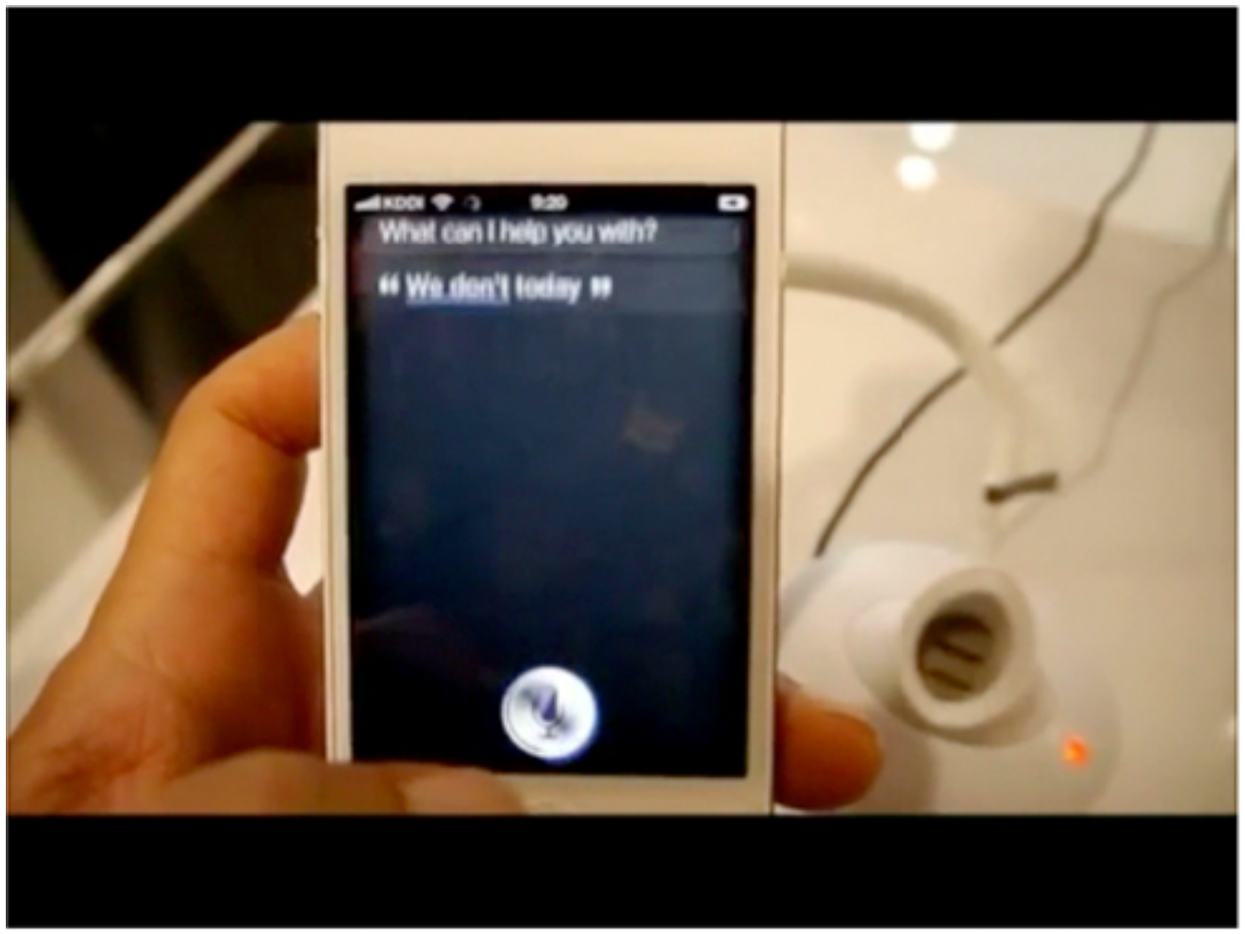
If accurately recognized, speech is three times faster than QWERTY
(Basapur et al. '07)

Only plausible interaction modality for 800 million non-literate users

Design of Classical ASR Systems

- 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

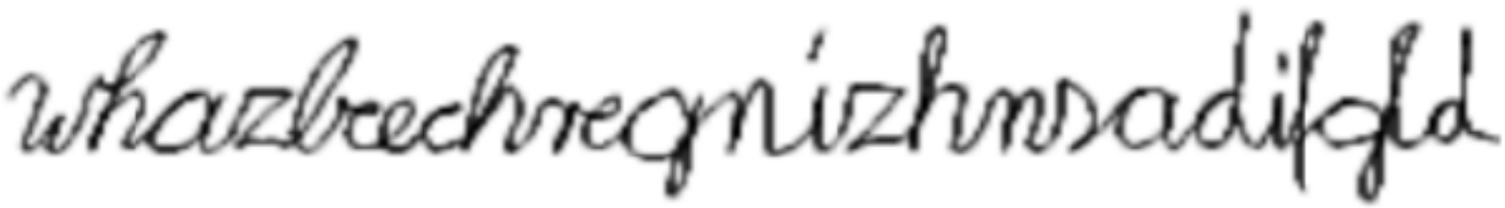
Current ASR Systems



- ✓ Adult Native English Speakers
- ✓ Clean Lab Environment
- ✗ Children
- ✗ Non-native Accent
- ✗ Noisy Environment
- ✗ Variable Speaking Rate

No “universal” speech recognizer – one must be developed and optimized for each use case

Why is ASR Hard ?

written text:	Why is speech Recognition so Difficult?
spontaneous:	why's speech recognition so difficult
continuous:	whysspeechrecognitionsodifficult
pronunciation:	whazbeechregnizhnsadifcld
acoustic variability:	
noise:	
Cocktail party-Effect:	

What are the factors that determine difficulty ?

COMPLEXITY

amount of data:	typically 32000 bytes per second (16khz)
class inventory:	50 phonemes, 5000 sounds, 100.000 words
combinatorial explosion:	exponential growth of possible sentences

SEGMENTATION

our perception:	Phones, syllables, words, sentences
actually there are:	no boundary markers, continuous flow of samples

VARIABILITY

speaker:	anatomy of vocal tract, speed, loudness, acoustic stress, mood, dialect, speaking style, context
channel, environment:	noise, microphones, channel conditions

AMBIGUITY

Homophones:	two vs. too,
Word Boundaries:	interface vs. in her face,
Semantics:	He saw the Grand Canyon flying to New York,
Pragmatics:	Time flies like an arrow.

Why is HSR easy ?

"The main prerequisite of the uniquely human communication is that speaker and listener must have a common understanding that out of all possible sounds man can produce and hear, only a few have linguistic significance."

(Olli Aaltonen& Esa Uusipaikka: Why Speaking Is so Easy? – Because Talking Is Like Walking with a Mouth)

- Important feature of speech perception: we hear sounds either as speech or non-speech
- Once defined as speech, we hear a sequence of vowels and consonants not as buzzes and hisses, the segmentation into words happens on the fly
- Abstract away from sound variability - we use an enormous database of background knowledge: phonotactics, morphology, syntax, semantics, pragmatic knowledge

Robustness of HSR

"The main prerequisite of the uniquely human communication is that speaker and listener must have a common understanding that out of all possible sounds man can produce and hear, only a few have linguistic significance."

(Olli Aaltonen& Esa Uusipaikka: Why Speaking Is so Easy? – Because Talking Is Like Walking with a Mouth)

- Important feature of speech perception: we hear sounds either as speech or non-speech
- Once defined as speech, we hear a sequence of vowels and consonants not as buzzes and hisses, the segmentation into words happens on the fly
- Abstract away from sound variability - we use an enormous database of background knowledge: phonotactics, morphology, syntax, semantics, pragmatic knowledge

We also use multiple modalities - McGurk effect

Problems and Challenges

- Speech Recognition ("speech-to-text")
  - Finding Robust Acoustic Representations of Speech (how do things sound)
  - Dictionary Learning (how to decompose words into units)
  - Language Modeling (what is likely to be said)
  - Decoding (how to get an answer in finite time)
- Adaptation and robustness of models and techniques to changing conditions
  - Multi-modal and multi-task learning, audio-visual processing (deep learning)
- Language-universal (?) modeling
  - Can we port resources across languages, to enable processing of new, unwritten, low-resource languages?
  - How do Humans do it? Language acquisition?

Problems and Challenges

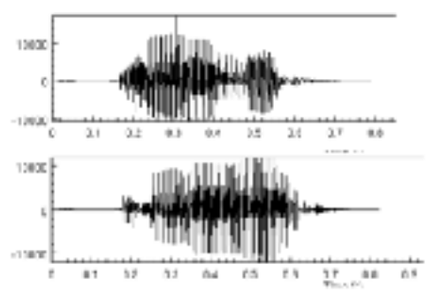
- Meta-data extraction (what is „not in text“)
  - Speaker identification (age, gender, ...)
  - Emotions, personalities, ...
  - Languages, dialects, ...
- Optimality Criterion?
  - Speech-to-text useful? No, unless dictation
  - Optimize directly “speech-to-meaning” or “speech-to-action”
  - All neural architectures
  - Jointly optimized speech-to-X systems
  - Speech synthesis

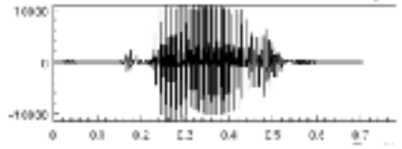
Template Matching

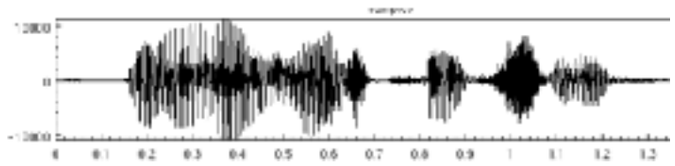
A Template based ASR System

- Some applications only have limited vocabulary — Voice Dialing System

- Library
  - Mom
  - Dad
  - Bob
  - Mario's Pizza



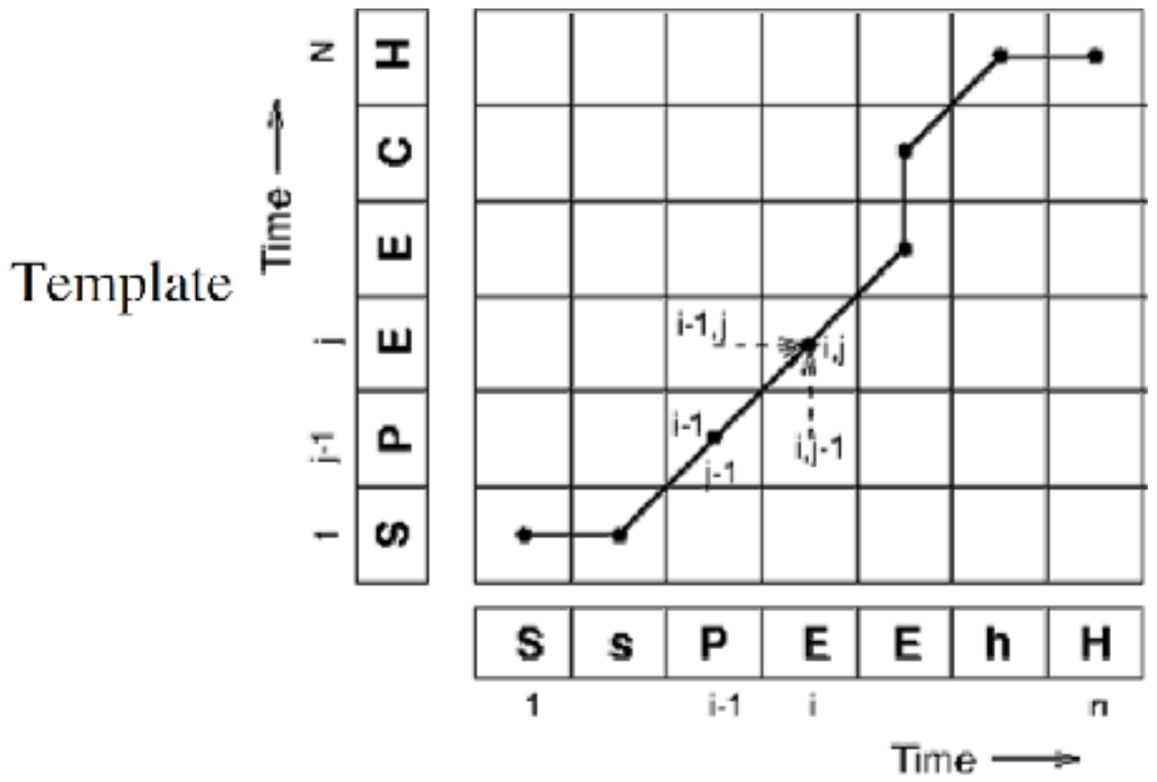




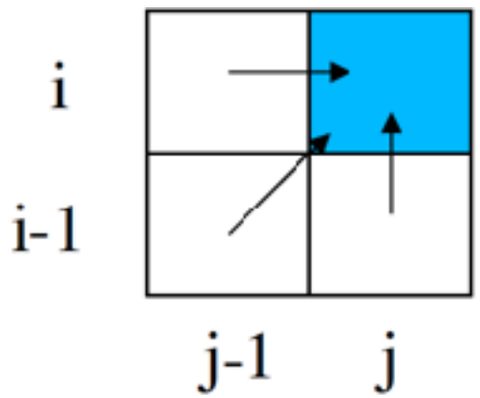
The matching problem

- We change durations
  - two utterances are never the same

Dynamic Time Warping



Dynamic Time Warping



- *For each square*
  - *Dist(template[i],sample[j]) +*
    smallest_of (Dist(template[i-1],sample[j])
    Dist(template[i],sample[j-1])
    Dist(template[i-1],sample[j-1])

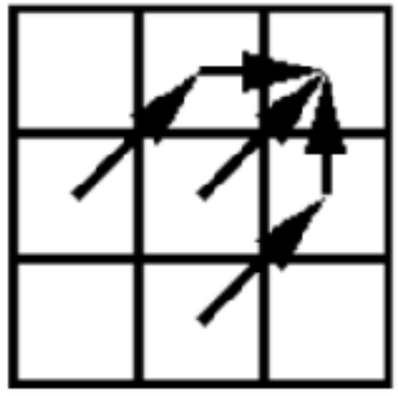
Remember which choice your took (count path)

Dynamic Time Warping Paths

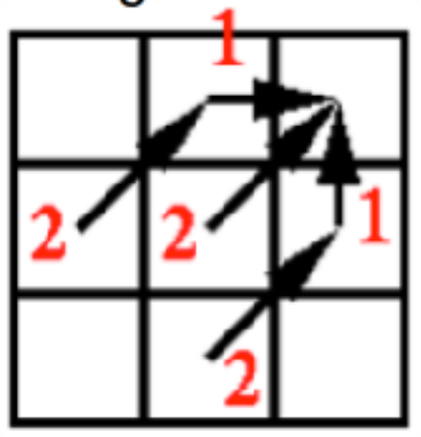
Many different warping steps are possible and have been used.

Examples:

Itakura



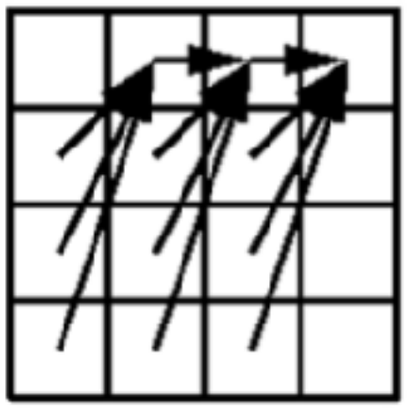
weighted



General rule is:

Cumulative cost of destination = best-of (cumulative cost of source + cost of step + distance in destination)

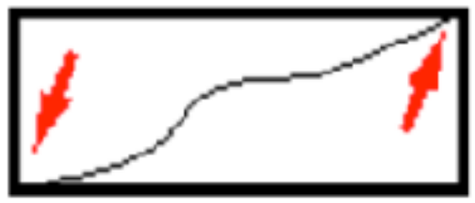
symmetric
(editing distance)



Bakis

Dynamic Time Warping Path Constraints

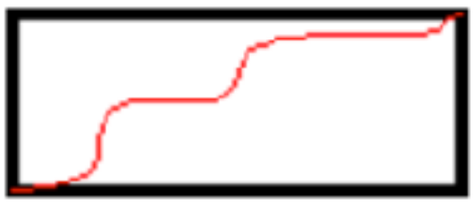
Endpoint constraints: we want the path to not skip a part at the beginning or end of the utterance



Monotonicity conditions: we can‘t go back in time (for any utterance)

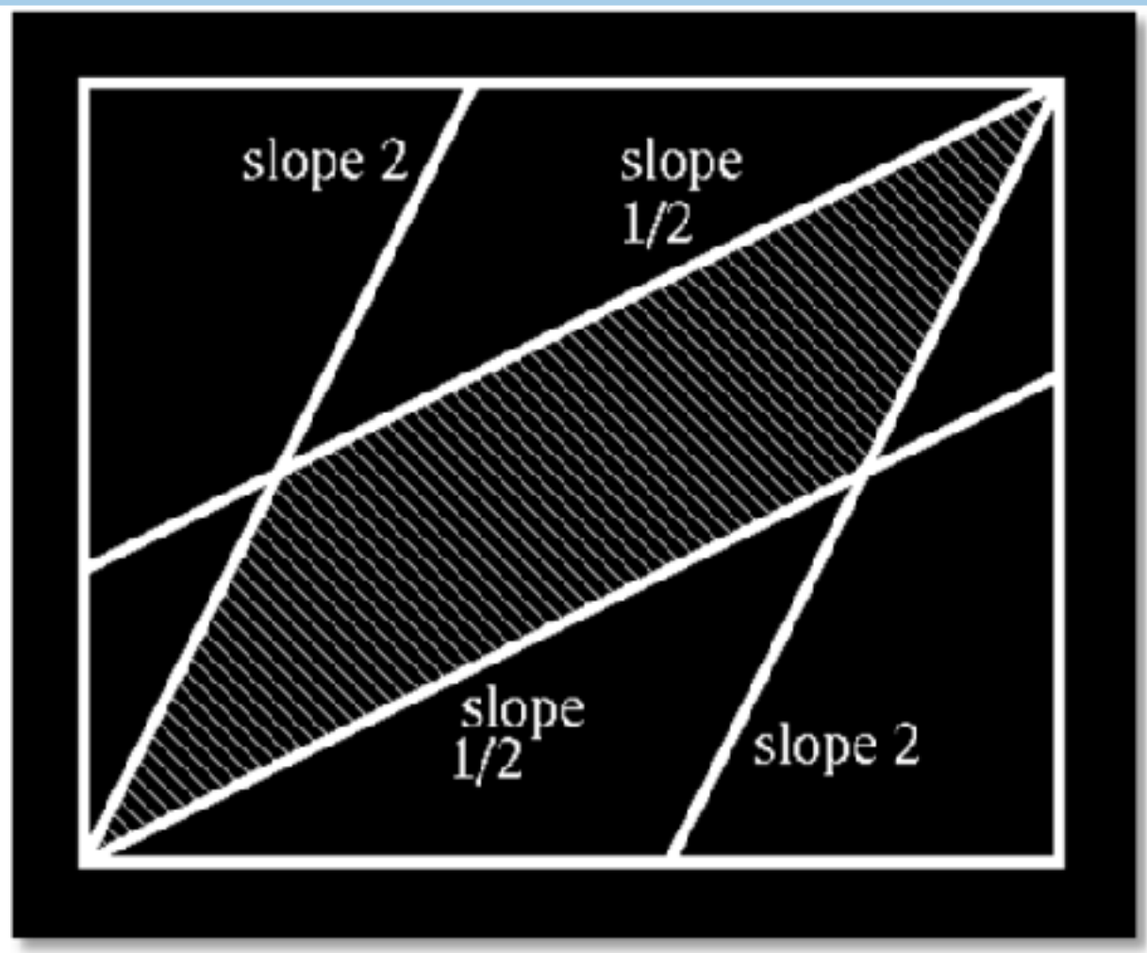
Local continuity: no jumps etc

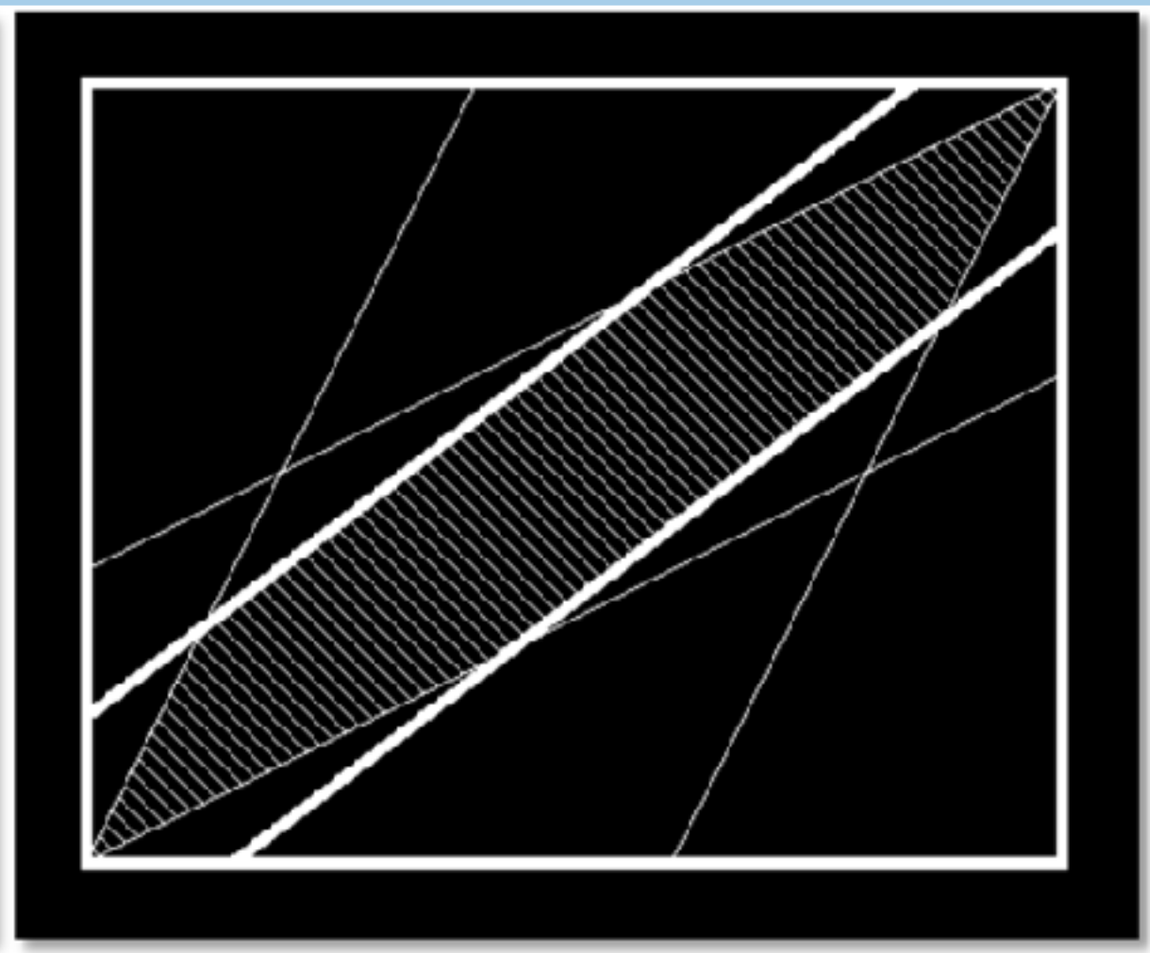
Global path constraints: path should be close to diagonal

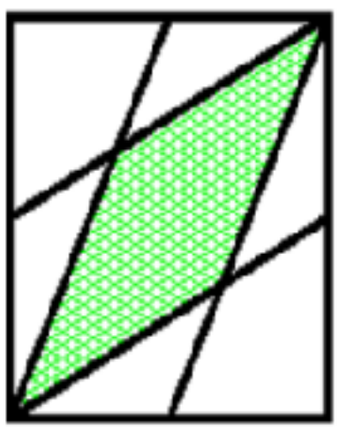


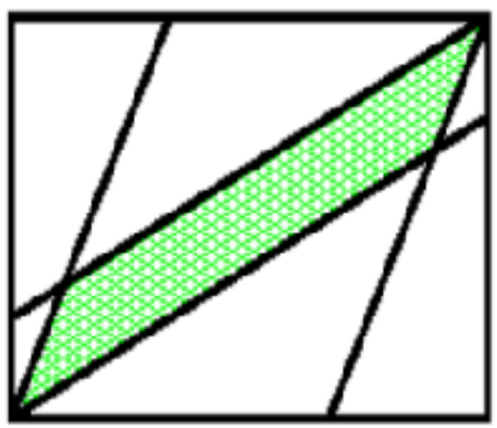
Slope weighting: we believe the DTW path sjould be somewhat „smooth“

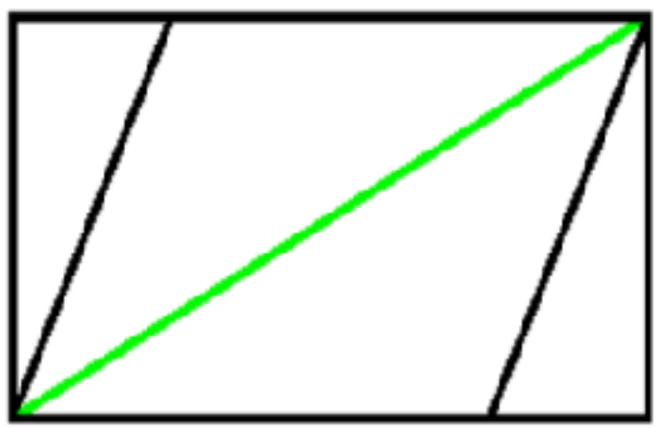
Dynamic Time Warping Path Constraints











← only one path

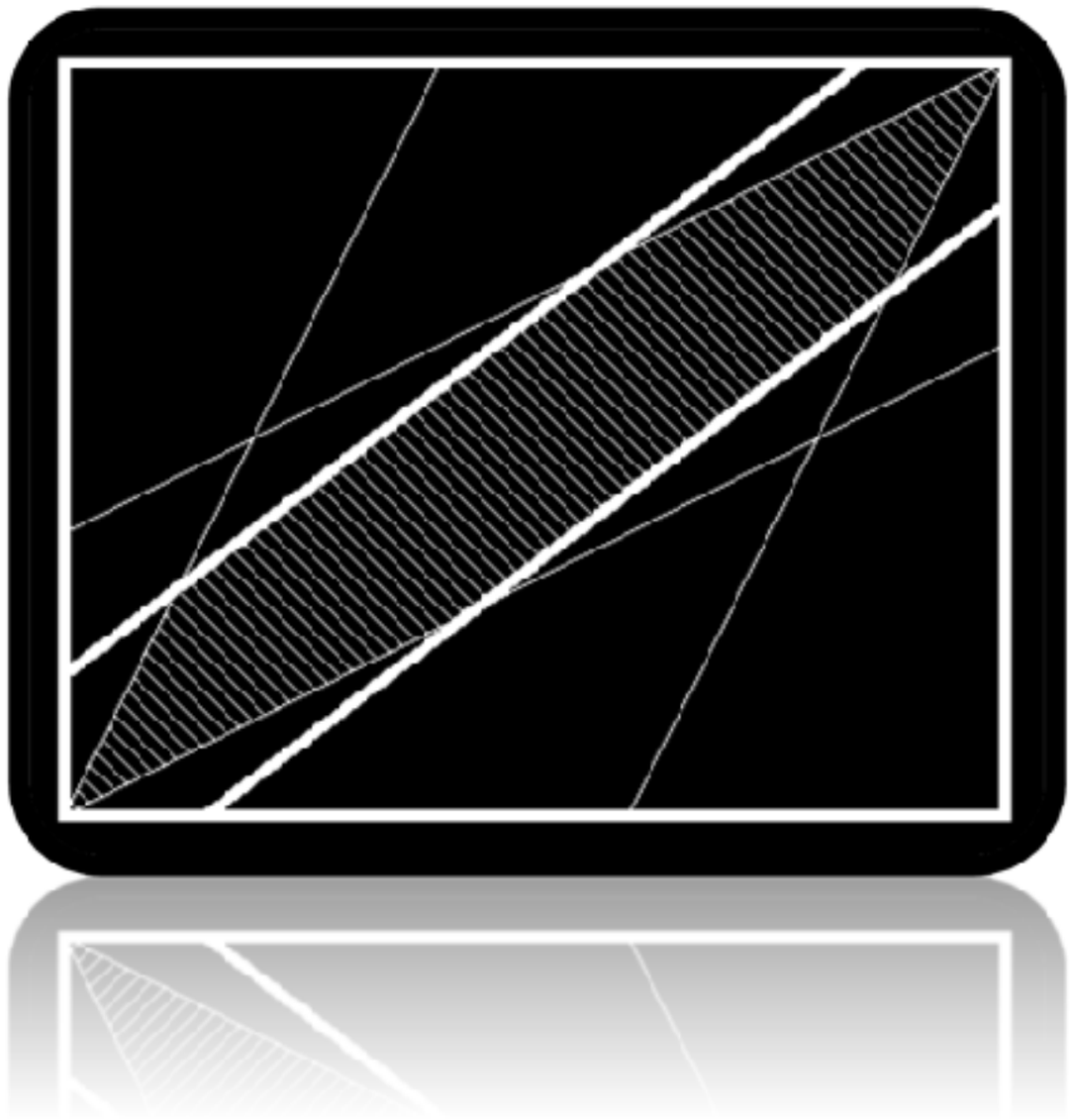
DTW Search Space

Already suggested: restrict search space by window around diagonal. Caveats:

- Silence period in one utterance can cause "edgy" path
- Search area becomes too restricted when utterance durations differ a lot

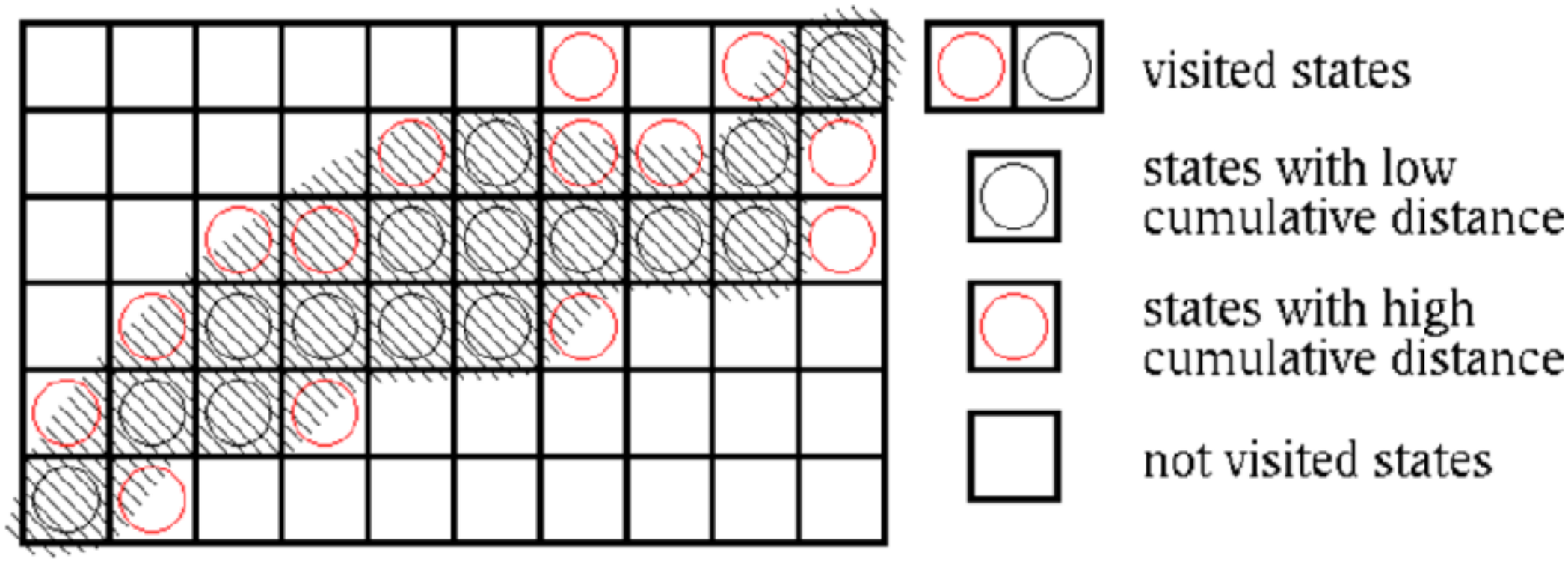
Other reason (besides global path constraints) for restricting search space:

- **Save time:** A window that has a constant width, reduces the search effort from $O(n^2)$ to $O(n)$
- To overcome caveats of "diagonal window" restriction, use: **beam search.**



DTW with Beam Search

Idea: do not consider steps to be possible out of states that have “too high” cumulative distances.

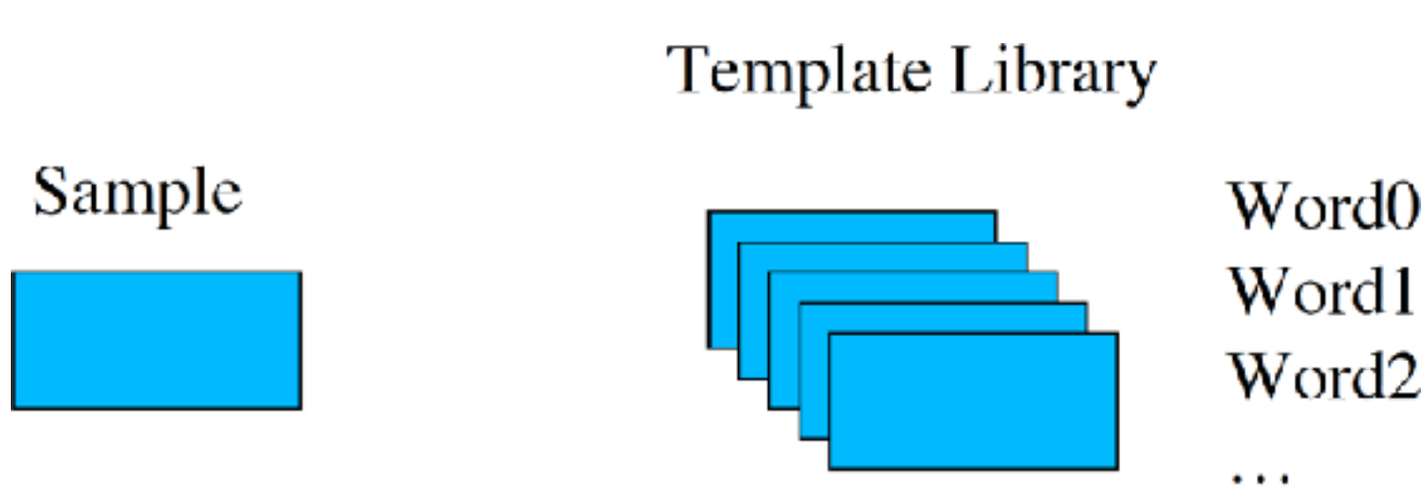


Approaches:

- “expand” only a fixed number of states per column of DTW matrix
- expand only states that have a cumulative distance less than a factor (the “beam”) times the best distance so far

DTW and Multiple Templates

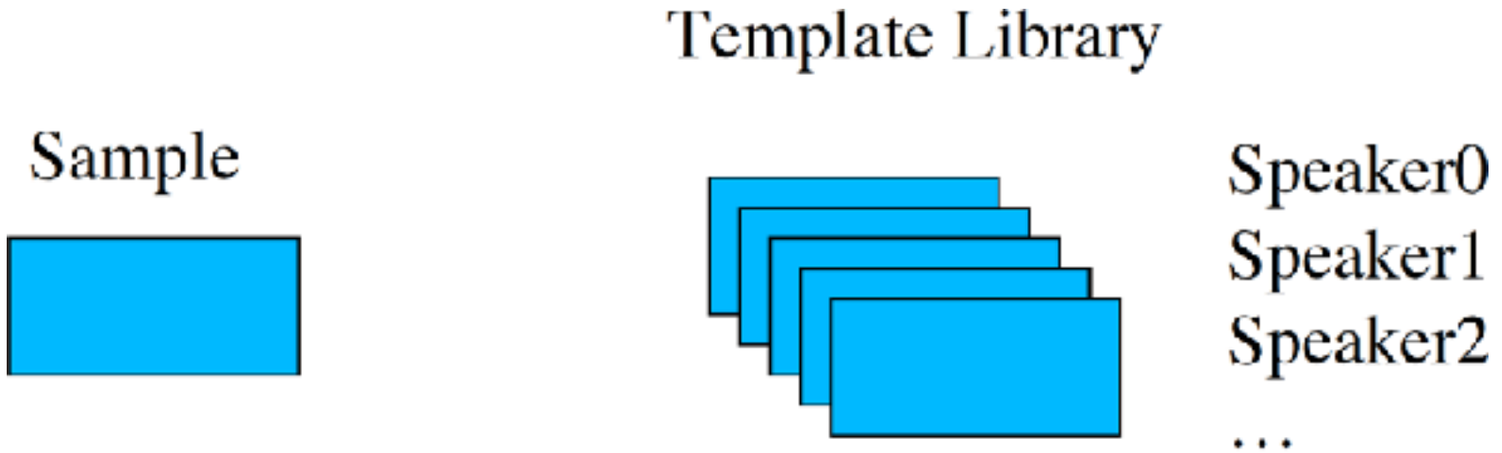
- *Compare against each*
- *Find closest*
- *Need to normalize scores*
  - *(divide by length of matches)*




```
For Word in Templates
        Score = dtw(Template[Word], Sample);
        if (Score < BestScore)
                BestWord = Word;
DoAction(Action[BestWord])
```

Also adapted for Speaker ID

- *Compare against each*
- *Find closest*
- *Need to normalize scores*
  - *(divide by length of matches)*




```
For Speaker in Templates
        Score = dtw(Template[Speaker], Sample);
        if (Score < BestScore)
                BestSpeaker = Speaker;
```

Template Matching

- *Advantages*
  - *Works well for small number of templates (<20)*
  - *Language independent*
  - *Speaker specific*
  - *Easy to train (end user controls it)*
- *Disadvantages*
  - *Limited number of templates*
  - *Speaker specific*
  - *Need actual training examples*

Computing distances

- Distance metric
  - Euclidean $\sqrt{\sum_{i=0}^{N}(T_i - S_i)^2}$
- But some distances are bigger than others
  - Silence is pretty similar
  - Fricatives are quite larger
    - A longer fricative might give large score
    - A longer vowel might give smaller score

Isolated Word Recognition using Template Matching

- For each word in the vocabulary, store at least one reference pattern
- When multiple reference patterns are available, either use all of them or compute an average
- During recognition
  - Record a spoken word
  - Perform pattern matching with all stored patterns (or at least with those that can be used in the current context)
  - Compute a DTW score for every vocabulary word (when using multiple references, compute one score out of many, e.g. average or max)
  - Recognize the word with the best DTW score
- This approach works only for very small vocabularies and/ or for speaker-dependent recognition
- Forms the basis of Hidden Markov Model based recognizers

More reliable distances

- Instead of Euclidean distance
  - Doesn't care about the standard deviation

$$\sqrt{\sum_{i=0}^{N}(T_i - S_i)^2}$$

- Use Mahalanobis distance
  - Care about means and standard deviation

$$\sqrt{\sum_{i=0}^{N}(\frac{(\mu_i - S_i)}{\sigma_i})^2}$$

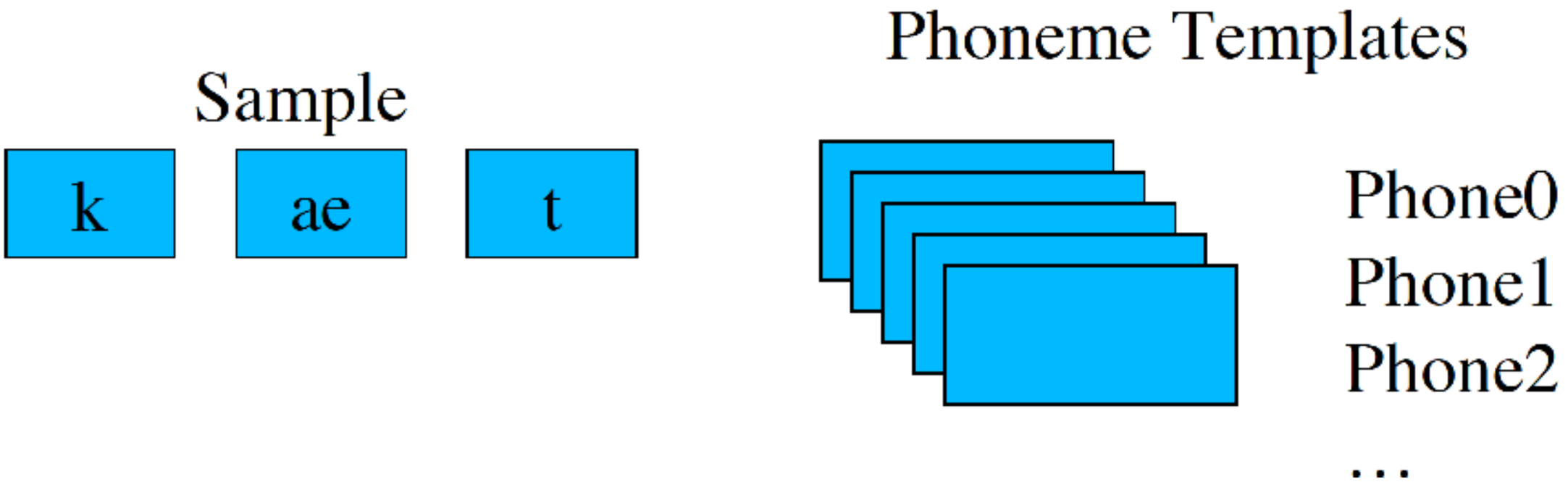
More reliable matching

- Having multiple template examples
  - Individual matches or
  - Average them together
- DTW align all of the examples
- Collect statistics as a Gaussian
  - Mean and standard deviation for each coeff

$$\{\mu_o, \sigma_0, \ \mu_1, \sigma_1, \ \mu_2, \sigma_2, \ \ldots\}$$

Beyond Template Matching

- *String phoneme templates together*
  - *A template model for each phoneme*



Automatic speech recognition

Automatic speech recognition

- Instead of starting from the waveform, we will often start from **speech features** (MFCC, etc.) through the **feature extraction** module
- Let's think of the conversion from speech feature O to text W

Speech recognition with a probabilistic formulation

- **MAP decision theory**: Estimate the most probable word sequence among all possible word sequences (I'll omit the domain sometimes)

$\hat{W}$ $\mathcal{W}$

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$

Speech recognition with a probabilistic formulation

- **MAP decision theory**: Estimate the most probable $\hat{W}$ word sequence $\mathcal{W}$ among all possible word sequences (I'll omit the domain sometimes)

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$

- The following parts will discuss how to make this equation tractable
- To do that we need to prepare some basic math

Notation

Type	Font, case	Latex command	Looks like
Scalar variable	Italic font, lower case	x	x
Vector variable	Bold font, lower case	$\mathbf{x}$	$\mathbf{x}$
Matrix variable	Bold font, upper case	$\mathbf{X}$	$\mathbf{X}$

Notation

- Please specify the domain of variables
  - D-dimensional continuous vector: $\mathbf{o} \in \mathbb{R}^D$
  - $(D \times D)$-dimensional matrix: $\mathbf{\Sigma} \in \mathbb{R}^{D \times D}$
  - Word with vocabulary $\mathcal{V}$: $w \in \mathcal{V}$
- Set: calligraphic font, upper case, a set of elements are represented with curly brackets

  $$\mathcal{V} = \{\text{”one”}, \text{”two”}, \text{”three”}, \cdots\}$$

- Sequence: italic font, upper case, a sequence of elements are represented with round brackets

  $$O = (\mathbf{o}_1, \mathbf{o}_2, \cdots) \qquad O = (\mathbf{o}_t \in \mathbb{R}^D | t = 1, \cdots, T)$$

Notation

- Subsequence

$$O = (\mathbf{o}_1, \cdots, \mathbf{o}_{t_1}, \underbrace{\mathbf{o}_{t_1+1}, \cdots, \mathbf{o}_{t_2}}, \cdots, \mathbf{o}_T)$$

$$o_{t_1+1:t_2} = (\mathbf{o}_{t_1+1}, \cdots, \mathbf{o}_{t_2})$$

or

$$O_{t_1+1:t_2} = (\mathbf{o}_{t_1+1}, \cdots, \mathbf{o}_{t_2})$$

Speech recognition cases

- T-length speech feature sequence (D-dimensional vector)

$$O = (\mathbf{o}_t \in \mathbb{R}^D | t = 1, \ldots, T)$$

- N-length word sequence with vocabulary $\mathcal{V}$

$$W = (w_n \in \mathcal{V} | n = 1, \ldots, N)$$

Notation cont'd

- Operation: non-italic

Operation type	Latex command
	\log()
	\arg \max (), \text{argmax}(), etc.
	\text{sigmoid}()

- Index: subscript, italic

$W_n,\ \mathbf{o}_t$

- Type of variables, functions: superscript, non-italic

$x^{\mathrm{HMM}},\ x^{\mathrm{DNN}}$

$p^{\mathrm{HMM}}(x),\ p^{\mathrm{DNN}}(x)$

Probabilistic rules

- **Product rule**

$$p(x|y)p(y) = p(x, y)$$

- **Sum rule**

$$p(y) = \sum_x p(x, y)$$

- **Conditional independence assumption**

$$p(x|y, z) = p(x|z) \qquad p(x, y|z) = p(x|z)p(y|z)$$

Other rules

- **Bayes rule**

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_x p(y|x)p(x)}$$

- **Probabilistic chain rule**

$$p(x_1, \cdots, x_N) = \prod_{n=1}^{N} p(x_n|x_{1:n-1}) \quad \text{where} \quad p(x_1|x_{1:0}) = p(x_1)$$

- Both are derived with a combination of the product and/or sum rules

Other approximation

- **Viterbi approximation**

$$p(x|y) = \sum_z p(x, z|y) \approx \max_z p(x, z|y)$$

We often use this approximation to avoid $\sum_z$

- **Set an actual distribution**, e.g.,
  - $p(x)$: Gaussian distribution, Gamma distribution, softmax probability obtained by a neural network etc.

Now let us use **product** and **sum** rules and **conditional independence assumption** to formulate the speech recognition problem

- End-to-End Speech Recognition
- Classical speech recognition
  - Pipeline

Speech recognition

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$

End-to-end speech recognition

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$

O W

"I want to go to campus"

How to obtain the posterior $p(W|O)$

- We just replace it with a neural network-based function $f^{\mathrm{nn}}(\cdot)$

$$\underset{W}{\operatorname{argmax}}\, p(W|O) = \underset{W}{\operatorname{argmax}}\, f^{\mathrm{nn}}(W|O)$$

  - Easy and simple, no math (in this level), however
- W is a **sequence**! $W = (w_n \in \mathcal{V} | n = 1, \ldots, N)$
  - Very difficult to deal with it
  - Say $N = 10, \left|\mathcal{V}\right| = 100,$ we have to deal with 100^{10} possible sequences
  - Also, the length N is variable
  - We have to use a **special neural network** (e.g., attention, CTC, and RNN-transducer)

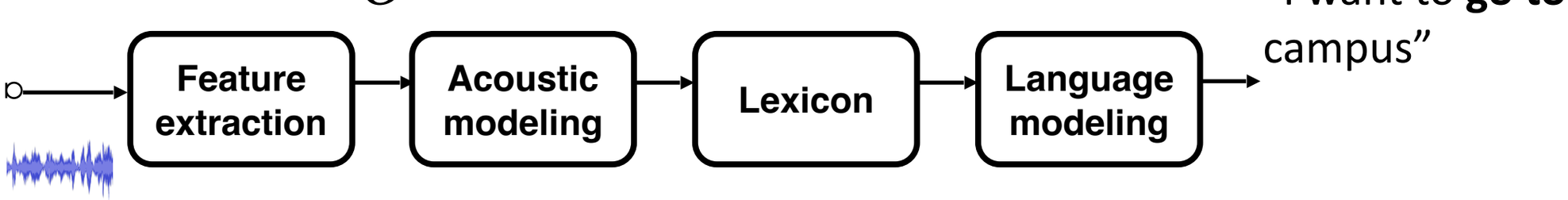
- Classical speech recognition
  - Pipeline

Speech recognition

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$

Classical (non-end-to-end) speech recognition

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$



Speech → Text

Speech O:

ASR

Text W: I want to go to the campus

Speech → Phoneme → Text

Speech o:

ASR

Phoneme L: AY W AA N T T UW G OW T UW K AE M P AH S

ASR

Text W: I want to go to campus

How to obtain the posterior $p(W|O)$

- Factorize the model with **phoneme**
  - Let $L = (l_i \in \{/\mathrm{AA}/, /\mathrm{AE}/, \cdots\} | i = 1, \cdots, J)$ be a phoneme sequence

$$
\begin{aligned}
\arg\max_W p(W|O) &= \arg\max_W \sum_L p(W, L|O) && \text{Sum rule} \\
&= \arg\max_W \sum_L \frac{p(O|W, L)p(L|W)p(W)}{p(O)} && \text{Bayes+ Product rule} \\
&= \arg\max_W \sum_L p(O|W, L)p(L|W)p(W) && \text{Ignore } p(O) \text{ as it does not depend on } W \\
&= \arg\max_W \sum_L p(O|L)p(L|W)p(W) && \text{Conditional independence assumption}
\end{aligned}
$$

Note: the right hand side is not the probability as it lacks a sum to one constraint

Noisy channel model

$$\underset{\mathrm{W}}{\operatorname{argmax}} p(W \mid O) = \underset{W}{\operatorname{argmax}} p(O \mid W) p(W) \approx \underset{W}{\operatorname{argmax}} \sum_{L} p(O \mid L) p(L \mid W) p(W)$$

- **Speech recognition**
  - $p(O \mid L)$: Acoustic model (Hidden Markov model)
  - $p(L \mid W)$: Lexicon
  - $p(W)$: Language model (n-gram)

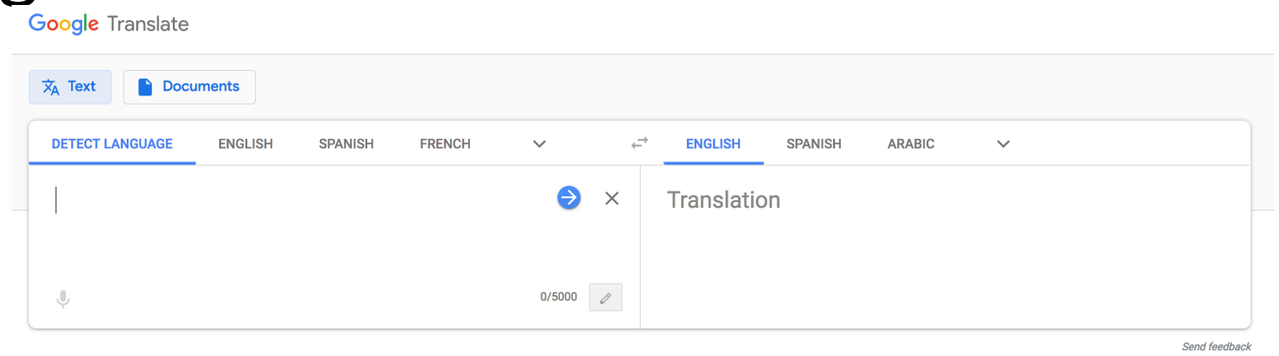
Noisy channel model

$$\underset{W}{\operatorname{argmax}} p(W \mid Y) = \underset{W}{\operatorname{argmax}} p(Y \mid W) p(W)$$

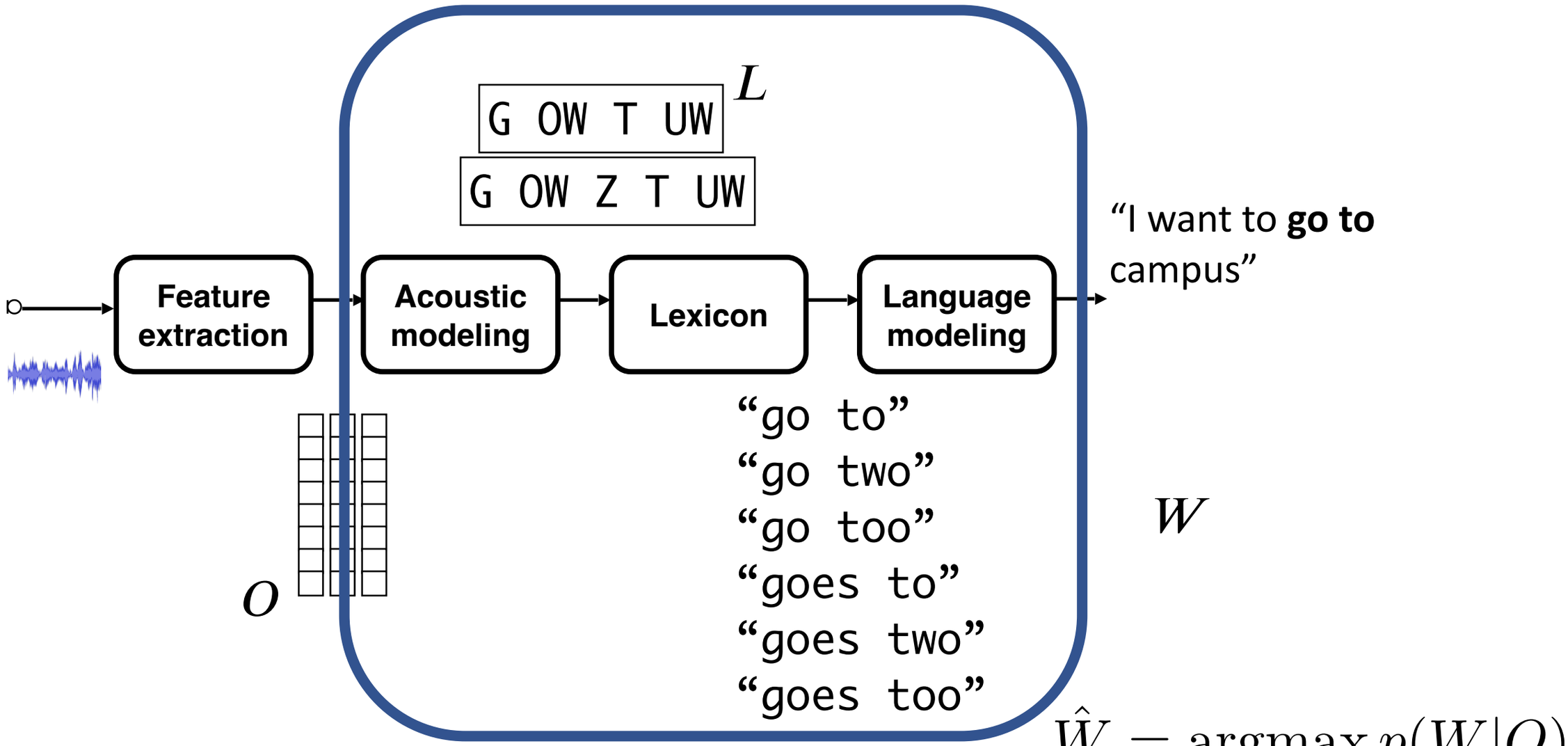
W: Target language text

Y: Source language text

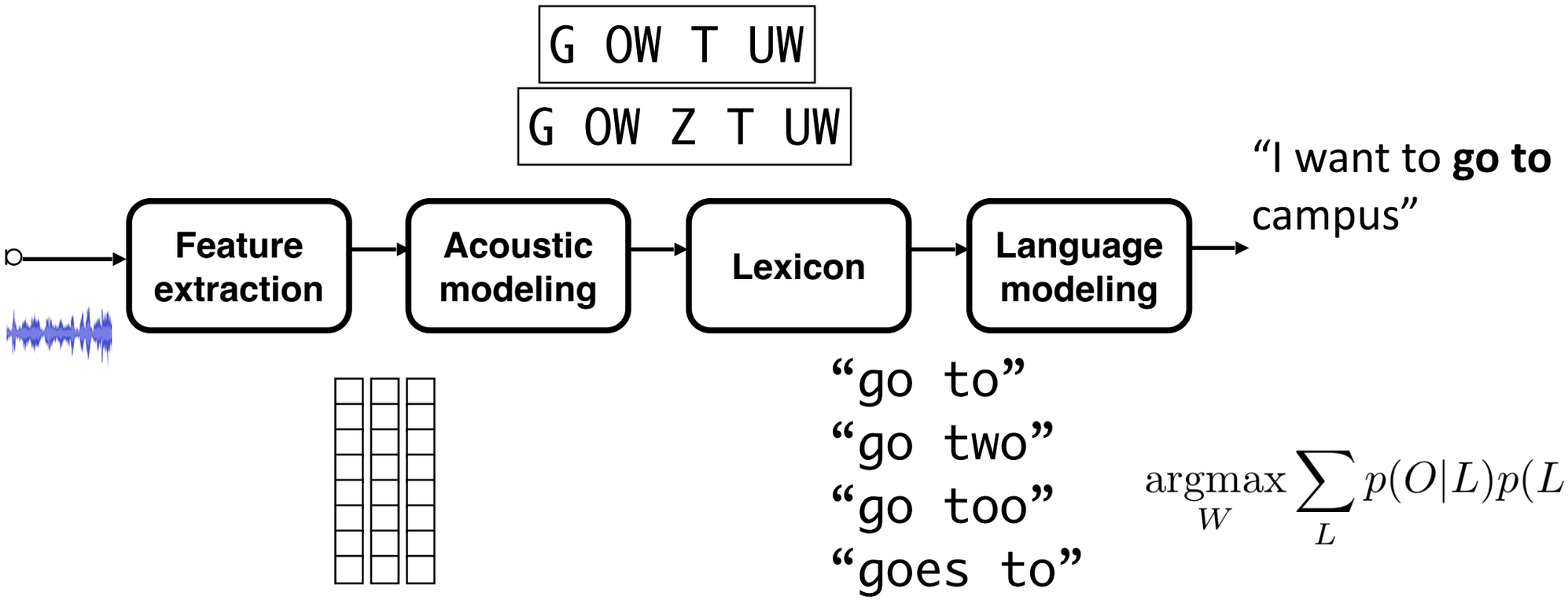
- **Machine translation**
  - $p(Y \mid W)$: Translation model
  - $p(W)$: Language model



Speech recognition pipeline



Speech recognition pipeline



$$\operatorname*{argmax}_{W} \sum_{L} p(O|L)p(L|W)p(W)$$

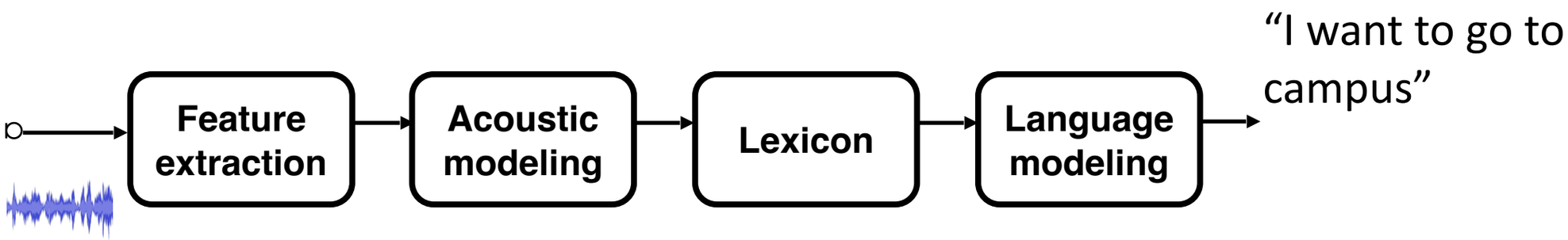
$p(O|L)$ $p(L|W)$ $p(W)$

Please remember the noisy channel model

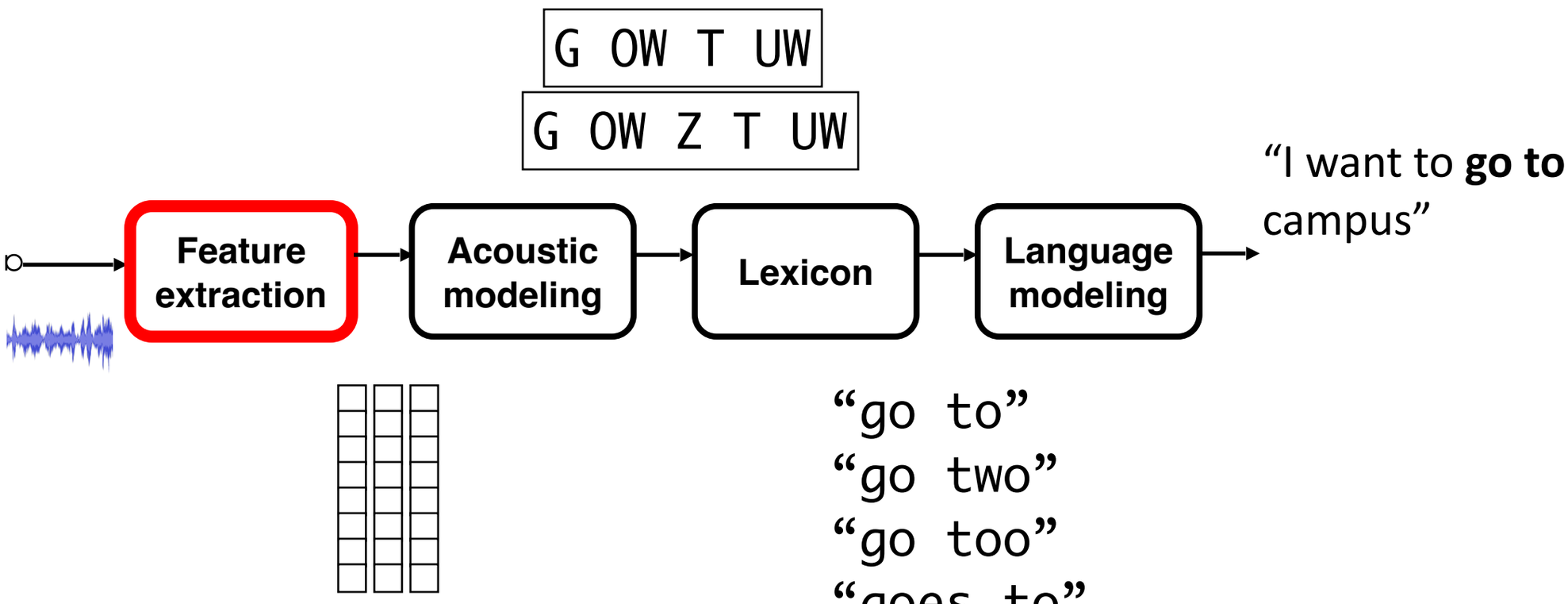
- **Factorization**
- **Conditional independence (Markov) assumptions**

We can elegantly factorize the speech recognition problem with a tractable subproblem

Main blocks of Classical ASR

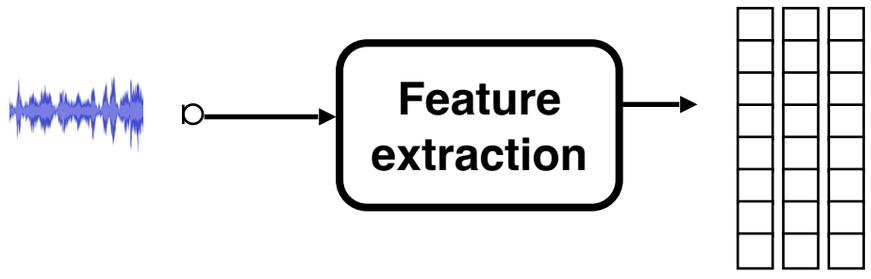


Speech recognition pipeline



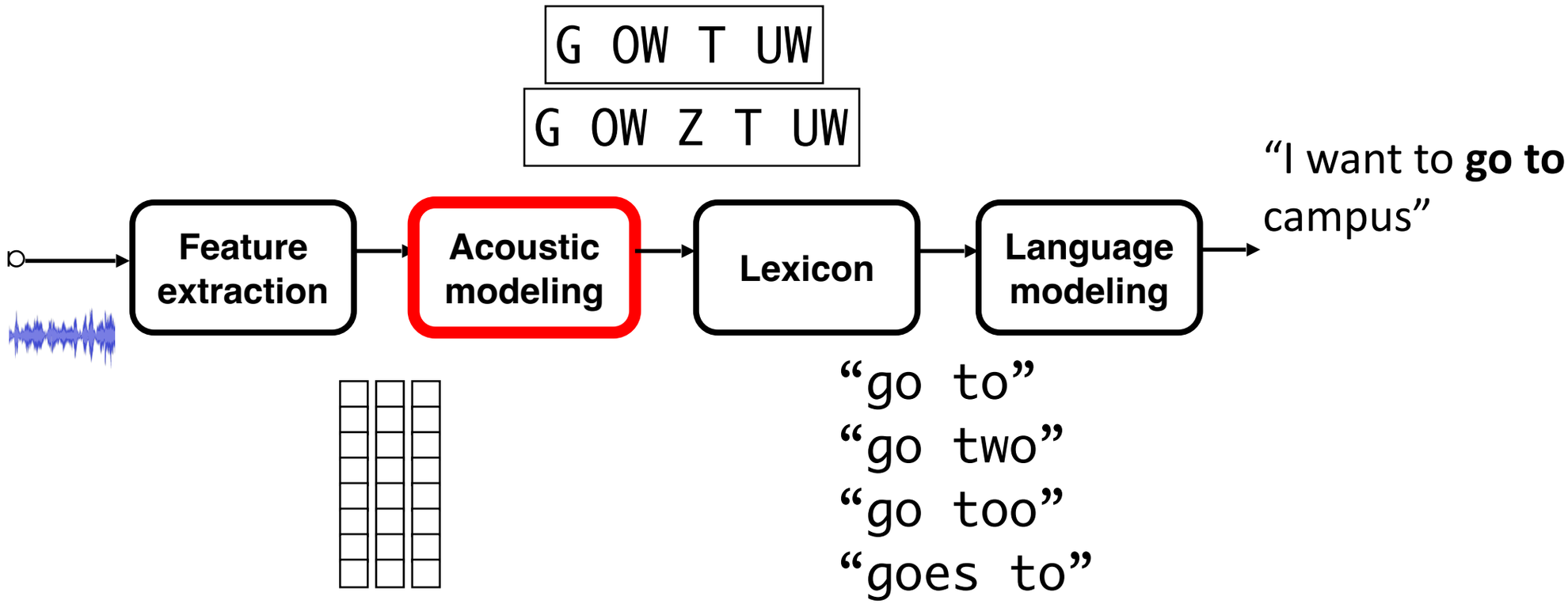
$$p(O|L) \quad p(L|W) \quad p(W)$$

Waveform to speech feature



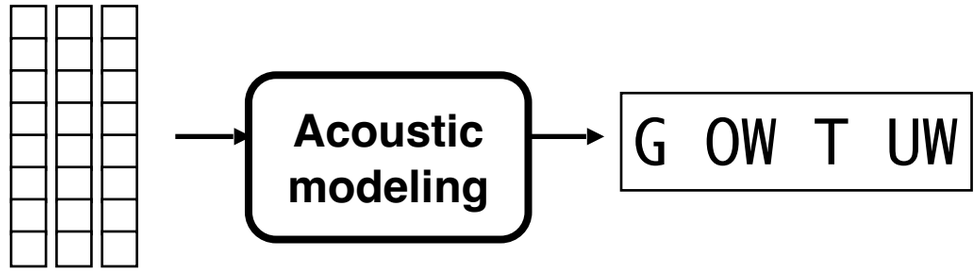
- Performed by so-called **feature extraction** module
  - Mel-frequency cepstral coefficient (MFCC), Perceptual Linear Prediction (PLP) used for **Gaussian mixture model (GMM)**
  - Log Mel filterbank used for **deep neural network (DNN)**
- Time scale
  - 0.0625 milliseconds (16kHz) to 10 milliseconds
- Type of values
  - Scalar (or discrete) to 12—40 dimensional vector

Speech recognition pipeline



$$p(O|L) \quad p(L|W) \quad p(W)$$

Speech feature to phoneme

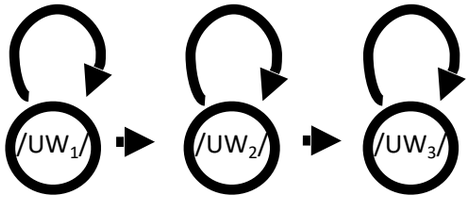


- Performed by so-called **acoustic modeling** module
  - Hidden Markov model (HMM) with **GMM** as an emission probability function
  - Hidden Markov model (HMM) with **DNN** as an emission probability function
- Time scale
  - 10 milliseconds to ~100 milliseconds (depending on a phoneme)
- Type of values
  - 12-dimensional continuous vector to 50 categorical value (~6bit)
- The most critical component to get the ASR performance
- It can be a probability of possible phoneme sequences, e.g.,

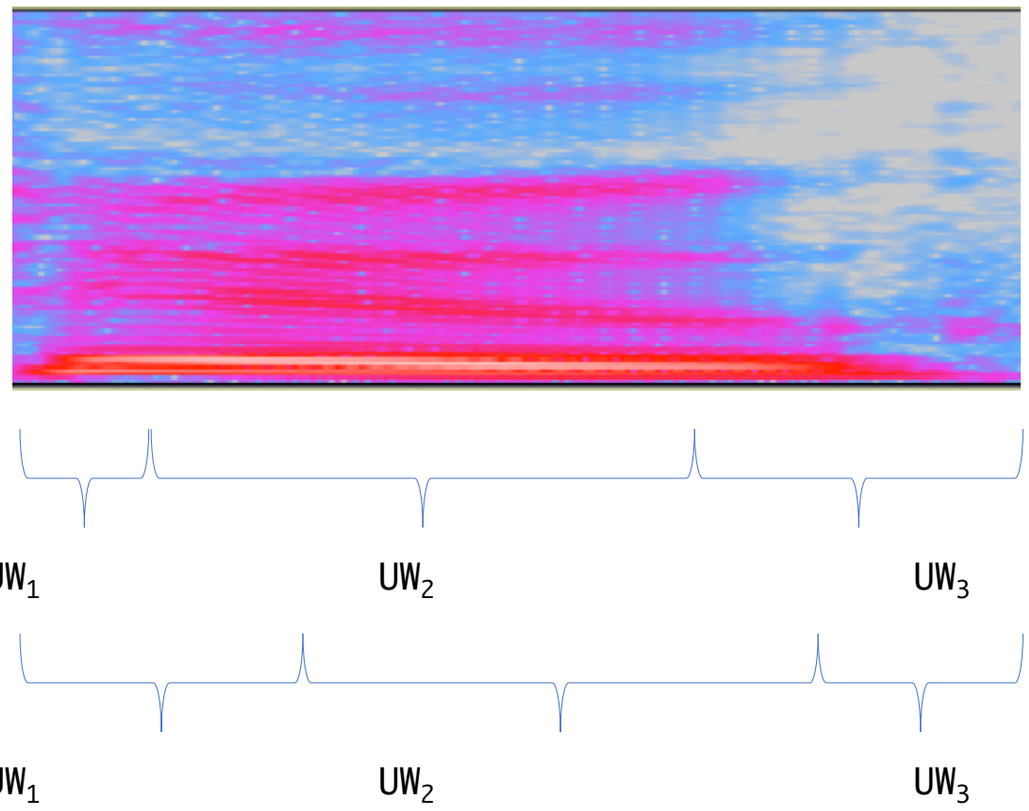
  G OW T UW or G OW Z T UW with some scores

Acoustic model $p(O \mid L)$

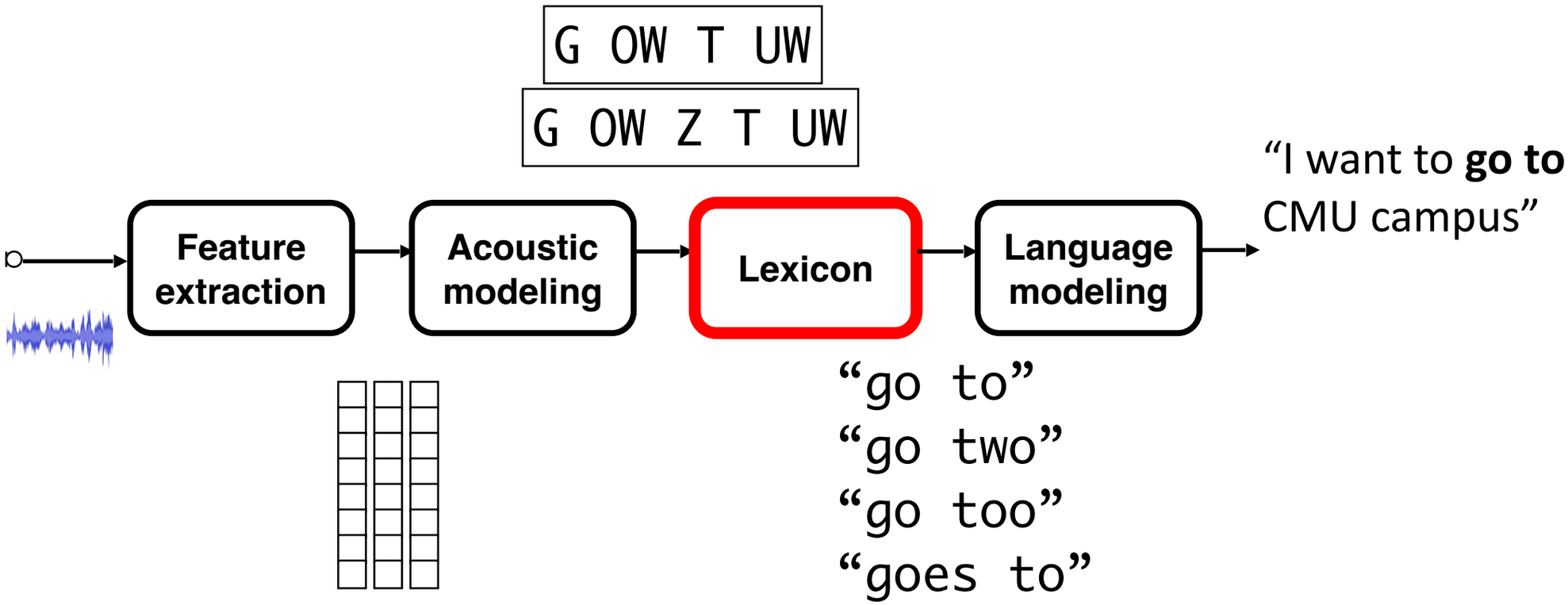
- O and L are **different lengths**
- **Align** speech features and phoneme sequences by using HMM



- Provide $p(O \mid L)$ based on this alignment and model
- The most important problem in speech recognition



Speech recognition pipeline



Phoneme to word

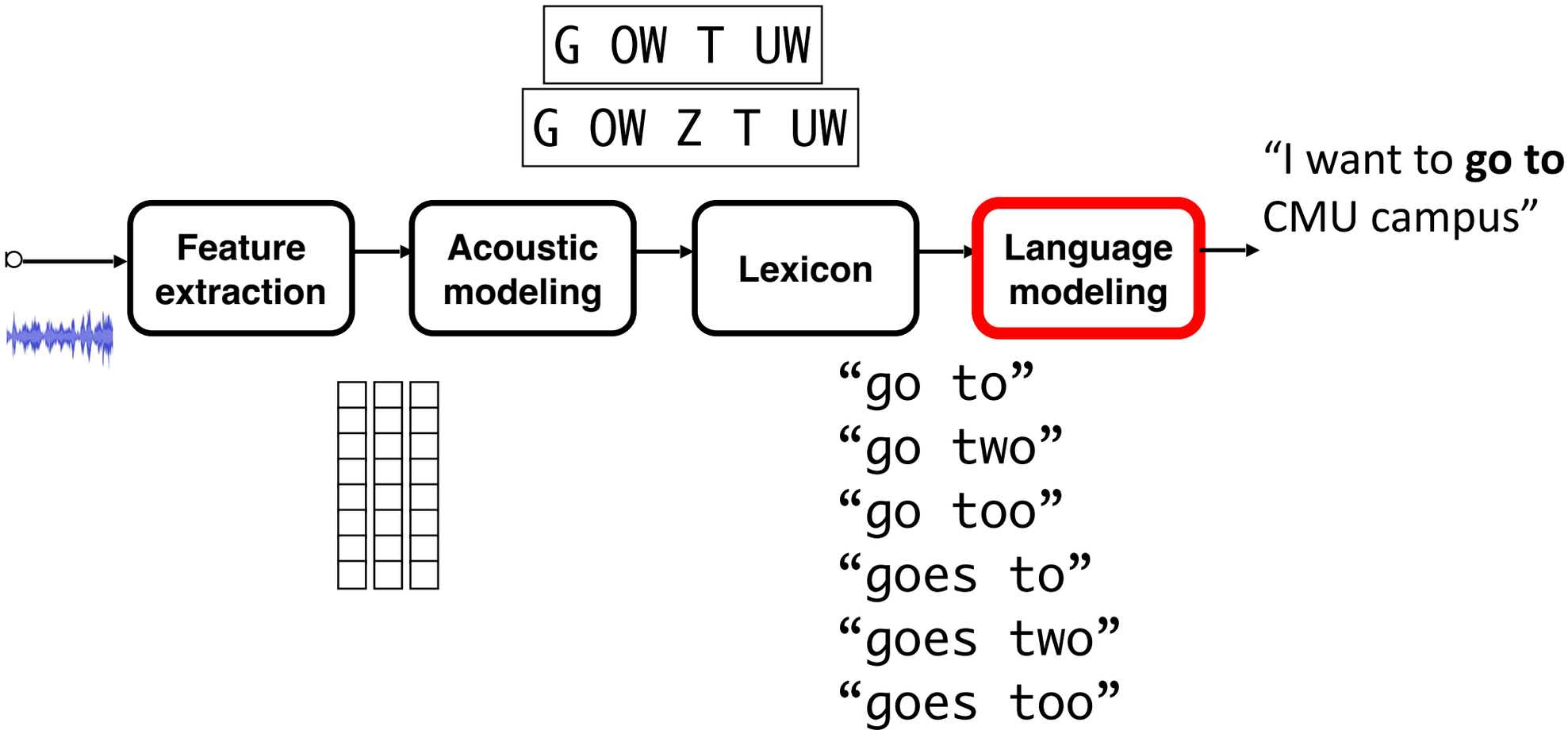
- Performed by **lexicon** module
  - American English: CMU dictionary
- Time scale
  - 100 milliseconds (depending on a phoneme) to 1 second (depending on a word and also language)
- Type of values
  - 50 categorical value (~6bit) to 100K categorical value (~2Byte)
- **We need a pronunciation dictionary**
- It can be **multiple** word sequences (one to many)

Lexicon $p(L \mid W)$

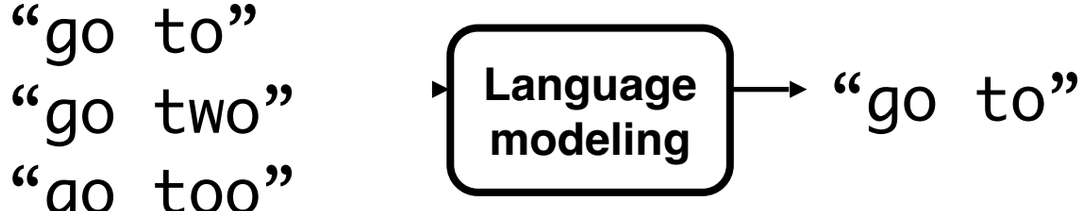
- Basically use a pronunciation dictionary, and map a word to the corresponding phoneme sequence
  - with the probability = 1.0 when single pronunciation
  - with the probability = $1.0/J$ when multiple (J) pronunciations

$$p(L|W) = p(/\mathrm{T}/, /\mathrm{OW}/|\text{"two"}) = 1.0$$

Speech recognition pipeline



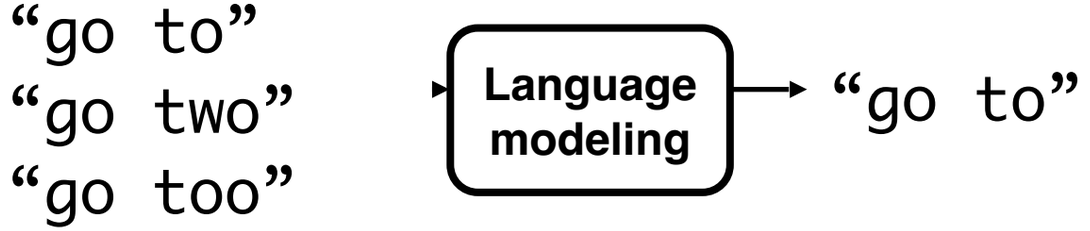
Word to text



- Performed by **language modeling** module $p(W)$
  - N-gram
  - Neural language model (recurrent neural network or transformer)
- From training data, we can basically find how possibly ”to”, “two”, and “too” will be appeared after “go”
  - Part of WSJ training data, 37,416 utterances
    - “go to”: **51** times
    - “go two”:
    - “go too”:

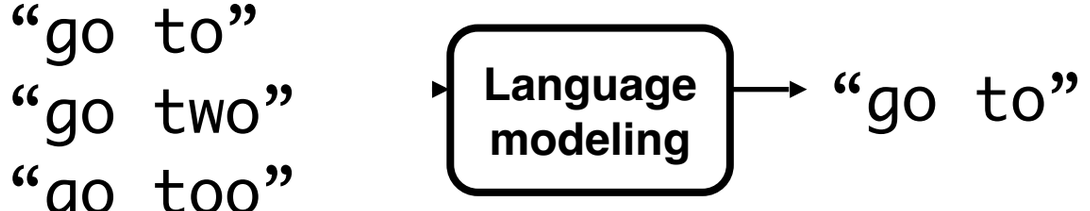
THE WALL STREET JOURNAL.

Word to text



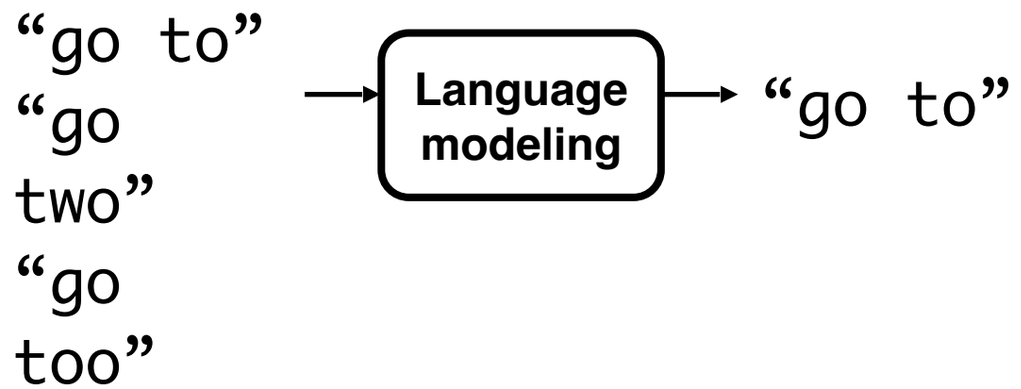
- Performed by **language modeling** module $p(W)$
  - N-gram
  - Neural language model (recurrent neural network or transformer)
- From training data, we can basically find how possibly ”to”, “two”, and “too” will be appeared after “go”
  - Part of WSJ training data, 37,416 utterances
    - “go to”: **51** times
    - “go two”: **0** times
    - “go too”: **0** times

Word to text



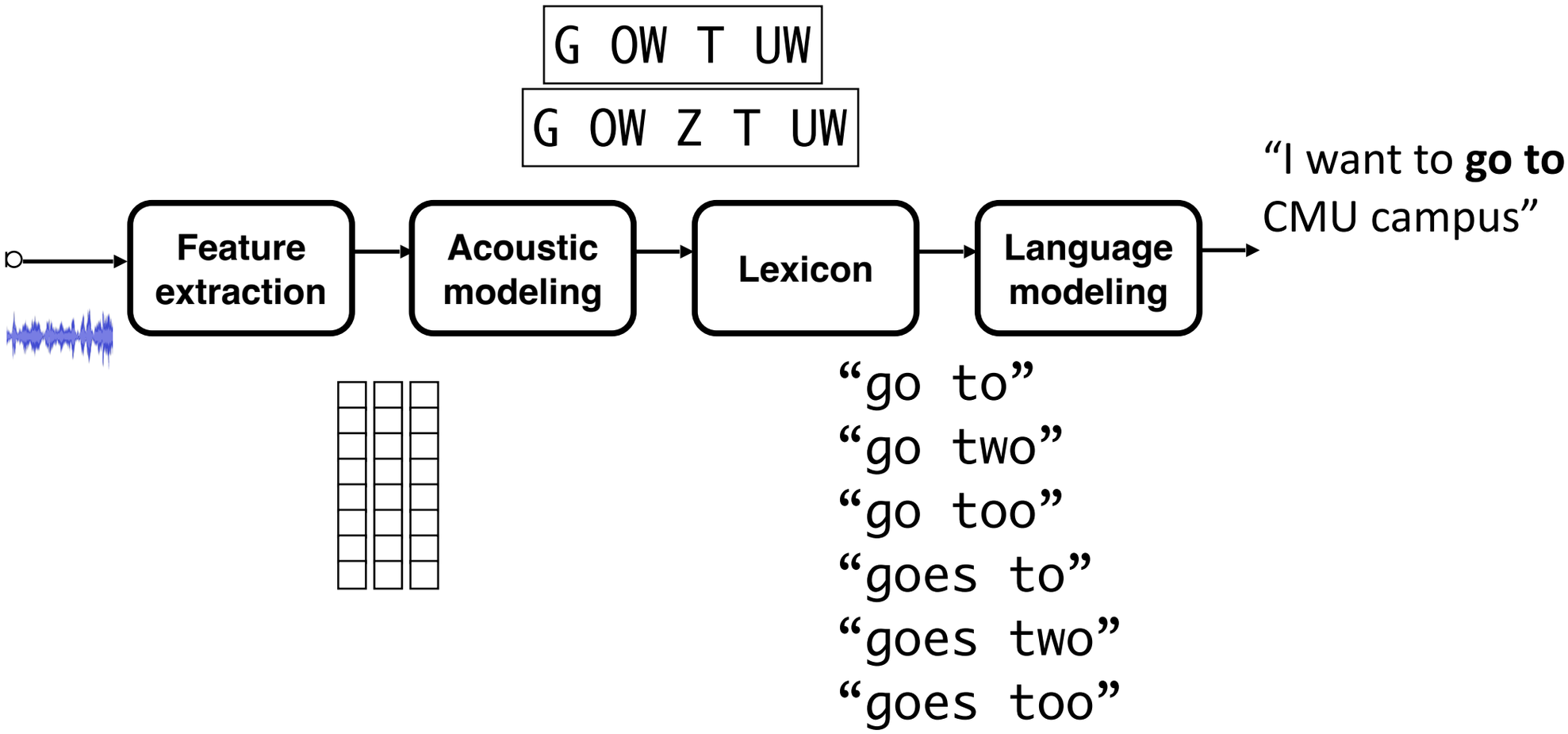
- Performed by **language modeling** module $p(W)$
  - N-gram
  - Neural language model (recurrent neural network or transformer)
- From training data, we can basically find how possibly ”to”, “two”, and “too” will be appeared after “go”
  - WSJ all text data, 6,375,622 sentences
    - “go to”: **2710** times
    - “go two”:
    - “go too”:

Word to text



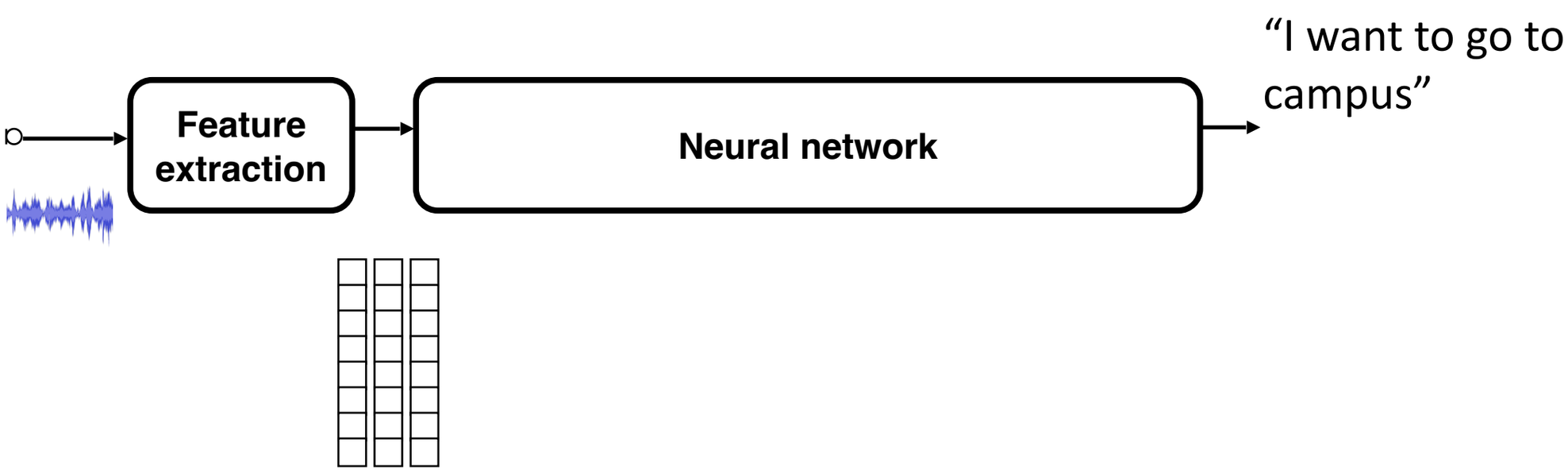
- Performed by **language modeling** mod
  - N-gram
  - Neural language model (recurrent neural network or transformer)
- From training data, we can basically find how possibly ”to”, “two”, and “too” will be appeared after “go”
  - WSJ all text data, 6,375,622 sentences
    - “go to”: **2710** times
    - “go two”: **2** times, e.g., “those serving shore plants often go two hundred miles or more”
    - “go too”:

Building speech recognition was really difficult…



- We need to develop all components
- Each component requires a lot of background knowledge
- We need to tune hyper-parameters in each module

Next: End-to-end speech recognition



- We can simply the complicated models
- Optimize all components by using back propagation
- We still need some formulations to make a problem tractable