Some applications and summary

- Natural language processing
  - Basic analysis
  - Machine translation
  - Question answering
- Computer vision
  - Object recognition
- Course summary
- Final remarks

Outline

- **Natural language processing**
  - Basic analysis
  - Machine translation
  - Question answering
- Computer vision
  - Object recognition
- Course summary
- Final remarks

Natural language processing

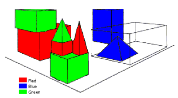
Goal: build systems that can process human language

language → $\boxed{\text{System}}$ → something useful

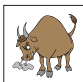
Cartoon history:

- Rule-based methods (1960s-1990s): deep and narrow
- Statistical methods (1990s-today): shallow and broad
- Holy grail: **deep and broad**

SHRDLU [Winograd 1971]



Person: Pick up a big red block.
Computer: OK.
Person: Grasp the pyramid.
Computer: I don't understand which pyramid you mean.
Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.
Computer: By "it", I assume you mean the block which is taller than the one I am holding.
Computer: OK.
Person: What does the box contain?
Computer: The blue pyramid and the blue block.
Person: What is the pyramid supported by?
Computer: The box.
Person: How many blocks are not in the box?
Computer: Four of them.
Person: Is at least one of them narrower than the one which I told you to pick up?
Computer: Yes, the red cube.



Some headlines

[slide credit: Dan Klein]

Enraged Cow Injures Farmer With Ax

Hospitals Are Sued by 7 Foot Doctors

Ban on Nude Dancing on Governor's Desk

Iraqi Head Seeks Arms

Local HS Dropouts Cut in Half

Juvenile Court to Try Shooting Defendant

Stolen Painting Found by Tree

Kids Make Nutritious Snacks

Ambiguity $\Rightarrow$ funny for humans, wrong answer for machines

Outline

- Natural language processing
  - **Basic analysis**
  - Machine translation
  - Question answering
- Computer vision
  - Object recognition
- Course summary
- Final remarks

Basic linguistic analysis

Beethoven was born in Bonn and displayed his musical talents at an early age.

Tokenization: "*an early age.*" to ["*an*", "*early*", "*age*", "*.*"]

Part-of-speech tagging: *Beethoven* is proper noun, *was* is past tense verb

Syntactic parsing: what's the subject and object of *displayed*?

Coreference resolution: *his* refers to *Beethoven*

Named entity recognition: *Beethoven* is a person, *Bonn* is a location

[Stanford CoreNLP demo]

Named-entity recognition

$\boldsymbol{y}$:	B-PER	I-PER	I-PER	O	O	O	B-LOC
$\boldsymbol{x}$:	*Ludwig*	*van*	*Beethoven*	*was*	*born*	*in*	*Bonn*

Define features $\phi(x, y) \in \mathbb{R}^d$:

- Number of times [word] was tagged [tag] (e.g., B-LOC,*Bonn*)
- Number of times [tag] follows [tag] (e.g., B-LOC,I-LOC)
- Number of times [tag] follows [part-of-speech of previous word] (e.g., B-LOC,preposition)
- Number of times [word cluster] was tagged [tag] (e.g., B-LOC,{*Bonn,Paris,London,...*})

Model: conditional random field (special case of Markov network)

$$\mathbb{P}_\theta(y \mid x) \propto \exp(\theta \cdot \phi(x, y))$$

Named-entity recognition

$\boldsymbol{y}$:	B-PER	I-PER	I-PER	O	O	O	B-LOC
$\boldsymbol{x}$:	*Ludwig*	*van*	*Beethoven*	*was*	*born*	*in*	*Bonn*

Objective: maximimum likelihood

$$\max_\theta \sum_{(x,y) \in \text{Train}} \log \mathbb{P}_\theta(y \mid x)$$

Algorithm: stochastic gradient descent

$$\theta \leftarrow \theta + \eta_t (\underbrace{\phi(x, y)}_{\text{target}} - \underbrace{\sum_{y'} \mathbb{P}_\theta(y' \mid x)\phi(x, y')}_{\text{prediction}})$$

"Prediction" term in gradient involves variable elimination

Sequence tagging: combine **machine learning** and **graphical models**

Results: train on 15K newswire sentences, get 90% test accuracy

Outline

- Natural language processing
  - Basic analysis
  - **Machine translation**
  - Question answering
- Computer vision
  - Object recognition
- Course summary
- Final remarks

[Source]

Language translation in the wild

Airline ticket office, Copenhagen: *We take your bags and send them in all directions.*

Cocktail lounge, Norway: *Ladies are Requested Not to have Children in the Bar.*

Car rental brochure, Tokyo: *When passenger of foot heave in sight, tootle the horn. Trumpet him melodiously at first, but if he still obstacles your passage then tootle him with vigor.*

Language translation in the wild

Machine translation

Today: machine translation good enough to be useful

Eine Folge von Ereignissen bewirkte, dass aus Beethovens Studienreise nach Wien ein dauerhafter und endgültiger Aufenthalt wurde. Kurz nach Beethovens Ankunft, am 18. Dezember 1792, starb sein Vater. 1794 besetzten französische Truppen das Rheinland, und der kurfürstliche Hof musste fliehen.

[Google translate demo]

Machine translation

x: *natürlich hat John spass am spiel*

$\downarrow$

y: *of course John has fun with the game*

Challenges:

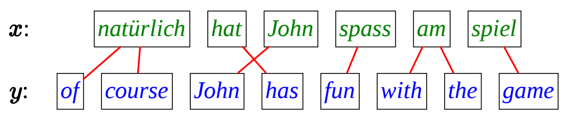
- Huge search space: outputs y are possible sentences!
- Alignment: which parts of x correspond to parts of y?

Components of a phrase-based MT system:

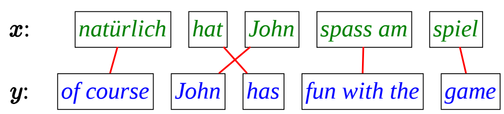
- **Phrase table**: mapping from input phrases to output phrases
- **Language model**: makes output y "look like" English
- **Decoder**: algorithm to search for good outputs y

Machine translation: phrase table

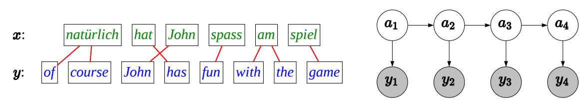
Word alignment: allows reordering, many-to-many correspondence



Phrasal alignment: captures idioms / higher-order dependencies



Word alignment model



Generative process: HMM model for word alignment

For each English position j:

Choose German position $a_j \sim p_2(a_j \mid a_{j-1})$

Choose English word $y_j \sim p_1(y_j \mid x_{a_j})$

Parameters:

- p_1(English word y_j | German word x_i)
- p_2(English position a_j | previous English position a_{j-1})

Unsupervised word alignment

English/German sentence pairs

$\downarrow$

EM algorithm on word alignment model

$\downarrow$

word alignments

$\downarrow$

phrase table

spiel $\rightarrow$ game	0.8
spiel $\rightarrow$ play	0.2
spass am $\rightarrow$ fun with the	0.3
...	

Data

Language model:

- n-gram model $\mathbb{P}(y_j \mid y_{j-1}, y_{j-2})$ with smoothing
- Define $\text{LMScore}(y_j, y_{j-1}, y_{j-2}) = \log p(y_j \mid y_{j-1}, y_{j-2})$
- Estimated from tons of raw English sentences (fully automatic)

Phrase table:

- Mapping from German to English phrases
- Define $\text{TranslationScore}(x_{i:k}, y_{j:\ell}) = \log p(y_{j:\ell} \mid x_{i:k})$
- Estimated from ≈ 10 million English/German sentence pairs (human translations)

Decoder

x: *natürlich* | *hat* | *John* | *spass am* | *spiel*

y: *of course* | *John* | *has* | *fun with the* | *game*

Objective: Given input sentence x, find

$$\arg\max_y [\text{LMScore}(y) + \text{TranslationScore}(x, y)]$$

Search problem:

- State: set of German words translated and last English words
- Actions: pick a contiguous sequence of German words and choose a English phrase
- Costs: contribution of LMScore and TranslationScore
- State space is huge, use beam search

Outline

- Natural language processing
  - Basic analysis
  - Machine translation
  - **Question answering**
- Computer vision
  - Object recognition
- Course summary
- Final remarks

Question answering

Where is the Louvre Museum located?

↓

System

↓

Paris, France

General idea:

- Run web search on rewritten queries to fetch relevant documents
- Extract candidate answers (e.g., detect named entities, etc.)
- Choose correct answer from candidates (using learned classifier)

Successful systems: AskMSR (2002), IBM Watson (2011)

Machine reading

A lot of human knowledge is contained in natural language text. How can we tap into this? Turn an agent loose on entire web.

Open Information Extraction (UW): [Demo]

- Use CRF to extract relations: "is located in"("Louvre", "Paris")
- 1 billion web pages yields 5 billion "facts"

Never-Ending Language Learning (CMU): [Demo]

- Learn classifiers to extract relations from sentences
- Bootstrapping (EM): generate new examples by running on new sentences
- Constraints on classifiers: $\forall x\, \textbf{Athlete}(x) \rightarrow \textbf{Person}(x)$

Summary of NLP

NLP draws from all the techniques:

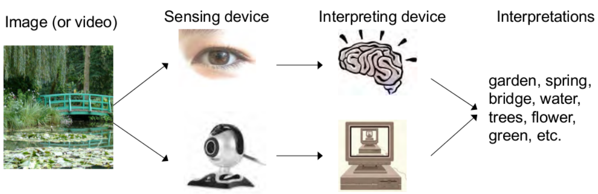
- Machine learning: cope with ambiguity / uncertainty
- Graphical models: model context dependence of words
- Search: cope with large output spaces
- Logic: represent knowledge / relations

Outline

- Natural language processing
  - Basic analysis
  - Machine translation
  - Question answering
- **Computer vision**
  - Object recognition
- Course summary
- Final remarks

[slide credit: Fei-Fei Li]

What is computer vision?



[slide credit: Fei-Fei Li]

Long-term goal: scene understanding

[slide credit: Fei-Fei Li]

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What humans see

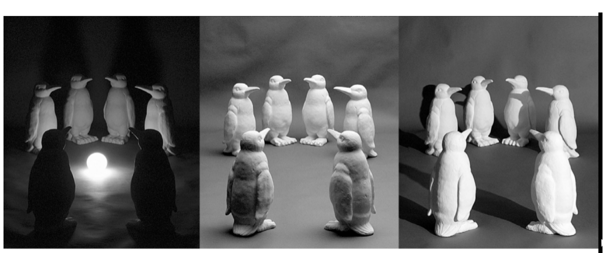
What machines see

[slide credit: Fei-Fei Li, Fergus, Torralba]

Challenge: variation in viewpoint

[slide credit: Fei-Fei Li, J. Koenderink]

Challenge: variation in illumination



[slide credit: Fei-Fei Li, Fergus, Torralba]

Challenge: intra-class variation

Outline

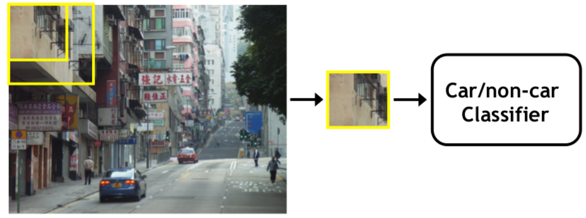
- Natural language processing
  - Basic analysis
  - Machine translation
  - Question answering
- Computer vision
  - **Object recognition**
- Course summary
- Final remarks

[K. Grauman, B. Leibe]

Object detection: sliding windows

Strategy:

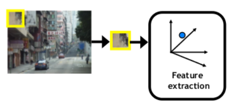
- Train a classifier mapping image region to car/non-car
- Slide a window over the image and call classifier

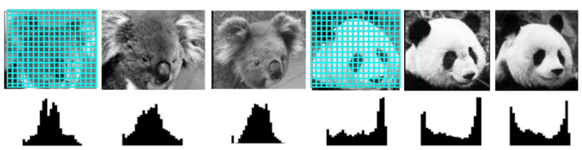


[K. Grauman, B. Leibe]

Global appearance features

Feature extraction: global appearance





Simple holistic descriptions of image content

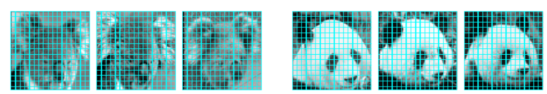
- grayscale / color histogram
- vector of pixel intensities

[K. Grauman, B. Leibe]

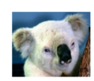
Global appearance features

Feature extraction: global appearance

- **Pixel-based representations sensitive to small shifts**



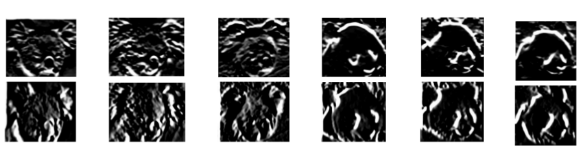
- **Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation**



Cartoon example: an albino koala

Gradient-based representations

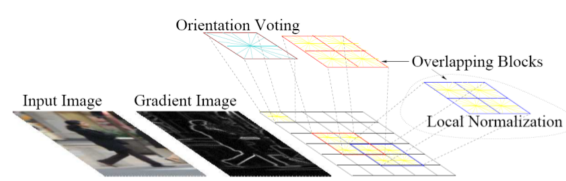
Desiderata: local features invariant to translation, rotation, scaling, deformation

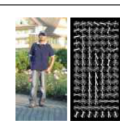


Idea: measure changes in intensity between adjacent pixels

[Dalal, Triggs, 2005]

Histograms of oriented gradients (HOG)



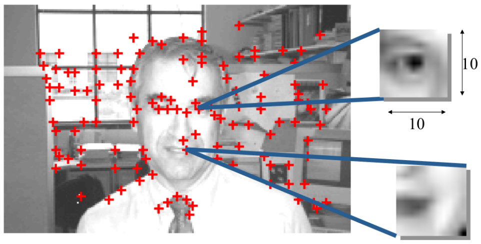


Map each grid cell in the input window to a histogram counting the gradients per orientation.

[slide credit: Fei-Fei Li]

Visual bag of words

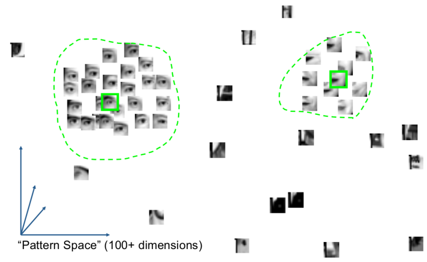
Run feature descriptor (e.g., HOG) on tons of images



[slide credit: Fei-Fei Li]

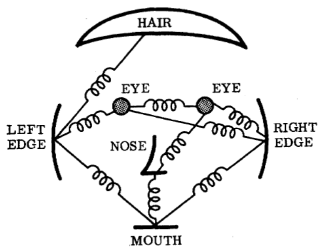
Visual bag of words

Use K-means to cluster features into visual code words



[Felzenszwalb et al. 2011]

Deformable part-based models



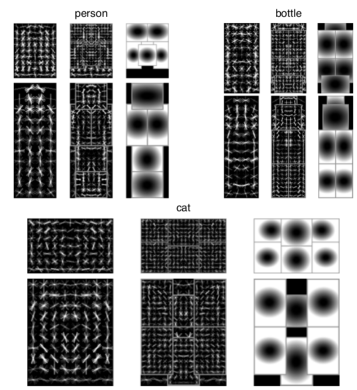
Given image x, score defined in terms of a latent configuration z:

$$\text{Score}(x) = \max_z \mathbf{w} \cdot \phi(x, z)$$

Method: SVM with latent variables, stochastic gradient descent

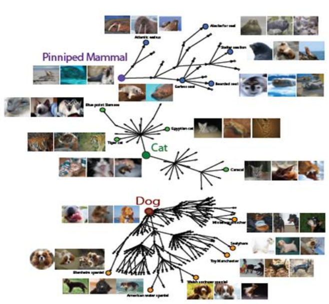
[Felzenszwalb et al. 2011]

Deformable part-based models



ImageNet

14 million images, 22K categories



[Browse]

Outline

- Natural language processing
  - Basic analysis
  - Machine translation
  - Question answering
- Computer vision
  - Object recognition
- **Course summary**
- Final remarks

Maximize expected utility

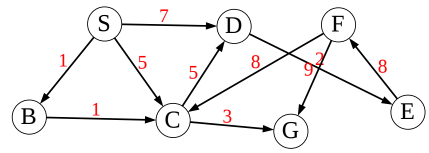
Separate the "what" (objective) from the "how" (algorithm):

- Search problems: minimize path cost
- MDPs: maximize expected cumulative rewards
- Weighted CSPs: find maximum weight assignment
- Markov networks: compute conditional probability queries
- Learning: minimize regularized training loss
- Logic: compute semantic entailment

Strategy: Define **locally** (edge costs, factors), think **globally** (run inference algorithm)

State space models

Ingredients:
$\text{States}, \text{Actions}(s), \text{Cost}(s,a), \text{Succ}(s,a), \text{IsGoal}(s)$

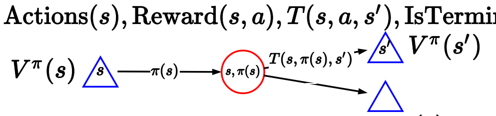


State: minimal information about history to keep relevant to evaluate/choose future actions (e.g., budget constraints)

Algorithms: DAG search, DFS, BFS, UCS, A*, Bellman-Ford work under different model assumptions

MDPs

Ingredients:
$\text{States}, \text{Actions}(s), \text{Reward}(s,a), T(s,a,s'), \text{IsTerminal}(s)$



Solution: policy π mapping each state s to action $\pi(s)$

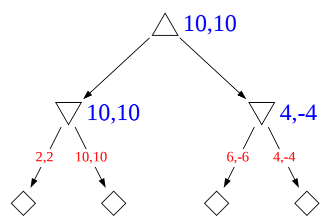
Recurrence:

$$V(s) = \max_{a \in \text{Actions}(s)} \underbrace{\text{Reward}(s,a)}_{\text{immediate reward}} + \underbrace{\sum_{s'} T(s,a,s') V(s')}_{\text{future reward}}$$

Algorithms: value iteration / policy iteration repeatedly apply recurrence

Games

Setup: assume opponent is not random but adversarial (minimax)

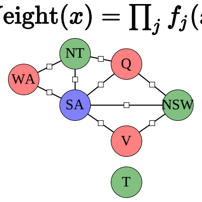


Improve efficiency: alpha-beta pruning (exact), evaluation functions (approximate)

Variable-based models

Ingredients: variables and factors

$$\text{Weight}(x) = \prod_j f_j(x)$$



Operations to remove variables:

- Conditioning (observe one value): break up the graph
- Elimination (consider all values): connect Markov blanket

Tree-width: minimum over orderings of maximum arity of new factor

Variable-based models

Markov networks:

- $\mathbb{P}(X = x) \propto \text{Weight}(x)$
- To compute $\mathbb{P}(A \mid B)$: (i) condition on B, (ii) eliminate all except A, (iii) normalize distribution.

Bayesian networks:

- Easy/fun to define models (generative story)
- Factors are local conditional probability tables
- Inference: eliminate nodes with no evidence attached for free, then treat as Markov network

Learning

Objective: regularized loss minimization (remember, only surrogate for true utility)

$$\min_{\mathbf{w}} \sum_{(x,y)\in\text{Train}} \text{Loss}(x, y, \mathbf{w}) + \lambda\text{Penalty}(\mathbf{w})$$

Algorithm: stochastic gradient descent

$$\mathbf{w} \to \mathbf{w} - \eta_t(\text{prediction} - \text{target})$$

Unsupervised learning: reconstruction loss, chicken-egg problem, K-means / EM

Reinforcement learning: exploration/exploitation tradeoff, Q-learning (online regression)

Logic

Formula f: represents a set of models, defined compositionally via interpretation function

Objective: semantic entailment $\text{KB} \models f$

Algorithm: define inference rules, and apply rules (e.g., resolution, Modus ponens)

Types of logic: propositional (propositions), first-order (relations/objects), restrictions to definite clauses, etc.

Purpose: allows operating at a higher-level of abstraction

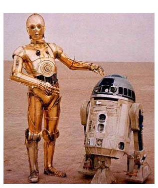
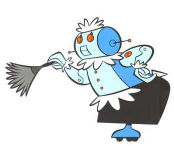
Outline

- Natural language processing
  - Basic analysis
  - Machine translation
  - Question answering
- Computer vision
  - Object recognition
- Course summary
- **Final remarks**

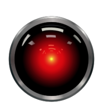
Other AI-related courses

- CS228: Probabilistic Graphical Models
- CS229: Machine Learning
- **CS229T: Statistical Learning Theory**
- **CS246: Mining Massive Data Sets**
- CS224N: Natural Language Processing
- **CS224U: Natural Language Understanding**
- CS224W: Social and Information Network Analysis
- **CS223A: Introduction to Robotics**
- CS225B: Robot Programming Lab
- CS227B: General Game Playing
- CS231A: Introduction to Computer Vision
- CS276: Information Retrieval and Web Search
- CS334A: Convex Optimization
- STAT315A/B/C: Modern Applied Statistics: Elements of Statistical Learning
- PSYCH204: Computation and Cognition: the Probabilistic Approach

Future of AI?





Future of AI?

1950s: seek general-purpose intelligence in a top-down manner, but failed due to limited methods, computation, and data

Last 50 years: AI has fragmented into different subcommunities (language/vision/robotics/planning), address specific problems (quite successfully!)

Next 50 years:

- Need to connect the subcommunities (we have unprecedented amount of rich multimedia data — how do we exploit it?)
- Build robust and interpretable systems to guard against bad behavior

Utility to society: efficiency, safety, health, environment, etc.

Please provide feedback in course evaluations.

Thanks for an exciting first quarter!