Overview of Speech Recognition

January 17, 2020
Professor Marie Meteer
Brandeis University

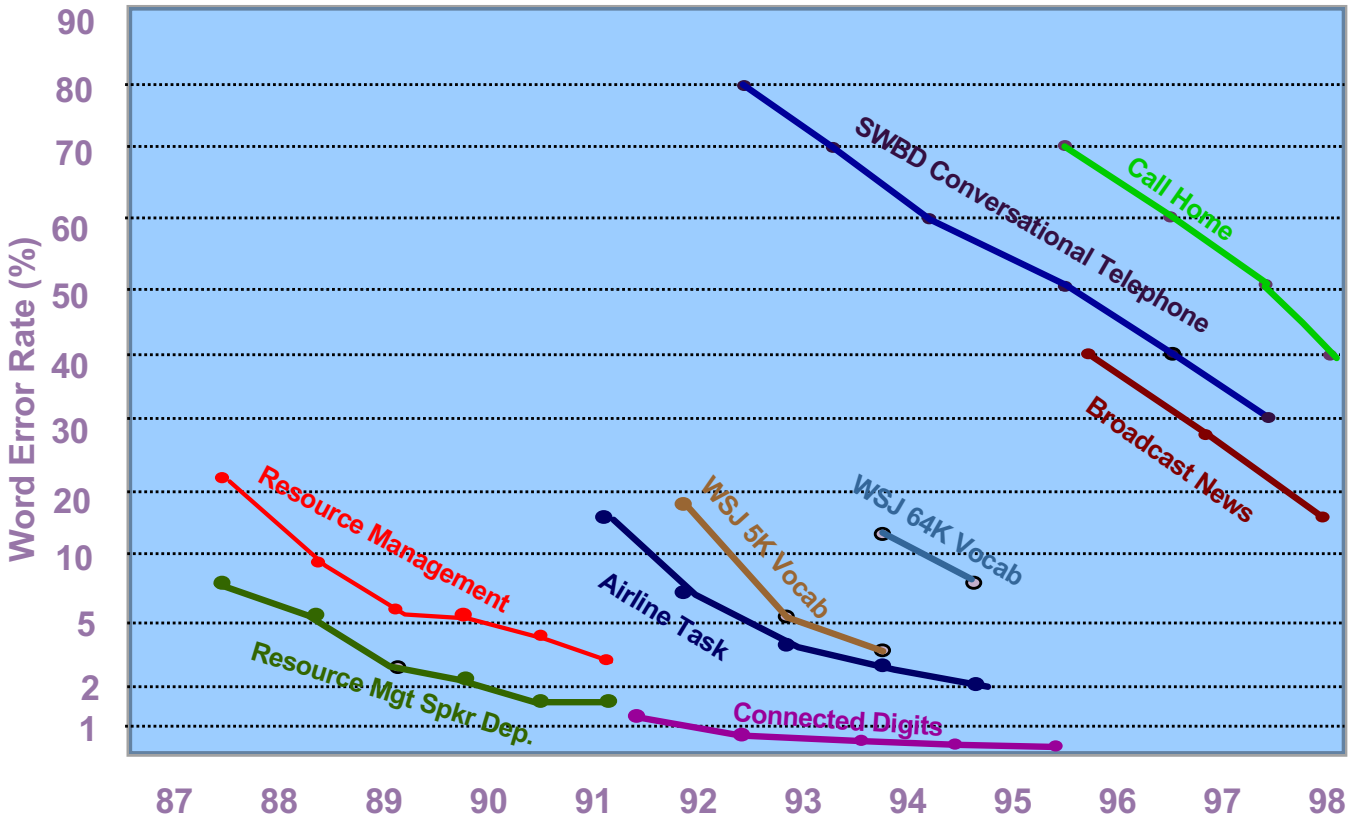
Today

- The speech problem
- Units of speech
- Hidden Markov Models
- Phonetic HMMs
- Recognition architecture
  - Training
  - Decoding

1985: First significant breakthrough in Speech Recognition

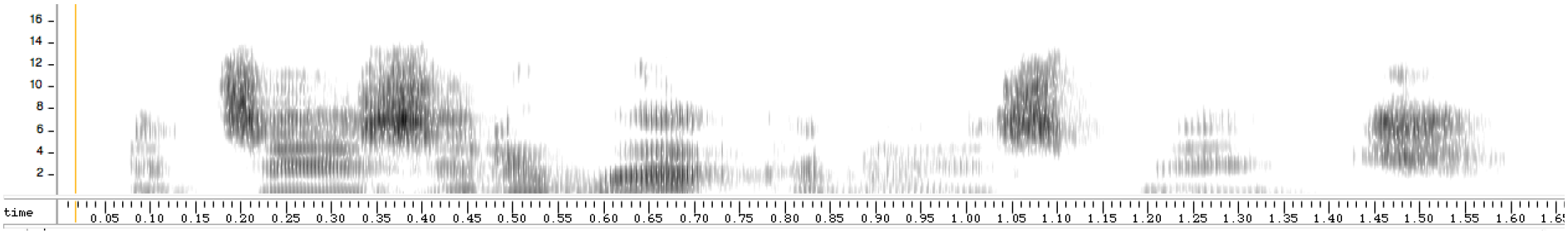
- Hidden Markoff Models
  - Mathematical framework
  - Ability to model time and spectral variability simultaneously
  - Ability to automatically estimate parameters given data
  - Not longer need to hand segment into phonemes
  - Segmentation and modeling done in one step
  - Data driven → Standard scientific procedures
  - Empirical!

A Decade of Progress in Speech Recognition



The speech problem

- Continuous-time signal →



- Sequence of discrete entities

ih	t	S	iy	z	iy	t	u	r	eh	k	o	g	n	iy	z	s	p	iy	ch
It's			easy			to		recognize								speech			

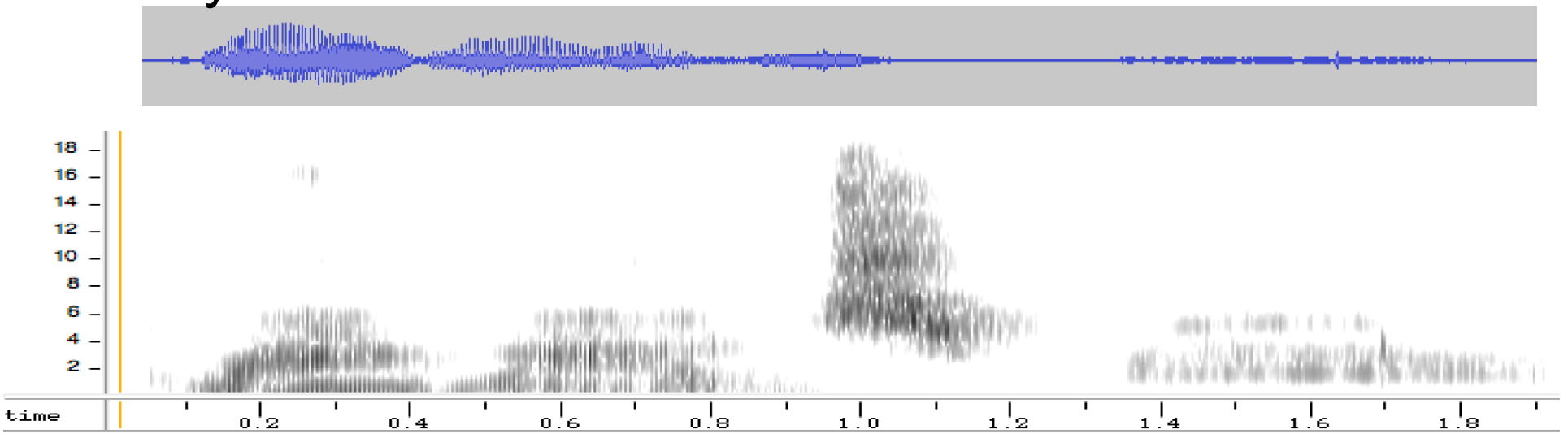
- Or did she say
  - It's easy to wreck a nice beach?

Challenge: Variability

- Linguistic
  - Can say many different things
  - Phonetics, phonology, syntax, semantic, discourse
- Speaker
  - Physical characteristics of speaker
  - Co-articulation (mouth has to transition between sounds
  - Native language/dialect
- Channel
  - Background noise
  - Transmission channel (microphone/telephone quality)

A look at the speech sounds

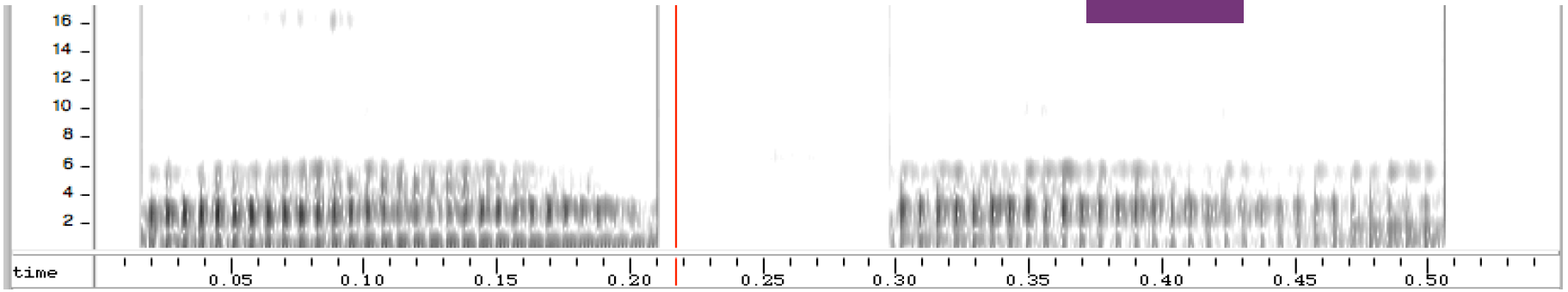
- Grey whales



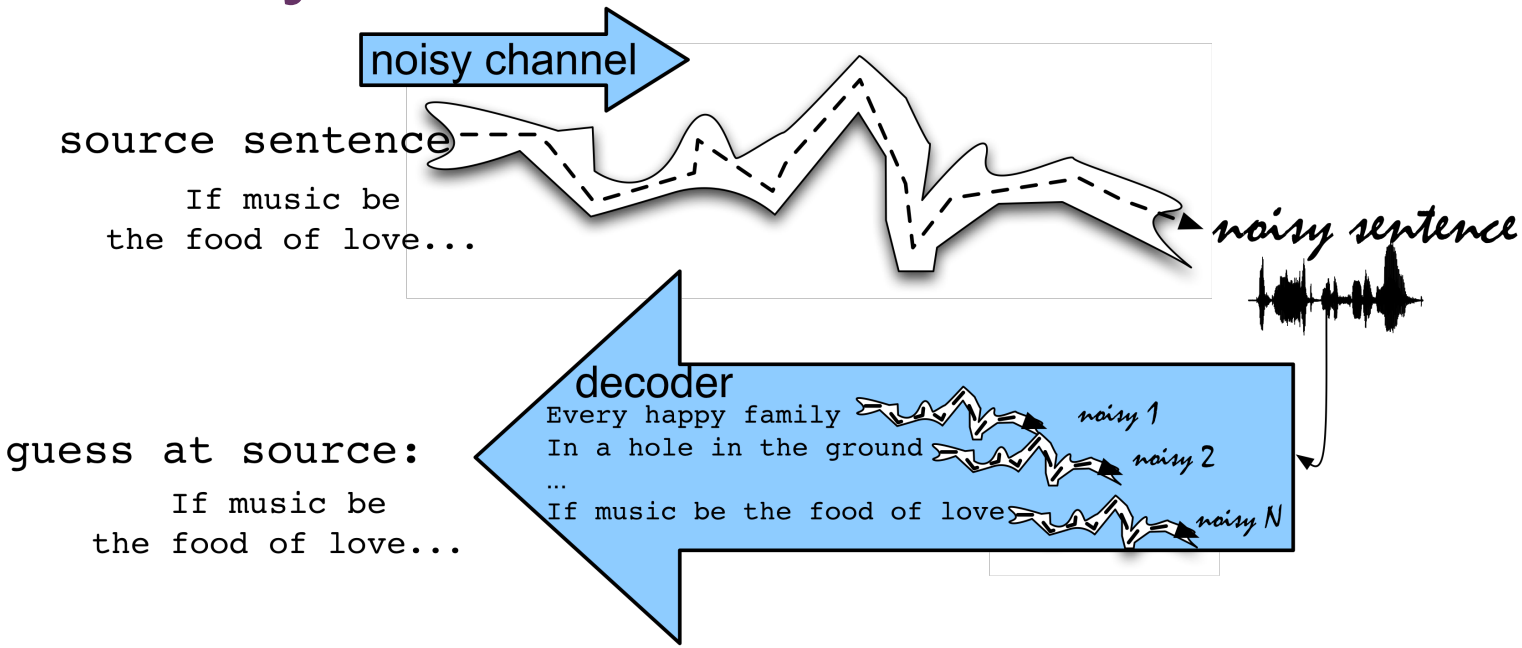
Words	Grey			Whales				meow
Phonemes	g	r	ey	w	ey	l	z	?
Triphones	- g r	g r ey	r ey w	ey w y	w ey l	ey l z	l z -	

Contextual variability

- Ey in grey, ey in whales



The Noisy Channel Model



- **Search** through space of all possible sentences.
- Pick the one that is **most probable** given the waveform.

Thanks to Dan Jurafsky for these slides

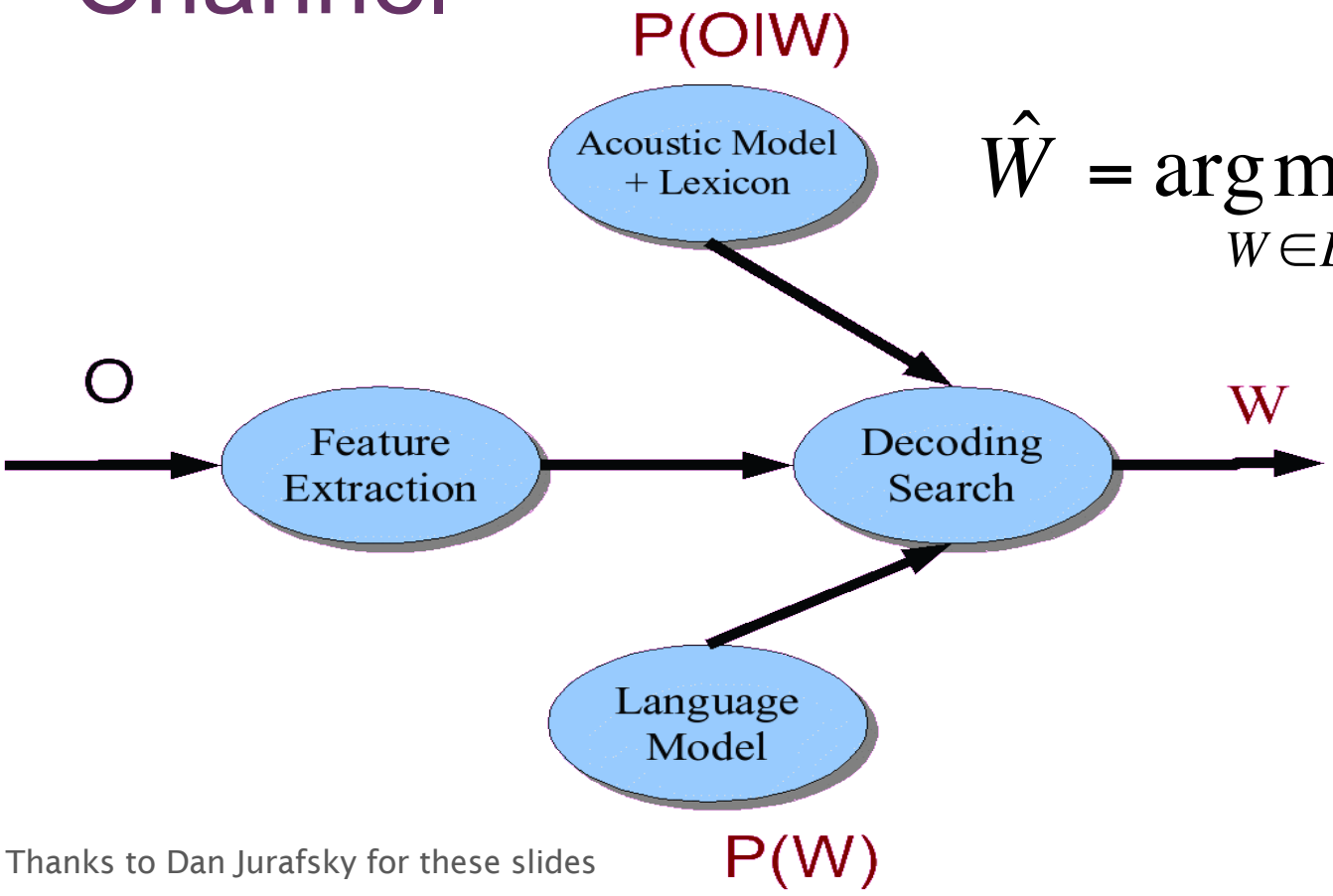
Design Intuition

- Build a statistical model of the speech sounds-to-words
  - Collect lots and lots of speech, and transcribe all the words.
  - Train the model on the labeled speech
- Create a “search space” of all the possible word combinations
  - Write a grammar
  - “Learn” likely sequences from lots of text
- Paradigm: Supervised Machine Learning + Search
  - Use the model to find the best sequence of words given the input sounds

Speech Architecture meets Noisy Channel

likelihood prior

$$\hat{W} = \arg\max_{W \in L} P(O \mid W) P(W)$$



Thanks to Dan Jurafsky for these slides

The Noisy Channel Model (Preview)

- What is the most likely sentence out of all sentences in the language L given some acoustic input O?
- Treat acoustic input O as sequence of individual observations
  - $O = o_1,o_2,o_3,\ldots,o_t$
- Define a sentence as a sequence of words:
  - $W = w_1,w_2,w_3,\ldots,w_n$

Thanks to Dan Jurafsky for these slides

Noisy Channel Model

- Probabilistic implication: Pick the highest prob S:

$$\hat{W} = \arg\max_{W \in L} P(W \mid O)$$

- We can use Bayes rule to rewrite this:

$$\hat{W} = \arg\max_{W \in L} \frac{P(O \mid W)P(W)}{P(O)}$$

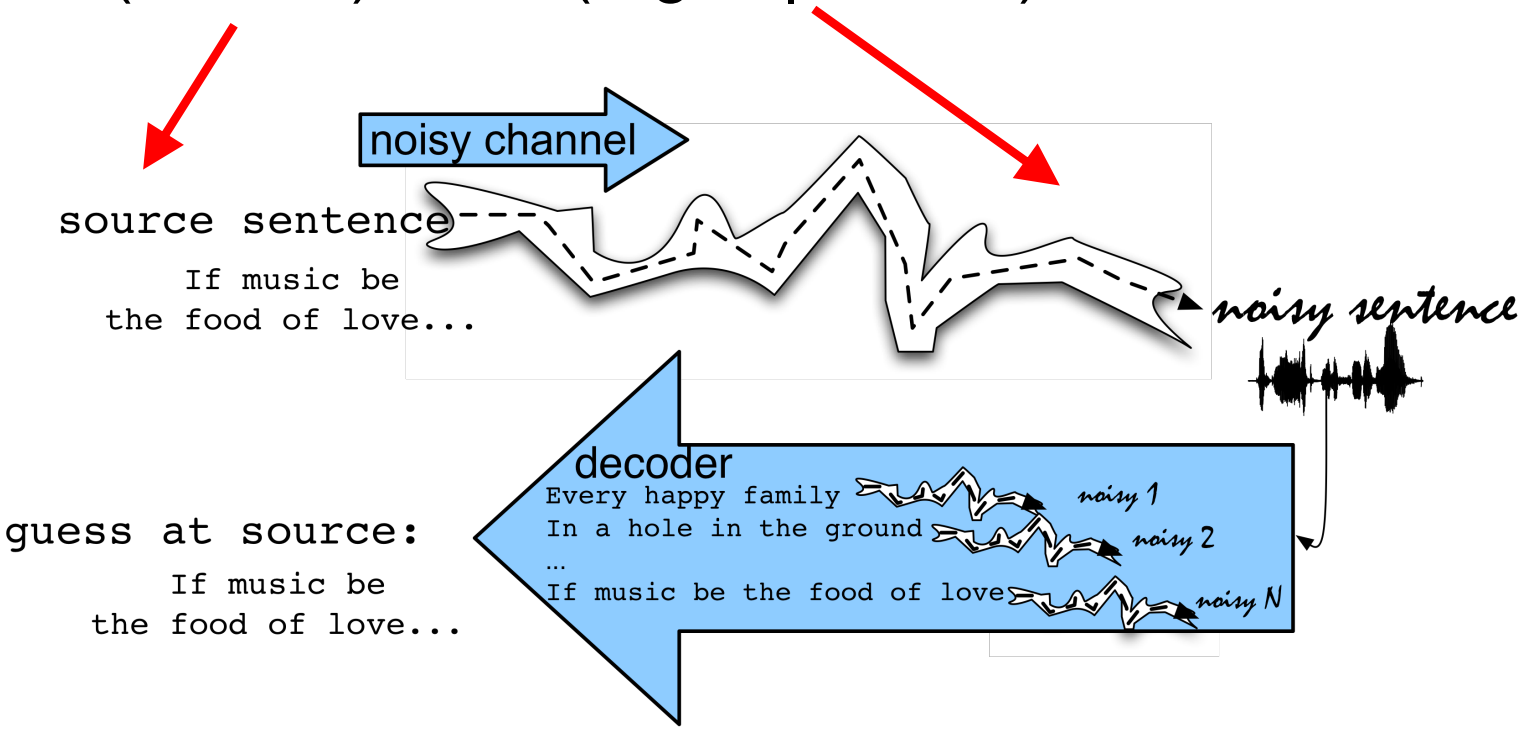
- Since denominator is the same for each candidate sentence W, we can ignore it for the argmax:

$$\hat{W} = \arg\max_{W \in L} P(O \mid W)P(W)$$

Thanks to Dan Jurafsky for these slides

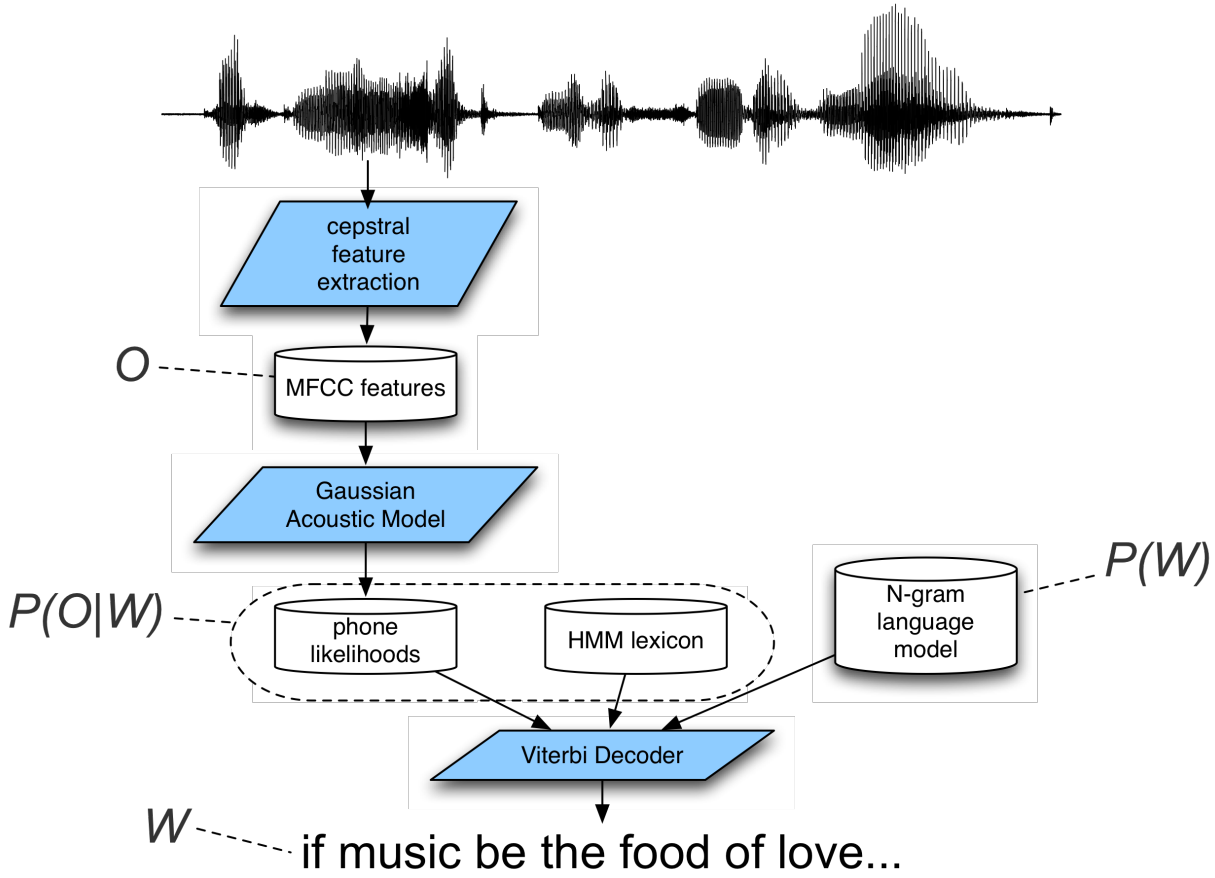
The noisy channel model

- Ignoring the denominator leaves us with two factors: P(Source) and P(Signal|Source)



Thanks to Dan Jurafsky for these slides

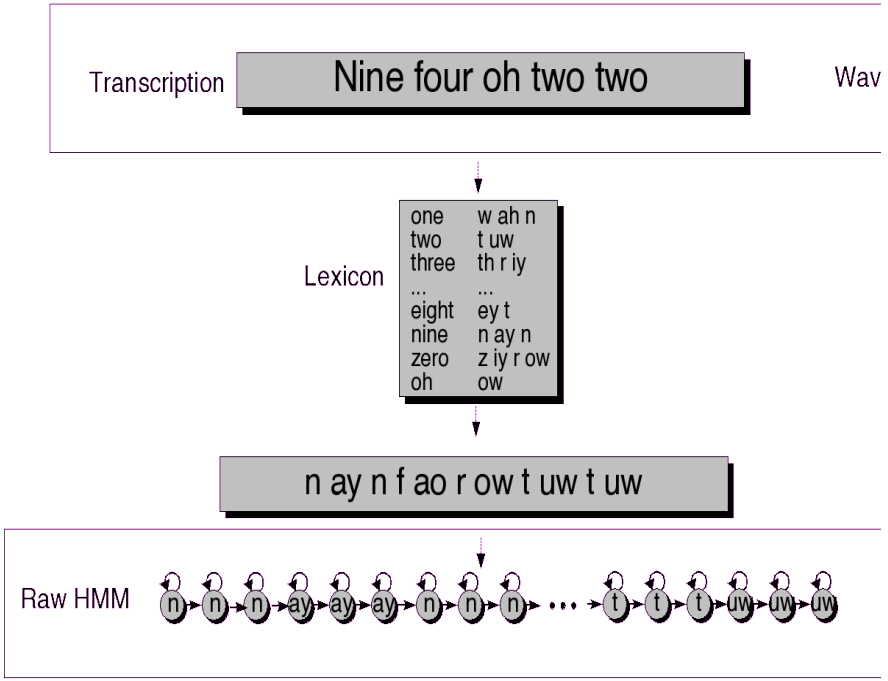
+ Speech Recognition Architecture



Thanks to Dan Jurafsky for these slides

How is this connected to language?

- The Dictionary
- Hand built knowledge source that ties words to sounds
- Sequence of words →
  - sequence of phonemes →
    - HMM states



From Phonetics to the Dictionary

- Dictionaries spell out the phonemes for each word
- Variations in pronunciation come from
  - Slight differences in shape of the vocal tract
  - Slight variations in articulation (accents)
  - Co-articulation with neighboring phonemes
  - Predictable variation from the position of phoneme in the word (e.g the “ps” in “tap” and “pat” are different)
- Some words have multiple pronunciations

  >RECORD R-EH-K-AXR-D R-IX-K-AO-R-D

+ Base + Domain Dictionary

>ABOUND	AX-B-AW-N-D
>ABOUNDED	AX-B-AW-N-D-IX-D
>ABOUNDING	AX-B-AW-N-D-IX-NX
>ABOUNDS	AX-B-AW-N-D-Z
>ABOUT	AX-B-AW-T
>ABOUT'S	AX-B-AW-T-S
>ABOVE	AX-B-AH-V
>ABOVEBOARD	AX-B-AH-V-B-AO-R-D
>ABPLANALP	AE-B-P-L-AX-N-AE-L-P
>ABRA	AA-B-R-AX
>ABRACADABRA	AE-B-R-AX-K-AX-D-AE-B-R-AX
>ABRAHAM	EY-B-R-AX-HH-AE-M

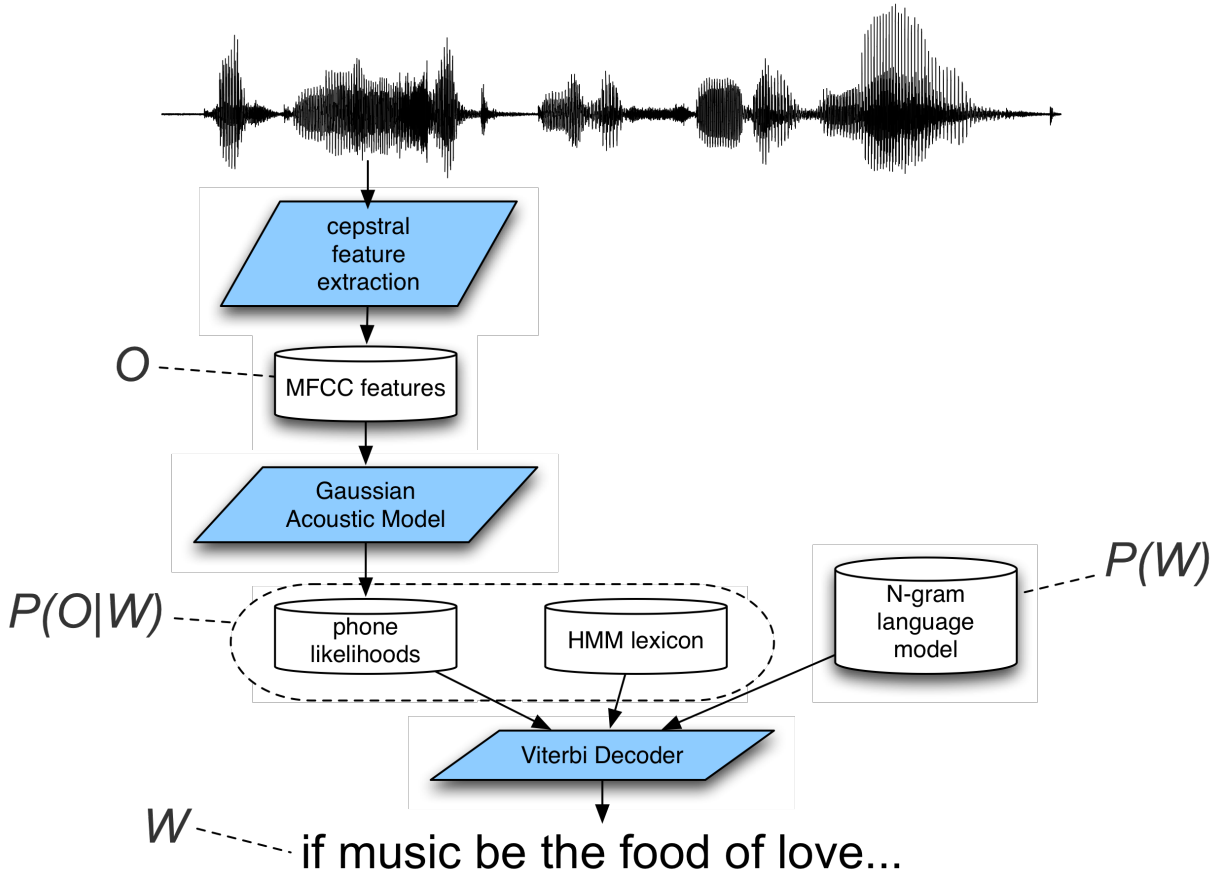
>AIRFORCE	EI-R-F-OW-R-S
>APPROACHING	AX-P-R-OW-CH-IX-N
	AX-P-R-OW-CH-IX-NX
	P-R-OW-CH-IX-N
>ARAC	EY-R-AE-K
>ATIS	EY-T-IX-S
>AZIMUTH	AE-Z-M-EH-TH
>BLACKCAT	B-L-AE-K-AE-T
>CAIRNS	K-AE-R-IX-N-Z

Architecture: Five easy pieces

- Feature extraction
- HMMs, Lexicons, and Pronunciation
- Acoustic Modeling
- Language Modeling
- Decoding

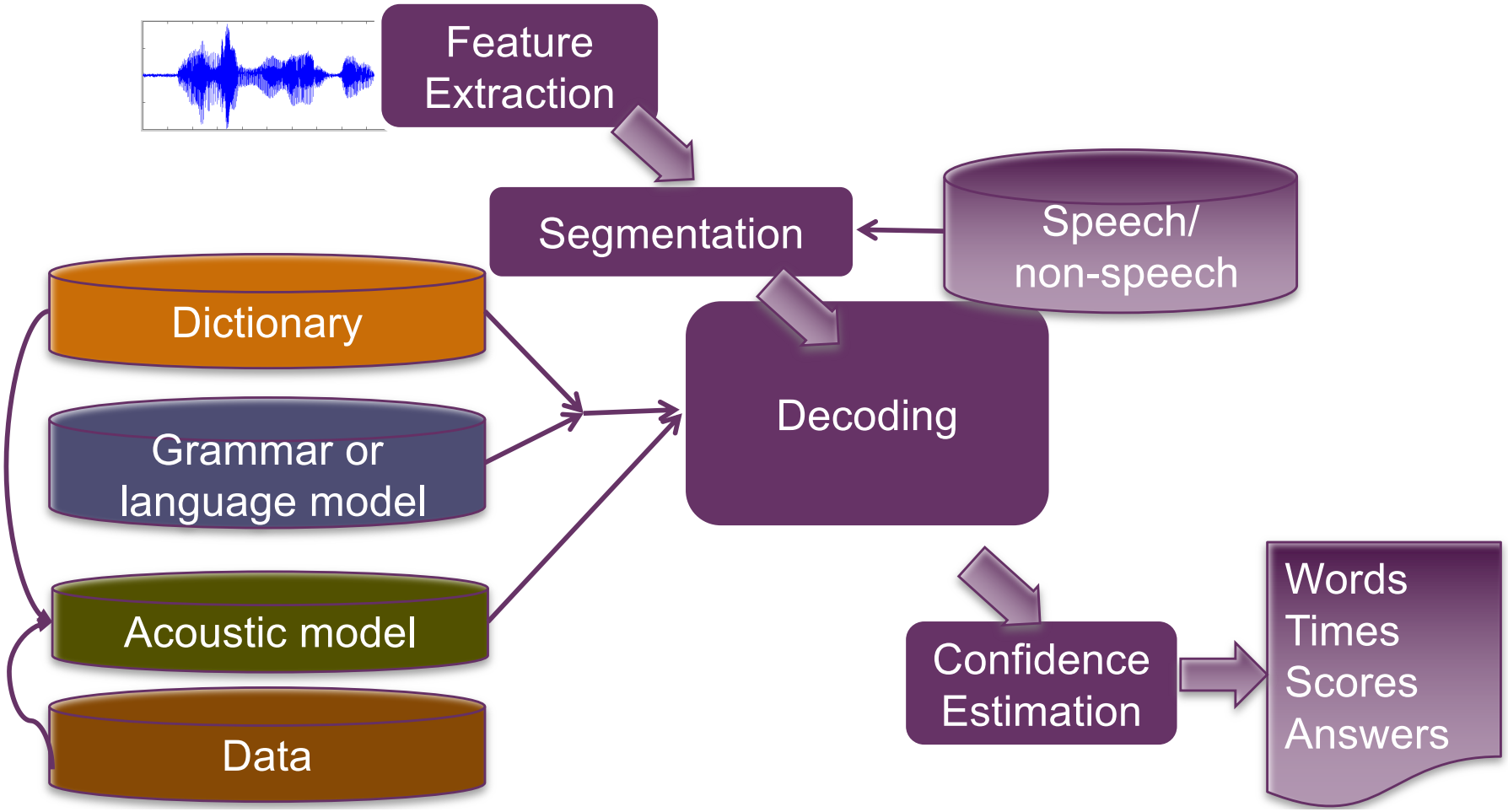
Thanks to Dan Jurafsky for these slides

+ Speech Recognition Architecture

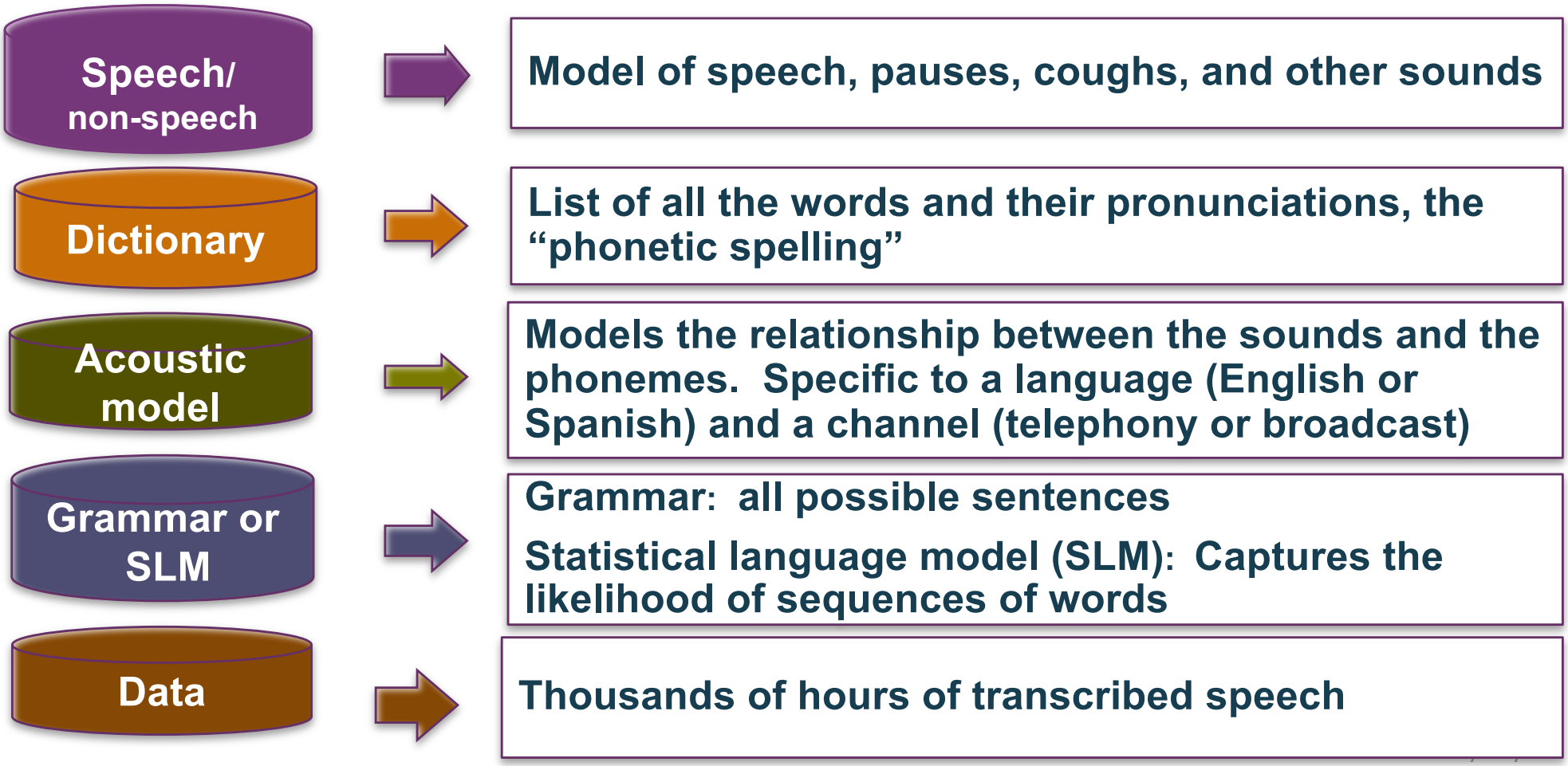


Thanks to Dan Jurafsky for these slides

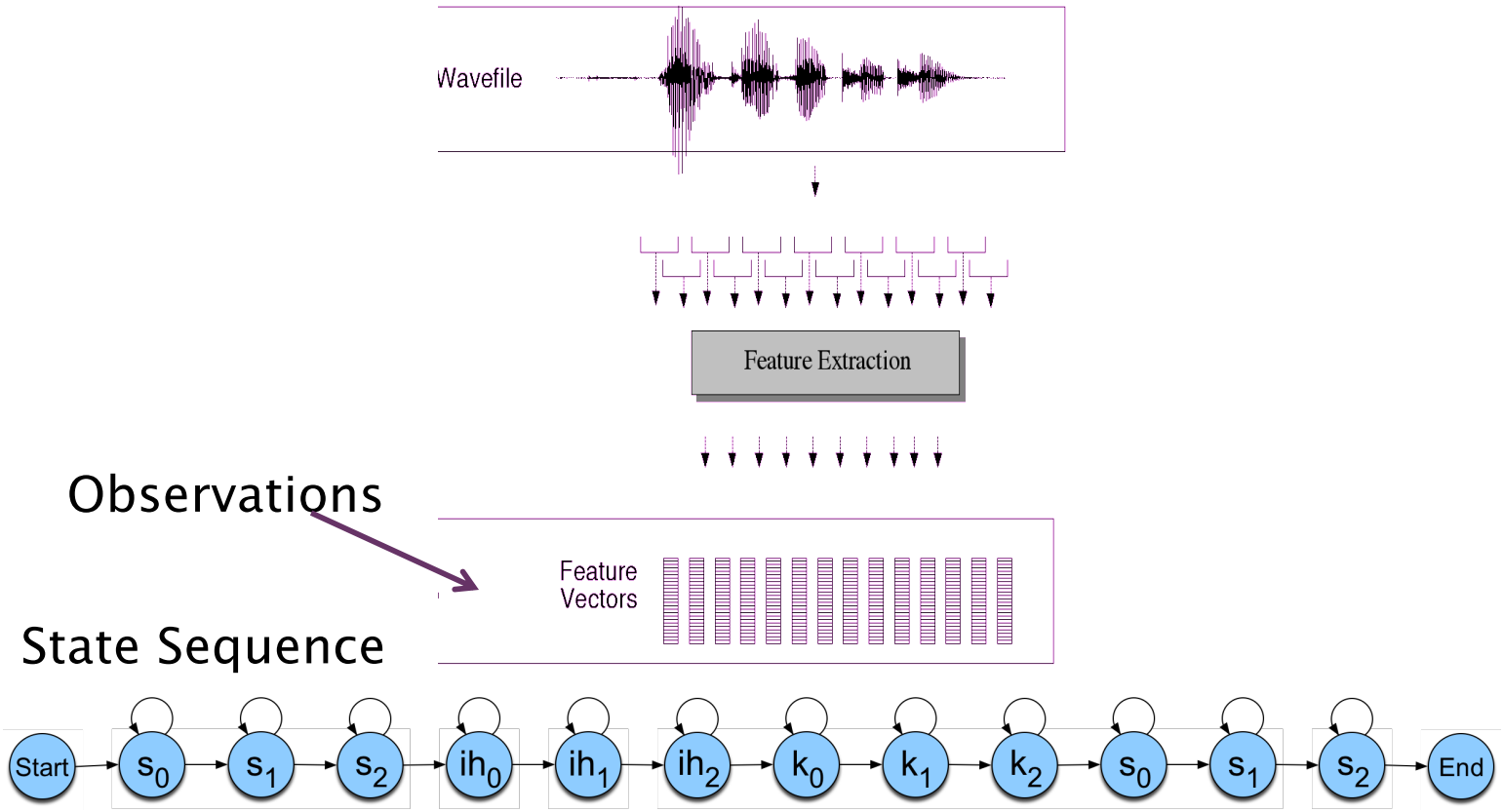
Speech Recognition Pipeline



+ Speech Recognition Knowledge Sources



+ Front End

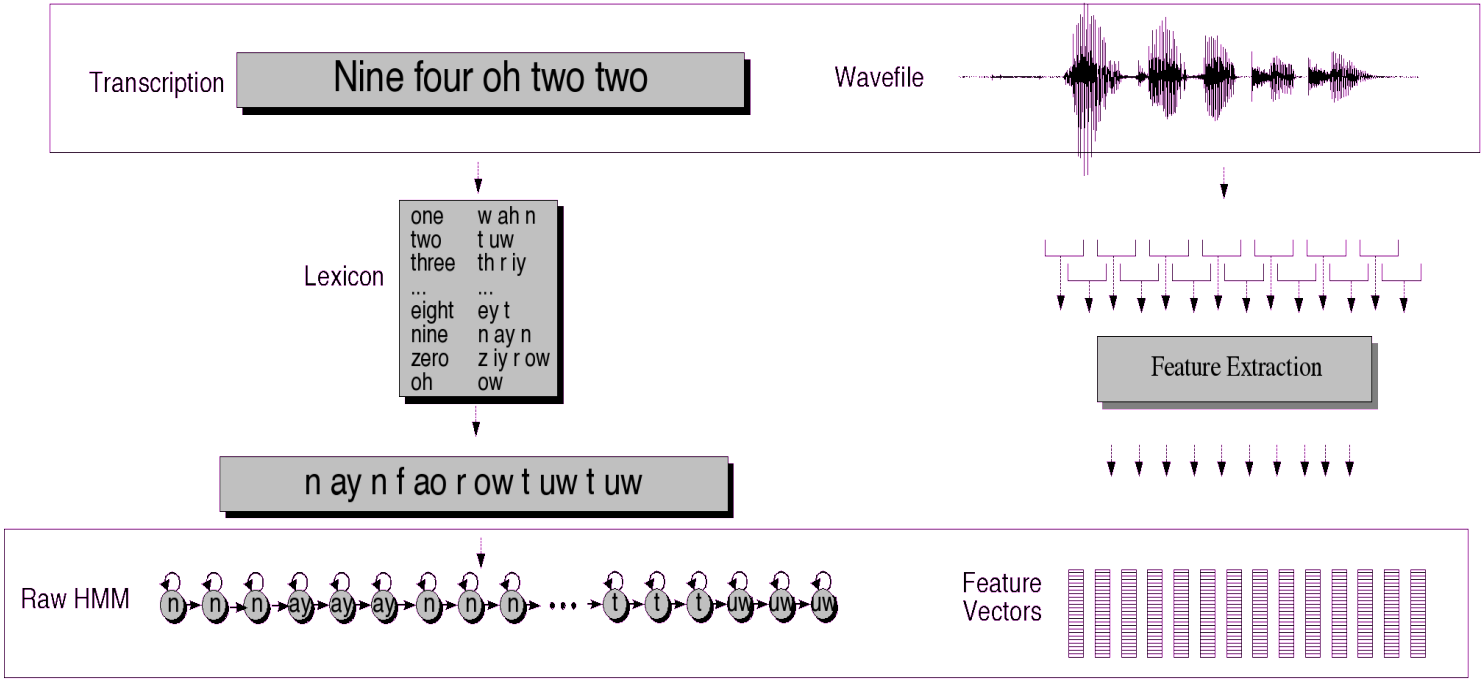

Wavefile
Feature Extraction
Observations
Feature
Vectors
State Sequence
Start
s_0
s_1
s_2
ih_0
ih_1
ih_2
k_0
k_1
k_2
s_0
s_1
s_2
End

Acoustic Modeling Training

learns the relationship between feature vectors and triphones

Words	Nine	night
Phonemes	N-AY-N	N-AY-T
Triphones	<S>N_{AY}-$N$$AY_{N}$-$AY$$N$<S>	<S>N_{AY}-$N$$AY_{T}$-$AY$$T$<S>

+ Embedded Training



Statistical Language Model (SLM)

- Captures how words are used in a particular domain
  - Specific to dialect (“in the hospital” vs. “in hospital”)
  - Specific to domain (frequency of different words)
  - Disambiguate homophones (“disc” vs. “disk)
- But allows any word order
- Trigram model
  - Probability of a word given the previous two words

What is the missing word?

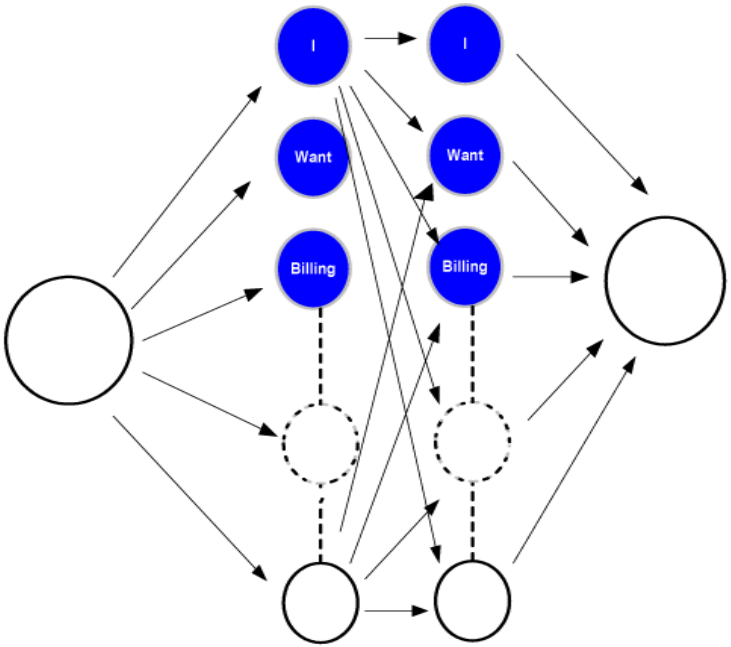
“I love _____”

NOW what is the missing word?

“My favorite TV show was I love _____”

Language (or Domain) Model: Statistical Grammar

- Word order is guided by statistics
- Data is used to count the likelihood a word given the previous
  - “in the hospital”
  - “the in hospital”
  - “the hospital in”
- Creates a lattice that is searched
- Training data must be relevant!



Bigram

+ Back to HMMs

- Why Markov model?

Markov chain models sequences	→	Speech is sequential can be modeled as a sequence of states specified by the dictionary
Transitions in the chain are probabilistic	→	Probabilities model uncertainly well

- Why Hidden Markov model?
  - Output symbols are probabilistic distribution over all labels
  - The actual sequence of states for a particular output is “Hidden”
  - There is one sequence that is the most probable to generate the output symbols

Hidden Markov Models formally

- States $Q = q_1, q_2 \ldots q_N$;
- Observations $O = o_1, o_2 \ldots o_N$;
  - Each observation is a symbol from a vocabulary $V = \{v_1, v_2, \ldots v_V\}$
- Transition probabilities
  - Transition probability matrix $A = \{a_{ij}\}$

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \le i, j \le N$$

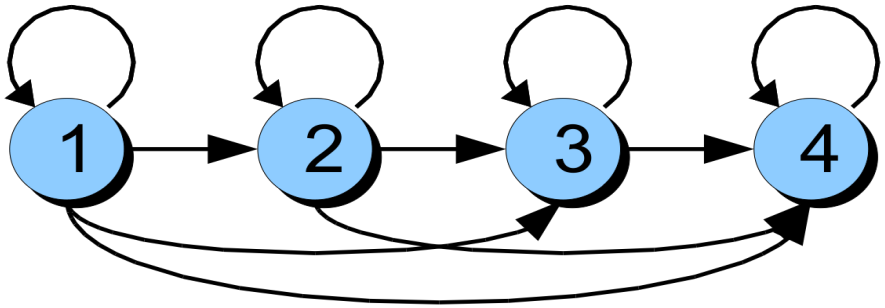
- Observation likelihoods
  - Output probability matrix $B = \{b_i(k)\}$

$$b_i(k) = P(X_t = o_k \mid q_t = i)$$

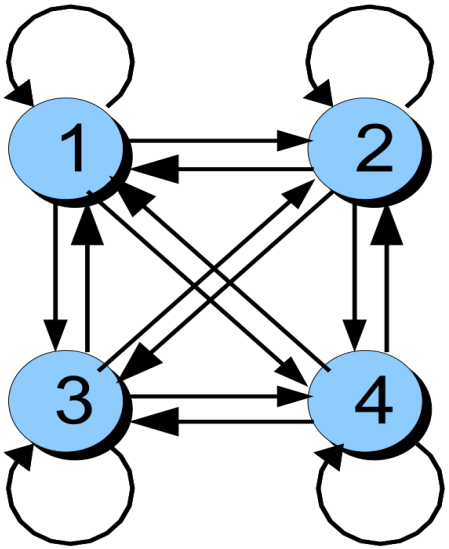
- Special initial probability vector π

$$\pi_i = P(q_1 = i) \quad 1 \le i \le N$$

Different types of HMM structure



Bakis = left-to-right (allowing skips)



Ergodic = fully-connected

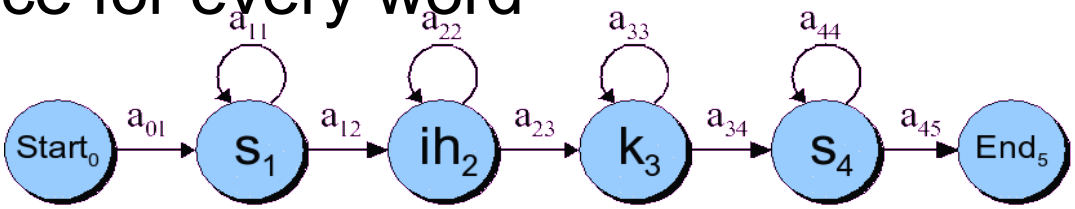
Thanks to Dan Jurafsky for these slides

HMMs for speech

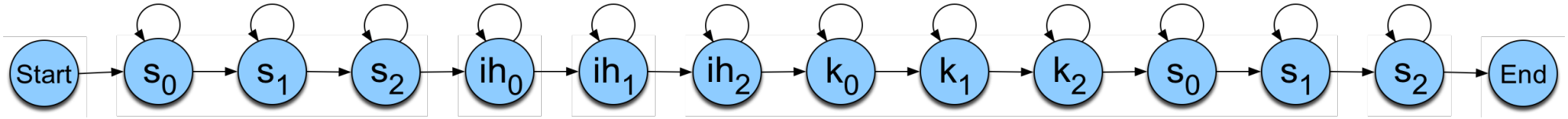
- Dictionary

  SIX S IH K S

- State sequence for every word

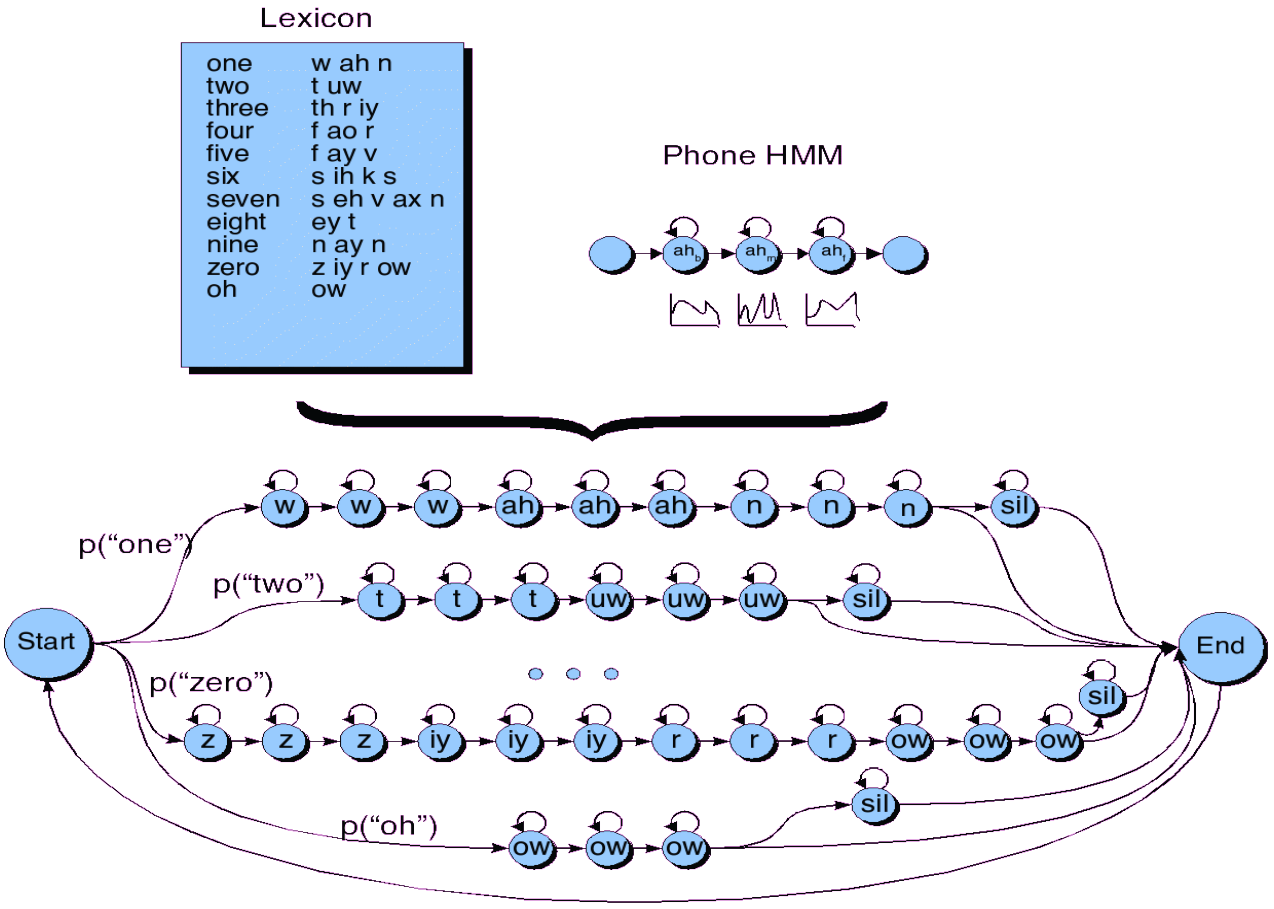


- Each phone has 3 subphones



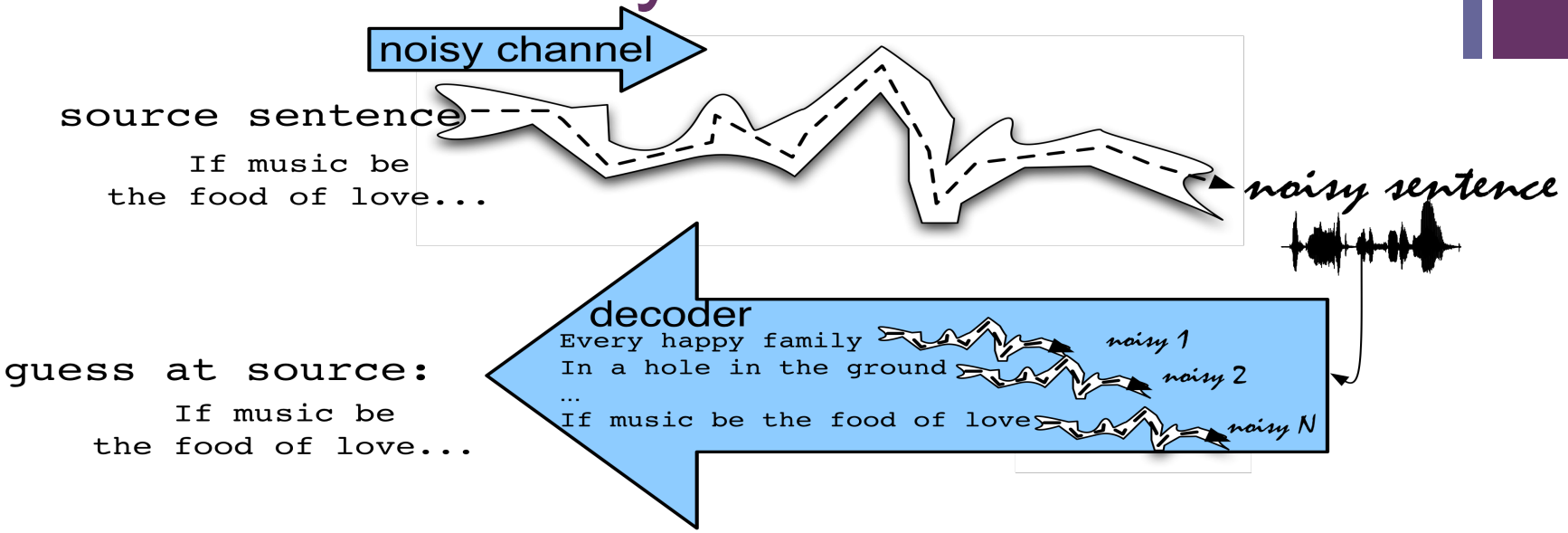
Thanks to Dan Jurafsky for these slides

HMM for digit recognition task



Thanks to Dan Jurafsky for these slides

Back to the Noisy Channel Model



- Search through space of all possible sentences
  - **Defined by the HMM**
- Pick the one that is most probable given the waveform.
  - **Based on the transition and output probabilities in the HMM**

Thanks to Dan Jurafsky for these slides

The Noisy Channel Model

- What is the most likely sentence out of all sentences in the language L given some acoustic input O?
- Treat acoustic input O as sequence of individual observations
  - $O = o_1,o_2,o_3,\ldots,o_t$
- Define a sentence as a sequence of words:
  - $W = w_1,w_2,w_3,\ldots,w_n$

Thanks to Dan Jurafsky for these slides

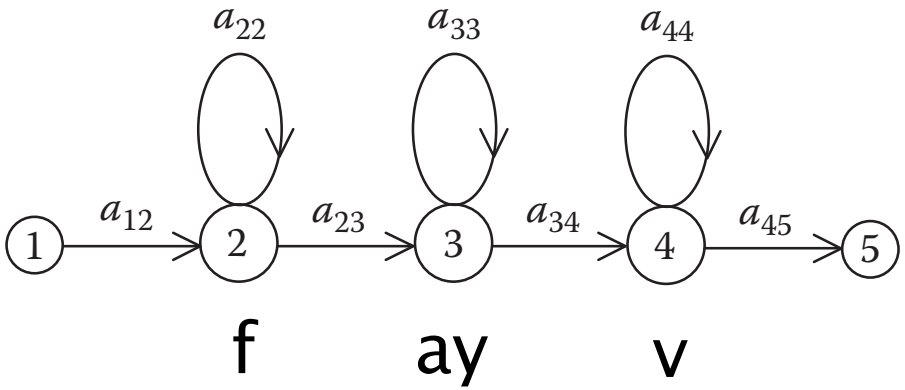
The Evaluation (forward) problem for speech

- The observation sequence O is a series of MFCC vectors
- The hidden states W are the phones and words
- For a given phone/word string W, our job is to evaluate P(O|W)
- Intuition: how likely is the input to have been generated by just that word string W

Thanks to Dan Jurafsky for these slides

HMM for speech: Consider all different paths!

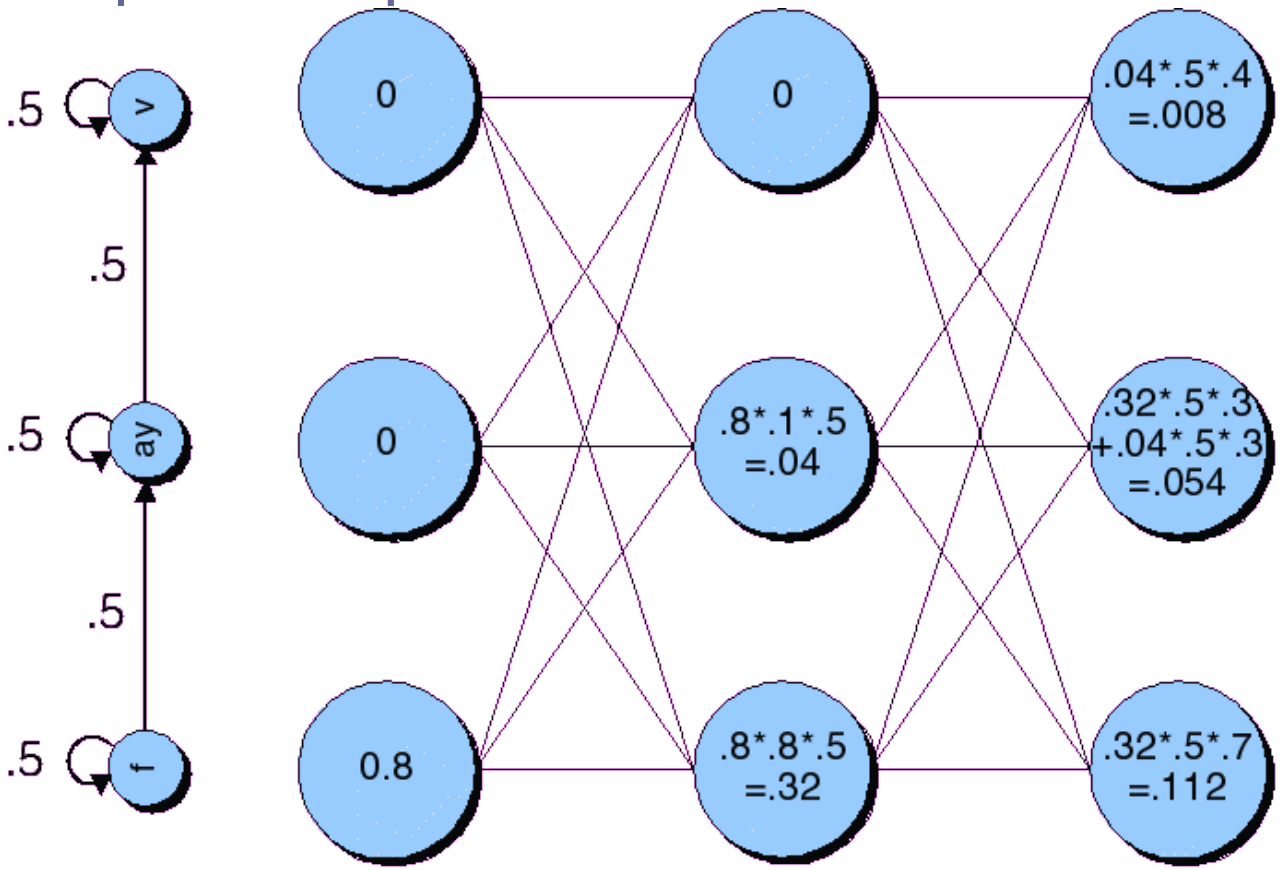
- f ay ay ay ay v v v v
- f f ay ay ay ay v v v
- f f f f ay ay ay ay v
- f f ay ay ay ay ay ay v
- f f ay ay ay ay ay ay ay ay v
- f f ay v v v v v v v v



Thanks to Dan Jurafsky for these slides

The forward lattice for “five”

Computes all possible paths



Thanks to Dan Jurafsky for these slides

+ Forward trellis for “five”

states										
	V	0.0000	0.0000	0.0080	0.0093	0.0114	0.0070	0.0069	0.0036	0.0021	0.0011
	AY	0.0000	0.0400	0.0540	0.0664	0.0355	0.0160	0.0051	0.0016	0.0004	0.0001
	F	0.8000	0.3200	0.1120	0.0224	0.0045	0.0009	0.0002	0.0001	0.0000	0.0000
		1	2	3	4	5	6	7	8	9	10
		time									

“Emission” probabilities (observation likelihood for the observation *o* at each frame)

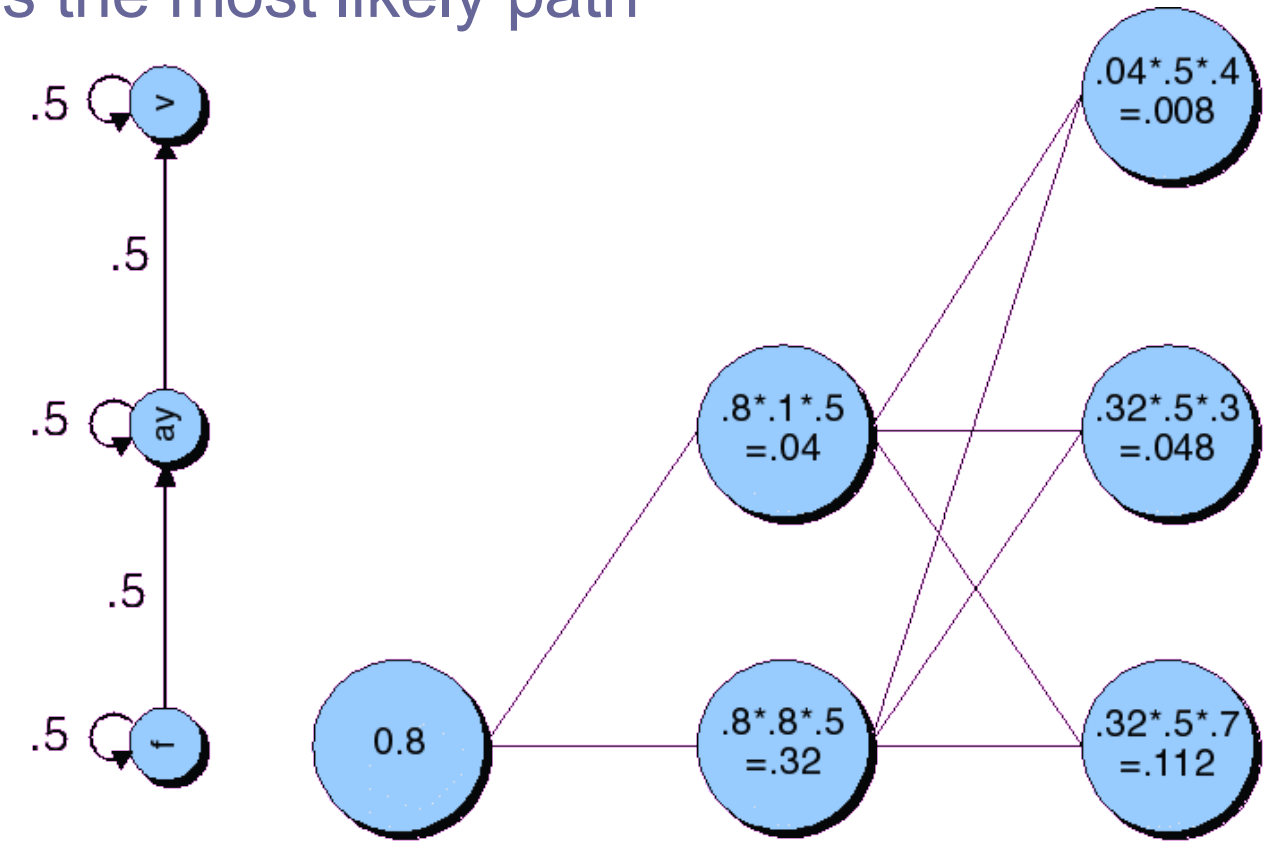
f	0.8	0.8	0.7	0.4	0.4	0.4	0.5	0.5	0.5	0.5
ay	0.1	0.1	0.3	0.8	0.8	0.8	0.8	0.6	0.5	0.4
v	0.6	0.6	0.4	0.3	0.3	0.3	0.3	0.6	0.8	0.9
p	0.4	0.4	0.2	0.1	0.1	0.1	0.1	0.1	0.3	0.3
iy	0.1	0.1	0.3	0.6	0.6	0.6	0.5	0.5	0.5	0.4

“Transition” probabilities

	F	AY	V
F	0.5	0.5	0.0
AY	0.0	0.5	0.5
V	0.0	0.0	0.5

Viterbi trellis for “five”

Computes the most likely path



Thanks to Dan Jurafsky for these slides

+ Viterbi trellis for "five"

states										
	V	0.0000		0.064	0.0168	0.00672	0.002688	0.0010752	0.00086016	0.000344064	0.000137626
	AY	0.0000	0.04	0.048	0.0448	0.01792	0.007168	0.0028672	0.00086016	0.00021504	6.88128E-05
	F	0.8000	0.32	0.112	0.0224	0.00896	0.003584	0.001792	0.0007168	0.00021504	8.6016e-05
		1	2	3	4	5	6	7	8	9	10
		time									

"Emission" (observation) probabilities

f	0.8	0.8	0.7	0.4	0.4	0.4	0.5	0.5	0.5	0.5
ay	0.1	0.1	0.3	0.8	0.8	0.8	0.8	0.6	0.5	0.4
v	0.6	0.6	0.4	0.3	0.3	0.3	0.3	0.6	0.8	0.9
p	0.4	0.4	0.2	0.1	0.1	0.1	0.1	0.1	0.3	0.3
iy	0.1	0.1	0.3	0.6	0.6	0.6	0.5	0.5	0.5	0.4

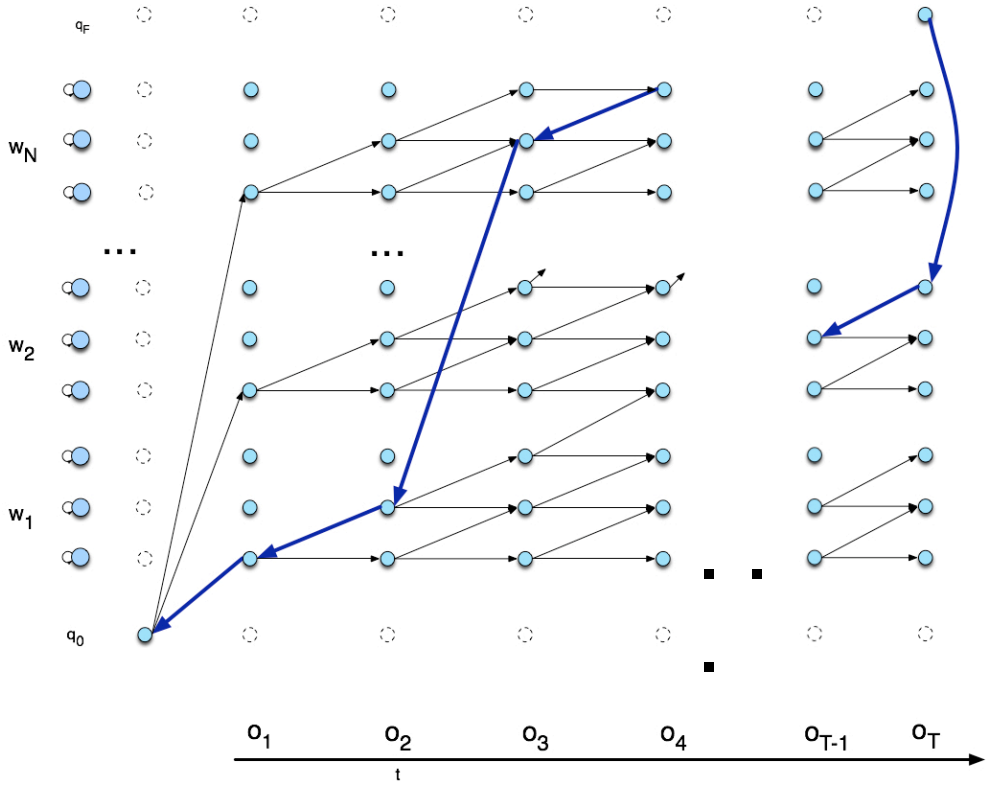
Transition probabilities

	F	AY	V
F	0.5	0.5	0.0
AY	0.0	0.5	0.5
V	0.0	0.0	0.5

Same as for the forward trellis

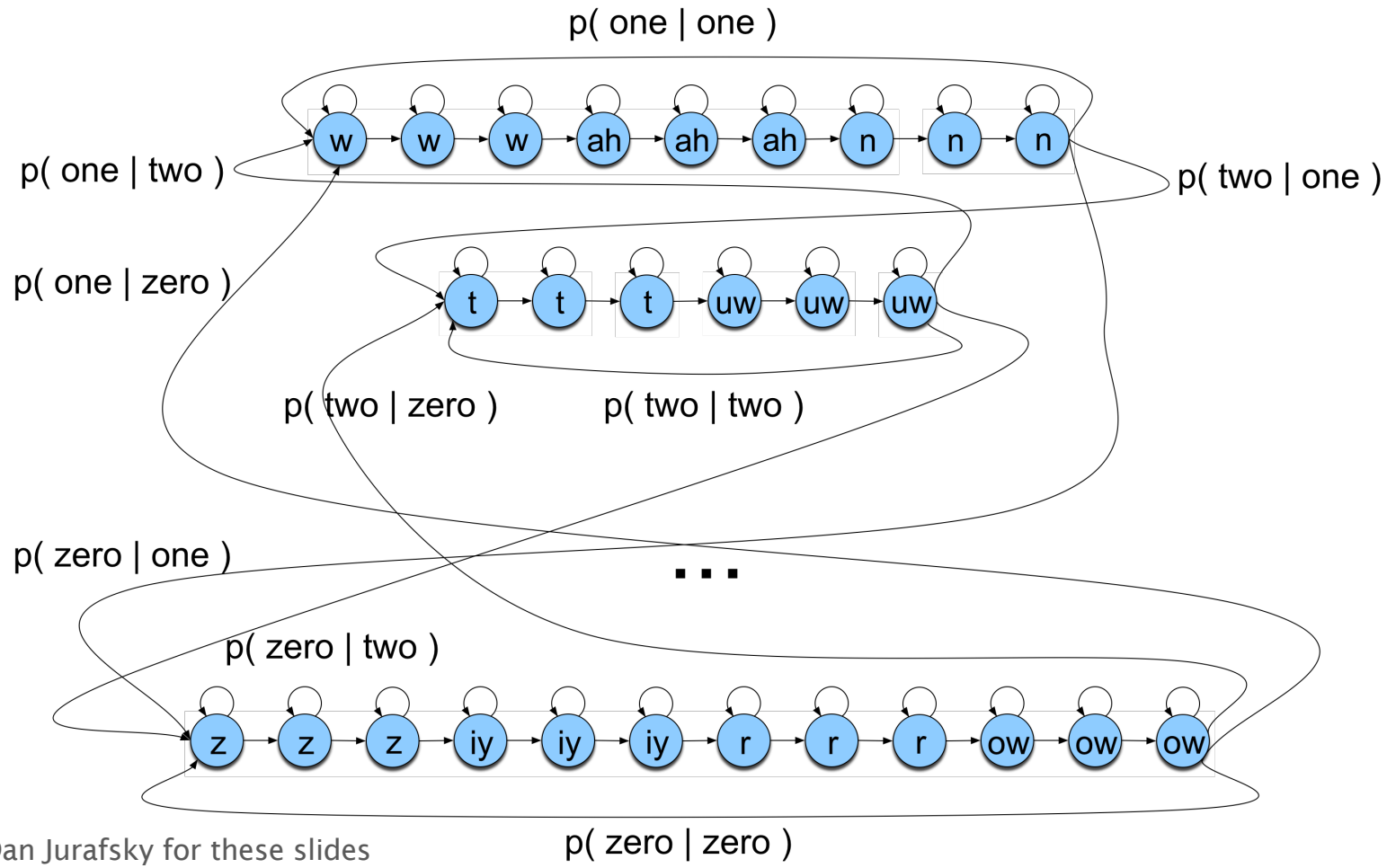
Thanks to Dan Jurafsky for these slides

+ Viterbi backtrace



Thanks to Dan Jurafsky for these slides

+ Search space with bigrams



Thanks to Dan Jurafsky for these slides

“Output symbols”: the great hack

- Markov models are “generative” models: Find the most likely sequence that generates the output symbols
- “Output symbols” = Observations
  - What we normally think of as input