Introduction

www.scss.tcd.ie/Tim.Fernando/AI

Key Phrases:

Can machines think?
- Turing test & ELIZA effect
- AI-complete

Agent & environment
- Cognitive Revolution & Big Data

Levels of intelligence
Can machines think? (Turing 1950)

**Turing test**: can C tell A from B?

From Wikipedia, (Juan Alberto Sánchez Margallo)

Intelligence operationalized: subject to testing
Can machines think? (Turing 1950)

**Turing test:** can C tell A from B?

Intelligence operationalized: subject to testing
... cheating?

From Wikipedia, (Juan Alberto Sánchez Margallo)
ELIZA (Weizenbaum, 1964-66) & artful deception
- use pattern matching and substitution to fake understanding

**ELIZA effect**: humans are inclined to see computers as humans
e.g. when ATM says “thank you”
ELIZA (Weizenbaum, 1964-66) & artful deception
- use pattern matching and substitution to fake understanding

**ELIZA effect**: humans are inclined to see computers as humans

  e.g. when ATM says “thank you”

An AI problem is **AI-complete** if any AI problem is mechanically reducible to it (i.e., it is at least as hard as any other).

E.g. Natural Language Understanding

  *The town councilors refused to give the demonstrators a permit because they feared violence.*

  *Who feared violence?*

  T. Winograd
ELIZA (Weizenbaum, 1964-66) & artful deception
- use pattern matching and substitution to fake understanding

**ELIZA effect**: humans are inclined to see computers as humans

  e.g. when ATM says “thank you”

An AI problem is **AI-complete** if any AI problem is mechanically reducible to it (i.e., it is at least as hard as any other).

E.g. Natural Language Understanding

*The town councilors refused to give the demonstrators a permit because they advocated violence.*

*Who advocated violence?*  

T. Winograd
ELIZA (Weizenbaum, 1964-66) & artful deception
- use pattern matching and substitution to fake understanding

**ELIZA effect:** humans are inclined to see computers as humans

e.g. when ATM says “thank you”

An AI problem is **AI-complete** if any AI problem is mechanically reducible to it (i.e., it is at least as hard as any other).

E.g. Natural Language Understanding

>The town councilors refused to give the demonstrators a permit because they advocated violence.

Who advocated violence?  

T. Winograd

**Caution:** Programs may appear to work better than they do

**Siri rage** (Urban dictionary):

When you get enraged because Siri just doesn’t get it.
Locating intelligence (black box)

Poole & Mackworth

Intelligence: \((\text{abilities, goals, \ldots, experience}) \mapsto \text{action}\)
Locating intelligence (black box)

Intelligence: (abilities, goals, ... , experience) $\mapsto$ action

Turing test: what to say $\rightsquigarrow$ what to do

Poole & Mackworth
### Between agent and environment

<table>
<thead>
<tr>
<th>agent</th>
<th>environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>program</td>
<td>data</td>
</tr>
<tr>
<td>Cognitive Revolution</td>
<td>Big Data</td>
</tr>
<tr>
<td>hard-wired</td>
<td>experienced</td>
</tr>
<tr>
<td>rationalist</td>
<td>empiricist</td>
</tr>
<tr>
<td>nativist</td>
<td>behaviorist</td>
</tr>
<tr>
<td>innate</td>
<td>tabula rasa</td>
</tr>
<tr>
<td>nature</td>
<td>nurture</td>
</tr>
</tbody>
</table>

Turing machine & specialized automaton
## Between agent and environment

<table>
<thead>
<tr>
<th>agent</th>
<th>environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>program</td>
<td>data</td>
</tr>
<tr>
<td>Cognitive Revolution</td>
<td>BIG DATA</td>
</tr>
<tr>
<td>hard-wired</td>
<td>experienced</td>
</tr>
<tr>
<td>rationalist</td>
<td>empiricist</td>
</tr>
<tr>
<td>nativist</td>
<td>behaviorist</td>
</tr>
<tr>
<td>innate</td>
<td>tabula rasa</td>
</tr>
<tr>
<td>nature</td>
<td>nurture</td>
</tr>
</tbody>
</table>

Turing machine &
specialized automaton

Learning (from environment)
trial & error: “data as oil”
Between agent and environment

<table>
<thead>
<tr>
<th>agent</th>
<th>environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>program</td>
<td>data</td>
</tr>
<tr>
<td>Cognitive Revolution</td>
<td>Big Data</td>
</tr>
<tr>
<td>hard-wired</td>
<td>experienced</td>
</tr>
<tr>
<td>rationalist</td>
<td>empiricist</td>
</tr>
<tr>
<td>nativist</td>
<td>behaviorist</td>
</tr>
<tr>
<td>innate</td>
<td>tabula rasa</td>
</tr>
<tr>
<td>nature</td>
<td>nurture</td>
</tr>
</tbody>
</table>

Turing machine &
specialized automaton

Learning (from environment)
trial & error: “data as oil”

Moving target: changing agent & environment
  e.g. change in state
What & how

Real-world task

Modeling

Formal task (model)

Algorithms

Program

unstructured information $\leadsto$ actionable knowledge

Demis Hassabis

www.theguardian.com/technology/2016/feb/16/demis-hassabis-artificial-intelligence-deepmind-alphago
We now embark on our tour of the topics in this course. The topics correspond to types of models that we can use to represent real-world tasks. The topics will in a sense advance from low-level intelligence to high-level intelligence, evolving from models that simply make a reflex decision to models that are based on logical reasoning.

**Traditional approach**

A spell checker:

- **input** "hte"
- **complex program**
- **output** "the"

**Problem**: complexity becomes unwieldy

**Machine learning approach**

- **Learning algorithm**
- **Training examples**
  - hte ⇒ the
  - jeopardy ⇒ jeopardy
  - affedave ⇒ affidavit
  - misilous ⇒ miscellaneous

- **simple program**
  - parameters = [3.2, 1.2, ...]

**input** → **output**
We now embark on our tour of the topics in this course. The topics correspond to types of models that we can use to represent real-world tasks. The topics will in a sense advance from low-level intelligence to high-level intelligence, evolving from models that simply make a reflex decision to models that are based on logical reasoning.

**Traditional approach**

A spell checker:

input "hte" → complex program → output "the"

**Problem:** complexity becomes unwieldy

**Machine learning approach**

- **Training examples**
  hte ⇒ the
  jeopardy ⇒ jeopardy
  affedave ⇒ affidavit
  misilous ⇒ miscellaneous

- **Learning algorithm**

- **Simple program**
  parameters = [3.2, 1.2, ...]

input → simple program → output

**Search problems**
- Markov decision processes
- Adversarial games

**Constraint satisfaction problems**
- Bayesian networks

**Reflex**

"Low-level intelligence"

**States**

**Variables**

"High-level intelligence"

**Logic**

**Machine learning**
Back in Trinity

Undergraduate ML modules

- CS4404 Machine Learning
  Michaelmas Term (5 ECTS)

- CS4LL5 Advanced Computational Linguistics
  Michaelmas Term (5 ECTS)
  unsupervised ML for natural language processing
Back in Trinity

Undergraduate ML modules

- CS4404 Machine Learning
  Michaelmas Term (5 ECTS)

- CS4LL5 Advanced Computational Linguistics
  Michaelmas Term (5 ECTS)
  unsupervised ML for natural language processing

CS3061: a taste building on CS3011 (Prolog)

- logic & agents as Turing machines
- search
- Q-learning & Markov decision processes
- Constraint satisfaction
- Bayesian and Markov networks