ARTIFICIAL INTELLIGENCE

17BTCS701

Prepared by: Dr. Nilima Kulkarni, MIT SOE.

Acknowledgement

This Ppt’s are prepared using

1. Stuart Russell and Peter Norvig (1995), “Articial Intelligence: A Modern Approach”, Third edition, Pearson, 2003.

2. Parag Kulkarni and Prachi Joshi, “Artificial Intelligence Building Intelligent Systems”, PHI learning Pvt. Ltd., ISBN 978-81-203-5046-5, 2015. 5.

3. Online source

ARTIFICIAL INTELLIGENCE

In computer science, **artificial intelligence** (**AI**), sometimes called **machine intelligence**, is intelligence demonstrated by machines, in contrast to the **natural intelligence** displayed by humans.

Leading AI textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals

It is the study of how to train the computers so that computers can do things which at present human can do better."

ARTIFICIAL INTELLIGENCE

AI can be defined in different ways

1. Machine which can think and have a capability to react like human beings
2. System that respond intelligently in the same way as the human do
3. Computational modals to solve various complex decision making problems
4. Study of intelligent agents

ARTIFICIAL INTELLIGENCE

AI stands for Artificial intelligence, where intelligence is defined acquisition of knowledge intelligence is defined as a ability to acquire and apply knowledge.

The aim is to increase chance of success and not accuracy.

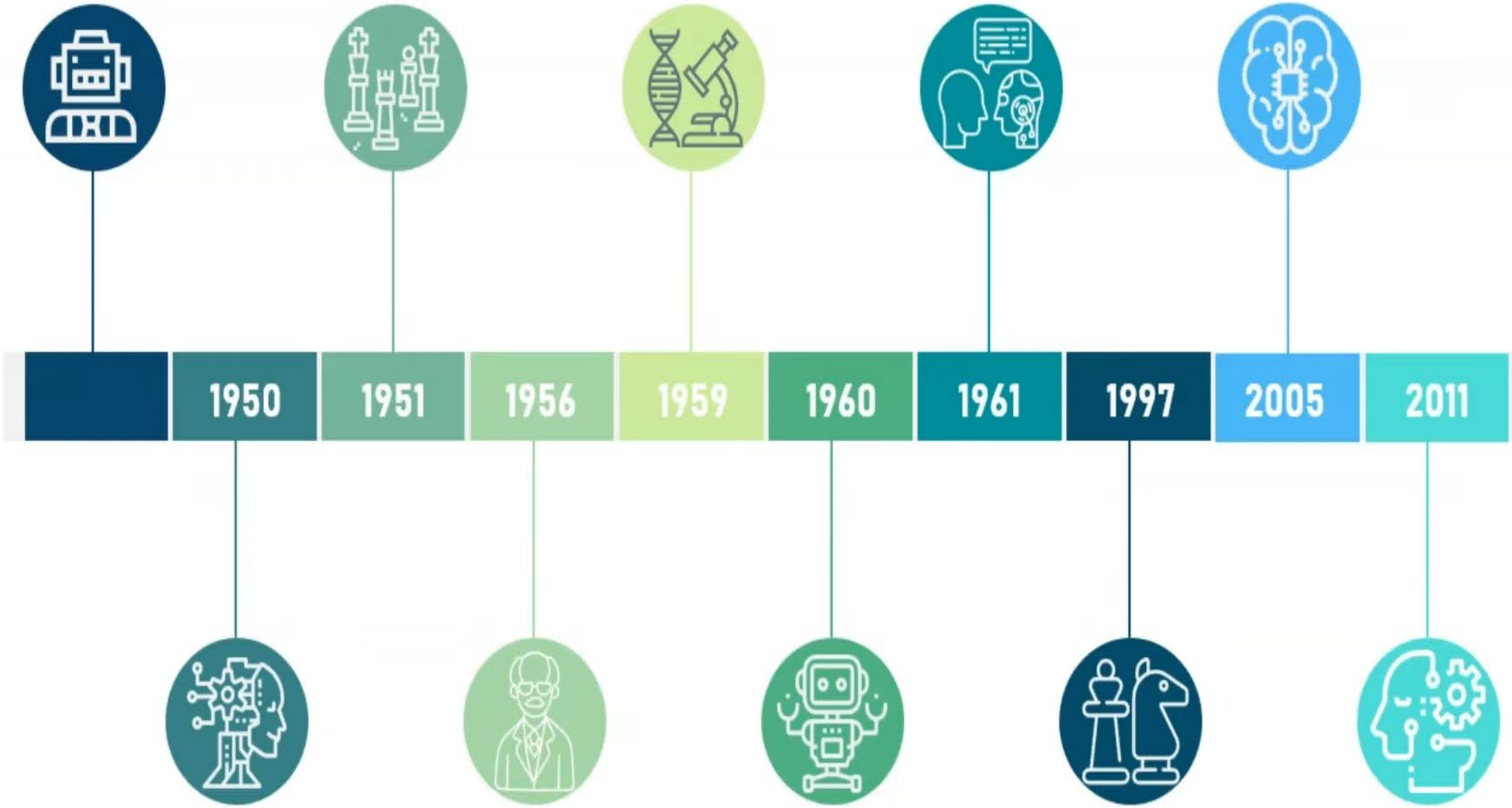
It work as a computer program that does smart work

The goal is to simulate natural intelligence to solve complex problem

AI is decision making.

01

HISTORY OF AI



2011 – IBM Watson

IBM's question answering system, *Watson*, defeated the two greatest Jeopardy! champions, Brad Rutter and Ken Jennings,.

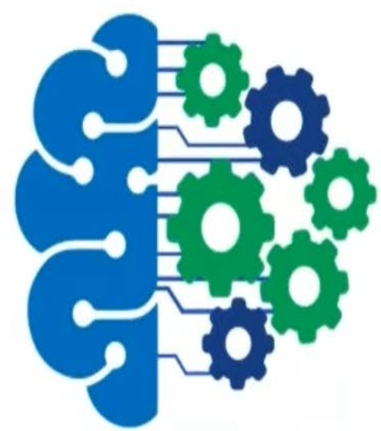
- Now?
- Now AI covers almost all the fields

Knowledge Base

Expert Systems

Deep Learning

Computer Vision



Machine Learning

Natural Language Processing

DEMAND FOR AI

More Computational Power

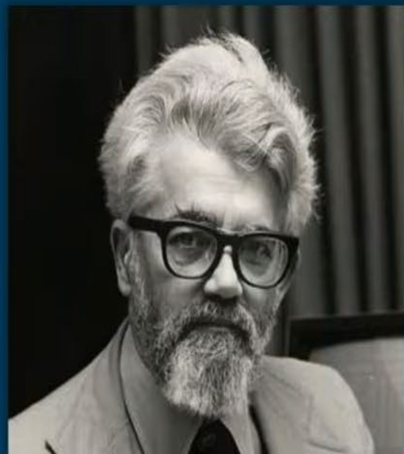
More Data

Better algorithms

Broad Investment

- What is AI

Birth Of AI



John McCarthy first coined the term Artificial Intelligence in the year 1956.

John McCarthy defined Artificial Intelligence as the science and engineering of making intelligent machines.

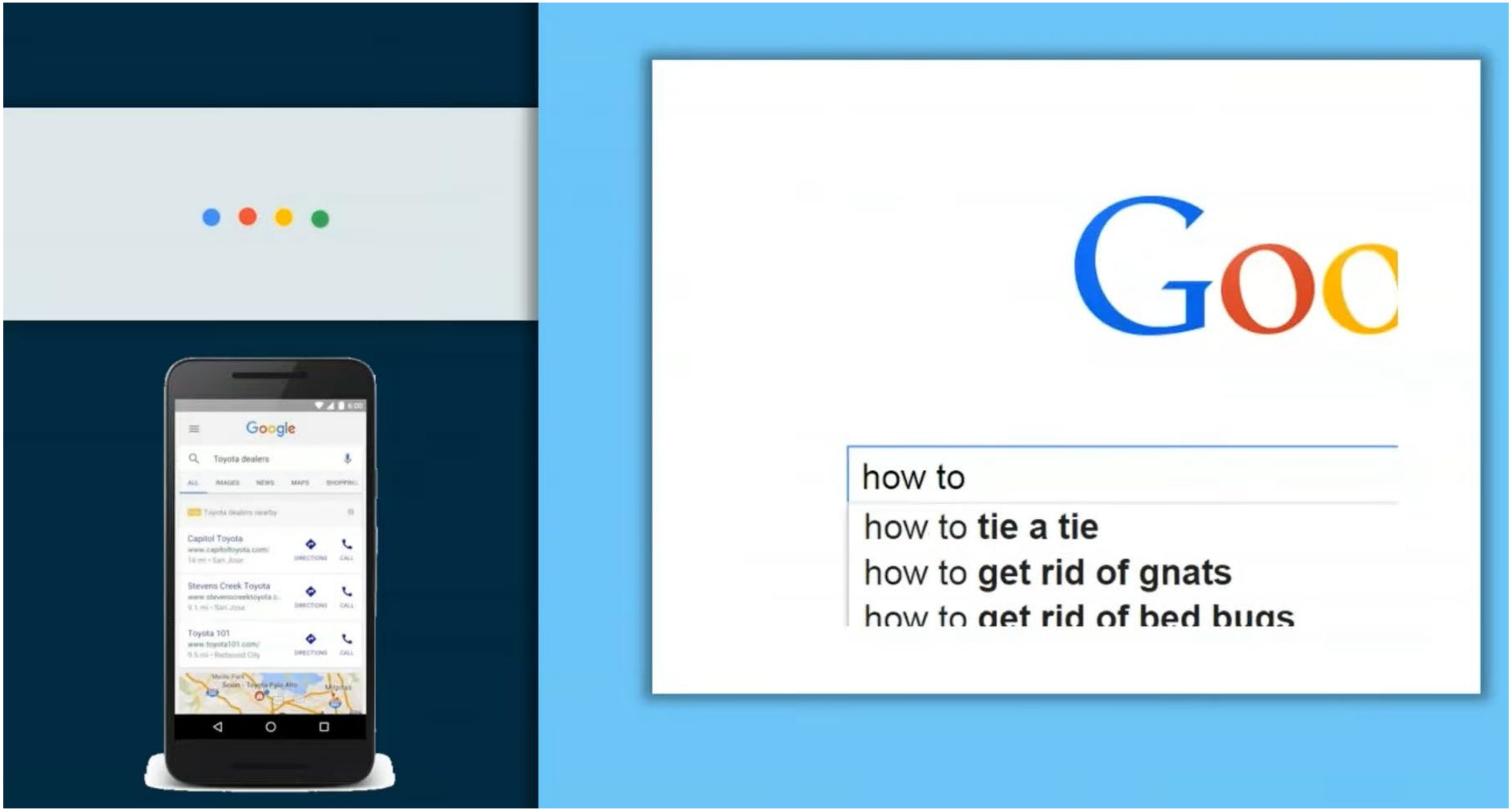
WHAT IS AI?

The theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making and translation between languages.

Where AI Used?

AI APPLICATIONS

Prepared by: Dr. Nilima Kulkarni, MIT SOE.



Google Recommendations:

Google gives recommendation (based on your age, gender, place and other details)

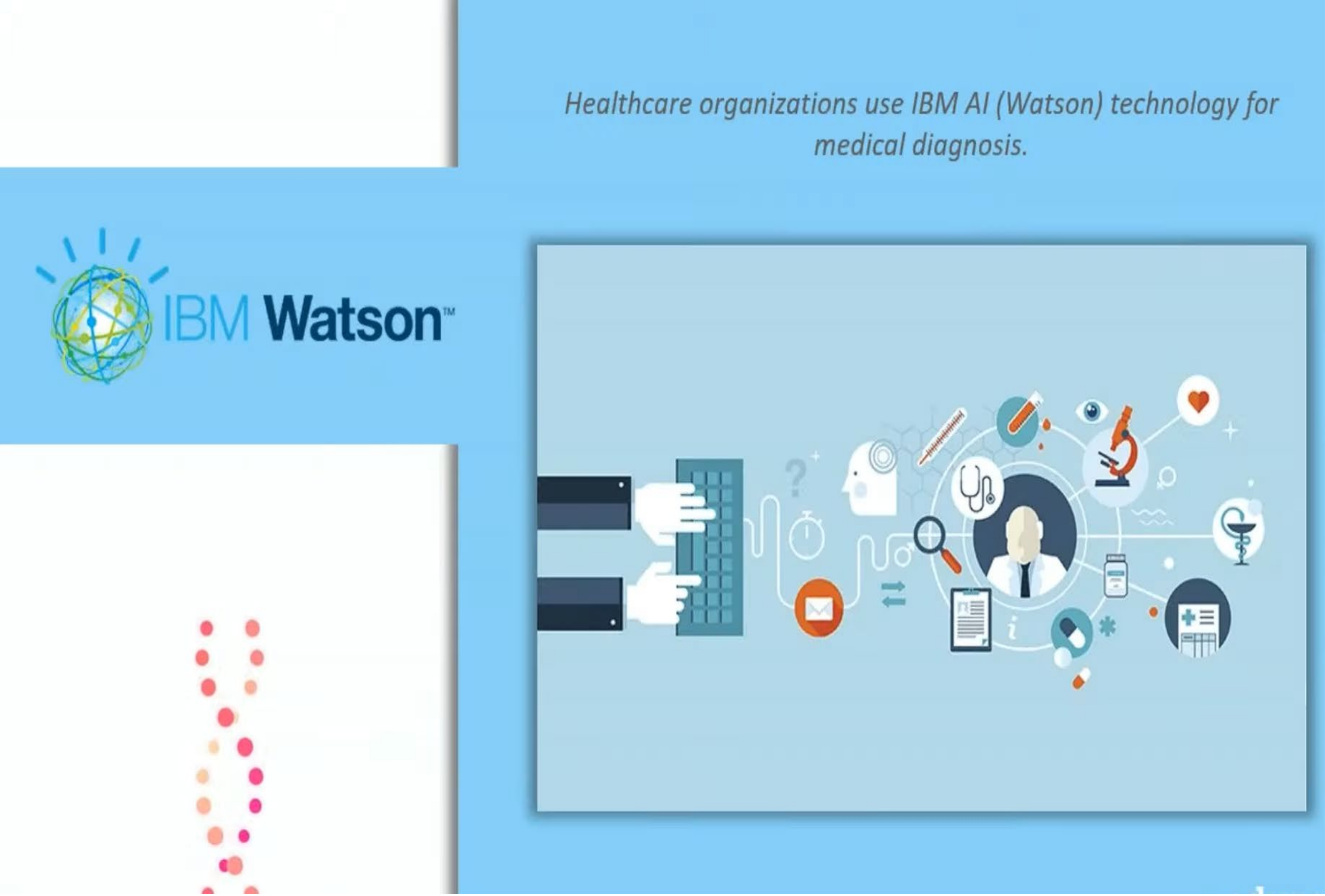
It uses NLP, ML, DL

J.P.Morgan

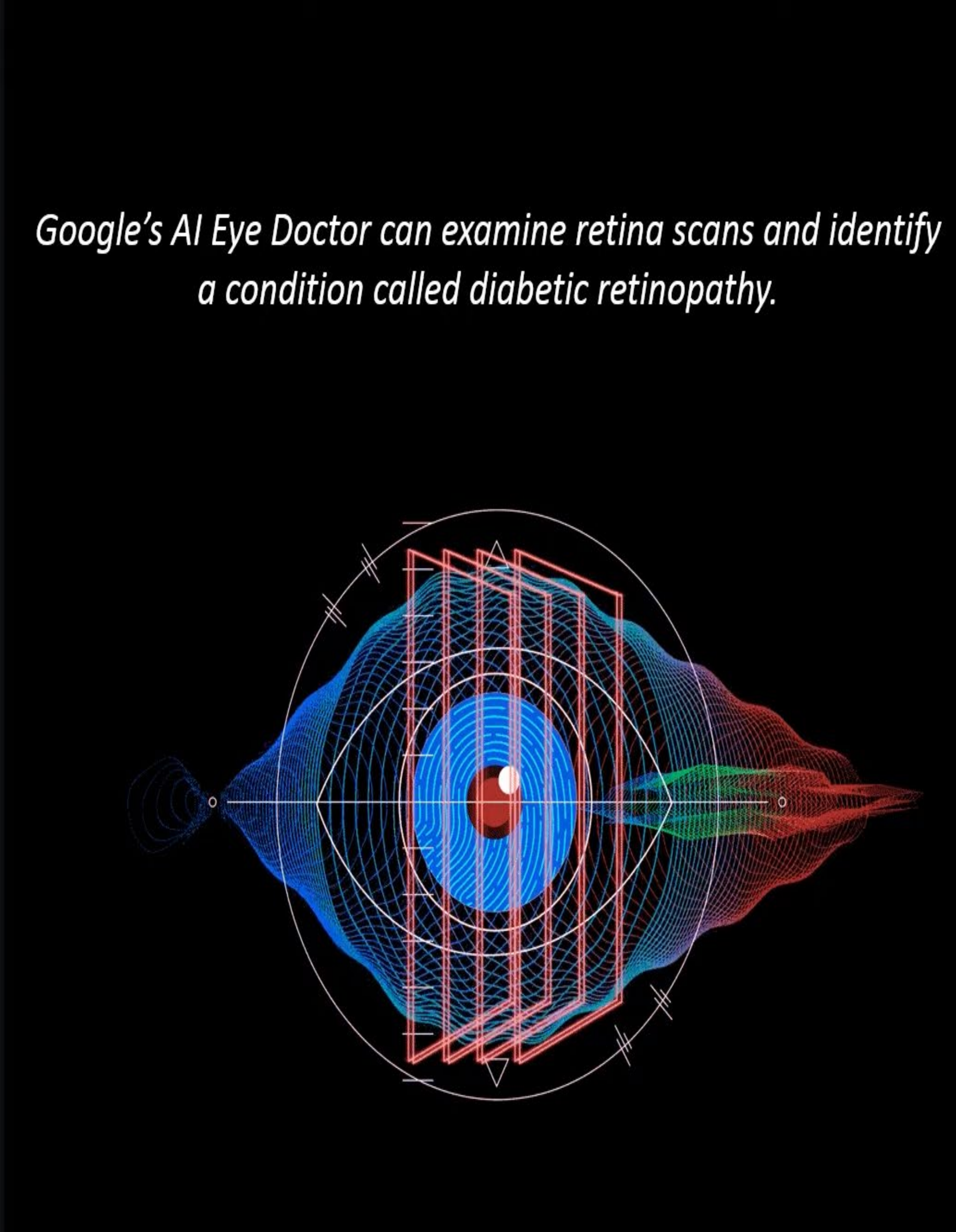
JPMorgan Chase's Contract Intelligence (COiN) platform uses AI, machine learning and image recognition software to analyze legal documents.

Manually reviewing 12,000 agreements takes over 36,000 hours

AI does this in Sec.



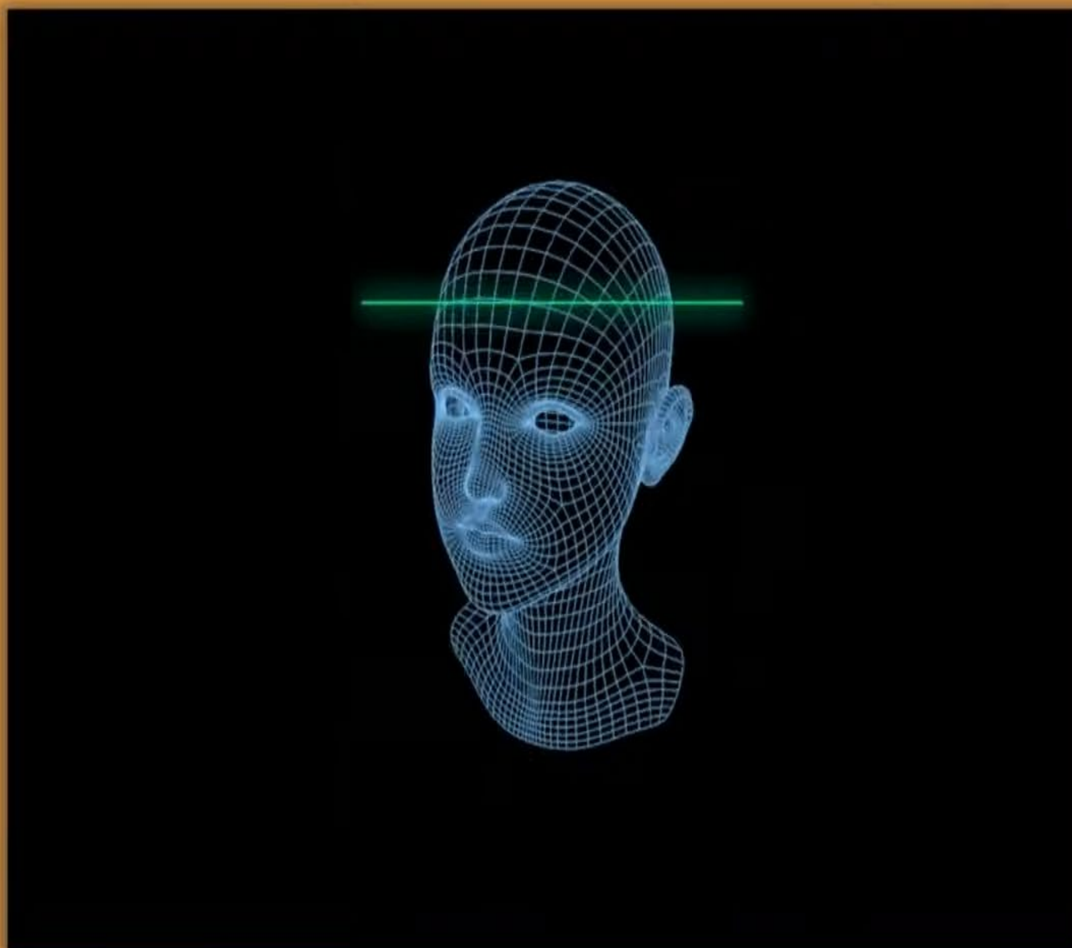
-Used by many
- Able to scan millions of medical record and diagnosis different diseases.



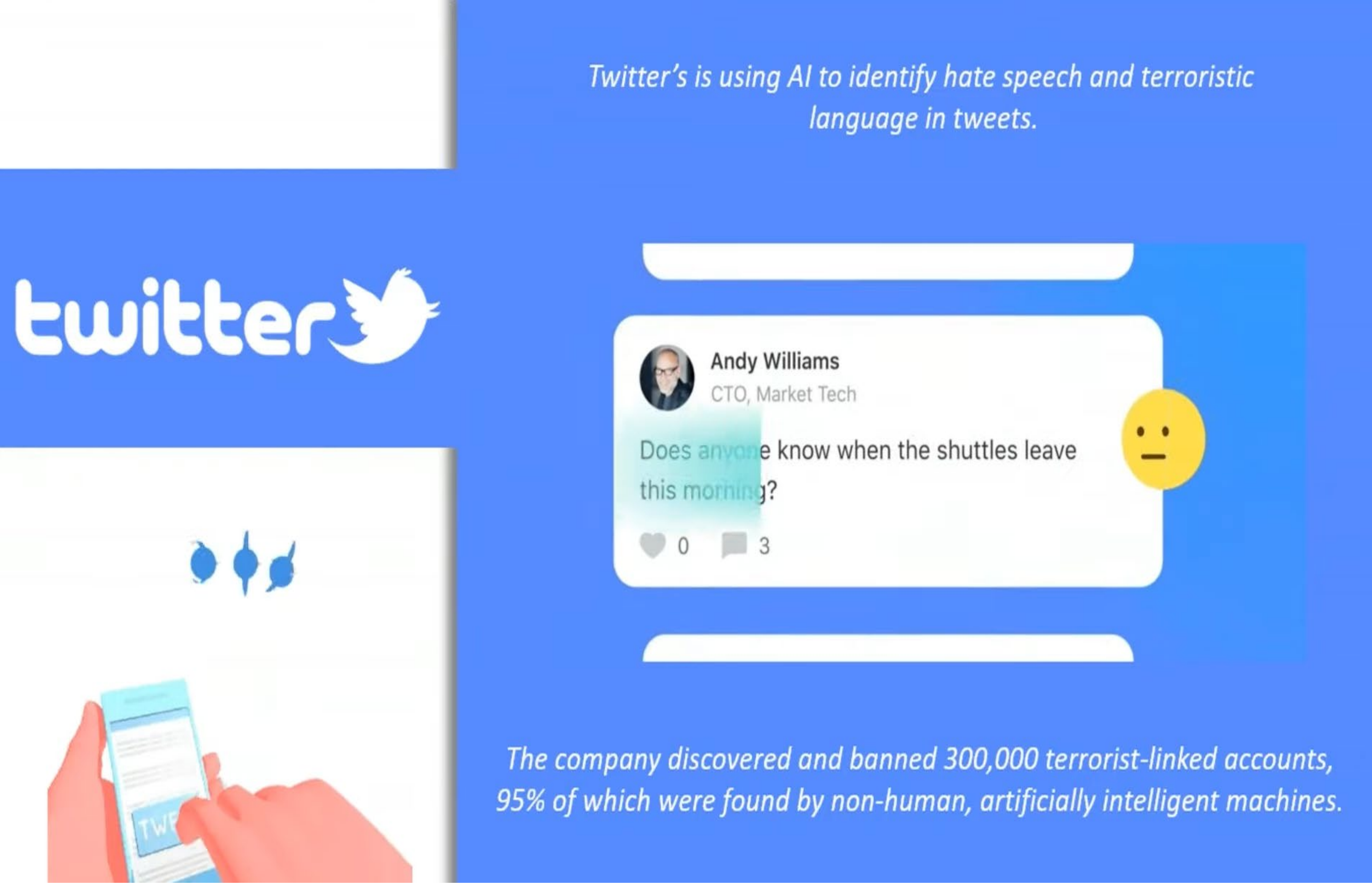
Google

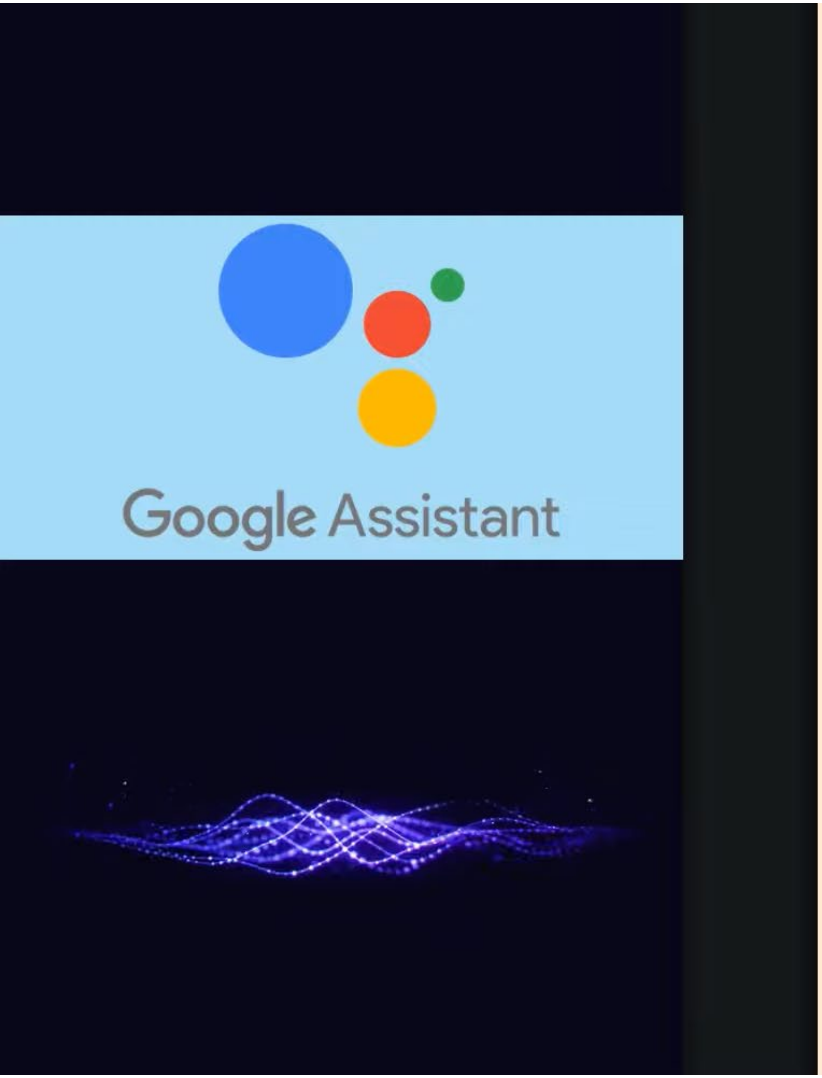
Google working with Indian hospital

Facebook uses Machine Learning & Deep Learning to detect facial features and tag your friends



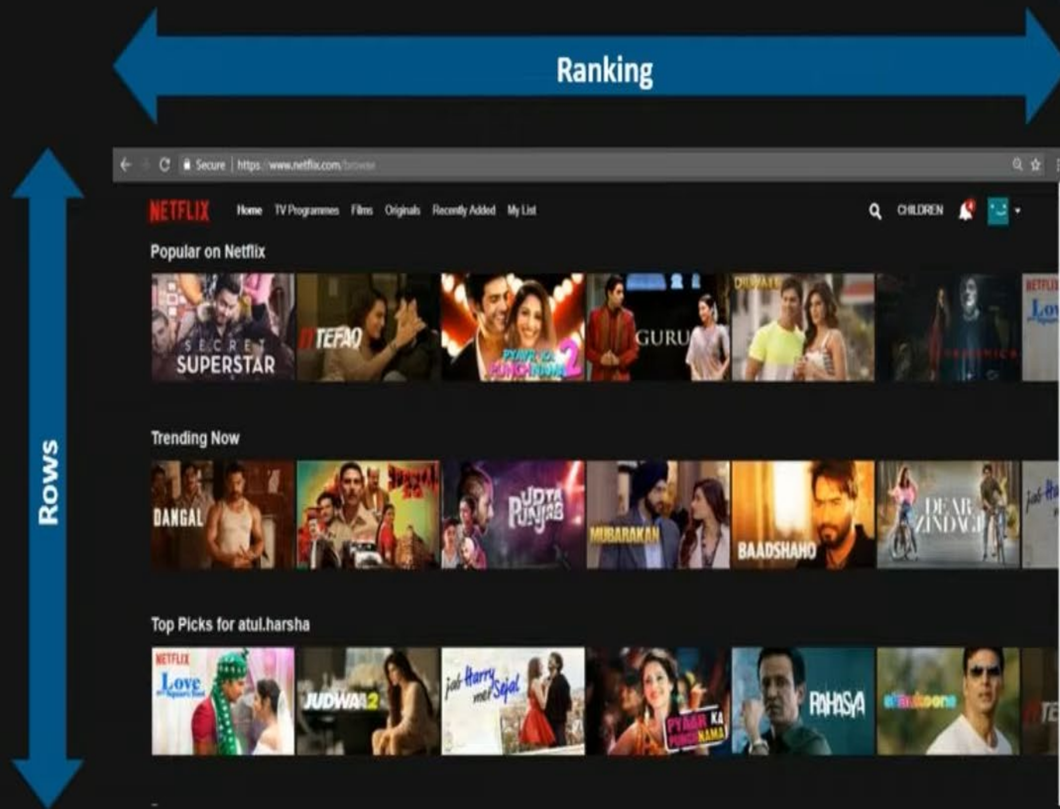
facebook


Twitter's is using AI to identify hate speech and terroristic language in tweets.
twitter
Andy Williams
CTO, Market Tech
Does anyone know when the shuttles leave this morning?
0
3
The company discovered and banned 300,000 terrorist-linked accounts, 95% of which were found by non-human, artificially intelligent machines.



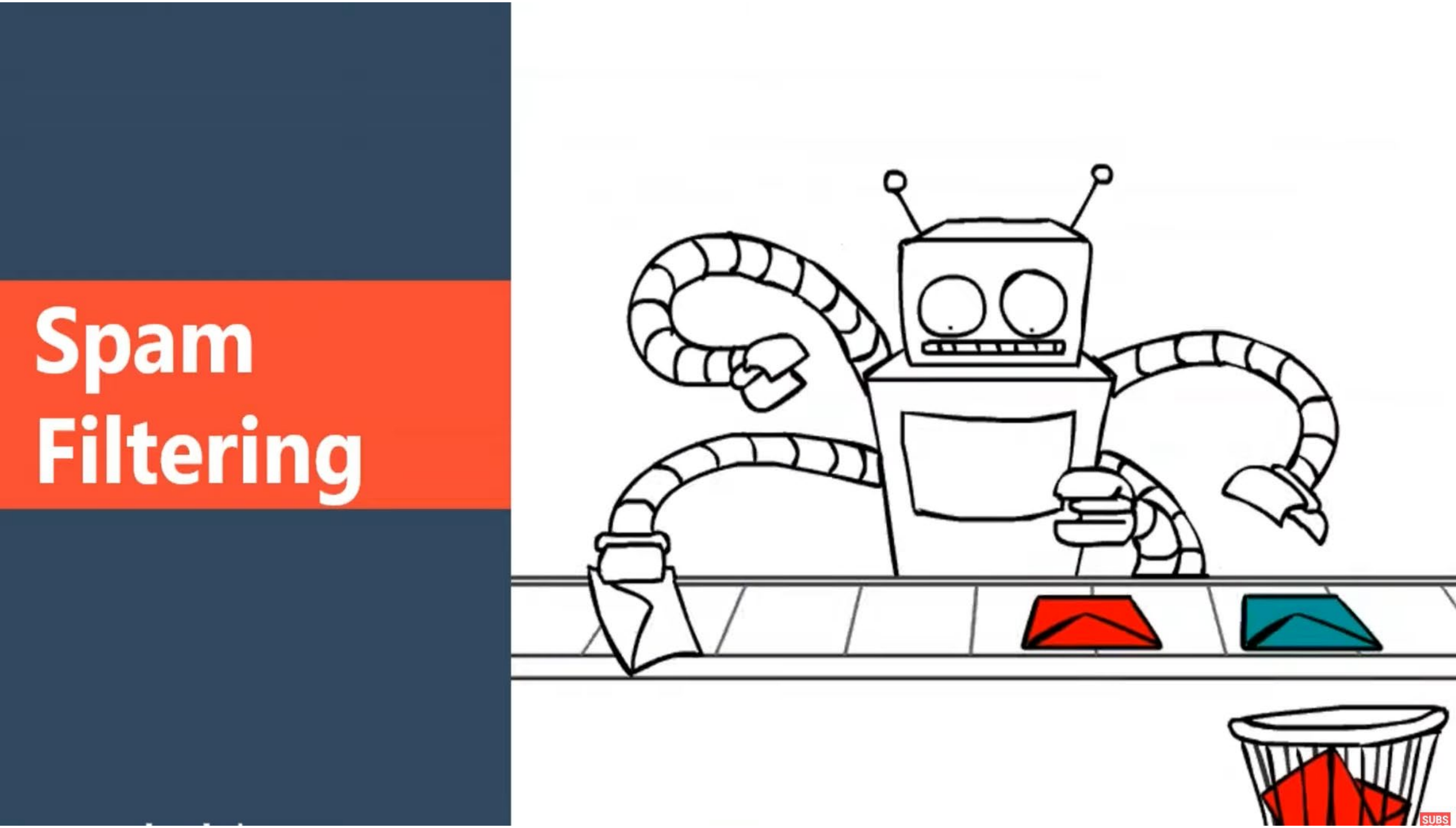
The Google Duplex can not only respond to calls and book appointments for you, it adds a human touch

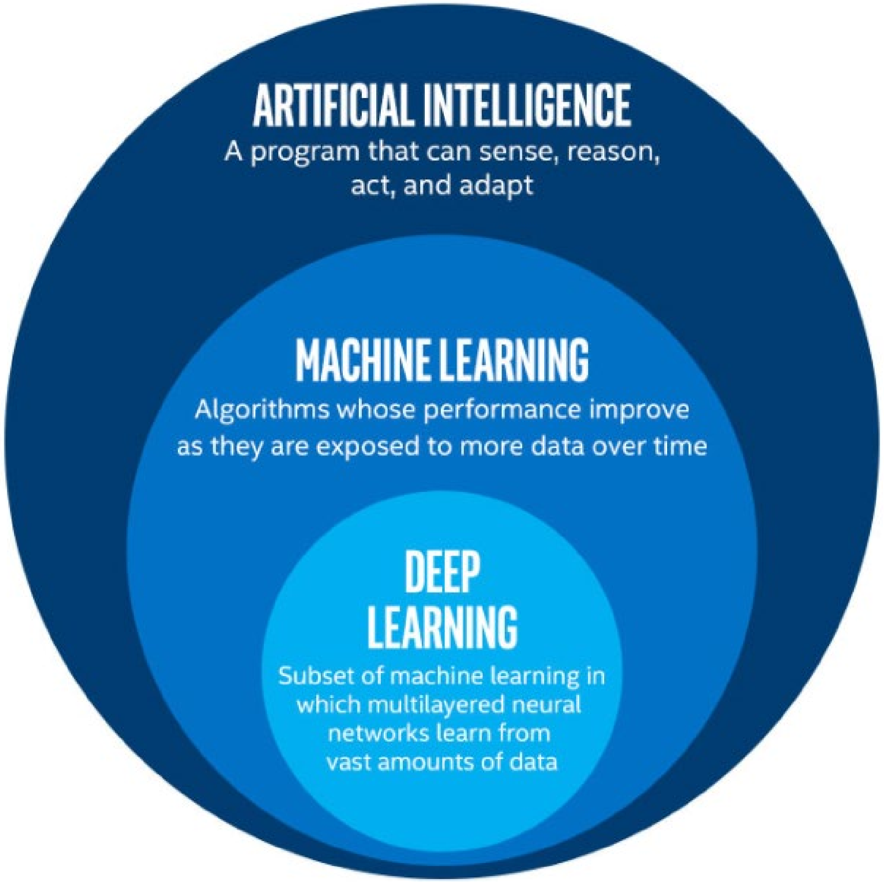
“Tesla will have fully self-driving cars ready by the end of the year and a "robo taxi" version – one that can ferry passengers without anyone behind the wheel – ready for the streets next year”.



Over 75% of what you watch is recommended by Netflix

Recommendations are made by machine learning


Spam
Filtering
SUBS


ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

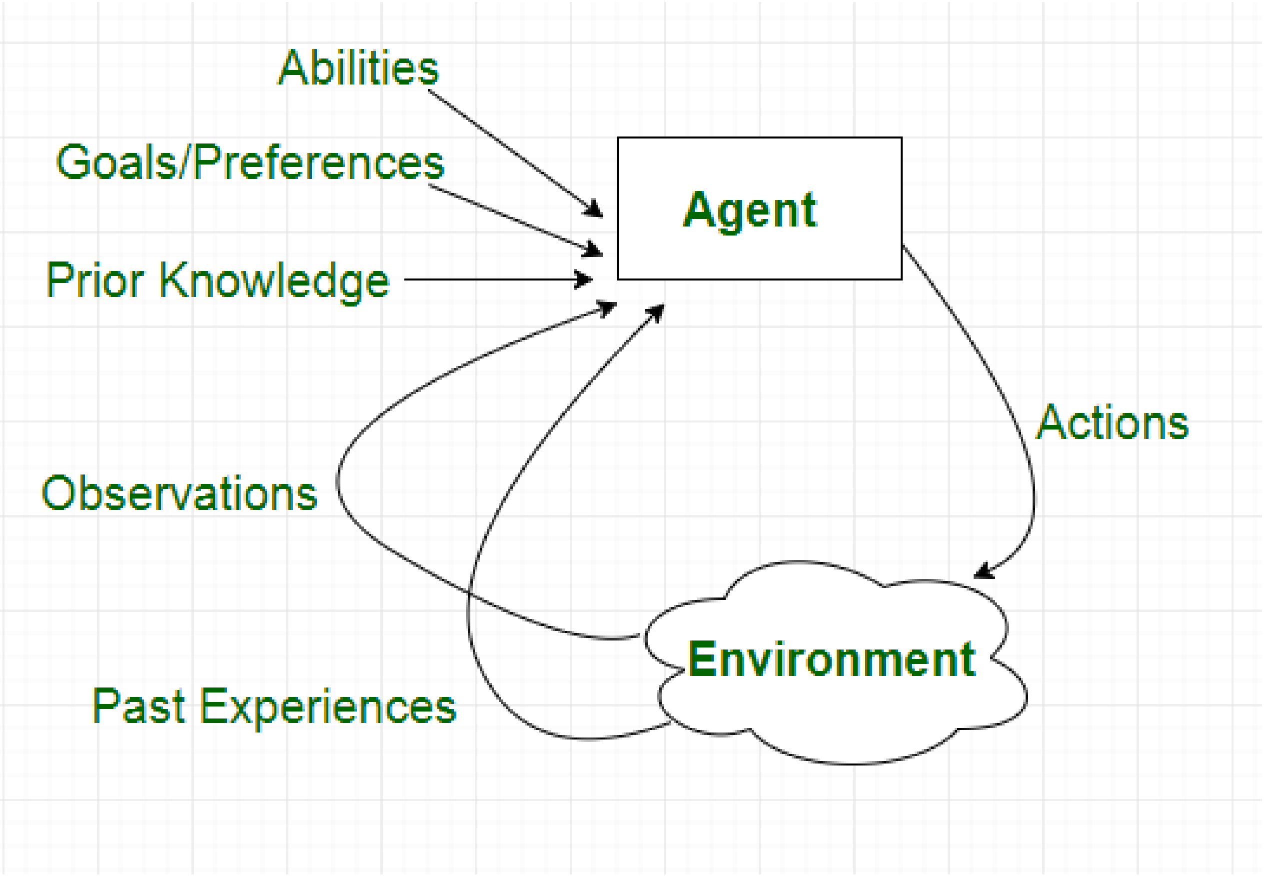
Agents in Artificial Intelligence

- An AI system can be defined as the study of the rational agent and its environment.
- The agents sense the environment through sensors and act on their environment through actuators.
- An AI agent can have mental properties such as knowledge, belief, intention, etc.

What is an Agent?

- An agent can be anything that perceive its environment through sensors and act upon that environment through actuators.
- An Agent runs in the cycle of **perceiving**, **thinking**, and **acting**. An agent can be:
  - **Human-Agent:** A human agent has eyes, ears, and other organs which work for sensors and hand, legs, vocal tract work for actuators.
  - **Robotic Agent:** A robotic agent can have cameras, infrared range finder, NLP for sensors and various motors for actuators.
  - **Software Agent:** Software agent can have keystrokes, file contents as sensory input and act on those inputs and display output on the screen.
- Hence the world around us is full of agents such as thermostat, cellphone, camera, and even we are also agents.

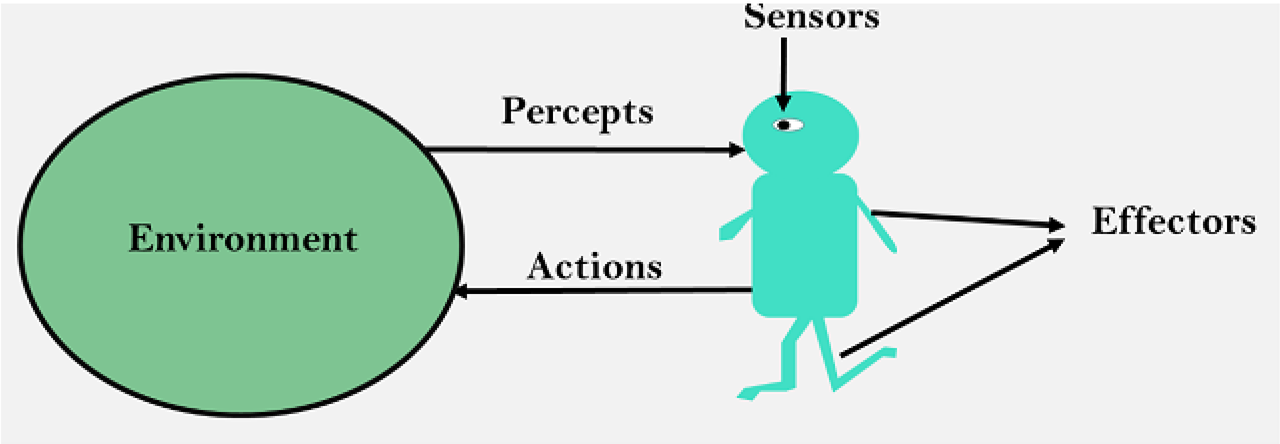
What is an Agent?



What is an Agent?

- **Sensor:** Sensor is a device which detects the change in the environment and sends the information to other electronic devices. An agent observes its environment through sensors.
- **Actuators:** Actuators are the component of machines that converts energy into motion. The actuators are only responsible for moving and controlling a system. An actuator can be an electric motor, gears, rails, etc.
- **Effectors:** Effectors are the devices which affect the environment. Effectors can be legs, wheels, arms, fingers, wings, fins, and display screen.

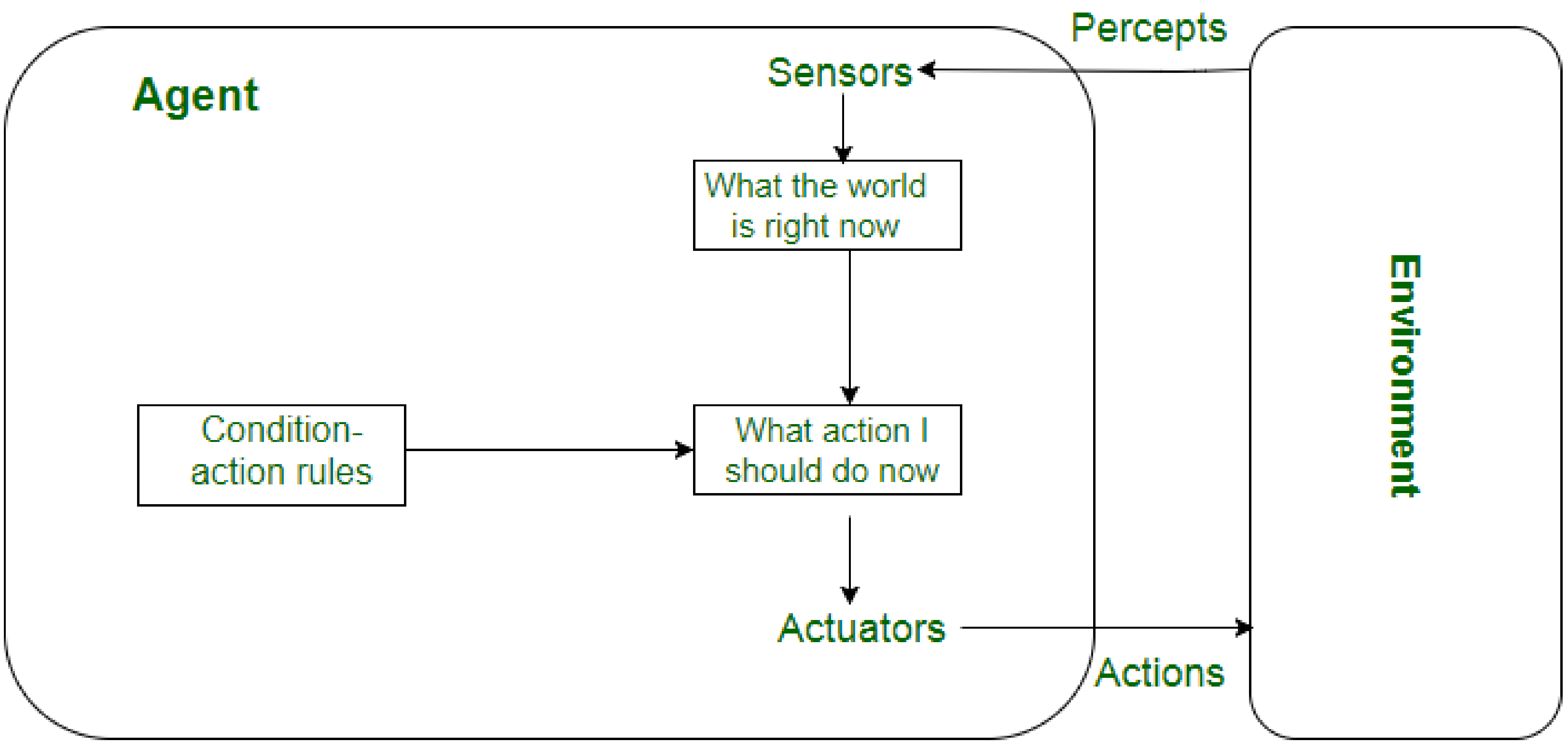
What is an Agent?



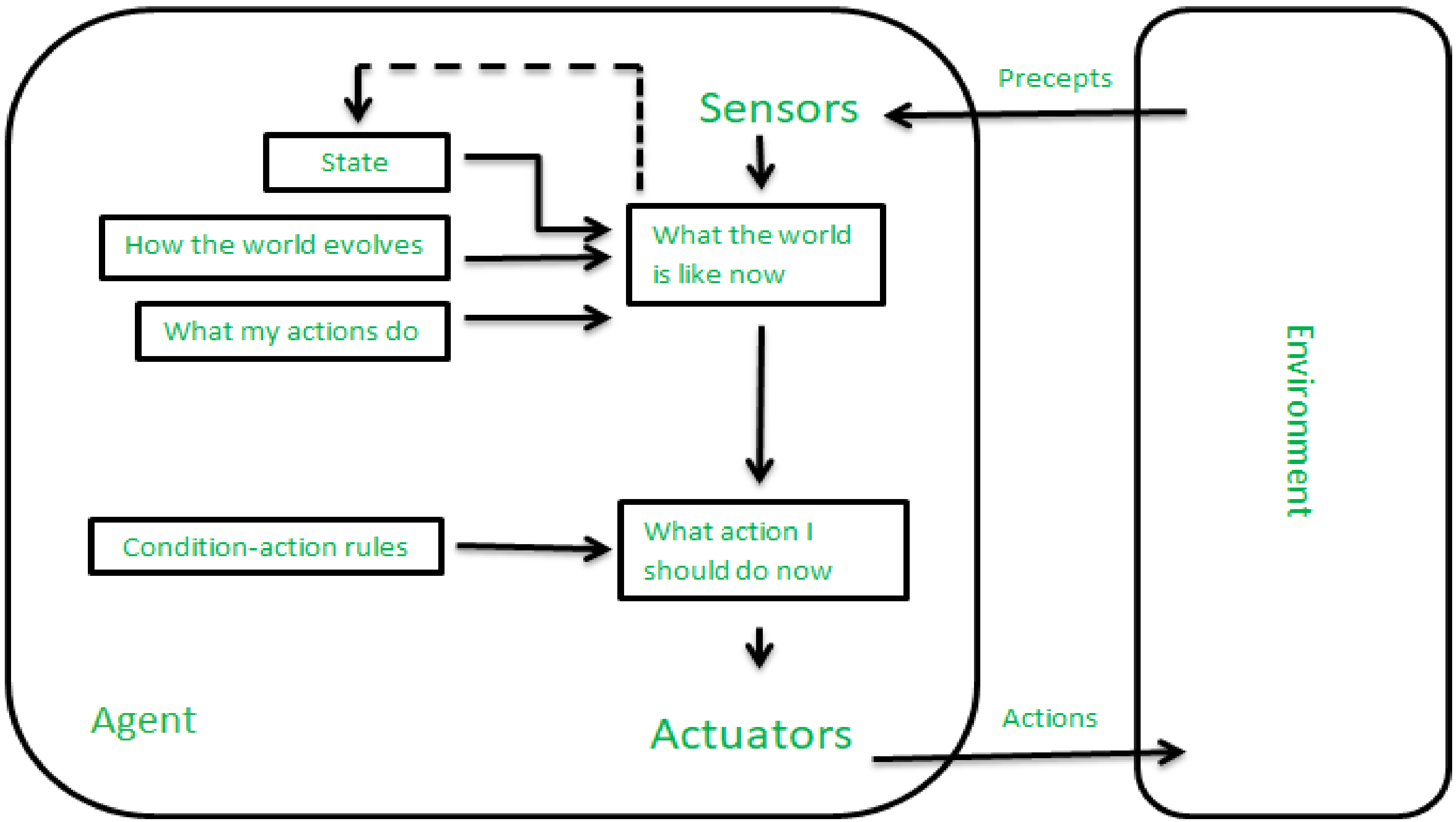
Types of Agents

- Agents can be grouped into following classes based on their degree of **perceived intelligence** and **capability** :
  1. Simple Reflex Agents
  2. Model-Based Reflex Agents
  3. Goal-Based Agents
  4. Utility-Based Agents
  5. Learning Agent

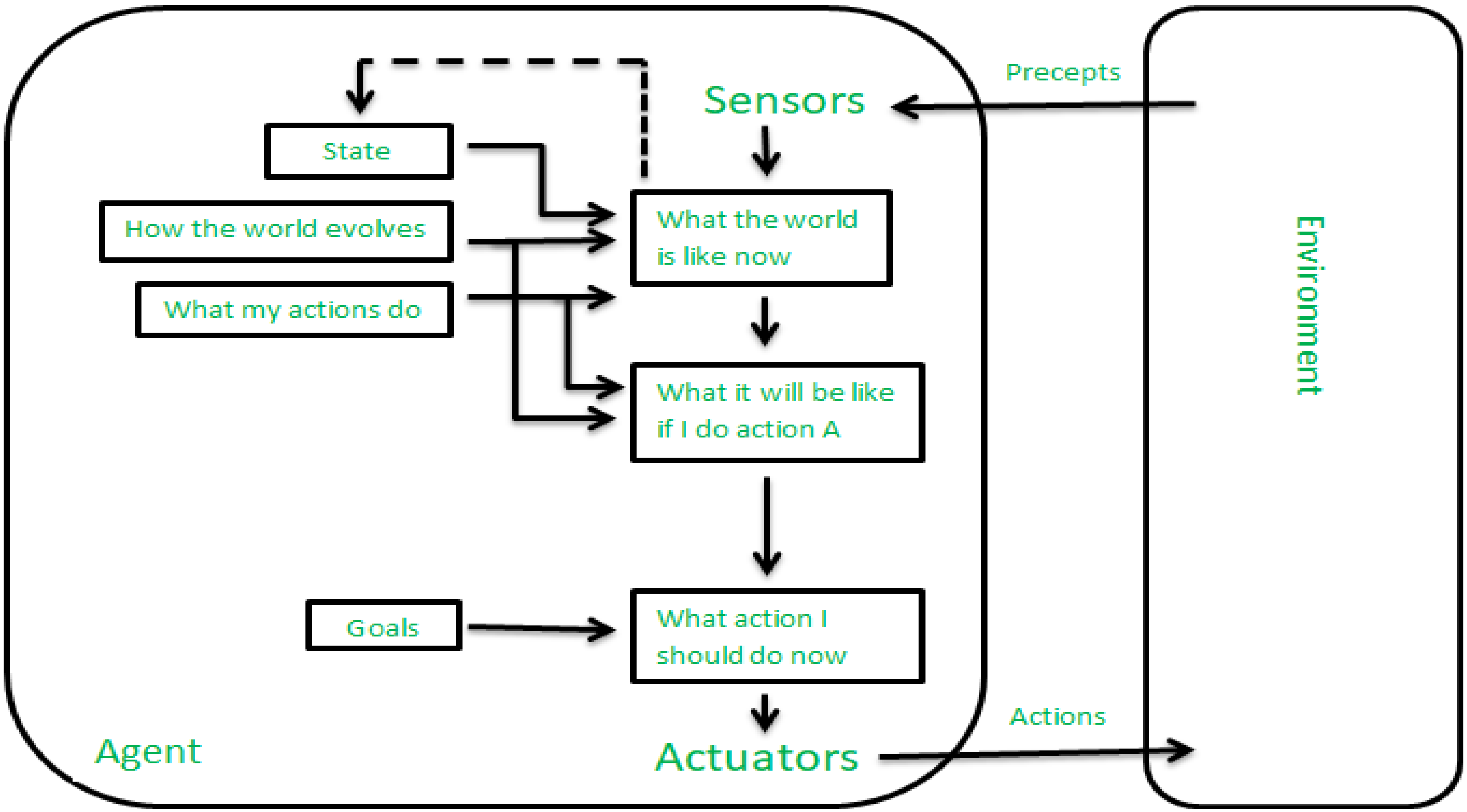
Simple reflex agents



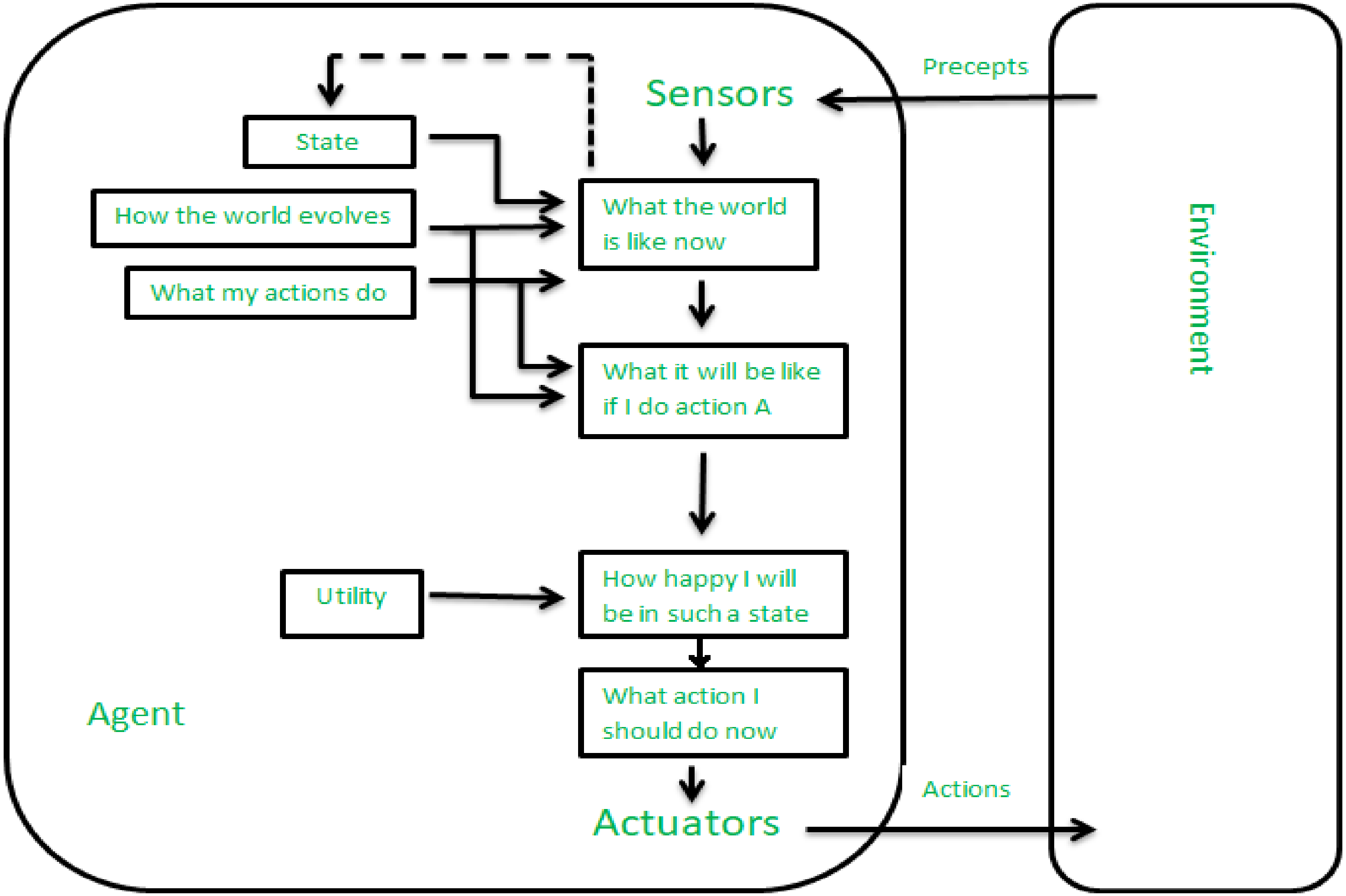
Model-based reflex agents



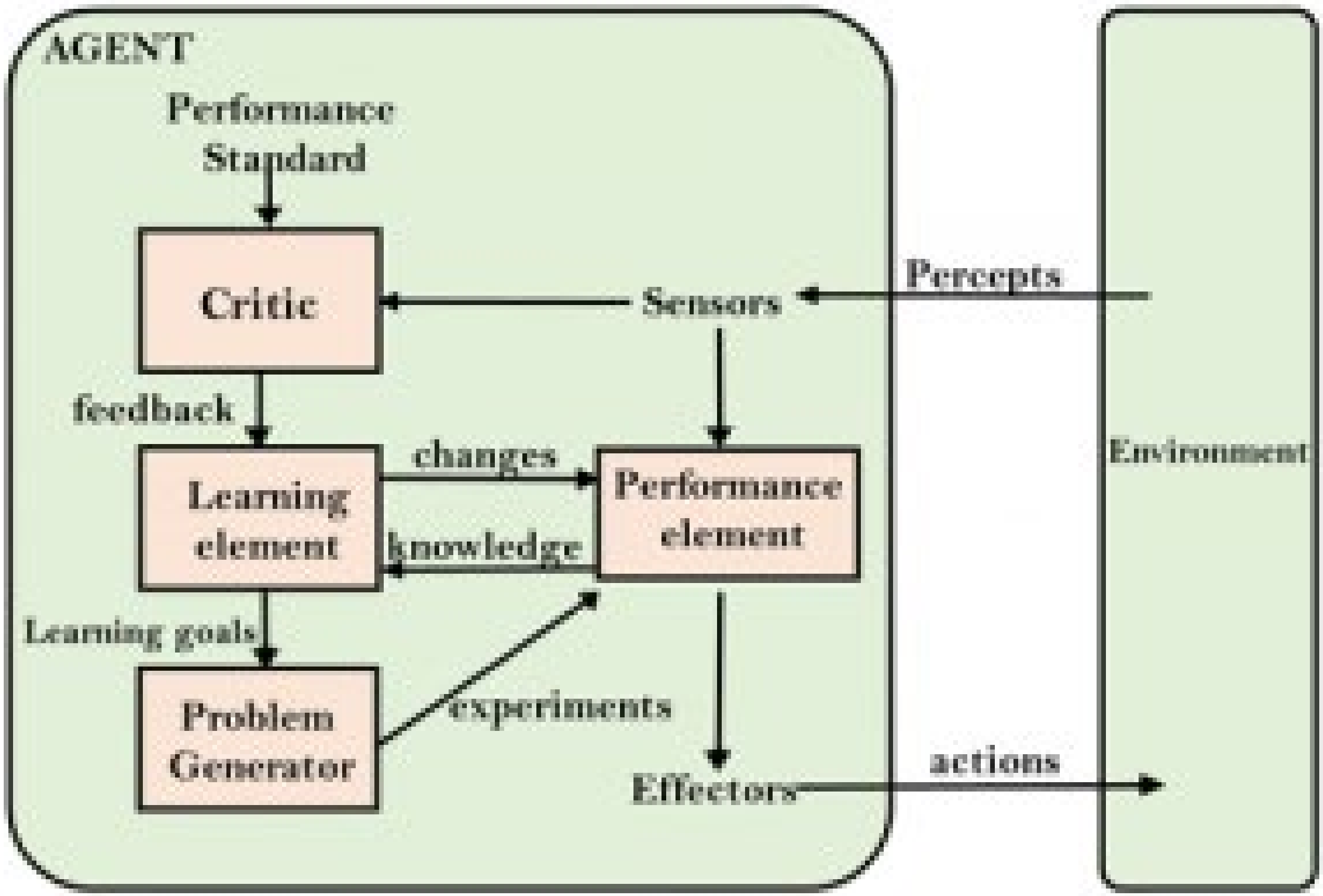
Goal-based agents



Utility-based agents



Learning Agent



Intelligent Agents

- An intelligent agent is an autonomous entity which act upon an environment using sensors and actuators for achieving goals.
- An intelligent agent may learn from the environment to achieve their goals.
- A thermostat is an example of an intelligent agent.
- Following are the main four rules for an AI agent:
  - **Rule 1:** An AI agent must have the ability to perceive the environment.
  - **Rule 2:** The observation must be used to make decisions.
  - **Rule 3:** Decision should result in an action.
  - **Rule 4:** The action taken by an AI agent must be a rational action.

Application of Intelligent agents

- Intelligent agents are applied as automated online assistants, where they function to perceive the needs of customers in order to perform individualized customer service.
- Such an agent may basically consist of a dialog system, an avatar, as well an expert system to provide specific expertise to the user.
- They can also be used to optimize coordination of human groups online Intelligent Agents

Rational Agent

- A rational agent is an agent which has clear preference, models uncertainty, and acts in a way to maximize its performance measure with all possible actions.
- **A rational agent is said to perform the right things**. AI is about creating rational agents to use for game theory and decision theory for various real-world scenarios.
- For an AI agent, the rational action is most important because in AI reinforcement learning algorithm, for each best possible action, agent gets the positive reward and for each wrong action, an agent gets a negative reward.

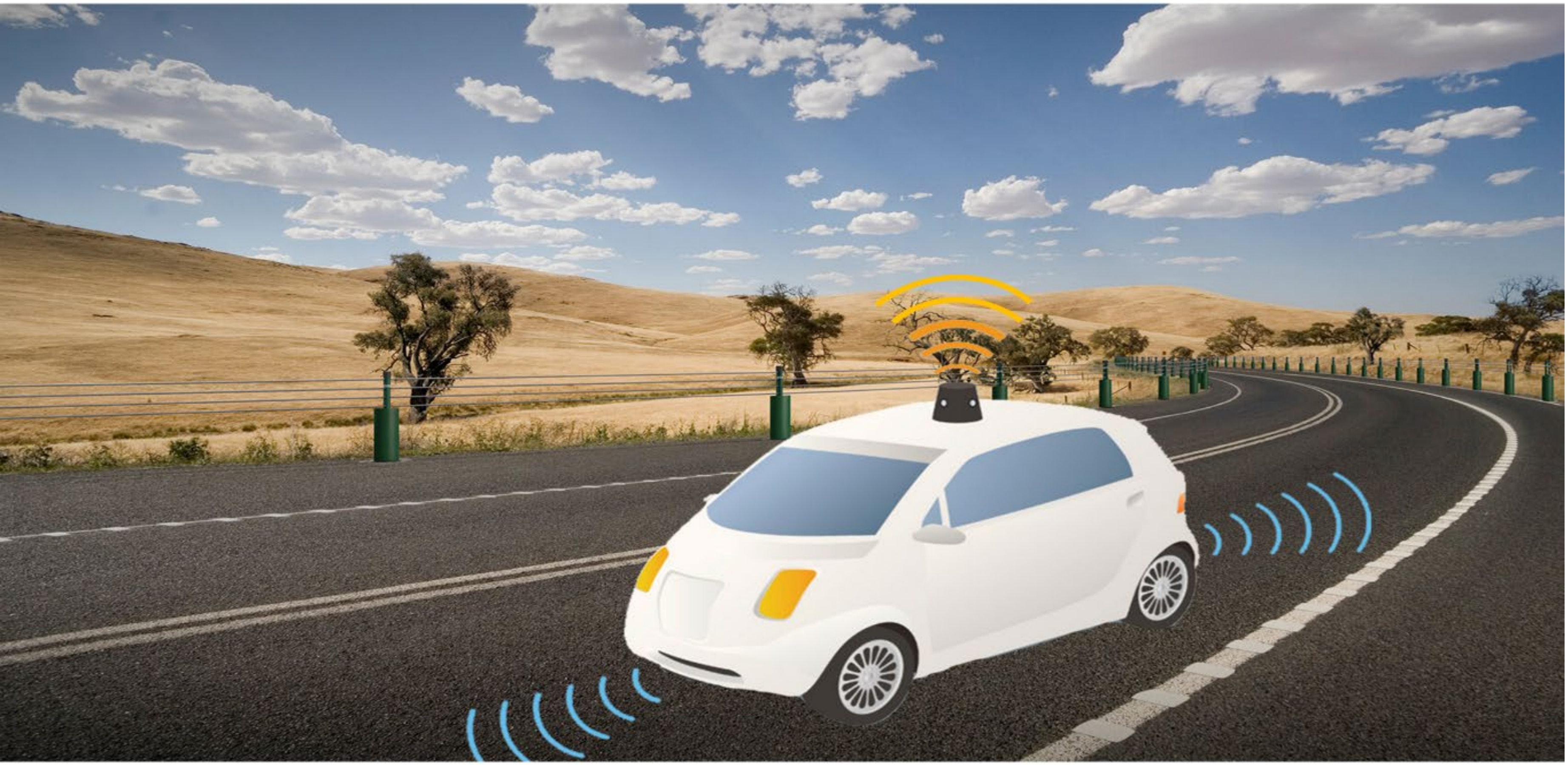
Rational Agent

- Rationality:
- The rationality of an agent is measured by its performance measure. Rationality can be judged on the basis of following points:
  - Performance measure which defines the success criterion.
  - Agent prior knowledge of its environment.
  - Best possible actions that an agent can perform.
  - The sequence of percepts.

Agent

- PEAS Representation
- PEAS is a type of model on which an AI agent works upon. When we define an AI agent or rational agent, then we can group its properties under PEAS representation model. It is made up of four words:
  - **P:** Performance measure
  - **E:** Environment
  - **A:** Actuators
  - **S:** Sensors
- Here performance measure is the objective for the success of an agent's behavior.

PEAS for self-driving cars



PEAS for self-driving cars

- Let's suppose a self-driving car then PEAS representation will be:
- **Performance:** Safety, time, legal drive, comfort
- **Environment:** Roads, other vehicles, road signs, pedestrian
- **Actuators:** Steering, accelerator, brake, signal, horn
- **Sensors:** Camera, GPS, speedometer, odometer, accelerometer, etc.

Example of Agents with their PEAS representation

Agent	Performance measure	Environment	Actuators	Sensors
1. Medical Diagnose	•Healthy patient •Minimized cost	•Patient •Hospital •Staff	•Tests •Treatments	Keyboard (Entry of symptoms)
2. Vacuum Cleaner	•Cleanness •Efficiency •Battery life •Security	•Room •Table •Wood floor •Carpet •Various obstacles	•Wheels •Brushes •Vacuum Extractor	•Camera •Dirt detection sensor •Cliff sensor •Bump Sensor •Infrared Wall Sensor
3. Part - picking Robot	•Percentage of parts in correct bins.	•Conveyor belt with parts, •Bins	•Jointed Arms •Hand	•Camera •Joint angle sensors.

Types of Environment

- An AI agent operates in an *environment*.
- For example, for a self-driving car, the environment is the road and for a chess-playing agent, the environment is the chessboard.
- Further, an environment might also have other agents operating on it.
- Like other autonomous vehicles in the example of the self-driving car.
- The nature of *environment* also varies based on the problem that the AI agent is intended to solve.

Types of Environment

- Fully observable (accessible) vs. partially observable (inaccessible)
- Deterministic vs. stochastic (non-deterministic):
- Episodic vs. sequential:
- Discrete vs. continuous:
- Static vs. dynamic:
- Single agent vs. multi agent:

Thank you

Unit 2

PROBLEM SOLVING AND SEARCH

Problem space and search

- Search is a general algorithm that helps in finding the path in the state space.
- Every problems can be solved with the help of search
- It has one or more paths every path may lead solution or might be dead end.
- In case of dead end, backtracking will work.
- The search algorithm makes use of control strategy.
- In broader sense there are two types of strategies
  - Informed search
  - Uninformed search

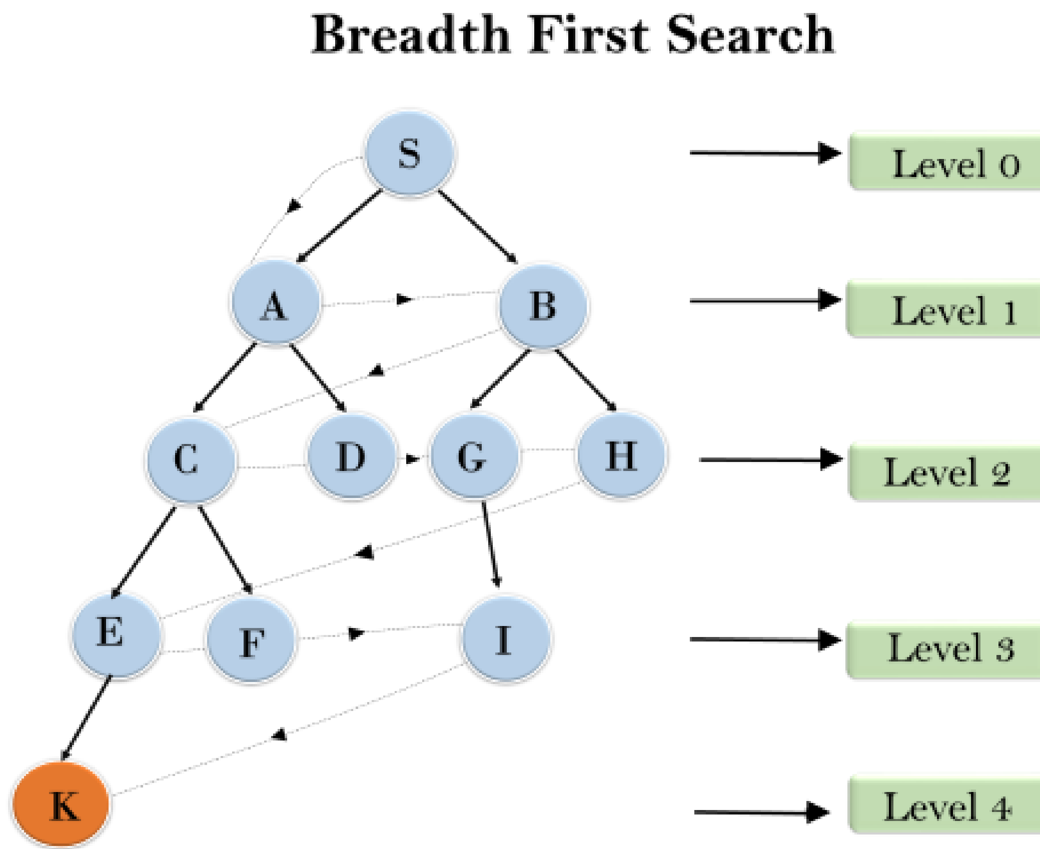
Uninformed Search

Uninformed search strategies that do not take into account the location of the goal.

Intuitively, these algorithms ignore where they are going until they find a goal and report success.

Uninformed Search

BFS

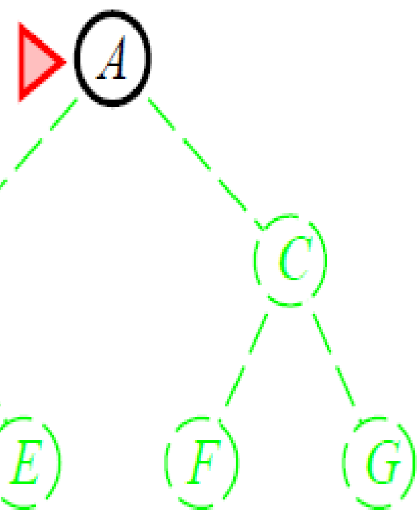


Uninformed Search

BFS

Implementation:

frontier is a FIFO queue, i.e., new successors go at end

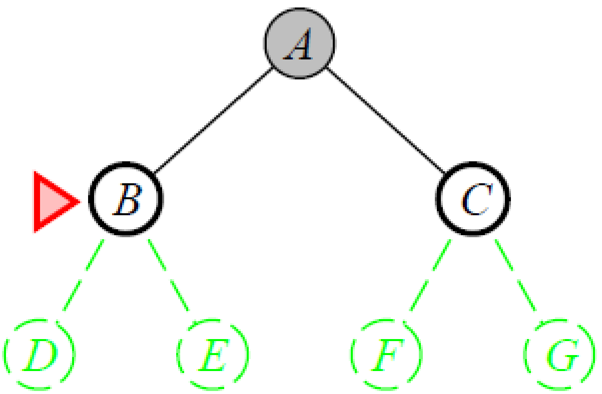


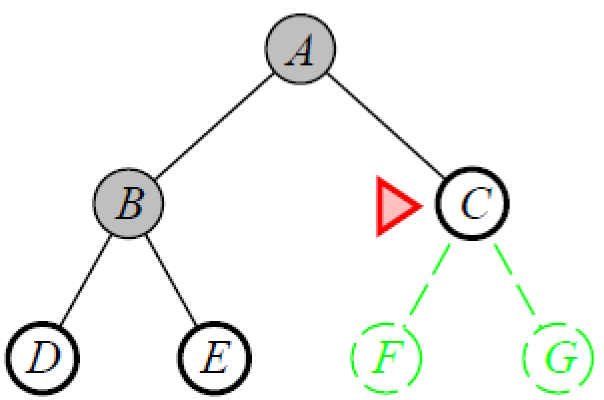
Uninformed Search

Implementation:

frontier is a FIFO queue, i.e., new successors go at end

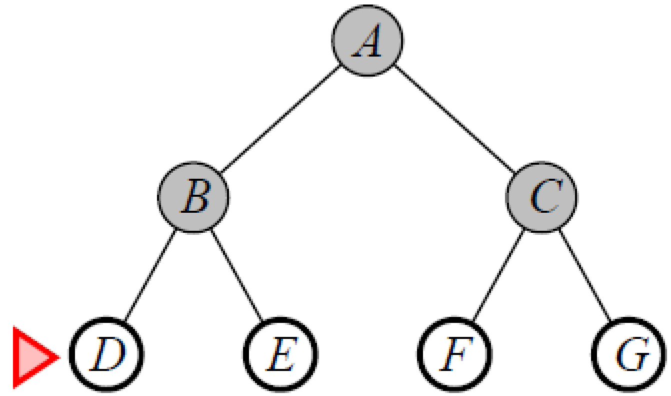
BFS





Uninformed Search

BFS



BFS

Uninformed Search

Time Complexity: Time Complexity of BFS algorithm can be obtained by the number of nodes traversed in BFS until the shallowest Node. Where the d= depth of shallowest solution and b is a node at every state.

$T(b) = 1+b^2+b^3+.......+ b^d= O(b^d)$

Space Complexity: Space complexity of BFS algorithm is given by the Memory size of frontier which is $O(b^d)$.

Completeness: BFS is complete, which means if the shallowest goal node is at some finite depth, then BFS will find a solution.

Optimality: BFS is optimal if path cost is a non-decreasing function of the depth of the node.

Uninformed Search

Depth-first Search

- Depth-first search is a recursive algorithm for traversing a tree or graph data structure.
- It is called the depth-first search because it starts from the root node and follows each path to its greatest depth node before moving to the next path.
- DFS uses a stack data structure for its implementation.
- The process of the DFS algorithm is similar to the BFS algorithm.

Uninformed Search

Depth-first Search

Advantage:

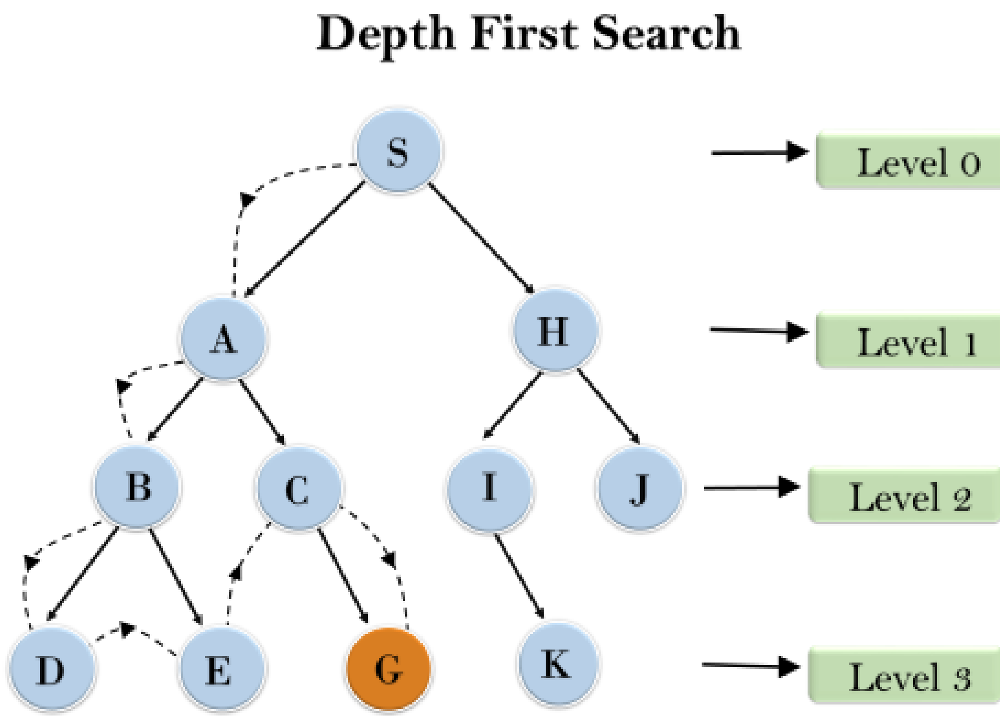
- DFS requires very less memory as it only needs to store a stack of the nodes on the path from root node to the current node.
- It takes less time to reach to the goal node than BFS algorithm (if it traverses in the right path).

Disadvantage:

- There is the possibility that many states keep re-occurring, and there is no guarantee of finding the solution.
- DFS algorithm goes for deep down searching and sometime it may go to the infinite loop.

Uninformed Search

Depth-first Search



Uninformed Search

Depth-Limited Search Algorithm:

- A depth-limited search algorithm is similar to depth-first search with a predetermined limit. Depth-limited search can solve the drawback of the infinite path in the Depth-first search. In this algorithm, the node at the depth limit will treat as it has no successor nodes further.
- Depth-limited search can be terminated with two Conditions of failure:
- Standard failure value: It indicates that problem does not have any solution.
- Cutoff failure value: It defines no solution for the problem within a given depth limit.

Uninformed Search

Depth-Limited Search (DLS) Algorithm:

1. Set the depth limit to the maxdepth to search
2. Initial node=current node

   if initial node=goal, retuen

3. If depth(intal node)> depth limit

   return

Else

   Expand (initial node)

   save(successors) using stack

   Go to step 2

Uninformed Search

Depth-Limited Search Algorithm:

Advantages:

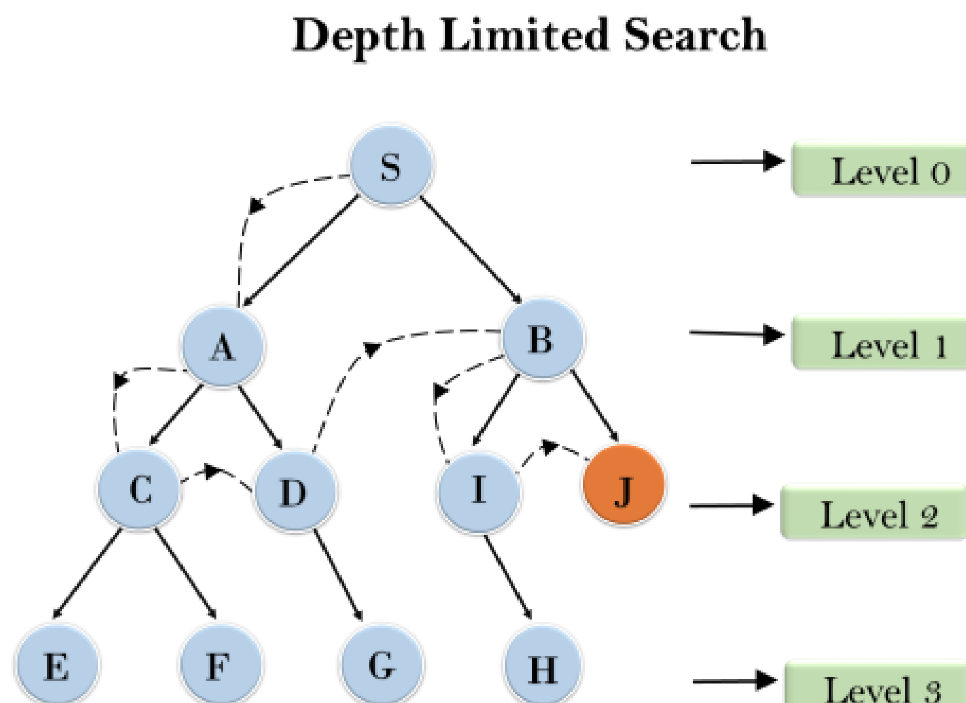
- Depth-limited search is Memory efficient.

Disadvantages:

- Depth-limited search also has a disadvantage of incompleteness.
- It may not be optimal if the problem has more than one solution.

Uninformed Search

Depth-Limited Search Algorithm:



Uninformed Search

Uniform Cost Search

- Uniform-cost search is a searching algorithm used for traversing a weighted tree or graph.
- This algorithm comes into play when a different cost is available for each edge.
- The primary goal of the uniform-cost search is to find a path to the goal node which has the lowest cumulative cost.
- Uniform-cost search expands nodes according to their path costs form the root node.
- It can be used to solve any graph/tree where the optimal cost is in demand.
- A uniform-cost search algorithm is implemented by the priority queue.

Uninformed Search

Uniform Cost Search

- It gives maximum priority to the lowest cumulative cost.
- Uniform cost search is equivalent to BFS algorithm if the path cost of all edges is the same.

Advantages:

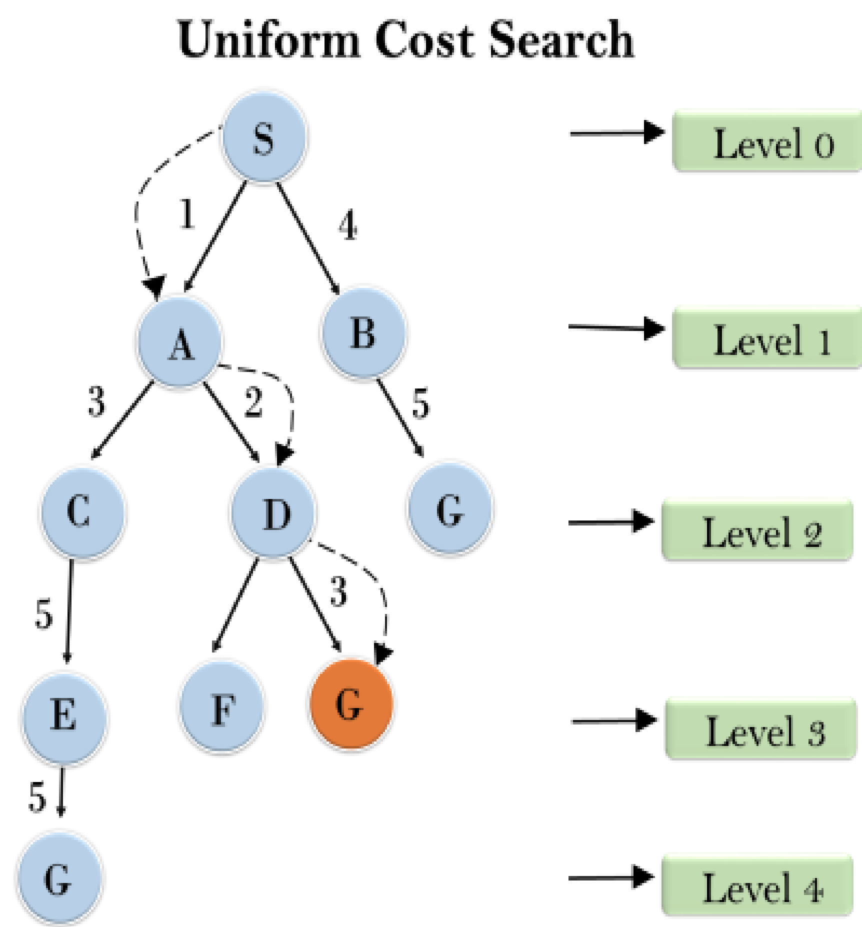
- Uniform cost search is optimal because at every state the path with the least cost is chosen.

Disadvantages:

- It does not care about the number of steps involve in searching and only concerned about path cost. Due to which this algorithm may be stuck in an infinite loop.

Uninformed Search

Uniform Cost Search



Uninformed Search

Uniform Cost Search

- **Completeness:**

Uniform-cost search is complete, such as if there is a solution, UCS will find it.

- **Optimal:**

Uniform-cost search is always optimal as it only selects a path with the lowest path cost.

Uninformed Search

Iterative deepening depth-first Search:

- The iterative deepening algorithm is a combination of DFS and BFS algorithms. This search algorithm finds out the best depth limit and does it by gradually increasing the limit until a goal is found.
- This algorithm performs depth-first search up to a certain "depth limit", and it keeps increasing the depth limit after each iteration until the goal node is found.
- This Search algorithm combines the benefits of Breadth-first search's fast search and depth-first search's memory efficiency.
- The iterative search algorithm is useful uninformed search when search space is large, and depth of goal node is unknown.

Uninformed Search

Iterative deepening depth-first (IDDFS) Search algorithm:

1. Set depth-limit $\leftarrow$ 0
2. Do

    Solution= DLS(depth-limit, initial node)

    If (solution= goal state) then return

    Else

    Depth-limit=depth-limit+1

    Continue

Uninformed Search

Advantages:

- It combines the benefits of BFS and DFS search algorithm in terms of fast search and memory efficiency.

Disadvantages:

- The main drawback of IDDFS is that it repeats all the work of the previous phase.

Bi-directional Search

- A *bidirectional search* is a searching technique that runs two way. It works with two who searches that run simultaneously, first one from source too goal and the other one from goal to source in a backward direction. In in an optimal state, both the searches will meet in the middle off the data structure.

- The bidirectional search algorithm works on a directed graph to find the shortest path between the source(initial node) to the goal node. The two searches will start from their respective places and the algorithm stops when the two searches meet at a node.

Bi-directional Search

- **Importance of the bidirectional approach**– it is a faster technique, and improves the amount of time required for traversing the graph.
- This approach is efficient in for the case when the starting node and goal node are unique and defined. branching factor is same for both directions.
- **Performance measures**
- **Completeness** – Bidirectional search is complete if BFS is used in both searches.
- **Optimality** – It is optimal if BFS is used for search and paths have uniform cost.
- **Time and Space Complexity** – Time and space complexity is **O(b^{d/2})**

Comparison of Uninformed search technique

Criterion	Breadth First	Depth First	Depth Limit	Bidirectional	Uniform Cost	Interactive Deepening
Time	b^d	b^d	b^l	$b^{d/2}$	b^d	b^d
Space	b^d	bd	bl	$b^{d/2}$	b^d	bd
Optimality	Yes	No	No	Yes	Yes	Yes
Completeness	Yes	No	No	Yes	Yes	Yes

Heuristic Search Methods

- Heuristic is a function which is used in Informed Search, and it finds the most promising path.
- It takes the current state of the agent as its input and produces the estimation of how close agent is from the goal.
- The heuristic method, however, might not always give the best solution, but it guaranteed to find a good solution in reasonable time.
- Heuristic function estimates how close a state is to the goal.
- It is represented by h(n), and it calculates the cost of an optimal path between the pair of states.
- The value of the heuristic function is always positive.

Heuristic Search Methods

- Pure heuristic search is the simplest form of heuristic search algorithms. It expands nodes based on their heuristic value h(n).
- It maintains two lists, OPEN and CLOSED list.
- In the CLOSED list, it places those nodes which have already expanded and in the OPEN list, it places nodes which have yet not been expanded.
- On each iteration, each node n with the lowest heuristic value is expanded and generates all its successors and n is placed to the closed list.
- The algorithm continues until a goal state is found.

Heuristic Search Methods

- In the informed search we will discuss two main algorithms which are given below:
  1. **Best First Search Algorithm(Greedy search)**
  2. **A* Search Algorithm**

Best First Search

- Greedy best-first search algorithm always selects the path which appears best at that moment.
- It is the combination of depth-first search and breadth-first search algorithms.
- It uses the heuristic function and search.
- Best-first search allows us to take the advantages of both algorithms.
- With the help of best-first search, at each step, we can choose the most promising node.
- In the best first search algorithm, we expand the node which is closest to the goal node and the closest cost is estimated by heuristic function, i.e.

Best First Search

- **Best first search algorithm:**

  Step 1: Place the starting node into the OPEN list.

  Step 2: If the OPEN list is empty, Stop and return failure.

  Step 3: Remove the node n, from the OPEN list which has the lowest value of h(n), and places it in the CLOSED list.

  Step 4: Expand the node n, and generate the successors of node n.

  Step 5: Check each successor of node n, and find whether any node is a goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to Step 6.

  Step 6: For each successor node, algorithm checks for evaluation function f(n), and then check if the node has been in either OPEN or CLOSED list. If the node has not been in both list, then add it to the OPEN list.

  Step 7: Return to Step 2.

Best First Search

- **Advantages:**

  Best first search can switch between BFS and DFS by gaining the advantages of both the algorithms.

  This algorithm is more efficient than BFS and DFS algorithms.

- **Disadvantages:**

  It can behave as an unguided depth-first search in the worst case scenario.

  It can get stuck in a loop as DFS.

  This algorithm is not optimal.

A* Search Algorithm

- Search problems, where you need to find a path from one point to another, say, point A to point B.
- To approximate the shortest path in real-life situations, like- in maps, games where there can be many hindrances.
- A* algorithm comes up as an answer to these problems.
- A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.
- Informally speaking, **A* Search algorithms, unlike other traversal techniques, it has "brains". What it means is that it is really a smart algorithm which separates it from the other conventional algorithms.**
- Many games and web-based maps use this algorithm to find the shortest path very efficiently (approximation).

A* Search Algorithm

Now to understand how A* works, first we need to understand a few terminologies:

- **Node** (also called **State**) — All potential position or stops with a unique identification
- **Transition** — The act of moving between states or nodes.
- **Starting Node —** Whereto start searching
- **Goal Node —** The target to stop searching.
- **Search Space** — A collection of nodes, like all board positions of a board game

A* Search Algorithm

- **Cost** — Numerical value (say distance, time, or financial expense) for the path from a node to another node.

  g(n) — this represents the ***exact cost*** of the path from the **starting node** to any node **n**

  h(n) — this represents the heuristic ***estimated cost*** from node **n** to the goal node.

  f(n) — lowest cost in the neighboring node n

- Each time A* enters a node, it calculates the cost, f(n)(n being the neighboring node), to travel to all of the neighboring nodes, and then enters the node with the lowest value of f(n).
- These values we calculate using the following formula:

$$f(n) = g(n) + h(n)$$

Thank you