CSC 244/444 Sept 5/23

LECTURE 1: The Role of Logic in AI

Intelligent systems need large amounts of knowledge, and the ability to use that knowledge for reasoning and planned, purposeful action. Logic is an essential tool in the REPRESENTATION and SOUND USE of much of this knowledge.

The need for knowledge

The crucial dependence of intelligent behavior on having large amounts of knowledge is by now a truism in AI; it was not obvious at first. But no general problem solving strategy, search strategy or learning strategy *by itself* has taken us very far toward human-level problem solving and planning. LLMs can suggest plans and lines of argument based on analogy with the vast number of related tasks they have been exposed to, and DNNs can also provide remarkably good "intuitions" in game-laying situations and other narrow tasks. But they are unable to reason deeply and combinatorially, especially for novel problems, using the abundant, miscellaneous world knowledge encoded in their parameters.

Some contrasting examples:

1. Puzzle solving

- *In the simplified missionaries and cannibals problem, two missionaries and two cannibals must cross a river using a boat which can carry at most two people, under the constraint that, for both banks, if there are missionaries present on the bank, they cannot be outnumbered by cannibals (if they were, the cannibals would eat the missionaries). Apart from this, the cannibals will do what they're told by the missionaries, in order to accomplish the transportation*. Can you solve the simplified missionaries and cannibals problem? ChatGPT already got the missionaries eaten up on the second move. One might expect it to do better on the full, 3 missionaries & 3 cannibals problem, which is more familiar, but it went wrong in counting how many individuals were on each side after a few moves.
- *In the register-exchange problem, we have three registers, R1, R2, and R3, where R1 and R2 have different positive numbers in them and R3 is empty, and we want to interchange the numbers in registers R1 and R2. All you're allowed to do is to move the contents of one register to another register, among the three given registers*. Can you solve the register-exchange problem? ChatGPT solved this one, as this is a well-known little problem in computer science and AI, and ChatGPT has also seen many programming problems.

2. Mysteries

- *Someone has killed Aunt Agatha in Dreadbury Mansion. The only people who were recently in Dreadbury Mansion, besides Agatha herself, were her butler and her cook. Both her butler and her cook are Quakers. Who most likely killed Agatha*? Because ChatGPT fails to draw on information it has about Quakers, it claims there isn't enough information. Hinting that Quakers are pacifists, and even telling ChatGPT that the butler and the cook would never harm anyone, still didn't lead ChatGPT to the likely answer. It wanted more specific evidence.

- *Another little puzzle: Either Bonnie or Clyde shot Albert. Bonnie had no access to a gun, so who shot Albert?* ChatGPT figured this out, as the required reasoning is minimal. (Viz.: If you don't have access to a gun, you don't have a gun; and if you don't have a gun, you can't shoot.)

3. Story understanding

This is LLMs' forte. Still, if alternatives are involved, they may struggle.

- *Alicia drove back home from her holiday visit to the US*. What borders did she cross? Where might she live? how do we infer this? Because this is an unfamiliar sort of problem, ChatGPT struggles to get it right. Even with much prompting, it didn't see the disjunctive answer.
- *When Randy learned that he had terminal metastatic cancer, he decided against further chemotherapy and to enter a hospice program for palliative care*. (Explain his decision.) Because terminal cancer and palliative care are much discussed on the web, and follow familiar patterns, ChatGPT does well on this.

For familiar fairy tales, such as the story of Little Red Riding Hood, ChatGPT does particularly well. Chung Hee Hwang and I did some detailed computational studies of examples from that story. One was concerned with the part of the story just after the point where the wolf has knocked on the door of grandmother's cottage, pretending to be Little Red Riding Hood:

> The worthy grandmother was in bed, not being very well, and cried out to him, "Pull out the peg and the latch will fall."
>
> The wolf drew out the peg and the door flew open. Then he sprang upon the poor old lady and ate her up in less than no time, for he had been more than three days without food.

In a detailed analysis of the last passage (see Hwang and Schubert in Minds and Machines 3(4), Nov.93, or in Iwanska & Shapiro's *Natural Language Processing and Knowledge Representation*, 2000, or http://www.cs.rochester.edu/~schubert/papers/el-meets-lrrh.pdf), the following facts, among quite a few others, were found to be necessary for a *reasoned* understanding of the passage. Essentially what a "reasoned understanding" means here is figuring out from general and specific knowledge about the world and about the story how the described events and states "hang together" in a causally coherent way, including agents' goals, intentions, beliefs, & behavioral dispositions:

- Causation is transitive (e.g., pulling out the peg causes the latch to fall, and hence the door to open)
- (Roughly) whatever a part of an event causes, the event as a whole causes as well
- If the conjunction of two facts holds because of a third fact, then each of the conjuncts holds because of the third fact
- If something flies open then it opens very quickly
- If something is "very P" then it is P
- If a person does a certain kind of action, and believes that this kind of action is a possible way to bring about a certain kind of event that is harmless to them, then they may ***intend*** that their action bring about that kind of event.
- If someone wants the door of a room or house to open while they are outside of it, that's probably because they want to enter it.

- If one enters a room, one will then be inside the room.
- Individuals who are in the same room are near each other.

- Eating requires food
- Creatures are very hungry when they have not eaten for more than a day
- Two successive event sentences in a narrative usually indicate that the two events happened in succession
- etc., etc

But because this is a familiar story (and abstractly resembles other fairy tales), ChatGPT does remarkably well just by analogy. This came as a big surprise, and suggests that perhaps much of our own story understanding and language understanding more generally is based much more on experience with stories and language, than on reasoning.

3. *Medical diagnosis*

We should first note that tasks like interpreting X-ray images, where *specialized supervised training on large datasets* can be performed, are more and more being supported effectively by AI systems.

However, when it comes to general medical diagnosis, treatment and prognosis, such as might be performed by an internist, LLMs can only provide lists of possibilities that may be helpful to the diagnostician (or lay person), but they can't be trusted in ranking likelihoods or dealing with very unusual cases; indeed, traditional AI methods using on Bayesian networks or rule-based systems are more trustworthy.

In general, the amount of medical knowledge that needs to be taken into account in diagnosis, treatment and prognosis is enormous. Have a look at some medical text, such as Krupp et al., *Current Medical Diagnosis and Treatment*! LLMs have been exposed to a lot of medical texts, and though they seem to answer confidently about quite technical questions, they will hallucinate answers where their exposure is sparse, and doctors so far certainly don't trust them.

The same goes for any other technical field or branch of mathematics – especially the latter, where combinatorial reasoning (i.e., exploring a large space of possibilities) is essential.

People know at least many millions of particular and general facts, such as the following

- the names, personal characteristics (gender, approximate age, general appearance, interests, skills, occupations, personal history, etc.) of many family members, friends, associates, and public figures
- general facts about people (appearance, body-parts, that they eat and breathe and sleep, how they come into being, mature, age, and die, ***how they think and feel*****,** how they interact socially, or in education, in business and commerce, etc.)
- where they live, what their daily routine is, what their belongings are, their particular characteristics and where they are located
- general facts about classes of natural objects, artifacts and substances such as what dogs look like, that they bark, are generally friendly domestic animals, have fur, are mammals; that pieces of writing paper are flat and rectangular and of a certain size, and can be written, typed or drawn on, folded, crumpled, burnt, etc.; that parking lots typically have large paved level surfaces, and markings indicating permissible locations of cars, etc

- particular and general facts about history, movies, sports, politics, music, crime, math, science, literature, etc.
- What sorts of situations and events are apt to occur in the world -- dogs barking, traffic moving on roads, people working at jobs, people eating, sleeping, opening doors, chatting with family, friends, co- workers, watching TV, etc., the patterns of day/night and seasons, clouds moving overhead, the sun and other light sources casting shadows, etc., etc. -- we appear to know tens of millions of such ***often-encountered situations and patterns of events***.
- particular and general facts about words and their meanings
- etc., etc.

People can ***state*** these facts in ordinary language, which suggests that they are available in an internal language (what cognitive scientists have called “mentalese”), at least at an abstract level. And they can ***use*** them for reasoning and action. I use the term "reasoning" very generally, especially to include such things as

- deducing logical consequences of given facts;
- figuring out a plan to achieve a goal;
- inductively forming theories – general concepts and relationships -- from a given set of facts, based on their shared features.

Perhaps at the intersection of analogy and reasoning, we have skills like

- engaging in familiar patterns of behavior & interaction (chatting, dining,...)
- inferring causal connections between what we see or are told;
- inferring explanations for given events or situations;
- predicting what will happen in a given situation
- recognizing certain types of situations and events as (un)familiar;
- figuring out what a speaker meant or intended;

If we are going to endow machines with human-like intelligence, they will have to acquire an amount and variety of knowledge comparable to that of people. To the extent that LLMs seem to possess a great deal of this knowledge, the question is how it can be brought “to the surface” in a form where it can be used *rationally* for the more demanding, more combinatorial types of reasoning and planning. And what should that form be, supporting reasoning steps that we can identify as being rational? This is where we can use logic as a guide.

While the goal of endowing machines with commonsense knowledge and reasoning abilities is the most central in AI, we should also note the need for representing formal mathematical and technical knowledge and reasoning. For example, it is a long-standing dream to build an automated mathematician or mathematician's assistant that would greatly boost progress in mathematics, not only helping with very complex proofs but also formulating mathematical concepts that are natural and relevant in a particular subdomain. Similarly we would like reasoners that can design or verify complex circuits or programs or architectural structures, or help with the complex mathematics, reasoning, and concept formation involved in attempting to unify quantum mechanics and general relativity.

The road to knowledge and commonsense reasoning

The ***knowledge acquisition bottleneck*** was already alluded to in "lecture 0"; it refers to the difficulty of doing the above. It may appear that the problem has been solved by LLMs, but as

noted repeatedly, the knowledge in these systems is not in a form where it can support nontrivial, combinatory reasoning and planning. How do we endow computers with this vast amount of knowledge in a form where it can be used for such reasoning? Some possible answers:

- *Knowledge mining from LLMs*: Using appropriate prompts, such as "*What are the major parts and major uses of a claw hammer?",* or *"What are the preconditions for picking up an object with your hand, how would you break the action down into simple steps, and what is the resultant situation brought about by the action?".* One problem is how to get out *all* the general knowledge of this type, as self-contained items, which could then be used in a separate inference engine. There's also the problem of integrating results derived by the inference engine with the analogical thinking the LLM performs, and indeed supporting the efforts of the inference engine with "guesswork" from the LLM.
- *Integrating a reasoner directly into an LLM*: We use prompts to "persuade" an LLM to work out solutions to logical problems and planning problems one step at a time, allowing if possible for hierarchical breakdown of problems into subproblems, with no *a prior* limit on depth or length of the solution attempt. Various attempts in this direction exist, such as chain-of-thought (CoT) prompting, tree-of-thought (ToT) prompting, and graph-of-thought (GoT) prompting. The problem is that the premises used in the reasoning are in a loose NL form, and the way they are combined is uncontrolled and unreliable. So this has severe limits in accuracy, and in the complexity of the problems to be solved.
- *Building a controllable reasoner that uses an LLM as a tool*: Suppose we have a reasoner that is able to rationally combine premises, and search in a space of possibilities towards a solution of a given problem, provided that the premises it starts with or is given along the way are in a form that enables the rational combination of premises. It would have a *working memory* in which to maintain currently relevant pieces of knowledge and (sub)goals, and would judge what would be a good subgoal to work on at any moment. It may decide to use a current fact in working memory to advance the chosen subgoal, or it may request "*some facts relevant to the subgoal*" from the LLM, would format the pieces of knowledge proffered by the LLM in the required inference-enabling form, and go from there, until the problem is solved. The problems here are,

    - ensuring that no hallucinated premises are offered by the LLM; this probably requires specific facts about specific entities involved in the reasoning to be drawn from a separate *episodic memory*; (the *general* facts offered by LLMs are relatively reliable);
    - conversion from NL-premises offered by the LLM to a structured inference-enabling form; further conversion to a vector representation to enable "analogical softening" of inferences if necessary;
    - somehow keeping the LLM informed of the status of the solution attempt (in particular, the premises and steps that lead up to the subgoal for which the reasoner is requesting relevant new information); this is needed for retrieving relevant facts; and
    - also using the LLM to make guesses about what subgoals might be most usefully pursued next, so as to guide the search through the combinatorial space of possible next steps. This is analogous to what happens in AlphaGo.)

- *Knowledge engineering:* Having considered three LLM-based approaches, let's also look back at the traditional approaches to breaking through the knowledge acquisition bottleneck. The goal of knowledge engineering is to find a general way to *represent* knowledge; have "knowledge engineers" code up the knowledge (e.g., Doug Lenat, CACM 38(11)); or acquire the knowledge *through language* (being told, reading), perception/physical exploration, and

through concept formation and generalization learning. A special case of coding up knowledge is what the Big Tech companies have been doing: They are code factories employing tens of thousands of coders to engineer “skills” that make their products, such as Alexa or Siri, marketable. Their goal till recently has not been to find a well-founded basis for general understanding and problem solving; But the future for them lies in combining general language understanding with those engineered skills. However, their thinking is more like bullet 2 above, where they also enable an LLM to call up plug-ins to find solutions to problems for which they have well-engineered code to solve them. This doesn’t help with combinatory reasoning and planning using miscellaneous world knowledge.

- *Rule-based systems:* Build mechanisms for making inferences, basically by pattern-matching: we use rules that say if you have a set of facts matching such and such patterns (i.e., having a certain form), then you can draw such-and-such a conclusion (having a certain form based on the input facts). E.g., two such rules might be (one is quite generic and "sound", the other not so much):

Given: [fact1 or fact2], not(fact1)	Given: bird(x)
Conclude: fact2	Conjecture: flies(x)

- *Neobehaviorism:* Mount a big "behavioral engineering" effort; instead of worrying about representation and inference, focus on what an intelligent agent ***does,*** what its skills and modes of behavior are; build small subsystems to capture these skills and modes of behavior (using inspiration from natural systems, as well as from math, control theory, engineering, introspection, etc.); learn how to put them together so that the right behaviors are "in charge" at the right time, and they cooperate; layer them into ever larger, more ambitious systems, ...

- *Universal ML:* Build a very general learning machine, based as closely as possible on neural mechanisms and brain organization, with a built-in tendency to act in a way that optimizes "rewards" (positive feedback); embody it & place it in a stimulating, realistic learning environment; it should learn facts and inference methods implicitly, just like people (not like LLMs). E.g., see the Blue Brain project (now focused on the mouse brain)
  http://bluebrain.epfl.ch/cms/lang/en/pid/56882,
  or Chris Eliasmith's approach to building a brain (focused on neuromorphic engineering),
  http://arts.uwaterloo.ca/~celiasmi/
  The DNN aspects of Google's DeepMind (AlphaZero, AlphaFold, etc),
  https://en.wikipedia.org/wiki/DeepMind
  might also evolve toward trying to achieve more human-like learning and thinking by modeling more aspects of human brain function than LLMs do.

- *Simulated evolution/ genetic programming:* Build virtual creatures capable of rapid self-reproduction, and genetic change, letting many variants of them thrive and reproduce in some virtual environment that contains hazards and rewards... Intelligence and learning, including learning of facts, will come automatically as the artificial beings evolve...

My belief, of late, is that the approach in the third bullet above is the most promising; it also extends to autonomous agents with a continual planner/ executor “in control”. I doubt very much that we can engineer all the knowledge needed for intelligent problem solving, or that we can build a human-like agent just by engineering skills and behaviors, or that we can build the ultimate learning machine by borrowing more heavily from human brain structure and function, or by

simulating evolution. At the same time, it seems to me that we need explicit, structured knowledge representations as traditionally studied (and extended beyond these), in order to support trustworthy reasoning in the context of the bullet-3 approach (even if we “soften” these representations using vector embeddings of predicates and other terms).

Of course, there are also kinds of "how-to" knowledge that probably are best represented in some non-symbolic information processing mechanism:
- how the tune for "Twinkle, twinkle" goes
- how to interpret images falling on our retinas
- how to visualize a clown standing on a horse galloping along the seashore (etc.!)
- how to ride a bike, catch a ball, use chopsticks, tie shoelaces
- how to speak grammatically
- how to learn

Levels of description, declarativism, proceduralism

I think in discussions of these issues there is often a confusion about ***levels of description*** of a knowledge-based agent (see R & N, p152-2):

- the ***knowledge level***, or ***epistemological level:*** what an agent knows, and knows how to do

- the ***logical level:*** the actual form (syntax) and semantics of the representations used

- the ***implementation level:*** the code and data structures

For instance, when we talk about neural nets in terms of the layers of neurons, their interconnections, activation functions, and backpropagation, we are describing a (possible) agent largely at the implementation level. We will also have to learn to describe it at the logical and epistemological level before we can successfully engineer (trustworthy) reasoning or planning systems using neural nets.

Finally, a word about the distinction between ***declarative*** and ***procedural*** knowledge. Declarative knowledge consists of explicit, interpretable representations of facts, such as

(forall x) dog(x) => can-bark(x),
dog(Odie)

B & L emphasize that declarative representations are (i) transparent, (ii) modifiable, and (iii) potentially usable for many purposes. By contrast, procedural knowledge (e.g., how to multiply two numbers, invert a matrix, ... or catch a ball, ride a bicycle, or pronounce a word) is typically special-purpose, and computer code that does such tasks is hard to understand and modify, or use for a different purpose.

This is true, but I think the key distinction is not so much the declarative/procedural one as the distinction between representations that contain ***externally meaningless symbols*** and those that don't. For instance, consider the following two procedural representations of a routine that makes a robot say "hello" when a person comes near:

1. declare (G,string,"hello")
2. declare (F,boolean,nil)
3. declare (F',boolean,nil)
4. turn-on(proximity-sensor)

5. Repeat-at-intervals(1, sec):
   a. let F' := read(proximity-sensor)
   b. IF F' and (not F) THEN write-stream(voice-output,G)
   c. let F := F'.

1. Turn-on(proximity-sensor)
2. Repeat-at-intervals(1, sec):
   IF on(proximity-sensor, now) AND
      off(proximity-sensor,ago(1,sec))
   THEN utter("hello").

When we consider knowledge relevant to a particular domain, ***externally meaningless symbols*** are symbols that don't denote anything in that domain -- typically, they are parameters and operators concerned with ***internal*** book-keeping and data manipulation.

In the first version, the following symbols are externally meaningless:
declare, G, F, F', string, boolean, nil, let, :=, read,
write-stream, voice-output.
They denote nothing ***in the domain,*** and have only internal significance.

In other words, clarity, modifiability, and re-usability require that we ***abstract away*** from the level of internal data manipulation, specifying actions strictly at the level of domain entities, properties, and operators, as in the second procedure above.

NOTE: Cooking recipes and furniture assembly instructions can be just as understandable as statements of fact or stories -- yet they are procedural!

A comment on B & L's "Snow is white" example: Their first prolog program directly encodes that the correct response to the query "Color of snow?" is to print "Snow is white", while the second program bases its response on the FACT that snow is white, explicitly represented (along with other color knowledge) as a prolog clause. This nicely illustrates that basing "what to do" on factual knowledge (rather than simply behavioral rules) provides flexibility and generality. But observe that both programs still embody behavioral rules, and apart from the "!" operator ("don't do any goal-chaining on the remaining literals of the clause) are quite comprehensible at the domain level -- i.e., how to answer certain color questions. In that sense, the examples don't impugn procedural representations at all.

CSC 444: Comments on Brachman & Levesque's introduction

Their emphasis is on defining knowledge representation and reasoning, and justifying an approach to AI based on KR&R.

They define knowledge in terms of propositions that are known, believed, expected, hoped-for, etc., by an agent.

According to Brian Smith's "KR hypothesis", mechanized intelligence will require internal symbolic representations of propositions that are interpretable by the designer (and other observers), and that are the basis for the intelligent agent's operation.

Reasoning is the derivation of new propositions from ones explicitly represented. They give a nice example:

Suppose you are told that
    John is allergic to penicillin; and
    Anyone allergic to penicillin is also allergic to ampicillin.

We would want to conclude that John is allergic to ampicillin. Another example they give is more verbally oriented:

Suppose you are told that
    John loves Mary (and these are persons); and
    Mary is coming to John's party.

Then we would agree that "Someone that John loves is coming to his party".

One point they make is that it is hard to see how facts like those mentioned in the examples (or ones like "Nearly half of Peru's population lives in the Andes") could be committed to memory by an agent *without* some form of propositional representation. (But from an LLM-impacted perspective, these propositional representations may be hidden at an abstract level of a DNN.) Keep in mind that these facts could be used in combination with a variety of other facts to make inferences; and there seems to be no procedural representation that could make use of those facts in all situations.

The point could be further dramatized by imagining learning from a book. For example, suppose you read about the geography, history, and culture of Peru. After doing the reading (if you make some mental effort in the process!) you'll know quite a bit of what you've read, and could answer questions, plan a vacation that suits your tastes, make conjectures about the future of Peru, etc. Surely the simplest assumption to make about how this is possible is that we represent the memorized information in some symbolic internal language, at least at an abstract level, perhaps not far removed from language. (Since we use language to communicate knowledge, it would be surprising if the internal encoding of knowledge were vastly different from language at all levels of structural and functional abstraction.)

They relate the kinds of inferences mentioned above to ***logical entailment*** (and I think it's a good idea to introduce this concept so early in the book). They explain entailment in terms of imagining how the world would have to be if the given propositions are true. For example, in the case of the inference concerning penicillin and ampicillin, if we imagine a world in which John is allergic to penicillin, and everyone who is allergic to penicillin is also allergic to ampicillin, then in such a world John ***must*** be allergic to ampicillin.

They justify the use of logic in terms of entailment, pointing out that one way of looking at logic is as a theory of entailment relations that hold between symbolically represented propositions. It's those entailment relations (among looser ones) we want to capture in a combinatory reasoner.

They also already introduce the notions of ***soundness and completeness***: a logical system is sound if the conclusions it allows you to derive are always entailments of the premises; and it is complete if it allows you to derive ***all*** entailments of any given premises.

A caveat:

At a couple of points, B & L imply that it is appropriate to think of the ***beliefs*** (or knowledge) of an agent, assuming that it represents propositions symbolically, as all the entailments of its explicitly represented propositions.

But this is a notion of "belief" (or knowledge) that is badly out of joint with intuitive notions of belief. Do you know the rules of chess, or Go? If so, do you also believe that the first player can, in principle, force a win? Or that the second player can force a win? Or that neither is the case? I doubt it very much -- even though one of those conclusions follows from the rules!

In general, what you can ***infer*** is not the same as what you ***know*** (or ***believe***); and it is entirely possible that a proposition that is entailed by what we know is unknown to us -- yet becomes known when the entailment is pointed out. Example: though you may know that the assassins of Lincoln and JFK were John Wilkes Booth and Harvey Lee Oswald, you probably didn't know (until now!) that their names have same number of letters -- even though this is an easy consequence of your prior knowledge!