All-Inclusive [?] Generative Tasks for AI

Dustin Smith
dustin@media.mit.edu
2010

draft: version 3
This is a proposal for unifying areas of AI in order to stitch them together to make progress in new directions.

The particular solution outlined within is intended to create new research opportunities, clarify problem formulations, but certainly not to dictate the direction of the entire field or usurp AI’s diversity of goals.

Benchmarks pose a risk, particularly in the hands of research funders, of prematurely optimizing a field. AI has been stunted by bad benchmarks in the past.

This proposal is incomplete; a starting point that needs development and I am distributing it to encourage critique from researchers in all areas of AI.
Previous Suggestions for Canonical AI Tasks

- Turing Test [Turing 1950]
- Chess [Shannon 1950]
- Compression of English Wikipedia Text [Hutter 2006]
Previous Suggestions for Canonical AI Tasks

- Turing Test [Turing 1950]
- Chess [Shannon 1950]
- Compression of English Wikipedia Text [Hutter 2006]
- Too difficult? Sufficient but not necessary condition for intelligence.

- Indivisible. Does not give insight into sub-problems and is too hard for researchers to get started on (it’s easier to imagine short-cut solutions than all of the intermediate steps necessary to actually solve the problem)

- Irreproducible: Tests cannot be easily replicated
Previous Suggestions for Canonical AI Tasks

- Turing Test [Turing 1950]
- Chess [Shannon 1950]
- Compression of English Wikipedia Text [Hutter 2006]
Claude Shannon had the spirit of the task in mind -- chess’ enormous state-space, he thought, would prohibit the direct engineering of chess playing machines and would encourage research on pattern recognition, learning and other general qualities of problem solving.

- We must assume AI solutions will “overfit” the task specification

- The solution is in software engineering, we must expect straightforward solutions that do not generalize to other classes of problems,

... unless the task itself is an unlimited class of problems!
Previous Suggestions for Canonical AI Tasks

- Turing Test [Turing 1950]
- Chess [Shannon 1950]
- Compression of English Wikipedia Text [Hutter 2006]
The Hutter Prize
- Hutter prize, 50000€, for compressing human knowledge (wikipedia text).

- Reductionist design: Human knowledge reduces to [the ability to generate redundancies in] Wikipedia articles. Wikipedia articles reduce to a binary string that can be compressed - Similar in spirit to algorithmic information theory; intelligence as a program generator?

- Turing Tar Pit:

Theoretician: All Flash animations can be written in Assembly language.

Software Engineer: So what?
Qualities of a good task

- **Easily understandable**: *i.e.* a researcher studying language acquisition can have gross understanding of how his model affects planning and learning.

- **Tests only relevant components** of intelligence, abstracting away “irrelevant” parts.

- **Continual/Scalable benchmarks** that extend from easy to difficult.

- **Modular whenever possible**, recognizing the merits of specialization (and the limitations of humans).
Qualities of a good task

- **Easily understandable**: *i.e.* a researcher studying language acquisition can have gross understanding of how his model affects planning and learning.

- **Tests only relevant components** of intelligence, abstracting away “irrelevant” parts.

- **Continual/Scalable benchmarks** that extend from easy to difficult.

- **Modular whenever possible**, recognizing the merits of specialization (and the limitations of humans).
The task: An agent is plopped in a strange new world, needs to learn about the environment, learn a language, and use the language. Finally, he is tested on his ability to solve problems as explained to him by others in their common language.

Sound familiar?
The task: An agent is plopped in a strange new world, needs to learn about the environment, learn a language, and use the language. Finally, he is tested on his ability to solve problems as explained to him by others in their common language.

1. Download and edit *initialization* file,
2. Program **generates** world to specs
3. AI learns and is evaluated in this world

have a visual representation of agent’s problem solving!
Qualities of a good task

- **Easily understandable**: *i.e.* a researcher studying language acquisition can have gross understanding of how his model affects planning and learning.

- **Tests only relevant components** of intelligence, abstracting away “irrelevant” parts.

- **Continual/Scalable benchmarks** that extend from easy to difficult.

- **Modular whenever possible**, recognizing the merits of specialization (and the limitations of humans).
Relevant components of intelligence

- **Adaptive**: can learn to solve many problems and goals, which may not be fixed in advance.

- **Resourceful**: can solve problems in many ways

- **Reflective**: reasons about itself and is pragmatically rational (will reason about itself, to spend time on problems proportionate to expected utility)

- **Communicational**: can communicate with humans in natural languages.

- “**Useful**”: helps us to live and better understand Earth (intelligence of a human form, not just any intelligence).
Relevant components of intelligence

- **Adaptive**: can learn to solve many problems and pursue goals, which may not be fixed in advance. Test the agent on its learning (generate new problems and constrain agent’s life span)

- **Resourceful**: can solve problems in many ways. Complex environment favors ingenuity in problem solving; evaluation penalizes stubbornness.

- **Reflective**: reasons about itself and is pragmatically rational (will reason about itself, to spend time on problems proportionate to expected utility) Give agent many goals, evaluate overall achievement

- **Communicational**: can communicate with humans in natural languages. Evaluate ability to learn and communicate knowledge by how well the agent/listener can use the knowledge to solve problems.

- **“Useful”**: helps us to live and better understand our world. Generate problems that are relevant to human-level tasks. Use human languages.
Generating a planning space

Initialization file

- objects: 3  # generates three unnamed objects with random
  # len, width, height, mass [?]
Generating a planning space

Initialization file

- objects: 3  # generates three unnamed objects with random
  # len, width, height, mass [?]

Simulator produces random objects in world:

```
"o1"
"o2"
"o3"
```
Generating a planning space

Initialization file

- objects: {“bacon”, {name:“pig”, color:pink}, “box”}

Simulator produces random objects in world:

“bacon”
“pig”
“box”
Generating a planning space

Initialization file

- objects: {"bacon", "pig", "box"}  # with default names
- actions: {left, right, forward, back, pickup, drop}

Predefined set of actions.
Could be richer predicate representations. move(direction={l,r,f,b})
Generating a planning space

Initialization file

- objects: {“bacon”, “pig”, “box”} # with default names
- actions: {left, right, forward, back, pickup, drop}
- obj-obj-relations: {on, in, holding}
Generating a planning space

Initialization file

- objects: {"bacon", "pig", "box"}  # with default names
- actions: {left, right, forward, back, pickup, drop}
- obj-obj-relations: {on, in, holding}
- goal_states: 3

Simulator produces random goals from object/relations and values (these are hidden and must be learned)

G1: in(pig,box) = 2
G2: on(bacon,pig) = 5
G3: \All in(pig,box) = 100
Generating a planning space

Initialization file

- objects: {"bacon", "pig", "box"}  # with default names
- actions: {left, right, forward, back, pickup, drop}
- obj-obj-relations: {on, in, holding}
- goal_states: 3
- event_relations: {during, before, after}
Generating a planning space

Initialization file

- objects: {"bacon", "pig", "box"}  # with default names
- actions: {left, right, forward, back, pickup, drop}
- obj-obj-relations: {on, in, holding}
- goal_states: 3
- event_relations: {during, before, after}
- events: 2

Simulator produces random hidden external conditions and events:

E1: on(bacon,pig) -> green(pig)
E2: in(bacon,pig) -> in(Pig, box)
Phase 1: Exploration

Agent builds model of own actions, differentiating the world in order to achieve goals and avoid anti-goals.
Phase 2: Language Learning

**Implicit goal #1:** the agent is rewarded when it *utters* the correct label when presented with the corresponding example. Envision a Rosetta Stone kind of interface, iterating through examples and counter-examples of:

- **basic properties:** map between regions in geometric perceptual space and color/size/etc words (adjectives)

- **object labels:** map between collections of properties and nouns

- **object relation labels:** map between (objects re-presented in visual space?) and prepositions denoting spatial configurations.

- **action labels:** map between actions (possibly with relations to objects, etc) and verbs

- **event labels:** map between states or transitions between states and verbs.

- **event relation labels:** map between event states; temporal event relations (durative, punctual, composition, sequence)

Language labels are structured (taxonomic) but doesn’t matter yet.
Phase 2: Language Learning

Implicit goal #1: the agent is rewarded when it *utters* the correct label when presented with the corresponding example. Envision a Rosetta Stone kind of interface, iterating through examples and counter-examples of:

- **basic properties**: map between regions in geometric perceptual space and color/size/etc words (adjectives)

- **object labels**: map between collections of properties and nouns

- **object relation labels**: map between (objects re-presented in visual space?) and prepositions denoting spatial configurations.

- **action labels**: map between actions (possibly with relations to objects, etc) and verbs

- **event labels**: map between states or transitions between states and verbs.

- **event relation labels**: map between event states; temporal event relations (durative, punctual, composition, sequence)
Phase 2: Language Learning

- **basic properties:** map between regions in geometric perceptual space and color/size/etc words (adjectives)

Kinds of English Adjectives [Wikipedia]
1. article or pronouns used as adjectives
2. intensifier
3. quality
4. size
5. age
6. **color**
7. participle
8. proper adjective
9. noun used as adjectives
10. headnoun
Phase 3: Communication

Another agent (blank slate?) is plopped into your world, but doesn’t get to explore. You have to tell ‘em what you’ve learned through exploration [implicit goal #2].

- Using only language, tell another agent:
  - what to do / not do,
  - how to do it / not do it
Phase 3: Communication

- **Story Telling:** Tell ‘em what you’ve learned.
  - Using only language, tell another agent
    - *what* to do / not do,
    - *how* to do it / not do it

- **Story Listening:** Learn what to do!
  - Listen to another agent describe how to solve problems in *his* foreign world!
Overall Evaluation

5 Researchers program 5 AI agents.
Overall Evaluation

Step 1: 5 researchers share a configuration file -- agreeing on the set of labels (words), objects, relations, to evaluate their agents with.

- goals: 3
- events: 20

Researchers and their AI agents.

Shared environment initialization file.
Overall Evaluation

Step 1: 5 researchers share a configuration file -- agreeing on the set of labels (words), objects, relations, to evaluate their agents with.

- goals: 3
- events: 20

Researchers and their AI agents.

Step 2. Each simulator randomly generates hidden goals and events in each environment.
Overall Evaluation

Step 3. Each agent explores the environment and then learns the common language to map to states in their environment.

![Diagram showing the relationship between agents and environments](image)

**Learning Agents**

**Generated Environments**

Step 4: ...
Overall Evaluation

Env 1
Agent 1

Env 2
Agent 2

Env 3
Agent 3

Env 4
Agent 4

Env 5
Agent 5
Overall Evaluation

Bo’s agent 1 explains goals in env 1 and how to solve them in
Overall Evaluation

All agents try to follow Agent 1’s stories to solve problems in Env 1 without exploring.
Dustin’s agent 2 explains which goals are in env 2 and how to solve them.
Overall Evaluation

All agents try to follow Agent 2’s stories to solve problems in Env 2.
Overall Evaluation

... and so on ...
- **Explaining**: representing how well the Agent $i$ performed relative to the other agent’s performance in Environment $i$.

- **Listening**: representing how well the agent was able to solve problems in the other environments.

<table>
<thead>
<tr>
<th>Agent</th>
<th>Explaining</th>
<th>Listening</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent 1</td>
<td>good !</td>
<td>impressive!</td>
</tr>
<tr>
<td>Agent 2</td>
<td>amazing !</td>
<td>unprecedented!</td>
</tr>
<tr>
<td>Agent 3</td>
<td>fantastic !</td>
<td>super!</td>
</tr>
<tr>
<td>Agent 4</td>
<td>splendid !</td>
<td>wonderful!</td>
</tr>
<tr>
<td>Agent 5</td>
<td>significant !</td>
<td>terrific!</td>
</tr>
</tbody>
</table>
- A report card

<table>
<thead>
<tr>
<th>Agent 1</th>
<th>Explaining</th>
<th>Listening</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>good !</td>
<td>impressive!</td>
</tr>
<tr>
<td>Agent 2</td>
<td>amazing !</td>
<td>unprecedented!</td>
</tr>
<tr>
<td>Agent 3</td>
<td>fantastic !</td>
<td>super!</td>
</tr>
<tr>
<td>Agent 4</td>
<td>splendid !</td>
<td>wonderful!</td>
</tr>
<tr>
<td>Agent 5</td>
<td>significant !</td>
<td>terrific!</td>
</tr>
</tbody>
</table>

- **Explaining**: representing how well the Agent $i$ performed relative to the other agent’s performance in Environment $i$.

- **Listening**: representing how well the agent was able to solve problems in the other environments.
Qualities of a good task

- **Easily understandable**: *i.e.* a researcher studying language acquisition can have gross understanding of how his model affects planning and learning.

- **Tests only relevant components** of intelligence, abstracting away “irrelevant” parts.

- **Continual/Scalable benchmarks** that extend from easy to difficult.

- **Modular whenever possible**, recognizing the merits of specialization (and the limitations of humans).
Continuously Scaling

This simple task framework can improve as researchers build better agents.

- Objects in environment become richer, with textures, parts, mass, shapes, ...

- Agents get more detailed sensory input, and motor controls (actions they can accomplish)

- Researchers hand code common goal structures as benchmarks:

- Other agents in environment, leading to model social phenomenon.

- Extends to the full set of prepositions (spatial and temporal relations) and increases vocabulary.

- Lexicon becomes structured, hierarchical labels (“thing”/”moveable”/”dog”)

Thursday, February 18, 2010
Qualities of a good task

- **Easily understandable**: *i.e.* a researcher studying language acquisition can have gross understanding of how his model affects planning and learning.

- **Tests only relevant components** of intelligence, abstracting away “irrelevant” parts.

- **Continual/Scalable benchmarks** that extend from easy to difficult.

- **Modular whenever possible**, recognizing the merits of specialization (and the limitations of humans).
An agent

- The components of an agent are modular, and can be shared between researchers (hopefully).
Extra Slides
(about more specific ideas)
Many ways to learn

An agent is plopped in a strange new world, needs to learn about the environment, learn a language, and use the language. Finally, he is tested on his ability to solve problems as explained to him by others in their common language.

Many ways to learn:

- By observing the effects of your actions (blind search, experimentation,...)
- By observing others (their direct actions, inferring implicit intentions)
- By being told (by someone you trust)
- By reasoning (finding and using patterns in knowledge)
Many ways to learn

An agent is plopped in a strange new world, needs to learn about the environment, learn a language, and use the language. Finally, he is tested on his ability to solve problems as explained to him by others in their common language.

Many ways to learn:

- By observing the effects of your actions (blind search, experimentation, ...)

- By observing others (their direct actions, inferring implicit intentions)

  By being told (by someone you trust)

- By reasoning (finding and using patterns in knowledge)
Many ways to learn: By Being Told

The simplest (and most general) model of action/world mapping: a tensor mapping state1, action to state2.

Markov assumption, all states 2s only depend on previous state 1.

MDP: learn action/state conditional probabilities, find optimal action for any state (a policy).

POMDP, the probabilistic generalization of MDPs, introduces observed states, which can be different than actual states.
Many ways to learn: By Being Told

Consider some of the things you know about (poaching an egg, installing RAM, how to placate a baby, copying a DVD) with a quadratic transition space, how can you pick what to do next?

- state spaces are full of rich, reusable structure.
- when you are planning, you can cache/ignore large branches of your decision space.

- *Imprimer:* puts constraints into planning space (desired or avoided states)
Language Learning

An agent is plopped in a strange new world, needs to learn about the environment, learn a language, and use the language. Finally, he is tested on his ability to solve problems in explained to him by others in their common language.

Language: A reflective level operation

- “situated” in the context of problem solving (semantics of language is now the mapping to the agent’s acquired planning knowledge)

- Adjectives: map to sub-spaces of geometric spaces, for example perceptual topologies.

- Nouns: map to combinations of properties (the regions denoted by adjectives)

- Verbs: map to categories of events (composite relations between objects and properties) and orderings between events.