Automatic Speech Recognition - Part I

Dr. Mathew Magimai Doss

September 21, 2022

Outline

Outline

String Matching

Matching two sequences of symbols

	D	3	2	2	1	2
	C	2	1	1	2	3
n	B	1	1	2	3	4
	A	0	1	2	3	4
		A	C	C	D	E
				m		

symbols can be alphabets of a language or words in a language

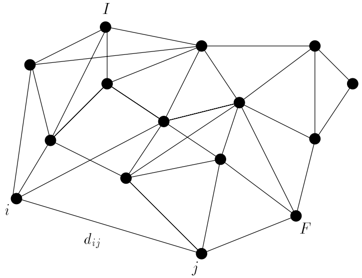
Dynamic Programming (DP)

Bellman, 1960

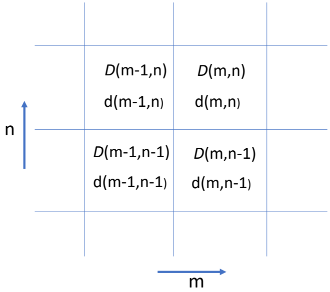
"*Optimal policy is composed of optimal sub-policies*".

Other Applications:

- Cargo loading problem, VLSI design, etc...
- Finding the shortest path between two points in a graph



String matching using DP



local score $d(m,n)$:
if $\mathrm{str}(m) = \mathrm{str}(n)$
$d(m,n) = 0$
else
$d(m,n) = 1$

1. Initial condition: path starts at $(1,1)$
2. Recursion:

$$D(m,n) = d(m,n) + min[D(m-1,n), D(m-1,n-1), D(m,n-1)]$$

$$\mathrm{Path}(m,n) = \arg\min[D(m-1,n), D(m-1,n-1), D(m,n-1)]$$

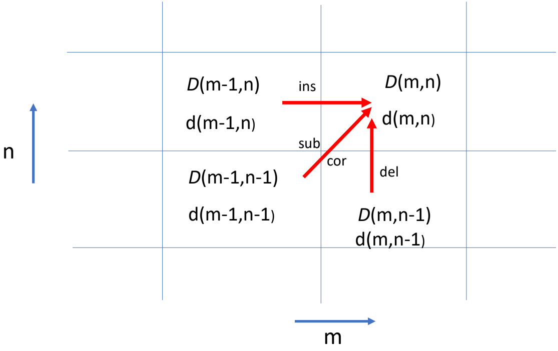
$$\forall m \in \{1 \dots M\} \text{ and } n \in \{1,\dots N\}$$

3. Final condition: path ends at (M,N) and $D(M,N)$ is the global score

$\mathrm{Path}(m,n)$ denotes the path index. Path can be traced back from $\mathrm{Path}(M,N)$

Local constraints

$$D(m,n) = d(m,n) + min[D(m-1,n), D(m-1,n-1), D(m,n-1)]$$



ins: insertion, cor: correct, sub: substitution, del: deletion

String Matching

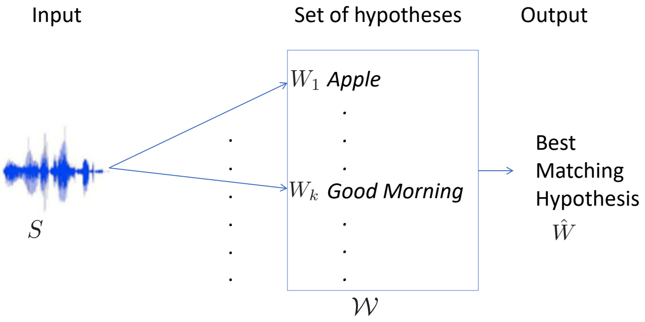
D	3	2	2	1	2
C	2	1	1	2	3
B	1	1	2	3	4
A	0	1	2	3	4
	A	C	C	D	E

n

m

Outline

Automatic speech recognition (ASR)



$$\hat{W} = \underset{W_k \in \mathcal{W}}{\arg\max}\, \mathrm{Match}(W_k, S)$$

How to match an observed speech signal S with a word hypothesis W_k?

Abstract formulation for matching S and W_k

W_k

$A = y_1 \ldots\ldots\ldots\ y_n \ldots\ldots\ldots\ y_N$

$B = z_1 \ldots\ldots\ldots\ z_m \ldots\ldots\ldots\ z_M$

S

Core Idea

1. Map S and W_k to a shared latent symbol space
2. Match the resulting two latent symbol sequences A and B

Four sub questions

Q1: What is the shared latent symbol set?

Q2: How to map S to a latent symbol sequence B?

Q3: How to map W_k to a latent symbol sequence A?

Q4: How to match the two latent symbol sequences A and B?

Different ASR methods mainly differ on how these four sub questions are addressed.

ASR methods

1. Knowledge-based approach
2. Instance-based approach
3. Model-based approach

Knowledge-based ASR approach

Q1: Phones (linguistic knowledge-based)

W_k

Q3. Apply linguistic knowledge

y_1=/b/ y_2=/ae/ .... y_n=/k/ .... y_N=/t/

z_1=/p/ z_2=/ae/......... z_m=/g/ .... z_M=/t/

Q4: Match two phone sequences (string matching)

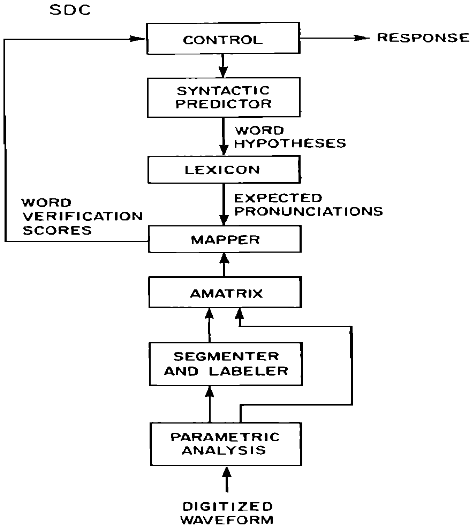
Q2. Segment and label based on acoustic-phonetic knowledge

S

Limitations:

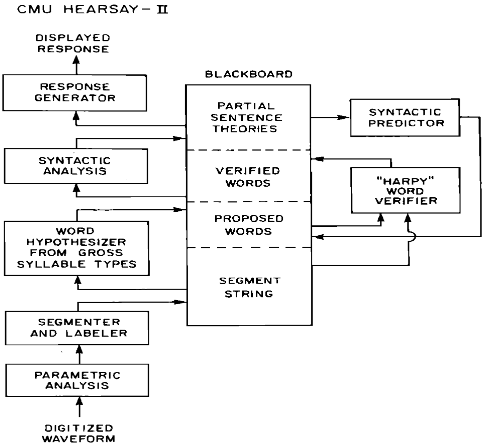
- Overly relies on knowledge
- Makes early decision so difficult to recover from errors such as, segmentation and labeling errors

Knowledge-based ASR system (1)



Source: D. H. Klatt. Review of the ARPA speech understanding project. J. Acoust. Soc. Amer., 62(6):1345-1366, December 1977.

Knowledge-based ASR system (2)



Source: D. H. Klatt. Review of the ARPA speech understanding project. J. Acoust. Soc. Amer., 62(6):1345-1366, December 1977.

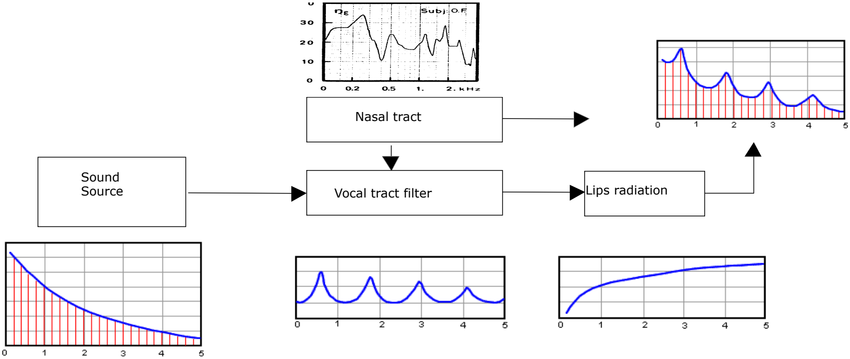
Outline

Instance-based ASR approach

In instance-based (also called template-based) approach W_k is represented by a speech signal

For example, record each word

Motivation (1)

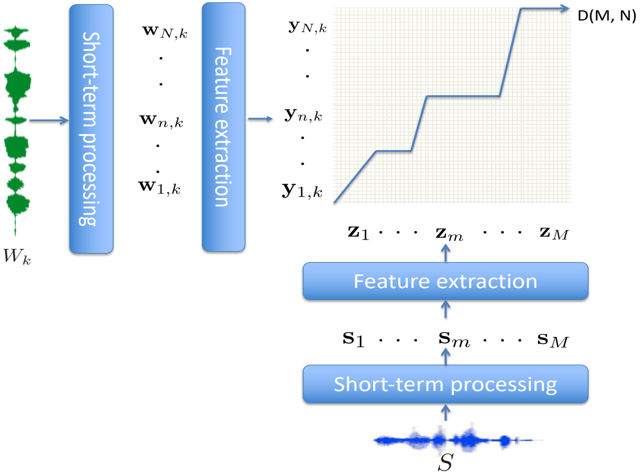


Credits: Lindqvist-Gauffin, Sundberg, Stevens, Mannel

Motivation (2)

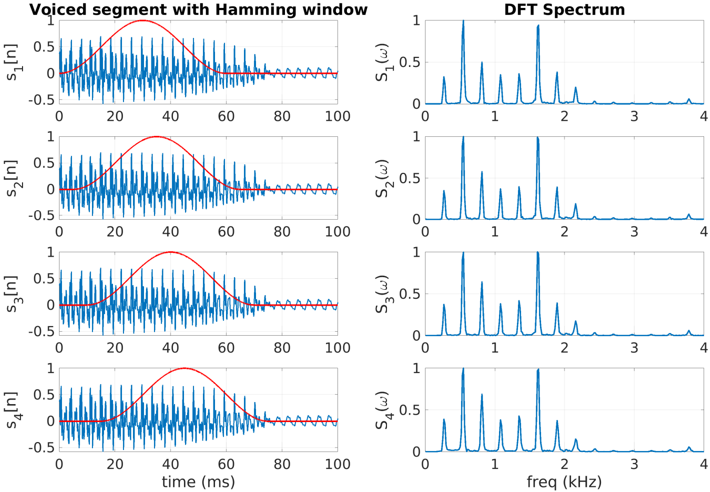
- Speech signal can be deconvolved into source and system components and synthesized back by putting these components. (e.g., linear prediction, cepstral analysis)
- Vocal tract shape is different for different sounds (caution: there are pair of sounds that differ mainly in terms of voicing, e.g., /p/ and /b/)
- Parametrize the vocal tract system information integrating speech perception knowledge (e.g. MFCCs, PLP cepstral coefficients) and compare S and W_k .

Matching S and W_k

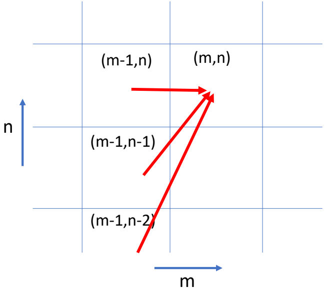


- s_m and $w_{n,k}$ denote of frame of speech signal
- z_m and $y_{n,k}$ denote the corresponding feature vectors

Short-term spectral processing



Dynamic Time Warping (DTW)



local score $d(m, n)$:

- Cepstral features: Euclidean distance between z_m and $y_{n,k}$
- Linear prediction coefficients: Itakura distance between z_m and $y_{n,k}$
- Spectral information: Itakura-Saito distance between z_m and $y_{n,k}$

1. Initial condition: path starts at $(1, 1)$
2. Recursion:

$$\begin{aligned} D(m,n) =& d(m,n) + min[D(m-1,n), \\ & D(m-1,n-1), \\ & D(m-1,n-2)] \\ \mathrm{Path}(m,n) =& \arg\min[D(m-1,n), \\ & D(m-1,n-1), \\ & D(m,n-2)] \end{aligned}$$

$\forall m \in \{1 \ldots M\}$ and $n \in \{1, \ldots N\}$

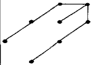
3. Final condition: path ends at (M, N) and $D(M, N)$ is the global score

$\mathrm{Path}(m, n)$ denotes the path index. Path can be traced back from $\mathrm{Path}(M, N)$

DTW local constraints

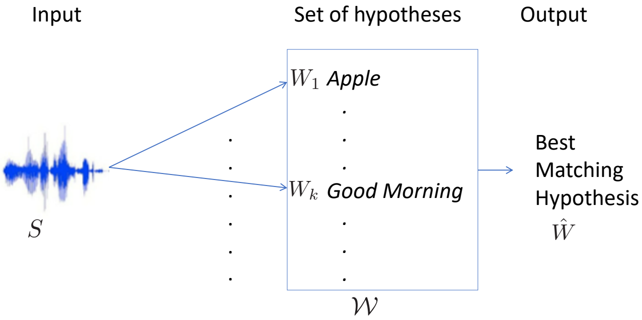
TABLE I

SYMMETRIC AND ASYMMETRIC DP-ALGORITHMS WITH SLOPE CONSTRAINT CONDITION $P = 0, \frac{1}{2}, 1$, AND 2

P	Schematic explanation	Symmetric / Asymmetric	DP-equation $g(i, j) =$
0		Symmetric	$\min\begin{bmatrix} g(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-1,j)+d(i,j) \end{bmatrix}$
		Asymmetric	$\min\begin{bmatrix} g(i,j-1) \\ g(i-1,j-1)+d(i,j) \\ g(i-1,j)+d(i,j) \end{bmatrix}$
1/2		Symmetric	$\min\begin{bmatrix} g(i-1,j-3)+2d(i,j-2)+d(i,j-1)+d(i,j) \\ g(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-2,j-1)+2d(i-1,j)+d(i,j) \\ g(i-3,j-1)+2d(i-2,j)+d(i-1,j)+d(i,j) \end{bmatrix}$
		Asymmetric	$\min\begin{bmatrix} g(i-1,j-3)+(d(i,j-2)+d(i,j-1)+d(i,j))/3 \\ g(i-1,j-2)+(d(i,j-1)+d(i,j))/2 \\ g(i-1,j-1)+d(i,j) \\ g(i-2,j-1)+d(i-1,j)+d(i,j) \\ g(i-3,j-1)+d(i-2,j)+d(i-1,j)+d(i,j) \end{bmatrix}$
1		Symmetric	$\min\begin{bmatrix} g(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-2,j-1)+2d(i-1,j)+d(i,j) \end{bmatrix}$
		Asymmetric	$\min\begin{bmatrix} g(i-1,j-2)+(d(i,j-1)+d(i,j))/2 \\ g(i-1,j-1)+d(i,j) \\ g(i-2,j-1)+d(i-1,j)+d(i,j) \end{bmatrix}$
2		Symmetric	$\min\begin{bmatrix} g(i-2,j-3)+2d(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-3,j-2)+2d(i-2,j-1)+2d(i-1,j)+d(i,j) \end{bmatrix}$
		Asymmetric	$\min\begin{bmatrix} g(i-2,j-3)+2(d(i-1,j-2)+d(i,j-1)+d(i,j))/3 \\ g(i-1,j-1)+d(i,j) \\ g(i-3,j-2)+d(i-2,j-1)+d(i-1,j)+d(i,j) \end{bmatrix}$

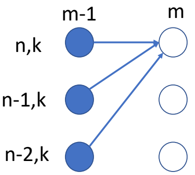
Source: H. Sakoe and S. Chiba,"Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Trans. on Acoustics, Speech, and Signal Processing, 26(1), 1978.

Instance-based ASR

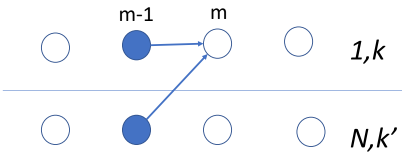


$$\hat{W} = \arg\min_{W_k \in \mathcal{W}} \mathrm{DTW}(W_k, S)$$

Across word local constraint



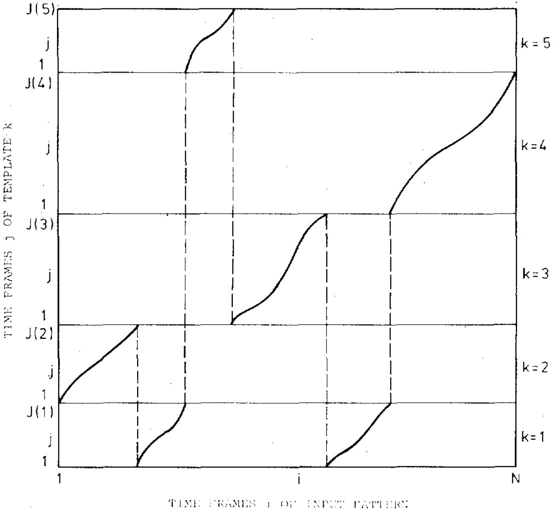
Within word constraint



Across word constraint

$$\forall k^{'} \in \{1, \cdots K\}$$

Continuous speech recognition



Source: H. Ney,"The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition ", IEEE Trans. on Acoustics, Speech, and Signal Processing, 32(2), 1984.

Four sub questions for instance-based approach

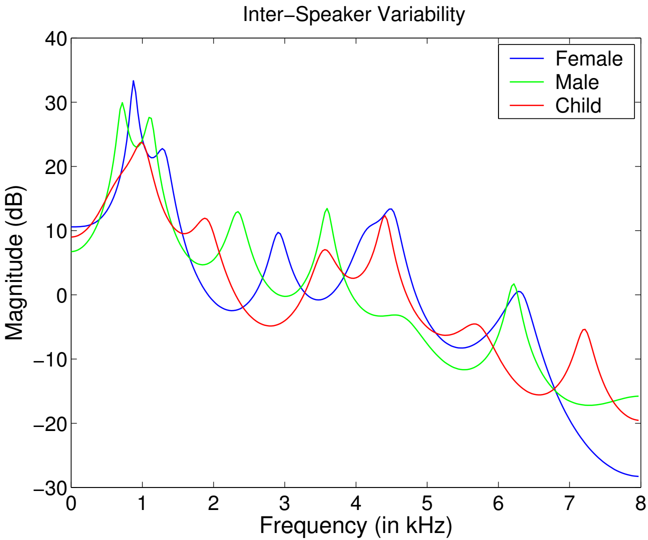
Q1 : Short-term spectral feature vectors are the latent symbols. The set of symbols is undefined, as there is no unique feature vector representation for speech sounds due to variabilities.

Q2 : Short-term speech processing-based feature extraction

Q3 : Short-term speech processing-based feature extraction

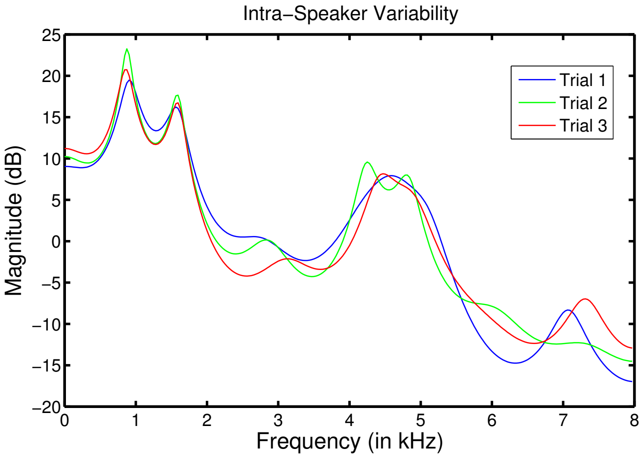
Q4 : Dynamic programming, i.e. DTW, with appropriate local score and local constraints

Inter-speaker variability



Linear prediction spectrum of a frame of sustained vowel /aa/ from different speakers

Intra-speaker variability



Linear prediction spectrum of a frame of sustained vowel /aa/ from the same speaker

Summary

Limitations

- Works well for speaker-dependent, clean and controlled conditions
  - Late 1990s name dialing on mobile phones
- Generalization across speakers and conditions is a highly challenging problem
- Reference templates typically represent word units. Every new word needs a new reference template.
- Getting phone-based reference templates is a non-trivial task
- Large amount of CPU and memory requirements

Pro: No training needed

Holy grail: Find the short-term speech processing based feature representation that carries linguistic unit (phone/syllable) related information and is robust to undesirable variabilities.

Outline

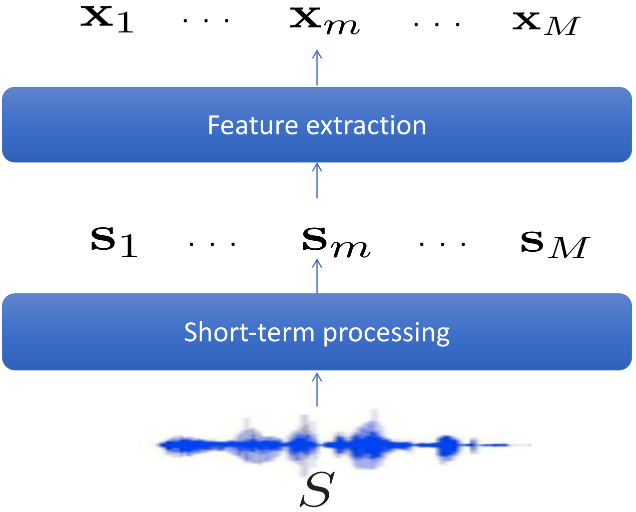
Statistical formulation for matching S and W_k

$$\hat{W} = \arg\max_{W_k \in \mathcal{W}} P(W_k|S) = \arg\max_{W_k \in \mathcal{W}} \frac{p(W_k, S)}{p(S)}$$

- Likelihood-based approach
  $p(W_k, S)$
- Posterior-based approach
  $P(W_k|S)$

<u>Model-based approach</u>

Speech signal S representation



Word hypothesis W_k representation

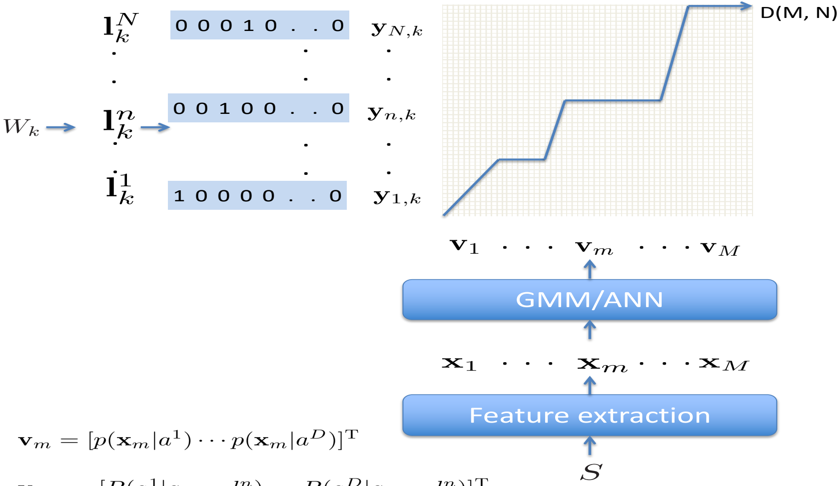
$\mathbf{l}_k^1 \quad \dots \quad \mathbf{l}_k^n \quad \dots \quad \mathbf{l}_k^N$ /dh/ /iy/ sp /k/ /ae/ /t/

Lexicon

$\mathbf{w}_k^1 \quad \dots \quad \mathbf{w}_k^j \quad \dots \quad \mathbf{w}_k^J$ The Cat

W_k

Matching S and W_k: $P(W_k, S)$



$$\mathbf{v}_m = [p(\mathbf{x}_m|a^1) \cdots p(\mathbf{x}_m|a^D)]^{\mathrm{T}}$$

$$\mathbf{y}_{n,k} = [P(a^1|q_m = l_k^n) \cdots P(a^D|q_m = l_k^n)]^{\mathrm{T}}$$

Thank you for your attention!

Dr. Mathew Magimai Doss

Idiap Research Institute, Martigny, Switzerland