Knowledge Based AI Systems

Ramón Béjar

Universitat de Lleida

Modelling AI Problems

Many problems in AI can be modelled as a search problem:

- The AI system works with a model of the problem to solve, as a set of different sets. At a given time the system will be on one of these states.
- The system can change the current state by **executing** some action.
- Solving the problem means finding a sequence of steps that allow the AI system to reach some target state.

However, in some problems, there is no such target state, the AI system has only to answer in some way for each input state provided. So, the system works always from the given input state.

Using Classical AI Search Algorithms

Assumptions behind classical-search (DFS, A*, ...) based AI systems:

1. Every state s belongs to a finite set of states S.
2. We know **for sure** where we start (initial state): s_0
3. For every possible state (s), we know **for sure** what actions can be executed $(A(s))$
4. For every action $a \in A(s)$, we know **for sure** the resulting state after its execution
5. The set of goal sets $G \subseteq S$ is well defined, i.e. it can be recognized with an algorithm.

Using Classical AI Search Algorithms

These assumptions are fair enough when working on off-line scenarios, that can be **completely** simulated with no uncertainty on the information used

Tipically, these assumptions are true when working on:

- There is complete information about the current state and the effect of every possible action
- for any state s and a possible action a from s, there is a well determined resulting state s':

$$f(s, a) \rightarrow s'$$

Using Classical AI Search Algorithms

About the initial state:
Suppose we start on a situation where we only know for sure some properties of the starting state s_0, but not all the relevant properties

However, the information we have available may allow to infer that we can be only on a state from a particular subset of S:

$$s_0 \in S_0 = \{s_{i_1}, s_{i_2}, \ldots, s_{i_n}\}$$

The **more** we can infer from the initial information, the smaller will be the set of possible initial states S_0

Using Classical AI Search Algorithms

About the possible actions to execute:
If there is no complete information about the current state s, how can we be sure about all the actions $A(s)$ we can execute ?

However, the information we have available may allow us to infer that some actions $A'(s) \subseteq A(s)$ can be executed

The **more** we can infer from the information we have about the current state, the **closer** will be the set $A'(s)$ to the whole set of actions $A(s)$

Using Classical AI Search Algorithms

About the effect of an action:
If there is no complete information about the current state s, knowing that some action a can be executed from s, may not be enough to know for sure the resulting state

That is, the effect of executing action a from a state s for which we do not know all the properties, may lead as a result a set of possible resulting states: $\{s_{j_1}, s_{j_2}, \ldots, s_{i_n}\}$.

Using Classical AI Search Algorithms

We cannot always define explicitely goal states for any AI system

1. Some times we know the properties we want to acheive, but we do not know explicitely all the states that satisfy these properties
2. However, given some properties of the current state and an action, we may know if the resulting state would satisfy the desired properties

Using Classical AI Search Algorithms

Even more, on some AI systems the goal is to execute, at every step, some "valid" action that keeps the system in a safe state

Curiosity Mars Rover:

Suppose the Rover knows at the current state:

- There is an obstacle in front of the Rover
- There is a martian to the left of the Rover. We do not know if he is friendly or not
- There is a person to the right of the Rover. Our Ontology Reasoner identifies that person as the famous "Jose Luis Barcenas"

The inmediate goal is:
Select an action that will keep the Rover in a safe place

Using Classical AI Search Algorithms

Some times, the goal of the system is to maximize some cumulative reward obtained after executing a sequence of actions

The Broker Agent:

Suppose an (artificial) Broker Agent knows:

- The current value of stock options on the market
- Some evidences about next operations of some companies
- What other broker agents have done in the past

The long-term goal is:
Achieve a maximum cumulative reward, after executing a sequence of sell / buy operations in some time horizon

This is an example in which usually the agent has very little knowledge about the whole situation, and how it will evolve in the future

So, the **basic** problem we want to address is:

Given that we have:

1. A basic knowledge in which we trust at any time
2. Some information that is true at the current time

Can we discover:

1. Any additional information that must also be true at the current time ?
2. Is there any action we can execute so that a desired property will hold true ?

Knowledge Based Systems - Answer Systems

A similar problem arises for AI Systems such that the state does not change, but the information provided to the system can be modified during the lifetime of the system

A Medical Diagnosis System:

Given a KW base with medical diagnosis knowledge about a set of medical deseases

1. Given a set of initial medical evidences (sintoms), can we infer what is the desease of the patient ?
2. If we gather new evidence in the future (more clinical tests are performed), could the conclusion of the system change ?

Talking with Eliza

In 1965, Joseph Veizenbaum implemented an AI system, called Eliza, that caused quite a sensation in his time. Eliza was explained in this paper:

> *"ELIZA A Computer Program For the Study of Natural Language Communication Between Man And Machine"*

Talking with Eliza

From the Introduction of the Eliza paper:

> *"But once a particular program is **unmasked**, once its inner workings are explained in language sufficiently plain to induce understanding, its magic crumbles away; it stands revealed as a mere collection of procedures, each quite comprehensible."*

It uses a set of inference or transformation rules:

For every question, it uses a single rule to provide an answer

Talking with Eliza

Eliza allowed **certain kinds** of *natural language* conversation between a man and a computer.

Conversation example

C: How do you do. please tell me your problem
P: I am desperate
C: Is it because you are desperate that you came to me
P: yes
C: you seem quite positive
P: Men are all alike
C: In what way
P: Are you listening me?
C: Why are you interested in whether I am listening you or not
...

Under the hood of Eliza

To allow different kinds of conversation, ELIZA uses a set of rules that follow the following general idea:

> If query MATCHES a certain pattern then
>   TRANSFORM the query to generate an answer

The rule to use is determined by the existence of certain keywords in the query of the person.
If different keywords are present in the same query, a preference order is used.

Under the hood of Eliza - Conversation Example

P: I am desperate
C: Is it because you are desperate
that you came to me

rule used →

keyword: I
pattern found: I am X
answer: Is it because you are X
that you came to me

P: yes
C: you seem quite positive

rule used →

keyword: yes,
pattern found: yes
answer: you seem quite positive

Under the hood of Eliza - Conversation Example

P: Men are all alike
C: In what way

rule used →

keyword: alike
pattern found: X
answer: In what way

P: Are you listening me?
C: Why are you interested in whether I am listening you or not

rule used →

keyword: are
pattern found: are you X
answer: Why are you interested in whether I am X or not

note: ”me” is always transformed to ”you”

Eliza and the Turing Test

Eliza was probably the first attemp to build an AI system that was able ”to chat” with a person in a very ”human-like” style.

What is Eliza really doing?

- Is Eliza thinking ?
- Or is she cheating ? (it makes you believe it understands what you are saying?)

At least, answers were syntactically correct.

But one would say it would not pass a ”Turing Test”.

Two possible assumptions:

Perfect knowledge

1. All the information on our KW base may be necessarily true at any state
2. All the information we provide to our system is true (at the given time)
3. The system never receives contradictory information

No perfect knowledge

1. Not all the information on our KW base may be necessarily true at any state
2. Not all the information we provide to our system may be necessarily true
3. Sometimes, we may collect contradictory information, or even evidences that may leed to contradictory conclusions

Inference and Learning with Perfect Information

Representation of knowledge and any provided information with logical languages (CP0 or CP1)

Discovery of new information with inference algorithms:

Perfect information query problem:

Given:

1. The knowledge on the system
2. Evidence from the outside
3. Query property Q

Is the answer to Q necessarily true given (1) and (2) ?

Inference and Learning with No Perfect Information

The set of all properties that can be true or false, will be associated with a joint probability distribution

No perfect information query problem:

Given:

1. Knowledge used for discovering more information: belief networks
2. Evidence from the outside
3. Query property Q

Can we discover the most probable answer for a query Q ?

A belief network will allow to infer the probability of a given property Q being true (or false), given the collected evidence from the outside

Knowledge Is Power ?

One limitation of the knowledge based systems we want to study, is that they are focused towards answering very concrete questions, where the information needed to answer them is needed to be encoded directly into the system.

Imagine a possible AI system targeted towards answering questions in a very broad domain of knowledge, and where questions do not follow a predetermined format.

Knowledge Is Power ?

For example, consider this already famous query, comming from a research paper 'Logic and the complexity of reasoning':

Could a crocodile run a steeplechase ?

What knowledge should we incorporate into a system to answer such kind of questions ? How should this information be represented into the system ?

Knowledge Is Power ?

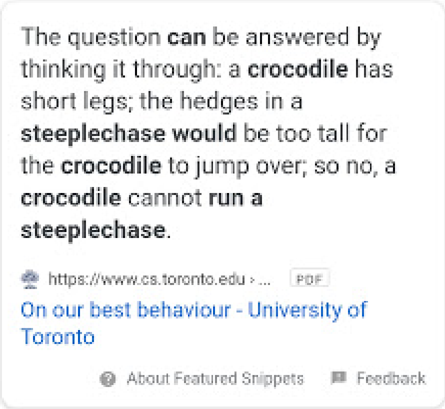
This question can be seen as one example of a common sense reasoning.

Could a crocodile run a steeplechase ?

What knowledge should we incorporate into a system to answer such kind of questions ? How should this information be represented into the system ?

Knowledge Is Power ? - Google Assistant and Ontologies

This question can be seen as one example of a common sense reasoning. How does google assistant resolve this question ?



Knowledge Is Power ? - Google Assistant and Ontologies

In this case, it could be that google is using the knowledge in his knowledge graph (a kind of ontology), to answering the question using properties of cocodriles and steeplechases. Read the google blog for a nice introduction:

https://www.blog.google/

Or for a more technical description of the original ontology that google uses for his knowledge graph, here you have a very complete paper:

https://arxiv.org/pdf/1805.03885.pdf

Knowledge Is Power ? - Google Assistant and Ontologies

However, checking carefully the answer provided by google, we discover that the answer is a **literal** copy of the answer provided in the research paper they link to:

'On Our Best behaviour', by Hector J. Levesque

So, the reasoning by google was not as smart as we could think at first sight

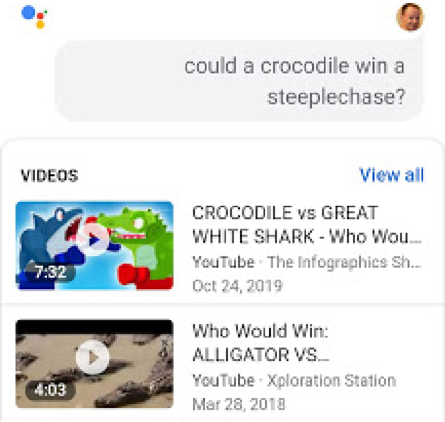
Knowledge Is Power ? - Google Assistant and Ontologies

But, what if we change a little bit the question:

Could a crocodile win a steeplechase ?

Then, it seems to be a completely different question for google...

Knowledge Is Power ? - Google Assistant and Ontologies



Knowledge Is Power ?

To know more about the challenges of common sense reasoning, read the paper that google uses to give us an answer to the question:

'On Our Best behaviour', by Hector J. Levesque

https://www.cs.toronto.edu/∼hector/Papers/ijcai-13-paper.pdf

They propose the use of questions like the ones from the Winograd benchmark set.

Language Models based on Deep Neural Networks

Recently, there has been a big success for answering general questions using a language model based on HUGE deep neural networks (more than 175 BILLIONS of conexions between neurons distributed in 96 layers)

The model gives you the most probable next word, given a sequence of previous words:

$$f(\text{Could a crocodile run a steeplechase ?}) \rightarrow \text{yes, no ?}$$

Longer answers are obtained by making new questions to the system, appending the previously generated words to the input.

Language Models based on Deep Neural Networks

But such models, like GPT-3, are trained (learning of its set of parameters) using a HUGE dataset of texts, around 45TB of compressed plaintext obtained from the WWW

The idea of GPT-3 is to have a general purpose language model, able to answers queries from different domains.

Are General Language Models good in any task?

It is enough to learn with a HUGE general data set to be able to get a general AI system that can answer queries of any domain?

1. Scientific/technical questions
2. finding bugs in programs
3. Art/Literature
4. ...

Are General Language Models good in any task?

If we consider some questions from the Winograd benchmark (related to common sense reasoning), some of them are well answered by models like GPT-3 and ChatGPT:

The trophy and the suitcase:

Query: The trophy would not fit in the brown suitcase because it was so small. What was so small?
Answer: The brown suitcase was so small.

GOOD !

But the data set used for learning is contaminated with specific data with questions and right answers from the Winograd test!

Are General Language Models good in any task?

But there exists some other benchmarks that have been checked to not contain questions that can be answered thanks to a strong bias towards the right answer in the WWW, and in that case models like GPT-3 fail to get a good success rate.

The trophy and the suitcase:

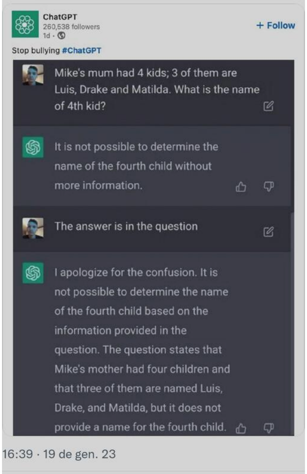
Query: The trophy would not fit in the brown suitcase despite the fact that it was so small. What was so small?
Answer: The brown suitcase was so small despite the fact that the trophy would not fit in it.

WRONG !

Are General Language Models good in any task?

It can also fail on questions that for us (humans) seem to be trivial (no need to read the whole WWW to be able to answer it)

So, not always more knowledge is more (intelligent) power...



Are General Language Models good in any task?

What about scientific knowledge?

LF What's a fun fact about Lance Fortnow?

I'm sorry, I don't know who Lance Fortnow is and I also don't have any fun fact about him. He is not a widely known person. Can you please provide me with more information about him so I can assist you better?

MO what about papers of Lance Fortnow

Lance Fortnow is a computer scientist and complexity theorist who has published many papers on computational complexity and its related topics. Here are a few examples of his papers:

1. "The Status of the P vs. NP Problem" (2009)
2. "The Golden Ticket: P, NP, and the Search for the Impossible" (2010)
3. "Time-Space Lower Bounds for Satisfiability" (1990)
4. "Interactive Proofs and the Hardness of Approximating Optimization Problems" (1991)
5. "The Complexity of Formal Languages" (1989)
6. "The Complexity of Perfect Information Equilibrium" (1994)
7. "Resource-Bounded Measure and the P-NP Hierarchy" (1989)
8. "The Game of Life is PSPACE-complete" (1982)

You can find more of his papers by searching for them on Google Scholar or on his personal website, if you have access to it.

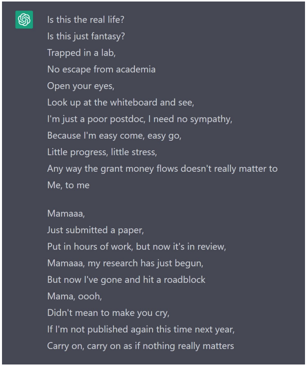
Answers given by such Language models can be inconsistent (contradictory)

Even more, in the second answer there are invented references !

Are General Language Models good in any task?

Can they be creative ?
That is, can they invent new content mixing/modyfing his HUGE learned knowledge ? It depends on how you define what is to be creative.

Query: full lyrics of bohemian rapsody + rewrite them to be the life of a postdoc



But we need to provide the full lyrics in the query as a very specific context