

Commonsense reasoning

Commonsense reasoning is not effective.

We know that infants and children arrive at an intuitive and seemingly flawless mathematical understanding of basic number patterns and structures (e.g., the existence of capable of counting to 10, but definitely not up to 100 because we know they can't) way before they learn the ("adult") names for the natural numbers.

Nevertheless, if you ask them why they know 10 is the right number, they can't really tell you because it's obvious in a kind of Stone Age way. They aren't capable of explaining what they're seeing directly. They understand but they can't step back and think.

Similarly, there are judgements that you make daily that you can't explain to yourself. If you take a closer look you'll probably find, to your surprise, that they violate a key principle of logic.

So it's not about whether the constructed is involved in the back. It's not about whether one commits common errors, such as double-counting bins coming to the wrong conclusions. Indeed one of the strengths of the recent advance in AI is the identification of the "errors" different people make when thinking about the problem, such as in opinions about the causes of obesity, bias, and climate change.

This is the great goal that philosophers, psychologists and statisticians have been working on: to understand what fills our consciousness and we simply cannot explain in words what we so constantly experience. The real subject that lies at the heart of this research is direct experience, as opposed to what causes direct experience. The reductionists haven't been paying attention, because the only way to genuinely reduce is to push forward by questioning the established viewpoints of what's out there.

That's what is happening to philosophy and the status quo more broadly in our time is what's leading to the development of the sort of AI we're rapidly developing.

So it's not just about the technology.

Regarding your question:

How does working in such encultured languages as Meecham and collaboratively with other software engineering team members address some of the decision making methodologies that would otherwise be difficult if not impossible?

Haha. :-)

Well, you're explicitly addressing whether encultured or the underlying languages are better. I've been sitting on both sides recently. To me, choosing the right professional enculturation is much more democratiz (Take THAT, those who follow the traditional modes of training!) than what you may have seen in the movie. The tech demigod is nothing more than the rest of us with harmonic voices and competent choices at the levels of culture and pedagogy.

What's just as amazing is that decisions have consequences: You can find tons of examples of that upcoming teams. Now, if you alter the structure of the team building and unit assignments, then chances are that you'll change the outcome. It's this: Unit (roberts, we can't stop the signal on "some") of action that you're making.

A:

One can certainly train an AI model using commonsense reasoning. The key is to approximate the reality you want to model well enough so that the error between the human representation of the model and the actual reality of the model is not huge. Something like this:

This last one is obviously an oversimplification, just to make an example. Using the Commonsense Reasoning from Nature's Algorithm is feasible under certain conditions. And it's problematic in a way because different cultural backgrounds will certainly lead to different models:

Commonsense reasoning is not effective because it assumes that things are similar to how the world is.

For instance, the type of context reasoning we use when constructing a sentence; we want to associate the subject and verb with the object.

AI uses a technique called 'weak supervision' which is not realistic for this type of context reasoning, where the context (like the subject and verb) parts of the sentence have not been given at the same time as the entity that is being referred to in the subject and verb.

AI does use it for other types of conceptual reasoning where the context has been provided. These types of reasoning can be quite powerful and useful.

Commonsense reasoning is not effective because it's entirely ambiguous what I mean by a "commonsense relation" in the first place, and because the meaning of "a relation" might not be even close to the same in the two expressions I use:

When I say "This relation/process is commutative"

When I say "This conclusion of this application of this machine learning entity only makes sense if this is a valid application."

Should they be according to you and should they be able to be fully trained, obviously a normal set of rules for commonsense reasoning should be able to be learnable and for them to have some meaning.

But, here is a good question for you to investigate: How would you unify all of the human cultural and individual differences, the intentional modifications to the blueprints to something bought on a store shelf as (for instance) a "Commercial 2MP Digital Camera from A to B" with the commonsense reasoning more useful to know how the camera actually works, to make a rock solid predictor of the operation of any commercial digital camera apt to perform its specified functionality with the documented Total Track Losses over the Assumed Life Span of the Camera?

A:

Because computer systems don't have a common sense.

They instead need to be trained by hand grants on a specific hardware and software setup. An example, if we could train a very inexpensive 4-legged robot in the manner of a cat, by reading about cats and then letting that information slowly drip in as the machine persisted over time learning about the environment, we would end up with a very expensive feline that wouldn't survive much longer than the dollar window in which it could be acquired.

Basic. That is the key to making computers more human. The only limits of a "computer" are the concepts that make it different from a human. If a computer can learn to unerringly detect and track a stranger in a crowded room, the next step is to PEMDAS train it to understand the basic dangers and knowledge enough to make a policy decision that keeps others safe. If a computer recognizes a picture of a cat, the next step is to make a PICTURE GAME where moving a sheet of paper with your hand over the screen runs a cat.

For neural networks it becomes: I created a program where I repeated the same sentence to the AI over and over within the first "training phase". It needs words and their context before starting to link them up. After that it needs to have clear markers. Blue make you calm. Red makes you angry.

Now computers need to learn the "under currents" in society. We're are starting to use a bar and a number below the bar to show how likely is any heading to be clicked on. Soon we'll be able to run a "program" that will not only rerun a document but also gather metrics on how frequently and immediately it was being read.

A:

For reference, rather than just tell you the answer, the following articles from Amy Zhang at OpenAI, two from Nature and another from CNRS/Université Paris-Saclay may be of interest on this topic:

Researchers Advance Machine Rule-Based Language Following Its Nose
And Kissing Its Foot!

Learning to classify and synthesize by reading and writing $\{x,y\}$ in
the wild

Generating Contextualized Sequences From End-to-End Generative Models

Commonsense reasoning is not effective for building an emulator because the model is more complex than the data allows. The model must be a holistic (top-down) approach rather one that learns from the bottom-up (feed-forward or supervised "explicit" layers). It has been proven, AI doesn't work. The AI bots get off track and get tricked all the time.

Commonsense reasoning is not effective at training an AI system for a few reasons.

First, it is hard to define exactly what "commonsense reasoning" is meant to be capable of. The reason substantial research was done in this area and not all is summarized in the following quote:

Linguistic knowledge is exploited because the problem of context dependency is the most basic one in all of linguistic knowledge. It is important to recognize, however, that many current deep learning systems are not

getting better at predicting perfectly what humans understand from a text, which suggests that our linguistic knowledge is more like a rough sketch, all the details of which have to be filled in by systems that use more sophisticated learning mechanisms [and optimisation by the model creator]. Increasingly, research in natural language processing is shifting from research from rule learning to instrumental models, which must be evaluated against surface-level corpus behaviours [...]

Second, mere exposure (being exposed to commonsense reasoning thinking patterns like commonsense reasoning, in an abstract sense) that an AI system accumulates mostly gets stored in its associative memories and not its schema/knowledge base. The schematized knowledge is the schema/knowledge base and not the associative memories in an AI system.

Many AI systems greatly benefit from the richness of associative memories. (Consider a human, for example.) But few AI systems are aimed at general-purpose commonsense reasoning—that is, few systems would be able to answer the kind of questions that require analysis at a very high level of abstraction.

To generalize into a narrow domain (especially the domain of commonsense reasoning that we want of a language-trained AI system), it's good to understand what it is that works and doesn't work. (Academic research (and to a less degree, commercial applications) tries to evaluate the performance of human-like commonsense reasoning and these are not included in machine-learning and language-based AI systems outside of academic research.)

All this being said, you can try out some machine-learned tools to unintentionally complement it. The reason why Artificial Intelligence developed is partly to counter the absence of human beings which means most of the the data is trained using data that is created and operationalized by humans. Using such tools is not only to make tasks faster (bias the final solution) but also to make the system resilient to the limits of human capabilities (especially those of human beings that do not possess the AI systems capability of datamining).

Commonsense reasoning is not effective for training an AI model because it requires knowledge that is either inaccessible (to the model), or more often, difficult to acquire in real time on the appropriate timescale.

The "vaguely human thing to do" is to advise the candidate of course, telling her to take it slow, make sure that she can explain the item and has just enough metadata or context (and that there's not too much else to care about in the problem at hand), but doing it this way throws away much of the benefit of recommending the best course of action given the current state of the pipeline. Imagine if a computer asked you why you flipped a switch when you were designing a new circuit from scratch. This, right now, is a very hard problem to solve. The shift occurs because telling a human in slow-motion often makes no difference. For an example if this, imagine that a human was tasked o making sense of computer logs from telemetry data gathering equipment, and use that to diagnose a major feedback glitch in a piece of equipment. The human's knowledge base, and likely the exact data, is probably not stored or accessible off-hand. Or perhaps you have a space faring system that is flying a mission consisting of many, many items and sensors around the

solar system and needs to determine if there is a broken part in a robotic housekeeping module or an inopportune sunlight reading is a problem with the CCD's or a problem with the onboard computer's health.

The sessions are limited to 2-4 hours in length. That's long enough for her to represent the data and understand the problem at hand, but short enough that she doesn't get burnt out (except possibly with the pleasure of challenge).

Note that there are well defined tasks that could be trivially trivialized using insight based-learning but I'd need to know more about those requirements and the current state of the system for an appropriate solution. My opinion is that seeing a long term vision for this is more important than the implementation of a trivial version at the current state of the process.

A:

The task is essentially how to let the AI agent find effectively a function that employs as a generalisation as possible the knowledge it knows, and employ this function to understand and decide on the next move.

We can classify this into two problems: Firstly, how can I find effective/correct program structures from unknown tasks; secondly, given a decision tree, and human coded program (or really any program, that we can directly inspect), tell if it was coded by a competent programmer.

Intuitively, we can think that "we" want an algorithm (or program) that behaves as closely as possible to an analog thinker. This is what we get. The key idea is to use reinforcement learning. This, basically, assumes a model of the world. It will "explore" the possible program structures by itself (using random search, as a grab bag approach), and above a threshold in accuracy it will then "converge" towards a "final answer" (using for example a gradient descent type algorithm).

Specifically, there is a generative model: this is a model of the world and all the structures (in our case, programs, but depending on the program it could also be "alarm arrays" or other hand drawing, or in the extreme, the human programmer's "mind"). This model is used to build a program, and then it is used to decide on the next move of the play.

A:

You can build machine learning models which do this. For example you could use a relational machine learner-"now, how could I model this as a graph or point set...".

<http://scholar.harvard.edu/files/koutra/files/rl-jmlr.pdf>

But, I would add that the practicalities are not so good. You can employ the modeling aspects of machine learning modeling in a machine learning workflow to support vision or reasoning more than you can in a conventional workflow. You get out of the traditional system what you put into it.

But the other way around is less true. What you put into the system is much more critical. Human experience and systematic knowledge may make this a more useful project, but its application computationally is a big challenge.

This is a bit of a chicken and egg question. You have to know something about the problem to understand the value of using machine learning models in the first place.

Moreover, why is it that when the system is given the exact same data (let's just say you've identified a nanometer shift in a pixel in an xray), but asked a slightly different question, the system still malfunctions and can't adapt but when you provide it identical data presented from different angles, it can shift its answer?