

Automatic Speech Recognition: An Overview

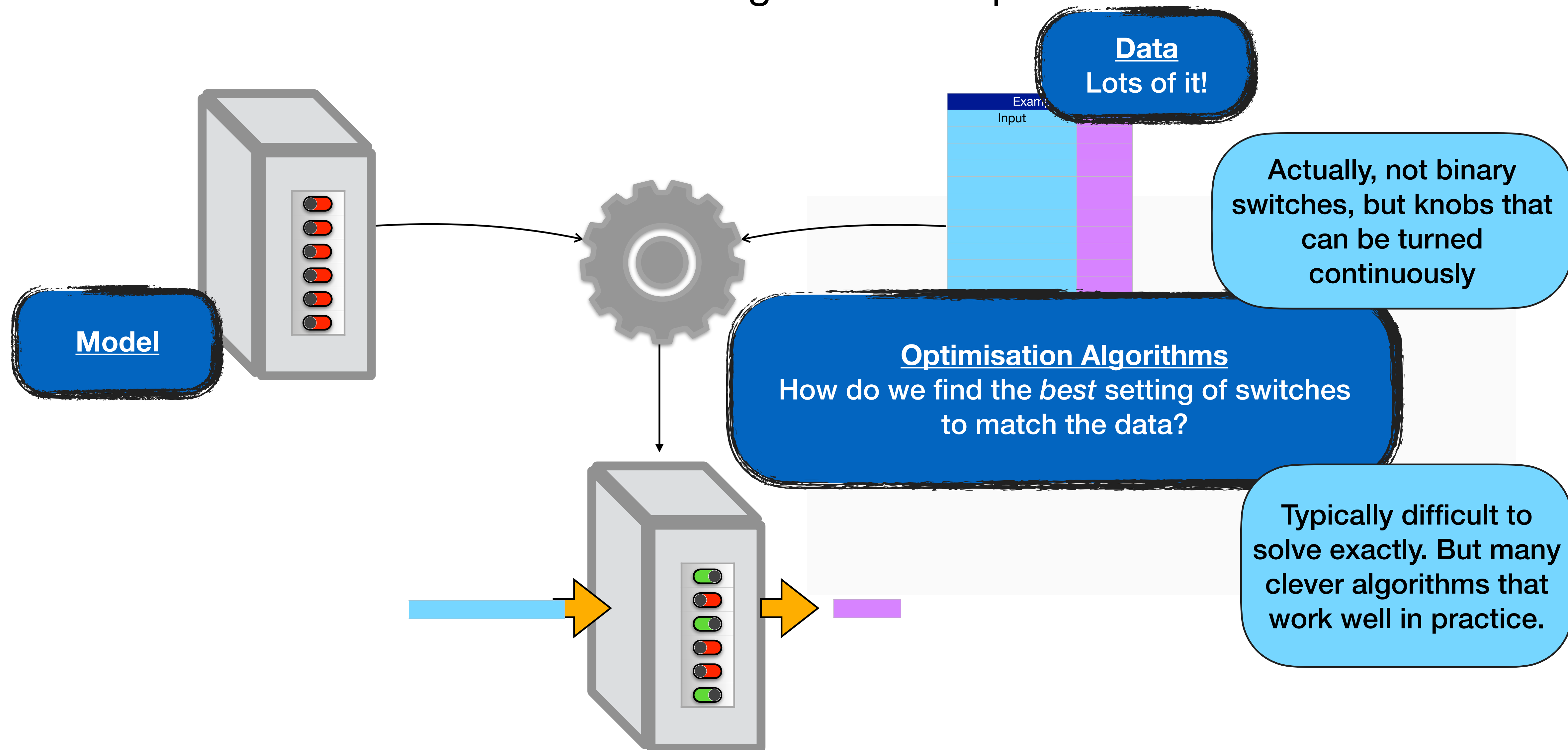
Preethi Jyothi, IIT Bombay

NCM-CEP AI/ML Course, Feb 23, 2023







Components of Machine Learning

- Optimization is integral to machine learning
- Mathematics is essential to understanding what this optimization entails



Many Courses on Mathematics for ML



About Swayam | All Courses | [SIGN-IN / RE](#)

Bro

Math

Le

Math

appl

142

Develop

Math

and

Learn

imple

4.3 ★★

Created

Last

W

✓


✓

Courses >

Essential Mathematics for Machine Learning

By Prof. Sanjeev Kumar, Prof. S. K. Gupta | IIT Roorkee

Learners enrolled: 5023



Summary

Course Status :	Completed
Course Type :	Elective
Duration :	12 weeks
Start Date :	26 Jul 2021
End Date :	15 Oct 2021
Exam Date :	24 Oct 2021 IST
Enrollment Ends :	09 Aug 2021
Category :	◦ Mathematics
Credit Points :	3

Why is Probability Important for ML Speech Processing?

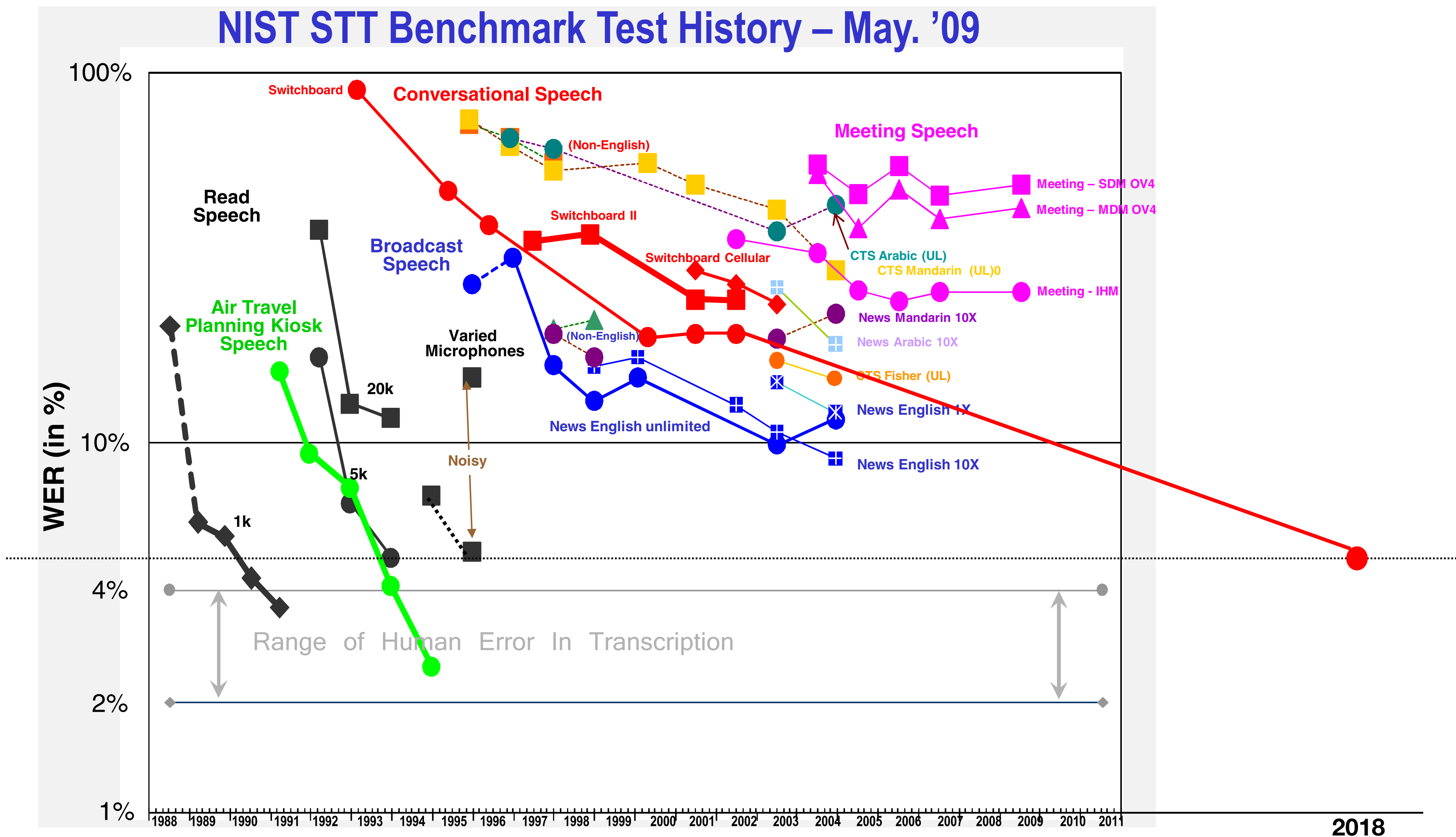
Many sources of variability when dealing with a task

- Noisy data
- Unpredictable environment
- Incomplete data
- Model is incomplete
- New domain

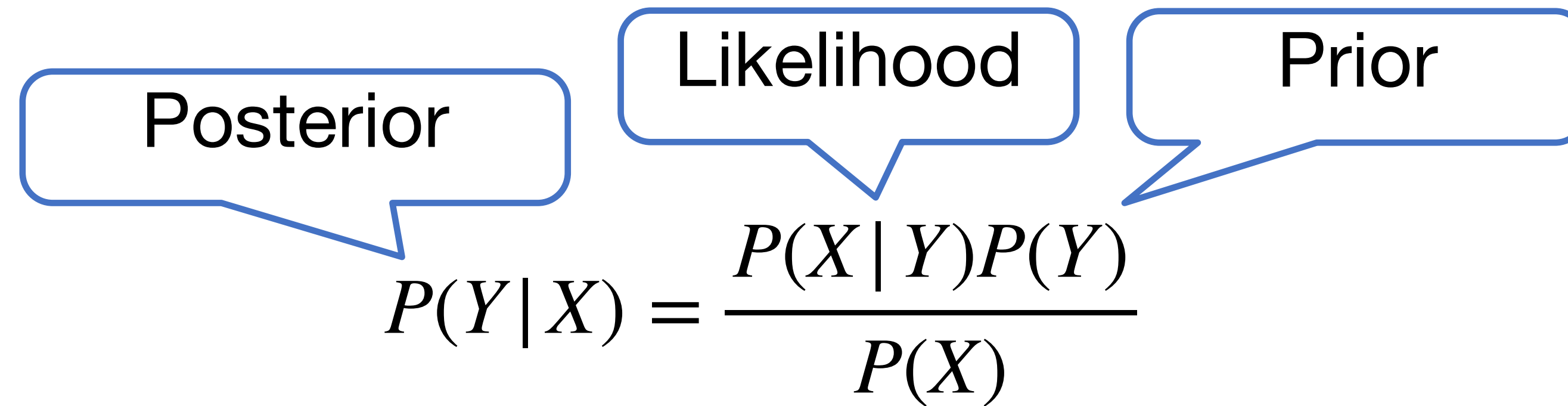
Automatic Speech Recognition (ASR)

- Problem statement: Transform a spoken utterance into a sequence of tokens (words, syllables, phonemes, characters)
- Many downstream applications of ASR. Examples:
 - Spoken language understanding
 - Spoken translation
 - Intelligent video editing
 - ASR from brain signals
- Speech demonstrates variabilities at multiple levels: Speaker style, accents, room acoustics, microphone properties, etc.

ASR Performance Over the Years



Recall Bayes' Theorem



The diagram shows the Bayes' Theorem equation with three callout boxes. The 'Posterior' box points to the left side of the equation, the 'Likelihood' box points to the numerator, and the 'Prior' box points to the second part of the numerator.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Of great interest in machine learning, where we often want to infer about *unobserved* random variables given we have observations for other random variables

Employed in traditional/statistical ASR systems

Statistical Speech Recognition

Pioneer of ASR technology, Fred Jelinek (1932 - 2010): Cast ASR as a channel coding problem

Let \mathbf{O} be a sequence of acoustic features corresponding to a speech signal. That is, $\mathbf{O} = \{O_1, \dots, O_T\}$, where $O_i \in \mathbb{R}^d$ refers to a d -dimensional acoustic feature vector and T is the length of the sequence.

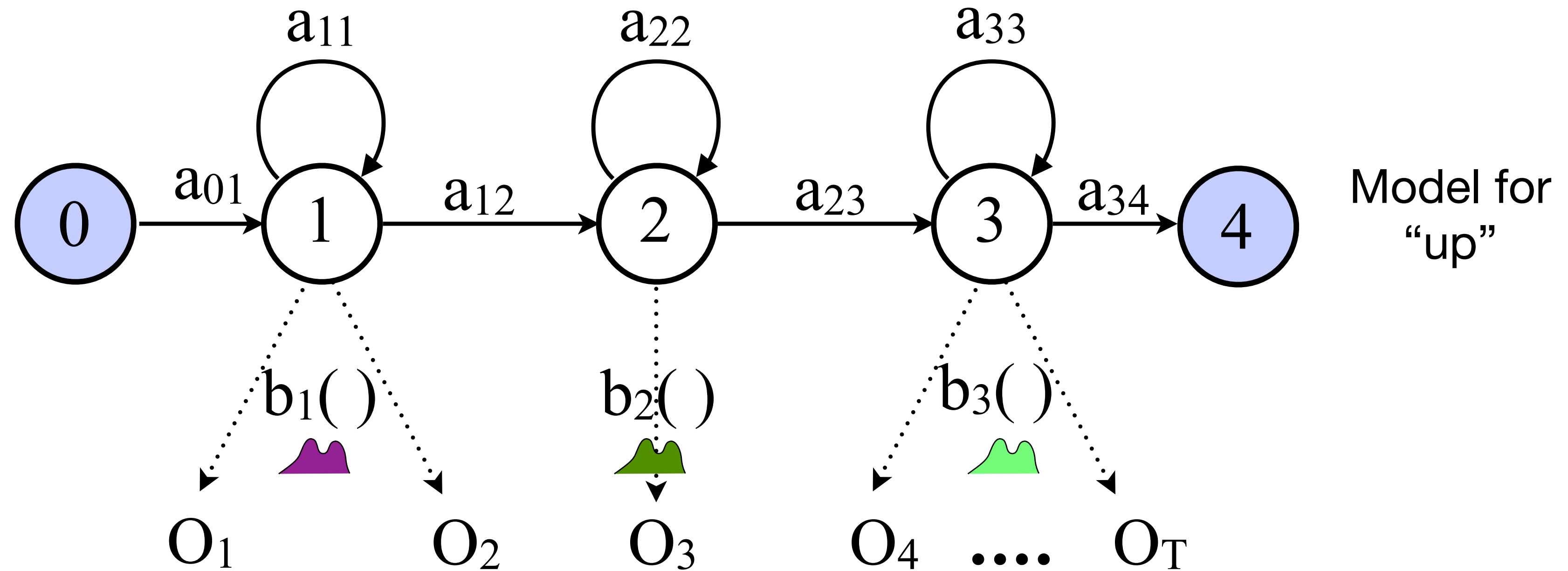
Let \mathbf{W} denote a word sequence. An ASR decoder solves the foll. problem:

$$\begin{aligned}\mathbf{W}^* &= \arg \max_W \Pr(\mathbf{W} | \mathbf{O}) \\ &= \arg \max_W \Pr(\mathbf{O} | \mathbf{W}) \Pr(\mathbf{W})\end{aligned}$$

Simple Example of Isolated Word ASR

- Task: Recognize utterances which consist of speakers saying either “up” or “down” or “left” or “right” per recording.
- Vocabulary: Four words, “up”, “down”, “left”, “right”
- Data splits
 - Training data: 30 utterances from different speakers containing “up”, “down”, “left”, “right”
 - Test data: 20 utterances from a disjoint set of speakers
- Let’s parameterize $\Pr_{\theta}(\mathbf{O} \mid \mathbf{W})$ using a hidden Markov model with parameters θ

Word-based Acoustic Model

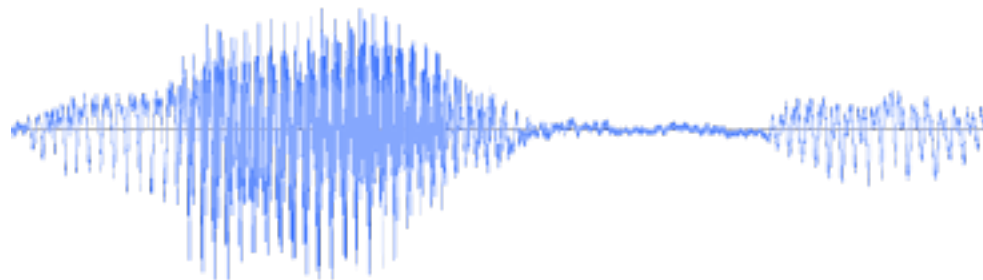
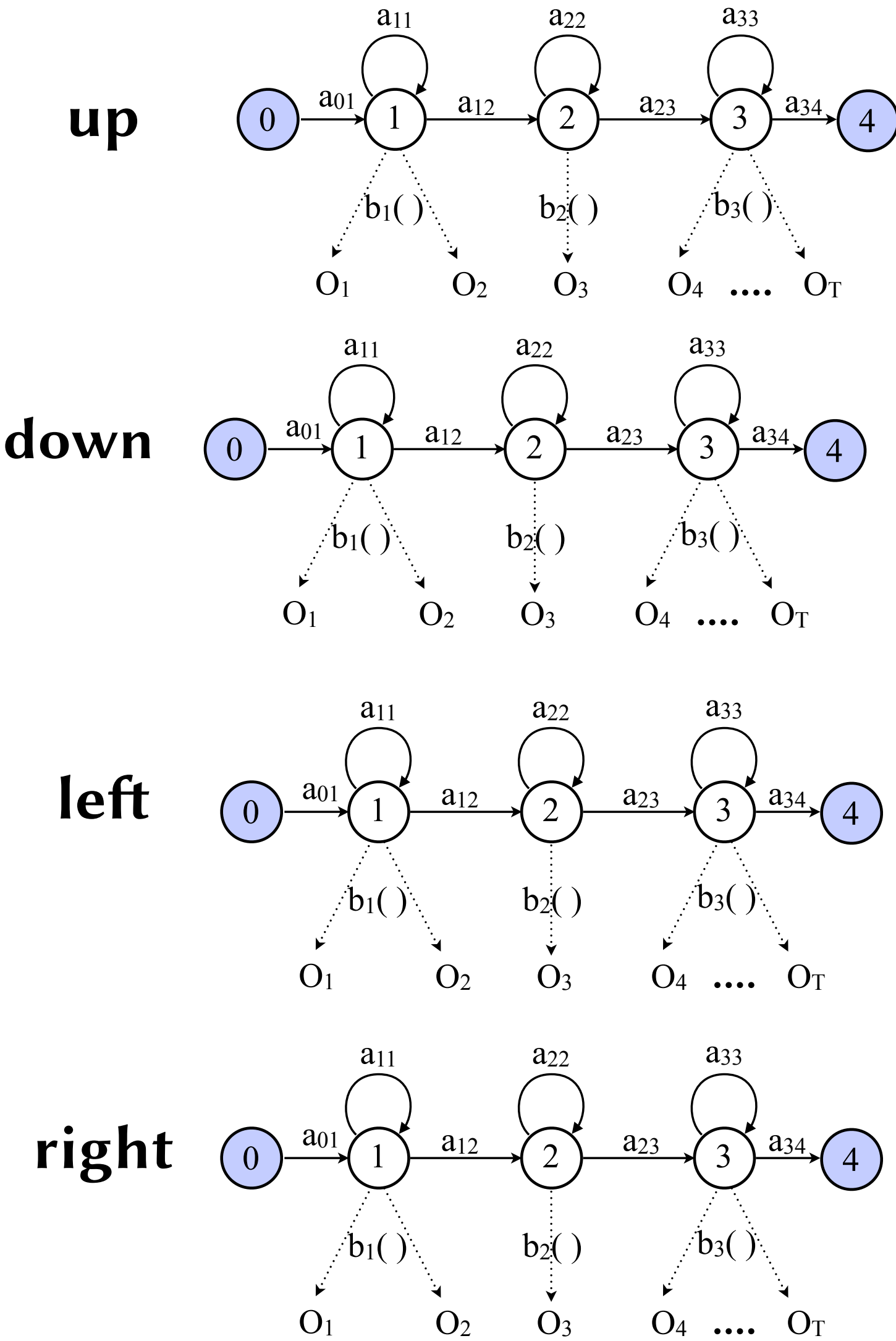


$a_{ij} \rightarrow$ Transition probabilities going from state i to state j
 $b_j(O_i) \rightarrow$ Probability of generating O_i from state j

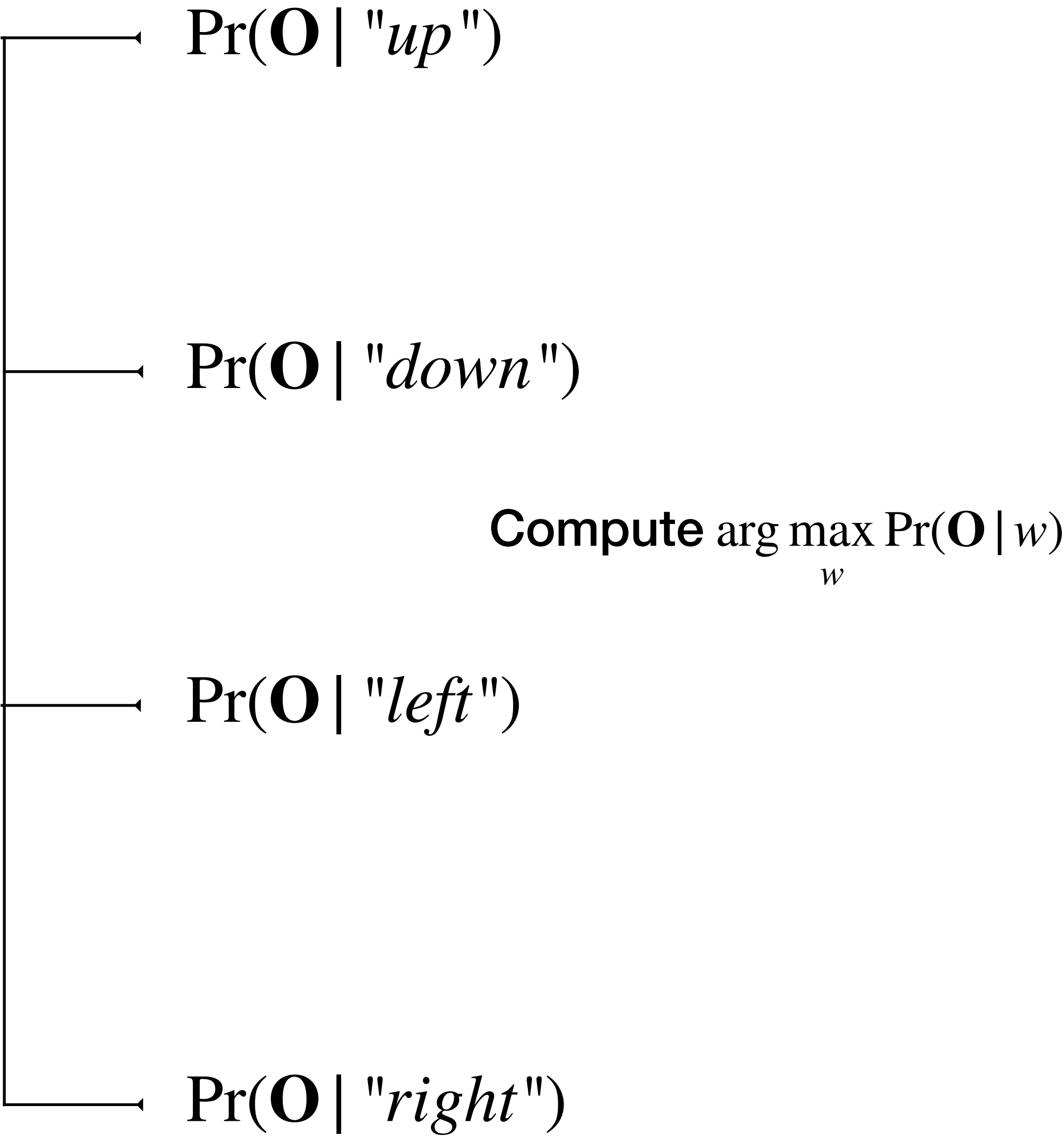
} θ

Compute $\Pr_{\theta}(\mathbf{O} \mid \text{"up"}) = \sum_{\mathbf{Q}} \Pr_{\theta}(\mathbf{O}, \mathbf{Q} \mid \text{"up"})$ where \mathbf{Q} is the hidden-state sequence

Isolated Word Recognition

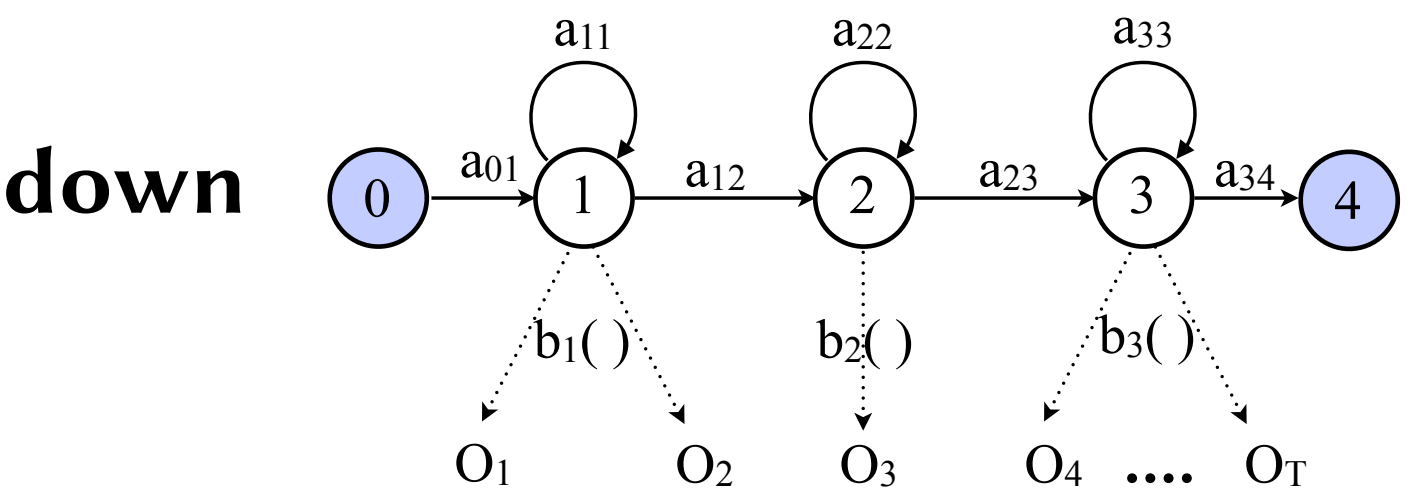
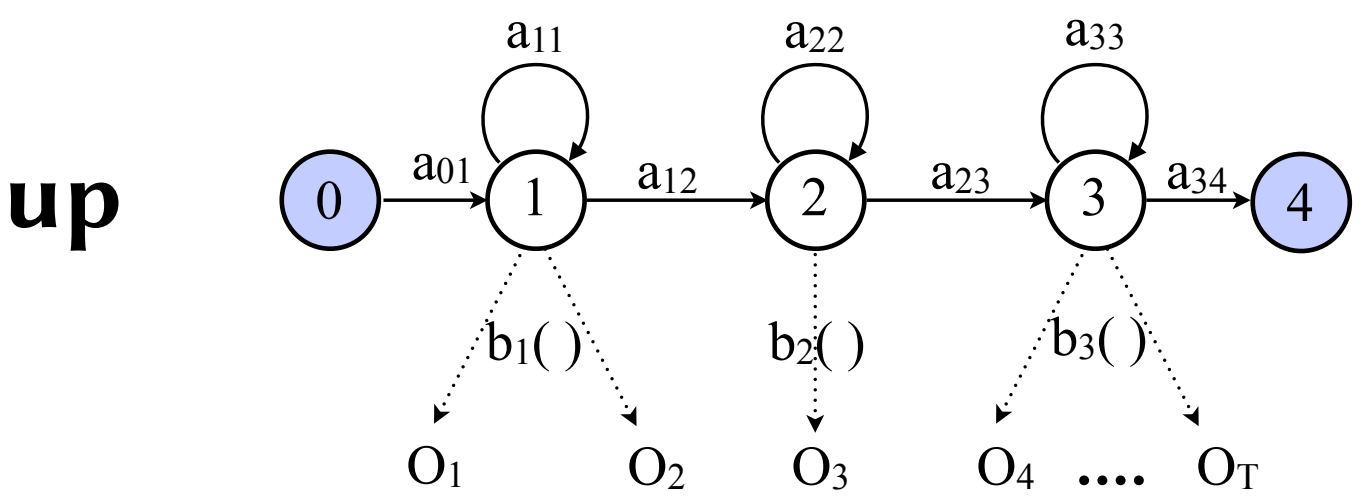


acoustic
features
 \mathbf{O}



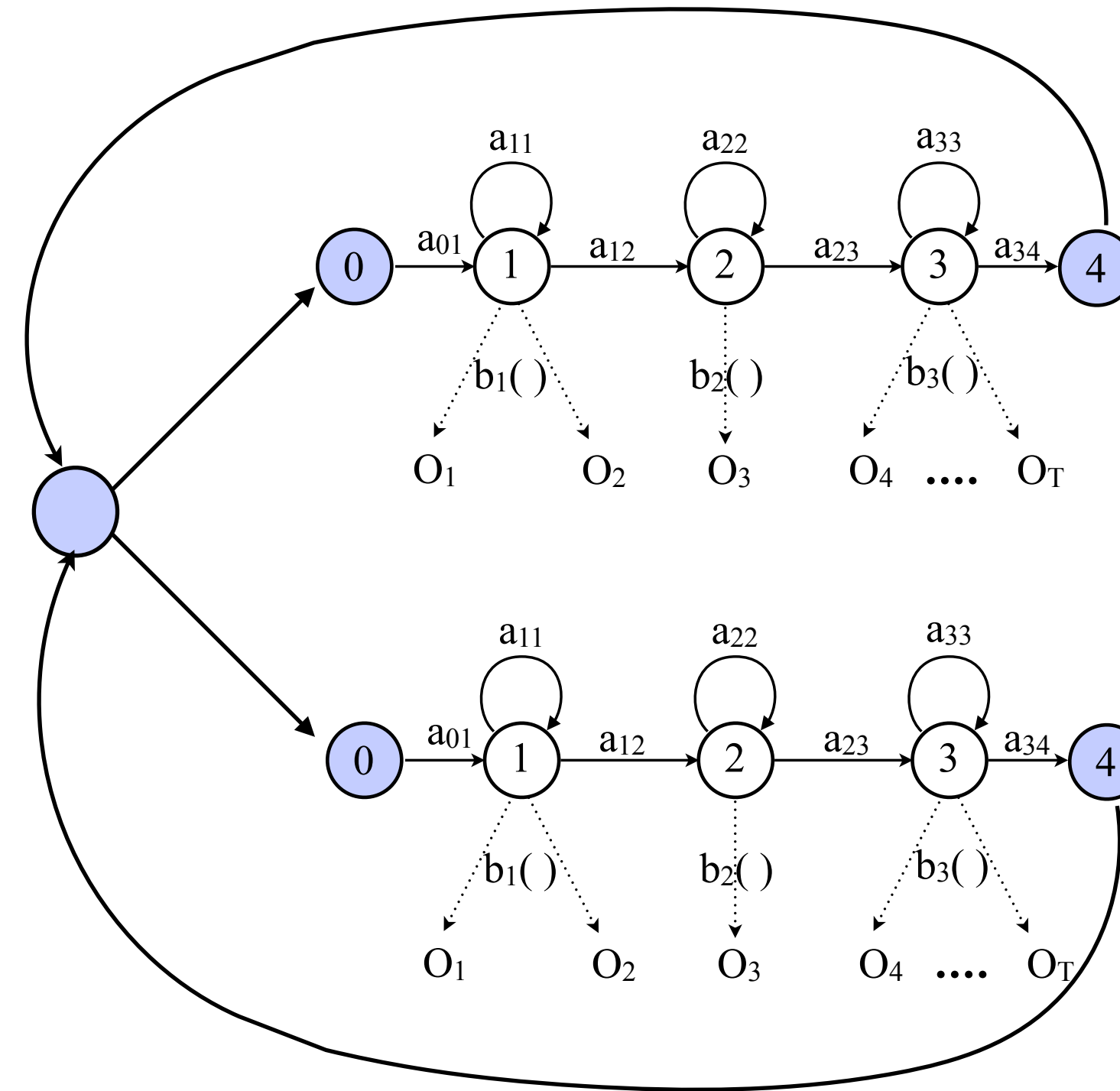
Small Tweak

- Task: Recognize utterances which consist of speakers saying either “up” or “down” **multiple times** per recording.



Small Tweak

- Task: Recognize utterances which consist of speakers saying either “up” or “down” **multiple times** per recording.



Search within this graph

Small Vocabulary ASR

- Task: Recognize utterances which consist of speakers saying one of 1000 words **multiple times** per recording.
- Not scalable anymore to use words as speech units
- Model using phonemes instead of words as individual speech units
 - Phonemes are subword speech units analogous to the written alphabet for text
- What's an advantage of using phonemes over entire words?
Hint: Think of words with zero coverage in the training data.

Statistical Speech Recognition

Let \mathbf{O} be a sequence of acoustic features corresponding to a speech signal. That is, $\mathbf{O} = \{O_1, \dots, O_T\}$, where $O_i \in \mathbb{R}^d$ refers to a d -dimensional acoustic feature vector and T is the length of the sequence.

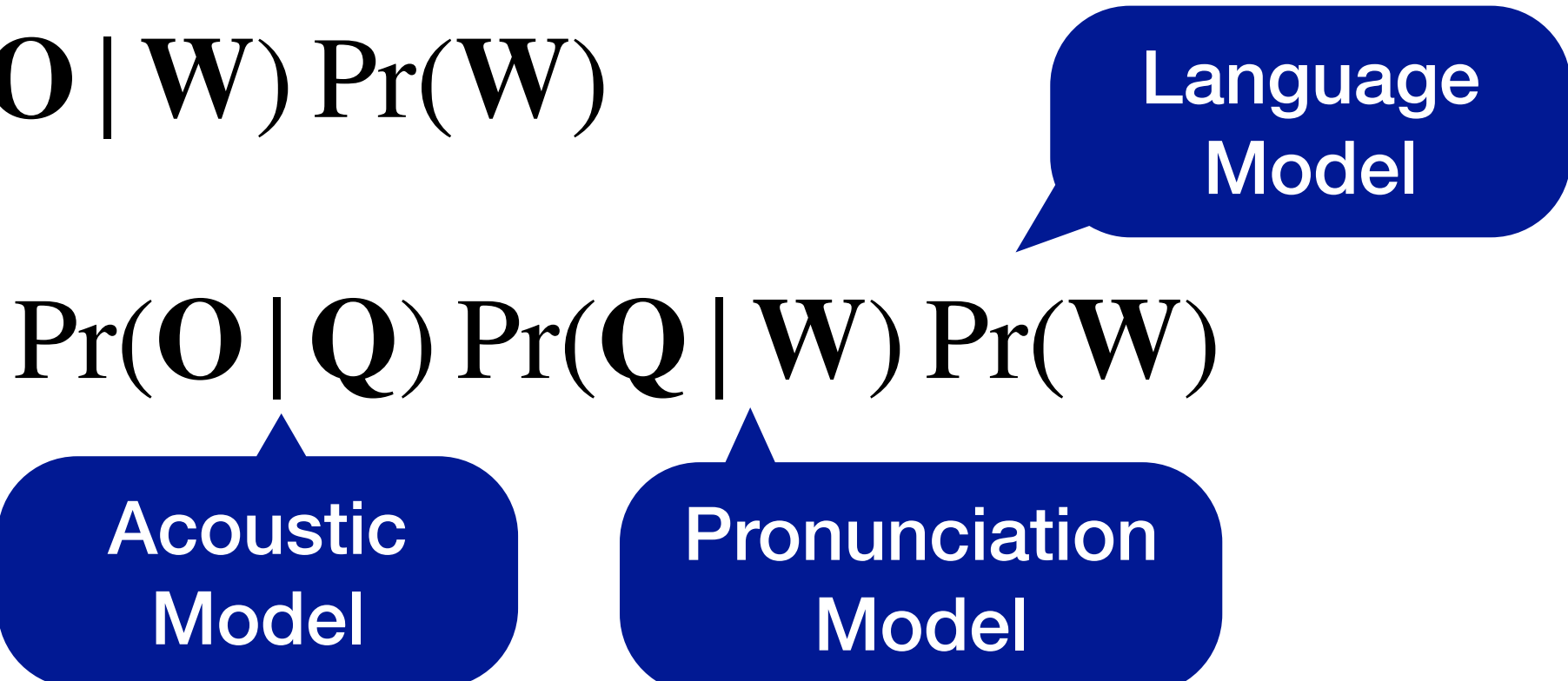
Let \mathbf{W} denote a word sequence. An ASR decoder solves the foll. problem:

$$\begin{aligned}\mathbf{W}^* &= \arg \max_W \Pr(\mathbf{W} | \mathbf{O}) \\ &= \arg \max_W \Pr(\mathbf{O} | \mathbf{W}) \Pr(\mathbf{W})\end{aligned}$$

Statistical Speech Recognition

Let \mathbf{O} be a sequence of acoustic features corresponding to a speech signal. That is, $\mathbf{O} = \{O_1, \dots, O_T\}$, where $O_i \in \mathbb{R}^d$ refers to a d -dimensional acoustic feature vector and T is the length of the sequence.

Let \mathbf{W} denote a word sequence. An ASR decoder solves the foll. problem:

$$\begin{aligned}\mathbf{W}^* &= \arg \max_W \Pr(\mathbf{W} | \mathbf{O}) \\ &= \arg \max_W \Pr(\mathbf{O} | \mathbf{W}) \Pr(\mathbf{W}) \\ &\approx \arg \max_W \sum_Q \Pr(\mathbf{O} | \mathbf{Q}) \Pr(\mathbf{Q} | \mathbf{W}) \Pr(\mathbf{W})\end{aligned}$$


The diagram illustrates the decomposition of the ASR equation into three models. Three blue callout boxes are present: 'Acoustic Model' points to $\Pr(\mathbf{O} | \mathbf{Q})$, 'Pronunciation Model' points to $\Pr(\mathbf{Q} | \mathbf{W})$, and 'Language Model' points to $\Pr(\mathbf{W})$.

Language Models

A language model can be characterized by a probability distribution P that:

- provides information about word reordering

$$P(\text{"she class taught a"}) < P(\text{"she taught a class"})$$

- provides information about the most likely next word

$$P(\text{"she taught a class"}) > P(\text{"she taught a speech"})$$

Why do we need a language model?

Consider recognizing speech corresponding to the following two sentences:

“i know you”
“eye no you”

Statistical Language Models

- Given a word sequence, $W = \{w_1, \dots, w_n\}$, what is $P(W)$?

Decompose $P(W)$ using the chain rule:

$$P(w_1, \dots, w_n) = P(w_1)P(w_2 | w_1)P(w_3 | w_1, w_2) \dots P(w_n | w_1, \dots, w_{n-1})$$

- Ngram models: Use a limited memory of previous word history. Specifically, only the last $m - 1$ words are included in an m gram model.
- Bigram models:

$$\Pr(w_1, w_2, \dots, w_{n-1}, w_n) \cong \Pr(w_2 | w_1, \langle s \rangle) \Pr(w_3 | w_1, w_2) \dots \Pr(w_n | w_{n-2}, w_{n-1})$$

- How do we estimate these probabilities? Compute normalized counts
 - E.g. $\Pr(\text{“class”} | \text{“she taught a”}) = \frac{\text{count(“she taught a class”)}}{\text{count(“she taught a”)}}$

count(“she taught a”)

Word Representations

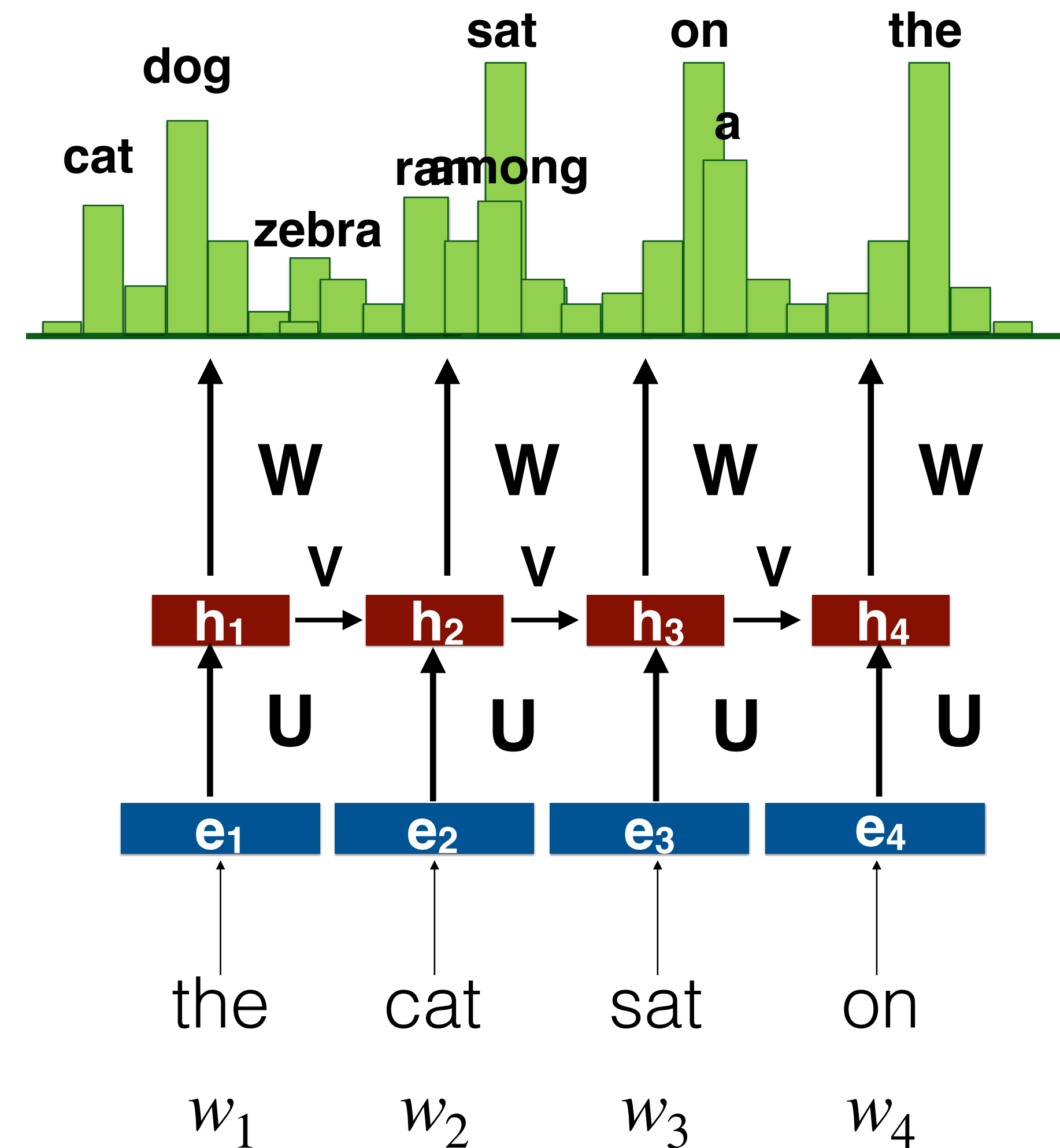
- In n gram models, words are represented in the discrete space involving the vocabulary
 - 1-hot representation: Each word is given an index in $\{1, \dots, V\}$, V is the vocabulary size
- Limits the possibility of truly interpolating probabilities of unseen n grams
- Can we build a representation for words in the continuous space?
 - **Word embeddings:** Each word is associated with a dense vector.
E.g. dog \rightarrow $\{-0.02, -0.37, 0.26, 0.25, -0.11, 0.34\}$
 - Trained using principles based on the “*distributional hypothesis*”:
Firth 1957 summarised this as “*you shall know a word by the company it keeps*”
- Word embeddings capture semantic properties (such as *man* is to *woman* as *boy* is to *girl*) and morphological properties (*glad* is similar to *gladly*).

Recurrent Neural Network (RNN)-based Language Model

output probability:
 $\hat{y}_t = \text{softmax}(\mathbf{W}\mathbf{h}_t + \mathbf{b}')$

hidden layer:
 $\mathbf{h}_t = \tanh(\mathbf{U}\mathbf{e}_t + \mathbf{V}\mathbf{h}_{t-1} + \mathbf{b})$

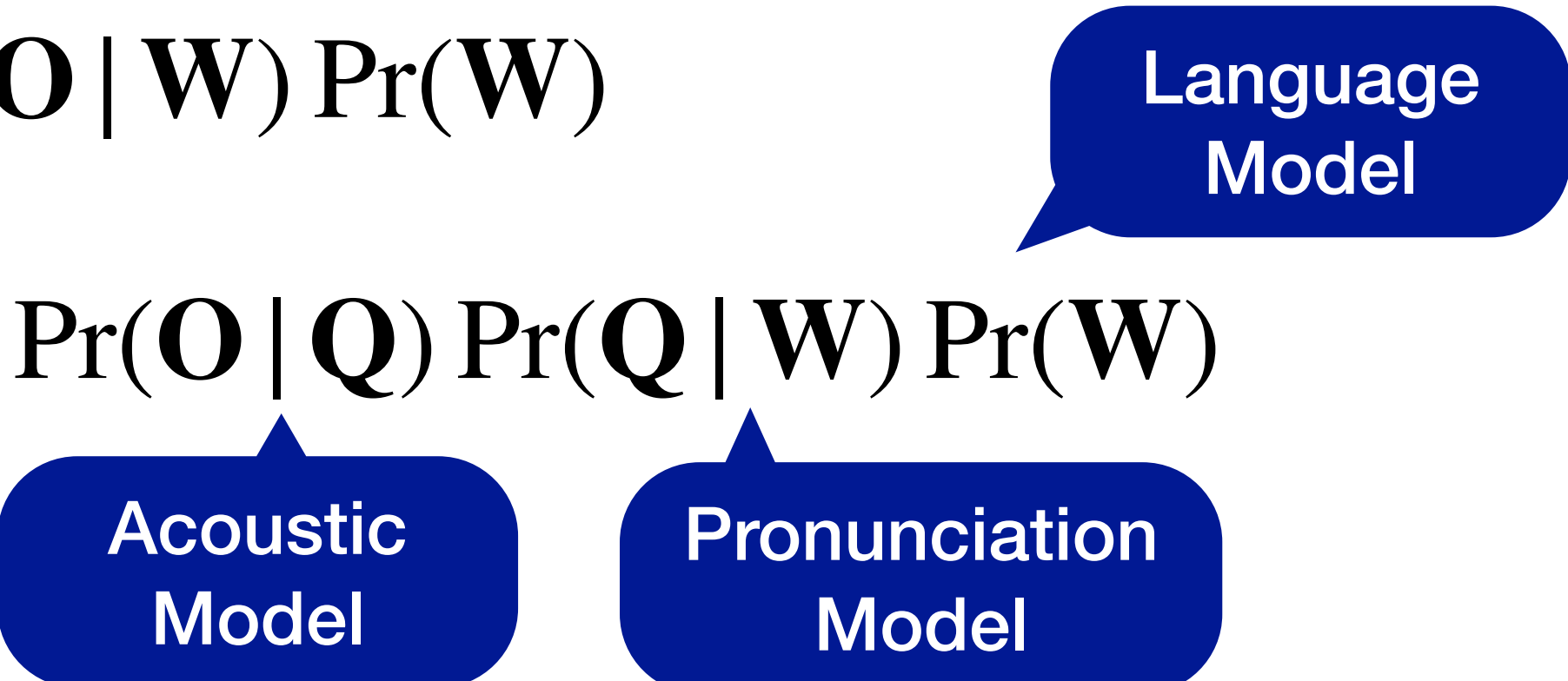
words



Statistical Speech Recognition

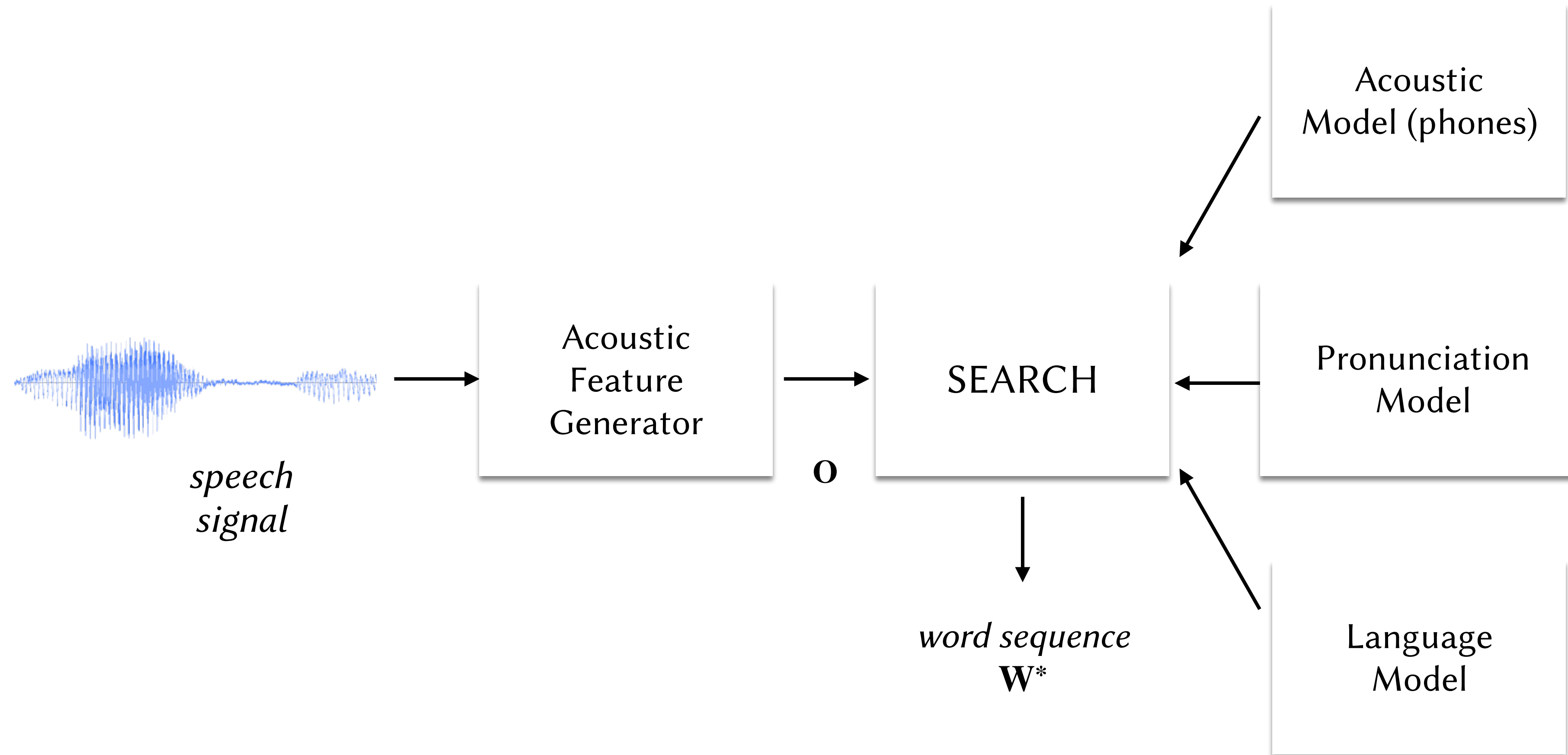
Let \mathbf{O} be a sequence of acoustic features corresponding to a speech signal. That is, $\mathbf{O} = \{O_1, \dots, O_T\}$, where $O_i \in \mathbb{R}^d$ refers to a d -dimensional acoustic feature vector and T is the length of the sequence.

Let \mathbf{W} denote a word sequence. An ASR decoder solves the foll. problem:

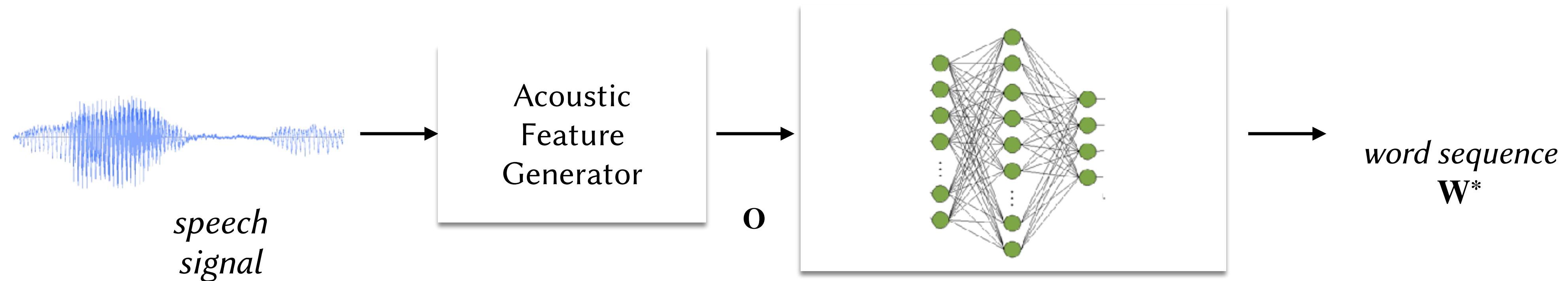
$$\begin{aligned}\mathbf{W}^* &= \arg \max_W \Pr(\mathbf{W} | \mathbf{O}) \\ &= \arg \max_W \Pr(\mathbf{O} | \mathbf{W}) \Pr(\mathbf{W}) \\ &\approx \arg \max_W \sum_Q \Pr(\mathbf{O} | \mathbf{Q}) \Pr(\mathbf{Q} | \mathbf{W}) \Pr(\mathbf{W})\end{aligned}$$


The diagram illustrates the decomposition of the ASR equation into three models. Three blue callout boxes are present: 'Acoustic Model' points to $\Pr(\mathbf{O} | \mathbf{Q})$, 'Pronunciation Model' points to $\Pr(\mathbf{Q} | \mathbf{W})$, and 'Language Model' points to $\Pr(\mathbf{W})$.

Architecture of a Cascaded ASR System



Cascaded ASR \Rightarrow End-to-end ASR



Single end-to-end model that directly learns a mapping from speech to text

End-to-End ASR Systems

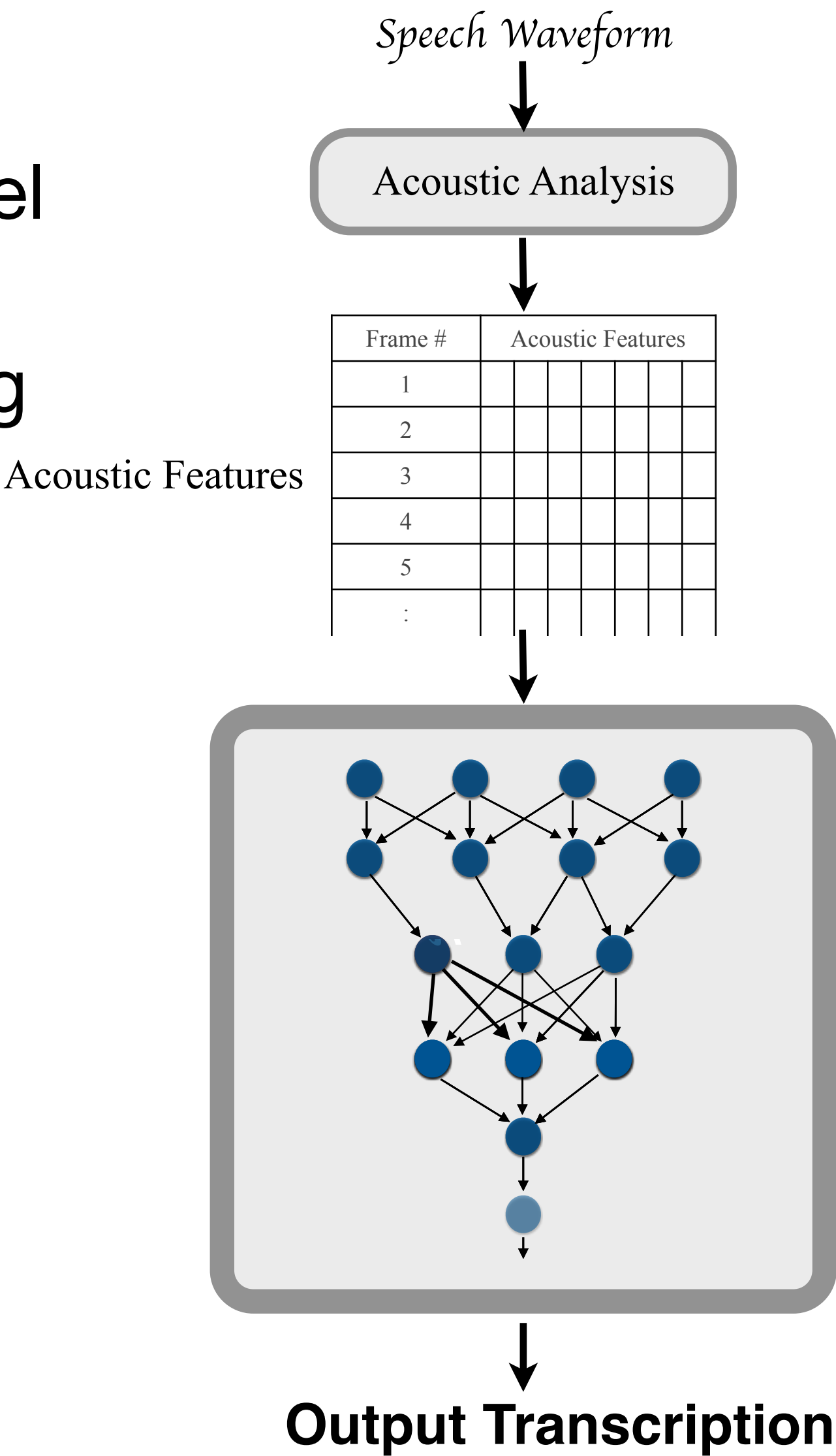
- All components trained jointly as a single end-to-end model
- Trained using pairs of speech clips and their corresponding text transcripts
- End-to-end models, with sufficient data, outperform conventional ASR systems

	dev	test
DNN-HMM	4.0	4.4
E2E (Attention)	4.7	4.8

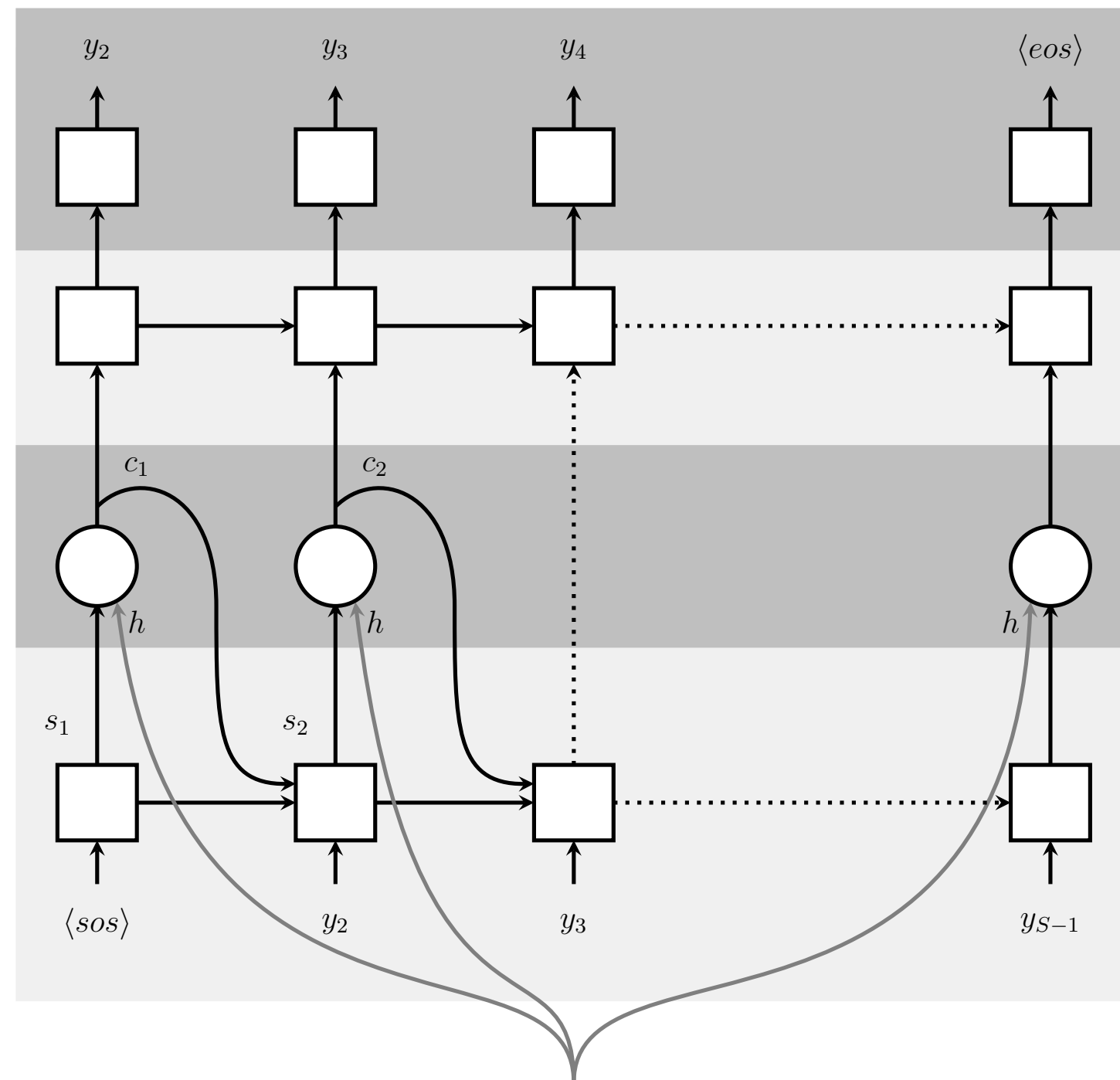
Librispeech-960

	dev	test
DNN-HMM	5.0	5.8
E2E (Attention)	14.7	14.7

Librispeech-100

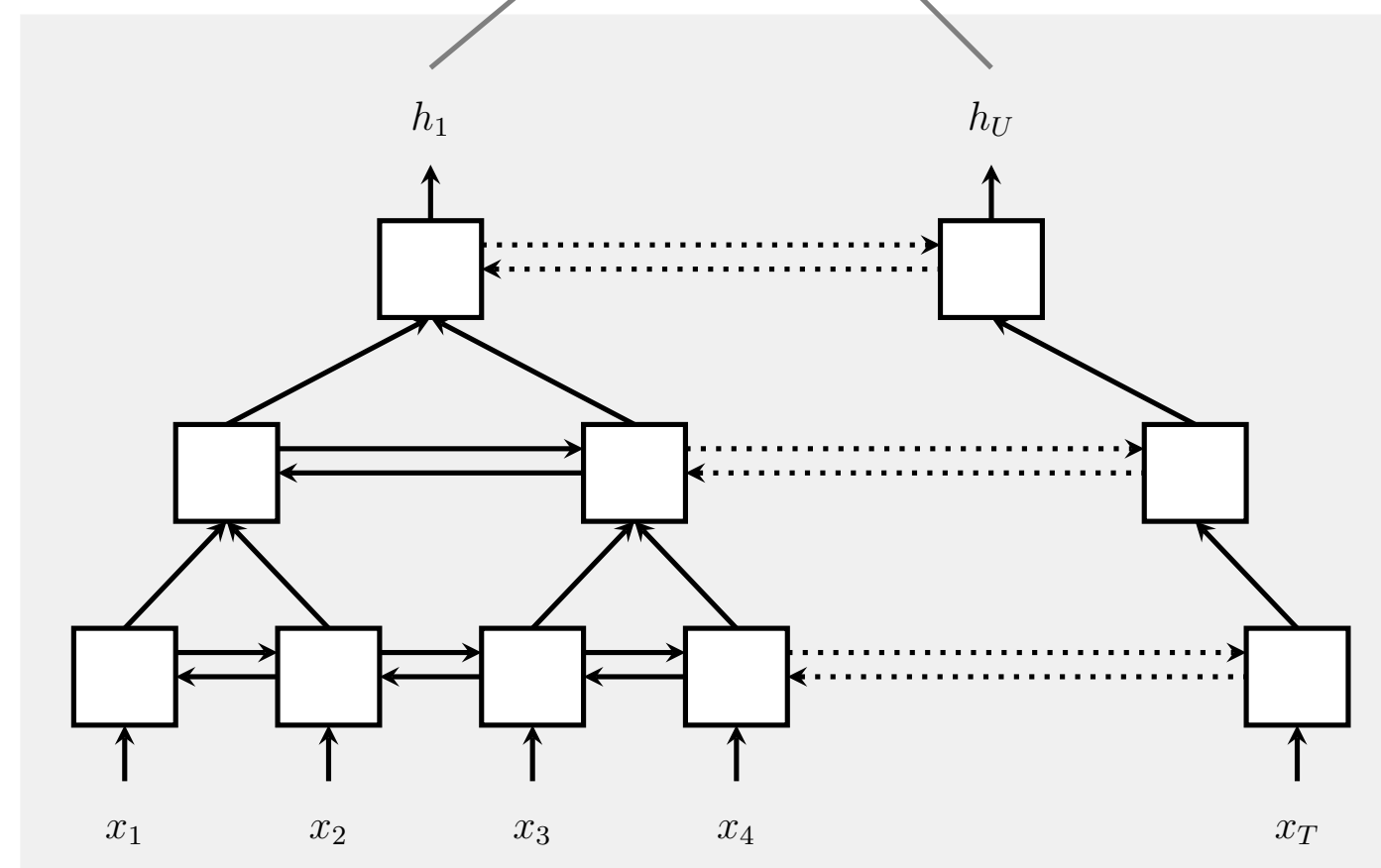


Speller



$h = (h_1, \dots, h_U)$

Listener



The LAS Model

- The Listen, Attend & Spell (LAS) architecture is a sequence-to-sequence model consisting of
- a Listener (Listen): An acoustic model encoder. Deep BLSTMs with a pyramidal structure: reduces the time resolution by a factor of 2 in each layer.
- a Speller (AttendAndSpell): An attention-based decoder. Consumes \mathbf{h} and produces a probability distribution over characters.

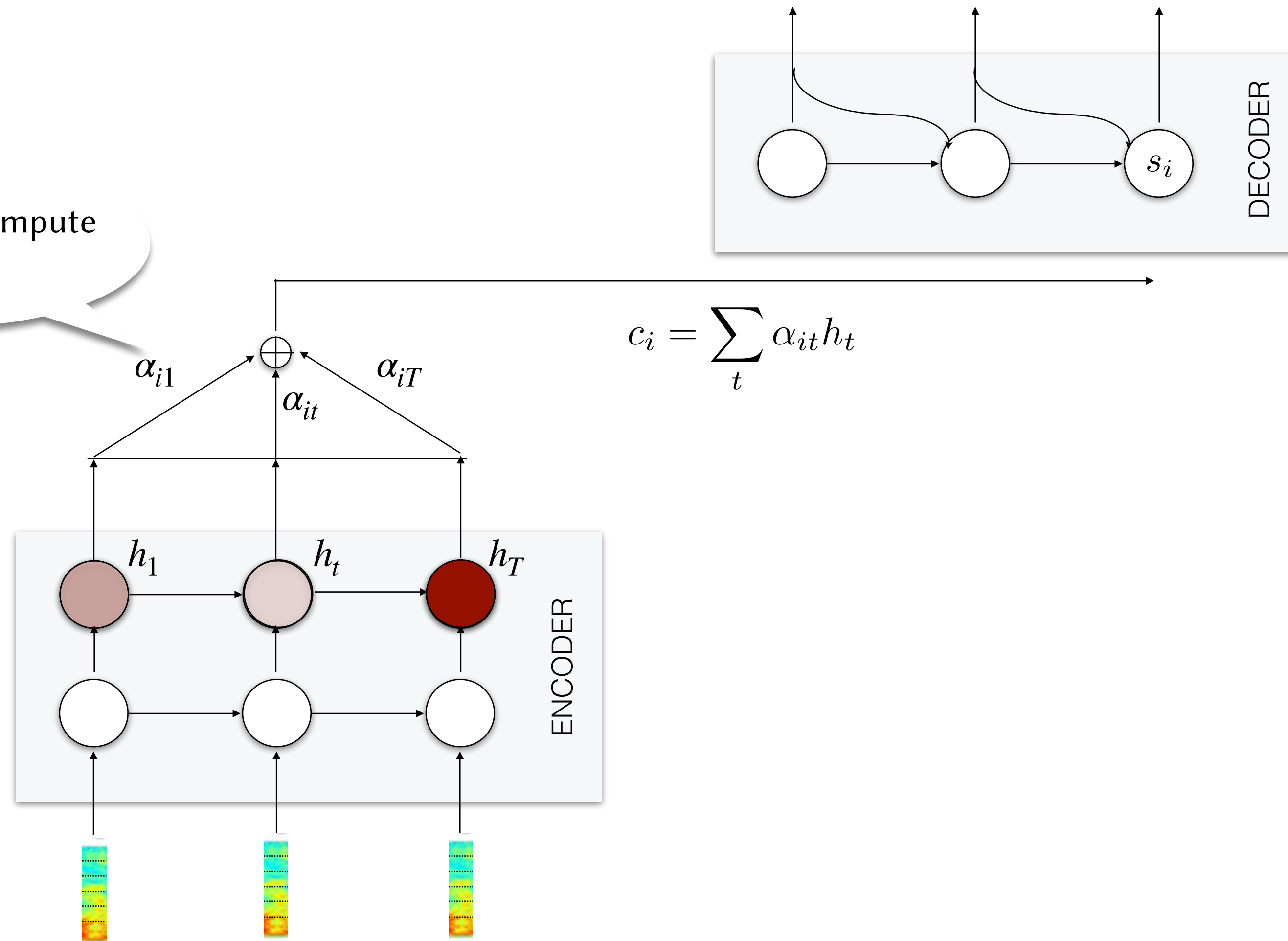
$$\mathbf{h} = \text{Listen}(\mathbf{x})$$

$$P(y_i | \mathbf{x}, y_{<i}) = \text{AttendAndSpell}(y_{<i}, \mathbf{h})$$

Sequence to sequence models

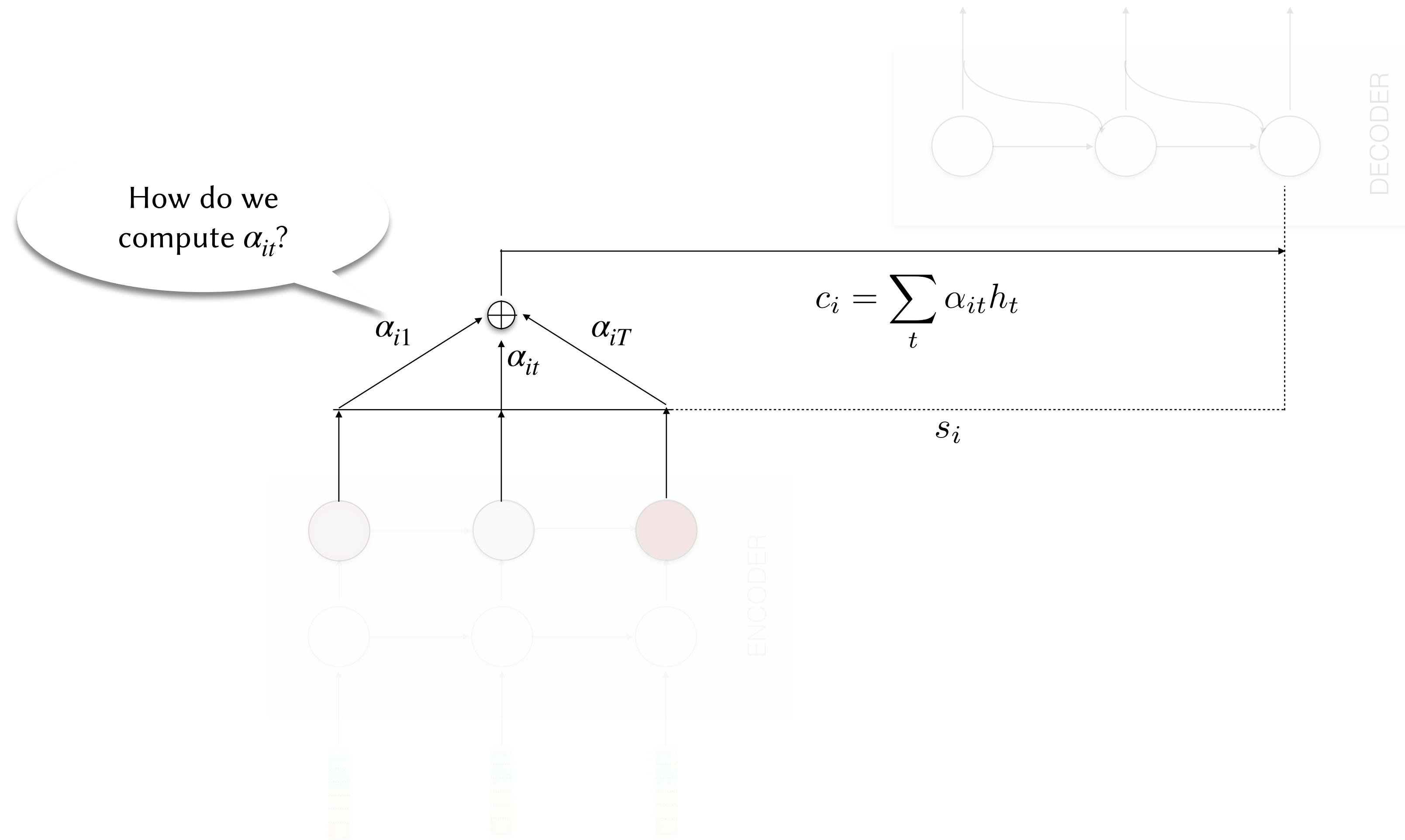
Encoder-decoder architecture with attention

How do we compute α_{it} ?



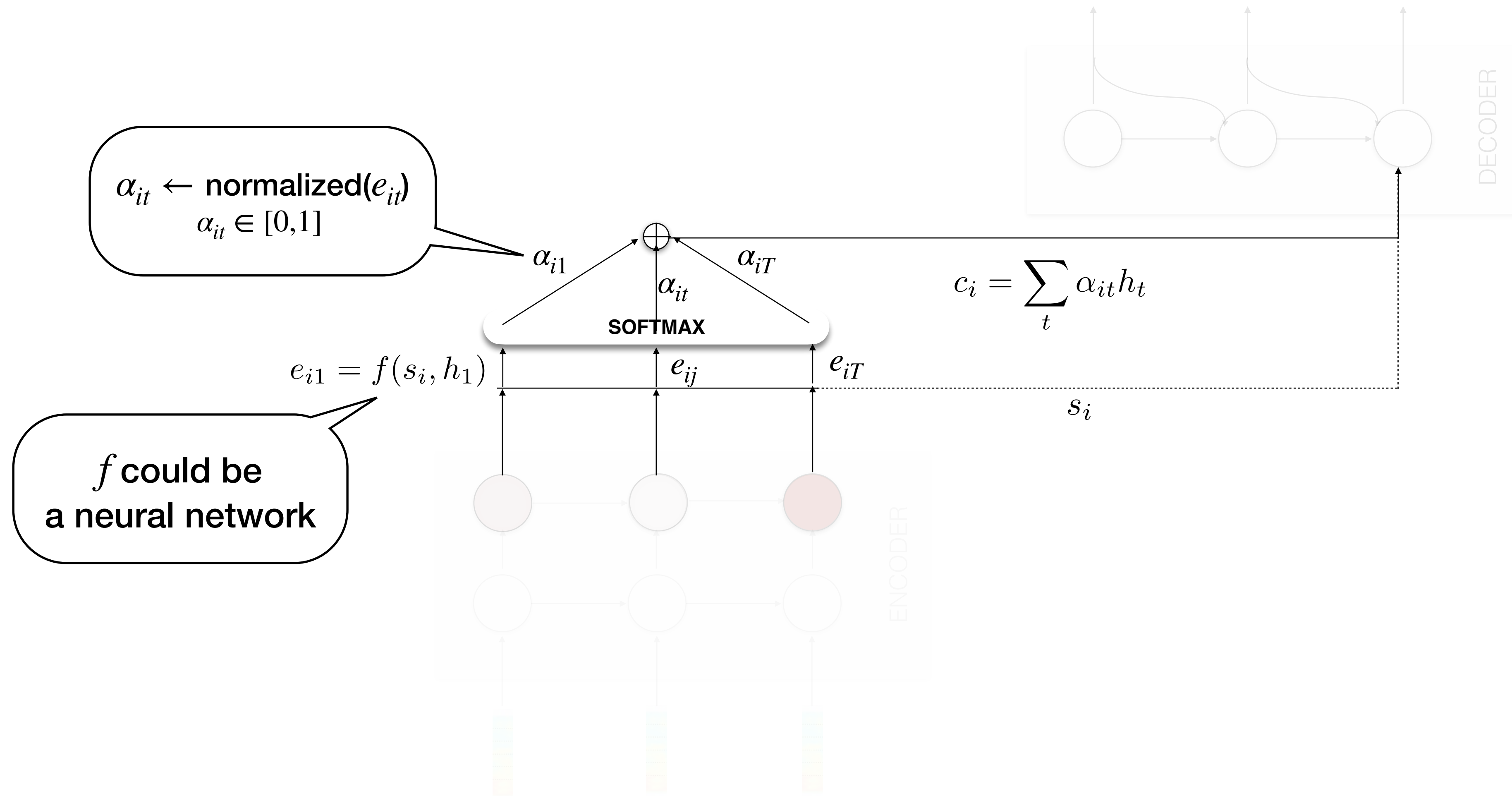
Sequence to sequence models

Encoder-decoder architecture with attention

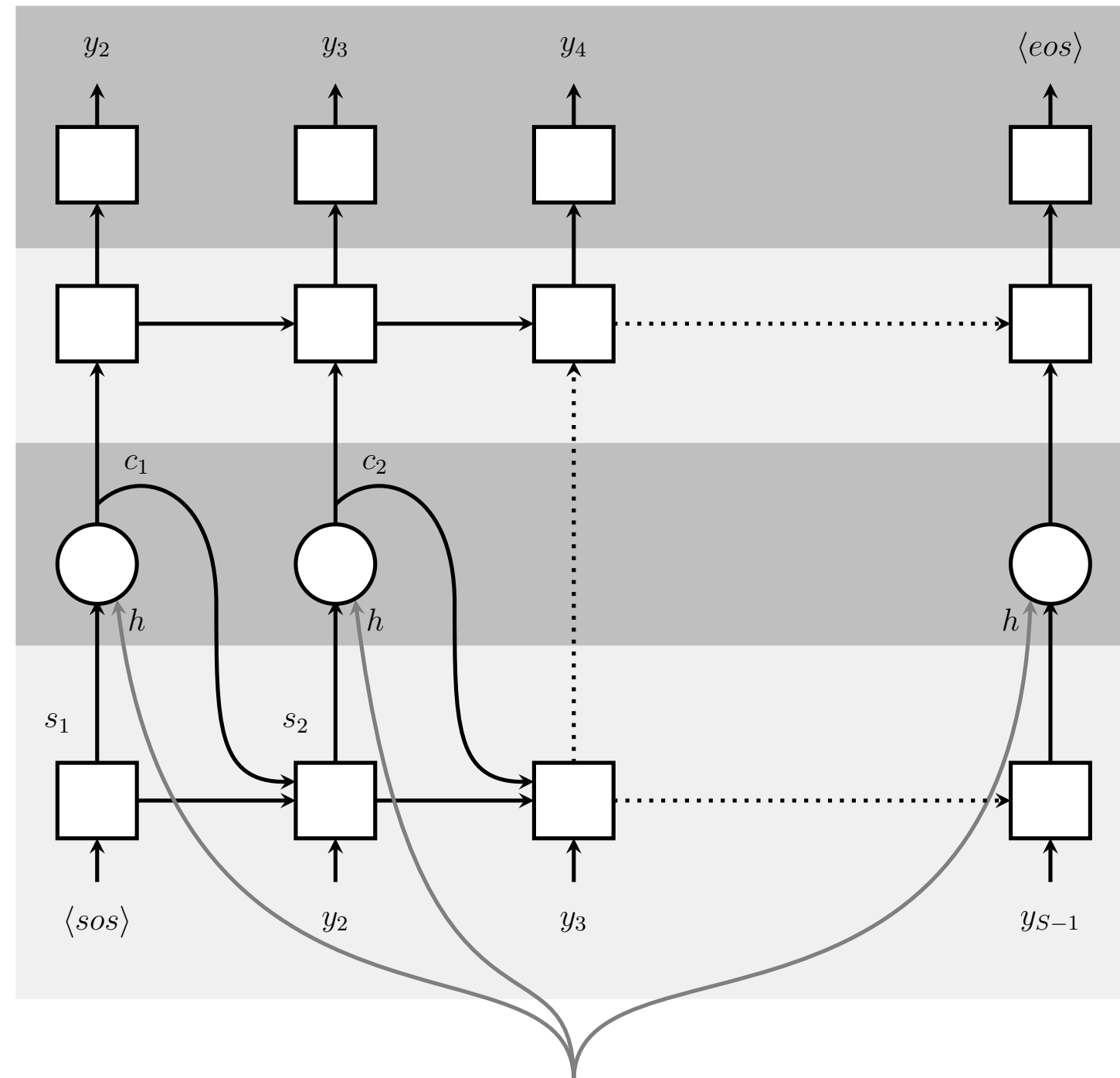


Sequence to sequence models

Encoder-decoder architecture with attention

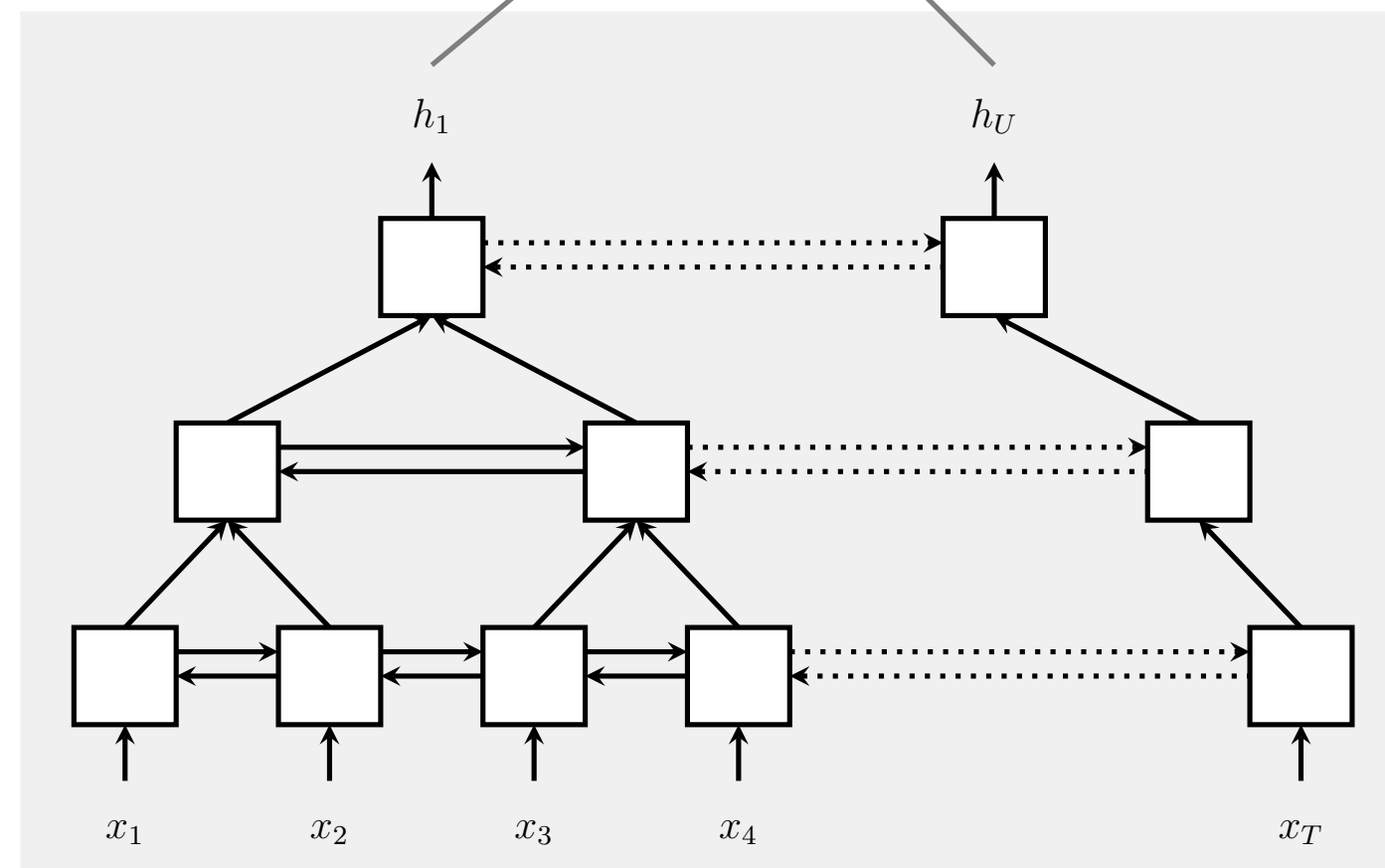


Speller



$h = (h_1, \dots, h_U)$

Listener



Attend and Spell

- Produces a distribution over characters conditioned on all characters seen previously

$$c_i = \text{AttentionContext}(s_i, \mathbf{h})$$

$$s_i = \text{RNN}(s_{i-1}, y_{i-1}, c_{i-1})$$

$$P(y_i | \mathbf{x}, y_{<i}) = \text{CharacterDistribution}(s_i, c_i)$$

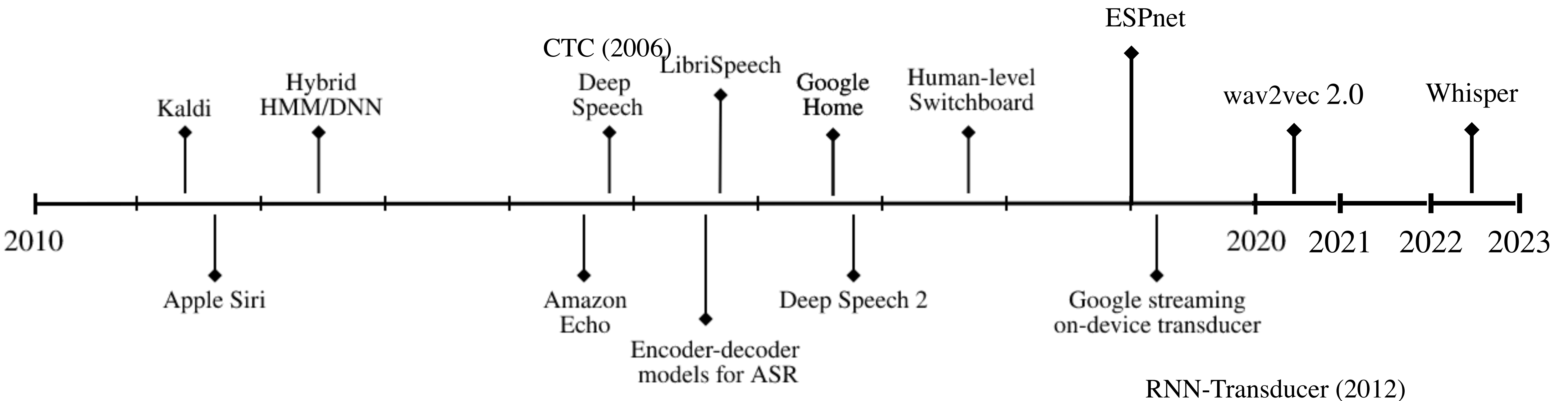
- At each decoder time-step i , AttentionContext computes a score for each encoder step u , which is then converted into softmax probabilities that are linearly combined to compute c_i

$$e_{i,u} = \langle \phi(s_i), \psi(h_u) \rangle$$

$$\alpha_{i,u} = \frac{\exp(e_{i,u})}{\sum_{u'} \exp(e_{i,u'})}$$

$$c_i = \sum_u \alpha_{i,u} h_u$$

Progress in ASR Over the Last Decade

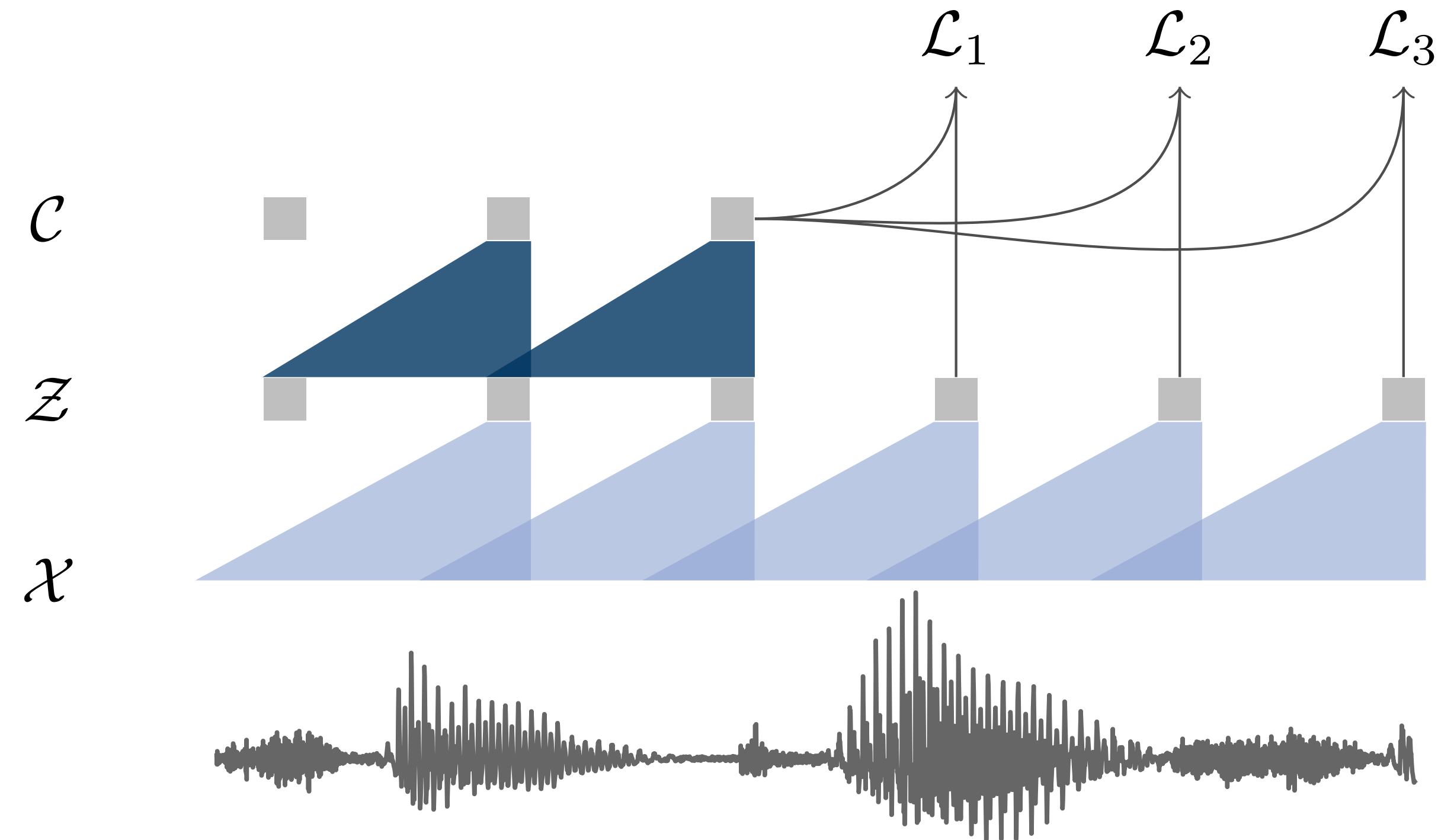


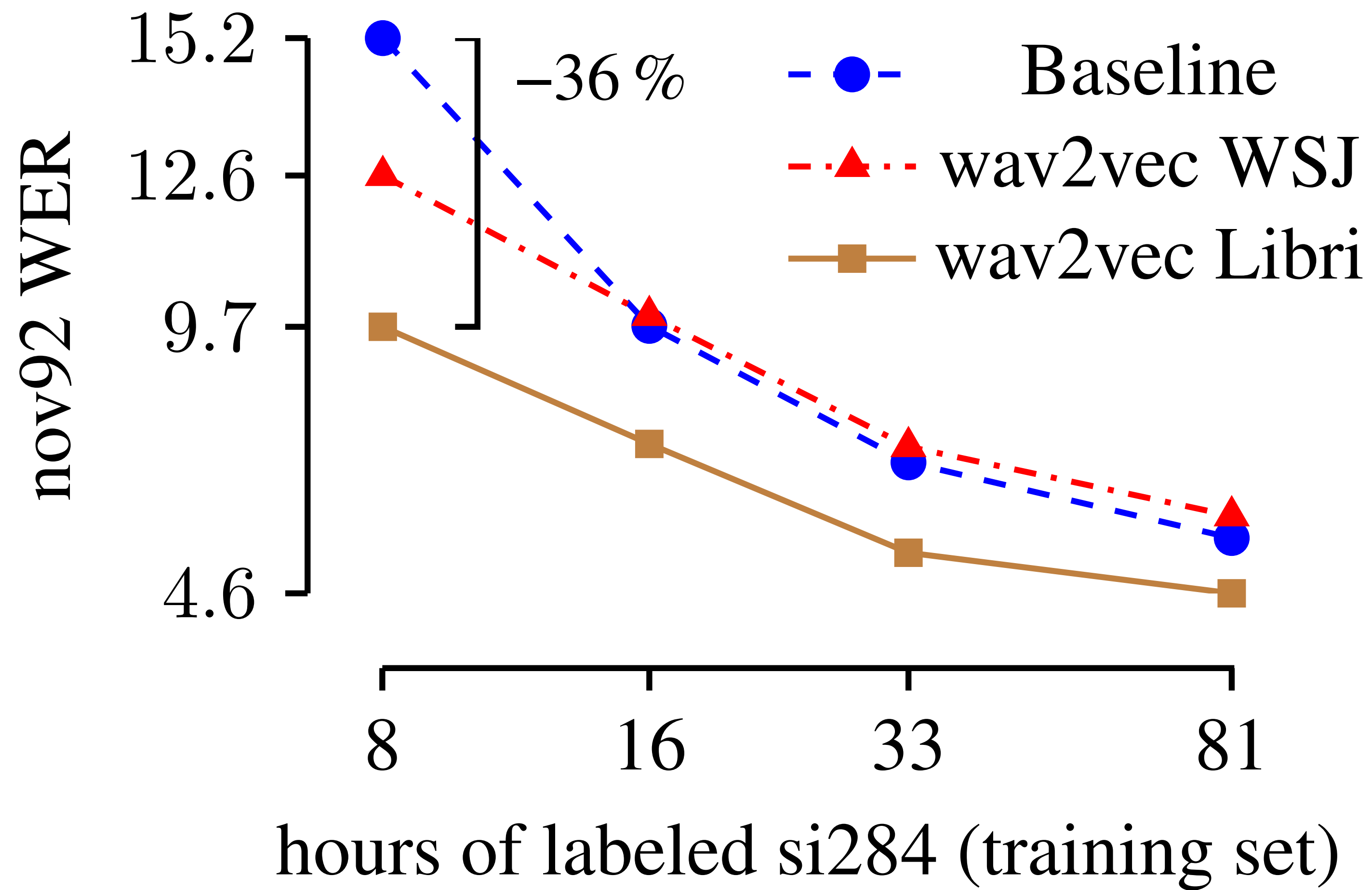
wav2vec

- Algorithm that uses raw audio to learn speech representations (“**self-supervised**” approach)

- Encoder network embeds raw audio into a latent representation ($f : \mathcal{X} \rightarrow \mathcal{Z}$) and a context network combines multiple encoded representations into a contextualised embedding ($g : \mathcal{Z} \rightarrow \mathcal{C}$)
- Train the model to minimize the following contrastive loss:

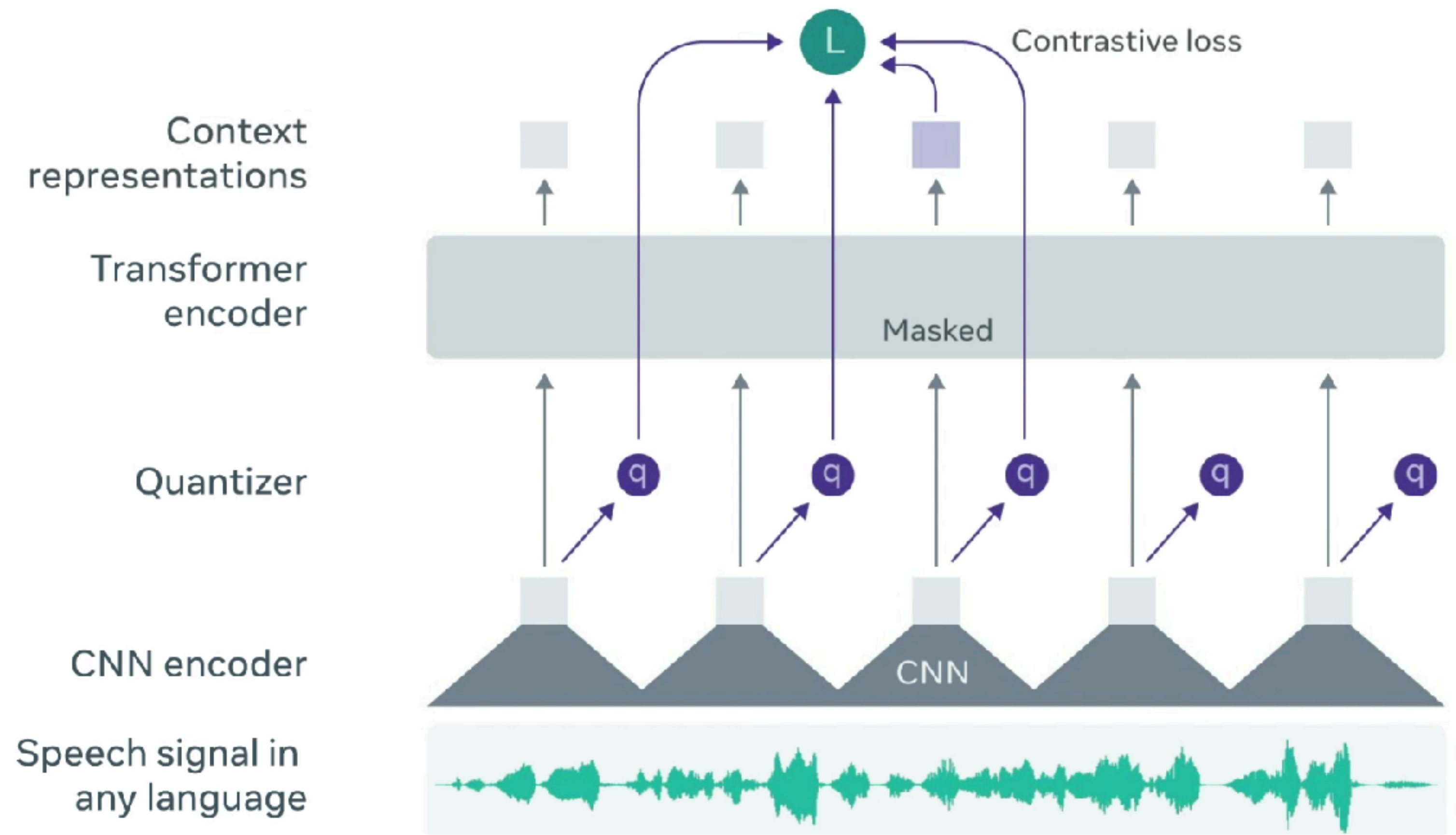
$$\mathcal{L}_k = - \sum_{i=1}^{T-k} \log \frac{\exp(\text{sim}(c_i, z_{i+k}))}{\sum_{\tilde{z}} \exp(\text{sim}(c_i, \tilde{z}))}$$



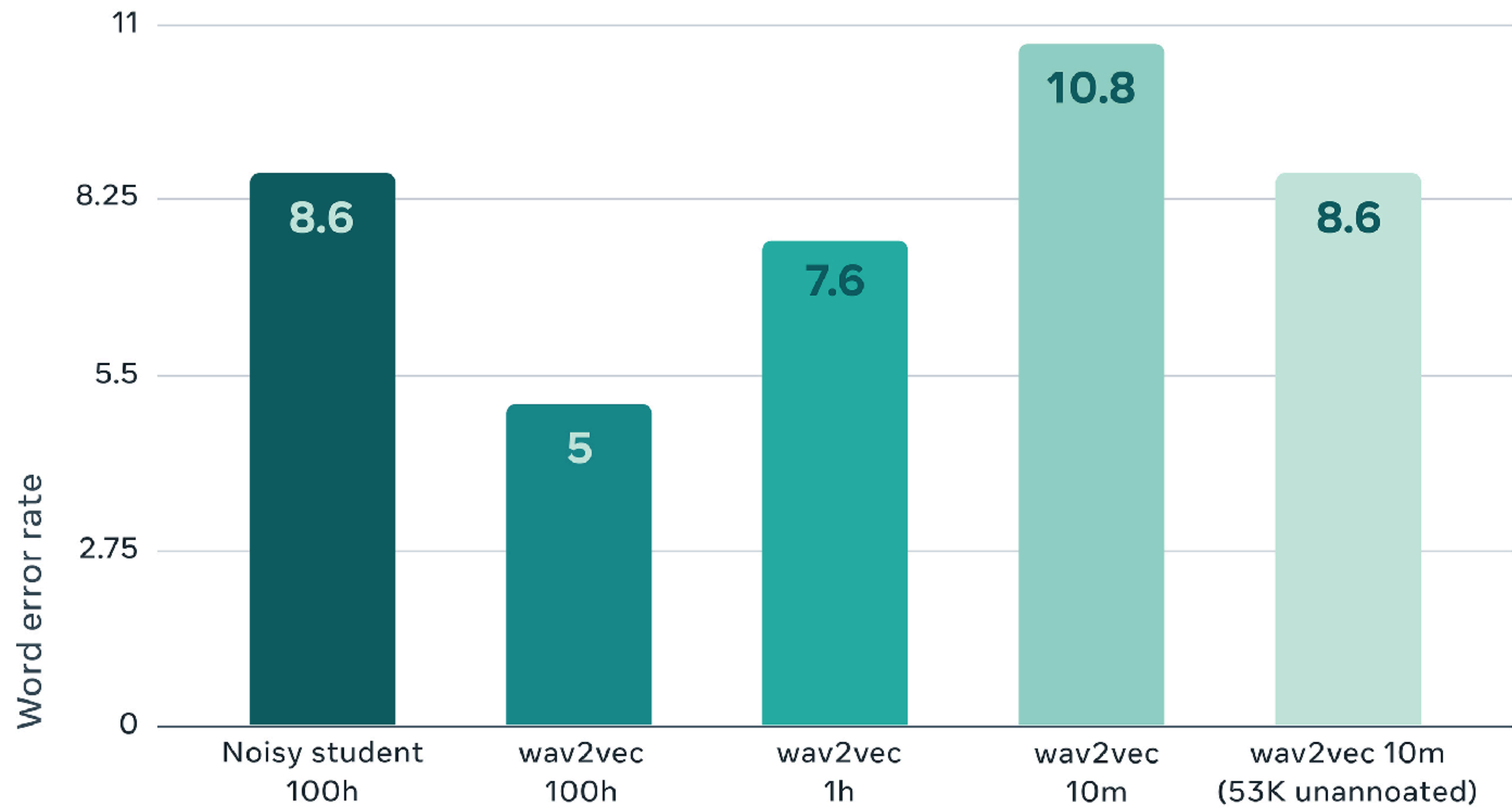


wav2vec 2.0: Learning Speech Representations from Raw Audio

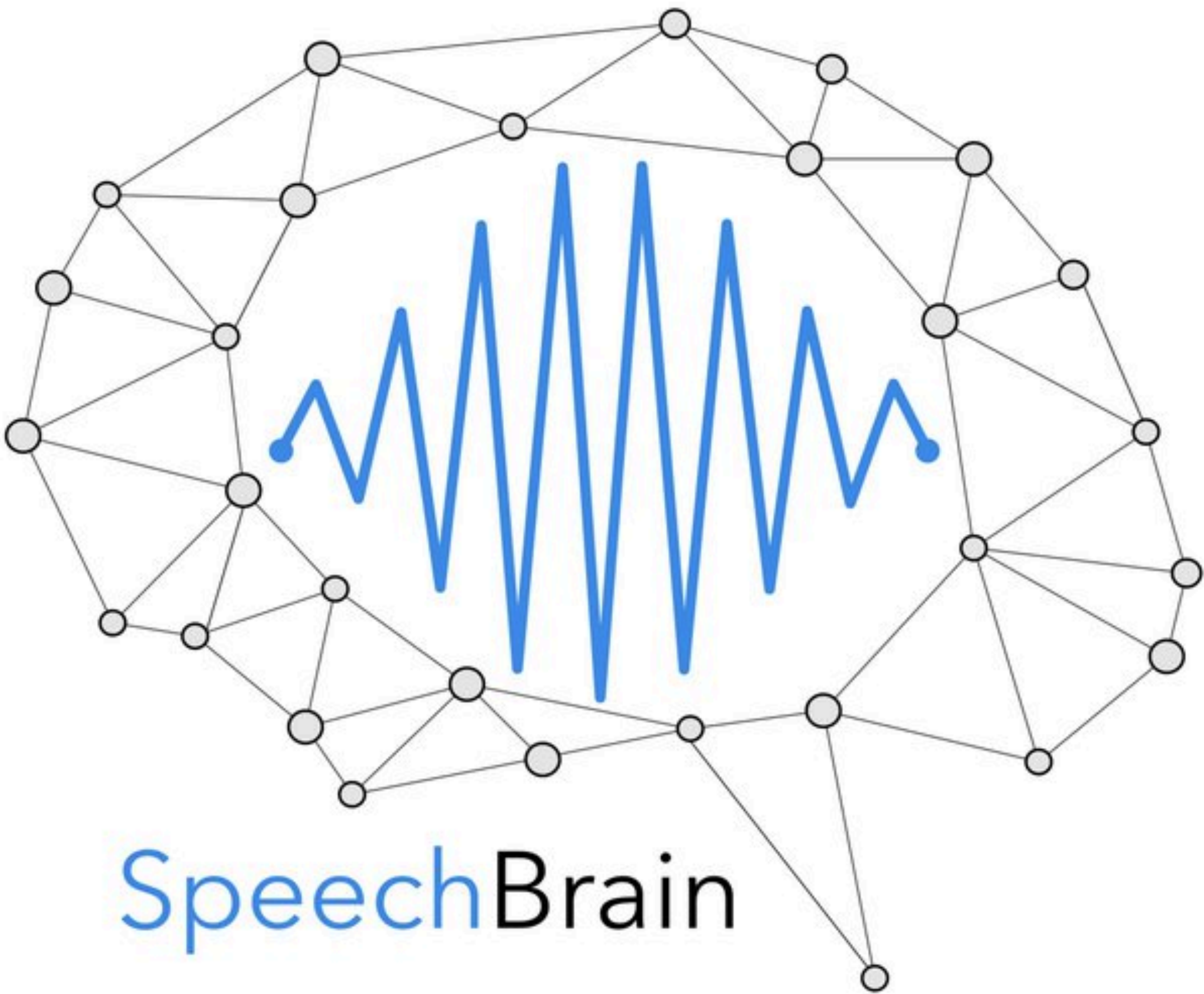
- Similar to wav2vec. Outputs from the encoder are further quantized.
- Masks spans of speech representations (as in masked language modelling for BERT [1])
- Training objective is to recover the masked representations among a set of distractors.



wav2vec 2.0: Results on English



Exciting Time to do Speech Research



Coqui, Freeing Speech

Coqui, a startup providing open speech tech for everyone 🐸
Sign up with your email address to receive the Coqui newsletter.

Over
80%+
Of respondents
are actively using
ASR to transcribe
speech data.
However...

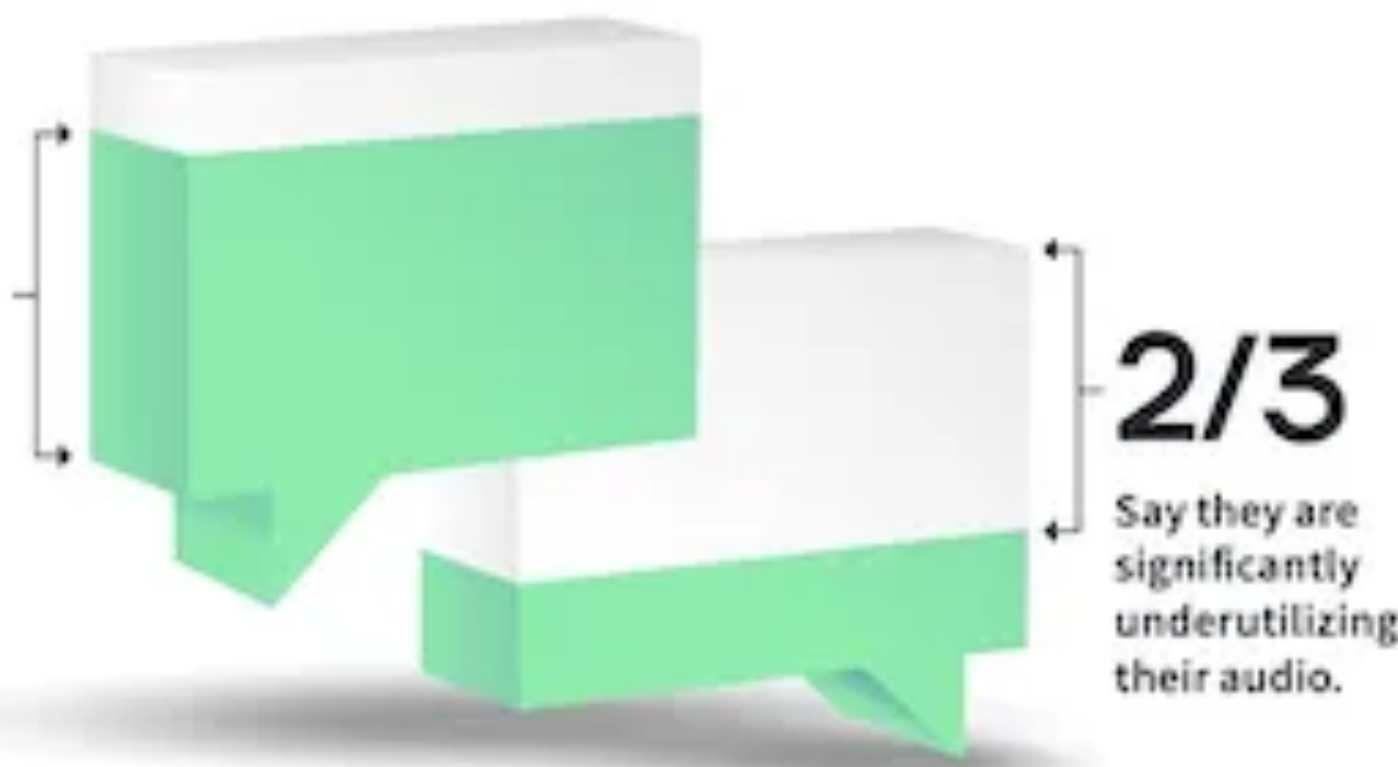


Image from: <https://coqui.ai/>
<https://speechbrain.github.io/>

Exciting Time to do Speech Research



Coqui, Freeing Speech

Coqui, a startup providing open speech tech for everyone 🐸
Sign up with your email address to receive the Coqui newsletter.

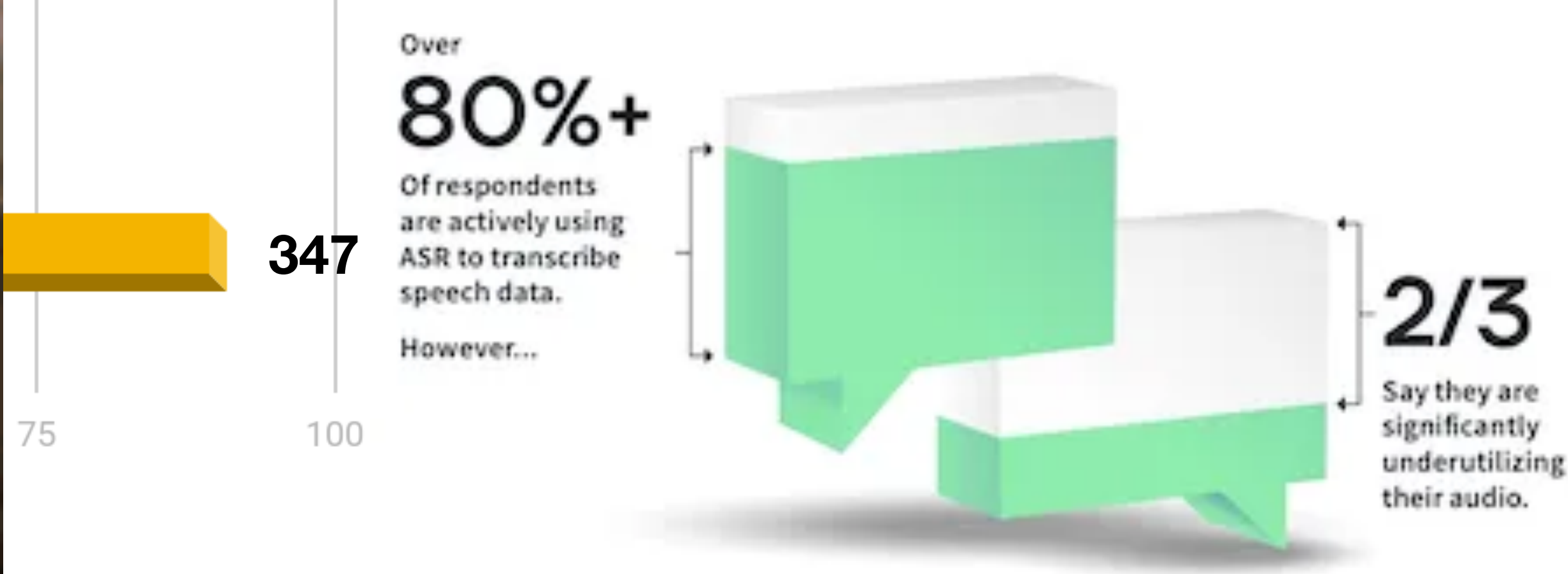


Image from: <https://coqui.ai/>

<https://speechbrain.github.io/>

<https://opusresearch.net/wordpress/2021/03/03/opus-research-report-2021-state-of-automatic-speech-recognition/>

Many Unsolved Problems Related to ASR

- State-of-the-art ASR systems do not work well on heavy regional accents, dialects.
- Code-switching is hard for ASR systems to deal with.
- How do we deploy such high-capacity ASR systems in low-power devices?
- How do we rapidly build competitive ASR systems for a new language?
Low-resource ASR, multilingual ASR and pretrained models.
- How do we recognize speech from meetings where a primary speaker is speaking amidst other speakers?