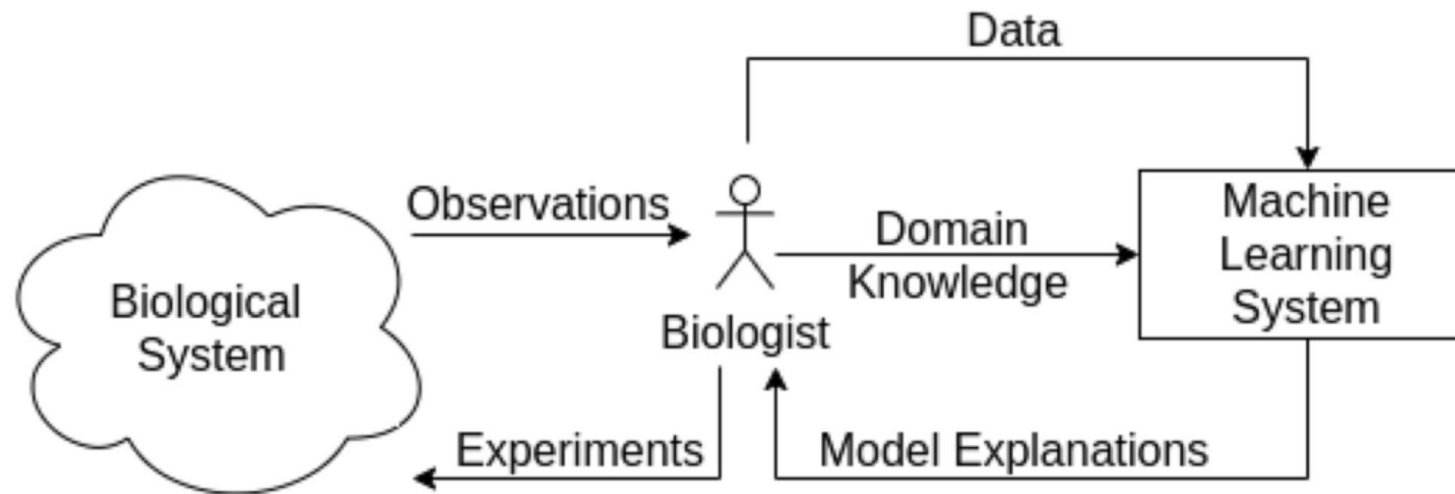


Expert systems

Expert system

1. Domain knowledge
2. Act under rules
3. Problem solving
4. Decision making
5. High performance

Domain knowledge



Feed back


Chain-of-thoughts

- **Practical challenge DNN:** Deep network structures constrained via logical domain-knowledge may not always be scalable to large datasets.
- The models based on attention raises the possibility of incorporating domain-knowledge through conversations.
- More learning : <https://www.nature.com/articles/s41598-021-04590-0#Fig1>


Rule-based

- knowledge base
- Database
- inference engine
- User interface
- Working memory

Decision making





 **You**

A farmer with a wolf, a goat, and a cabbage must cross a river by boat. The boat can carry only the farmer and a single item. If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage. How can they cross the river without anything being eaten? Describe your reasoning step by step.

 **ChatGPT**

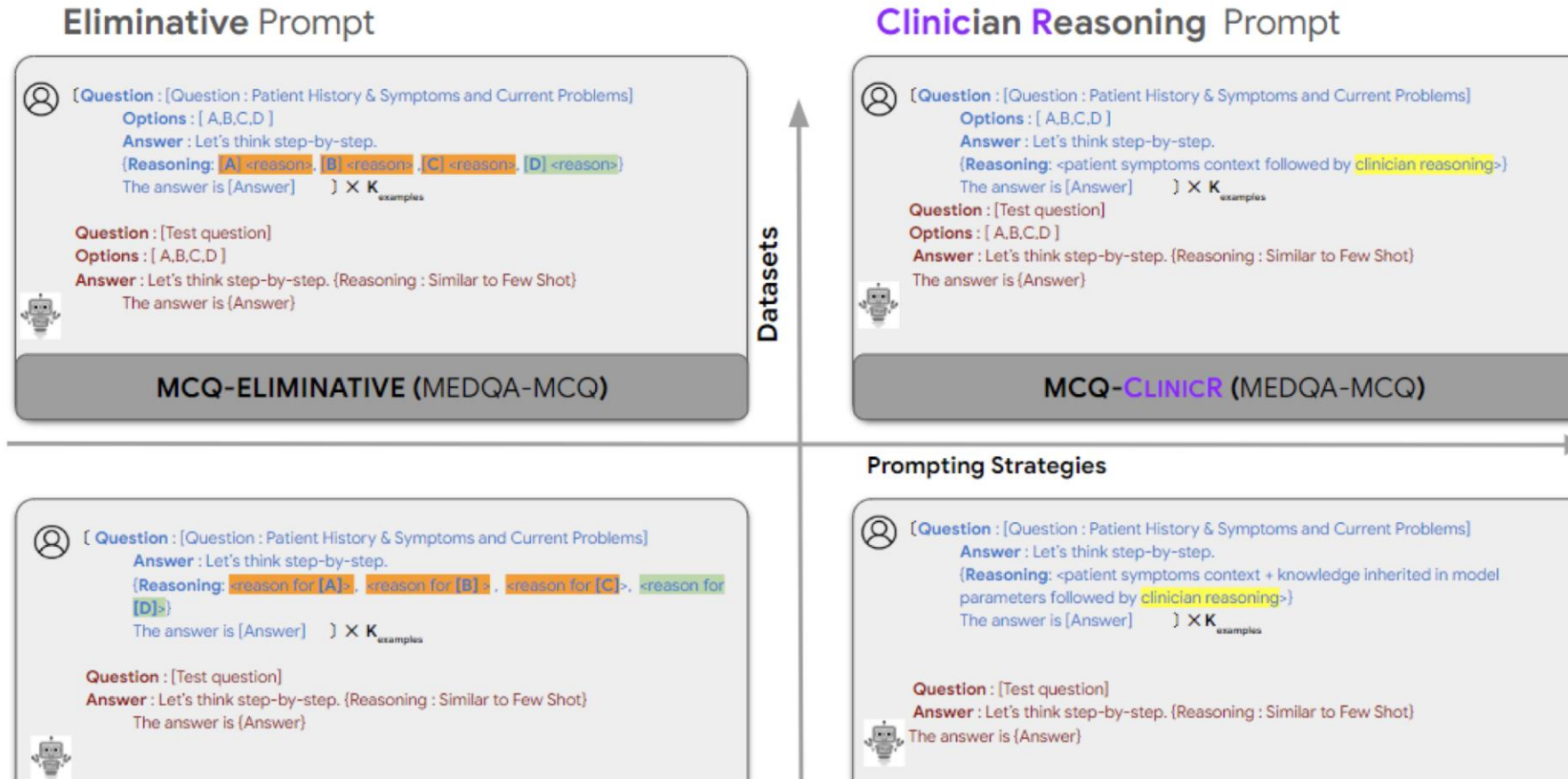
1. **First crossing:** The farmer takes the goat across the river and leaves it on the other side. This move ensures that the goat is safe from the wolf and the cabbage is safe from the goat because the farmer is with the goat and the wolf and cabbage are left behind. **State:** Farmer and goat on the new side; wolf and cabbage on the original side.
2. **Second crossing:** The farmer returns alone to the original side to pick up either the wolf or the cabbage. Let's say he takes the wolf. This keeps the cabbage safe from the goat since the goat is on the opposite side. **State:** Farmer and wolf on the new side; goat on the new side; cabbage on the original side.
3. **Third crossing:** The farmer leaves the wolf on the new side and takes the goat back to the original side. This action is critical because it prevents the wolf from eating the goat when the farmer is not there to supervise. **State:** Farmer and goat on the original side; wolf on the new side; cabbage on the original side.
4. **Fourth crossing:** The farmer then takes the cabbage across the river, leaving the goat on the original side. Now, the cabbage is on the new side with the wolf, and neither can harm the other. **State:** Farmer and cabbage on the new side; wolf on the new side; goat on the original side.
5. **Fifth crossing:** Finally, the farmer returns alone to the original side to pick up the goat. Bringing the goat across the river ensures that all are safely on the new side and none can harm the other. **Final state:** Farmer, wolf, goat, and cabbage all on the new side.

This sequence ensures that the wolf is never left with the goat without the farmer's supervision, and the goat is never left with the cabbage without the farmer's supervision, ensuring safe passage for all across the river.

chain-of-thought
prompting

Decision making



<https://arxiv.org/pdf/2403.04890>

High performance

A dominant part of the design of expert systems entails translating an expert's knowledge and experience into a form that the expert system can digest. Knowledge engineers have recognized this process as a serious bottleneck that slows expert system development.

1996<https://cdn.aaai.org/SIGMAN/1996/SIGMAN96-004.pdf>

- Minicomputers, workstations, mainframes.
- LISt Programming (LISP) and PROgrammation en LOGique (PROLOG).
- Large databases.

What is an Expert System?

- An Expert System (ES) is software that attempts to reproduce the performance of one or more human experts, typically in a specific problem domain.
- ES employs human knowledge represented in a computer to solve problems that ordinarily require human expertise.
- ES imitate the expert's reasoning processes to solve specific problems.

Why should we Use Expert System?

Expert Systems:

- Capture and preserve irreplaceable human expertise
- Provide expertise needed at a number of locations at the same time or in a hostile environment that is dangerous to human health
- Provide unemotional objective solutions faster than human experts
- Provide expertise that is expensive or rare
- Share human expertise with a large number of people

History of Expert Systems

In 1959 Newell and Simon described General Problem Solver (GPS).

GPS:

- Intended to solve general problems across domains. Example: theorem proof, geometric problem and chess playing.
- First computer program to separate knowledge from strategy (generic solver engine).
- Predecessor to ES.

History of Expert Systems

- ES introduced by Stanford Heuristic Programming Project led by Feigenbaum.
- Mid-1960s: Early ES programs:
 - MYCIN:
 - Diagnose infectious diseases such as bacteremia and meningitis.
 - Recommend antibiotics.
 - Dosage adjusted for patient's body weight.
 - Name derived from antibiotics (suffix – “mycin”).

History of Expert Systems

Mid-1970s:

–Recognition of the role of knowledge

- Power of an ES comes from the specific knowledge it possesses, not from the inference schemes it employs.
- Development of knowledge representation theories.
- Development of decision making procedures and inferences.

Design an expert system

- Choosing a problem
- Knowledge Engineering
- Expert system architecture
- Problem solving methods
 - Forward and backward chaining
 - Differential diagnosis
 - Rules

Applications of Expert Systems

Generic Categories of Expert System Applications

- Classification - identify an object based on stated characteristics
- Diagnosis Systems - infer malfunction or disease from observable data
- Monitoring - compare data from a continually observed system to prescribe behavior
- Process Control - control a physical process based on monitoring
- Design - configure a system according to specifications
- Scheduling & Planning - develop or modify a plan of action
- Generation of Options - generate alternative solutions to a problem