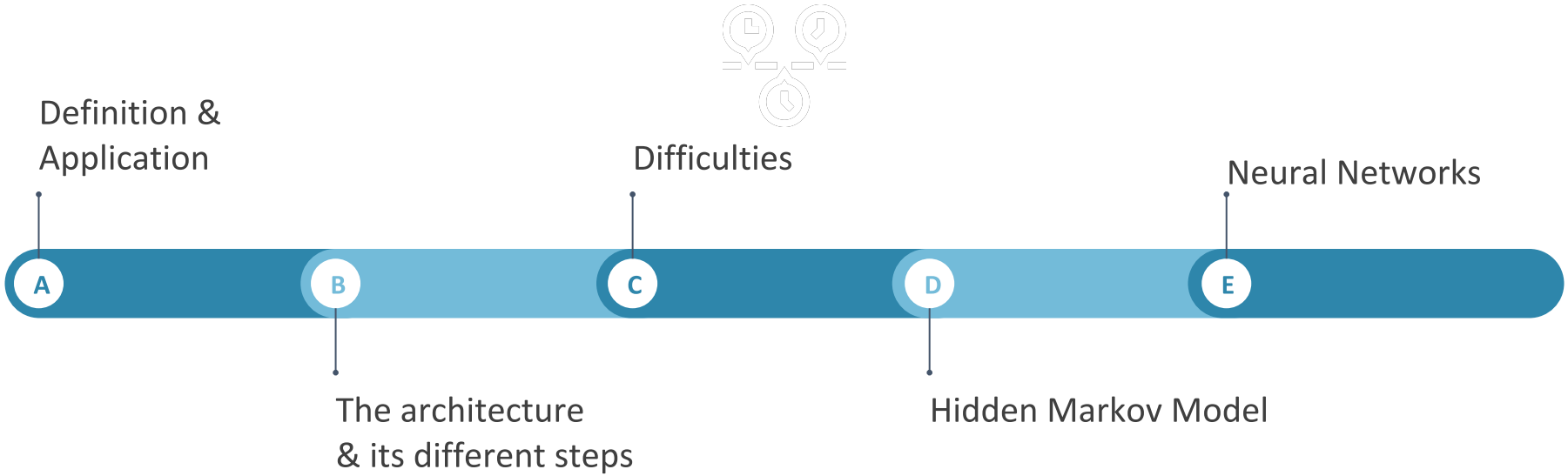
Automatic Speech Recognition

Machine Learning Seminar

Eva, Malte, Hsin-Yu

Outline


Definition &
Application
A
B
The architecture
& its different steps
Difficulties
C
D
Hidden Markov Model
Neural Networks
E

Definition & Application

“the process of converting a speech signal to a sequence of words (i.e., spoken words to text) by means of an algorithm implemented as a computer program”

Karpagavalli, Chandra

Dictation systems

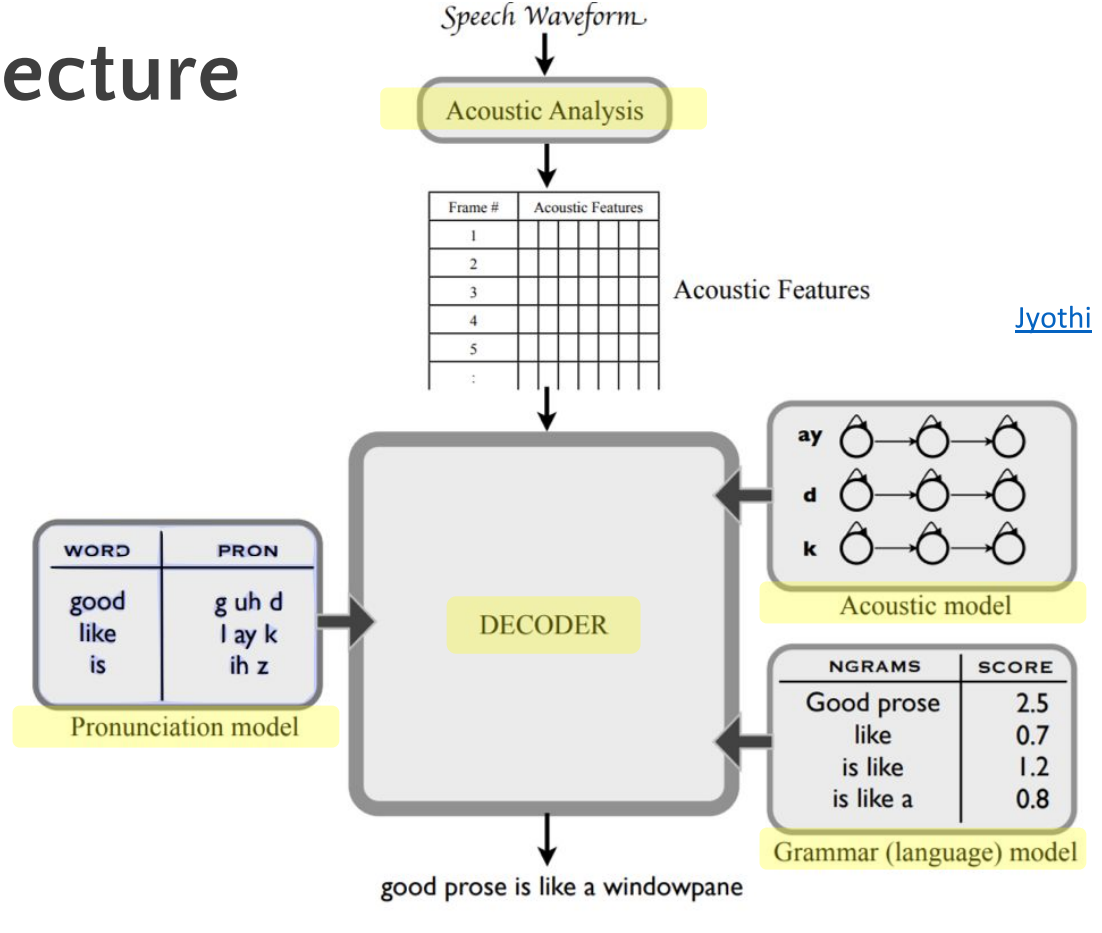
Youtube captioning

Call centers

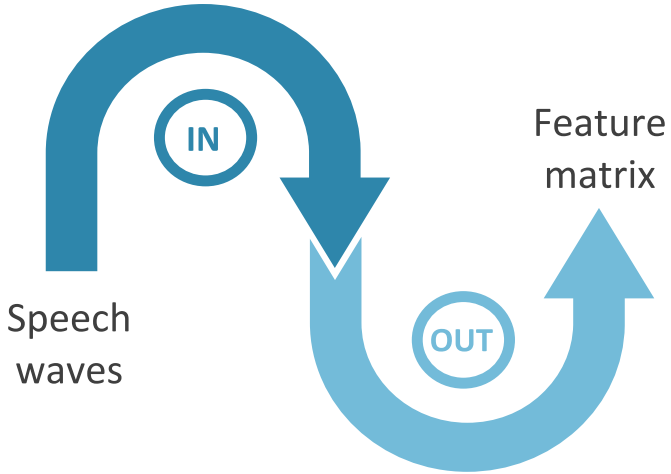
Voicemail transcription

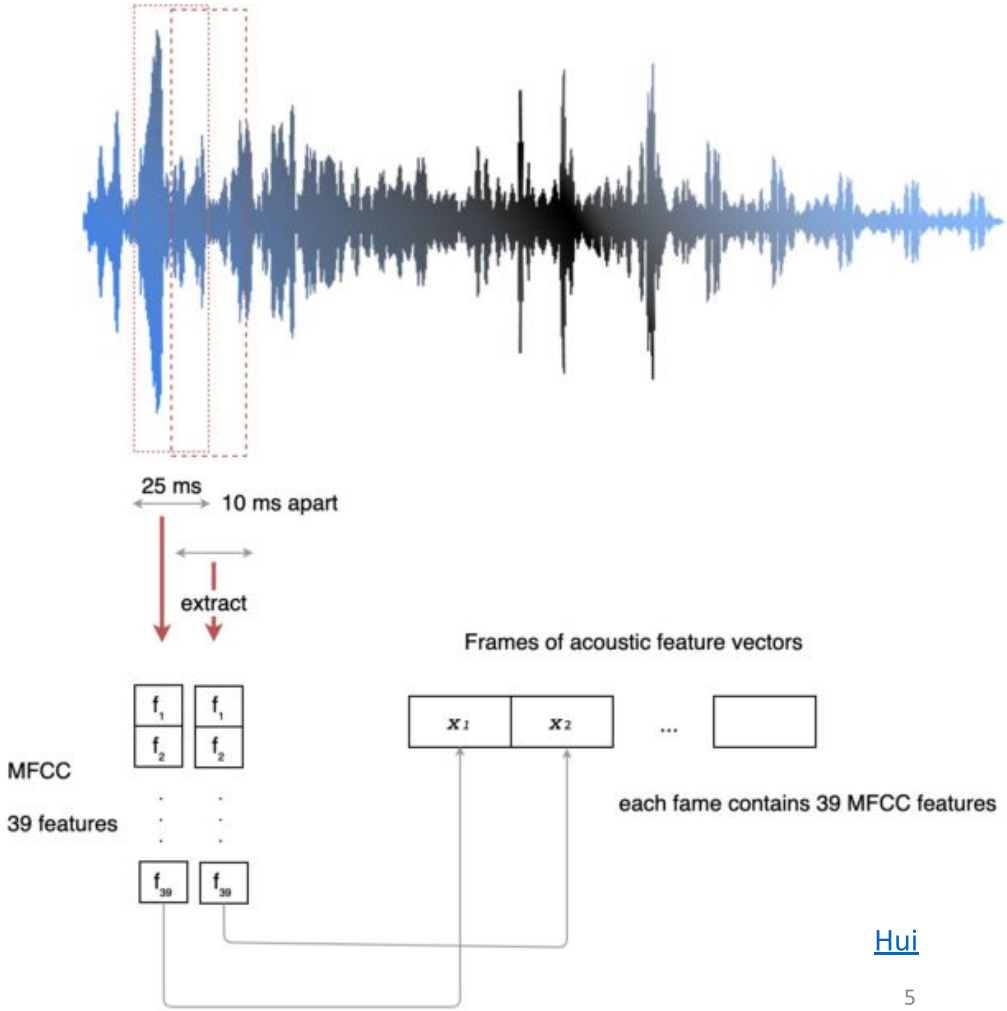
Virtual assistants like Siri, Alexa, ...

The architecture



Acoustic Analysis

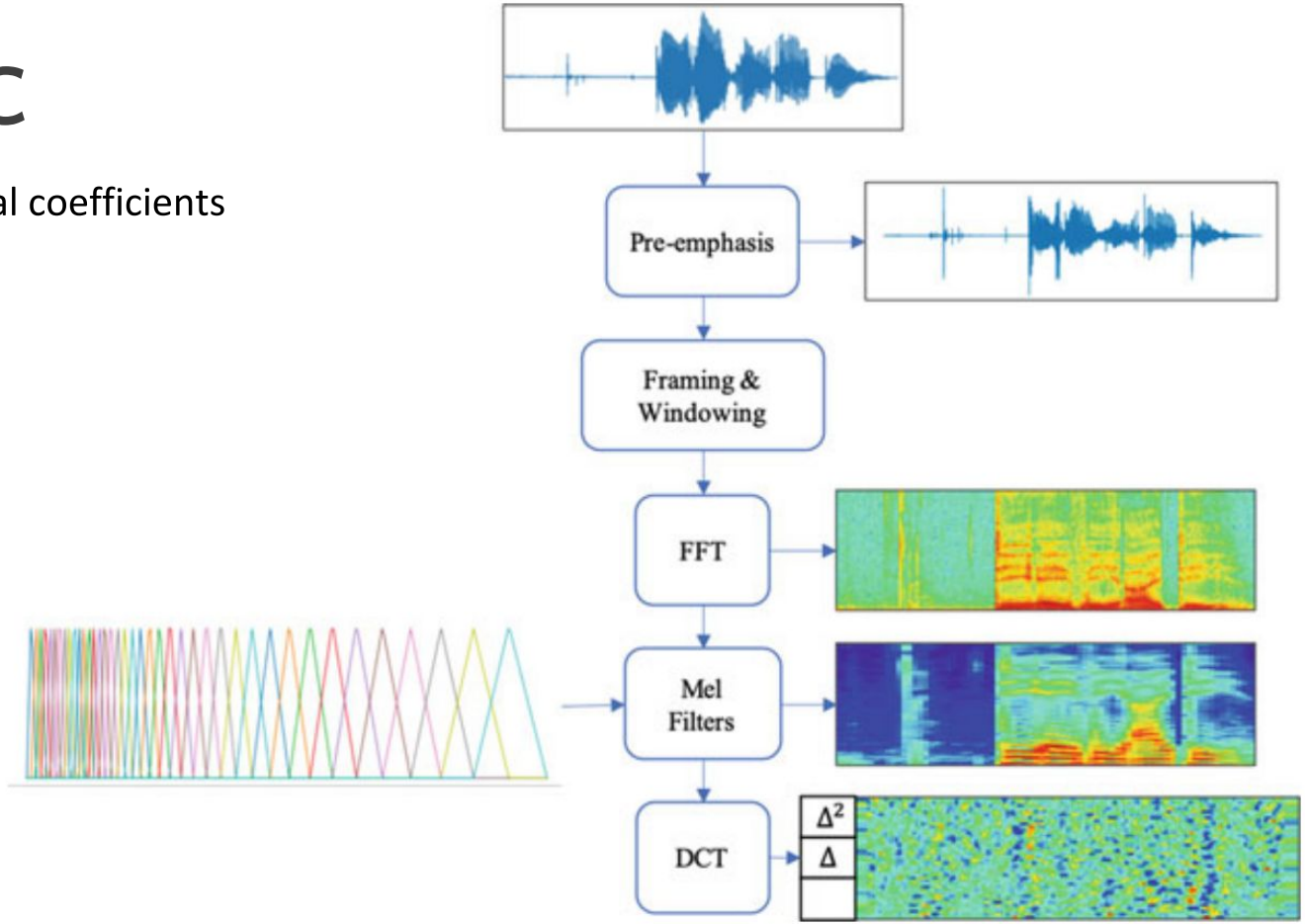




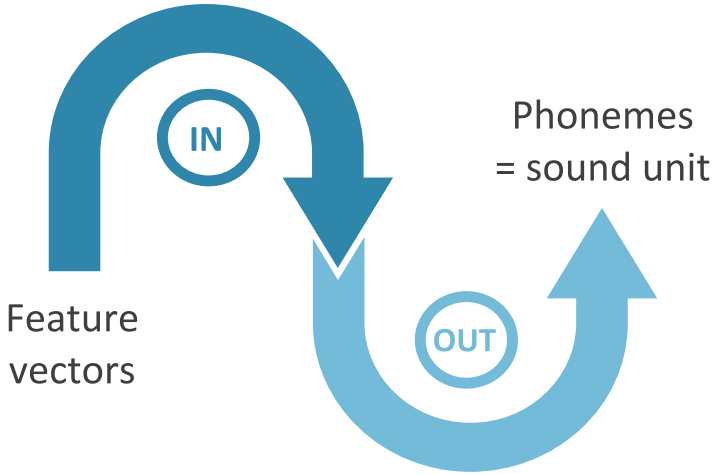
Hui

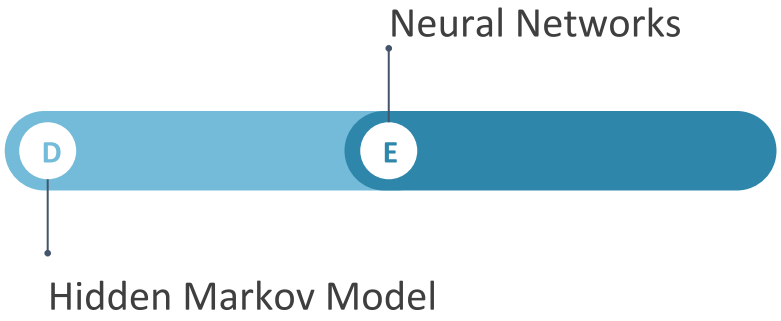
MFCC

Mel frequency cepstral coefficients



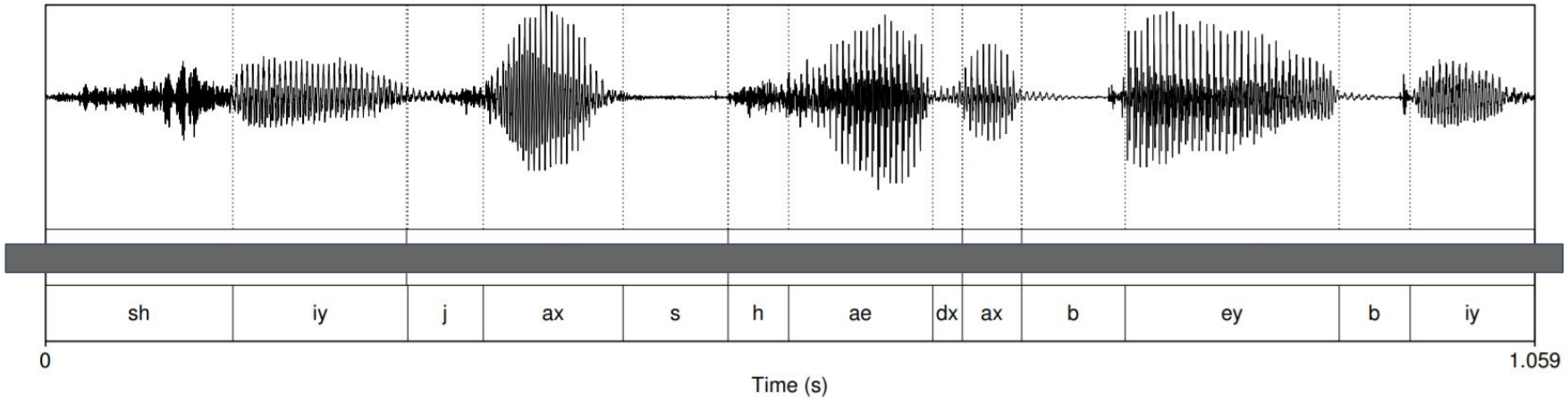
Acoustic Model



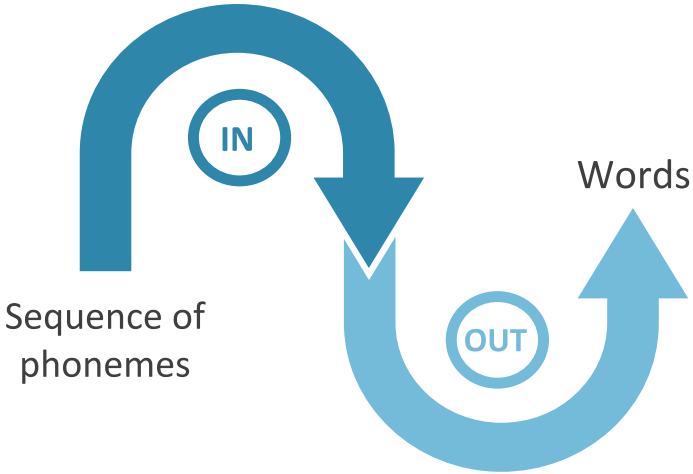


Acoustic Model

Jurafsky, Martin



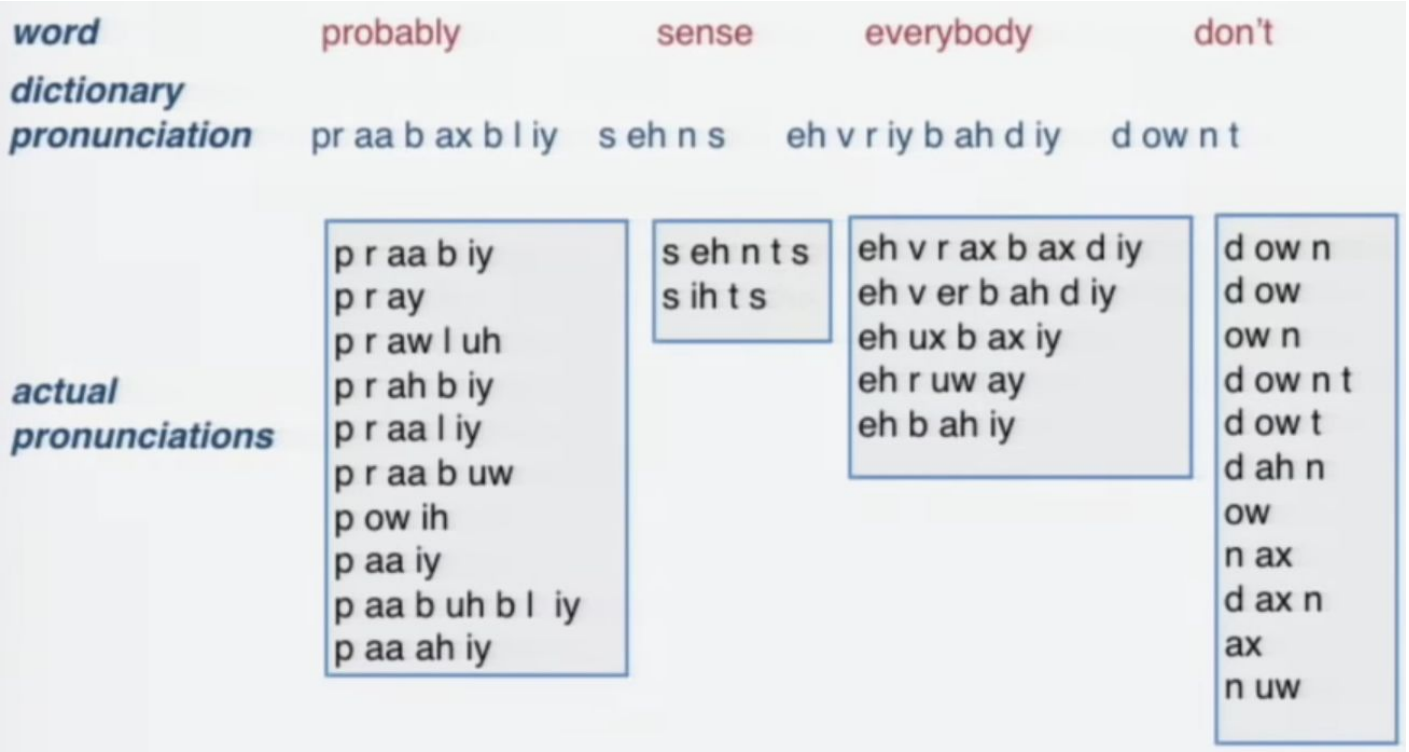
Pronunciation Model



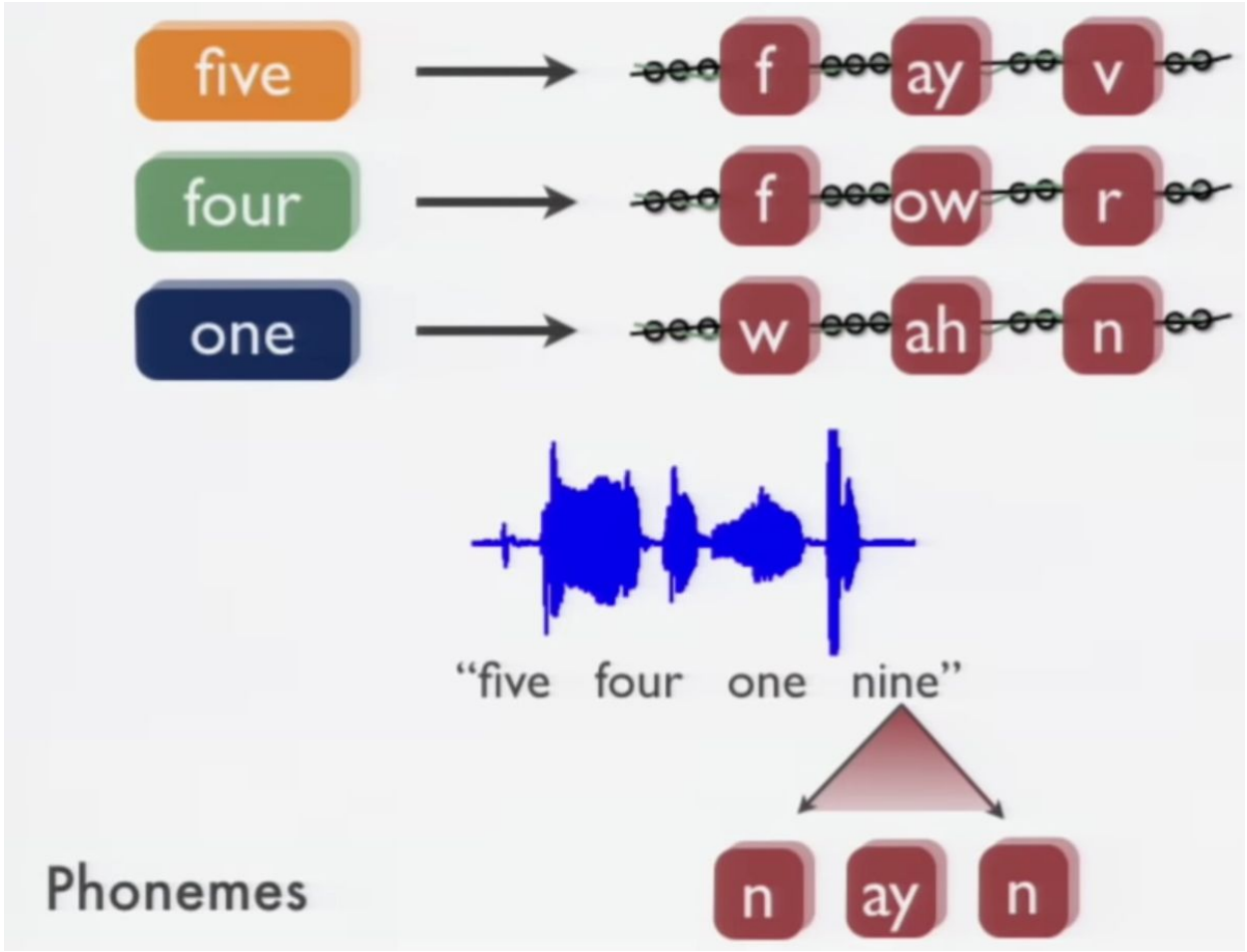
Word	ARPAbet Transcription
parsley	[p aa r s l iy]
tea	[t iy]
cook	[k uh k]
bay	[b ey]
dill	[d ih l]
garlic	[g aa r l ix k]
mint	[m ih n t]
nutmeg	[n ah t m eh g]
baking	[b ey k ix ng]
flour	[f l aw axr]
clove	[k l ow v]
thick	[th ih k]
those	[dh ow z]
soup	[s uw p]
eggs	[eh g z]
squash	[s k w aa sh]
ambrosia	[ae m b r ow zh ax]
cherry	[ch eh r iy]
jar	[jh aa r]
licorice	[l ih k axr ix sh]
kiwi	[k iy w iy]
rice	[r ay s]
yellow	[y eh l ow]
honey	[h ah n iy]

Jurafsky, Martin

Pronunciation Model

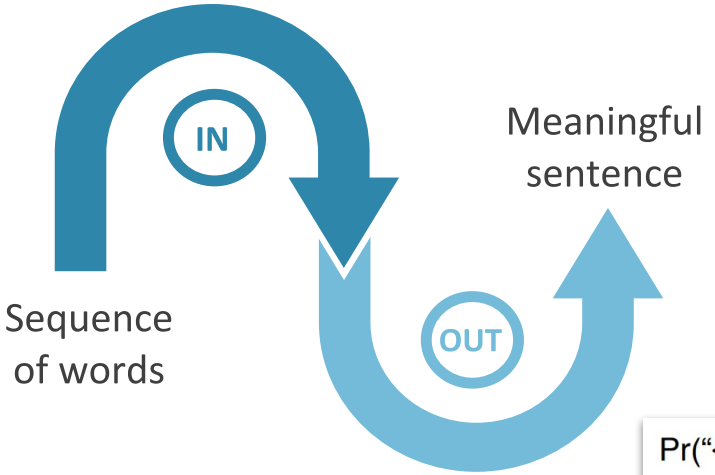


Words vs. Phonemes



Language Model

Jyothi



GRAMMAR

$\Pr(\text{“she class taught a”}) < \Pr(\text{“she taught a class”})$

VOCABULARY

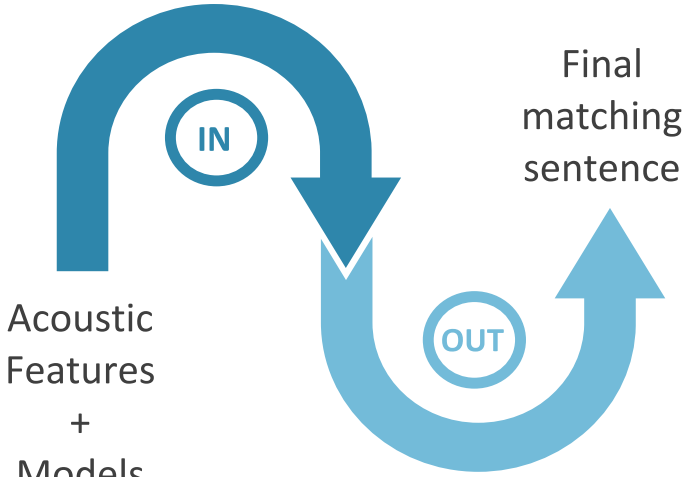
$\Pr(\text{“she taught a class”}) > \Pr(\text{“she taught a speech”})$

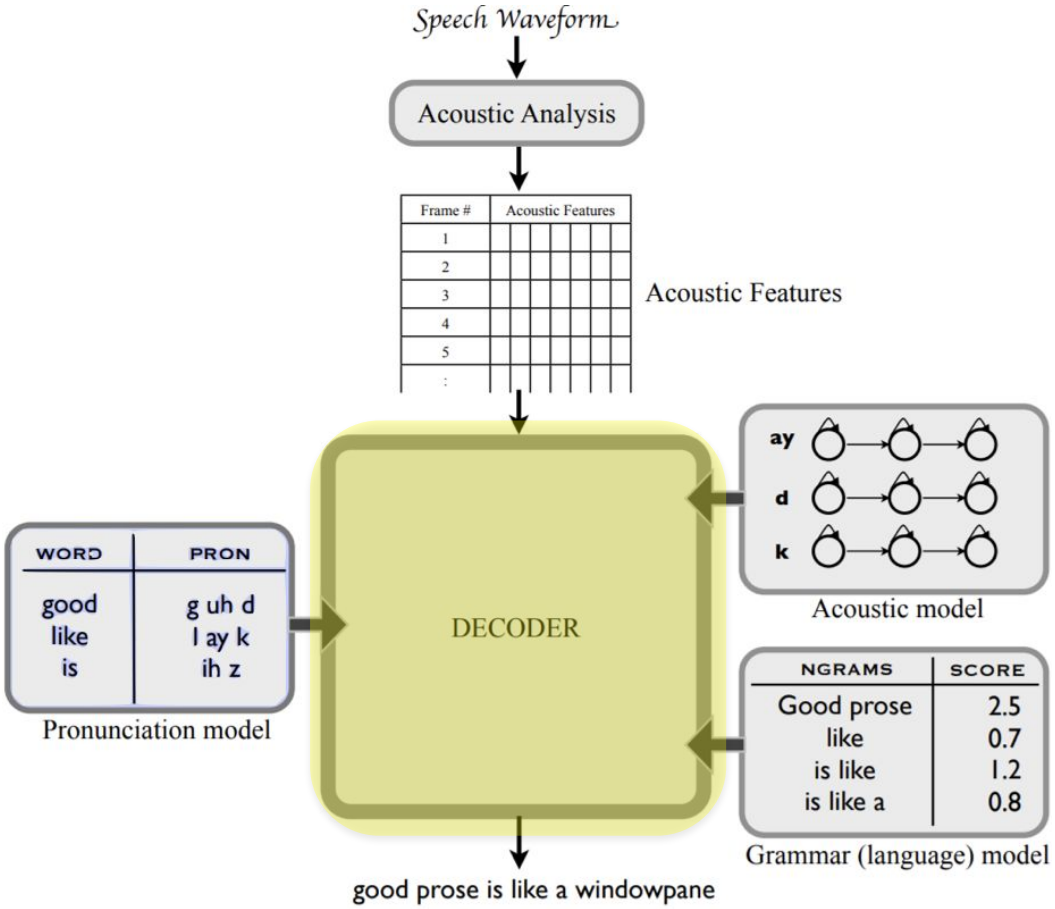
NGRAM MODEL

$\Pr(\text{“<s> The cat chased a mouse </s>”}) =$

$\Pr(\text{“The}|\text{<s>”}) \cdot \Pr(\text{“cat}|\text{The”}) \cdot \Pr(\text{“chased}|\text{cat”}) \cdot \Pr(\text{“a}|\text{chased”}) \cdot \Pr(\text{“mouse}|\text{a”}) \cdot \Pr(\text{“</s>}|\text{mouse”})$

Decoder


IN
OUT
Acoustic Features + Models
Final matching sentence


Speech Waveform
Acoustic Analysis
Frame #
Acoustic Features
1
2
3
4
5
Acoustic Features
ay
d
k
Acoustic model
WORD
PRON
good
like
is
g uh d
l ay k
ih z
Pronunciation model
DECODER
NGRAMS
SCORE
Good prose
like
is like
is like a
2.5
0.7
1.2
0.8
Grammar (language) model
good prose is like a windowpane

Difficulties

Change of language within a speech input

Speaker characteristics

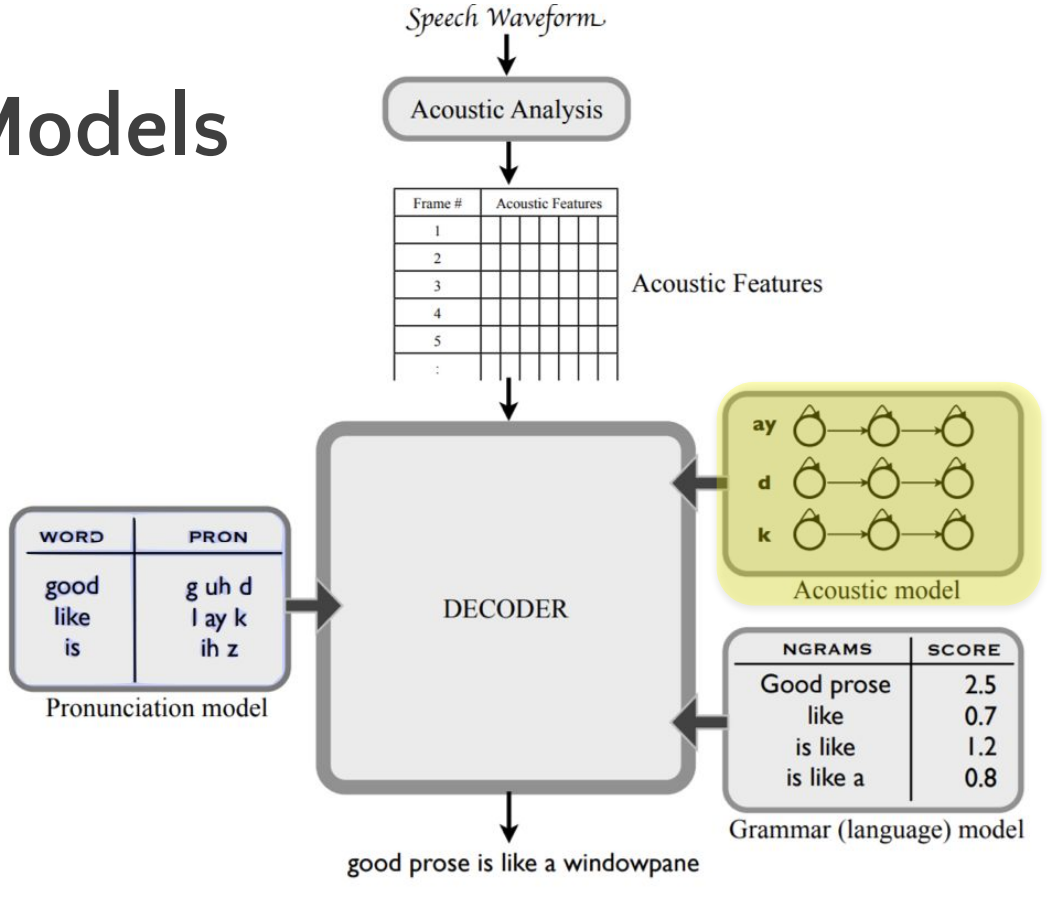
Application to a new language

Noisy settings

Pronunciation variability

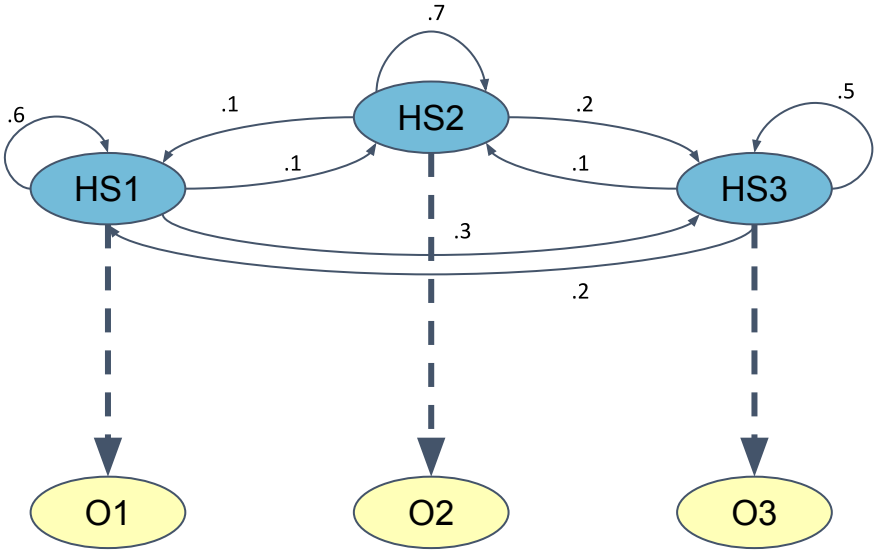
Hidden Markov Models (HMMs) in ASR

Relevant in the acoustic model



Hidden Markov Models (HMMs)

What are HMMs?



.5 Transition probabilities

Hidden states

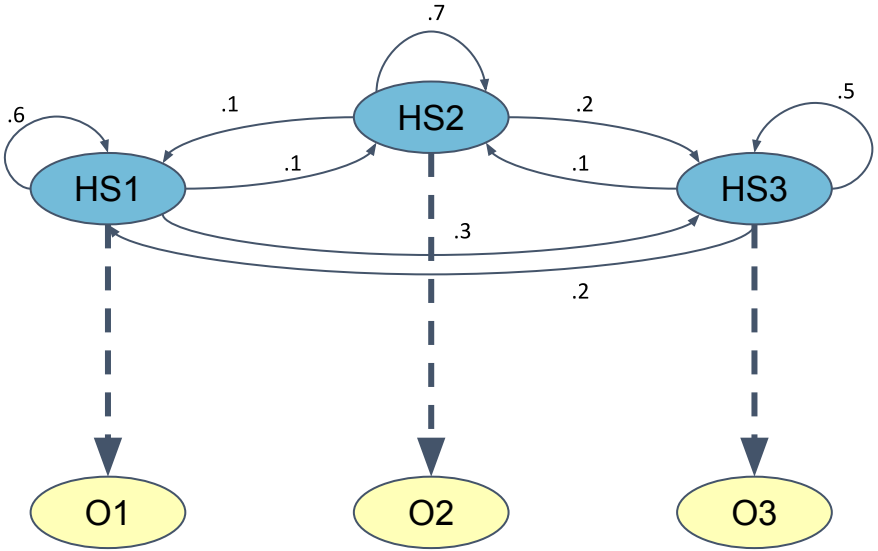
Emission probabilities

Observed events

π Initial probability distribution

Hidden Markov Models (HMMs)

… in ASR?



.5 Transition probabilities

Hidden states

↳ Phonemes or sub-phones

Emission probabilities

↳ Modeled by GMM ➜ GMM-HMM

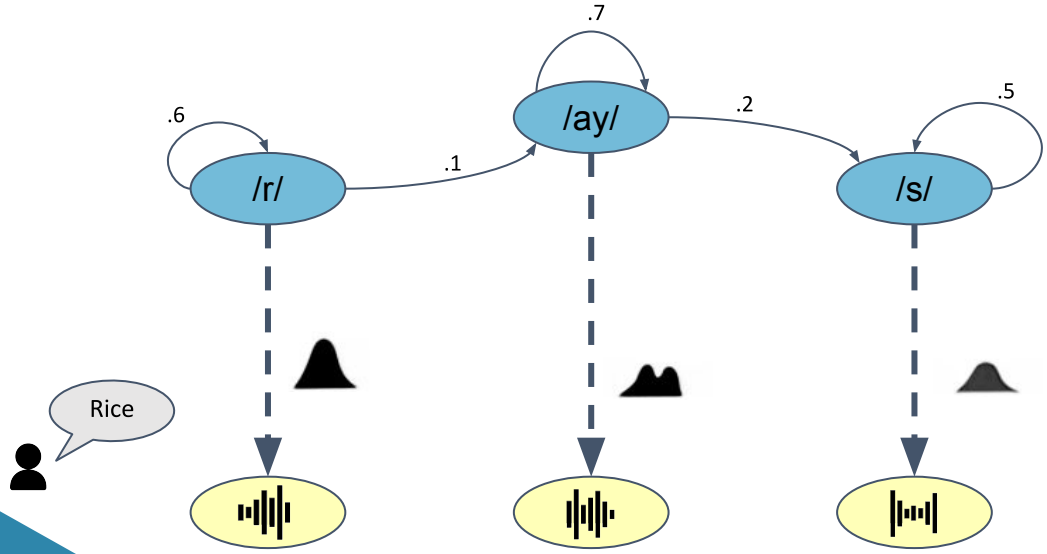
Observed events

↳ Recorded audio/feature vector

π Initial probability distribution

Hidden Markov Models (HMMs)

… in ASR?



.5 Transition probabilities

Hidden states

↳ Phonemes or sub-phones

Emission probabilities

↳ Modeled by GMM ➜ GMM-HMM

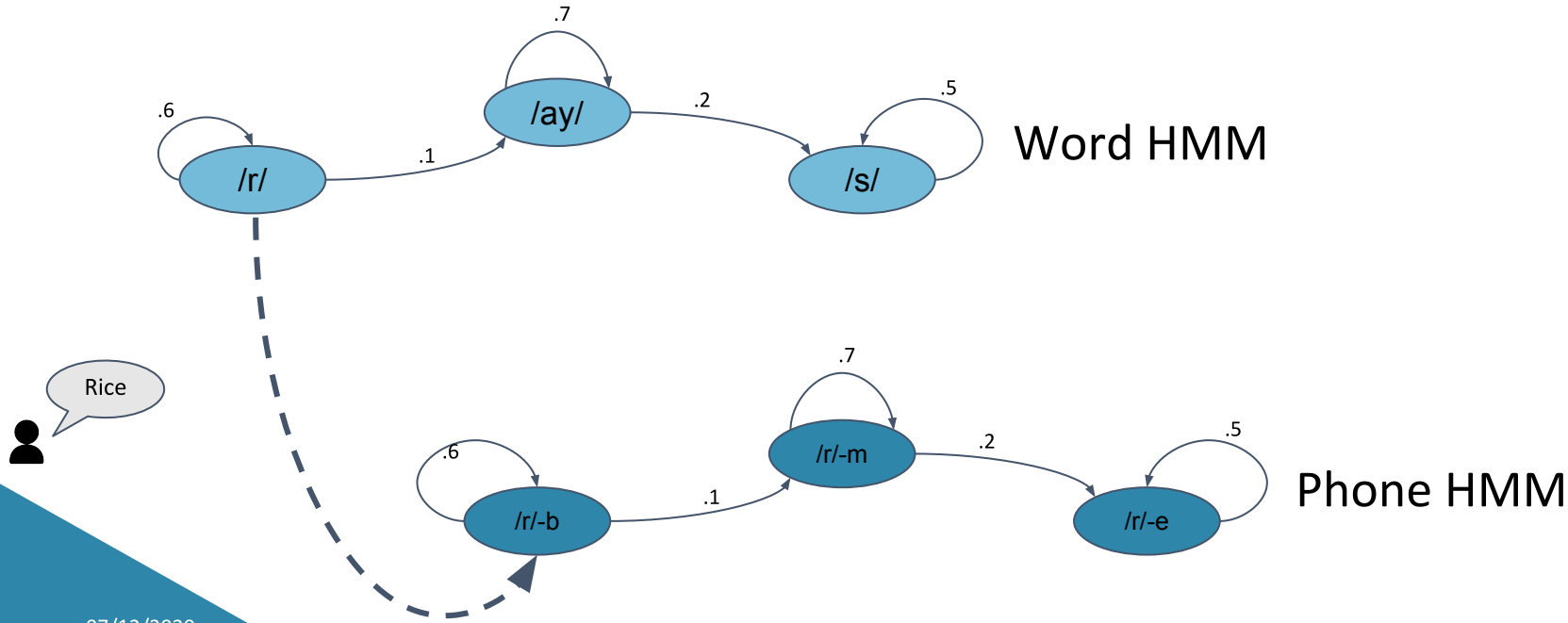
Observed events

↳ Recorded audio/feature vector

π Initial probability distribution

Hidden Markov Models (HMMs)

Phonemes and sub-phones

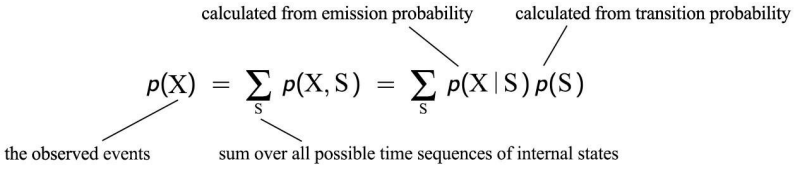


Hidden Markov Models (HMMs)

Algorithms

- Likelihood of observations: *forward algorithm*
- Finding internal states: *Viterbi algorithm*
- Learning: *Baum–Welch/forward-backward algorithm*

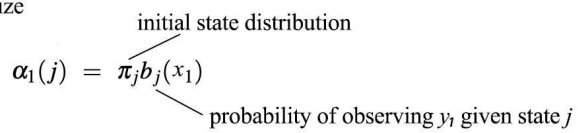
Forward Algorithm



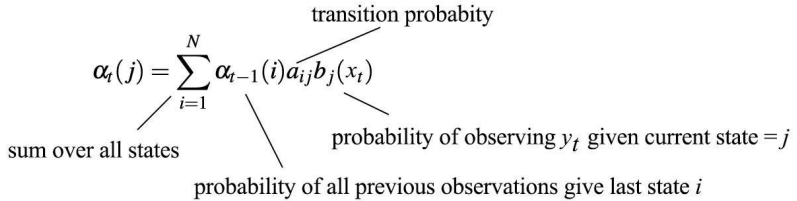
Exponential complexity

Therefore: using result from time t-1 and/or t+1 at time t.

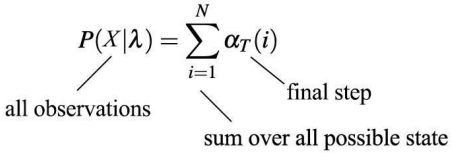
1. initialize



2. For each time step

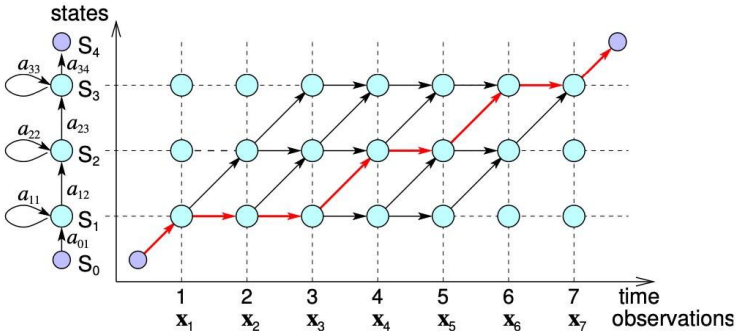


3. Result



Viterbi Algorithm

Similar to forward algorithm but instead of summing all possible state sequences, the Viterbi algorithm tries to find the most likely path of state sequences.



$$p(\mathbf{X}, \text{path}_\ell \,|\, \boldsymbol{\lambda}) = p(\mathbf{X} \,|\, \text{path}_\ell, \boldsymbol{\lambda}) P(\text{path}_\ell \,|\, \boldsymbol{\lambda})$$

likelihood: $\sum_{\{\text{path}_\ell\}} p(\mathbf{X}, \text{path}_\ell \,|\, \boldsymbol{\lambda})$

decode: $\max_{\text{path}_\ell} p(\mathbf{X}, \text{path}_\ell \,|\, \boldsymbol{\lambda})$

Baum–Welch/forward–backward Algorithm

Special form of the EM-algorithm, solving the problem in iteration steps.

Estimation step

Forward

1. $\alpha_i(1) = \pi_i b_i(y_1)$,
2. $\alpha_i(t+1) = b_i(y_{t+1}) \sum_{j=1}^{N} \alpha_j(t) a_{ji}$.

Backward

1. $\beta_i(T) = 1$,
2. $\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) a_{ij} b_j(y_{t+1})$.

Update

$$\gamma_i(t) = P(X_t = i|Y,\theta) = \frac{P(X_t = i, Y|\theta)}{P(Y|\theta)} = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$$

equals α (forward) × β (backward) for state i at time t

$$\xi_{ij}(t) = P(X_t = i, X_{t+1} = j|Y,\theta) = \frac{P(X_t = i, X_{t+1} = j, Y|\theta)}{P(Y|\theta)} = \frac{\alpha_i(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})}$$

equals α for state i at time t × tranistion prob. between i and j × β for state j at time $t+1$ × observe y_{t+1} for state j

Maximization step

$$\pi_i^* = \gamma_i(1)$$

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

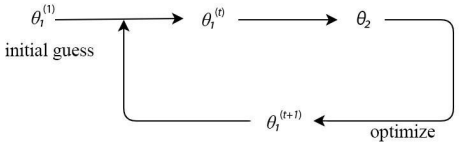
$P(X_t = i, X_{t+1} = j|Y,\theta)$ / $P(X_t = i|Y,\theta)$

$$b_i^*(v_k) = \frac{\sum_{t=1}^{T} 1_{y_t = v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

(sum γ over all time steps where the observation y_t is the same as v_t at time t)

where

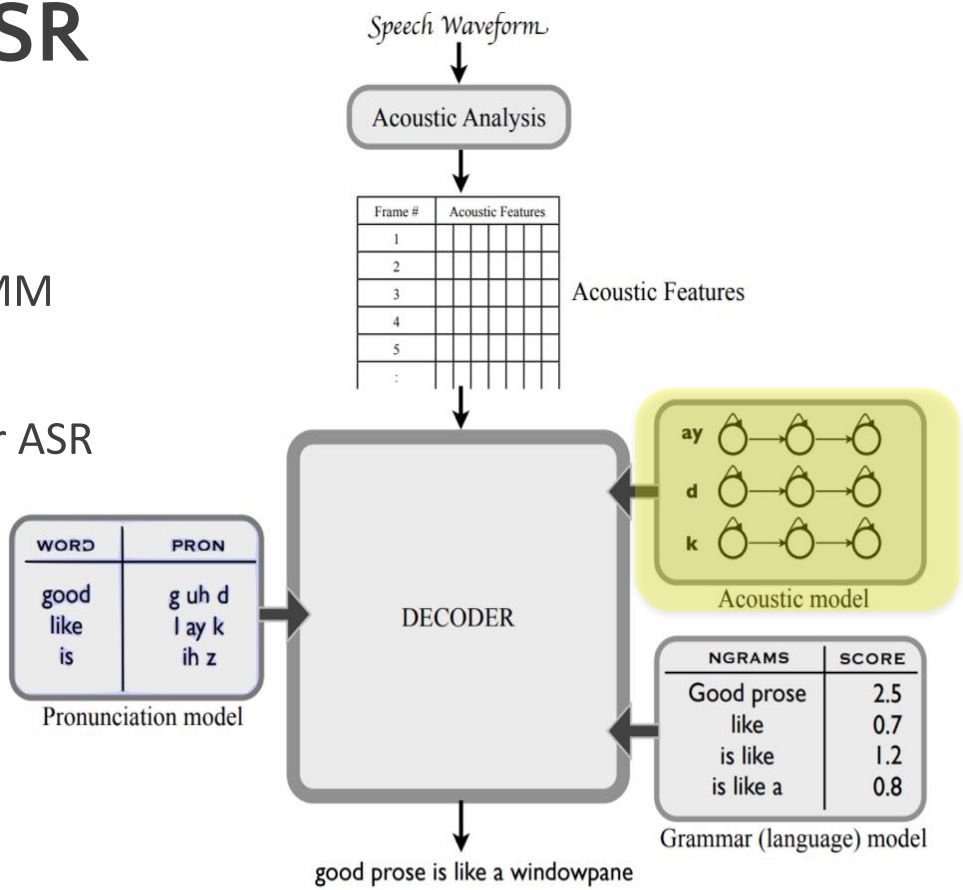
$$1_{y_t = v_k} = \begin{cases} 1 & \text{if } y_t = v_k, \\ 0 & \text{otherwise} \end{cases}$$



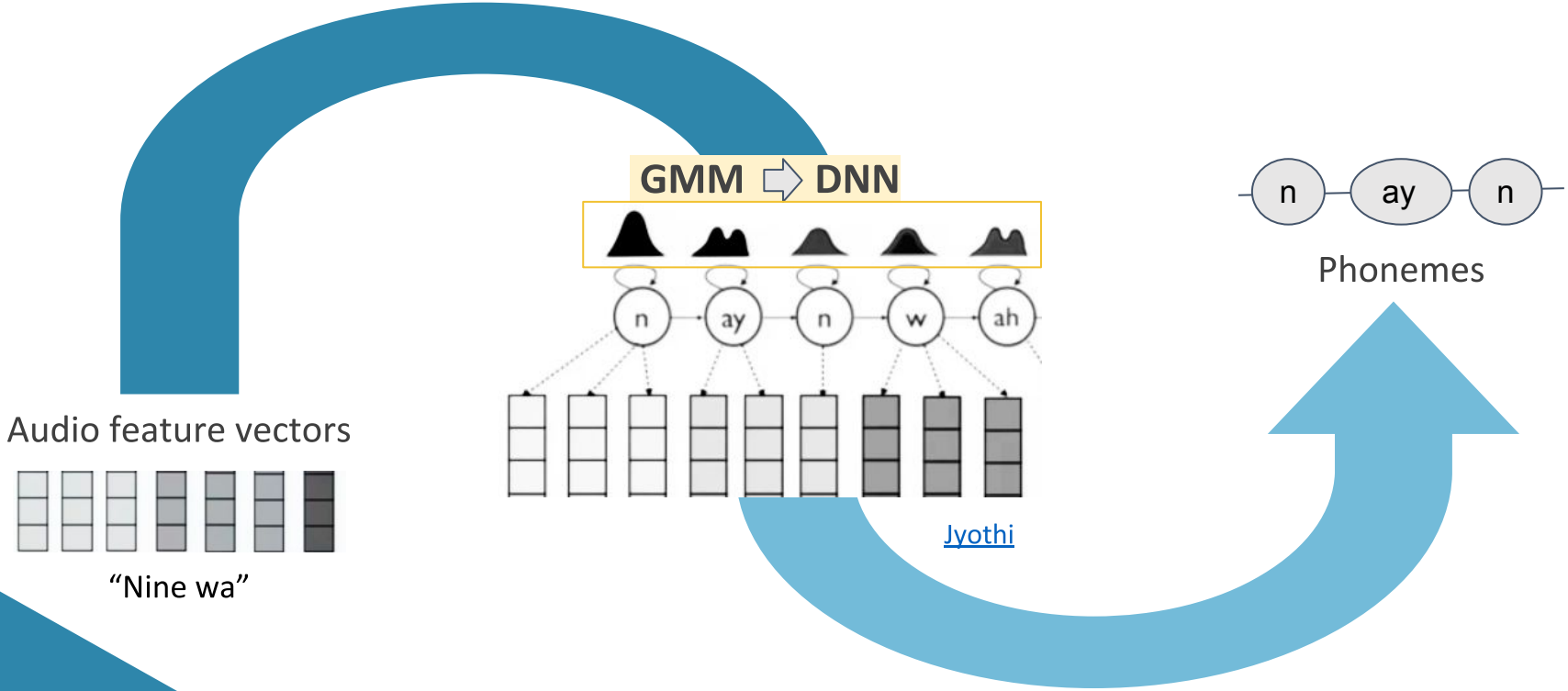
Neural network for ASR

- 1980s | GMM-HMM
- 1990s | (Artificial) Neural Network + HMM
- 2006~ | The era of deep learning starts,
  Deep neural network (DNN) for ASR

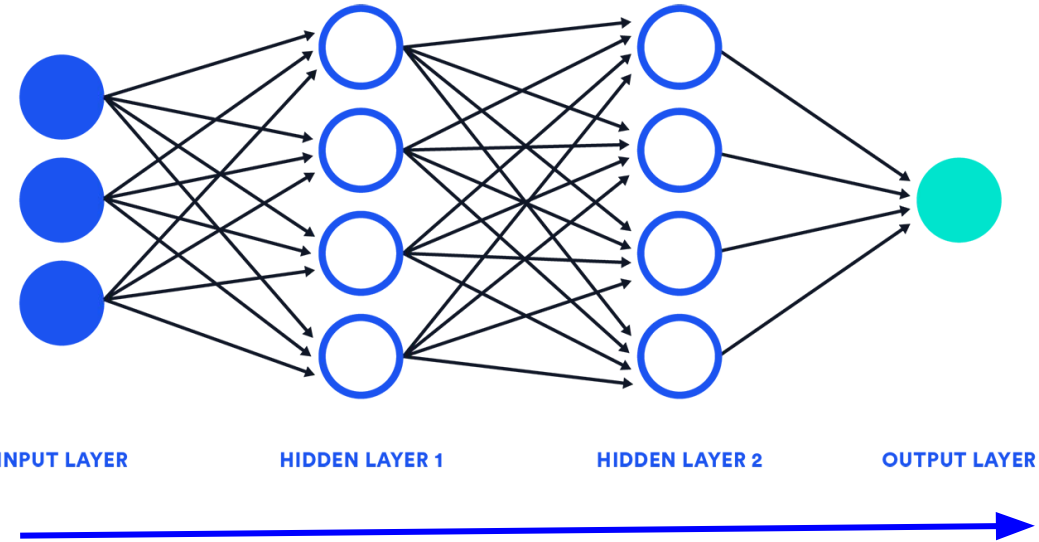
DNN - HMM for acoustic models



DNN-HMM for the acoustic model



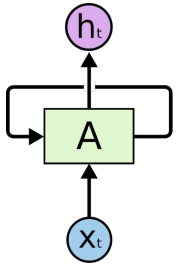
DNN : feed forward structure



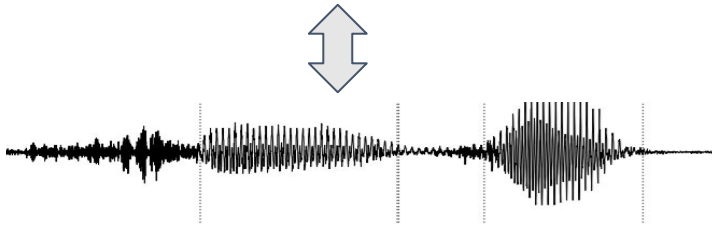
The **feed forward structure of DNN** has limited abilities to deal with:

- temporal dependencies in sequence data like speech

Recurrent Neural Network (RNN)



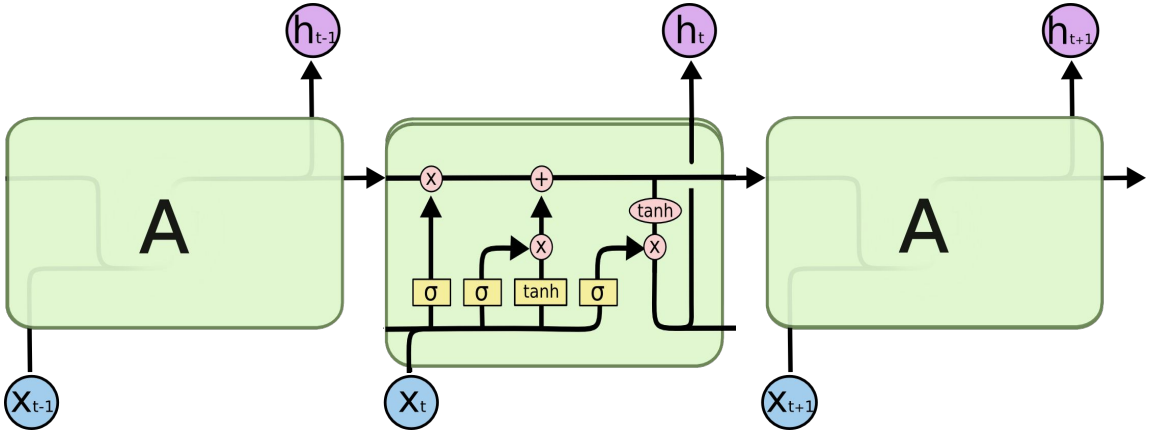
Christopher Olah



The chain-like structure of RNN enables:

- dealing with sequence data (text, speech etc)
- handling **temporal dependency**

Long-short-term-memory (LSTM)

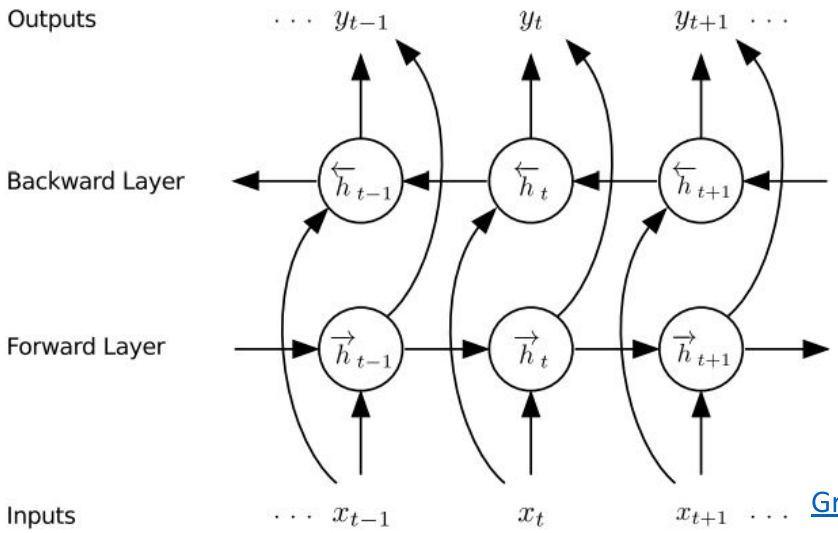


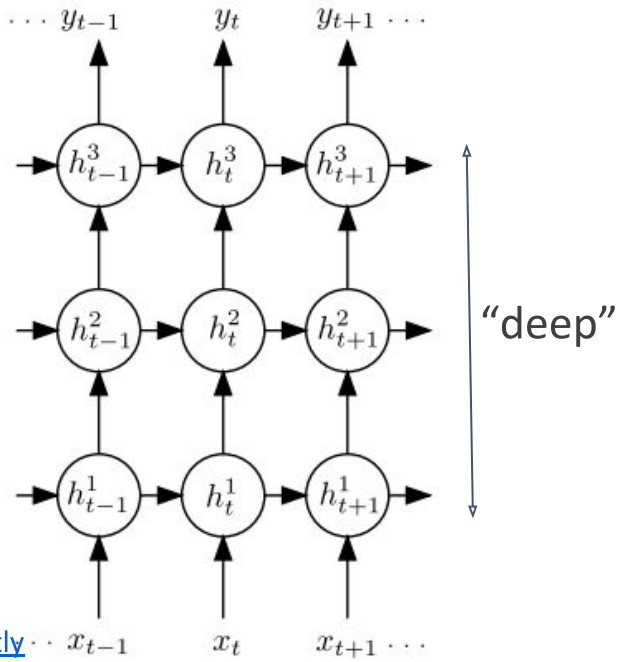
Christopher Olah

LSTM with a specific designed memory cell:

- can deal with **long-term** dependencies in data
- more effective than conventional RNN in acoustic modeling

Variations of LSTM for ASR

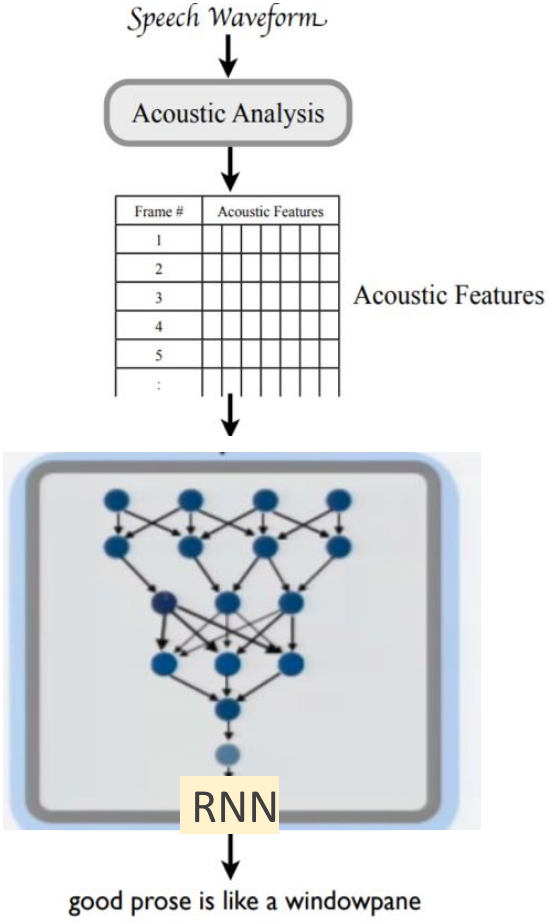




Graves & Jaitly

- Bidirectional RNN (LSTM) : exploits both previous and future context. (left)
- Deep RNN (LSTM) : stacks multiple RNN hidden layers. (right)

From partial to end-to-end



Jyothi

References

Christopher Olah. (2015). *Understanding LSTM*. Retrieved November 28, 2020, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 30–42.

Gales, M., & Young, S. (2007). The application of hidden markov models in speech recognition. *Foundations and Trends® in Signal Processing*, *1*(3), 195–304.

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645–6649.

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, II–1764–II–1772.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.

Hui, J. (2019, December 20). *Speech recognition — phonetics* [Medium]. Retrieved November 28, 2020, from https://jonathan-hui.medium.com/speech-recognition-phonetics-d761ea1710c0

Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing* (2nd ed.) [Series Title: Always learning]. Pearson Education.

References

Jyothi, P. (2019). *Automatic speech recognition.* https://www.cse.iitb.ac.in/~pjyothi/cs753/slides/lecture1.pdf

Karpagavalli, S., & Chandra, E. (2016). A review on automatic speech recognition architecture and approaches. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *9*(4), 393–404.

Microsoft Research. (2017, September 11). *Automatic speech recognition - an overview.*

O'Shaughnessy, D. (2008). Invited paper: Automatic speech recognition: History, methods and challenges. *Pattern Recognition*, *41*(10), 2965–2979.

OxfordUniversityPress (Ed.). (2020). Artificial intelligence. In *Oxford advanced learner's dictionary.*

Pearce, D., Wood, L., & Novello, F. (1991). Improved vocabulary-independent sub-word hmm modelling. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 181–184.

Rabiner, L. R., & Juang, B.-H. (1993). *Fundamentals of speech recognition* [Series Title: Prentice Hall signal processing series]. Prentice Hall PTR.

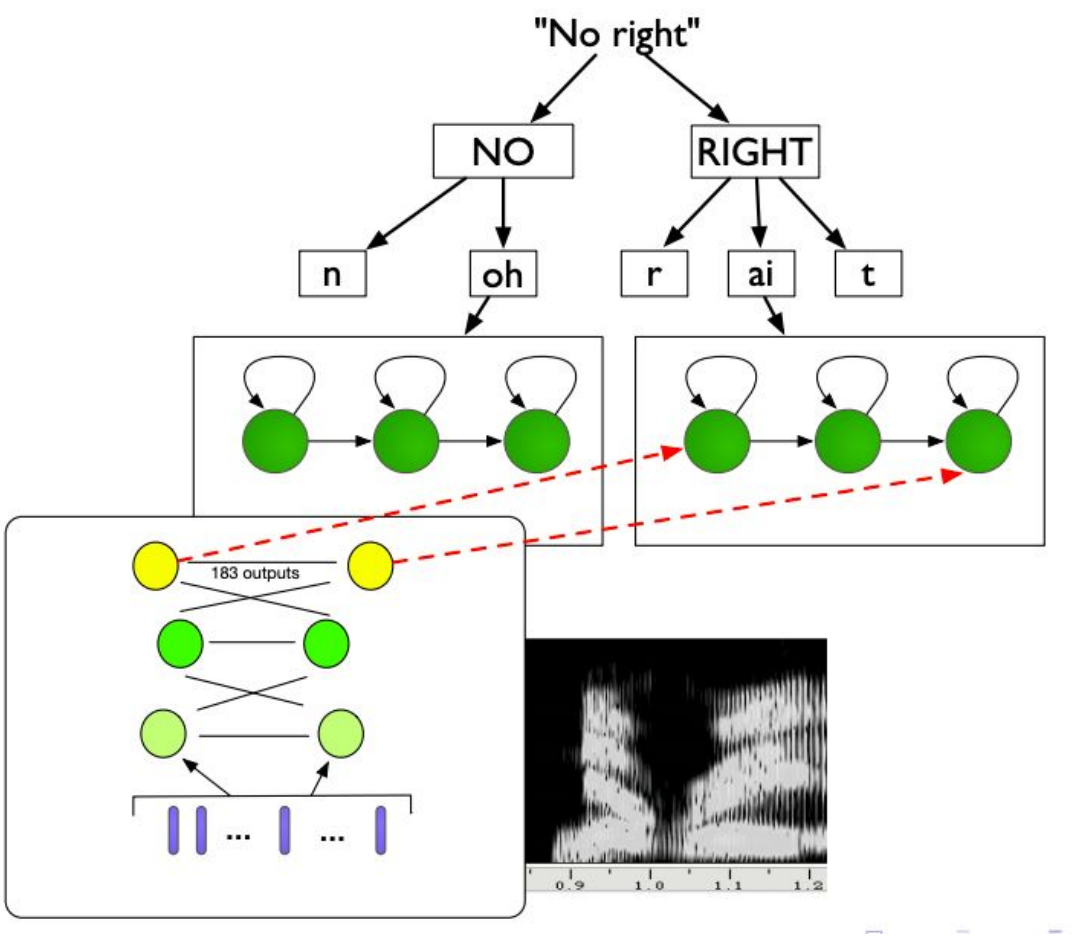
Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *INTERSPEECH.*

Sen, S., Dutta, A., & Dey, N. (2019). *Audio processing and speech recognition: Concepts, techniques and research overviews.* Springer Singapore.

Shewalkar, A., Nyavanandi, D., & Ludwig, S. A. (2019). Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU. *Journal of Artificial Intelligence and Soft Computing Research*, *9*(4), 235–245.

Yu, D., & Deng, L. (2015). *Automatic speech recognition.* Springer London.

Thank You


"No right"
NO
RIGHT
n
oh
r
ai
t
183 outputs
0.9
1.0
1.1
1.2

Summary of DNN-HMM acoustic models

Comparison against HMM-GMM on different tasks

[TABLE 3] A COMPARISON OF THE PERCENTAGE WERs USING DNN-HMMs AND GMM-HMMs ON FIVE DIFFERENT LARGE VOCABULARY TASKS.

TASK	HOURS OF TRAINING DATA	DNN-HMM	GMM-HMM WITH SAME DATA	GMM-HMM WITH MORE DATA
SWITCHBOARD (TEST SET 1)	309	18.5	27.4	18.6 (2,000 H)
SWITCHBOARD (TEST SET 2)	309	16.1	23.6	17.1 (2,000 H)
ENGLISH BROADCAST NEWS	50	17.5	18.8	
BING VOICE SEARCH (SENTENCE ERROR RATES)	24	30.4	36.2	
GOOGLE VOICE INPUT	5,870	12.3		16.0 (>> 5,870 H)
YOUTUBE	1,400	47.6	52.3	

Hybrid DNN-HMM systems consistently outperform GMM-HMM systems (sometimes even when the latter is trained with lots more data)

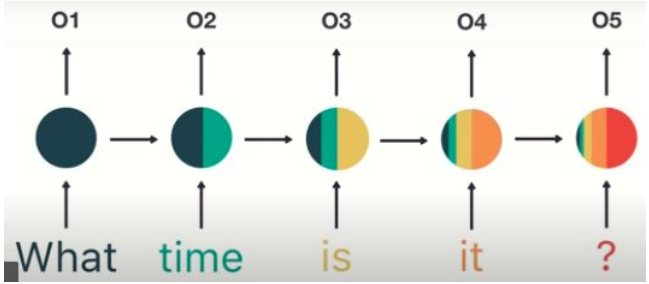
Hinton et al.

RNN: mathematical equations

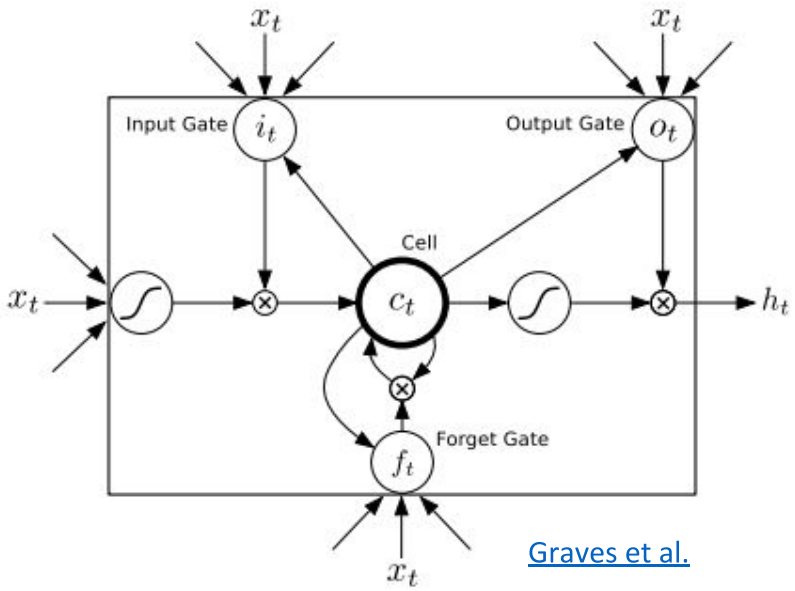
RNN calculates its output by iteratively calculating the following two equations

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \quad (1)$$

$$y_t = W_{hy}h_t + b_y, \quad (2)$$



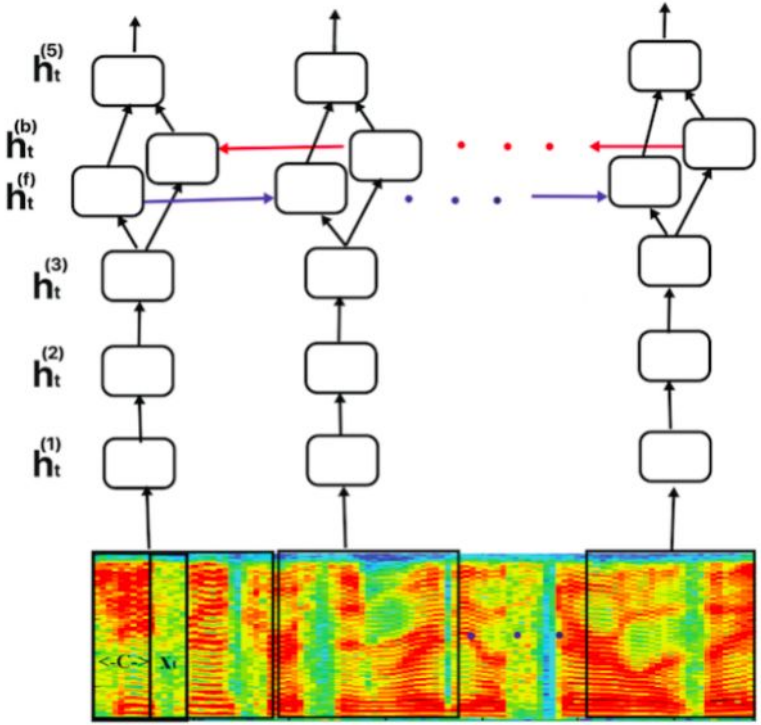
LSTM memory cell



Graves et al.

Amazing! This box of cereal gave me a **perfectly balanced breakfast**, as all things should be. I only ate half of it but **will definitely be buying again!**

Deep bidirectional RNN (LSTM) for e2e model



Shewalkar et al.

End-to-end results by character sequence

target: *TO ILLUSTRATE THE POINT A PROMINENT MIDDLE EAST ANALYST IN WASHINGTON RECOUNTS A CALL FROM ONE CAMPAIGN*
output: *TWO ALSTRAIT THE POINT A PROMINENT MIDILLE EAST ANALYST IM WASHINGTON RECOUNCACALL FROM ONE CAMPAIGN*

target: *T. W. A. ALSO PLANS TO HANG ITS BOUTIQUE SHINGLE IN AIRPORTS AT LAMBERT SAINT*
output: *T. W. A. ALSO PLANS TOHING ITS BOOTIK SINGLE IN AIRPORTS AT LAMBERT SAINT*

target: *ALL THE EQUITY RAISING IN MILAN GAVE THAT STOCK MARKET INDIGESTION LAST YEAR*
output: *ALL THE EQUITY RAISING IN MULONG GAVE THAT STACRK MARKET IN TO JUSTIAN LAST YEAR*

target: *THERE'S UNREST BUT WE'RE NOT GOING TO LOSE THEM TO DUKAKIS*
output: *THERE'S UNREST BUT WERE NOT GOING TO LOSE THEM TO DEKAKIS*