Introduction

AI has made dramatic strides, and gained great public attention is recent years, thanks (in large part) to the capabilities of deep (multi-layer) neural nets (DNNs) applied to vast oceans of data. Yet, machines still do not actually THINK.

"Representation and Reasoning (KR&R)" (also traditionally called "Logical Foundations of AI") is primarily concerned with enabling machines to think -- the emphasis is on reasoning and planning, and how machines can internally represent the knowledge needed to automate these distinctively human activities, and to acquire such knowledge. (However, we'll not have much time for delving into knowledge acquisition -- in many respects an unsolved problem.)

Thinking is required for meaningful conversation, understanding what you're reading or are being told, planning your dinner, or a trip, or how to spend your weekend, figuring out what to do about a broken window, or about a friend that's showing signs of paranoia, and so on. Of course, our thinking can also be focused on specialized areas like writing a program, repairing a car, solving a math or logic problem, etc.

Our approach to representing and using knowledge begins with "logic". Most people associate the word "logic" with a precise syntax and totally reliable deductive reasoning methods, but while we will start with that notion, we should understand that there are really two quite separate, equally important aspects of a logical symbolism: its "aboutness" and its use.

"Aboutness" refers to the fact that symbols in logic can be viewed formally as corresponding to entities, properties, and relationships in the world; in other words, they can be viewed as being about the world. As a result, logical "sentences" (formulas) can be viewed as representing factual information, events and actions in the world (or false claims, depending on whether they are true or false). We'll come back to this below, in mentioning denotational semantics. If we choose the symbols of our logic to be close to those of ordinary language, we can also understand intuitively what our logical sentences are saying about the world. (In fact, logical languages are just a formalization of ordinary language.) This intuitiveness is important both in our efforts to impart useful knowledge to machines, and in validating ("curating") that knowledge, and any conclusions drawn by the machine, in relation to the world.

The second aspect, "use", concerns how you reason with the represented knowledge. Certainly deduction is an important mode of reasoning, but many modes of human reasoning we'd like to emulate in computers are more like "educated guesswork". Often, such guesswork is based on regularities observed in the world. As children, when we encounter some creatures referred to as "dogs", and find that they are furry, friendly, and capable of running, barking, etc., we form the idea that dogs in general have these characteristics. This leads to more or less reliable "predictions" when we encounter another dog. When we've gained some experience in what's involved in eating at a restaurant, we'll know not only what to expect but also what's appropriate to do when we, or others, go to a restaurant. Besides learning characteristics of things and behavioral patterns from experience, we also learn them from being told and by reading. These
are profoundly symbolic activities, and suggest that there's an intimate connection between our thoughts and language.

What is reasoning? It's a kind of pattern transduction. For example, suppose we've learned the following general rule: If everything with property P also has property Q, and a certain entity X has property P, then conclude that X has property Q. Then how do we apply this rule to Snoopy, given that all dogs are furry and Snoopy is a dog? First, we match the claim that all dogs are furry to the pattern that all P are Q; this instantiates P to "dog" and Q to "furry". Then we match the claim that Snoopy is a dog to the pattern that X is a P, which succeeds because P is already bound to "dog". Finally, in the transduction step, we generate the conclusion that Snoopy is furry, by instantiating the pattern saying that X has property Q, with the bindings we obtained from the "premise" matches. Other reasoning patterns are less reliable. For example, If a person X is a nonsmoker, we may well guess that a person Y whom we don't know, but who is friends with X, is also a nonsmoker. Planning, too, can be understood in terms of pattern matching/transduction processes, either instantiating familiar patterns of action for achieving given goals, or synthesizing new plans by manipulating symbolic models of how various kinds of actions change the world.

So these are the kinds of *mechanisms* we will study. But, coming back to "aboutness", we also need to understand what sorts of mechanisms lead to true, or at least plausible, conclusions from given premises, or lead to plans that actually work. We want our AI systems to be sound thinkers! Denotational semantics enables this kind of analysis. In our study of knowledge representation and reasoning, we will separately study denotational semantics -- what the symbols and expressions of a symbolic language can refer to or mean, and the reasoning methods that can be employed. We will see that reasoning methods can be justified using denotational semantics.

**Current status in mechanizing thinking**

Don't the recent triumphs of AI indicate that thinking machines are just around corner? After all, we have Siri, Alexa, Google Assistant, Cortana, Bixby, etc. to talk to and provide information, reminders, play music, etc; machine translation (MT) is steadily getting better; images are being automatically captioned; face recognition is highly developed; there's the Alpha Go/Zero victory, IBM Watson's Jeopardy victory, experimental self-driving cars, robots that can carry out simple verbal instructions, and so on.

Well, thinking machines are not just around the corner. For example, when I said to Siri, "I have a cat -- is it an animal?", Siri replied "I don't understand 'I have a cat is it an animal?' But I can search the web for it". (Even "I have a cat" produced that response.) As far as games like Go are concerned, these do indeed require thinking, but of an extremely specialized sort; human thinking is marked by breadth and versatility. IBM's Watson does have breadth, but rather than thinking, it searches terabytes of stored texts for phrases embedded in contexts resembling the question (in certain relevant ways). A good reality check is a 2020 article by Gary Marcus, “The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence”:


Also, here is a good assessment by James Vincent of a recent AI sensation (from OpenAI), GPT3 (Generative Pretrained Transformer, version 3 – you’ve probably heard of it):

https://www.theverge.com/21346343/gpt-3-explainer-openai-examples-errors-agi-potential

Another recent example, Google’s LaMDA, is capable of astonishingly human-like conversations:

but statistical mimicry is not the same as thinking, is opaque, and subject to quirky errors – no-one would trust such a system for reasoning or planning, and potential abuses are plentiful.

Current attempts to push DNNs towards thinking usually consist of setting up multiple-choice questions, to be answered from short texts. For example, the so-called Winograd Schema Challenge involves examples like the following: In the sentence variants "The vase didn't fit into the suitcase because it was too small/too big", what does "it" refer to? Or in the sentence variants "The city councillors refused the women a parade permit because they feared/advocated violence", who does "they" refer to? (Terry Winograd's own example, circa 1970!). The site http://www.thelowdownblog.com/2016/07/turing-tests-indicate-that-if-we-want.html provides a commentary.

Current DNNs do very well on multiple-choice questions, but I think such AI "challenges" are misguided. The problem is that just because such questions involve commonsense thinking when humans answer them doesn’t mean they require any thinking when tackled by DNN methods. That's because DNN methods train on many thousands of test examples (often after learning "what's the most likely next word" models of language on millions or billions of texts), and in doing so learn to combine numerous superficial "cues" (features of words and the way they are arranged) to pick out the most likely choices. This has been called the "Clever Hans" phenomenon, after the horse that seemed to be able to do simple additions by stamping its leg, when in fact it was picking up superficial bodily cues from its unsuspecting trainer. Mind you, DNNs do seem to acquire some higher-level knowledge in their higher layers, but their puzzling errors, and inability to explain their "thinking", even when trained on virtually everything "out there" on the web, indicates that they don't actually think or understand. One thing is for sure: Learning by mimicry of vast collections of texts is nothing like human learning; humans can pick up novel information and extrapolate from it based on very little input (you are doing it right now), and this ability is almost entirely lacking in deep learning systems.

While statistical and DNN-based AI has held the research community -- and the public as well as funding agencies -- in thrall for 25 years, quite a lot has been accomplished in automating thinking over the 50 years or so since the beginnings of AI. We have some pretty good ideas about how to represent knowledge and how to reason with it (in various ways), and there are many noteworthy successes and applications. These include goal-directed planning for robots, dialogue-based problem solving systems, rule-based (or "description logic"-based) systems for many applications in industry, and proofs of difficult theorems in mathematics (or about the correctness of VLSI chip designs, nuclear plant shut-down protocols, or operating systems). The main problem is still that of acquiring the vast amounts of knowledge that are needed for general commonsense reasoning (as opposed to reasoning about narrow application domains); this is the knowledge acquisition (KA) bottleneck.

The future

When will machines be able to think as well as humans? And once they reach this point, won't they quickly surpass us, bringing about "the Singularity" (or "intelligence explosion" -- I.J. Good, 1965; Vernor Vinge's essay 1993)? My view is that this could happen within 2 or three decades, if the right questions are pursued. But expert views differ greatly, as do opinions about how wonderful or disastrous this would be. See https://futureoflife.org/background/benefits-risks-of-artificial-intelligence/ and the related links at my 244/444 website, http://www.cs.rochester.edu/~schubert/444/.
My own greatest worry is that civilization will collapse before the achievement of human-level AI, as a result of climate change, population growth, pandemics, and nuclear bombs as well as new means of exterminating ourselves. Can we still save ourselves? I tend to agree with Daniel Schmachtenberger that saving ourselves will require help from AI systems that can make sense of vastly more information than humans:

https://www.youtube.com/watch?v=8XCXvzQdcug

(warning: very lengthy video, in sophisticated sociological language). I also agree with novelist Richard Powers that we are "pitched in a final footrace... between inventiveness and built-in insanity".

Course goals & logistics

The main goal in Knowledge Representation and Reasoning in AI is to learn about techniques for representing factual knowledge (typically, the sort of knowledge that we can easily express in ordinary language, though also some kinds of knowledge calling for specialized representations, such as 3-D entity structure and motion, and temporal & taxonomic relations), and using such knowledge for making inferences (deductive or uncertain), and for planning courses of action to achieve goals. Students will thereby gain an appreciation of the fundamental role that knowledge representation and reasoning (KR&R) play in intelligent cognition.

Practically speaking, students will reach the point where they can write programs for basic KR&R tasks, and make sense of the current research literature in this area; CSC 444 students will also learn about a subtopic of their choice in greater depth, by reading some recent papers or books and (perhaps) writing programs that implement some ideas they have.

The grade for the course will be based on (usually) homeworks assigned every other week, some Lisp programming assignments, a midterm and final test, and, for 444 students, their research/essay project. More details are on the tentative course schedule. For grading details, see the TA homepage for 244/444 (when those details become available).


The "Course Schedule" on the course website gives tentative lecture topics, assignments, exams, weights, office hours, TAs (Pranay Mundra, Riya Sharma, Draco Xu).

Distinction between 244/444: As noted, 444 will include some extra technical topics and assignments. Most notably: a grad essay/project that should be thought about and started soon (handout next week); Abstract, with reference list: Tue. Nov. 15/21

How to use text and supplementary readings: For any technical material you’re finding difficult, it's advantageous to read the specified (usually Brachmann & Levesque) text material to get a slightly different perspective. The Russell & Norvig book is entirely optional, but can be useful as a very comprehensive, up to date source about AI (much broader scope than this course, but not much concerned with neural nets), for those interested in the subject; it has value for other courses too (242, 247/447, 246/446).
- for a different, more chatty presentation for students who find some things in the course or in B & L (or G & N) hard to understand (but less depth)
- more thorough coverage of probabilistic inference, decision making
- a possible source of ideas for grad student essays/projects (browse and see what interests you; look at bibliographic/historical notes)

Lisp: The traditional text is Wilensky's *Common LISPcraft*, which is still a good guide to learning Lisp. Another book that students seem to like is Peter Seibel's *Practical Common Lisp*. There are also excellent online guides, which you can find on the course web pages. A standard reference manual if you know Lisp already is Steele's *Common LISP: The Language*. This is available in electronic form at http://www.cs.cmu.edu/Groups/AI/html/cltl/cltl2.html (Warning: this gives the syntactic details and meaning of the various constructs but few clues about how to use them!).

All undergrads in the class and MS students should already (or shortly) have access to the instructional network (csug) -- the lab staff are taking care of this. You can activate Steel Bank Common Lisp by typing SBCL. This is available via ssh. For CSC 444 students on the grad network, Allegro Common Lisp is activated by typing acl (well, you may have to define this as an alias (in .bashrc) for /p/lisp/acl/linux/latest/alisp).