Deep Learning for Audio

YUCHEN FAN, MATT POTOK, CHRISTOPHER SHROBA

Motivation

Text-to-Speech

- Accessibility features for people with little to no vision, or people in situations where they cannot look at a screen or other textual source
- Natural language interfaces for a more fluid and natural way to interact with computers
- Voice Assistants (Siri, etc.), GPS, Screen Readers, Automated telephony systems

Automatic Speech Recognition

- Again, natural language interfaces
- Alternative input medium for accessibility purposes
- Voice Assistants (Siri, etc.), Automated telephony systems, Hands-free phone control in the car

Music Generation

- Mostly for fun
- Possible applications in music production software

Outline

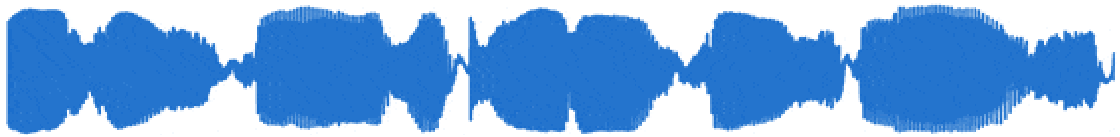
Automatic Speech Recognition (ASR)

- Deep models with HMMs
- Connectionist Temporal Classification (CTC)
- Attention based models

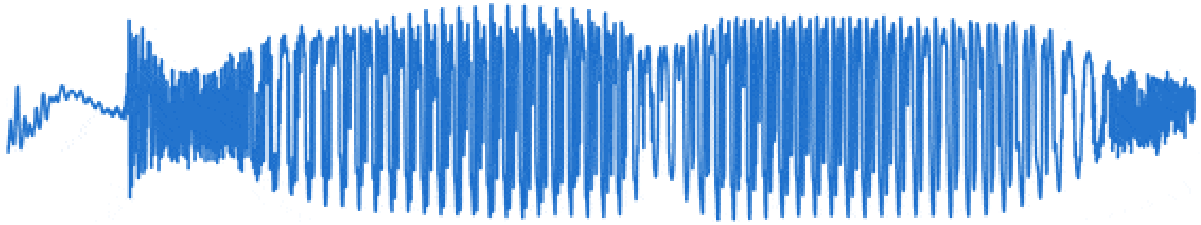
Text to Speech (TTS)

- WaveNet
- DeepVoice
- Tacotron

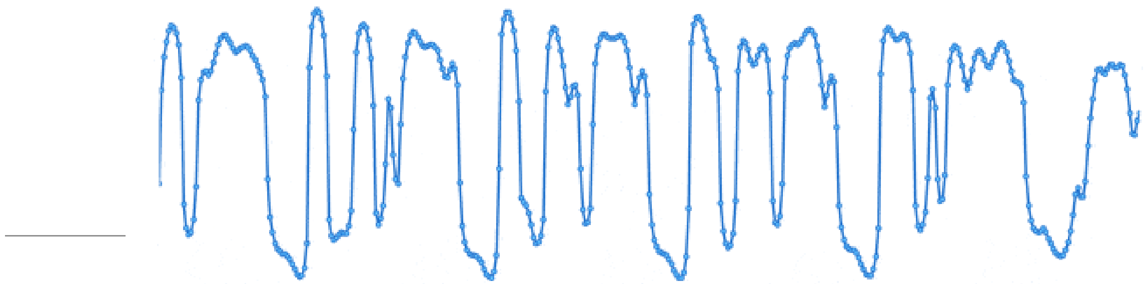
Bonus: Music Generation



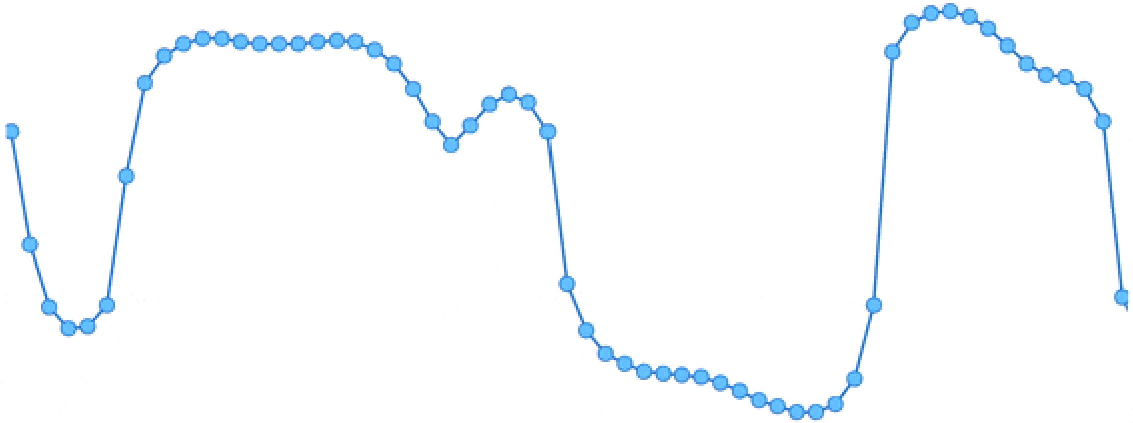
1 Second



100 milliseconds



10 milliseconds



1 millisecond

Automatic Speech Recognition (ASR)

Outline

History of Automatic Speech Recognition

Hidden Markov Model (HMM) based Automatic Speech Recognition

- Gaussian mixture models with HMMs
- Deep models with HMMs

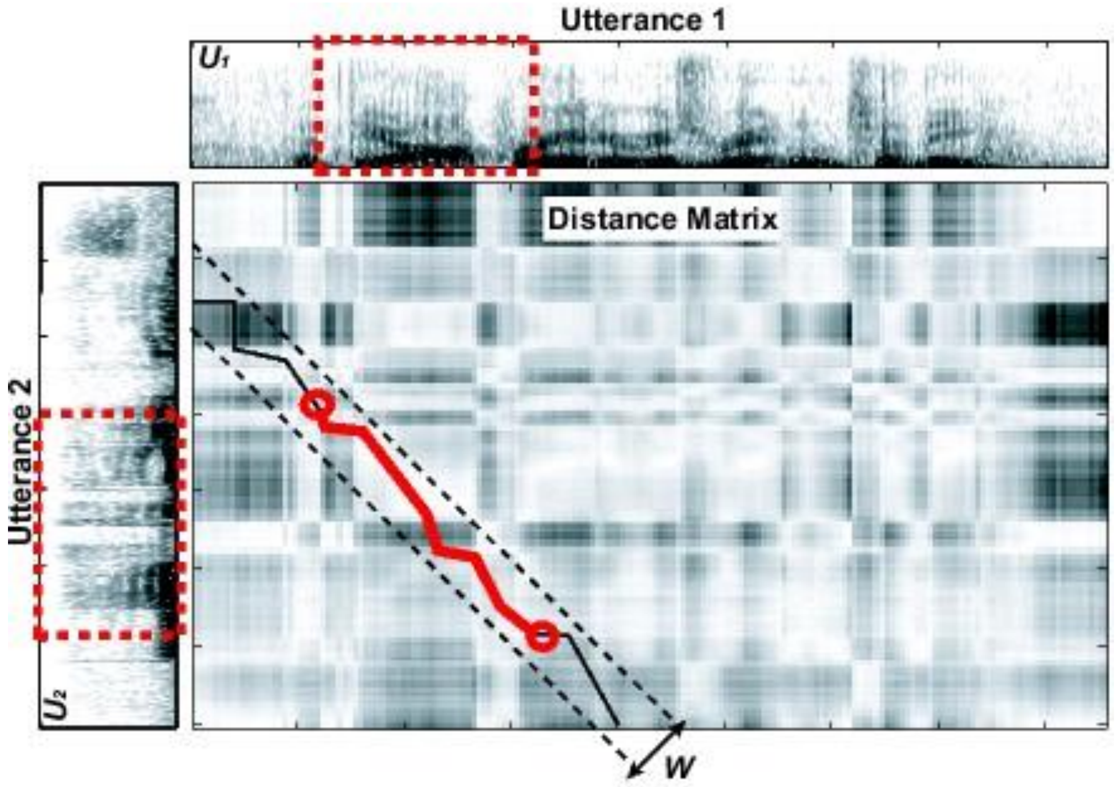
End-to-End Deep Models based Automatic Speech Recognition

- Connectionist Temporal Classification (CTC)
- Attention based models

History of Automatic Speech Recognition

Early 1970s: Dynamic Time Warping (DTW) to handle time variability

- Distance measure for spectral variability

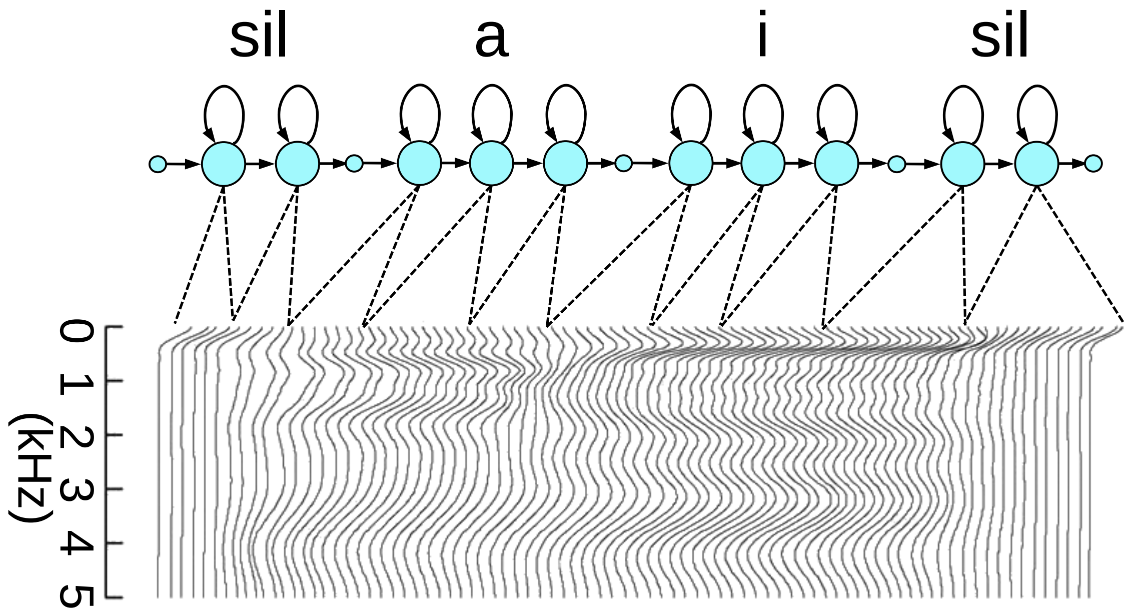


http://publications.csail.mit.edu/abstracts/abstracts06/malex/seg_dtw.jpg

History of Automatic Speech Recognition

Mid-Late 1970s: Hidden Markov Models (HMMs) – statistical models of spectral variations, for discrete speech.

Mid 1980s: HMMs become the dominant technique for all ASR



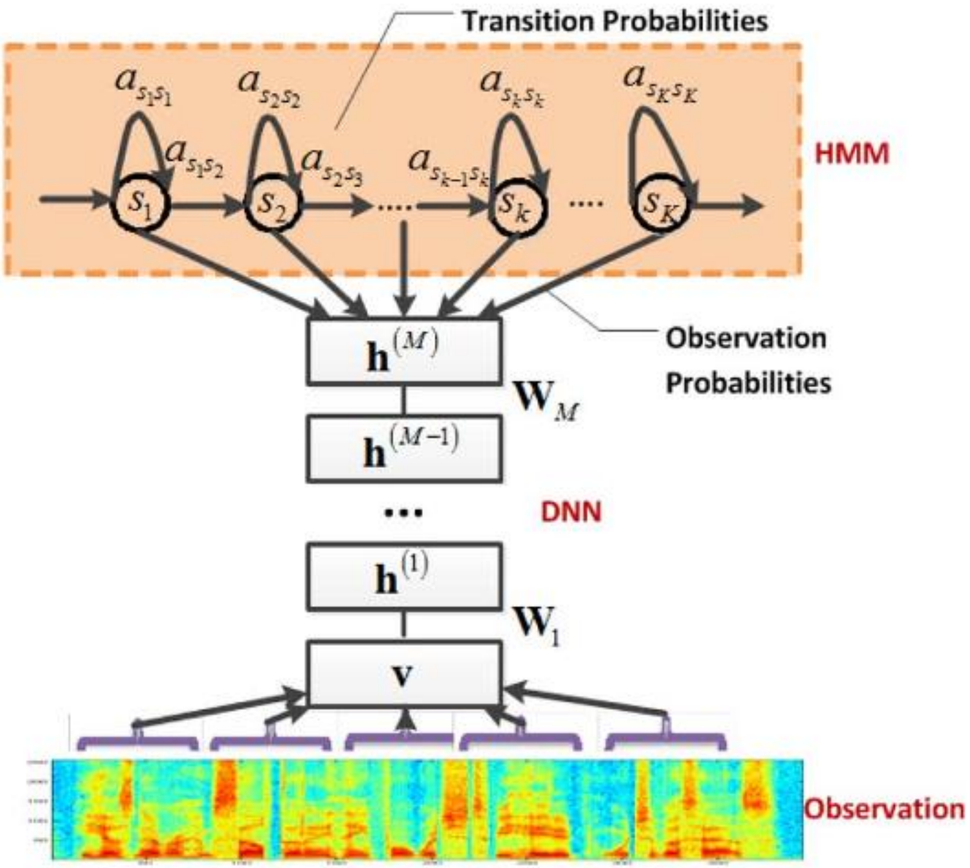
http://hts.sp.nitech.ac.jp/archives/2.3/HTS_Slides.zip

History of Automatic Speech Recognition

1990s: Large vocabulary continuous dictation

2000s: Discriminative training (minimize word/phone error rate)

2010s: Deep learning significantly reduce error rate



George E. Dahl, et al. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Trans. Audio, Speech & Language Processing, 2012.

Aim of Automatic Speech Recognition

Find the most likely sentence (word sequence) $\boldsymbol{W}$, which transcribes the speech audio $\boldsymbol{A}$:

$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}} P(\boldsymbol{W}|\boldsymbol{A}) = \underset{\boldsymbol{W}}{\operatorname{argmax}} P(\boldsymbol{A}|\boldsymbol{W})P(\boldsymbol{W})$$

- Acoustic model $P(\boldsymbol{A}|\boldsymbol{W})$
- Language model $P(\boldsymbol{W})$

Training: find parameters for acoustic and language model separately

- Speech Corpus: speech waveform and human-annotated transcriptions
- Language model: with extra data (prefer daily expressions corpus for spontaneous speech)

Language model

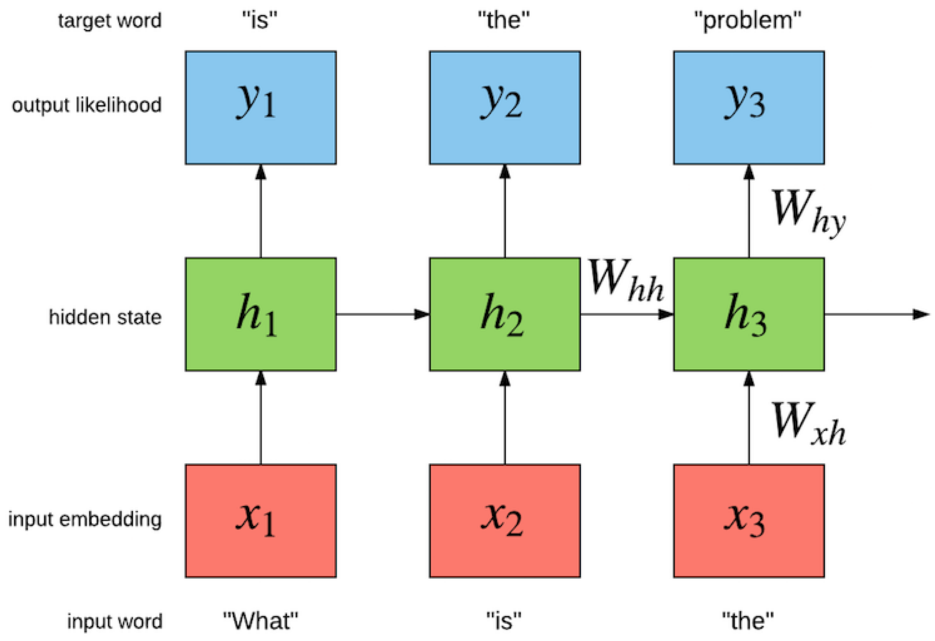
Language model is a probabilistic model used to

- Guide the search algorithm (predict next word given history)
- Disambiguate between phrases which are acoustically similar
  - Great wine vs Grey twine

It assigns probability to a sequence of tokens to be finally recognized

N-gram model $P(w_N | w_1, w_2, \cdots, w_{N-1})$

Recurrent neural network

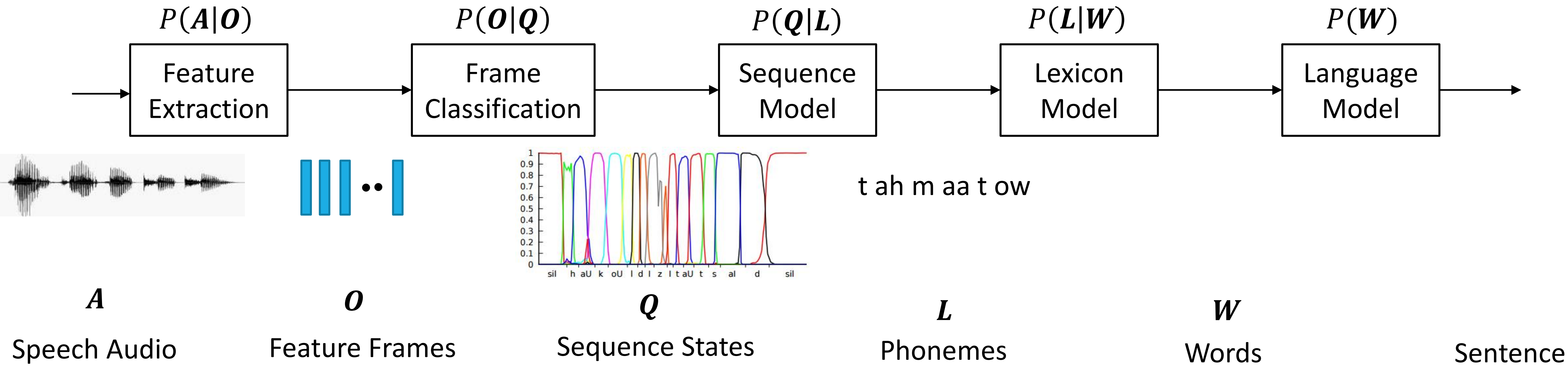


Recurrent neural network based Language model

http://torch.ch/blog/2016/07/25/nce.html

Architecture of Speech Recognition

$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}} P(\boldsymbol{W}|\boldsymbol{O}) = \underset{\boldsymbol{W}}{\operatorname{argmax}} P(\boldsymbol{A}|\boldsymbol{O})P(\boldsymbol{O}|\boldsymbol{Q})P(\boldsymbol{Q}|\boldsymbol{L})P(\boldsymbol{L}|\boldsymbol{W})P(\boldsymbol{W})$$



$P(\boldsymbol{A}|\boldsymbol{O})$ Feature Extraction

$P(\boldsymbol{O}|\boldsymbol{Q})$ Frame Classification

$P(\boldsymbol{Q}|\boldsymbol{L})$ Sequence Model

$P(\boldsymbol{L}|\boldsymbol{W})$ Lexicon Model

$P(\boldsymbol{W})$ Language Model

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

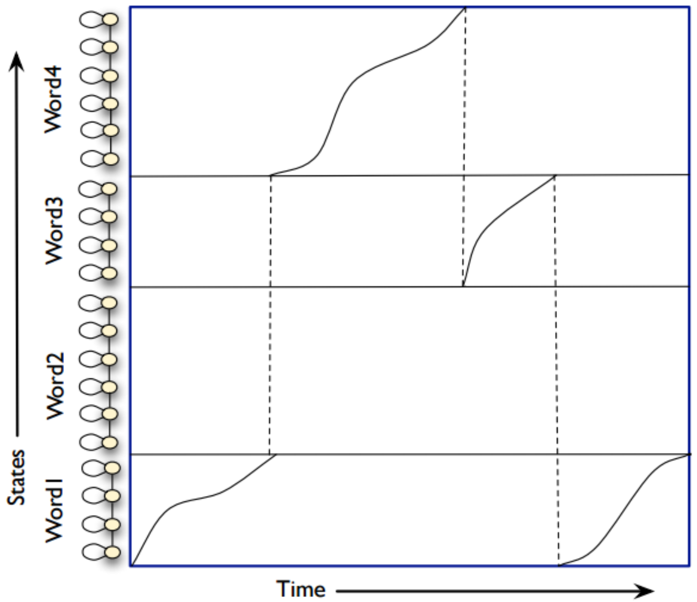
$\boldsymbol{L}$ Phonemes

$\boldsymbol{W}$ Words

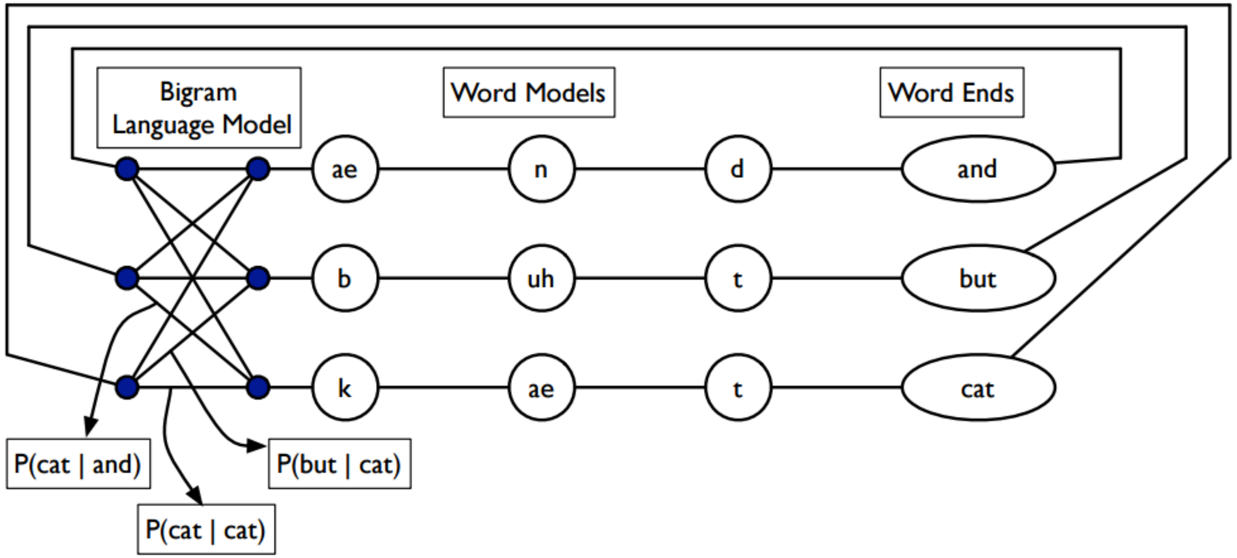
Sentence

Decoding in Speech Recognition

Find most likely word sequence of audio input



Without language models

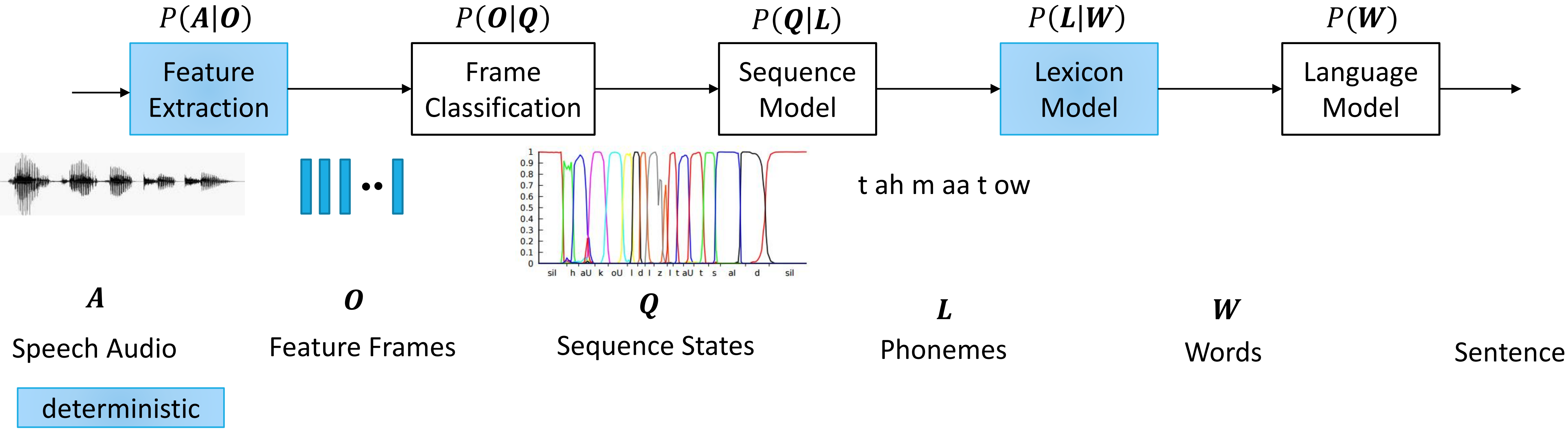


With language models

https://www.inf.ed.ac.uk/teaching/courses/asr/2011-12/asr-search-nup4.pdf

Architecture of Speech Recognition

$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{W}|\boldsymbol{O}) = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{A}|\boldsymbol{O})P(\boldsymbol{O}|\boldsymbol{Q})P(\boldsymbol{Q}|\boldsymbol{L})P(\boldsymbol{L}|\boldsymbol{W})P(\boldsymbol{W})$$



$P(\boldsymbol{A}|\boldsymbol{O})$ Feature Extraction

$P(\boldsymbol{O}|\boldsymbol{Q})$ Frame Classification

$P(\boldsymbol{Q}|\boldsymbol{L})$ Sequence Model

$P(\boldsymbol{L}|\boldsymbol{W})$ Lexicon Model

$P(\boldsymbol{W})$ Language Model

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

$\boldsymbol{L}$ Phonemes

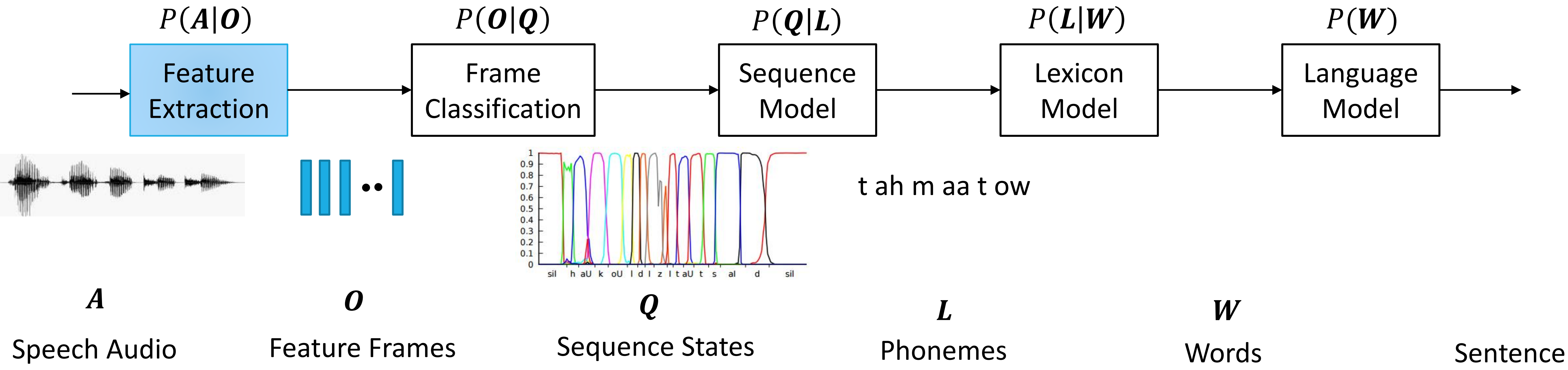
$\boldsymbol{W}$ Words

Sentence

deterministic

Architecture of Speech Recognition

$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}} P(\boldsymbol{W}|\boldsymbol{O}) = \underset{\boldsymbol{W}}{\operatorname{argmax}} P(\boldsymbol{A}|\boldsymbol{O})P(\boldsymbol{O}|\boldsymbol{Q})P(\boldsymbol{Q}|\boldsymbol{L})P(\boldsymbol{L}|\boldsymbol{W})P(\boldsymbol{W})$$



$P(\boldsymbol{A}|\boldsymbol{O})$ Feature Extraction

$P(\boldsymbol{O}|\boldsymbol{Q})$ Frame Classification

$P(\boldsymbol{Q}|\boldsymbol{L})$ Sequence Model

$P(\boldsymbol{L}|\boldsymbol{W})$ Lexicon Model

$P(\boldsymbol{W})$ Language Model

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

$\boldsymbol{L}$ Phonemes

$\boldsymbol{W}$ Words

Sentence

Feature Extraction

Raw waveforms are transformed into a sequence of feature vectors using signal processing approaches

Time domain to frequency domain

Feature extraction is a deterministic process

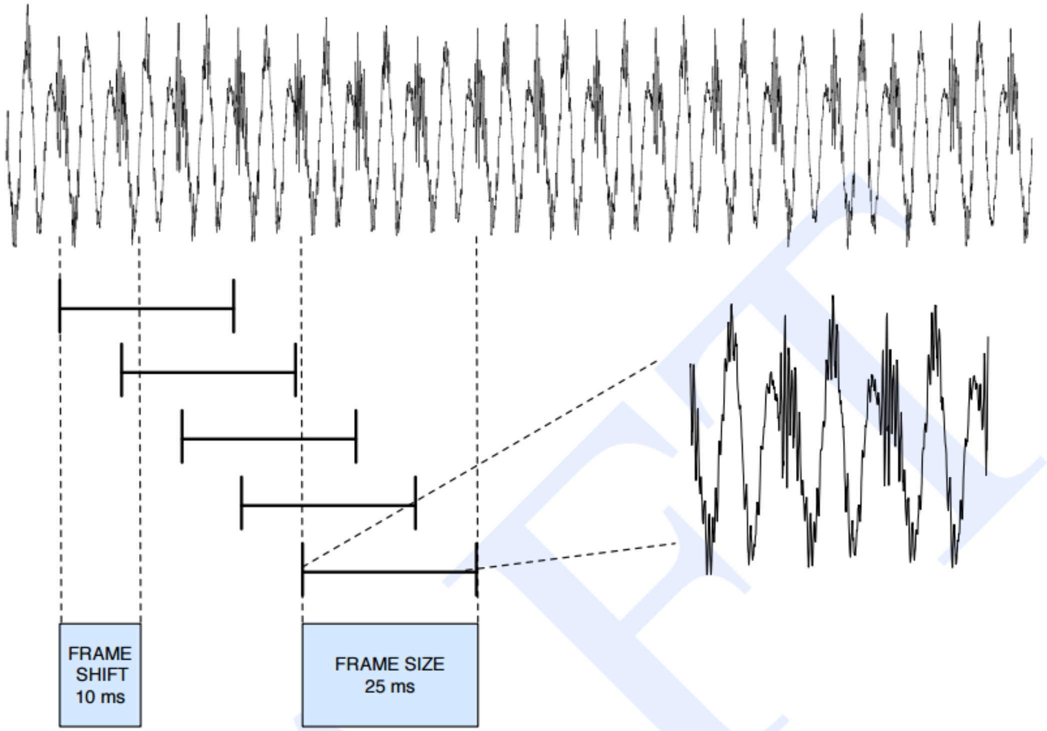
$$P(\boldsymbol{A}|\boldsymbol{O}) = \delta(A, \hat{A}(O))$$

Reduce information rate but keep useful information

- Remove noise and other irrelevant information

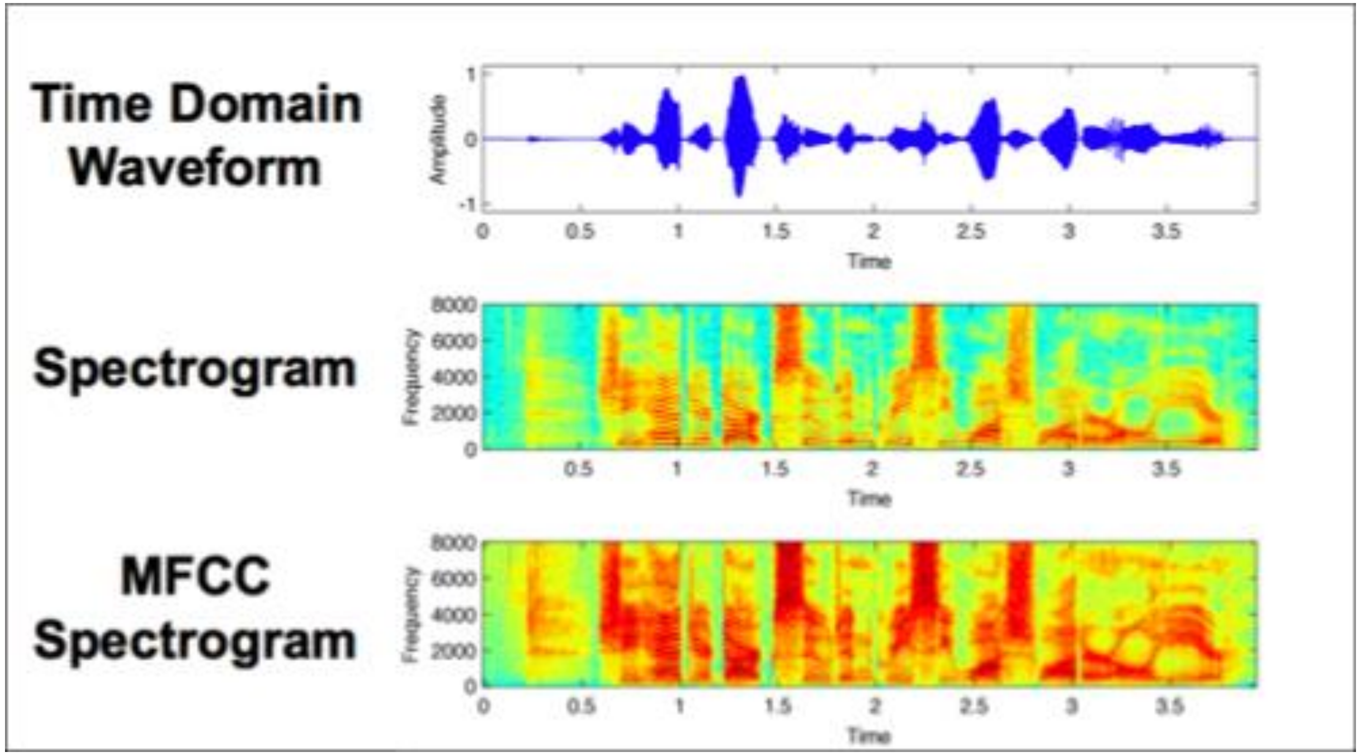
Extracted in 25ms windows and shifted with 10ms

Still useful for deep models



http://www2.cmpe.boun.edu.tr/courses/cmpe362/spring2014/files/projects/MFCC%20Feature%20Extraction.pdf

Different Level Features



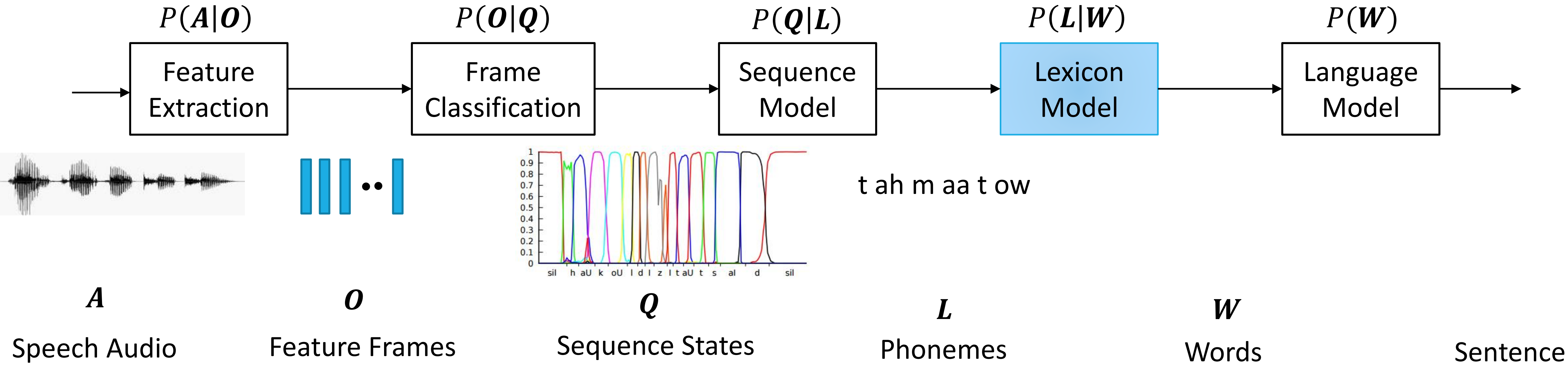
Better for deep models

Better for shallow models

http://music.ece.drexel.edu/research/voiceID/researchday2007

Architecture of Speech Recognition

$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{W}|\boldsymbol{O}) = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{A}|\boldsymbol{O})P(\boldsymbol{O}|\boldsymbol{Q})P(\boldsymbol{Q}|\boldsymbol{L})P(\boldsymbol{L}|\boldsymbol{W})P(\boldsymbol{W})$$



$P(\boldsymbol{A}|\boldsymbol{O})$ Feature Extraction

$P(\boldsymbol{O}|\boldsymbol{Q})$ Frame Classification

$P(\boldsymbol{Q}|\boldsymbol{L})$ Sequence Model

$P(\boldsymbol{L}|\boldsymbol{W})$ Lexicon Model

$P(\boldsymbol{W})$ Language Model

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

$\boldsymbol{L}$ Phonemes

$\boldsymbol{W}$ Words

Sentence

Lexical model

Lexical modelling forms the bridge between the acoustic and language models

Prior knowledge of language

Mapping between words and the acoustic units (phoneme is most common)

Deterministic

Word	Pronunciation
TOMATO	t ah m aa t ow
	t ah m ey t ow
COVERAGE	k ah v er ah jh
	k ah v r ah jh

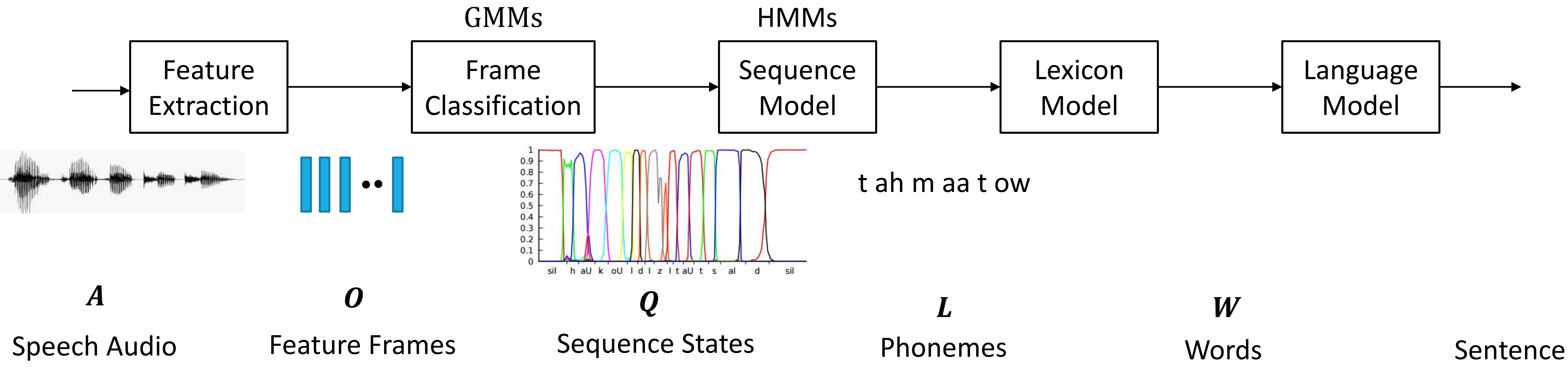
Probabilistic

Word	Pronunciation	Probability
TOMATO	t ah m aa t ow	0.45
	t ah m ey t ow	0.55
COVERAGE	k ah v er ah jh	0.65
	k ah v r ah jh	0.35

GMM-HMM in Speech Recognition

GMMs: Gaussian Mixture Models

HMMs: Hidden Markov Models



GMMs

HMMs

Feature Extraction

Frame Classification

Sequence Model

Lexicon Model

Language Model

t ah m aa t ow

$\boldsymbol{A}$

Speech Audio

$\boldsymbol{O}$

Feature Frames

$\boldsymbol{Q}$

Sequence States

$\boldsymbol{L}$

Phonemes

$\boldsymbol{W}$

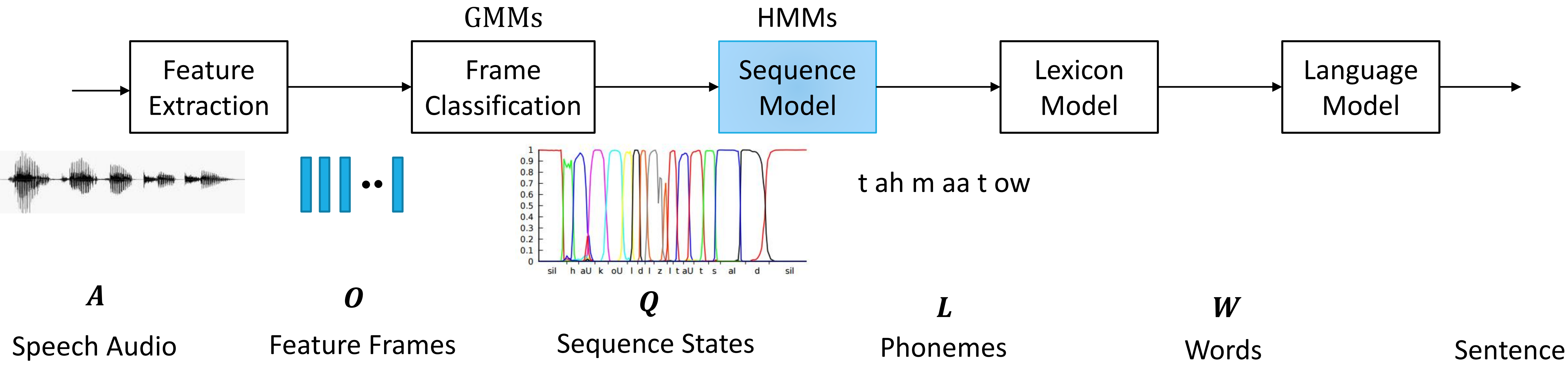
Words

Sentence

GMM-HMM in Speech Recognition

GMMs: Gaussian Mixture Models

HMMs: Hidden Markov Models



GMMs

HMMs

Feature Extraction

Frame Classification

Sequence Model

Lexicon Model

Language Model

t ah m aa t ow

A

Speech Audio

O

Feature Frames

Q

Sequence States

L

Phonemes

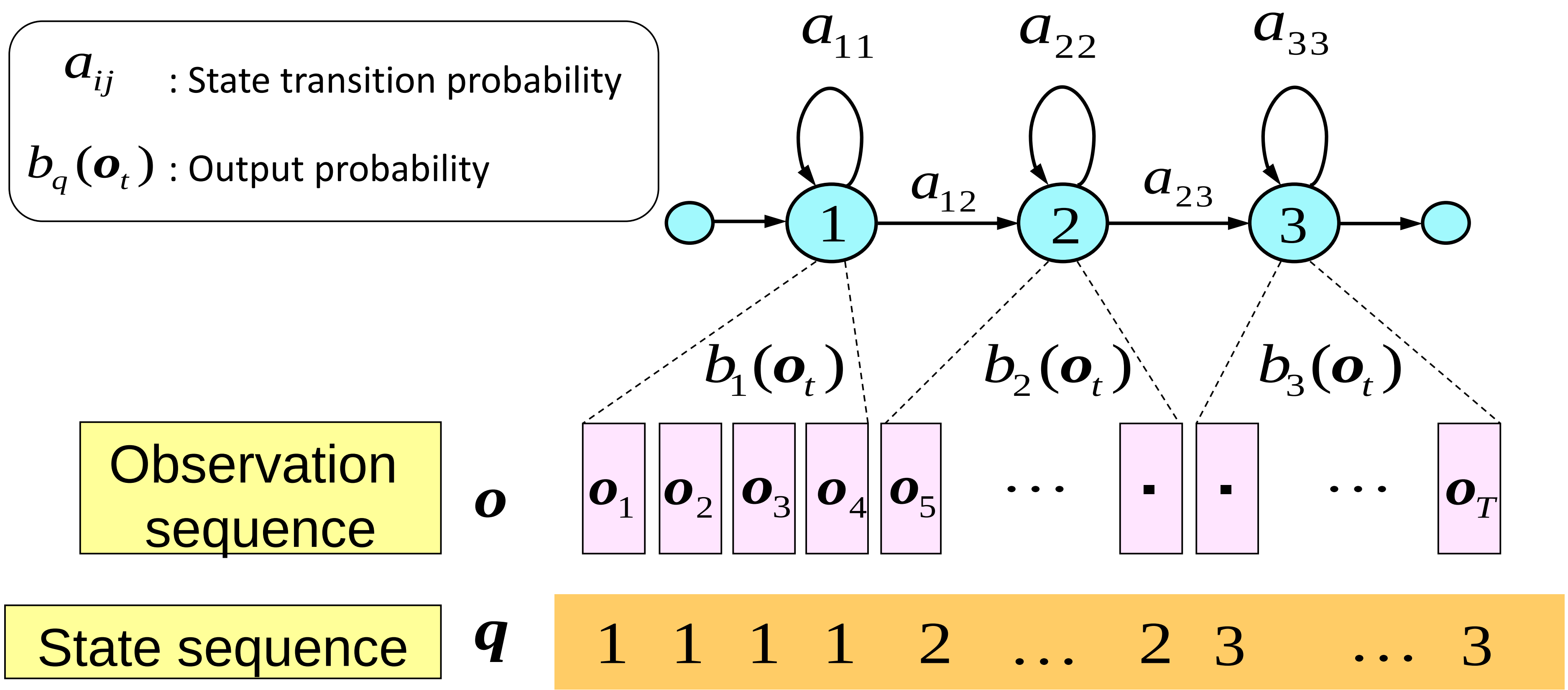
W

Words

Sentence

Hidden Markov Models

The Markov chain whose state sequence is unknown



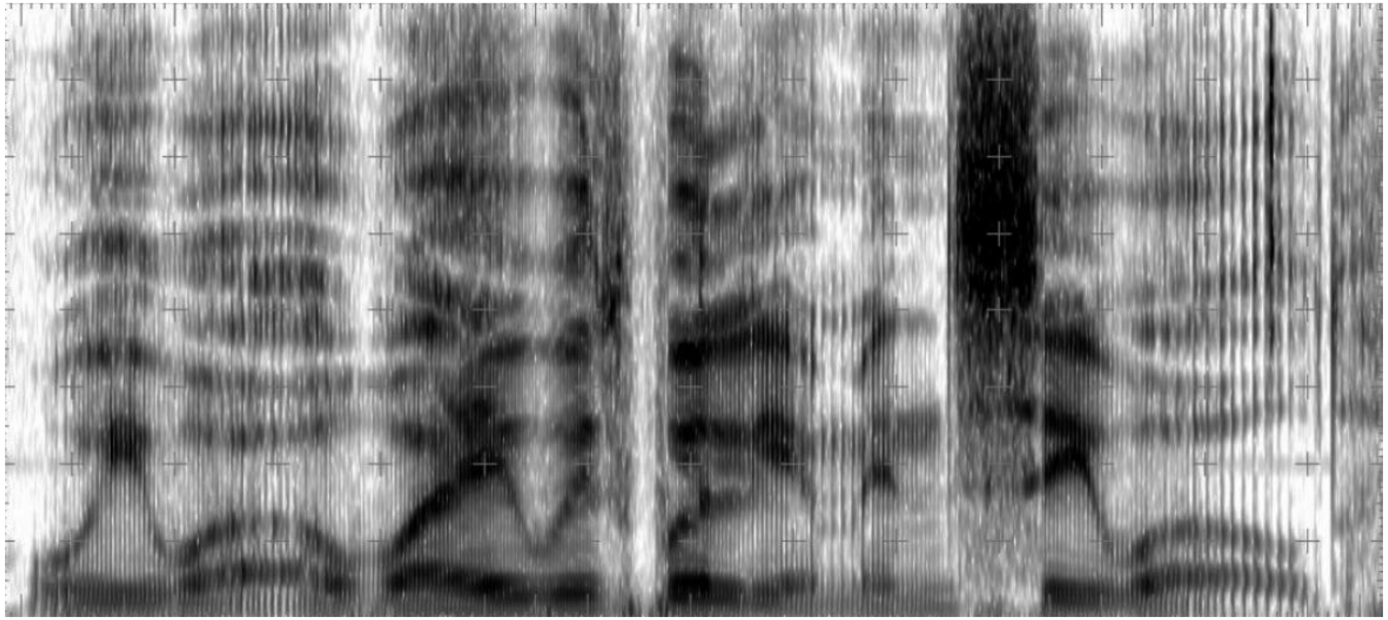
a_{ij} : State transition probability

$b_q(\boldsymbol{o}_t)$: Output probability

http://hts.sp.nitech.ac.jp/archives/2.3/HTS_Slides.zip

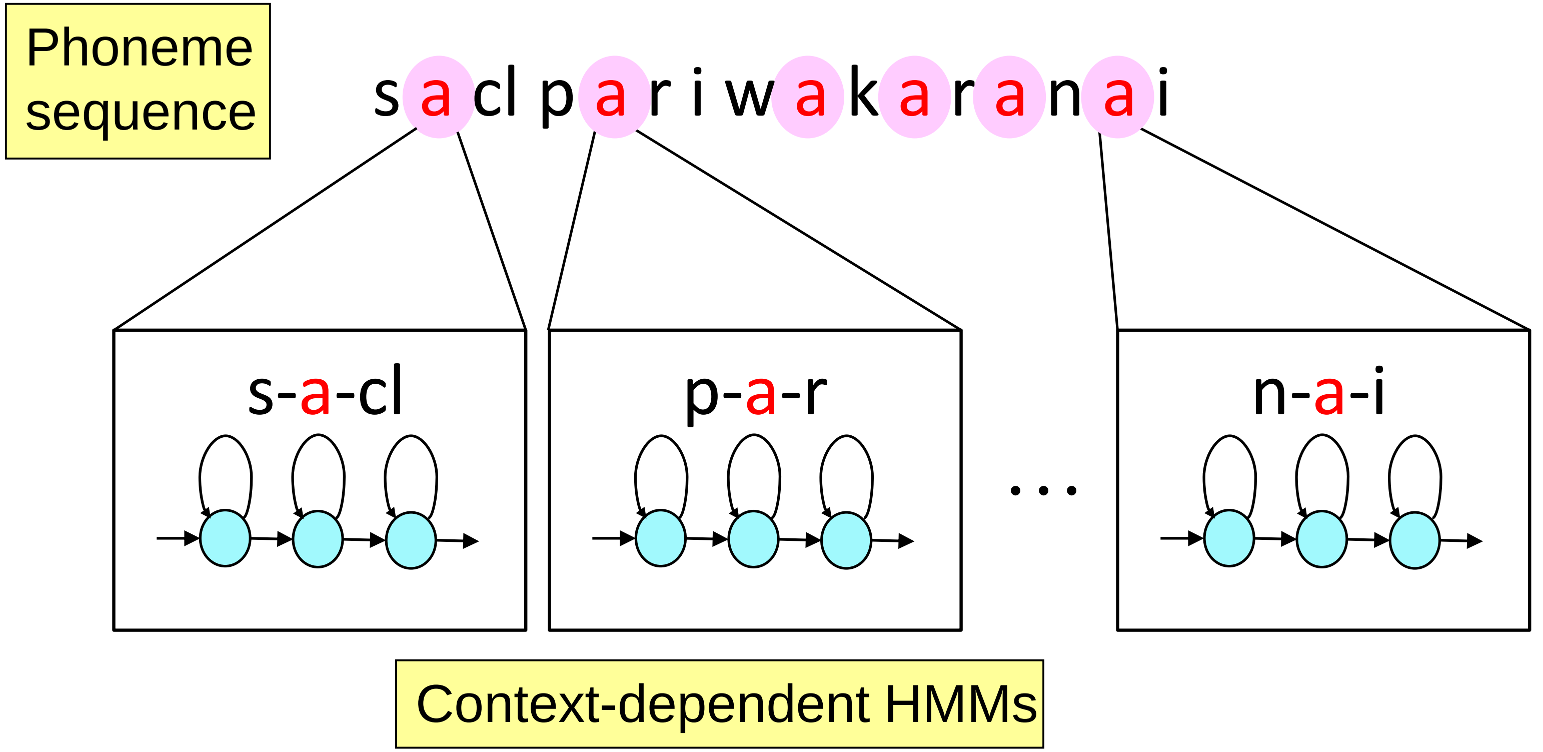
Context Dependent Model

We were away with William in Sea World



Realization of w varies but similar patterns occur in the similar context

Context Dependent Model



In English

#Monophone : 46

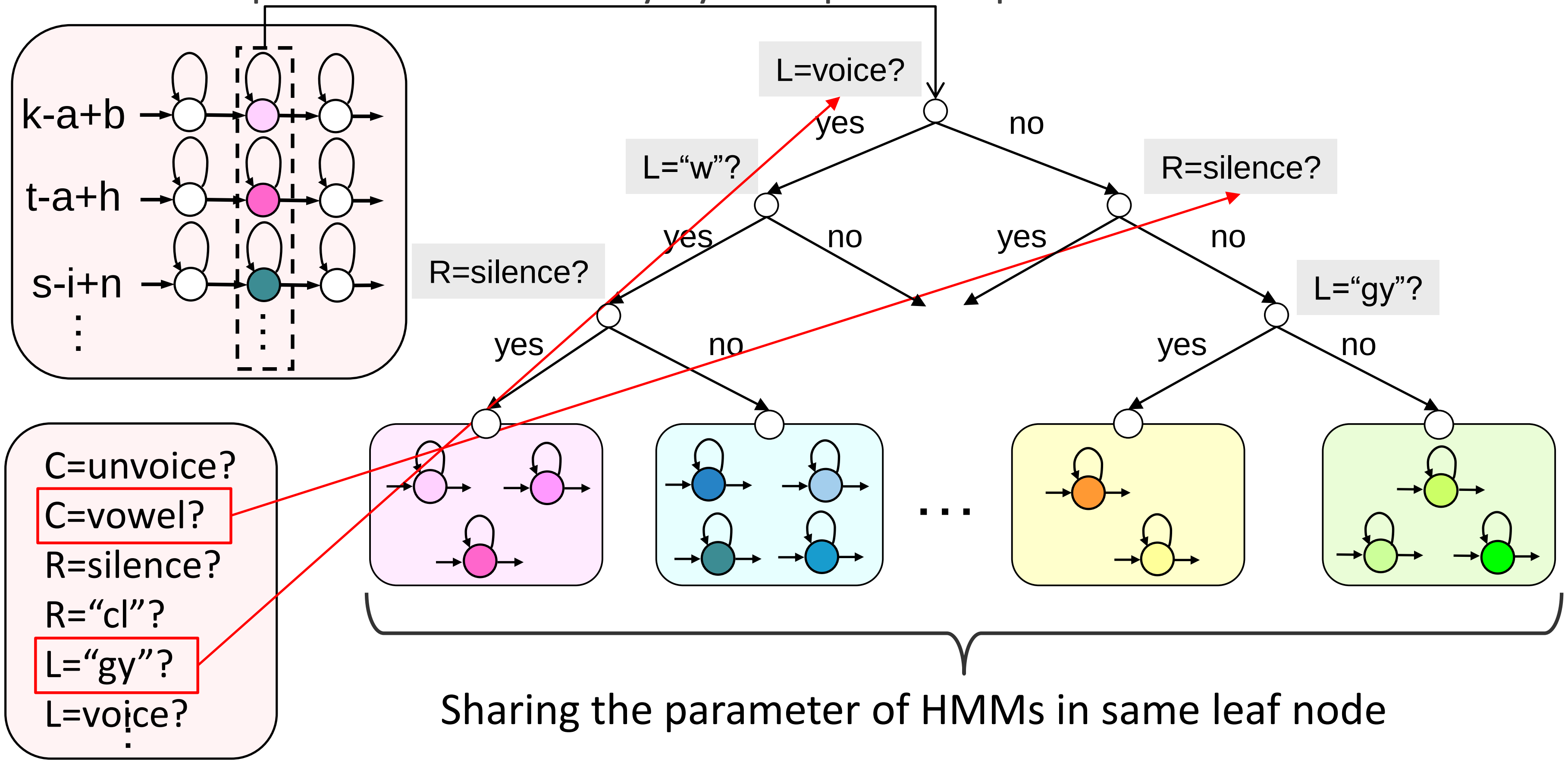
#Biphone: 2116

#Triphone: 97336

http://hts.sp.nitech.ac.jp/archives/2.3/HTS_Slides.zip

Decision Tree-based State Clustering

Each state separated automatically by the optimum question

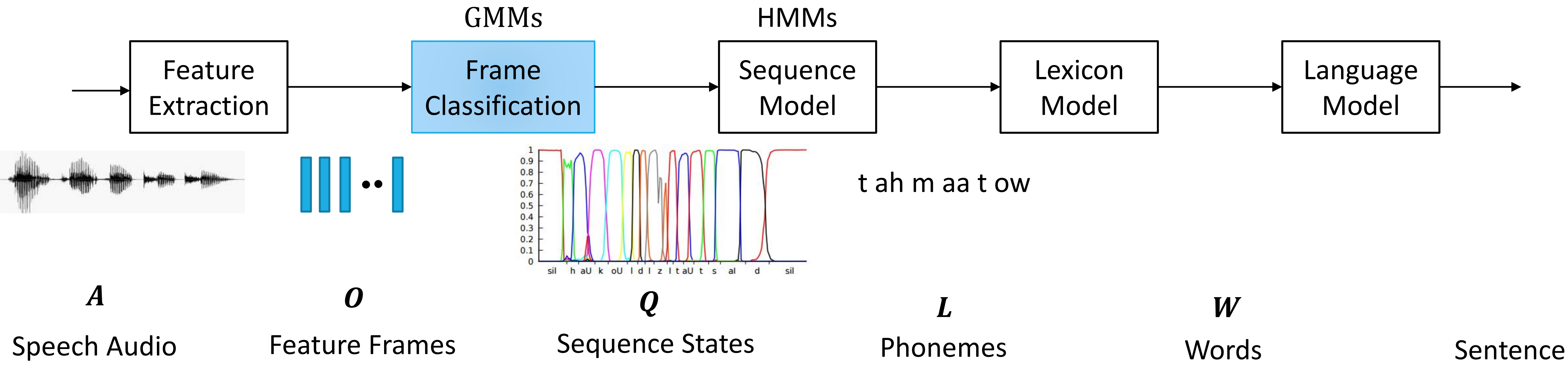


Sharing the parameter of HMMs in same leaf node

GMM-HMM in Speech Recognition

GMMs: Gaussian Mixture Models

HMMs: Hidden Markov Models



GMMs

HMMs

Feature Extraction → Frame Classification → Sequence Model → Lexicon Model → Language Model

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

$\boldsymbol{L}$ Phonemes

$\boldsymbol{W}$ Words

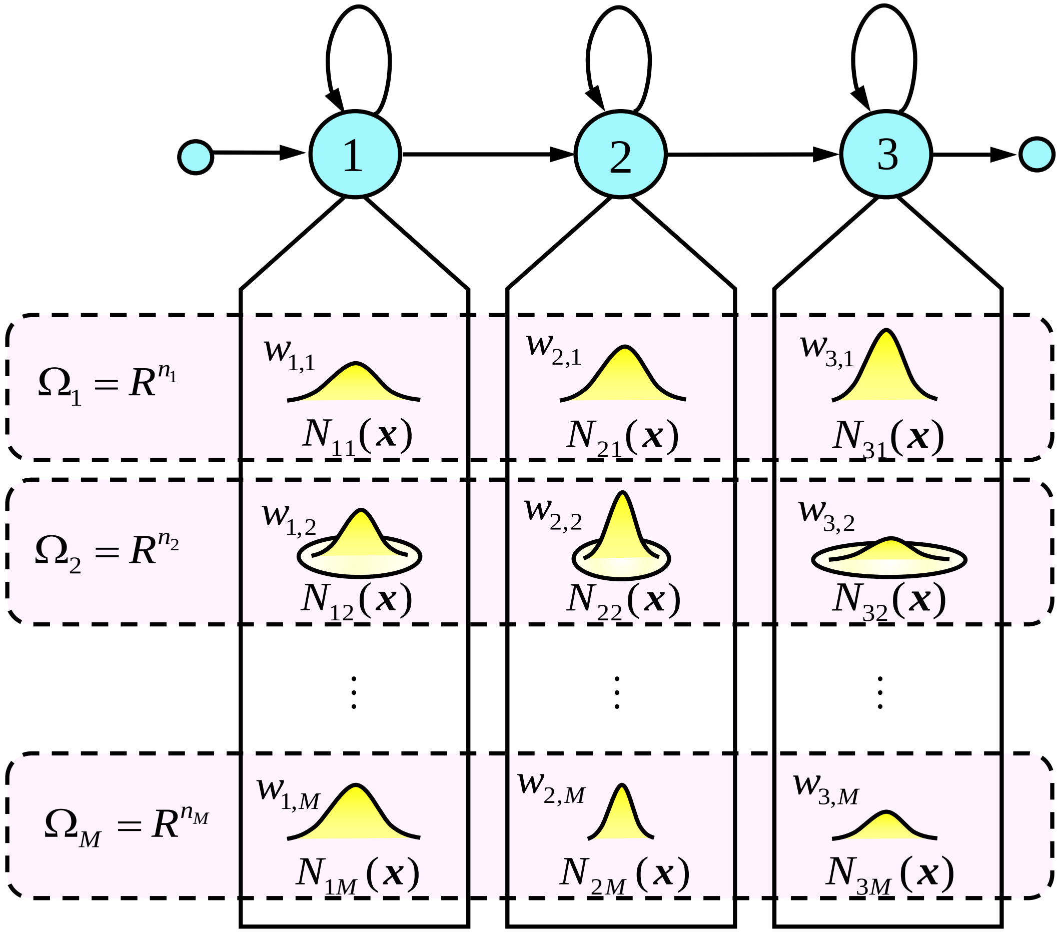
Sentence

Gaussian Mixture Models

Output probability is modeled by Gaussian mixture models

$$b_{\boldsymbol{q}}(\boldsymbol{o}_t) = b(\boldsymbol{o}_t|\boldsymbol{q}_t)$$

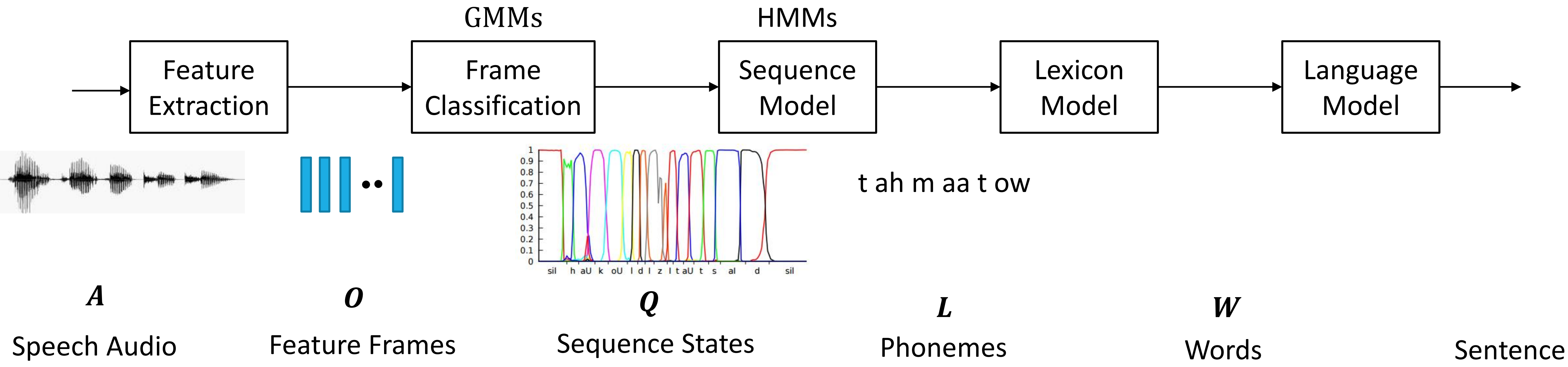
$$= \sum_{m=1}^{M} w_{\boldsymbol{q}_t,m} N(\boldsymbol{o}_t|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$



GMM-HMM in Speech Recognition

GMMs: Gaussian Mixture Models

HMMs: Hidden Markov Models



GMMs

HMMs

Feature Extraction

Frame Classification

Sequence Model

Lexicon Model

Language Model

t ah m aa t ow

$\boldsymbol{A}$

Speech Audio

$\boldsymbol{O}$

Feature Frames

$\boldsymbol{Q}$

Sequence States

$\boldsymbol{L}$

Phonemes

$\boldsymbol{W}$

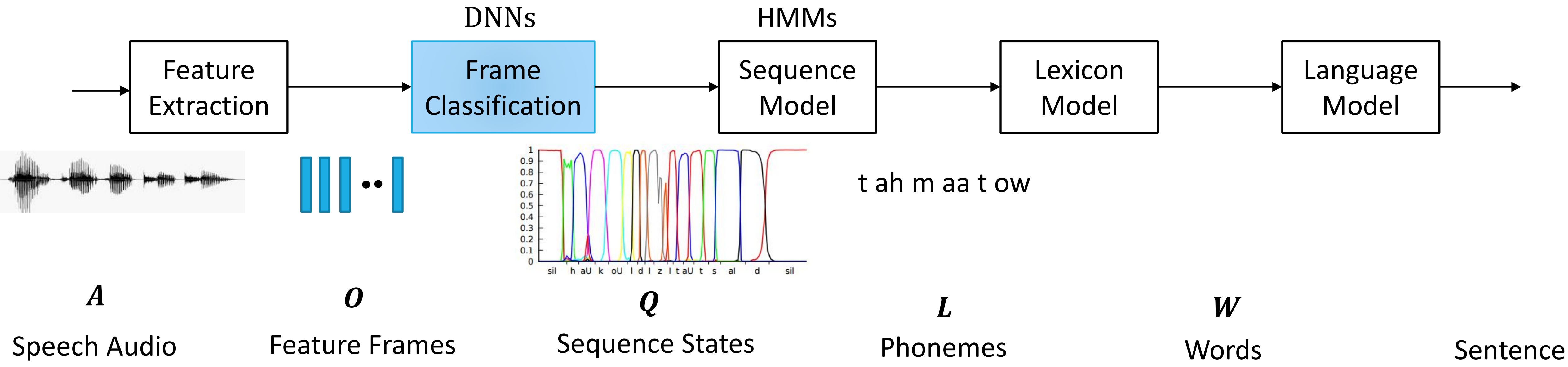
Words

Sentence

DNN-HMM in Speech Recognition

DNN: Deep Neural Networks

HMMs: Hidden Markov Models



DNNs

HMMs

Feature Extraction

Frame Classification

Sequence Model

Lexicon Model

Language Model

t ah m aa t ow

$\boldsymbol{A}$

Speech Audio

$\boldsymbol{O}$

Feature Frames

$\boldsymbol{Q}$

Sequence States

$\boldsymbol{L}$

Phonemes

$\boldsymbol{W}$

Words

Sentence

Ingredients for Deep Learning

Acoustic features

- Frequency domain features extracted from waveform
- 10ms interval between frames
- ~40 dimensions for each frame

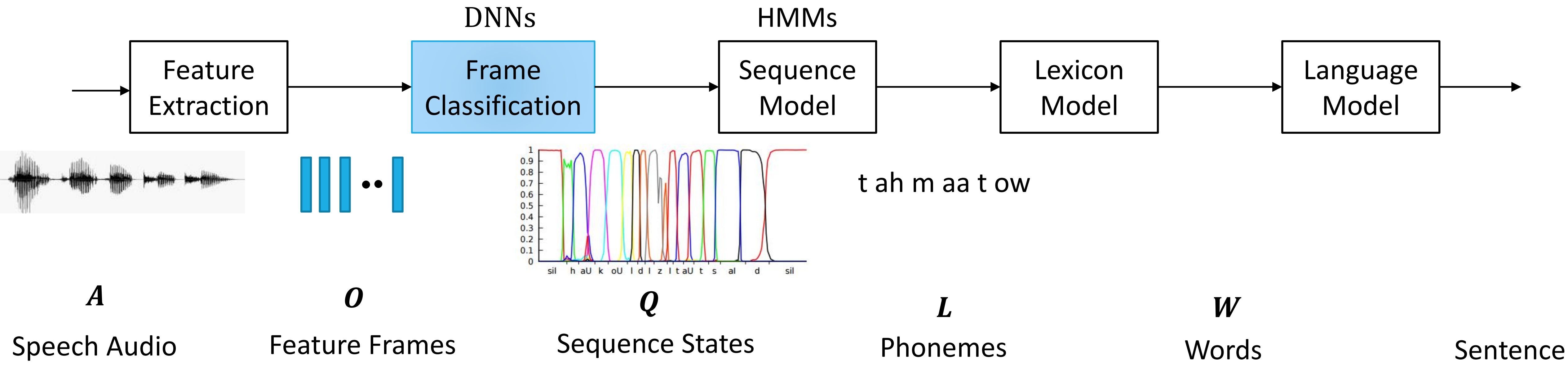
State alignments

- Tied-state of context-dependent HMMs
- Mapping between acoustic features and states

DNN-HMM in Speech Recognition

DNN: Deep Neural Networks

HMMs: Hidden Markov Models



DNNs

HMMs

Feature Extraction → Frame Classification → Sequence Model → Lexicon Model → Language Model →

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

$\boldsymbol{L}$ Phonemes

$\boldsymbol{W}$ Words

Sentence

Deep Models in HMM-based ASR

Classify acoustic features for state labels

Take softmax output as a posterior $P(state|\boldsymbol{o}_t) = P(\boldsymbol{q}_t|\boldsymbol{o}_t)$

Work as output probability in HMM

$$b_{\boldsymbol{q}}(\boldsymbol{o}_t) = b(\boldsymbol{o}_t|\boldsymbol{q}_t) = \frac{P(\boldsymbol{q}_t|\boldsymbol{o}_t)P(\boldsymbol{o}_t)}{P(\boldsymbol{q}_t)}$$

where $P(\boldsymbol{q}_t)$ is the prior probability for states

Fully Connected Networks

Features including 2 X 5 neighboring frames

- 1D convolution with kernel size 11

Classify 9,304 tied states

7 hidden layers X 2048 units with sigmoid activation

G. Hinton, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition. Signal Processing Magazine (2012).

Fully Connected Networks

Pre-training

- Unsupervised: stacked restricted Boltzmann machine (RBM)
- Supervised: iteratively adding layers from shallow model

Training

- Maximum cross entropy for frames

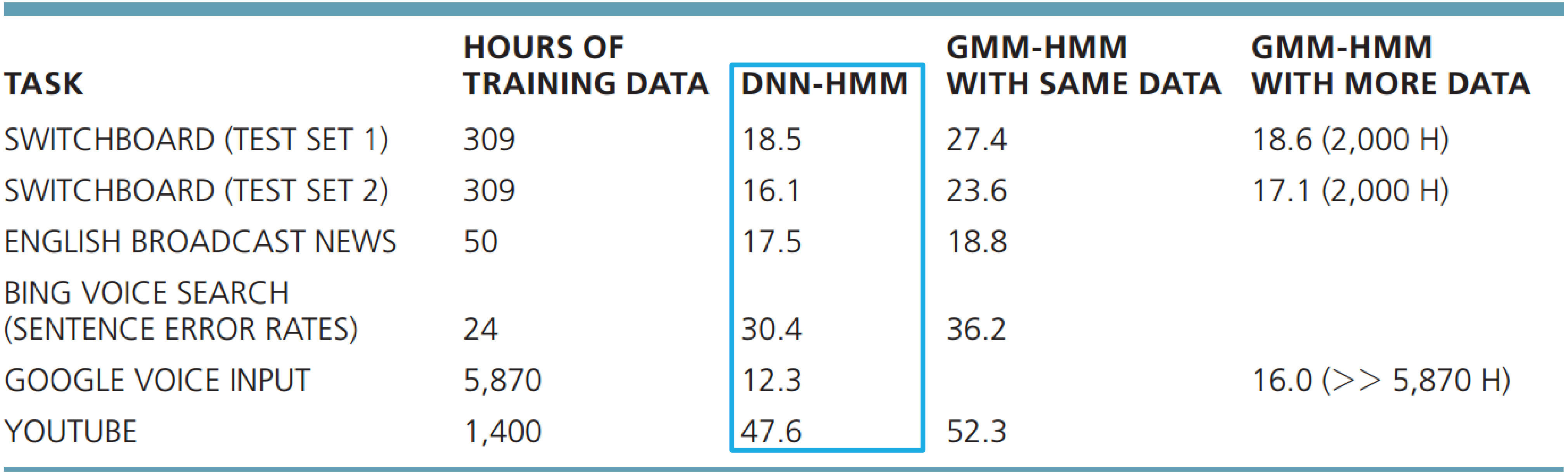
Fine-tuning

- Maximum mutual information for sequences

G. Hinton, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition. Signal Processing Magazine (2012).

Fully Connected Networks

Comparison on different large datasets



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	

G. Hinton, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition. Signal Processing Magazine (2012).

DNN-HMM vs. GMM-HMM

Deep models are more powerful

- GMM assumes data is generated from single component of mixture model
- GMM with diagonal variance matrix ignores correlation between dimensions

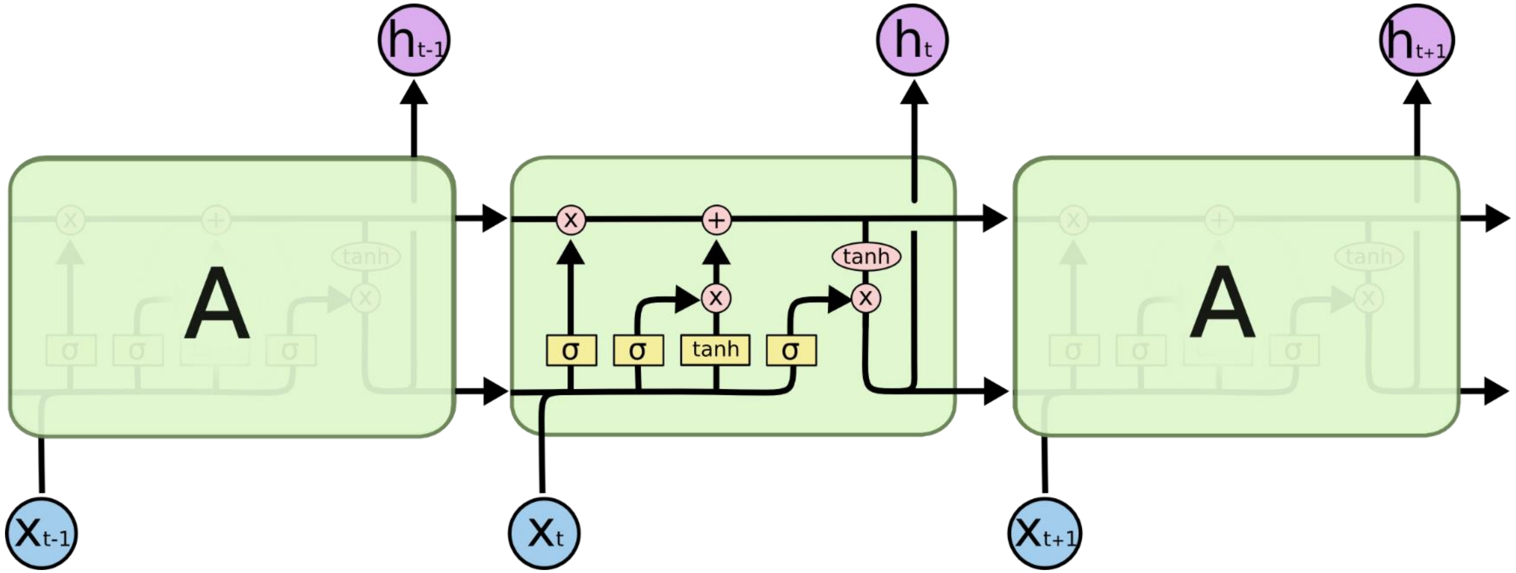
Deep models take data more efficiently

- GMM consists with many components and each learns from a small fraction of data

Deep models can be further improved by recent advances in deep learning

Recurrent Networks

Long Short Term Memory networks



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Recurrent Networks

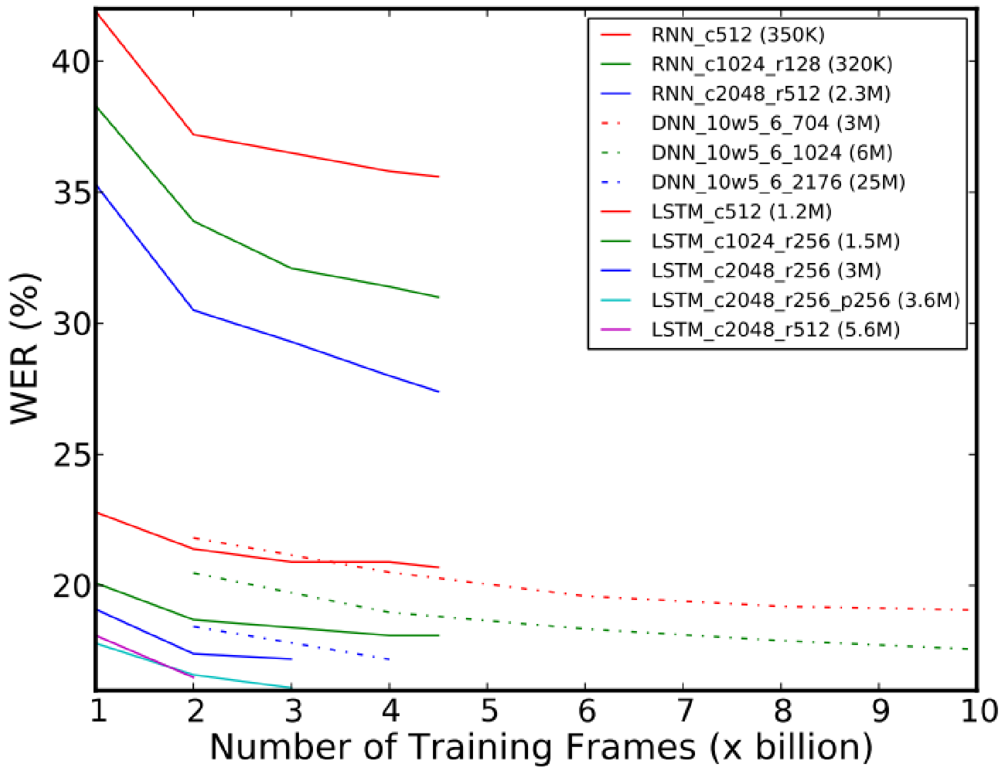


Fig. 5. 126 context independent phone HMM states.

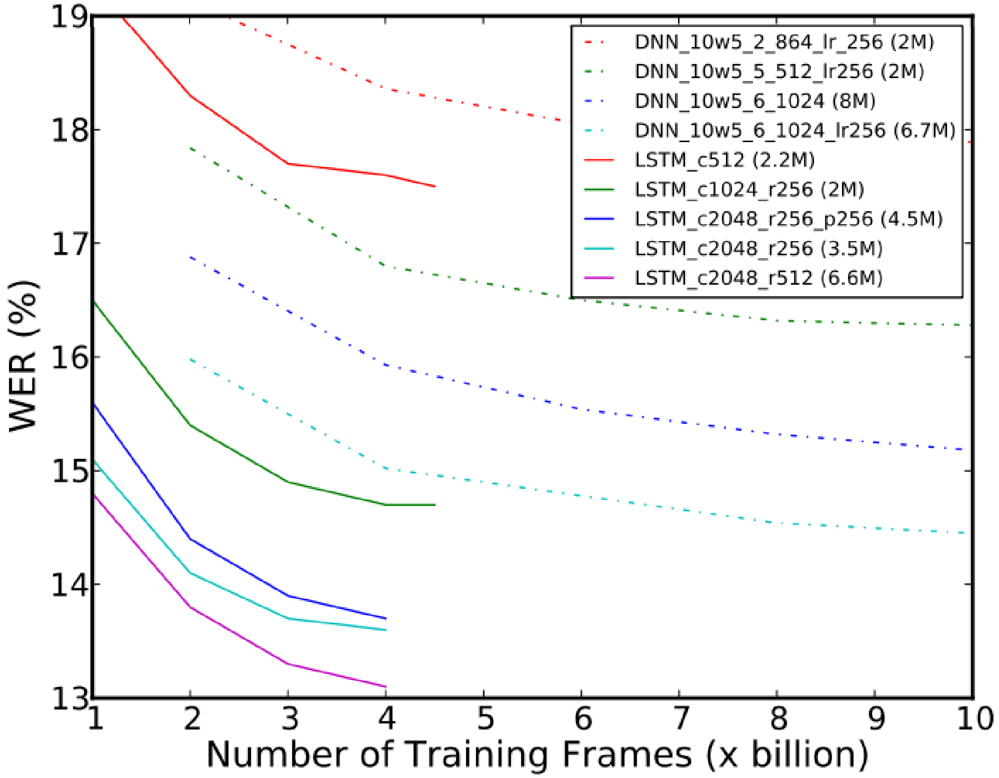
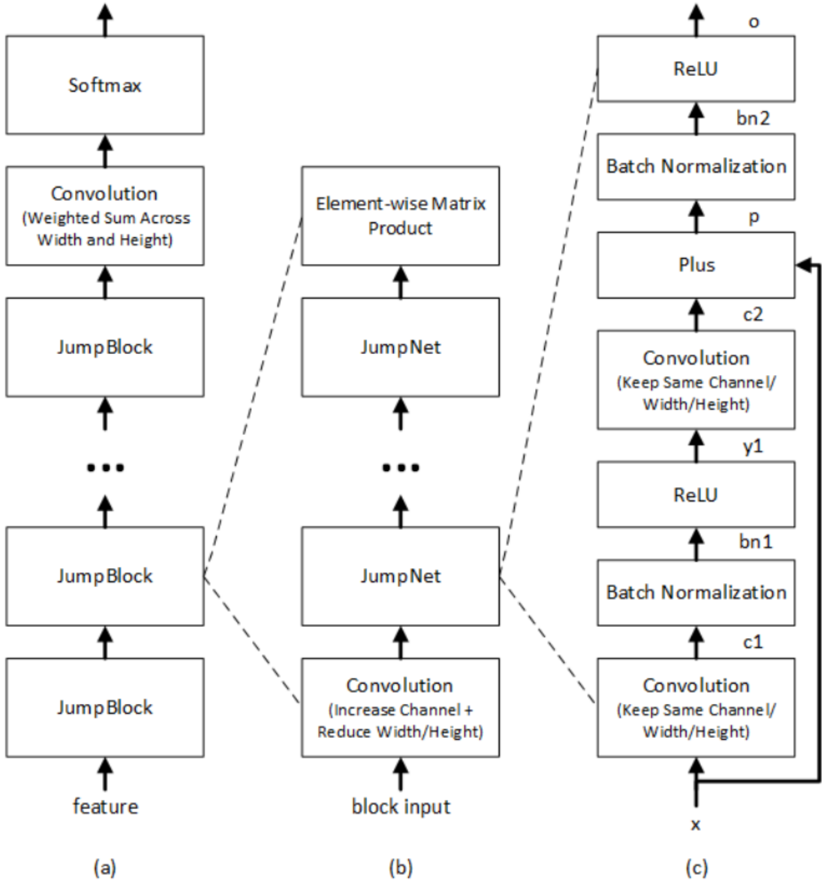


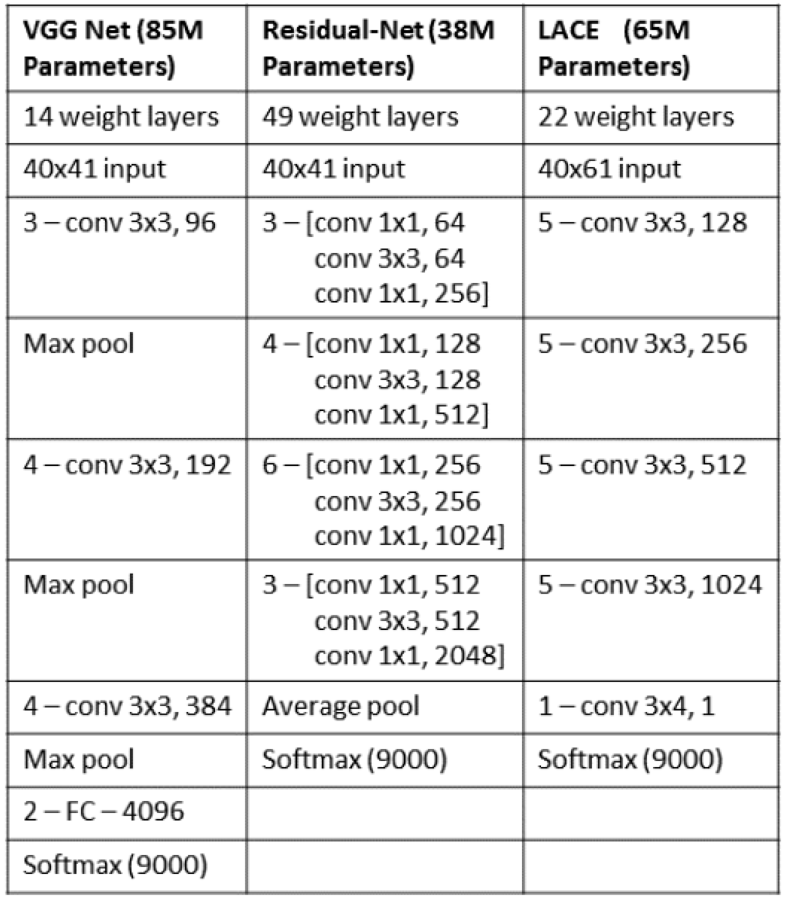
Fig. 6. 2000 context dependent phone HMM states.

H. Sak, A. Senior, and F. Beaufays. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. arXiv preprint arXiv:1402.1128v1 (2014).

Very Deep Networks

LACE





VGG Net (85M Parameters)	Residual-Net (38M Parameters)	LACE (65M Parameters)
14 weight layers	49 weight layers	22 weight layers
40x41 input	40x41 input	40x61 input
3 – conv 3x3, 96	3 – [conv 1x1, 64 conv 3x3, 64 conv 1x1, 256]	5 – conv 3x3, 128
Max pool	4 – [conv 1x1, 128 conv 3x3, 128 conv 1x1, 512]	5 – conv 3x3, 256
4 – conv 3x3, 192	6 – [conv 1x1, 256 conv 3x3, 256 conv 1x1, 1024]	5 – conv 3x3, 512
Max pool	3 – [conv 1x1, 512 conv 3x3, 512 conv 1x1, 2048]	5 – conv 3x3, 1024
4 – conv 3x3, 384	Average pool	1 – conv 3x4, 1
Max pool	Softmax (9000)	Softmax (9000)
2 – FC – 4096		
Softmax (9000)		

W. Xiong, et al. Achieving Human Parity in Conversational Speech Recognition. arXiv preprint arXiv:1610.05256 (2016).

Very Deep Networks

Speaker adaptive training

- Addition speaker id embedded vector as input

Language model with LSTM

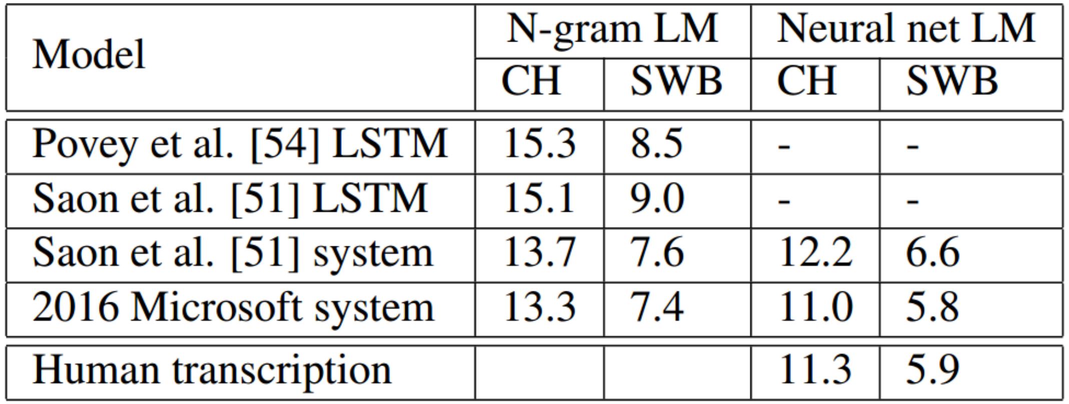
System combination

- Greedy-searched weight to combine multiple model system

Reuslts on the test set: CH and SWB

Word error rates	CH	SWB
ResNet	14.8	8.6
VGG	15.7	9.1
LACE	15.0	8.4

Exceed human accuracy



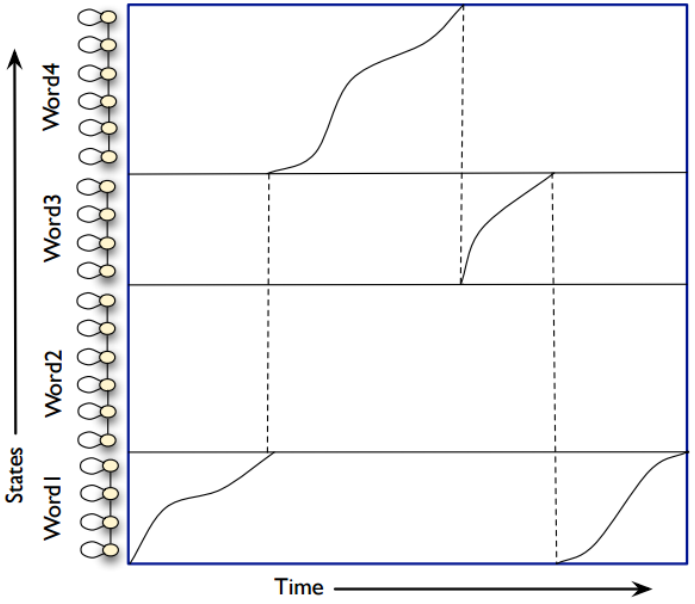
Model	N-gram LM		Neural net LM	
	CH	SWB	CH	SWB
Povey et al. [54] LSTM	15.3	8.5	-	-
Saon et al. [51] LSTM	15.1	9.0	-	-
Saon et al. [51] system	13.7	7.6	12.2	6.6
2016 Microsoft system	13.3	7.4	11.0	5.8
Human transcription			11.3	5.9

Limitations of DNN-HMM

Markov models only depend one previous states

History is discretely represented by 10k states

Decoding is slow to keep all 10k state in dynamic programing

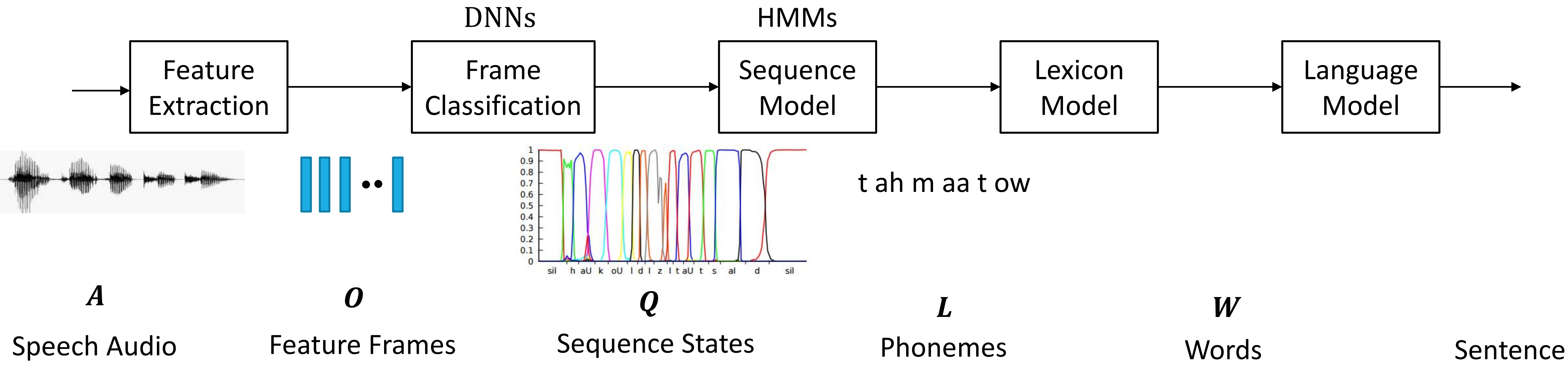


Decoding with dynamic programing

DNN-HMM in Speech Recognition

DNN: Deep Neural Networks

HMMs: Hidden Markov Models



DNNs

HMMs

Feature Extraction → Frame Classification → Sequence Model → Lexicon Model → Language Model →

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

$\boldsymbol{L}$ Phonemes

$\boldsymbol{W}$ Words

Sentence

End-to-End Deep Models based Automatic Speech Recognition

Connectionist Temporal Classification (CTC) based models

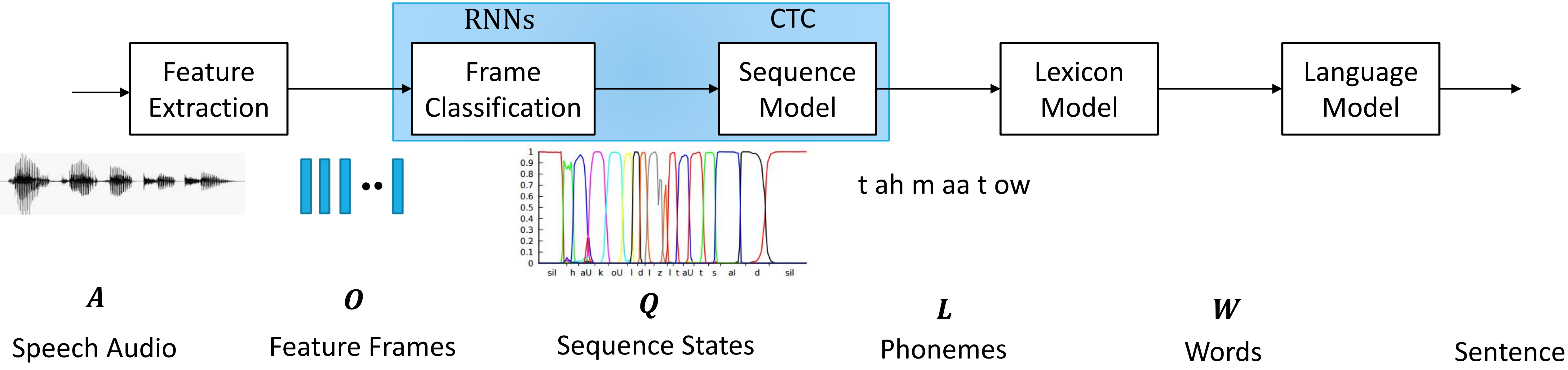
- LSTM CTC models
- Deep speech 2

Attention based models

CTC in Speech Recognition

RNN: Recurrent Neural Networks

CTC: Connectionist Temporal Classification



RNNs

CTC

Feature Extraction

Frame Classification

Sequence Model

Lexicon Model

Language Model

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

$\boldsymbol{L}$ Phonemes

$\boldsymbol{W}$ Words

Sentence

Connectionist Temporal Classification (CTC)

Method for labeling unsegmented data sequences

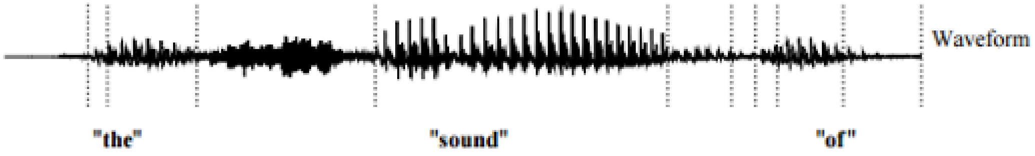
- Raw waveforms and text transcription

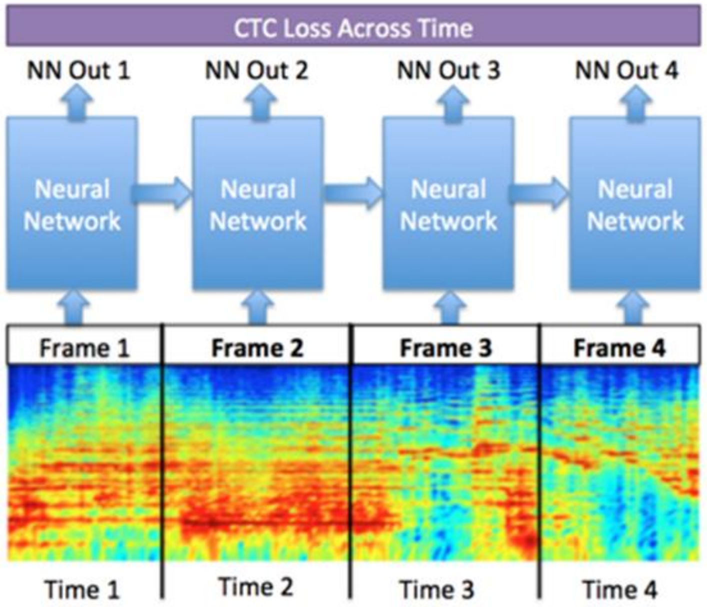
Differentiable objective function

- Gradient based training
- $O^{ML}(S, \mathcal{N}_w) = -\sum_{(x,z)\in S} \ln\big(p(z|x)\big)$

Used in various ASR and TTS architectures

- DeepSpeech (ASR)
- DeepVoice(TTS)





https://gab41.lab41.org/speech-recognition-you-down-with-ctc-8d3b558943f0, ftp://ftp.idsia.ch/pub/juergen/icml2006.pdf

CTC Problem

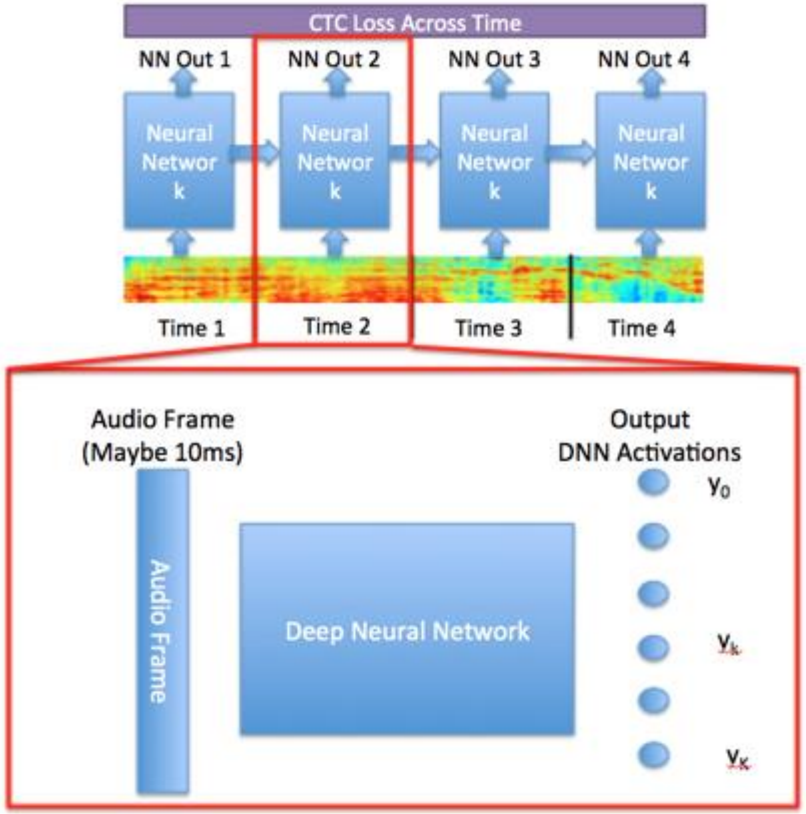
Training examples $S = \{(x_1, z_1), \dots \{x_N, z_N\}) \in \mathcal{D}_{\mathcal{X}\times\mathcal{Z}}$

- $x \in \mathcal{X}$ are waveforms
- $z \in \mathcal{Z}$ are text transcripts

Goal: train temporal classifier $h : \mathcal{X} \rightarrow \mathcal{Z}$

Architecture

- Input layer accepts audio frame
- Some network (usually CNN or RNN)
- Softmax output over phones



https://gab41.lab41.org/speech-recognition-you-down-with-ctc-8d3b558943f0, ftp://ftp.idsia.ch/pub/juergen/icml2006.pdf

CTC Description

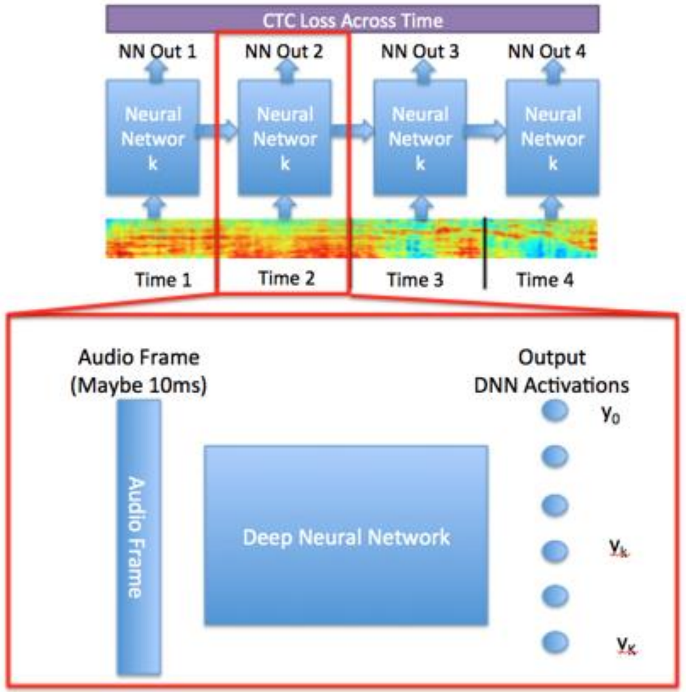
Let $\mathcal{N}_w(x) = \{y_k^t\}$ be NN with a softmax output

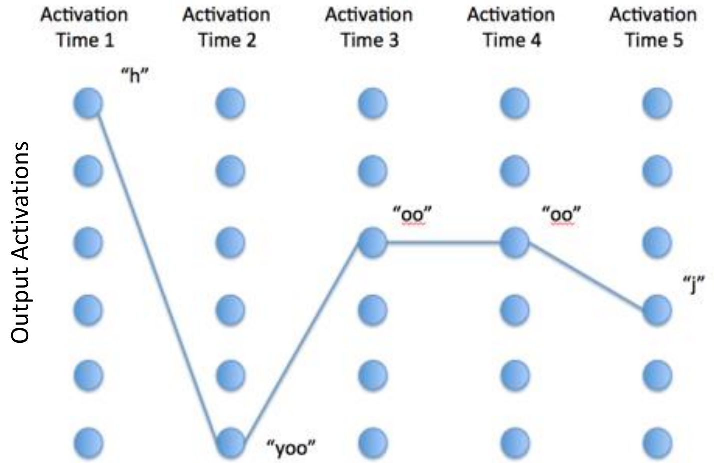
- y_k^t is activation of output unit k at time frame t
- Activations over time define distribution over L^T

Sequences over $L^T \triangleq \pi = \{\pi_1, \dots, \pi_T\}$ are paths

Optimize for best path:

$$\mathrm{P}(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall \pi \in L^T$$





https://gab41.lab41.org/speech-recognition-you-down-with-ctc-8d3b558943f0, ftp://ftp.idsia.ch/pub/juergen/icml2006.pdf

CTC Objective

Paths are not equivalent to the label, $\pi \neq l$

Optimize for best label:

$$P(l|x) = \sum_{\pi} P(l|\pi)P(\pi|x)$$

Solve objective above with dynamic time warping

- Forward-backward algorithm
  - Forward variables α
  - Backward variables β

$$P(l|x) = \sum_{s=1}^{|l|} \frac{\alpha_t(s)\beta_t(s)}{y_{l_s}^t}$$

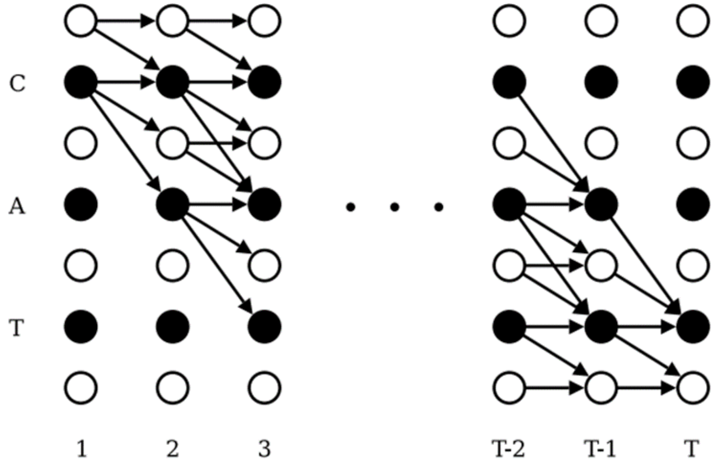
- Maps and searches only paths that correspond to target label

$l = \{a\}$

aa______
aaa_____
_aaa____
___aaaa_
_aaaaaaa

$l = \{bee\}$

bbbeee_ee
_bb_ee__e
__bbbe_e_



https://github.com/yiwangbaidu/notes/blob/master/CTC/CTC.pdf, ftp://ftp.idsia.ch/pub/juergen/icml2006.pdf

CTC Objective and Gradient

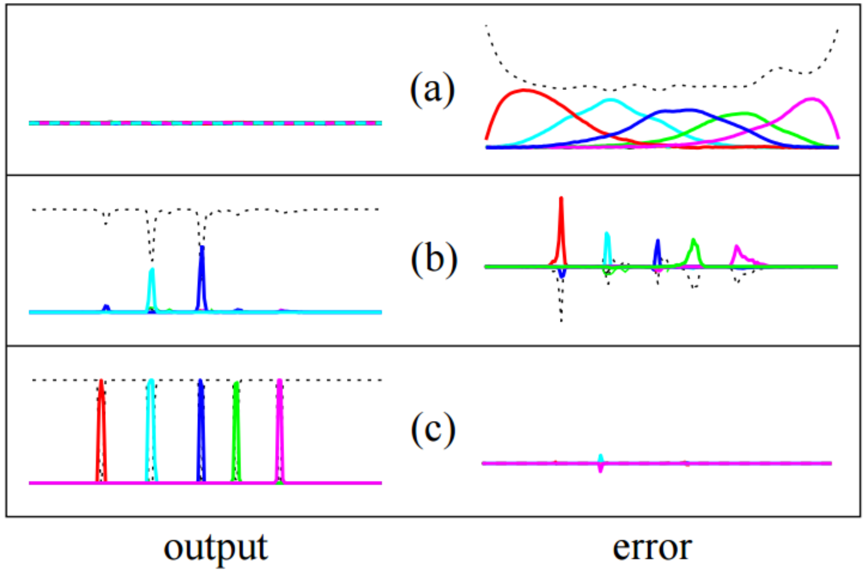
Objective function:

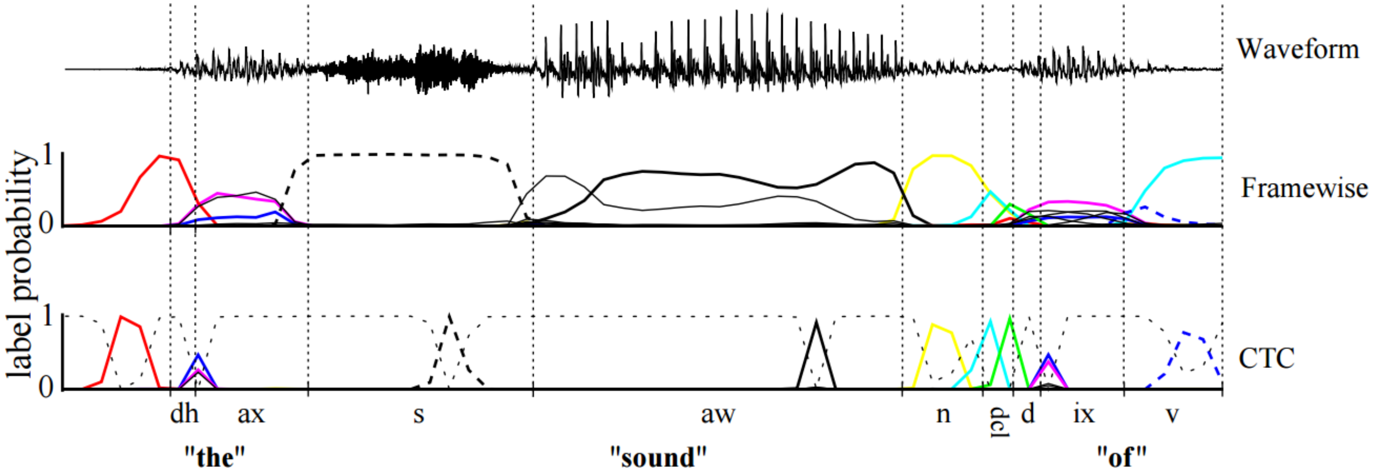
$$O^{ML}(S, \mathcal{N}_w) = -\sum_{(x,z)\in S} \ln\big(p(z|x)\big)$$

Gradient:

$$\frac{\partial O^{ML}(\{(x,z)\}, \mathcal{N}_w)}{\partial u_k^t} = y_k^t - \frac{1}{y_k^t Z_t} \sum_{s\in lab(z,k)} \hat{\alpha}_t(s)\hat{\beta}_t(s)$$

$$\text{where } Z_t \triangleq \sum_{s=1}^{|l'|} \frac{\alpha_t(s)\beta_t(s)}{y_{l'_s}^t}$$





ftp://ftp.idsia.ch/pub/juergen/icml2006.pdf

LSTM CTC Models

Word error rate compared with HMM based models

	Context	Error Rate (%)
LSTM-HMM	Uni	8.9
	Bi	9.1
LSTM-CTC	Uni	9.4
	Bi	8.5

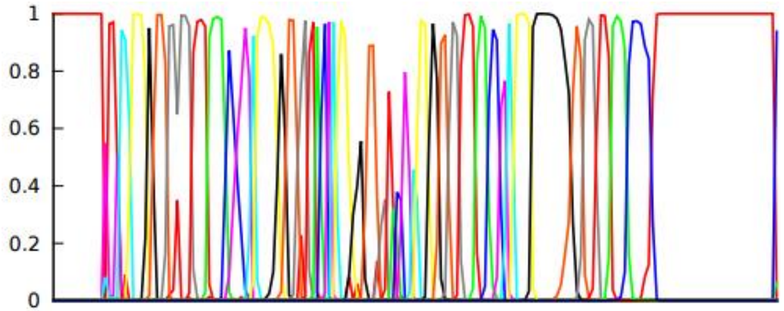
Bi-direction LSTM is more essential for CTC

Sak, Haşim, et al. "Learning acoustic frame labeling for speech recognition with recurrent neural networks." ICASSP, 2015.

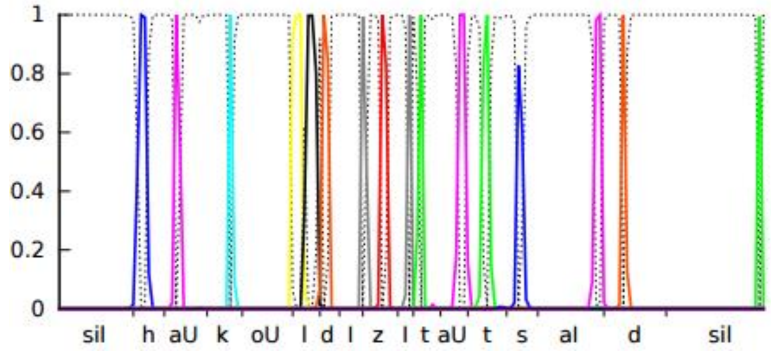
CTC vs HMM

Output probability of LSTM

HMM



CTC



CTC embeds history in continuous hidden space, more capable than HMM

CTC has spiky predictions, more discriminable between states than HMM

CTC with less states (40) is significantly faster for decoding than HMM (10k states)

Sak, Haşim, et al. "Learning acoustic frame labeling for speech recognition with recurrent neural networks." ICASSP, 2015.

Deep Speech 2

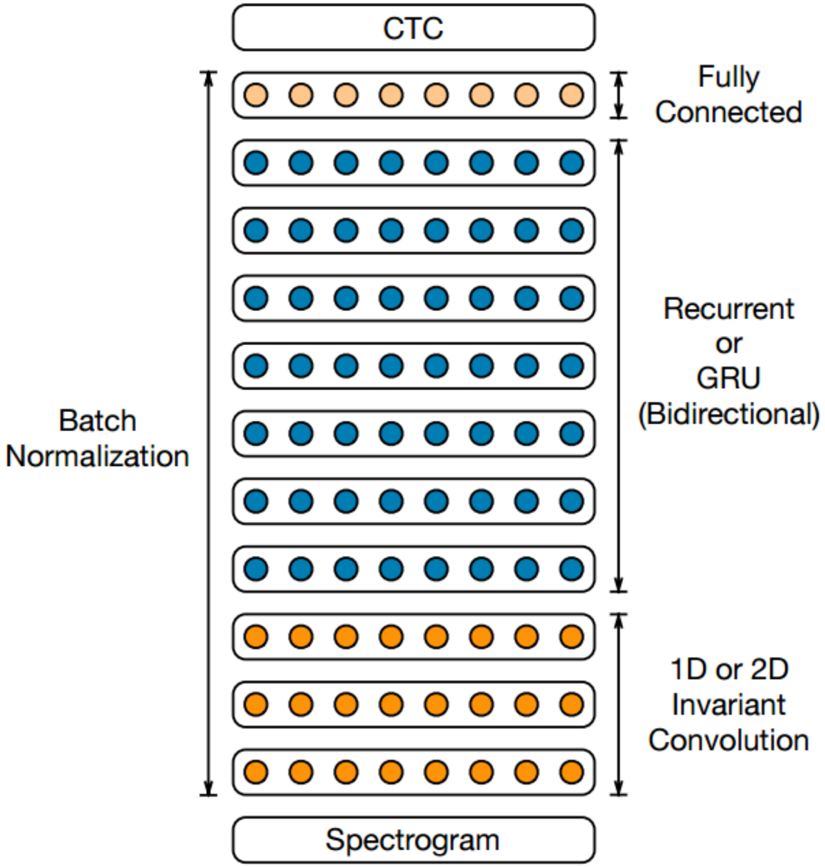
3 layers of 2D-invariant convolution

7 layers of bidirectional simple recurrence

100M parameters

Word error rates

Test set	Deep speech 2	Human
WSJ eval’92	3.60	5.03
WSJ eval’93	4.98	8.08
LibriSpeech test-clean	5.33	5.83
LibriSpeech test-other	13.25	12.69

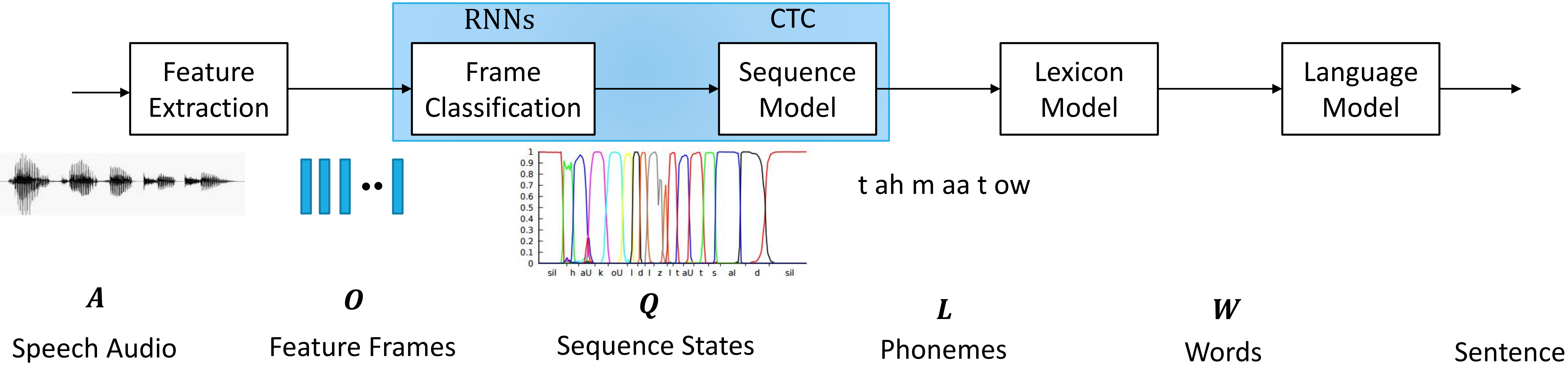

CTC
Fully Connected
Recurrent or GRU (Bidirectional)
Batch Normalization
1D or 2D Invariant Convolution
Spectrogram

Amodei, Dario, et al. "Deep speech 2: End-to-end speech recognition in english and mandarin." *arXiv preprint arXiv:1512.02595* (2015).

CTC in Speech Recognition

RNN: Recurrent Neural Networks

CTC: Connectionist Temporal Classification



RNNs

CTC

Feature Extraction

Frame Classification

Sequence Model

Lexicon Model

Language Model

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

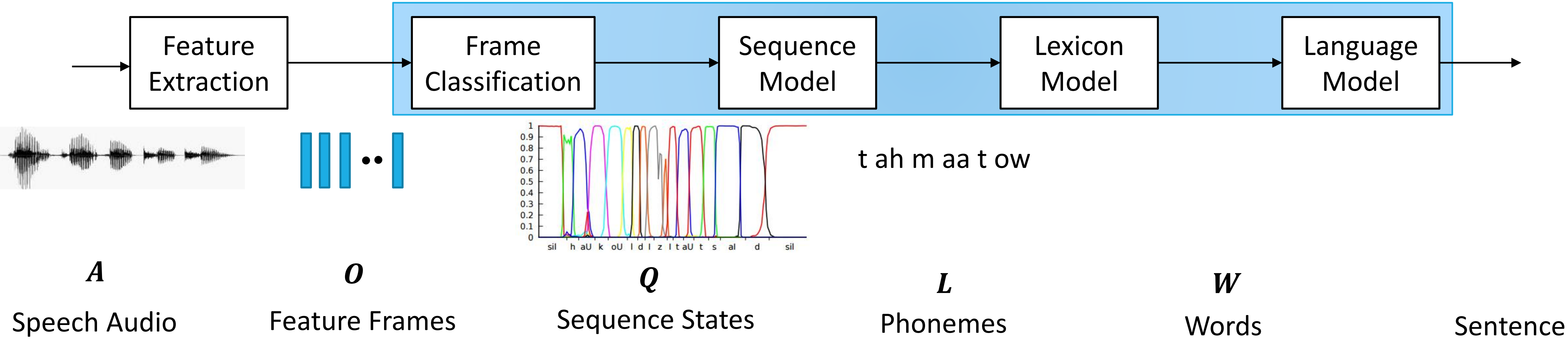
$\boldsymbol{L}$ Phonemes

$\boldsymbol{W}$ Words

Sentence

Attention in Speech Recognition

Predict character sequence directly



Feature Extraction

Frame Classification

Sequence Model

Lexicon Model

Language Model

t ah m aa t ow

$\boldsymbol{A}$ Speech Audio

$\boldsymbol{O}$ Feature Frames

$\boldsymbol{Q}$ Sequence States

$\boldsymbol{L}$ Phonemes

$\boldsymbol{W}$ Words

Sentence

Attention based Models

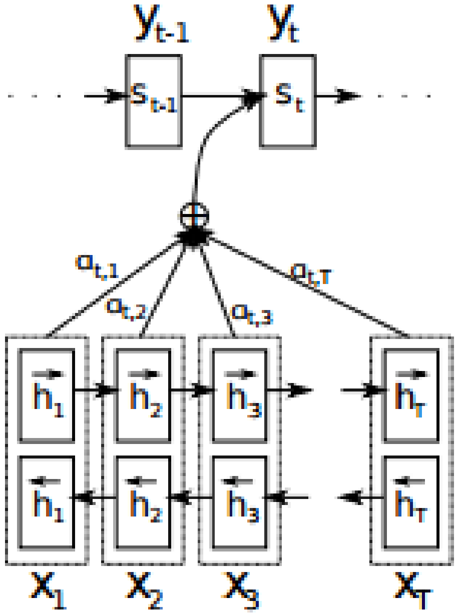
Recall previous lecture

- Image generation
- Image text embedding
- Question answering
- Machine translation

Weighted sum of previous history

$$\alpha_{tj} = softmax(e_{tj}) = \frac{\exp(e_{tj})}{\sum_{k=1}^{T} \exp(e_{tk})}$$

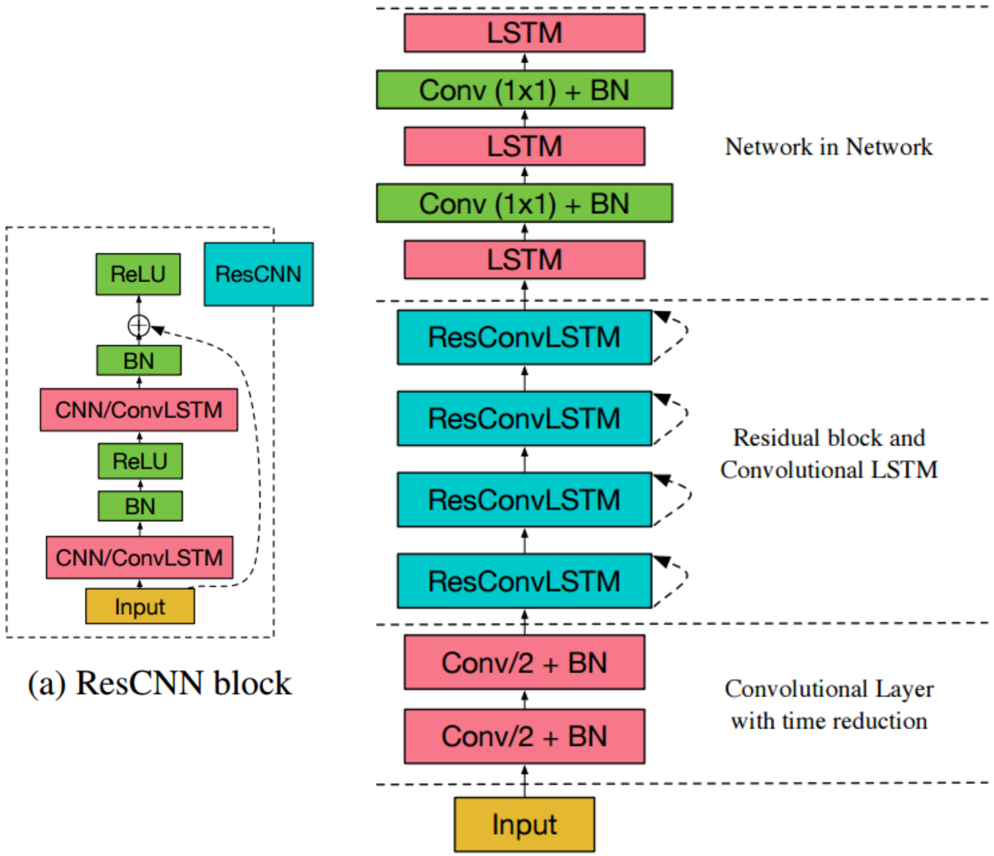
Essential for modeling long sequences like speech



Attention based Models

Word error rate w/o language models
on Wall Street Journal test eval92

	WER
CTC	30.1
Proposed	10.53

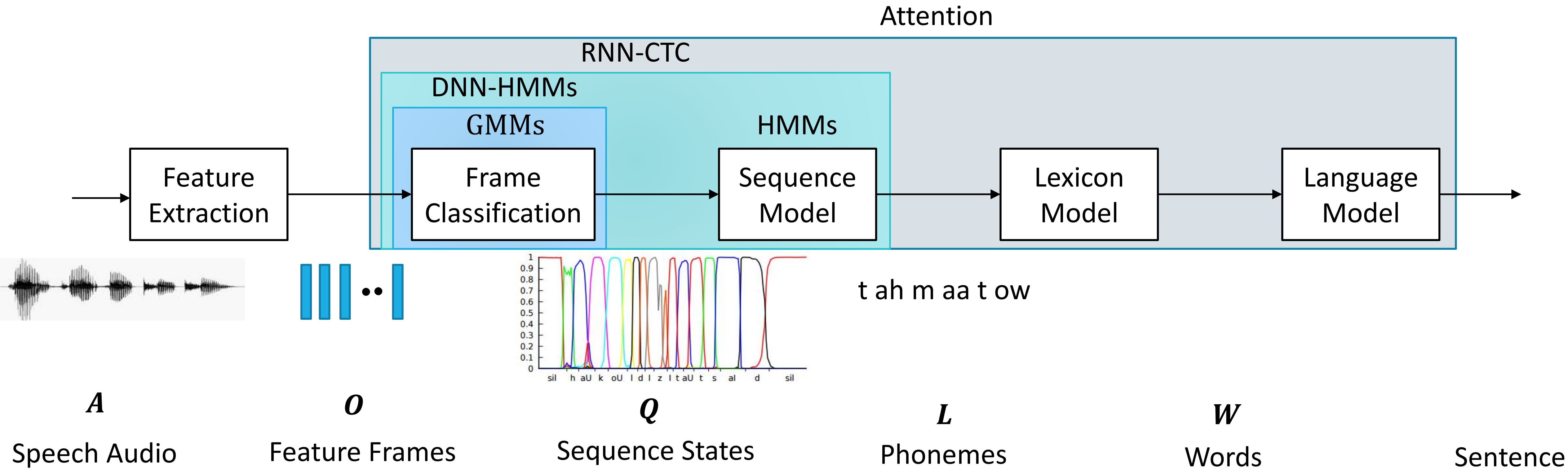


(a) ResCNN block

Encoder Network

Zhang, Yu, William Chan, and Navdeep Jaitly. "Very deep convolutional networks for end-to-end speech recognition." *arXiv preprint arXiv:1610.03022* (2016).

Deep learning in Speech Recognition



Attention

RNN-CTC

DNN-HMMs

GMMs

HMMs

Feature Extraction

Frame Classification

Sequence Model

Lexicon Model

Language Model

t ah m aa t ow

$\boldsymbol{A}$

Speech Audio

$\boldsymbol{O}$

Feature Frames

$\boldsymbol{Q}$

Sequence States

$\boldsymbol{L}$

Phonemes

$\boldsymbol{W}$

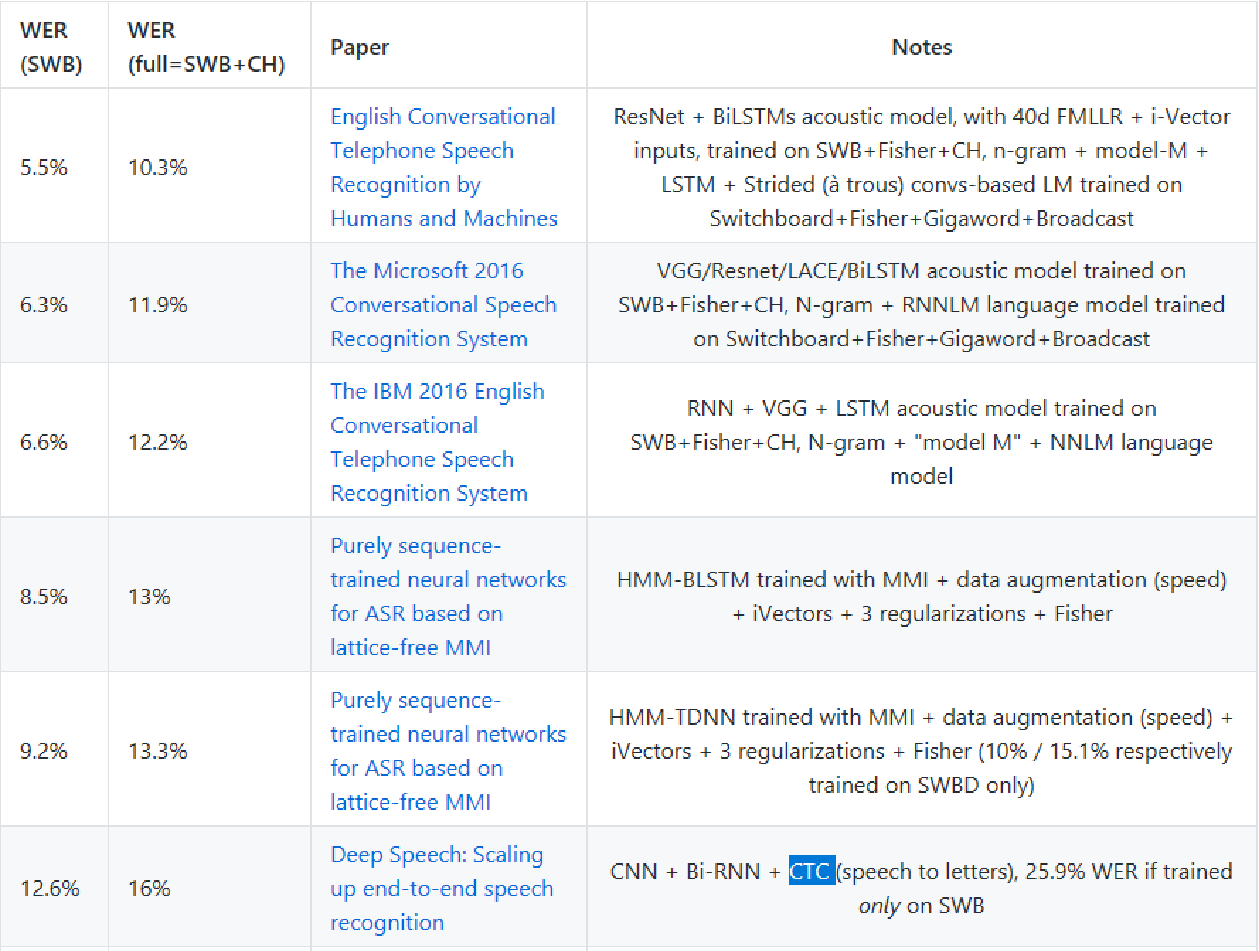
Words

Sentence

Deep learning in Speech Recognition

Top systems on dataset Switchboard

HMM based system still performs better than End-to-End system on large scale dataset



WER (SWB)	WER (full=SWB+CH)	Paper	Notes
5.5%	10.3%	English Conversational Telephone Speech Recognition by Humans and Machines	ResNet + BiLSTMs acoustic model, with 40d FMLLR + i-Vector inputs, trained on SWB+Fisher+CH, n-gram + model-M + LSTM + Strided (à trous) convs-based LM trained on Switchboard+Fisher+Gigaword+Broadcast
6.3%	11.9%	The Microsoft 2016 Conversational Speech Recognition System	VGG/Resnet/LACE/BiLSTM acoustic model trained on SWB+Fisher+CH, N-gram + RNNLM language model trained on Switchboard+Fisher+Gigaword+Broadcast
6.6%	12.2%	The IBM 2016 English Conversational Telephone Speech Recognition System	RNN + VGG + LSTM acoustic model trained on SWB+Fisher+CH, N-gram + "model M" + NNLM language model
8.5%	13%	Purely sequence-trained neural networks for ASR based on lattice-free MMI	HMM-BLSTM trained with MMI + data augmentation (speed) + iVectors + 3 regularizations + Fisher
9.2%	13.3%	Purely sequence-trained neural networks for ASR based on lattice-free MMI	HMM-TDNN trained with MMI + data augmentation (speed) + iVectors + 3 regularizations + Fisher (10% / 15.1% respectively trained on SWBD only)
12.6%	16%	Deep Speech: Scaling up end-to-end speech recognition	CNN + Bi-RNN + CTC (speech to letters), 25.9% WER if trained *only* on SWB

https://github.com/syhw/wer_are_we

Text to Speech (TTS)

Outline

Automatic Speech Recognition (ASR)

- Deep models with HMMs
- Connectionist Temporal Classification (CTC)
- Attention based models

Text to Speech (TTS)

- WaveNet
- DeepVoice
- Tacotron

Bonus: Music Generation

Traditional Methods

Concatenative approaches vs. Statistical Parametric approach

- Concatenative: More natural sounding, Less flexible, Take up more space
- Statistical Parametric: Muffled, more flexible, Smaller models

Traditionally, two stages: frontend and backend

- Frontend analyzes text and determines phonemes, stresses, pitch, etc.
- Backend generates the audio

Moving towards models which can convert directly from text to audio, with the model itself learning any intermediate representations necessary

WaveNet: A Generative Model for Raw Audio (2016)

Operates on **Raw Waveform** and generates a raw waveform (audio samples)

Each audio sample's predictive distribution conditioned on all previous ones

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

Problems to overcome:

- Tons of samples: 16kHz = 16,000 samples per second
- Correlations in both small time scales and large time scales
- 16 bit audio means 65,536 probabilities per time step

Problem: Too many samples, and different time scales of correlation

Solution: Causal Convolutions

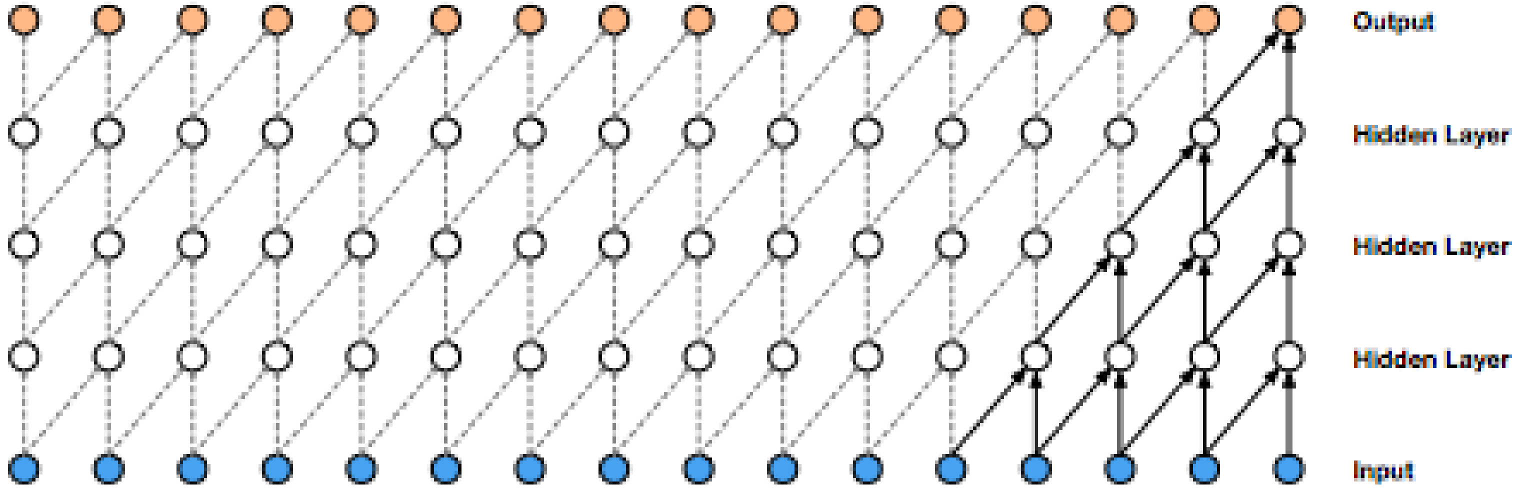


Figure 2: Visualization of a stack of causal convolutional layers.

https://arxiv.org/pdf/1609.03499.pdf

Causal Convolutions

Prediction of an audio sample can only depend on timestamps before it (nothing from the future)

No recurrent connections (Not an RNN) => Faster to train than RNN’s

- Especially with very long sequences

Glaring problem: Requires many layers to increase the receptive field

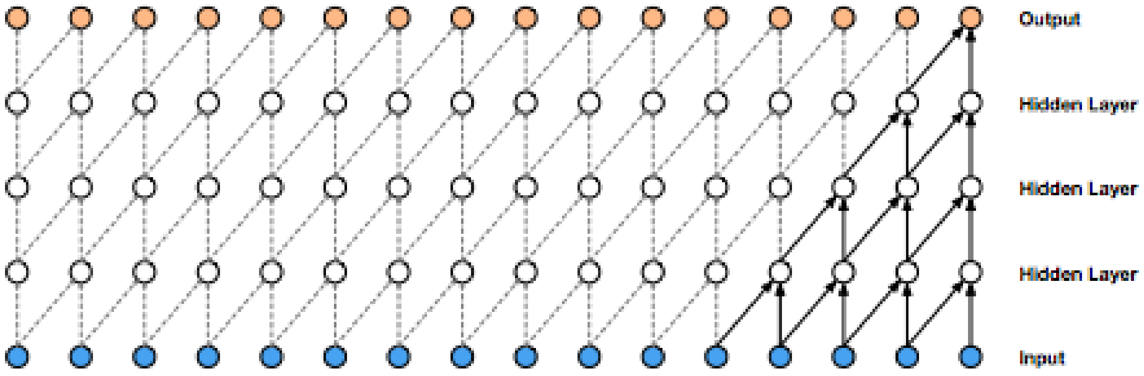


Figure 2: Visualization of a stack of causal convolutional layers.

https://arxiv.org/pdf/1609.03499.pdf

Dilated Causal Convolutions

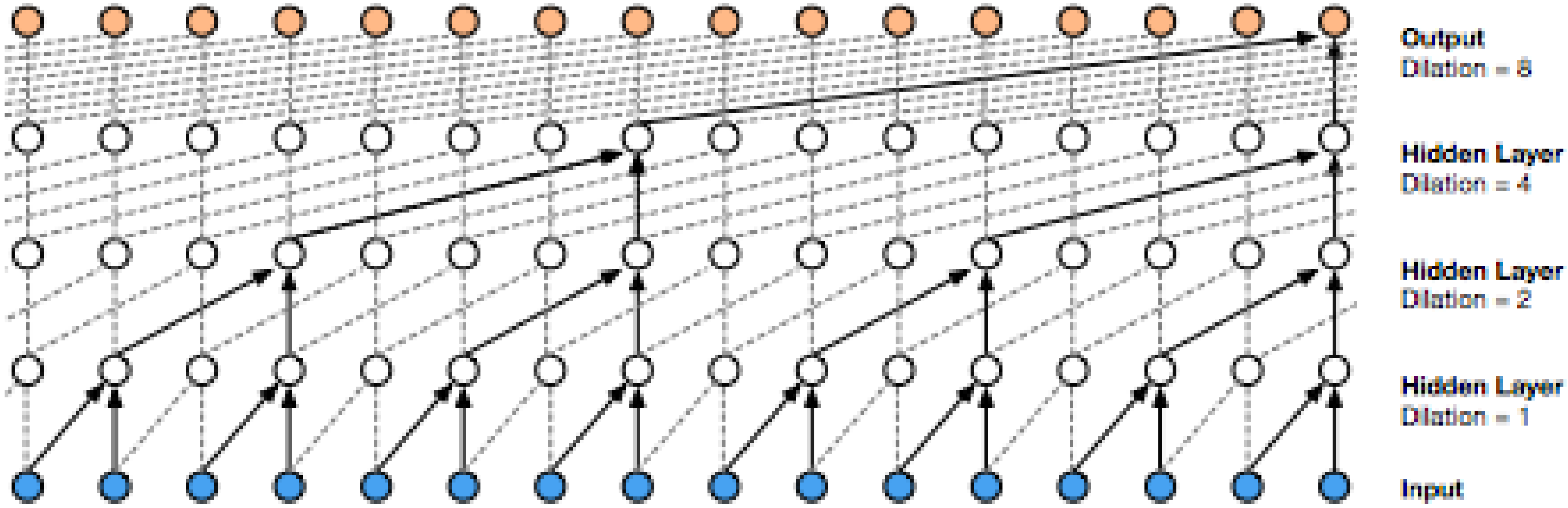


Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

https://arxiv.org/pdf/1609.03499.pdf

Dilated Causal Convolutions

Filter is applied to area much larger than its length by skipping inputs at each layer: Large receptive field

Allows information to be gleaned from both temporally close and far samples

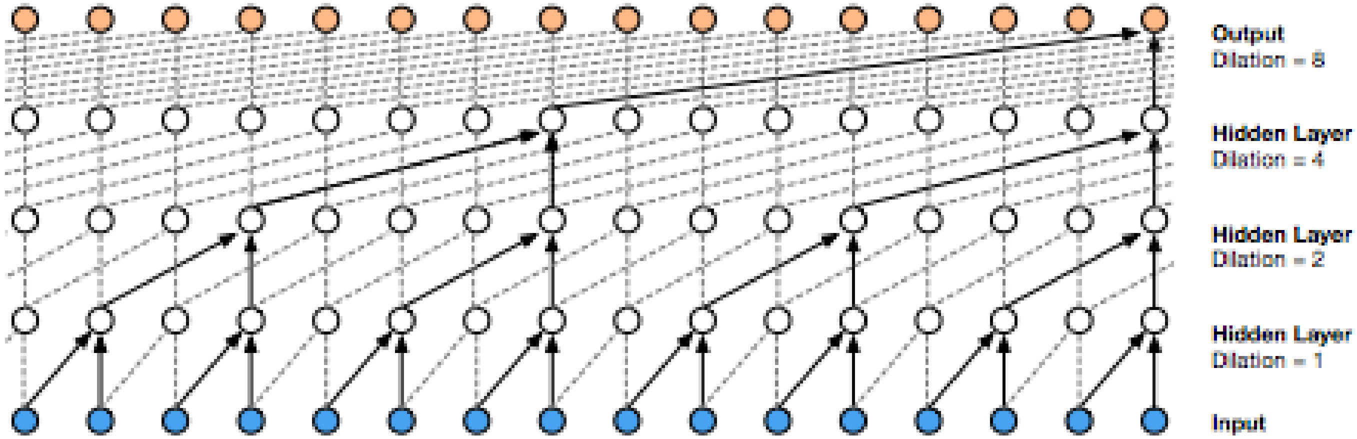


Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

https://arxiv.org/pdf/1609.03499.pdf

Skip connections are used throughout the network

Speeds up convergence

Enables training of deeper models

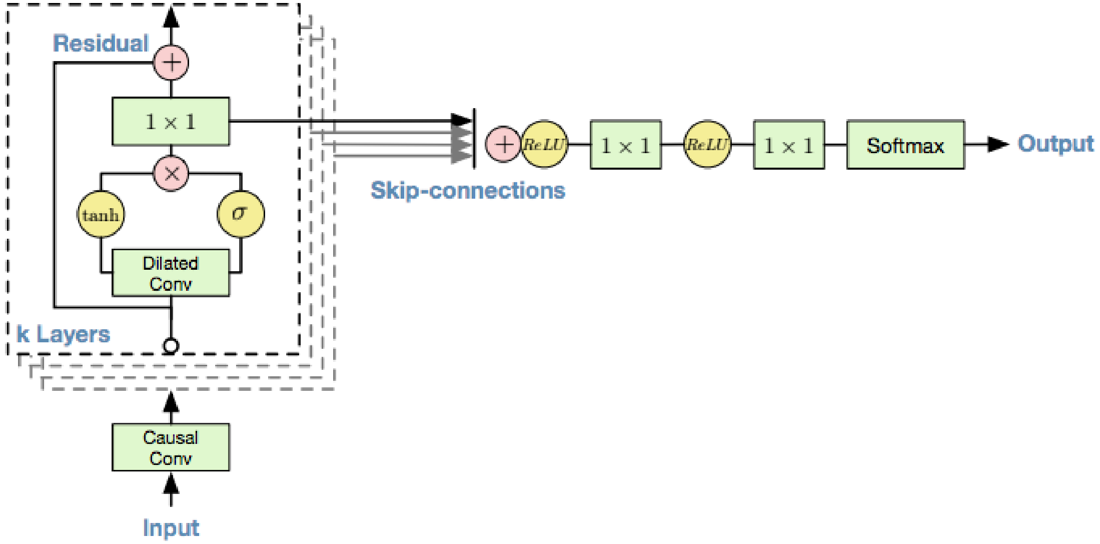


Figure 4: Overview of the residual block and the entire architecture.

https://arxiv.org/pdf/1609.03499.pdf

Where are we now?

No input => Generic framework

https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Add in other input

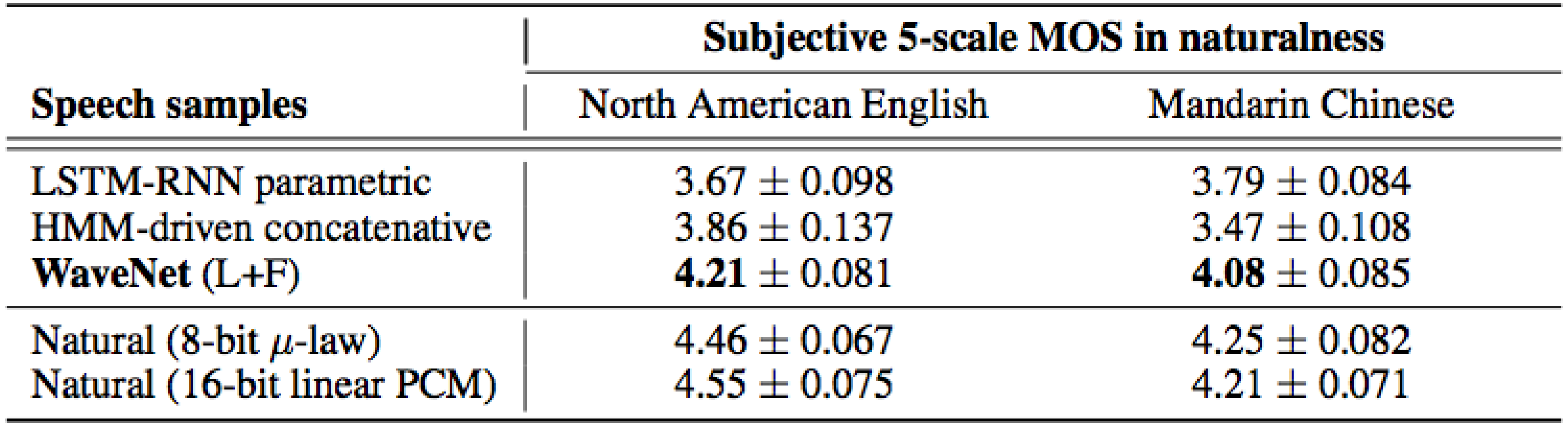
Generic enough to allow for both universal and local input

- Universal: Speaker Identity
- Local: Phonemes, inflection, stress

$$p(\mathbf{x} \mid \mathbf{h}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}, \mathbf{h})$$

Results

https://deepmind.com/blog/wavenet-generative-model-raw-audio/



	Subjective 5-scale MOS in naturalness	
Speech samples	North American English	Mandarin Chinese
LSTM-RNN parametric	3.67 ± 0.098	3.79 ± 0.084
HMM-driven concatenative	3.86 ± 0.137	3.47 ± 0.108
WaveNet (L+F)	**4.21** ± 0.081	**4.08** ± 0.085
Natural (8-bit μ-law)	4.46 ± 0.067	4.25 ± 0.082
Natural (16-bit linear PCM)	4.55 ± 0.075	4.21 ± 0.071

SampleRNN

How to deal with different time scales? Modularize.

Use different modules operating at different clock rates to deal with varying levels of abstraction

Sample-Level Modules and Frame-Level Modules

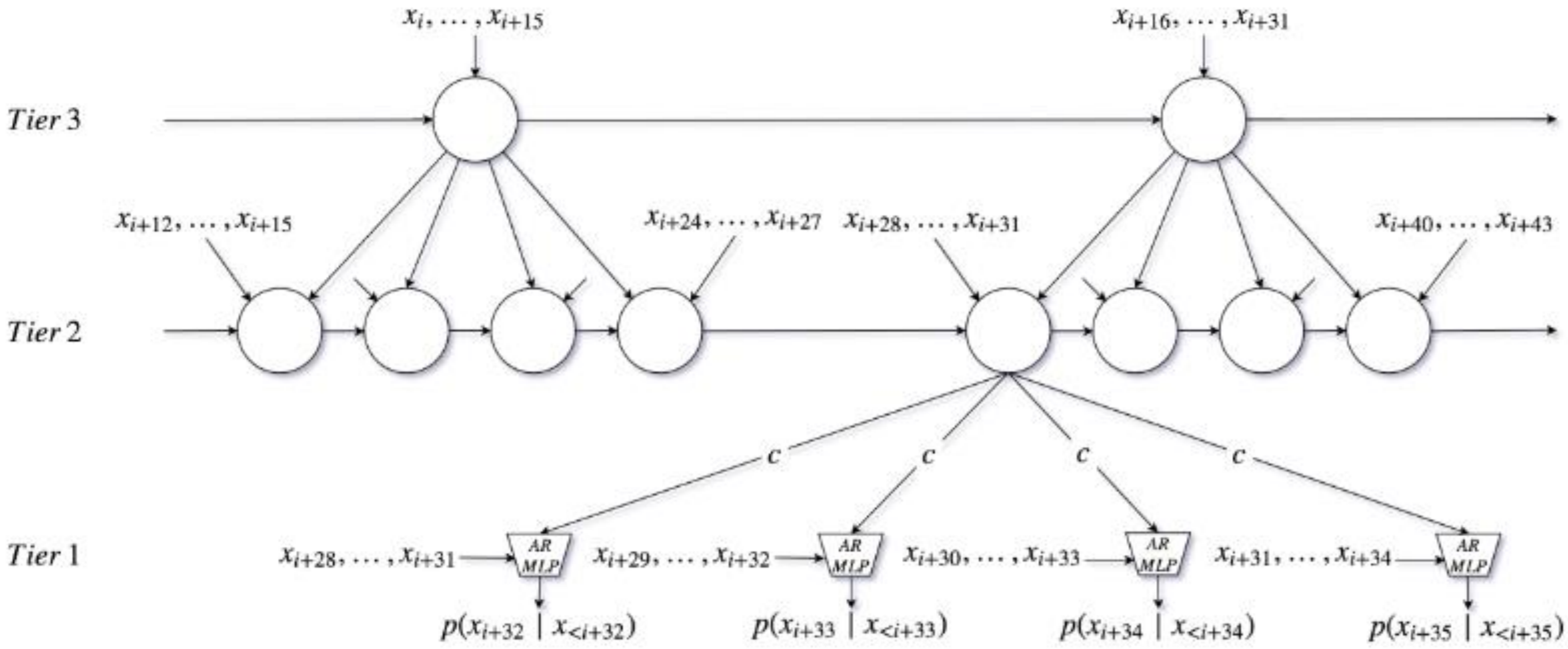

$x_i, \ldots, x_{i+15}$
$x_{i+16}, \ldots, x_{i+31}$
Tier 3
$x_{i+12}, \ldots, x_{i+15}$
$x_{i+24}, \ldots, x_{i+27}$
$x_{i+28}, \ldots, x_{i+31}$
$x_{i+40}, \ldots, x_{i+43}$
Tier 2
c
c
c
c
Tier 1
$x_{i+28}, \ldots, x_{i+31}$
AR MLP
$x_{i+29}, \ldots, x_{i+32}$
AR MLP
$x_{i+30}, \ldots, x_{i+33}$
AR MLP
$x_{i+31}, \ldots, x_{i+34}$
AR MLP
$p(x_{i+32} \mid x_{<i+32})$
$p(x_{i+33} \mid x_{<i+33})$
$p(x_{i+34} \mid x_{<i+34})$
$p(x_{i+35} \mid x_{<i+35})$

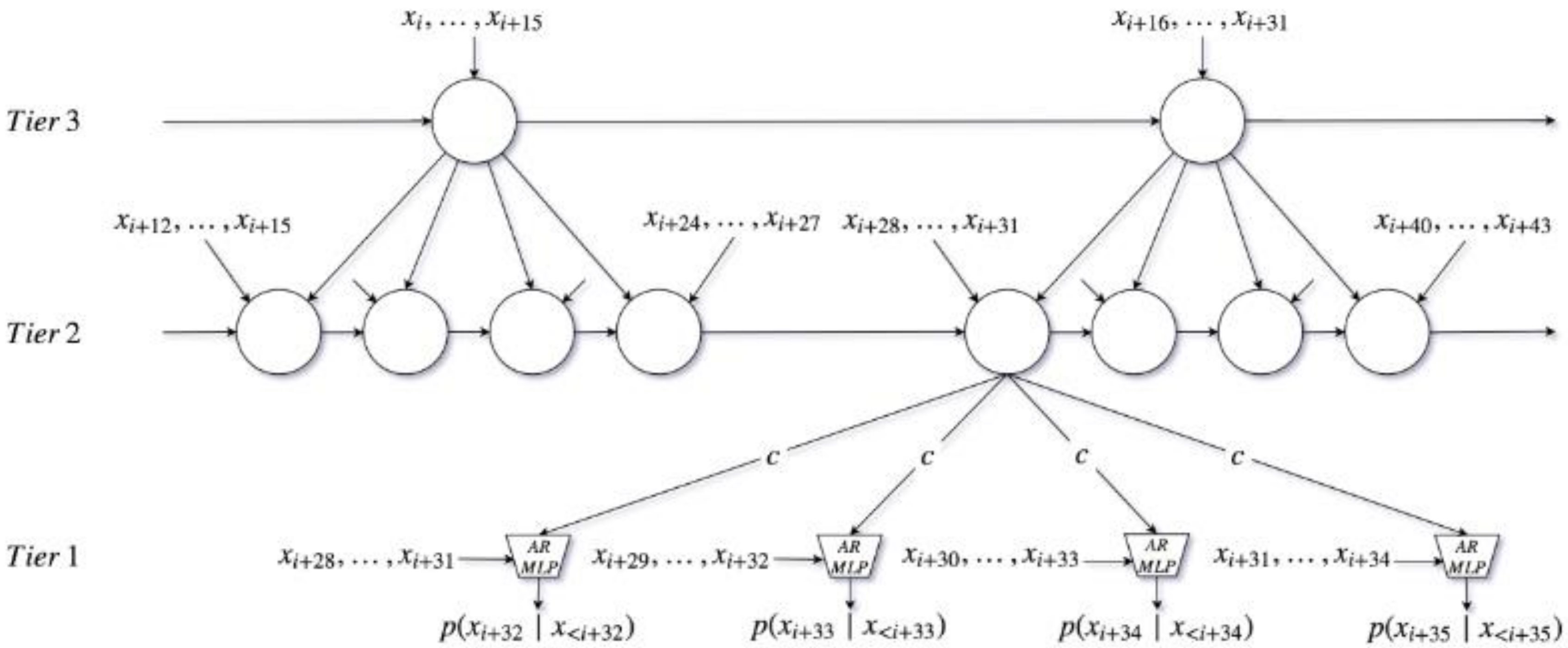
Figure 1: Snapshot of the unrolled model at timestep i with $K = 3$ tiers. As a simplification only one RNN and up-sampling ratio $r = 4$ is used for all tiers.

https://arxiv.org/abs/1612.07837

Frame-Level Modules

“Each frame-level module is a deep RNN which summarizes the history of its inputs into a conditioning vector for the next module downward.”

Each module takes as input its corresponding frame, as well as the conditioning vector of the layer above it.


$x_i, \ldots, x_{i+15}$
$x_{i+16}, \ldots, x_{i+31}$
Tier 3
$x_{i+12}, \ldots, x_{i+15}$
$x_{i+24}, \ldots, x_{i+27}$
$x_{i+28}, \ldots, x_{i+31}$
$x_{i+40}, \ldots, x_{i+43}$
Tier 2
c
c
c
c
Tier 1
$x_{i+28}, \ldots, x_{i+31}$
AR MLP
$x_{i+29}, \ldots, x_{i+32}$
AR MLP
$x_{i+30}, \ldots, x_{i+33}$
AR MLP
$x_{i+31}, \ldots, x_{i+34}$
AR MLP
$p(x_{i+32} \mid x_{<i+32})$
$p(x_{i+33} \mid x_{<i+33})$
$p(x_{i+34} \mid x_{<i+34})$
$p(x_{i+35} \mid x_{<i+35})$

Sample Level Modules

Conditioned on the conditioning vector from the frame above it, and on some number of preceding samples.

Since this number is generally small, they use a multilayer perceptron here instead of an RNN, to speed up training

“When processing an audio sequence, the MLP is convolved over the sequence, processing each window of samples and predicting the next sample.”

“At generation time, the MLP is run repeatedly to generate one sample at a time.”

Other aspects of SampleRNN

Linear Quantization with q=256

RNN’s (not used in WaveNet) are powerful if they can be trained efficiently

Truncated Back Propagation Through Time (BPTT)

- Split sequence into subsequences and only propagate gradient to beginning of subsequence
- Interestingly, able to train well on subsequences of only 32ms

Results

https://soundcloud.com/samplernn

DeepVoice: Real-time Neural TTS

Composed end-to-end TTS pipeline by Baidu

- Composed of five individual components
    - Segmentation model
    - Grapheme-to-phone conversion model
    - Phoneme duration prediction model
    - Fundamental frequency prediction model
    - Audio synthesis model
- Few hours of manual effort minus training time
- Real-time audio synthesis

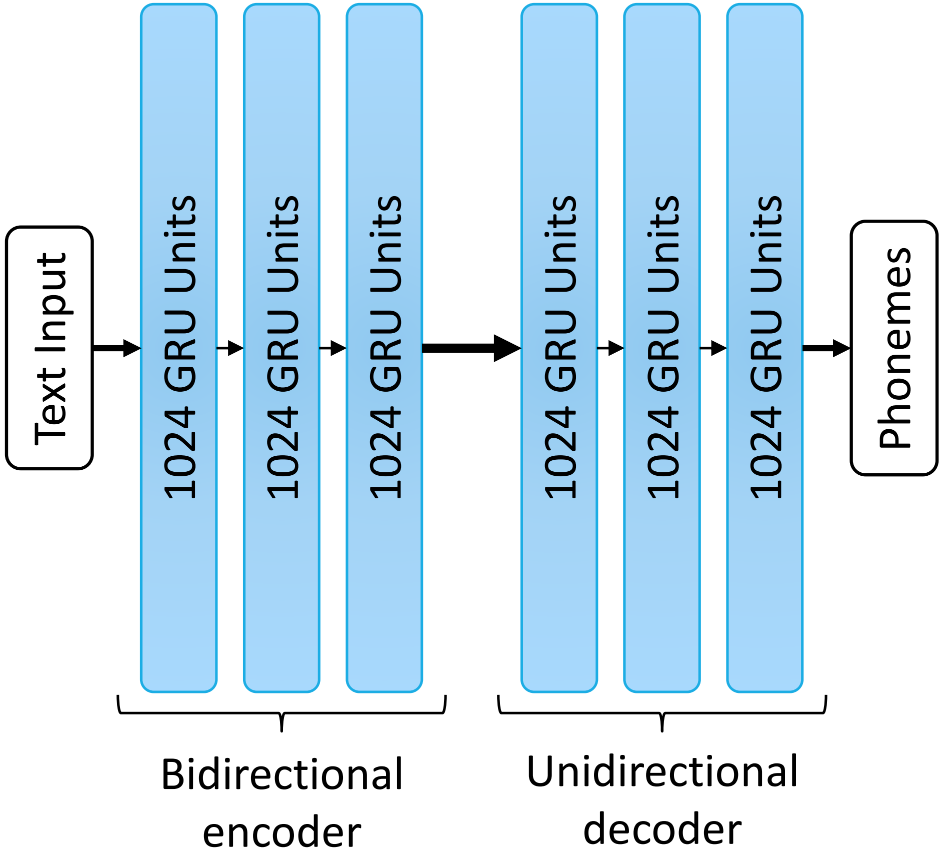
https://arxiv.org/abs/1702.07825

DeepVoice Grapheme-to-Phoneme Model

Encoder-decoder architecture from *“Sequence-to-Sequence Neural Net Models for Grapheme-to-Phoneme Conversion”*

Architecture

- Trained with teacher forcing
- Decode phonemes with beam search of width 5



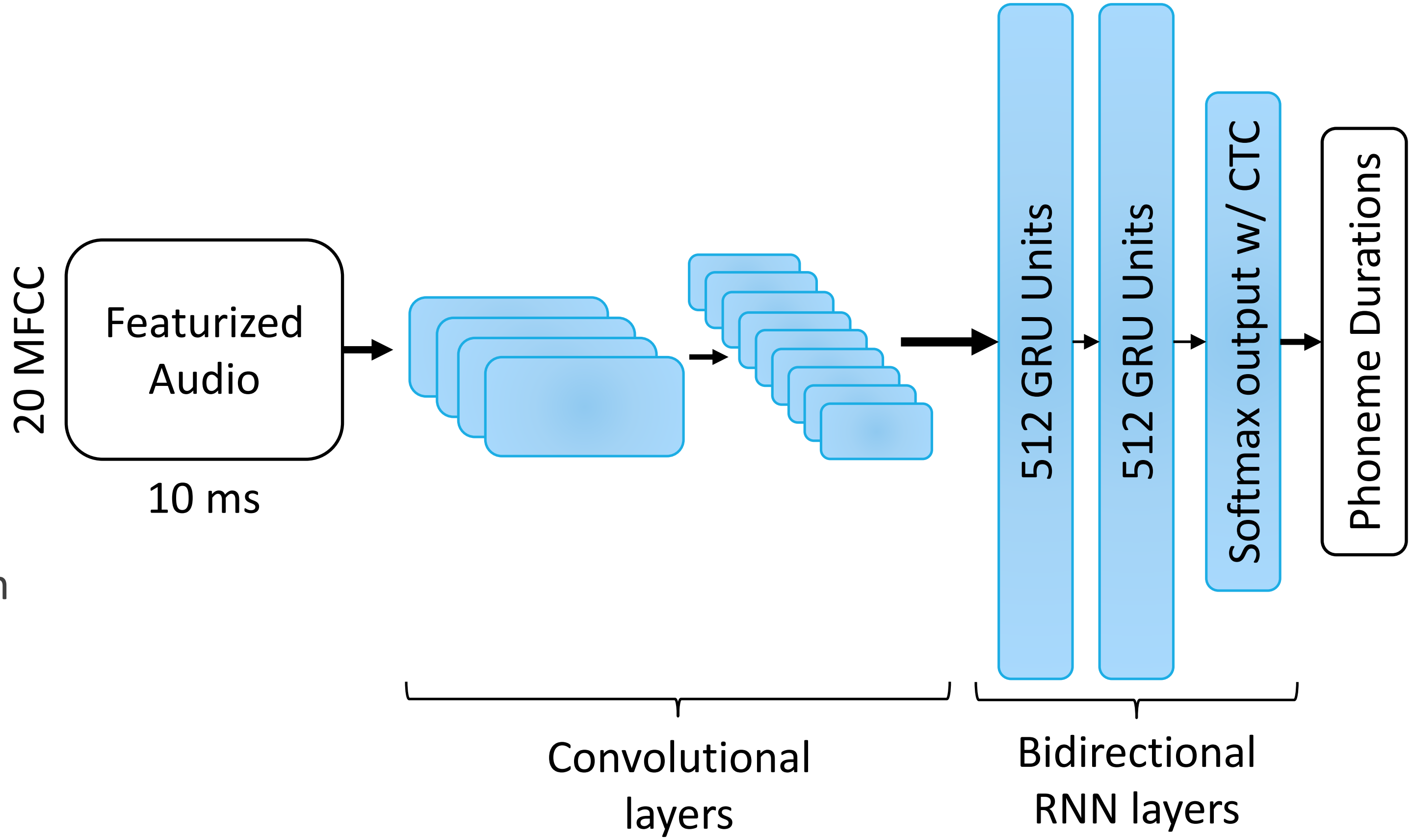
https://arxiv.org/abs/1702.07825

DeepVoice Segmentation Model

Convolutional recurrent neural network architecture from *"DeepSpeech 2: End-to-End Speech Recognition in English and Mandarin"*

Architecture

- 2D convolutions in time and frequency
- Softmax layer uses CTC to predict phoneme pairs
  - Output spikes from CTC close to phoneme boundaries
- Decode phoneme boundaries with beam search of width 50



https://arxiv.org/abs/1702.07825

DeepVoice Phoneme Duration and Fundamental Frequency Model

Outputs

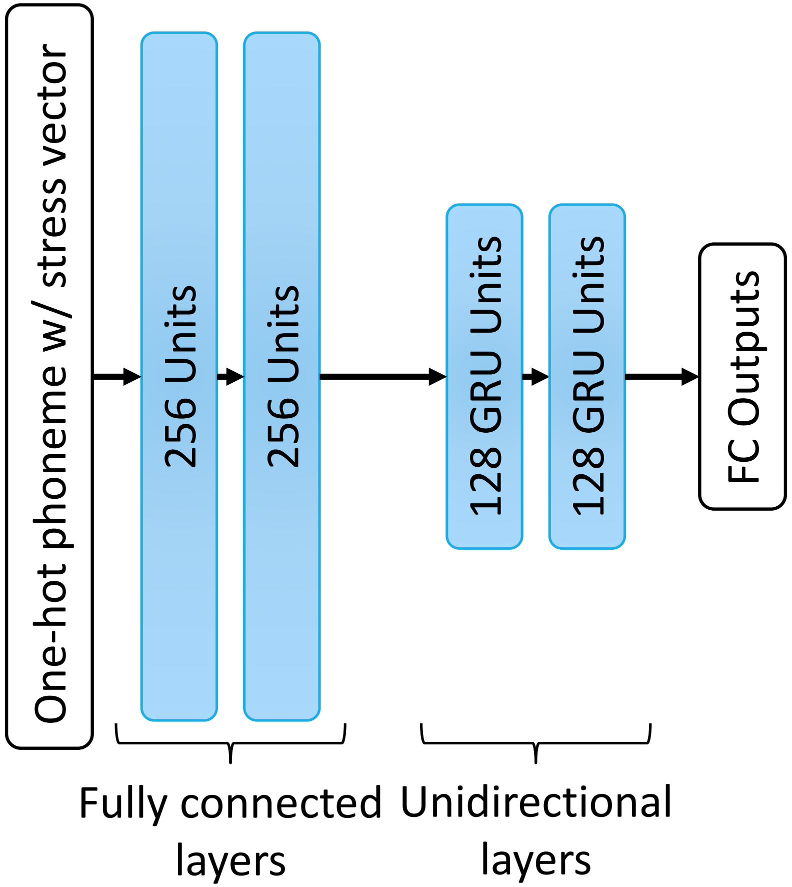
- Phoneme duration
- Probability phoneme is voiced (has F0)
- 20 time-dependent F0 values

Loss

$$L_n = |\hat{t}_n - t_n| + \lambda_1 \mathrm{CE}(\hat{p}_n, p_n) + \lambda_2 \sum_{t=0}^{T-1} |\widehat{F0}_{n,t} - F0_{n,t}|$$

$$+\lambda_3 \sum_{t=0}^{T-2} |\widehat{F0}_{n,t+1} - F0_{n,t}|$$

- λ_i are tradeoff constants
- $\hat{t}_n, t_n$ are durations of n^{th} phoneme
- $\hat{p}_n, p_n$ are probabilities n^{th} phoneme is voiced
- $\widehat{F0}_{n,t}, F0_{n,t}$ are fundamental frequency of n^{th} phoneme

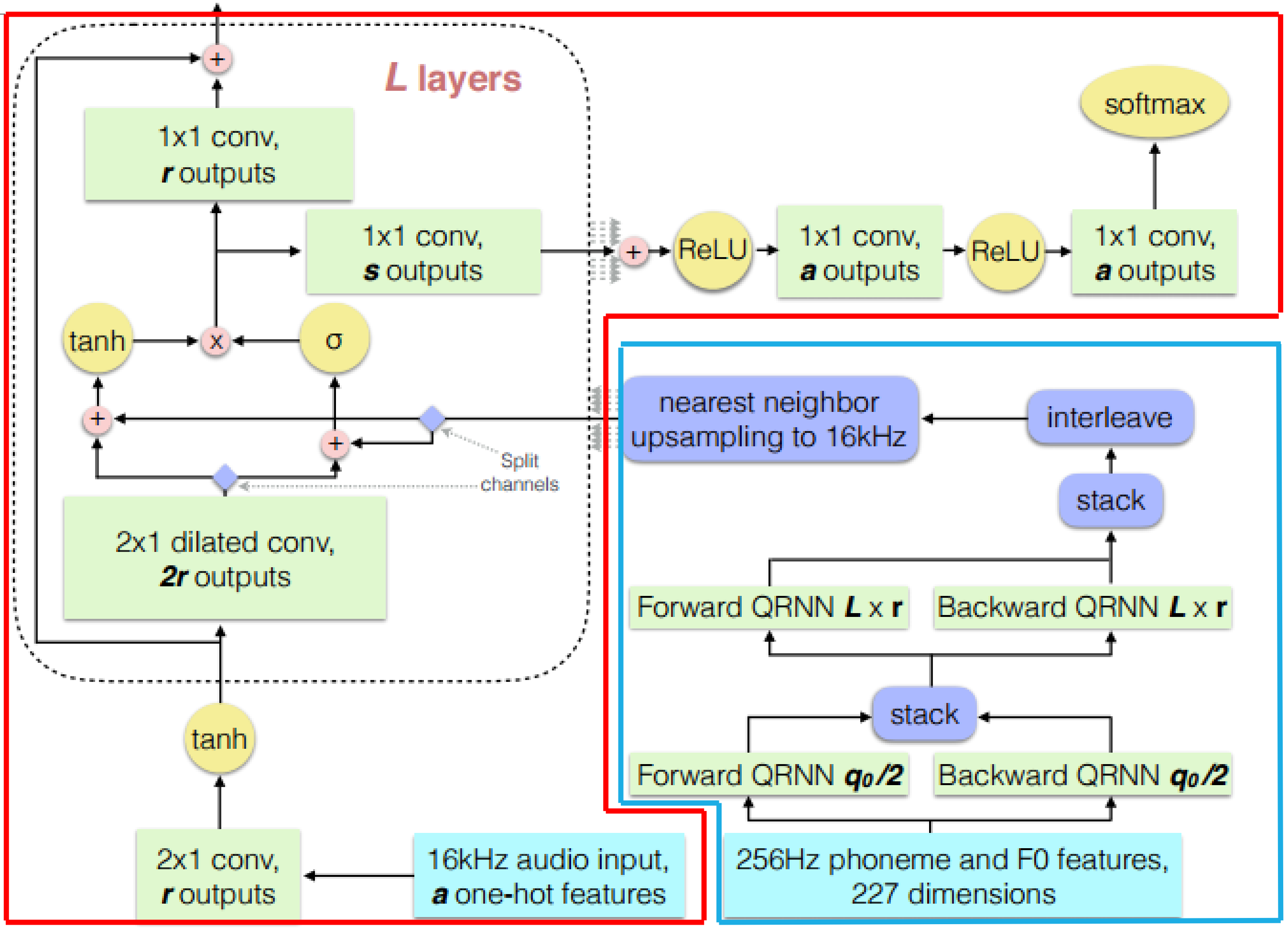


https://arxiv.org/abs/1702.07825

DeepVoice Audio Synthesis Model

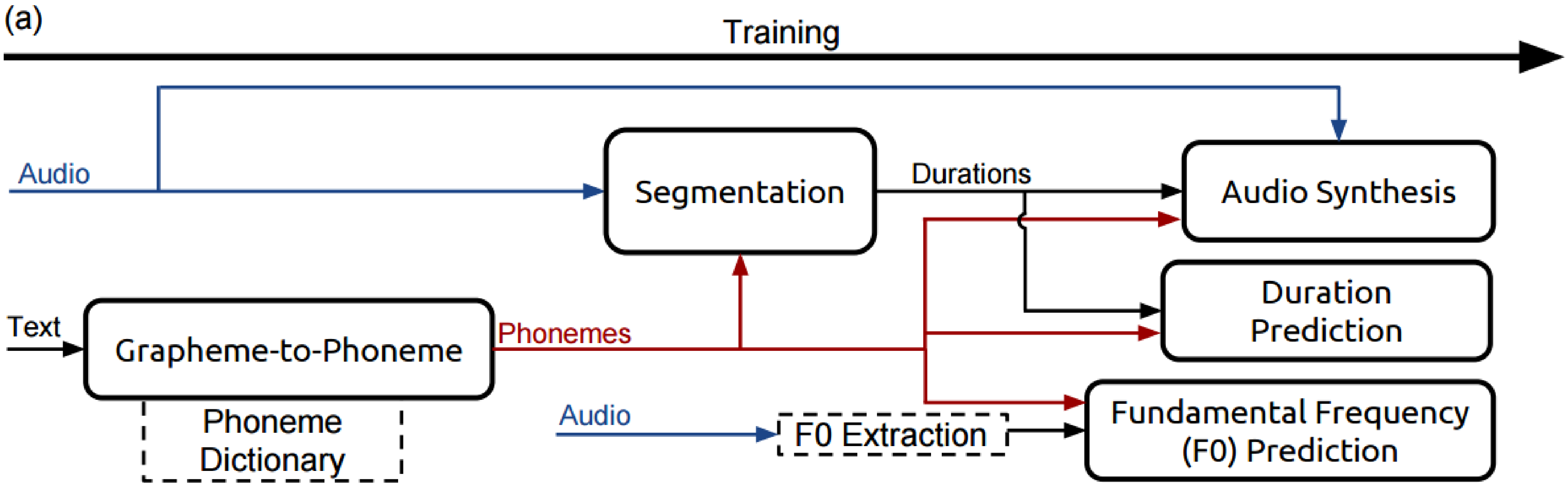
Architecture

- Red WaveNet
  - Same structure, different l, r, s values
- Blue Conditioning network
  - Two bidirectional QRNN layers
  - Interleave channels
  - Upsample to 16kHZ



https://arxiv.org/abs/1702.07825

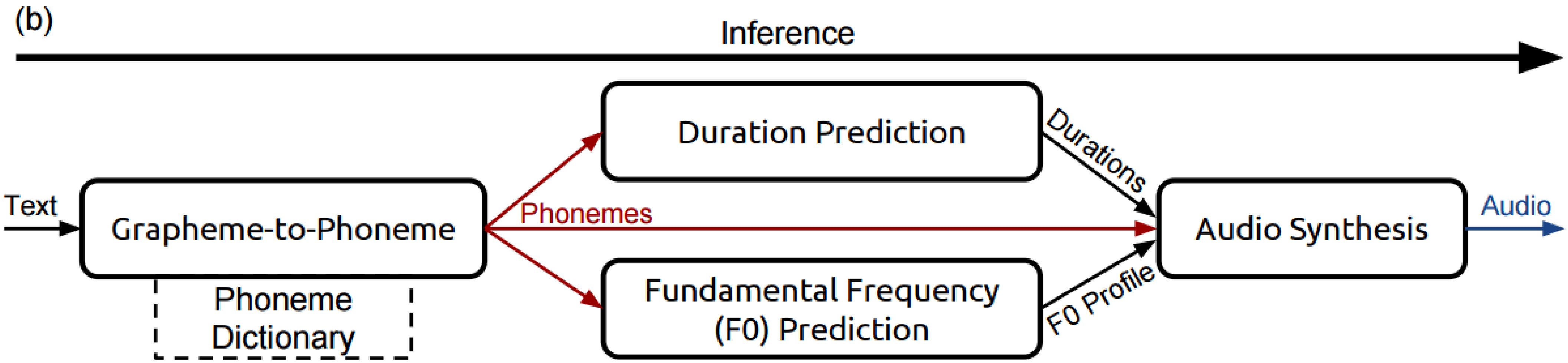
DeepVoice Training



Grapheme-to-Phoneme model used as backup to phoneme dictionary (CMUdict)

https://arxiv.org/abs/1702.07825

DeepVoice Inference



Segmentation model only annotates data in training and not used in inference

https://arxiv.org/abs/1702.07825

DeepVoice Results

Audio Synthesis Results

Type	Model Size	MOS ± CI
Ground Truth (48 kHz)	None	4.75 ± 0.12
Ground Truth (16 kHz)	None	4.45 ± 0.16
Synthesized (Audio only)	$l = 40, r = 64, s = 256$	3.94 ± 0.26
Synthesized (Synthesized Duration & F0)	$l = 40, r = 64, s = 256$	2.00 ± 0.23
Synthesized (2x real-time inference, audio only)	$l = 20, r = 32, s = 128$	2.74 ± 0.32
Synthesized (1x real-time inference, audio only)	$l = 20, r = 64, s = 128$	3.35 ± 0.31

Inference Results

Model	Platform	Data Type	# of Threads	Speed-up Over Real Time
$l = 20, r = 32, s = 128$	CPU	Float32	6	2.7
$l = 40, r = 64, s = 128$	CPU	Float32	6	1.11
$l = 20, r = 32, s = 128$	GPU	Float32	N/A	0.39

https://arxiv.org/abs/1702.07825

DeepVoice Implementation Details

CPU implementation

- Parallelizing work via multithreading
- Pinning threads to physical cores (or disabling hyperthreading)
- Replacing nonlinearities with high-accuracy approximations (only during inference)

$$\tanh(x) \approx \operatorname{sign}(x)\frac{\tilde{e}(x) - \frac{1}{\tilde{e}(x)}}{\tilde{e}(x) + \frac{1}{\tilde{e}(x)}} \text{ and } \sigma(x) \approx \begin{cases} \frac{\tilde{e}(x)}{1+\tilde{e}(x)} & x \geq 0 \\ \frac{1}{1+\tilde{e}(x)} & x \leq 0 \end{cases} \text{ where } e^{|x|} \approx \tilde{e}(x) = 1 + |x| + 0.5658x^2 + 0.143x^4$$

- Weight matrices quantized to *int16*
- Custom AVX assembly kernels for matrix-vector multiplication

GPU implementation

- Persistant RNNs to generate all samples in one kernel launch
- Split model across register file of SMs
- Round robin execution of kernels

https://arxiv.org/abs/1702.07825

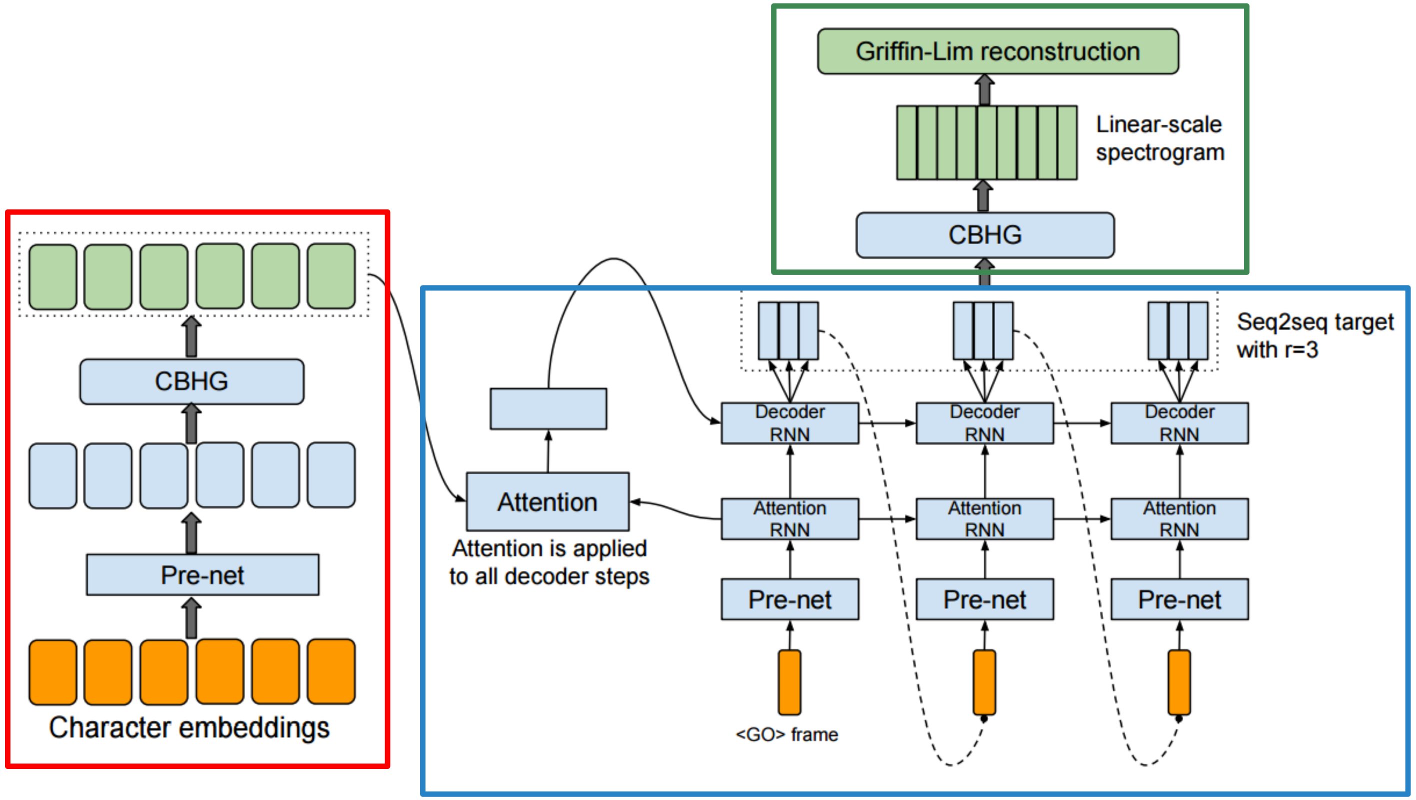
Tacotron: Towards End-to-End Speech Synthesis

Truly end-to-end TTS pipeline by Google

- Reduces feature engineering
- Allows conditioning on various attributes

Architecture

- One network based on the sequence-to-sequence with attention paradigm
- Red Encoder
- Blue Decoder
- Green Post-processing net



https://arxiv.org/abs/1703.10135

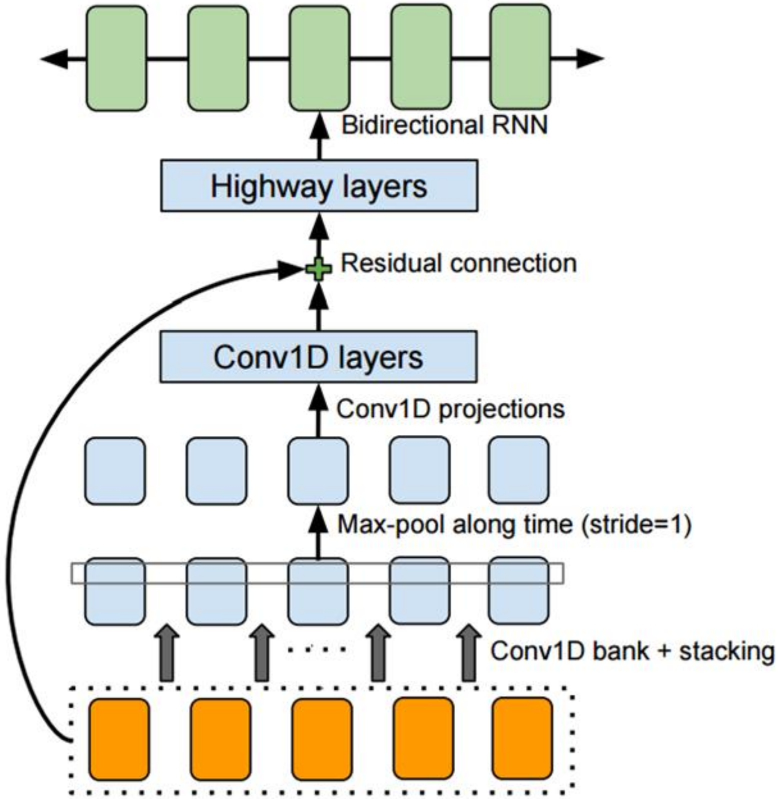
Tacotron CBHG Module

1D Convolutional Bank + highway network + bidirectional GRU (CBHG)

- Module for extracting representations from sequences
- Inspired by work from “Fully Character-Level Neural Machine Translation without Explicit Segmentation”

Architecture

- Bank of 1D convolutional filters
- Highway networks
  - Gating unit learn to regulate flow of information through network
- Bidirectional GRU RNN



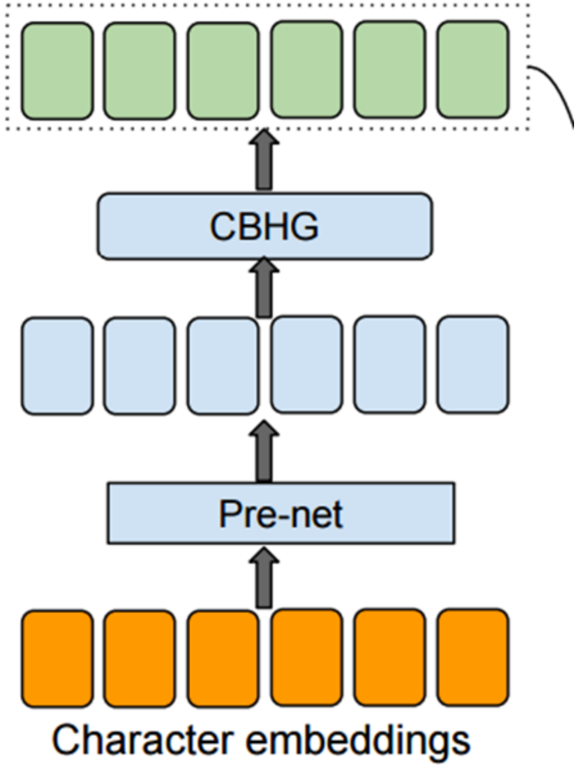
https://arxiv.org/abs/1703.10135

Tacotron Encoder

Extracts robust sequential representation of text

Architecture

- Input: one-hot vector of characters embedded into a continuous sequence
- Pre-Net, a set of on non-linear transformations
- CBHG module



https://arxiv.org/abs/1703.10135

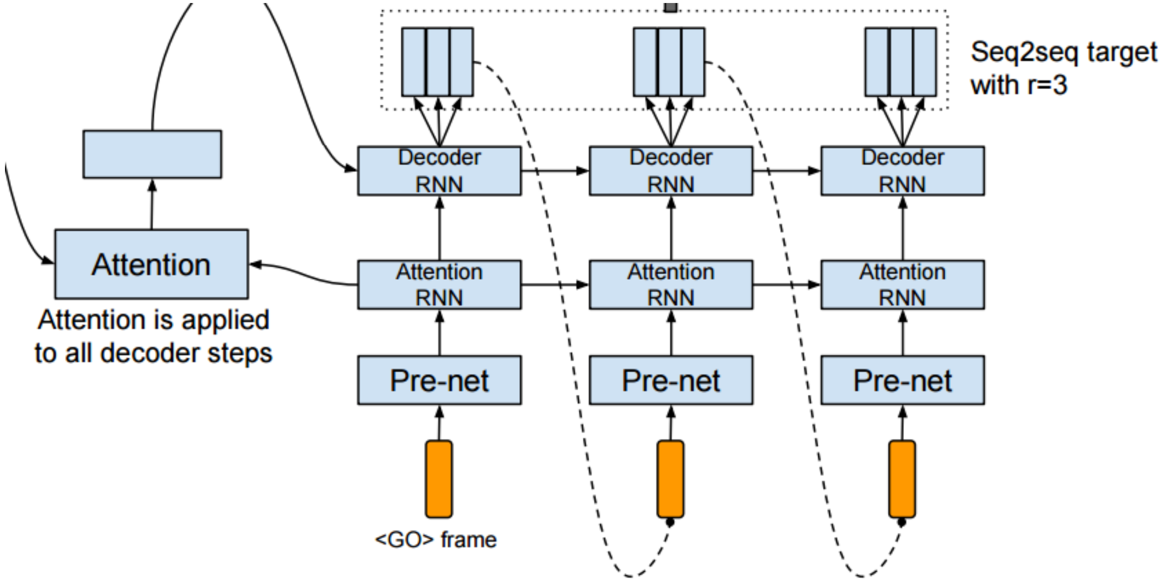
Tacotron Decoder

Target is 80-band mel-scale spectrogram rather than raw waveform

- Waveform is highly redundant representation

Architecture

- Attention RNN: 1 layer 256 GRU cells
- Decoder RNN: 2 layer residual 256 GRU cells
- Fully connected output layer



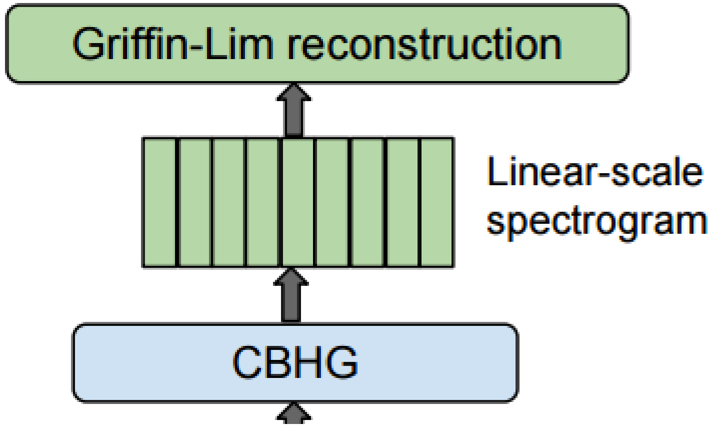
https://arxiv.org/abs/1703.10135

Tacotron Post-Processing Network

Convert sequence-to-sequence target to waveform though can be used to predict alternative targets

Architecture

- CBHG
- Griffin-Lim algorithm
  - Estimates waveform from spectrogram
  - Uses an iterative algorithm to decrease MSE
  - Used for simplicity



https://arxiv.org/abs/1703.10135

Tacotron Results

Tacotron results

	Model	MOS ± CI
1)	Tacotron	3.82 ± 0.085
	Parametric	3.69 ± 0.109
	Concatenative	4.09 ± 0.119

1)

The quick brown fox jumps over the lazy dog

Generative adversarial network or variational autoencoder

DeepVoice results

	Type	Model Size	MOS ± CI
	Ground Truth (48 kHz)	None	4.75 ± 0.12
	Ground Truth (16 kHz)	None	4.45 ± 0.16
2)	Synthesized (Audio only)	$l = 40, r = 64, s = 256$	3.94 ± 0.26
3)	Synthesized (Synthesized Duration & F0)	$l = 40, r = 64, s = 256$	2.00 ± 0.23
	Synthesized (2x real-time inference, audio only)	$l = 20, r = 32, s = 128$	2.74 ± 0.32
	Synthesized (1x real-time inference, audio only)	$l = 20, r = 64, s = 128$	3.35 ± 0.31

3)

Either way you should shoot very slowly

She set out to change the world and to change us

2)

She set out to change the world and to change us

https://arxiv.org/abs/1702.0782, https://arxiv.org/abs/1703.10135

Summary

Automatic Speech Recognition (ASR)

- Deep models with HMMs
- Connectionist Temporal Classification (CTC)
  - Method for labeling unsegmented data
- Attention based models

Text to Speech (TTS)

- WaveNet
  - Audio synthesis network with intermediate representation
- DeepVoice
  - Composed end-to-end learning
- Tacotron
  - Truly end-to-end learning

Thank you!
Questions?

Bonus: Music Generation

WaveNet, revisited

Train on 60 hours of Solo YouTube Piano music in one experiment

Train on MagnaTagATune dataset: 29 second clips, 188 tags. No published results unfortunately.

$$p\left(\mathbf{x} \mid \mathbf{h}\right) = \prod_{t=1}^{T} p\left(x_t \mid x_1, \ldots, x_{t-1}, \mathbf{h}\right)$$

$$p\left(\mathbf{x}\right) = \prod_{t=1}^{T} p\left(x_t \mid x_1, \ldots, x_{t-1}\right)$$

https://deepmind.com/blog/wavenet-generative-model-raw-audio/

“Deep Learning for Music"

Aims to create pleasant music with deep neural nets alone

Questions:

- “Is there a meaningful way to represent notes in music as a vector?” (like word2vec)
- “Can we build interesting generative neural network architectures that effectively express the
- notions of harmony and melody?”

Input

- Midi Data
- Piano Roll data (which notes are on?)

The Network

2 Layer LSTM

Map tokens (either MIDI messages or note combinations) to a learned vector respresentation

Feed output into softmax layer

Allows user to adjust hyperparamters including number of layers, hidden unit size, sequence length, batch size, and learning rate.

Input a “seed” sequence

Two approaches to generation:

- Choose maximum from softmax
- Choose anything from the softmax distribution

Results

Music trained solely on Bach was better than a mix of classical pianists

Music trained using the Piano Roll format was better than that of the MIDI messages

No published audio, unfortunately

References

Text to Speech (TTS)

- A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. arXiv preprint arXiv:1609.03499 (2016).
- S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio. SampleRNN: An Unconditional End-to-End Neural Audio Generation Model. arXiv preprint arXiv:1612.07837v2 (2017).
- J. Sotelo, S. Mehri, K. Kumar, J. Santos, K. Kastner, A. Courville, and Y. Bengio. Char2Wav: End-to-End Speech Synthesis. ICLR (2017).
- S. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi. Deep Voice: Real-time Neural Text-to-Speech. arXiv preprint arXiv:1702.07825v2 (2017).
- Y. Wang, and et al. Tacotron: A Fully End-to-End Text-to-Speech Synthesis Model. arXiv preprint arXiv:1703.10135v1 (2017).

References

Automatic Speech Recognition (ASR)

- A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. ICML (2006).
- G. Hinton, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition. Signal Processing Magazine (2012).
- A. Graves, A. Mohamed, and G. Hinton. Speech Recognition with Deep Recurrent Neural Networks. arXiv preprint arXiv:1303.5778v1 (2013).
- H. Sak, A. Senior, and F. Beaufays. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling.. rXiv preprint arXiv:1402.1128v1 (2014).
- O.Abdel-Hamid, et al. Convolutional Neural Networks for Speech Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing (2014).
- A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Ng. Deep Speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567v2 (2014).
- D. Amodei, et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. arXiv preprint arXiv:1512.02595v1 (2015).
- Y. Wang. Connectionist Temporal Classification: A Tutorial with Gritty Details. Github (2015).
- D. Bahdanau, et al. End-to-End Attention-based Large Vocabulary Speech Recognition. arXiv preprint arXiv:1508.04395 (2015).
- W. Xiong, et al. Achieving Human Parity in Conversational Speech Recognition. arXiv preprint arXiv:1610.05256 (2016).

References

Music Generation

- A. Huang, and R. Wu. Deep Learning for Music. arXiv preprint arXiv:1606.04930v1 (2016).
- V. Kaligeri, and S. Grandhe. Music Generation Using Deep Learning. arXiv:1612.04928v1 (2016).