

# Choices for Acoustic Modeling

Professor Marie Roch

Huang et al. 9.4-9.5

## Phonetic modeling

- Small vocabulary → word/phrase models
- Large vocabulary →
  - word/phrase models are impractical
  - subword models are necessary
  - what issues?

## Phonetic modeling considerations

- *accuracy* – The model should represent what happens in different contexts.
- *trainability* – There must be enough data to train the unit.
  - i.e. syllable units might be nice, but some languages like English have a lot of them.
- *generalization* – We should be able to derive new words from the models.

## Phones as models?

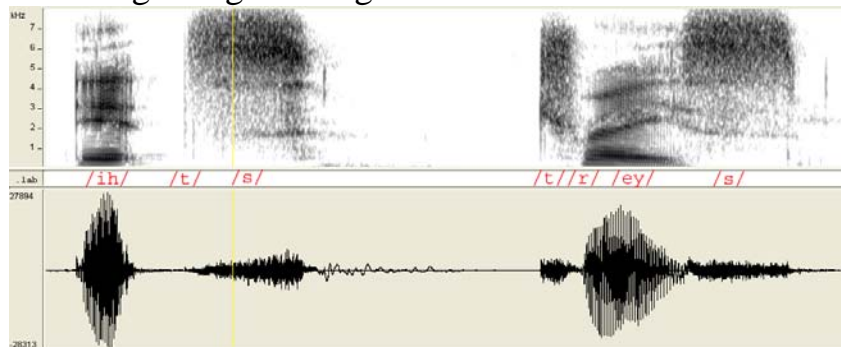
- Only about 50 in US English
- A few hundred sentences is sufficient training data.
- Phone models can't handle coarticulation.
  
- Phones are certainly *trainable*, and *accurate*, but we will see that they are not always *generalizable*.

## Syllables as models?

- Suitable for some languages:
  - About 50 syllables in Japanese.
  - Approximately 1,200 tone-dependent syllables in Mandarin Chinese.
- but not all:
  - English has about 30,000 syllables.

## Importance of context

- Suppose we only see /t s/ in training.
- Recognizing /t r/ might be difficult.



## Context dependence

- To help alleviate the problem from the last slide, we can use triphone models.
  - Create a model based upon the phoneme as well as portions of the preceding and following phonemes.
  - Permits us to model allophones quite easily.

## Triphone limitations

- Cross-word triphones are somewhat unpredictable.
  - We don't know what might follow "hope."
  - Suppose no /f/ after /p/ in training, but in test someone says "hope floats."
  - Some of the most serious coarticulation effects occur across words.

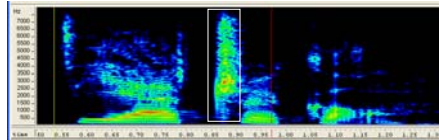
## Triphone limitations

- The same context can have different realizations:

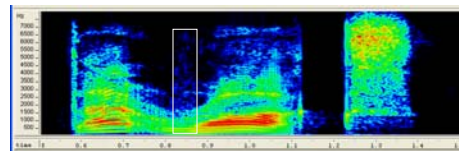
– /t/ in “theatrical” versus “that rocks”

- Both have the same context.
- First realization is almost a /ch/ while the second /t/ is almost extinct.

/t/ in “theatrical”



/t/ in “that rocks”



## Free stress versus bound stress

- Free stress: Stress can appear anywhere in a word.
- Bound stress: Stress is always in the same place.
  - Example language: French
    - *J'ai bien mangé*. Stress is always at the end of a word.

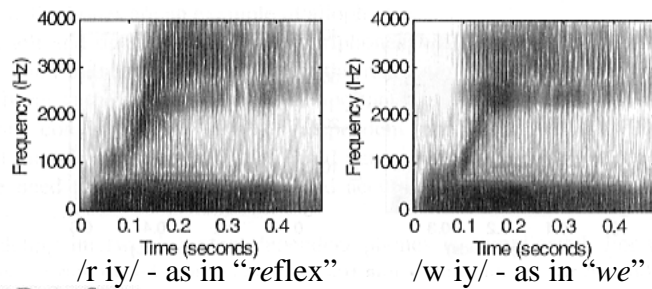


## Sentence level stress

- Sentence level stress is also possible:
  - You took out the trash, **didn't** you?
- However, it is difficult to model without high-level knowledge and is not part of most recognizers.

## Clustering acoustic-phonetic units

- Sometimes, a group of neighboring phones have similar coarticulatory effects
- i.e. liquids /r/ /w/



## Clustering acoustic-phonetic units

- Other similar prefixes/postfixes exist
  - i.e. the labial stops /b/ and /p/
- In many cases, it may make sense to merge the units:
  - triphone units: /b – ae – t/ and /p – ae – t/
  - could be merged to clustered triphone:  
/ [bp] – ae – t/

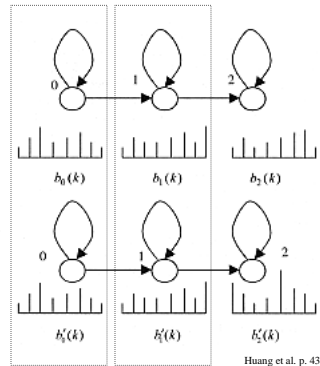
## Senones

- This has been generalized to a state-level clustering scheme.
- States with similar pdfs are tied.
- The clustered states are called senones and become the states of the HMMs.

## Senone example

- Consider the following two discrete HMMs

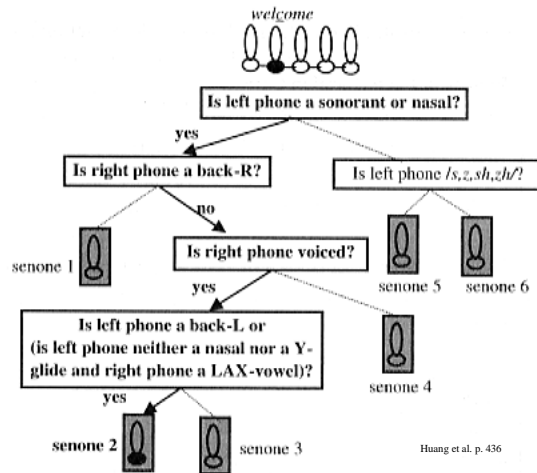
States 0 and 1 have similar pdfs and could be tied with other similar states to form a senone.



## Coping with unseen triphones

- American English > 100,000 triphones → trainability is an issue
- Clustering permits us to create generalizable models.
- How do we know what to cluster?

## Senone decision trees



Huang et al. p. 436

## Coping with unseen triphones

- Other possibilities exist for handling triphones for which there is not enough data.
- One such possibility is to create a context-independent unit and to interpolate.

## Performance of acoustical units

Units	Relative Error Reductions
Context-independent phone	Baseline
Context-dependent phone	+25%
Clustered triphone	+15%
Senone	+24%

Huang et al. p. 436

## Lexical baseforms

- Describes the transcription of a word into subword units.
- Issues
  - pronunciations due to dialects, i.e. “tomato”
  - coarticulation
    - across words, “you” /y uw/ versus “did you ...” /jh uh/
  - common contractions

## CMU Pronouncing Dictionary

- Over 100,000 entries
- 39 phonemes
- Transcription examples:

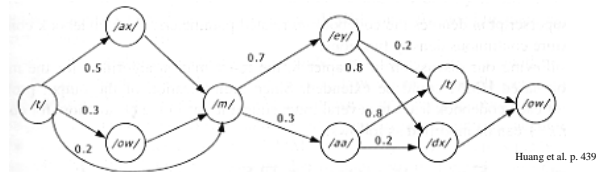
DOLPHIN	D AA1 L F AH0 N
TOMATO	T AH0 M EY1 T OW2
TOMATO(2)	T AH0 M AA1 T OW2
YOU'VE	Y UW1 V

## Proper names

- Proper names are difficult.
- Approximately 1-2 million names in the US
- Letter to sound conversion desirable
  - Rule based is impractical – too many exceptions
  - Machine learning techniques are appropriate:  
neural nets, HMMs, decision trees, etc.

## Probabilistic models for baseforms

- We can construct finite state machines to model variability:



– /t ax m ey t ow/, /t ow m ey t o/, /t ax m ey dx ow/, etc

## Probabilistic models for baseforms

- Produces relative error reductions of ~ 5-10%
- It is common to simply list the different baseforms instead of using a probabilistic model.

## Training isolated-word HMMs

- Gather all instances (tokens) of a specific word from the training material.
- Use the forward-backward algorithm as modified for multiple tokens to train an HMM with the “appropriate” number of states.

## Training sub-word models

- Suppose we wish to train a phone-level recognizer for connected digits.
- We would need:
  - dictionary
  - phone level HMMs
  - and labeled training speech.

## Dictionary and models

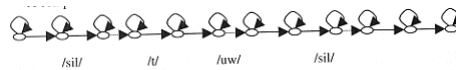
- Dictionary from CMU pronunciation dictionary.
  - Note pronunciation variants.
- Need to create models for the set of phonemes used.

EIGHT	EY1 T
FIVE	F AY1 V
FOUR	F AO1 R
NINE	N AY1 N
OH	OW1
ONE	W AH1 N
ONE(2)	HH W AH1 N
SEVEN	S EH1 V AH0 N
SIX	S IH1 K S
THREE	TH R IY1
TWO	T UW1
ZERO	Z IH1 R OW0
ZERO(2)	Z IY1 R OW0

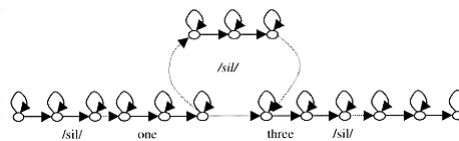
## Training

- Using the dictionary and null transitions, we can create networks of HMMs that correspond to the labeled training data:

- “two”



- “one three”

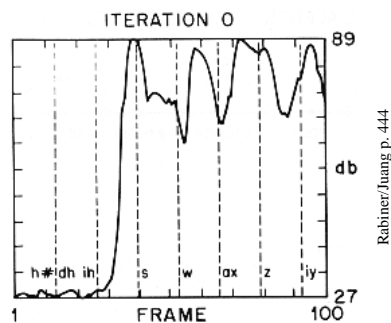


## Training the models

- The dead start
  - When we first begin, we do not know anything about the alignment.
  - Make a naive assumption that there is a linear alignment between the network and the training data:

## Dead start: “This was easy”

- Initialization: Equally segmented

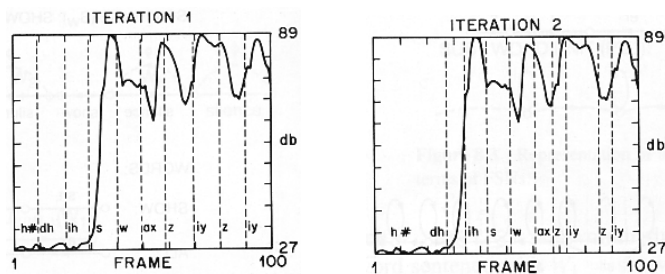


## Dead start

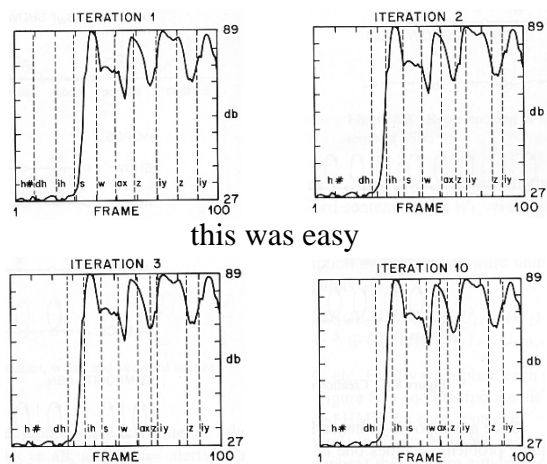
- From the dead start segmentation
  - Initialize the models given our algorithms for “good guesses”
  - Reestimate the models based upon the current segmentation using the EM algorithm (Baum-Welch/forward-backward algorithm).
  - Partition the data by performing a Viterbi decode.

## Training the models

- Using the new partitioning, reestimate the models and resegment.
- Repeat the process



# Subsequent iterations...



this was easy

Rabiner/Juang p. 444