# SPEECH/VISION COGNITIVE SYSTEMS

Dr TIAN Jing

tianjing@nus.edu.sg

# Module objective

## Knowledge and understanding

- Understand the fundamentals of speech recognition systems, including statistical acoustic modelling, and end-to-end system using machine learning
- Understand basic concepts of vision cognitive systems

## Key skills

- Design, build, implement and evaluate speech recognition approach in Python
- Design, build, implement and evaluate various visual question and answering approach in Python

# Major reference

- [Introduction] CS131: ***Computer Vision: Foundations and Applications***, http://vision.stanford.edu/teaching/cs131_fall1718/syllabus.html
- [Comprehensive] ***Computer Vision Crash Course***, https://filebox.ece.vt.edu/~jbhuang/
- [Introduction] ***Automatic Speech Recognition***,
  https://github.com/ekapolc/ASR_course
- [Comprehensive] CS224S, ***Spoken Language Processing***,
  http://web.stanford.edu/class/cs224s/

# Topics

- Vision cognition systems
- Speech recognition systems
- Workshop: Design and build speech recognition system in Python

# Vision cognition

The first computer vision project in 1966.

Abstract: The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of “pattern recognition”.

Tasks

- Figure ground: Divide a picture into regions such as likely objects, likely background areas.
- Region description: Analysis of shape and surface properties.
- Object identification: Name objects by matching them with a vocabulary of known objects.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

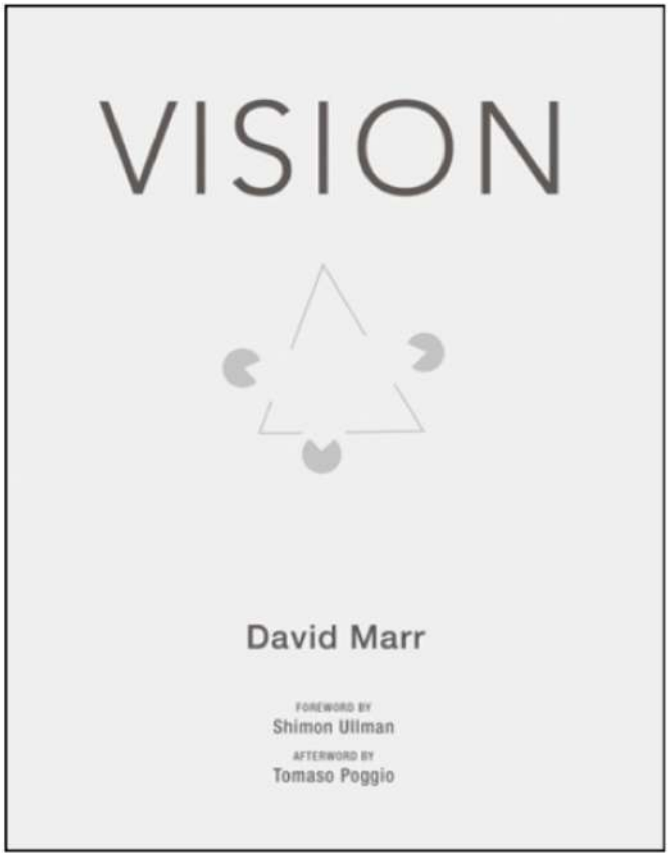
Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Reference: http://people.csail.mit.edu/brooks/idocs/AIM-100.pdf

# Vision cognition: Marr theory

I am not sure that Marr would agree, but I am tempted to add learning as the very top level of understanding, above the computational level. (T. Poggio, 2010)




**Learning**

**Computational**

- Computations relating inputs to outputs

**Algorithmic**

- How the computation is executed at the level of inforamtion processing

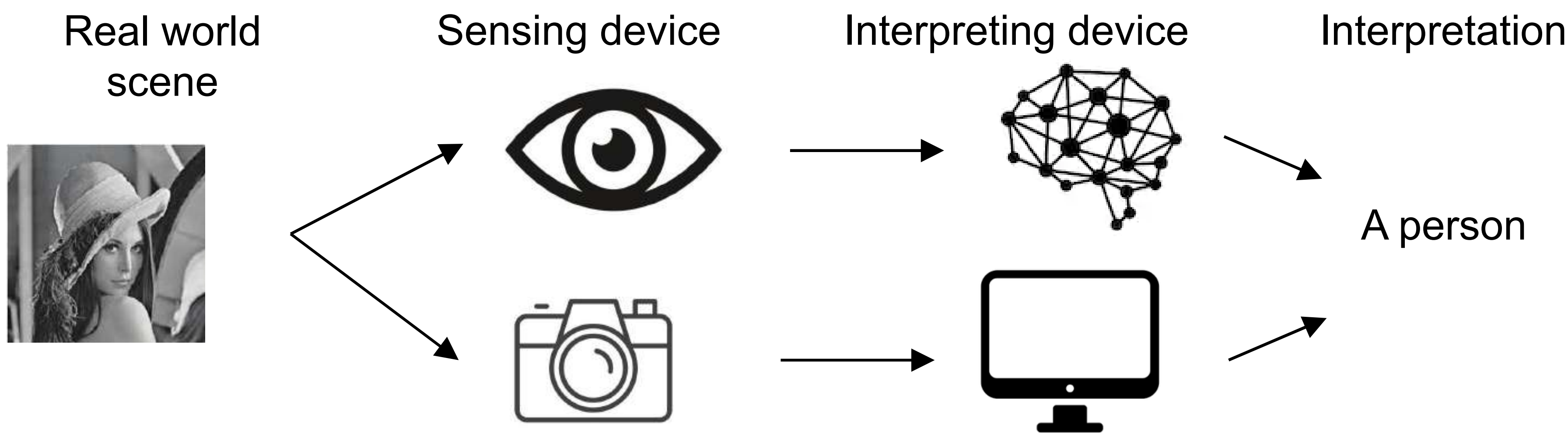
**Implementational**

- How algorithm is embodied as a physical process

Reference
- https://en.wikipedia.org/wiki/David_Marr_(neuroscientist)
- T. Poggio, Vision (2010, The MIT Press), Afterword, P.367

# Thinking humanly

- Humans use their eyes and brains to visually sense the world.
- Computers use their cameras and computation to visually sense the world.




| Computers | Brains |
|---|---|
| Fixed architecture | Evolving architecture |
| Modular, (primarily) serial | Massively parallel |
| Separate hardware, software | No distinction between hardware and software |
| Separate computation, memory | No distinction between computation and memory |

Reference: http://scienceblogs.com/developingintelligence/2007/03/27/why-the-brain-is-not-like-a-co/

# Key cognitivist vision tasks

A concept is named entity, e.g., cat, human

- Learn concepts given labeled examples
- Localize concepts given labeled examples
- Count concepts
- Search for examples similar to this concept
- Explain evidence for concepts
- Estimate variance intrinsic or extrinsic to concept

Photo: https://cathumor.net/human-i-request-your-assistance/

| | |
|---|---|
| Behaviour decision | Cat should take off hanger |
| High-level interpretation | Cat hung on hanger |
| Scene description | Cat on the flat floor |
| Visual objects | Car, hanger |
| Integrated features | Histogram of color/gradients |
| Low-level features | Colors and textures |
| Pre-processing | Enhance contrast of images |

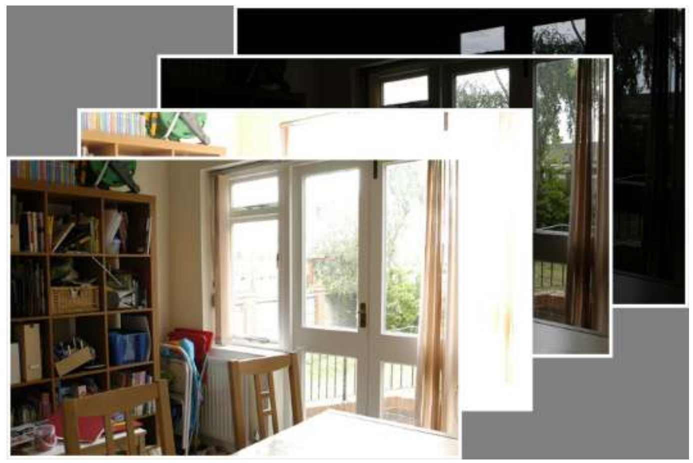
# Vision cognitive system pipeline

Application domain → Image acquisition → Image enhancement → Image restoration → Image processing and analysis → Feature representation & description → Object detection and recognition → Scene understanding → Analytics task

Reference:
- Mo Zi, 400 B. C., https://en.wikipedia.org/wiki/Camera_obscura
- Picture source, http://www.sohu.com/a/140776287_736731
- https://www.indiamart.com/proddetail/street-pole-cctv-camera-14049923862.html

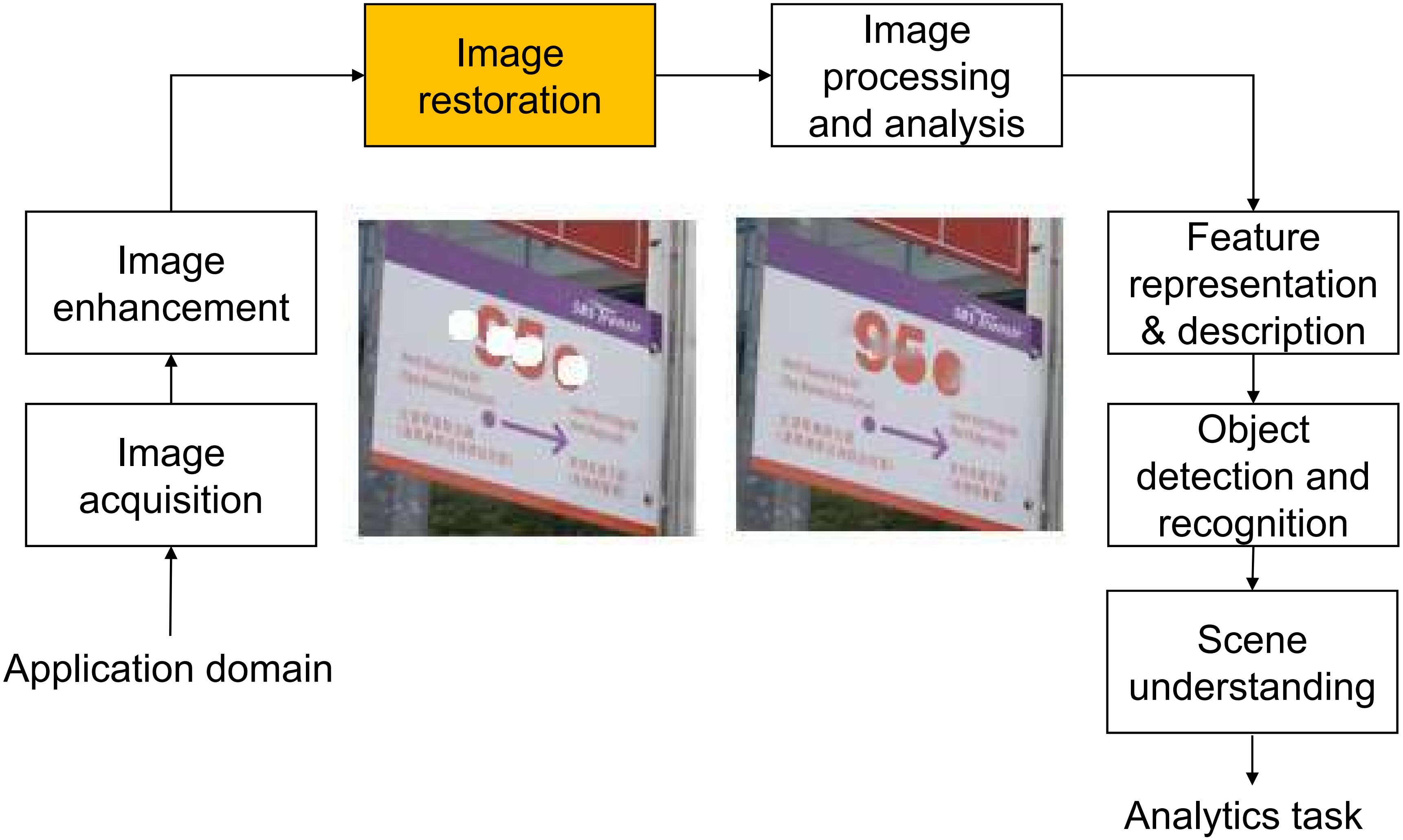
# Vision cognitive system pipeline

Application domain → Image acquisition → Image enhancement → Image restoration → Image processing and analysis → Feature representation & description → Object detection and recognition → Scene understanding → Analytics task



Online demo: http://ipolcore.ipol.im/demo/clientApp/demo.html?id=230

# Vision cognitive system pipeline




Online demo: http://demo.ipol.im/demo/54/

# Vision cognitive system pipeline

Application domain

Image acquisition

Image enhancement

Image restoration

Image processing and analysis


Feature representation & description

Object detection and recognition

Scene understanding

Analytics task

Online demo: http://bigwww.epfl.ch/demo/ip/demos/edgeDetector/

# Vision cognitive system pipeline

Image restoration

Image processing and analysis

Image enhancement

Feature representation & description

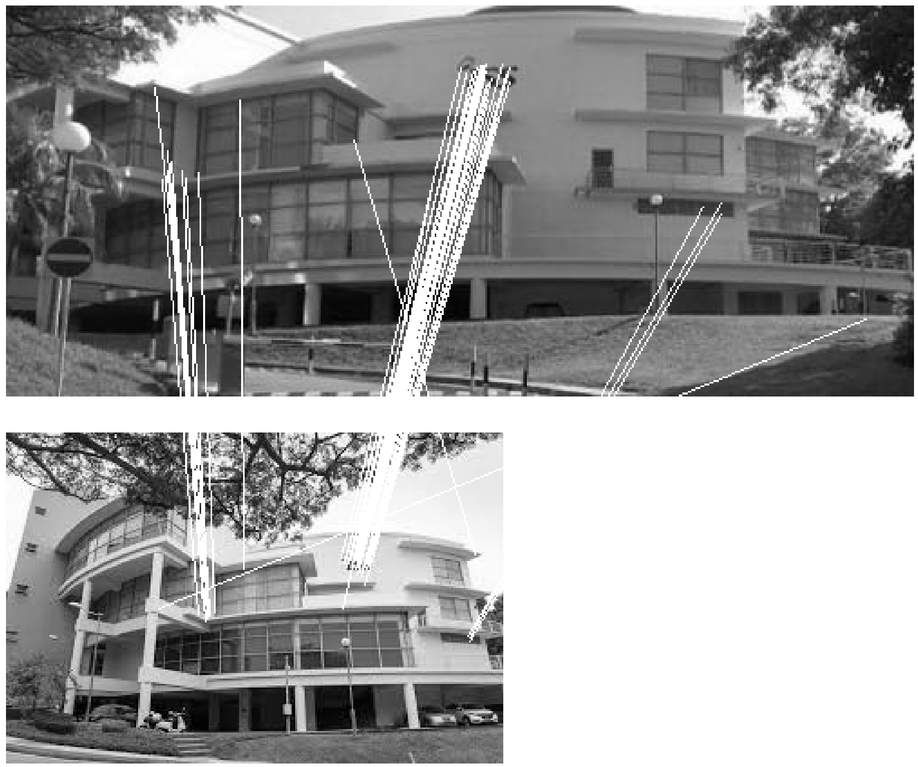
Image acquisition

Object detection and recognition

Application domain

Scene understanding

Analytics task



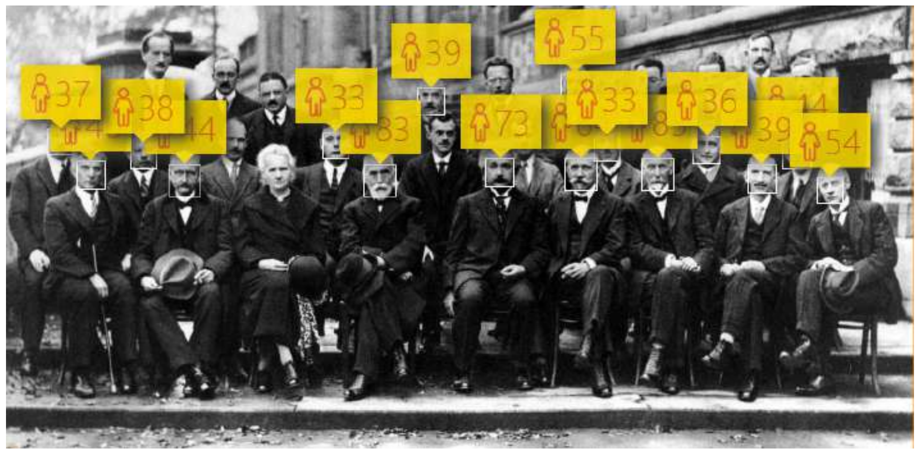
Online demo: http://demo.ipol.im/demo/my_affine_sift/

# Vision cognitive system pipeline

Image acquisition → Image enhancement → Image restoration → Image processing and analysis → Feature representation & description → **Object detection and recognition** → Scene understanding

Application domain → Image acquisition

Scene understanding → Analytics task




Demo website: https://www.how-old.net

# Vision cognitive system pipeline

Image acquisition → Image enhancement → Image restoration → Image processing and analysis → Feature representation & description → Object detection and recognition → Scene understanding

Application domain

Analytics task

One Look Is Worth A Thousand Words--

One look at our line of Republic, Firestone, Miller and United States tires can tell you more than a hundred personal letters or advertisements.

WE WILL PROVE THEIR VALUE BEFORE YOU INVEST ONE DOLLAR IN THEM.

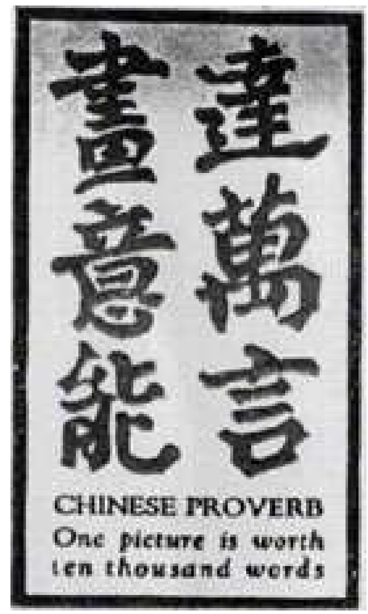
Ever consider buying Supplies from a catalog?

What's the use! Call and see what you are buying. One look at our display of automobile and motorcycle accessories will convince you of the fact.

THAT WE HAVE EVERYTHING FOR THE AUTO

Piqua Auto Supply House

133 N. Main St.—Piqua, O.




Machine learning can be applied in every component.

Reference

- https://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words
- https://www.phrases.org.uk/meanings/a-picture-is-worth-a-thousand-words.html

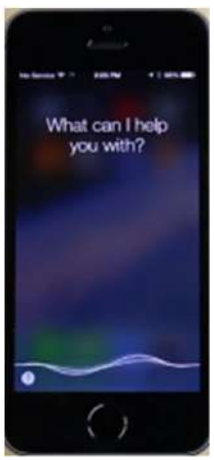
# Automatic speech recognition

Amazon Echo
2015

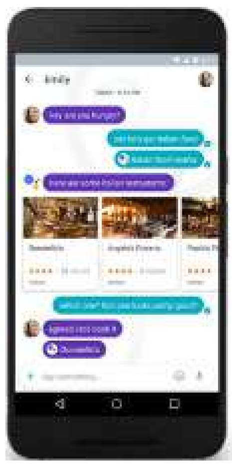
Google Home
2016

Facebook M
2015

Anki Cozmo
2016

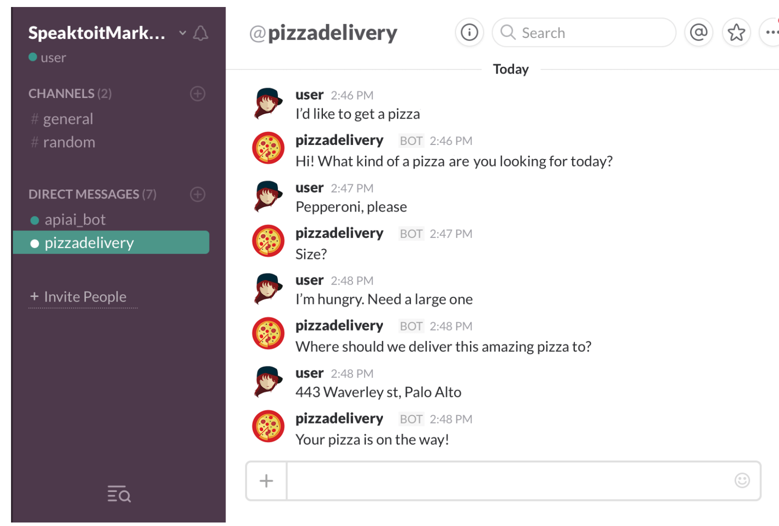



Apple
Siri
2011



Google
Assistant
2016

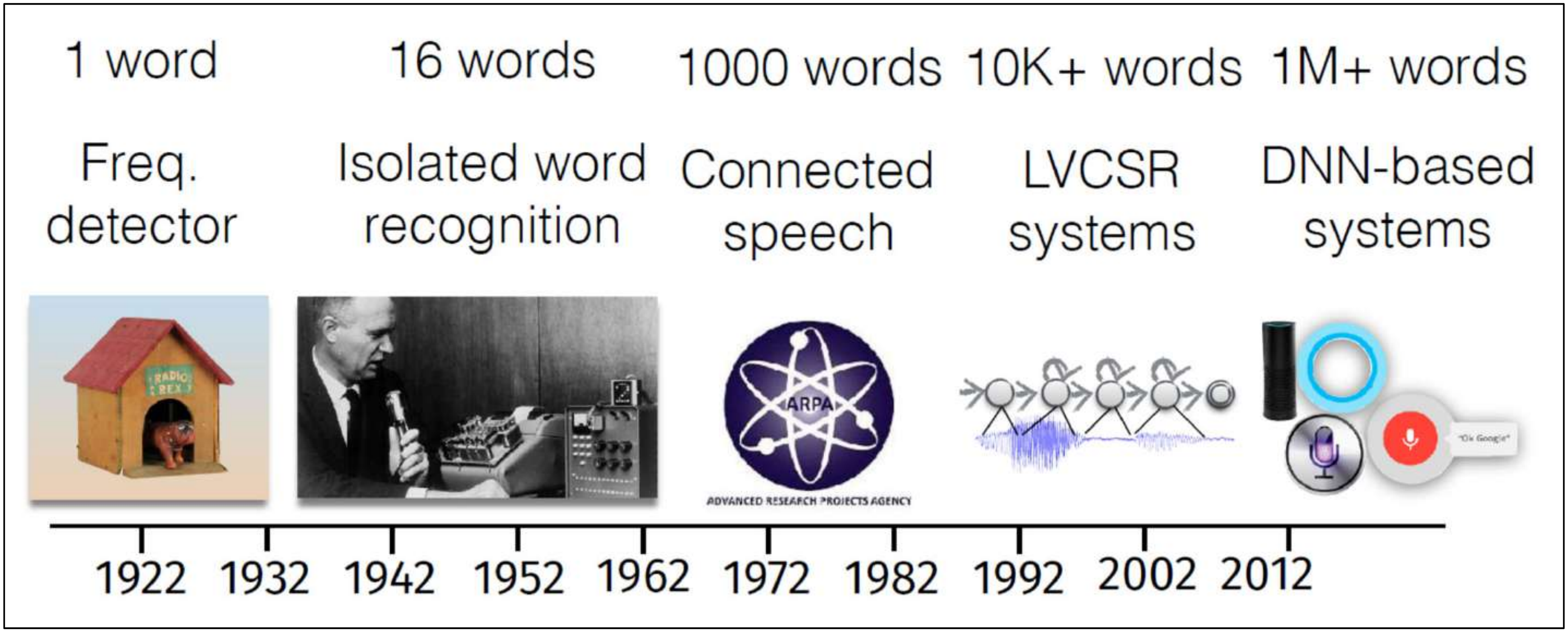

Microsoft
Cortana
2014




Slack Bot API
2015

Source: CS224S Spoken Language Processing, http://web.stanford.edu/class/cs224s/

# Automatic speech recognition



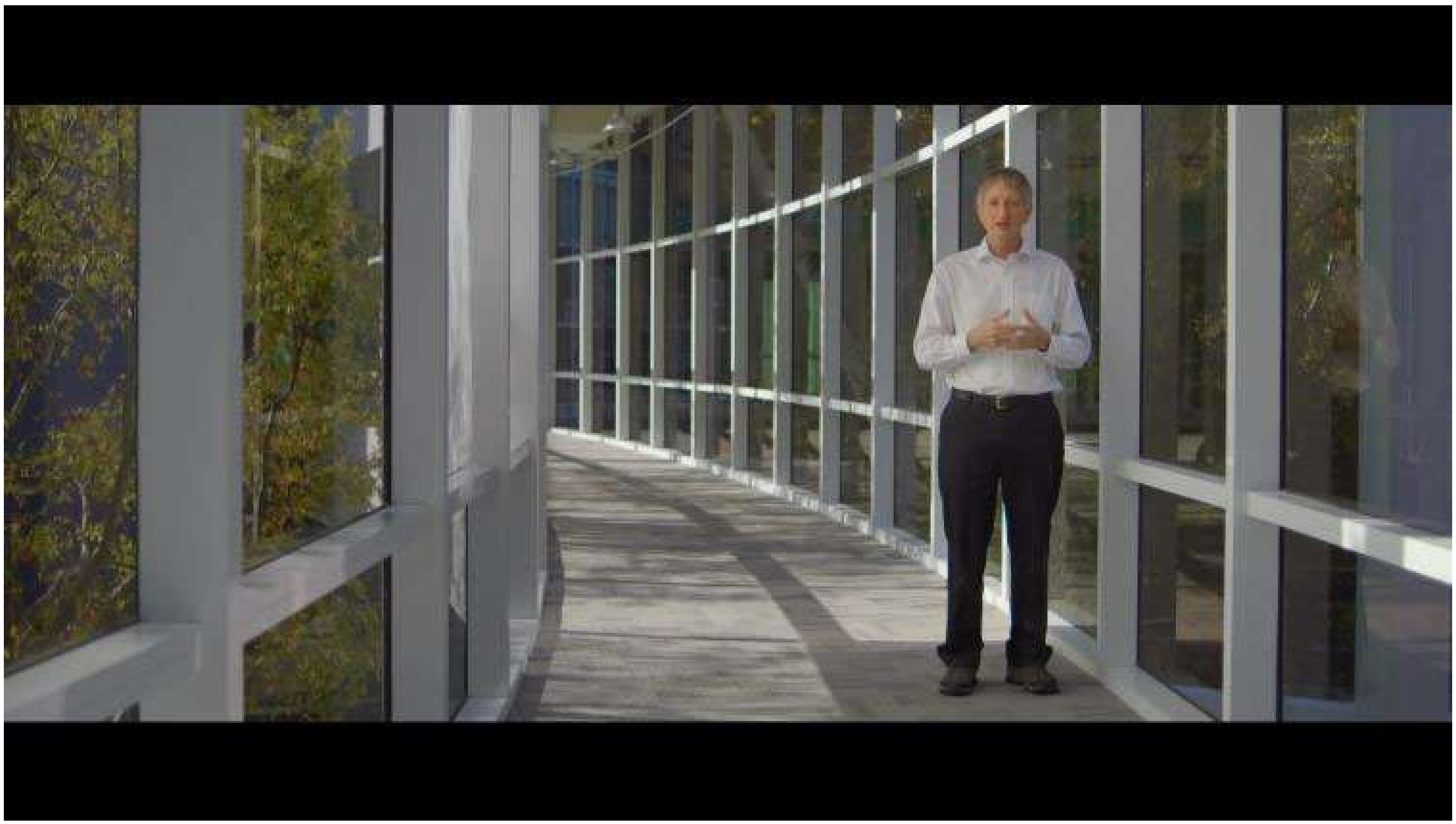

More introductions on history of automatic speech recognition can be found at

- https://ileriseviye.wordpress.com/2011/02/17/speech-recognition-in-1920s-radio-rex-the-first-speech-recognition-machine/
- https://machinelearning-blog.com/2018/09/07/a-brief-history-of-asr-automatic-speech-recognition/

Source: Automatic Speech Recognition (CS753), Lecture 1: Introduction to Statistical Speech Recognition, https://www.cse.iitb.ac.in/~pjyothi/cs753/

# Automatic speech recognition

Behind the Mic: The Science of Talking with Computers, (6 minutes)
https://www.youtube.com/watch?v=yxxRAHVtafI
Language is easy for humans to understand (most of the time), but not so easy for computers. This video talks about speech recognition, language understanding, neural nets, and using our voices to communicate with the technology around us.



# Automatic speech recognition

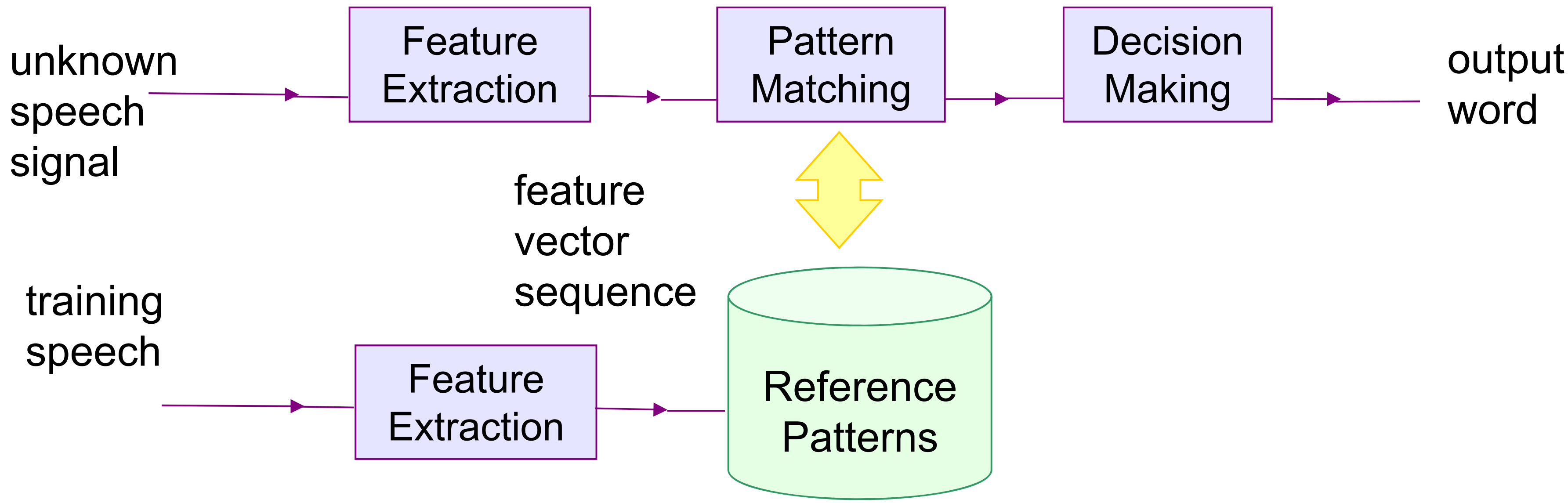
- Human-machine Interaction
  - Automatic Speech Recognition
  - Speech Synthesis / Text-to-Speech (TTS)
  - Natural Language Understanding (NLU)
  - Natural Language Generation (NLG)
- Telecommunication
  - Speech Coding
- Language
  - Statistical Machine Translation (SMT)
- Language Acquisition
  - Pronunciation Training
- Security/Forensics
  - Speaker ID
  - Speaker Verification
- Medical Applications
  - Diagnosis of Diseases
- Information Retrieval
  - Video/Audio Transcribing
  - Audio/Text Summarizing
- Speech Manipulation
  - Speaking Rate Adjusting

# Automatic speech recognition

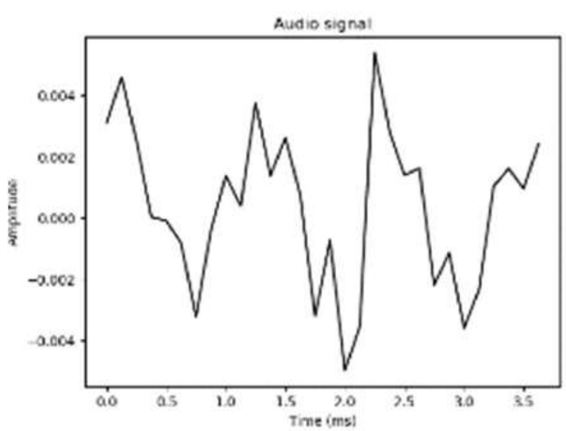
## Challenges of speech recognition

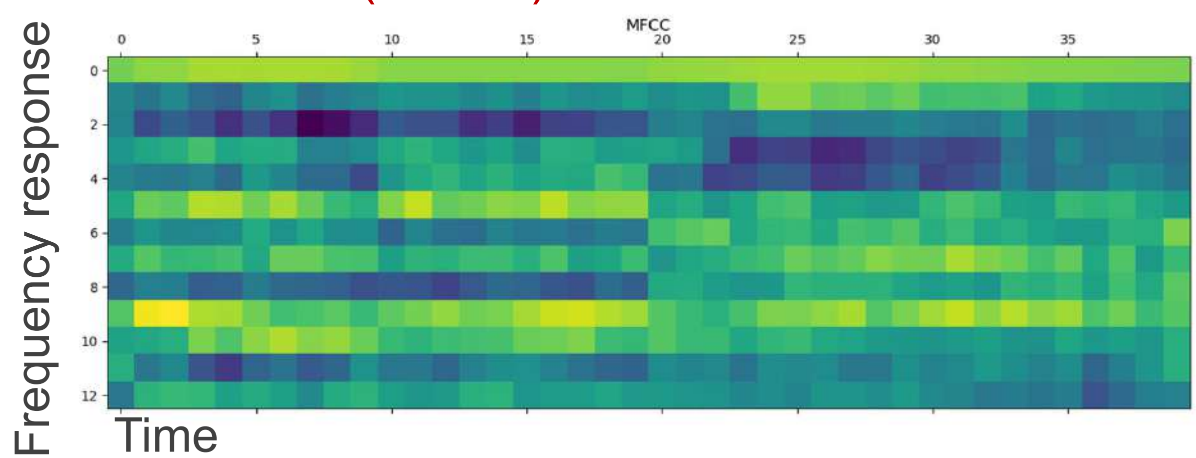
- Style: Read speech or spontaneous (conversational) speech?
- Continuous natural speech or command & control?
- Speaker characteristics: Rate of speech, accent, prosody (stress, intonation), speaker age, pronunciation variability even when the same speaker speaks the same word
- Channel characteristics: Background noise, room acoustics, microphone properties, interfering speakers
- Task specifics: Vocabulary size (very large number of words to be recognized), language-specific complexity, resource limitations

# Speech recognition as pattern recognition problem




Speech signal represented in time domain and frequency-domain features, e.g., *Mel frequency cepstral coefficient* (MFCC)

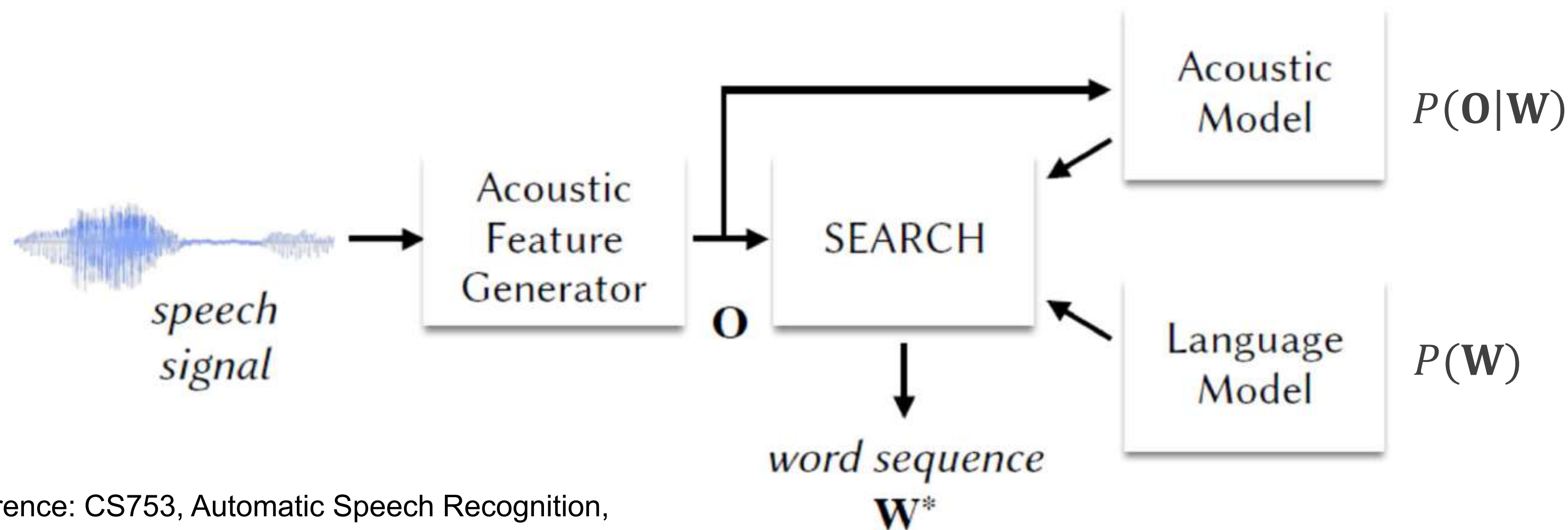


# Statistical speech recognition

- Let $\mathbf{O}$ represent a sequence of acoustic feature observations (i.e., $\mathbf{O} = \{o_1, o_2, \cdots, o_t\}$, and $\mathbf{W}$ denote a word sequence. Then the speech recognizer decodes $\mathbf{W}^*$ as

$$\mathbf{w}^* = \underset{\mathbf{W}}{\operatorname{argmax}} P(\mathbf{W}|\mathbf{O}) = \underset{\mathbf{W}}{\operatorname{argmax}} \frac{P(\mathbf{O}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{O})} \propto \underset{\mathbf{W}}{\operatorname{argmax}} P(\mathbf{O}|\mathbf{W})\mathrm{P}(\mathbf{W})$$




Reference: CS753, Automatic Speech Recognition, https://www.cse.iitb.ac.in/~pjyothi/cs753/

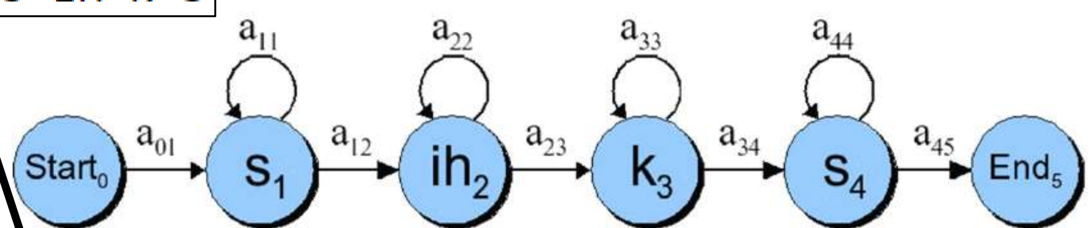
# Statistical speech recognition

- Further introduce three definitions, acoustic signal $\mathbf{A}$, phoneme $\mathbf{L}$, and state $\mathbf{Q}$. The optimization problem statement is changed to be

$$\mathbf{w}^* = \underset{\mathbf{W}}{\operatorname{argmax}}\, P(\mathbf{O}|\mathbf{W})\mathrm{P}(\mathbf{W}) = \underset{\mathbf{W}}{\operatorname{argmax}}\, P(\mathbf{A}|\mathbf{O})P(\mathbf{O}|\mathbf{Q})P(\mathbf{Q}|\mathbf{L})P(\mathbf{L}|\mathbf{W})P(\mathbf{W})$$

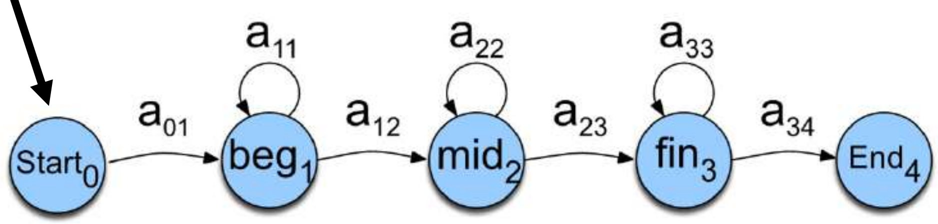
- SIX
- S IH K S

Key challenge is how to estimate sequence.
→ Hidden Markov model (HMM).

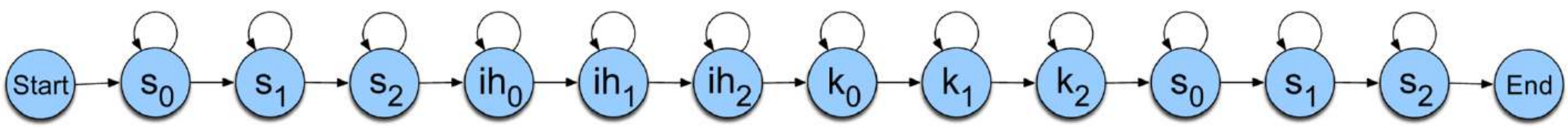



Phoneme model

According to linguistic study, each phoneme has 3 states: (1) the transition part at the begin of the phoneme, (2) the stationary part, (3) the transition at the end.

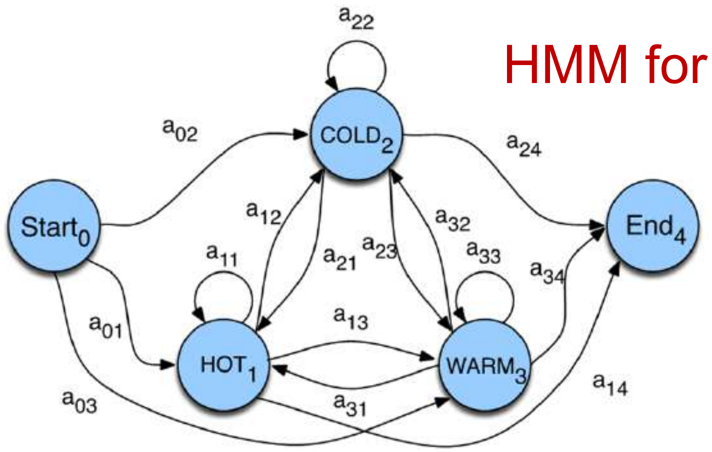


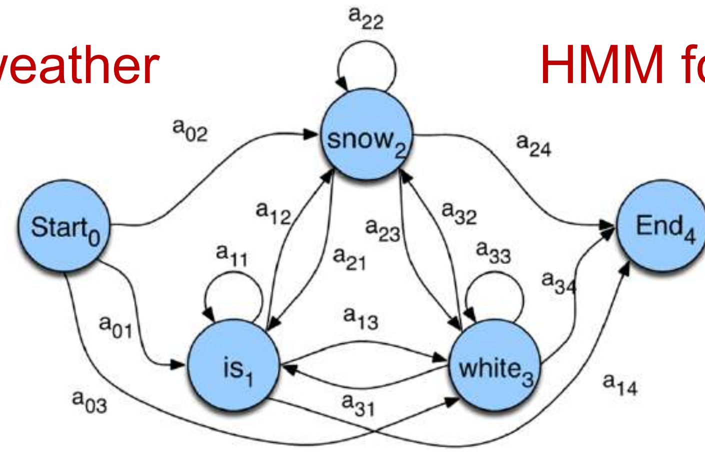

Details will be studied in acoustic model in following slides.




Reference: http://www.speech.cs.cmu.edu/cgi-bin/cmudict?in=six

# Hidden Markov model (HMM): Idea




| Notation | Descriptions | |
|---|---|---|
| $\mathbf{Q} = \{q_1, q_2, \cdots, q_t\}$ | A set of $N$ states for observations | Each observation has one state |
| $\mathbf{A} = \{a_{11}, a_{12}, \cdots, a_{nn}\}$ | A state transition probability matrix $\mathbf{A}$, each $a_{ij}$ representing the probability of moving from the state $i$ to the state $j$, $s.t. \sum_{j=1}^{n} a_{ij} = 1$ | Learned from speech training dataset |
| $\mathbf{O} = \{o_1, o_2, \cdots, o_t\}$ | A sequence of $T$ observations | Observed speech data |
| $B = b_i(o_t)$ | An observation likelihoods, also called emission probabilities, each expressing the probability of an observation $o_t$ being generated from a state $i$ | Learned from speech training dataset |
| $\mathbf{S}$ | A set of states (e.g., HOT$_1$, COLD$_2$, WARM$_3$, is$_1$, snow$_2$, white$_3$), a special start state Start$_0$ and an end state End$_4$ that are not associated with observations, together with their transition probabilities out of the start state and into the end state. For example, each phoneme has 3 states (see slide 23). | |

Reference: CS224S, Spoken Language Processing, http://web.stanford.edu/class/cs224s/

# HMM: A toy example

| Transition probability | | | | Observation likelihood | | |
|---|---|---|---|---|---|---|
| Today weather | Tomorrow weather | | | Weather | Probability of | |
| | Sunny ($S$) | Raining ($R$) | Cloudy ($C$) | | Umbrella ($U$) | No umbrella ($N$) |
| Sunny ($S$) | 0.8 | 0.05 | 0.15 | Sunny ($S$) | 0.1 | 0.9 |
| Raining ($R$) | 0.2 | 0.6 | 0.2 | Raining ($R$) | 0.8 | 0.2 |
| Cloudy ($C$) | 0.2 | 0.3 | 0.5 | Cloudy ($C$) | 0.3 | 0.7 |

Q: Given that today weather is $S$, what is the probability that tomorrow is $S$ and the day after is $R$?

Markov assumption

$$P(q_2 = S, q_3 = R | q_1 = S) = P(q_3 = R | q_2 = S, q_1 = S) P(q_2 = S | q_1 = S)$$
$$= P(q_3 = R | q_2 = S) P(q_2 = S | q_1 = S) = 0.05 \times 0.8 = 0.04$$

Q: Given that you don't use umbrella ($N$) for three days, calculate the probability for the weather on these three days to be $\{q_1 = S, q_2 = C, q_3 = S\}$. Note that the prior probability for the start state as sunny ($S$) on day one is assumed to be 1/3 (three weather has same probability).

$$P(q_1 = S, q_2 = C, q_3 = S | o_1 = N, o_2 = N, o_3 = N)$$
$$= P(o_1 = N | q_1 = S) P(o_2 = N | q_2 = C) P(o_3 = N | q_3 = S) P(q_1 = S) P(q_2 = C | q_1 = S) P(q_3 = S | q_2 = C)$$
$$= 0.9 \times 0.7 \times 0.9 \times 1/3 \times 0.15 \times 0.2 = 0.0057$$

Reference: http://www.iitg.ac.in/samudravijaya/tutorials/hmmTutorialBarbaraExercises.pdf

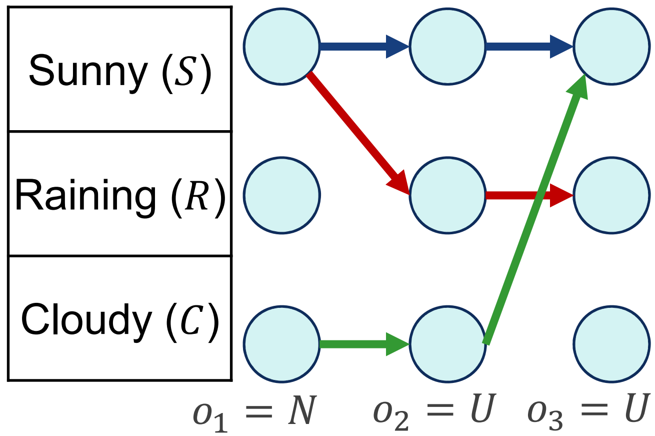
# HMM: Sequence estimation

Q: Given that three days your umbrella observations are: {no umbrella ($N$), umbrella ($U$), umbrella ($U$)}, find the most probable weather-sequence.

Idea 1: If we ignore the weather as a ‘sequence’ and treat each day weather separately, the most probable weather are Sunny ($S$), Raining ($R$), Raining ($R$).

Idea 2: Exhaustively evaluate probability of each sequence. Consider following three possible sequences, which is most probable?

- Blue sequence: Sunny ($S$), Sunny ($S$), Sunny ($S$)
- Red sequence: Sunny ($S$), Raining ($R$), Raining ($R$)
- Green sequence: Cloudy ($C$), Cloudy ($C$), Sunny ($S$)




Idea 3: Design an efficient method to evaluate all possible sequence and find the most probable one.

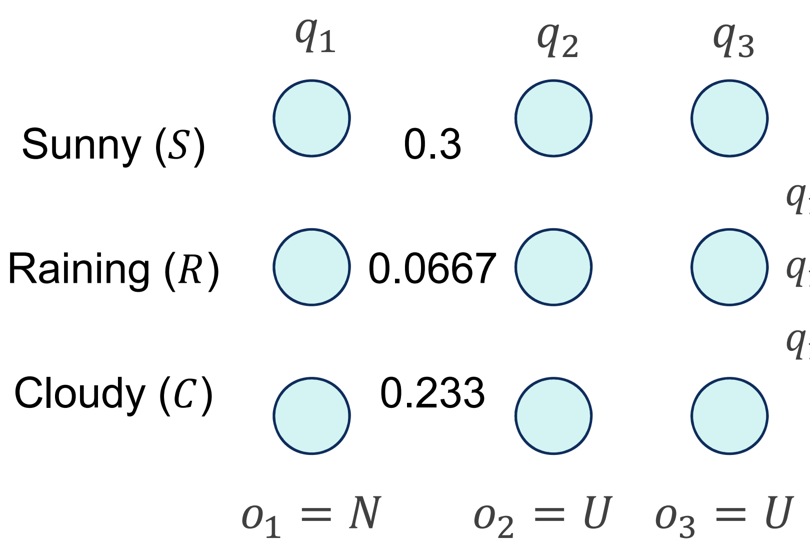
→ We will study Viterbi algorithm in next few slides.

Viterbi: A single-line $.predict(O)$ function in hmmlearn library

# HMM: Viterbi algorithm

Key idea: “Optimal policy is composed of optimal sub-policies”.

1. Initialization: Calculate probability of the first day state based on first day observation and (equal) prior probability starting from all possible states.
2. Recursion: For all following days, calculate probability of each state based on current observation and the largest transition probability from the previous day. Record the ‘best path’ ending at current state from the previous day.
3. Termination and back tracing: For the last day, choose the state with the highest probability. Trace back according to the recorded most probable path.




$q_1$ is $S$: $P(q_1 = S) \times P(o_1 = N | q_1 = S) = \frac{1}{3} \times 0.9 = 0.3$

$q_1$ is $R$: $P(q_1 = R) \times P(o_1 = N | q_1 = R) = \frac{1}{3} \times 0.2 = 0.0667$

$q_1$ is $C$: $P(q_1 = C) \times P(o_1 = N | q_1 = C) = \frac{1}{3} \times 0.7 = 0.233$

# HMM: Viterbi algorithm

Key idea: "Optimal policy is composed of optimal sub-policies".

1. Initialization: Calculate probability of the first day state based on first day observation and (equal) prior probability starting from all possible states.
2. Recursion: For all following days, calculate probability of each state based on current observation and the largest transition probability from the previous day. Record the 'best path' ending at current state from the previous day.
3. Termination and back tracing: For the last day, choose the state with the highest probability. Trace back according to the recorded most probable path.

$q_2$ is $S$: $\max(P(q_1 = S)\alpha_{ss}, P(q_1 = R)\alpha_{rs}, P(q_1 = C)\alpha_{cs})\, P(o_2 = U|q_2 = S)$

$= \max(0.3 \times 0.8, 0.0667 \times 0.2, 0.233 \times 0.2) \times 0.1 = 0.24 \times 0.1 = 0.024$

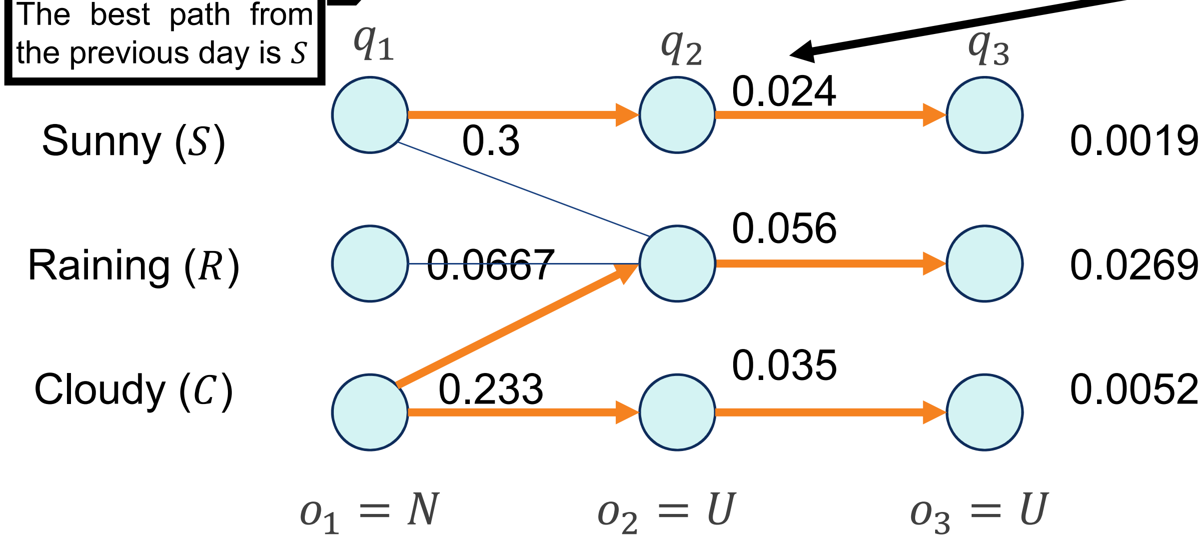
The best path from the previous day is $S$

The best path from the previous day is $R$

$q_3$ is $R$:

$\max(P(q_2 = S)\alpha_{sr}, P(q_2 = R)\alpha_{rr}, P(q_2 = C)\alpha_{cr})\, P(o_3 = U|q_3 = R)$

$= \max(0.024 \times 0.05, 0.056 \times 0.6, 0.035 \times 0.3) \times 0.8 = 0.0336 \times 0.8 = 0.0269$



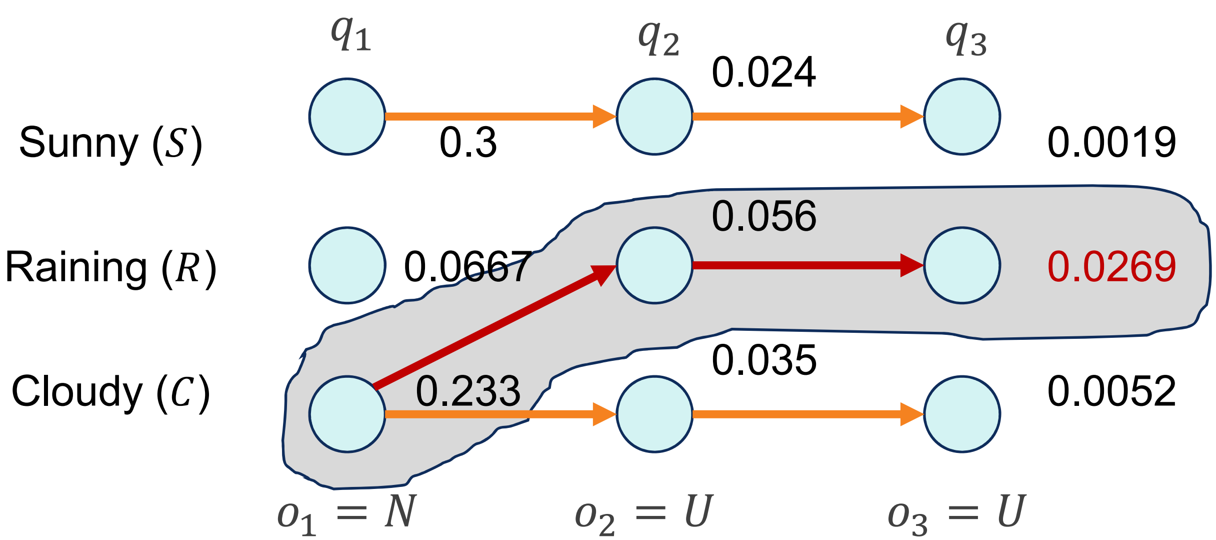

Note: $\alpha$ is transition probability, $\alpha_{ss} = 0.8$ from $R$ to $R$ (see slide 26)

# HMM: Viterbi algorithm

Key idea: "Optimal policy is composed of optimal sub-policies".

1. Initialization: Calculate probability of the first day state based on first day observation and (equal) prior probability starting from all possible states.
2. Recursion: For all following days, calculate probability of each state based on current observation and the largest transition probability from the previous day. Record the 'best path' ending at current state from the previous day.
3. Termination and back tracing: For the last day, choose the state with the highest probability. Trace back according to the recorded most probable path.

The optimal sequence: Cloudy ($C$), Raining ($R$), Raining ($R$).
Recall that the result (in Idea 2 in slide 26) is Sunny ($S$), Raining ($R$), Raining ($R$).




How to use it for speech recognition?

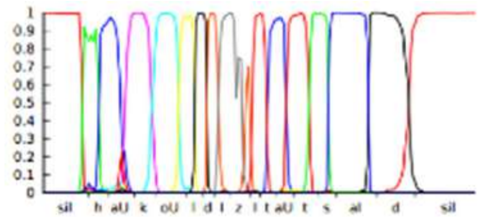
- Weather → phoneme state
- Umbrella → audio features

# Summary: Statistical speech recognition

$$\mathbf{w}^* = \underset{\mathbf{W}}{\operatorname{argmax}} P(\mathbf{O}|\mathbf{W})\mathrm{P}(\mathbf{W}) = \underset{\mathbf{W}}{\operatorname{argmax}} P(\mathbf{A}|\mathbf{O})P(\mathbf{O}|\mathbf{Q})P(\mathbf{Q}|\mathbf{L})P(\mathbf{L}|\mathbf{W})P(\mathbf{W})$$

$\mathbf{A}$

Speech Audio




$\mathbf{Q}$

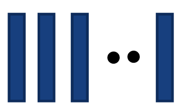
Sequence States

$\mathbf{W}$

Words

$P(\mathbf{A}|\mathbf{O})$ → Feature Extraction → $P(\mathbf{O}|\mathbf{Q})$ Frame Classification → $P(\mathbf{Q}|\mathbf{L})$ Sequence Model → $P(\mathbf{L}|\mathbf{W})$ Lexicon Model → $P(\mathbf{W})$ Language Model →

Sentence



$\mathbf{O}$

Feature Frames

t ah m aa t ow

$\mathbf{L}$

Phonemes

# Statistical speech recognition: Lexicon model

- Lexical modelling forms the bridge between the acoustic and language models
- Each one with a pronunciation in terms of phones
- CMU dictionary: 127K words, http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Deterministic model

| Word | Pronunciation |
|---|---|
| TOMATO | t ah m aa t ow |
| | t ah m ey t ow |
| COVERAGE | k ah v er ah jh |
| | k ah v r ah jh |

Probabilistic model

| Word | Pronunciation | Probability |
|---|---|---|
| TOMATO | t ah m aa t ow | 0.45 |
| | t ah m ey t ow | 0.55 |
| COVERAGE | k ah v er ah jh | 0.65 |
| | k ah v r ah jh | 0.35 |

# Statistical speech recognition: Language model

$N$-gram models: Build the language model by calculating probabilities from text training corpus: how likely is one word to follow another.

| | i | want | to | eat | chinese | food | lunch | spend |
|---|---|---|---|---|---|---|---|---|
| **i** | 5 | 827 | 0 | 9 | 0 | 0 | 0 | 2 |
| **want** | 2 | 0 | 608 | 1 | 6 | 6 | 5 | 1 |
| **to** | 2 | 0 | 4 | 686 | 2 | 0 | 6 | 211 |
| **eat** | 0 | 0 | 2 | 0 | 16 | 2 | 42 | 0 |
| **chinese** | 1 | 0 | 0 | 0 | 0 | 82 | 1 | 0 |
| **food** | 15 | 0 | 15 | 0 | 1 | 4 | 0 | 0 |
| **lunch** | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **spend** | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

Example: Bi-gram counts for eight of the words (out of $V$ = 1446) in the Berkeley Restaurant Project corpus of 9332 sentences

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

# Statistical speech recognition: Data

- Collect corpora appropriate for recognition task
  - Small speech + phonetic transcription to associate sounds with symbols (Acoustic Model)
  - Large (>= 60 hrs) speech + orthographic transcription to associate words with sounds (Acoustic Model)
  - Very large text corpus to identify $N$-gram probabilities or build a grammar (Language Model)

# Speech recognition: Evaluation

- Word Error Rate (WER): Minimum Edit Distance: Distance in words between the system output and the reference transcription (truth)

$$WER = \frac{S + D + I}{N}$$

- $S$ is the number of substitutions,
- $D$ is the number of deletions,
- $I$ is the number of insertions and
- $N$ is the number of words in the reference

| | |
|---|---|
| Truth: What a bright day | System: What a day |
| Deletion: "Bright" was deleted by the system | |
| Truth: What a day | System: What a bright day |
| Insertion: "Bright" was inserted by the system | |
| Truth: What a bright day | System: What a light day |
| Substitution: "Bright" was substituted by "light" by the system | |

Reference: https://martin-thoma.com/word-error-rate-calculation/

# Language, vision and actions

Language
- Instruction following
- Question answering
- Dialog

Vision
- Image / video understanding
- 3D environment perception

Actions
- Camera motion
- Robotics / Manipulation


Reference:
- Connecting language and vision to actions, https://lvatutorial.github.io/
- Reference: Natural language interaction with robots, https://bringmeaspoon.org/

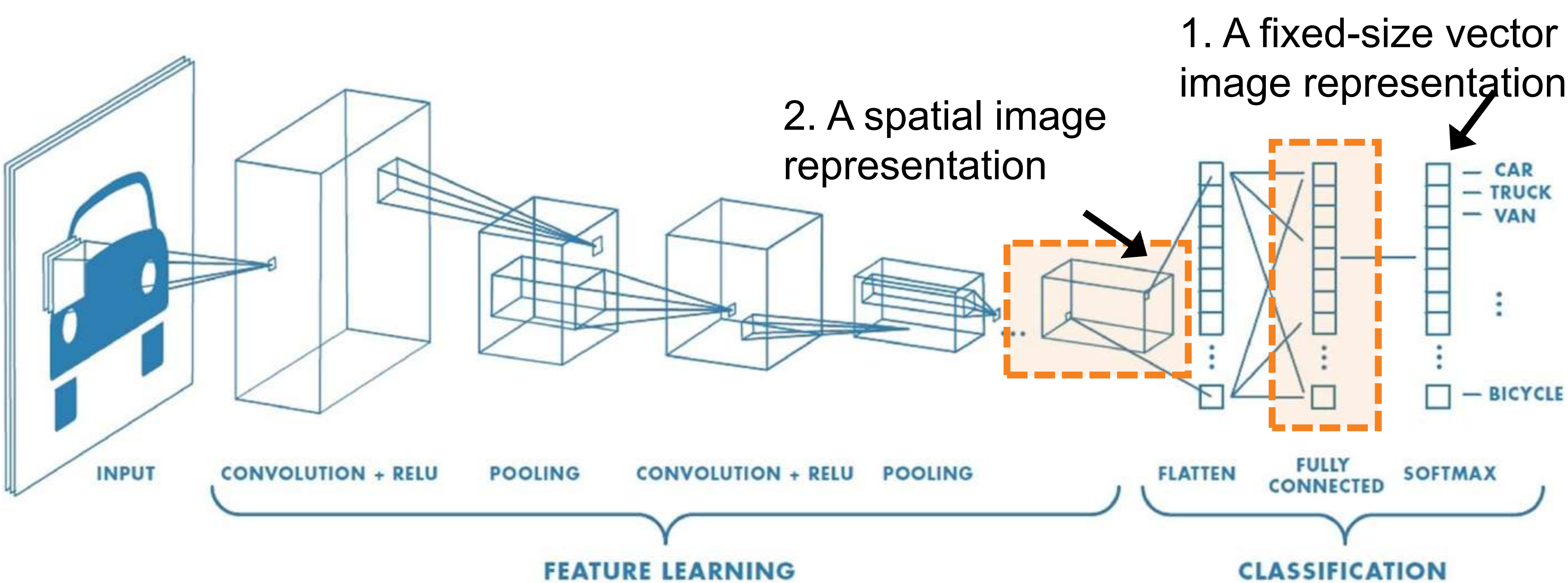
# Visual question answering (VQA)

- Task: Given an image and a natural language open-ended question, generate a natural language answer. This is reasoning techniques using both language and vision knowledge.


Reference: P. Wang, Q. Wu, C. Shen, A. Hengel, A. Dick, FVQA: Fact-Based Visual Question Answering, https://arxiv.org/abs/1606.05433

# Visual question answering (VQA)

Recap: We can treat convolutional neural networks (CNNs) as black boxes that can output following two things



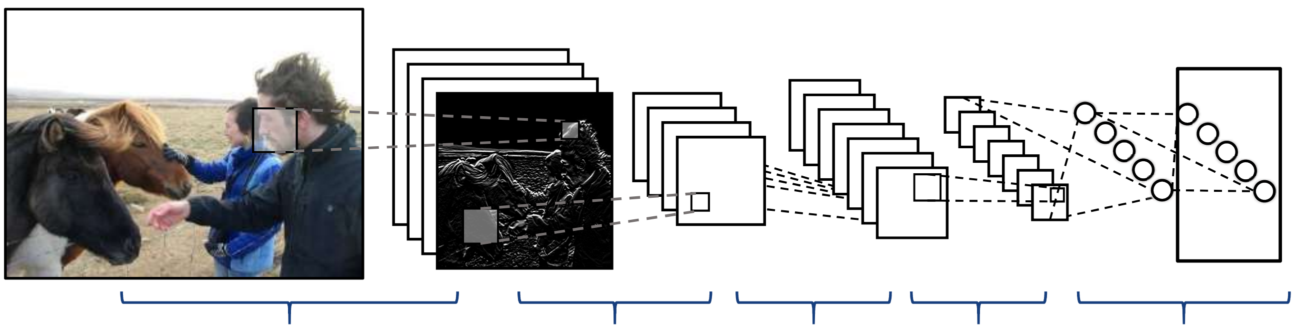

Reference: Connecting language and vision to actions, https://lvatutorial.github.io/

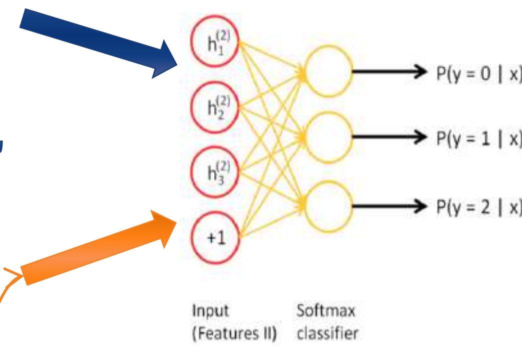
# Visual question answering (VQA)

Demo website: http://vqa.cloudcv.org/

## Image

Neural Network over top answers

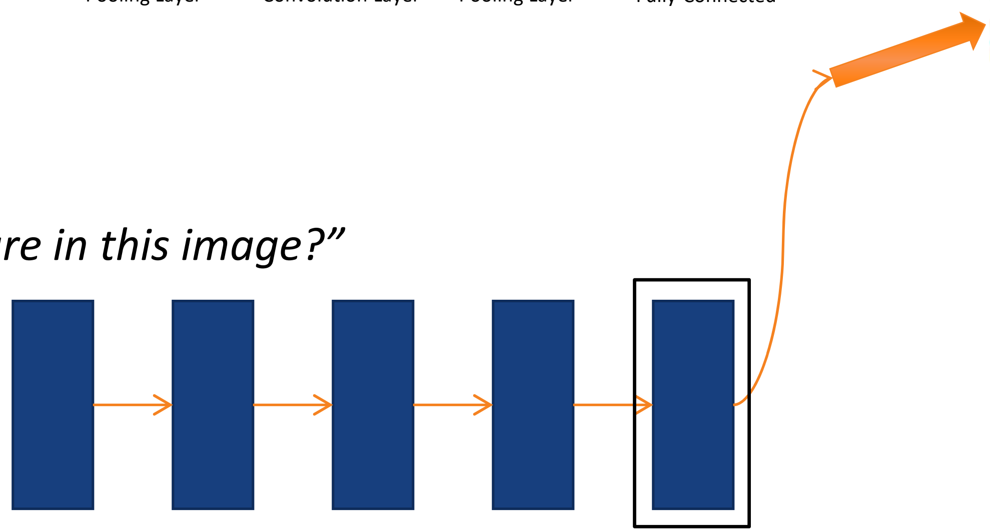



## Question

*"How many horses are in this image?"*

LSTM



Reference: Aishwarya Agrawal, et al, VQA: Visual Question Answering, https://arxiv.org/abs/1505.00468

# Visual question answering (VQA)

- Output: (classification) 1000 answers.
- Input: Image feature (VGG16 model, 4096 dimensional vector, 'fc2'), text feature (tokens in the question are first embedded into 300 dimensional GloVe vectors and then passed through LSTM). Both multimodal data points are then passed through a dense layer and a final softmax layer.

```
Layer (type)                     Output Shape          Param #
=================================================================
reshape_2_input (InputLayer)     (None, 4096)          0
_________________________________________________________________
input_2 (InputLayer)             (None, 30, 300)       0
_________________________________________________________________
reshape_2 (Reshape)              (None, 4096)          0
_________________________________________________________________
lstm_2 (LSTM)                    (None, 512)           1665024
_________________________________________________________________
dense_4 (Dense)                  (None, 1024)          4195328
_________________________________________________________________
concatenate_2 (Concatenate)      (None, 1536)          0

_________________________________________________________________
dense_5 (Dense)                  (None, 1024)          1573888
_________________________________________________________________
dense_6 (Dense)                  (None, 1000)          1025000
=================================================================
Total params: 8,459,240
Trainable params: 8,459,240
Non-trainable params: 0
```

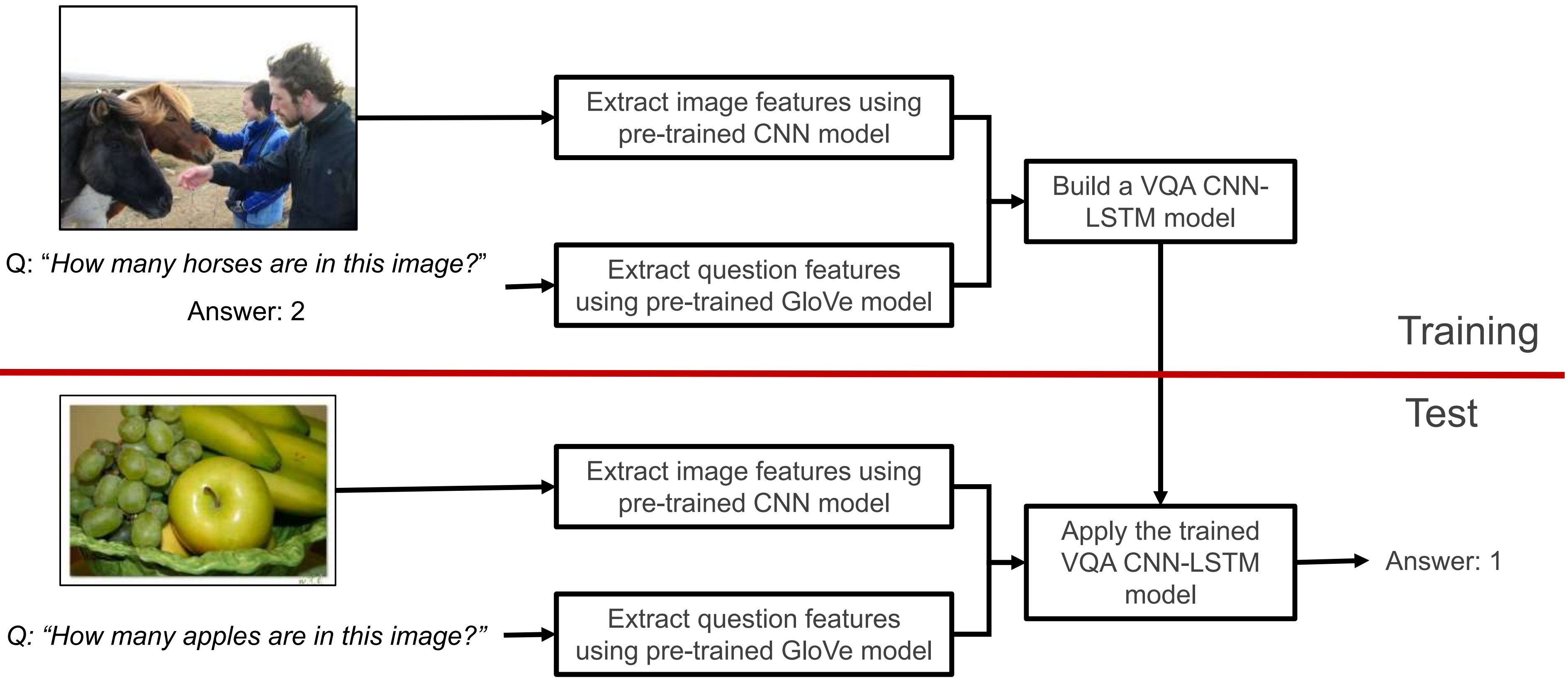
Toy model architecture


Example training images and question/answer

# Visual question answering (VQA)

The full VQA pipeline

- Pre-process both image and question/answer text
- Design a model architecture and train the model
- Deploy the model and process the new test image and question input



# Workshop: Speech cognitive systems

Objective

- Build a HMM-based speech recognizer

Reference

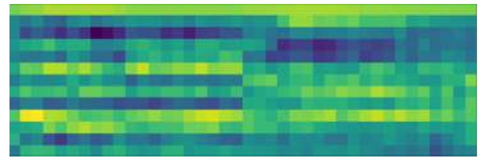
- Prateek Joshi, Python Machine Learning Cookbook, Packt Publishing, 2016, Code available at https://github.com/PacktPublishing/Python-Machine-Learning-Cookbook

# Workshop: Speech cognitive systems

apple
banana
kiwi
lime
orange
peach
pineapple

Training data:
7 classes (isolated word), 15 samples (audio clip) each class

apple01.wav
apple02.wav
apple03.wav
apple04.wav
apple05.wav
apple06.wav
apple07.wav
apple08.wav
apple09.wav
apple10.wav
apple11.wav
apple12.wav
apple13.wav
apple14.wav
apple15.wav



Extract speech features

Build a HMM model for each class

Trained HMM models
(totally, 7 models for 7 classes)

Training

Test

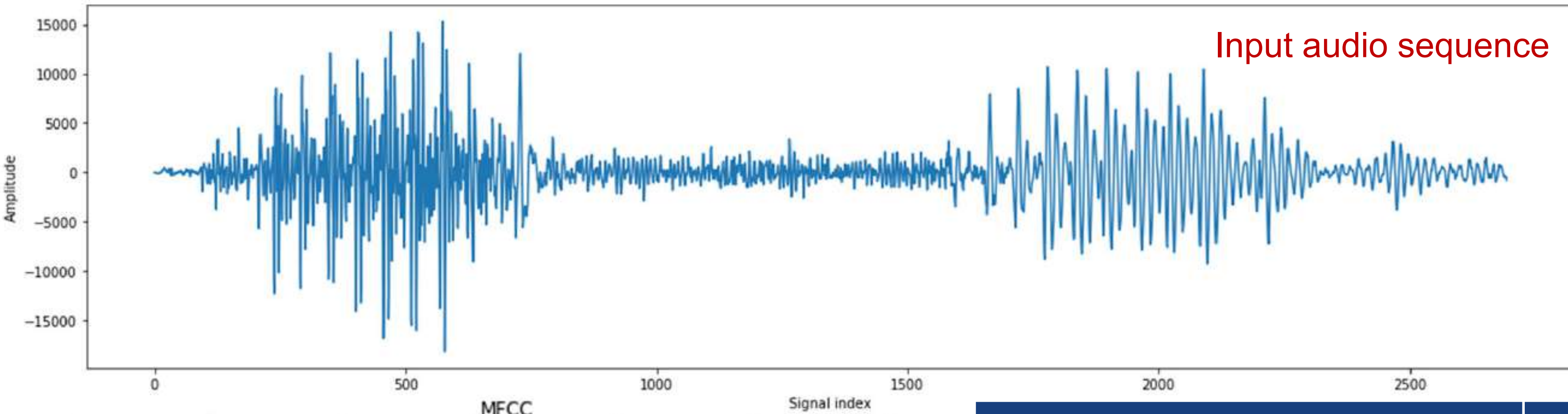
Test data

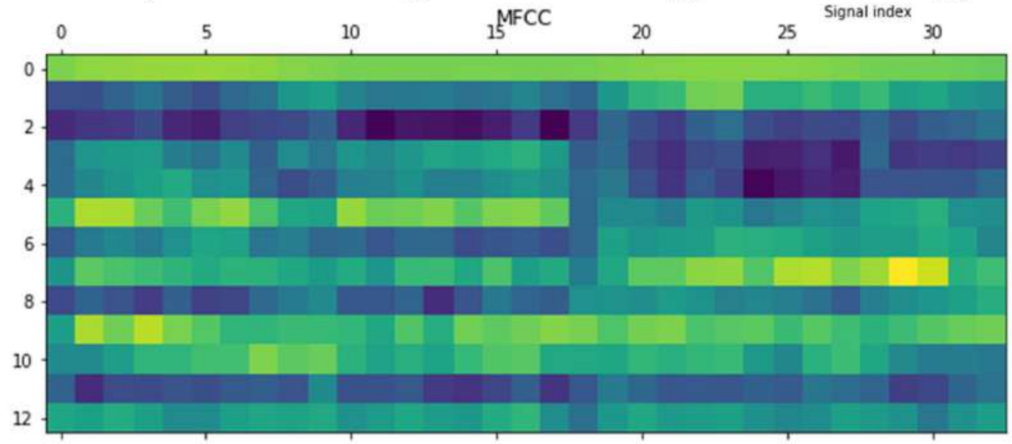
banana15.wav

Evaluate with seven models and choose the model with the largest confidence score output

Recognition result: *Banana*

# Workshop: Speech cognitive systems




| Variables | Dimensions |
|---|---|
| Input audio (apple01.wav) | (2694,) |
| MFCC feature of input signal (# windows, # features) | (33, 13) |
| HMM transition probability matrix with 3 components | (3, 3) |
| HMM state sequence | (33,) |

```
[[9.32529578e-01, 4.36394206e-26, 6.74704225e-02]
 [1.53171049e-39, 9.05902521e-01, 9.40974787e-02]
 [1.24927288e-01, 1.02681348e-01, 7.72391365e-01]]
```

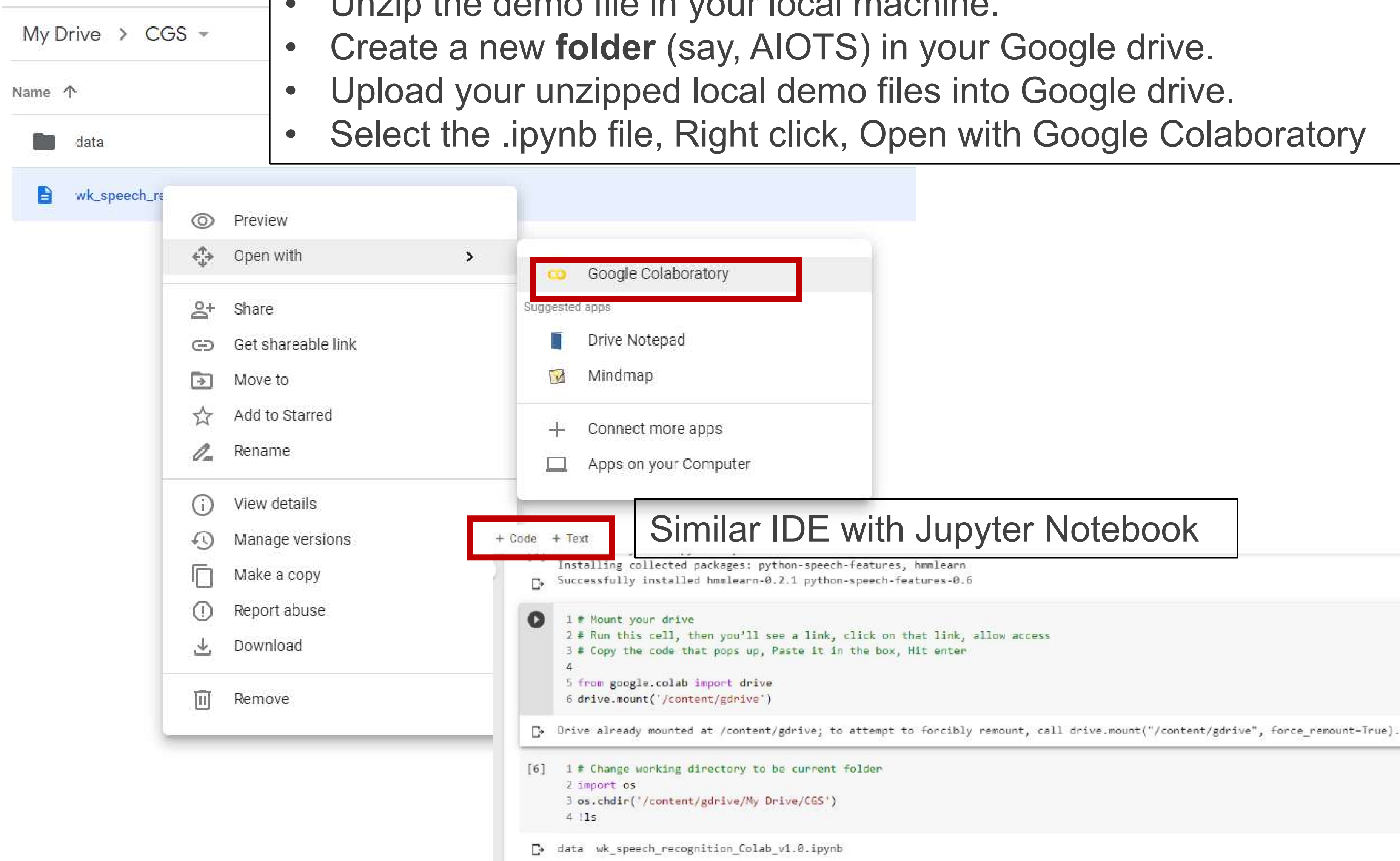
HMM transition probability matrix

```
[0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2]
```

HMM state sequence

# DEMO: Colab

- Unzip the demo file in your local machine.
- Create a new **folder** (say, AIOTS) in your Google drive.
- Upload your unzipped local demo files into Google drive.
- Select the .ipynb file, Right click, Open with Google Colaboratory




Similar IDE with Jupyter Notebook

# What we have learnt

- A typical vision cognitive system pipeline
- A statistical speech cognitive system framework
- Isolated word speech recognition using Hidden Markov model (HMM)

# Thank you!

Dr TIAN Jing
Email: tianjing@nus.edu.sg