FINITE STATE TRANSDUCERS AND NEURAL NETS IN ASR

YU / HUGHES

OCTOBER 28/NOVEMBER 2, 2021

UMass Amherst

HIDDEN MARKOV MODEL

REVIEW

REVIEW QUESTION: GUMBALL MACHINE HMMS

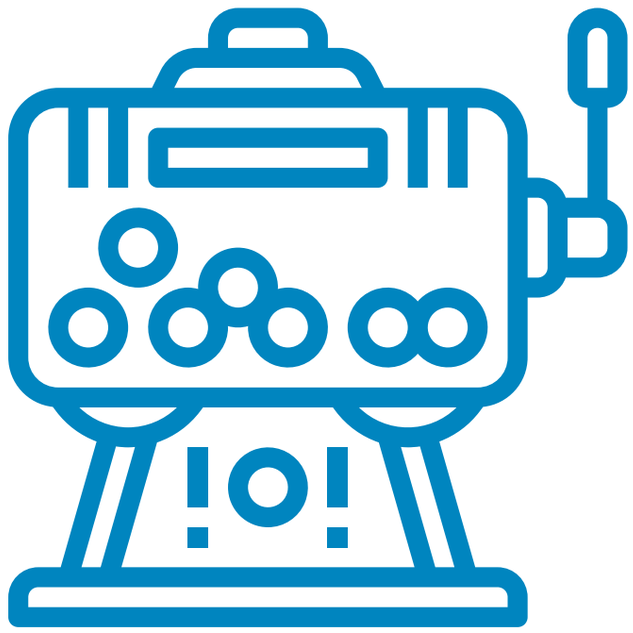
- Our review question today comes courtesy of Mark Hasegawa-Johnson, a senior speech processing professor at the University of Illinois Urbana-Champaign

https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf

Gumball machines in a Diner at Dallas, Texas, in 2008," Andreas Praefcke, public domain image.

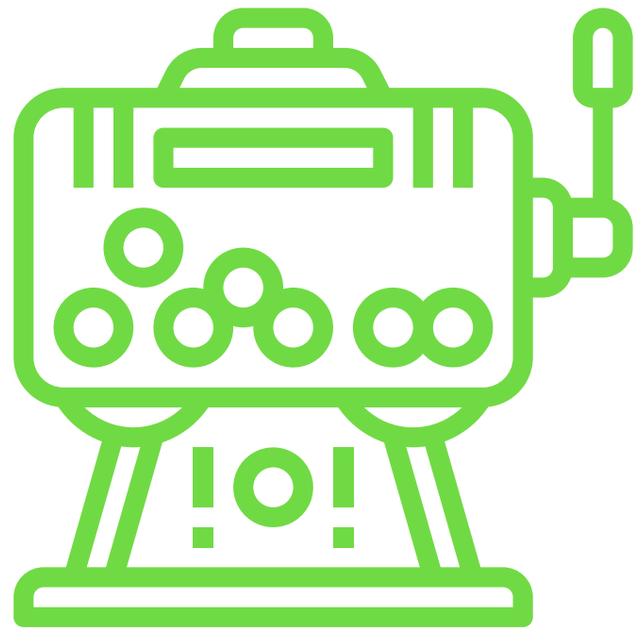
REVIEW QUESTION: OBSERVATION PROBABILITIES

Machine q_1



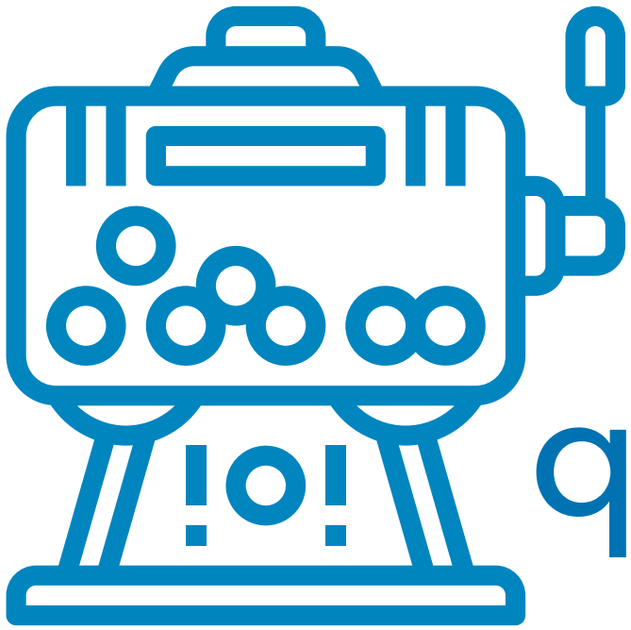
60% grapefruit
40% apple

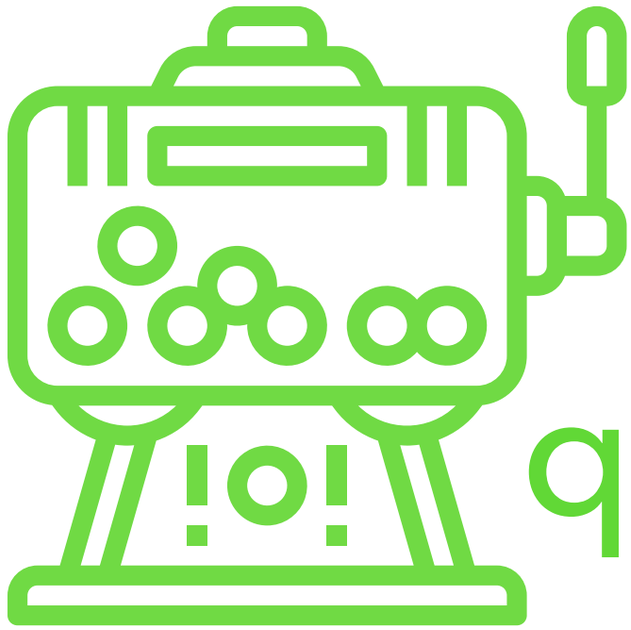
Machine q_2

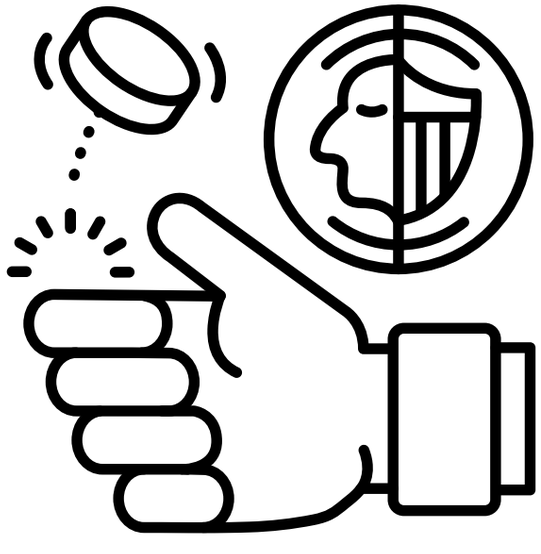


10% grapefruit
90% apple

REVIEW QUESTION: INITIAL STATE PROBABILITIES





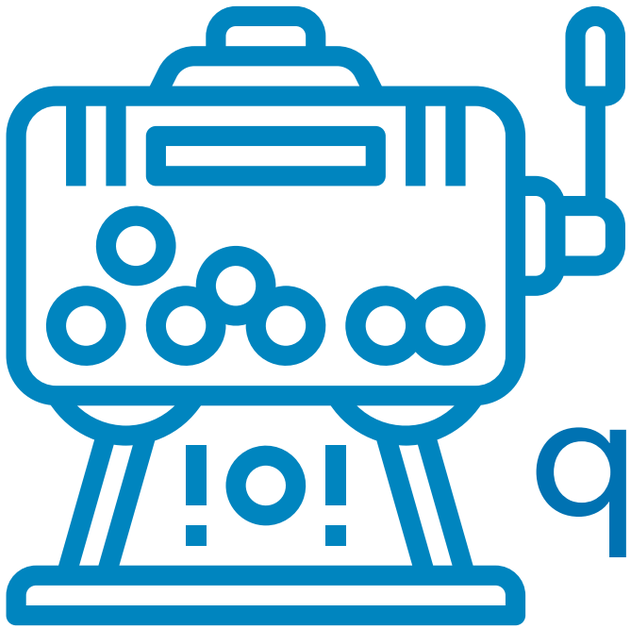


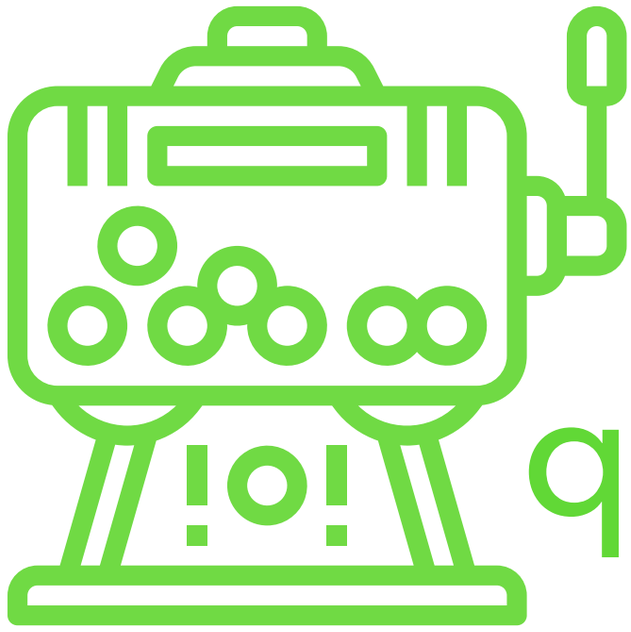
Flip fair coin: start in?

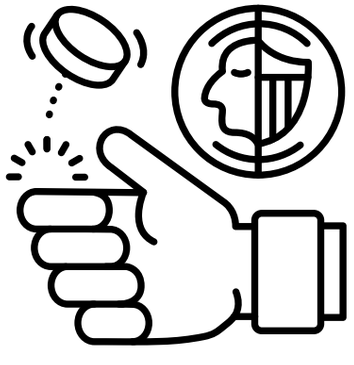
Heads → q_1

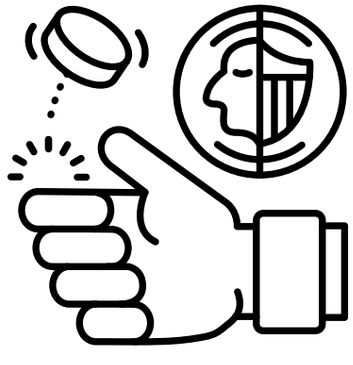
Tails: → q_2

REVIEW QUESTION: TRANSITION PROBABILITIES







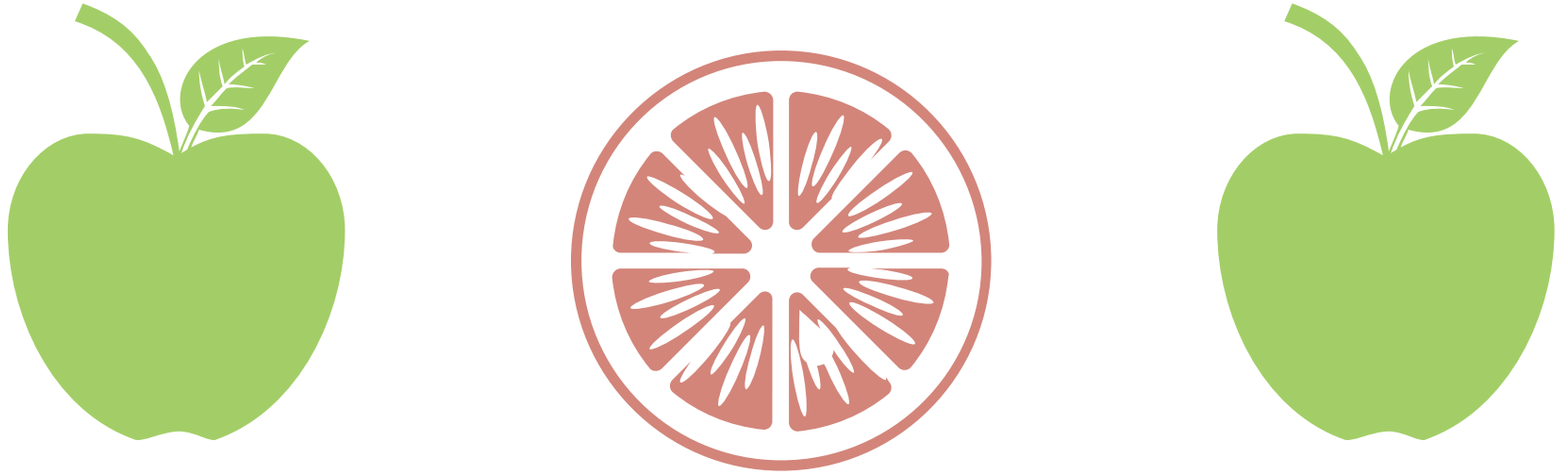


After already used a machine…
Flip 2 fair coins: stay or change?

HH → change
else → stay

REVIEW QUESTION: OBSERVATION SEQUENCE

Cerys bought 3 gumballs, with the following order:

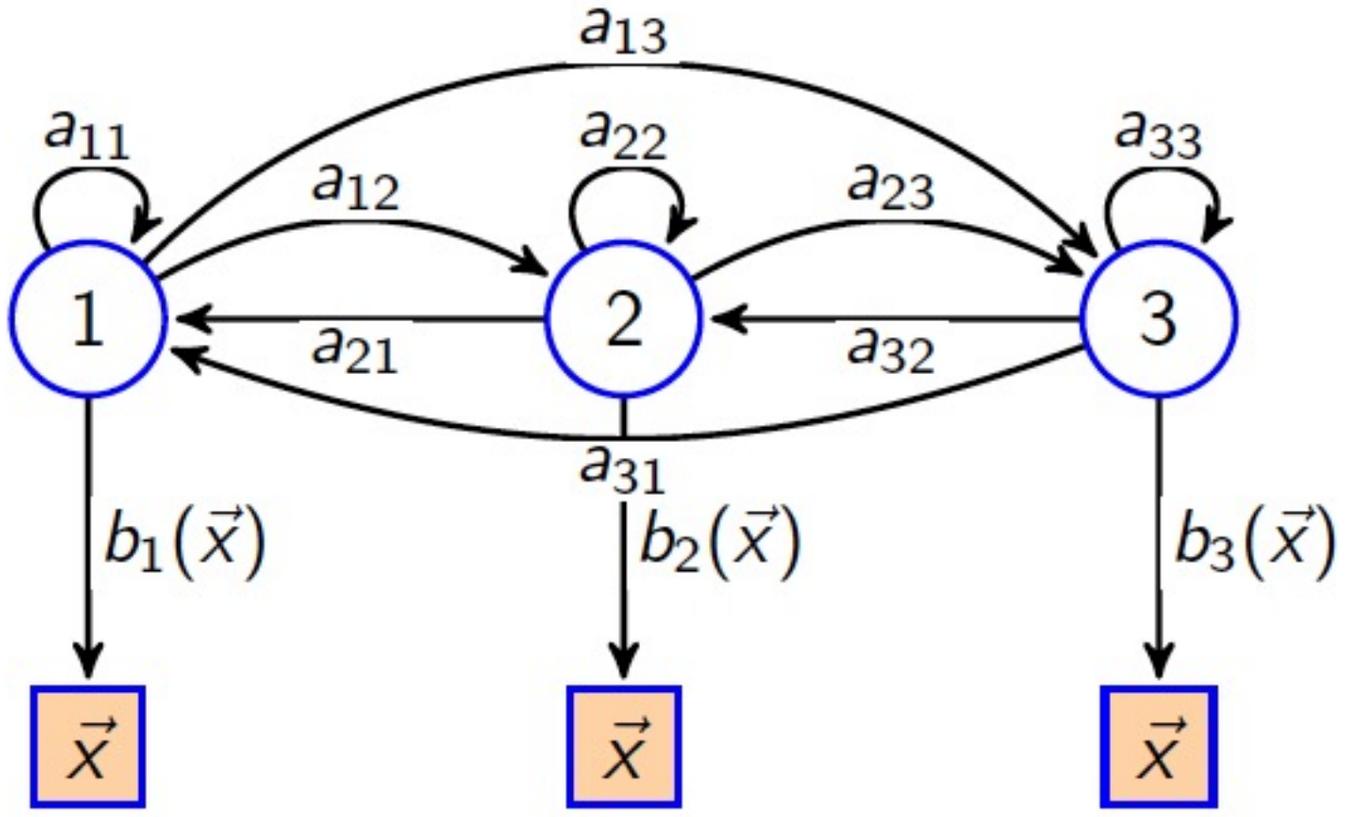


REVIEW QUESTION: DEFINITION OF OBS AND HMM

- Use the description of the gumball scenario to fill out the following and define the observation sequence of T consecutive observations, $\mathbf{X} = \langle x_1, x_2, \ldots, x_j, \ldots, x_T \rangle$, the hidden state sequence $\mathbf{Q} = \langle q_1, q_2, \ldots, q_i, \ldots, q_T \rangle$, and the HMM $\Lambda_{gumball} = \{\pi_i, a_{ij}, \mathbf{b}_j(x) \, \forall i, j\}$:
  - Observation sequence $\mathbf{X} = ?$
  - Hidden state sequence $\mathbf{Q} = ?$
  - Initial state probabilities $\pi_i = p(q_1 = i) \, \forall i = ?$
  - Transition probabilities $a_{ij} = p(q_t = j \,|\, q_{t-1} = i) \, \forall i, j = ?$
  - Observation probabilities $\mathbf{b}_j \forall j = ?$ (n.b. $b_j(\mathbf{x}) = p(\mathbf{x_t} = \mathbf{x} \,|\, q_t = j)$

REVIEW QUESTION: DEFINITION OF OBS AND HMM

https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf



1. Start in state $q_t = i$, for some $1 \leq i \leq N$.
2. Generate an observation, $\vec{x}$, with pdf $b_i(\vec{x})$.
3. Transition to a new state, $q_{t+1} = j$, according to pmf a_{ij}.
4. Repeat steps #2 and #3, T times each.

THREE BASIC PROBLEMS FOR HMMS

C. The Three Basic Problems for HMMs[5]

Given the form of HMM of the previous section, there are three basic problems of interest that must be solved for the model to be useful in real-world applications. These problems are the following:

Problem 1: Given the observation sequence $O = O_1 O_2 \cdots O_T$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?

Problem 2: Given the observation sequence $O = O_1 O_2 \cdots O_T$, and the model λ, how do we choose a corresponding state sequence $Q = q_1 q_2 \cdots q_T$ which is optimal in some meaningful sense (i.e., best "explains" the observations)?

Problem 3: How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

Rabiner (1989)

- Come up with an example of each problem for our gumball scenario!

Recognition:

Given Λ_1, Λ_2, X,

Is $p(X|\Lambda_1) > p(X|\Lambda_2)$?

Segmentation:

What is $p(q_t = i|X, \Lambda)$?

Training:

Given Λ, X,

Can we find Λ' s.t.

$$p(X|\Lambda') > p(X > \Lambda)$$

https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf

THREE BASIC PROBLEMS FOR HMMS

Recognition:

Given Λ_1, Λ_2, X,

Is $p(X \mid \Lambda_1) > p(X \mid \Lambda_2)$?

Compute overall likelihood:

Forward algorithm

Segmentation:

What is $p(q_t = i \mid X, \Lambda)$?

Decode most likely state sequence:

Viterbi algorithm

Training:

Given Λ, X,

Can we find Λ' s.t.

$p(X \mid \Lambda') > p(X > \Lambda)$

Estimate most likely parameters:

EM (Forward-Backward) algorithm

A SEGMENTATION/STATE DECODING PROBLEM

- Cerys bought three gum balls. The first was an apple, the second a grapefruit and the third an apple
- Unfortunately, the second of the three quarters Cerys used was my special birth year quarter I'd be hanging onto for a few decades since I got it from the tooth fairy and I want that back!
- Which gumball machine do I need to break into to get my special quarter back?

https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf

DECODING AND ALIGNMENT: STEP BY STEP

- Observation sequence is: $X = \langle A, G, A \rangle$
- Given the observed sequence, what is the probability that we were in Machine 1 in time step 2? The probability that we were in Machine 2 in time step 2?
- Compute the posterior state probability for time step 2: $\gamma_2(i)$

$$\gamma_t(i) = \frac{p(X, q_t = i \mid \Lambda)}{\sum_{k=1}^{N} p(X, q_t = k \mid \Lambda)}$$

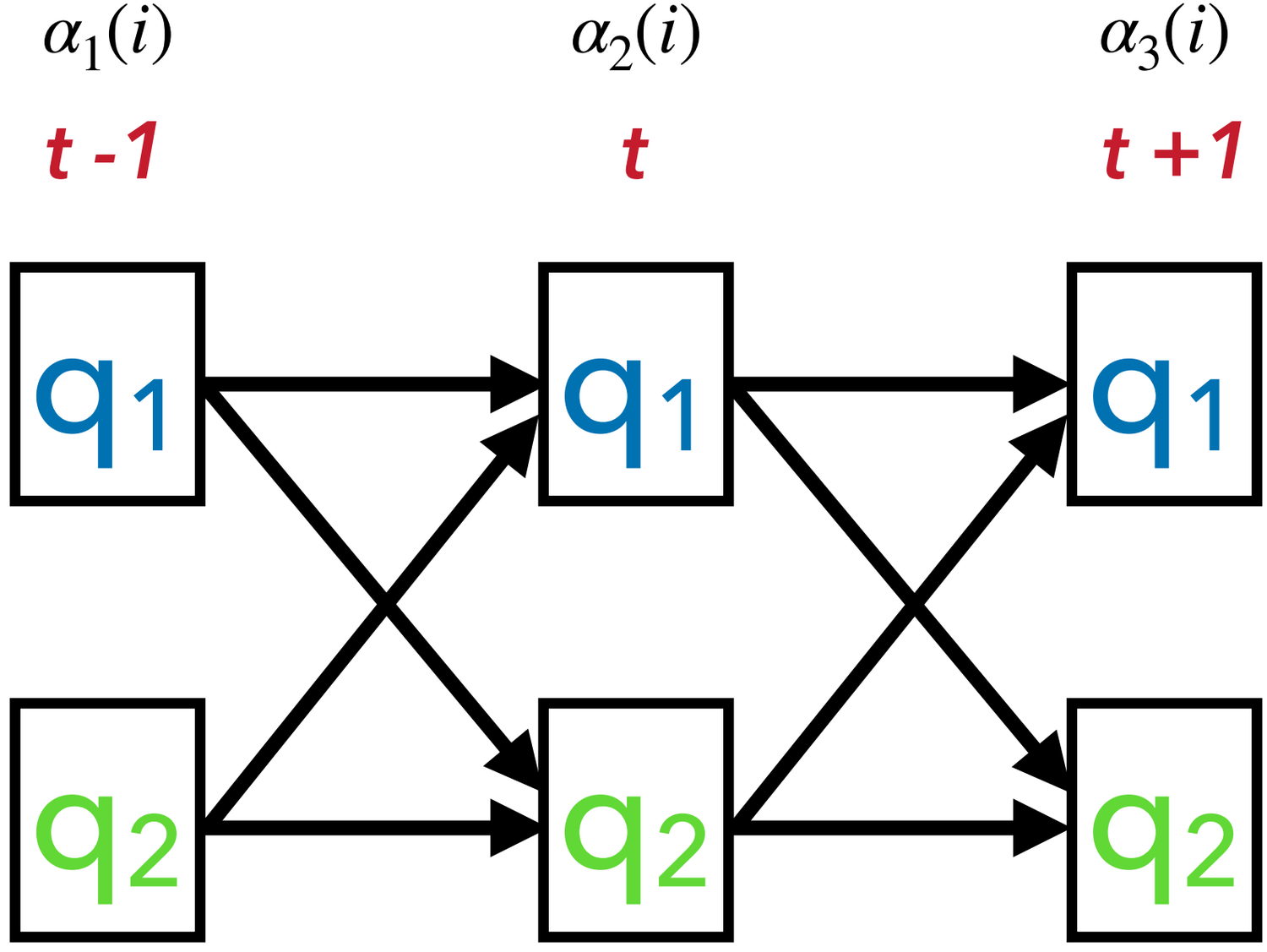
https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf

DECODING AND ALIGNMENT: STEP BY STEP

- Observation sequence is: $X = \langle A, G, A \rangle$
- Compute the forward probability
  $\alpha_t(i) = p(\mathbf{x_1}, \ldots, \mathbf{x_t}, q_t = i \,|\, \Lambda)$ for time step 1, i.e.,
  $\alpha_1(i) = p(\mathbf{x_1}, q_1 = i \,|\, \Lambda)$
- What is $\alpha_1(i)$ in terms of initial state probabilities π_i and observation probability $b_i(\mathbf{x_1})$?

https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf

FORWARD ALGORITHM AS STATE-TIME TRELLIS



Exploit Markov assumption: probability of being in some state j at next time step dependent only on state at current time state

DECODING AND ALIGNMENT: STEP BY STEP

- Observation sequence is: $X = \langle A, G, A \rangle$
- Compute the forward probability
  $\alpha_t(i) =_{def} p(\mathbf{x_1}, \ldots, \mathbf{x_t}, q_t = i \mid \Lambda)$ for time step 1, i.e.,
  $\alpha_1(i) =_{def} p(\mathbf{x_1}, q_1 = i \mid \Lambda)$
- Compute the forward probability for time step 2:
  what is $\alpha_2(j) = \sum_{i=1}^{2} \alpha_1(i) a_{ij} b_j(\mathbf{x_2})$?

https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf

DECODING AND ALIGNMENT: STEP BY STEP

- Observation sequence is: $X = \langle A, G, A \rangle$
- Compute the backward probability
  $\beta_t(i) =_{def} p(\mathbf{x_{t+1}}, \ldots \mathbf{x_T} \,|\, q_t = i, \Lambda)$ for time step 2, i.e.,
  $\beta_2(i) =_{def} p(\mathbf{x_{2+1}}, \mathbf{x_2} \,|\, q_2 = i, \Lambda)$
- Note that $\beta_3(i) = 1, 1 \leq i \leq N$ (Why?)
- What is $\beta_2(i) = \sum_{j=1}^{2} a_{ij} b_j(\mathbf{x_3}) \beta_3(j)$?

https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf

DECODING AND ALIGNMENT: LAST STEP

- Observation sequence is: $X = \langle A, G, A \rangle$
- Given the observed sequence, what is the probability that we were in Machine 1 in time step 2? The probability that we were in Machine 2 in time step 2?
- Compute the posterior state probability for time step 2:

$$\gamma_2(i) = \frac{\alpha_2(i)\beta_2(i)}{\sum_{k=1}^{2} \alpha_2(k)\beta_2(k)}$$

$$\gamma_t(i) = \frac{p(X, q_t = i \mid \Lambda)}{\sum_{k=1}^{N} p(X, q_t = k \mid \Lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{k=1}^{N} \alpha_t(k)\beta_t(k)}$$

https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf

SPEECH RECOGNITION ARCHITECTURE

FUNDAMENTAL EQUATION OF SPEECH RECOGNITION

Most likely word sequence

Word sequence

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X})$$

Observations (Acoustic feature vectors)

FUNDAMENTAL EQUATION OF SPEECH RECOG

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X})$$

$$P(\mathbf{W} \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid \mathbf{W})P(\mathbf{W})}{P(\mathbf{X})}$$

$$\propto P(\mathbf{X} \mid \mathbf{W})P(\mathbf{W})$$

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{X} \mid \mathbf{W}) \; P(\mathbf{W})$$

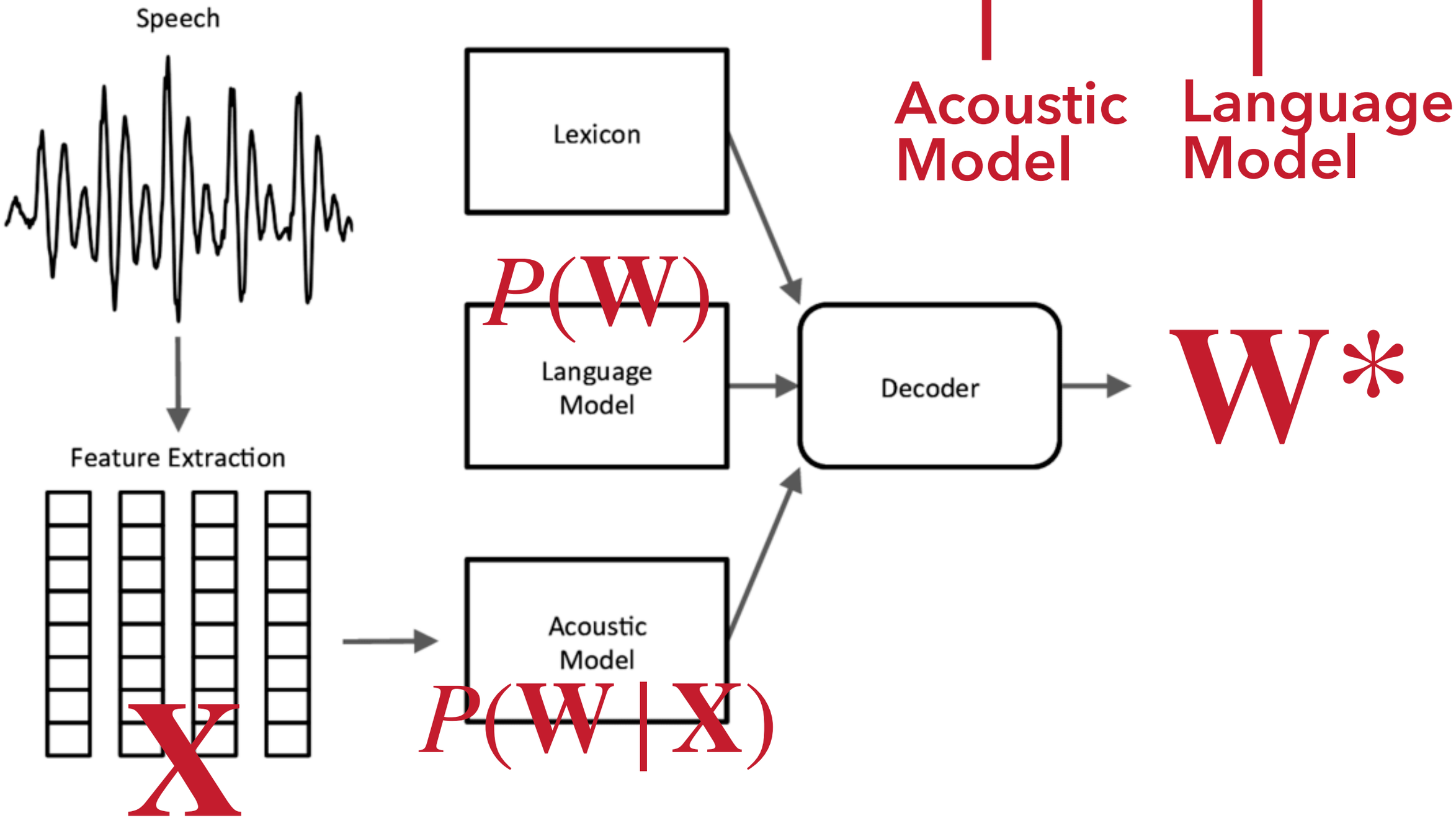
Acoustic Model

Language Model

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{X} \mid \mathbf{W}) \, P(\mathbf{W})$$

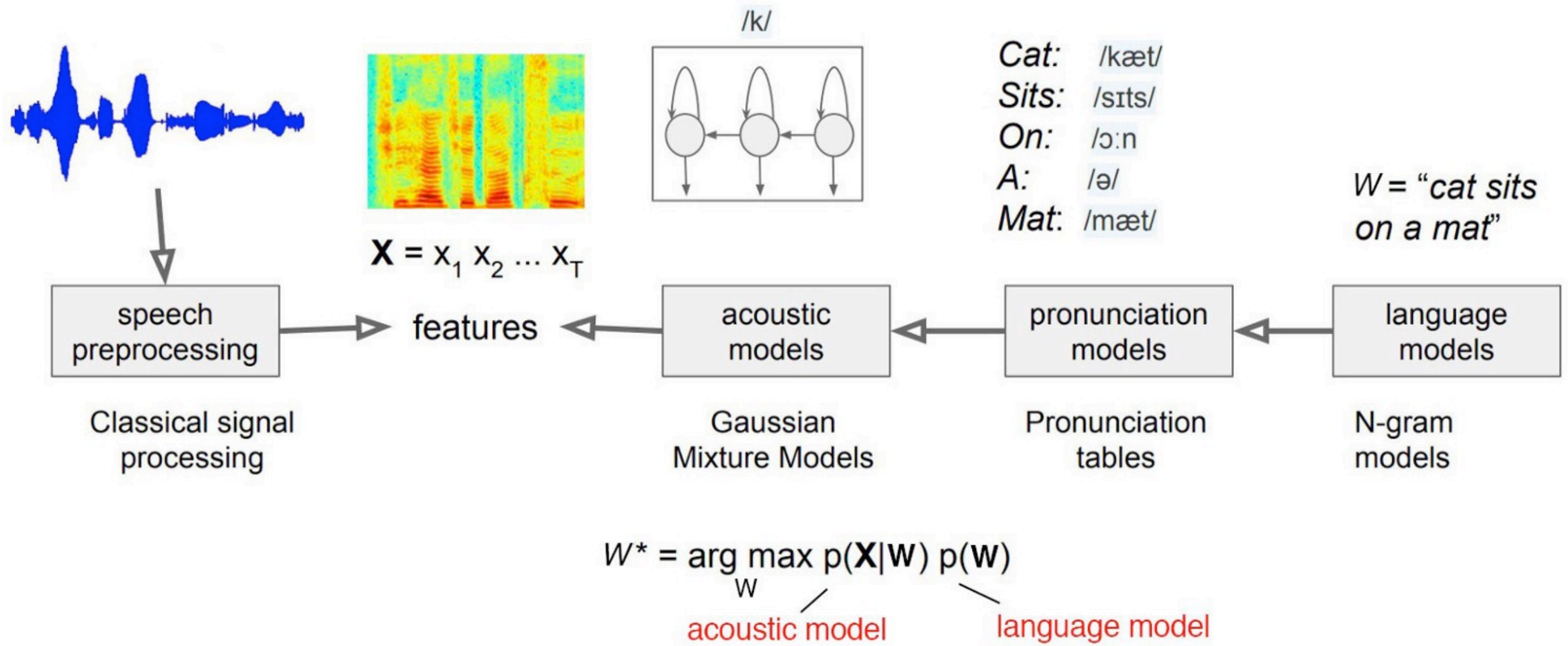
Acoustic Model

Language Model



https://www.researchgate.net/profile/Tom-Nuorivaara/publication/307560344/figure/fig1/AS:401591556427776@1472758296782/The-structure-of-a-typical-ASR-system.png

CLASSICAL SPEECH RECOGNITION



Pronunciation model = Lexicon

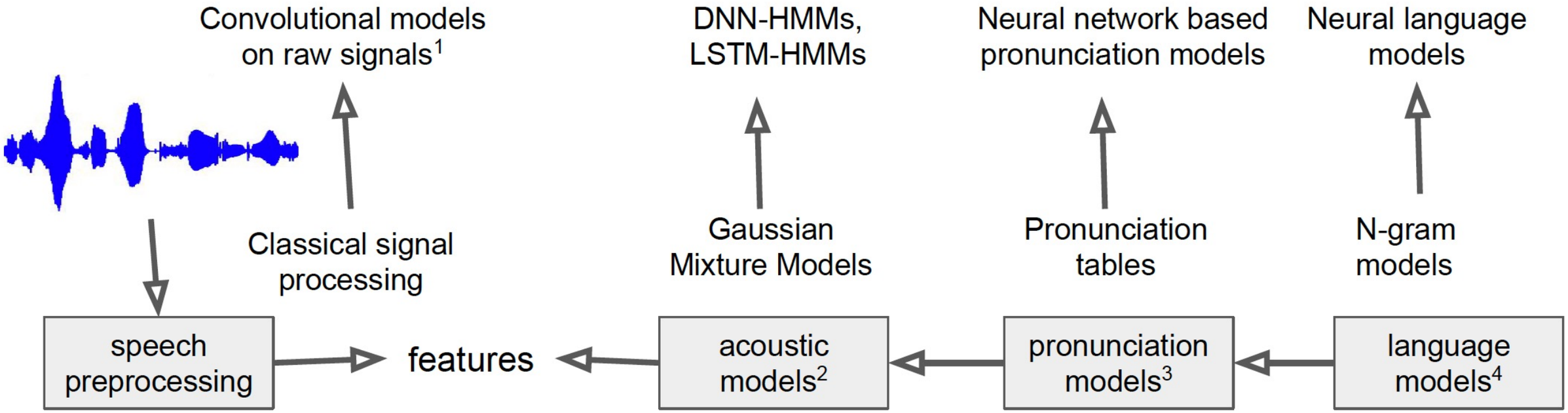
https://jonathan-hui.medium.com/speech-recognition-gmm-hmm-8bb5eff8b196

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lectures/cs224n-2017-lecture12.pdf

NEURAL NETWORK "INVASION"

Speech Recognition -- the neural network invasion

- Each of the components seems to be better off with a neural network



1. Jaitly, Navdeep, and Geoffrey Hinton. "Learning a better representation of speech soundwaves using restricted boltzmann machines." *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011.
2. Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." *IEEE Signal Processing Magazine* 29.6 (2012): 82-97.
3. Rao, Kanishka, et al. "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks." *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.
4. Mikolov, Tomas, et al. "Recurrent neural network based language model." *Interspeech*. Vol. 2. 2010.

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lectures/cs224n-2017-lecture12.pdf

INTERSPEECH 2021

https://www.interspeech2021.org/tutorials

30 August – 3 September 2021
INTERSPEECH 2021 BRNO | CZECHIA
Speech everywhere!

Home Authors Participants Program Students Venue/Travel Partners Registration Ab

TUTORIALS

Interspeech 2021 will feature six high-quality tutorials.

Expand all

Monday August 30, 11:00-14:00 CEST

E104 **Intonation Transcription and Modelling in Research and Speech Technology Applications - Amalia Arvaniti et al.**

E105 **Neural target speech extraction - Marc Delcroix et al.**

E112 **Speech Recognition with Next-Generation Kaldi (k2, Lhotse, Icefall) - Piotr Żelasko et al.**

Monday August 30, 15:00-18:00 CEST

E104 **An Introduction to Automatic Differentiation with Weighted Finite-State Automata - Awni Hannun**

E112 **SpeechBrain: Unifying Speech Technologies and Deep Learning With an Open Source Toolkit - Mirco Ravanelli et al.**

E105 **Concept to Code: Semi-Supervised End-To-End Approaches For Speech Recognition - Omprakash Sonie et al.**

NEXT-GEN KALDI

E112 **Speech Recognition with Next-Generation Kaldi (k2, Lhotse, Icefall) - Piotr Żelasko et al.**

This tutorial introduces k2, the cutting-edge successor to the Kaldi speech processing, which consists of several Python-centric modules to enable building speech recognition systems, along with its enabling counterparts, Lhotse and Icefall. The participants will learn how to perform swift data manipulation with Lhotse; how to build and leverage auto-differentiable weighted finite state transducers with k2; and how these two can be combined to create Pytorch-based, state-of-the-art hybrid ASR system recipes from Snowfall, the precursor to Icefall.

https://github.com/k2-fsa/k2

“The vision of k2 is to be able to seamlessly integrate Finite State Automaton (FSA) and Finite State Transducer (FST) algorithms into autograd-based machine learning toolkits like PyTorch and TensorFlow.”

WEIGHTED FSA AND AUTODIFF

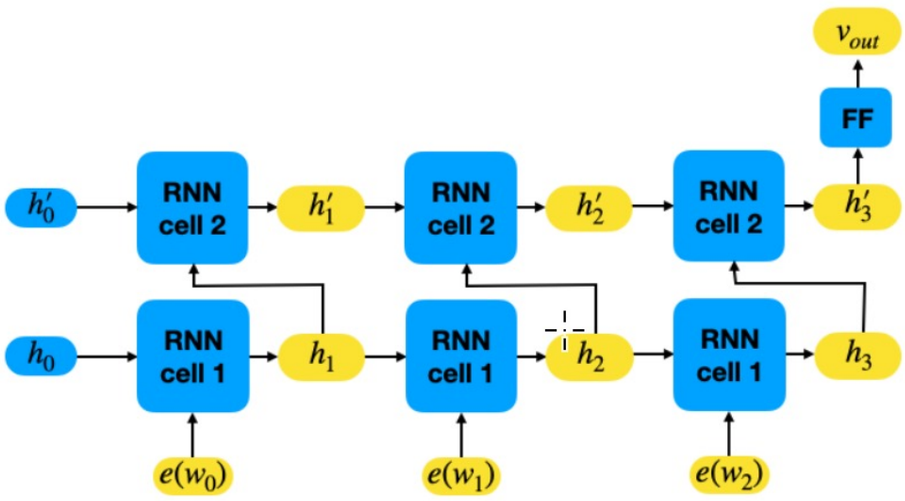
E104 **An Introduction to Automatic Differentiation with Weighted Finite-State Automata - Awni Hannun**

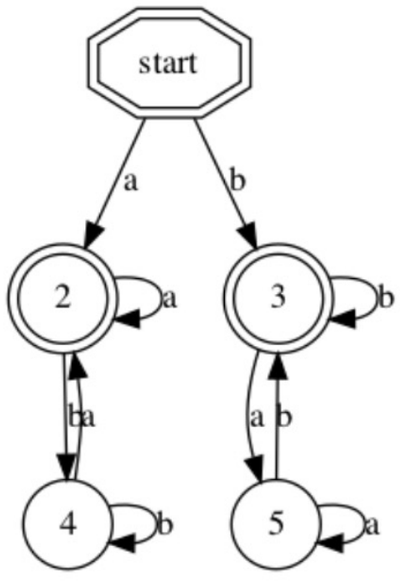
Weighted finite-state automata (WFSAs) have been a critical building block in modern automatic speech recognition. However, their use in conjunction with "end-to-end" deep learning systems is limited by the lack of efficient frameworks with support for automatic differentiation. This limitation is being overcome with the advent of new frameworks like GTN and k2. This tutorial will cover the basics of WFSAs and review their application in speech recognition. We will then explain the core concepts of automatic differentiation and show how to use it with WFSAs to rapidly experiment with new and existing algorithms. We will conclude with a discussion of the open challenges and opportunities for WFSAs to grow as a central component in automatic speech recognition and related applications.

https://awnihannun.com/writing/automata_ml.html

AUTOMATA-THEORETIC NEURAL NETS

Formal Abstractions of Neural Sequence Models





https://icgi2020.lis-lab.fr/speakers/#Gail

FINITE STATE TRANSDUCERS IN ASR

REFERENCES FOR WEIGHTED FSTS IN ASR

Moh97

Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311, 1997. URL: https://cs.nyu.edu/~mohri/pub/cl1.pdf.

MPR02

Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002. URL: https://cs.nyu.edu/~mohri/postscript/csl01.pdf.

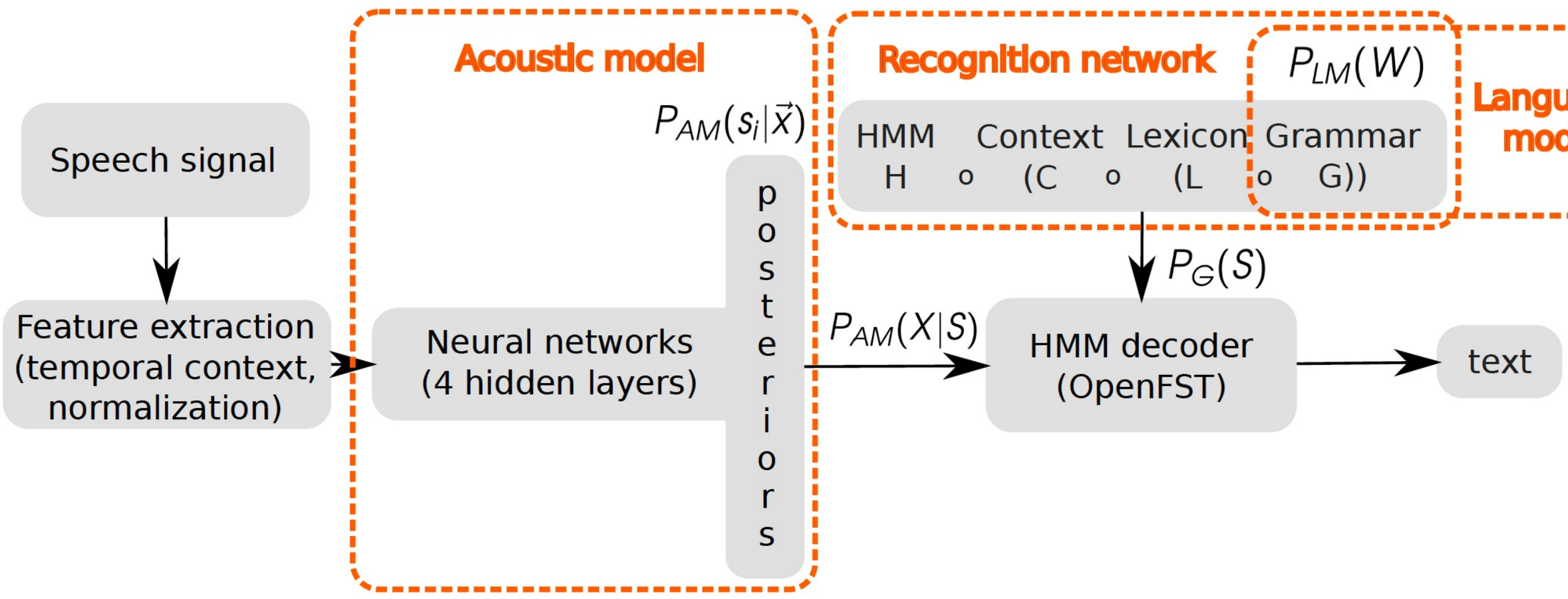
MPR08

Mehryar Mohri, Fernando Pereira, and Michael Riley. Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing*, pages 559–584. Springer, 2008. URL: https://wiki.eecs.yorku.ca/course_archive/2011-12/W/6328/_media/wfst-lvcsr.pdf.

https://k2-fsa.github.io/k2/core_concepts/index.html#a-simple-fsa-example

FST COMPOSITION

$$H \circ (C \circ (L \circ G))$$

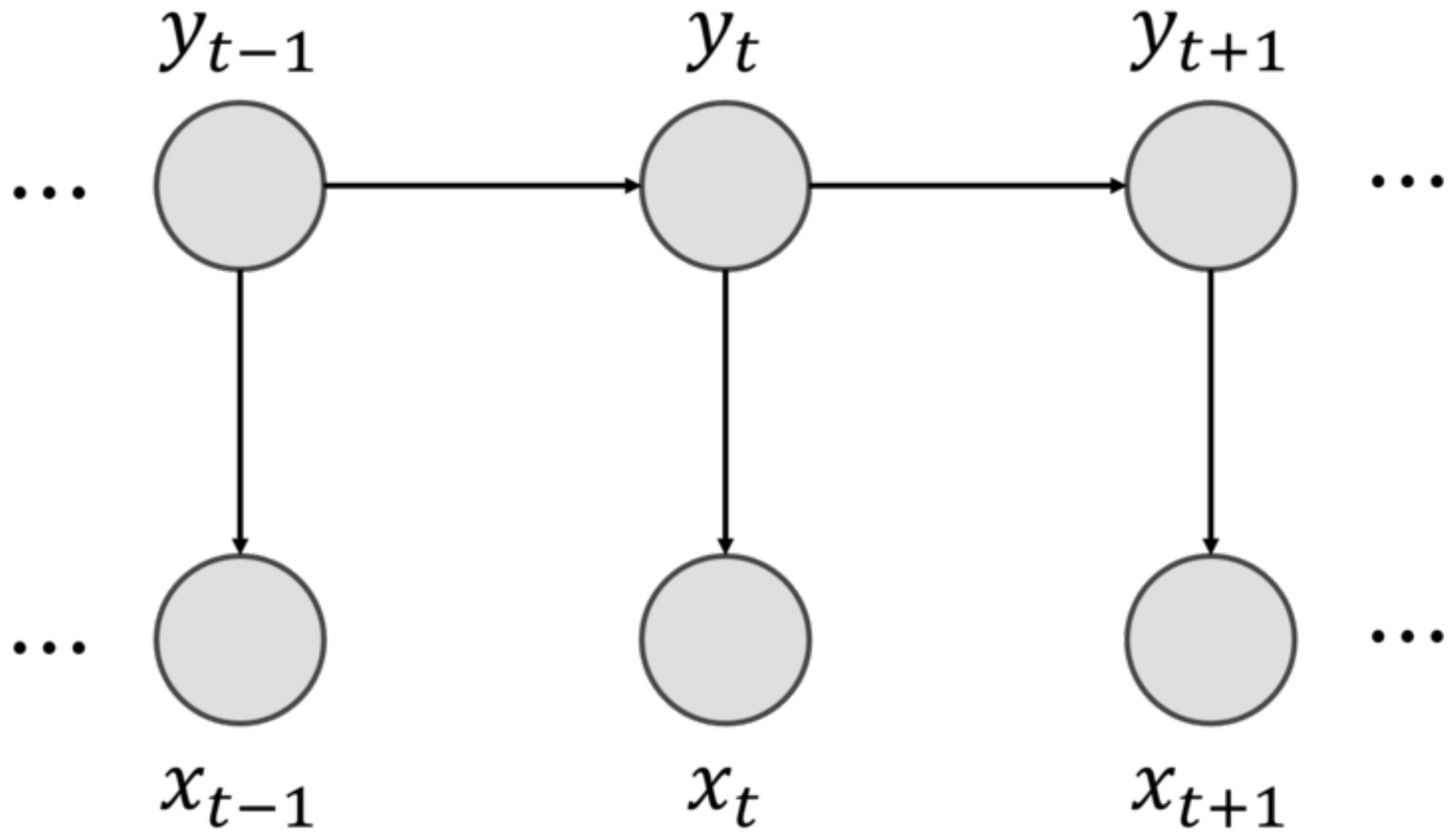


The decoding formula:

$$\hat{W} = \text{wrds}\left(\underset{S}{\text{argmax}}\, P_{AM}(\mathbf{X}|S)^{\kappa}\ P_G(S)^{\rho}\right)$$

http://www.fit.vutbr.cz/~grezl/ZRE/lectures/2019_KaldiSlidesForZRE.pdf

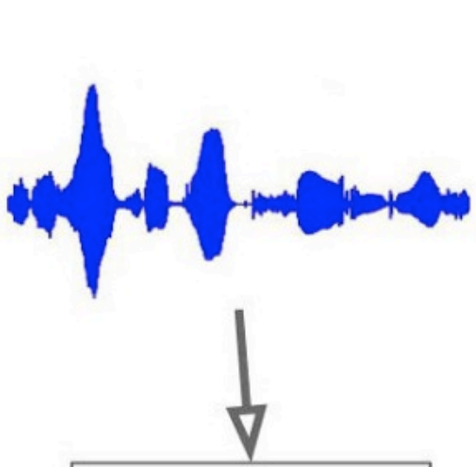
HIDDEN MARKOV MODEL: GRAPHICAL MODEL

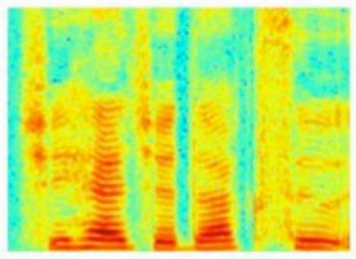


https://commons.wikimedia.org/wiki/File:Graphical_model_for_HMM.PNG

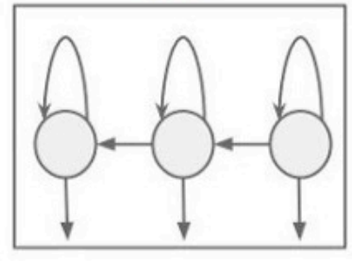
HMM: FINITE STATE AUTOMATON

$$H \circ (C \circ (L \circ G))$$





$\mathbf{X} = x_1\ x_2\ \ldots\ x_T$



Cat: /kæt/
Sits: /sɪts/
On: /ɔ:n
A: /ə/
Mat: /mæt/

W = "*cat sits on a mat*"

speech preprocessing → features ← acoustic models ← pronunciation models ← language models

Classical signal processing

Gaussian Mixture Models

Pronunciation tables

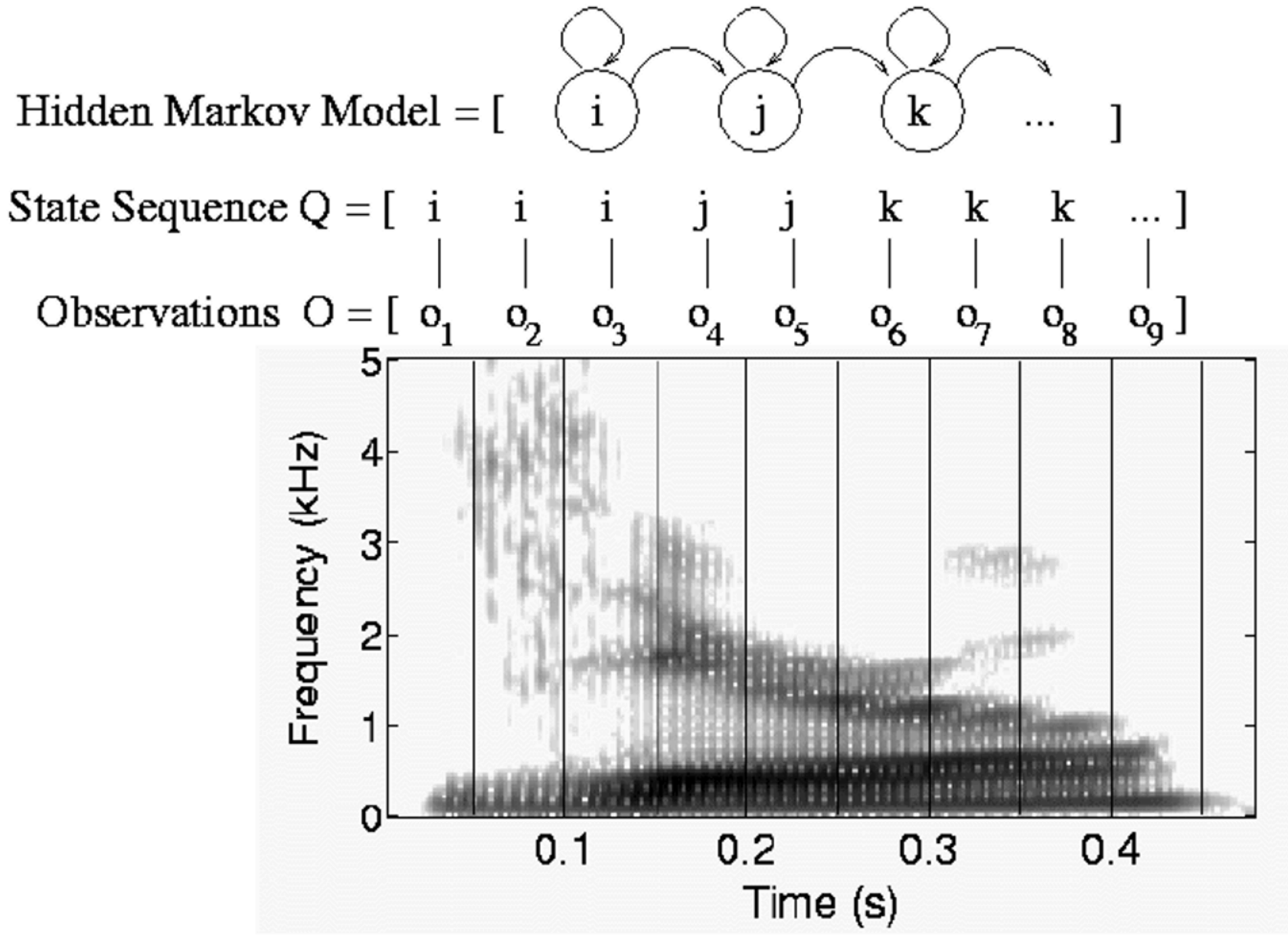
N-gram models

$$W^* = \arg\max_W p(\mathbf{X}|\mathbf{W})\ p(\mathbf{W})$$

acoustic model

language model

HMM FINITE STATE ACCEPTOR

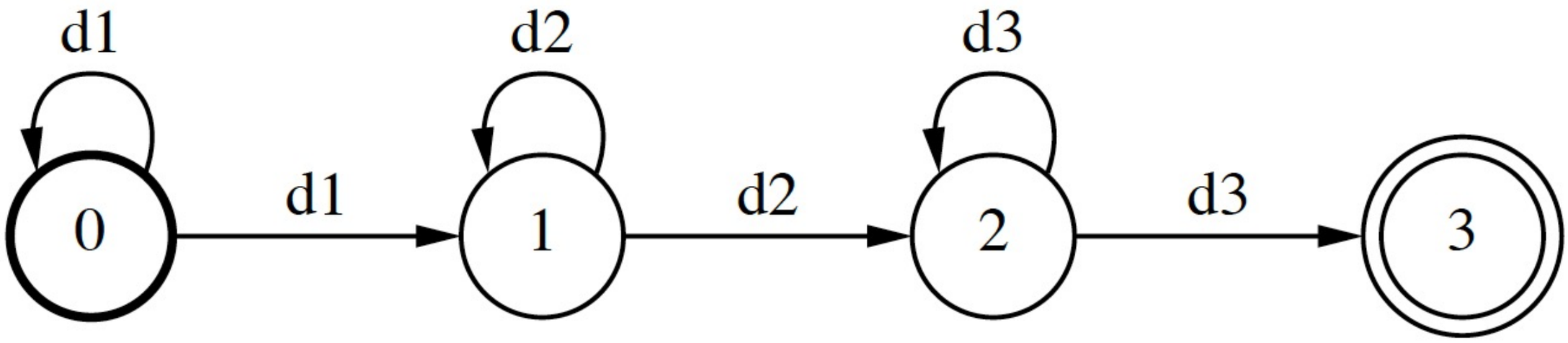


https://courses.physics.illinois.edu/ece417/fa2021/lectures/lec14.pdf

HMM FINITE STATE ACCEPTOR

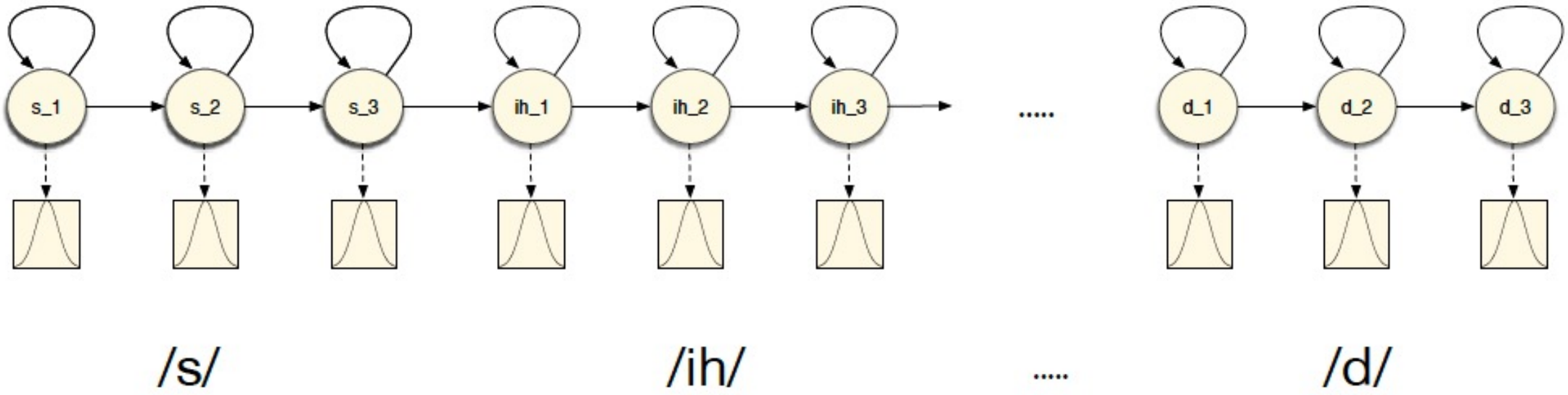
$$\boxed{H} \circ (C \circ (L \circ G))$$

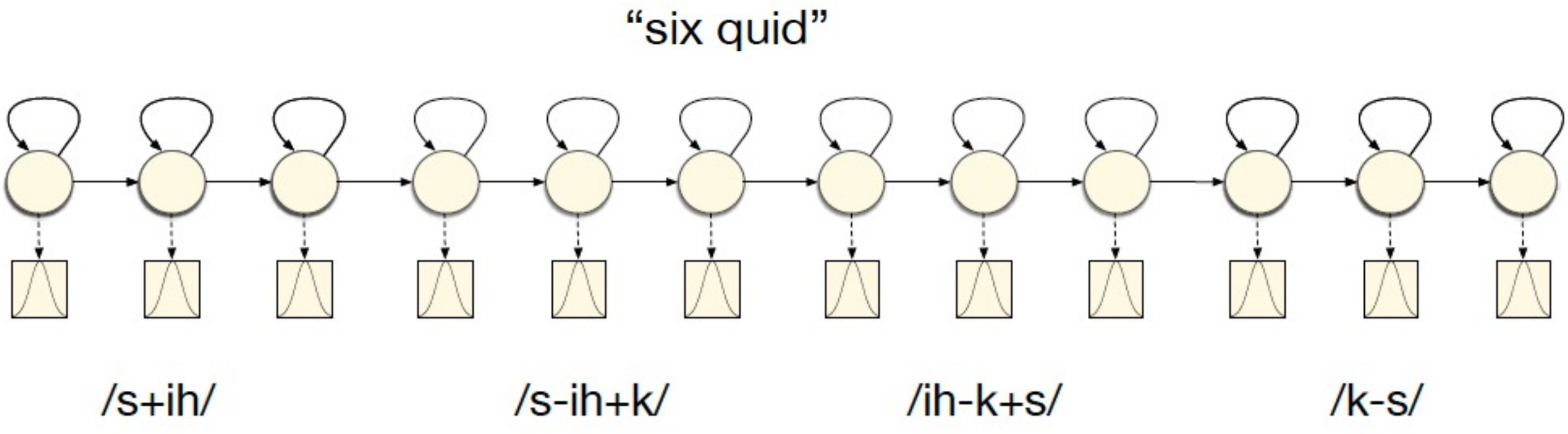
Beginning Middle End



Input: *d1d2d2d3*,
~~*d1d3d2d3*~~

CONTEXT INDEPENDENT VS. DEPENDENT

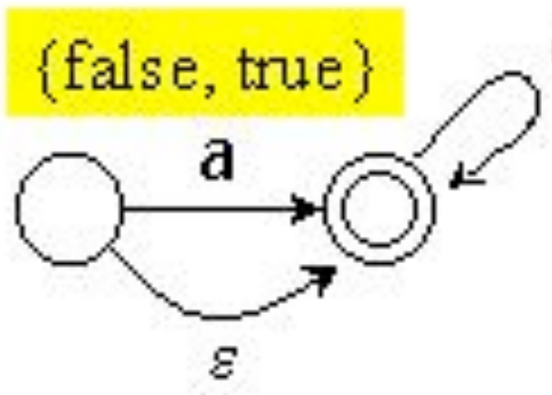
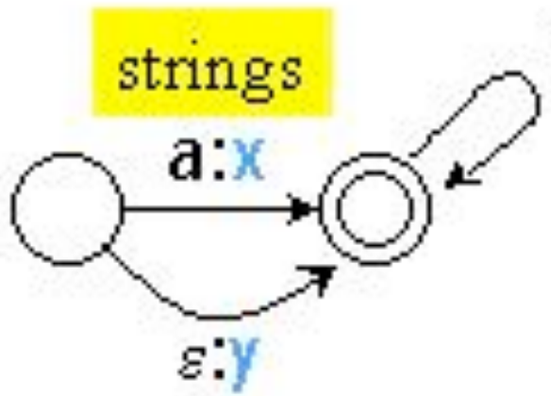
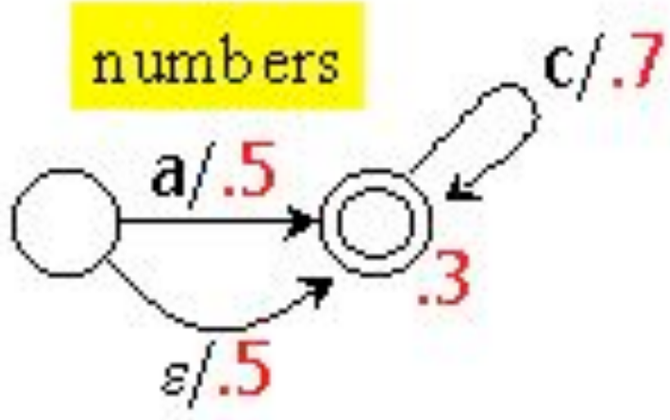
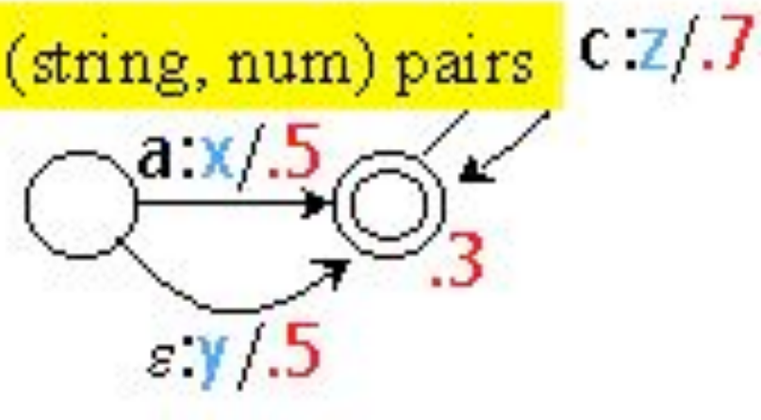




https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr07-cdhmm.pdf

FOUR TYPES OF FINITE STATE AUTOMATA

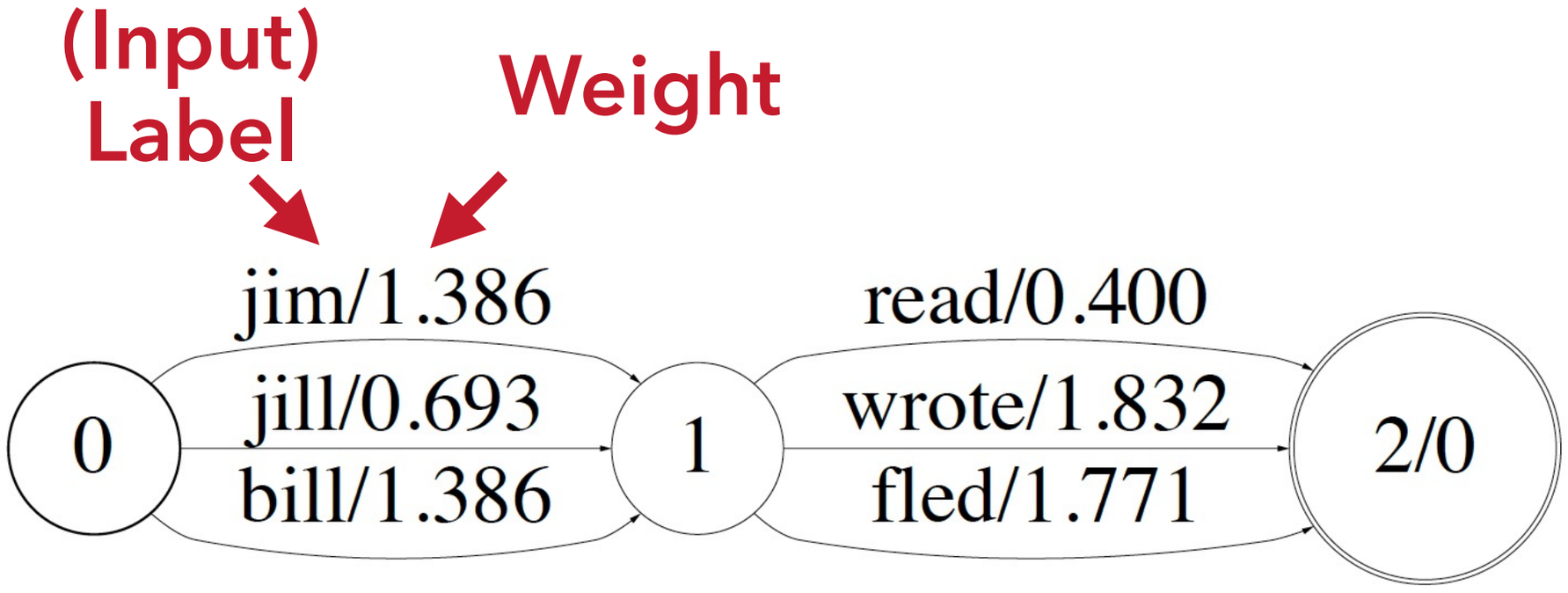
Function from strings to ...

	Acceptors	Transducers
Unweighted	{false, true} 	strings 
Weighted	numbers 	(string, num) pairs 

https://www.cs.jhu.edu/~jason/405/lectures1-2/sld065.htm

A FINITE STATE GRAMMAR

$$H \circ (C \circ (L \circ G))$$



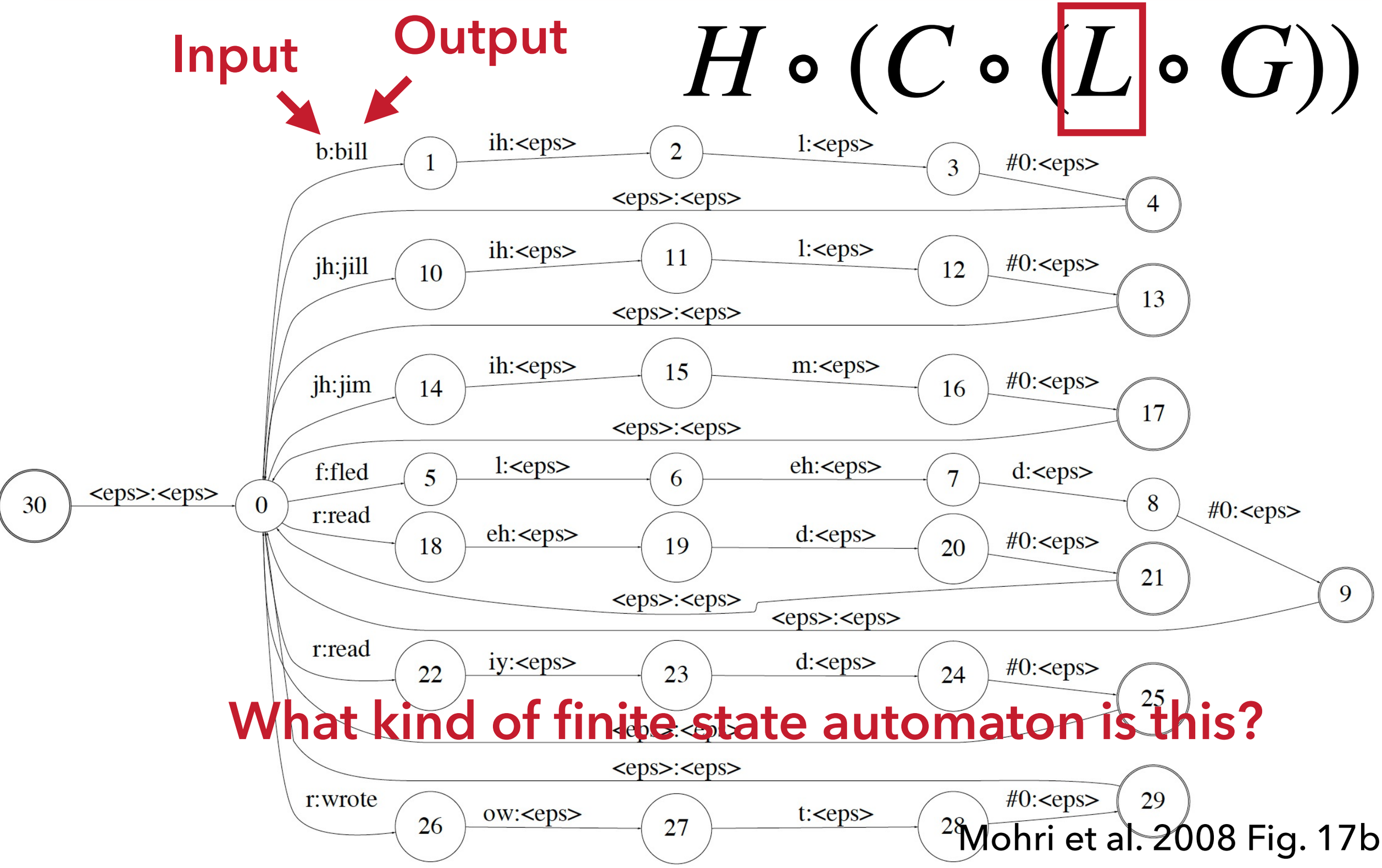
What kind of finite state automaton is this?

Mohri et al. 2008 Fig. 17a

A FINITE STATE LEXICON

Input Output

$$H \circ (C \circ (L \circ G))$$

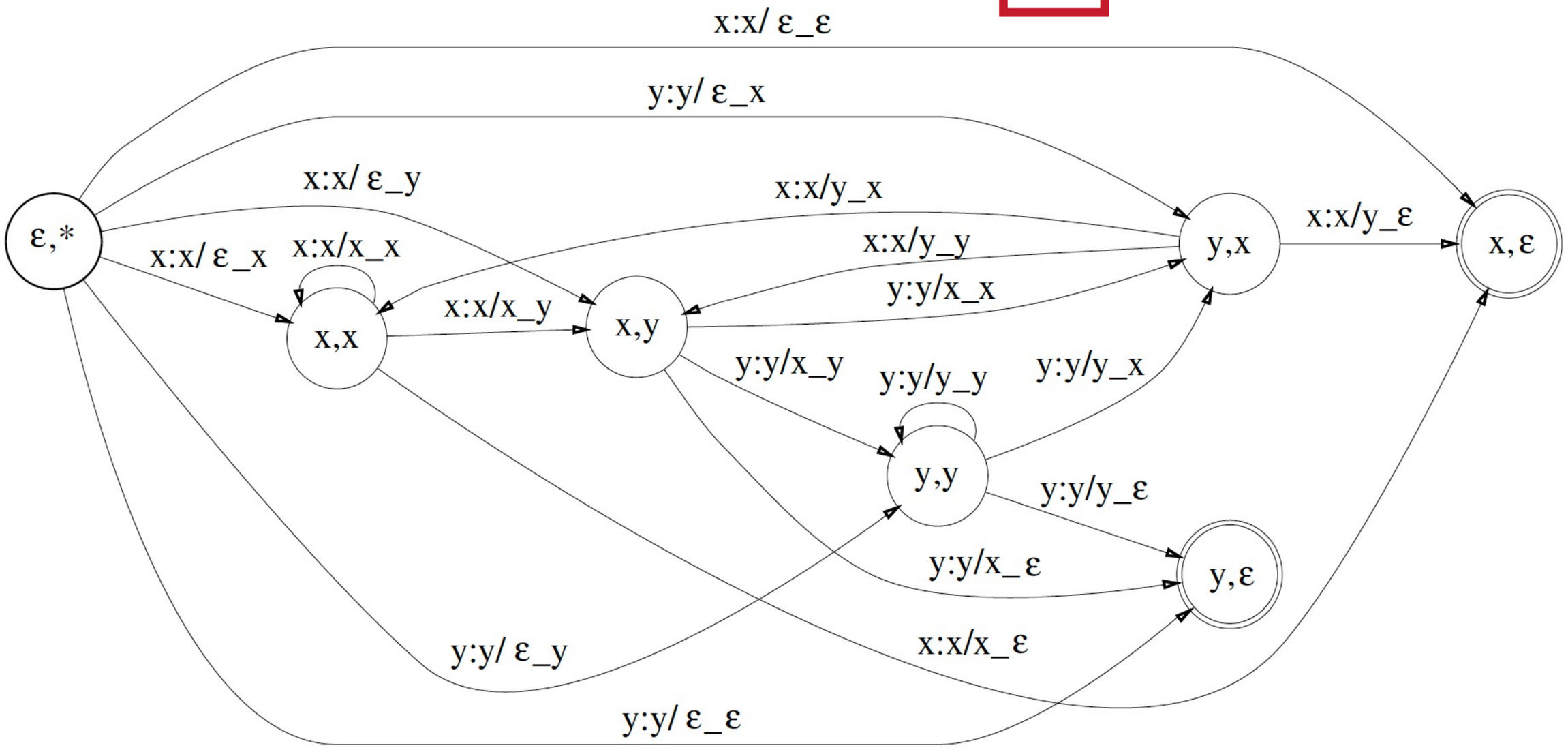


What kind of finite state automaton is this?

Mohri et al. 2008 Fig. 17b

A FINITE STATE CONTEXT-DEPENDENCY MAP

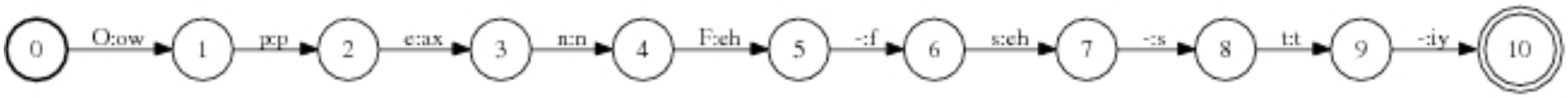
$$H \circ (C \circ (L \circ G))$$



What kind of finite state automaton is this?

Mohri et al. 2008 Fig. 15a

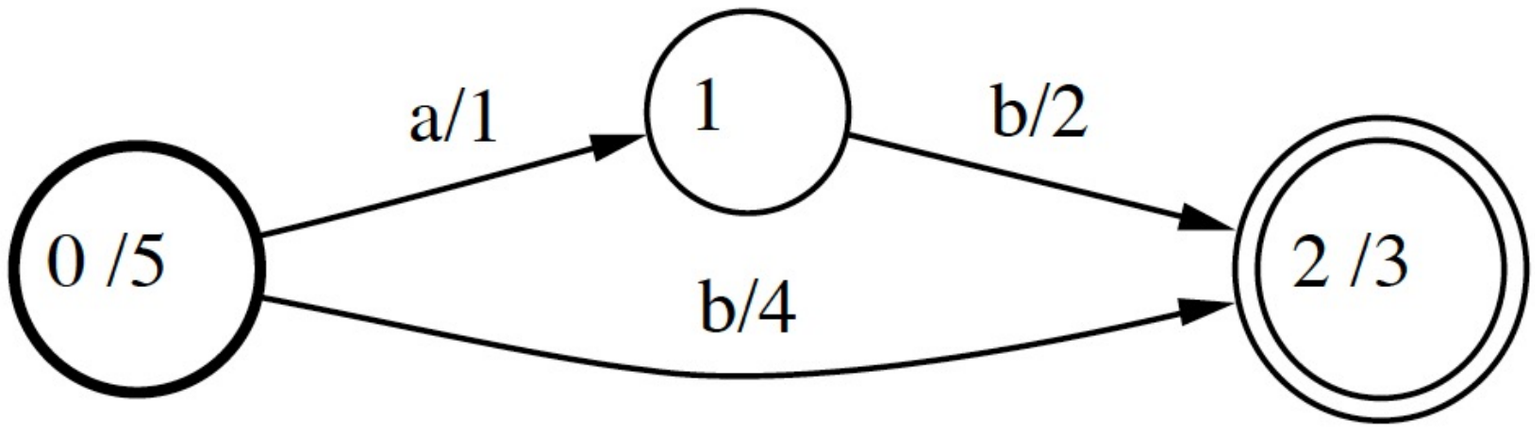
COMPUTATIONS USE OPEN FST



- See background materials at
- https://www.openfst.org/twiki/bin/view/FST/FstBackground

https://www.openfst.org/

WEIGHTED FINITE STATE ACCEPTOR



Input: *ab*

Weight: 5 + 1 + 2 + 3 = 11

- In ASR, weights typically probability or -log(probability)
- Computation with weights determined by algebraic structures called **semirings**

Mohri 1997, Figure 9

SEMIRINGS

No requirement for additive inverse!

A *semiring* $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ = a ring that may lack negation.

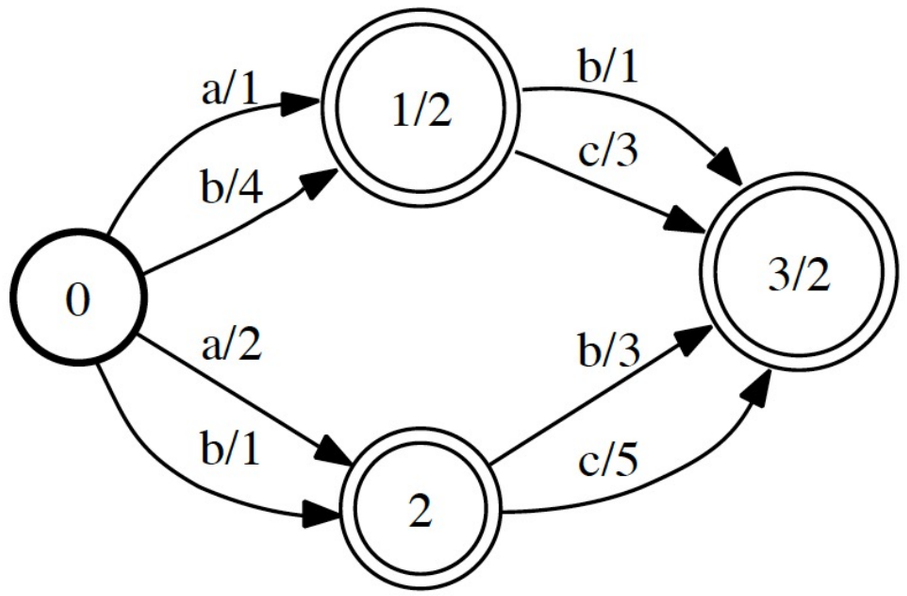
- Sum: to compute the weight of a sequence (sum of the weights of the paths labeled with that sequence).
- Product: to compute the weight of a path (product of the weights of constituent transitions).

SEMIRING	SET	$\oplus$	$\otimes$	$\overline{0}$	$\overline{1}$
Boolean	$\{0,1\}$	$\vee$	$\wedge$	0	1
Probability	$\mathbb{R}_+$ Or [0,1]!	$+$	$\times$	0	1
Log	$\mathbb{R} \cup \{-\infty, +\infty\}$	$\oplus_{\log}$	$+$	$+\infty$	0
Tropical	$\mathbb{R} \cup \{-\infty, +\infty\}$	min	$+$	$+\infty$	0
String	$\Sigma^* \cup \{\infty\}$	$\wedge$	$\cdot$	∞	ϵ

$\oplus_{\log}$ is defined by: $x \oplus_{\log} y = -\log(e^{-x} + e^{-y})$ and $\wedge$ is longest common prefix. The string semiring is a *left semiring*.

From https://www.openfst.org/twiki/pub/FST/FstHltTutorial/tutorial_part1.pdf

SEMIRING EXAMPLE: ACCEPTOR

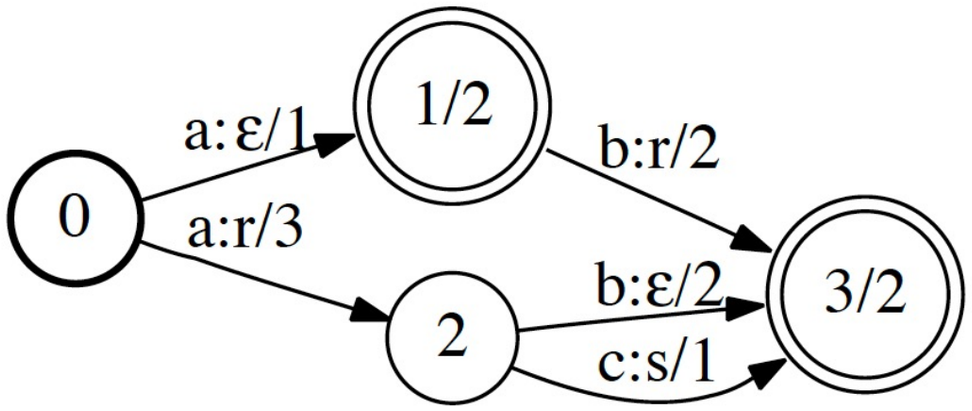


Probability semiring $(\mathbb{R}_+, +, \times, 0, 1)$	Tropical semiring $(\mathbb{R}_+ \cup \{\infty\}, \min, +, \infty, 0)$
$[\![A]\!](ab) = 14$	$[\![A]\!](ab) = 4$
$(1 \times 1 \times 2 + 2 \times 3 \times 2 = 14)$	$(\min(1 + 1 + 2, 3 + 2 + 2) = 4)$

From https://www.openfst.org/twiki/pub/FST/FstHltTutorial/tutorial_part1.pdf

SEMIRING EXAMPLE: TRANSDUCER

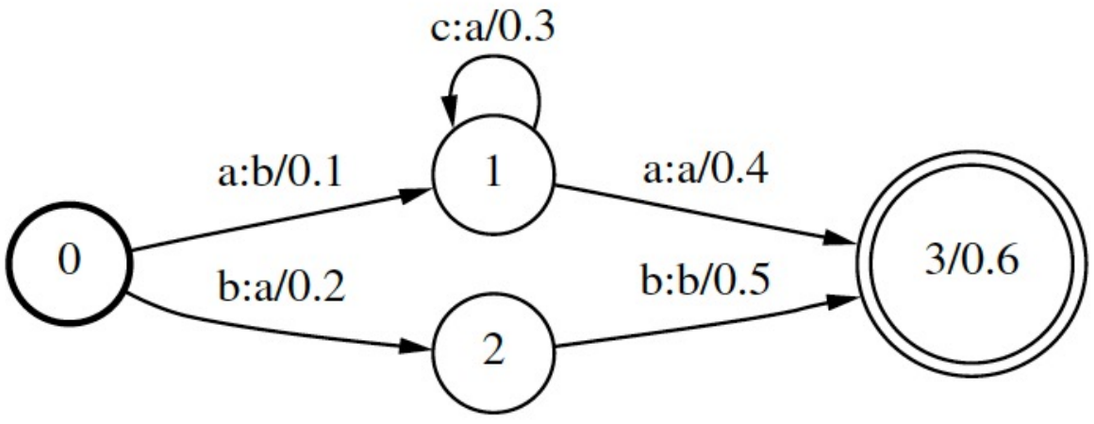
Weighted Transducer



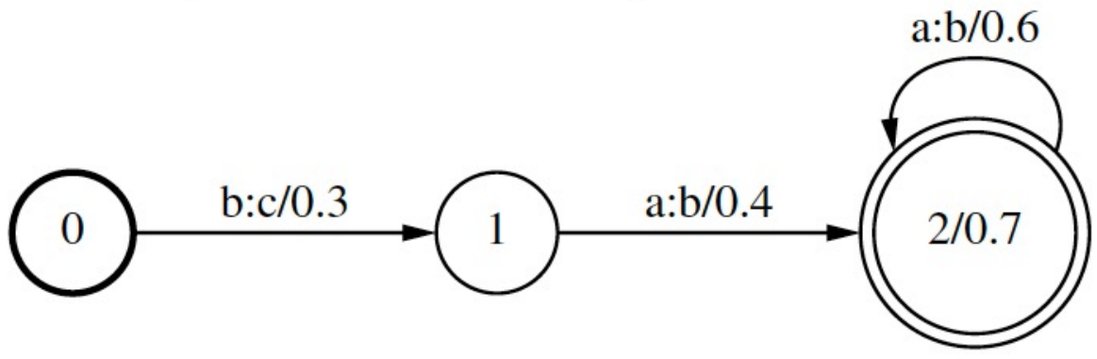
Probability semiring $(\mathbb{R}_+, +, \times, 0, 1)$	Tropical semiring $(\mathbb{R}_+ \cup \{\infty\}, \min, +, \infty, 0)$
$[\![T]\!](ab, r) = 16$	$[\![T]\!](ab, r) = 5$
$(1 \times 2 \times 2 + 3 \times 2 \times 2 = 16)$	$(\min(1 + 2 + 2, 3 + 2 + 2) = 5)$

From https://www.openfst.org/twiki/pub/FST/FstHltTutorial/tutorial_part1.pdf

COMPOSITION EXAMPLE



First, ignore weights



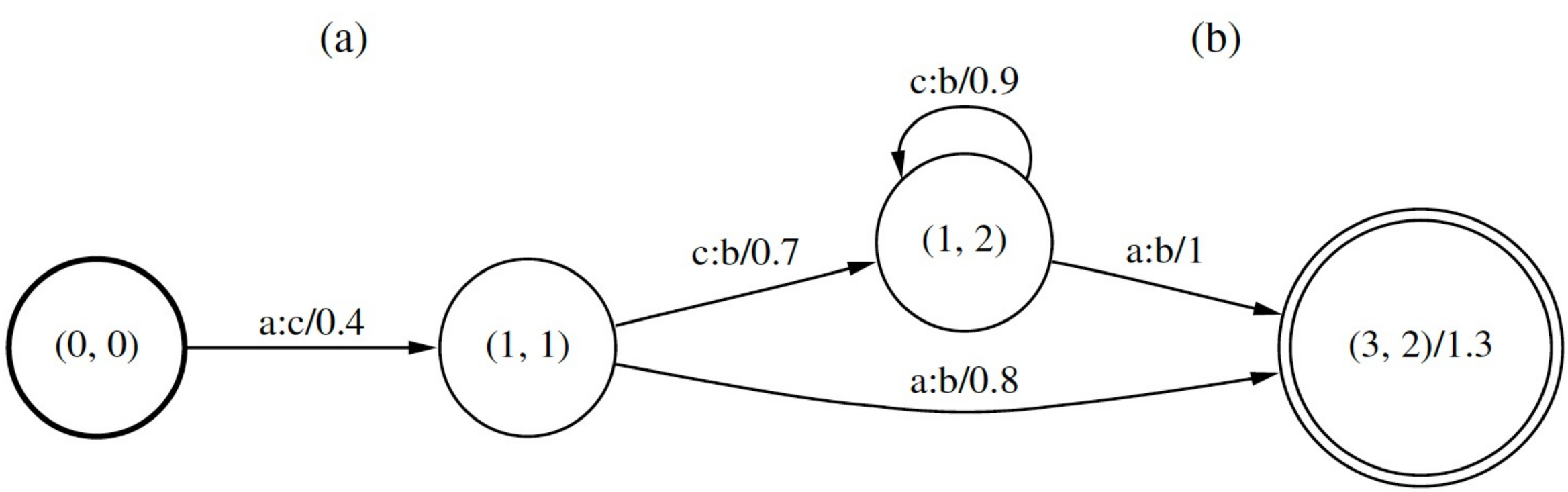


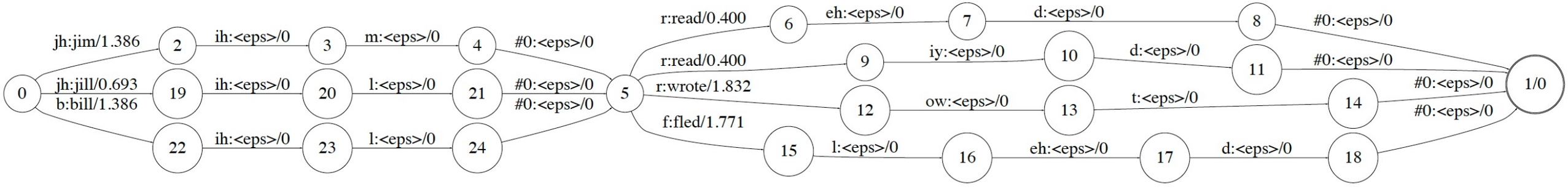
Figure 3. Example of transducer composition.

Mohri et al. 2002 Fig. 3

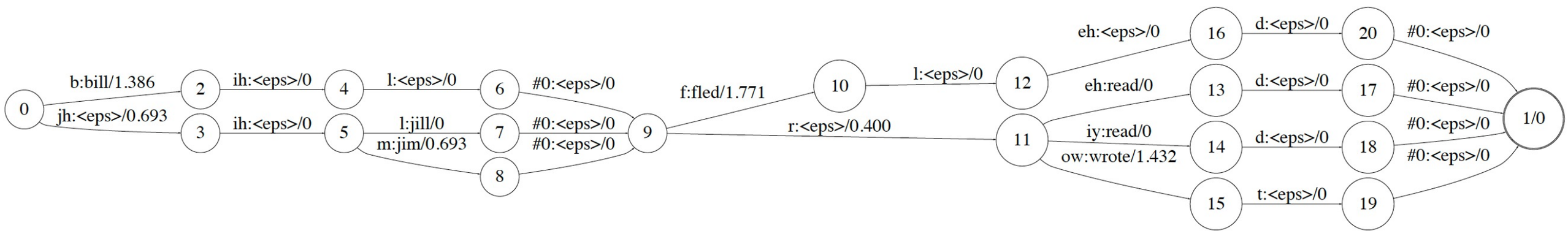
COMPOSITION OF LEXICON AND GRAMMAR

Weighted finite state transducer

$$L \circ G$$



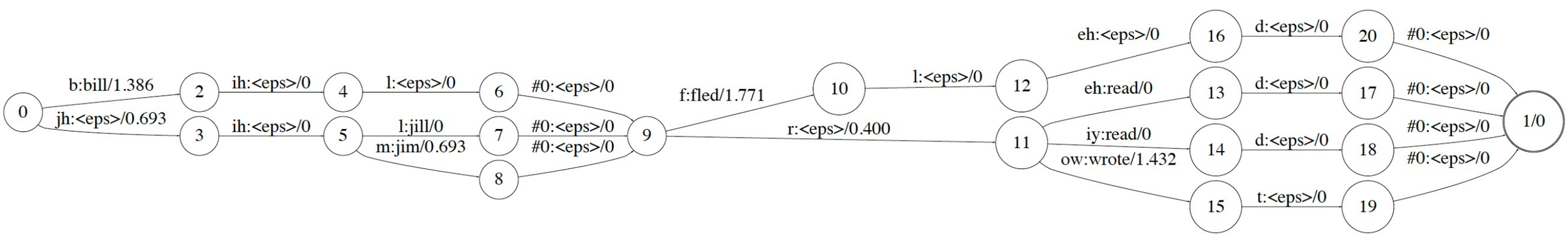
$$\det(L \circ G)$$



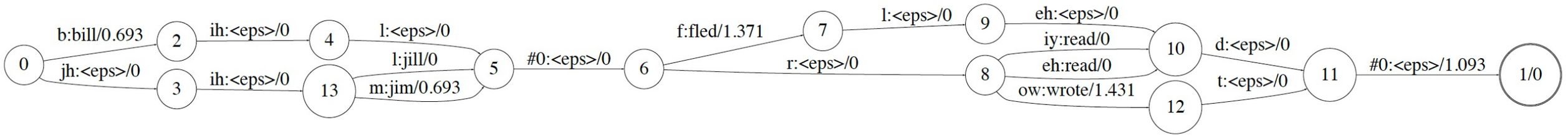
Mohri et al. 2008 Fig. 17c, d

DETERMINIZATION AND MINIMIZATION

$$\det(L \circ G)$$



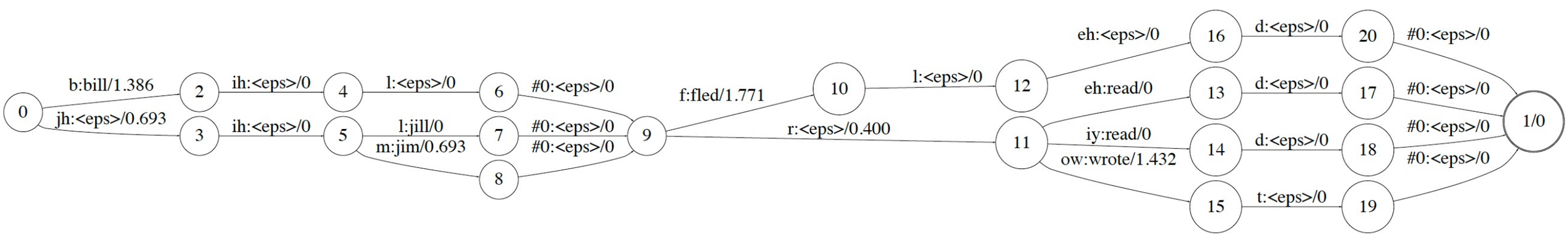
$$\min_{\text{trop}}(\det(L \circ G))$$



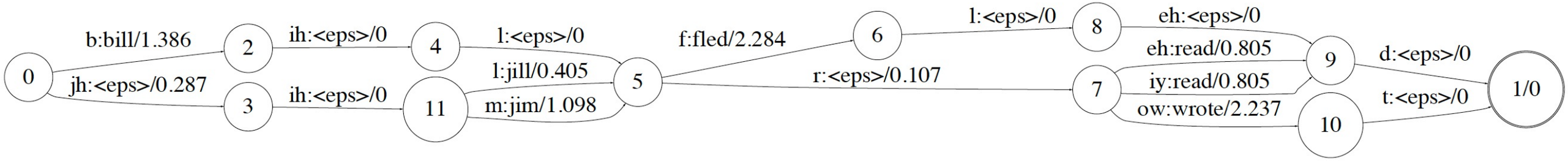
Mohri et al. 2008 Fig. 17d, e

DETERMINIZATION AND MINIMIZATION

$$\det(L \circ G)$$



$$\min_{\log}(\det(L \circ G))$$



Mohri et al. 2008 Fig. 17d, f

INCREASING COMPUTATIONAL EFFICIENCY

TABLE I. Size of the first-pass recognition transducers in the NAB 40 000-word vocabulary task

Transducer	States	Transitions
G	1 339 664	3 926 010
$L \circ G$	8 606 729	11 406 721
$\det(L \circ G)$	7 082 404	9 836 629
$C \circ \det(L \circ G)$	7 273 035	10 201 269
$\det(H \circ C \circ L \circ G)$	18 317 359	21 237 992
F	3 188 274	6 108 907
$\min(F)$	2 616 948	5 497 952

TABLE II. Recognition speed of the first-pass transducers in the NAB 40 000-word vocabulary task at 83% word accuracy

Transducer	×Real-time
$C \circ L \circ G$	12·5
$C \circ \det(L \circ G)$	1·2
$\det(H \circ C \circ L \circ G)$	1·0
$\min(F)$	0·7

Mohri et al. 2008

HMMS FOR NEURONS!

https://www.nature.com/articles/s41467-019-10994-4

Article | Open Access | Published: 30 July 2019

Real-time decoding of question-and-answer speech dialogue using human cortical activity

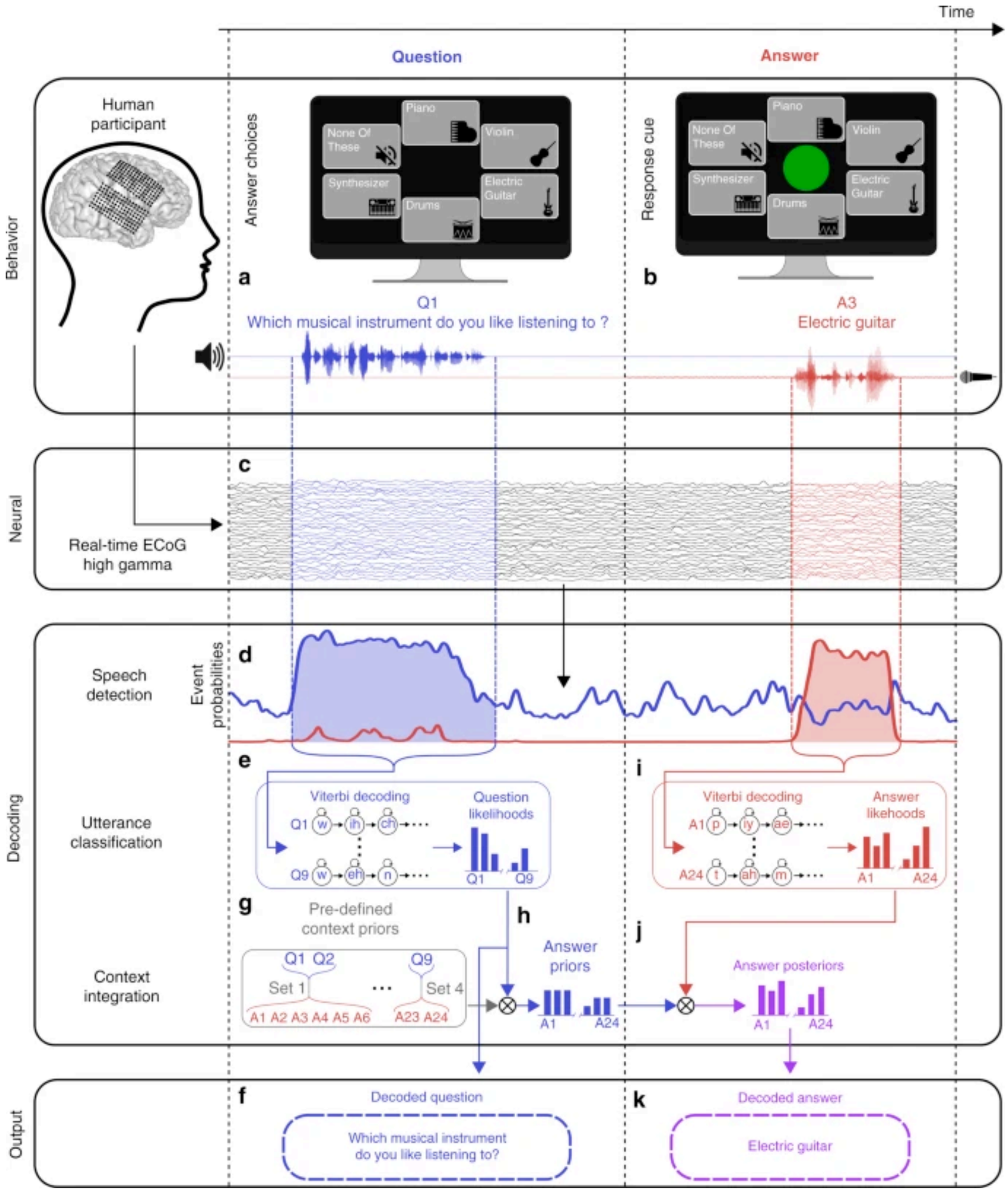
David A. Moses, Matthew K. Leonard, Joseph G. Makin & Edward F. Chang

Nature Communications **10**, Article number: 3096 (2019) | Cite this article

53k Accesses | **46** Citations | **1173** Altmetric | Metrics

Abstract

Natural communication often occurs in dialogue, differentially engaging auditory and sensorimotor brain regions during listening and speaking. However, previous attempts to decode speech directly from the human brain typically consider listening or speaking tasks in isolation. Here, human participants listened to questions and responded aloud with answers while we used high-density electrocorticography (ECoG) recordings to detect when they heard or said an utterance and to then decode the utterance's identity. Because certain answers were only plausible responses to certain questions, we could dynamically update the prior probabilities of each answer using the decoded question likelihoods as context. We decode produced and perceived utterances with accuracy rates as high as 61% and 76%, respectively (chance is 7% and 20%). Contextual integration of decoded question likelihoods significantly improves answer decoding. These results demonstrate real-time decoding of speech in an interactive, conversational setting, which has important implications for patients who are unable to communicate.

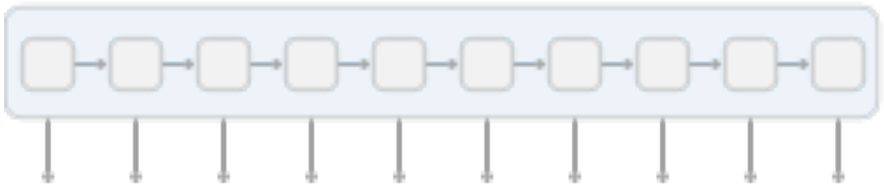

Time
Question
Answer
Behavior
Human participant
Answer choices
Response cue
Piano
None Of These
Violin
Synthesizer
Electric Guitar
Drums
a
b
Q1
Which musical instrument do you like listening to ?
A3
Electric guitar
Neural
c
Real-time ECoG high gamma
Decoding
Speech detection
Event probabilities
d
Utterance classification
e
i
Viterbi decoding
Question likelihoods
Answer likehoods
Q1 w ih ch
Q9 w eh n
A1 p iy ae
A24 t ah m
Context integration
g
Pre-defined context priors
h
j
Answer priors
Answer posteriors
Q1 Q2
Q9
Set 1
Set 4
A1 A2 A3 A4 A5 A6
A23 A24
Output
f
Decoded question
Which musical instrument do you like listening to?
k
Decoded answer
Electric guitar

https://www.nature.com/articles/s41467-019-10994-4

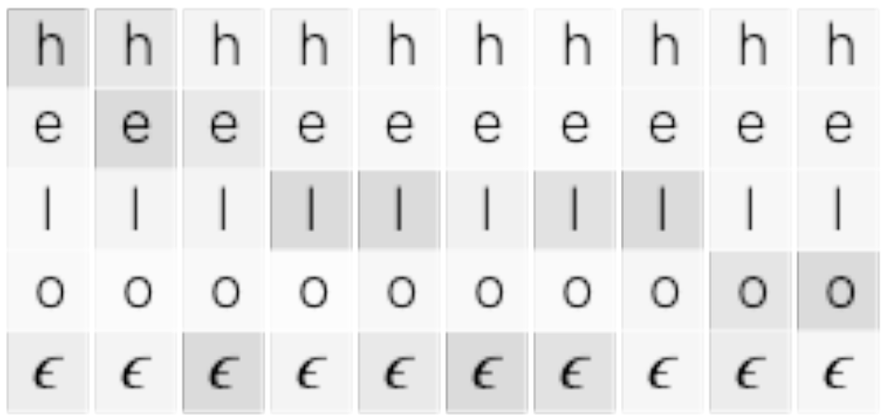
Our results were achieved with methods that have been used successfully in ASR research and applications[29,36], specifically Viterbi decoding with hidden Markov models (HMMs), except here we used neural activity as features during decoding instead of acoustic signals. We selected an HMM model architecture for several reasons, with arguably the most important being its inherent robustness to certain kinds of variability in the structure of speech. During Viterbi decoding, the answer classifiers were robust to variability in the exact duration and pronunciations of the produced answers because the amount of time each HMM could spend in each phone state was flexible. Similarly, both the question and answer classifiers were robust to slight inaccuracies in the detected speech onsets and offsets because each HMM started and ended with a silence state. The phone likelihood models underlying these utterance classifiers relied on discriminable phonetic encoding in the neural activity, which has been described in previous studies with both perceived[4,12] and produced[15,16,37] speech. Although other methods such as deep neural network modeling may be able to overcome these and other types of variability, the demonstrated methodologies we used allow for robust decoding of continuous speech from neural activity, including in data-limited settings such as clinical recordings with epilepsy patients.

CONNECTIONIST TEMPORAL CLASSIFICATION

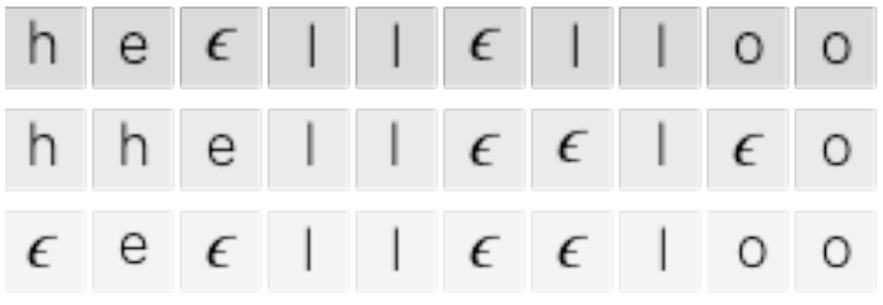
We start with an input sequence, like a spectrogram of audio.



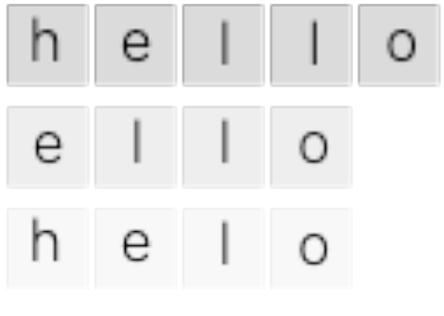
The input is fed into an RNN, for example.



The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ} for each input step.



With the per time-step output distribution, we compute the probability of different sequences



By marginalizing over alignments, we get a distribution over outputs

https://distill.pub/2017/ctc/