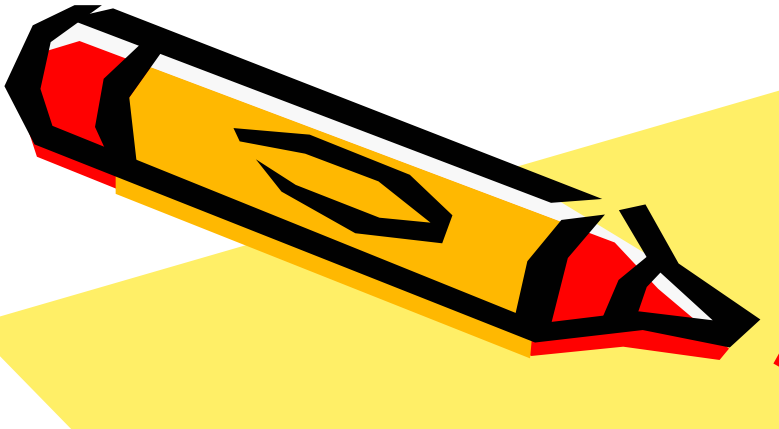
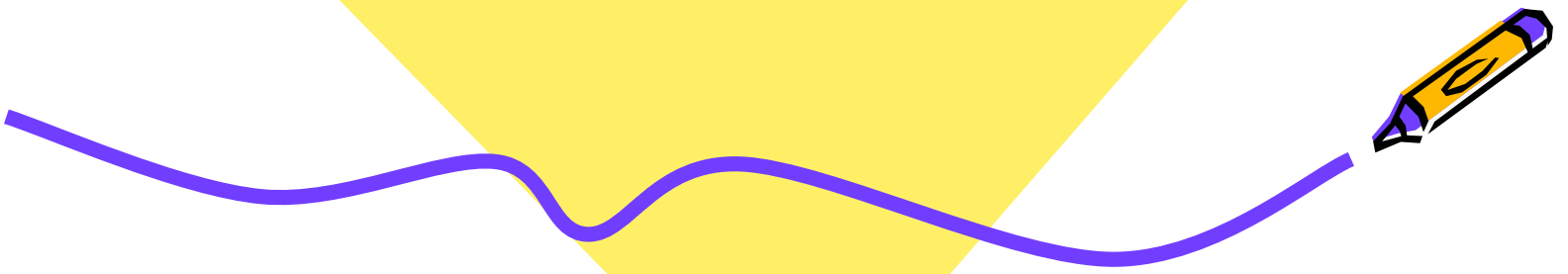


HMMS and Speech Recognition

Presented by Jen-Wei Kuo

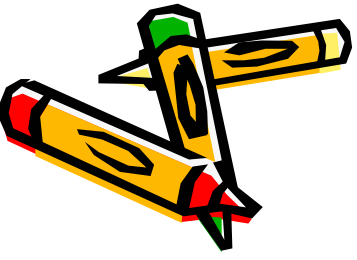


Reference

1. X. Huang et. al., Spoken Language Processing, Chapter 8

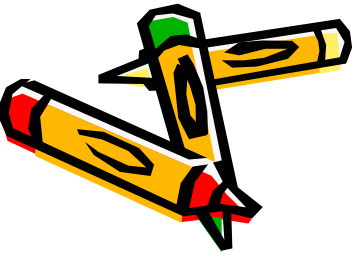
2. Daniel Jurafsky and James H. Martin, Speech and Language Processing, Chapter 7

3. Berlin Chen, Fall, 2002: Speech Signal Processing, Hidden Markov Models for Speech Recognition



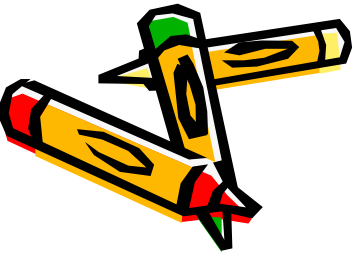
Outline

- Overview of Speech Recognition Architecture
- Overview of Hidden Markov Models
- The Viterbi Algorithm Revisited
- Advanced Methods for Decoding
  - A* Decoding
- Acoustic Processing of Speech
  - Sound Waves
  - How to Interpret a Waveform
  - Spectra
  - Feature Extraction



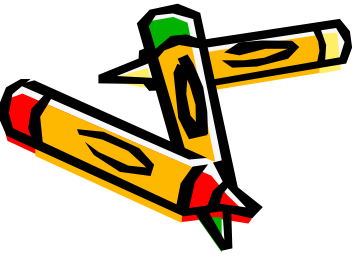
Outline (Cont.)

- Computing Acoustic Probabilities
- Training a Speech Recognizer
- Waveform Generation for Speech Synthesis
  - Pitch and Duration Modification
  - Unit Selection
- Human Speech Recognition
- Summary



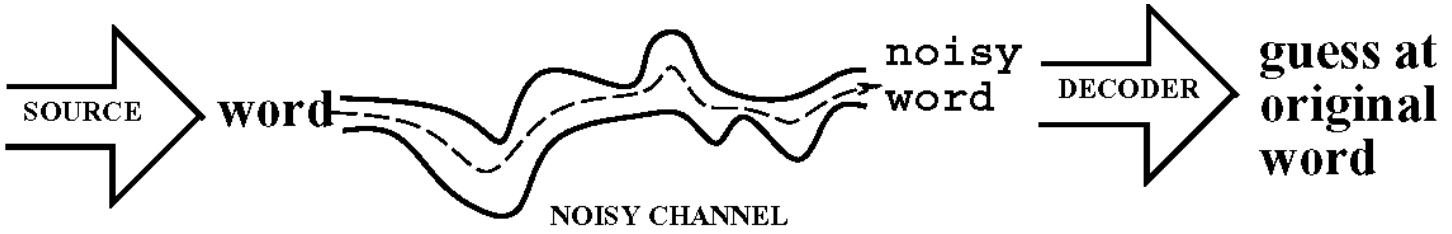
HMMs and Speech Recognition

- Application : Large – Vocabulary Continuous
- Speech Recognition (LVCSR)
- Large vocabulary : Dictionary size 5000 – 60000 words
- Isolated – word speech : each word followed by a pause
- Continuous speech : words are run together naturally
- Speaker-independent

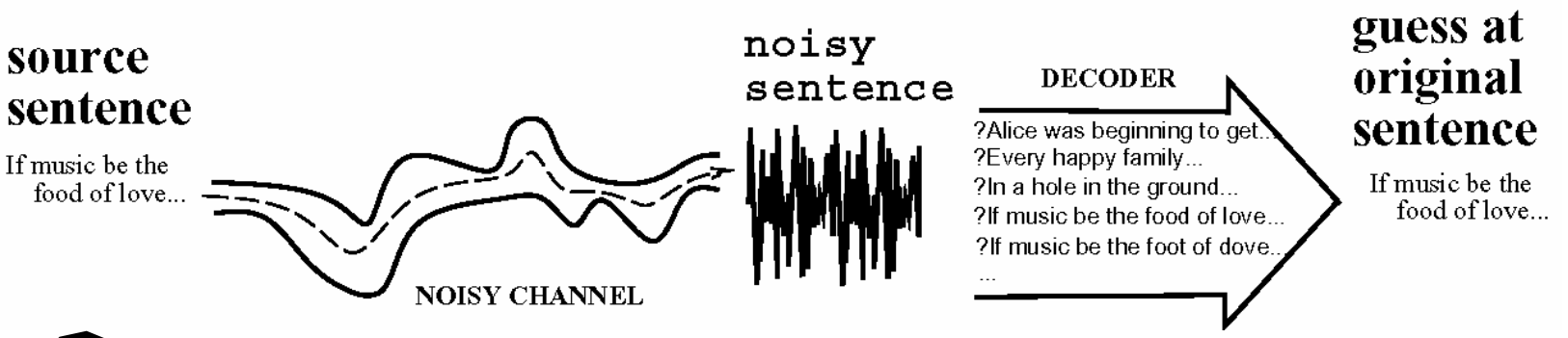


Speech Recognition Architecture

↓ Figure 5.1 The noisy channel model of individual words



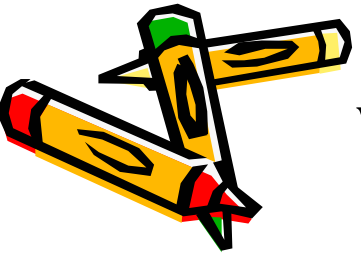
Acoustic input considered a noisy version of a source sentence.



↑ Figure 7.1 The noisy channel model applied to entire sentences

Speech Recognition Architecture

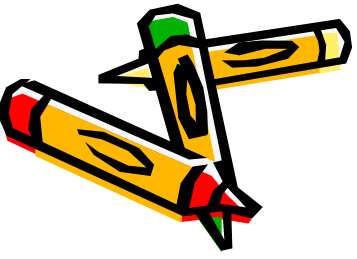
- Implementing the noisy-channel model have two problems.
  - Metric for selecting best match? *probability*
  - Efficient algorithm for finding best match? *A**
- Modern Speech Recognizer
  - Providing a search through a huge space of potential ” source” sentences.
  - And choosing the one which has the highest probability of generating this sentence.
  - So they use models to express the probability of words.
  - N-grams and HMMs are applied.



Speech Recognition Architecture

- The goal of the probabilistic noisy channel architecture for speech recognition can be summarized as follows :

What is the most likely sentence out of all sentences in the language L given some acoustic input O ?



Speech Recognition Architecture

- Observations : $O = o_1, o_2, o_3, \cdots, o_t$
- Word Sequences : $W = w_1, w_2, w_3, \cdots, w_n$
- Probabilistic implementation can be expressed :

$$\hat{W} = \arg\max_{W \in L} P(W \mid O)$$

- Then we can use Bayes' rule to break it down :

$$\hat{W} = \arg\max_{W \in L} P(W \mid O) = \arg\max_{W \in L} \frac{P(O \mid W)P(W)}{P(O)}$$

$$\left(\because P(W \mid O) = \frac{P(WO)}{P(O)} \quad and \quad P(O \mid W) = \frac{P(WO)}{P(W)} \right)$$

$$\therefore P(W \mid O) \cdot P(O) = P(WO) = P(O \mid W) \cdot P(W)$$

Speech Recognition Architecture

- For each potential sentence we are still examining the same observations O, which must have the same probability $P(O)$.

$$\hat{W} = \arg\max_{W \in L} P(W \mid O)$$

Posterior probability

$$= \arg\max_{W \in L} \frac{P(O \mid W) P(W)}{P(O)} = \arg\max_{W \in L} P(O \mid W) P(W)$$

Observation likelihood
Acoustic model

Prior probability
Language model

Speech Recognition Architecture

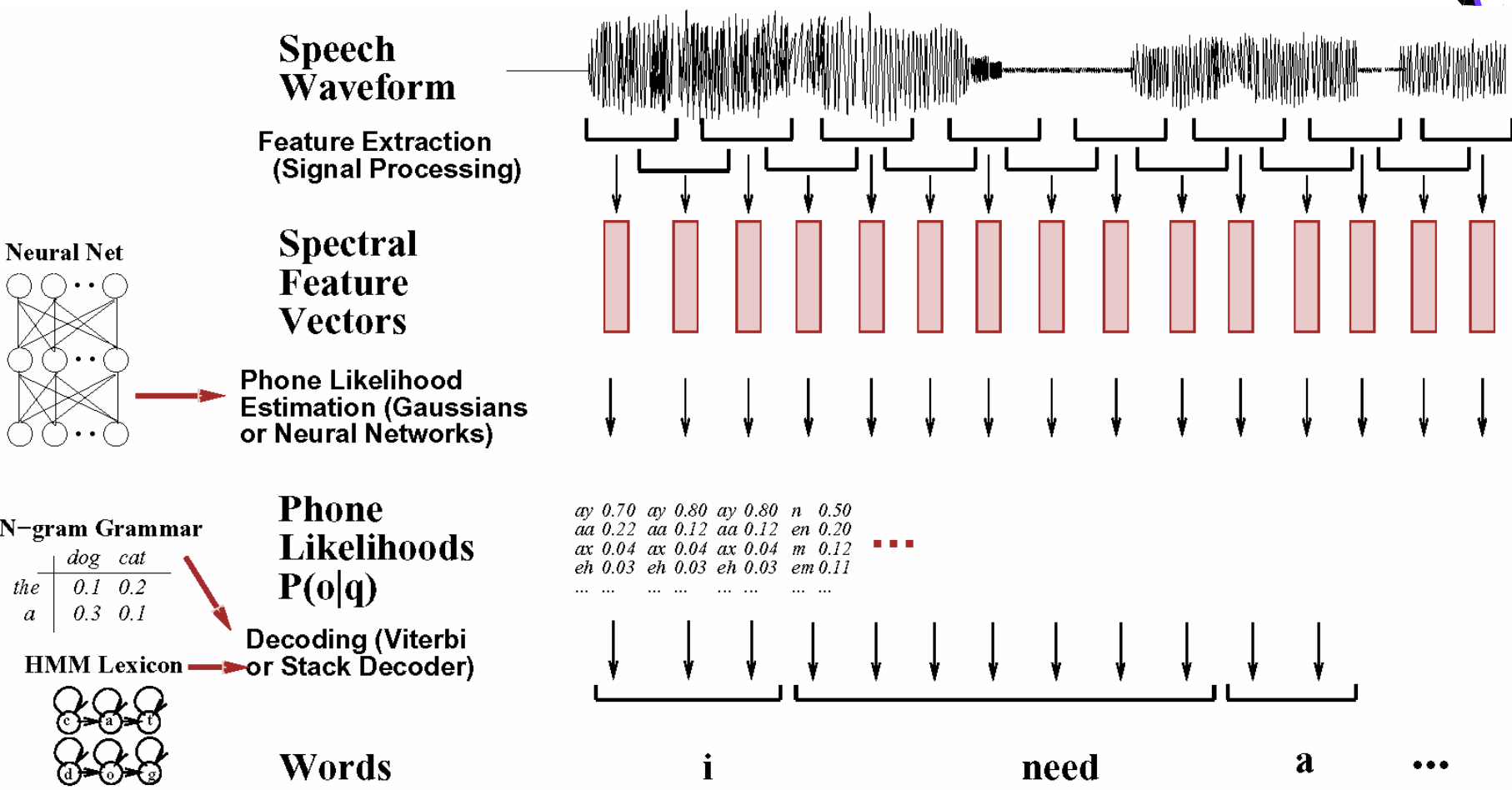
Errata ! page 239, line -7 : Change "can be computing" to "can be computed".

- Three stage for speech recognition system
  - Signal processing or Feature extraction stage :
    - Waveform is sliced up into frames.
    - Waveform are transformed into spectral features.
  - Subword or Phone recognition stage :
    - Recognize individual speech.
  - Decoding stage :
    - Find the sequence of words that most probably generated the input.

Errata ! page 240, line -12 : Delete extraneous closing paren. ") ("

Speech Recognition Architecture

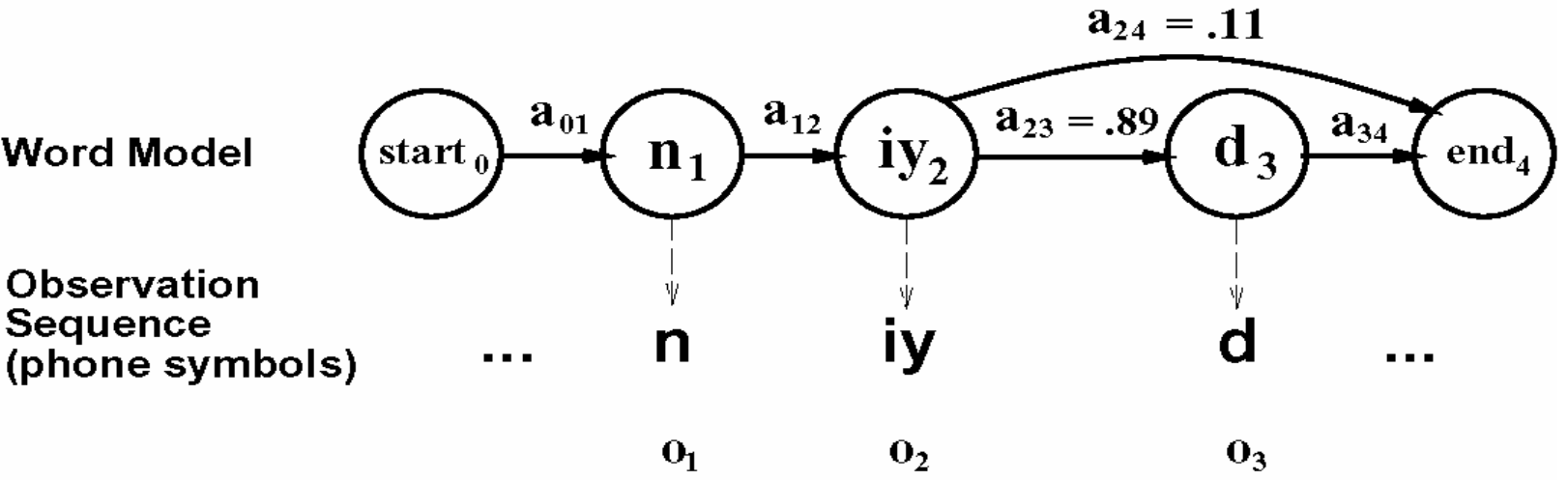
↓Figure 7.2 Schematic architecture for a speech recognition



Overview of HMMs

- Previously, Markov chains used to model pronounciation

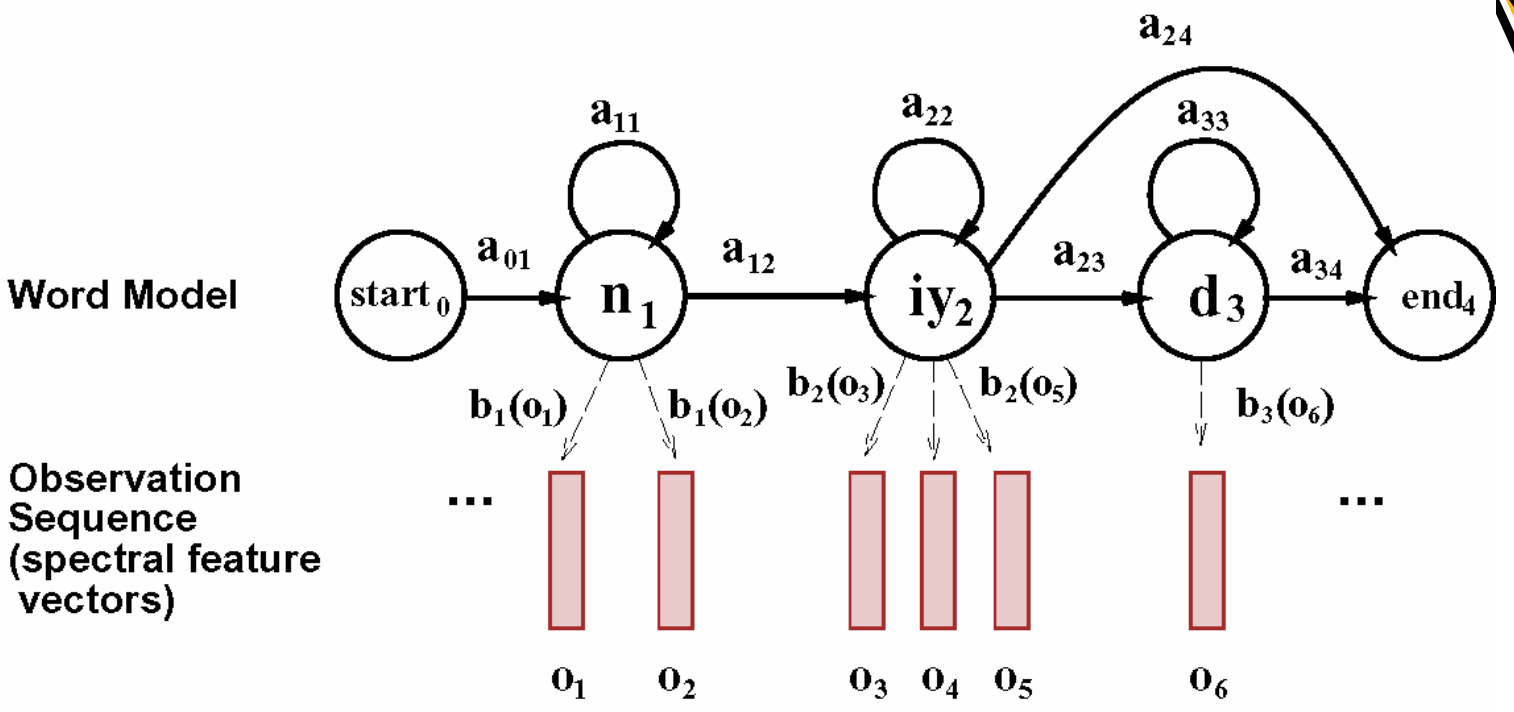
↓ Figure 7.3 A simple weighted automaton or Markov chain pronunciation network for the work *need.* The transition probabilities a_{xy} between two states x and y are 1.0 unless otherwise specified.



Overview of HMMs

- Forward algorithm：Phone sequences likelihood.
- Real input is not symbolic: Spectral features
- input symbols do not correspond to machine states
- HMM definition:
  - State set Q.
  - Observation symbols $O \neq Q$.
  - Transition probabilities $A =$ $a_{01}a_{02}a_{03}a_{n1}a_{nn}$
  - Observation likelihood $B =$ $b_j(o_t)$
  - Two special states：start state and end state
  - Initial distribution： π_i is the probability that the HMM will start in state i.

Overview of HMMs

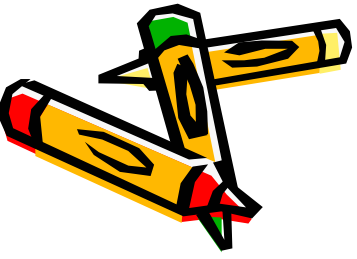


↑ Figure 7.4 An HMM pronunciation network for the word *need*.

- Compared with Markov Chain :
  - Separate set of *observation symbols O*.
  - Likelihood function *B* is not limited to 0 or 1.

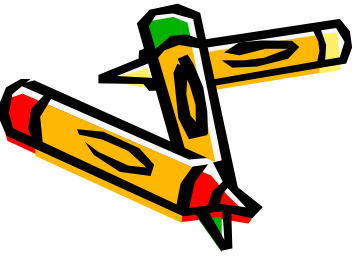
Overview of HMMs

- Visible (Observable) Markov Model
  - One state , one event.
  - States which the machine passed through is known.
  - Too simple to describe the speech signal characteristics.

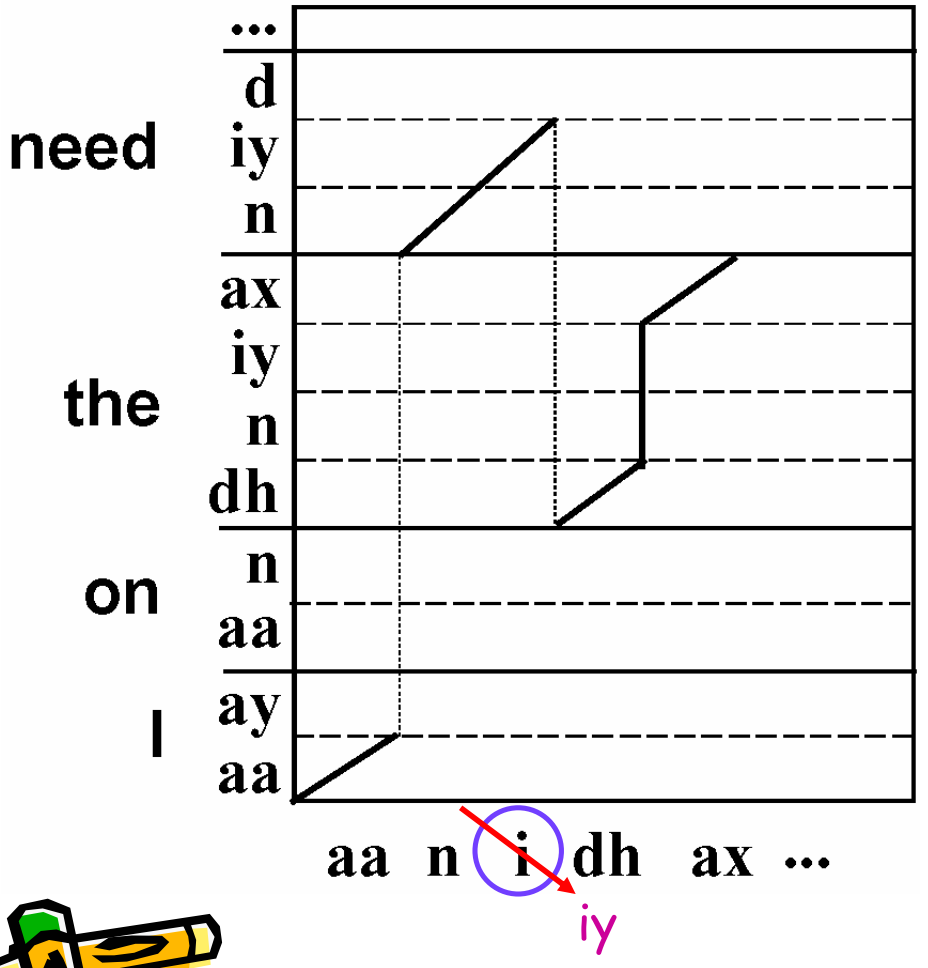


The Viterbi Algorithm Revisited

- Viterbi algorithm：
  - Find the most-likely path through the automaton
- Word boundaries unknow in continuous speech
  - If we know where the word boundaries.
    - we can sure the pronunciation came from one word.
    - Then, we only had some candidates to compare.
  - But it’ s the lack of spaces indicating word boundaries.
    - It make the task difficult.
  - Segmentation
    - The task of finding word boundaries in connected speech.
    - It will solve it by using the Viterbi algorithm.



The Viterbi Algorithm Revisited



Errata ! page 246, Figure 7.6 :
Change " i " to " iy " on x axis.

↑ Figure 7.6 Result of the Viterbi algorithm used to find the most-likely phone sequence

The Viterbi Algorithm Revisited

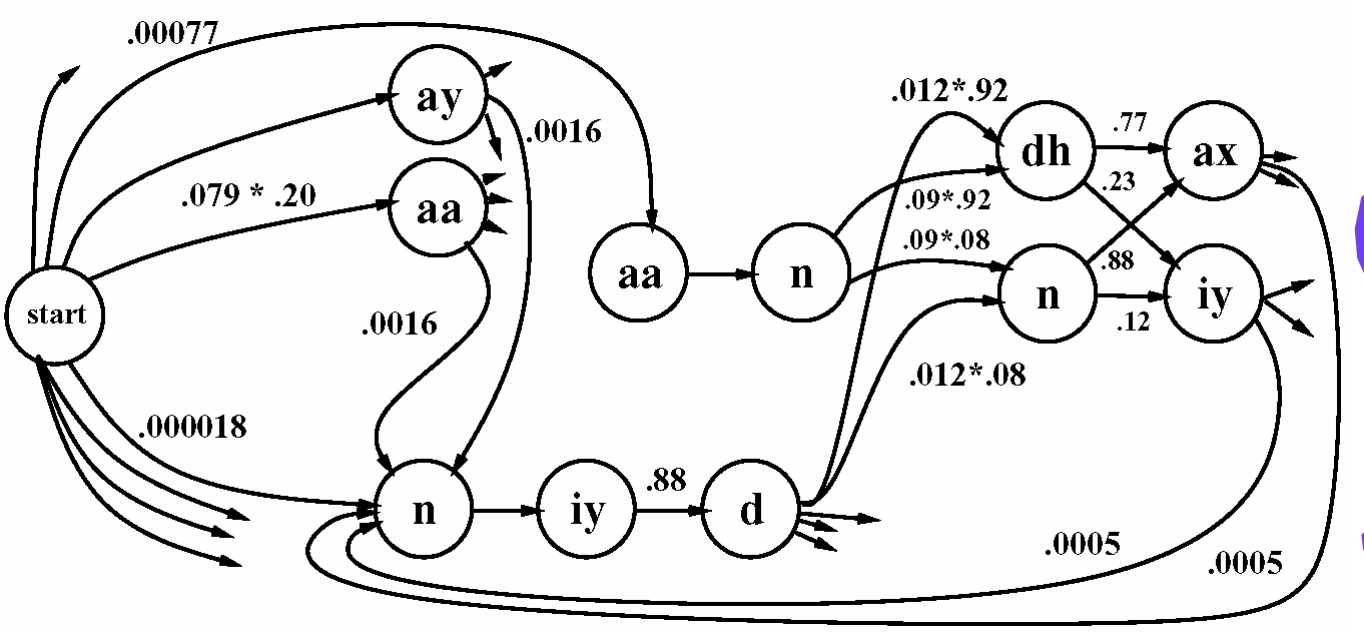
$$viterbi[t,j] = \max_{q_1,q_2,\ldots,q_{t-1}} P(q_1 q_2 \ldots q_{t-1}, q_t = j, o_1 o_2 \ldots o_t \mid \lambda)$$
$$= \max_i (viterbi[t-1,i] a_{ij}) b_j(o_t)$$

- Assumption of Viterbi algorithm：
  - Dynamic programming invariant
    - If ultimate best path for O includes state q_i , that this best path must include the best path up to state q_i
    - This doesn’ t mean that the best path at any time t is the best path for the whole sequence. (bad path →best path)
    - Does not work for all grammars, ex: trigram grammars

Errata ! page 247, line -2：Replace “Figure 7.9 shows” to “Figure 7.10 shows”

The Viterbi Algorithm Revisited

I need	0.0016	need need	0.000047	# Need	0.000018
I the	0.00018	need the	0.012	# The	0.016
I on	0.000047	need on	0.000047	# On	0.00077
I I	0.039	need I	0.000016	# I	0.079
the need	0.00051	on need	0.000055		
the the	0.0099	on the	0.094		
the on	0.00022	on on	0.0031		
the I	0.00051	on I	0.00085		



The Viterbi Algorithm Revisited

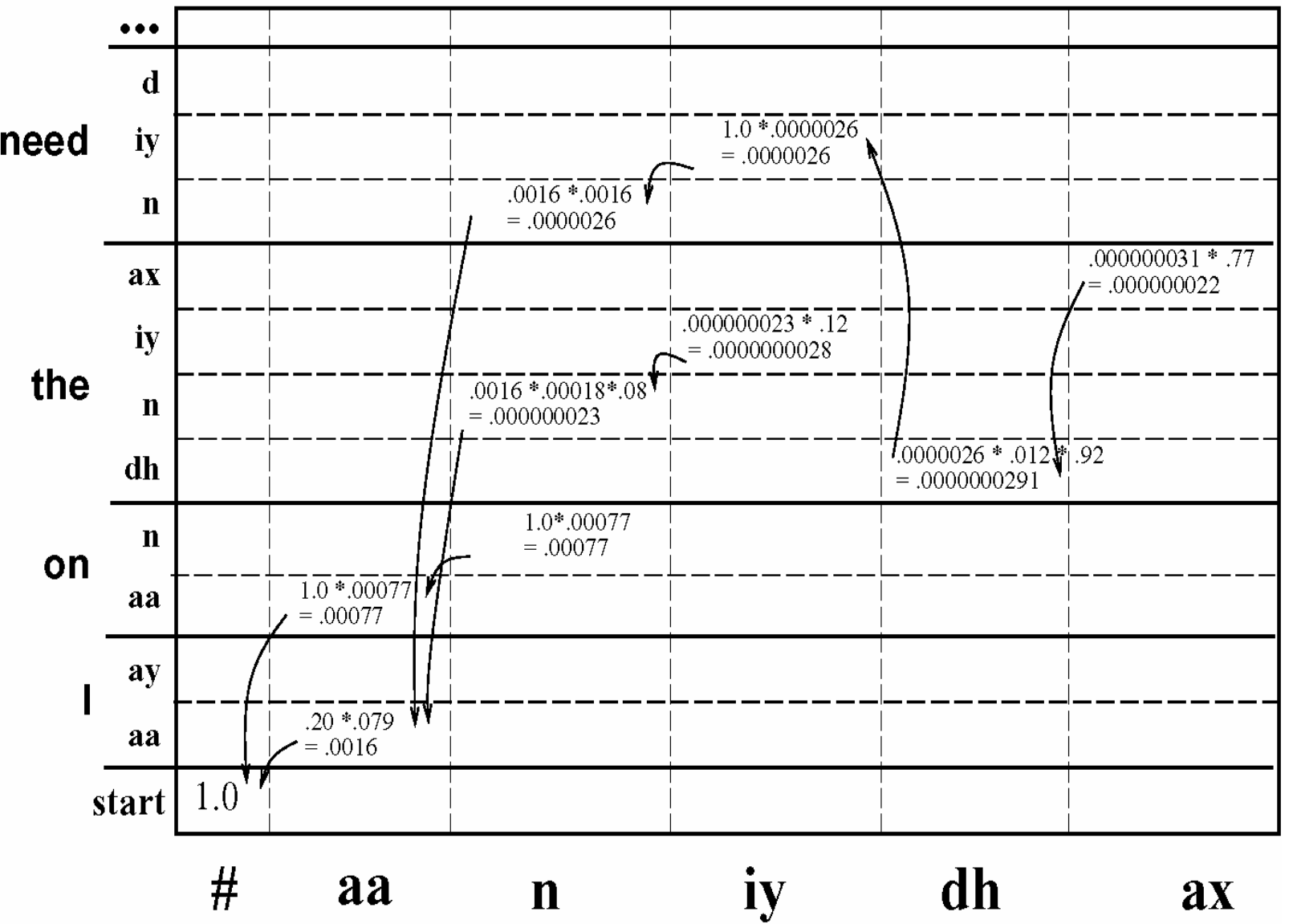
Errata ! page 248, line -6：Change "i dh ax" to "iy dh ax"

function VITERBI(*observations* of len *T*, *state-graph*) **returns** *best-path*

num_states ← NUM-OF-STATES(*state-graph*)
Create a path probability matrix *viterbi[num-states*+2,*T*+2*]*
viterbi[0,0]←1.0
for each time step *t* **from** 0 **to** *T* **do**
  for each state *s* **from** 0 **to** *num-states* **do**
    for each transition *s'* from *s* specified by *state-graph*
      new-score←*viterbi*[*s*,*t*]**a*[*s*,*s'*]**b*$_{s'}$(o_t)
      if ((*viterbi*[*s'*,*t*+1] = 0) || (*new-score* > *viterbi*[*s'*,*t*+1]))
        then
          viterbi[*s'*,*t*+1]←*new-score*
          back-pointer[*s'*,*t*+1]←*s*
Backtrace from highest probability state in the final column of *viterbi*[] and return path.

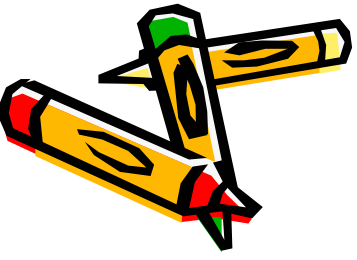
Errata ! page 249, Figure 7.9 caption：Change "minimum" to "maximum"

The Viterbi Algorithm Revisited



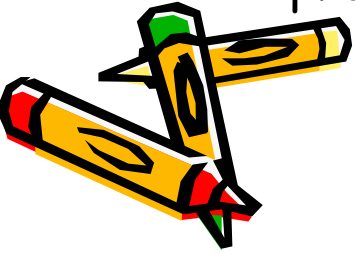
The Viterbi Algorithm Revisited

- Viterbi decoding are complex in three key way：
  - The input of HMM would not be phone
    - Instead, the input is a feature vector.
    - The observation likelihood probabilities will not simply take on the values 0 or 1.
    - It will be more fine-grained probability estimates.ex : Gaussian probability estimators.
  - The HMM states may not be simple phones
    - Instead, it may be subphones.
    - Each phone may be divided into more than one state.
    - This method could provide the intuition that the significant changes in the acoustic input happen.



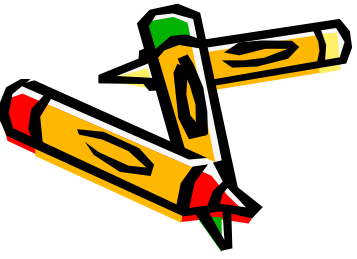
The Viterbi Algorithm Revisited

- It is too expensive to consider all possible paths in LVCSR
  - Instead, low probability paths are pruned at each time step.
  - This is usually implemented via **beam search**.
  - For each time step, the algorithm maintains a short list of high-probability words whose path probabilities are within some range.
  - Only transitions from these words are extended at next time step.
  - So, at each time step the words are ranked by the probability of the path.



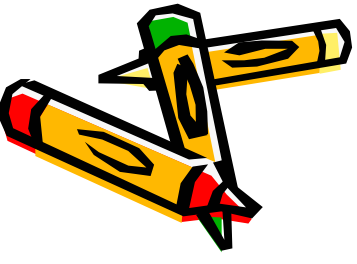
Advanced Methods for Decoding

- Viterbi decoder has two limitations：
  - Computes most probable state sequence, not word sequence
    - Sometimes the most probable sequence of phones does not correspond to the most probable word sequence.
    - The word has shorter pronunciation will get higher probability than the word has longer pronunciation.
  - Cannot be used with all language models
    - In fact, it only could be used in bigram grammar.
    - Since it violates the **dynamic programming invariant**.



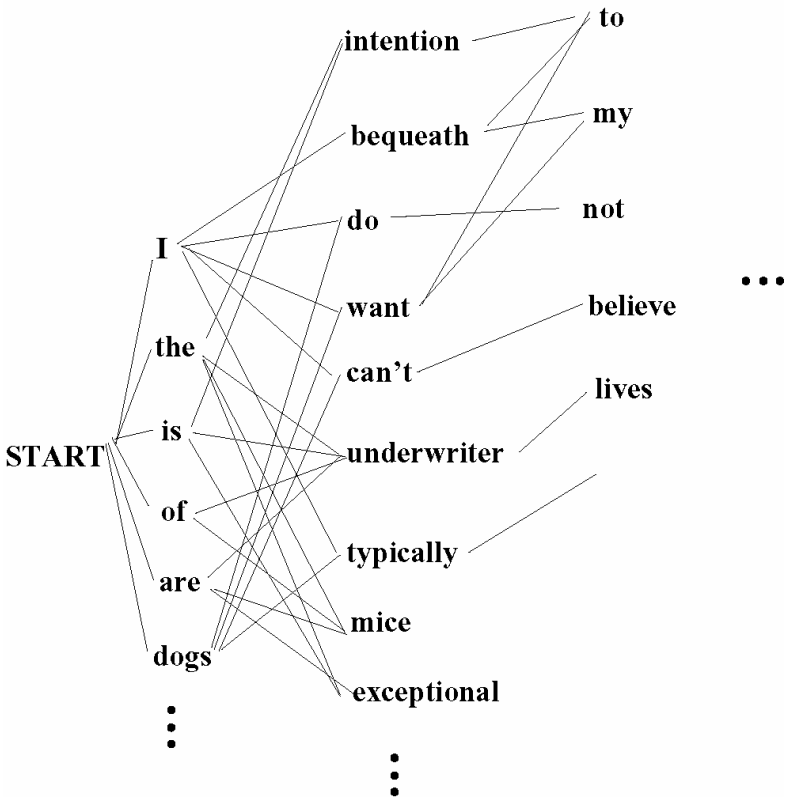
Advanced Methods for Decoding

- Two classes of solutions to viterbi decoder problems：
  - Solution 1：Multiple-pass decoding
    - N-best-Viterbi：Return N best sentences, sort with more complex model.
    - Word lattice： Return "directed word graph" and "word observation likelihoods", refine with more complex model.
  - Solution 2：A* decoder
    - Compared with Viterbi：
      - viterbi：Approximation of the forward algorithm, max instead of sum.
      - A*：Using the complete forward algorithm correct observation likelihoods, and allow us to use arbitrary language model.



Advanced Methods for Decoding

- A kind of best-first search of the lattice or tree.
- Keeping a **priority queue** of partial paths with scores.



↑ Figure 7.13 A word lattice

Advanced Methods for Decoding

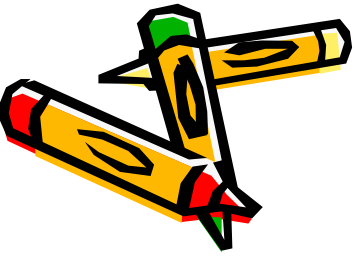
Errata ! page 255, line 1 : Change "a A*" to "an A*"

Errata ! page 256, Figure 7.14 : The main loop is missing from this pseudocode. Add a "While (queue is not empty A)" after the initialization of the queue and before the Pop

Errata ! page 256, Figure 7.14 caption : Change "possibly" to " possible"

function STACK-DECODING() **returns** *min-distance*

Initialize the priority queue with a null sentence.
Pop the best (highest score) sentence s off the queue.
If (s is marked end-of-sentence (EOS)) output s and terminate.
Get list of candidate next words by doing fast matches.
For each candidate next word w:
    Create a new candidate sentence $s+w$.
    Use forward algorithm to compute acoustic likelihood L of $s+w$
    Compute language model probability P of extended sentence $s+w$
    Compute "score" for $s+w$ (a function of L, P, and ???)
    if (end-of-sentence) set EOS flag for $s+w$.
    Insert $s+w$ into the queue together with its score and EOS flag

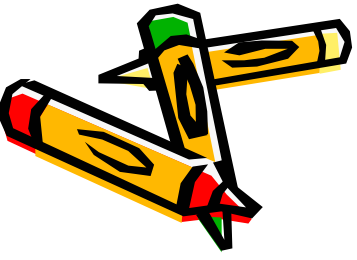


Advanced Methods for Decoding

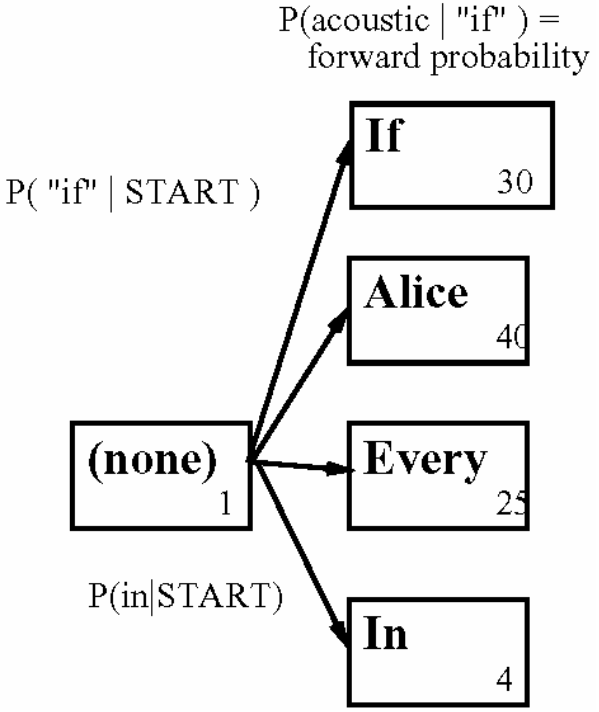
- A* Algorithm:
  - Select the highest-priority path (pop queue)
  - Create possible extensions (if none, stop)
  - Calculate scores for extended paths

    (from forward algorithm and language model)
  - Add scored paths to queue

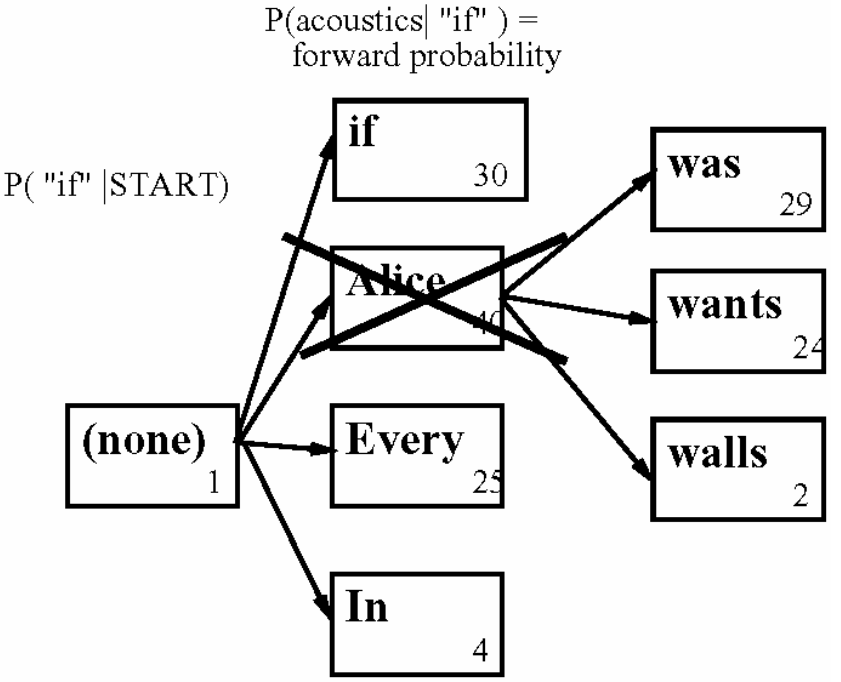
- Example: Search for the sentence "*If music be the food of love.*"



Advanced Methods for Decoding

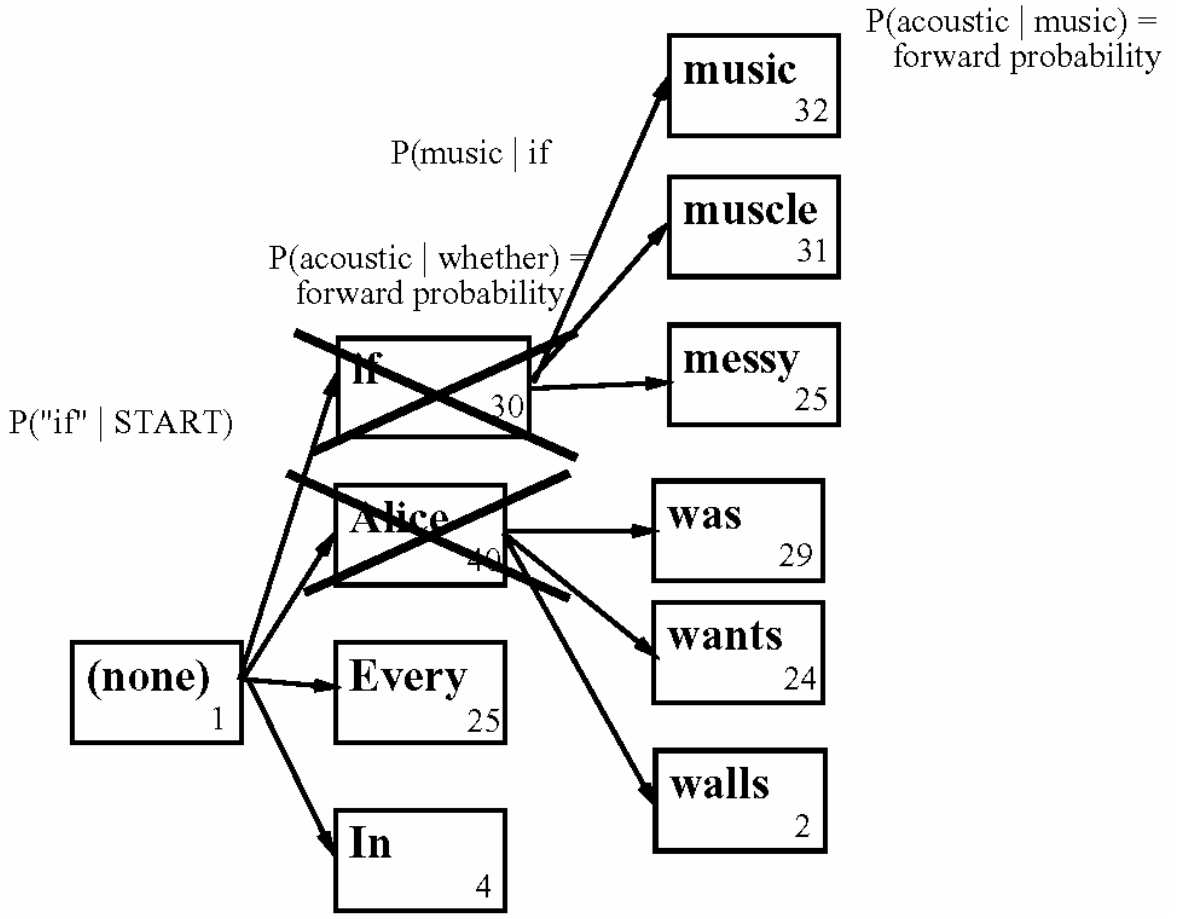


↑ Figure 7.15 Beginning



↑ Figure 7.16
Expanding "Alice" node

Advanced Methods for Decoding



↑ Figure 7.17
Expanding "if" node

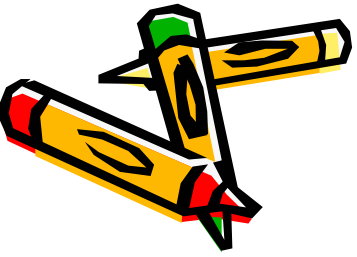
Advanced Methods for Decoding

- How to determine the score for each node ?
  - If we use $P(y_1^j \mid w_1^i)P(w_1^i)$
    - Then the probability will be much smaller for a longer path than a shorter one. (path prefix will have higher score)
    - Instead, we use " A* evaluation function" .
    - Given a partial path p :

$$f^*(p) = g(p) + h^*(p)$$

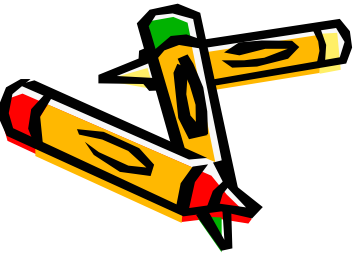
estimate | path p | heuristic

$$P(A \mid W)P(W)$$



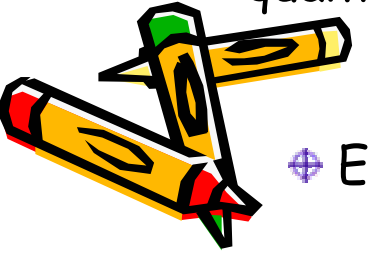
Acoustic Processing of Speech

- Two important characteristics of a wave
  - Frequency and Pitch
    - The frequency is the number of times a second that a wave repeats itself, or **cycles**.
    - Unit : **cycles per second** are usually clled **Hertz** (Hz)
    - The pitch is the perceptual correlate of frequency
  - Amplitude and loudness
    - The amplitude measures the amount of air pressure variation.
    - Loudness is the perceptual correlate of the power, which is related to the square of the amplitude.



Acoustic Processing of Speech

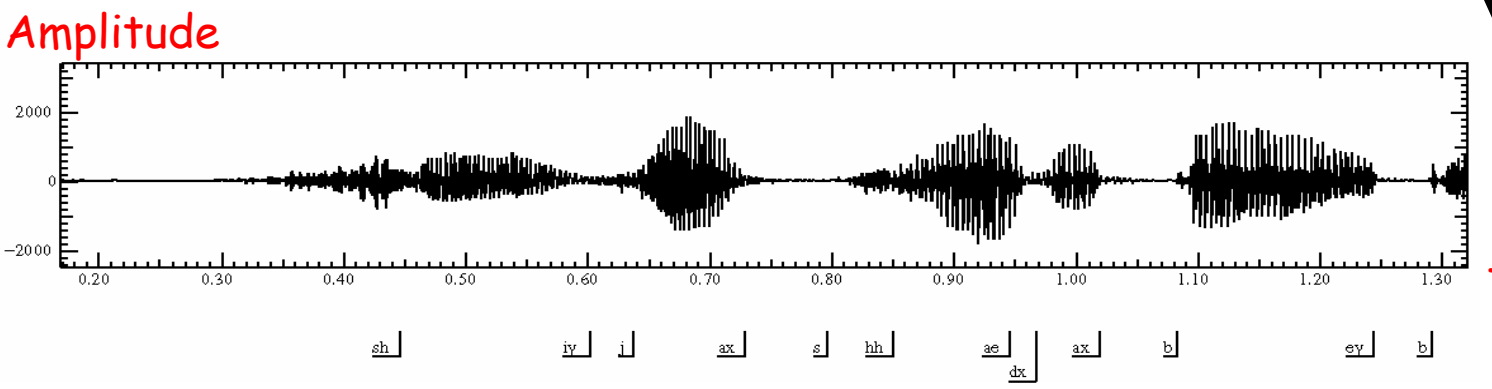
- Feature extraction
  - Analog-to-digital conversion
    - sampling : In order to accurately measure a wave, it is necessary to have at least two samples in each cycle :
      - One measuring the positive part of the wave
      - The other one measuring the negative part
      - Thus the maximum frequency wavethat can be measured is one whose frequency is half the sample rate.
      - This maximum frequency for a given sampling rate is called the **Nyquist frequency**.
    - quantization : Representing a real-valued number as an integer.



Errata ! page 266, line -13 : Change "a integer" to "an integer"

- Either 8-bit or 16-bit integer.

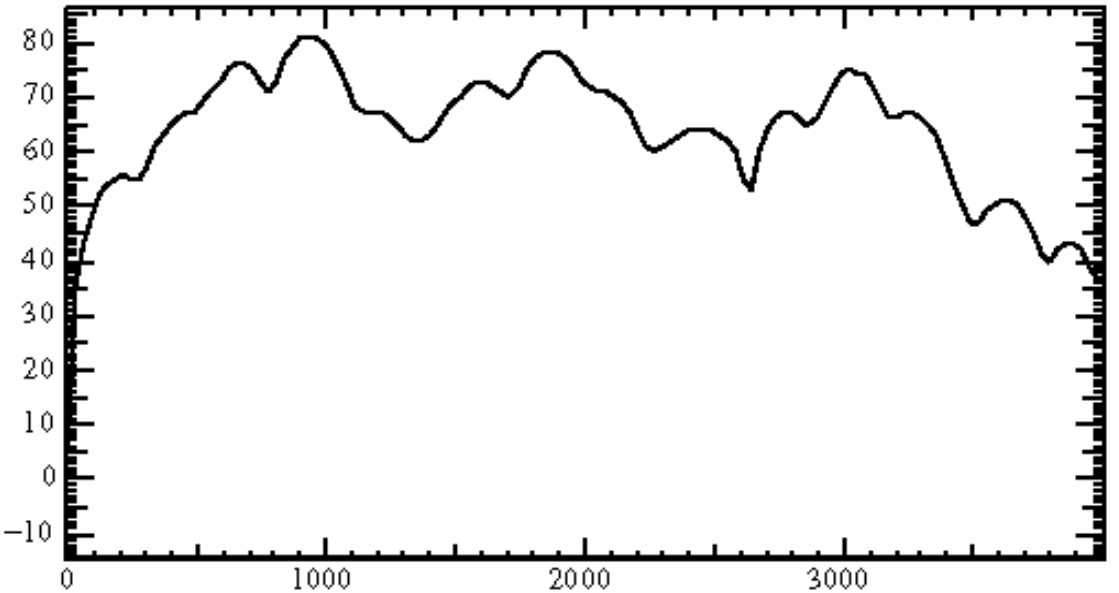
Acoustic Processing of Speech



- Feature extraction
  - Spectrum
    - Based on the insight of Fourier that every complex wave can be represented as a sum of many simple waves of different frequencies.
    - Spectrum is a representation of these different frequency components.

Acoustic Processing of Speech

- Feature extraction
  - Smoothing
    - Goal：Finding where the spectral peaks (formants) are, we could get the characteristic of different sounds.→ determining vowel identity
    - The most common methods are Linear Predictive (LPC) and Cepstral Analysis, or variants of these.



Acoustic Processing of Speech

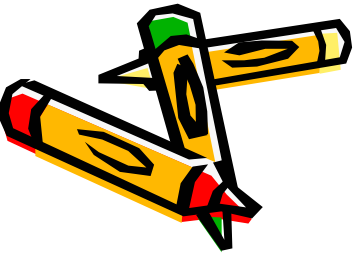
- Feature extraction
  - LPC spectrum (Linear Predictive Coding)
    - Represented by a vector of features.
    - It is possible to use LPC features directly as the observation of HMMs. However, further processing is often done to the features.
    - An all-pole filter with a sufficient number of poles is a good approximation to model the vocal tract (filter) for speech signals.
    - Try to " fit " the frequency response of an " all-pole filter ".
    - It predicts the current sample as a linear combination of its several past samples.

Acoustic Processing of Speech

- Feature extraction
  - Variations of LPC

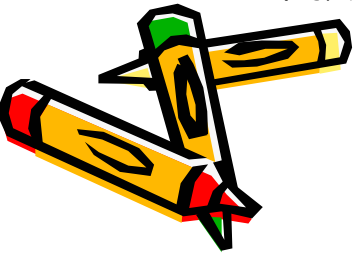
Errata！ page 266, line -4：Replace the definition of cepstral with the following： "One popular feature set is cepstral coefficients, which are computed by an efficient recursion which is conceptually equivalent to taking the inverse Fourier transform of the log spectrum corresponding to the predictor coefficients."

- PLP (Perceptual Linear Predictive analysis)：Takes the LPC features and modifies them in ways consistent with human hearing.



Computing Acoustic Probabilities

- Simple way
  - Vector quantization
    - Cluster features into discrete symbols.
    - Count the occurrences and computer the probability.
- Two modern way – HMM based
    - Calculate probability density function (pdf) over observations.
  - Gaussian observation – probability - estimator
    - Trained by a extension to the **forward-backward algorithm**.
  - Neural Network observation – probability - estimator
    - Trained by a different algorithm : **error back-propagation**.



Computing Acoustic Probabilities

- Gaussian observation – probability - estimator
  - Assumption
    - The possible values of the observation feature vector o_t are normally distributeds.
    - So we represent the observation probability function $b_j(o_t)$ as a Gaussian curve with mean vector μ_j and covariance matrix Σ_j .
    - Then our Gaussian pdf is $b_j(o_t) = \frac{1}{\sqrt{(2\pi)|\Sigma_j|}} e^{(o_t-\mu_j)^T \Sigma_j^{-1} (o_t-\mu_j)}$
    - Usually we make the assumption that the covariance matrix is diagonal.
  - Gaussian mixtures
    - One state has multiple Gaussians.
    - Parameter tying (tied mixtures) : Similar phone states might share Gaussians for some features, just the weights are difference.

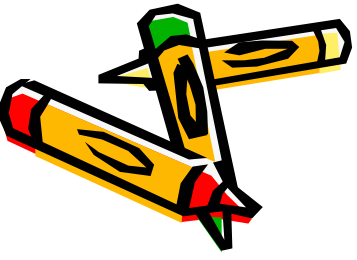
Computing Acoustic Probabilities

- Neural Network observation – probability - estimator
  - The Hybrid HMM-MLP approach
    - The observation probability is done by an MLP instead of a mixture of Gaussians.
    - The input to these MLPs is a representation of the signal at time *t* and some surrounding windows.
    - Thus the input to the network is a set of nine vectors, each vector having the complete set of real-valud spectral features for one time slice.
    - The network has one output unit for each phone; by constraining the values of all the output units to sum to 1, the net can be used to compute the probability of a state *j* given an observation o_t , or $P(o_t | q_j)$.

Computing Acoustic Probabilities

- Neural Network observation – probability - estimator
  - The Hybrid HMM-MLP approach
    - This MLP computes the probability of the HMM state j given an observation o_t , or $P(q_j|o_t)$.
    - But the observation likelihood we need for the HMM, $b_j(o_t)$ is $P(o_t|q_j)$.
    - The Bayes rule can help us see how to computer one from the other.
    - The net is computing... $$P(q_j|o_t)=\frac{P(o_t|q_j)P(q_j)}{P(o_t)}$$
    - We can rearrange the terms as follows:

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

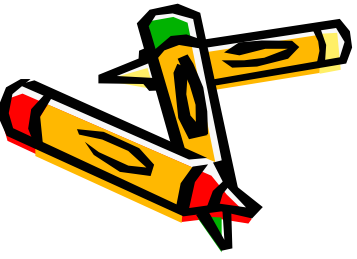


Computing Acoustic Probabilities

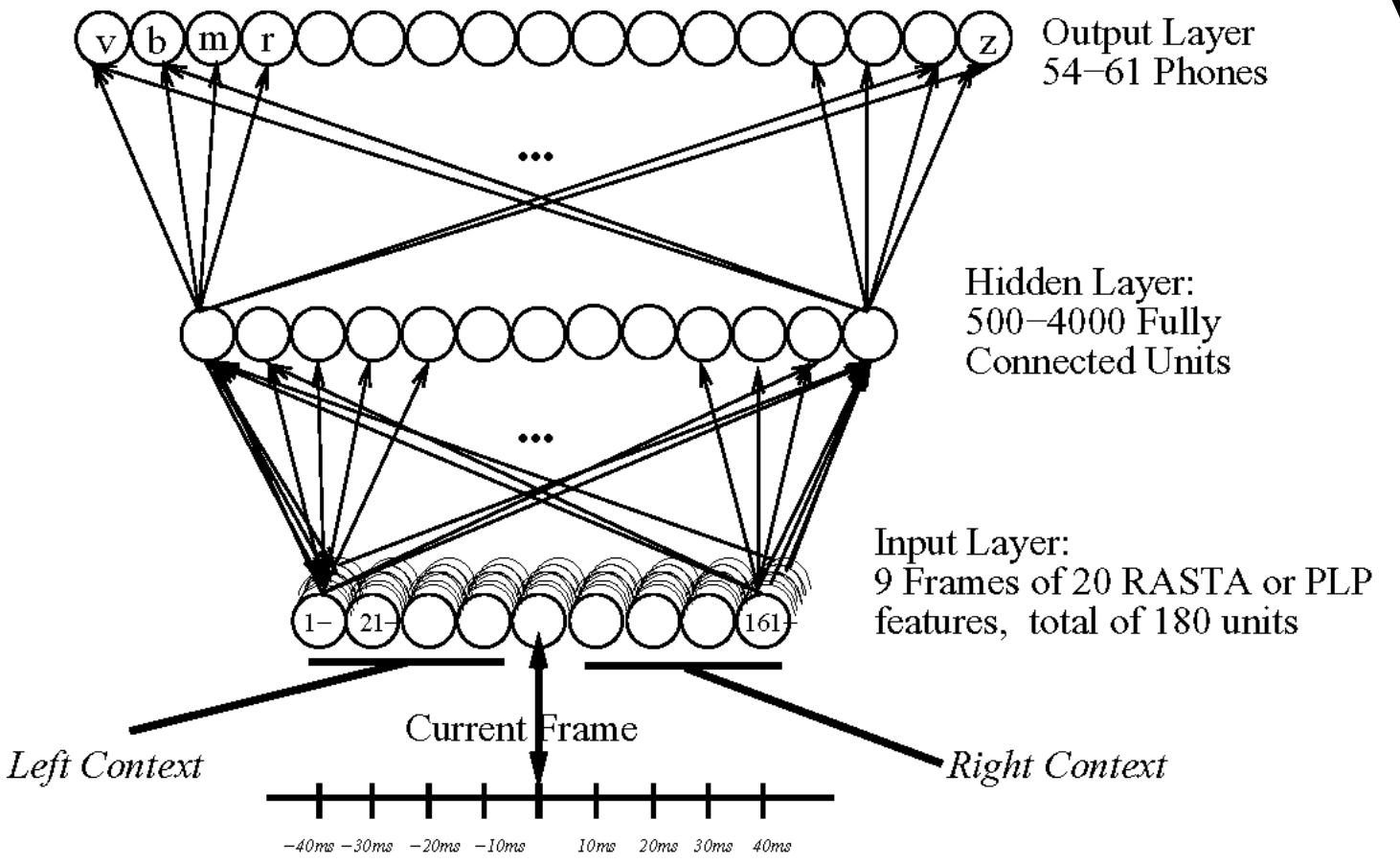
- Neural Network observation – probability - estimator
  - The Hybrid HMM-MLP approach
    - The two terms on the right-hand can be directly computed from the MLP; the numerator is the output of the MLP, and the denominator is the total probability of a given state, summing over all obervations (i.e., the sum over all t of $\sigma_j(t)$)
    - Thus although we cannot directly compute $P(o_t|q_j)$, we can use $\frac{P(o_t|q_j)}{P(o_t)} = \frac{P(q_j|o_t)}{P(q_j)}$ to compute $\frac{P(o_t|q_j)}{P(o_t)}$, which is known as a **scaled likelihood** (the likelihood divided by the probablility of the observation).
    - In fact, the scaled likelihood is just as good as the regular likelihood, since the probablility of the observation $P(o_t)$ is a constatnt during recognition and doesn't hurt us to have in the equation.

Computing Acoustic Probabilities

- Neural Network observation – probability - estimator
  - The Hybrid HMM-MLP approach
    - The error-back-propagation algorithm for training an MLP requires that we know the correct phone label q_j for each observation o_t.
    - Given a large training set of observations and correct labels, the algorithm iteratively adjusts the weights in the MLP to minimize the error with this training set.



Computing Acoustic Probabilities



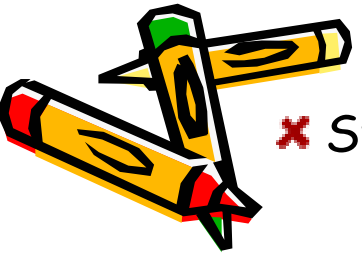
Training A Speech Recognizer

- Evaluation Metric
  - Word Error Rate
    - Compute minimum edit distance between hypothesized and correct string.
    - WER is defined :

$$\text{Word Error Rate} = 100\frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Total Words in Correct Transcript}}$$

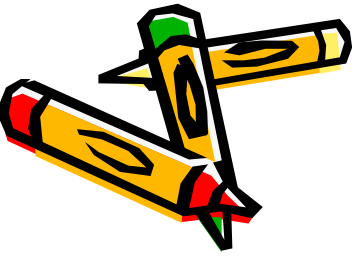
    - Ex : Correct: "I went to school yesterday."
      Hypothesis: "Eye went two yeah today."
      3 substitutions, 1 deletion, 1 insertion

      Word Error Rate = 100%.
    - State-of-the-art : 20% WER on natural-speech task.



Training A Speech Recognizer

- Models to be trained:
  - Language model: $P(w_i \mid w_{i-1} w_{i-2})$
  - Observation likelihoods: $b_j(o_t)$
  - Transition probabilities: a_{ij}
  - Pronounciation lexicon: HMM state graph structure
- Training data:
  - Corpus of speech wavefiles + word-transcription
  - Large text corpus for language model training
  - Smaller corpus of phonetically labeled speech



Training A Speech Recognizer

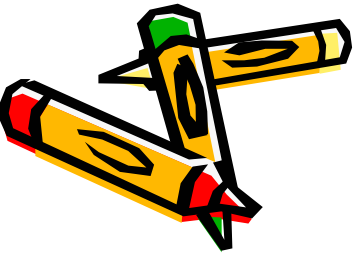
- N-gram language model :
  - Counting N-gram occurrences in large corpus.
  - Smoothing and normalizing the counts.
  - About the corpus...
    - The larger training corpus accurate the models.
    - text → less space → half a billion words of text.
- HMM lexicon structure :
  - Built by hand, by taking an off-the-shelf pronunciation dictionary. ex : PRONLEX , CMUdict

  - Uniphone , diphone or triphone ?

Training A Speech Recognizer

- HMM parameters:
  - About the corpus...
    - Labeled speech → Supply a correct phone label for each frame.
  - Initial estimate:
    - Transition probabilities and observation probabilities → All states are equal.
    - [Gaussian] Means and variances → Use means and variances of entire training set. (Less important)
    - [MLP] A hand-labeled bootstrap is the norm. ??

(More important)

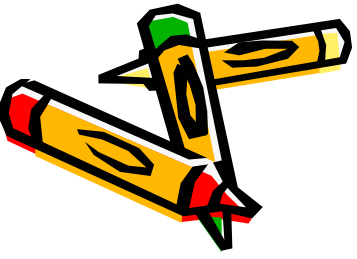


Training A Speech Recognizer

- HMM parameters:
  - Calculate ***a*** and ***b*** probability
    - [Gaussian] Forward-backward algorithm.
    - [MLP] Forced Viterbi alignment.
  - Forced Viterbi alignment
    - It takes as input the correct words in an utterance, along with the spectral feature vectors.
    - It produces the best sequence of HMM states, with each state aligned with the feature vectors.
    - It is thus a simplification of the regular Viterbi decoding algorithm, since it only has to figure out the correct phone sequence, but doesn't have to discoverthe word sequence.
    - It is called **forced** because we constrain the algorithm by requiring the best path to go through a particular sequence of words.

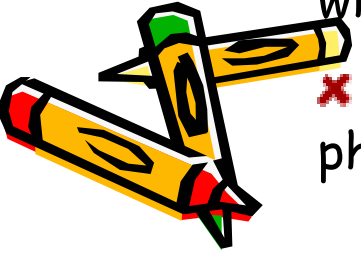
Training A Speech Recognizer

- HMM parameters:
  - Forced Viterbi alignment
    - It still requires the Viterbi algorithm since words have multiple pronunciations, and since the duration of each phone is not fixed.
    - The result of the forced Viterbi is a set of features vectors with "correct" phone labels, which can then be used to retrain the neural netword.
    - The counts of the transitions which are taken in the forced alignments can be used to estimate the HMM transition probabilities.



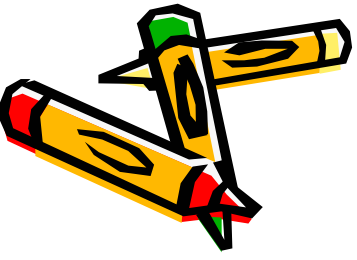
Waveform Generation for Speech Synthesis

- Text-To-Speech (TTS) System :
  - Text-To-Speech
    - Output is a phone sequence with durations and a FO pitch contour.
    - This specification is often called the **target**, as it is this that we want the synthesizer to produce.
  - Waveform concatenation
    - Such **concatenative synthesis** is based on a database of speech that has been recorded by a single speaker.
    - This database is segmented into a number of short units, which can be phones or words.
    - Simplest synthesizer : Phone unit and Single unit for each phone in the phone inventory.



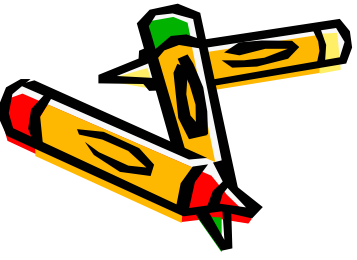
Waveform Generation for Speech Synthesis

- Text-To-Speech (TTS) System :
  - Waveform concatenation
    - Single phone don't produce good quality speech.
    - The triphone models are a popular choice, because they cover both the left and right contexts of a phone.
    - But there are too many combinations for triphones.
    - Hence **diphones** are often used in speech synthesis.
    - Diphone units normally start half-way through the first phone and end half-way through the second.
    - This because it is known that phones are more stable in the middle than at the edges.



Waveform Generation for Speech Synthesis

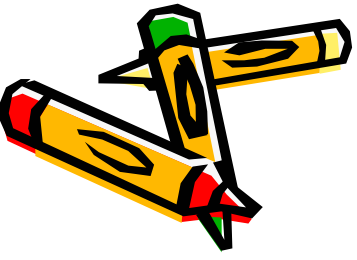
- Text-To-Speech (TTS) System :
  - Pitch and Duration Modification
    - Since the pitch and duration (i.e., the prosody) of each phone will be the same. ← disadvantage
    - So we use **signal processing techniques** to change the prosody of the concatenated waveform.
    - LPC model separates pitch from spectral envelope to modify pitch and duration:
      - Modify pitch:
        - Generate pulses in desired pitch.
        - Re-excite LPC coefficients.
        - Modified wave.



Errata ! page 275, line 12 : Add a "." after "spectral envelope" .

Waveform Generation for Speech Synthesis

- Text-To-Speech (TTS) System :
  - Pitch and Duration Modification
    - Modify duration: Contract/expand coefficient frames.
  - TD-PSOLA: frames centered around pitchmarks.
    - Change pitch: Make pitchmarks closer together/further apart.
    - Change duration: Duplicate/leave out frames.
    - Recombine: Overlap and add frames.
  - Problems with speech synthesis
    - 1 example/diphone is insufficient .
    - Signal processing → distortion.
    - Subtle effects not modeled.



Waveform Generation for Speech Synthesis

- Text-To-Speech (TTS) System :
  - Unit Selection
    - Collect several examples/unit with different pitch/duration/linguistic situation.
    - Selection method:
      - FO contour with 3 values/phone, large unit corpus.
        - Find candidates (closest phone, duration & FO) rank them by target cost (closeness).
        - Measure join quality of neighbour candidates rank joins by concatenation cost.
      - Pick best unit set: More natural speech.

Errata ! page 276, line -13 : Add a "." after "naturally occurring speech"

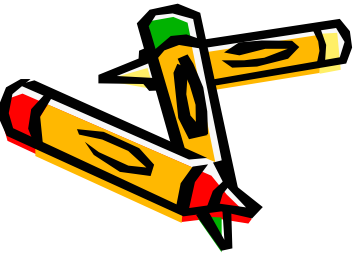
Errata ! page 277, line 7 : Change "a utterance" to "an utterance".

Human Speech Recognition

- The ideas of speech recognition from Human :
  - Signal processing algorithms like PLP inspired by human auditory system.
  - lexical access has common properties:
    - Frequency:
      - N-gram language models, human lexical access is sensitive to word **frequency**.
      - High-frequency spoken words are accessed faster or with less information than low-frequency words.
    - Parallelism:
      - Multiple words are active at the same time.
    - Neighborhood effects:
      - Words with large frequency-weighted neighborhoods are accessed slower than words with less neighbors.

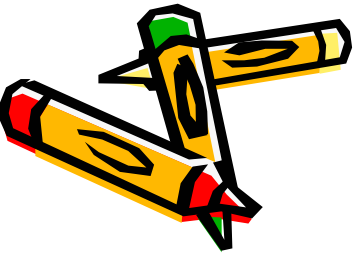
Human Speech Recognition

- The ideas of speech recognition from Human :
  - lexical access has common properties:
    - Cue-based processing:
      - Speech input is interpreted by integrating cues at many different levels.
      - Human perception of individual phones is based on the integration of multiple cues:
        - Acoustic cues: such as formant structure or the exact timing of voicing.
        - Visual cues: such as lip movement.
        - Lexical cues: such as the identity of the word in which the phone is placed. ex:Phoneme restoration effect.
        - Semantic word association cues: Words are accessed more quickly if a semantically related word has been heard recently.
        - Repetition priming cues: Words are accessed morequickly if they themselves have just been heard.



Human Speech Recognition

- The ideas of speech recognition from Human :
  - Difference between ASR models and human speech recognition:
    - time-course of the model:
      - ASR decoder return the best sentence at the end of the sentence.
      - But human processing is **on-line**:
        - People incrementally segment an utterance into words and assign it an interpretation as they hear it..
      - Close shadowers:
        - People who are able to shadow (repeat back) a passage as they hear it with lags as short as 250 ms.
        - When these shadowers made errors, they were syntactically and semantically appropriate with the context, indication that word segmentation, parsing, and interpretation took place within these 250 ms.



Human Speech Recognition

- The ideas of speech recognition from Human :
  - Difference between ASR models and human speech recognition:
    - Other cues:
      - Many other cues have been shown to play a role in human speech recognition but have yet to be successfully integrated into ASR.
      - The most important class of these missing cues is **prosody**.
      - Most multisyllabic English word tokens have stress on the initial syllable, suggesting in their metrical segmentation strategy (MSS) that stress should be used as a cue for word segmentation.

Errata ! page 278, line -10：Change “a utterance” to “an utterance”.

Errata ! page 279, line 2：Delete extraneous closing paren ”) ”.

Errata ! page 279, line -9：Replace "wound" with "sound".

Bibliographical and Historical Notes

- At the begining :
  - 1920s
    - "Radio Rex", a dog, it would move when you call it.
  - late 1940s ~ early 1950s
    - Bell Labs: 10-digits recognizer, 97-99% accuracy by chosing the best pattern.
  - 1959
    - Fry & Denes: Phoneme recognizer at University College, London, could recognize 4 vowels & 9 consonants. The first system to use phoneme transition probabilities.

Bibliographical and Historical Notes

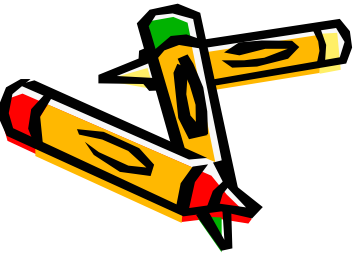
- HMM Used:
  - late 1960s ~ early 1970s
    - First important shifts: (Feature extraction)
      - Efficient FFT.
      - Application of Cepstral processing to speech.
      - Development of LPC for speech coding.
    - Second important shifts: (Handling warping→D.P.)
      - Stretching or shrinking the input signal to handle differences in speaking rate and segment length when matching against stored patterns.
      - Vintsyuk (1968)→Velichko & Zagoruyko (1970) and Sakoe & Chiba (1971)→Itakura (1975) combined the dynamic programming idea with the LPC coefficients.
      - The resulting system extracted LPC features for input signal and used dynamic programming to match them against stored LPC templates.

Bibliographical and Historical Notes

- HMM Used:
  - late 1960s ~ early 1970s
    - Third important shifts: (HMM used)
      - Statisticians: Baum and colleagues at the Institute for Defense Analyses in Princeton.
      - Baker's DRAGON system: James Baker learned of this work and applied HMMs to speech processing during his graduate work at CMU. (Using Viterbi decoding)
      - IBM's system: Frederick Jelinek, Robert Mercer, Lalit Bahl (influenced by the work of Shannon(1948)) applied HMMs to speech at the IBM Thomas J. Watson Research Center.
      - IBM:
        - N-grams
        - HMM-based part-of-speech tagging
        - statistical machine translation
        - the use of entropy / perplexity

Bibliographical and Historical Notes

- HMM Used:
  - late 1960s ~ early 1970s
    - HMM spread through the speech community:
      - Many research and development programs sponsored by the "Advanced Research Projects Agency" of the U.S. Department of Defense (ARPA).
      - The goal of this first program was to build speech understanding systems, and four systems were funded and compared against each other:
        - System development Corporation (SDC) system
        - Bolt, Beranek & Newman (BBN)'s HWIM system
        - Carnegie-Mellon University's Hearsay-II system
        - Carnegie-Mellon's Happy system
      - Happy system is a simplified version of HMM-based DRAGON system, and it was the best tested system.



Bibliographical and Historical Notes

- Recently:
  - mid-1980s
    - ARPA funded a number of new speech research programs.
    - Later speech recognition tasks moved away from read-speech to more natural domains:
      - Broadcast News (Hub-4)
      - CALLHOME and CALLFRIEND (Hub-5)
      - The Air Traffic Information System (ATIS)
    - Conference:
      - EUROSPEECH Conference
      - International Conference on Spoken Language Processing (ICSLP)
      - IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)