ARTIFICIAL INTELLIGENCE

Since the invention of computers or machines, their capability to perform various tasks went on growing exponentially. Humans have developed the power of computer systems in terms of their diverse working domains, their increasing speed, and reducing size with respect to time.

A branch of Computer Science named *Artificial Intelligence* pursues creating the computers or machines as intelligent as human beings.

What is Artificial Intelligence?

According to the father of Artificial Intelligence, John McCarthy, it is *“The science and engineering of making intelligent machines, especially intelligent computer programs”.*

Artificial Intelligence is a way of **making a computer, a computer -controlled robot, or a software think intelligently**, in the similar manner the intelligent humans think.

AI is accomplished by studying how human brain thinks, and how humans learn, decide, and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems.

Philosophy of AI

While exploiting the power of the computer systems, the curiosity of human, lead him to wonder,
“Can a machine think and behave like humans do?”

Thus, the development of AI started with the intention of creating similar intelligence in machines that we find and regard high in humans.

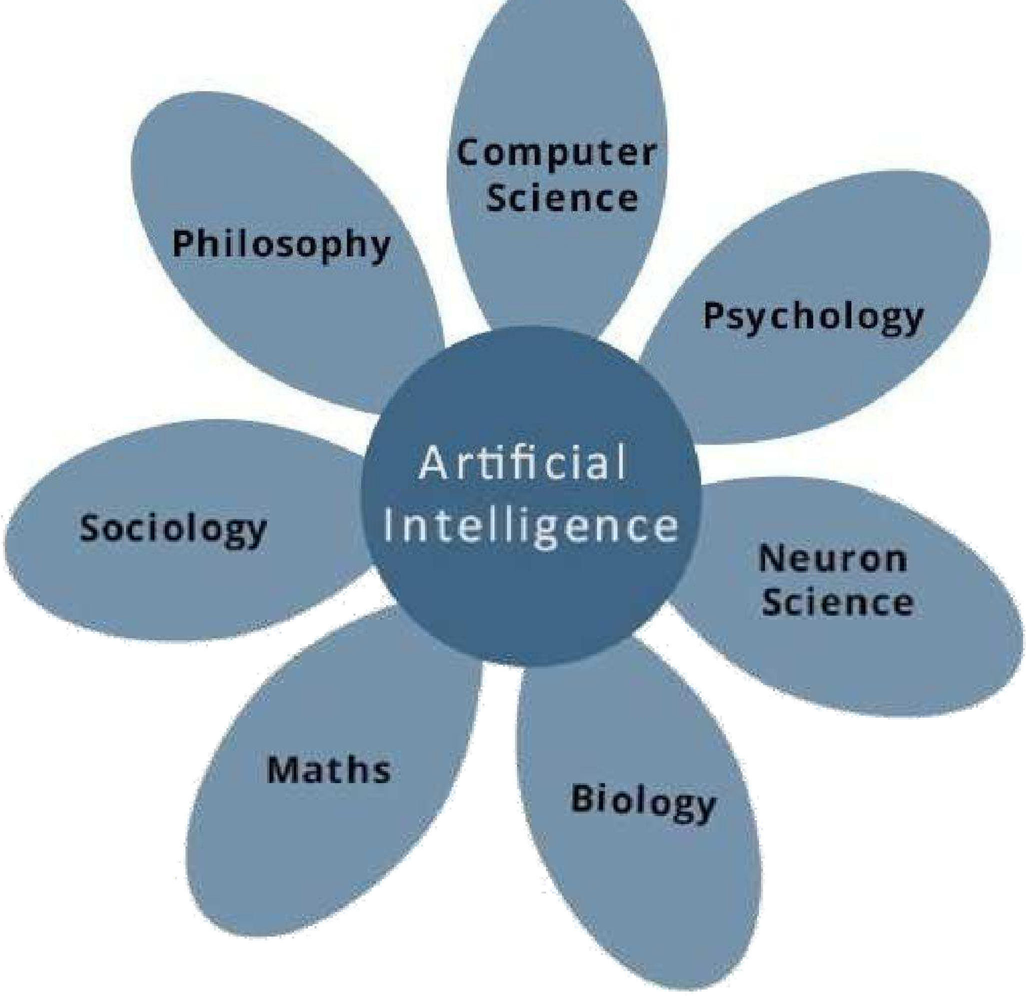
Goals of AI

- **To Create Expert Systems** − The systems which exhibit intelligent behavior, learn, demonstrate, explain, and advice its users.
- **To Implement Human Intelligence in Machines** − Creating systems that understand, think, learn, and behave like humans.

What Contributes to AI?

Artificial intelligence is a science and technology based on disciplines such as Computer Science, Biology, Psychology, Linguistics, Mathematics, and Engineering. A major thrust of AI is in the development of computer functions associated with human intelligence, such as reasoning, learning, and problem solving.

Out of the following areas, one or multiple areas can contribute to build an intelligent system.



Programming Without and With AI

The programming without and with AI is different in following ways −

Programming Without AI	Programming With AI
A computer program without AI can answer the **specific** questions it is meant to solve.	A computer program with AI can answer the **generic** questions it is meant to solve.
Modification in the program leads to change in its structure.	AI programs can absorb new modifications by putting highly independent pieces of information together. Hence you can modify even a minute piece of information of program without affecting its structure.
Modification is not quick and easy. It may lead to affecting the program adversely.	Quick and Easy program modification.

-
-
-

-
-
-

What is AI Technique?

In the real world, the knowledge has some unwelcomed properties −

Its volume is huge, next to unimaginable.

It is not well-organized or well-formatted.

It keeps changing constantly.

AI Technique is a manner to organize and use the knowledge efficiently in such a way that −

It should be perceivable by the people who provide it.

It should be easily modifiable to correct errors.

It should be useful in many situations though it is incomplete or inaccurate.

AI techniques elevate the speed of execution of the complex program it is equipped with.

Applications of AI

AI has been dominant in various fields such as −

Gaming − AI plays crucial role in strategic games such as chess, poker, tic-tac-toe, etc., where machine can think of large number of possible positions based on heuristic knowledge.

Natural Language Processing − It is possible to interact with the computer that understands natural language spoken by humans.

- **Expert Systems** − There are some applications which integrate machine, software, and special information to impart reasoning and advising. They provide explanation and advice to the users.

- **Vision Systems** − These systems understand, interpret, and comprehend visual input on the computer. For example,
  - A spying aeroplane takes photographs, which are used to figure out spatial information or map of the areas.
  - Doctors use clinical expert system to diagnose the patient.
  - Police use computer software that can recognize the face of criminal with the stored portrait made by forensic artist.
- **Speech Recognition** − Some intelligent systems are capable of hearing and comprehending the language in terms of sentences and their meanings while a human talks to it. It can handle different accents, slang words, noise in the background, change in human's noise due to cold, etc.

- **Handwriting Recognition** − The handwriting recognition software reads the text written on paper by a pen or on screen by a stylus. It can recognize the shapes of the letters and convert it into editable text.

- **Intelligent Robots** − Robots are able to perform the tasks given by a human. They have sensors to detect physical data from the real world such as light, heat, temperature, movement, sound, bump, and pressure. They have efficient processors, multiple sensors and huge memory, to exhibit intelligence. In addition, they are capable of learning from their mistakes and they can adapt to the new environment.

History of AI

Here is the history of AI during 20th century −

Year	Milestone / Innovation
1923	Karel Kapek's play named “Rossum's Universal Robots” *RUR* opens in London, first use of the word "robot" in English.
1943	Foundations for neural networks laid.
1945	Isaac Asimov, a Columbia University alumni, coined the term *Robotics*.
1950	Alan Turing introduced Turing Test for evaluation of intelligence and published *Computing Machinery and Intelligence.* Claude Shannon published *Detailed Analysis of Chess Playing* as a search.
1956	John McCarthy coined the term *Artificial Intelligence*. Demonstration of the first running AI program at Carnegie Mellon University.
1958	John McCarthy invents LISP programming language for AI.
1964	Danny Bobrow's dissertation at MIT showed that computers can understand natural language well enough to solve algebra word problems correctly.
1965	Joseph Weizenbaum at MIT built *ELIZA*, an interactive problem that carries on a dialogue in English.
1969	Scientists at Stanford Research Institute Developed *Shakey*, a robot, equipped with locomotion, perception, and problem solving.
1973	The Assembly Robotics group at Edinburgh University built *Freddy*, the Famous Scottish Robot, capable of using vision to locate and assemble models.
1979	The first computer-controlled autonomous vehicle, Stanford Cart, was built.

1985 Harold Cohen created and demonstrated the drawing program, *Aaron*.

1990 Major advances in all areas of AI −

- Significant demonstrations in machine learning
- Case-based reasoning
- Multi-agent planning
- Scheduling
- Data mining, Web Crawler
- natural language understanding and translation
- Vision, Virtual Reality
- Games

1997 The Deep Blue Chess Program beats the then world chess champion, Garry Kasparov.

2000 Interactive robot pets become commercially available. MIT displays *Kismet*, a robot with a face that expresses emotions. The robot *Nomad* explores remote regions of Antarctica and locates meteorites.

Artificial Intelligence

Intelligence vs Artificial Intelligence

- Intelligence is a property/ability attributed to people, such as to know, to think, to talk, to learn, to understand.

Intelligence = Knowledge + ability to perceive, feel, comprehend, process, communicate, judge, learn.

- Artificial Intelligence is an <u>interdisciplinary</u> field aiming at developing techniques and tools for solving problems that people at good at.

Definitions of AI

- Computers with the ability to mimic or duplicate the functions of the human brain.
- Artificial Intelligence (AI) is the study of how computer systems can simulate intelligent processes such as learning, reasoning, and understanding symbolic information in context.

- It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence.
- “The branch of computer science that is concerned with the automation of intelligent behaviour” (Luger and Stubblefield, 1993).

The State of the Art Example: The Semantic Web

The Semantic Web is the latest most prominent example of applied AI. It allows data to be linked across the web, and thus understood by computers so that they can perform increasingly sophisticated tasks which were previously delegated to humans.

https://www.cambridgesemantics.com/blog/semantic-university/intro-semantic-web/

“The word *semantic* itself implies *meaning* or *understanding*. As such, the fundamental difference between Semantic Web technologies and other technologies related to data (such as relational databases or the World Wide Web itself) is that the Semantic Web is concerned with the *meaning* and not the structure of data.”

Why "meaning" is the central concept of AI

- For an entity to be "intelligent", it must be able to understand the meaning of information.
- Information is acquired / delivered / conveyed in messages which are phrased in a selected representation language.
- There are two sides in information exchange: the source (text, image, person, program, etc.) and the receiver (person or an AI agent). They must speak the same "language" for the information to be exchanged in a meaningful way.
- The receiver must have the ability to interpret the information correctly according to the intended by the source meaning / semantics of it.

MEANING = SEMANTICS

Applications of AI:

- Natural Language Understanding
- Expert Systems
- Planning and Robotics
- Machine Learning
- Game Playing

Natural Language Processing

- To design and build software that will analyze understand and generate languages that human use naturally.

Modes of communication

- Text based.

- Dialog

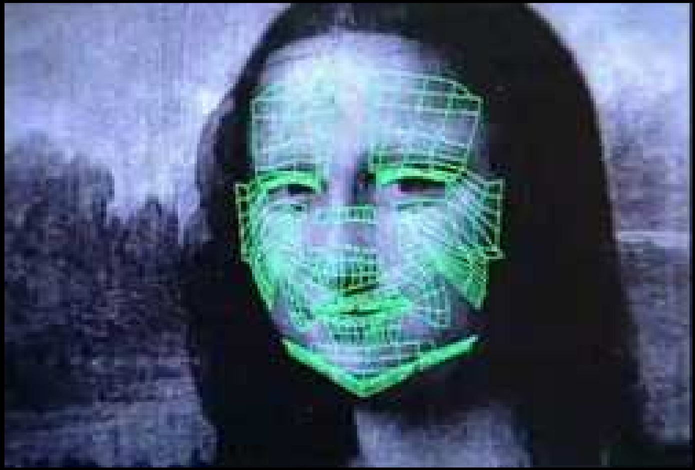
Speech Recognition

- Process of converting sound signal captured by microphone or mobile/telephone to a set of words.
- 70-100 words / min with accuracy of 90%

Computer Vision

- Ability of a machine to extract information from an image that is necessary to solve a task

- Image Acquisition
- Image Processing
- Image Analysis
- Image understanding

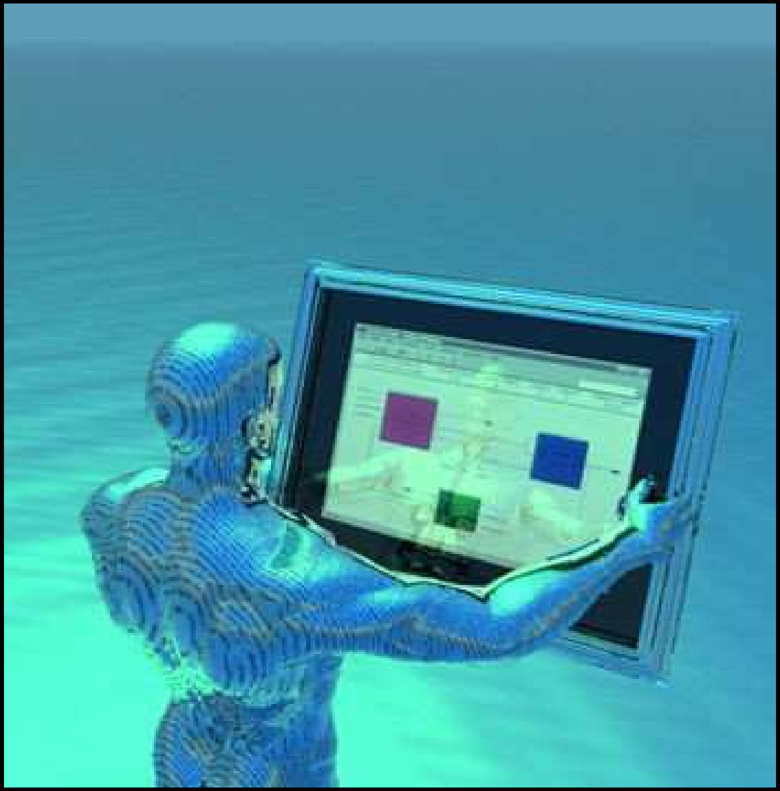


Intelligent Robot

- Tend to mimic human sensing and decision making abilities so that they can adopt themselves to certain conditions and modify their actions.

Expert Systems

- These are Softwares used for decision making .
- Automated Reasoning and Theorem Proving.
- Troubleshooting Expert Systems.



Foundations of Artificial Intelligence

Structure of intelligent agents and environments

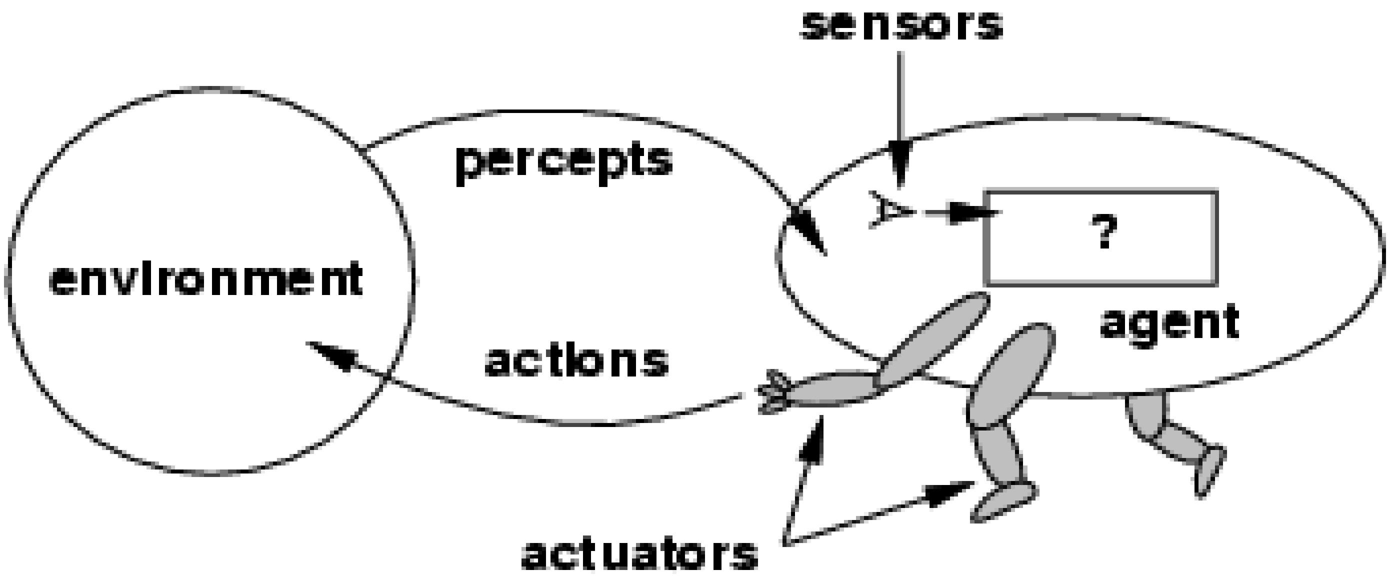
Agents

Definition: An agent perceives its environment via sensors and acts upon that environment through its actuators

---

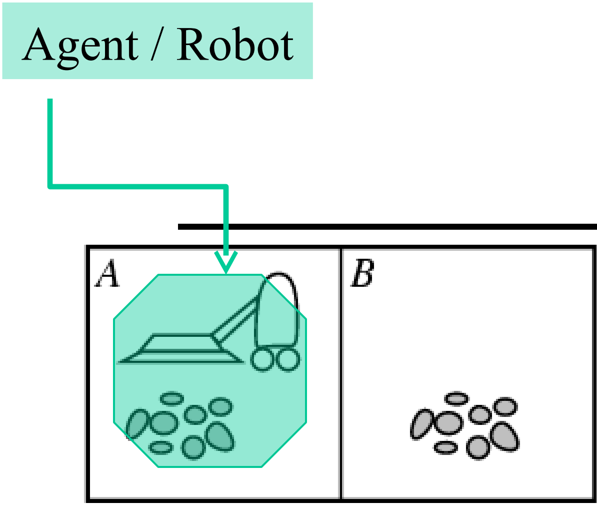
Human agent: eyes, ears, and other organs for sensors; hands, legs, mouth, and other body parts for actuators

Robotic agent: camera and microphone for sensors; various motors for actuators



Next: definition of rational agents. Sufficiently complex rational agents can be viewed as "intelligent agents."

E.g., vacuum-cleaner world



iRobot Roomba® 400
Vacuum Cleaning Robot

iRobot Corporation

Founder Rodney Brooks (MIT)

Percepts: location and contents,
e.g., [A, Dirty]

Actions: *Left*, *Right*, *Clean*, *NoOp*

- Powerful suction and rotating brushes
- Automatically navigates for best cleaning coverage
- Cleans under and around furniture, into corners and along wall edges
- Self-adjusts from carpets to hard floors and back again
- Automatically avoids stairs, drop-offs and off-limit areas
- Simple to use— just press the Clean button and Roomba does the rest

Rational agents

An agent should strive to "do the right thing", based on what:

- ~~it can perceive and~~
- the actions it can perform.

The right action is the one that will cause the agent to be most successful

Performance measure: An *objective criterion for success* of an agent's behavior.

Performance measures of a vacuum-cleaner agent: amount of dirt cleaned up, amount of time taken, amount of electricity consumed, level of noise generated, etc.

Performance measures self-driving car: time to reach destination (minimize), safety, predictability of behavior for other agents, reliability, etc.

Performance measure of game-playing agent: win/loss percentage (maximize), robustness, unpredictability (to "confuse" opponent), etc.

Definition of Rational Agent:

For each possible percept sequence, a rational agent should select an action that maximizes its performance measure (in expectation) given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Why "in expectation"?

Captures actions with stochastic / uncertain effects or actions performed in stochastic environments.
We can then look at the expected value of an action.

Rational agents

~~Rationality is distinct from omniscience ("all knowing"). We~~ can behave rationally even when faced with incomplete information.

Agents can perform actions in order to modify future percepts so as to obtain useful information: information gathering, exploration.

An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt).

Characterizing a Task Environment

Must first specify the setting for intelligent agent design.

PEAS: Performance measure, Environment, Actuators, Sensors

Example: the task of designing a self-driving car

- **Performance measure** **Safe, fast, legal, comfortable trip**
- **Environment** **Roads, other traffic, pedestrians**
- **Actuators** **Steering wheel, accelerator, brake, signal, horn**
- **Sensors** **Cameras, LIDAR (light/radar), speedometer, GPS, odometer**
- **engine sensors, keyboard**

Task Environments

1) Fully observable / Partially observable

- If an agent's sensors give it access to the complete state of the environment needed to choose an action, the environment is fully observable.

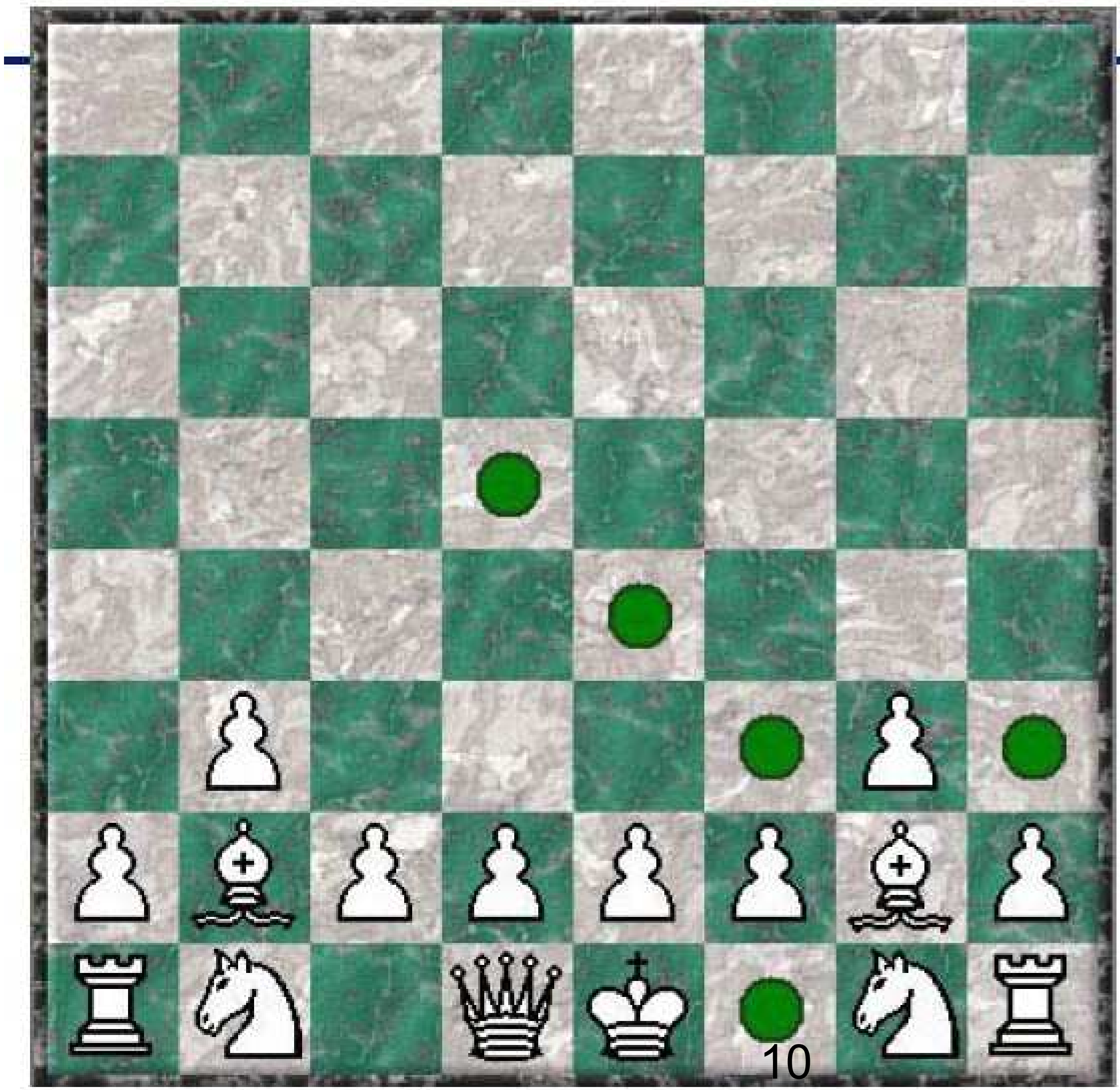
(e.g. chess – what about Kriegspiel?)

Making things a bit more challenging…
Kriegspiel --- you can't see your opponent!

Incomplete / uncertain information inherent in the game.

Balance **exploitation** (best move given current knowledge) and **exploration** (moves to explore where opponent's pieces might be).

Use probabilistic reasoning techniques.



2) Deterministic / Stochastic

- An environment is deterministic if the next state of the environment is completely determined by the current state of the environment and the action of the agent;
- In a stochastic environment, there are multiple, unpredictable outcomes. (If the environment is deterministic except for the actions of other agents, then the environment is strategic).

In a fully observable, deterministic environment, the agent need not deal with uncertainty.

Note: Uncertainty can also arise because of computational limitations. E.g., we may be playing an omniscient ("all knowing") opponent but we may not be able to compute his/her moves.

3) Episodic / Sequential

- **In an episodic environment, the agent's experience is divided into atomic episodes. Each episode consists of the agent perceiving and then performing a single action.**

- **Subsequent episodes do not depend on what actions occurred in previous episodes. Choice of action in each episode depends only on the episode itself.**

- **In a sequential environment, the agent engages in a series of connected episodes. Current decision can affect future decisions. (E.g., chess and driving)**

4) Static / Dynamic

- **A static environment does not change while the agent is thinking.**

- **The passage of time as an agent deliberates is irrelevant.**

- **The environment is semidynamic if the environment itself does not change**

5) Discrete / Continuous

- If the number of distinct percepts and actions is limited, the environment is discrete, otherwise it is continuous.

6) Single agent / Multi-agent

- If the environment contains other intelligent agents, the agent needs to be concerned about strategic, game-theoretic aspects of the environment (for either cooperative *or* competitive agents).
- Most engineering environments don’t have multi-agent properties, whereas most social and economic systems get their complexity from the interactions of (more or less) rational agents.

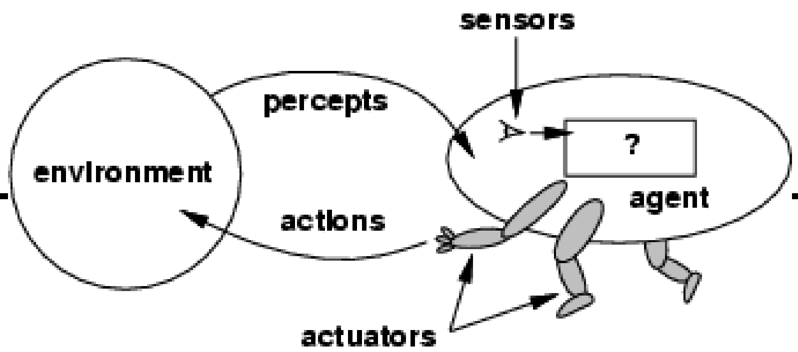
Example Tasks and Environment Types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Agents and environments



The agent function maps from percept histories to actions

$f: \mathcal{P}^* \rightarrow \mathcal{A}$

The agent program runs (internally) on the physical architecture to produce f

Types of Agents

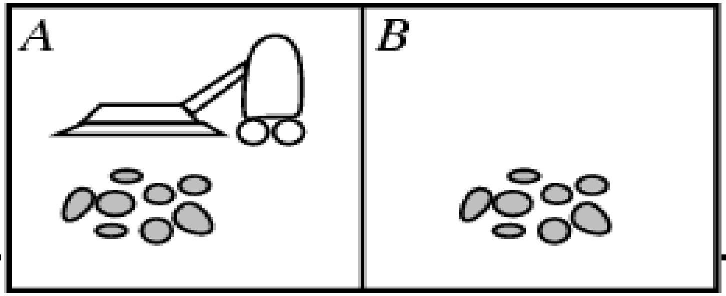
I) --- Table-lookup driven agents

Uses a percept sequence / action table in memory to find the next action. Implemented as a (large) lookup table.

Drawbacks:

- Huge table (often simply too large)
- Takes a long time to build/learn the table

Toy example:
Vacuum world.



Percepts: robot senses it's location and "cleanliness."

So, location and contents, e.g., [A, Dirty], [B, Clean].

With 2 locations, we get **4 different possible sensor inputs.**

Actions: *Left*, *Right*, *Clean*, *NoOp*

II) --- Simple reflex agents

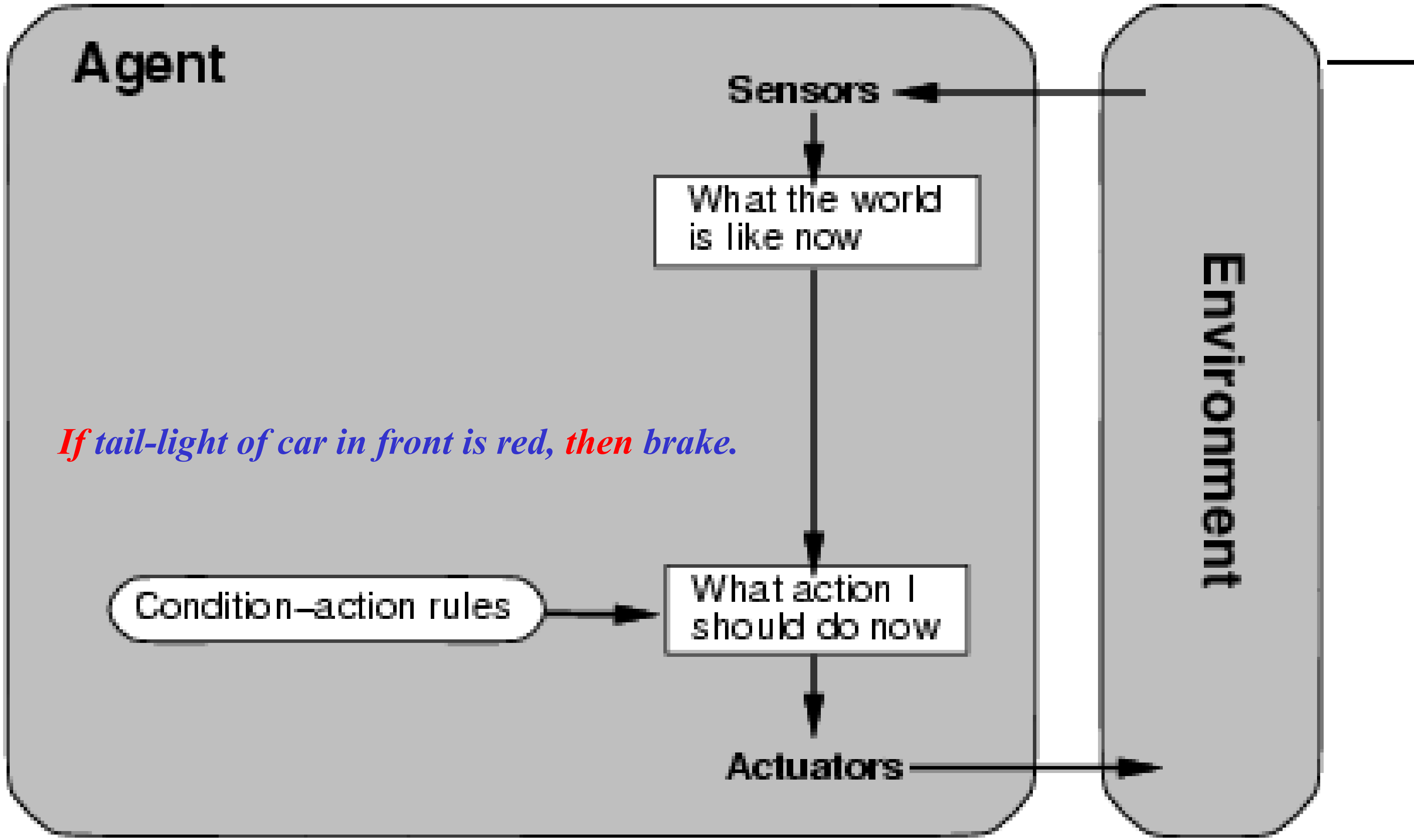
Agents **do not have memory** of past world states or percepts.

So, actions depend solely on **current percept**.

Action becomes a “reflex.”

Uses **condition-action rules**.

Agent selects actions on the basis of *current* percept only.



Simple reflex agents

Closely related to "behaviorism" (psychology; quite effective in explaining lower-level animal behaviors, such as the behavior of ants and mice).

The Roomba robot largely behaves like this. Behaviors are robust and can be quite effective and surprisingly complex.

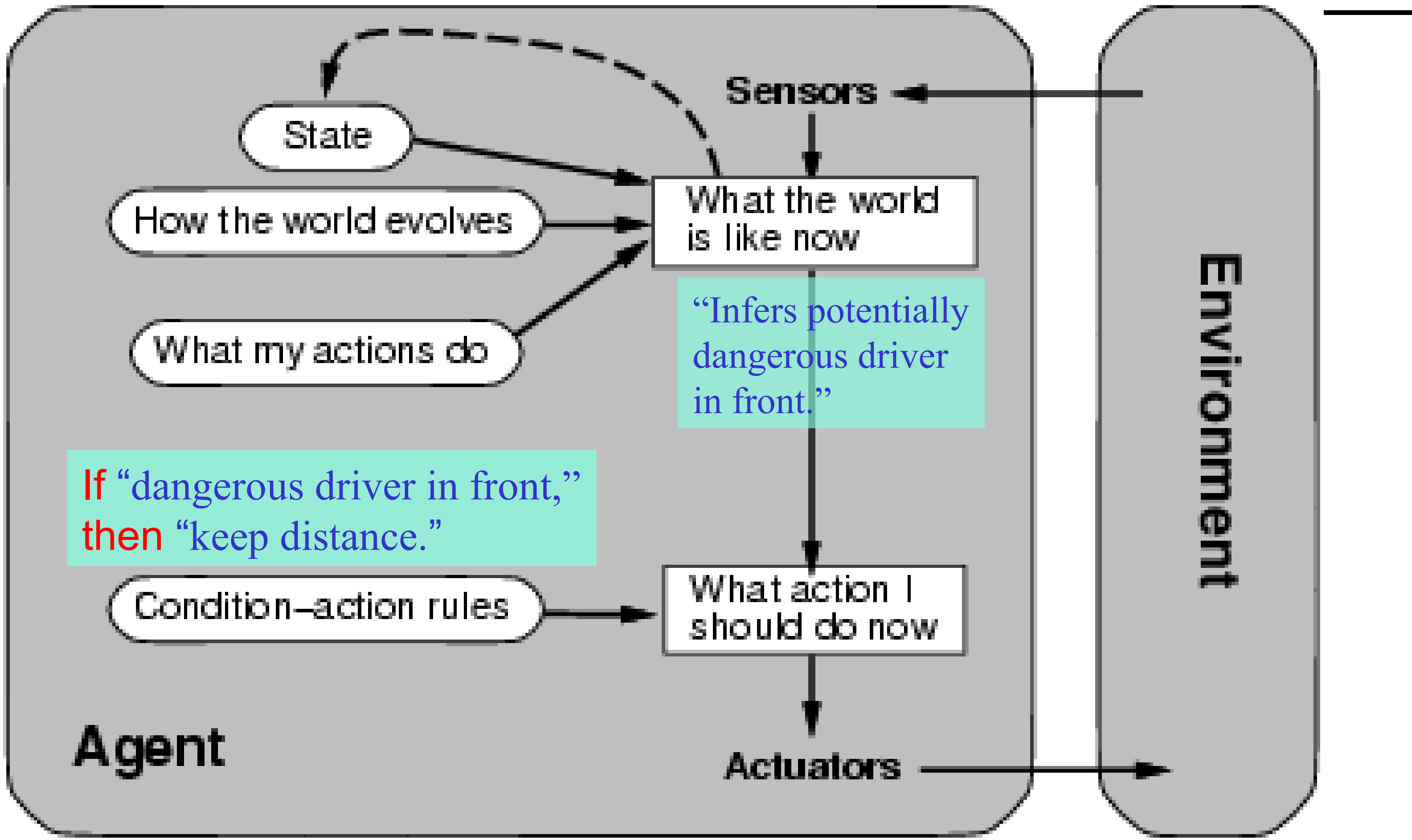
But, how does complex behavior arise from simple reflex behavior?
E.g. ants colonies and bee hives are quite complex.

III) --- Model-based reflex agents

Key difference (wrt simple reflex agents):

- Agents have **internal state**, which is used to keep track of past states of the world.
- Agents have the ability **to represent change in the World**.

Model-based reflex agents



IV) --- Goal-based agents

Key difference wrt Model-Based Agents:

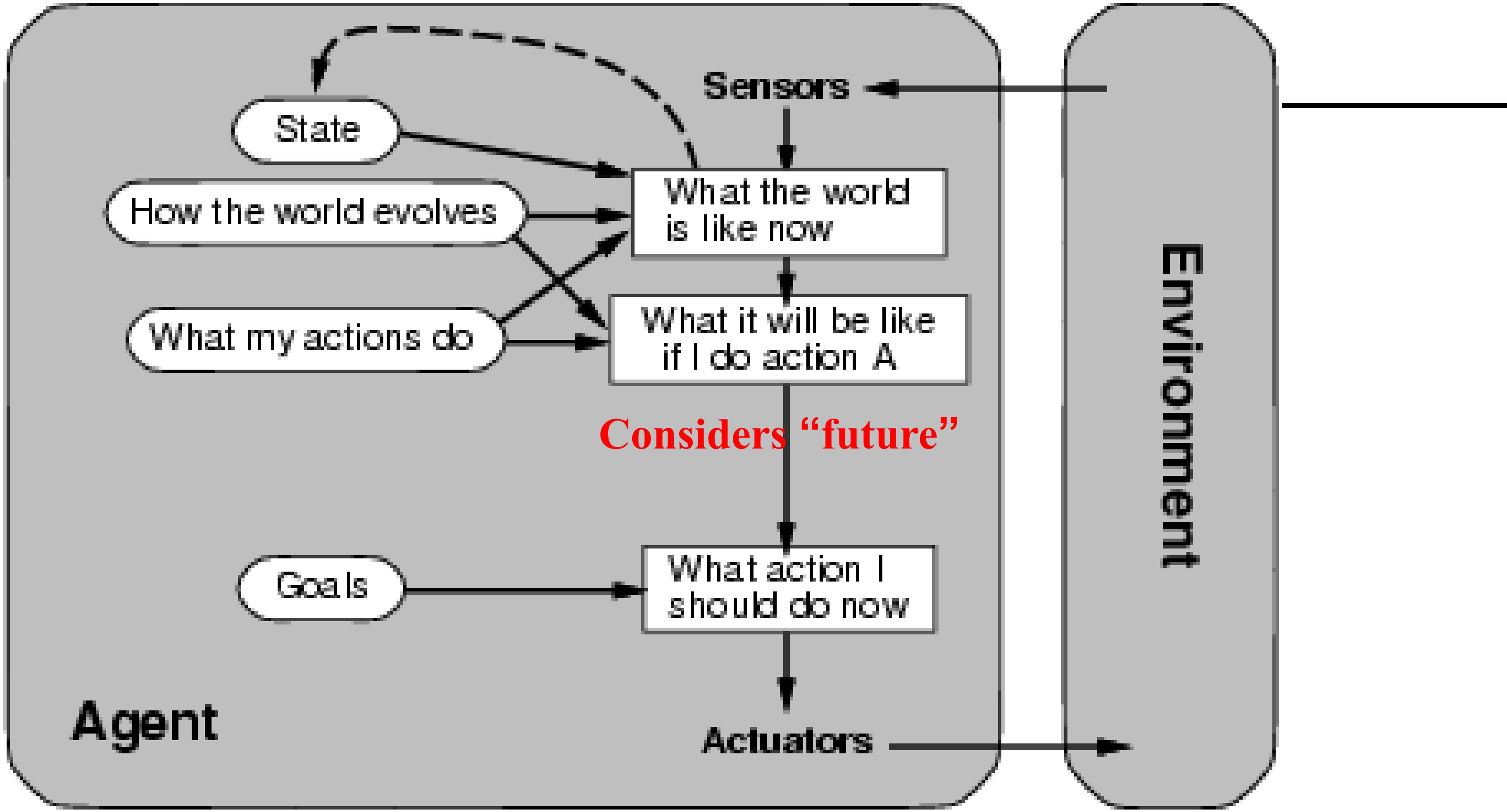
In addition to state information, have **goal information** that **describes desirable situations to be achieved**.

Agents of this kind take future events into consideration.

What *sequence* of actions can I take to achieve certain goals?

Choose actions so as to (eventually) achieve a (given or computed) goal.

Goal-based agents



Agent keeps track of the world state as well as set of goals it's trying to achieve: chooses actions that will (eventually) lead to the goal(s).

More flexible than reflex agents → may involve search and planning

V) --- Utility-based agents

When there are multiple possible alternatives, how to decide which one is best?

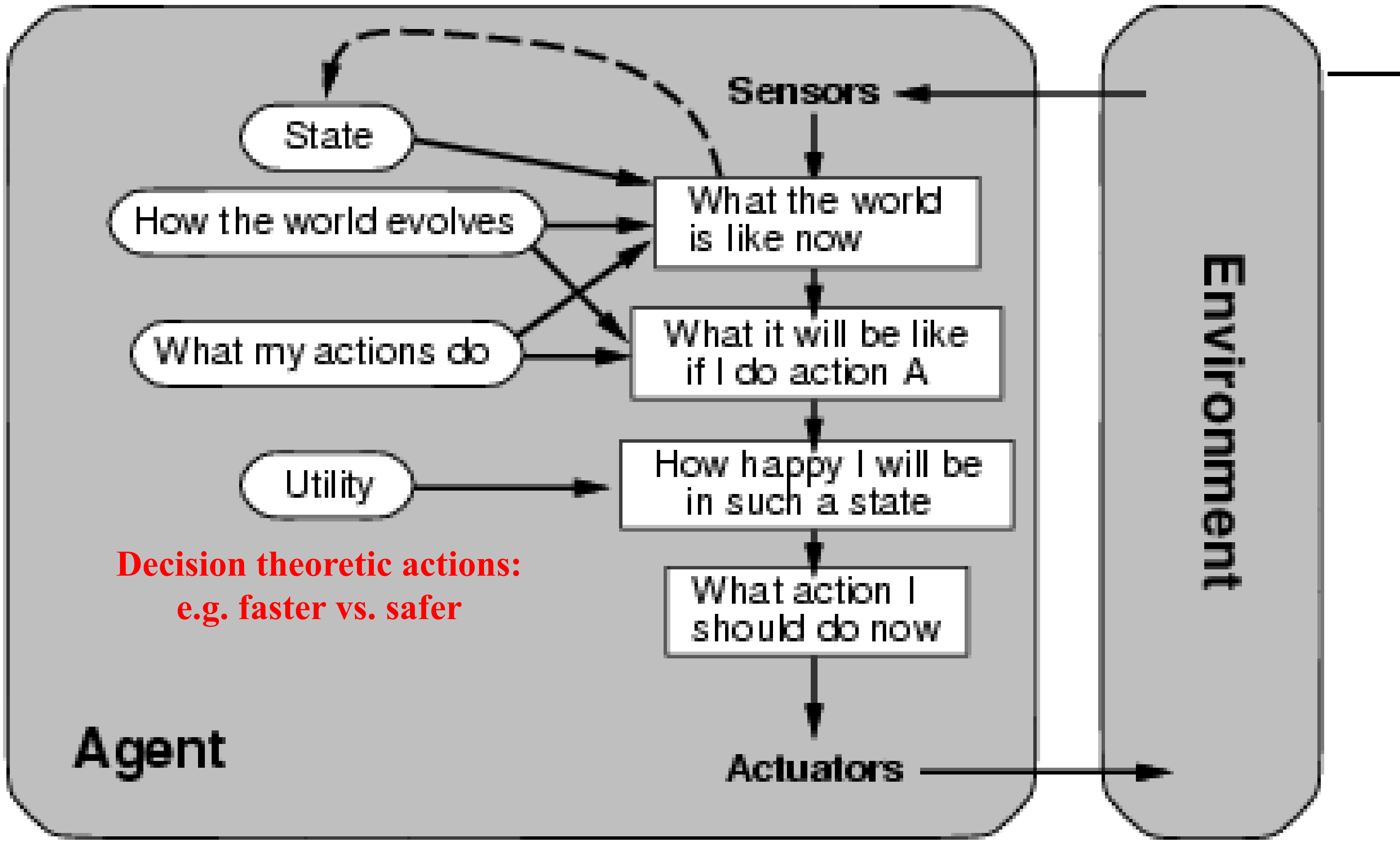
Goals are qualitative: A goal specifies a crude distinction between a happy and unhappy state, but often need a more general performance measure that describes "degree of happiness."

Utility function U: State   R indicating a measure of success or happiness when at a given state.

Important for making tradeoffs: Allows decisions comparing choice between conflicting goals, and choice between likelihood of success and importance of goal (if achievement is uncertain).

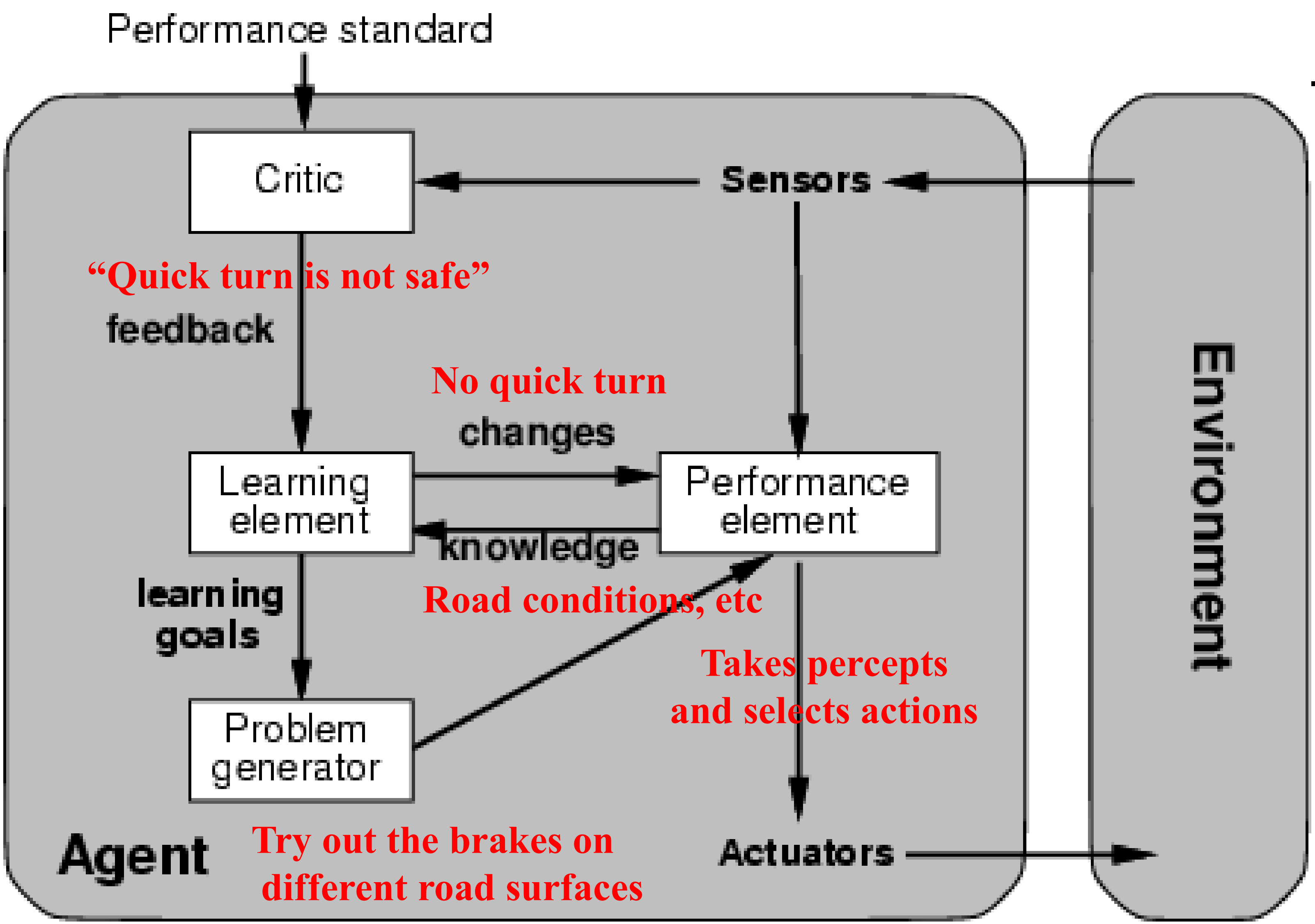
Use decision theoretic models: e.g., faster vs. safer.

Utility-based agents



VI) --- Learning agents
Adapt and improve over time

More complicated when agent needs to learn utility information: Reinforcement learning (based on action payoff)



Summary: agent types

(1) Table-driven agents

- use a percept sequence/action table in memory to find the next action. They ~~are implemented by a (large) lookup table.~~

(2) Simple reflex agents

- are based on condition-action rules, implemented with an appropriate production system. They are <u>stateless devices which do not have memory of past world states.</u>

(3) Agents with memory - Model-based reflex agents

- have internal state, which is used to keep track of past states of the world.

(4) Agents with goals – Goal-based agents

- are agents that, in addition to state information, have goal information that describes desirable situations. Agents of this kind take future events into consideration.

(5) Utility-based agents

- base their decisions on classic axiomatic utility theory in order to act rationally.

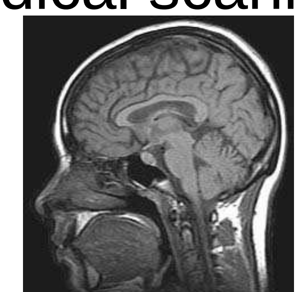
(6) Learning agents

- they have the ability to improve performance through learning.

Introduction to Computer Vision

What is Computer Vision?

- **Computer vision** is the science and technology of machines that see.
- Concerned with the theory for building artificial systems that obtain information from images.
- The image data can take many forms, such as a video sequence, depth images, views from multiple cameras, or multi-dimensional data from a medical scanner



Computer Vision

Make computers understand images and videos.

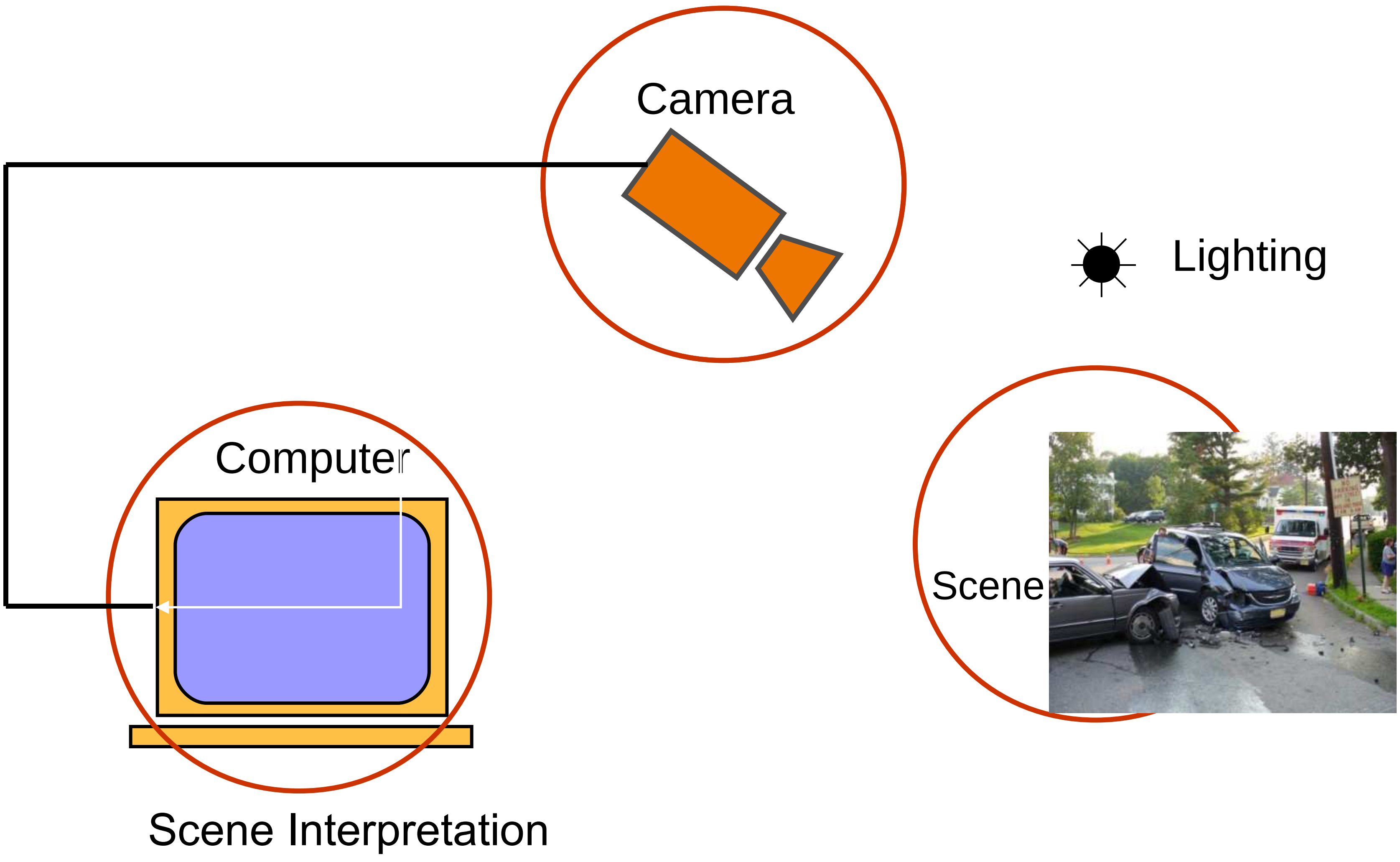
What kind of scene?

Where are the cars?

How far is the building?

…

Components of a computer vision system



Computer vision vs human vision

What we see

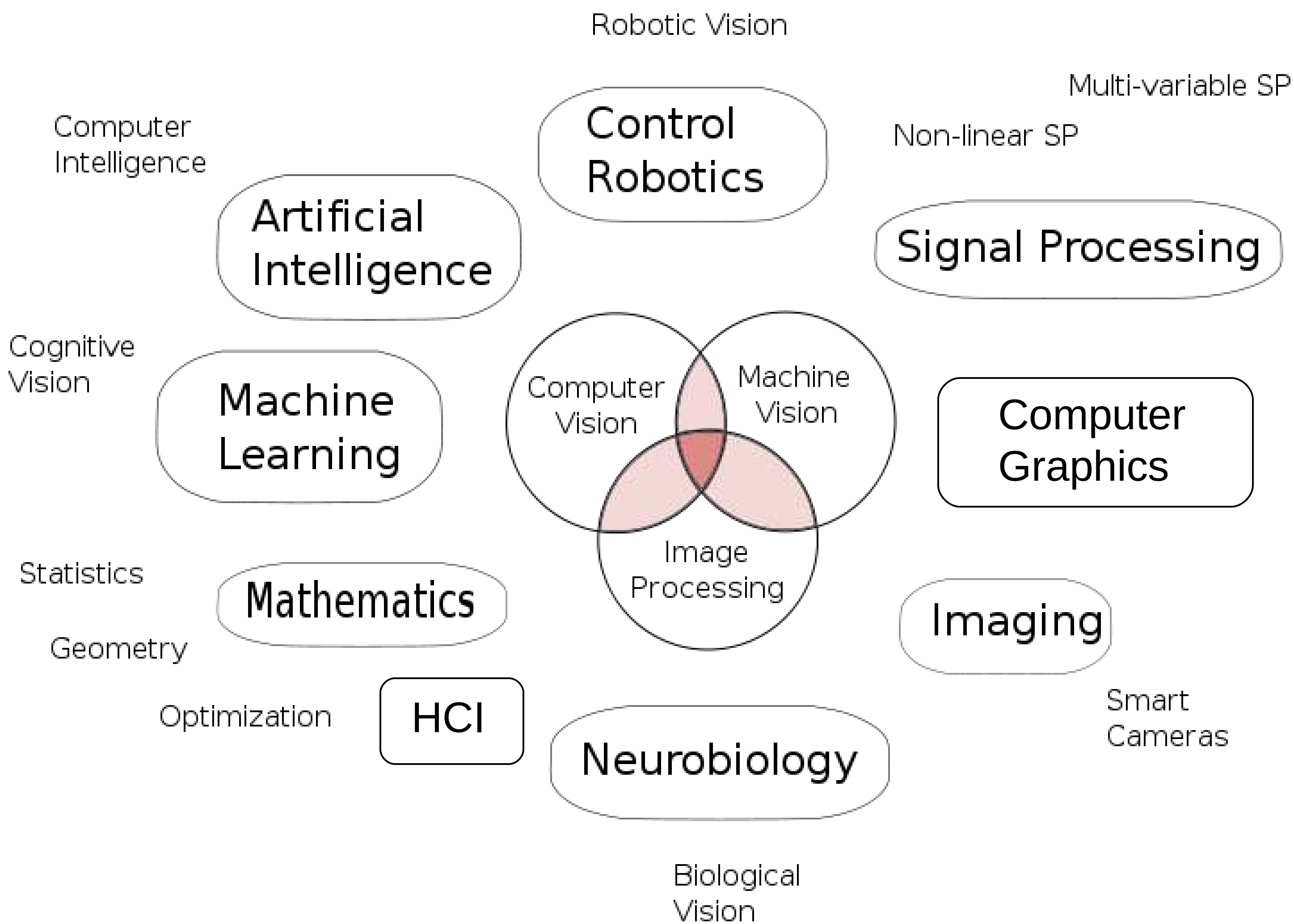
0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

Vision is really hard

- Vision is an amazing feat of natural intelligence
  - Visual cortex occupies about 50% of Macaque brain
  - More human brain devoted to vision than anything else

Vision is multidisciplinary



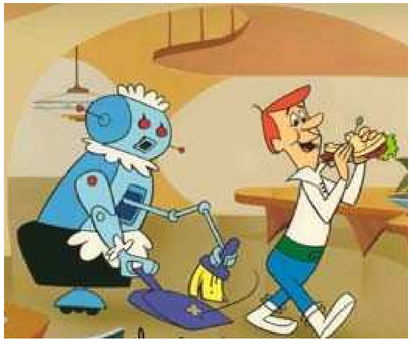
From wiki

Why computer vision matters

Safety

Health

Security



Comfort

Fun

Access

A little story about Computer Vision

In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to describe what it saw”. We now know that the problem is slightly more difficult than that. (Szeliski 2009, Computer Vision)

A little story about Computer Vision

Founder, MIT AI project

In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to describe what it saw”. We now know that the problem is slightly more difficult than that.

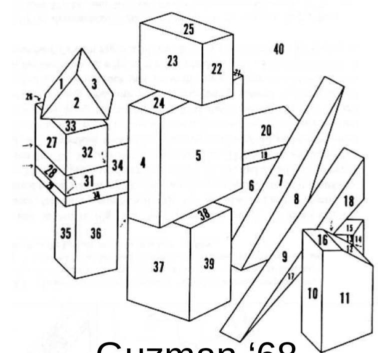
A little story about Computer Vision

In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to "spend the summer linking a camera to a computer and getting the computer to describe what it saw". We now know that the problem is slightly more difficult than that.

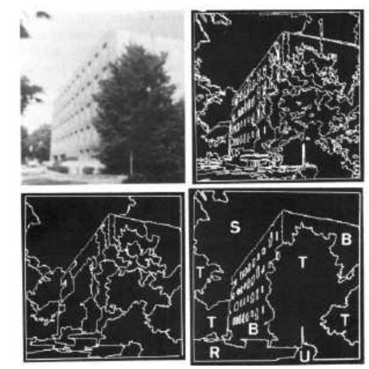
Image Understanding

Ridiculously brief history of computer vision

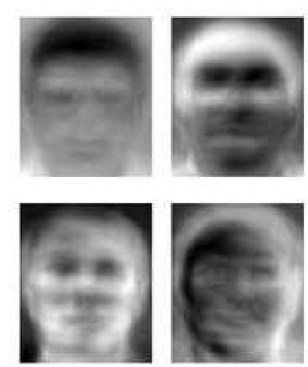
- 1966: Minsky assigns computer vision as an undergrad summer project
- 1960's: interpretation of synthetic worlds
- 1970's: some progress on interpreting selected images
- 1980's: ANNs come and go; shift toward geometry and increased mathematical rigor
- 1990's: face recognition; statistical analysis in vogue
- 2000's: broader recognition; large annotated datasets available; video processing starts; vision & graphis; vision for HCI; internet vision, etc.



Guzman '68



Ohta Kanade '78



Turk and Pentland '91

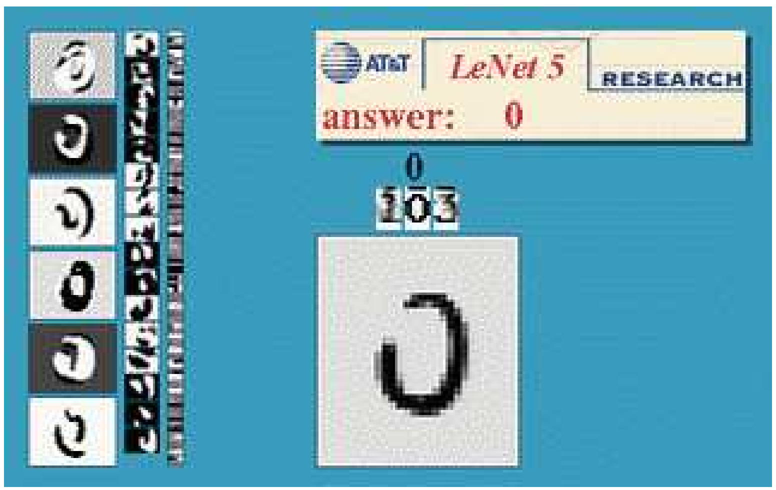
How vision is used now

- Examples of state-of-the-art

Optical character recognition (OCR)

Technology to convert scanned docs to text

- If you have a scanner, it probably came with OCR software



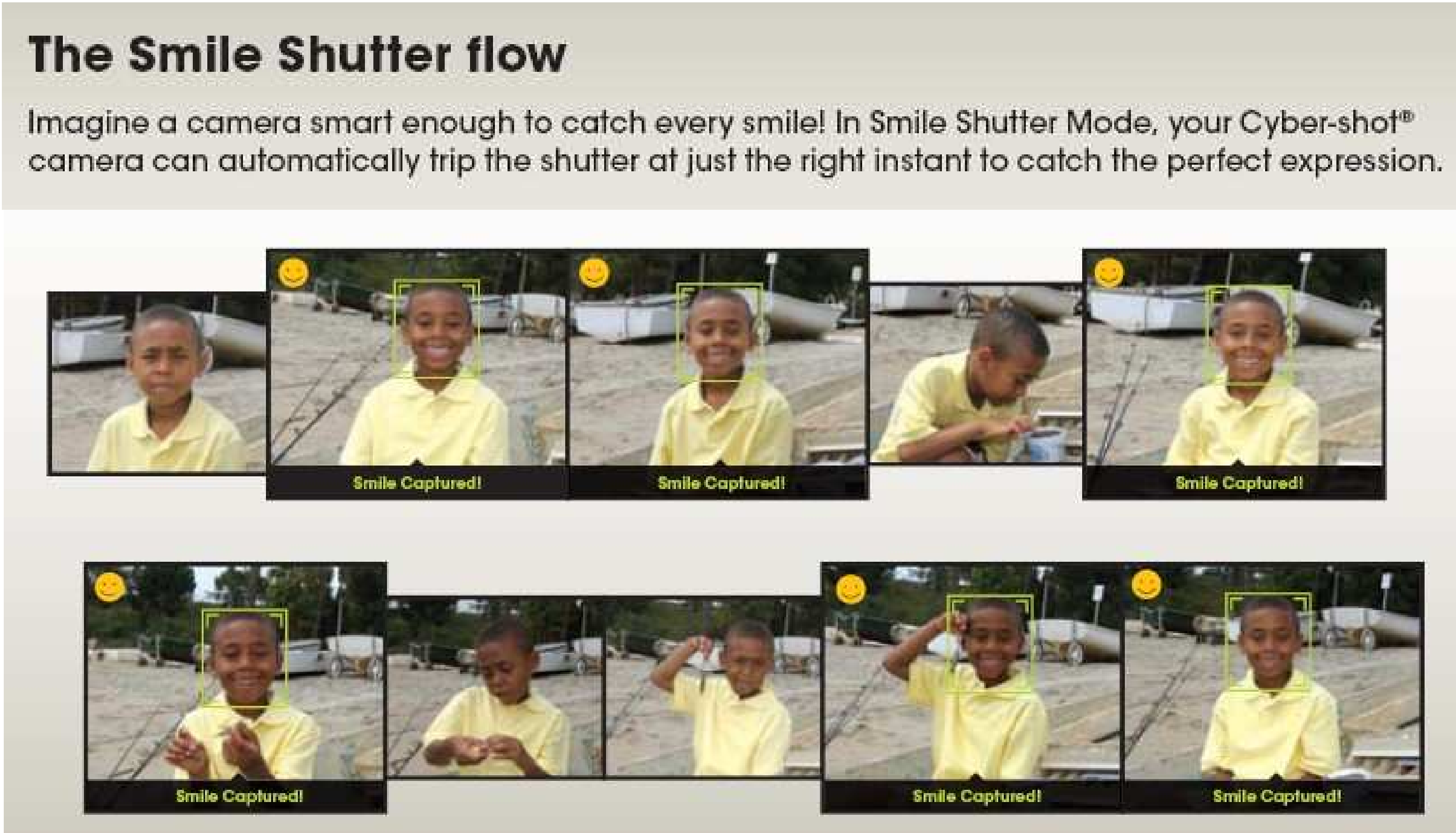
Digit recognition, AT&T labs
http://www.research.att.com/~yann/

License plate readers
http://en.wikipedia.org/wiki/Automatic_number_plate_recognition

Face detection

- Many new digital cameras now detect faces
  - Canon, Sony, Fuji, …

Smile detection



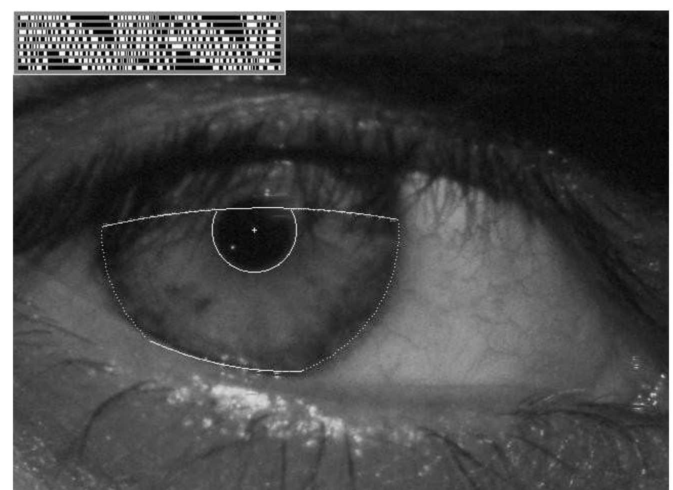
Sony Cyber-shot® T70 Digital Still Camera

Object recognition (in supermarkets)

LaneHawk by EvolutionRobotics
"A smart camera is flush-mounted in the checkout lane, continuously watching for items. When an item is detected and recognized, the cashier verifies the quantity of items that were found under the basket, and continues to close the transaction. The item can remain under the basket, and with LaneHawk,you are assured to get paid for it… "

Vision-based biometrics

"*How the Afghan Girl was Identified by Her Iris Patterns*" Read the story
wikipedia



Login without a password…

Fingerprint scanners on
many new laptops,
other devices

Face recognition systems now
beginning to appear more widely
http://www.sensiblevision.com/

Object recognition (in mobile phones)

Point & Find, Nokia

Google Goggles

Special effects: shape capture

The Matrix movies, ESC Entertainment, XYZRGB, NRC

Special effects: motion capture

Pirates of the Carribean, Industrial Light and Magic

Sports

Sportvision first down line
Nice explanation on www.howstuffworks.com

http://www.sportvision.com/video.html

Smart cars

Slide content courtesy of Amnon Shashua

- Mobileye [wiki article]
  - Vision systems currently in high-end BMW, GM, Volvo models
  - By 2010: 70% of car manufacturers.

Google cars

http://www.nytimes.com/2010/10/10/science/10google.html?ref=artificialintelligence

Vision-based interaction (and games)

Digimask: put your face on a 3D avatar.

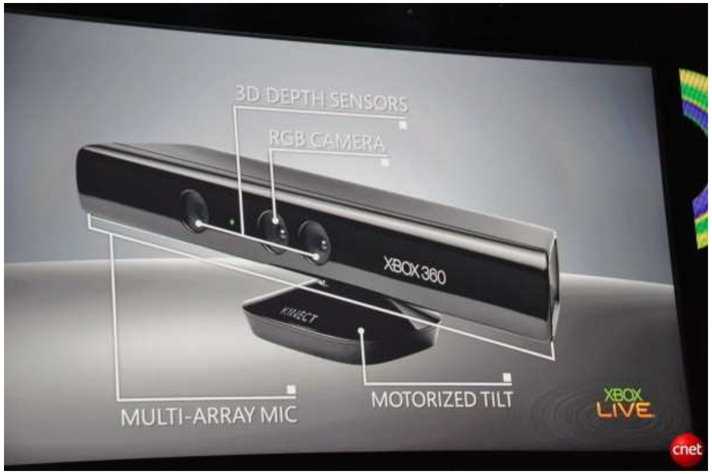
as camera-based IR
. See Lee's work atCMU on clever tricks on using it to
ouch display!

"Game turns moviegoers into Human Joysticks", CNET
Camera tracking a crowd, based on this work.

Interactive Games: Kinect

- Object Recognition: http://www.youtube.com/watch?feature=iv&v=fQ59dXOo63o
- Mario: http://www.youtube.com/watch?v=8CTJL5lUjHg
- 3D: http://www.youtube.com/watch?v=7QrnwoO1-8A
- Robot: http://www.youtube.com/watch?v=w8BmgtMKFbY
- 3D tracking, reconstruction, and interaction:

  http://research.microsoft.com/en-us/projects/surfacerecon/default.aspx



Vision in space

NASA'S Mars Exploration Rover Spirit captured this westward view from atop a low plateau where Spirit spent the closing months of 2007.

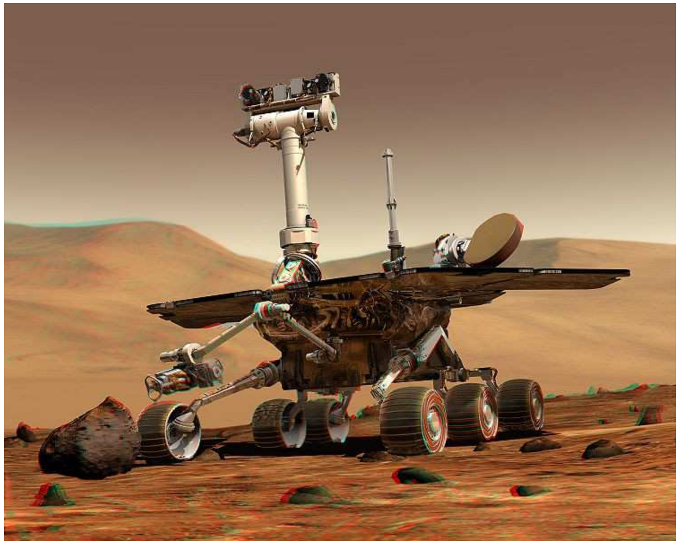
Vision systems (JPL) used for several tasks

- Panorama stitching
- 3D terrain modeling
- Obstacle detection, position tracking
- For more, read “Computer Vision on Mars” by Matthies et al.

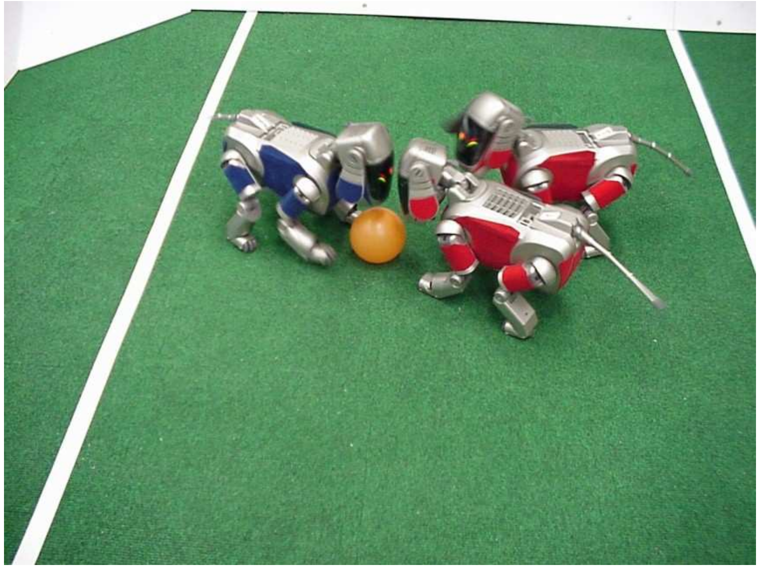
Industrial robots

Vision-guided robots position nut runners on wheels

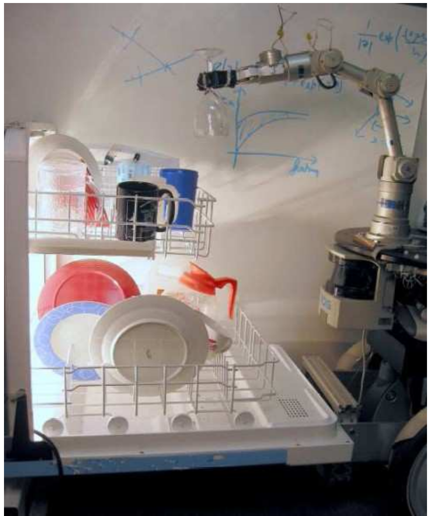
Mobile robots



NASA's Mars Spirit Rover
http://en.wikipedia.org/wiki/Spirit_rover

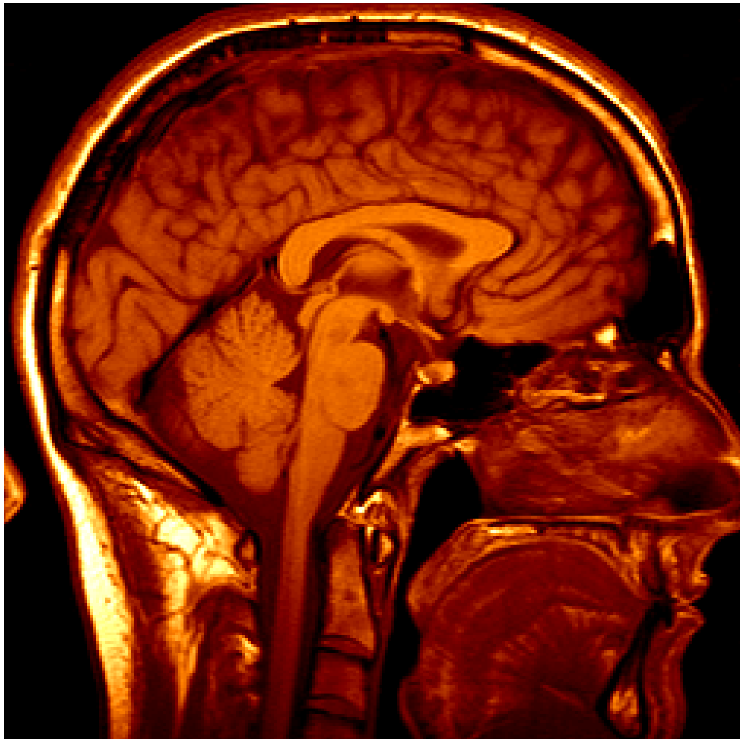


http://www.robocup.org/



Saxena et al. 2008
STAIR at Stanford

Medical imaging



3D imaging
MRI, CT

Image guided surgery
Grimson et al., MIT

Natural Language Processing

Artificial Intelligence

Natural Language Processing, topics : Introduction, definition, formal language, linguistic and language processing, terms related to linguistic analysis, grammatical structure of utterances - sentence, constituents, phrases, classifications and structural rules; Syntactic Processing - context free grammar (CFG), terminal, non-terminal and start symbols, parser, Semantics and Pragmatics.

Natural Language Processing

Artificial Intelligence

Topics

(Lecture 41 , 1 hours)

Natural Language Processing

What is NLP ?

- NLP is Natural Language Processing.
  Natural languages are those spoken by people.

- NLP encompasses anything a computer needs to understand natural language (typed or spoken) and also generate the natural language.

- Natural Language Processing (NLP) is a subfield of Artificial intelligence and linguistic, devoted to make computers "understand" statements written in human languages.

1. Introduction

Natural Language Processing (NLP) is a subfield of artificial intelligence and linguistic, devoted to make computers "understand" statements written in human languages.

1.1 Natural Language

A natural language (or ordinary language) is a language that is spoken, written by humans for general-purpose communication.
Example : Hindi, English, French, and Chinese, etc.

A language is a system, a set of symbols and a set of rules (or grammar).

- The Symbols are combined to convey new information.
- The Rules govern the manipulation of symbols.

Natural Language Processing (NLP)

NLP encompasses anything a computer needs to understand natural language (typed or spoken) and also generate the natural language.

- **Natural Language Understanding (NLU) :**

  The NLU task is understanding and reasoning while the input is a natural language.

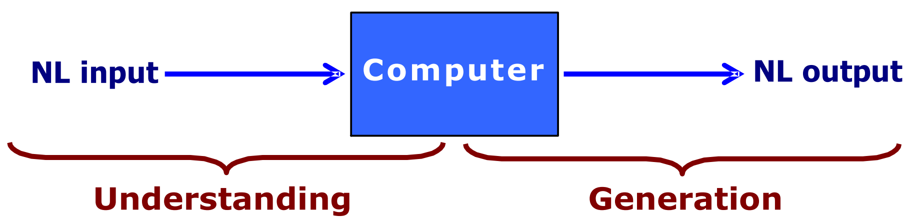
  Here we ignore the issues of natural language generation.

- **Natural Language Generation (NLG) :**

  NLG is a subfield of natural language processing NLP.

  NLG is also referred to text generation.

Natural Language Processing



1.2 Formal Language

Before defining formal language Language, we need to define symbols, alphabets, strings and words.

Symbol is a character, an abstract entity that has no meaning by itself.

e.g., Letters, digits and special characters

Alphabet is finite set of symbols;

an alphabet is often denoted by **Σ** (sigma)

e.g., **B = {0, 1}** says **B** is an alphabet of two symbols, **0** and **1**.

C = {a, b, c} says **C** is an alphabet of three symbols, **a, b** and **c**.

String or a word is a finite sequence of symbols from an alphabet.

e.g., **01110** and **111** are strings from the alphabet **B** above.

aaabccc and **b** are strings from the alphabet **C** above.

Language is a set of strings from an alphabet .

Formal language (or simply language) is a set **L** of strings over some finite alphabet Σ .

Formal language is described using formal grammars.

1.3 Linguistic and Language Processing

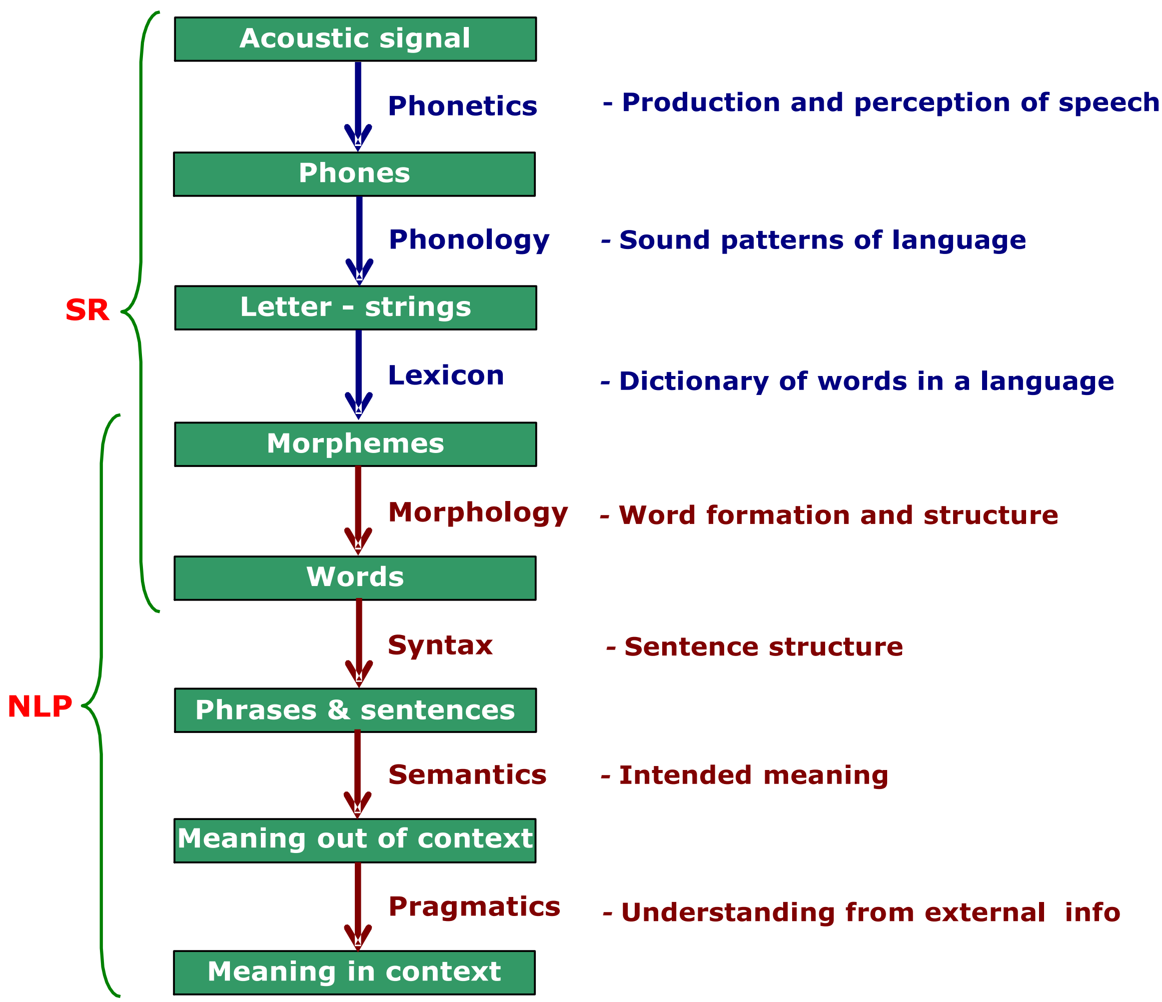
Linguistics is the science of language. Its study includes :

- sounds (phonology),
- word formation (morphology),
- sentence structure (syntax),
- meaning (semantics), and understanding (pragmatics) etc.

The levels of linguistic analysis are shown below.

- higher level corresponds to Speech Recognition (SR)
- lower levels corresponds to Natural Language Processing (NLP).

Levels Of Linguistic Analysis



- **Steps of Natural Language Processing (NLP)**

  Natural Language Processing is done at 5 levels, as shown in the previous slide. These levels are briefly stated below.

  - **Morphological and Lexical Analysis :**

    The lexicon of a language is its vocabulary, that include its words and expressions. Morphology is the identification, analysis and description of structure of words. The words are generally accepted as being the smallest units of syntax. The syntax refers to the rules and principles that govern the sentence structure of any individual language.

    Lexical analysis: The aim is to divide the text into paragraphs, sentences and words. the lexical analysis can not be performed in isolation from morphological and syntactic analysis

  - **Syntactic Analysis :**

    Here the analysis is of words in a sentence to know the grammatical structure of the sentence. The words are transformed into structures that show how the words relate to each others. Some word sequences may be rejected if they violate the rules of the language for how words may be combined.

    Example : An English syntactic analyzer would reject the sentence say :

    " Boy the go the to store ".

- **Semantic Analysis :**

  It derives an absolute (dictionary definition) meaning from context; it determines the possible meanings of a sentence in a context.

  The structures created by the syntactic analyzer are assigned meaning. Thus, a mapping is made between the syntactic structures and objects in the task domain. The structures for which no such mapping is possible are rejected.

  Example : the sentence **"Colorless green ideas . . . "** would be rejected as semantically anomalous because colorless and green make no sense.

- **Discourse Integration :**

  The meaning of an individual sentence may depend on the sentences that precede it and may influence the meaning of the sentences that follow it.

  Example : the word **" it "** in the sentence, **"you wanted it"** depends on the prior discourse context.

- **Pragmatic analysis :**

  It derives knowledge from external commonsense information; it means understanding the purposeful use of language in situations, particularly those aspects of language which require world knowledge; The idea is, what was said is reinterpreted to determine what was actually meant. Example : the sentence

  "Do you know what time it is ?"

  should be interpreted as a request.

1.4 Defining Terms related to Linguistic Analysis

The following terms are explained in next few slides.

Phones, Phonetics, Phonology, Strings, Lexicon, Words, Determiner, Morphology, Morphemes, Syntax, Semantics, Pragmatics, Phrase, and Sentence.

- **Terms**

  - **Phones**

    The Phones are acoustic patterns that are significant and distinguishable in some human language.

    Example : In English, the **L** - sounds at the beginning and end of the word **"loyal"**, are termed "light L" and "dark L" by linguists.

  - **Phonetics**

    Tells how acoustic signals are classified into phones.

  - **Phonology**

    Tells how phones are grouped together to form phonemes in particular human languages.

- **Strings**

  An alphabet is a finite set of symbols.

  Example : English alphabets

  { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z }

  A String is a sequence of symbols taken from an alphabet.

- **Lexicon**

  Lexicon is collection of information about words of a language.

  The information is about the lexical categories to which words belong.

  Example : **"bird"** is usually a noun **(N)**, but also occurs as a verb**(V)** and an adjective**(ADJ)**.

  Lexicon structure : as collection of lexical entries.

  Example : **("bird" N, V, ADJ)**

- **Words**

  Word is a unit of language that carries meaning.

  Example : words like bear, car, house are very different from
  words like run, sleep, think, and are different from
  words like in, under, about.

  These and other categories of words have names : nouns, verbs, prepositions, and so on.

  Words build phrases, which in turn build sentences.

- **Determiner**

  Determiners occur before nouns and indicate the kind of reference which the noun has.

  Example below shows determiners marked by "bold letters"

  the boy　　**a** bus　　**our** car

  these children　　**both** hospitals

- **Morphology**

  Morphology is the analysis of words into morphemes, and conversely the synthesis of words from morphemes.

- **Morphemes**

  A smallest meaningful unit in the grammar of a language.

  A smallest linguistic unit that has semantic meaning.

  A unit of language immediately below the 'word level'.

  A smallest part of a word that can carry a discrete meaning.

  Example : the word **"unbreakable"** has 3 morphemes:

1	"**un-**"	a bound morpheme;
2	"**-break-**"	a free morpheme; and
3	"**-able**"	a bound morpheme;

  Also **"un-"** is also a prefix; **"-able"** is a suffix; Both are affixes.

  Morphemes are of many types, stated in the next slide.

Types of Morphemes

‡ **Free Morphemes**

can appear stand alone, or "free" .

Example : **"town", "dog"** or with other lexemes **"town hall" , "dog house"**.

‡ **Bound Morphemes**

appear only together with other morphemes to form a lexeme.

Example : **"un-"** ; in general it tend to be prefix and suffix.

‡ **Inflectional Morphemes**

modify a word's tense, number, aspect, etc.

Example : **dog** morpheme with plural marker morpheme **s** becomes **dogs**.

‡ **Derivational Morphemes**

can be added to a word to derive another word.

Example : addition of **"-ness"** to **"happy"** gives "**happiness."**

‡ **Root Morpheme**

It is the primary lexical unit of a word; roots can be either free or bound morphemes; sometimes "root" is used to describe word minus its inflectional endings, but with its lexical endings.

Example : word **chatters** has the inflectional root or lemma **chatter**, but the lexical root **chat**.

Inflectional roots are often called stems, and a root in the stricter sense may be thought of as a mono-morphemic stem.

‡ **Null Morpheme**

It is an "invisible" affix, also called zero morpheme represented as either the figure zero **(0)**, the empty set symbol **Ø**, or its variant **ø**. Adding a null morpheme is called null affixation, null derivation or zero derivation; null morpheme that contrasts singular morpheme with the plural morpheme.

e.g., **cat = cat + -0 = ROOT("cat") + SINGULAR**
cats = cat + -s = ROOT("cat") + PLURAL

- **Syntax**

  Syntax is the structure of language. It is the grammatical arrangement of words in a sentence to show its relationship to one another in a sentence; Syntax is finite set of rules that specifies a language;

  Syntax rules govern proper sentence structure;

  Syntax is represented by Parse Tree, a way to show the structure of a language fragment, or by a list.

- **Semantics**

  Semantic is Meaning of words / phrases/ sentences/ whole texts.

  Normally semantic is restricted to "meaning out of context" - that is, meaning as it can be determined without taking context into account.

- **Pragmatics**

  Pragmatics tell how language is used; that is ‘meaning in context’.

  Example: if someone says **"the door is open"** then it is necessary to know which door **"the door"** refers to;

  Need to know what the intention of the speaker :

  could be a pure statement of fact,

  could be an explanation of how the cat got in, or

  could be a request to the person addressed to close the door.

1.5 Grammatical Structure of Utterances

Here sentence, constituent, phrase, classification and structural rule are explained.

- **Sentence**

  Sentence is a string of words satisfying grammatical rules of a language;

  Sentences are classified as simple, compound, and complex.

  Sentence is often abbreviated to "***S***".

  Sentence (S) : "The dog bites the cat".

- **Constituents**

  Assume that a phrase is a construction of some kind.

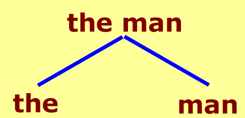
  Here construction means a syntactic arrangement that consists of parts, usually two, called "constituents".

  Examples : The phrase **the man** is a construction consists of two constituents **the** and **man**. A few more examples are shown below.

Phrase : the man

Constituents : the and man

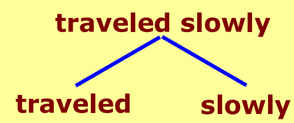
Construction :



Phrase : traveled slowly

Constituents : traveled and slowly.

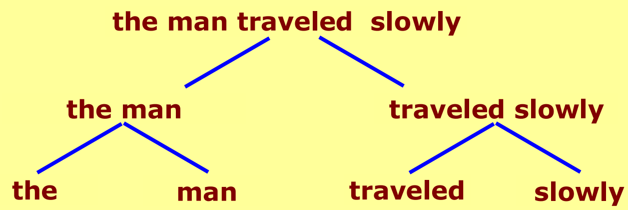
Construction :



Phrase : the man traveled slowly

Constituents four : the , man , traveled , slowly

Construction :



- **Phrase**

  A Phrase is a group of words (minimum is two) that functions as a single unit in the syntax of a sentence.

  e.g., 1: **"the house at the end of the street "** is a phrase, acts like noun.

  e.g., 2: **"end of the street "** is a phrase, acts like adjective;

  How phrases are formed is governed by phrase structure rules.

  Most phrases have a head or central word, which defines the type of phrase. Head is often the first word of the phrase. Some phrases, can be headless.

  e.g.,3: **"the rich"** is a noun phrase composed of a determiner and an adjective, but no noun.

  Phrases may be classified by the type of head they take.

  [Continued in next slide]

[Continued from previous slide]

Classification of Phrases : names (abbreviation)

The most accepted classifications for phrases are stated below.

‡ **Sentence (S) :** often abbreviated to "***S***".

‡ **Noun phrase (NP) :** noun or pronoun as head, or optionally accompanied by a set of modifiers; The possible modifiers include: determiners: articles (the, a) or adjectives (the red ball) etc ; example : "**the** black cat", "**a** cat on the mat".

‡ **Verb phrase (VP**) : verb as head, example : "**eat** cheese", "**jump** up and down".

‡ **Adjectival phrase (AP) :** adjective as head, example : "**full** of toys"

‡ **Adverbial phrase (AdvP) :** adverb as head, example : "**very carefully**"

‡ **Prepositional phrase (PP) :** preposition as head, example : "**in** love", "**over** the rainbow".

‡ **Determiner phrase (DP) :** determiner as head example : "**a** little dog", "**the** little dogs". In English, determiners are usually placed before the noun as a noun modifier that includes : articles (the, a), demonstratives (this, that), numerals (two, five, etc.), possessives (my, their, etc.), and quantifiers (some, many, etc.).

- **Phrase Structure Rules**

Phrase-structure rules are a way to describe language syntax. Rules determine what goes into phrase and how its constituents are ordered. They are used to break a sentence down to its constituent parts namely phrasal categories and lexical categories.

- Phrasal category include : noun phrase, verb phrase, prepositional phrase;
- Lexical category include : noun, verb, adjective, adverb, others.

Phrase structure rules are usually of the form **B C** ,

Meaning "constituent **A** is separated into two sub-constituents **B** and **C** " or simply " **A** consists of **B** followed by **C** " .

Examples :

- **S** **NP VP** Reads : **S** consists of an **NP** followed by a **VP** ; means a sentence consists of a noun phrase followed by a verb phrase.
- **NP** **Det N1** Reads : **NP** consists of an **Det** followed by a **N1** ; means a noun phrase consists of a determiner followed by a noun.

Phrase Structure Rules and Trees for Noun Phrase (NP)

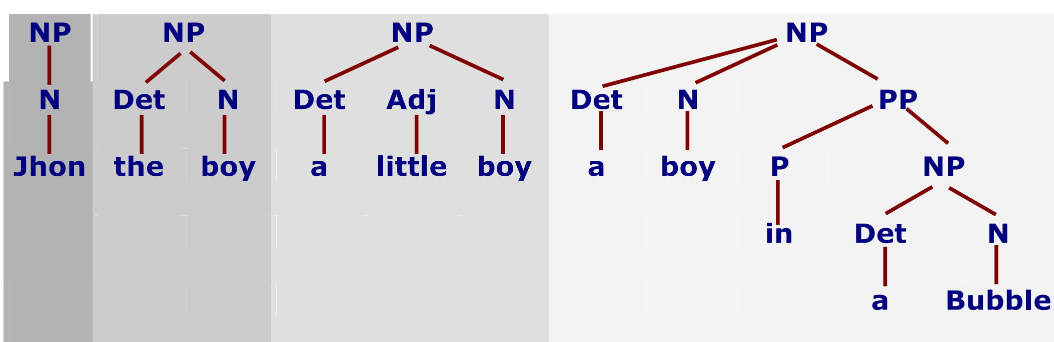
Noun Phrase (NP)

John	N
the boy	Det N
A little boy	Det Adj N
A boy in a bubble	Det N PP

Phrase Structure rules for NPs

NP (Det) (Adj) N (PP)

Phrase Structure trees for NPs



2. Syntactic Processing

Syntactic Processing converts a flat input sentence into a hierarchical structure that corresponds to the units of meaning in the sentence.

The Syntactic processing has two main components :

- one is called grammar, and
- other is called parser.

‡ **Grammar** :

It is a declarative representation of syntactic facts about the language.

It is the specification of the legal structures of a language.

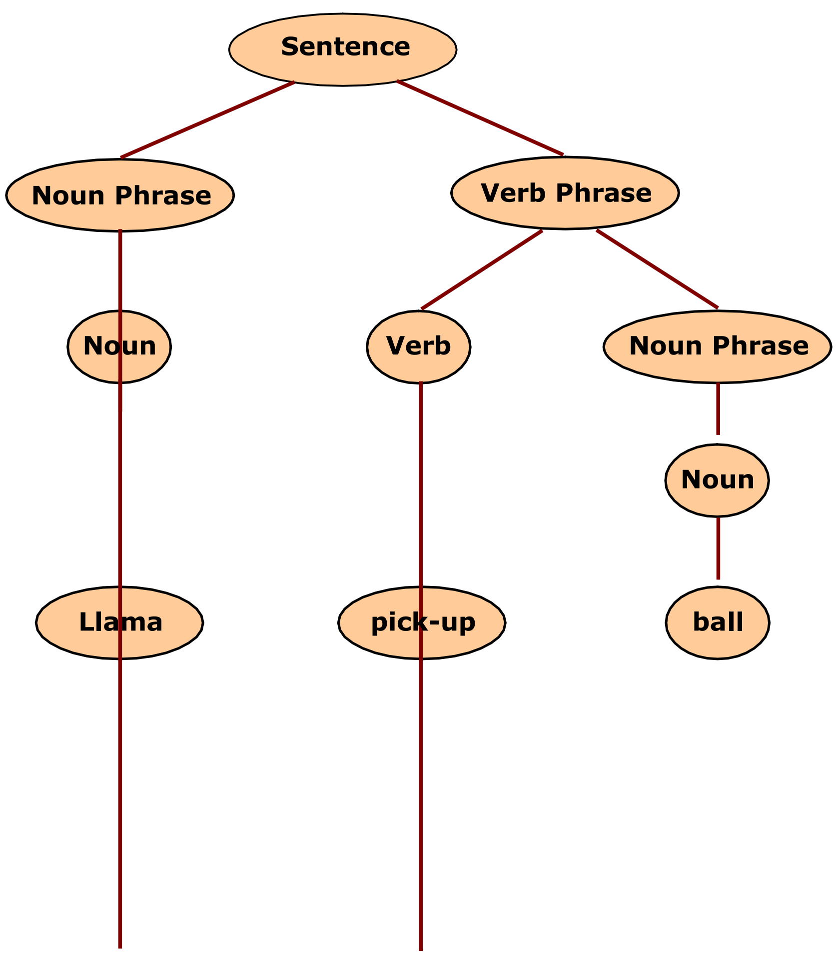
It has three basic components : terminal symbols, non-terminal symbols, and rules (productions) .

‡ **Parser** :

It is a procedure that compares the grammar against input sentences to produce a parsed structures called **parse tree**.

Example 1 : Sentence **"Llama pickup ball"**.

Parse Tree Structure (PS)



2.1 Context Free Grammar (CFG)

In formal language theory, a context free grammar is a grammar where every production rules is of the form: where is a single symbol called **non-terminal** , and is a **string** that is a sequence of symbols of terminals and/or non-terminals (possibly empty).

Note : The difference with an arbitrary grammars is that the left hand side of a production rule is always a single nonterminal symbol rather than a string of terminal and/or nonterminal symbols.

- **Terminal , Non-Terminal and Start Symbols**

  The terminal and non-terminal symbols are those symbols that are used to construct production rules in a formal grammar.

  ‡ **Terminal Symbol**

  Any symbol used in the grammar which does not appear on the left-hand-side of some rule (ie. has no definition) is called a terminal symbol. Terminal symbols cannot be broken down into smaller units without losing their literal meaning.

  ‡ **Non-Terminal Symbol**

  Symbols that are defined by rules are called non-terminal symbol. Each production rule defines the non-terminal symbol. Like the above rule states that "whenever we see an , we can replace it with ".

  ‡ A non-terminal may have more than one definition, in that case we use symbol "|" as the union operator;

  Example 1: | states that "whenever we see , we can replace it with or with ''.

  Similarly, if a rule is **NP** **Det N** | **Prop** then the vertical slash on the right side is a convention used to represent that the **NP** can be replaced either by **Det N** or by **Prop**. Thus, this is really two rules.

  Example 2: **S** **NP VP** states that the symbol **S** is replaced by the symbols **NP** and **VP**.

  ‡ One special non-terminal is called **Start symbol**, usually written **S**. The production rules for this symbol are usually written first in a grammar.

- **How Grammar works ?**

Grammar starts with the start symbol, then successively applies the production rules (replacing the L.H.S. with the R.H.S.) until reaches to a word which contains no non-terminals. This is known as a derivation.

‡ Anything which can be derived from the start symbol by applying the production rules is called a sentential form.

‡ Any grammar may have an infinite number of sentences;
The set of all such sentences is the language defined by that grammar.

‡ Example of grammar :

S X c X Y X Y a | b

The above grammar shows that it can derive all words which start arbitrarily and have many '**a**'s or '**b**'s and finish with a '**c**'. This language is defined by the regular expression **(a | b) * c**. The " * " indicates that the character immediately to its left may be repeated any number of times, including zero. Thus **ab*c** would match "**ac**", **"abc", "abbc", "abbbc", "abbbbbbbc"**, and any string that starts with an "**a**", is followed by a sequence of "**b**"'s, and ends with a "**c**".

‡ **Regular Expression**

Every regular expression can be converted to a grammar, but not every grammar can be converted back to a regular expression;
Any grammar which can be converted back to a regular expression is called a regular grammar; the language it defines is a regular language.

Regular Expression	**Grammar**
Regular Expression ←	**Regular Grammar**

‡ **Regular Grammars**

A regular grammar is a grammar where all of the production rules are of one of the following forms:

a or **a**

where and represent any single non-terminal, and

a represents any single terminal, or the empty string.

2.2 Parsar

A parser is a program, that accepts as input a sequence of words in a natural language and breaks them up into parts (nouns, verbs, and their attributes), to be managed by other programming.

- Parsing can be defined as the act of analyzing the grammaticality an utterance according to some specific grammar.
- Parsing is the process to check, that a particular sequence of words in a sentence correspond to a language defined by its grammar.
- Parsing means show how we can get from the start symbol of the grammar to the sequence of words using the production rules.
- The output of a parser is *a Parse tree.*

Parse Tree is a way of representing the output of a parser.

- Each phrasal constituent found during parsing becomes a branch node of the parse tree;
- the words of the sentence become the leaves of the parse tree;
- there can be more than one parse tree for a single sentence;

- **Parsing**

  To parse a sentence, it is necessary to find a way in which the sentence could have been generated from the start symbol. There two ways to do : One, Top-Down Parsing and the other, Bottom–UP Parsing.

  - **Top-Down Parsing**

    Begin with the start symbol and apply the grammar rules forward until the symbols at the terminals of the tree corresponds to the components of the sentence being parsed.

  - **Bottom–UP Parsing**

    Begin with the sentence to be parsed and apply the grammar rules backward until a single tree whose terminals are the wards of the sentence and whose top node is the start symbol has been produced.

  Note : The choice between these two approaches is similar to the choice between forward and backward reasoning in other problem solving tasks. The most important consideration is the branching factors. Some times these two approaches are combined in to a single method called bottom–up parsing with top-down filtering.

- **Modeling a Sentence using Phase Structure**

  Every sentence consists of an internal structure which could be modeled with the phrase structure.

  Algorithm : Steps

  ‡ Apply rules on an proposition

  ‡ The base proposition would be :

  S (the root, ie the sentence).

  ‡ The first production rule would be :

  (NP = noun phrase, VP = verb phrase)

  S -> (NP, VP)

  ‡ Apply rules for the 'branches'

  NP -> noun VP -> verb, NP

  ‡ The verb and noun have terminal nodes which could be any word in the lexicon for the appropriate category.

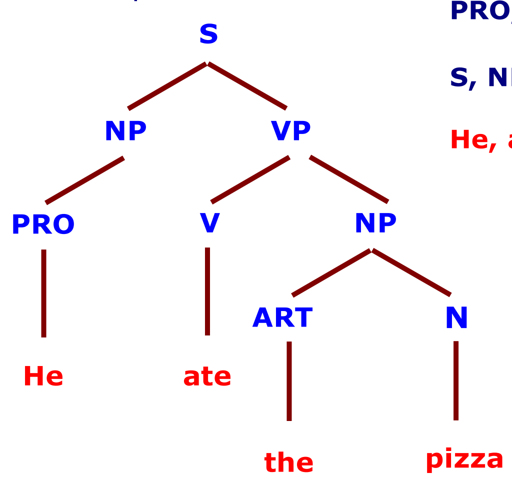
  ‡ The end is a tree with the words as terminal nodes, which is referred as the sentence.

Example : Parse tree

- sentence "He ate the pizza",
- apply the grammar with rules

  S -> NP VP, NP -> PRO, NP -> ART N, VP -> V NP,
- the lexicon structure is

  ("ate" V) ("he" PRO) ("pizza" N) ("the" ART)
- The parse tree is



PRO, V, ART, N - lexical non-terminals

S, NP, VP - phrasal non-terminal

He, ate, the, pizza - words or terminal

3. Semantics and Pragmatics

The semantics and pragmatics, are the two stages of analysis concerned with getting at the meaning of a sentence.

- In the first stage (semantics) a partial representation of the meaning is obtained based on the possible syntactic structure(s) of the sentence and the meanings of the words in that sentence.
- In the second stage (pragmatic), the meaning is elaborated based on : the contextual and the world knowledge.

For the difference between these stages, consider the sentence:

"He asked for the boss".

From knowledge of the meaning of the words and the structure of the sentence we can work out that :

- Someone (who is male) asked for someone who is a boss.
- We can't say who these people are and why the first guy wanted the second.
- If we know something about the context (including the last few sentences spoken/written) we may be able to work these things out.
- Maybe the last sentence was **"Fred had just been sacked.''**
- From our general knowledge that bosses generally sack people : if people want to speak to people who sack them it is generally to complain about it.
- We could then really start to get at the meaning of the sentence : **"Fred wants to complain to his boss about getting sacked".**

4. References : Textbooks

1. *"Artificial Intelligence", by Elaine Rich and Kevin Knight, (2006), McGraw Hill companies Inc., Chapter 15, page 377-426.*

2. *"Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig, (2002), Prentice Hall, Chapter 23, page 834-861.*

3. *"Artificial Intelligence: Structures and Strategies for Complex Problem Solving", by George F. Luger, (2002), Addison-Wesley, Chapter 15, page 619-632.*

4. *"Artificial Intelligence: Theory and Practice", by Thomas Dean, (1994), Addison-Wesley, Chapter 10, Page 489-538.*

5. *Related documents from open source, mainly internet. An exhaustive list is being prepared for inclusion at a later date.*