

Introduction

What AI is, where it came from, and how this module works

Slides: Week 1 lecture · Seminar: Week 1 sheet · Code: none this week. Before Week 2, download github.com/sakibanwar/comp5001-ai-code (see the end of Section 14)

These notes were written for COMP5001. They follow the Week 1 lecture and Chapters 1 and 2 of Russell and Norvig's *Artificial Intelligence: A Modern Approach* (4th ed.). The definition of an agent and the list of topics come from CS50's *Introduction to Artificial Intelligence with Python* by Brian Yu and David J. Malan (Harvard University), licensed under CC BY-NC-SA 4.0. The notes are shared under the same licence.

THIS WEEK

By the end of the week you should be able to:

- explain what the module covers, how it is assessed, and the rules on AI tools for the portfolio;
- give a working definition of AI and of an agent, and tell symbolic AI from statistical AI;
- outline the history of AI, including the two AI winters;
- describe the methods you used for the four seminar puzzles, and say which later week each one previews.

No code this week. Before Week 2, install Python and download the module code: the checklist is at the end of Section 14.

- | | |
|---|---|
| 1. The module | 8. Symbolic and statistical AI |
| 2. What is AI? | 9. Where the ideas came from |
| 3. Four goals for AI | 10. A short history of AI |
| 4. Agents | 11. Hype and winters |
| 5. Describing the task: PEAS and environments | 12. What can AI do today? |
| 6. Five kinds of agent | 13. Benefits and risks |
| 7. The eight topics | 14. The Week 1 puzzles, and before Week 2 |
| | 15. Check yourself, key terms and reading |

1 The module

COMP5001 is a Level 5 module about how AI works and how to use it well. Each week takes one AI topic: you see the idea in the lecture, discuss where it is used in the seminar, and run and change the code in the practical.

Learning outcomes

By the end of the module you should be able to:

- develop an understanding of the main areas of Artificial Intelligence, including symbolic and statistical approaches;
- understand different methods for search, planning and reasoning;
- design and discuss applications of Artificial Intelligence to a particular scenario;
- demonstrate awareness of the uses and implications of Artificial Intelligence technologies within businesses and organisations.

Outcomes (a) and (b) are about how AI works. Outcomes (c) and (d) are about using it well in a real organisation. The portfolio assesses all four.

The ten weeks

Week 1 is this introduction. Weeks 2 to 9 each take one topic (the table in Section 7). Week 10 is AI in practice and revision, with a portfolio clinic. Every week has three parts:

Lecture	one AI topic, with these notes to read afterwards
Seminar	AI in the real world: discussion and activities on paper. From Week 4, each seminar also has an “AI in practice” activity that practises Part 2 of the portfolio.
Practical	Python code that you run and change, from the module’s code repository

The portfolio

The module has one assessment: a portfolio worth 100% of the mark. It is set around **Kingsgate Couriers**, a made-up parcel company in Hampshire whose managing director wants to know where AI could help.

Part	What you do	Weight
1	Build a small AI system, starting from code you used in a practical	30%
2	Advise the company on AI: discussion and topic research (1,500 words)	55%
3	Reflect on your learning (500 words)	15%

Deadline: 3.30pm, Wednesday 6 January 2027. The practicals prepare you for Part 1 and the seminars for Part 2. For Part 3, keep a few lines each week on what you learned and what you found hard. The assessment brief on the module page has the details.

AI TOOLS IN YOUR PORTFOLIO: NOT PERMITTED

- **Not permitted** at any stage of the portfolio: chatbots (ChatGPT, Claude, Gemini, Copilot, DeepSeek, Perplexity), coding tools (GitHub Copilot, Cursor, Claude Code, Codex, Windsurf), AI features in other software, and AI translation or paraphrasing tools.
- **One exception:** Grammarly-style fixes to spelling, grammar and clarity in sentences you wrote yourself.
- **You can use:** the module slides, seminar sheets and textbooks; documentation (Python, scikit-learn, TensorFlow, NLTK); ordinary web searches, but not AI-generated summaries; help from the module team; and ideas from classmates, as long as your code and writing are your own.
- **In class,** AI tools are fine when an activity asks you to use them.
- **How it is checked:** you may be asked to explain your code and choices in a short conversation; keep your version history (for example Git commits); suspected AI use is handled as academic misconduct; and you sign a declaration on your portfolio's title page.

Why? The portfolio tests whether *you* can build a small AI system, explain how it works, and judge where AI fits in an organisation. An AI tool would be doing that work for you. Before you start, switch off AI code completion in your editor (for example GitHub Copilot in VS Code): some editors switch it on by default.

Reading and help

- Russell, S. and Norvig, P. (2020) *Artificial Intelligence: A Modern Approach*. 4th edn. Pearson.
- Poole, D.L. and Mackworth, A.K. (2023) *Artificial Intelligence: Foundations of Computational Agents*. 3rd edn. Cambridge University Press. Free online at artint.info.

The end of each week's notes lists the sections to read. **Office hours:** Dr Sakib Anwar, UWBS 005, Wednesdays 4 to 5pm, in 30-minute slots booked at calendly.com/sakib-anwar. Email: sakib.anwar@winchester.ac.uk.

2 What is AI?

There is no single agreed definition of artificial intelligence. In this module we use a working one.

ARTIFICIAL INTELLIGENCE

Getting computers to do tasks that would need intelligence if people did them.

Recognising a friend's face, understanding a spoken question, finding the fastest way across town, playing chess and deciding whether an email is a scam all need intelligence when a person does them. A computer that does them is doing AI.

TESTING THE DEFINITION

A calculator works out $4,817 \times 2,963$ in an instant. A person would need to think hard to do that, so is a calculator AI? Most people say no: it follows one fixed method, and it never chooses between options or learns. So the definition is a starting point, not a test you can apply blindly. Sections 3 and 4 make it sharper: most AI is about **choosing actions**.

The line also moves over time. Finding a route and playing chess were headline AI once; now most people call them ordinary software. This is sometimes called the **AI effect**: once a computer can do something reliably, people stop calling it AI. The seminar's "AI or not AI?" vote explores where the line is today.

3 Four goals for AI

People who build AI have aimed at different things. Russell and Norvig sort the aims using two questions. Should the machine **think**, or is it enough that it **acts**? And should it do what **people** do, or what is **logical** and gets the **best result**?

	Like people	Logically, for the best result
Think	Think like people how do people actually think? (cognitive science)	Think logically reach correct conclusions (logic)
Act	Act like people pass as a person (the Turing test)	Act to get the best result do the right thing (rational agents)

Act like people: the Turing test

In 1950 the British mathematician Alan Turing asked "Can machines think?" That question is hard to answer, so he suggested a test instead, which he called the **imitation game**. It is now called the **Turing test**.

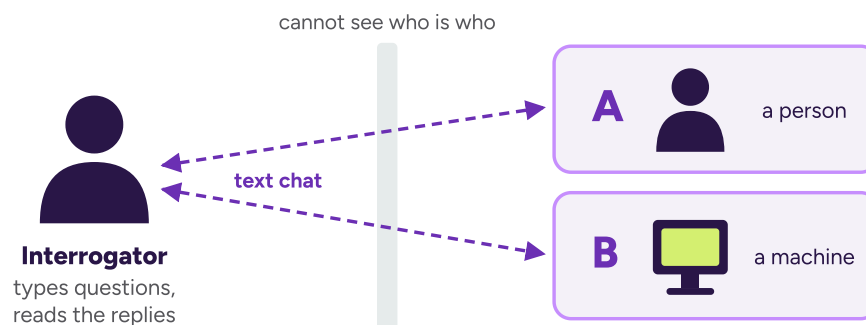


Figure 1. The Turing test. The interrogator chats by text with A and B, without seeing them. One is a person and one is a machine, but the interrogator is not told which. If the interrogator cannot reliably tell them apart, the machine passes.

To pass, a machine needs to understand and write language, store what it knows, reason with that knowledge to answer questions, and learn from the conversation. Those became four of the main areas of AI. Turing predicted that by 2000 a computer would fool an average interrogator about 30% of the time after five minutes of questions.

The test has weaknesses. It rewards *imitation*, not intelligence: in 1966 a simple program called ELIZA (Section 10) fooled some people with a few tricks. The result also depends on who asks the questions, so

two runs of the test can disagree. In a 2025 study, people judged a large language model to be the human 73% of the time in five-minute chats. Few researchers take that as proof that it thinks.

Think like people: cognitive science

To build a program that thinks the way people do, we first need to know how people think. That comes from psychology experiments and from studying the brain, and it is the subject of **cognitive science**. It is now a separate field from AI, but ideas pass both ways: neural networks (Week 8) were inspired by brain cells.

Think logically: the laws of thought

Over 2,000 years ago the Greek philosopher Aristotle wrote down patterns of correct reasoning, such as: "Socrates is a man. All men are mortal. So Socrates is mortal." Each line is a **statement**: a sentence that is either true or false. If the first two statements are true, the conclusion must be true. Logic like this is the language of **symbolic AI** (Section 8). Week 3 writes statements like these in symbols and works out what follows from them, and Week 4 uses them to describe facts and actions for a planner.

Logic alone is not enough, for two reasons. Much of what we know is uncertain ("it will probably rain"), which Week 5 handles with probability. And much intelligent behaviour is not reasoning at all: you pull your hand off a hot pan before you have time to think.

Act to get the best result: rational agents

A **rational** agent does the right thing: the action that is expected to achieve its goals best, given what it knows. It does not have to think like a person, and it does not have to use logic. Even a reflex can be rational, if it gets the best result. Section 4 shows how we score an agent to decide what "best" means.

Perfect rationality is usually out of reach, because there is not enough time to work out the perfect answer. A chess program cannot check every possible game (Week 2). So in practice we look for the best action we can find with the time and computer we have. Week 3's **heuristics**, quick guesses that point a search in the right direction, are one way to do this.

KEY IDEA

Most of AI today, and this module, takes the fourth view: AI is about building **agents that act to get the best result**. They do not have to think like people to do it.

4 Agents

AGENT

An entity that perceives its environment and acts upon that environment.

Russell and Norvig add *how* an agent does this. It perceives through **sensors**: cameras, microphones, touch sensors, or simply the data it is sent. It acts through **actuators**: wheels, motors, a speaker, a screen, or a message it sends. The **environment** is the world the agent is in.

What the agent senses at one moment is a **percept**. Everything it has sensed so far is its **percept sequence**. The agent's job is to choose the next action, using its percept sequence.

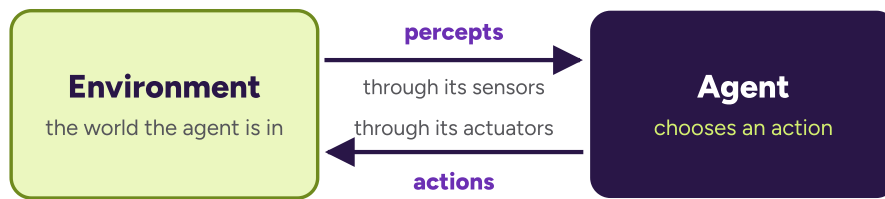


Figure 2. An agent in a loop: it senses the environment, chooses an action, and the action changes the environment, which it senses again.

A sat-nav, a robot vacuum, a spam filter and a chess app are all agents. So is a thermostat: it senses the room temperature and switches the heating on or off. But is it intelligent? It follows one fixed rule (“colder than the setting: heating on”). It never compares options, plans ahead or learns. Being an agent is not the same as being intelligent: the AI is in *how well* the agent chooses its actions.

Worked example: the vacuum world

Russell and Norvig use a tiny world to show these ideas. There are two squares, A and B, and a robot vacuum. Each square is either clean or dirty.

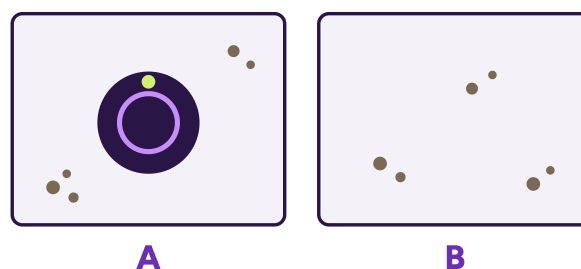


Figure 3. The vacuum world. The robot is in square A, and both squares are dirty.

The robot can sense two things: which square it is in, and whether that square is dirty. It cannot see the other square. It can do four things: move **Left**, move **Right**, **Suck** up the dirt, or do nothing.

We write a percept as a pair in square brackets. How to read it: **[A, Dirty]** means “I am in square A, and square A is dirty”. A table can then list the action to take for each percept:

Percept	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck

In Figure 3 the percept is **[A, Dirty]**, so the robot sucks. A complete table would need a row for every possible *sequence* of percepts (**[A, Clean]** then **[A, Dirty]**, and so on), so it would never end. Instead we write a short rule that gives the same answers:

```
if the square is Dirty: Suck
else if in square A: move Right
else if in square B: move Left
```

AGENT FUNCTION AND AGENT PROGRAM

The **agent function** says what the agent does for every possible percept sequence: the endless table.

The **agent program** is the code that actually produces it: the three lines above. A large part of AI is writing short agent programs that act well.

Rational, not perfect

Is this a good vacuum? To answer, someone has to decide how to score it: its **performance measure**. In the vacuum world we could give one point for each clean square at each moment. Or we could also take away a point for every move, to save electricity. With the second score, a vacuum that keeps moving between two clean squares loses points, so the best behaviour changes. People choose the score, and different scores lead to different “best” actions. Week 2 meets this again, when a sat-nav has to decide whether the best route is the fastest, the shortest or the one that uses least fuel.

GETTING THE SCORE WRONG

Suppose we score a robot vacuum by the **amount of dirt it sucks up**. A clever enough agent could tip the dirt out and suck it up again, over and over: a perfect score and a floor no cleaner than before. The fix is to score what we want to happen in the world (a clean floor), not what we think the agent should do (suck up dirt).

RATIONAL AGENT

An agent that chooses the action it expects to give the best score on its performance measure, given what it has sensed so far and what it knows.

Rational does not mean all-knowing, and it does not mean always successful. You look both ways, the road is clear, and you cross. Halfway over, a part falls off a passing plane and hits you. Crossing was still the rational choice: nobody could have known. A rational agent makes the best choice with the information it has; it cannot see the future.

So a rational agent should **gather information** before it acts (looking before crossing), **learn** from what it senses, and be **autonomous**: able to act on its own experience, not only on what its designer built in.

5 Describing the task: PEAS and environments

Before you design an agent, write down the job it has to do. Russell and Norvig call this the **task environment**, and describe it under four headings, remembered as **PEAS**.

PEAS

Performance measure

How the agent is scored.

Environment

The world it works in.

Actuators

How it acts on that world.

Sensors

How it senses that world.

Agent	Performance measure	Environment	Actuators	Sensors
Self-driving taxi	safe, fast, legal and comfortable trips; profit	roads, other traffic, pedestrians, weather, passengers	steering, accelerator, brake, indicators, horn, a screen for passengers	cameras, radar, GPS, speedometer, engine sensors, a touchscreen
Online shopping agent	low price, good quality, the right items, little time	shopping websites, sellers, delivery firms	showing results to the user, following links, filling in forms	the web pages it reads
Robot vacuum	a clean floor, little electricity, no bumps into furniture	the rooms, furniture, dirt, people and pets	wheels, brushes, suction	bump, dirt and edge sensors, a camera, battery level

Writing the PEAS first stops you building the wrong thing. Compare the taxi with the vacuum world: its performance measure alone has five parts, and some pull against each other (fast against safe).

Six questions about an environment

Some environments are much harder for an agent than others. Six questions show how hard. In each row, the middle column is the easier answer for the agent and the right-hand column is the harder one.

Question	Easier	Harder
Can the agent sense everything that matters?	Fully observable: the Week 1 maze, drawn on one page	Partially observable: poker, where the other players' cards are hidden
Does an action always have the same result?	Deterministic: chess, where a move does exactly what you expect	Stochastic (chance plays a part): backgammon, where dice decide; a taxi on an icy road
Does each decision stand alone?	Episodic: a robot checking each part on a production line, where one decision does not affect the next	Sequential: chess, where every move shapes the rest of the game

Question	Easier	Harder
Does the world wait while the agent thinks?	Static: a crossword	Dynamic: taxi driving, where the traffic keeps moving
Is there a fixed number of situations and choices?	Discrete: chess, with a limited number of positions and moves	Continuous: taxi driving, where speed and steering can take any value in a range
Is the agent on its own?	Single agent: solving a crossword	Multi-agent: chess against an opponent; a taxi among other drivers

A self-driving taxi is at the harder end of every row. So is most of the real world: partly hidden, uncertain, sequential, always changing, continuous and full of other agents.

KEY IDEA

The environment largely decides how the agent has to be built. Weeks 2 to 4 start at the easier end: the agent sees the whole maze, every action does what you expect, and nothing moves while it thinks. Week 4 calls these the assumptions of **classical planning** (“we know every fact”, “nothing else changes the world”, “actions always work”) and asks what a robot should do when one of them fails. Week 2’s games add an **opponent**, and Week 5 deals with what is **hidden** or left to **chance**.

6 Five kinds of agent

How should an agent choose its actions? Russell and Norvig describe four kinds of agent, each able to do more than the one before, and a way to make any of them learn.

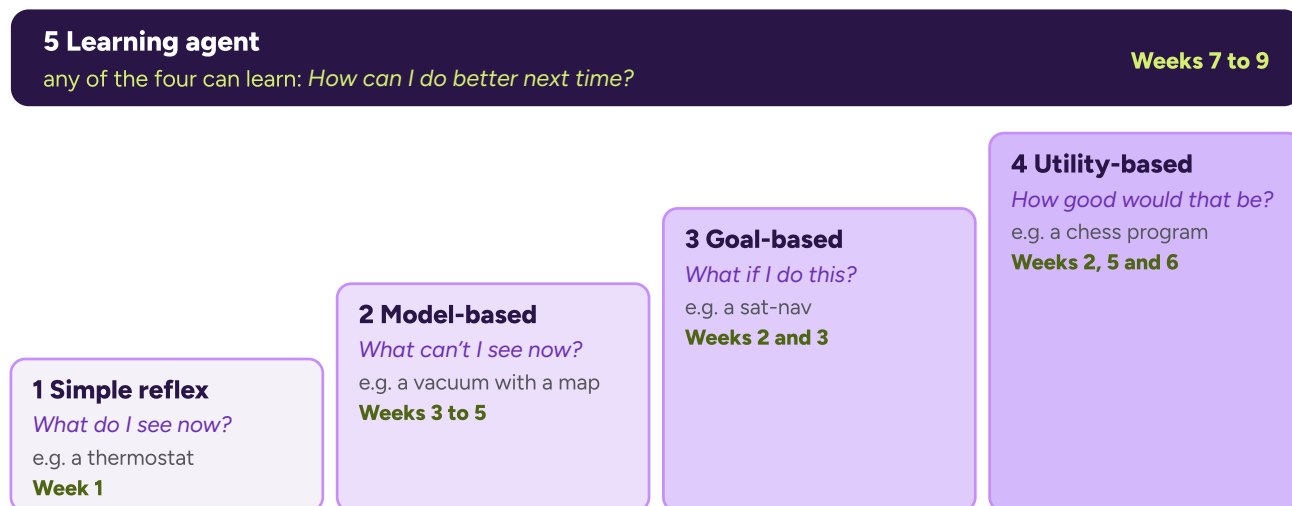


Figure 4. Each kind of agent can do everything the one before it can, and asks one more question. Any of them can be made into a learning agent.

1. Simple reflex agent. It chooses using only what it senses *now*, with if–then rules (“if the square is dirty, then suck”). The vacuum’s three-line rule and the thermostat are simple reflex agents. They are fast and easy to check, but they only work when the current percept tells them everything they need. Our vacuum cannot see the other square, so once both squares are clean it keeps moving from A to B and back for ever, wasting electricity.

2. Model-based agent. It also keeps track of the parts of the world it cannot sense right now: an internal picture, or **model**, of the world. It updates the model using what it knows about how the world changes and what its own actions do. A model-based vacuum remembers that it cleaned B a minute ago, so it can wait in A instead of wandering. A driver remembers the car that has just gone into the blind spot. Weeks 3 and 4 write such a picture as a list of facts and reason with it using logic, and Week 5 adds probability.

3. Goal-based agent. It knows what it is trying to achieve, and it looks ahead: “what will happen if I do this?” A sat-nav’s goal is your destination, and it looks for actions that get you there. To look ahead, it must know what each of its actions does: in Week 4 that is written as each action’s preconditions and effects. This is search and planning, in Weeks 2 to 4.

4. Utility-based agent. A goal is either reached or not. But there are usually many ways to reach it, and some are better than others: faster, cheaper or safer. A utility-based agent gives each outcome a score, its **utility**, and picks the action with the best score. When it cannot be sure what will happen, it picks the best score on average. Minimax in Week 2 uses utilities, and so do Weeks 5 and 6.

5. Learning agent. Any of the four can be made into a learning agent. One part of it, the *critic*, checks how well the agent is doing against the performance measure, and the agent changes its rules to do better. A spam filter that is retrained when people mark its mistakes is a learning agent. Weeks 7 to 9 are about learning.

KEY IDEA

Each kind of agent asks one more question than the last: **What do I see now? What can’t I see? What if I do this? How good would that be?** A learning agent adds: **how can I do better next time?** Most of this module is about building agents further up the steps.

7 The eight topics

AI is not one technique. Weeks 2 to 9 cover eight areas, and each answers a different question. Search is big enough for two weeks, so Week 3 starts by finishing it (with heuristics) before moving on to logic, and Week 4 adds planning.

Week	Topic	The question it answers	For example
2	Search	How do I find a way from where I am to my goal?	a sat-nav route; a chess move
3	Heuristics and Logic	How can a good guess speed up a search? What follows from what I know?	a sat-nav heading towards the destination; a puzzle solved by checking every possibility
4	Knowledge and Planning	How can I reason faster than checking every possibility? Which actions, in which order, reach a goal?	rules for deciding who gets a loan; a warehouse robot fetching an order
5	Uncertainty	How sure can I be when information is missing?	spotting a fraudulent card payment
6	Optimization	Which choice is best when there are millions?	an exam timetable with no clashes

Week	Topic	The question it answers	For example
7	Learning	How can a computer get better from data?	a spam filter; film recommendations
8	Neural Networks	How can a computer learn from images and sound?	face unlock on a phone
9	Language	How can a computer work with human language?	translation apps; chatbots

Part 2 of the portfolio asks you to pick the right method for each of the company's problems, so you need to know what each topic is *for*, not only how it works.

8 Symbolic and statistical AI

The eight topics fall into two broad families. Learning outcome (a) asks you to tell them apart.

TWO KINDS OF AI

Symbolic AI

AI that works with facts, rules and symbols that people write down.

Statistical AI

AI that learns patterns from data and works with probabilities.

WORKED EXAMPLE: A SPAM FILTER BUILT BOTH WAYS

Symbolic. People write the rules: "If the subject contains WINNER, or the sender is on the blocked list, then it is spam." Anyone can read the rules and check them. But scammers soon write W1NNER instead, and someone has to keep writing new rules.

Statistical. Collect 10,000 emails that people have already marked as spam or not spam. The program counts which words appear more often in spam, and gives each new email a probability of being spam, such as 0.97. It copes with new tricks once it is retrained on new examples. But it needs a lot of marked emails, and it is harder to explain why one particular email was flagged.

	Symbolic AI	Statistical AI
Knowledge comes from	people, who write it down	data, which it learns from
What you need	experts to write the rules	many examples
Explaining a decision	easy: show the rule it used	hard: the pattern is spread over many numbers
A case nobody planned for	often fails: no rule covers it	often copes, if the data had similar cases
In this module	mostly Weeks 2 to 4: search, logic, planning	mostly Weeks 7 to 9: learning, neural networks, language

The middle weeks sit between the two: Week 5 brings in probability, and the optimization methods of Week 6 are used by both families.

KEY IDEA

Real systems often **mix the two**. A sat-nav searches a road map for a route (symbolic), using travel times predicted from millions of past journeys (statistical). When you name a method in Part 2 of the portfolio, say whether it is symbolic or statistical, and what difference that makes.

9 Where the ideas came from

AI is a young subject, but it borrowed ideas from much older ones. Each row below comes back somewhere in the module.

Subject	What it gave AI	Where you meet it
Philosophy	rules of correct reasoning; the question of whether a mind could be a machine	Weeks 3 and 4 (logic)
Mathematics	logic, algorithms and probability	every week
Economics	how to make the decision that gives the best outcome, measured by a score called <i>utility</i>	Week 2 (game scores), Weeks 5 and 6
Neuroscience	how brain cells (neurons) pass signals to each other	Week 8 (neural networks)
Psychology	how people and animals think and learn	Weeks 7 and 8
Computer engineering	fast, cheap computers to run it all	the practicals
Control theory	machines that sense and correct themselves, like the thermostat	Section 4
Linguistics	how language is structured: words, grammar, meaning	Week 9 (language)

The story of AI itself starts just before 1950. In 1943 Warren McCulloch and Walter Pitts described a simple mathematical model of a brain cell: the first artificial neuron, and the seed of the neural networks in Week 8.

10 A short history of AI

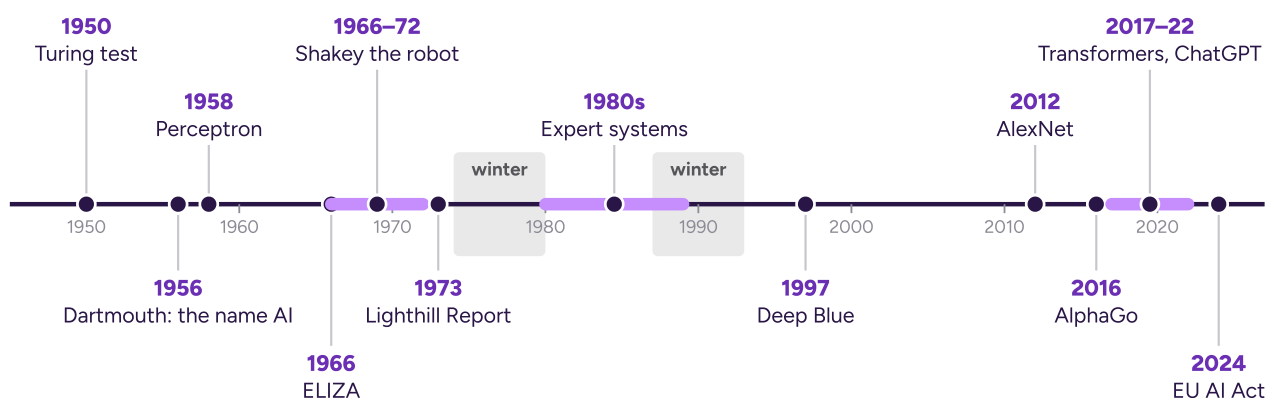


Figure 5. Twelve milestones in the history of AI. The bars show events that lasted several years. The grey bands are the two AI winters (Section 11), when money and interest dried up.

Year	What happened	Why it matters
1950	Alan Turing asks “Can machines think?” and proposes the imitation game.	The Turing test (Section 3).
1956	A summer workshop at Dartmouth College, USA, gives the field its name: artificial intelligence.	The organisers expected real progress in a single summer: an early example of big promises.
1958	Frank Rosenblatt’s perceptron , an early neural network, learns from examples.	The first learning machine. Weeks 7 and 8.
1966	Joseph Weizenbaum’s ELIZA plays a therapist by turning your sentences into questions, using simple pattern rules.	Some users were sure it understood them. Imitation is not understanding (Week 9).
1966–72	Shakey , at the Stanford Research Institute, is the first mobile robot that could plan its own actions.	A* search (1968) and STRIPS planning (1971) were invented for Shakey: A* is in Week 3 and STRIPS in Week 4.
1973	The Lighthill Report tells the UK government that AI has not delivered on its promises.	Methods that worked on small puzzles failed on real problems. UK funding was cut: the first winter.
1980s	Expert systems put human experts’ rules into software. One, R1, worked out the parts needed for customers’ computer orders at Digital Equipment Corporation, and was said to save the company about \$40 million a year.	The first big business use of AI: symbolic AI (Week 4).
1997	IBM’s Deep Blue beats the world chess champion, Garry Kasparov.	Search in games (Week 2), on very fast hardware.
2012	A deep neural network, AlexNet , wins an image recognition contest. It got about 15% of the test images wrong; the next best entry got 26% wrong.	The start of the deep learning boom (Week 8).

Year	What happened	Why it matters
2016	DeepMind's AlphaGo beats Lee Sedol, one of the world's best Go players, by four games to one.	Search and neural networks combined: symbolic and statistical together.
2017–22	The transformer is invented (2017), leading to large language models such as ChatGPT (November 2022).	AI that writes fluent text (Week 9).
2024	The EU AI Act , the first comprehensive law on AI, comes into force.	Rules for AI by level of risk. The Week 7 seminar and Part 2 of the portfolio use it.

Notice the swing between the two families in Section 8. From the 1950s to the 1980s, most of the excitement was about symbolic AI: logic, search, planning and expert systems. The perceptron was an early exception. Since the 1990s, and especially since 2012, statistical AI, which learns from data, has led. AlphaGo shows that the best results often come from combining the two.

11 Hype and winters

AI has gone round the same cycle more than once:

1. **Big promises** attract money and attention.
2. **Results fall short** of the promises.
3. **Funding is cut** and interest fades: an **AI winter**.
4. Work carries on quietly, until a new success starts the cycle again.

In 1958 Herbert Simon and Allen Newell predicted that within ten years a computer would be world chess champion. It took until 1997, nearly forty years.

The **first winter** (mid-1970s) came when early programs that solved small puzzles failed on real problems. The number of possibilities grows far too fast as a problem gets bigger, so a method that checks them all soon runs out of time. Remember the Week 1 seminar: would your maze method still work on a maze with a million squares? Week 3 shows how fast this happens: with 10 true-or-false symbols there are 1,024 possible worlds to check, and with 30 about a billion. The Lighthill Report made this criticism in 1973, and funding was cut in the UK and the USA.

The **second winter** (late 1980s) came after the expert systems boom. Expert systems were expensive to build and to keep up to date, and they failed on cases that their rules did not cover. Many AI companies went out of business.

Today there is huge investment, and big promises again. Nobody knows whether a third winter will come. What you can do is judge each claim on its own.

KEY IDEA

Ask of any AI claim: **what can it actually do, and how do we know?**

WORKED EXAMPLE: TESTING A CLAIM

A company says: "Our AI picks the best job applicants with 95% accuracy."

- **What can it actually do?** It reads CVs and ranks them. It does not interview anyone or see how people perform in the job.
- **How do we know?** "95% accurate" compared with what? Usually with past hiring decisions. If those decisions were unfair to some groups, then 95% agreement means copying that unfairness 95% of the time. Who tested it, on whose CVs, and what happens to the applicants it gets wrong?

The Week 4 and Week 7 seminars come back to screening job applications, and Part 2 of the portfolio asks the same questions about Kingsgate Couriers.

12 What can AI do today?

Russell and Norvig ask their readers which tasks AI can do "at present". Here is their list, lightly adapted. Before you read the right-hand column, cover it and say **yes**, **partly** or **not yet** for each task.

Task	In 2026
Play a decent game of table tennis	Yes, at amateur level. A research robot (Google DeepMind, 2024) beat all the beginners it played and about half the intermediate players.
Drive safely along a curving mountain road	Mostly yes. Self-driving cars cope well with clear roads.
Drive safely through a busy city centre	Partly. Driverless taxis carry passengers in several US cities, but only in areas mapped in detail, and they can still be confused by unusual situations.
Buy a week's groceries on the web	Yes. AI agents can now use shopping websites, though people usually check the basket before paying.
Buy a week's groceries at a busy market	Not yet. Finding, choosing and carrying real objects through a crowd is still too hard for robots.
Play a decent game of bridge	Yes. In 2022 a program beat eight world champions at the card-playing part of the game.
Discover and prove a new mathematical theorem	Yes, sometimes. In 1996 a program settled a maths problem that had been open for 60 years, and AI systems reached gold-medal standard at the International Mathematical Olympiad in 2025.
Design and run a research programme in molecular biology	Partly. AlphaFold predicts the shapes of proteins, and won its makers a share of the 2024 Nobel Prize in Chemistry. People still set the research goals.
Write an intentionally funny story	Debatable. Chatbots write jokes, and some are funny. Whether a chatbot <i>intends</i> anything is a question for Week 9.
Give competent legal advice in a specialised area of law	Partly, with risks. Chatbots answer many legal questions well, but lawyers in the US and the UK have been caught citing court cases that a chatbot made up.
Translate spoken English into spoken Swedish in real time	Yes. Translation apps on a phone can do this live, though not perfectly.

Task	In 2026
Hold a conversation with a person for an hour	Yes. Chatbots can keep a long conversation going (and see the 2025 Turing test study in Section 3).
Perform a complex surgical operation	Not on its own. Surgical robots are steered by surgeons. Operations done by a robot alone have only been tried in research, on animals.
Unload any dishwasher and put everything away	Not yet. Home robots are being built, but any dishwasher in any kitchen is still out of reach.

These answers change fast: several have moved from “no” to “yes” since Russell and Norvig’s 2020 edition. Always check the date of anything you read about what AI can do.

KEY IDEA

Look at which tasks are still “not yet”: they are things a child can do, like moving round a kitchen or picking up a cup. The tasks people find hard, such as chess, Go and maths, turned out to be easier for AI. This surprise is called **Moravec’s paradox**, after the robotics researcher Hans Moravec.

13 Benefits and risks

AI brings real benefits. It takes over repetitive work, so people can spend their time on the parts of a job that need a person. It lets a business produce more goods and services for the same effort. And it speeds up science: AlphaFold’s protein shapes are used in research on new medicines.

It also brings risks. Russell and Norvig list these:

Risk	For example
Autonomous weapons	weapons that choose and attack targets with no person deciding
Surveillance and persuasion	face recognition that tracks people in the street; fake videos and targeted messages that sway voters
Biased decisions	a loan or hiring system that learns past unfairness from its data
Jobs	tasks are automated: some jobs change, and some disappear
Safety-critical uses	a mistake by AI in a car or a hospital can injure or kill
Cybersecurity	AI helps attackers write convincing scam emails, as well as helping defenders spot them

Getting what you asked for

Some researchers also worry about the long term: machines that become far more capable than people. Russell and Norvig call this the **gorilla problem**. Humans and gorillas share an ancestor, but today the gorillas’ future depends on what humans decide, not on what gorillas want. Others think such machines are a long way off. Either way, there is a lesson for today’s systems.

In the Greek myth, King Midas asked that everything he touched would turn to gold. He got exactly what he asked for, and then could not eat or drink. This is the **King Midas problem**: an AI system that does exactly

what we asked, rather than what we meant. The robot vacuum in Section 4, tipping out dirt to suck it up again, is a small example.

KEY IDEA

AI systems should be designed so that their goal is what people really want, and so that people can **check and correct** them. The Week 3 seminar asks a first version of this question (a railway signalling system has been proved safe by logic: can we trust it completely?), and the Week 8 seminar comes back to it as *human oversight*.

In the seminar you choose an organisation (a supermarket, a bank, the NHS or a university), find three uses of AI, and give a benefit, a risk and who is affected for each. That is the start of Part 2 of the portfolio.

14 The Week 1 puzzles, and before Week 2

In the seminar you solved four puzzles with paper and a pen. Each one previews a method that comes back later in the module.

Puzzle	What you had to do	What it is called	Week
The maze	find a path, then write a method that works for any maze	search: depth-first and breadth-first search; then greedy best-first search and A*, which head towards the goal	2 and 3
The river crossing	write down the states and find a sequence of crossings	states and actions (Week 2); the same puzzle written as facts and actions for a planner (Week 4)	2 and 4
Knights and knaves	check which worlds fit what each person said	logic and model checking	3
The exam timetable	give each exam a day with no clashes, and undo a choice when stuck	constraint satisfaction and backtracking	6

In every puzzle you had to keep track of possibilities: squares still to try, states already visited, worlds still possible, days still free. When you solved the maze, you probably headed straight for B rather than trying every square: Week 3 turns that idea into greedy best-first search and A*. But looking at the picture only works for small puzzles. A method written down step by step, an **algorithm**, can be followed by a computer on a maze with a million squares. Weeks 2 to 6 are about such algorithms. The full solutions to the puzzles are in the Week 1 seminar solutions on the module page.

BEFORE WEEK 2

- Install Python 3 from python.org, or check that it is installed: `python3 --version`.
- Download the module code from github.com/sakibanwar/comp5001-ai-code (green **Code** button, then **Download ZIP**) and unzip it.
- In a terminal, go into the `week02-search` folder and run `pip install -r requirements.txt`.
- Run `python maze/maze.py maze/mazel.txt` and open the picture `maze.png` that it makes.

If `python` is not found, try `python3` on a Mac or `py` on Windows, and use the same word in every command after that. If something else goes wrong, bring the error message to the Week 2 practical.

Check yourself

Try these without looking back. The answers are at the end.

1. Which of the four goals for AI does the Turing test measure? Give one weakness of the test.
2. Write a PEAS description for a robot lawnmower. Then suggest a badly chosen performance measure, and say what could go wrong.
3. Which kind of agent (Section 6) is each of these: a smoke alarm; a sat-nav; a robot vacuum that remembers which rooms it has already cleaned; a spam filter that is retrained every week?
4. Use the six questions in Section 5 to describe the environment of a program that plays poker.
5. A bank wants to decide who gets a loan. Describe a symbolic approach and a statistical approach, and give one advantage of each.
6. Put these in order, earliest first: Deep Blue beats Kasparov; the Dartmouth workshop; AlexNet; the Lighthill Report; ELIZA.
7. What is an AI winter? Give one reason for each of the two winters.
8. Why is unloading any dishwasher harder for AI than beating the world chess champion?
9. A sat-nav company says its new AI "saves drivers 20% of their time". What would you ask before believing it?

Key terms

Artificial intelligence	Getting computers to do tasks that would need intelligence if people did them.
AI effect	Once a computer can do something reliably, people stop calling it AI.
Turing test	An interrogator chats with a hidden person and a hidden machine; the machine passes if it cannot be told apart.
Cognitive science	The study of how people think, from psychology and the brain.
Agent	An entity that perceives its environment and acts upon that environment.
Sensors / actuators	How an agent senses its environment / how it acts on it.

Environment / percept / percept sequence	The world the agent is in / what it senses at one moment / everything it has sensed so far.
Agent function / agent program	What the agent does for every possible percept sequence / the code that produces it.
Performance measure	How an agent is scored; chosen by people.
Rational agent	An agent that chooses the action it expects to give the best score, given what it has sensed and what it knows.
PEAS	Performance measure, Environment, Actuators, Sensors: a description of an agent's task.
Fully / partially observable	The agent can / cannot sense everything that matters.
Deterministic / stochastic	An action always has the same result / chance plays a part.
Episodic / sequential	Each decision stands alone / decisions affect later ones.
Static / dynamic	The world waits while the agent thinks / it keeps changing.
Discrete / continuous	A fixed number of situations and choices / values such as speed can be anything in a range.
Single agent / multi-agent	The agent works alone / other agents, such as opponents or other drivers, affect it.
Simple reflex agent	Chooses with if-then rules, using only what it senses now.
Model-based agent	Also keeps track of the parts of the world it cannot sense now.
Goal-based agent	Looks ahead for actions that reach its goal.
Utility-based agent / utility	Picks the action with the best expected score / that score for an outcome.
Learning agent	Improves its behaviour using feedback on how well it is doing.
Symbolic AI	AI that works with facts, rules and symbols that people write down.
Statistical AI	AI that learns patterns from data and works with probabilities.
Neural network	A program made of many simple connected units, loosely based on brain cells, that learns from examples. The perceptron was an early one.
Expert system	Software that holds rules written down from human experts.
AI winter	A period when funding and interest in AI fell after promises were not met (mid-1970s and late 1980s).
Large language model	A very large neural network trained on huge amounts of text, such as the model behind ChatGPT.
Moravec's paradox	Tasks that are easy for people, like moving and handling objects, are often hard for AI, and the reverse.
King Midas problem	An AI system does what we asked, not what we meant.
Algorithm	A method written down step by step, precisely enough for a computer to follow.

Reading and next week

- Russell & Norvig, *Artificial Intelligence: A Modern Approach* (4th ed.): Chapter 1, sections 1.1 to 1.5 (what AI is, its foundations, its history, the state of the art, and risks and benefits), and Chapter 2, sections 2.1 to 2.4 (agents and environments, rationality, PEAS and environment types, kinds of agent).
- Poole & Mackworth, *Artificial Intelligence: Foundations of Computational Agents* (3rd ed., free at artint.info): sections 1.1 to 1.3 (what AI is, its history, agents in environments), 1.5 (the agent design space, close to the six questions in Section 5) and 1.7 (social impact).

Next week: search. You will describe a problem as a search problem, and learn depth-first and breadth-first search. One of them is probably the maze method you wrote in the seminar. Then we look at games, where an opponent fights back.

Answers to Check yourself

1. **Acting like people.** The machine only has to behave like a person in a text chat; nobody checks how it thinks. Weaknesses: it rewards imitation rather than intelligence (ELIZA fooled some people with simple tricks); the result depends on who the interrogator is, so two runs can disagree; and a machine could be very intelligent at other things and still fail.
2. **P:** the share of the lawn kept at the right length, with no damage to flower beds or pets, and the battery never flat. **E:** the lawn and its edges, flower beds, weather, people and pets. **A:** wheels, the cutting blade, the connection to its dock. **S:** bump sensors, a boundary wire or GPS, a rain sensor, its battery level. A bad performance measure is “time spent cutting”: the mower could drive round the same patch all day and score well while the rest of the lawn grows.
3. **Smoke alarm:** a simple reflex agent (if there is smoke, sound the alarm). It is an agent but not intelligent: it never compares options, plans or learns. **Sat-nav:** goal-based, because it looks ahead for actions that reach the destination; utility-based if it also weighs time against fuel or tolls. **Vacuum that remembers:** model-based, because it keeps track of what it cannot see now. **Retrained spam filter:** a learning agent.
4. **Partially observable** (the other players’ cards are hidden); **stochastic** (the cards are dealt at random); **sequential** (each bet affects the rest of the hand); **static** (the game waits while the program decides); **discrete** (a fixed set of cards and bets); **multi-agent** (the other players).
5. **Symbolic:** loan officers write rules, such as “if income is over three times the repayments and there are no missed payments in the last year, then approve”. Advantage: easy to explain a decision and to check that the rules are fair. **Statistical:** train a model on thousands of past loans, marked as repaid or not, and give each applicant a probability of repaying. Advantage: it can find patterns nobody thought of, and it adapts when retrained on new data. (Risks: rules go out of date; the model can learn past unfairness and is hard to explain.)
6. Dartmouth workshop (1956), ELIZA (1966), Lighthill Report (1973), Deep Blue (1997), AlexNet (2012).

7. A period when funding and interest in AI fell, because results did not match the promises. **First winter (mid-1970s)**: methods that solved small puzzles failed on real problems, because the number of possibilities grows too fast; the Lighthill Report (1973) said so, and funding was cut. **Second winter (late 1980s)**: expert systems were expensive to build and keep up to date, and failed on cases their rules did not cover.
8. Chess has fixed rules, a small board and a clear goal, so a computer can search through moves very fast. A dishwasher needs a robot to see and grip many different objects (wet glasses, knives, odd-shaped pans) in a kitchen it has never seen, and to know where each thing goes. People find this easy because evolution has trained us for it over millions of years. This is Moravec's paradox.
9. The two questions from Section 11. **What can it actually do?** Does it choose routes, or only predict traffic? Does it work everywhere, or only in some cities? **How do we know?** Saves 20% compared with what: another sat-nav, or no sat-nav at all? On which journeys, and at what times of day? Who measured it: the company or someone independent? And does any other road (say, residential streets) pay the price for the time saved?

COMP5001 Artificial Intelligence, University of Winchester, 2026/27 · Dr Sakib Anwar ·

Sakib.Anwar@winchester.ac.uk

Written for COMP5001, following the Week 1 lecture and Russell and Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Chapters 1 and 2. The definition of an agent and the list of topics come from CS50's *Introduction to Artificial Intelligence with Python* by Brian Yu and David J. Malan, Harvard University, licensed under CC BY-NC-SA 4.0. Shared under the same licence.