Automatic Speech Recognition

Section 9.1 – 9.6 in Textbook

hannes@ru.is

T-725-MALV

Some Dimensions

- Vocabulary size (e.g. digits vs. free dictation)
- Kind of speech (e.g. words vs. continuous)
- Channel and noise (e.g. lab vs. outside)
- Speaker variation (e.g. native vs. foreign)

Typical Error Rates

2006 Data

Task	Vocabulary	Error Rate %
TI Digits	11 (zero–nine, oh)	.5
Wall Street Journal read speech	5,000	3
Wall Street Journal read speech	20,000	3
Broadcast News	64,000+	10
Conversational Telephone Speech (CTS)	64,000+	20

x4 for foreign accents
x4 for added automobile noise

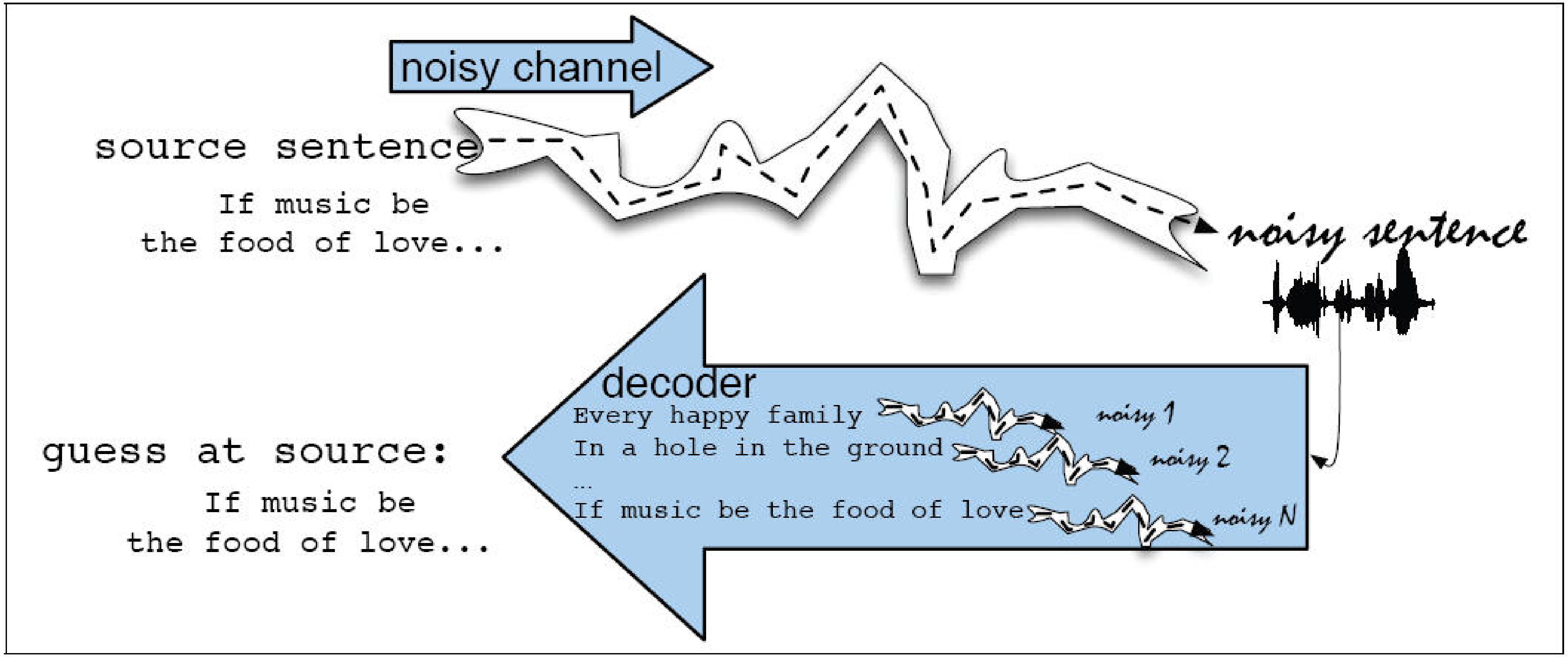
Focus on LVCSR

- Chapter focuses on
  - Large vocabulary (20 – 60 thousand words)
  - Continuous speech
  - Speaker independence
- We call this **LVCSR**
  - Large-Vocabulary Continuous Speech Recognition
- Dominant paradigm for LVCSR is the **HMM**

Noisy Channel Model

- **Goal**:
  - Build a model of the channel so we can figure out how it modifies the "true" sentence
- **Insight**:
  - We could run every possible sentence through the model and see which one matches the output
- **Issues**:
  - Exact match impossible → Use probabilities
  - Too many possibilities → Consider likely matches

Noisy Channel Model



Noisy Channel Model

- What is the most likely sentence out of all sentences in the language **L** given some acoustic input **O**?
- Treat acoustic input **O** as sequence of individual observations

  $O = o_1, o_2, o_3, \ldots, o_t$

- Define a sentence as a sequence of words:

  $W = w_1, w_2, w_3, \ldots, w_n$

Noisy Channel Model

- Probabilistic implication: Pick highest prob. W

$$\hat{W} = \underset{W \in L}{\arg\max}\, P(W \mid O)$$

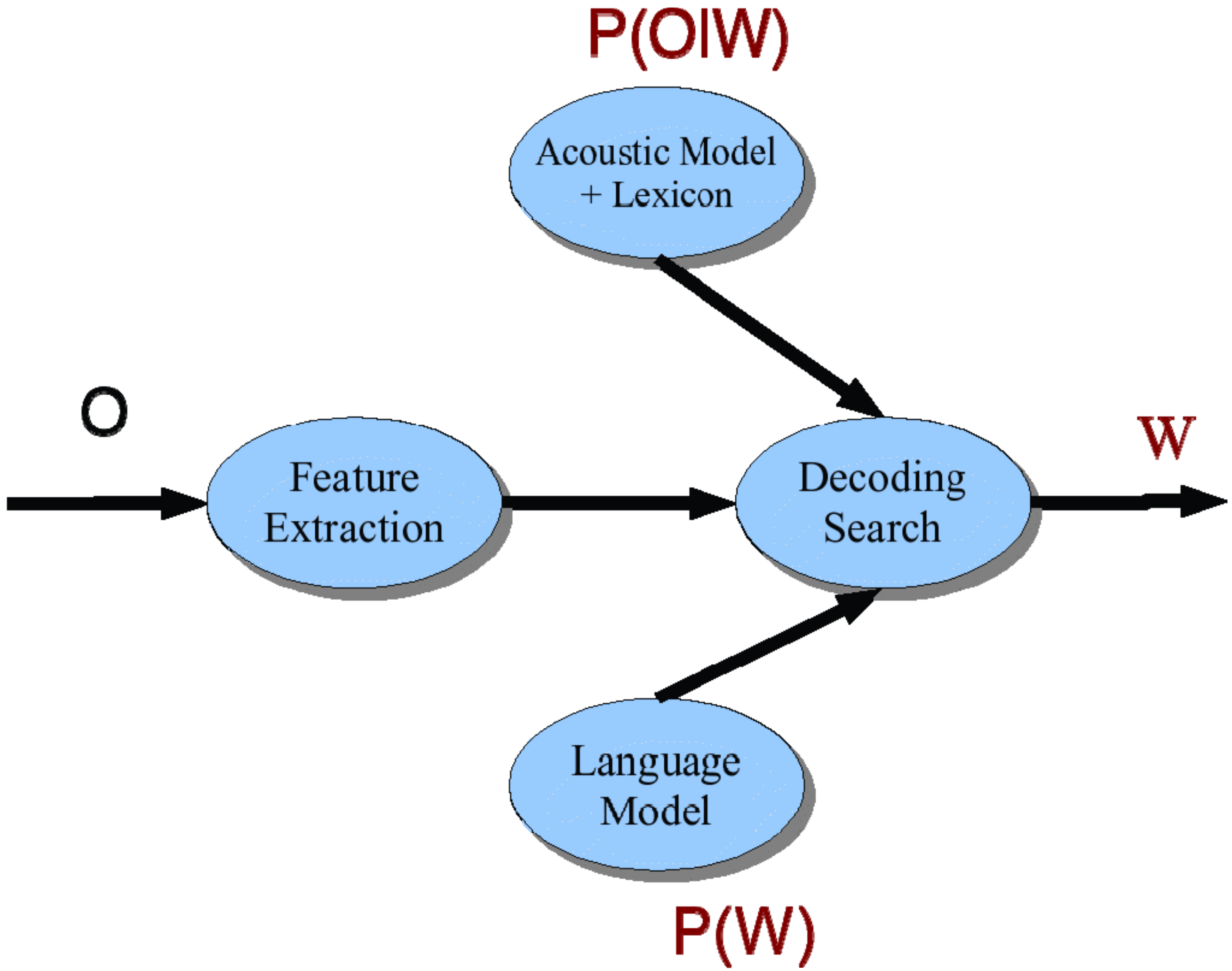
- We can use Bayes rule to rewrite this:

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

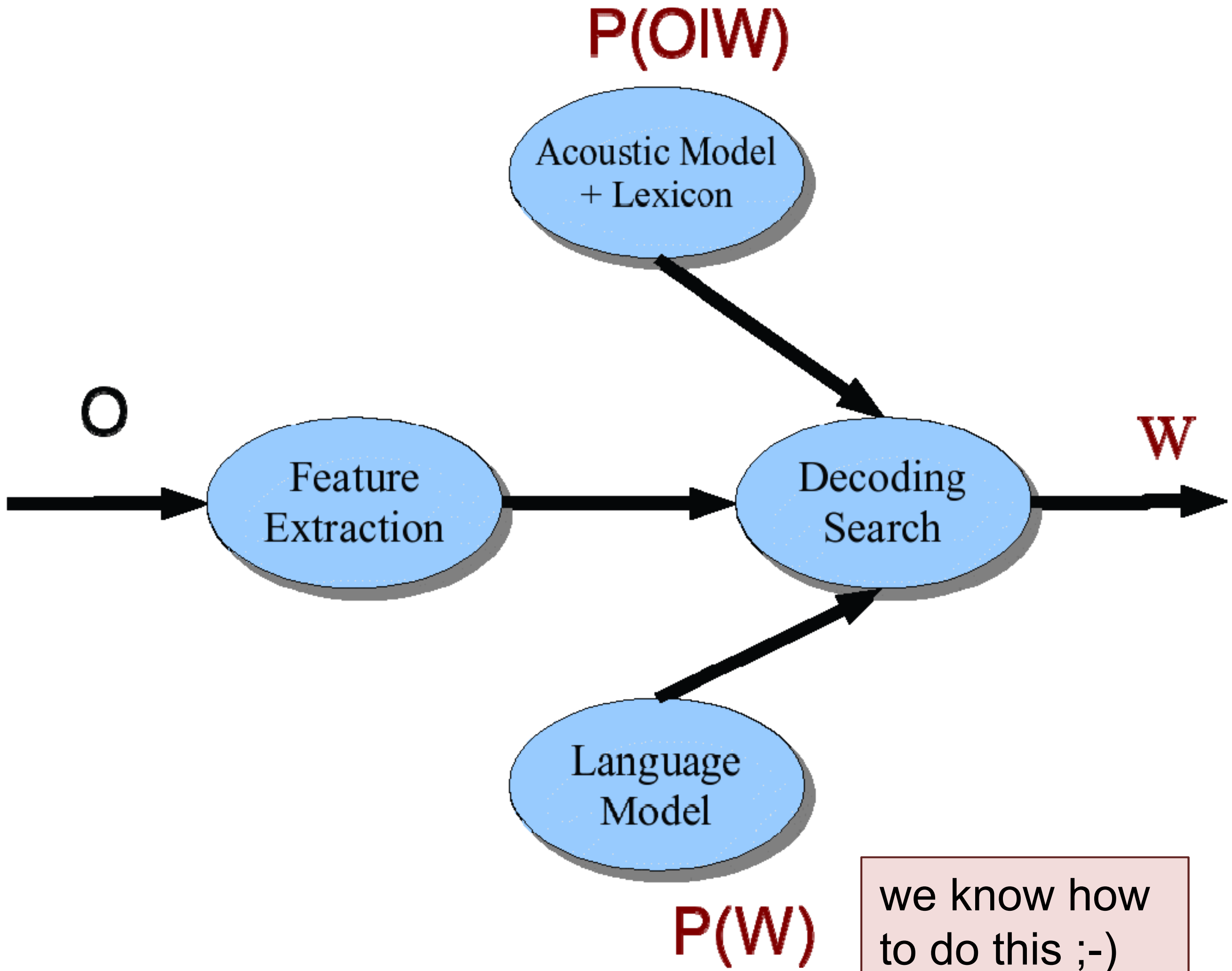
- Denominator same for each candidate W

$$\hat{W} = \underset{W \in L}{\arg\max}\, P(O \mid W)P(W)$$

Simple Architecture

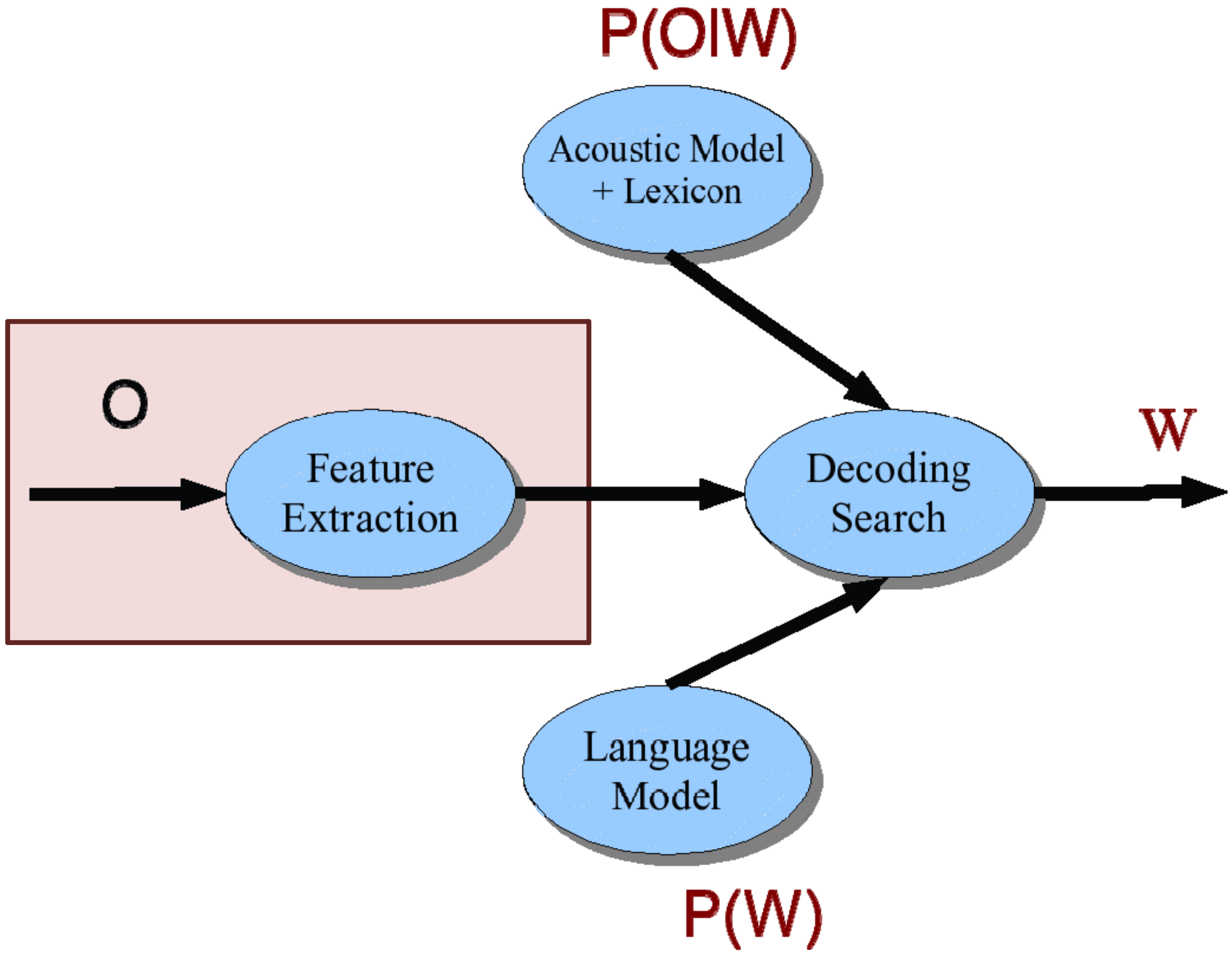


Simple Architecture



we know how
to do this ;-)

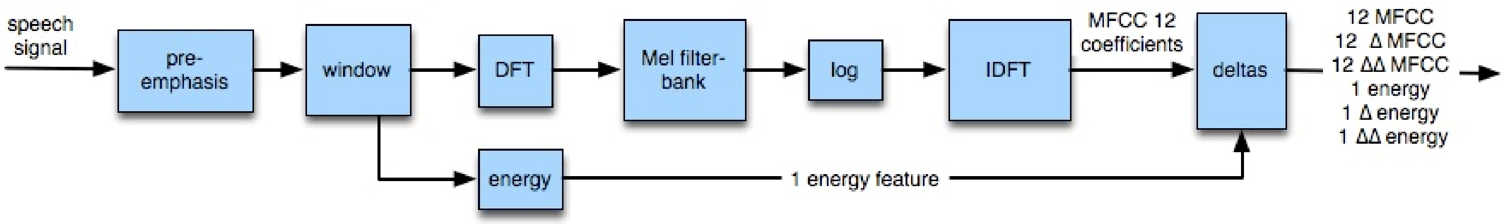
Simple Architecture



Feature Extraction

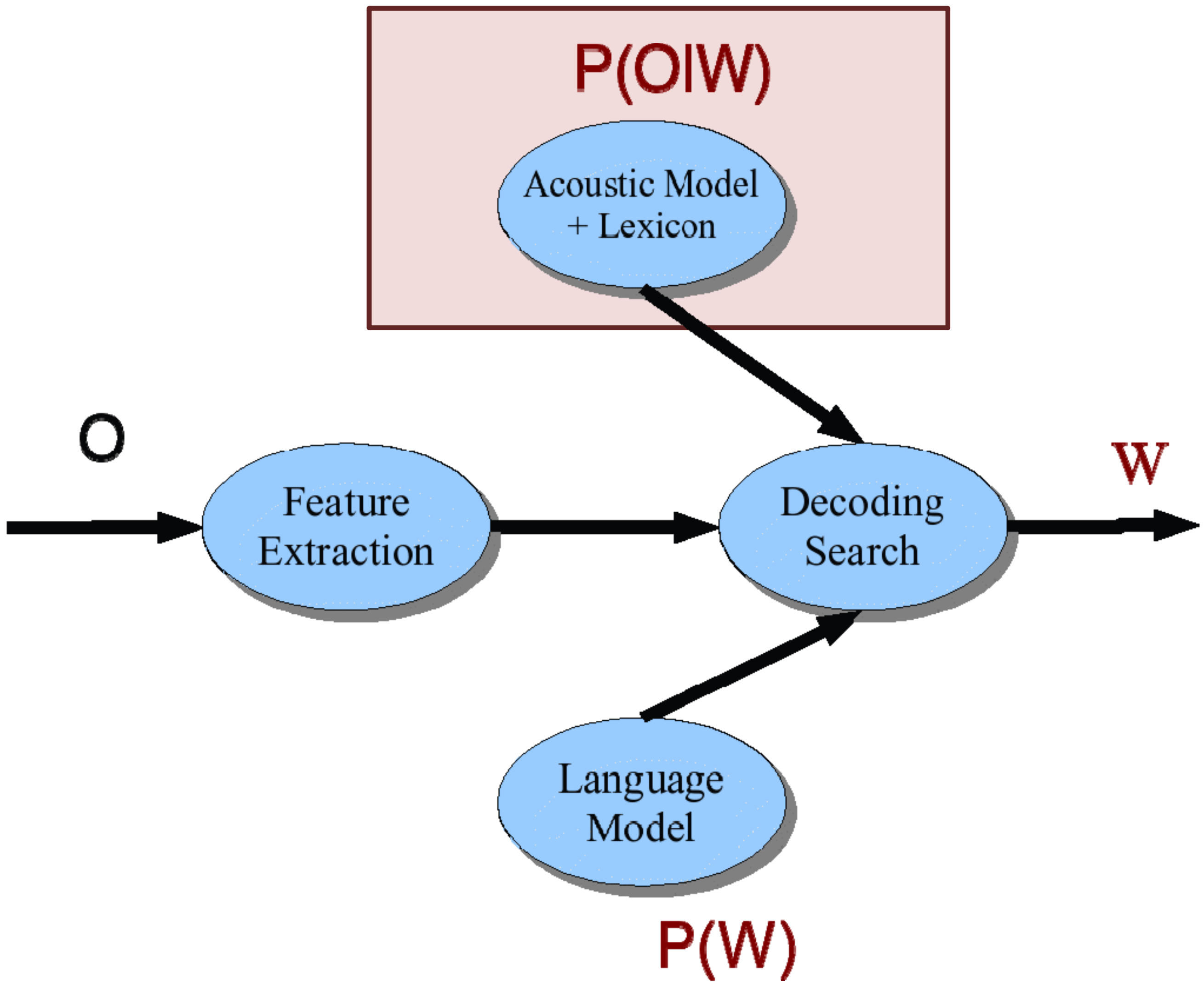
- Sample and Quantify the signal
- Boost high frequencies
- Focus on short windows
- Extract energy at different frequency bands
- Filter based on human hearing (mel filter)
- Extract coefficients for vocal tract filter separated from glottal source (cepstrum)

Feature Extraction



- 39 Features per 10 ms frame:
  - 12 MFCC features
  - 12 Delta MFCC features
  - 12 Delta-Delta MFCC features
  - 1 (log) frame energy
  - 1 Delta (log) frame energy
  - 1 Delta-Delta (log frame energy)

Simple Architecture

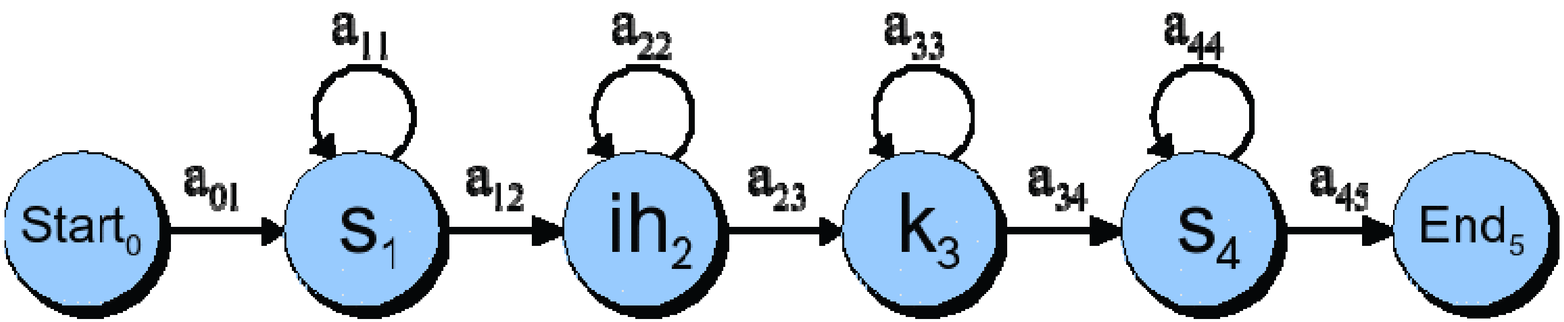


Lexicon

- A list of words
- Pronunciation in terms of phones
  - E.g. from CMU pronunciation dictionary
    CMU dictionary: 127K words
  - http://www.speech.cs.cmu.edu/cgi-bin/cmudict
- Lexicon represented as an HMM

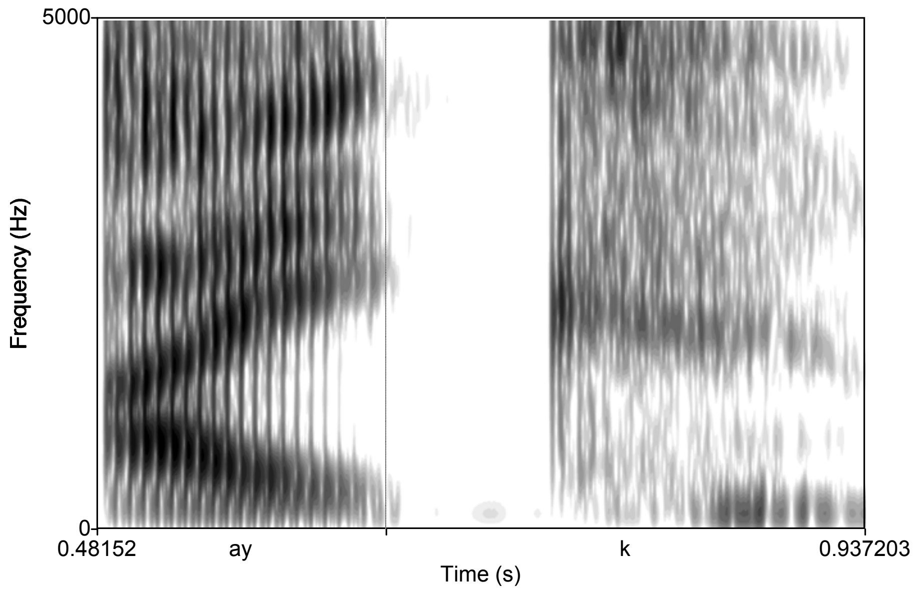
Lexicon

- HMMs for "six"

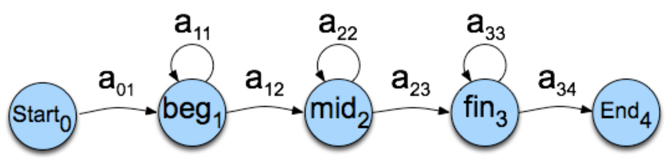


Lexicon

- Phones are not homogeneous

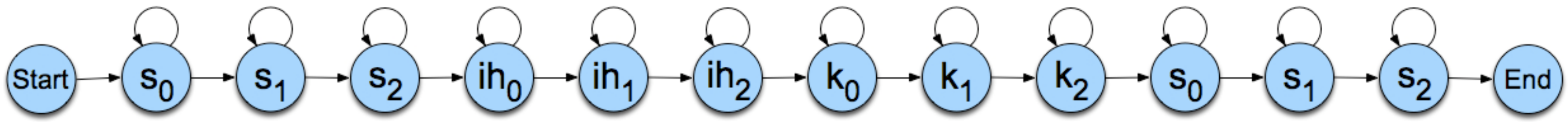


- Therefore represent subphones



Lexicon

- Resulting HMM for "six" with subphones

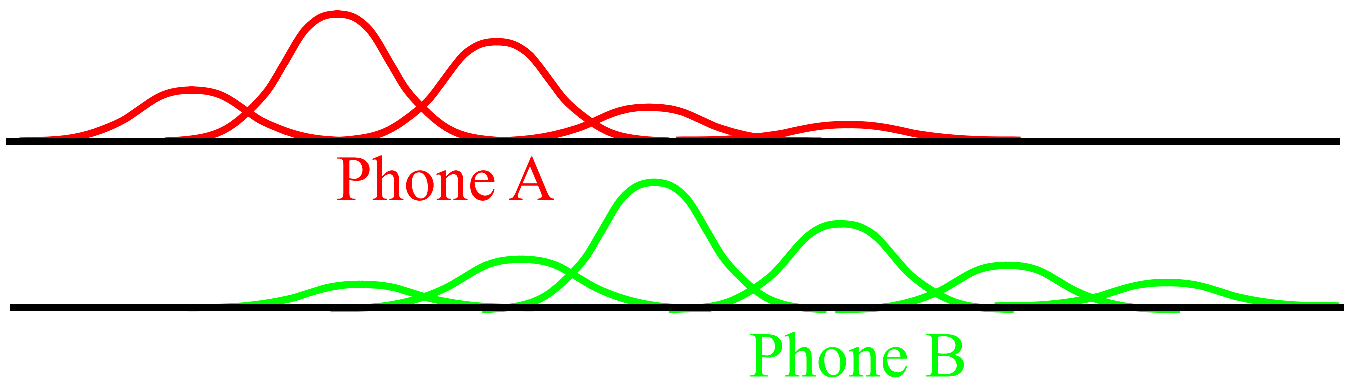


Acoustic Modeling

- For **each phone** in our lexicon we want to know what kind of **acoustic features** are associted with it
- The same phone may not always result in the same exact feature vector

Acoustic Modeling

- Instead of storing exact values, we store **Gaussian probability density functions**, mapping each possible feature to a likelihood of having been generated by this phone



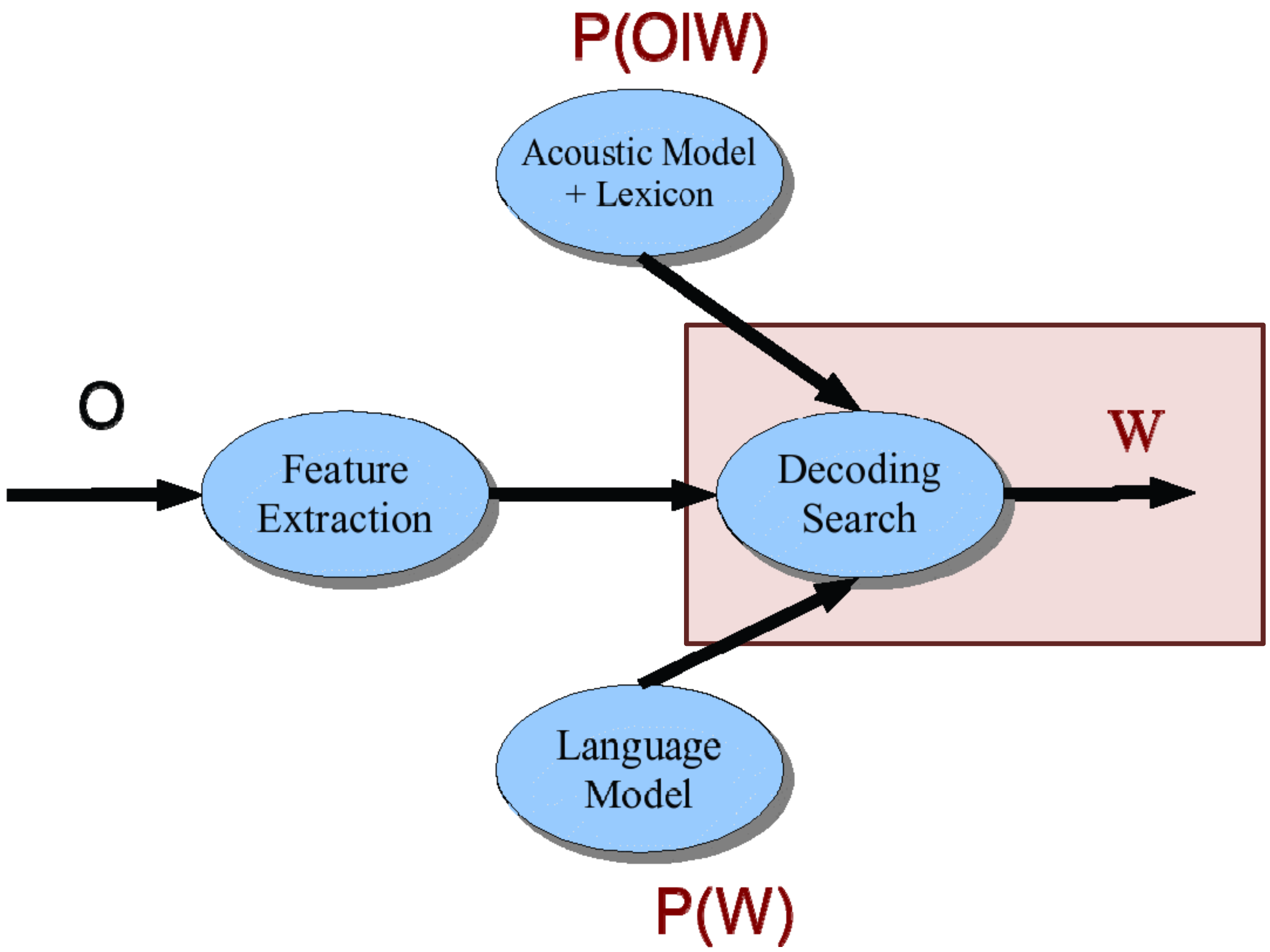
Acoustic Modeling

- Given a 39-dimensional **feature vector**, corresponding to the **observation** of one frame o_i and **phone q** we want to know:

$$p(o_i | q)$$

- We can now look this up in each phone's **Gaussian Mixture Model**

Simple Architecture



Decoding / Search

- The **observation sequence O** is a series of MFCC vectors
- The **hidden states** are the phones and words we wish to recover
- For a given phone/word string W in our lexicon, our job is to **evaluate P(O|W)**
- Intuition: how likely is the input to have been generated by that exact word string W?

Decoding / Search

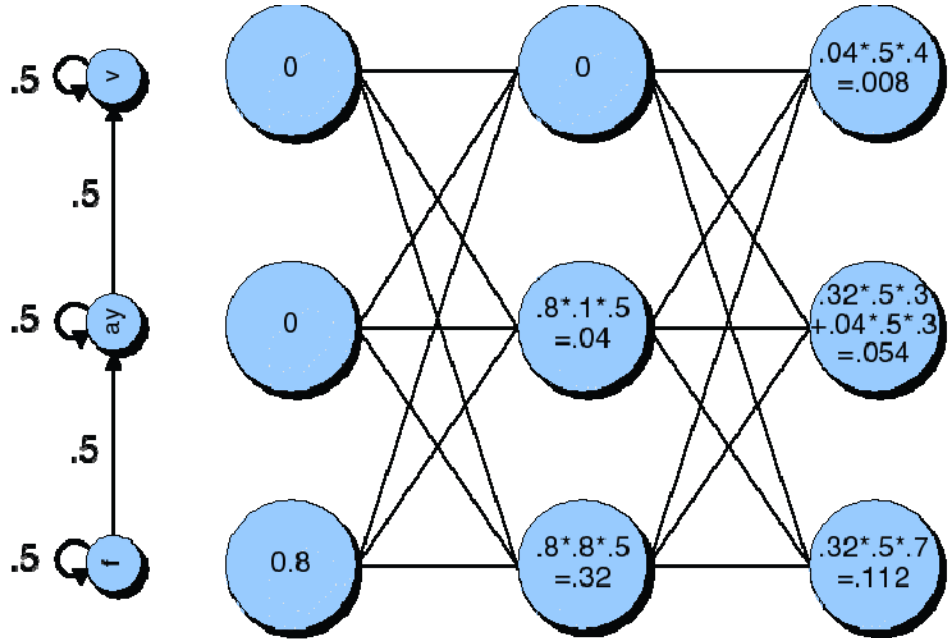
- We can construct word/sentence HMMs from phone HMMs and can therefore look-up the probability from our acoustic model

- Check all possible phone paths?

Decoding / Search

- The forward lattice for "five" (3 frames)

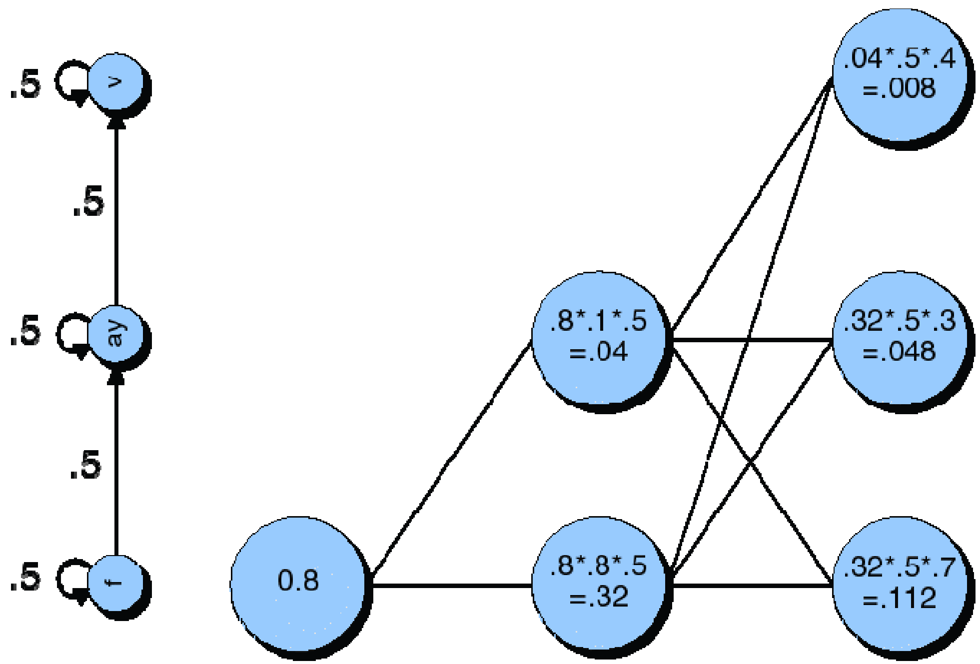
```
f ay ay ay ay v  v  v  v
f f  ay ay ay ay v  v  v
f f  f  f  ay ay ay ay v
f f  ay ay ay ay ay ay v
f f  ay ay ay ay ay ay ay v
f f  ay v  v  v  v  v  v
```



Decoding / Search

- The Viterbi trellis for "five" (3 frames)

```
f ay ay ay ay v  v  v  v
f f  ay ay ay ay v  v  v
f f  f  f  ay ay ay ay v
f f  ay ay ay ay ay ay v
f f  ay ay ay ay ay ay ay v
f f  ay v  v  v  v  v  v
```



(we need to prune search tree and be really "smart")