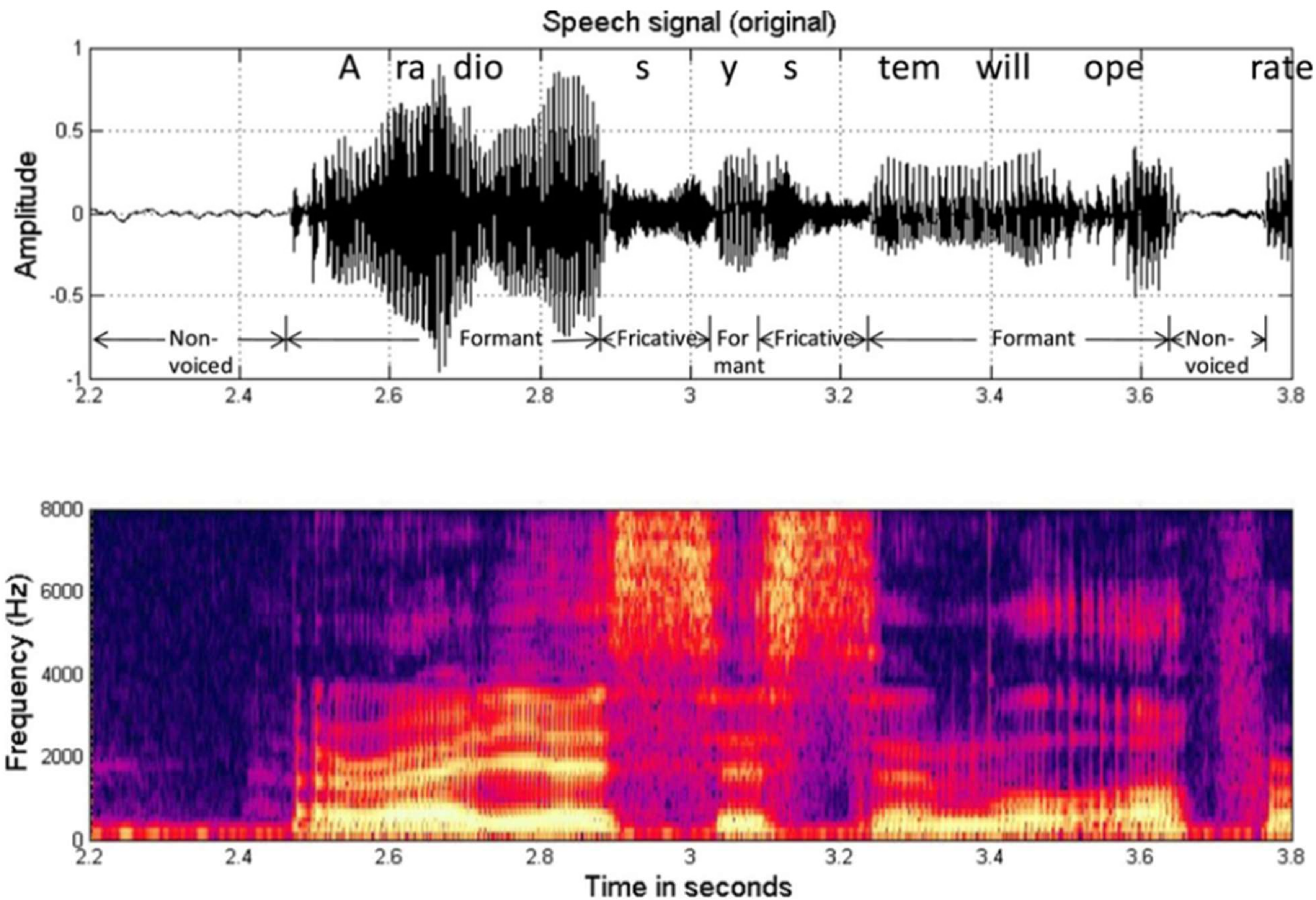
A Speech Signal in Time and Frequency Domains



Speech Recognition in Time or Frequency domain?

Time vs frequency domain:

- People are still working on time-domain recognition
- The best performing work are done in frequency domain

Advantages of working in frequency domain:

- Less data size
- Consistent feature without being affected by phase change
- Works in the same way as human ears

Difficulty in Speech Recognition

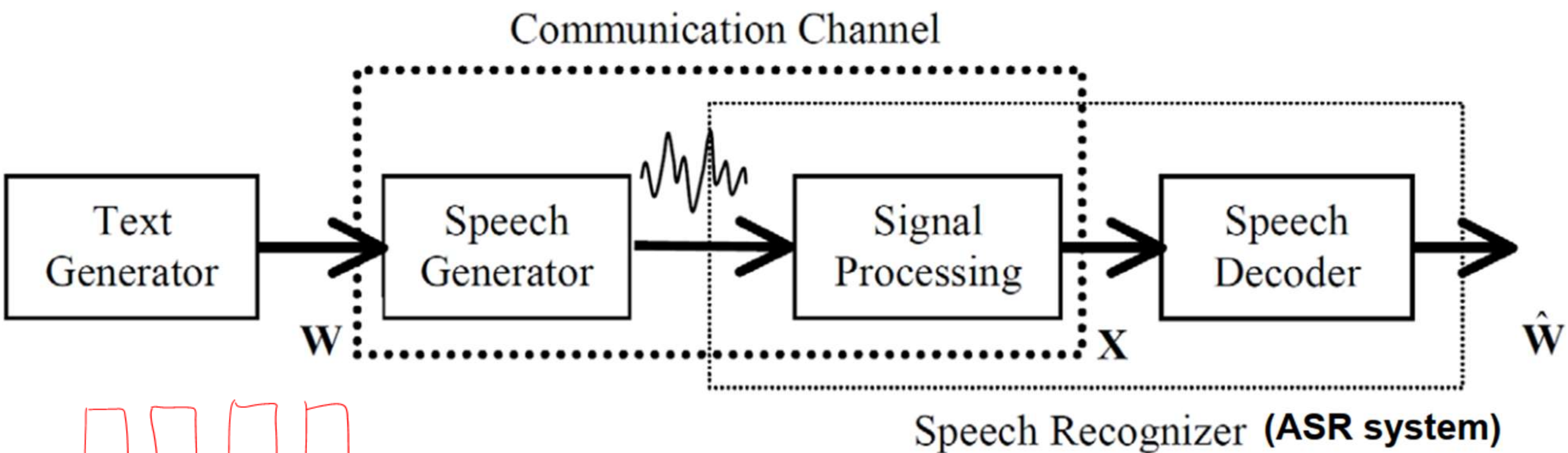
Nature of the signals:

- People says the same text in very diverse ways
  - Different length
  - Different pause
  - Different intonation
  - Different pitch
  - Different stress
- Recording is not perfect
  - Different background noises
  - Different channel responses (think about the equalizer)

Tools:

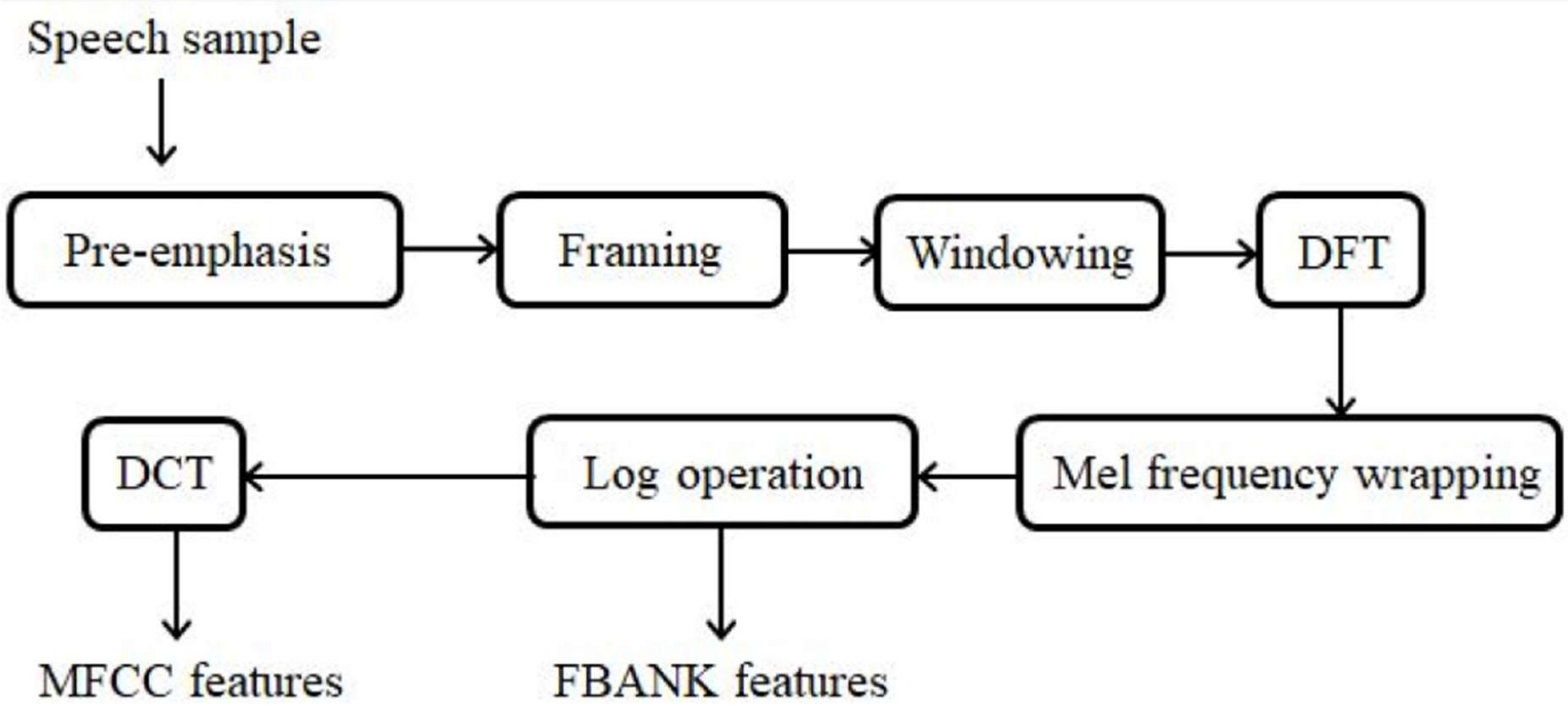
- Need to use many different mathematic tools and ideas
- Need to use many different programming tools and frameworks

A Simple Source-Channel Model for ASR



- W is the sequence of words from a certain speaker. It is called an utterance.
- X is the speech signal. in TD We can extract O, in FD the feature from X. O can have different format, but mostly in **frequency** domain.
- $\widehat{W}$ is the sequence we want to obtain given X or O.

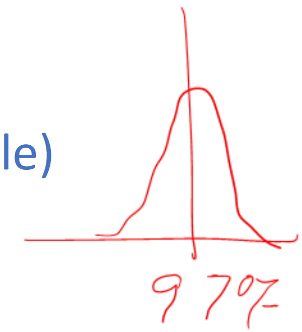
Frequency-domain Features Based on MFCC and Fbanks



- Each frame has 25 ms long.
- Skip from the previous frame is 10 ms.
- Use Mel frequency to better simulate the work of human ears.
- Use Log operation to reduce the range of changes.
- Use Discrete Cosine Transform to reduce the correlation between different dimensions. This is important for GMM we will use later.

Intro to Probability Theory

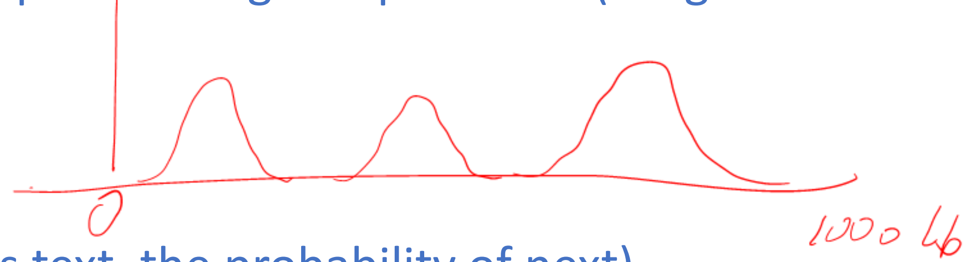
- Probability of an event (tossing a die)
- An event to a random variable (RV)
  - PFM for discrete RV (tossing a die)
  - PDF for continuous RV (measuring the body temperature of people)



- PDFs of common distributions
  - Uniform (quantization errors from analog signal to discrete)
  - Gaussian (white noise)
- PDFs based on parameters
  - Gaussian based on mean and variance (body temperature of patients)

GMM and HMM

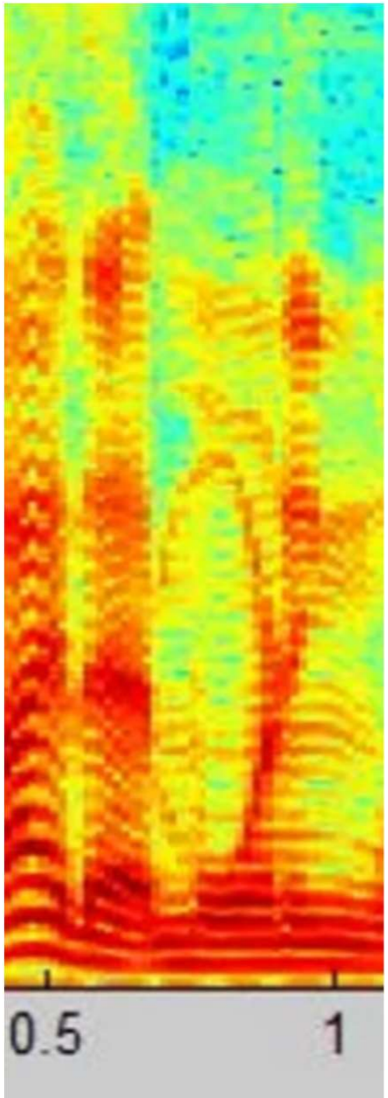
- Gaussian mixture model (GMM) for approximating complex PDFs (weight of cats)



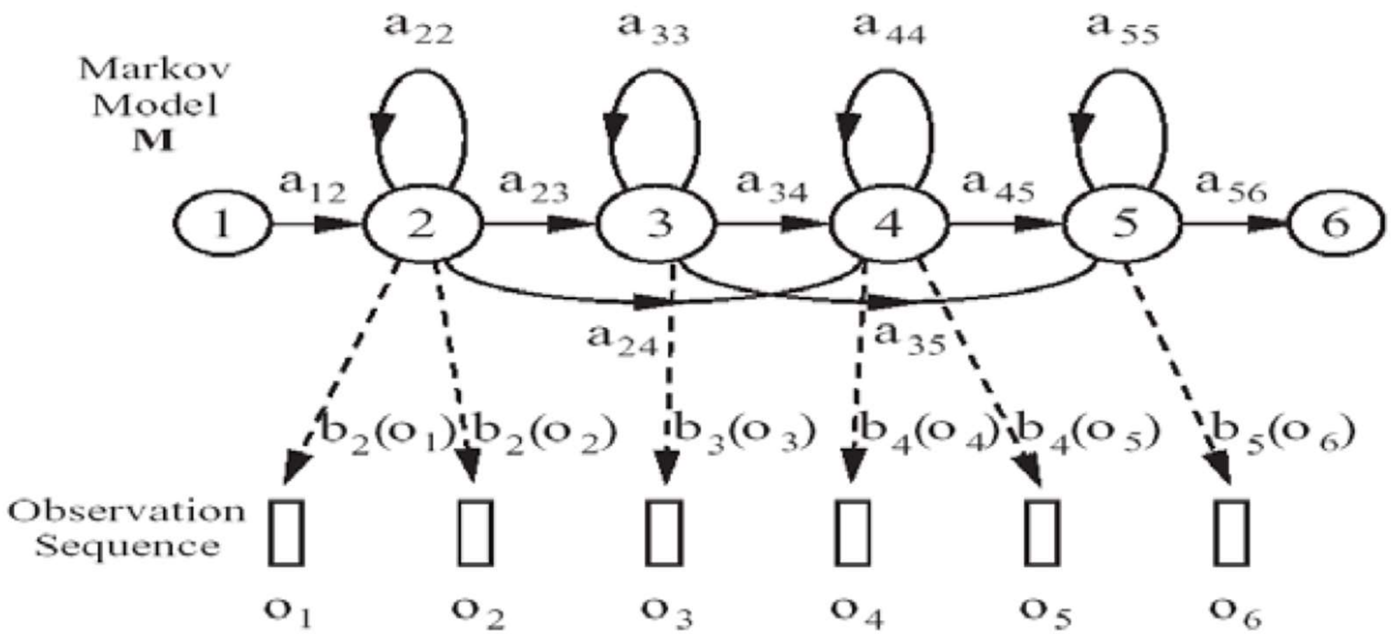
- Multiple RVs for a certain event (text)
- Conditional probability (given previous text, the probability of next)
- Bayesian theory ($p(a|b)$ to $p(b|a)$)
- Markov chain ($p(s2|s1, s0) = p(s2|s1)$)
- Hidden Markov model (HMM)
  - System described by states
  - State cannot be observed
  - State transitions in a non-backward manner
  - Each state transmits observable RV
  - Use observable RV to infer the state of the HMM

Phones and Triphones

- Phones are used to model the pronunciation of words in an utterance.
- Speech recognition is can be reformulated as phone recognition.
- We need to train the system to recognize different phones.
- These phones comes from a lexicon dictionary.
- To better model the transition of phones, we use tri-phones, the combination of three phones.



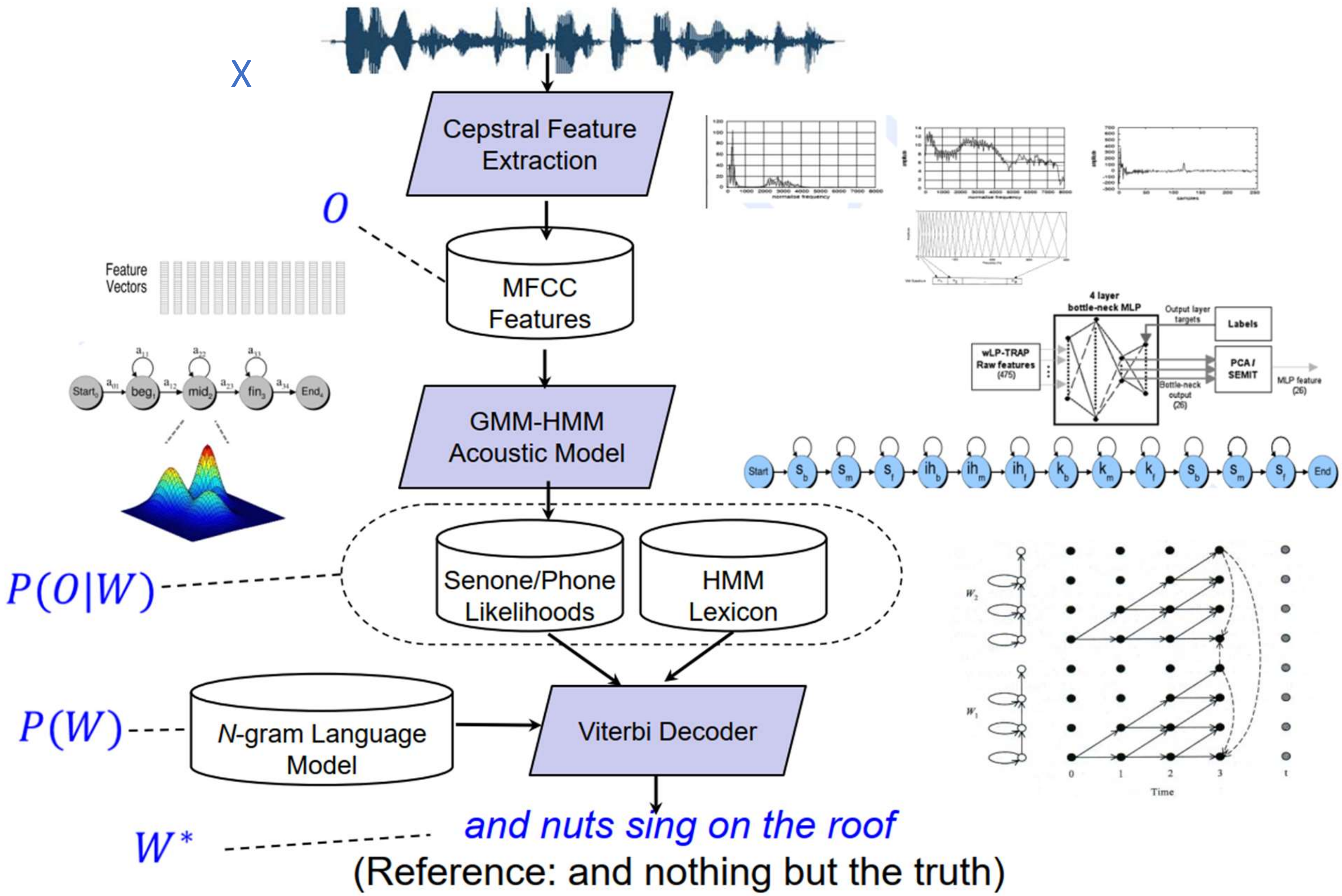
Using HMM to Match the Time-Varying Signals



Two Probability Models to Train

- Phones and triphones cannot be observed; they are modeled using an HMM each.
- Each HMM has several states to better model change of pronunciations.
- We need to train the probability model that describes the transition from one state to another using the training dataset.
- For a given state, the probability distribution of MFCC is described by using a GMM.
- We need to train this GMM using the training dataset as well. For simplicity, we assume the elements of feature vectors are independent. This is supported by using MFCC.
- The training is done by in an iterative approach, called EM (expectation maximization) as we don’t have well-aligned speech feature vs label (phones or triphones).
- We need to use the mono-phone model to estimate the aligned speech feature vs label pairs.
- Then, we move on to the tri-phones.

Traditional GMM-HMM-based ASR Systems



The Basic Formula for Speech Recognition

$$W_{opt} = \arg\min_{W \in \mathbf{W}} Risk(W|O)$$

$$= \arg\min_{W \in \mathbf{W}} \sum_{W' \in \mathbf{W}} Loss(W, W') P(W'|O)$$

For similar sequences, say, ok vs okay

$$\approx \arg\max_{W \in \mathbf{W}} P(W|O)$$

Assumption of Using the "0-1" Loss Function

$$P(W|O) = \frac{P(W,O)}{P(O)}$$

$$p(O|W) \cdot p(W)$$

$$= \arg\max_{W \in \mathbf{W}} \frac{p(O|W)P(W)}{p(O)}$$

$$= \arg\max_{W \in \mathbf{W}} \boxed{p(O|W)P(W)}$$

Linguistic Decoding

Feature Extraction & Acoustic Modeling

Language Modeling

Possible variations: speaker, pronunciation, environment, context, etc. and domain, topic, style, etc.

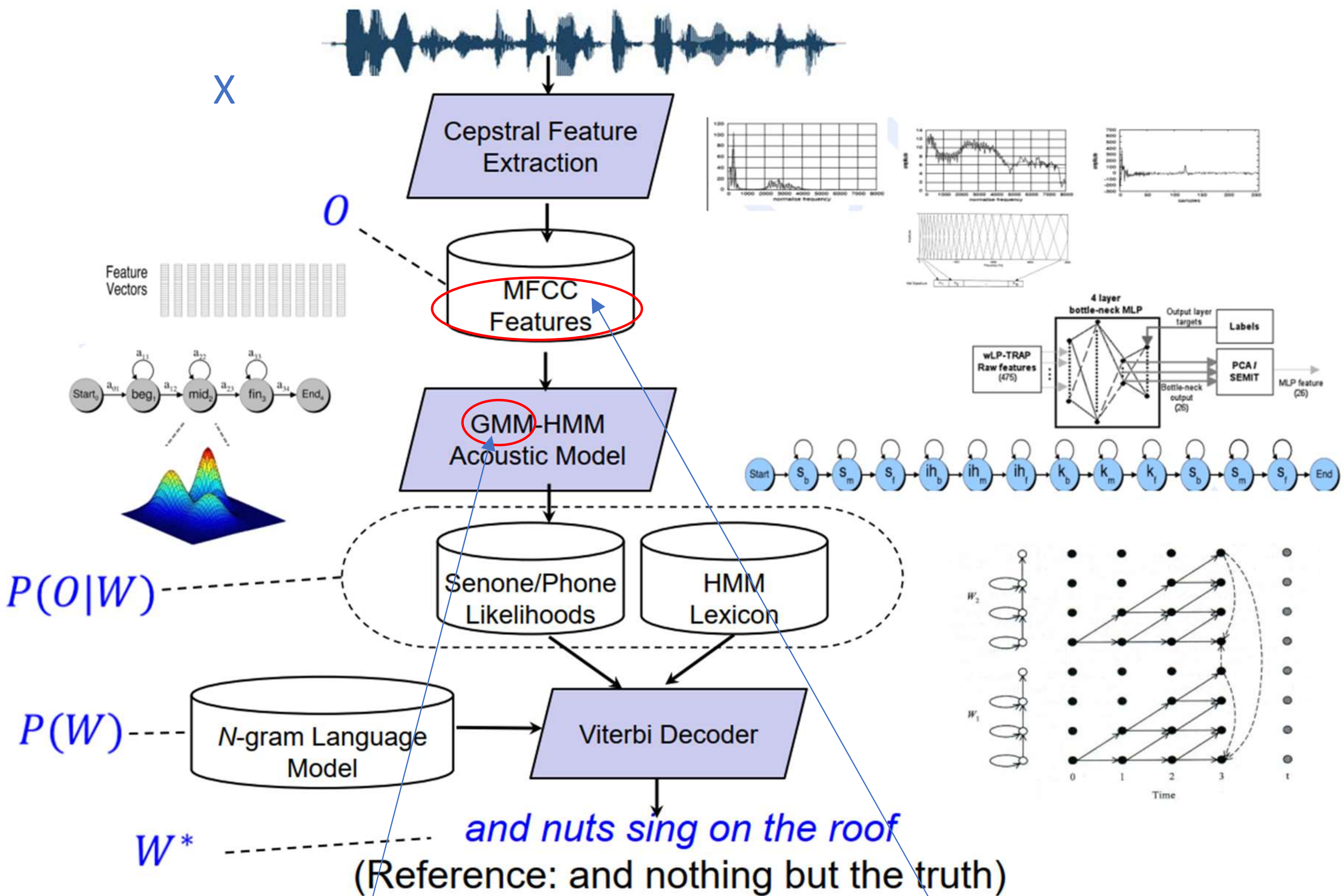
Decision Trees and Senone

- The total number of tri-phones is too big.
- We do not have resources to use all the tri-phones. (Both training and computation)
- Decision trees are used to significantly shrink the size of useful tri-phones.
- Decisions trees can be formed by linguists. They can also be learned, which is especially important to special applications or languages of little speakers.
- Each kept tri-phone is expressed using a HMM with three states excluding the start and end. These are called senone.
- To further reduce the number of parameters of the model, we can tie the GMM for some senones together---different states in the tri-phone HMM can share the same coefficients of GMMs.

Weighted Finite-State Transducer (WFST)

- When we know the phones, we can get the corresponding word using a weighted finite-state transducer
  - The input is a sequence of phones with weights
  - The output is a sequence of word(s)
  - Each word has a WFST
- There are four WFSTs used in the model:
  - G (grammar): words in words out
  - L (pronunciation lexicon): phones in words out
  - C (Context-dependency): tri-phones in phones out
  - H (HMM): HMM states in tri-phones out
- The above WFSTs can be combined to simplify the decoding.
- The HMM states are estimated based on $P(S|O)$ using a Viterbi algorithm. Here S is the state, and O is the feature vector.
- A latices is used for the decoding. Other algorithms, such as beam search can be used.

Modern DNN-HMM-based ASR Systems



By replacing MFCC with a higher dimension vector called FBank and GMM with a deep neural network (DNN), we have the modern DNN-HMM-based ASR systems.