Reminders

- Create a Course Intro
- Set up Python 3.8 virtual environment (I recommend using Anaconda).
- Reading for Tuesday: YLLATAILY Chapters 1 & 2
- HW 1 due Monday
Dangers of focusing on imitating human intelligence as our only model for AI

Type & conflation of multiple definitions of AI

Hey, narrow AI can be useful

Principles for AI engineering: benefits & harms

Is human intelligence the right focus? What are the alternatives?

AI: collaborating with humans

Interdisciplinary aspects of AI
“The problem had to do not just with data analysis per se, but with what database researchers call **provenance**—broadly, where did data arise, what inferences were drawn from the data, and how relevant are those inferences to the present situation?”
“Whether or not we come to understand ‘intelligence’ any time soon, we do have a major challenge on our hands in bringing together computers and humans in ways that enhance human life. While some view this challenge as subservient to the creation of artificial intelligence, another more prosaic, but no less reverent, viewpoint is that it is the creation of a new branch of engineering.”
“While one can foresee many problems arising in such a system—privacy issues, liability issues, security issues, etc.—these concerns should be viewed as **challenges, not show-stoppers.**

We now come to a critical issue: **is working on classical human-imitative AI the best or only way to focus on these larger challenges?**”
“This scope is less about the realization of science-fiction dreams or superhuman nightmares, and more about the need for **humans to understand and shape technology** as it becomes ever more present and influential in their daily lives.

Moreover, in this understanding and shaping, there is a need for a diverse set of voices from all walks of life, not merely a dialog among the technologically attuned. **Focusing narrowly on human-imitative AI prevents an appropriately wide range of voices from being heard.**”
Task-based AI
Our goal is to write programs that can solve tasks. This is sort of the goal of all computer science.

In AI, though, the tasks we focus on are ones that seem to require human intelligence. This is a moving standard- what seems impossible for a computer to solve one day may eventually become very easy.
Almost all AI tasks can be grouped into one of three main categories:

- Search
- Classification
- Generation
AI Agents
Agents

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.

This is extremely broad!
An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**.
Defining an environment
Making problems out of the real world

If our goal is to tackle hard real-world problems, we will need a way of describing them.

Agents exist in environments. How do we define environments?
Defining the environment

Consider my cat Captain Haddock as an agent.
Defining the environment

We could define his environment very simply. Here are two possible states:

State 1: foodInBowl = true

State 2: foodInBowl = false
Defining the environment

Here are Captain Haddock's actions in each state:

<table>
<thead>
<tr>
<th>State</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: foodInBowl = true</td>
<td>eat, wail</td>
</tr>
<tr>
<td>2: foodInBowl = false</td>
<td>wail</td>
</tr>
</tbody>
</table>
Defining the environment

If we represent more information about the environment, we can make more nuanced choices:

State 1: humansHome = true, foodInBowl = true
State 2: humansHome = true, foodInBowl = false
State 3: humansHome = false, foodInBowl = true
State 4: humansHome = false, foodInBowl = false
Defining the environment

Here are Captain Haddock's actions in each state:

<table>
<thead>
<tr>
<th>State</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: humansHome = true,</td>
<td>eat, wail, sleep</td>
</tr>
<tr>
<td>foodInBowl = true</td>
<td></td>
</tr>
<tr>
<td>2: humansHome = true,</td>
<td>wail, sleep</td>
</tr>
<tr>
<td>foodInBowl = false</td>
<td></td>
</tr>
<tr>
<td>3: humansHome = false,</td>
<td>eat, wail, sleep</td>
</tr>
<tr>
<td>foodInBowl = true</td>
<td></td>
</tr>
<tr>
<td>4: humansHome = false,</td>
<td>wail</td>
</tr>
<tr>
<td>foodInBowl = false</td>
<td></td>
</tr>
</tbody>
</table>
Rationality
A rational **agent** is an entity that acts to optimize a desired outcome (accomplish a task).
Given a goal, an AI agent must decide what the best action to take is in order to reach this goal.

For complex tasks, this can mean:

- gathering information
- coming up a set of possible actions
- weighing the best action
- acting
- updating and adapting based on changes to the environment
In task-based AI, our first step is to define a goal.

Captain Haddock has two goals in life:

1. eat
2. sleep
Actions

Given a goal, an AI agent must decide which actions to take.

Captain Haddock has several available actions:

- eat
- sleep
- wail
- stare at birds
Actions

Not all of these are possible in every environment:

- **eat** - successful only if food bowl is not empty
- **sleep** - always successful
- **wail** - always successful, but most effective if there is an audience
- **stare at birds** - successful only if birds
Assessing Consequences

Given a goal, an AI agent must decide which action to take by weighing their consequences.

well, well, well, if it isn't the consequences of my own actions
Some actions have costs as well as consequences!

**eat:**
- achieves goal of devouring food
- updates the environment by removing food

**sleep:**
- achieves goal of sleeping
- replenishes energy

**wail:**
- chance of leading humans to add food to bowl
- cost: lack of dignity

**stare at bird:**
- absolutely no effect
- cost: energy
Solutions

- A **solution** is a sequence of actions from the initial state to a goal state.
- A solution is **optimal** if no other solution has a lower cost.
**Agent Complexity**

**Reflex agent**: decides what to do based only on current state.

*is there food? then I will devour it!*
**Agent Complexity**

**Problem-solving agent**: capable of considering a sequence of actions that form a path to a goal state (planning ahead).

*if I wail, the human might refill my bowl. then I can eat more.*
Defining Search Problems
Formal Definition

1. States: a set $S$
2. An initial state $s_i \in S$
3. Actions: a set $A$
   $\forall s, \text{Action}(s) \rightarrow \text{set of actions that can be taken in } s$
4. Transition Model: $\forall s, \forall a \in \text{Action}(s) \rightarrow s_r$
   $s_r$ is a successor of $s$
5. Path cost: must be additive (sum of individual action costs)
   $c(s, a, s_r)$
6. Goal(s) returns true if $s$ is a goal state
Vacuum World

**States:** A state of the world says which objects are in which cells.

In a simple two cell version,

- the agent can be in one cell at a time
- each cell can have dirt or not
Vacuum World

**States:** A state of the world says which objects are in which cells.

In a simple two cell version,
- the agent can be in one cell at a time
- each cell can have dirt or not

2 positions for agent \( \times 2^2 \) possibilities for dirt = 8 states.

With \( n \) cells, there are \( n \times 2^n \) states.
Vacuum World

Actions:
- Clean
- Move Left
- Move Right

Transition:
- Clean: removes dirt
- Move: moves in that direction unless agent hits wall
Vacuum World
Art: Formulating a Search Problem

Decide:

Which properties matter & how to represent
  • Initial State, Goal State, Possible Intermediate States

Which actions are possible & how to represent
  • Operator Set: Actions and Transition Model

Which action is next
  • Path Cost Function

Formulation greatly affects combinatorics of search space and therefore speed of search
Hard subtask: Selecting a state space

Real world is absurdly complex
State space must be *abstracted* for problem solving
Useful Concepts

State space: the set of all states reachable from the initial state by any sequence of actions.

Path: sequence of actions from one state to another.

Solution: path from initial state to goal.

Search tree: way of representing paths through the state space.
Evaluation
Evaluating Solvers

- Completeness: if there is a solution, is the algorithm guaranteed to find it?

- Optimality: does the strategy find the optimal solution?

- Time complexity

- Space complexity