Tampere University

Automatic Speech Recognition

SGN 14007

Guest Lecture, 26 November 2019

GP Huang

Postdoc Research Fellow,

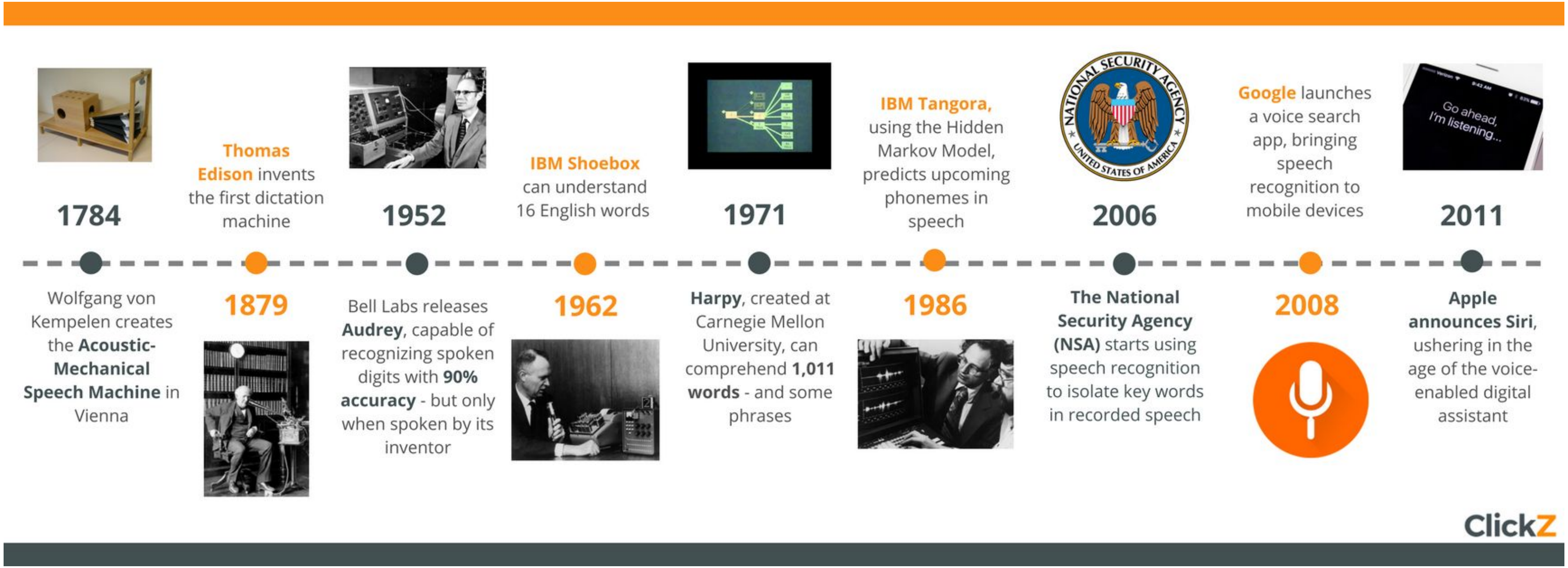
Audio Research Group, Tampere University

ASR Today

Conversational Agents:

- ➢ Apple Siri
- ➢ Amazon Echo & Alexa
- ➢ Google Home
- ➢ Microsoft Cortana … and many more

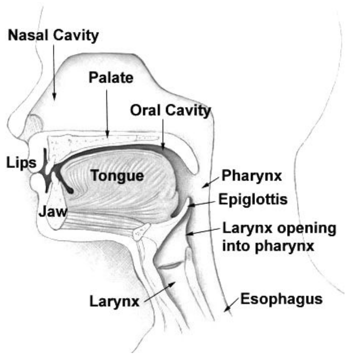
ASR History

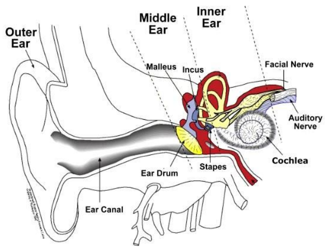

1784
Wolfgang von Kempelen creates the **Acoustic-Mechanical Speech Machine** in Vienna
1879
Thomas Edison invents the first dictation machine
1952
Bell Labs releases **Audrey**, capable of recognizing spoken digits with **90% accuracy** - but only when spoken by its inventor
1962
IBM Shoebox can understand 16 English words
1971
Harpy, created at Carnegie Mellon University, can comprehend **1,011 words** - and some phrases
1986
IBM Tangora, using the Hidden Markov Model, predicts upcoming phonemes in speech
2006
The National Security Agency (NSA) starts using speech recognition to isolate key words in recorded speech
NATIONAL SECURITY AGENCY
UNITED STATES OF AMERICA
2008
Google launches a voice search app, bringing speech recognition to mobile devices
2011
Apple announces Siri, ushering in the age of the voice-enabled digital assistant
Go ahead, I'm listening...
ClickZ

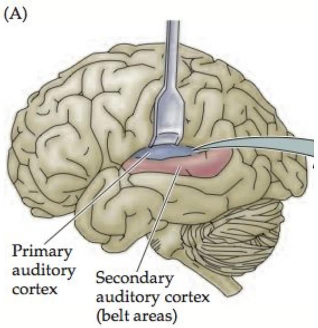
HSR

Three parts:

- Production: articulatory system ~ the mouth
- Perception: peripheral auditory system ~ the ear(s)
- Auditory nervous system - the brain







Tampere University

Thoughts…

What immediately came into your mind when your hear “speech recognition”?

- Human speech recognition (HSR), how?
- Automatic speech recognition (ASR), how, why, what?

Can you think of examples where HSR inspired ASR?

- Shall we **‘replicate’** the HSR parts?
- Shall we **‘mimic’** the HSR parts?

ASR: machine vs. human

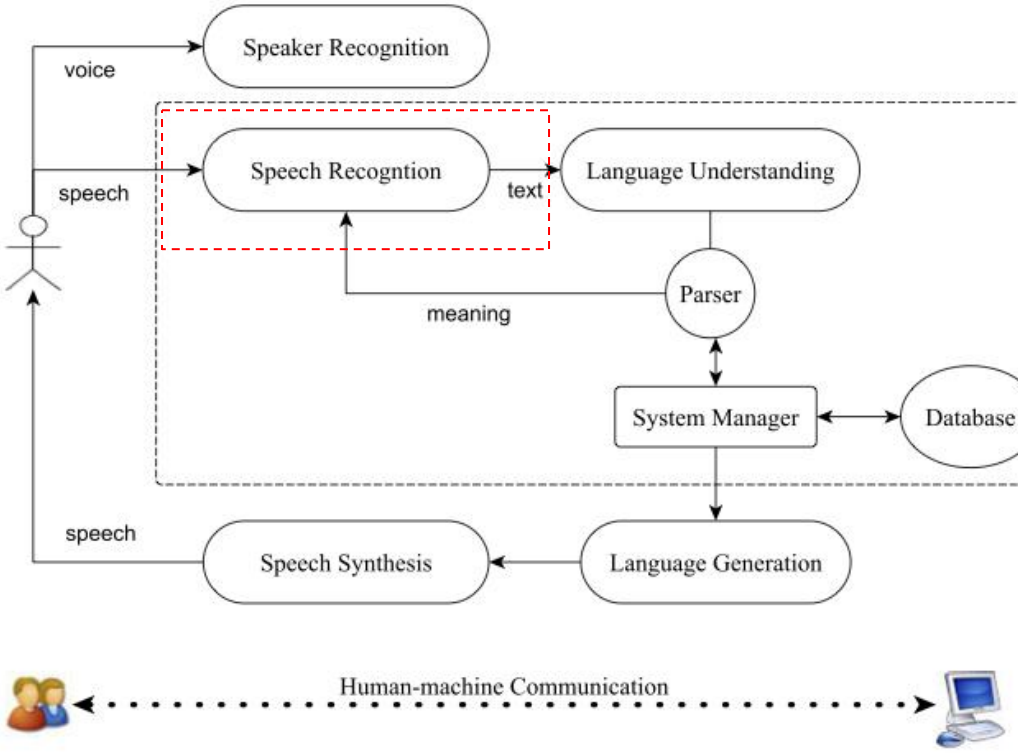
- Large Vocabulary Conversational Speech
  - Average human adult ~ 20K words
  - Conversational vs. read speech
  - Clean vs. noisy speech
  - Speaker-variations: dialects, gender, age, health ...

Task	Vocabulary	Word Error Rate (%)	
		Machine	Human
Continuous Digits	11	0.5	0.009
WSJ 1995 clean speech	5K	3	0.9
WSJ 1995 noisy speech	5K	9	1.1
Conversational Telephone	65K	6	3~4

What to expect from today:

1. Approach ASR as a research problem
2. Understand the challenges, status and potentials of the technology
3. Some of the state-of-the-art machine learning methods in ASR
4. Cats and dogs

Part of Speech & Language Processing:



Define the problem

- ASR: speech to text, audio to transcription

The **good:** speaking is **2 ~ 5** times faster than typing

The **bad**: “Recognizer speech” **vs.** “Wreck a nice beach”

The **ugly**:__w___rrr_eee_k___a_r__a_n__i_i_s_c_c_s___ssppp_e_____ eeeee_______eee____ch___hh____

Human Sounds

Speech

Singing

Humming

Hiccups

Sneezes

Whistle…

Human Language & Speech - “layers”

Language

⇩

Sentence (Prosody)

⇩

Phrase

⇩

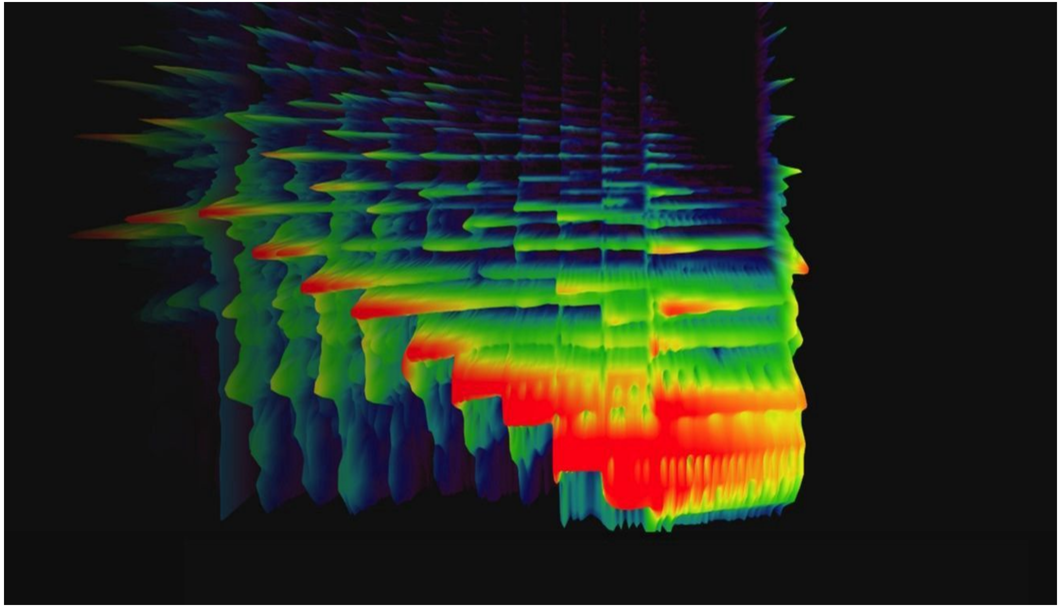
Word

⇩

Phoneme

⇩

Pronunciation (phone)



ASR: machine vs. human

- Large Vocabulary Conversational Speech
  - Average human adult ~ 20K words
  - Conversational vs. broadcast/read speech
  - Clean vs. noisy environment
  - Speaker-dependant vs. speaker-independent
  - Online vs. offline

Task	Vocabulary	Word Error Rate (%)	
		Machine	Human
Continuous Digits	11	0.5	0.009
WSJ 1995 clean speech	5K	3	0.9
WSJ 1995 noisy speech	5K	9	1.1
Conversational Telephone	65K	6	3~4

ASR Word Error Rate

- How to evaluate the word string output by a speech recognizer?
- Word Error Rate (WER):

$$\frac{100\ (\text{Insertions}+\text{Substitutions} + \text{Deletions})}{\text{Total Word in Reference/Correct Transcript}}$$

- Example (NIST sctk scoring with sclite):

REF: if **** MUSIC be THE food of love

HYP: if FOR MUSE be THEIR **** of love

Eval I S S D

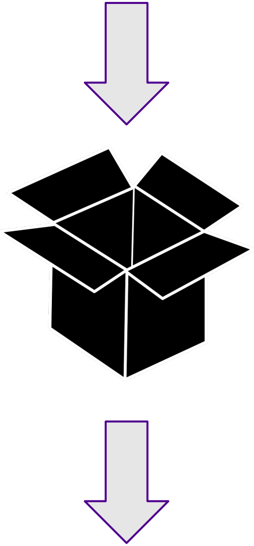
WER = 100 (1+2+1)/6 = 66.7%

Source: http://www.nist.gov/speech/tools/

Formulate ASR

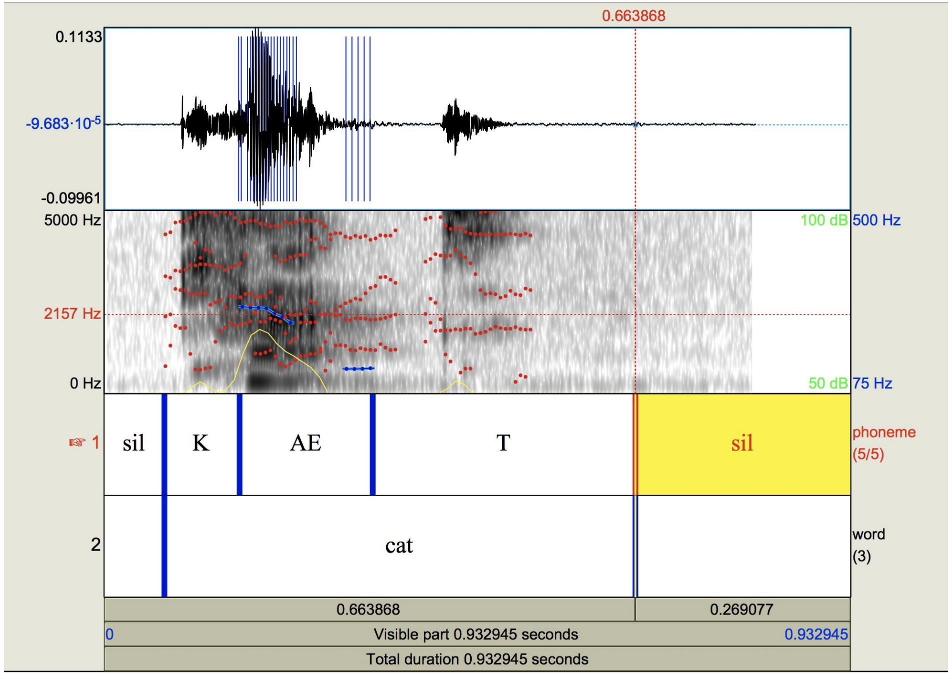


- ❖ Speech: time, dynamic signal, left-to-right, … : **beads on a string**
- ❖ What we want? - transcription
- ❖ What we have?
  - ➢ Math: probability, statistics …, (don’t panic)
  - ➢ Devices:
    - ■ Microphones
    - ■ Laryngograph
    - ■ Electromagnetic Articulograph
    - ■ ...
  - ➢ Computaters?
  - ➢ Knowledge:
    - ■ Linguists
    - ■ Phoneticians
    - ■ Mathematicians, ...

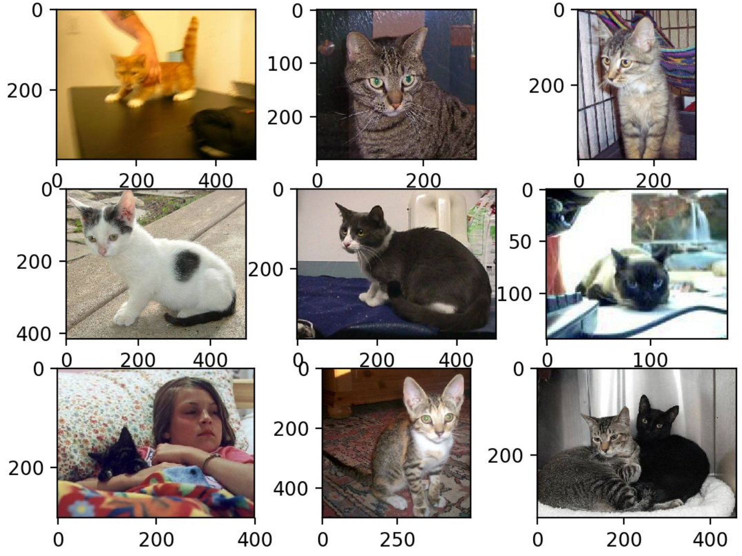


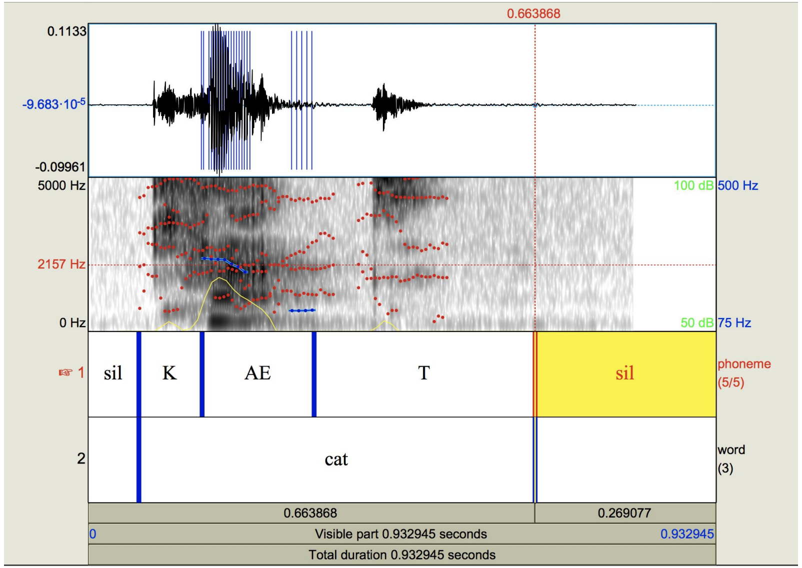
“if music be the food of love … ”

ASR: Audio & Text



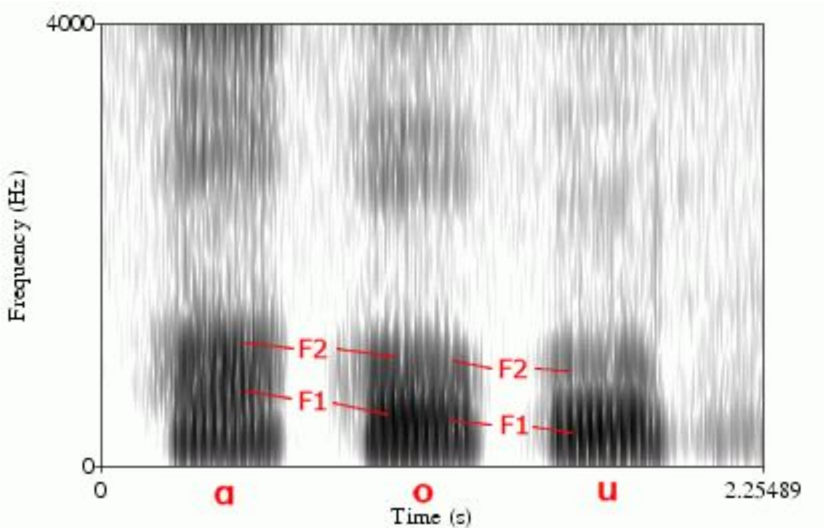
Speech vs. Image Processing





Speech vs. Image Processing

- Spectrogram reading: human and machine



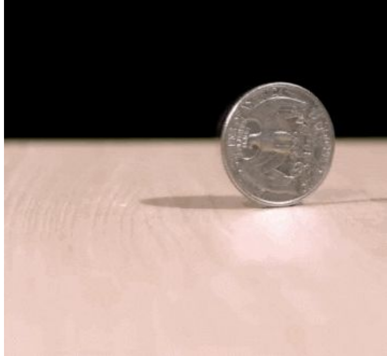
Thinking break (2 min)

What is ***similar / different / difficult*** about **speech** vs. **image** in machine learning solutions?

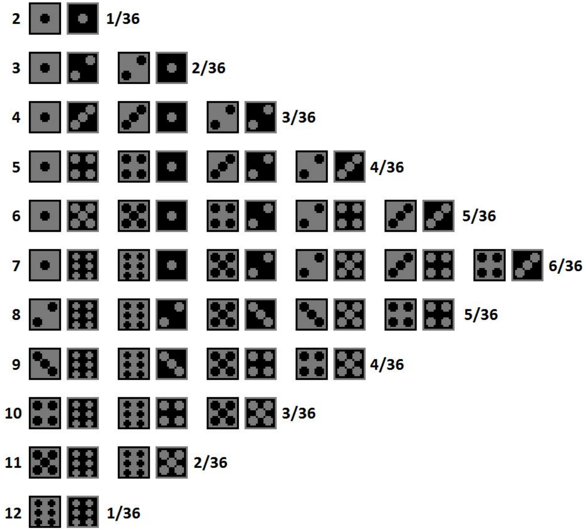
Consider other possible applications of ASR technologies in the next 5, 10 years?

About Probability

- Observation:
  - flipping a coin 1000 times (or roll the dice)
  - ~ 50% front side
  - ~ 50% back side



- Intuition/Hypothesis
  - Predict the outcome **likelihood**
  - Make choices
  - Probability theory



The Linda Problem

Linda is thirty-one years old, single, outspoken and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in antinuclear demonstrations."

A. Linda is a teacher in an elementary school.
B. Linda works in a bookstore and takes yoga classes.
C. Linda is active in the feminist movement.
D. Linda is a psychiatric social worker.
E. Linda is a member of the League of Women Voters.
F. Linda is a bank teller.
G. Linda is an insurance salesperson.
H. Linda is a bank teller and is active in the feminist movement.

The Linda Problem

Linda is thirty-one years old, single, outspoken and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in antinuclear demonstrations."

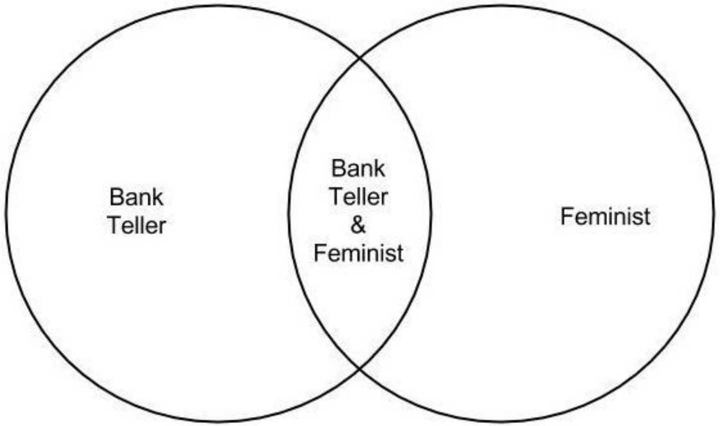
Which is more probable?

(A) Linda is a bank teller.

(B) Linda is a bank teller and is active in the feminist movement.

The Linda Problem

Conjunction fallacy: A - H

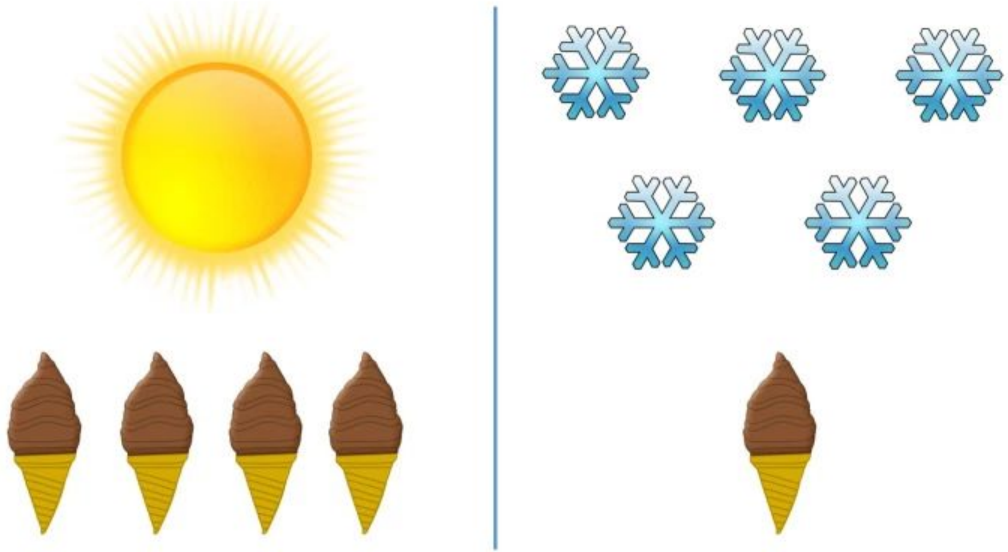


Bayes Rule:

- Base rate or prior: P(W) = P(Linda is a bank teller)
- Likelihood: P(O|W) = P(a bank teller who is also a feminist | someone is a bank teller)

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

About Ice-cream and Probability



About Ice-cream and Probability

- **Observation**:
  - Correlation?

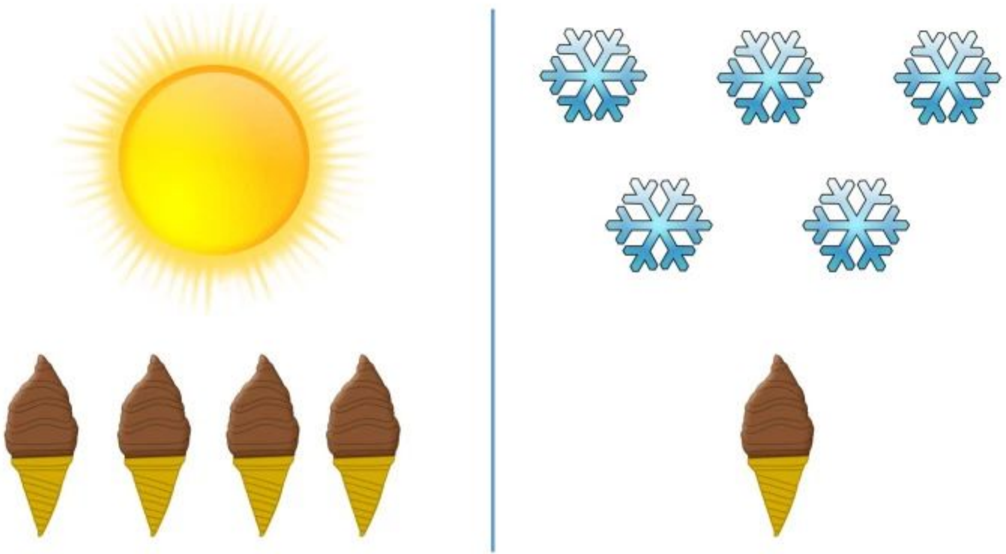
	D1	D2	...										Dn
No. of ice-creams	2	3	4	3	1	5	6	3	1	1	0	0	0	0
Weather	H	H	H	C	C	H	H	C	C	C	C	C	C	C

- **Hypothesis**:
  - Probability modeling
- **Estimation**:

	Day 1	Day 2	Day 3
No. of ice-creams	1	4	1
Weather: hot/cold	?	?	?

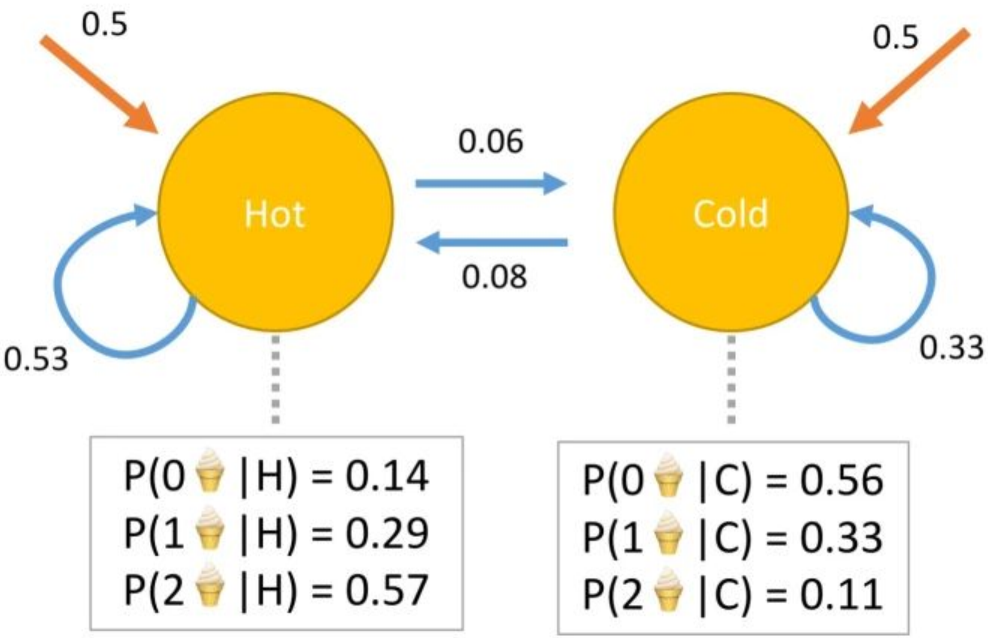
About Ice-cream and Probability

- Observation A
  - P(2 ice-creams | hot day) = 0.29
  - P(1 ice-cream | cold) = 0.33
- Observation B
  - P(C) = 0.5
  - P(H) = 0.5
  - P(H | C) = 0.08
  - P(C | H) = 0.06
  - P(H | H) = 0.53
  - ...



About Ice-cream and Probability

- State-graph: observations, transitions - hidden Markov model (HMM)



Predict the Weather with Ice-cream?

- P(H, H, C) = P(0|H) * P(H) + P(4|H) * P(H|H) + P(1|C) * P(C|H, H) = **0.089**
- P(H, H, H) = 0.043
- P(H, C, H) = 0.026
- p(C, H, H) = …
- P(C, C, C) = ...

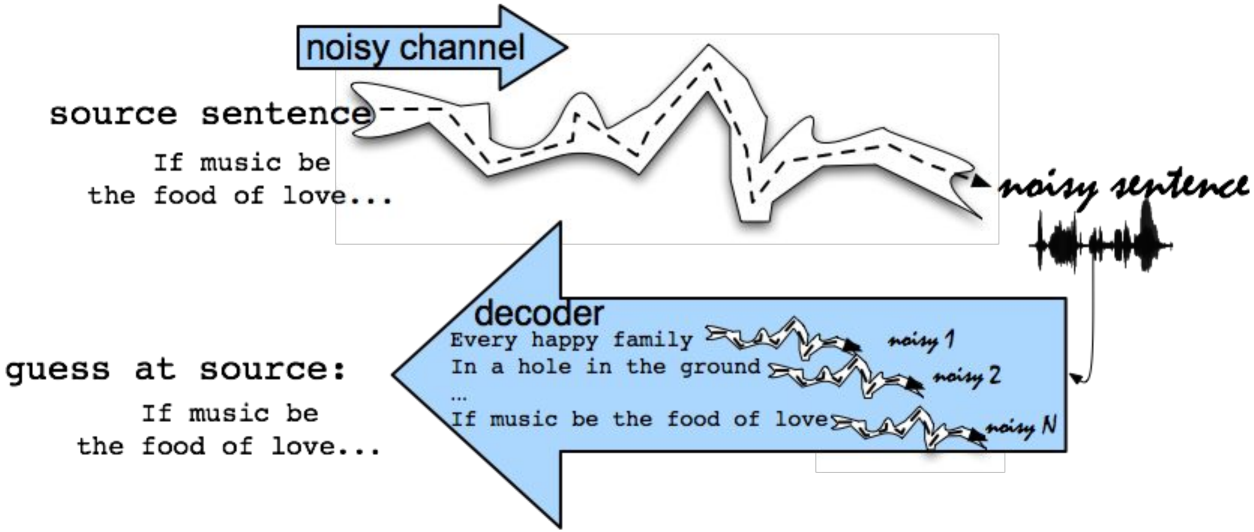
	Day 1	Day 2	Day 3
No. of ice-creams	1	4	1
Weather: hot/cold	?	?	?

Thinking break (2 min)

Consider other possible applications of the probability theory, e.g. Bayes rule?

Formulate ASR

- Assumption: left-to-right beads on a string
- What is the most likely sentence ***W*** out of all sentences in the language ***L*** given some acoustic input ***O***?



Formulate ASR: statistical model

- What is the most likely sentence ***W*** = *w1,w2,w3,…,wn*, out of all sentences in the language ***L*** given some acoustic input ***O*** = *o1,o2,o3,…,ot* ?

$$\hat{W} = \arg\max_{W \in L} P(W \mid O)$$

- Rewrite with Bayes Rule
  - denominator is the same for each candidate sentence W, ignored

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

  - AM likelihood: P (O | phone) = P(waveform | phone)
  - PM likelihood: P (phone | W) = P(phone | words)
  - LM prior: P (W) = P(words or sentence)

$$\hat{W} = \arg\max_{W \in L} \overbrace{P(O \mid W)}^{\text{likelihood}}\overbrace{P(W)}^{\text{prior}}$$

Machine Learning: signal to label

Signal

1. Features
2. **Model (s)**
3. Decode

Label

Deep learning:

- Train, validation, test
- K-fold cross validation
- Backpropagation
- Loss function
- Neural networks
- Hyper-parameters
- …

ASR: audio to text

Audio

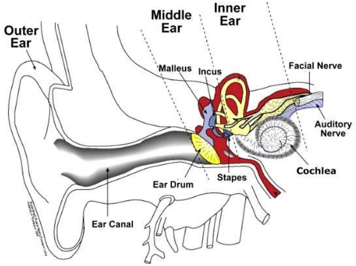
1. Features
2. **Acoustic** model - the sound
3. **Pronunciation** model - the dictionary
4. **Language** model - the words
5. Decode - Bayes Rule

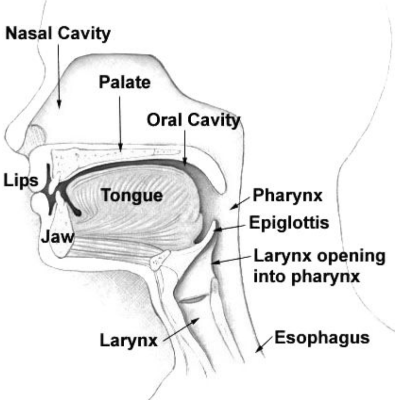
Text

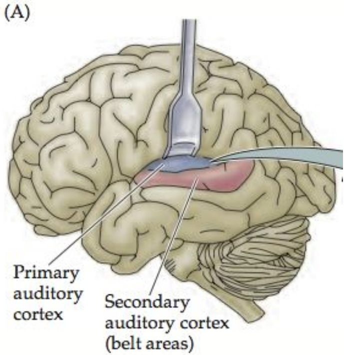
ASR & HSR

Similarities?

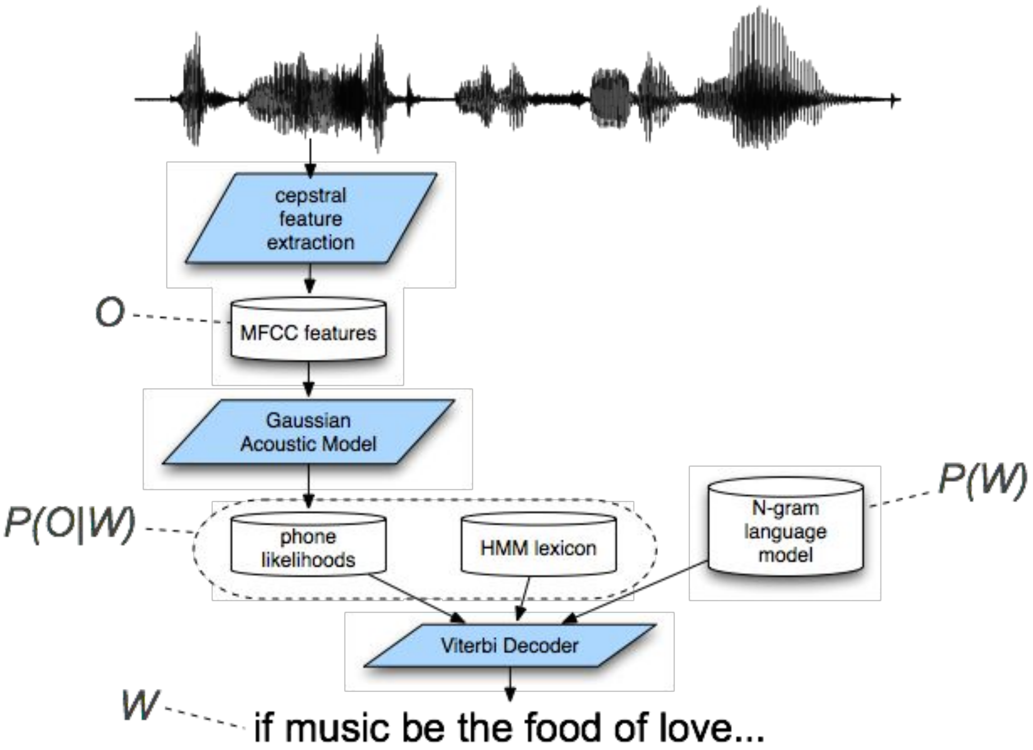
- AM: peripheral auditory system ~ the ear(s)
- PM: articulatory system ~ the mouth
- LM: Auditory nervous system - the brain



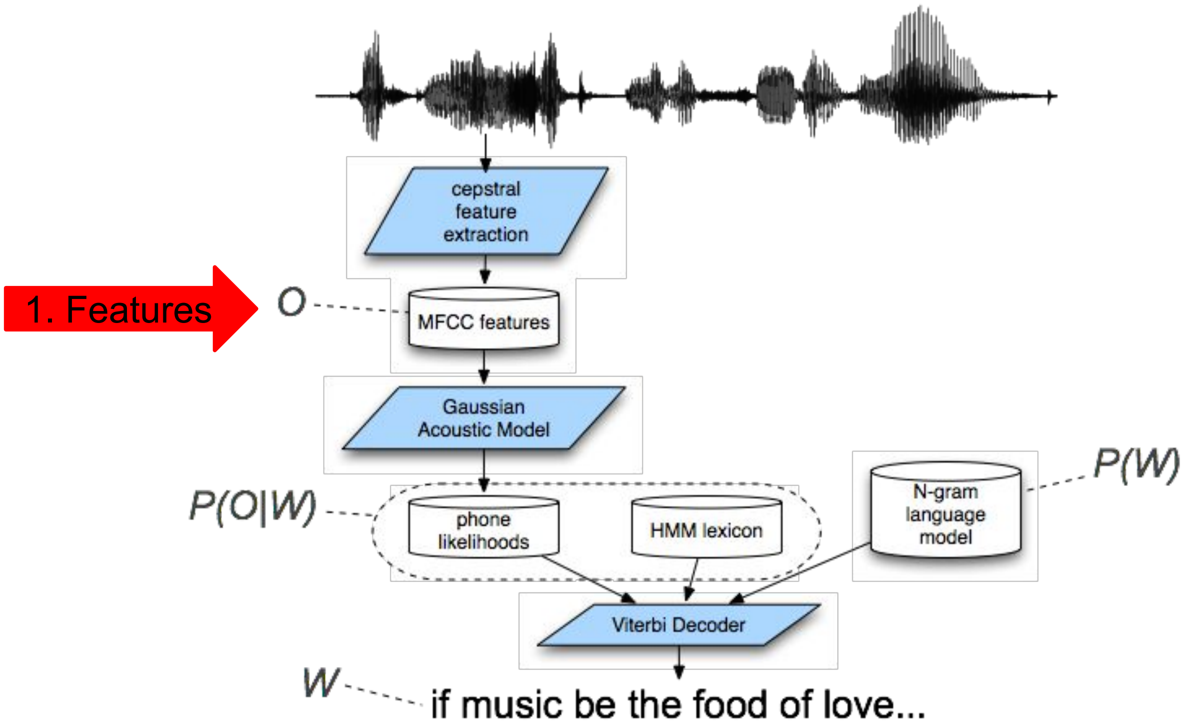




(Traditional) ASR System



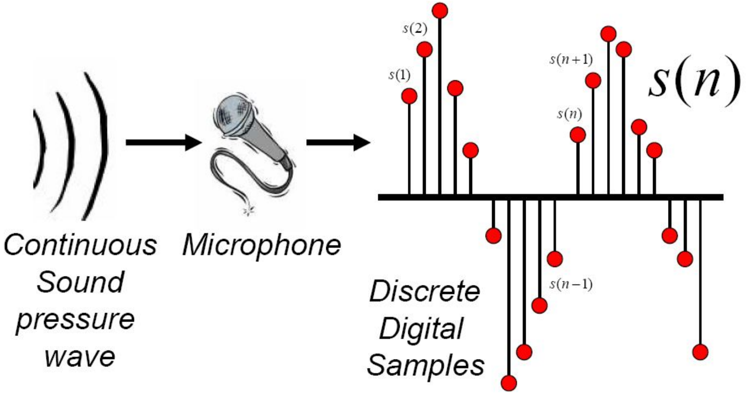
1. Feature extraction
2. Acoustic model
3. Lexicon/Pronunciation model
4. Language model
5. Decoder


1. Features
O
cepstral
feature
extraction
MFCC features
Gaussian
Acoustic Model
P(O|W)
phone
likelihoods
HMM lexicon
N-gram
language
model
P(W)
Viterbi Decoder
W
if music be the food of love...

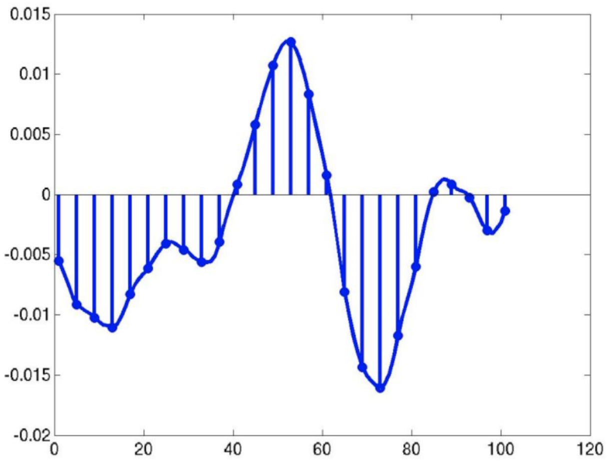
ASR: Feature Extraction-I

❖ Digitizing Speech Signal

a. Sampling: 16 kHz microphone, 8 kHz telephone, human speech < 10 kHz
b. Quantization: 8- or 16-bit
c. Formats; Headers: raw, Microsoft wav, Sun au

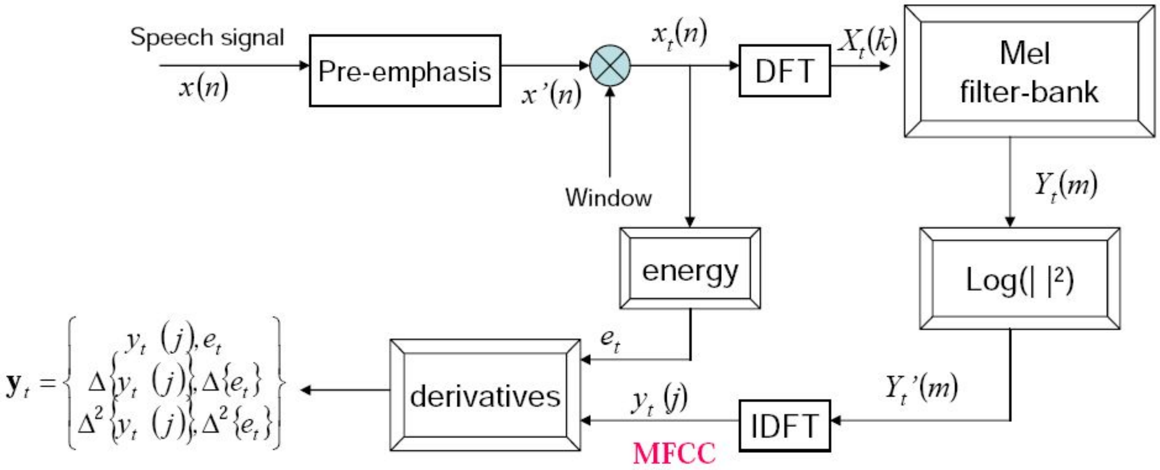


Analog vs discrete samples



ASR: Feature Extraction-II

- Commonly used Mel-Frequency Cepstral Coefficients (MFCCs)
  - Most widely used spectral representation in ASR
  - Knowledge about human speech perception and production



ASR: Feature Extraction-III

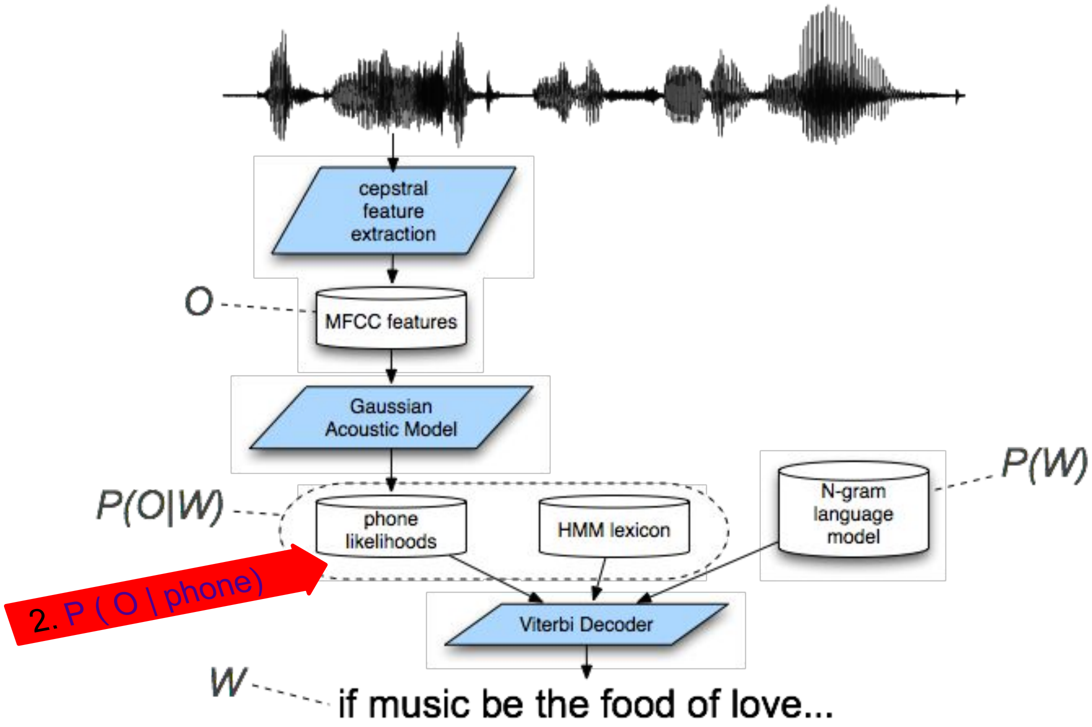
- Typical MFCCs
  - Window size: 25ms
  - Window shift: 10ms
  - Pre-emphasis coefficient: 0.97
  - MFCC:
    - 12 MFCC (mel frequency cepstral coefficients)
    - 1 energy feature
    - 12 delta MFCC features
    - 12 double-delta MFCC features
    - 1 delta energy feature
    - 1 double-delta energy feature
  - Total 39-dimensional features

- Properties:
  - Efficient to compute
  - Perceptual Mel frequency scale
  - Separates the source and filter

Thoughts…

What are other features that you can think of to represent speech, effectively if possible?

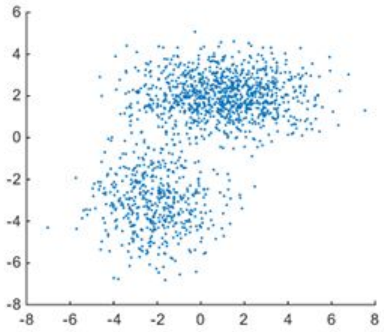
1. Feature extraction
2. Acoustic model
3. Lexicon/Pronunciation model
4. Language model
5. Decoder


cepstral feature extraction
O
MFCC features
Gaussian Acoustic Model
P(O|W)
phone likelihoods
HMM lexicon
N-gram language model
P(W)
2. P (O | phone)
Viterbi Decoder
W
if music be the food of love...

ASR: Acoustic Modelling-I

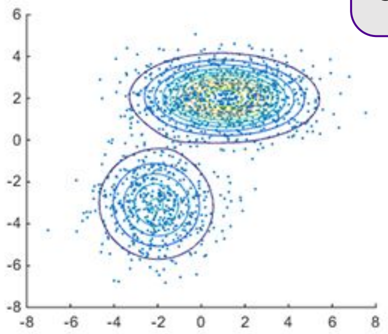
- Acoustic likelihood as Gaussians Mixture Models (GMMs)
  - Assume the possible values of the observation feature vector ***O*** are normally distributed.
  - Fit the observation likelihood with GMMs
    - Learn Gaussian over the distribution of MFCCs
    - Estimate probability of observation ***Ot***
    - Suited for speech recognition

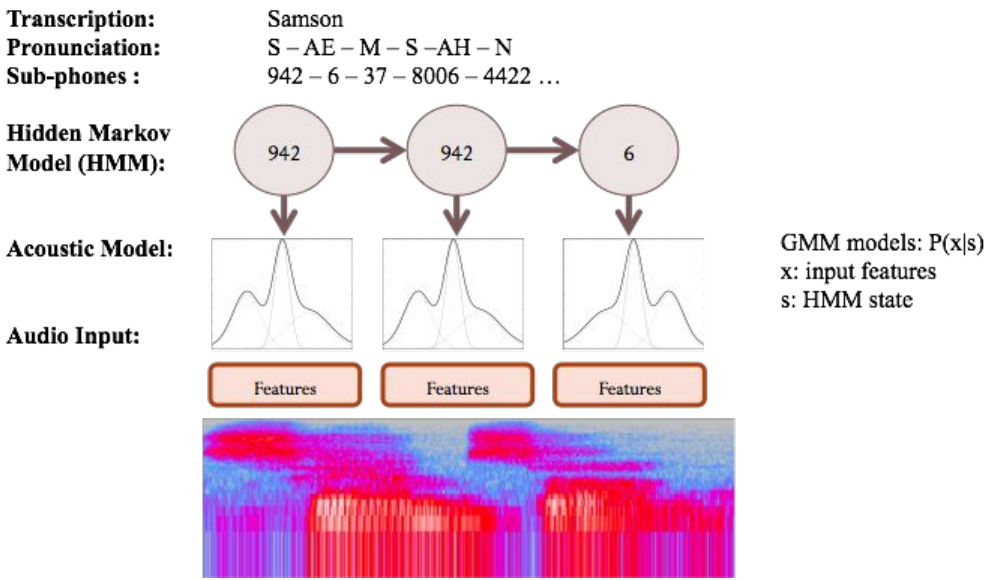
Example of data points in 2-D space



Fit GMMs

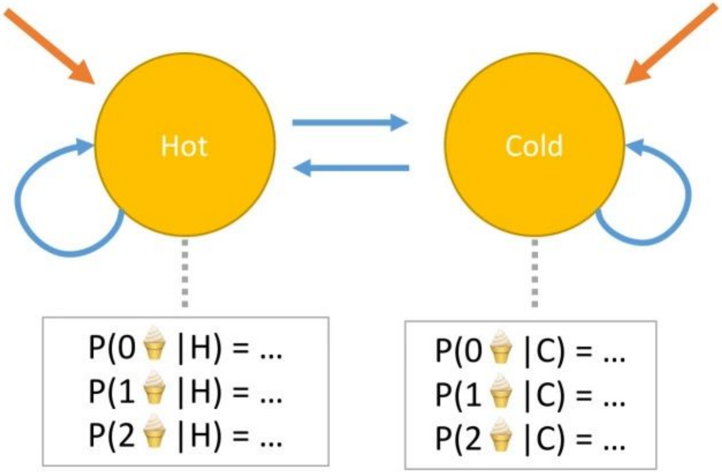
GMM Parameters:
Mean: μ1
Standard deviation: σ1




Transcription:
Samson
Pronunciation:
S – AE – M – S –AH – N
Sub-phones :
942 – 6 – 37 – 8006 – 4422 …
Hidden Markov Model (HMM):
942
942
6
Acoustic Model:
Audio Input:
Features
Features
Features
GMM models: P(x|s)
x: input features
s: HMM state

ASR: Acoustic Modelling-II

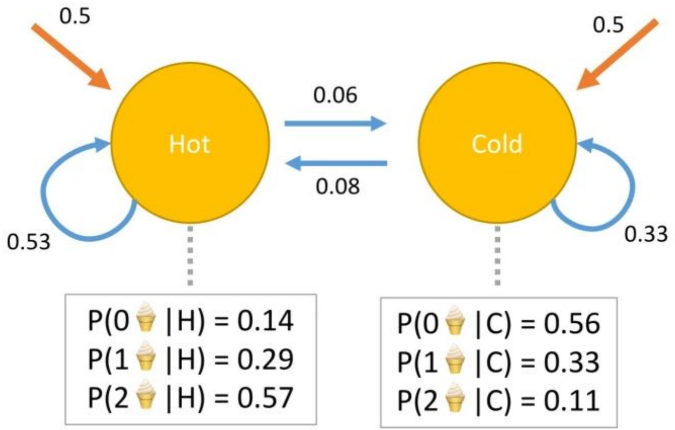
- GMM-HMM: the learning problem
  - Given an observation sequence O and the possible states in the HMM, learn the HMM transition parameters A and emission/observation parameters B
  - Forward-Backward algorithm



ASR: Acoustic Modelling-II

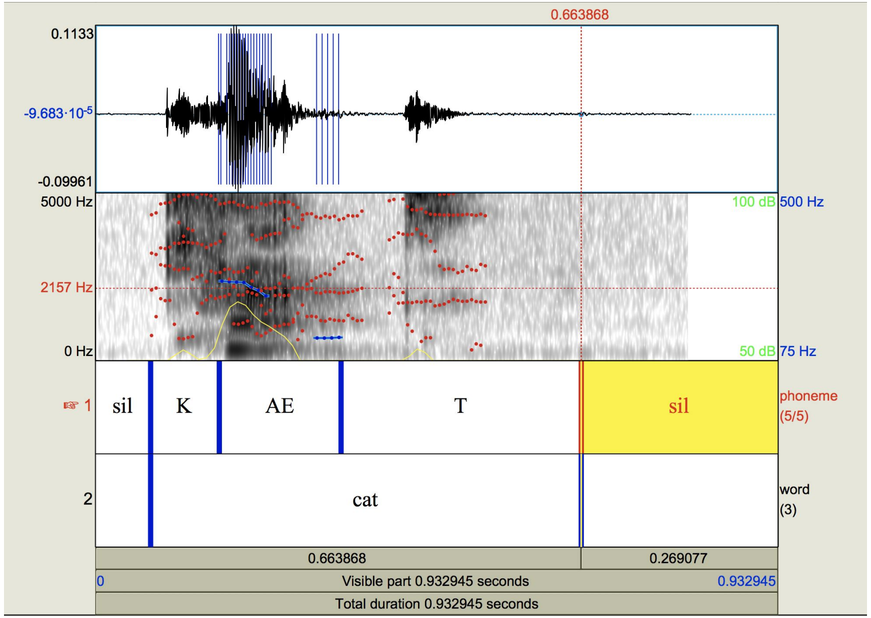
❖ Typical Training procedure in ASR

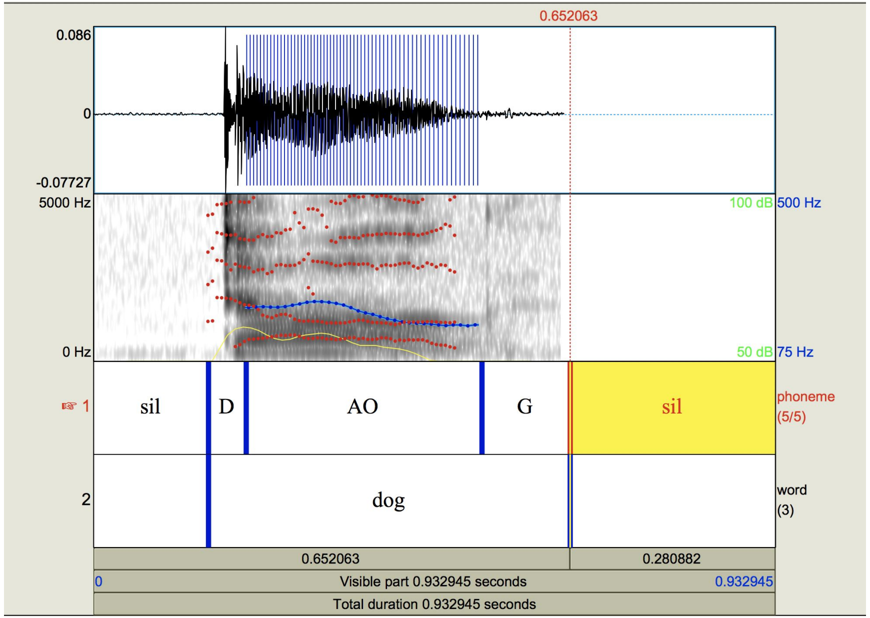
a. Initialization
b. Generate a forced alignment with existing model
c. Create new observation models from updated alignments
d. Repeat



ASR: audio

ASR: text


0.663868
0.1133
-9.683·10-5
-0.09961
5000 Hz
2157 Hz
0 Hz
100 dB
500 Hz
50 dB
75 Hz
sil
K
AE
T
sil
phoneme
(5/5)
1
2
cat
word
(3)
0.663868
0.269077
0
Visible part 0.932945 seconds
0.932945
Total duration 0.932945 seconds


0.652063
0.086
0
-0.07727
5000 Hz
0 Hz
100 dB
500 Hz
50 dB
75 Hz
sil
D
AO
G
sil
phoneme
(5/5)
1
2
dog
word
(3)
0.652063
0.280882
0
Visible part 0.932945 seconds
0.932945
Total duration 0.932945 seconds

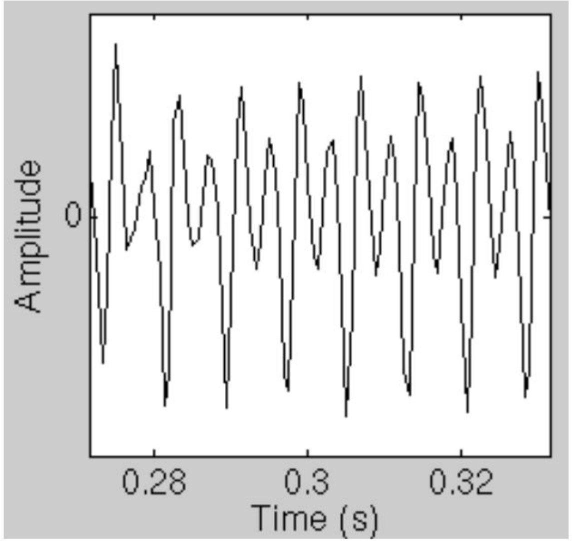
Thoughts…

What are the major difficulties when recognizing ‘cat’, ‘dog’ in this ASR example?

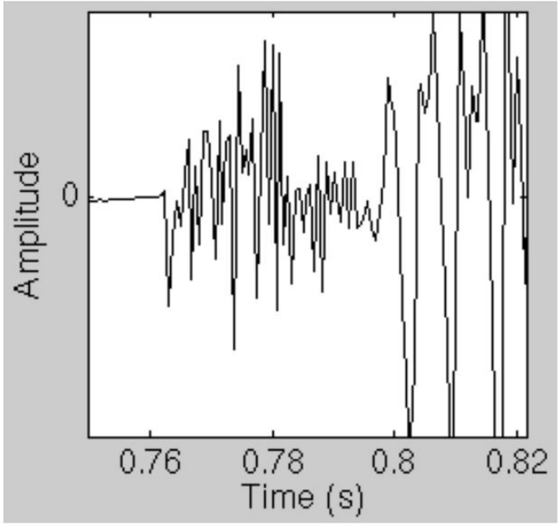
- Audio:
  - Pronunciation variations
  - Speaker variations
  - Coarticulation (in sentences)
  - Background noises (in actual situations)
- Text:
  - Collect, annotate, store audio with text description and annotation (task-specific)
  - Ambiguities: phoneme annotation

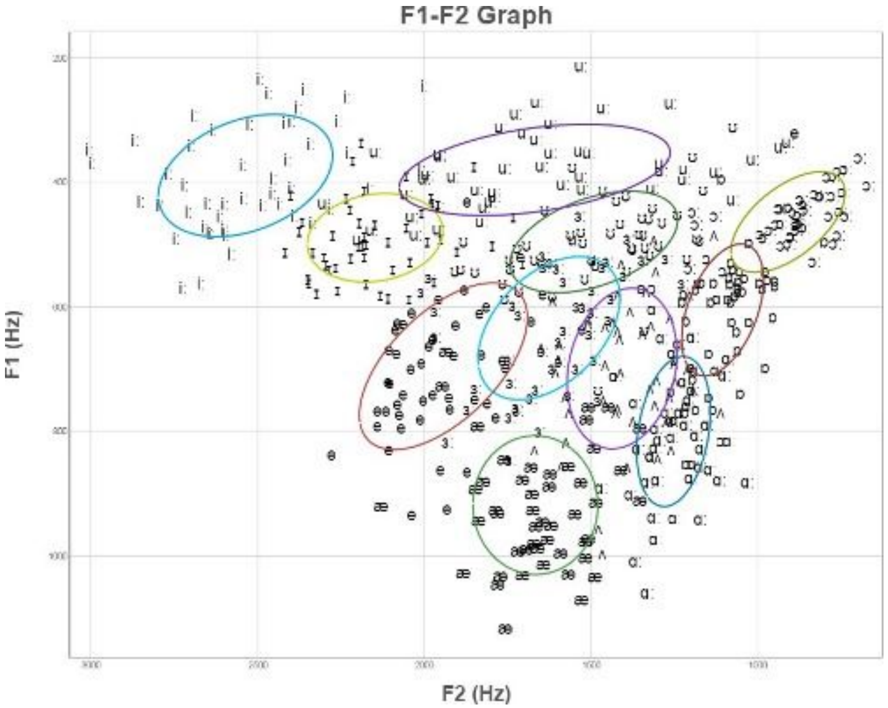
Break (15 minutes)

“e” in He (voiced: periodic)

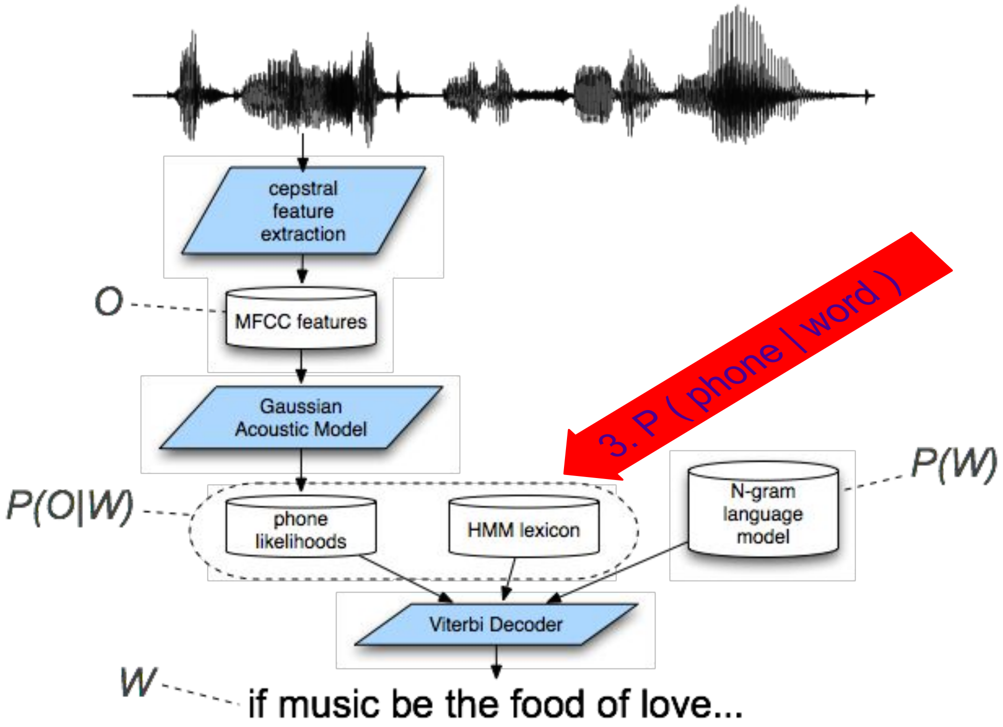


”t” in ”taboos” (unvoiced: ”noisy”)




F1-F2 Graph
F1 (Hz)
F2 (Hz)

1. Feature extraction
2. Acoustic model
3. Lexicon/Pronunciation model
4. Language model
5. Decoder


cepstral feature extraction
O
MFCC features
Gaussian Acoustic Model
P(O|W)
phone likelihoods
HMM lexicon
N-gram language model
P(W)
Viterbi Decoder
W
if music be the food of love...
3. P (phone | word)

English Pronunciations: phonetics (recap)

- Phoneme
  - The smallest linguistic unit which may change the meaning (kill vs. kiss)
  - The realization of phonemes are called phones
  - Phonemes are combined to form larger entities such as words

- Speech sounds
  - Consonant vs. vowel:
    - Consonants involve an obstruction in air stream above the glottis.
  - Voiced vs. voiceless:
    - Voiced if vocal cords vibrate
  - Nasal vs. oral
    - Nasal if air travels through nasal cavity and oral cavity closed

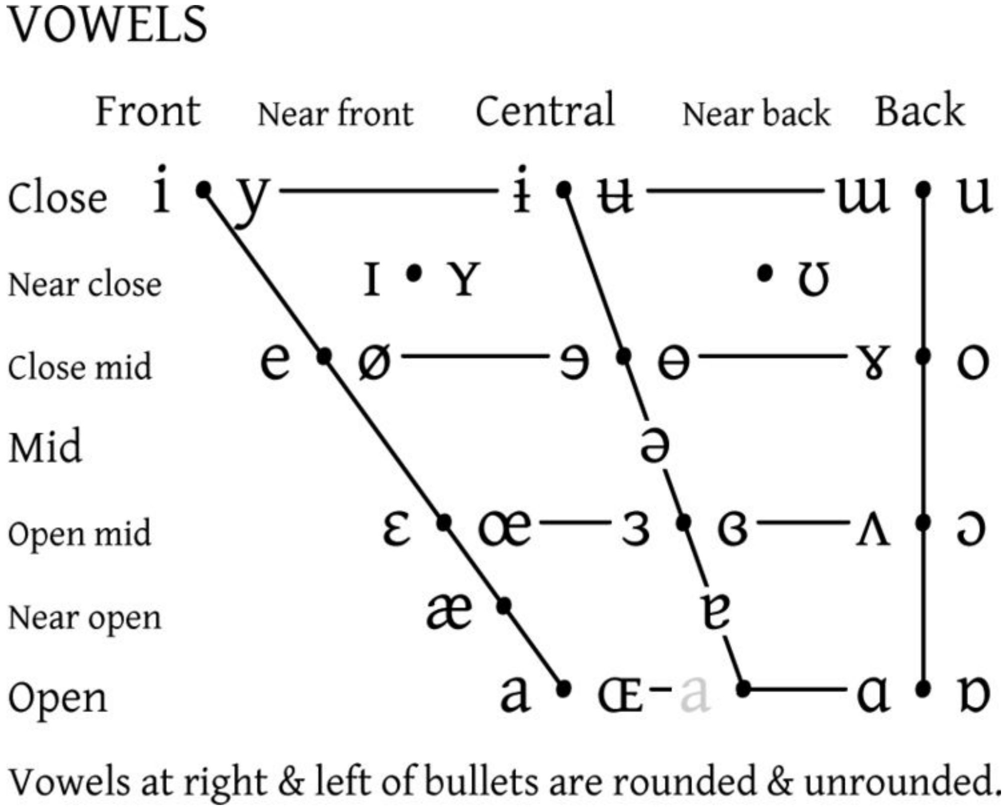
English Pronunciations:IPA

CONSONANTS (PULMONIC)

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap				ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Where symbols appear in pairs, the one to the right represents a voiced consonant. Shaded areas denote articulations judged impossible.

English Pronunciations:IPA



Annotations: IPA, ARPABET, X-SAMPA ...

Vowels[3]

ARPABET		IPA	Example(s)
1-letter	2-letter		
a	AA	ɑ	balm, bot
@	AE	æ	bat
A	AH	ʌ	butt
c	AO	ɔ	story
W	AW	aʊ	bout
x	AX	ə	comma
N/A	AXR[4]	ɚ	letter
Y	AY	aɪ	bite
E	EH	ɛ	bet
R	ER	ɝ	bird
e	EY	eɪ	bait

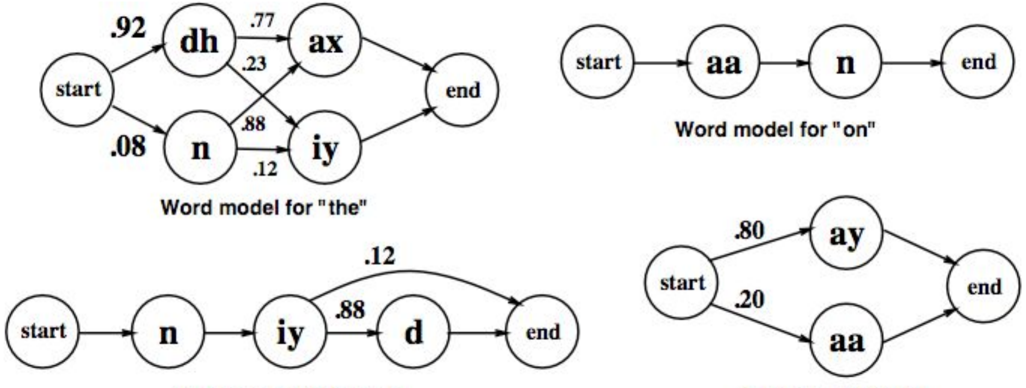
Consonants[3]

ARPABET		IPA	Example
1-letter	2-letter		
b	B	b	buy
C	CH	tʃ	China
d	D	d	die
D	DH	ð	thy
F	DX	ɾ	butter
L	EL	l̩	bottle
M	EM	m̩	rhythm
N	EN	n̩	button
f	F	f	fight
g	G	g	guy
h	HH *or* H[4]	h	high

ASR: Pronunciation Modeling

- Lexicon or Dictionary: A list of words, each with a pronunciation of phones
  - E.g., CMU dictionary: 127K words
    - ‘need’ : ['n', 'iy1', 'd']
    - ‘recognition’ : ['r', 'eh2', 'k', 'ah0', 'g', 'n', 'ih1', 'sh', 'ah0', 'n']

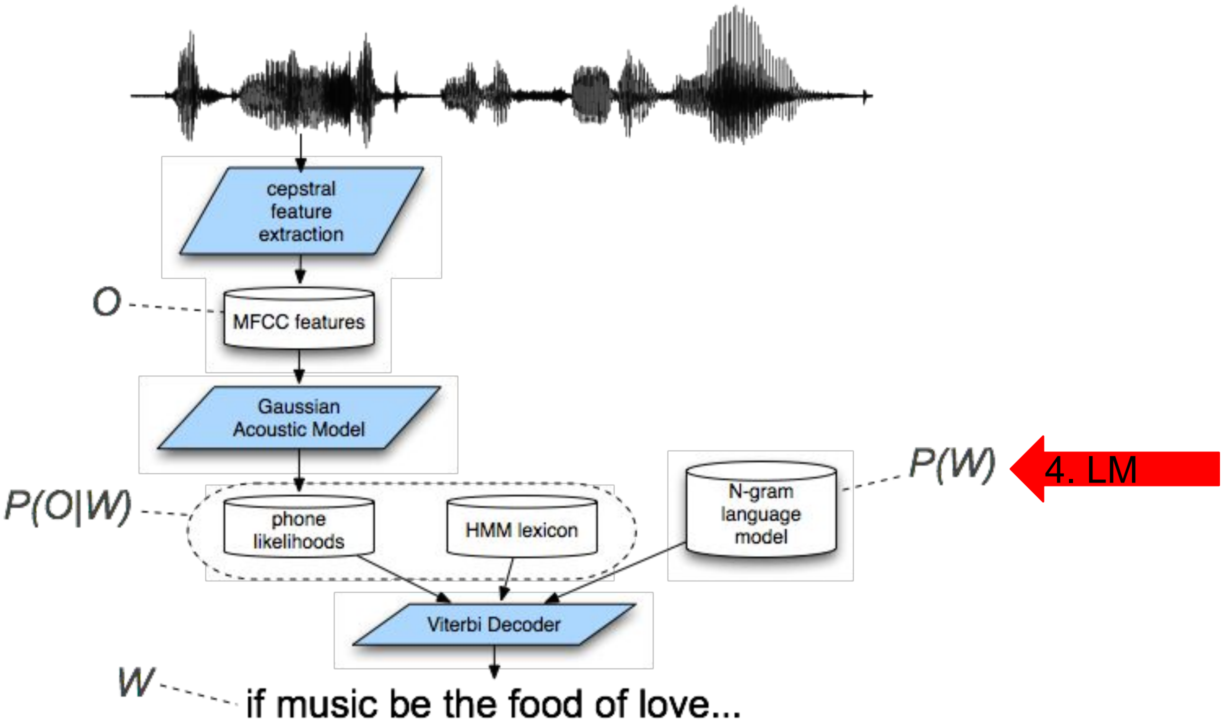
- Markov model for pronunciations P(phone | word)



Word model for "the"
Word model for "on"
Word model for "need"
Word model for "I"

ASR

1. Feature extraction
2. Acoustic model
3. Lexicon/Pronunciation model
4. Language model
5. Decoder


cepstral feature extraction
O
MFCC features
Gaussian Acoustic Model
P(O|W)
phone likelihoods
HMM lexicon
N-gram language model
P(W)
4. LM
Viterbi Decoder
W
if music be the food of love...

ASR: Probabilistic Language Modeling

- N-grams
  - Approximate the probability of a sentence or a sequence of words ***P(W)***
  - Also used in machine translation, spell correction, speech recognition, etc.
- **Chain Rule**

P(if, music, be, the, food, of, love)

= P(if) * P(music|if) * P(be|if, music) * P(the|if, music, be) *...P(love|if, …, of)

Or ≅ P(if) * P(music|if) * P(be|music) * P(the|be) *...P(love|of) 2-gram or bigram LM

Or ≅ P(if) * P(music|if) * P(be|if, music) * P(the|music, be) *...P(love|food, of) 3-gram or trigram LM

Where ***P(music|if) = count(if music) / count(if)***

ASR: Probabilistic Language Modeling

- Case study: bigram estimates of sentence probabilities
  - Out of 9222 sentences (artificial sentences for illustration purpose)
  - Unigrams

if	music	be	the	food	of	love	then
2533	927	2417	746	158	1093	341	278

  - Bigrams

	if	music	be	the	food	of	love	then
if	5	827	0	9	0	0	2	2
music	2	0	608	1	6	6	5	1
be	2	0	4	686	2	0	6	211
the	0	0	2	0	16	2	42	0
food	1	0	0	0	0	82	1	0
of	15	0	15	0	1	4	0	0
love	2	0	0	0	0	1	0	0
then	1	0	1	0	0	0	0	0

ASR: Probabilistic Language Modeling

- Case study: bigram estimates of sentence probabilities
  - Usually done in log domain to avoid underflow
  - Adding is faster than multiplying
  - Google Book N-grams https://books.google.com/ngrams

Recall: P(music|if) = count(if music) / count(if)

P(<S> if music be the food of love </s>)

= P(if|<s>) * P(music|if) * P(be|music) * P(the|be) *... * P(love|of)

= 2533/8486 * 827/2533 * 608/927 * … * 1/341

= 0.000031

- Typical Training procedure of LM
  - Training of LMs on a set of sentences
  - Evaluate the LM with perplexity on held-out data
  - Deal with data sparsity: smoothing, interpolation, backoff ...

Thoughts…

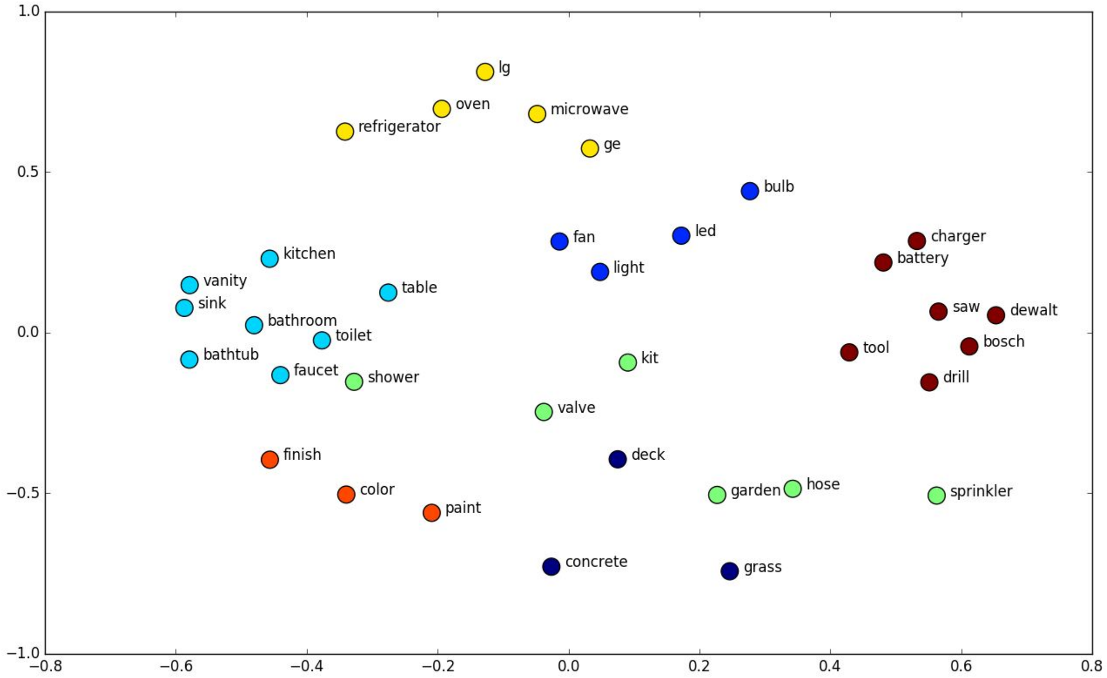
- What are other uses of LMs?
- Do you think LMs have limitations?

(Neural) Language Modeling

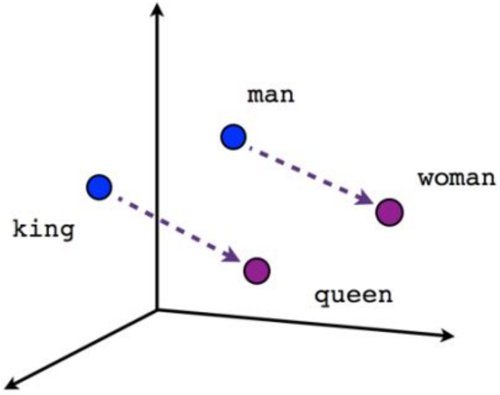
France + Paris = Italy + ?

Cold + Hot = Big + ?

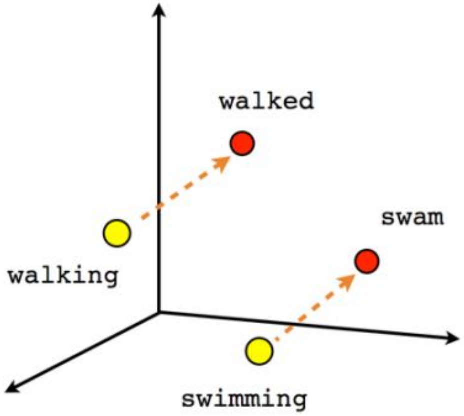
LM: Word Embedding



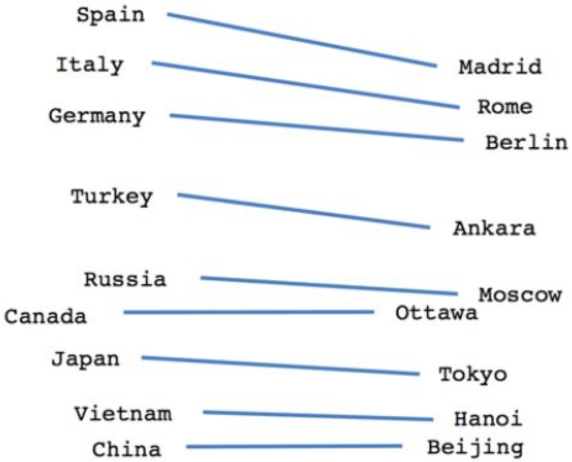
LM: Word Embedding Vectors



Male-Female

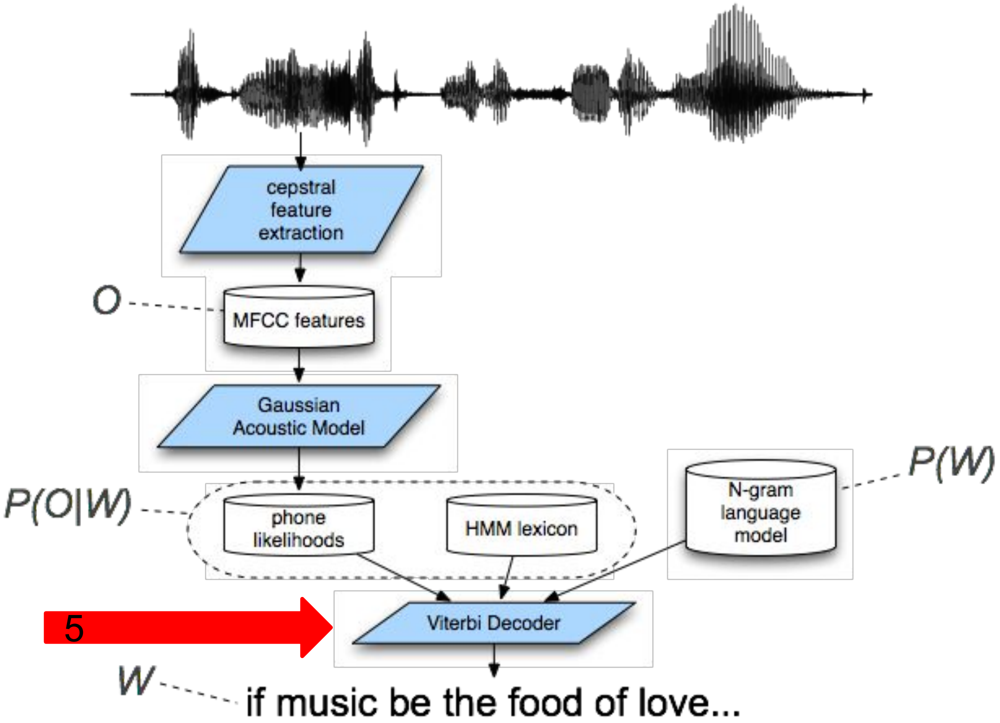


Verb tense



Country-Capital

1. Feature extraction
2. Acoustic model
3. Lexicon/Pronunciation model
4. Language model
5. Decoder


cepstral feature extraction
O
MFCC features
Gaussian Acoustic Model
P(O|W)
phone likelihoods
HMM lexicon
N-gram language model
P(W)
5
Viterbi Decoder
W
if music be the food of love...

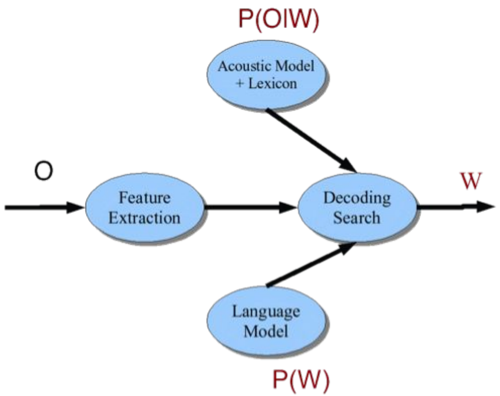
ASR: Decoding

- What we are searching for

likelihood prior

$$\hat{W} = \underset{W \in L}{\operatorname{argmax}} \underbrace{P(O \mid W)}_{\text{likelihood}} \underbrace{P(W)}_{\text{prior}}$$

- Example: viterbi beam search
  - How to weigh Acoustic-, Pronunciation-, and Language-models
  - Most common search algorithm for ASR
  - Generates N-best lists



ASR: Decoding

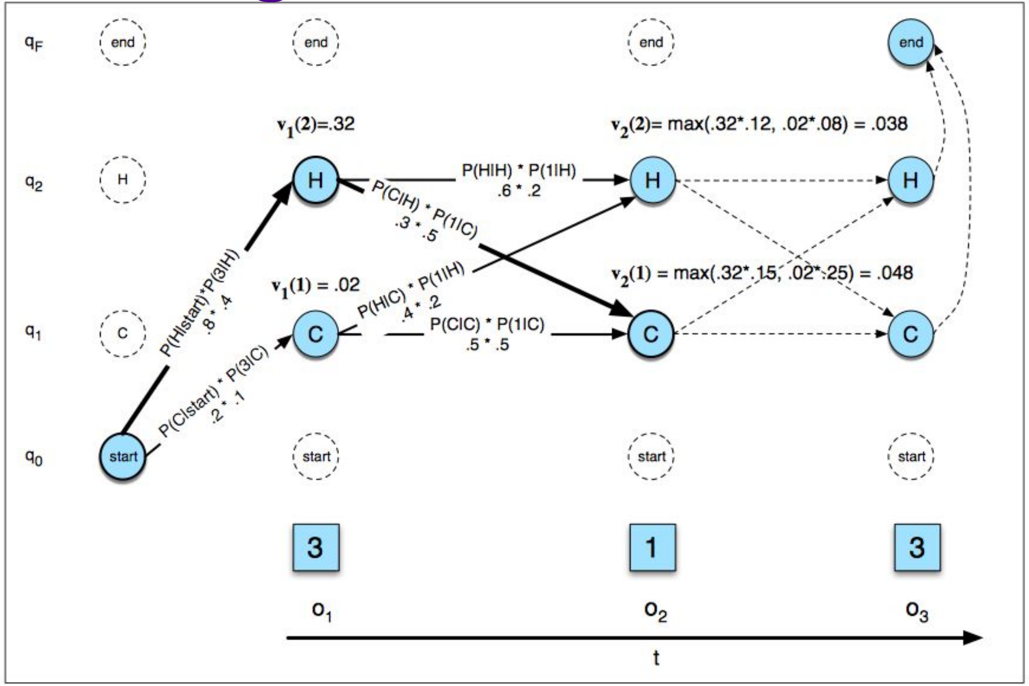


Figure 9.10 The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events *3 1 3*. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $v_t(j)$ for two states at two time steps. The computation in each cell follows Eq. 9.19: $v_t(j) = \max_{1\leq i\leq N-1} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$. The resulting probability expressed in each cell is Eq. 9.18: $v_t(j) = P(q_0, q_1, \ldots, q_{t-1}, o_1, o_2, \ldots, o_t, q_t = j|\lambda)$.

Source: (Jurafsky, 2008)

ASR: Decoding

❖ Viterbi beam search

function VITERBI(*observations* of len T, *state-graph* of len N) **returns** *best-path*

create a path probability matrix *viterbi[N+2,T]*

for each state s **from** 1 **to** N **do** ; initialization step

$viterbi[s,1] \leftarrow a_{0,s} * b_s(o_1)$

$backpointer[s,1] \leftarrow 0$

for each time step t **from** 2 **to** T **do** ; recursion step

for each state s **from** 1 **to** N **do**

$viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$

$backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s}$

$viterbi[q_F,T] \leftarrow \max_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$; termination step

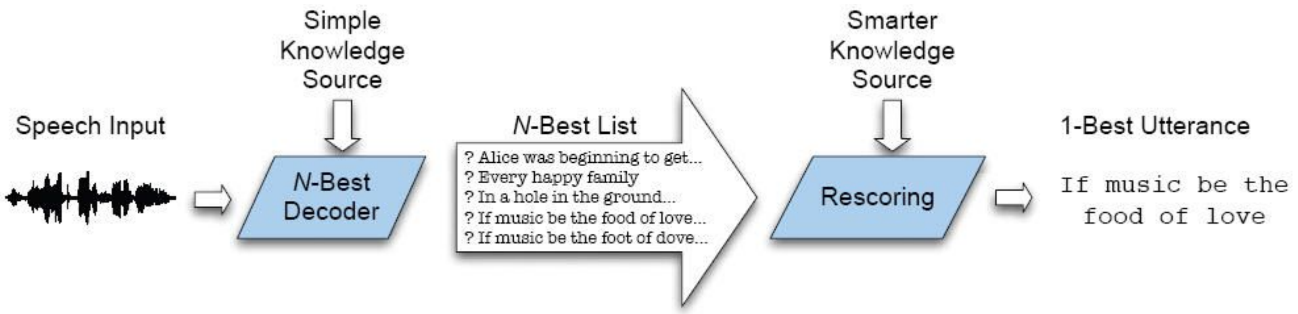
$backpointer[q_F,T] \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$; termination step

return the backtrace path by following backpointers to states back in time from $backpointer[q_F,T]$

Figure 9.11 Viterbi algorithm for finding optimal sequence of hidden states. Given an observation sequence and an HMM $\lambda = (A,B)$, the algorithm returns the state path through the HMM that assigns maximum likelihood to the observation sequence. Note that states 0 and q_F are non-emitting.

ASR: Decoding

- Viterbi beam search
  - Problem: hard to integrate priors or knowledge sources, e.g., trigram grammars
  - Solution: find multiple hypotheses, N-best sentences list, and rescore them



AM logprob	LM logprob
-7193.53	-20.25
-7192.28	-21.11
-7221.68	-18.91
-7189.19	-22.08
-7198.35	-21.34
-7220.44	-19.77
-7205.42	-21.50
-7195.92	-21.71
-7217.34	-20.70
-7226.51	-20.01

Slide from John-Paul Hosom

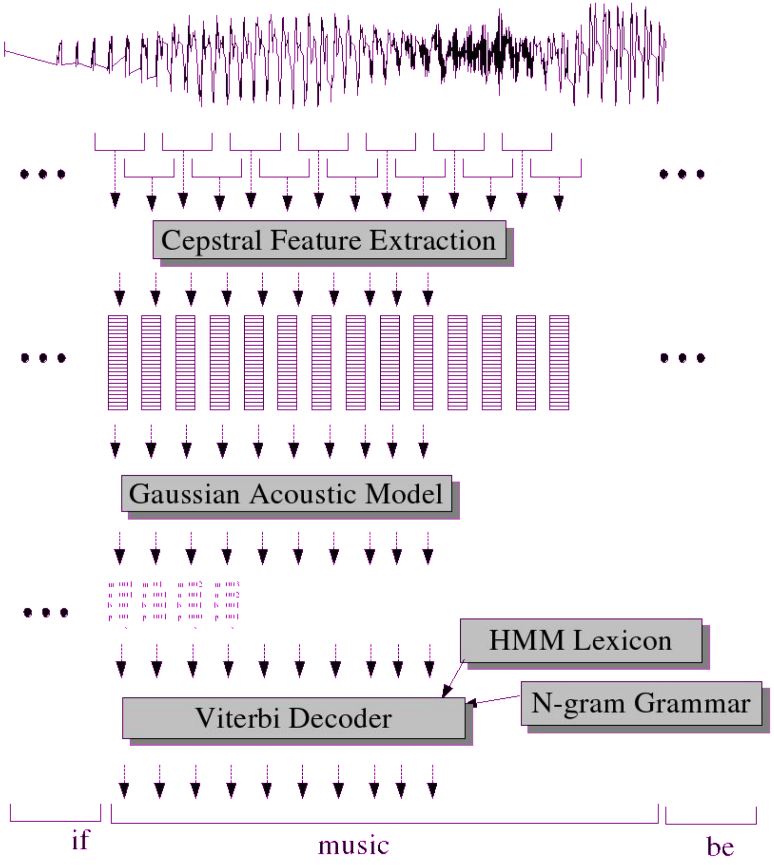
1.1^20 = 1.1 * 1.1 * 1.1 * … *1.1 = 6.7274

0.9^20 = 0.9 * 0.9 * 0.9 * 0.9 * … * 0.9 = 0.1216

ASR: audio to text

1. Feature extraction
   - 39 ***MFCCs***
2. Acoustic model
   - GMMs for computing ***p(o)***
3. Pronunciation model:
   - 42 **phonemes**
4. Language model
   - N-grams for computing ***p(w)***
5. Decoder
   - Viterbi algorithm to find best path

$$\hat{W} = \underset{W \in L}{\operatorname{argmax}} \underbrace{P(O \mid W)}_{\text{likelihood}} \underbrace{P(W)}_{\text{prior}}$$



Thinking break (2 min)

What are the limitations of the traditional approach to ASR?

Which part of the ASR system would you improve, or get rid of?

Have you thought about deep neural networks?

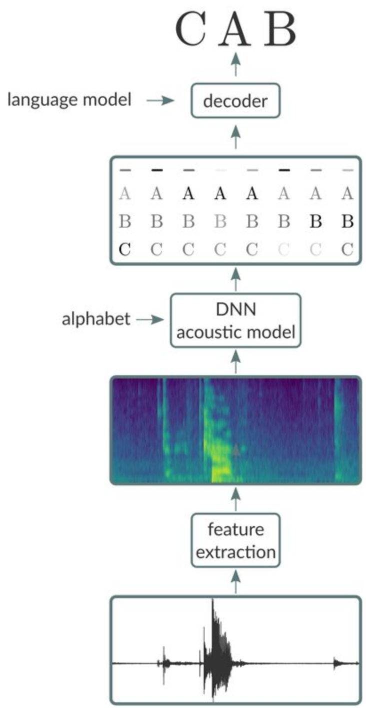
Thoughts…

- ❖ Advantages of GMM-HMM based ASR systems
  - ➢ Probabilistic modeling
  - ➢ Explicit modeling
- ❖ Disadvantages
  - ➢ Assumption of ‘beads on a string’
  - ➢ AM: speech variability
  - ➢ LM: long-term dependencies of context
- ❖ Challenges
  - ➢ Accurate word-, phoneme-level labeling of speech utterances
  - ➢ Rare languages or dialects

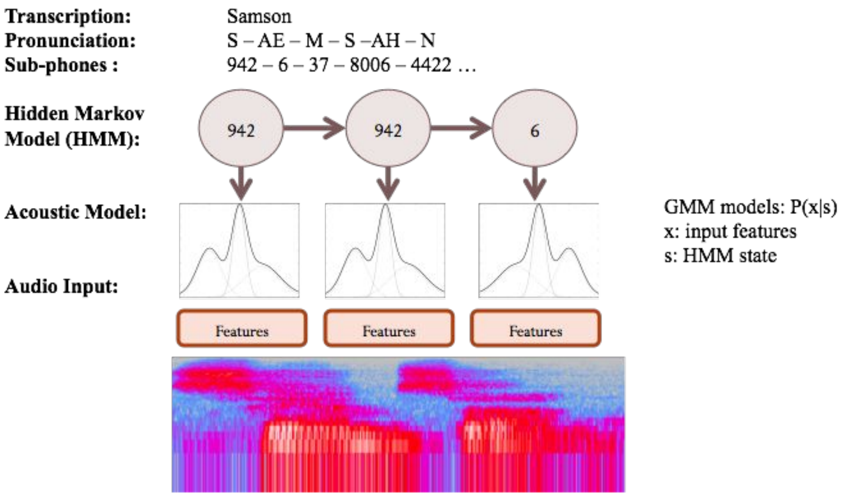
And… deep learning?

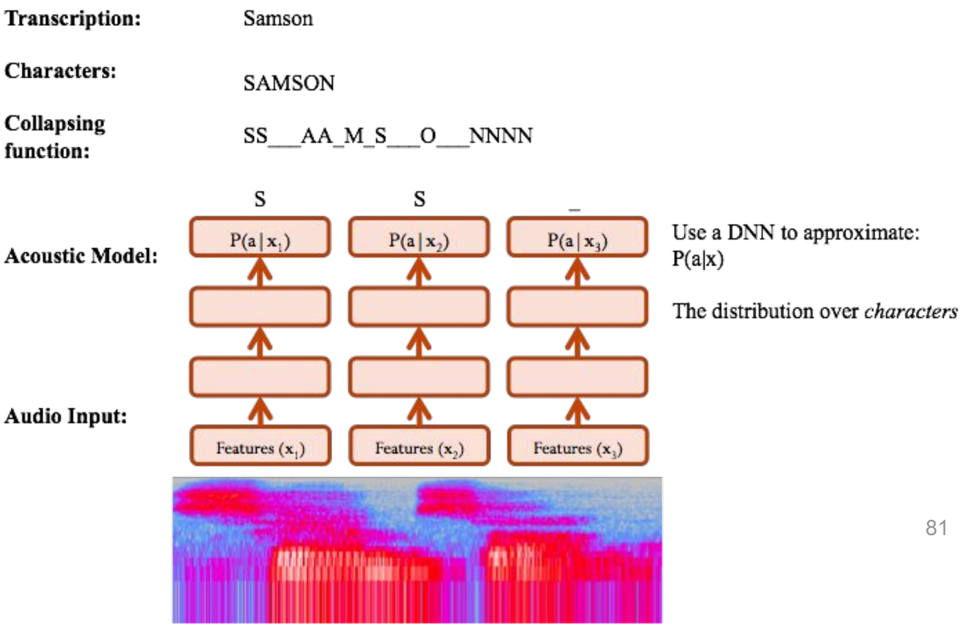
ASR: 2010 ~

- End-to-End System with Neural networks
  - Simplicity, big-data, ...
  - Has taken over HMM systems
  - Albeit similar principles as seen before



ASR: Statistical vs. DNN model





(Graves & Jaitly. 2014)

Tampere University

DNN Per-frame Output (argmax)

____yy__ee_________tt_

____a_____

_rr__e________hh__________b___ii_______lll__i_____tt______aa______tt_______iio__n___ ___cc_____rrr_u_____________________

________ii___ss ______________o__________nn_____________hhh_a___________________nnddd ________________i__n___ __thh_e_____

____bb_uuii_______llllldd____ii____nng_____

__________________________________l___o___o_g__g___ii____nng______

____b___rr_ii________ck__s____p___ll__a________sstt_________eerr__ ______a___nnd_

___b___lll_uu____ee__pp___r___i________nnss_ ________________f______oou____________rrr________

_____________f_____oo__rrr__tt_y____ _____t____www_oo__________ ____nn___ew___________________

____b___e_______t__________i____n___

____e________pp_____aa___rr___tt____mm_ee___nnntss

After collapsing:

yet a rehbilitation cru is onhand in the building loogging bricks plaster and blueprins four forty two new betin epartments

Reference:

yet a rehabilitation crew is on hand in the building lugging bricks plaster and blueprints for forty two new bedroom apartment

(Hannun, Maas, Jurafsky, & Ng. 2014)

Add Language Model

yet a rehbilitation cru is onhand in the building loogging bricks plaster and blueprins four forty two new betin epartments

yet a rehabilitation crew is on hand in the building lugging bricks plaster and blueprints for forty two new bedroom apartments

...

this parcle guna come back on this iland som day soo

the sparkle gonna come back on this island someday soon

...

trade representigd juider warants that the u s wont backcoff its push for trade barior reductions

trade representative yeutter warns that the u s wont back off its push for trade barrier reductions

...

treasury secretary bager at rohie wos in auggral pressed four arise in the value of koreas currency

treasury secretary baker at roh tae woos inaugural pressed for a rise in the value of koreas currency

ASR: statistical vs. DNN

- Similar modules
  - Feature extraction: e.g., MFCC
  - Language modeling: e.g., ngrams
- Differences
  - Acoustic modeling
  - Dictionary/lexicon modeling
  - Decoding may differ: e.g., CTC loss function
- Big-Data
  - Quantity: The more, the better?
  - Quality?
- Generalization vs. specialization

Demo: ASR

```python
print('Build model...' + type_feat + ' based cnnlstm system')
if type_feat == 'mfcc+d+a':
    main_input = Input(shape=(None, 60))
if type_feat == 'mfcc':
    main_input = Input(shape=(None, 20))
y = main_input
if scl:
    y = Lambda(scale_features, arguments={
            "mean": scl.mean_, "scale": scl.scale_}, name="feature_scaler")(y)
if 'cnn' in network_type:
    y = Lambda(lambda k: K.expand_dims(k, axis=-1))(y)
for num in range(num_cnn_layers):
    y = Conv2D(cnn_size, kernel_size, padding="same")(y)
    y = BatchNormalization()(y)
    y = Activation("relu")(y)
    y = Dropout(drop_out_rate)(y)
```

```python
for num in range(num_dense_layers):
    x = Dense(dense_size, activation='relu')(x)
    x = Dropout(drop_out_rate)(x)
output = Dense(num_classes, activation='softmax')(x)
model = Model(inputs=[main_input, auxiliary_input], outputs=output)

model.compile(loss=loss, optimizer=optimizer, metrics=['accuracy'])
model.summary()
```

Useful ASR Toolkits

- Developers
  - Tensorflow (Keras)
  - Torch (PyTorch)
  - HTK
  - KALDI
- End-Users
  - Google
  - Github
  - ...
- Speech/Audio processing
  - Sox: sound manipulation
  - Praat: voice analysis, pitch tracking, spectral analysis, simple speech synthesis
  - SRILM: language modeling toolkits
  - NLTK: python toolkit for language modeling

More resources

- Behind the Mic: The Science of Talking with Computers. A short film about speech processing by Google. https://www.youtube.com/watch?v=yxxRAHVtafI&feature=youtu.be
- A Historical Perspective of Speech Recognition by Huang, Baker and Reddy. Communications of the ACM (2014). This article provides an in-depth and scholarly look at the evolution of speech recognition technology.
- The Voice in the Machine: Building Computers That Understand Speech, Pieraccini, MIT Press (2012). An accessible general-audience book covering the history of, as well as modern advances in, speech processing.
- Fundamentals of Speech Recognition, Rabiner and Juang, Prentice Hall (1993). Rabiner, a researcher at Bell Labs, was instrumental in designing some of the first commercially viable speech recognizers. This book is now over 20 years old, but a lot of the fundamentals remain the same.
- Automatic Speech Recognition: A Deep Learning Approach, Yu and Deng, Springer (2014). Yu and Deng are researchers at Microsoft and both very active in the field of speech processing. This book covers a lot of modern approaches and cutting-edge research but is not for the mathematically faint-of-heart.

Summary

1. Review on probability: the Linda problem, **Bayes Rule**, ice-cream and the weather
2. **Why, how to formulate** the ASR problem
3. Implement ASR system with **HMM**: AM, PM, LM, decoder
4. End-to-end ASR system with **DNN**
5. Resources

After-thoughts…

Acoustic:

- How would you analyze your own speaking voice, reading 10 digits or the alphabets?
- Are there features that are ‘unique’ to yourself, e.g. versus your friends on the same words?

Linguistic:

- Can you find/predict what is the most frequent words in your favorite books, movies, songs?
- Is this useful to recommend authors/genres that could interest you e.g. on Netflix, IMDB?

At home:

- If you have used SIRI or Google Voice, what are the most common errors?
- Use what you learned in this lesson to analyze the ASR performance e.g. WER on one sentence
- Use available tools to implement an ASR system that recognizes 10 digits

References

1. Dan Jurafsky and James H. Martin. *Speech and Language Processing (3rd ed. draft),* 2000
2. Yoav Goldberg. *A Primer on Neural Network Models for Natural Language Processing*
3. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press.
4. Andrew Maas, CS 224S Spoken Language Processing, Stanford University, Spring 2017
5. Graves, Alex, and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." In International Conference on Machine Learning, pp. 1764-1772. 2014.
6. Awni et al. "First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs." arXiv preprint arXiv:1408.2873 (2014).

Items[a]	Probability estimates	
	Mean (%)[b]	Median (%)
Linda will be a teacher in elementary school. (**P**)	29.3 (3.7)	20
Linda will be active in the feminist movement. (**F**)	71.3 (2.9)	80
Linda will be a bank teller. (T)	22.5 (3.4)	10
Linda will take Yoga classes. (Y)	42.5 (4.4)	50
Linda will be a bank teller or will be active in the feminist movement. (T ∨ **F**)	61.5 (4.0)	65
Linda will take Yoga classes or will be a teacher in elementary school. (Y ∨ **P**)	46.3 (3.9)	50

[a] In the version given to the participants, the labels P, F, T, Y, $T \vee F$ and $Y \vee P$ were omitted

[b] Standard errors with 95 % confidence intervals are in parentheses. Data indicates no significant difference on the disjunction statements, respectively relative to the likely target items F and Y ($p < .05$)