The Statistical Approach to Speech Recognition

Frederick Jelinek
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD

March 2004

Terminology

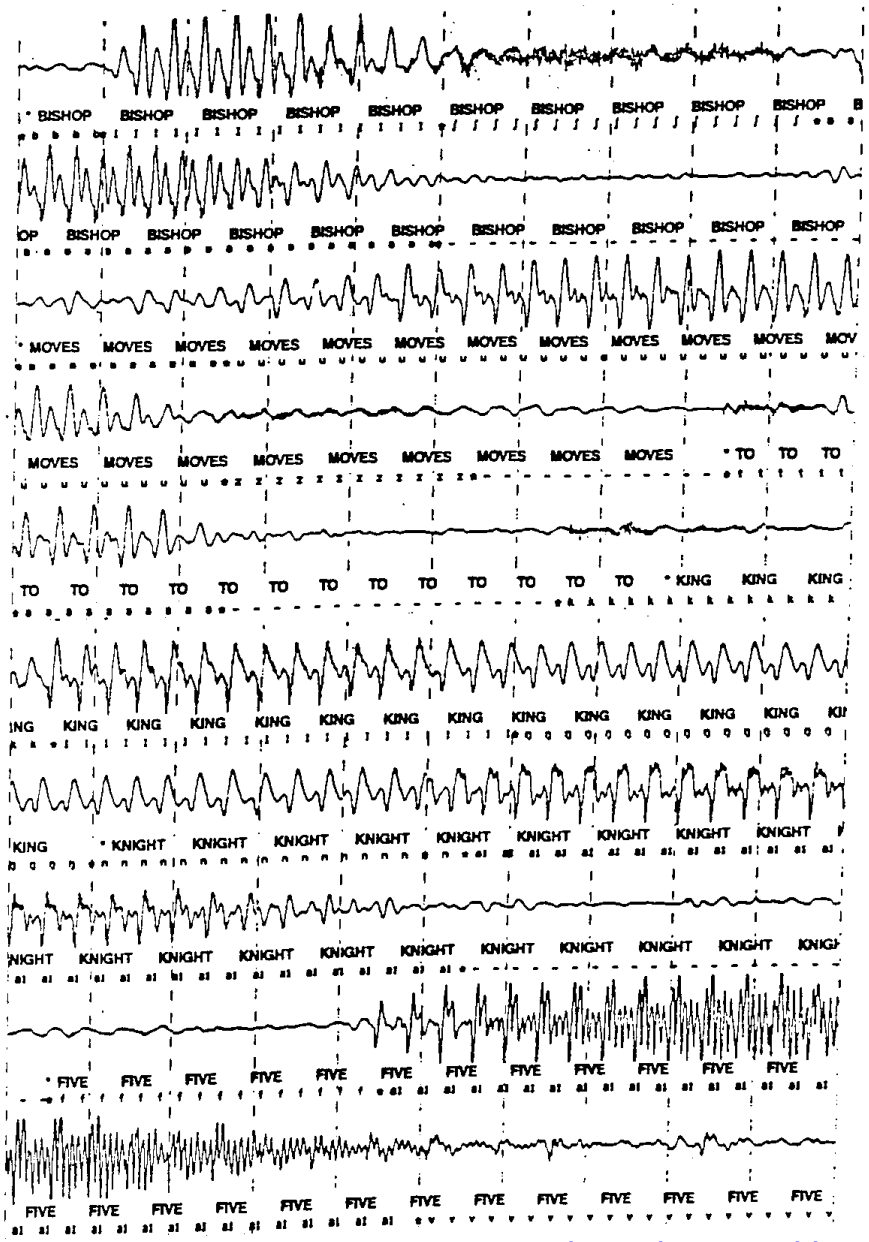
Speech recognition:

Automatic transcription of the sound of speech into text

Speech understanding:

Determination of intended meaning of observed speech

Bishop Moves to King Knight Five



A mathematical formulation

- Let A denote the acoustic evidence (data)
  - A is a sequence of symbols taken from some (possibly very large) alphabet A :

$$\mathbf{A} = a_1, a_2, ..., a_m \qquad a_i \in \mathcal{A}$$

- Let

$$\mathbf{W} = w_1, w_2, ..., w_n \qquad w_i \in \mathcal{V}$$

denote a string of n words, each belonging to a fixed and known vocabulary V.

A mathematical formulation (Cont.)

- If $P(W|A)$ denotes the probability that the words W were spoken, given that the evidence A was observed, then the recognizer should decide in favor of a word string $\widehat{\mathbf{W}}$ satisfying

$$\widehat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{A})$$

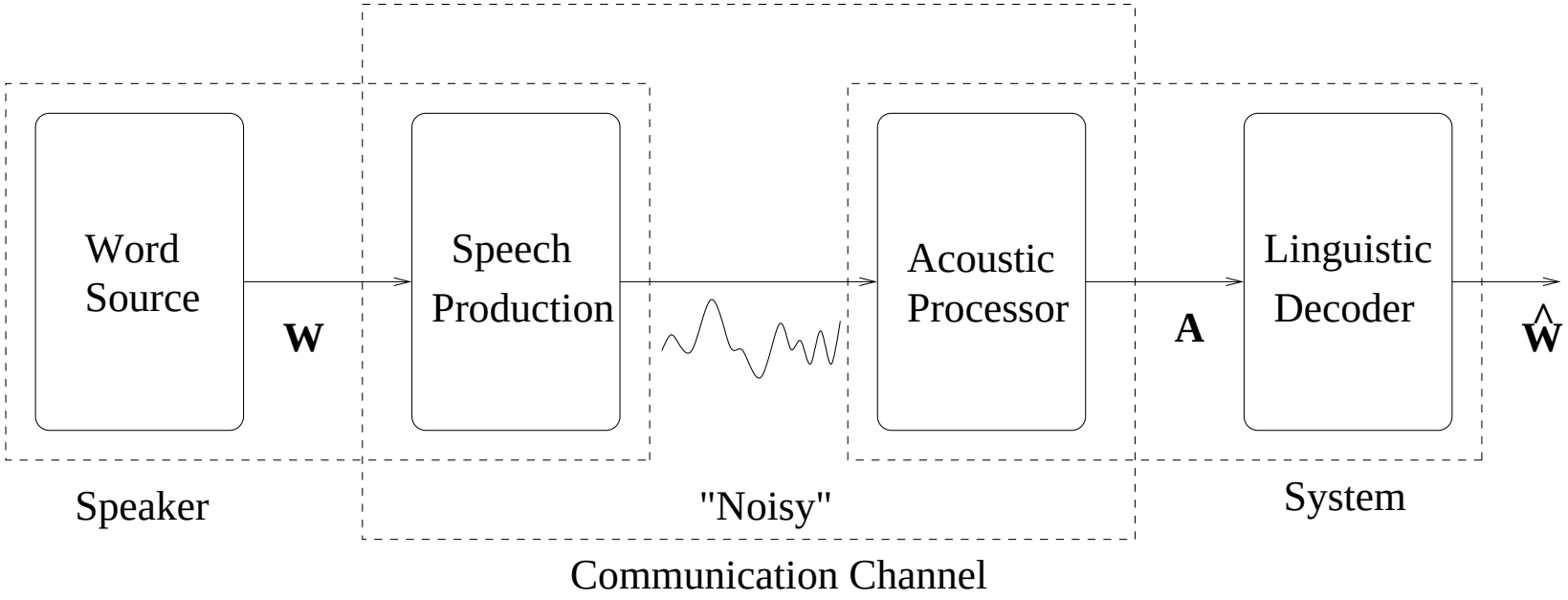
- Bayes' formula of probability theory allows us to re-write the right-hand side probability as

$$P(\mathbf{W}|\mathbf{A}) = \frac{P(\mathbf{W})P(\mathbf{A}|\mathbf{W})}{P(\mathbf{A})}$$

- Since the maximization is carried out with the variable A fixed,

$$\widehat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W})P(\mathbf{A}|\mathbf{W})$$

Communication theory diagram



Components of a speech recognizer

- Acoustic processing: What acoustic data A will be observed?
  - Decide on a "front end"
- Acoustic modeling: Determine the value $P(A|W)$.
  - To compute $P(A|W)$ on the fly, we need a statistical acoustic model
- Language modeling: Compute for every word string W the à priori probability $P(W)$ that the speaker wishes to utter W.
  - Since

$$P(\mathbf{W}) = \prod_{i=1}^{n} P(w_i|w_1, ..., w_{i-1})$$

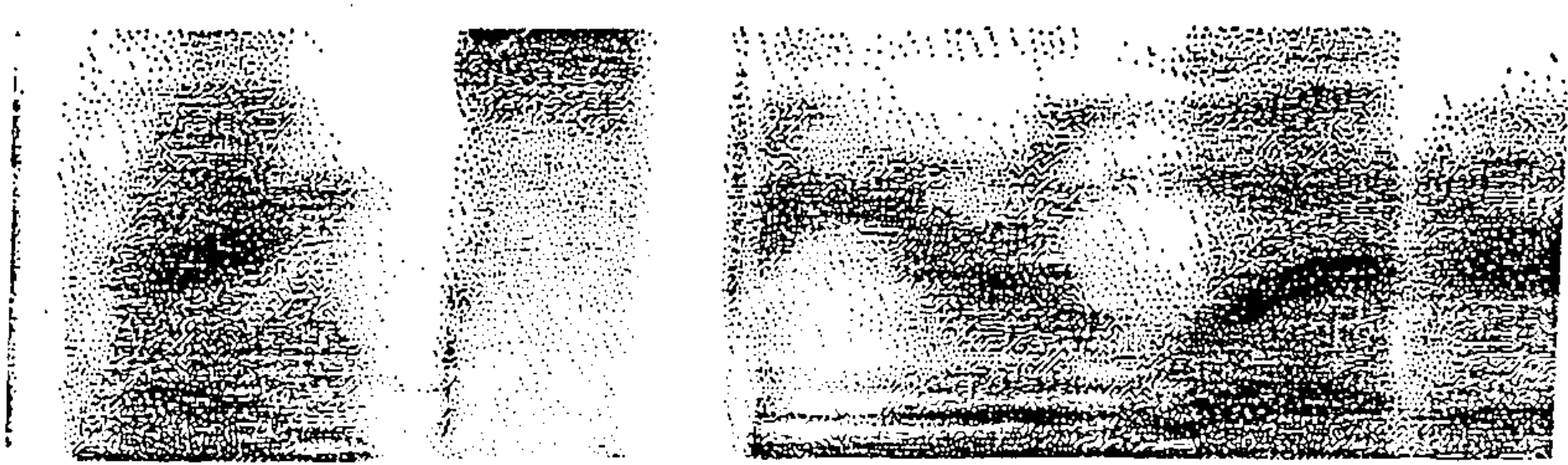
We must determine estimates of the probabilities $P(w_i|w_1, ..., w_{i-1})$.

Components of a speech recognizer (Cont.)

- Hypothesis search: We must search over all possible word strings W to find the maximizing $\hat{W}$.
  - No brute force: space of W s is astronomically large.
  - Search limited to word strings that are suggested by the acoustics A observed.

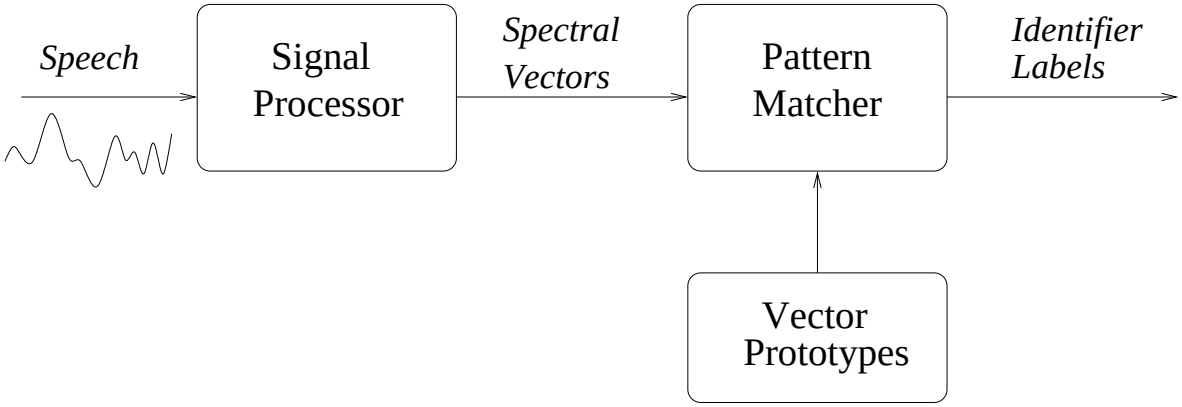
How to generate observed acoustic symbols $\mathbf{A}$

- Example of a *spectrogram*: ”visible speech”



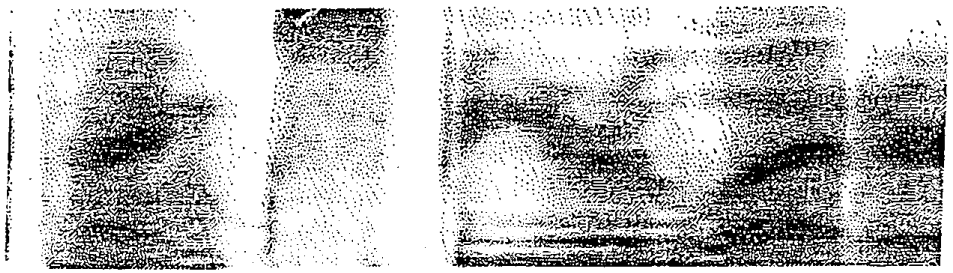
Acoustic processor

1. At regular time intervals (100 times per second) the signal processor outputs a vector of the speech energies measured in k selected frequency bands
   - By vector we mean a slice through the spectrogram!
2. The pattern matcher compares that vector with pre-stored prototypes and finds the nearest prototype.
3. Prototypes can be selected directly from speech data without any human intervention.
4. The processor output is the identifier label of the nearest prototype

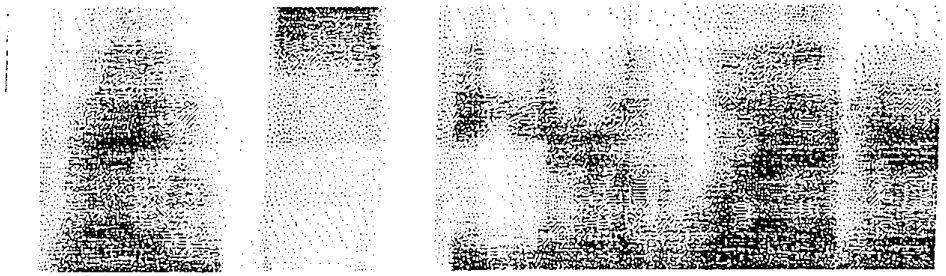


Comparison of original with the approximate spectrogram derived from Acoustic Processor output labels

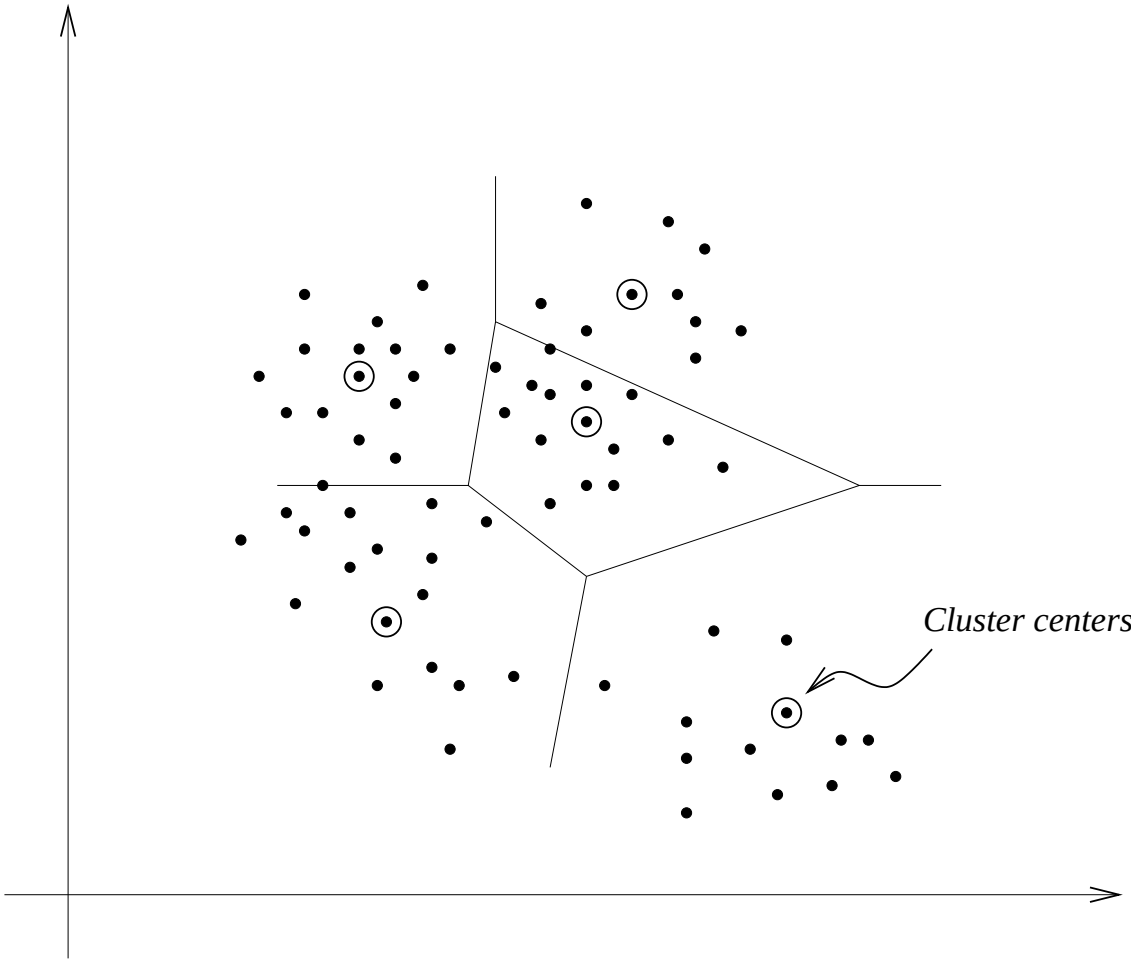
Before Vector Quantization

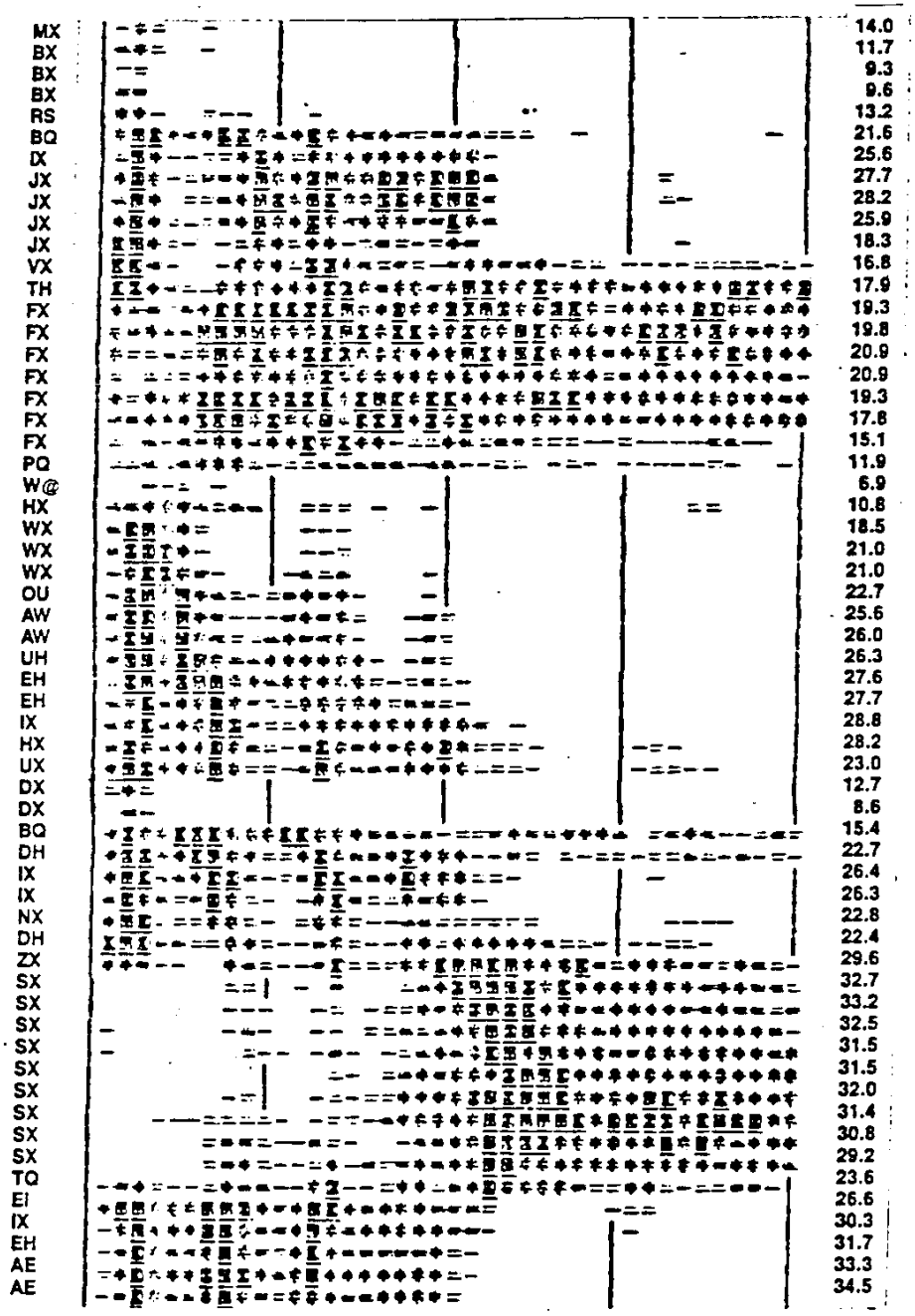


After Vector Quantization



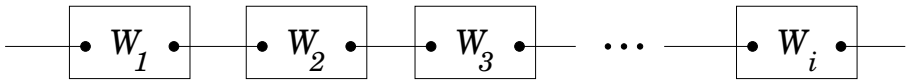
K-means clustering



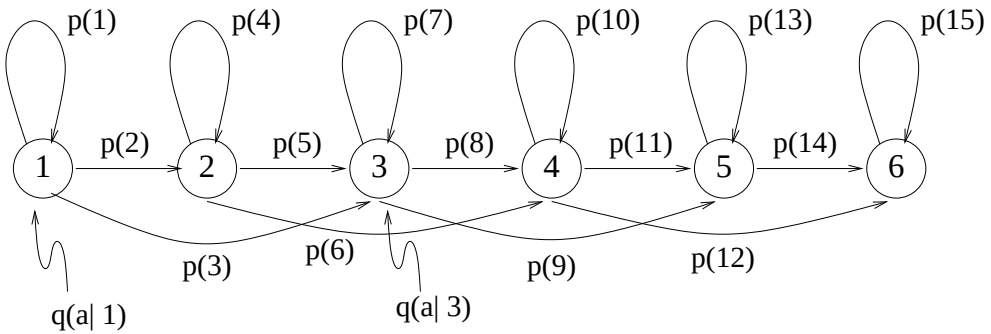


Statistical models of production of acoustic label string $\mathbf{A}$ by word string $\mathbf{W}$

- Model must be constructed from building blocks.
  - Model for $W = w_1 w_2 ... w_i ...$ is the concatenation of models for the individual words w_i.



- The Hidden Markov Model (HMM) generates the string of acoustic symbols a_1 ,a_2 ,a_3 ,... . It has states connected by transitions.
  - A starting and an ending state
  - A transition t is taken with probability $p(t)$
  - When state s is reached the model generates output $a \in A$ with probability $q(a|s)$.

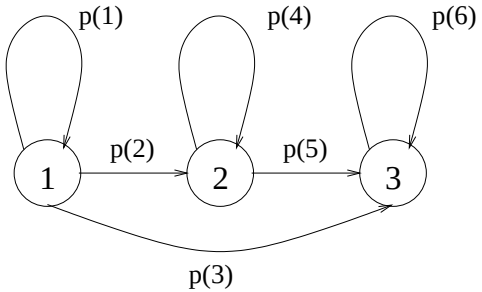


Production of acoustic label string $\mathbf{A}$ by word string $\mathbf{W}$ (Cont.)

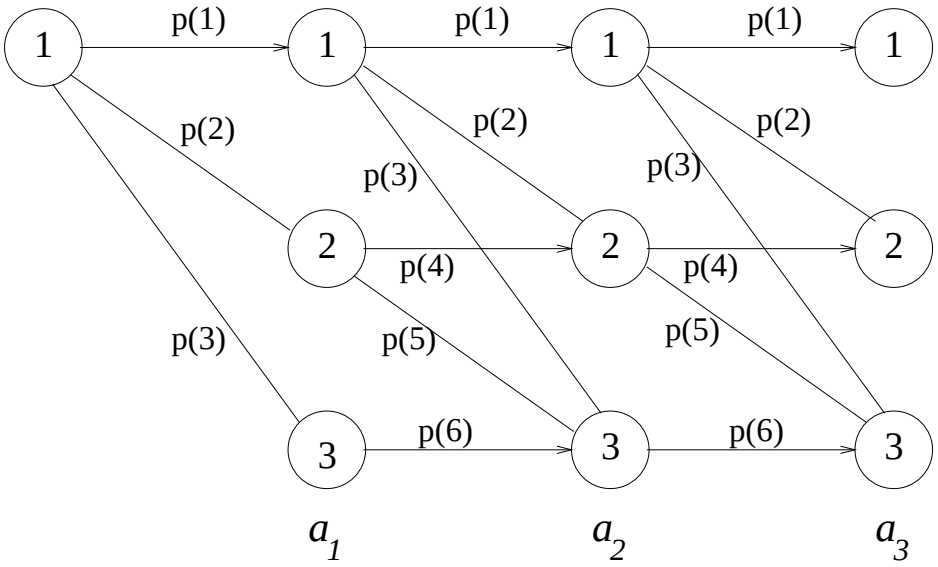
- Every word $v \in V$ will have its own HMM. The difference between models are values of model parameters:
  - the number of states M
  - the transition probabilities $p(t)$ between them
  - the acoustic label output probabilities $q(a|s)$

Visualizing the action of HMMs

- Example of a simple HMM

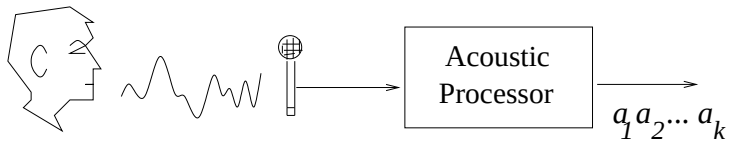


- The HMM trellis: unwinding of its generating action in time



Three questions about HMMs

- How do we compute the probability of an observed string $A = a_1 a_2 ... a_k$?
  - This string can be generated by any state sequence along any path that leads from the starting to the terminal state.
- How do we find the most probable path through the HMM?
  - This path will yield the most probable sequence of words spoken.
- How do we determine the transition probabilities $p(t)$ and output probabilites $q(a|s)$?
  - We need to get them from data!
  - User reads prepared text $W = w_1 w_2 ... w_n$ into acoustic processor
  - Acoustic processor generates corresponding output $A = a_1 a_2 ... a_k$
  - System creates composite HMM for the text W.



The answer: There are relatively simple algorithms that can accomplish all three tasks!

The basic pronunciation model

- The system contains a finite pronunciation lexicon specifying a correspondence between each word and its baseform expressed as a phonetic sequence:

$$\text{chair} \quad \leftrightarrow \quad \text{ČÉR}$$

- An utterance W is transformed into a phonetic string by replacing each of its words by its baseform followed by a delimiter

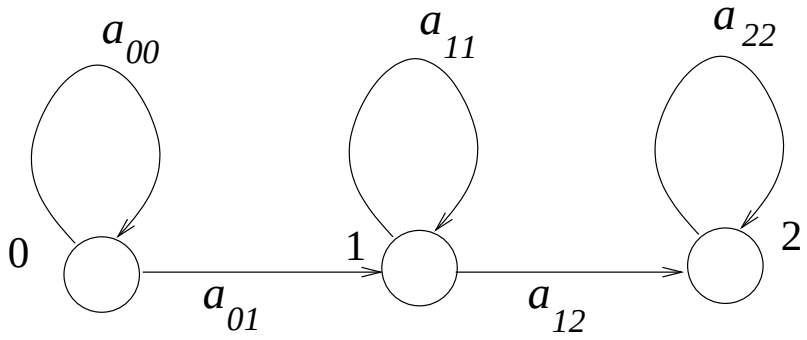
$$\text{blue chair} \quad \leftrightarrow \quad | \;\; \text{B} \;\; \text{L} \;\; \text{Ú} \;\; | \;\; \text{Č} \;\; \text{É} \;\; \text{R} \;\; |$$

Phones are pronounced according to their immediate context: the acoustic model of a phone is a tri-phone

$$| - \text{B} + \text{l}, \quad \text{b} - \text{L} + \text{ú}, \quad \text{l} - \text{Ú} + |, \quad \text{ú} - | + \text{č}, \quad | - \text{Č} + \text{é}, \quad \text{č} - \text{É} + \text{r}, \quad \text{é} - \text{R} + |$$

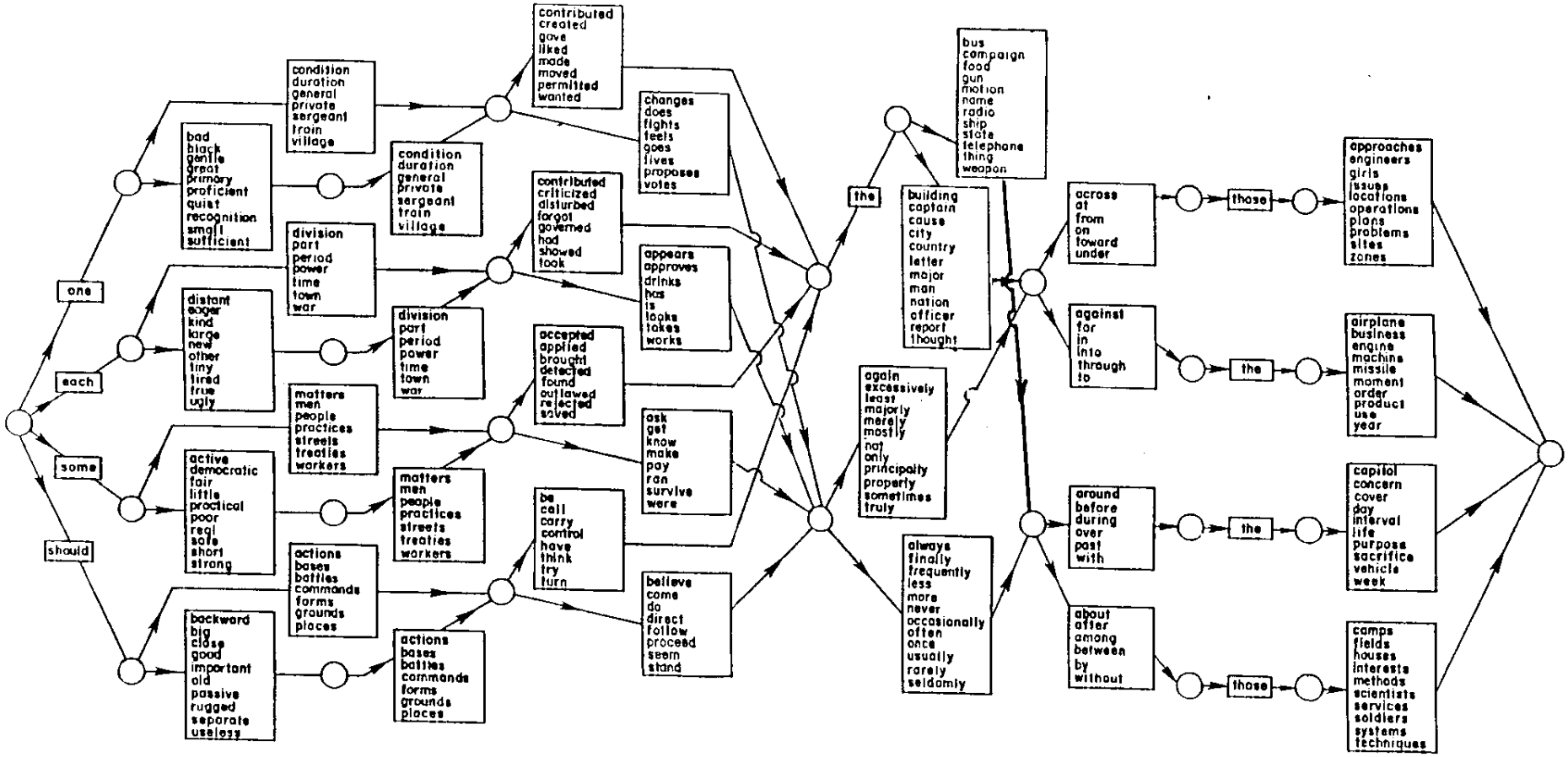
The basic acoustic model

- The microphone input is transformed by a signal processor into a sequence $a_1 a_2 ... a_k ...$ of vectors of cepstral coefficients
  - Vectors are generated 100 times a second
- Each tri-phone corresponds to a hidden Markov model (HMM) of the same structure:



- Transitions take place once every centi-second. States generate normally distributed vectors.
  - Tri-phones differ in that their statistical parameters have different values.
  - The parameter values are estimated from transcribed speech data by the EM algorithm.

New Raleigh Language revisited



The compensatory power of statistics derived from data

An experiment on New Raleigh Language task involving smaller building blocks from which word models were constructed:

- Phonetic baseforms:

$$through \Longleftrightarrow \; TH \quad R \quad \acute{U}$$

Model statistics estimated by experts:

35% correct recognition

- Phonetic baseforms:

$$through \Longleftrightarrow \; TH \quad R \quad \acute{U}$$

Model statistics estimated automatically from data:

75% correct recognition

- Orthographic baseforms:

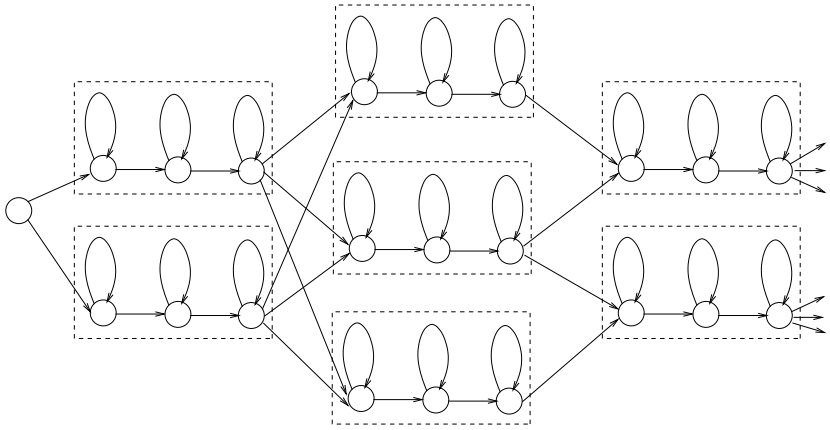
$$through \Longleftrightarrow \; T \quad H \quad R \quad O \quad U \quad G \quad H$$

Model statistics estimated automatically from data:

43% correct recognition.

Advantages of the HMM formulation

- Simple and uniform structure
- The complete model for the language (e.g., the New Raleigh Model) is one large composite HMM:
  - the transitions between words are ordinary HMM transitions between the final state of the previous word and the initial state of the next word



- The search for the best word sequence $\widehat{\mathbf{W}}$ is just a search for the best path through the composite HMM.
- It turns out that we can determine the values of the model parameters directly from speech data:
  - Aplies to all languages
  - No experts are needed!

Equivalence Classification for Language Modeling

- $P(W)$ can be formally decomposed as

$$P(\mathbf{W}) = \prod_{i=1}^{n} P(w_i | w_1, ..., w_{i-1})$$

The past $w_1, ..., w_{i-1}$ is referred to as history and is denoted by h_i.

- For a vocabulary of size $|V|$ there are $|V|^{i-1}$ different histories! For $|V| = 5000$ and $i = 3$, $|V|^i$ is equal to 125 billion!
- Let Φ be a (many to one) mapping of histories into M of equivalence classes. If $\Phi(w_1, ..., w_{i-1})$ denotes the equivalence class of the string $w_1, ..., w_{i-1}$, then

$$P(\mathbf{W}) = \prod_{i=1}^{n} P(w_i | \Phi(w_1, ..., w_{i-1}))$$

- If at time $i - 1$ class $\Phi_{i-1} \in \{1, 2, ..., M\}$ is reached,

$$P(\mathbf{W}) = \prod_{i=1}^{n} P(w_i | \Phi_{i-1})$$

Centrality of prediction (and of statistics)

- $P(w_0|\Phi(\mathbf{h}))$ where $\mathbf{h} = w_{-1}, w_{-2}, ...$
- The future is unknown. Can only be predicted via statistics.
- Equivalence classification is *central:*
  - Φ is ideally a function of *meaning* and *grammar*
  - In principle, a better Φ indicates a better theory about language
  - Entropy is an operational measure of the quality of Φ
    - measure provided by Information Theory
    - measure related to maximum likelihood

The equivalence of entropy and maximum likelihood

- We have

$$P(\mathbf{W}) = \prod_{i=1}^{n} P(w_i|\Phi_i) = \prod_{v,\Phi} P(v|\Phi)^{C(v,\Phi)}$$

where

$$f(v,\Phi) \doteq \frac{1}{n} C(v,\Phi) \doteq \frac{1}{n} \sum_{i=1}^{n} \delta(w_i, v)\, \delta(\Phi_i, \Phi)$$

and the Kronecker delta function

$$\delta(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases}$$

- Therefore,

$$\frac{1}{n} \log P(\mathbf{W}) = \sum_{v,\Phi} f(v,\Phi) \log P(v|\Phi)$$

The equivalence of entropy and maximum likelihood (Cont.)

- We would like to have estimated $P(v|\Phi)$ so as to maximize $\frac{1}{n}\log P(\mathbf{W})$. That is, we want

$$\frac{\vartheta}{\vartheta(P(v|\Phi))}\left[\sum_{v,\Phi} f(v,\Phi)\log P(v|\Phi) - \sum_{\Phi}\lambda_{\Phi}\sum_{v} P(v|\Phi)\right] = 0$$

- The solution is

$$P(v|\Phi) = \frac{f(v,\Phi)}{f(\Phi)}$$

The equivalence of entropy and maximum likelihood (Cont.)

- The above solution represents the maximum likelihood assignment of $P(v|\Phi)$.
- Therefore we want to find an equivalence classification Φ that would minimize the entropy

$$H = -\sum_{v,\Phi} f(v,\Phi) \log f(v|\Phi)$$

- **However,** we actually want to maximize $P(\mathbf{W})$ over *test* data, not training data. So *cross-entropy*

$$-\sum_{v,\Phi} f(v,\Phi) \log P(v|\Phi)$$

is a measure of that.

Estimation of probabilities

- Run the text's word sequences through equivalence classifier

$$\begin{array}{cccccccccc} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 & w_8 & w_9 & w_{10} & \ldots \\ \Phi_1 & \Phi_2 & \Phi_3 & \Phi_4 & \Phi_5 & \Phi_6 & \Phi_7 & \Phi_8 & \Phi_9 & \Phi_{10} & \ldots \end{array}$$

- Accumulate counts $C(w, \Phi)$ of the number of times the word w occurred after history was in class Φ.

$$C(\Phi) = \sum_w C(w, \Phi)$$

Relative frequency

$$f(w_i|\Phi_i = \Phi) = \frac{C(w_i, \Phi)}{C(\Phi)}$$

- First order approximation:

$$P(w_i|\Phi_i) \cong f(w_i|\Phi_i)$$

Equivalence classification requirements:

1. The classification must be sufficiently refined to provide adequate information about the history $\mathbf{h}$ so it can serve as a basis for prediction.

2. It must yield its M possible classes frequently enough so that the probabilities $P(w|\Phi)$ can be reliably estimated (not necessarily by the above crude relative frequency approach).

Linear Smoothing

- Approximation $P(w_i|\Phi_i) \cong f(w_i|\Phi_i)$ not good enough (gives 0 probability to events that were not observed but are still possible)
- Experiment:
  - Vocabulary: 1K words
  - 1.5 M words used for training
  - 300K words to test
  - Result: 23% of trigrams in test were absent from training
- Linear smoothing

$$P(w_i|\Phi_i) = \lambda\, f(w_i|\Phi_i) + (1-\lambda)\, g(w_i)$$

where $g(\bullet)$ is a suitable distribution with full support.

- To be more accurate, λ may be made to depend on the confidence we have in $f(w_i|\Phi_i)$:

$$\lambda = \lambda(C(\Phi))$$

- We choose λ to maximize our estimate of heldout data (separate from development data used to estimate $f(w_i|\Phi_i)$)

$$\widehat{\lambda} = \arg\max_{\lambda} \prod_{i=1}^{N} P_{\lambda}(w_i|\Phi_i)$$

The trigram language model

- The language model that is most frequently used is the tri-gram model

$$P(\mathbf{W}) = \prod_{i=1}^{n} \widehat{P}(w_i|w_{i-2}, w_{i-1})$$

- We have

$$\widehat{P}(w_i|w_{i-2}, w_{i-1}) = \lambda\, f(w_i|w_{i-2}, w_{i-1}) + (1-\lambda)\, \widehat{P}(w_i|w_{i-1})$$

- Where

$$\widehat{P}(w_i|w_{i-1}) = \gamma\, f(w_i|w_{i-1}) + (1-\gamma)\, f(w_i)$$

- And

$$\lambda = \lambda(C(w_{i-2}, w_{i-1}))$$

$$\gamma = \gamma(C(w_{i-1}))$$

Backing off

- The basic idea is

$$\widehat{P}(w_3|w_1,w_2) = \begin{cases} f(w_3|w_1,w_2) & \text{if } C(w_1,w_2,w_3) \geq K \\ \alpha Q_T(w_3|w_1,w_2) & \text{if } 1 \leq C(w_1,w_2,w_3) < K \\ \beta(w_1,w_2)\,\widehat{P}(w_3|w_2) & \text{otherwise} \end{cases}$$

  where α and β are appropriately chosen so that the probability $\widehat{P}(w_3|w_1,w_2)$ is properly normalized.

- Furthermore,

$$\widehat{P}(w_3|w_2) = \begin{cases} f(w_3|w_2) & \text{if } C(w_2,w_3) \geq L \\ \alpha Q_T(w_3|w_2) & \text{if } 1 \leq C(w_2,w_3) < L \\ \beta(w_2)\,f(w_3) & \text{otherwise} \end{cases}$$

- α and β are chosen so that the total probability assigned to all events that have been seen exactly once in the training data is equal to the total probability of events never seen in training data.

1	The	are	**to**	know	the	issues	necessary
2	This	will		have	this	problems	data
3	One	the		understand	these	**the**	information
4	Two	would		do	problems		above
5	A	also		get	any		other
6	Three	do		the	a		time
7	Please	**need**		use	problem		people
8	In			provide	them		operators
9	**We**			insert	**all**		tools
•				•			•
•				•			•
•				•			•
93				request			factors
94				respond			facts
95				supply			I
96				write			jobs
97				me			MVS
98				**resolve**			old
•							•
•							•
•							•
1636							mailroom
1637							marketplace
1638							provision
1639							reception
1640							shop
1641							**important**

1	role	and	**the**	**next**	be	meeting	of
2	thing	from			**two**	months	•
3	that	in				years	
4	to	to				meetings	
5	contact	are				to	
6	parts	with				week	
7	point	were				**days**	
8	for	requiring					
9	**issues**	still					
•		•					
•		•					
•		•					
61		being					
62		during					
63		I					
64		involved					
65		would					
66		**within**					

Another example of the power of trigrams

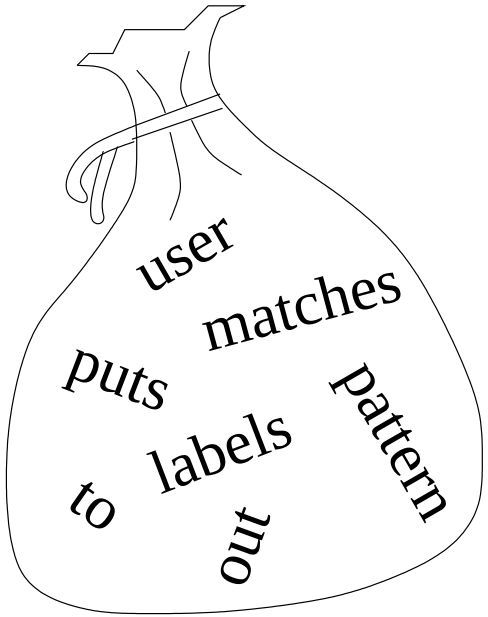
Reconstruction of a short sentence from a bag of words:

- Scramble words of a sentence
- Use trigram language model to find most probable word order, i.e.,
  - From set $\{v_1, v_2, ..., v_n\}$ find the sequence

$$w_1 = v_{i_1}, w_2 = v_{i_2}, ..., w_n = v_{i_n}$$

that will maximize the value of

$$P(w_1 w_2 ... w_n) \doteq P(w_1) P(w_2|w_1) P(w_3|w_1 w_2) ... P(w_n|w_{n-2} w_{n-1})$$



Sentence reconstruction results

- 38 randomly selected sentences of $n \leq 10$ words
- 24 sentences reconstructed exactly (63%).
- 9 more reconstructions have same meaning as originals (24%)
- Reconstruction error only 13%.

Reconstruction examples

- Meaning preserved:

would	*I*	*report*	*directly*	*to*	*you*	?
I	*would*	*report*	*directly*	*to*	*you*	?

now	*let*	*me*	*mention*	*some*	*of*
let	*me*	*mention*	*some*	*of*	*the*

the	*disadvantages*	.
disadvantages	*now*	.

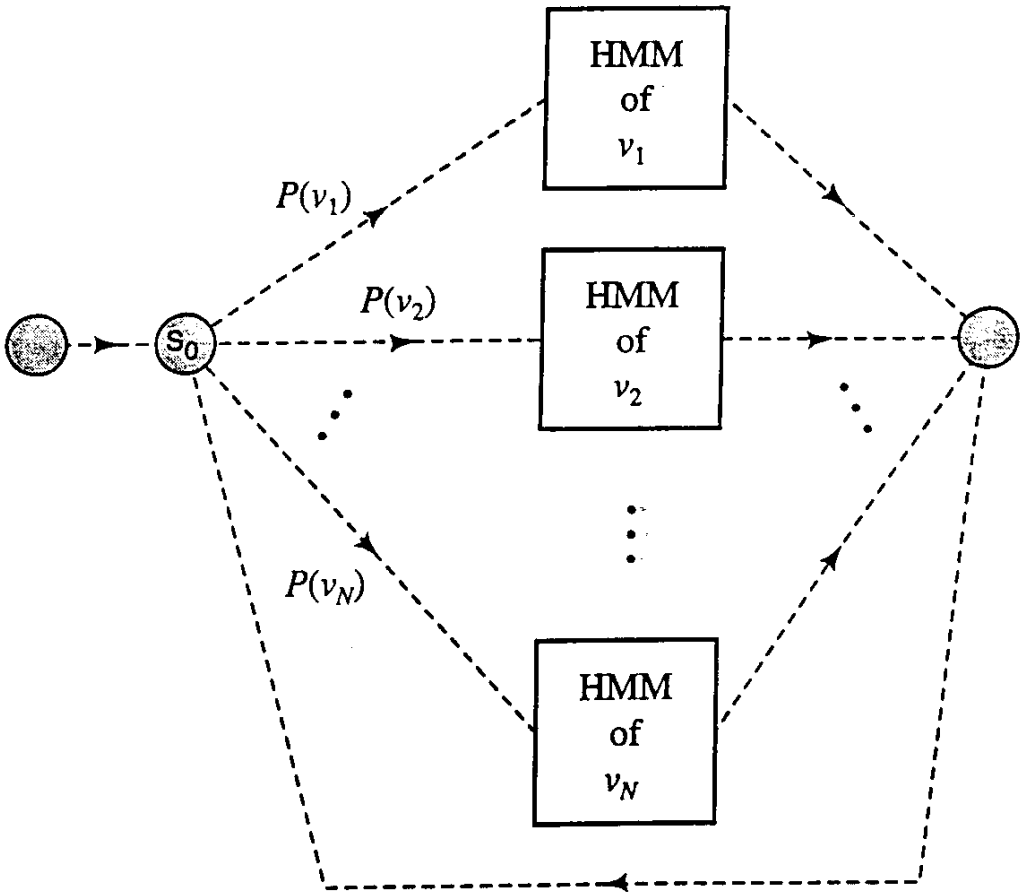
he	*did*	*this*	*several*	*hours*	*later*	.
this	*he*	*did*	*several*	*hours*	*later*	.

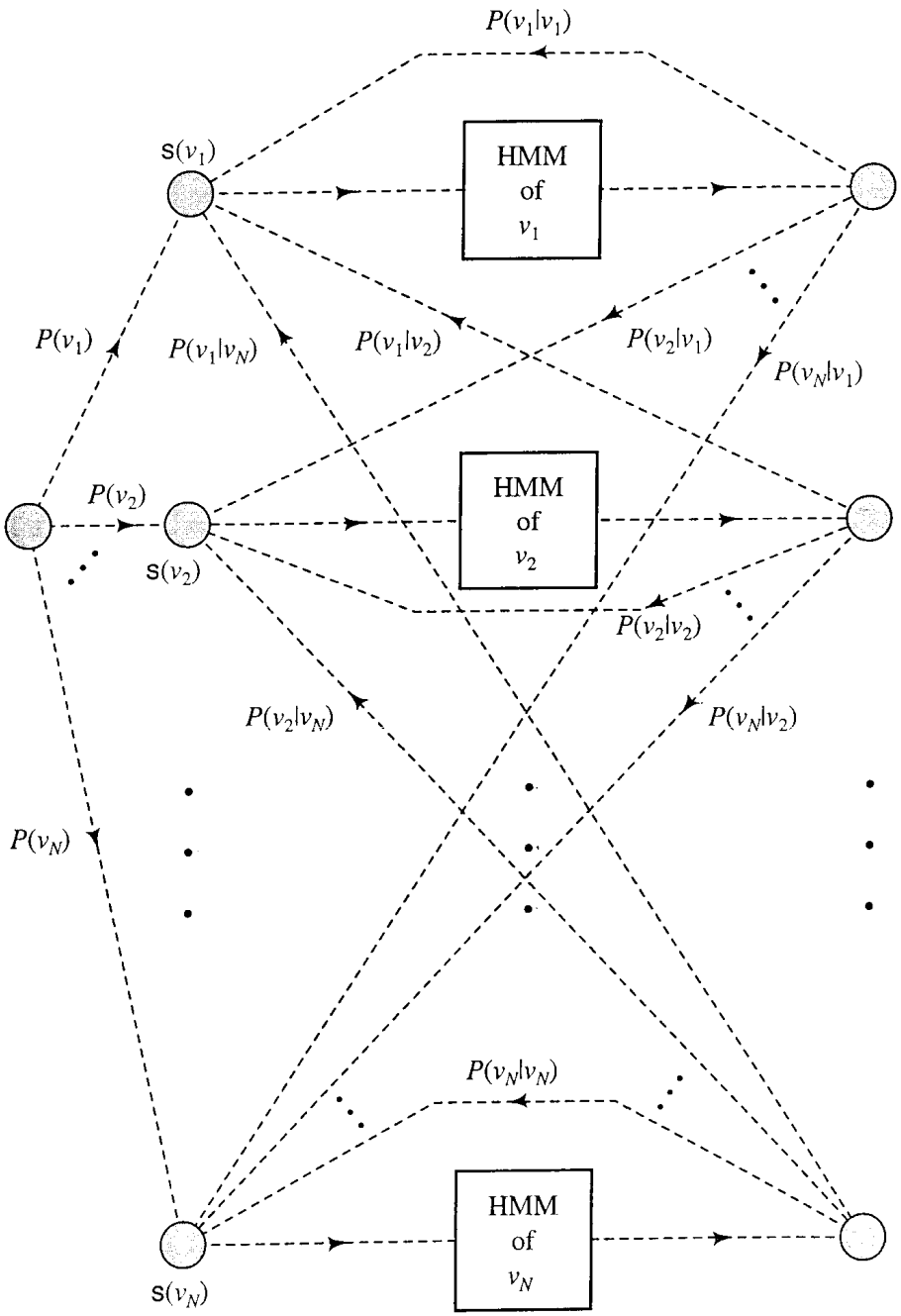
- Meaning destroyed:

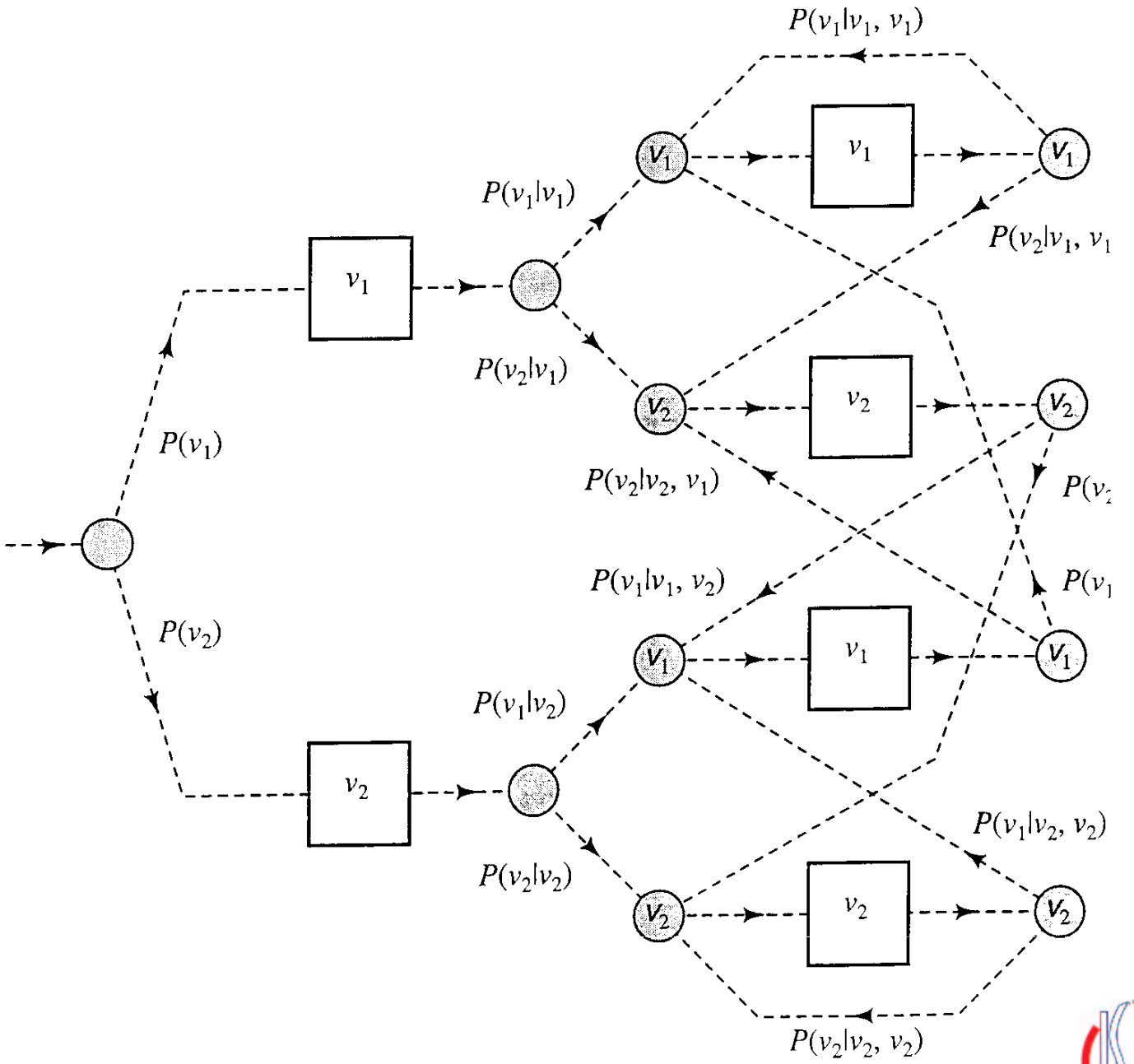
in	*our*	*organization*	*research*	*has*	*two*	*missions*	.	
in	*our*	*missions*	*research*	*organization*	*has*	*two*	.	

exactly	*how*	*this*	*might*	*be*	*done*	*is*	*not*	*clear*	.
clear	*is*	*not*	*exactly*	*how*	*this*	*might*	*be*	*done*	.

HMM graph: unigram language model




$P(v_1|v_1)$
$s(v_1)$
HMM of v_1
$P(v_1)$
$P(v_1|v_N)$
$P(v_1|v_2)$
$P(v_2|v_1)$
$P(v_N|v_1)$
$P(v_2)$
$s(v_2)$
HMM of v_2
$P(v_2|v_2)$
$P(v_2|v_N)$
$P(v_N|v_2)$
$P(v_N)$
$P(v_N|v_N)$
HMM of v_N
$s(v_N)$


$P(v_1|v_1, v_1)$
$P(v_1|v_1)$
$P(v_2|v_1, v_1)$
$P(v_2|v_1)$
$P(v_1)$
$P(v_2|v_2, v_1)$
$P(v_1|v_1, v_2)$
$P(v_2)$
$P(v_1|v_2)$
$P(v_1|v_2, v_2)$
$P(v_2|v_2)$
$P(v_2|v_2, v_2)$

Language Model Constructed via Maximum Entropy Estimation

- Must approximate language model probability $P(w|\mathbf{h}) \cong P(w|\Phi(\mathbf{h}))$ where $\Phi(\mathbf{h})$ denotes the equivalence class to which the history $\mathbf{h}$ belongs.
- Equivalence classification necessary
  1. to have fewer parameters to estimate,
  2. so that available data be sufficient for the estimation,
  3. so the probability can be constructed in a timely manner at recognition time from parameter values occupying limited storage.
- Basic idea: construct the probability $P(w, \mathbf{h})$ by insisting that
  - $P(w, \mathbf{h})$ should satisfy certain linear constraints,
  - $P(w, \mathbf{h})$ should reflect our ignorance about everything not specified by these constraints.

Example: Trigram language model constraints

- Consider the case $\mathbf{h} = wz,\ w \in V, z \in V$ and constraints

$$
\begin{array}{ll}
P(x, \mathbf{h}) = f(x, \mathbf{h}) & \text{if } C(x, \mathbf{h}) \geq K \\
P(x, w) = f(x, w) & \text{if } C(x, w) \geq M \\
P(x, z) = f(x, z) & \text{if } C(x, z) \geq M \\
P(x) = f(x) & \text{if } C(x) \geq L \\
P(\mathbf{h}) = f(\mathbf{h}) & \text{if } C(\mathbf{h}) \geq L \\
\sum_{x,\mathbf{h}} P(x, \mathbf{h}) = 1 &
\end{array}
$$

- The first set of constraints is equivalent to

$$\sum_{x',\mathbf{h}'} P(x', \mathbf{h}')\, k(x', \mathbf{h}' | x, \mathbf{h}) = f(x, \mathbf{h})$$

where

$$k(x', \mathbf{h}' | x, \mathbf{h}) = \begin{cases} 1 & \text{if } x' = x, \mathbf{h}' = \mathbf{h}, C(x, \mathbf{h}) \geq K \\ 0 & \text{otherwise} \end{cases}$$

- Remaining 5 constraint sets can be similarly expressed

The General Solution

- Let $\mathbf{x} = x_1, x_2, ..., x_n$ denote a sequence of n random variables.

  Let $k(\mathbf{x}|i)$ denote the i^{th} constraint function

- Determine $P(\mathbf{x})$ so that
  - it satisfies $\sum_{\mathbf{x}} P(\mathbf{x})k(\mathbf{x}|i) = d(i)$
    for given constraint targets $d(i),\ i = 1, 2, ..., m,$
  - the entropy $H(X)$ is maximal

- For $\sum P(\mathbf{x}) = 1,$ we must add the 0^{th} constraint function

$$k(\mathbf{x}|0) = d(0) \qquad \text{for all } \mathbf{x}$$

with the constraint target $d(0) = 1$.

The General Solution (Cont.)

- Straightforward application of calculus results in solution

$$P(\mathbf{x}) = e^{\lambda_0}\ e^{\sum_i \lambda_i k(\mathbf{x}|i)}$$

where multipliers λ_i are chosen to satisfy the constraints

$$e^{\lambda_0} \sum_{\mathbf{x}} e^{\sum_i \lambda_i k(\mathbf{x}|i)}\, k(\mathbf{x}|j) = d(j) \quad \text{for } j = 0, 1, ..., m$$

- Observe that the derived probability $P(x_1, ..., x_n)$ is equal to a product of factors (e^{λ_i}), one for each constraint in which the particular argument $x_1, ..., x_n$ participates (i such that $k(x_1, ..., x_n|i) = 1$).

The practical problem

- Two basic questions:
  1. how to choose the constraints,
  2. how to solve for the parameters λ_i .

- One method for finding λ_i is called iterative projection:
  1. Guess at the values of $\lambda_i, i = 1, 2, ..., m$.
  2. For $j = 0$ to m , do:

     keeping λ_i , $i \neq j$ fixed, find λ_j^* so as to satisfy the j^{th} constraint;

     set $\lambda_j = \lambda_j^*$;

     end;
  3. If all the constraints are sufficiently satisfied, then stop. Else go to 2.

- The convergence of iterative projection may be slow, particularly if the number of constraints m is large.

Some unsolved problems

- How to find appropriate constraints
- How to find target values $d(i)$
  - If these targets are marginal probabilities, better estimates than relative frequencies exist
  - How do we use better estimates and preserve the mutual consistency of all the constraints?
- There will always be uncertainty about the value of the targets $d(i)$. How are we to incorporate it into the formulation?

Essentials of the statistical approach

- A clear statement of the problem and of the goal
  - Communication theory formulation of the recognition process
  - Search for $\widehat{\mathbf{W}}$ maximizing $P(A|W)P(W)$
- Based on data related to the process:
  - Speech transcribed and raw (for A and $P(A|W)$)
  - Training text (for $P(W)$)
- Choice of parametric models
  - HMMs for $P(A|W)$
  - Trigrams for $P(W)$
- Equivalence classification of data states
  - Speech vector prototypes
  - HMM building blocks
  - Equivalence sets for language model
- Estimation of model parameters based on a clearly defined criterion
  - Find $\widehat{\theta}$ maximizing $P_\theta(A|W)$
  - Find $\widehat{\phi}$ maximizing $P_\phi(W)$
- A unified point of view

In Conclusion

- The statistical approach provides us with a unified point of view applicable to all languages and requiring a minimum of expert preparation
- A clear statement of the problem and of the goal
  - Search for $\widehat{\mathbf{W}}$ maximizing $P(A|W) \times P(W)$
- The entire design of the recognizer is based on actual data related to the process:
  - Raw and transcribed speech (for A and $P(A|W)$)
  - Training text (for $P(W)$)
- Modern speech recognizers are capable of transcribing natural dictated speech using tens of thousands of words as the vocabulary with less than a 10% error rate.
- The next challenges:
  - Transcription of telephone conversations
  - Real speech understanding
  - Language translation