Computer intelligence and formalization

Kai A. Olsen, Molde College and University of Bergen

The movie I, Robot shows how robots, masterminded by a "positronic" brain, try to take control in some futuristic society. The movie is as exciting as the short stories written by Isaac Asimov that it is based on. He started on the series as early as 1940. This was perhaps his luck. The intelligent robots make much more sense with their vague "positronic" brain than with a *computer*.

The intelligent computer was a hot topic in the sixties and seventies. After the most optimistic prophets had lived to see that their "in twenty-five years" predictions of an intelligent computer had failed to appear, the few remaining are cleverly dating their forecasts several hundred years into the future. In addition there is also a change of content in their predictions of computer intelligence. Many have now moderated their statements, now mentioning "applications that would have needed intelligence if humans were doing the job." But that is something quite different. The Vikings, for example, navigated by using the sun, moon, stars, and wind directions. Even bird flight and floating seaweed may have aided an intelligent or creative navigator. Today, we also use stars for navigating, but artificial stars, satellites, which send a radio beacon that is used by any GPS system to pinpoint the position on the surface of the earth with an accuracy of a few meters. It does a better navigating job than the Vikings just by computing a simple function, but no intelligence is needed.

Smart functions?

There are some "intelligent" functions embedded in the standard software that we use today. For example, MS Word can detect typos and replace these automatically with the right word. Excel offers an autocomplete function where it automatically will suggest the rest of the data that you are typing. Mail systems can alert you immediately when messages arrive, popping these up on your screen. Windows will tell when there are unused icons on your desktop, and help to get rid of them. It can update itself automatically; an important function in these virus times.

The early intelligence prophets would have been disappointed to see how mundane the implementation of their expectations has become, but even these smart functions do not work unfailingly. The autocorrect function in Word causes problems, especially outside the US. In Norway the preposition "i" (meaning in) is often erroneously capitalized to an "I". Newer versions of the software have a language-dependent autocorrect function, and will only perform the latter in English text. But still there are many examples where the word processing system has taken control from the author. The menu in my favourite Chinese Restaurant has all I's capitalized. Probably the system perceived the text as English.

Some days ago I was typing in student grades in Excel; C+ to the first student, and C to the next three. Looking up from my notes I see that all had gotten a C+. The autocomplete function had suggested the text C+ after each C using the enter key as a confirmation, not as a jump to the next line as I had intended.

When I use my laptop during presentations and have the desktop up on the big screen, Windows may send silly messages to the audience, for example the note on the unused icons. But these are minor compared to what may happen. I recently attended a conference together with several

hundred others, when a mail message popped up on the big screen. Luckily it was an innocent message from the speaker's wife, but we all know that it could have been much worse.

Context

The problem, of course, is context. As humans we usually have some sense of the semantics, the overall idea behind what is happening. We do not interrupt a speaker with casual remarks. We understand that someone is very busy and try not to interfere. When proofreading, our marks are based on an idea of what the writer is trying to express. But the automatic spelling checkers, autocomplete functionality and automatic suggestions all work on a lexical level, without any idea of either the overall context or the semantics.

There are ways of adding context to automatic systems. But they can be data-intensive, and not always practical. The process of maintaining the correct temperature at home during winter is a good example. An on/off button on the electric furnace is the simplest mechanism, but by adding a thermostat we can avoid letting the room get too hot. We can add a timer and operate with a lower night temperature to save energy. But this system may freeze out the guest if they stay to long on a Saturday night. We can add a sensor that keeps the furnace on if there is movement in the room, but we may then end up using a lot of energy to keep the room heated for the cat. By adding more equipment, even systems where the temperature can be remotely controlled, we may get a system that gives a good balance between comfortable temperature and acceptable energy bills, but even then we will have to provide data both for standard and exceptional situations.

Sometimes we get context for free. Years ago I had to go down the hallway to ask my US-born colleague if it was "in the West Coast", "on the West Coast" or "at the West Coast". Today I ask Google. It will give me 590,000 votes for the first, 10,200,000 for the second and 188,000 for the last. An extremely useful function, especially when we write in a second language where the idea is to follow the norm. We can also use Google itself as an example. Their scheme for presenting search results offers a practical implementation of "importance".

Computer takeover?

The drawback of letting a system use context is that the outcome gets less deterministic for the average user. We may be wondering why it is so cold in the house, forgetting that we came home earlier than expected. We should, of course, have sent a message to the heating system telling it about our change of plans, but is this really how we want to shape our future? I see a situation where the heating system is warning us that if we get home that early it will not be able to heat the house, or where we have a discussion with our automatic fridge if we can go out for dinner when it planned a home meal.

I feel we are nearly there when Windows tries to restart after an update. Yes, it is smart to activate updates, but just not now when we are trying to get everything ready for the next deadline. So, we irritably hit the "restart later" button. But our dumb assistant is back ten minutes later, with the same message. It reminds me of my kids asking for ice cream on a sunny day. However, with Windows (not with the kids) it is possible to set the repetition time, but it is rather complicated and few users know how to do this.

Can we see the "restart later" message as an indication that the computer is trying to take over? Certainly - if we are paranoid. But modern software and hardware try to take control in many cases, from requiring updates to suggesting actions. The idea is to aid the user, but often the effect is just the opposite. Even systems to prevent errors may fail if they don't handle context correctly. Yes, it was nice to get a warning that the metro ticket machine did not give change, but I became pretty frustrated when it would not sell me a ticket for my 10 € bill. I was more than

willing to waive the two euro change in order to get on the train and catch my international flight. Modern printers have similar problems. Earlier it was possible to coach a sick printer along in order to get a needed hardcopy, today a much “smarter” device refuses to go on if an error has been detected.

Formalization

Formalization is strongly related to context. We need to formalize the application in a way where we cover all circumstances. Spam filters are a good example. An often used method is to discriminate spam from genuine messages by looking at the words in the message, i.e., by differentiation on a lexical level. This simple formalization works in most cases, but the difference between a semantic and lexical level causes the filters to let some spam through. On the other hand they also remove some genuine messages. There are efforts to use similar methods in discriminating pornographic content both in text or pictures, but here the formalization problems are even greater.

Those of us who have followed the Great Robot Race have been impressed to see how cars can be steered by a computer through the Mojave Desert, using GPS, 3-D mapping systems, laser and video cameras. They follow dirt roads, avoid obstacles, go through tunnels, turn round narrow corners and pass other cars as if they were controlled by a human driver. Does this imply that we have automatic cars on the road in a few years? The clue here is not how much intelligence we can build into the cars, but how we can formalize driving. While driving can be very relaxing at times, allowing us to listen to music, look at the countryside, and have a conversation, there are situations when we use all the sensory organs and all our brain capacity to make the right decisions. We may unexpectedly see something in the road ahead. Do we have to brake, veer, or can we just go straight ahead? If a robot has to take these decisions the tasks must be formalized. The decision program will have to distinguish between a rock (brake or veer), a small snowdrift (go ahead), an empty cardboard box (go ahead?) or a full box (brake or veer). It must be able to interpret the intentions of fellow drivers (is he really going to turn, why is she slowing down, does he want me to pass?). Clearly, the other alternative—to formalize the road to a higher level, seems to be more promising. This can be done by putting “virtual rails” in the road, cables that gadgets in the cars can follow or by using detailed GPS waypoints to determine the route as in the Great Robot Race. Laser-guided vision, sensors and video cameras can be used to control the distance to nearby cars. Such a system may reduce the driver’s workload, but exceptions would still be a problem for an automatic system.

The examples we have today of computer controlled trains illustrate the formalization issue. These run in an environment where exceptions are reduced to a minimum, for example by having systems where it is not possible for people to get access to the tracks. We should be able to find similar applications for robot trucks where these can run on enclosed (company) roads. This should be an interesting challenge for the near future.

Computer intelligence, a matter of formalization

In the end we see that computer “intelligence” is really a matter of formalization, i.e., how we can formalize tasks and context. GPS navigation, the telephone, IP-addressing, credit card payment, account numbers and bar codes are some success stories. Robots already perform many tasks in industry, and are driving forklifts, trains and experimental trucks. Automatic vacuum cleaners and lawn movers have been on the market for several years. Robots have an application wherever physical output is needed. They may be given “intelligence” in the form of being able to perform choices within limited worlds, where the context is formalized and well-defined.

The Turing test is the ultimate challenge for intelligent systems. Is it possible to make a computer system that can answer like a human being, so well that the user at the terminal cannot

make the distinction? Is this a test about thinking and intelligence? No, it is about formalization. Type in some stories and let the machine and the human classify these as dull, sad or humorous. We have an intelligent machine the day the robot can laugh at the right places.