Introduction to Speech Recognition

Jian Li

History

- 30s single-speaker
  small vocabulary
  isolated word recognition
- 70s larger vocabulary
  speaker-independent
  continuous speech recognition
- 80s using statistical modeling techniques like HMMs
- 90s commercially successful speech recognition technologies
- 2009 deep learning for speech recognition

Application

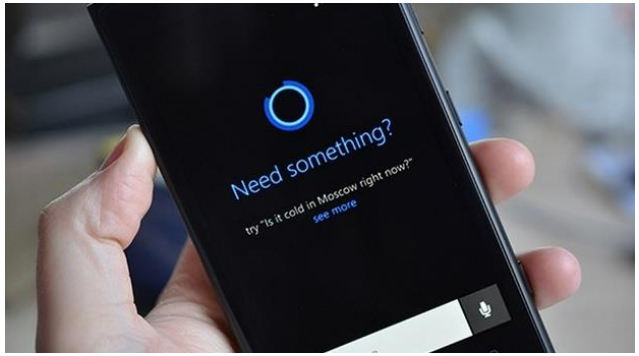
In-car systems

Usage in education and daily life

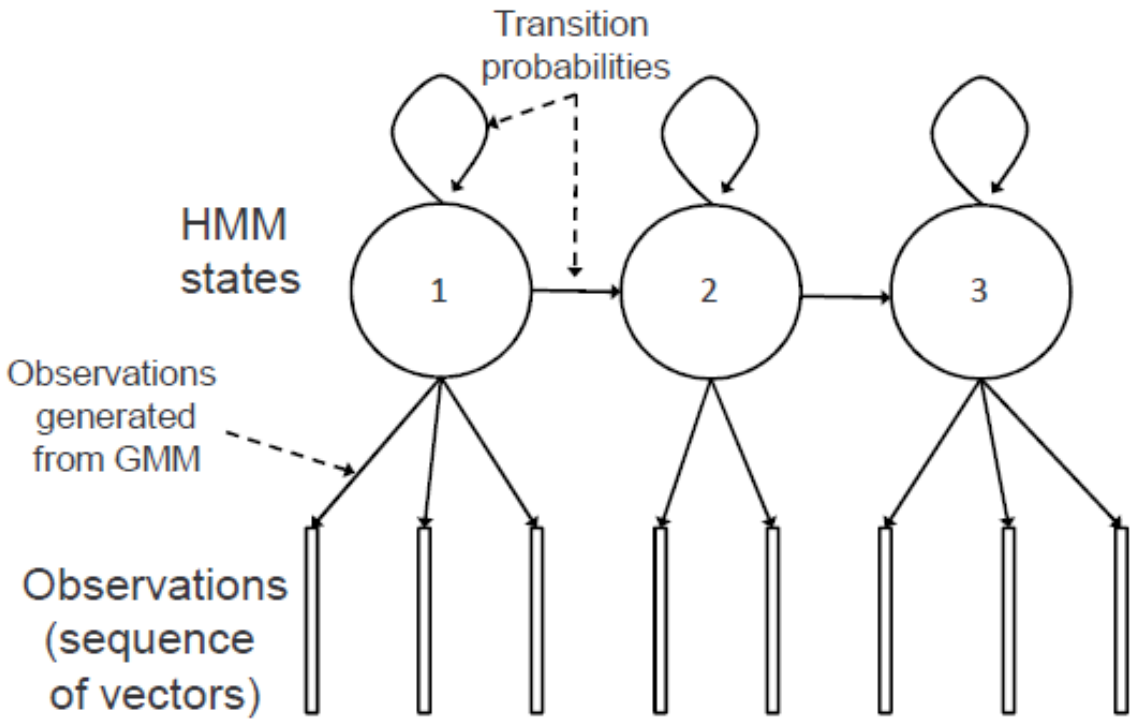
People with disabilities

……

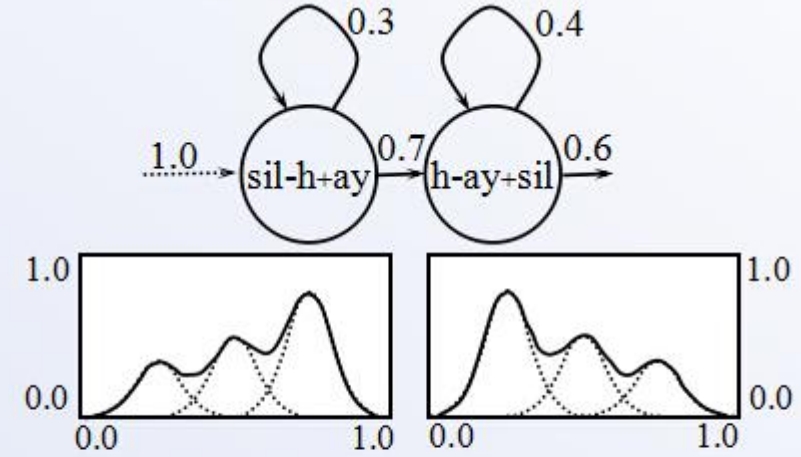
CMU Sphinx



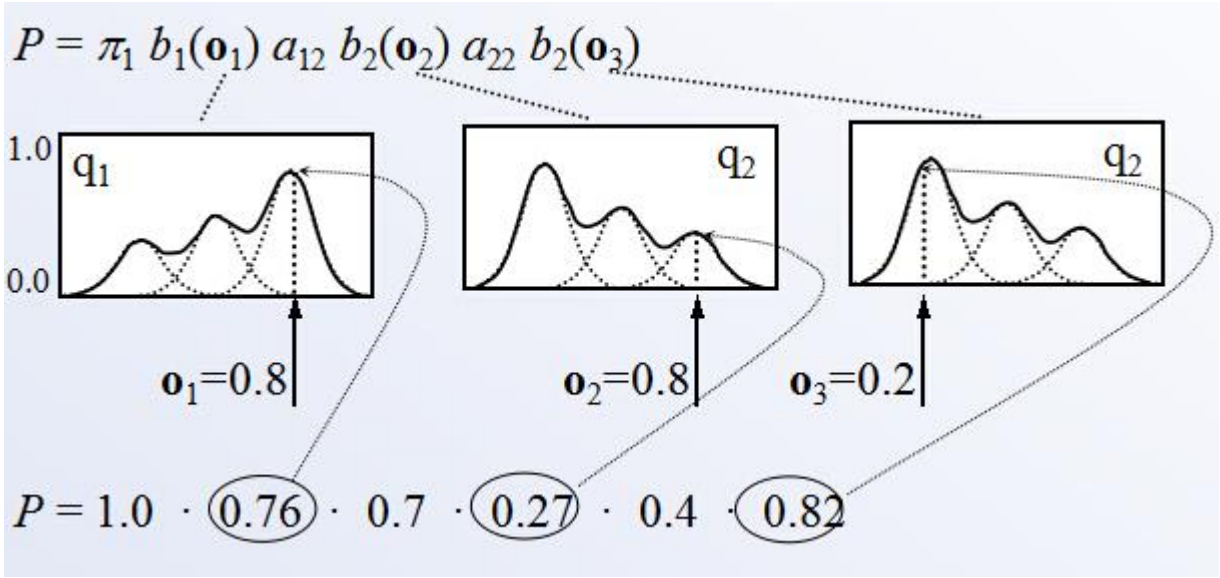
GMM-HMM



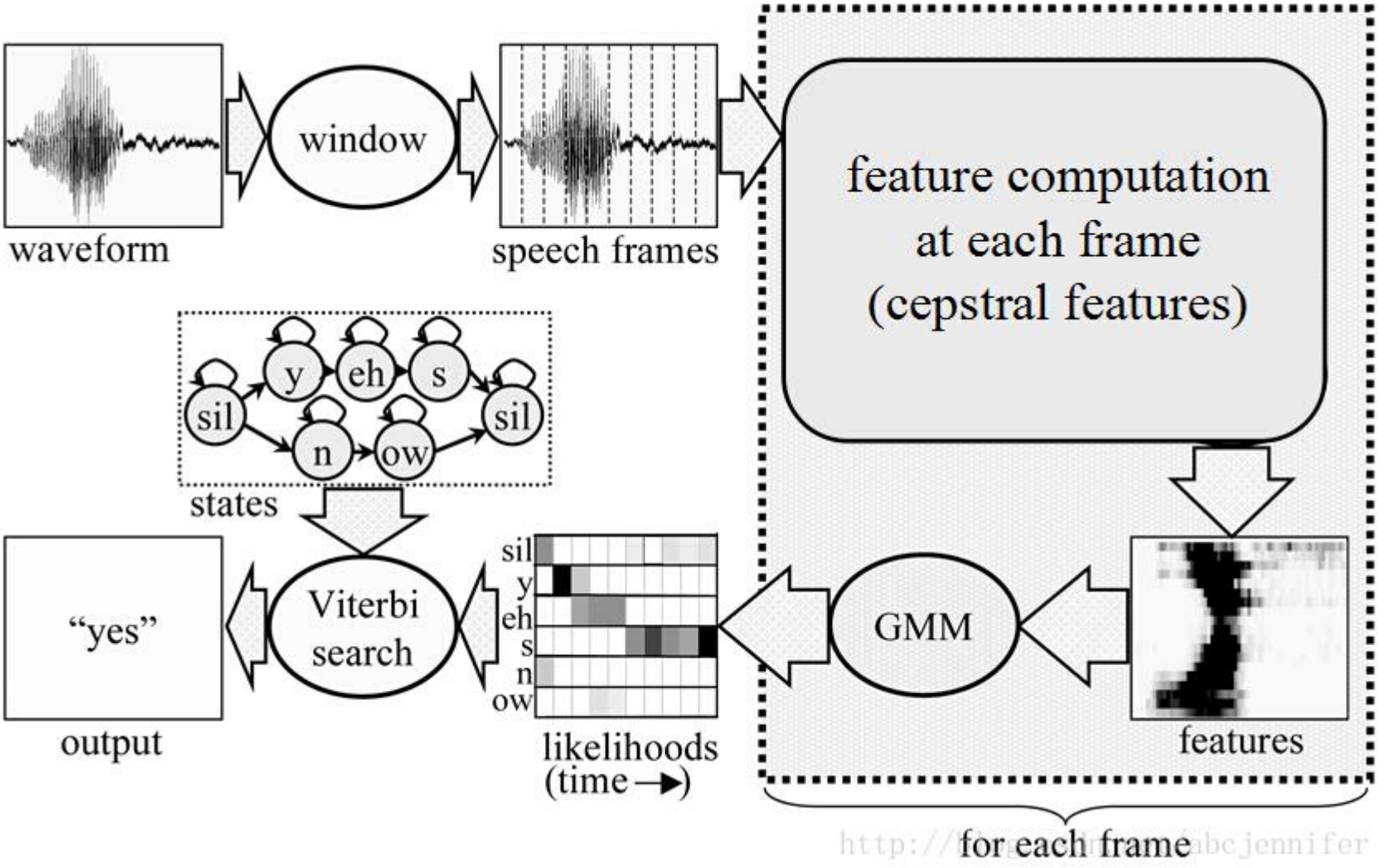
Ask ---->> ah + s + k



- observed features: $\mathbf{o}_1 = \{0.8\}$
  $\mathbf{o}_2 = \{0.8\}$
  $\mathbf{o}_3 = \{0.2\}$



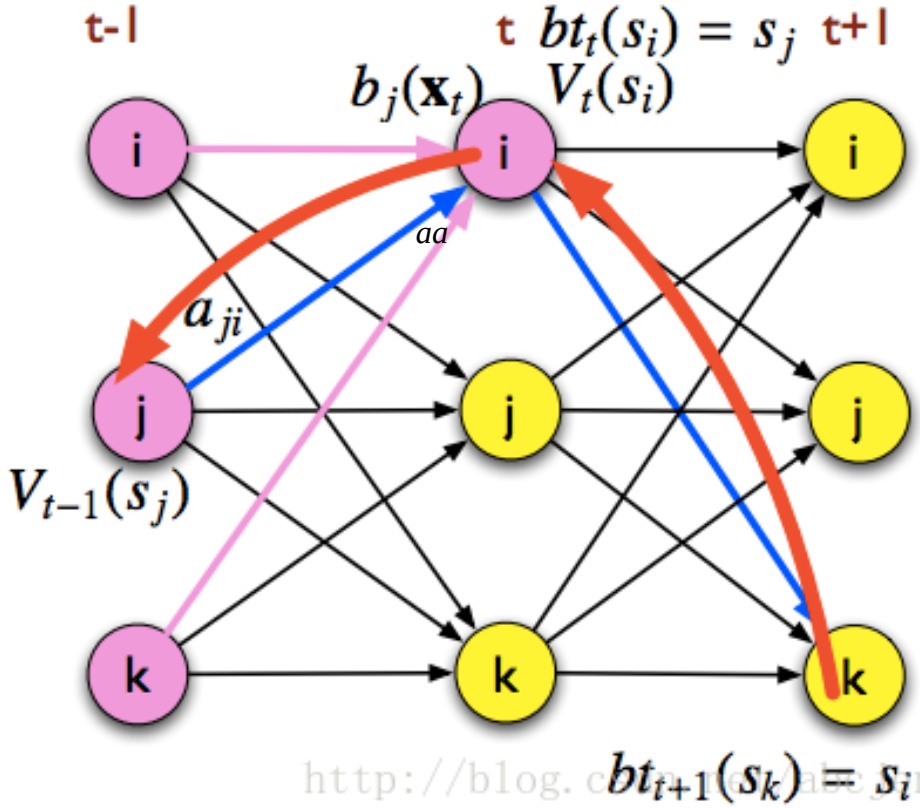
GMM-HMM


window
waveform
speech frames
feature computation
at each frame
(cepstral features)
sil
y
eh
s
sil
n
ow
states
"yes"
Viterbi
search
sil
y
eh
s
n
ow
likelihoods
(time →)
GMM
features
output
for each frame

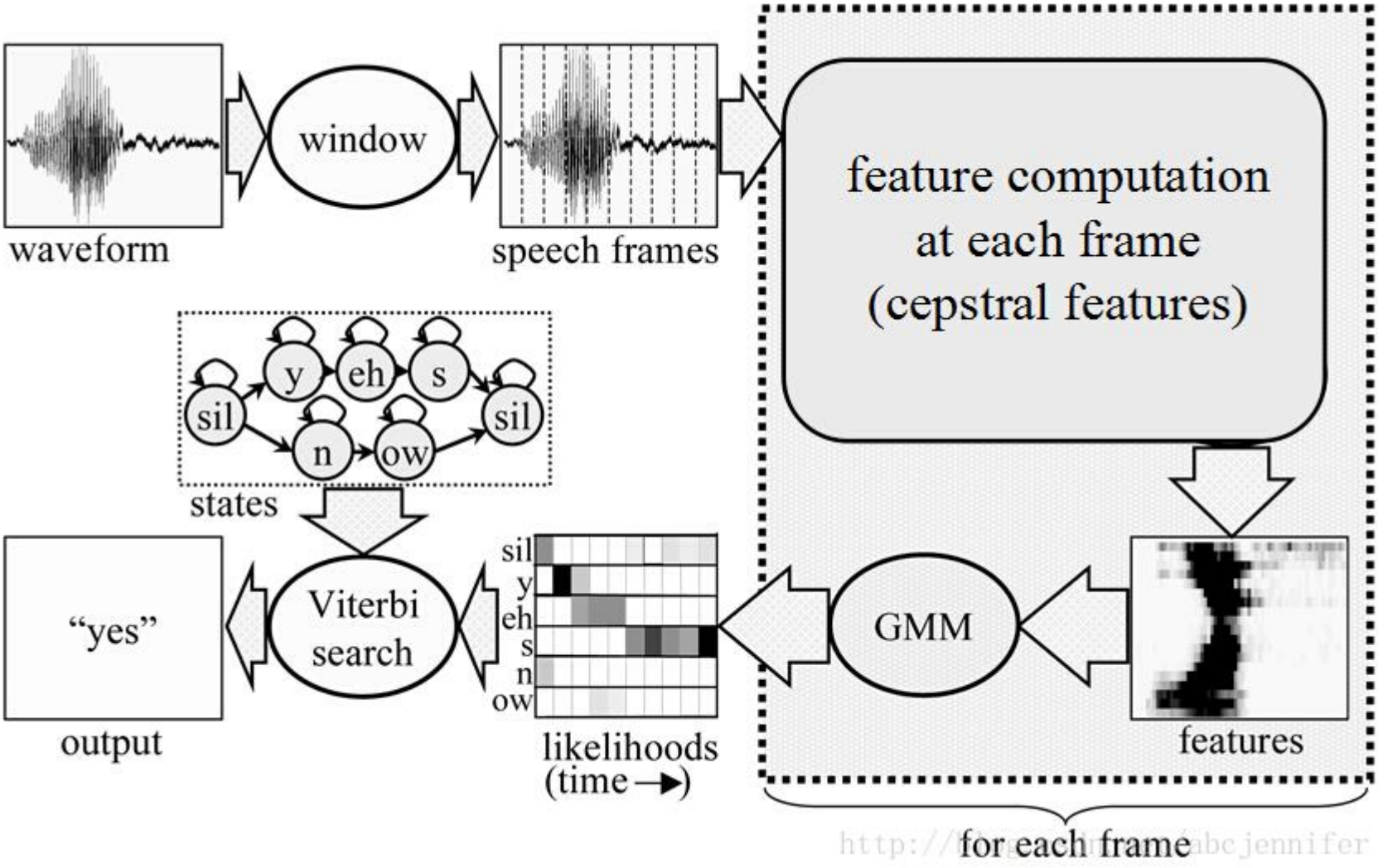
$$V_t(s_j) = \max_{i=1}^{N} V_{t-1}(s_i) a_{ij} b_j(\mathbf{x}_t)$$

$$bt_t(s_j) = \arg\max_{i=1}^{N} V_{t-1}(s_i) a_{ij} b_j(\mathbf{x}_t)$$

Backtrace to find the state sequence of the most probable path



GMM-HMM


window
waveform
speech frames
feature computation
at each frame
(cepstral features)
sil
y
eh
s
sil
n
ow
states
"yes"
Viterbi
search
sil
y
eh
s
n
ow
likelihoods
(time →)
GMM
features
output
for each frame

context-independent (**CI phone**) → context-dependent (**CD phones)**

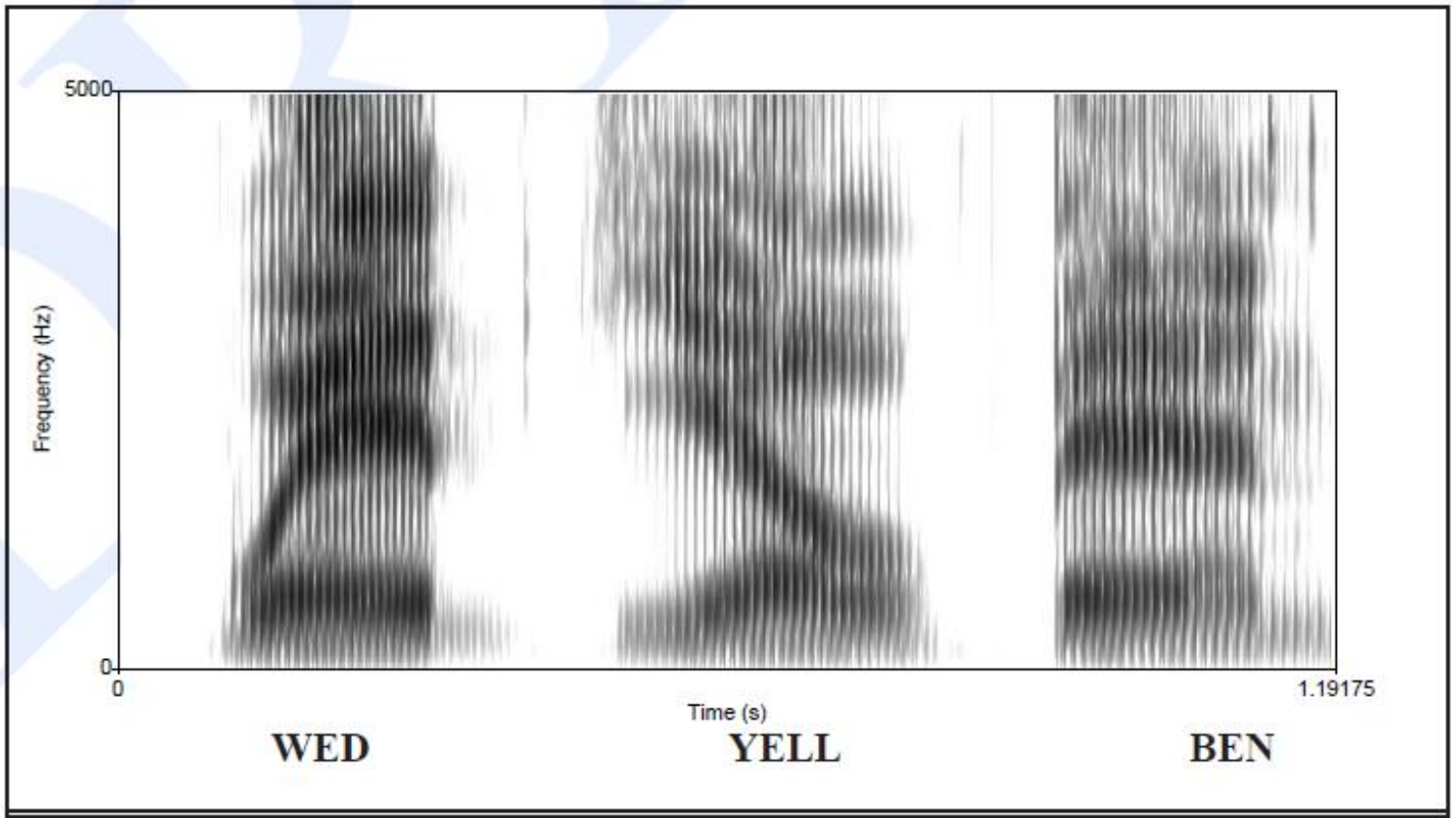
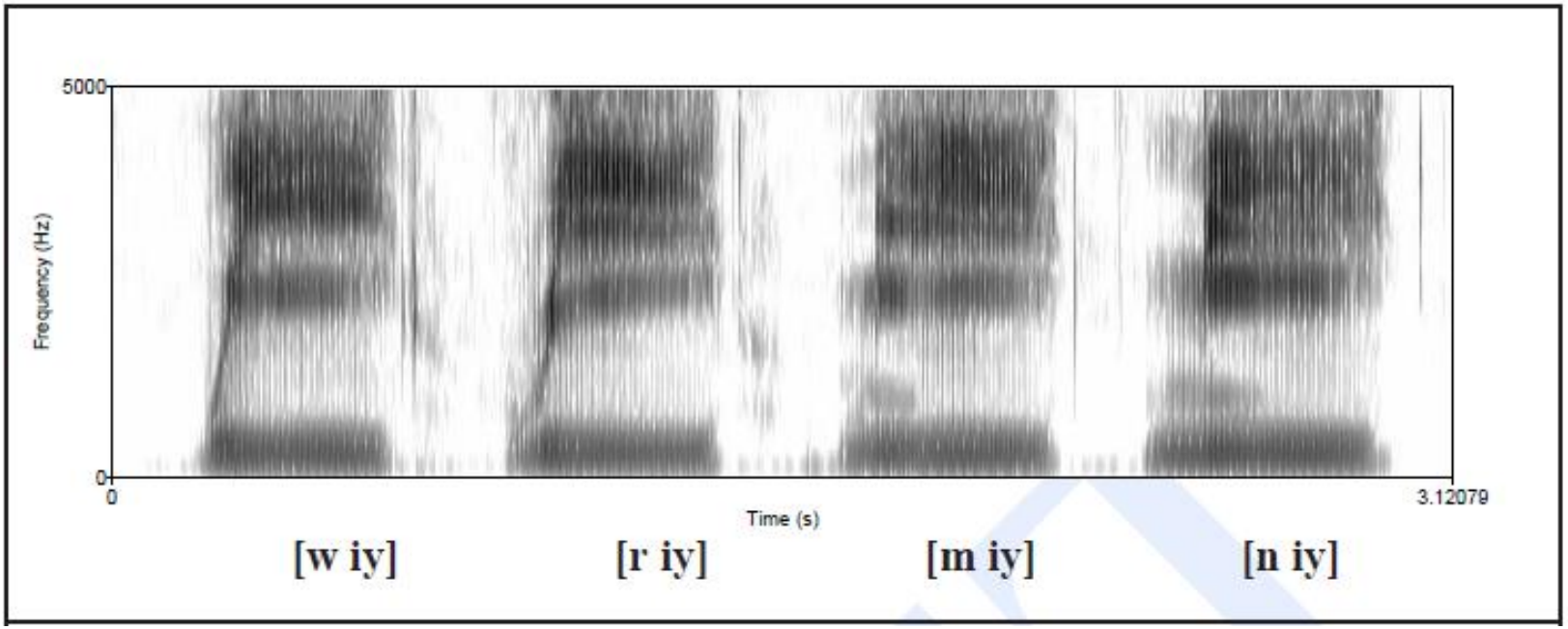
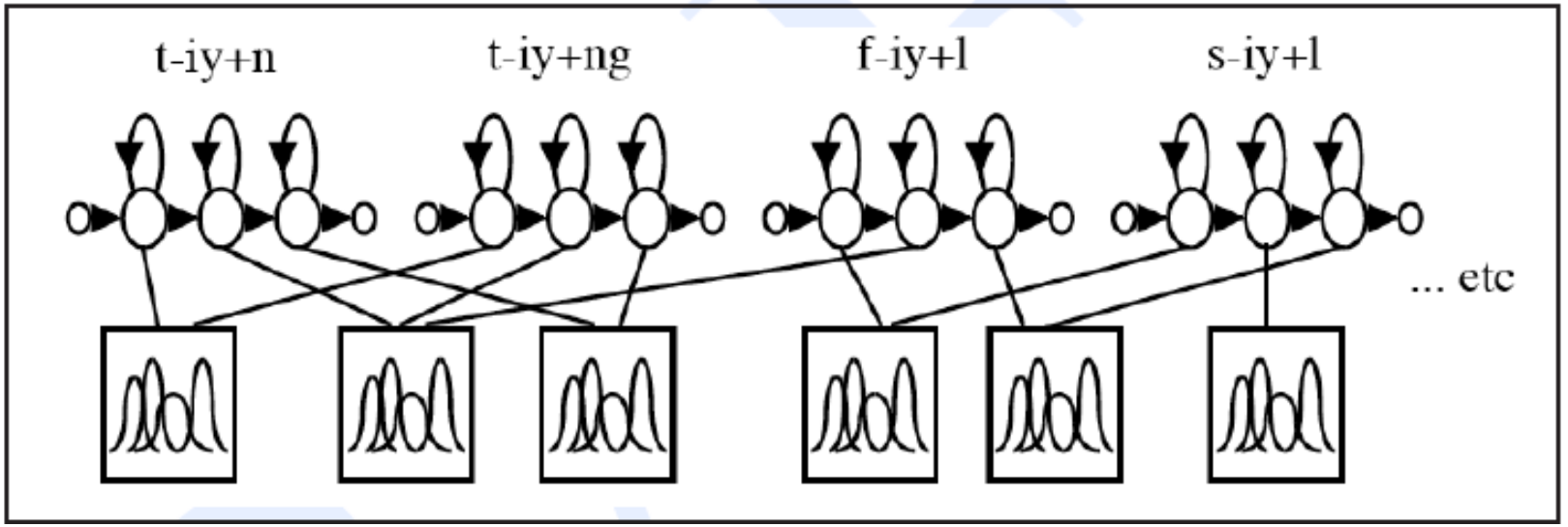
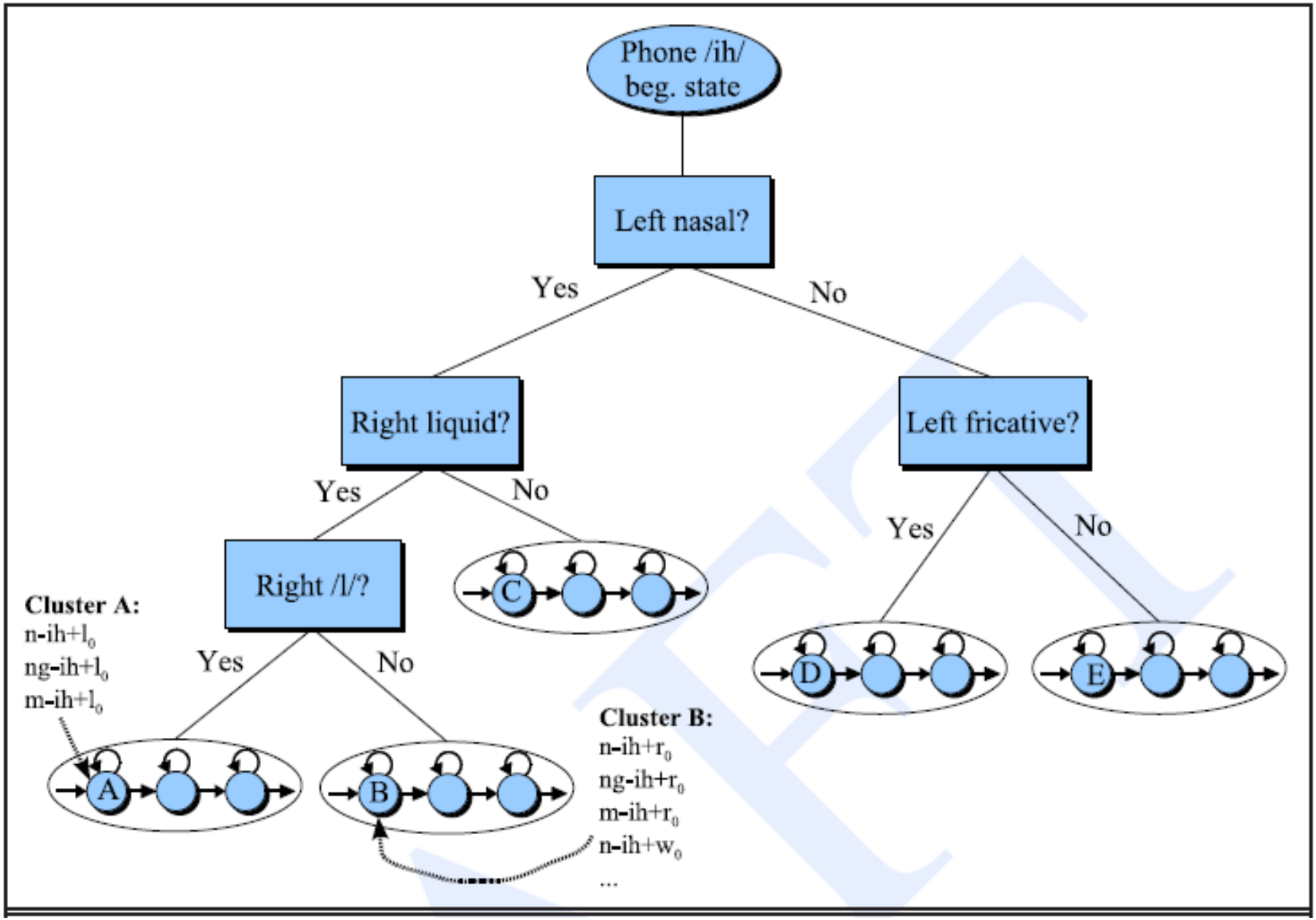


Figure 10.11 The vowel [eh] in three different triphone contexts, in the words *wed*, *yell*, and *Ben*. Notice the marked differences in the second formant (F2) at the beginning and end of the [eh] in all three cases.


5000
Frequency (Hz)
0
0
3.12079
Time (s)
[w iy]
[r iy]
[m iy]
[n iy]


t-iy+n
t-iy+ng
f-iy+l
s-iy+l
... etc



Decision tree for choosing which triphone states (subphones) to tie together.

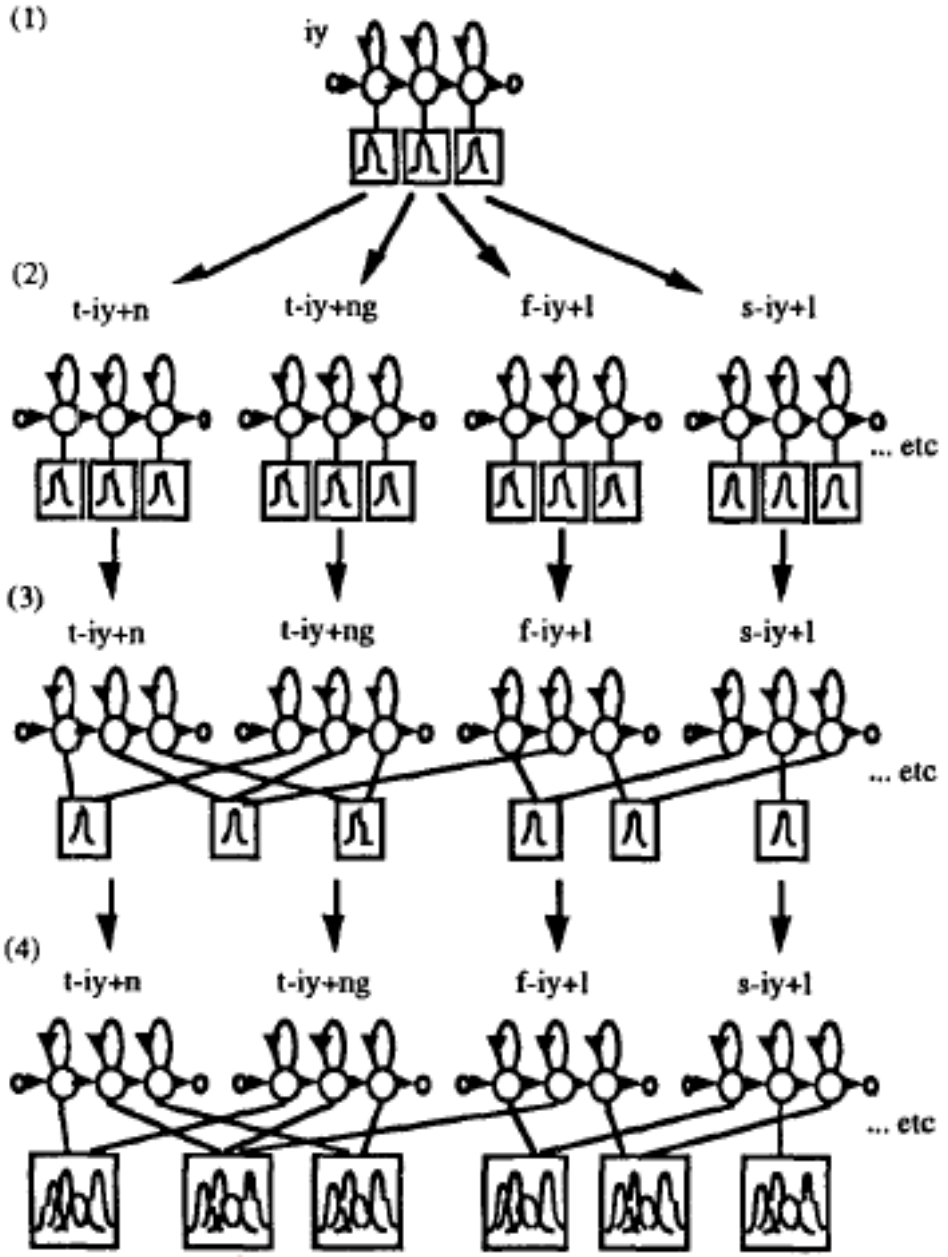
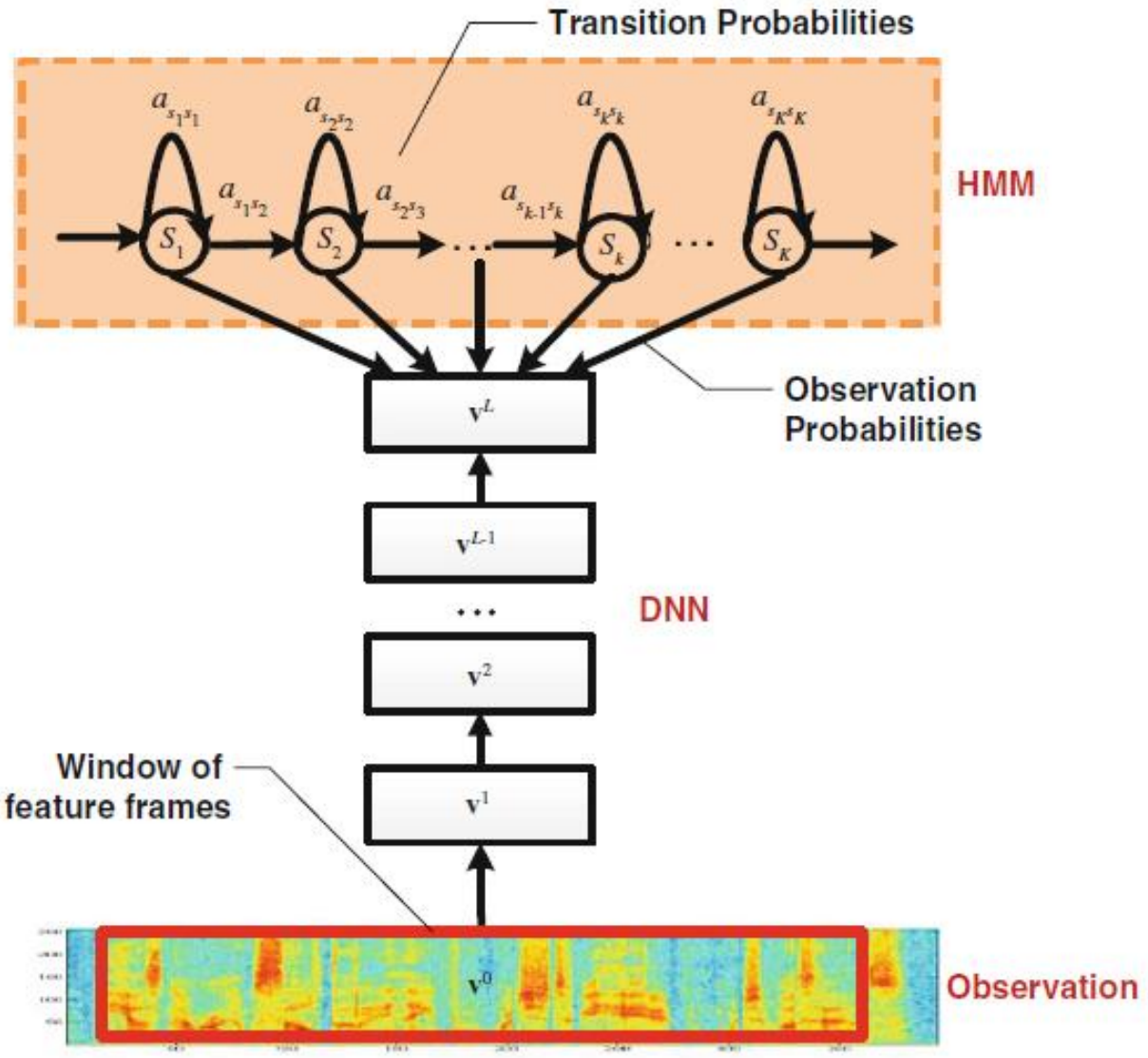
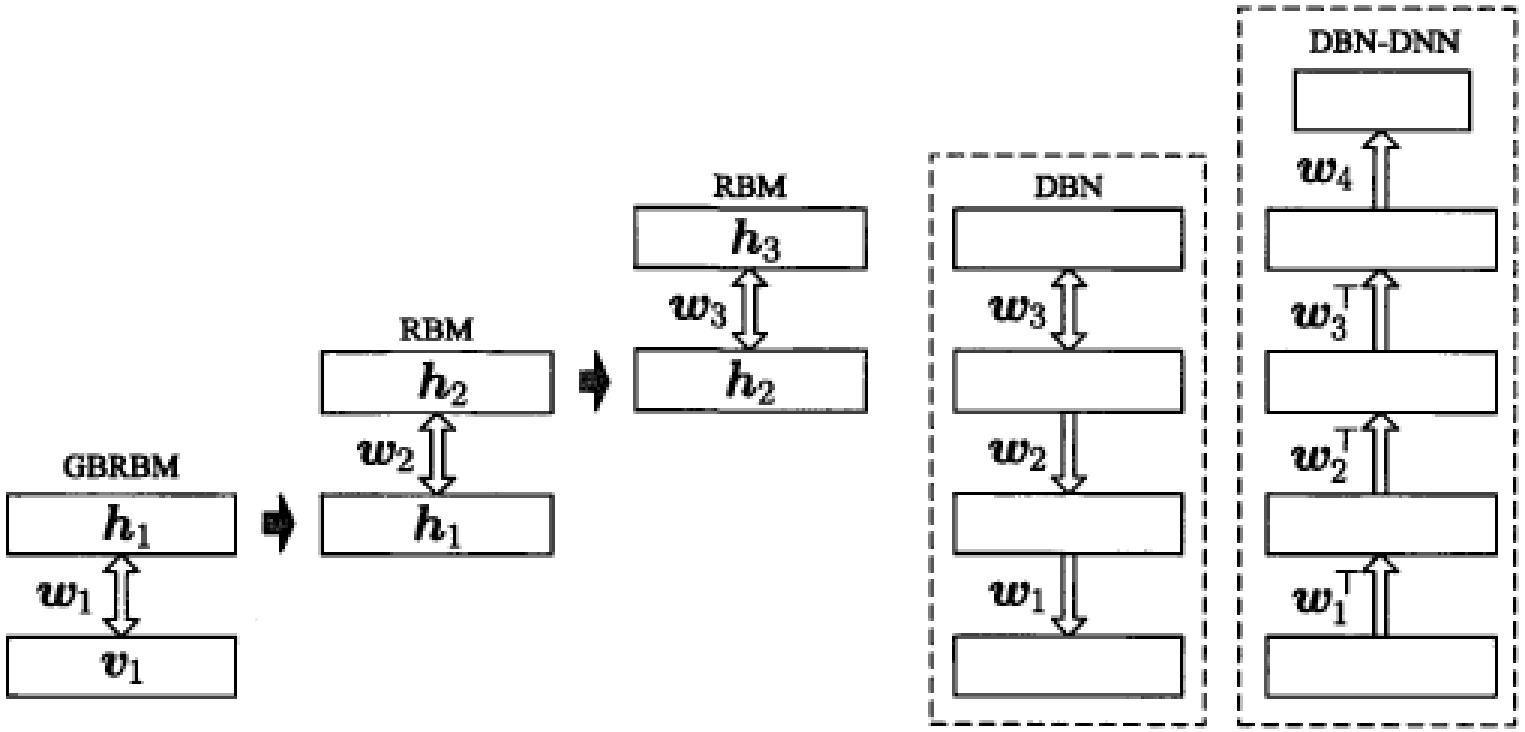


Figure 1: The Tied-State HMM System Build Procedure

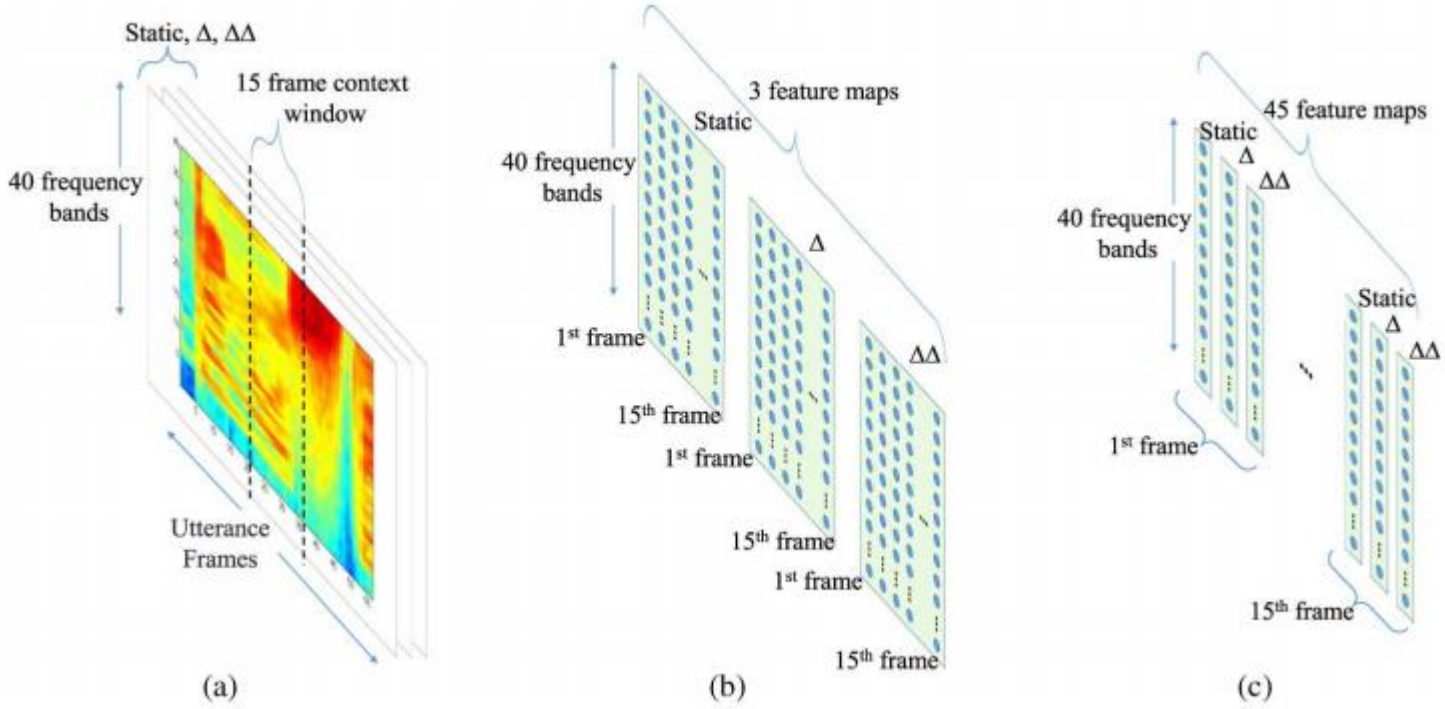
CD-DNN-HMM


Transition Probabilities
HMM
Observation Probabilities
DNN
Window of feature frames
Observation

CD-DNN-HMM


GBRBM
h_1
w_1
v_1
RBM
h_2
w_2
h_1
RBM
h_3
w_3
h_2
DBN
w_3
w_2
w_1
DBN-DNN
w_4
w_3^T
w_2^T
w_1^T

CNN-HMM


Static, Δ, ΔΔ
15 frame context window
40 frequency bands
Utterance Frames
(a)
3 feature maps
Static
Δ
ΔΔ
40 frequency bands
1st frame
15th frame
1st frame
15th frame
1st frame
15th frame
(b)
45 feature maps
Static
Δ
ΔΔ
40 frequency bands
Static
Δ
ΔΔ
1st frame
15th frame
(c)

CNN-HMM

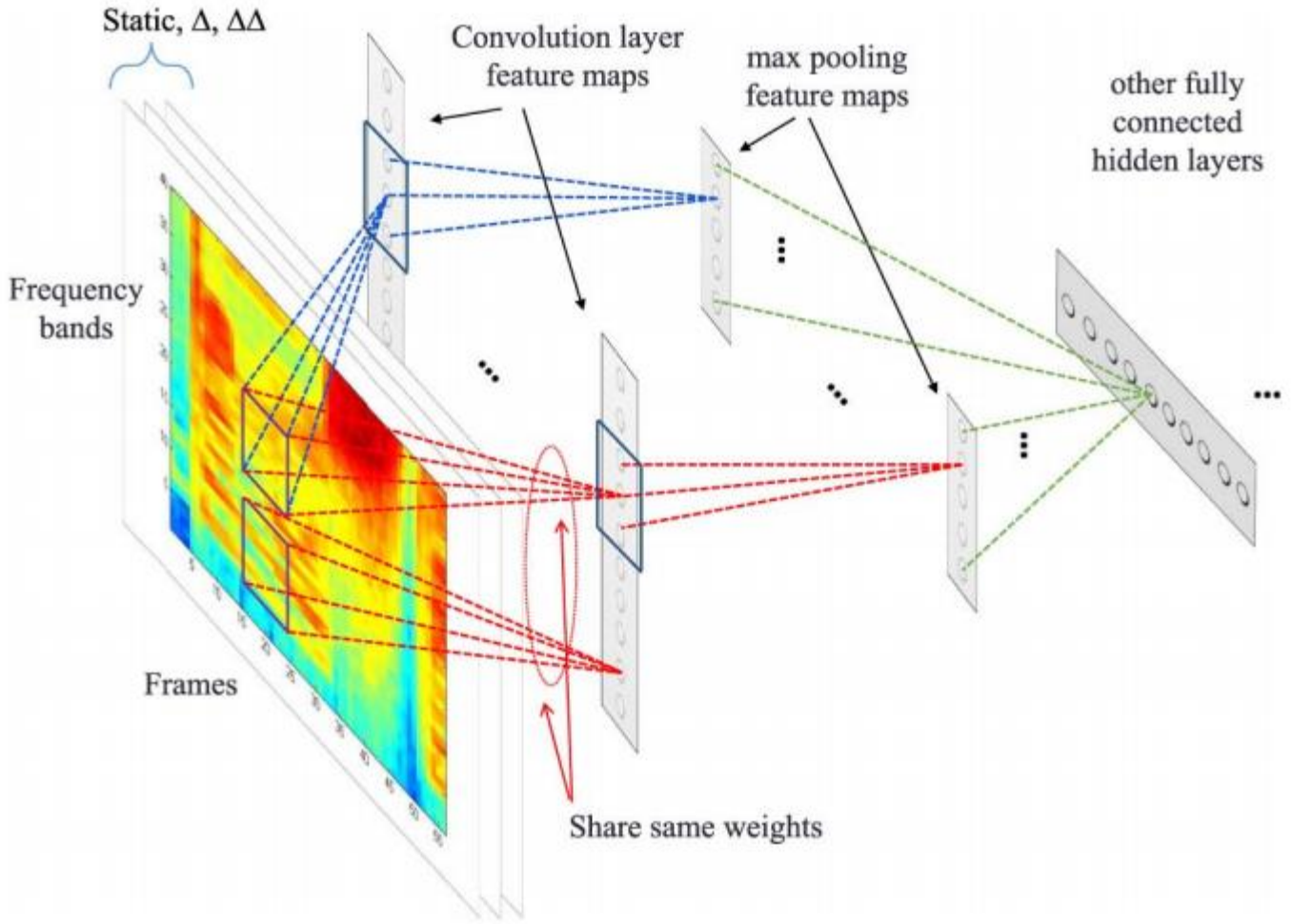
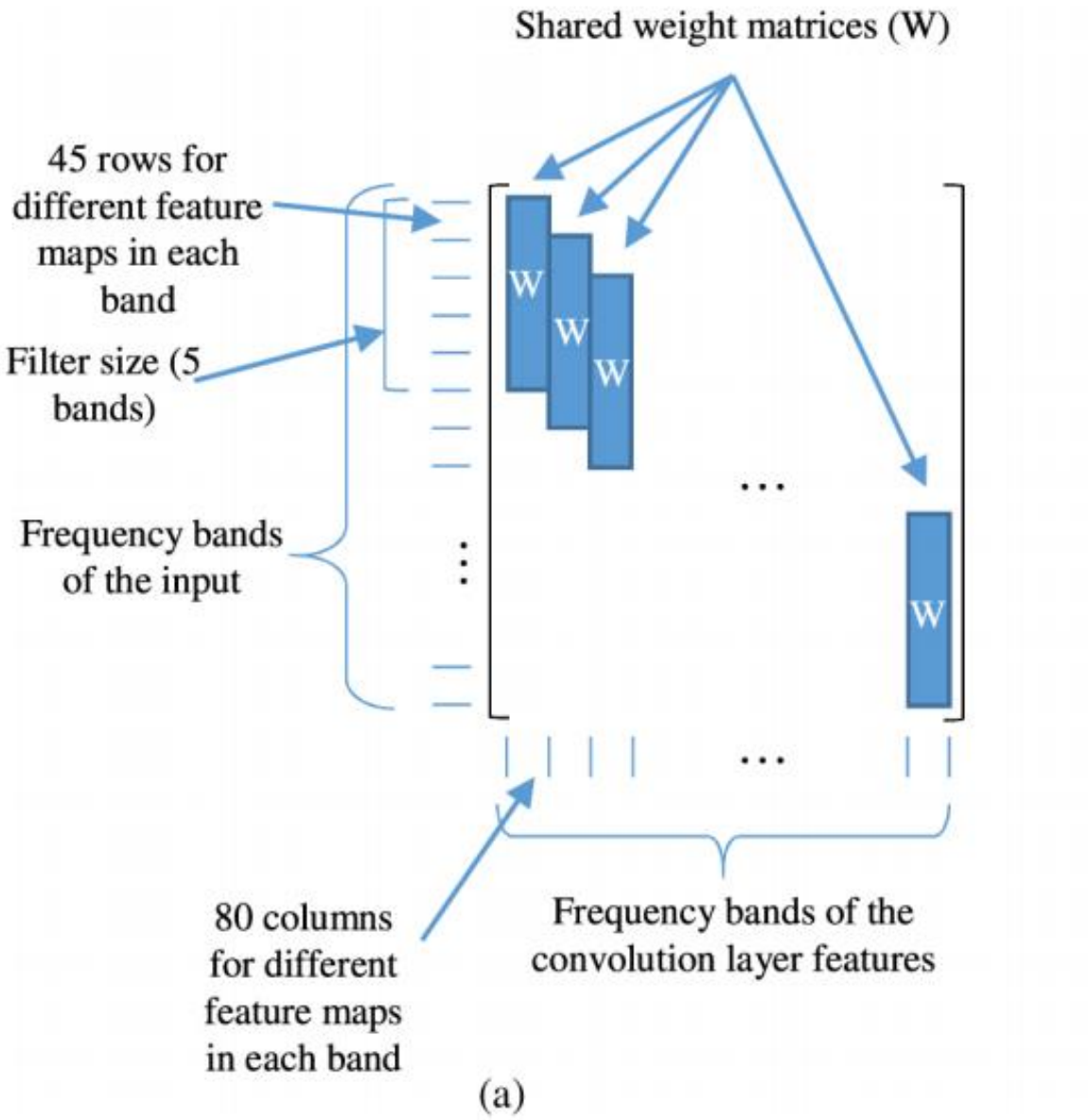


Fig. 3. An illustration of the regular CNN that uses so-called full weight sharing. Here, a 1-D convolution is applied along frequency bands.

CNN-HMM


Shared weight matrices (W)
45 rows for different feature maps in each band
Filter size (5 bands)
Frequency bands of the input
W
W
W
W
80 columns for different feature maps in each band
Frequency bands of the convolution layer features
(a)

LWS-CNN-HMM

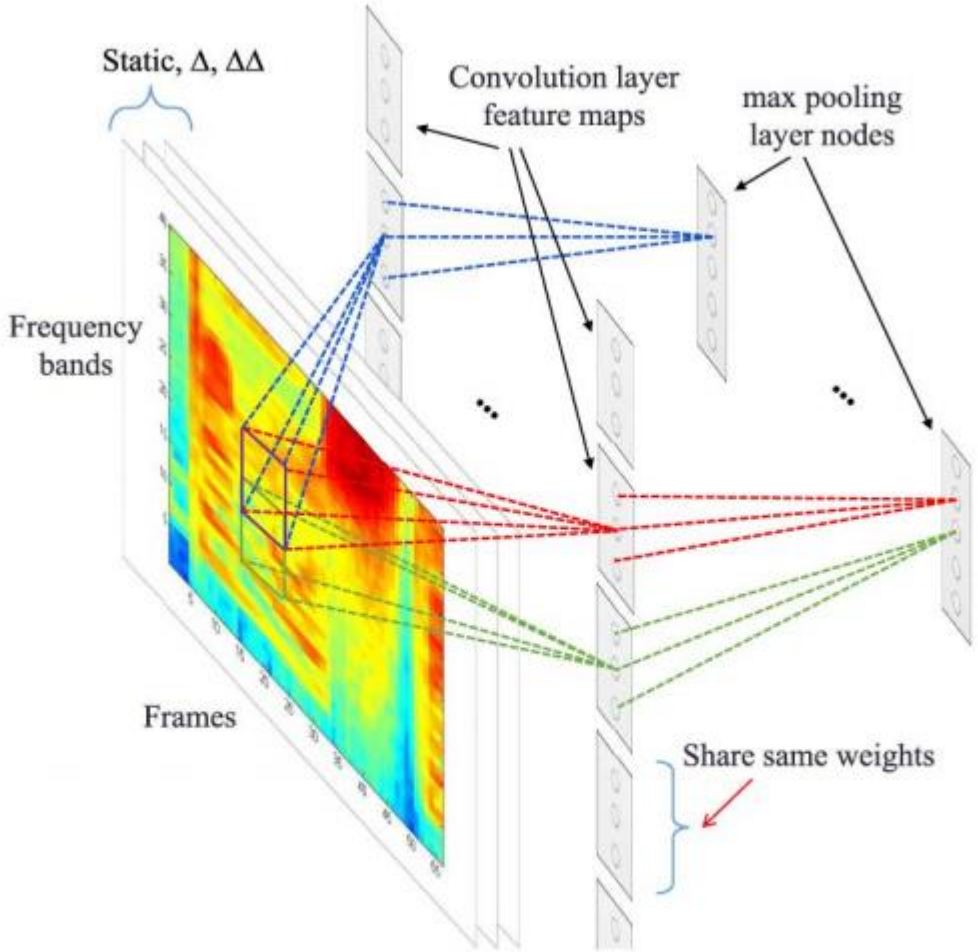


Fig. 5. An illustration of a CNN with limited weight sharing. 1-D convolution is applied along the frequency bands.

LWS-CNN-HMM

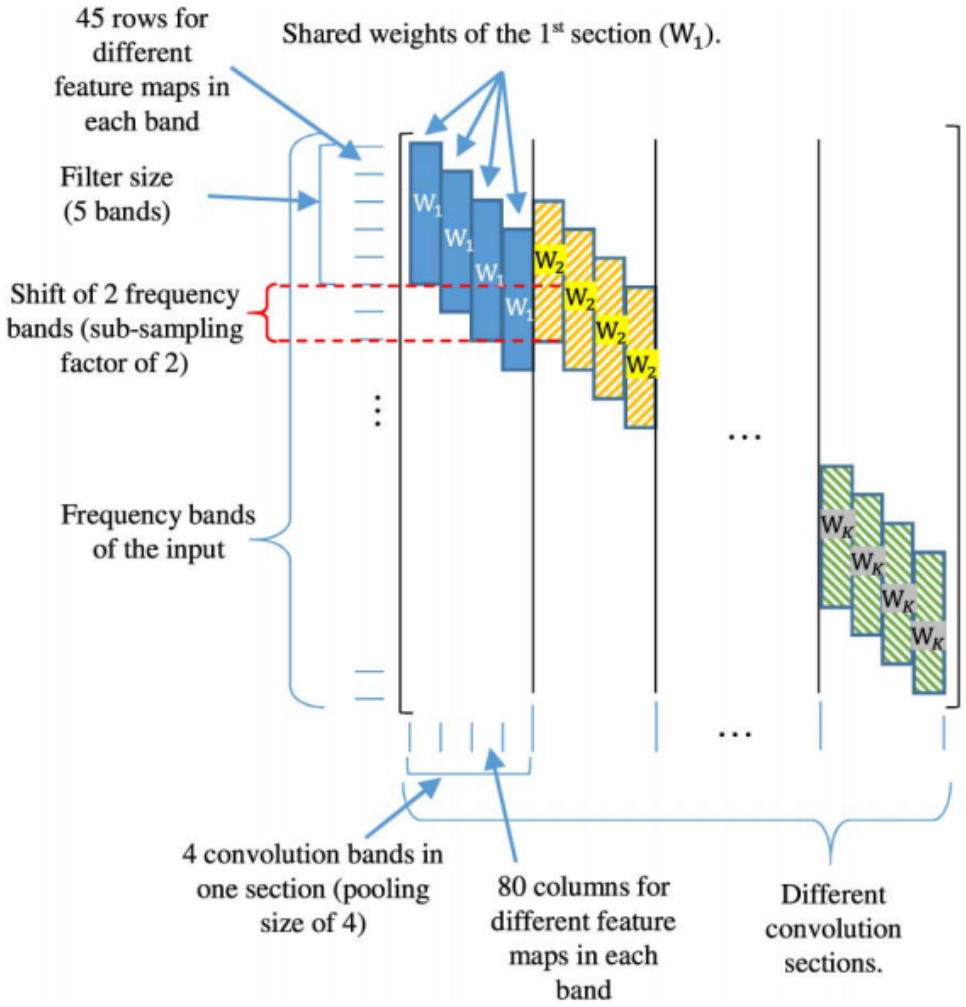


Fig. 6. The CNN layer using limited weight sharing (LWS) can also be represented as matrix multiplication using a large sparse matrix where local connectivity and weight sharing are represented in matrix form. The above figure assumes a filter size of 5, a pooling size of 4, 45 input feature maps, and 80 feature maps in the convolution ply.

DeepSpeech-RNN based

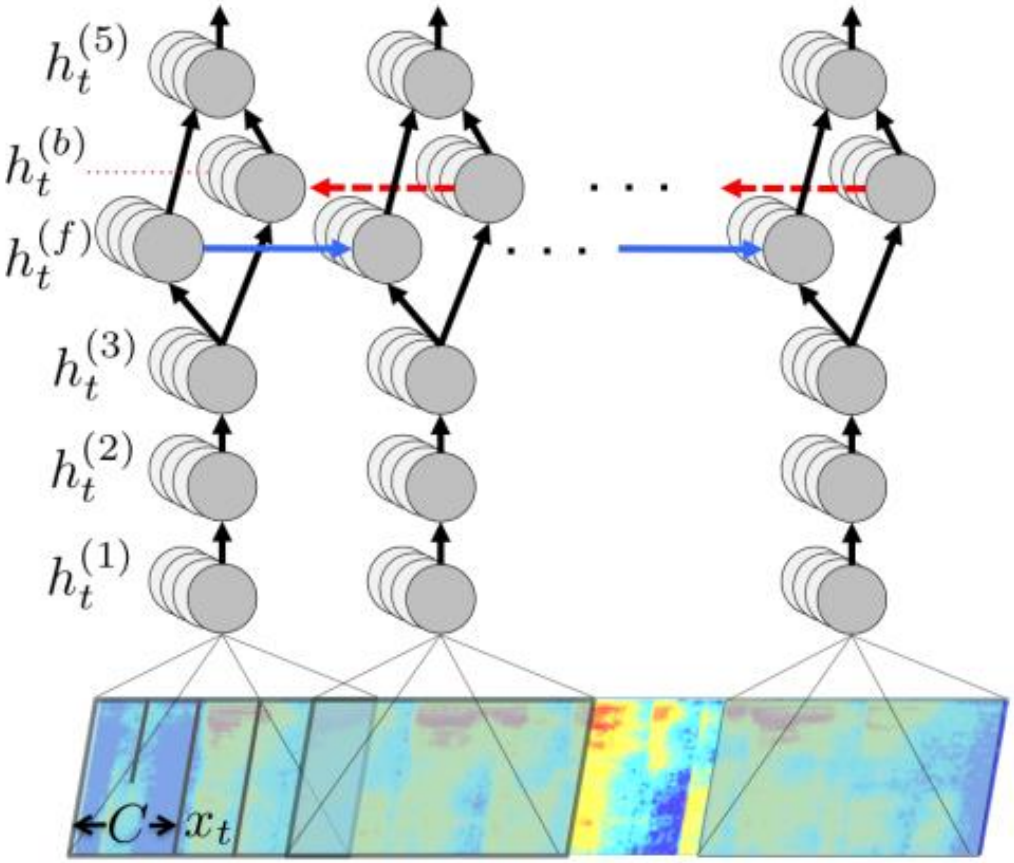


Figure 1: Structure of our RNN model and notation.

These outputs define the probabilities of all possible ways of aligning all possible label sequences with the input sequence. The total probability of any one label sequence can then be found by summing the probabilities of its different alignments.

RNN output	Decoded Transcription
what is the weather like in bostin right now	what is the weather like in boston right now
prime miniter nerenr modi	prime minister narendra modi
arther n tickets for the game	are there any tickets for the game

Table 1: Examples of transcriptions directly from the RNN (left) with errors that are fixed by addition of a language model (right).

References

- Dahl G E, Yu D, Deng L, et al. Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition[J]. IEEE Transactions on Audio Speech & Language Processing, 2012, 20(1):30 - 42.

- Abdel-Hamid O, Mohamed A R, Jiang H, et al. Convolutional Neural Networks for Speech Recognition[J]. IEEE/ACM Transactions on Audio Speech & Language Processing, 2014, 22(10):1533-1545.

- Hannun A, Case C, Casper J, et al. Deep Speech: Scaling up end-to-end speech recognition[J]. Eprint Arxiv, 2014.

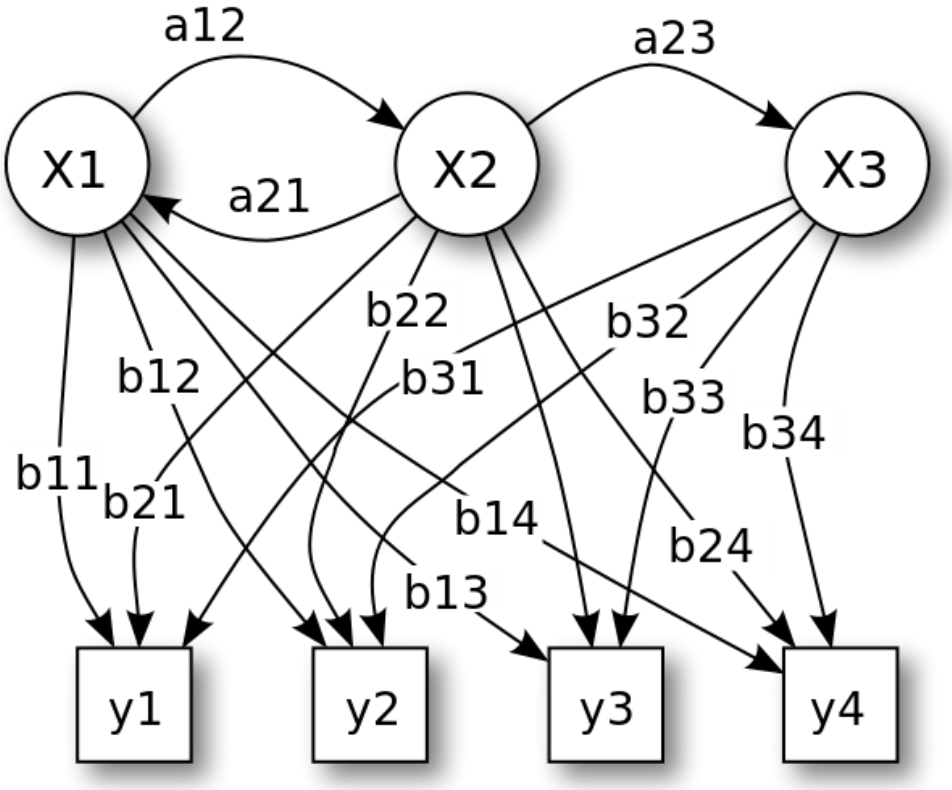
- Yu D, Deng L. Automatic speech recognition : a deep learning approach[M]// Springer, 2015.

- Speech and Language Processing: An introduction to natural language proce ssing,computational linguistics, and speech recognition. Daniel Jurafsky & Ja mes H. Martin.Copyright c 2006, All rights reserved. Draft of June 25, 2007

- http://blog.csdn.net/abcjennifer/article/details/27346787

Thanks!

HMM



X — states
y — possible observations
a — state transition probabilities
b — output probabilities

Disadvantage of Viterbi：

1、通过Viterbi Decoder，并没有计算最优的字序列，而是计算出了最优的状态序列，也可以说是音素序列。

2、该算法是通过记录拥有最大概率的状态路径，来得出最优学列的，而实际上这只是对于一个观察序列生成该状态序列的概率的近似，例如，一个字词有很多种发音，那么分配给不同的状态路径的概率自然是比那些只有单种发音的字词的，此时，该算法会倾向于选择后者。

3、不能够有效地使用更多的领域知识，例如只能使用bigram grammar， 而不能使用 trigram， 因为其违反了dynamic programming invariant，举个例子就是，本来在trigram grammar上，Wx 在Wy和Wz给定的情况下，出现的概率是很高的，但是Wy在给定Wz和任意其它状态情况下，出现概率很低，那么实际上就不会出现给定Wy和Wz的情况。

Solution:

1、Multipass Decoding: *N*-best

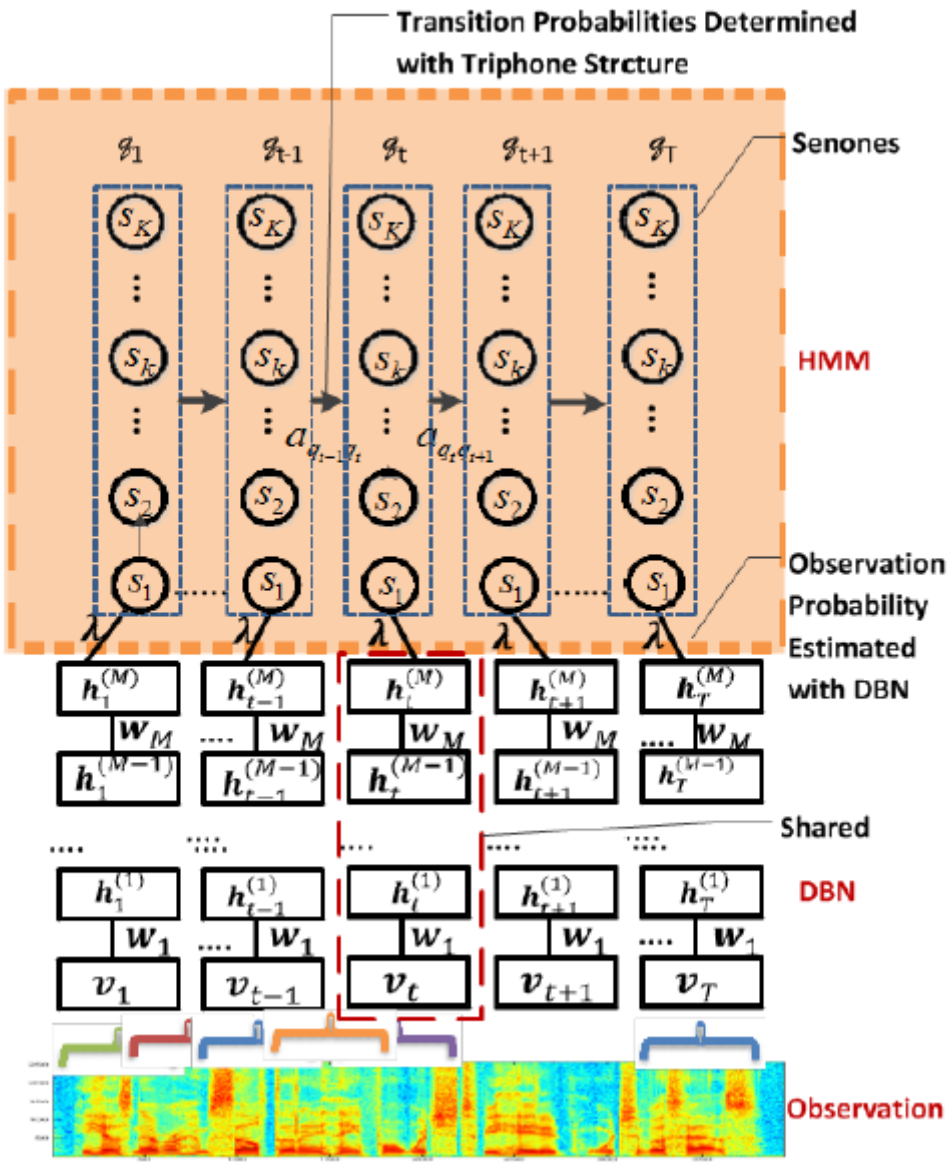
2、A* (Stack) Decoding

Discriminative Training

$$b_j(o_t) = P(o_t|q_j) = \frac{P(q_j|o_t) * P(o_t)}{P(q_j)}$$

- Uses a larger window of acoustic information, i.e. a total of 9 cepstral feature vectors instead of the single one that the Gaussian model uses
- The supervised learning algorithms for training a SVM or MLP posterior phone classifiers require that we know the correct phone label for each observation .

CD-DBN-HMM


Transition Probabilities Determined with Triphone Strcture
Senones
HMM
Observation Probability Estimated with DBN
Shared
DBN
Observation

CD-DBN-HMM

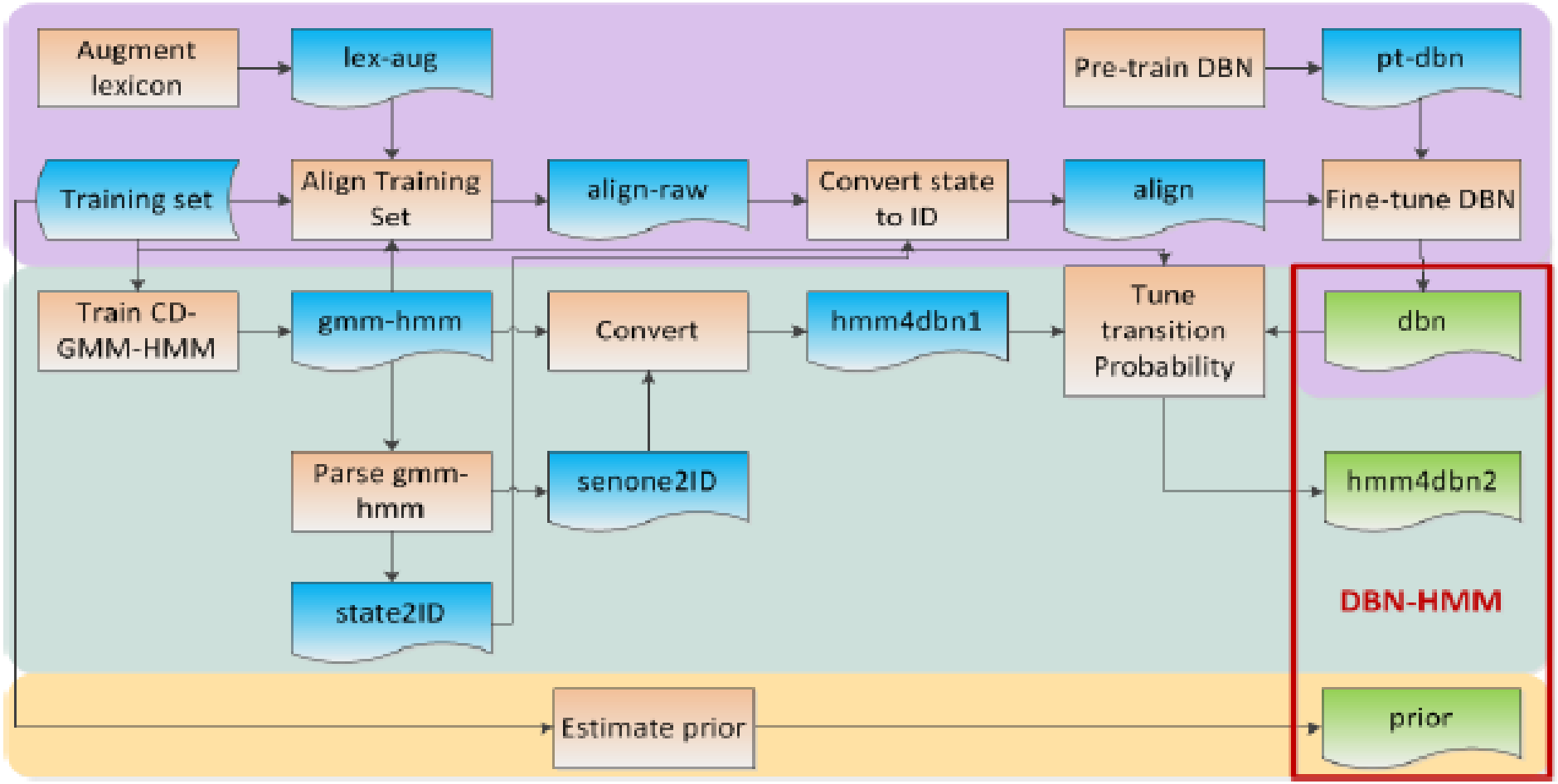
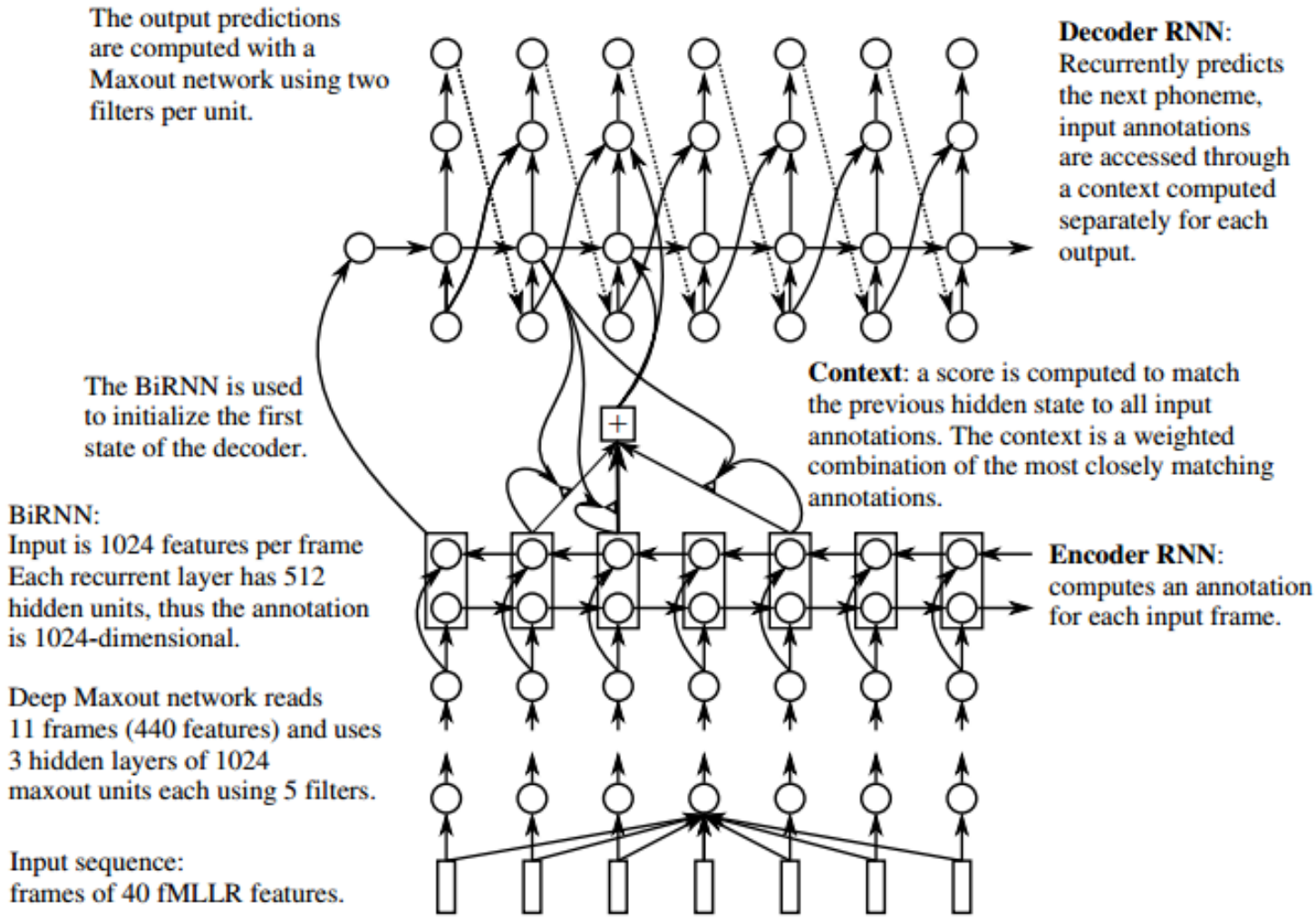


Figure 2: The procedure to train CD-DBN-HMMs.


The output predictions are computed with a Maxout network using two filters per unit.
Decoder RNN: Recurrently predicts the next phoneme, input annotations are accessed through a context computed separately for each output.
The BiRNN is used to initialize the first state of the decoder.
Context: a score is computed to match the previous hidden state to all input annotations. The context is a weighted combination of the most closely matching annotations.
BiRNN: Input is 1024 features per frame Each recurrent layer has 512 hidden units, thus the annotation is 1024-dimensional.
Encoder RNN: computes an annotation for each input frame.
Deep Maxout network reads 11 frames (440 features) and uses 3 hidden layers of 1024 maxout units each using 5 filters.
Input sequence: frames of 40 fMLLR features.